Pattern Recognition Letters
journal homepage: www.elsevier.com

Bi-path Combination YOLO for Real-time Few-shot Object Detection

Ruiyang Xia[a,b], Guoquan Li[a,**], Zhengwen Huang[c], Hongying Meng[c], Yu Pang[d]

[a] *School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China*
[b] *Group of Artificial Intelligence and System Optimization, BUL-CQUPT Innovation Insitute, Chongqing, 400065, China*
[c] *Department of Electronic and Electrical Engineering, Brunel University London, London, UB8-3PH, United Kingdom*
[d] *Key Laboratory of Photoelectric Information Sensing and Transmission Technology, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China*

Article history:

keywords :
Few-shot object detection
Transfer learning
Real-time
Bi-path Combination You Only Look Once
Attentive DropBlock

ABSTRACT

Few-shot object detection (FSOD) has more attention in recent years as the quantitative limitation of instances during the model training. Previous works based on meta-learning and transfer learning focus on the detection precision but ignore the inferring speed, which is difficult to apply in amounts of applications. In this letter, to keep a high inferring speed and a comparable detection precision, we propose a real-time detector entitled Bi-path Combination You Only Look Once (BC-YOLO) for FSOD. BC-YOLO can be categorized as a transfer learning based one-stage object detector with a two-phase training scheme. It is particularly composed of bi-path parallel detection branches which detect base and novel class objects respectively and commonly detect objects with a discriminator in the inferring stage. Moreover, to elevate the model generalization trained from few-shot objects, we further propose an Attentive DropBlock algorithm to make the detector focus on the entire details of objects instead of the local discriminative regions. Extensive experiments on PASCAL VOC 2007 and MS COCO 2014 datasets demonstrate that our method can achieve a better tradeoff between speed and precision than state-of-the-art methods.

1. Introduction

Object detection is one of the most important tasks in computer vision. There are many detectors proposed based on convolutional neural network (CNN) [1, 2, 3, 4, 5] or vision Transformer [6, 7, 8, 9, 10] with high detection performance. However, the community of these models is that the performance is achieved at the cost of massive data. When the number of data is limited, the detection precision will drop rapidly as the complexities of objects and the enormousness of model parameters. Therefore, few-shot object detection (FSOD) is received more attention in recent years.

To better adapt the quantitative limitation of instances, there are currently few-shot object detectors based on two types of mainstream thinking, i.e., meta-learning and transfer learning. For the meta-learning based methods [11, 12, 13, 14, 15], the aim is to build the feature relevance between the query image and the few support samples. Although the detection performance gets improved, the computational complexity also increases severely as the feature extractor in the few-shot branch, the relation builder between input features and few support features, and the number of object categories. For the transfer learning based approaches [16, 17, 18, 19], the goal is to make the detector that has already possessed the ability of feature representation adapt in few-shot objects well. However, to elevate the detection precision, most methods focus on the two-stage detectors such as Faster-RCNN [3] or Mask-RCNN [4], which is cumbersome during the inferring stage as the input images should be large and the proposals should be generated in Region Proposal Network (RPN).

**Corresponding author
e-mail: S190101135@stu.cqupt.edu.cn (Ruiyang Xia), ligq@cqupt.edu.cn (Guoquan Li), Zhengwen.Huang@brunel.ac.uk (Zhengwen Huang), hongying.meng@brunel.ac.uk (Hongying Meng), pangyu@cqupt.edu.cn (Yu Pang)

In this letter, to achieve a fast inferring speed for FSOD with a comparable detection precision, we propose a real-time detector called Bi-path Combination You Only Look Once (BC-YOLO). BC-YOLO is a transfer learning based model which consists of backbone, detection neck, and bi-path parallel branches in detection heads to concentrate on base and novel class objects, respectively. During the inferring stage, two branches will commonly detect objects and output the bounding boxes after going through a discriminator. In addition, to circumvent the model overfitting and enhance the generalization trained from few-shot objects. We hence propose an Attentive DropBlock algorithm to guide the model to focus on the entire object semantic features by masking the local discriminative regions with higher probability. To our best knowledge, we are the first to focus on real-time FSOD and achieve a better tradeoff between speed and precision at the same time.

Our contributions can be summarized as three-folds:

- We propose a real-time detector called BC-YOLO based on transfer learning with a two-phase training scheme to elevate the detection efficiency for FSOD. It owns two parallel detection branches for the sake of detecting base and novel class objects and commonly detecting objects with a discriminator in the inferring stage.

- We propose an Attentive DropBlock algorithm to decrease the influence of local discriminative regions and guide the model to concentrate on the entire object semantic features during the few-shot tuning to increase the model generalization.

- We carry out experiments on PASCAL VOC 2007 [20] and MS COCO 2014 [21] datasets to demonstrate the effectiveness of our method. Extensive experimental results indicate that our proposed detector can achieve a better tradeoff between speed and precision than state-of-the-art methods.

2. Related works

2.1. Few-shot Learning

To precisely classify unseen categories with limited quantities. Two mainstreams can be concluded as metric learning based [22, 23, 24] and meta-learning based approaches [25, 26]. Metric learning focuses on building strong feature embeddings to close and enlarge feature vectors with the same and different classes, respectively. There are various metric loss functions used to distinguish feature vectors such as cosine loss [22] and triplet loss [26]. Meta-learning endows the meta-model a strong knowledge representation and can make the model quickly adapt into few-shot samples [25, 26].

2.2. Few-shot Object Detection

Motivated from the effectiveness in image classification by meta-learning based approaches. Some meta-detectors are proposed and achieve good detection performance [11, 12, 13, 14, 15]. For example, FSRW [11] is proposed to extract a few sample features and reweight to query features in channel dimension. Meta-RCNN [13] follows this insight but focuses on Region of Interests (RoIs). However, two parallel backbones located in few-shot and base samples lead to high computational complexity. Besides, the computational complexity is also a positive correlation to the number of categories and relation builder between query and few-shot samples, which means training the meta-detector will be hard if there are too many categories or the relation builder is complex. To make the training simpler and more efficient, there are some few-shot object detectors based on transfer learning [17, 18, 19]. For instance, the work in [18] proposed TFA by adopting a two-phase training scheme based on transfer learning. CoRPN [19] follows this strategy and builds multiple RPNs to build proposals more precisely. Nevertheless, these two-stage few-shot object detectors are hard to achieve fast inferring speed which impedes amounts of engineering applications such as autonomous driving.

2.3. Regulization

To elevate the model generalization, there are some regulization methods proposed to randomly drop the feature units. Specifically, DropOut [27] randomly drops features from arbitrary dimensions. DropConnect [28] then convert to drop network weights before computing with extracted features. Spatial DropOut [29] randomly drops features in a specific dimension and DropBlock [30] drops a square region features. Except to feature, input and network module can also be considered to drop by adopting CutOut [31] and Stochastic Depth [32], respectively.

3. Proposed method

In this section, we describe BC-YOLO from the model architecture, training scheme, and detection process in inferring stage, respectively. Then, the Attentive DropBlock algorithm is detailed in the second part.

3.1. Overview

The architecture of BC-YOLO is shown in Fig. 1. The main components include CNN backbone, Feature Pyramid Network (FPN) [33], and bi-path detection branches with a discriminator, that focus on extracting image features, providing semantic features with different scales, and detecting base and novel class objects, respectively. The main reasons for utilizing bi-path detection branches are to avoid the model degradation that appeared in detecting the base class objects when the model is trained on few-shot objects [34]. In addition, according to the knowledge distillation [35], the novel detection branches Det_n can be generalized better by learning from the strong base detection branches Det_b. After extracting the high semantic features in the backbone, Spatial Pyramid Pooling (SPP) layer [36] is then equipped behind the last layer of the backbone to further enlarge the receptive fields of these features.

A two-phase training scheme is needed during training for BC-YOLO. The first phase called base training is trained for base class objects C_b with abundant data D_b offering and the

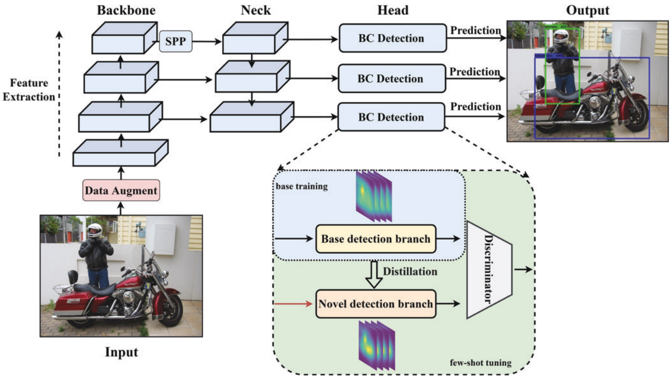


Fig. 1. The architecture of the proposed BC-YOLO detector. The red line indicates the place we use the Attentive DropBlock algorithm. Bounding boxes about the novel and base class objects are represented in terms of blue and green color, respectively.

second phase called few-shot tuning is trained for novel class objects C_n with limited data K for each class.

Firstly, the whole network is trained except Det_n such that the backbone and detection neck can own the strong knowledge representation [18] in the base training. The loss function trained on D_b is,

$$L_{\text{base training}} = L_{box} + L_{cls} + L_{obj} \tag{1}$$

Where L_{box} is the combination of GIoU loss [37] and smooth L1 loss [3] for coordinate regression. L_{cls} and L_{obj} are focal loss [38] and binary cross-entropy loss functions, respectively.

Secondly, in the few-shot tuning stage, the backbone, detection neck, and Det_b are frozen to keep the strong generalization. Det_n and the SPP layer is trained for novel class objects in the few-shot tuning. However, with our trials, we find the detection precision is low when only the novel class objects are adopted. The reason might be that the similarity existed in base and novel classes so that Det_n generates many false positive bounding boxes. We hence increase the categories of the object as $C_b \cup C_n$ by randomly taking K instances from D_b for each base class. In addition, with the consideration that Det_b owns strong generalization trained from D_b, Det_n should learn the soft weights from Det_b to get better generalization. We hence build a base distillation loss L_b between Det_b and Det_n branches computed as follow:

$$L_b = \frac{1}{N}\left(\sum_{i=1}^{N} l\left(O_{b,i}^{cls_{base}}, O_{n,i}^{cls_{base}}\right)\right) \tag{2}$$

Where N donates the batch size. l is the sum of absolute error function. $O_{b,i}^{cls_{base}}$ and $O_{n,i}^{cls_{base}}$ indicate the base classification scores from the Det_b output O_b and the Det_n output O_n for i-th image, respectively. Therefore, the loss function trained on few-shot objects can be summarized as:

$$L_{\text{few-shot tuning}} = L_{box} + 2L_{cls} + L_{obj} + \lambda \cdot L_b \tag{3}$$

Where λ is a weight that controls the influence of base distillation learning and we set this weight as 0.1 in default.

Algorithm 1 Attentive DropBlock

Input: Feature map F; parameter $keep\_prob$; parameter $block\_size$ and model state $mode$

Output: Feature map F'

```
if mode == Inference then
    F' = F
else
    Compute f_C by applying global max pooling function in each channel dimension
    Compute f_S by applying global average pooling function in each spatial dimension
    Compute γ: γ = (1−keep_prob)/(block_size^2) · (σ(f_C)×σ(f_S))/α
    Build mask M: M_{i,j} ∼ Bernoulli (γ)
    Each zero position in M is set as the center for a square zero mask with the length equals block_size
    Compute F': F' = F ∗ A
    Normalize F': F' = F' ∗ count(M)/sum(M)
end if
return F'
```

In the inferring stage, Det_b and Det_n jointly detect objects. However, resolving O_b added with O_n severely prolongs the inferring process. We hence incorporate a discriminator behind these two branches to choose the most probable one. Specifically, the discriminators only output one of O_b and O_n by comparing their combination result $R_b = max\left(O_b^{cls}\right)*O_b^{obj}$ from Det_b and $R_n = max\left(O_n^{cls}\right) * O_n^{obj}$ from Det_n, where $max\left(O^{cls}\right)$ and O^{obj} respectively denote the maximum of classification scores and object confidence from the output as follow:

$$O_d(i,j) = \begin{cases} O_b(i,j) & \text{if } R_b(i,j) \geq R_n(i,j), \\ O_n(i,j) & \text{otherwise.} \end{cases} \tag{4}$$

Where $O_d(i,j)$ denotes the discriminator output for a specific spatial gird (i,j).

3.2. Attentive DropBlock

To further elevate the model generalization during the few-shot tunning, we propose an Attentive DropBlock algorithm which is influenced not only by the parameters of $keep\_prob$ and $block\_size$, but also the object semantic features. Specifically, the DropBlock [39] algorithm which sets a constant coefficient for all positions within a feature map as follow:

$$\gamma = \frac{1-keep\_prob}{block\_size^2} \cdot \frac{feat\_size^2}{(feat\_size - block\_size + 1)^2} \tag{5}$$

Where $keep\_prob$ and $block\_size$ are the hyparameters that influence the frequency and the square size of dropping. Different from the original DropBlock, γ is a dynamic coefficient which is also dependent on the extracted feature map in Attentive DropBlock algorithm.

To be specific, given a feature map $F \in \mathbb{R}^{B\times C\times H\times W}$, we compute $f_C \in \mathbb{R}^{B\times C\times 1\times 1}$ by applying the global max pooling function in each channel dimension and $f_S \in \mathbb{R}^{B\times 1\times H\times W}$ by applying

Table 1. Few-shot object detection results on different datasets

		PASCAL VOC						MS COCO	
		Novel Set 1		Novel Set 2		Novel Set 3		Novel Set	
Method	Backbone	5	10	5	10	5	10	10 / FPS	30
LSTD [16]	Darknet-19	29.1	38.5	15.7	31.0	27.3	36.3	3.2 / -	6.7
YOLO-ft-full [11]	Darknet-19	24.8	38.6	16.1	33.9	32.2	38.4	3.1 / -	7.7
FsDetView [17]	ResNet-101	36.1	42.3	22.6	29.2	33.2	39.8	7.6 / -	12.0
FRCN-ft-full [13]	ResNet-101	41.5	45.6	31.6	39.1	35.0	45.1	6.5 / -	11.1
RepMet [15]	ResNet-101	38.6	41.3	28.3	35.8	34.3	37.2	- / -	-
FSRW [11]	Darknet-19	33.9	47.2	30.1	39.2	40.6	41.3	5.6 / -	9.1
NP-RepMet [14]	ResNet-101	47.3	49.4	**43.4**	**49.1**	41.5	44.8	- / -	-
CoRPNs w/cos [19]	ResNet-101	54.1	55.7	36.2	41.3	51.6	49.6	- / -	-
MetaDet[12]	VGG-16	36.8	49.6	31.7	43.0	43.9	44.1	7.1 / -	11.3
Meta R-CNN [13]	ResNet-101	45.7	51.5	34.8	45.4	41.2	48.1	8.7 / 11.7	12.4
TFA w/cos [18]	ResNet-101	**55.7**	56.0	35.1	39.1	**49.5**	**49.8**	**10.0** / -	**13.7**
BC-YOLO	Darknet-53	47.3	55.1	37.7	41.5	40.9	45.4	8.2 / **43.6**	12.0
BC-YOLO*	Darknet-53	50.4	**57.6**	38.9	43.3	42.5	49.1	9.0 / 5.8	12.9

* donates the results with multi-scale testing

the global average pooling function in each spatial information. Then the dynamic $\gamma \in \mathbb{R}^{B\times C\times H\times W}$ can be calculated as follow:

$$\gamma = \frac{1 - keep\_prob}{block\_size^2} \cdot \frac{\sigma(f_C) \times \sigma(f_S)}{\alpha} \quad (6)$$

Where σ is the sigmoid function that controls the weight scale of attention information and α is the amplification factor. Algorithm 1 describes how Attention DropBlock works for a given feature map. Note that the *count* and *sum* indicate the number and the sum of elements, respectively.

Finally, the notations utilized in our method are summarized in Appendix A.5

4. Experiments

We firstly introduce the implementation details of BC-YOLO and Attentive DropBlock. Then we compare our approach with other state-of-the-art methods on PASCAL VOC 2007 [20] and MS COCO 2014 [21] datasets, respectively. The ablation studies and qualitative results on PASCAL VOC 2007 are presented later.

4.1. Implementation Details

To increase the data diversity indirectly, we use Mixup [40] with random affine transformation and multi-scale strategy with label smoothing [41] to augment the limited instances. The optimizer used is SGD with the weight decay and the momentum set as 0.0005 and 0.9, respectively. The cosine learning rate schedule [42] from 0.001 to 0.00001 in base training and few-shot tunning for 300 epochs. BC-YOLO is trained over 4 GPUs with 64 images per batch size. Moreover, the *keep_prob*, *block_size* and α set in Attentive DropBlock as 0.9, 7 and 0.1, respectively.

4.2. Comparison With State-of-the-Art

To ensure the fairness of comparison, the data and class splits adopted are the same as the settings from previous works [11, 12, 13, 14, 16, 17, 18, 19], i.e., the overall categories in PASCAL VOC are divided into 15 base and 5 novel classes with three different splits. For MS COCO, all 20 categories in PASCAL VOC can be seen as novel classes and the rest of 60 categories are base classes. We report 5, 10-shot results on PASCAL VOC and 10, 30-shot results on MS COCO as the extremely few-shot objects lead to the large variances that exist in detection results. Moreover, we report the mean Average Precision (mAP) on MS COCO and mean Average Precision with 0.5 IoU threshold on PASCAL VOC (mAP@50), respectively. Table 1 shows the detection results compared with other state-of-the-art methods.

Note that the 10-shot results in the column of MS COCO respectively represent the mAP (left) and FPS (right). It can be observed that BC-YOLO outperforms some of the state-of-the-art methods on mAP and mAP@50. More importantly, our model uses a relatively small backbone (Darknet-53 vs ResNet-101) and achieves real-time FSOD (43.6 FPS) on 10-shot MS COCO novel set, which is nearly 4 times faster than Meta-RCNN with only 0.5 and 0.4 mAP gap for 10-shot and 30-shot MS COCO, respectively. After adopting multi-scale testing strategy, all detection results can be improved and even surpass the state-of-the-art precision. Therefore, these results plausibly demonstrate that BC-YOLO can have a better tradeoff between speed and precision.

4.3. Ablation Studies and Qualitative Results

In this part, we analyze the effectiveness of each component in our model on the PASCAL VOC dataset. Experimental results are shown in Table 2. nAP, bAP, and aAP donate the mAP@50 on novel class, base class, and all class objects, respectively. It can be apparently observed that each component

Table 2. Ablation studies from 10-shot PASCAL VOC novel set 3

Two-phase Training Scheme	Bi-path Combination	Base Distillation Loss	Attentive DropBlock	nAP	bAP	aAP	Param (M)
				32.7	48.3	45.7	64.3
✓				41.9	69.8	62.9	**64.3**
✓	✓			44.3	71.3	64.6	70.5
✓	✓	✓		44.7	72.5	65.6	70.5
✓	✓	✓	✓	**45.4**	**72.5**	**65.7**	70.5

(a) Novel class object detection (b) Base class object detection

Fig. 2. Qualitative results on PASCAL VOC dataset. The top and bottom row is the detection results of YOLO* and BC-YOLO without L_b and Attentive DropBlock, respectively.

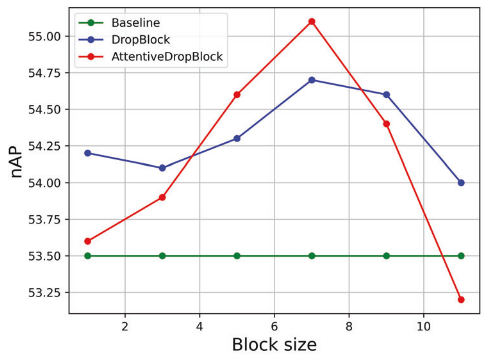
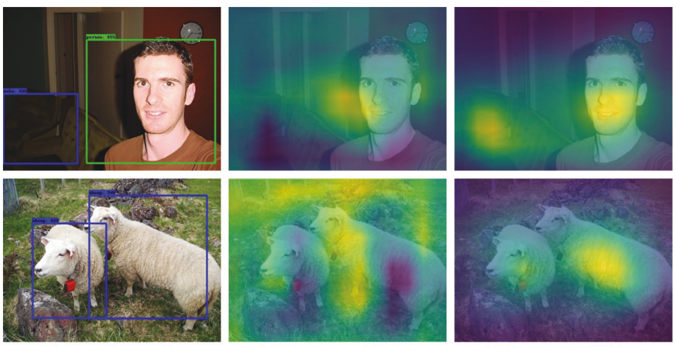


Fig. 3. Detection results from 10-shot PASCAL VOC novel set 1 for BC-YOLO, *keep_prob* we set is 0.9 for DropBlock and Attentive DropBlcok.



(a) Detection results (b) Responses of Det_b (c) Responses of Det_n

Fig. 4. Object responses from 10-shot PASCAL VOC novel set 3 for Det_b and Det_n, respectively.

can bring gains to different extents. Specifically, after adopting our two-phase training scheme, the nAP can significantly be elevated by 9.2% (41.9% vs 32.7%).

Then we incorporate the bi-path parallel detection branches with a discriminator into the model, it promotes the nAP and bAP about 2.4% and 1.4%. To further demonstrate their strength, qualitative results are shown in Fig 2. It can be seen that, for novel class objects, BC-YOLO can figure out them even are close to the base class object, own special gesture, or small scale with occlusion. For base class objects, thanks to the strong Det_b and the discriminator, the base class objects can also be detected without forgetting in the inferring stage.

The base distillation loss can bring 1.2% bAP and 0.4% nAP increase, respectively. We conjecture it because the generalization learned from Det_b can effectively influence Det_n and make our model better distinguish objects whether they belong to base or novel categories.

Table 3. Detection results from 10-shot PASCAL VOC novel set 2 for different factors in discriminator.

Factor	nAP	bAP	aAP
O^{obj}	41.1	70.2	62.9
$max(O^{cls})$	40.3	70.4	62.9
$max(O^{cls}) * O^{obj}$	**41.5**	**70.8**	**63.5**

Attentive DropBlock is also beneficial to the model generalization, which promotes the model 0.7% nAP. To further demonstrate its effectiveness, we compare it with the original DropBlock [39]. The results are shown in Fig. 4. It can be noticed that the curve of Attentive DropBlock is more dynamic than the DropBlock one as the former algorithm pays more attention to the object. Attentive DropBlock can get better nAP when the *block_size* equals 5 and 7 than DropBlock, which means it is the significance of considering the object semantic features.

To further explore the influence of the components in our model, a suitable discriminator we found is helpful to BC-YOLO. Table 3 shows the results by considering different factors. It is interesting to observe from the results that the maximum classification score and the object confidence are suitable determinations for base and novel class objects, respectively. However, the best results generated from the combination of object confidence and classification score indicate its superiority.

Table 4. Detection results from 10-shot PASCAL VOC novel set 3 for different detection branches

Detection branch	nAP	bAP	aAP
Det_b	0.0	70.0	52.5
Det_n	45.1	67.6	62.0
$Det_n + Det_b$	**45.4**	**72.5**	**65.7**

A potential problem is whether the Det_b can identify novel class objects during the base training as the input images also exist these objects except their ground-truth information. Therefore, to clearly observe the responses of Det_b and Det_n, Table 4 respectively shows the results of different detection branches. To the Det_b, it can be noticed that they are by no means interested in novel class objects but generate high responses in base class objects. However, the Det_n generates responses for all objects. After combining them in the discriminator, nAP, bAP, and aAP can get the best results. Lastly, Fig. 3 further illustrates the responses of the Det_b and Det_n, respectively.

5. Conclusion

To achieve real-time FSOD with comparable detection precision, we proposed BC-YOLO detector and an Attentive DropBlock algorithm. BC-YOLO has bi-path parallel detection branches which respectively focus on base and novel class objects and commonly detect objects with a discriminator in inferring stage. Attentive DropBlock can further elevate the model generalization by masking local discriminative regions with a higher probability. Extensive experiments on PASCAL VOC 2007 and MS COCO 2014 datasets demonstrate that our model can achieve a better tradeoff between speed and precision than state-of-the-art methods.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work presented in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61971079); the Brunel University London BREIF Award (No. 11937115); the National Key Research and Development Program of China (No. 2019YFC1511300); the Basic Research and Frontier Exploration Project of Chongqing (No. cstc2019jcyj-msxmX0666) and the Innovative Group Project of the National Natural Science Foundation of Chongqing (No. cstc2020jcyj-cxttX0002).

Appendix A. Notations

Table A.5. Lookup table for notations in the paper

Notation	Description
Det_b	detection branches of base class objects
Det_n	detection branches of novel class objects
C_b	list of base class
C_n	list of novel class
D_b	dataset of base class
D_n	dataset of novel class
O_b	output of base class detection branches
O_n	output of novel class detection branches
O_d	output of discriminators
R_b	combination results of base class objects
R_n	combination results of novel class objects
K	number of few-shot instances
l	sum of absolute error function
f_C	max spatial feature in each channel dimension
f_S	average channel feature in each spatial dimension
λ	weight of base distillation loss
α	amplification factor of attention information
γ	coefficient of controling the dropping unit
σ	sigmoid function

References

[1] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767 (2018).
[2] S. Chen, Z. Cheng, L. Zhang, Y. Zheng, Snipedet: Attention-guided pyramidal prediction kernels for generic object detection, Pattern Recognition Letters (2021).
[3] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, IEEE transactions on pattern analysis and machine intelligence 39 (6) (2016) 1137–1149.
[4] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.
[5] H. Peng, S. Chen, Bdnn: Binary convolution neural networks for fast object detection, Pattern Recognition Letters 125 (2019) 91–97.
[6] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision, Springer, 2020, pp. 213–229.
[7] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable detr: Deformable transformers for end-to-end object detection, arXiv preprint arXiv:2010.04159 (2020).
[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, arXiv preprint arXiv:2103.14030 (2021).

[9] Y. Fang, B. Liao, X. Wang, J. Fang, J. Qi, R. Wu, J. Niu, W. Liu, You only look at one sequence: Rethinking transformer in vision through object detection, arXiv preprint arXiv:2106.00666 (2021).

[10] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable detr: Deformable transformers for end-to-end object detection, arXiv preprint arXiv:2010.04159 (2020).

[11] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, T. Darrell, Few-shot object detection via feature reweighting, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 8420–8429.

[12] Y.-X. Wang, D. Ramanan, M. Hebert, Meta-learning to detect rare objects, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9925–9934.

[13] X. Yan, Z. Chen, A. Xu, X. Wang, X. Liang, L. Lin, Meta r-cnn: Towards general solver for instance-level low-shot learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9577–9586.

[14] Y. Yang, F. Wei, M. Shi, G. Li, Restoring negative information in few-shot object detection, arXiv preprint arXiv:2010.11714 (2020).

[15] L. Karlinsky, J. Shtok, S. Harary, E. Schwartz, A. Aides, R. Feris, R. Giryes, A. M. Bronstein, Repmet: Representative-based metric learning for classification and few-shot object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5197–5206.

[16] H. Chen, Y. Wang, G. Wang, Y. Qiao, Lstd: A low-shot transfer detector for object detection, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 32, 2018.

[17] Y. Xiao, R. Marlet, Few-shot object detection and viewpoint estimation for objects in the wild, in: European Conference on Computer Vision, Springer, 2020, pp. 192–210.

[18] X. Wang, T. E. Huang, T. Darrell, J. E. Gonzalez, F. Yu, Frustratingly simple few-shot object detection, arXiv preprint arXiv:2003.06957 (2020).

[19] W. Zhang, Y.-X. Wang, D. A. Forsyth, Cooperating rpn's improve few-shot object detection, arXiv preprint arXiv:2011.10142 (2020).

[20] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International journal of computer vision 88 (2) (2010) 303–338.

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.

[22] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, W. Liu, Cosface: Large margin cosine loss for deep face recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5265–5274.

[23] J. Snell, K. Swersky, R. S. Zemel, Prototypical networks for few-shot learning, arXiv preprint arXiv:1703.05175 (2017).

[24] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.

[25] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, R. Hadsell, Meta-learning with latent embedding optimization, arXiv preprint arXiv:1807.05960 (2018).

[26] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1126–1135.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (1) (2014) 1929–1958.

[28] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, R. Fergus, Regularization of neural networks using dropconnect, in: International conference on machine learning, PMLR, 2013, pp. 1058–1066.

[29] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, C. Bregler, Efficient object localization using convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 648–656.

[30] G. Ghiasi, T.-Y. Lin, Q. V. Le, Dropblock: A regularization method for convolutional networks, arXiv preprint arXiv:1810.12890 (2018).

[31] T. DeVries, G. W. Taylor, Improved regularization of convolutional neural networks with cutout, arXiv preprint arXiv:1708.04552 (2017).

[32] G. Huang, Y. Sun, Z. Liu, D. Sedra, K. Q. Weinberger, Deep networks with stochastic depth, in: European conference on computer vision, Springer, 2016, pp. 646–661.

[33] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.

[34] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, Advances in neural information processing systems 30 (2017) 6467–6476.

[35] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531 (2015).

[36] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE transactions on pattern analysis and machine intelligence 37 (9) (2015) 1904–1916.

[37] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 658–666.

[38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[39] G. Ghiasi, T.-Y. Lin, Q. V. Le, Dropblock: A regularization method for convolutional networks, arXiv preprint arXiv:1810.12890 (2018).

[40] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, arXiv preprint arXiv:1710.09412 (2017).

[41] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[42] I. Loshchilov, F. Hutter, Sgdr: Stochastic gradient descent with warm restarts, arXiv preprint arXiv:1608.03983 (2016).