Hierarchical Shot Detector

Jiale Cao[1], Yanwei Pang[1]*, Jungong Han[2], Xuelong Li[3]
[1]Tianjin University [2]University of Warwick [3]Northwestern Polytechnical University
connor@tju.edu.cn, pyw@tju.edu.cn, jungong.han@warwick.ac.uk, li@nwpu.edu.cn

Abstract

Single shot detector simultaneously predicts object categories and regression offsets of the default boxes. Despite of high efficiency, this structure has some inappropriate designs: (1) The classification result of the default box is improperly assigned to that of the regressed box during inference, (2) Only regression once is not good enough for accurate object detection. To solve the first problem, a novel reg-offset-cls (ROC) module is proposed. It contains three hierarchical steps: box regression, the feature sampling location predication, and the regressed box classification with the features of offset locations. To further solve the second problem, a hierarchical shot detector (HSD) is proposed, which stacks two ROC modules and one feature enhanced module. The second ROC treats the regressed boxes and the feature sampling locations of features in the first ROC as the inputs. Meanwhile, the feature enhanced module injected between two ROCs aims to extract the local and non-local context. Experiments on the MS COCO and PASCAL VOC datasets demonstrate the superiority of proposed HSD. Without the bells or whistles, HSD outperforms all one-stage methods at real-time speed.

1. Introduction

Object detection based on deep convolutional neural network can be mainly divided into two classes: two-stage methods [15, 40, 16] and one-stage methods [39, 31, 28]. Two-stage methods firstly extract some candidate object proposals and then classify and regress these proposals. One-stage methods directly predict object categories and regression offsets of dense default boxes (anchors). Compared with two-stage methods, one-stage methods are more efficient but inferiorly accurate.

In one-stage methods, class imbalance problem between the positive and negative samples is considered as a main challenge, which has been solved by OHEM [42] and focal loss [28] in some degree. Besides class imbalance problem, there are two inappropriate designs which can be further improved: (1) Most one-stage methods simultaneously conduct the classification and regression of default box during training. As a result, the classification result of default box is improperly assigned to that of the regressed box during inference. In fact, we want to output the true classification result of the regressed box. Meanwhile, training the classification task with the default boxes ignores some accurately regressed boxes, which are helpful for object detection. (2) Only one regression of the default box is not good enough to accurately detect the object. Recently, cascade structure has been proposed for accurate detection in both two-stage methods and one-stage methods [54, 2]. However, each stage in the cascade structure still suffers from the above problem of classification inconsistency.

To better solve the above problems, a novel hierarchical shot detector (HSD) is proposed in Fig. 1 to hierarchically conduct regression and classification. In HSD, the key and novel structure is the *reg-offset-cls* (ROC) module. Instead of simultaneous classification and regression, ROC firstly

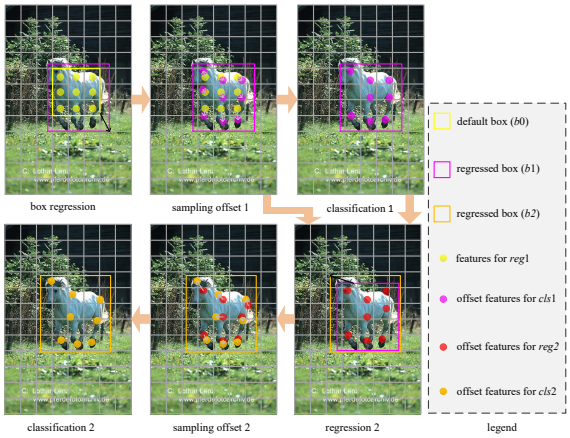


Figure 1. The detection pipeline. HSD consists of two stacked *reg-offset-cls* (ROC) modules. Each ROC firstly regresses the box, secondly calculates the feature sampling offset based on the box offset, and finally classifies the regressed box by the features of offset sampling locations. The feature sampling locations for the first classification and second regression are the same. To simplify, the feature enhanced module between two ROCs is not shown.

*Yanwei Pang is the corresponding author. Code will be available at https://github.com/JialeCao001/HSD.

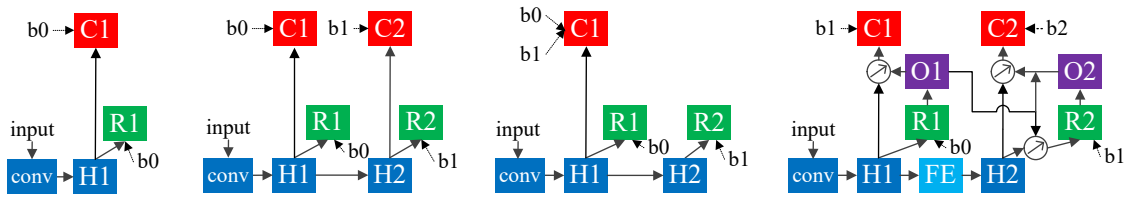


Figure 2. The architectures of some one-stage methods. 'conv' means the backbone network. 'H' is the convolution head. 'C' is the predication of classification branch. 'R' is the predication of regression branch. 'b' is the box ($b0 \mapsto b1 \mapsto b2$), where $b0$ is the default box, and $b1$ and $b2$ are the regressed boxes. 'O' in (d) is the convolutional module to calculate feature sampling offset by the box regression output. '$\nearrow$' in (d) means the convolutional operation for classification or regression by considering the feature sampling offset.

predicts regression offsets of the default box ($b0$), then generates the feature sampling offsets, and finally classifies the regressed box ($b1$) by the features of offset locations. Based on ROC module, the features of accurate sampling locations for classification and the more accurately regressed boxes can be used to boost performance. To further improve detection, HSD hierarchically stacks two ROC modules. The regressed boxes ($b1$) and the feature sampling offsets generated by the first ROC are used as the inputs. Meanwhile, to enhance the feature discrimination, a feature enhanced (FE) module is injected before the second ROC to exploit more local and non-local contextual information. The main contributions of this paper can be summarized as follows:

(1) A novel *reg-offset-cls* (ROC) module is proposed. It reconstructs simultaneous classification and regression by three hierarchical steps: the default box regression, the feature sampling offset predication, and the regressed box classification with offset features. Moreover, it is light-weight.

(2) Based on the proposed ROC module, the hierarchical shot detector (HSD) is proposed, which hierarchically stacks two ROC modules and one feature enhanced (FE) module. The proposed HSD can be seen as a generalization of one-stage methods for accurate object detection.

(3) Experimental results on the MS COCO [29] and PASCAL VOC [11] demonstrate the effectiveness of proposed HSD. Moreover, the proposed HSD achieves state-of-the-art performance at real-time speed.

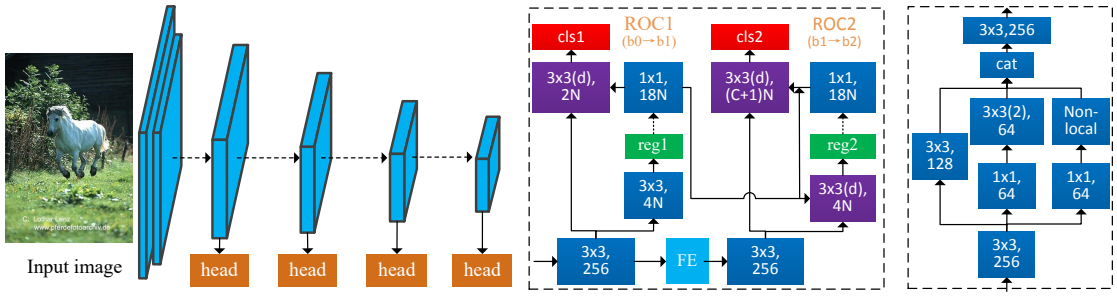
2. Related works

In the past few years, object detection has achieved the great success based on deep convolutional neural networks [24, 43, 18, 46, 37]. Depending on whether or not the candidate proposals (i.e., ROI) are used, the methods are split into the two-stage methods [15, 40, 27, 16] and the one-stage methods [39, 31, 28, 49].

Two-stage methods are proposal-based methods, which firstly generate some candidate object proposals and then use a ROI head-network to classify and regress these proposals. RCNN [15] and its extensions (Fast RCNN [14] and Faster RCNN [40]) are the most representative two-stage methods. After that, many variants have been proposed. To encode position information and reduce computational cost of Faster RCNN, R-FCN [8] replaces the ROI pooling by the position-sensitive ROI (PSROI) pooling. GA-RPN [48] leverages semantic features to guide the anchor generation. To solve scale variance problem, some feature pyramid methods (e.g., MSCNN [1], MCF [3], and FPN [27]) and image pyramid methods (e.g., SNIP [44] and SNIPER [45]) detect objects by multi-scale feature maps or multi-scale images. Mask R-CNN [16] extends object detection to instance segmentation by an extra segmentation branch.

One-stage methods are proposal-free methods, which simultaneously output classification scores and regression offsets of dense default boxes (see Fig. 2(a)). YOLO [39] and SSD [31] are two representative methods. YOLO [39] splits the original image into the $N \times N$ grids and predicts object probabilities existed in each grid. SSD [31] uses the different layers of the backbone to detect the objects of different scales. Different from two-stage methods, one-stage methods need to process a large number of positive and negative samples and face the class imbalance problem. To address this problem, OHEM [42] and focal loss [28] pay more attention to the hard samples. Recently, many works have been proposed to promote the progress of one-stage methods. Some methods [12, 23, 20, 57] aim to enhance the feature semantics of predication layers. Some methods [55, 10, 34, 4, 7] add segmentation supervision to boost detection performance. To avoid the handcrafted anchors, some anchor-free methods [25, 59, 47] are proposed recently. Compared with two-stage methods, one-stage methods are of high efficiency and low accuracy. Thus, we aim to improve detection accuracy of one-stage methods with high efficiency.

Cascade structure is very useful for accurate object detection [2, 19, 54, 32, 60, 35]. Specifically, two-stage detectors Cascade RCNN [2] and IoU-Net [19] have a sequence of ROI detectors, while one-stage detectors RefineDet [54] and ALFNet [32] use multiple fully-convolutional head-networks for predications (see Fig. 2(b)). At each stage, these methods conduct simultaneous classification and re-



(a) The overall architecuture of HSD (b) The head-network (c) The FE module

Figure 3. The overall architecture of HSD in (a), which detects objects at multiple layers by multiple head-networks. The head-network in (b) consists of two stacked ROC modules and one feature enhanced (FE) module in (c). 'd' means the deformable convolution. 'C' means the number of object categories. 'N' means the number of anchors.

gression. As a result, they does not consider that there exists the classification inconsistency between the default box and regressed box. Recently, ConRetinaNet [22] uses both the default box and the regressed box for classification (see Fig. 2(c)) at the training stage, which is an initial attempt to solve the inconsistency. Compared with ConRetinaNet, our method (see Fig. 2(d)) further considers two important inconsistency: the feature inconsistency between default box and regressed box, and the threshold inconsistency of the positive/negative samples for regression and classification.

Contextual information has been demonstrated to be helpful for object detection and semantic segmentation. On the one hand, some methods [21, 27, 26, 51, 50]) use the encoder-decoder structure to combine the feature maps of different scales. On the other hand, some methods [17, 5, 56, 38, 36, 6]) adopt the spatial pyramid structure with multiple branches to extract multi-scale contextual information. Meanwhile, non-local contextual information extracted by self-attention module [52, 13] is also useful. Thus, a natural idea is to combine local and non-local contextual information for object detection. In our cascade structure, the local and non-local contextual information is added before the second ROC which needs more discriminative features for more accurate object detection.

3. Our proposed method

In this section, we give a detailed description about our proposed method (HSD). Fig. 3(a) shows the overall architecture. Given an image, the backbone network (e.g., VGG16 [43]) and two extra convolutional blocks are used to generate the feature maps of different resolutions. Then, the head-network, which contains two *cls-offset-reg* (ROC) modules and one feature enhanced (FE) module, is respectively attached at these feature maps. Finally, the detection results of each head-network are combined by the non-maximum suppression.

In the following section, we firstly introduce the core *reg-offset-cls* (ROC) module. Then, we explain how to hierarchically stack two ROCs and how to extract the local and non-local context by the feature enhanced (FE) module.

3.1. The reg-offset-cls (ROC) module

Generally, one-stage methods simultaneously predict object category and regression offset of the default box $b0$ during training, which have some following drawbacks: (1) the classification score of the default box $b0$ is mistakenly assigned to that of regressed box $b1$ during inference; (2) some regressed boxes accurately detecting objects may be labelled as the negative samples during training, which can be used to improve detection performance. To solve the above problems, a novel *reg-offset-cls* (ROC) module is proposed, which reconstructs regression and classification by three hierarchical steps: firstly predicts the regression offset of default box $b0$, secondly calculates the feature sampling offsets upon the regression output, and finally classifies the regressed box $b1$ with the features of offset locations.

The regression loss (i.e., L^1_{reg}) of the ROC module is the same as that of SSD [31]:

$$L^1_{reg} = \frac{1}{N^1_{reg}} \sum_i L_{reg}(v^{b0}_i, t^{b0}_i(\mathbf{w}^{b0}_{reg}, \mathbf{x}^1)), \qquad (1)$$

where v^{b0} and t^{b0} are respectively the true regression offset and the predicted regression offset of the default box $b0$. $\mathbf{w}^{b0}_{reg}$ and $\mathbf{x}^1$ are respectively the regression convolutional weights and the input features of the ROC. N^1_{reg} is the number of samples for box regression.

Based on the regressed box $b1$, the classification loss (i.e., L^1_{cls}) of the ROC module can be expressed as follows:

$$L^1_{cls} = \frac{1}{N^1_{cls}} \sum_i L_{cls}(u^{b1}_i, p^{b1}_i(\mathbf{w}^{b1}_{cls}, \mathbf{x}^1(\mathbf{\Delta}^{b1}))), \qquad (2)$$

where u^{b1} and p^{b1} are respectively the true label and the predicted score of the regressed box $b1$. N^1_{cls} is the number of samples for box classification. With the technique of deformable convolution [9], the predicted score p^{b1} can

be calculated by convolving the classification weights $\mathbf{w}_{cls}^{b1}$ with the features $\mathbf{x}^1(\mathbf{\Delta}^{b1})$ of offset locations.

The feature sampling offset $\mathbf{\Delta}^{b1}$ of regressed box $b1$ is 18-d, which is learned from the regression output t^{b0} (i.e., $\Delta^{x0}, \Delta^{y0}, \Delta^{w0}, \Delta^{h0}$) of the default box $b0$ as follows:

$$\mathbf{\Delta}^{b1} = \mathbf{w}_1^1 \otimes (\mathbf{w}_2^1 \otimes t^{b0}), \tag{3}$$

where $\mathbf{w}_1^1$ and $\mathbf{w}_2^1$ respectively represent the weights of two 1×1 convolutional layers.

The structure of the ROC module is shown at the left of Fig. 3(b). The feature maps from the main network are fed to a 3×3 convolutional layer to generate the input feature maps (F1) of the first ROC. After that, the regression branch uses a 3×3 convolution to predict the regression offsets. The channel number of regression output is $4N$, where N is the number of anchors. Based on the regression offsets, the feature sampling offset (O1) is calculated by two 1×1 convolutions. To simplify, one convolution is shown. The channel number of sampling offset is $18N$. With F1 and O1, the classification branch uses a 3×3 deformable convolution of group N to output classification results. If HSD uses only one ROC module, the channel number of classification output is equal to $(C + 1)N$, where C is the number of object categories. If HSD uses two stacked ROC module, the channel number of classification output in the first ROC is equal to $2N$, which aims to filter many negative boxes.

Compared with simultaneous regression and classification of default box, our ROC can use more accurately regressed boxes for better training and use the features of more accurate sampling locations for better classification. Moreover, the proposed ROC merely adds two 1×1 convolutions, which is relatively light-weight.

3.2. Two hierarchical ROC modules

For accurate object detection, multiple consecutive regressions and classifications with cascade structure have been demonstrated to be effective [54, 2]. In this paper, two ROC modules are hierarchically stacked to further improve detection accuracy. Based on the regressed boxes and the feature sampling offsets generated by the first ROC module, the second ROC module further predicts the box regression and classifies the regressed boxes by the features of offset locations. At the training stage, the training loss is equal to the loss sum of the two ROC modules.

For the first ROC module, the regression loss (i.e., L_{reg}^1) and the classification loss (i.e., L_{cls}^1) have been shown in Section 3.1. For the second ROC module, the regression loss (i.e., L_{reg}^2) can be written as follows:

$$L_{reg}^2 = \frac{1}{N_{reg}^2} \sum_i L_{reg}(v_i^{b1}, t_i^{b1}(\mathbf{w}_{reg}^{b1}, \mathbf{x}^2(\mathbf{\Delta}^{b1}))), \tag{4}$$

where v^{b1} and t^{b1} are respectively the true regression offset and the predicted regression offset of the box $b1$ by the first ROC module. Similarly, the deformable convolution [9] is used to predict the regression offset t^{b1} of box $b1$ by convolving the regression weights $\mathbf{w}_{reg}^{b1}$ with the features $\mathbf{x}^2(\mathbf{\Delta}^{b1})$ of offset locations.

For the second module, the classification loss (i.e., L_{cls}^2) can be then expressed as

$$L_{cls}^2 = \frac{1}{N_{cls}^2} \sum_i L_{cls}(u_i^{b2}, p_i^{b2}(\mathbf{w}_{cls}^{b2}, \mathbf{x}^2(\mathbf{\Delta}^{b1+b2}))), \tag{5}$$

where u^{b2} and p^{b2} are the true label and the predicted score of the regressed box $b2$. The feature sampling offset $\mathbf{\Delta}^{b1+b2}$ for the box $b2$ is $\mathbf{\Delta}^{b1+b2} = \mathbf{\Delta}^{b1} + \mathbf{\Delta}^{b2}$, where $\mathbf{\Delta}^{b1}$ is calculated by Equation 3 and $\mathbf{\Delta}^{b2}$ is calculated by

$$\mathbf{\Delta}^{b2} = \mathbf{w}_1^2 \otimes (\mathbf{w}_2^2 \otimes t^{b1}), \tag{6}$$

where t^{b1} is regression output (i.e., Δ^{x1}, Δ^{y1}, Δ^{w1}, Δ^{h1}) of the box $b1$. Similar to the convolutional weights (i.e., $\mathbf{w}_1^1$ and $\mathbf{w}_2^1$) in Equation 3, the weights (i.e., $\mathbf{w}_1^2$ and $\mathbf{w}_2^2$) of the convolutional layers are also learned at the training stage.

The detailed structure of two stacked ROC modules can be seen in Fig. 3(b). The detection pipeline of two ROCs is *first regression* $\mapsto$ *first sampling offset* $\mapsto$ *first classification* $\mapsto$ *second regression* $\mapsto$ *second sampling offset* $\mapsto$ *second classification*. The *first classification* uses a 3×3 deformable convolution to consider the feature *sampling offset* caused by the *first regression* of default box, while the *second classification* uses a 3×3 deformable convolution to consider the feature *sampling offset* caused by both the *first regression* and the *second regression* of default box.

3.3. Feature enhanced module

In HSD, the second ROC module aims to generate more accurate classification and location. To further improve feature discrimination, a feature enhanced (FE) module is injected between the two ROC modules, which extracts more local and non-local contextual information to enrich the input features of the second ROC module.

Fig. 3(c) shows the structure of the FE module. Specifically, it consists of three different branches: a convolutional branch, a local context branch, and a non-local context branch. The convolutional branch has a 3×3 convolution. To reduce computational cost, the local context branch and the non-local context branch firstly go through a 1×1 convolutional layer to reduce the channel number of the feature maps. After that, the local context branch goes through a 3×3 convolution with dilation rate of 3, and the non-local context branch goes through a non-local module used in [52]. After that, the output maps of three branches are concatenated together and fed to a 3×3 convolutional layer to generate the input for the second ROC. To achieve a little better performance when the input size is of 512×512, FE module concatenates the feature map at current scale and the upsampled map at next scale as the input.

method	backbone	input size	#reg	#cls	context	AP	AP@0.5	AP@0.75	AP_s	AP_m	AP_l
(a) baseline (SSD-like)	VGG16	320×320	1	1	✗	27.8	46.7	28.4	10.3	27.8	43.1
(b) one ROC	VGG16	320×320	1	1	✗	30.3	51.2	31.0	13.1	32.2	46.4
(c) two ROCs	VGG16	320×320	2	2	✗	32.6	51.5	34.7	15.0	34.9	49.1
(d) two ROCs+FE	VGG16	320×320	2	2	✓	33.3	52.8	36.1	16.1	35.3	49.3
(e) two ROCs+FE	VGG16	512×512	2	2	✓	38.5	57.8	42.2	22.2	42.3	52.3

Table 1. The ablation study of the proposed HSD on the COCO `minival` set [29]. '#reg' and '#cls' means the number of classification and regression. 'context' means the local and non-local context extracted by the proposed FE module.

4. Experiments

In this section, the experiments on the challenging MS COCO [29] and the classic PASCAL VOC [11] are conducted to demonstrate the effectiveness of proposed method and compare with some sate-of-the-art methods.

4.1. Datasets and evaluation metrics

MS COCO [29] is a famous and challenging computer vision benchmark for object detection and instance segmentation, which contains about $115k$ images for training (i.e., `trainval35k`), $5k$ images for ablation experiments (i.e., `minival`), and about $20k$ images for comparing with other methods (i.e., `test-dev`). There are 80 object categories. Mean average precision under different IoU thresholds (0.5:0.95) is used for performance evaluation.

PASCAL VOC [11] is the classic object detection dataset, which mainly contains VOC2007 and VOC2012. VOC2007 has 5011 images for training and 4952 images for testing, while VOC2012 has 5717 images for training, 5823 images for validation, and 10991 images for testing. There are 20 object categories. In this paper, the train set in VOC2007 and the train and validation sets in VOC2012 are used for training, while the test set in VOC2007 is used for testing. The average precision with the IoU threshold of 0.5 is used for performance evaluation.

4.2. Implementation details

The proposed HSD adopts the VGG16 [43], ResNet101 [18], or ResNext101 [53] pre-trained on the ImageNet [41] as the backbone, and further fine-tunes the pre-trained network on the specific object detection datasets. For the COCO benchmark [29], there are 160 epochs at the training stage. The initial learning rate is 0.004. It deceases by a factor of 10 at 90, 120, and 140 epochs for small input size and at 90 and 140 epochs for large input size. For the PASCAL VOC dataset [11], there are 250 epochs at the training stage. The initial learning rate is 0.004 and deceases by a factor of 10 at the 150 and 200 epochs. On both the COCO and PASCAL VOC datasets, each mini-batch has 32 images for training. At the test stage, the threshold of non-maximum suppression (NMS) is 0.45, and the top 200 maximum scoring boxes per image after NMS are saved.

4.3. Ablation experiments

The effectiveness of the proposed HSD The baseline (SSD-like) in Table 1(a) simultaneously predicts object category and regression offset of the default box. Compared with SSD [31], SSD-like removes the L2-Norm layer and adds the 3×3 convolutional layer before each prediction head-network. Compared with the baseline, HSD firstly regresses the default box and then classifies the regressed box with the features of offset locations. The local and non-local contextual information extracted by the feature enhanced (FE) module is injected before the second ROC. Table 1(b)-(e) give the detection results of the proposed HSD. For fair comparisons, they are implemented with the similar parameter settings. The analysis of Table 1 is given as follows:

(1) The HSD in Table 1(b) and the baseline in Table 1(a) both have one regression and one classification. By comparing Table 1(a) and 1(b), it can be seen that HSD with one ROC module outperforms the baseline by 2.5%. Meanwhile, HSD with one ROC module outperforms the baseline at all the scales (see AP_s, AP_m, and AP_l). It can be concluded that the proposed ROC is more effective than the baseline (i.e., simultaneous classification and regression). Compared with the baseline, the proposed ROC only adds two 1×1 convolutions to predict feature sampling offsets. Thus, it does not add many network parameters.

(2) When two ROC modules are stacked together, the performance of HSD can be significantly improved. By comparing Table 1(c) and Table 1(b), HSD with two ROCs outperforms that with one ROC by 2.3%. It is further found that the improvement of AP@0.75 has a large improvement. It means that two stacked ROC modules can provide more precise detection than one ROC module.

(3) When injecting the feature enhanced (FE) module into the two stacked ROC modules, the performance can be further improved, which mainly comes from the improvement of small-scale object detection. The reason may be that detecting small-scale objects is more difficult and needs more contextual information.

(4) With the large input size of 512×512 during training and inference, HSD achieves a better accuracy (i.e., 38.5%).

Comparison with related one-stage methods To further demonstrate the effectiveness of proposed method, some related one-stage methods (i.e., SSD [31] and Re-

method	backbone	cascade	AP	AP@05	AP@075	T (ms)
(a) SSD [31]	VGG16	✗	25.3	42.0	26.2	10
(b) our HSD	VGG16	✗	30.3	51.2	31.0	12
(c) RefineDet [54]	VGG16	✓	29.9	50.2	31.1	13
(d) our HSD	VGG16	✓	32.6	51.5	34.7	16

Table 2. Comparison with two related one-stage methods (i.e., SSD [31] and RefineDet [54]). 'T' means the forward time. For fair comparison, FE module is not used in our HSD.

method	box offset	sampling offset	AP	AP@0.5	AP@0.75
(a) baseline	✗	✗	27.8	46.7	28.4
(b) baseline$_{\theta=0.4}$	✗	✗	28.4	48.2	28.8
(c) baseline$_{\theta=0.6}$	✗	✗	27.1	45.8	27.4
(d) ROC$_{\theta=0.5}$	✓	✗	28.0	49.1	27.9
(e) ROC$_{\theta=0.6}$	✓	✗	29.9	50.2	31.0
(f) ROC$_{\theta=0.7}$	✓	✗	29.8	49.0	31.3
(g) ROC$_{\theta=0.8}$	✓	✗	28.2	45.5	30.9
(h) ROC$_{one\ conv}$	✓	✓	30.2	50.8	31.0
(i) ROC$_{two\ convs}$	✓	✓	30.3	51.2	31.0

Table 3. Ablation study of the ROC module. Box offset means that it firstly conducts box regression and secondly conducts regressed box classification. Sampling offset means that it considers the feature sampling offset for classification due to the box regression.

fineDet [54]) are compared with our HSD in Table 2. They are re-implemented with similar parameter settings. (1) SSD and our HSD with one ROC are both one regression/classification. By comparing Table 2(a) and (b), HSD with one ROC outperforms SSD by 5.0%. (2) RefineDet and our HSD with two ROCs are both two regressions/classifications. By comparing Table 2(c) and (d), HSD with two ROCs outperforms RefineDet by 2.7%.

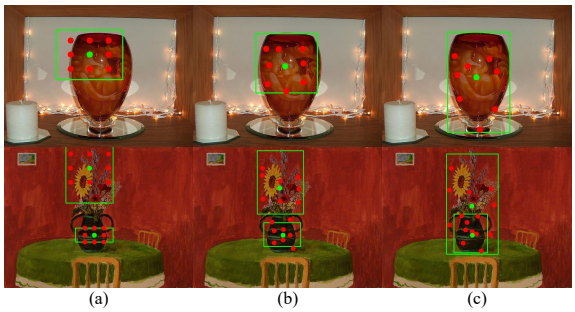
Meanwhile, compared with SSD and RefineDet, our HSD greatly improves detection accuracy nearly without additional computational cost. For example, our HSD outperforms SSD by 5.0% with a merely extra 2ms forward time. Meanwhile, the proposed HSD nearly does not increase network parameters. Based on the above analysis, the proposed HSD is superior to other classical and state-of-the-art one-stage methods (i.e., SSD and RefineDet).

Ablation study of the ROC module Because the ROC module reconstructs the regression and the classification into three hierarchical steps, using the original designs in one-stage methods is not good enough for proposed ROC. The detailed differences are two following folds:

(1) The first one is that how to set the IoU threshold θ of positive/negative samples for the classification after regression. A natural idea is to use the same threshold (0.5) as the regression like most one-stage methods [31, 22]. By comparing Table 3(d) and (a), it can be seen that using the same threshold (e.g., θ=0.5) has very limited improvements. The accuracy of AP@0.75 even decreases. The reason may be

method	sampling offset	AP	AP@0.5	AP@0.75
(a) one ROC	✗	29.9	50.2	31.0
(b) one ROC	✓	30.3	51.2	31.0
(c) two ROCs	✗	31.1	49.4	33.2
(d) two ROCs	✓	32.6	51.5	34.7

Table 4. The effect of feature sampling offset which means whether or not the offset features are used for classification.



(a) (b) (c)

Figure 4. The progress of box regression and the feature sampling locations. The green rectangle means the detection box, the red point means the feature sampling center of 3×3 convolution, and the red point means other sampling locations of 3×3 convolution. (a) the default box $b0$ and feature locations for the first regression. (b) the regressed box $b1$ and feature sampling locations for the first classification and second regression. (c) the regressed box $b2$ and feature sampling locations for the second classification.

that more inaccurately regressed boxes (merely above 0.5 IoU with GT) interfere the training. With the stricter IoU threshold for classification, the performance becomes much better. When the threshold θ is 0.6 in Table 2(e), the performance is 2.0% better than that of baseline. It means that the classification of accurately regressed box by the ROC is better than the classification of default box by the baseline.

Moreover, the detection results that the baseline uses the IoU thresholds of 0.4 and 0.6 are shown in Table 3(b) and (c), which are inferior to that of our ROC. It means that the improvement of ROC cannot be simply achieved by the baseline with more positives or the stricter threshold θ.

(2) The second one is that how to predict the sampling offset based on the regression output. Table 3(h) uses one 1×1 convolution, and Table 3(i) uses two 1×1 convolutions. By comparing Table 3(h) and (i), it can be seen that ROC with two convolutions has a little better performance.

The effect of sampling offsets Table 4 compares the effect of feature sampling offsets on the proposed HSD. It can be seen that the performances of one ROC and two ROCs respectively have 0.4% and 1.5% improvements by using the offset features. It can be explained that using the feature sampling offset can extract more accurate features to classify the regressed box.

Visualization of box regression and sampling locations Fig. 4 visualizes the regressed box and the feature

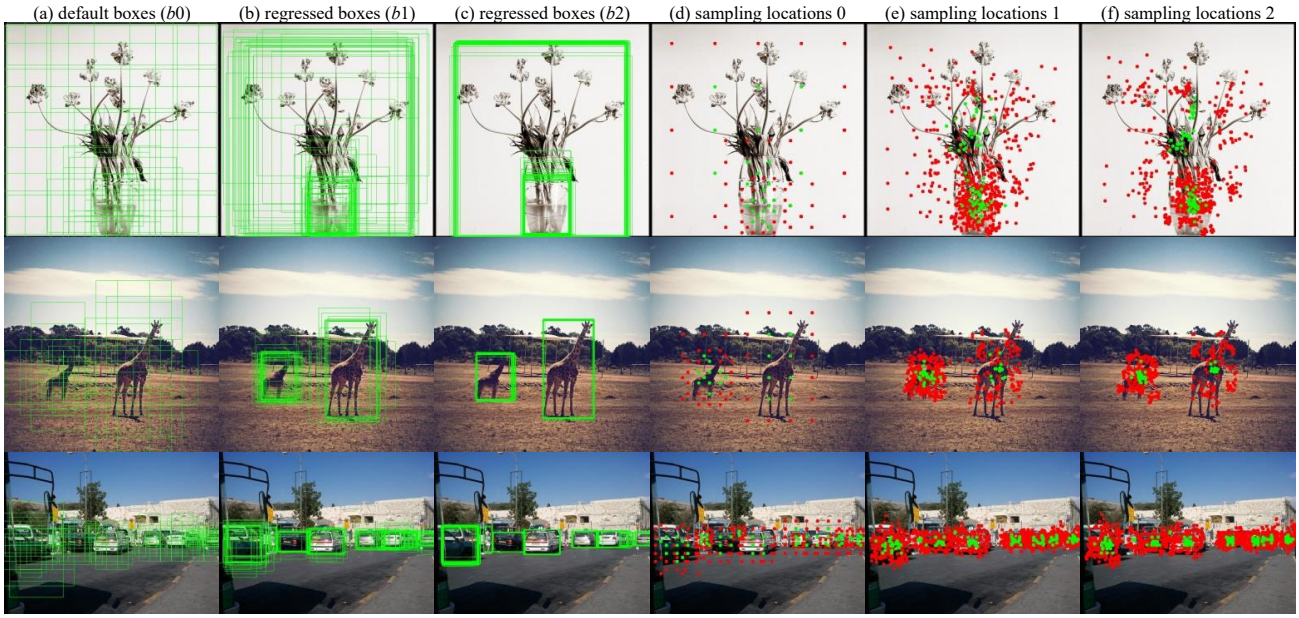


Figure 5. Detection boxes recognized as objects and the sampling locations of features before NMS. The green rectangle means the detection box, the green point means the feature sampling center of 3×3 convolution, and the red point means the other sampling location of 3×3 convolution. (a)-(c) show the progress of box regression ($b0 \mapsto b1 \mapsto b2$). (d)-(f) show the change of feature sampling locations.

method	AP
(a) FE for the second ROC	33.3
(b) FE for the first ROC	32.7
(c) FE without non-local context for the second ROC	33.0

Table 5. The ablation study of feature enhanced (FE) module in the proposed HSD.

sampling locations of the proposed HSD. It can be seen that the box (green rectangle) can accurately detect the objects after two regressions. Meanwhile, the sampling locations of features coincide with the regressed box, which means that the sampling offsets can be accurately predicted by the regression output. Fig. 5 further gives more complete samples of box regression and feature sampling locations before NMS. It can be seen that many boxes become around the objects after two regressions. As a result, the features for box classification are accurately extracted around objects.

Feature enhanced (FE) module To demonstrate the importance of the proposed FE module, Table 5 gives some experiments as follows: (1) The FE module is added to only the first ROC. By comparing Table 5(a) and (b), it can be seen that the FE module is more important to the second ROC. The reason is that the second ROC needs more discriminative features to deal with many hard boxes around objects. (2) Without the non-local context (Table 5(c)), the performance decreases by 0.3%.

Detection results Fig. 6 gives some detection results of the proposed HSD (Table 1(e)). It can be seen that the proposed method achieves a good performance on both small-scale objects and occluded objects in complex scene.

Figure 6. Detection results of the proposed HSD on the COCO `minival` set. It can be seen that the proposed HSD has a good detection performance on small-scale objects and occluded objects.

4.4. Comparisons on the COCO benchmark

In this section, the proposed HSD is compared with some state-of-the-art methods on the `test-dev` set of the COCO benchmark in Table 6. With the same VGG16 and the input size of 512×512, HSD respectively outperforms RefineDet [54] and RFBNet [30] by 5.8% and 4.4%. With the same ResNet101 and the input size of 512×512, HSD respectively outperforms DFPR [21] by 5.6%. Based on the

method	backbone	input size	time	AP	AP@0.5	AP@0.75	AP_s	AP_m	AP_l
Two-stage methods									
FPN [27]	ResNet101	$\sim 1000 \times 600$	172ms	36.2	59.1	39.0	18.2	39.0	48.2
Cascade RCNN [2]	ResNet101	$\sim 1333 \times 800$	140ms	42.8	62.1	46.3	23.7	45.5	55.2
One-stage methods									
SSD [31]	VGG16	300×300	12ms	25.1	43.1	25.8	6.6	25.9	41.4
DSSD [12]	ResNet101	321×321	-	28.0	46.1	29.2	7.4	28.1	47.6
STDN [58]	DenseNet169	300×300	-	28.0	45.6	29.4	7.9	29.7	45.1
DES [55]	VGG16	300×300	-	28.3	47.3	29.4	8.5	29.9	45.2
RefineDet [54]	VGG16	320×320	19ms†	29.4	49.2	31.3	10.0	32.0	44.4
RFBNet [30]	VGG16	300×300	15ms	30.3	49.3	31.8	11.8	31.9	45.9
SSD [31]	VGG16	512×512	28ms	28.8	48.5	30.3	10.9	31.8	43.5
DSSD [12]	ResNet101	512×512	-	33.2	53.3	35.2	13.0	35.4	51.1
STDN [58]	DenseNet169	512×512	-	31.8	51.0	33.6	14.4	36.1	43.4
DES [55]	VGG16	512×512	-	32.8	53.2	34.6	13.9	36.0	47.6
RefineDet [54]	VGG16	512×512	40ms†	33.0	54.5	35.5	16.3	36.3	44.3
RFBNet [30]	VGG16	512×512	33ms	34.4	55.7	36.4	17.6	37.0	47.6
DFPR [21]	ResNet101	512×512	-	34.6	54.3	37.3	14.7	38.1	51.9
TripleNet [4]	ResNet101	512×512	-	37.4	59.3	39.6	18.5	39.0	52.7
CornerNet [25]	Hourglass104	511×511	244ms	40.5	56.5	43.1	19.4	42.7	53.9
ExtremeNet [59]	Hourglass104	511×511	322ms	40.2	55.5	43.2	20.4	43.2	53.1
RetinaNet [28]	ResNet101	$\sim 1333 \times 800$	198ms	39.1	59.1	42.3	21.8	42.7	50.2
ConRetinaNet [22]	ResNet101	$\sim 1333 \times 800$	-	40.1	59.6	43.5	23.4	44.2	53.3
FSAF [60]	ResNet101	$\sim 1333 \times 800$	-	40.9	61.5	44.0	24.0	44.2	51.3
our HSD	VGG16	320×320	25ms†	33.5	53.2	36.1	15.0	35.0	47.8
our HSD	VGG16	512×512	43ms†	38.8	58.2	42.5	21.8	41.9	50.2
our HSD	ResNet101	512×512	48ms†	40.2	59.4	44.0	20.0	44.4	54.9
our HSD	ResNext101	512×512	66ms†	41.9	61.1	46.2	21.8	46.6	57.0
our HSD	ResNet101	768×768	92ms†	42.3	61.2	46.9	22.8	47.3	55.9

Table 6. Detection results and detection time of some state-of-the-art methods on COCO `test-dev` set. All results reported are based on single-scale test. † means detection time, including forward time and NMS time, tested by us with Pytorch0.40 on NVIDIA Titan Xp.

method	backbone	input size	mAP
SSD [31]	VGG16	300×300	77.5
DES [55]	VGG16	300×300	79.7
DSSD [12]	ResNet101	321×321	78.6
STDN [58]	DenseNet169	300×300	79.3
BlitzNet [10]	ResNet50	300×300	79.1
DFPR [21]	VGG16	300×300	79.6
RFBNet [30]	VGG16	300×300	80.5
RefineDet [54]	VGG16	320×320	80.0
SSD [31]	VGG16	512×512	79.5
DES [55]	VGG16	512×512	81.7
DSSD [12]	ResNet101	512×512	81.5
STDN [58]	DenseNet169	512×512	80.9
BlitzNet [10]	ResNet50	512×512	81.5
DFPR [21]	VGG16	512×512	81.1
RFBNet [30]	VGG16	512×512	82.2
RefineDet [54]	VGG16	512×512	81.8
our HSD	VGG16	320×320	81.7
our HSD	VGG16	512×512	83.0

Table 7. Detection results of the single-stage methods on the VOC 2007 test set without COCO pre-training and multi-scale test.

strong backbone Hourglass [33], CornerNet [25] has a little better performance compared to our HSD with the backbone ResNet101. By using a strong backbone ResNext101 [53], our HSD can also outperform CornerNet [25] and ExtremeNet [59] by 1.4% and 1.7%. Meanwhile, the proposed method has a faster detection speed. With the input size of 768×768, our HSD can outperform RetinaNet [28], ConRetinaNet [22], and FSAF [60].

4.5. Comparisons on the VOC2007 dataset

In this section, the proposed HSD is compared with some state-of-the-art methods on the VOC2007 test set in Table 7. With the small input size of 300×300 or 320×320, HSD respectively outperforms RFBNet [30] and RefineDet [54] by 1.2% and 1.7%. With the large input size of 512×512, HSD respectively outperforms RFBNet and RefineDet by 0.8% and 1.2%.

5. Conclusion

In this paper, we proposed a novel pipeline for accurate object detection (called ROC). Instead of simultaneous classification and regression, ROC firstly conducts box regression, secondly predicts the feature sampling locations for box classification, and finally classifies regressed boxes with the features of offset locations. To achieve the better detection accuracy, a hierarchical shot detector is proposed by stacking two ROC modules. Meanwhile, the contextual information is also incorporated to enrich the features of the second ROC module. HSD achieves the state-of-art performance on both the COCO and PASCAL VOC datasets.

Acknowledgments This work was supported by National Natural Science Foundation of China (Nos. 61632018, 61876140), Postdoctoral Program for Innovative Talents (No. BX20180214), and China Postdoctoral Science Foundation (No. 2018M641647).

References

[1] Zhaowei Cai, Quanfu Fan, Rogerio S. Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. *Proc. European Conf. Computer Vision*, 2016.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018.

[3] Jiale Cao, Yanwei Pang, and Xuelong Li. Learning multilayer channel features for pedestrian detection. *IEEE Trans. Image Processing*, 26(7):3210–3220, 2016.

[4] Jiale Cao, Yanwei Pang, and Xuelong Li. Triply supervised decoder networks for joint detection and segmentation. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[6] Hisham Cholakkal, Jubin Johnson, and Deepu Rajan. Backtracking spatial pyramid pooling-based image classifier for weakly supervised topdown salient object detection. *IEEE Trans. Image Processing*, 27(12):6064–6078, 2018.

[7] Hisham Cholakkal, Guolei Sun, Fahad Shahbaz Khan, and Ling Shao. Object counting and instance segmentation with image-level supervision. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019.

[8] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *Proc. Advances in Neural Information Processing Systems*, 2016.

[9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *Proc. IEEE International Conf. Computer Vision*, 2017.

[10] Nikita Dvornik, Konstantin Shmelkov, Julien Mairal, and Cordelia Schmid. Blitznet: A real-time deep network for scene understanding. *Proc. IEEE International Conf. Computer Vision*, 2017.

[11] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[12] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C. Berg. Dssd: Deconvolutional single shot detector. *arXiv:1701.06659*, 2017.

[13] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019.

[14] Ross Girshick. Fast r-cnn. *Proc. IEEE International Conf. Computer Vision*, 2015.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2014.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. *Proc. IEEE International Conf. Computer Vision*, 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *Proc. European Conf. Computer Vision*, 2014.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proc. IEEE International Conf. Computer Vision*, 2016.

[19] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. *Proc. European Conf. Computer Vision*, 2018.

[20] Seung-Wook Kim, Hyong-Keun Kook, Jee-Young Sun, Mun-Cheon Kang, and Sung-Jea Ko. Parallel feature pyramid network for object detection. *Proc. European Conf. Computer Vision*, 2018.

[21] Tao Kong, Fuchun Sun, Wenbing Huang, and Huaping Liu. Deep feature pyramid reconguration for object detection. *Proc. European Conf. Computer Vision*, 2018.

[22] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, and Jianbo Shi. Consistent optimization for single-shot object detection. *arXiv:1901.06563*, 2019.

[23] Tao Kong, Fuchun Sun, Anbang Yao, Huaping Liu, Ming Lu, and Yurong Chen. Ron: Reverse connection with objectness prior networks for object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Proc. Advances in Neural Information Processing Systems*, 2012.

[25] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. *Proc. European Conf. Computer Vision*, 2018.

[26] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017.

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017.

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *Proc. IEEE International Conf. Computer Vision*, 2017.

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *Proc. European Conf. Computer Vision*, 2014.

[30] Songtao Liu, Di Huang, and Yunhong Wang. Receptive field block net for accurate and fast object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018.

[31] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Proc. European Conf. Computer Vision*, 2016.

[32] Wei Liu, Shengcai Liao, Weidong Hu, Xuezhi Liang, and Xiao Chen. Learning efficient single-stage pedestrian detectors by asymptotic localization fitting. *Proc. European Conf. Computer Vision*, 2018.

[33] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. *Proc. European Conf. Computer Vision*, 2016.

[34] Junhyug Noh, Soochan Lee, Beomsu Kim, and Gunhee Kim. Improving occlusion and hard negative handling for single-stage pedestrian detectors. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018.

[35] Yanwei Pang, Jiale Cao, and Xuelong Li. Cascade learning by optimally partitioning. *IEEE Trans. Cybernetics*, 47(12):4148–4161, 2017.

[36] Yanwei Pang, Yazhao Li, Jianbing Shen, and Ling Shao. Towards bridging semantic gap to improve semantic segmentation. *Proc. IEEE International Conf. Computer Vision*, 2019.

[37] Yanwei Pang, Manli Sun, Xiaoheng Jiang, and Xuelong Li. Convolution in convolution for network in network. *IEEE Trans. Neural Networks and Learning Systems*, 29(5):1587–1597, 2018.

[38] Yanwei Pang, Jin Xie, and Xuelong Li. Visual haze removal by a unified generative adversarial network. *IEEE Trans. Circuits and Systems for Video Technology*, 2018.

[39] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016.

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Proc. Advances in Neural Information Processing Systems*, 2015.

[41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

[42] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016.

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[44] Bharat Singh and Larry S. Davis. An analysis of scale invariance in object detection snip. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018.

[45] Bharat Singh, Mahyar Najibi, and Larry S. Davis. Sniper: Efficient multi-scale training. *Proc. Advances in Neural Information Processing Systems*, 2018.

[46] Hanqing Sun and Yanwei Pang. Glancenets efficient convolutional neural networks with adaptive hard example mining. *SCIENCE CHINA Information Sciences*, 61(10):109–101, 2018.

[47] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *Proc. IEEE International Conf. Computer Vision*, 2019.

[48] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019.

[49] Tiancai Wang, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Learning rich features at high-speed for single-shot object detection. *Proc. IEEE International Conf. Computer Vision*, 2019.

[50] Wenguan Wang, Jianbing Shen, Ming-Ming Cheng, and Ling Shao. An iterative and cooperative top-down and bottom-up inference network for salient object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019.

[51] Wenguan Wang, Hongmei Song, Shuyang Zhao, Jianbing Shen, Sanyuan Zhao, Steven C. H. Hoi, and Haibin Ling. Learning unsupervised video object segmentation through visual attention. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019.

[52] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018.

[53] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017.

[54] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. Single-shot refinement neural network for object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018.

[55] Zhishuai Zhang, Siyuan Qiao, Cihang Xie, Wei Shen, Bo Wang, and Alan L. Yuille. Single-shot object detection with enriched semantics. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018.

[56] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017.

[57] Qijie Zhao, Tao Sheng, Yongtao Wang, Zhi Tang, Ying Chen, Ling Cai, and Haibin Ling. M2det: A single-shot object detector based on multi-level feature pyramid network. *Proc. AAAI Conf. Artificial Intelligence*, 2018.

[58] Peng Zhou, Bingbing Ni, Cong Geng, Jianguo Hu, and Yi Xu. Scale-transferrable object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018.

[59] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019.

[60] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019.