DuBox: No-Prior Box Objection Detection via Residual Dual Scale Detectors

Shuai Chen Jinpeng Li Chuanqi Yao Wenbo Hou Shuo Qin Wenyao Jin Tang Xu
Baidu Inc.
chenshuai11,lijinpeng,yaochuanqi,houwenbo,qinshuo01,jinwenyao,tangxu02@baidu.com

Abstract

Traditional neural objection detection methods use multi-scale features that allow multiple detectors to perform detecting tasks independently and in parallel. At the same time, with the handling of the prior box, the algorithm's ability to deal with scale invariance is enhanced. However, too many prior boxes and independent detectors will increase the computational redundancy of the detection algorithm. In this study, we introduce Dubox, a new one-stage approach that detects the objects without prior box. Working with multi-scale features, the designed dual scale residual unit makes dual scale detectors no longer run independently. The second scale detector learns the residual of the first. Dubox has enhanced the capacity of heuristic-guided that can further enable the first scale detector to maximize the detection of small targets and the second to detect objects that cannot be identified by the first one. Besides, for each scale detector, with the new classification-regression progressive strapped loss makes our process not based on prior boxes. Integrating these strategies, our detection algorithm has achieved excellent performance in terms of speed and accuracy. Extensive experiments on the VOC, COCO object detection benchmark have confirmed the effectiveness of this algorithm.

1. Introduction

Object detection has been a challenging issue in the field of computer vision for a long time. With the development of deep neural networks (DNN), significant progress has been made in object detection in recent years. It is a prerequisite for a variety of industrial applications, such as autonomous driving [11] and face analysis[25]. Due to the advancement of deep convolutional neural networks [24, 6] and well-annotated data sets[3, 14], the performance of object detectors has been significantly improved.

Images in the real world contain different scale objects. Scale variation has become a challenging problem in the field of objection detection. To achieve scale invariance, state-of-the-art approaches typically combine features of multiple levels to construct a feature pyramid or multi-layer feature tower. Meanwhile, to improve the detection performance, the multi-scale method uses multiple detectors in parallel at various scales. For example, RetinaNet[13] has five scale detectors ($p3$-$p7$) that are detected in parallel on the feature pyramid structure[12]. YOLOv3[20] has three detectors running on the main network.

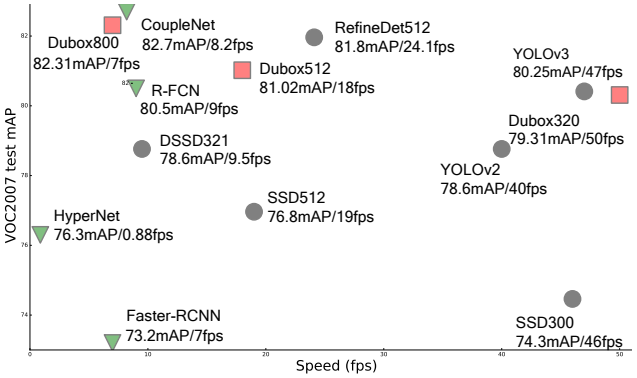

Figure 1. Some comparisons with the precision and speed to classical algorithms on VOC07, ▽ is two-stage method, ◯ denote the one-stage algorithm, □ is Dubox.

In addition, the prior box is considered to be an effective means for dealing with scale invariance. It is fundamental for lots of detectors, e.g., anchors in Faster RCNN[21] and YOLOv2[19], default boxes in SSD[16]. Prior boxes are a bunch of boxes with pre-defined sizes and aspect ratios that tile the feature map in a sliding window manner, to serve as detection candidates. The prior box discretizes the space of possible output bounding-box shapes, and DNN regresses bounding-boxes based on a specific prior-box taking advantage of prior information. Hybridising of multi-scale detection and prior boxes is a common practice in state-of-the-art detectors, which takes advantage of multi-scale features and pre-computed bounding box statistics.

In multi-scale detectors, a specific feature level is responsible for objects with similar scales. The correspondence between the object scale and the feature level is found independently of each other by heuristic-guided feature selection and different detectors. However, one of the drawbacks of this design is computation redundancy. Spatial scales of features are discrete, which violate the scale continuity in the real world. Each scale detector will try to de-

tect objects lying in the intermediate scales of them. To regress multiple scale objects, more detectors are needed in this design. Meanwhile, a large number of anchor boxes are required to overlap with object bounding boxes sufficiently. For example, there are more than 100k anchors in RetinaNet[13], which results in an imbalance between positive and negative samples. These strategies increased the computational complexity. Although, in early stage, some works have explored the no-prior box detection like YOLOv1 [18] and Densebox[7], they achieved high speed in a sacrifice of accuracy. Recently, some algorithms such as CornerNet[9] designed the paired key points to remove the anchor settings and delivered excellent performance.

Based on these observations, we explore two issues in this paper: Is it possible to achieve excellent performance using less scale detectors? Is it possible to regress accurate bounding boxes without anchor? In other words, can we design high-performance dual scale detectors algorithms that doesn't use the prior box?

In this paper, we introduce Dubox method, an one-stage approach for object detection without prior boxes. Dubox classifies objects and regresses bounding boxes directly in one network. The object detection problem becomes a pixel-wise classification-regression problem. To alleviate the influence of scales, we also design a dual-scale structure called dual scale residual unit, which allows dual scale detectors to no longer run independently. In Dubox, the second scale detector learns the residual of the first. In this residual design, we add some computational redundancy reduction strategies by enhancing the capability of heuristic-guided. This method can further encourage the first scale detector to maximize the detection ability of small targets, and let the second scale detector to detect objects that cannot be identified by the first one. Although the two independent scales do not have good performance when test separately, the joint inference results have achieved outstanding performance. At the same time, for each detector, our detection framework with the new classification-regression progressive strapped loss makes our process not based on prior boxes. Integrating these strategies, our detection algorithm has achieved excellent performance in terms of speed and accuracy. Extensive experiments on the VOC[3], COCO[14]object detection benchmark confirm the effectiveness of our method. Some comparisons with the precision and speed to classical algorithms are shown in Fig.1.

2. Related Work

Detection by single-scale

Single-scale detectors detect targets in a typical ratio and cannot identify objects in other proportions. To overcome this drawback, many algorithms use image pyramids, and each proportional object in the pyramid is fed into the detector. Such framework design is prevalent in algorithms that do not use deep learning, and often design some manual features such as HOG [2] or SIFT[17]. There are also algorithms using this method in the CNN network, like [10]. The detector only processes the features in a specific range, and based on this feature map, classifies and regresses the object box. Although this may reduce the detection difficulty of the detector, on the whole, it has a substantial computational cost, making it not easy to use on devices with low computing capability, which greatly limits its practicality.

Detection by multi-scale

The multi-scale detection algorithm only needs a fixed-scale input and detects objects with diverse sizes. YOLOv3[20] and RetinaNet[13] had fixed input sizes and detected in parallel at various scales by using multiple detectors. In general, detectors in the lower layers detect small targets and the uppers are more accessible to identify large objects. This is a heuristic-guided strategy. The addition of the anchor design further strengthens this guidance. However, because multiple levels of detectors operate independently in parallel in each scale feature, there is no cooperation between them, resulting in a large amount of detection redundancy. In the meanwhile, the common design of anchor in these detectors dramatically increases the number of output channels and aggravates the computation burden.

3. Our Method

In this section, we depict each component of our pipeline (Fig.3) in detail. We first devise the classification and regression target label maps for our no-prior box detector in Section 3.1.

The dual scale residual unit is designed for letting the high-level detector to learn the residual of the low-level one, which is described in Section 3.2. To get ride of prior boxes and make the detector's classification and regression work synergistically, we designed classification-regression progressive strapped loss, which will be explained in Section 3.3. Many redundancy reduction strategies are added to enhance the capability of heuristic guiding in 3.4. In Section 3.5, we represent the positive and negative sample balance and data augmentation strategies for the detector during the training phase.

3.1. No-prior Box Detection

In this section, we describe the method to generate targets for classification and regression. We transform bounding box ground truth represented by coordinates to pixel-wise label maps.

Hooks

Dubox is a single neural network unifying all necessary components of object detection. The detector design enables end-to-end training and real-time inference while maintaining high average precision.

Our network takes the whole image as input and predicts the result feature maps with the down-sampling level of s-times. Supposing the output map size is (h, w), we define the location (i, j) in output as hook, where $i \in [0,w)$ and $j \in [0,h)$. Dubox predicts each bounding box and its confidence scores of all categories at each hook on the output feature, as shown in Fig.2.

Note that hooks are parameters predefined by the network output. They represent the positions of each points on the output map. We will use this feature to design the target maps for classification and regression.

Classification and regression target map

Suppose there are $w \times h$ hooks in output. An bounding box (x_1, y_1, x_2, y_2) of an object in the output map represents its left-top and right-down corner points. They are sample mapping from location $(x_1 s, y_1 s, x_2 s, y_2 s)$ in origin image by stride s. we define the positive range Θ with the following condition:

$$(i - (x_2 + x_1)/2)^2 + (j - (y_2 + y_1)/2)^2 \leqslant r^2, \quad (1)$$

where $r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}/p$. That means if a hook (i, j) falls into the range Θ of an bounding box, then it's responsible for detecting the corresponding object. Each hook predicts one bounding box$(Pr_{\Delta w_1}, Pr_{\Delta w_2}, Pr_{\Delta h_1}, Pr_{\Delta h_2})$ and one confidence score Pr_{cls} for this object. This confidence score reflects how confident the model is that the hook is in the range of the object and also how accurate it thinks it is belong to an class is that it predicts. p is an predefined value for adjusting the range. The size of this value will affect the number and proportion of hooks that large objects and small objects occupy in detecting. We will discuss it further in Section 3.4.

For regression target, traditional methods[21, 16] regress the center(c_x, c_y), width w and height h. Each bounding box consist of 4 predictions: (c_x, c_y, w, h). However, with the position (i, j), because the regression box can adjust all of its offset values, but the locations of classification cannot change. This approach will result in inconsistency between classification and regression.

As shown in Fig.2, we designed a hook-based regression target that each bounding box consists of 4 predictions: $(\Delta w_1, \Delta w_2, \Delta h_1, \Delta h_2)$ which represent the offset to the positive hooks (i, j) in an object:

$$\begin{aligned} \Delta w_1 &= i - x_1, \Delta w_2 = x_2 - i \\ \Delta h_1 &= j - y_1, \Delta h_2 = y_2 - j. \end{aligned} \quad (2)$$

With such design, each hook center must reside in its own box prediction. Consequently, the result of the classification and the result of the regression will not have inconsistency by predicting different objects in the image. In inference phase, using the fixed hook (i, j), and predict offset

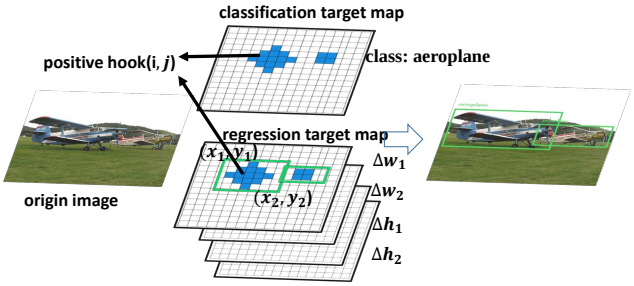

Figure 2. DuBox use fixed hook(i, j) to unite bounding box prediction and classification. The blue points are positive hooks, others are negatives.

$(Pr_{\Delta w_1}, Pr_{\Delta w_2}, Pr_{\Delta h_1}, Pr_{\Delta h_2})$, we can obtain the bbox results in origin image by:

$$\begin{aligned} x_1 &= (i - Pr_{\Delta w_1})\, s, x_2 = (i + Pr_{\Delta w_2})\, s \\ y_1 &= (j - Pr_{\Delta h_1})\, s, y_2 = (j + Pr_{\Delta h_2})\, s. \end{aligned} \quad (3)$$

In our Dubox detector, we use two different down-sampling scales $detector_1$ $s = 8$, $detector_2$ $s = 32$. Thus, with the input (w_{ori}, h_{ori}) image, our final prediction is a $\left(\frac{w_{ori}}{8} \times \frac{h_{ori}}{8}\right) \times C \times 5$ tensor in $detector_1$, $\left(\frac{w_{ori}}{32} \times \frac{h_{ori}}{32}\right) \times C \times 5$ tensor in $detector_2$, where C is the number of class. For PASCAL VOC dataset $C = 20$ and for COCO dataset $C = 80$.

3.2. Residual Dual Scale Detectors

Dual scale residual unit is a sub-structure based on a shared feature extraction backbone. The residual dual scale detector combines the features of different levels detectors by sharing the feature extraction network such as VGG-16[24], ResNet[6]. The structure of residual unit contains two detectors where the high-level detector will learn the residual of regression boxes found in low-level detector. The detailed structure is shown in Fig. 3. $Detector_1$ connects the features at s=8 and s=16, $detector_2$ adds features at s=32 and s=64. De-convolution (stride 2, 1×1 kernel and 256 channels) is used in up-sampling feature maps of different scales to the same spatial size. We do not connect the features of s = 32 and s = 16 used in FPN[12] structure.

Refine module

In our network, an object box in output map contains positive and negative hooks, which requires our system to consider the surrounding situation when classifying the hooks. Combining the different scales, hooks can learn features with enlarged receptive fields. After mixing feature maps, every scale are connected to a refine module, which is a simple implementation of channel and spatial attention model [27]:

$$\begin{aligned} \boldsymbol{\gamma} &= \Phi(\boldsymbol{V}), \\ \boldsymbol{x} &= f(\boldsymbol{V}, \boldsymbol{\gamma}), \end{aligned} \quad (4)$$

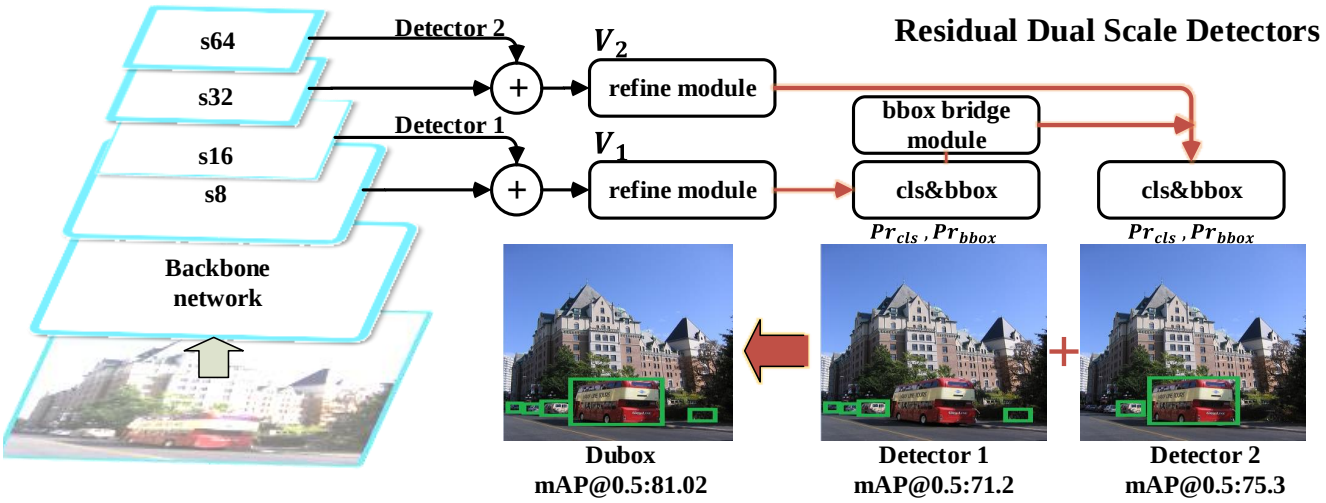

Figure 3. The residual dual scale detectors structure, the $detector_2$ will learn the residual of $detector_1$ through an down-sample bbox bridge module.

where $\boldsymbol{V}$ is the input feature of the refine module. $f()$ is a multiplication for feature map regions and corresponding region weights, Φ is refine module.

The detail design of refine module is shown in Fig.4, where $Sigmoid\,(x) = \frac{1}{1+e^{-x}}$ and $\mathrm{Re}LU\,(x) = \max\,(0,x)$. In the structure, we use an convolution (stride 2, 1×1 kernel and 256 channels) and De-convolution (stride 2, 1×1 kernel and 256 channels) to reduced by 2 times and enlarge the feature map back to the input size of the refine module, this technique helps our detector to push further the ability of considering the around feature for prediction.

Bbox bridge module

Bbox (bounding box) bridge module connects the regressions of low-level and high-level detector, so that the high-level regression is based on the low-level residuals. We inductively describe residual dual scale detectors as follows:

$$\begin{aligned} Pr^{b+1}_{bbox}\,(\boldsymbol{V_{b+1}}) &= \phi\left(Pr^{b}_{bbox}\,(\boldsymbol{V_b})\right) + \tau\,(\boldsymbol{V_b}\,) \\ Pr^{1}_{bbox}\,(\boldsymbol{V_1}) &= \tau\,(\boldsymbol{V_1}\,), \end{aligned} \tag{5}$$

where $\boldsymbol{V_b}$ is the input feature map of $detector_b$, $Pr_{bbox}\,(\boldsymbol{V_b})$ denote $detector_b$ predict the bbox with the input $\boldsymbol{V_b}$ and it equal to $\tau(\boldsymbol{V_b})$ when $b = 1$, $\phi()$ is the bbox bridge module, it contains two convolution (stride 2, 1×1 kernel,$4c$ channels). The bbox bridge module transmits the residual of low-level to high one by stride 4. The details structure is shown in Fig.4.

Consequently, the residual dual scale detectors make the $detector_2$ to perform residual learning based on the prediction of the $detector_1$. This method makes multi-detector in our design not independent as the higher scale depends on the low-level's results.

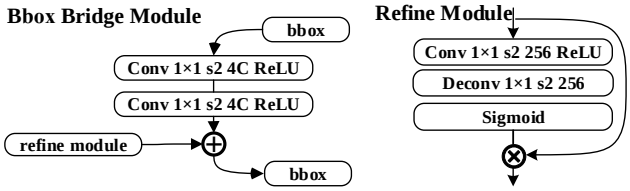

Figure 4. The detail structure of bbox bridge module and refine module.

3.3. Classification-Regression Progressive Strapped Loss

In the anchor-based method, with the help of the prior box, the detector has prior knowledge of box shapes. And it performs its prediction by adjusting the pre-defined anchor shape which boosts their fitting ability of around anchors. Dubox donot have any prior box shape, we have to design a more robust classification and regress strategy, primarily the loss function.

In order to regress the bounding box target of the offset $(\Delta w_1, \Delta w_2, \Delta h_1, \Delta h_2)$ to positive hooks without prior, a loss function which is robust to objects of varied shapes and scales is in need. IoU loss normalize the loss of boxes with different scales by their areas and show robustness to objects of various shapes and scales[26]. The mathematical form of IoU loss can be expressed as:

$$L_{bbox} = -\sum_{i,j\in\boldsymbol{\Theta}} ln\left(IoU\left(\boldsymbol{Pr^{i,j}_{bbox}}, \boldsymbol{Gt^{i,j}_{bbox}}\right)\right), \tag{6}$$

where $\boldsymbol{Gt^{i,j}_{bbox}}$ is the ground truth box of hook (i,j), $\boldsymbol{Pr^{i,j}_{bbox}}$ denotes the the predict bbox in hook(i,j). $IoU\left(\boldsymbol{Gt^{i,j}_{bbox}}, \boldsymbol{Pr^{i,j}_{bbox}}\right)$ denotes the Intersection-over-union (IoU) between the predicted bounding box and

ground truth. For regression, we only regress to the positive samples and ignore the negative ones. In the actual implementation and show in Fig.5, we use Sigmoid to normalize the prediction to $[0, 1]$. Correspondingly, we also map our predicted targets to $[0, 1]$ and Eq.2 change into

$$\begin{aligned} \Delta w_1 &= (i - x_1)/w, \Delta w_2 = (x_2 - i)/w \\ \Delta h_1 &= (j - y_1)/h, \Delta h_2 = (y_2 - j)/h. \end{aligned} \tag{7}$$

For classification problems, logistic regression with cross entropy loss is widely used in objection detection methods. The classification loss function can be described as:

$$L_{cls} = -\sum_{i,j=0}^{h,w} CE\left(\boldsymbol{Pr}_{\boldsymbol{cls}}^{\boldsymbol{i,j}}, \boldsymbol{T}_{\boldsymbol{cls}}^{\boldsymbol{i,j}}\right), \tag{8}$$

where $\boldsymbol{T}_{\boldsymbol{cls}}^{\boldsymbol{i,j}}$, $\boldsymbol{Pr}_{\boldsymbol{cls}}^{\boldsymbol{i,j}}$ is the class label and predict result of hook (i, j), $CE()$ is the cross entropy function.

However, this classification and regression is flawed as the two losses are independent which results in inconsistency during prediction. Our experiments shows that the detector often predicts a right bounding box which the classification fails to predict the right class.

Based on this observation we rebuild the classification loss progressive strap by IoU:

$$\begin{aligned} L_{cls} = &-\sum_{i,j\in\boldsymbol{\Theta}} CE\left(\boldsymbol{Pr}_{\boldsymbol{cls}}^{\boldsymbol{i,j}}, \boldsymbol{T}_{\boldsymbol{cls}}^{\boldsymbol{i,j}}\right)\sigma\left(\boldsymbol{Pr}_{\boldsymbol{bbox}}^{\boldsymbol{i,j}}, \boldsymbol{Gt}_{\boldsymbol{bbox}}^{\boldsymbol{i,j}}\right) \\ &-\sum_{i,j\notin\boldsymbol{\Theta}} CE\left(\boldsymbol{Pr}_{\boldsymbol{cls}}^{\boldsymbol{i,j}}, \boldsymbol{T}_{\boldsymbol{cls}}^{\boldsymbol{i,j}}\right), \end{aligned} \tag{9}$$

where $\sigma()$ is the IoU gate unit, it can be defined as:

$$\sigma\left(\boldsymbol{Pr}_{\boldsymbol{bbox}}^{\boldsymbol{i,j}}, \boldsymbol{Gt}_{\boldsymbol{bbox}}^{\boldsymbol{i,j}}\right) = \begin{cases} 1 & if\ IoU\left(\boldsymbol{Pr}_{\boldsymbol{bbox}}^{\boldsymbol{i,j}}, \boldsymbol{Gt}_{\boldsymbol{bbox}}^{\boldsymbol{i,j}}\right) > \epsilon \\ 0 & elsewise. \end{cases} \tag{10}$$

As shown in Fig.5, for positive hooks (i, j), the classification includes it as a positive sample, only if the predicted and ground truth of the regression match ϵ overlaps. Otherwise, it's ignored. In our experiment ϵ is 0.5.

With the dual scale detectors, the final loss funciton:

$$L = \sum_{b=1}^{2} \left(\lambda_{bbox} L_{bbox} + \lambda_{cls} L_{cls}\right), \tag{11}$$

where λ_{bbox}, $\lambda_{cls}$is the hyper-parameter used to keep the task of classification and regression of $detector_b$ in balance.

3.4. Reducing Redundancy Strategy

The primary goal of the residual dual scale detectors is to maximize the overall fitting capacity of a multi-detector. In general, high-level detector are better at detecting large objects, while low-levels are more sensitive to small ones in image. To enhance this capability of heuristic guiding and reduce redundancy we adopt the following strategies:

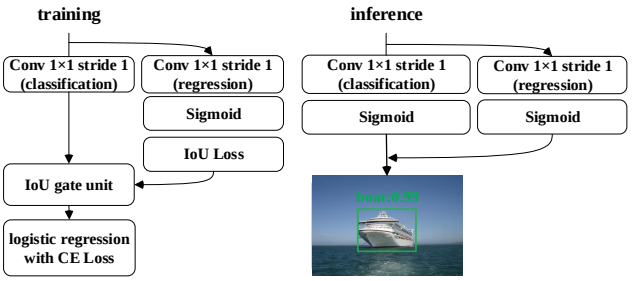

Figure 5. In the training phase, classification-regression progressive strapped loss(CRPS loss) working with the IoU gate unit. With the classification and regression hooks, we can get the detecting results by Eq.3 in inference phase.

Differentiate positive range

As mentioned in Eq.1, p is a predefined value for adjusting the positive range. The size of this value will affect the number and proportion of hooks between large and small objects occupy in detecting. In order to control the proportion of large and small samples in $detector_1$, we design p is 10 in $detector_1$ and 9 in $detector_2$. At the same time, add a constraint to the positive range of $detector_1$ that

$$r = arg\min(r, 3). \tag{12}$$

This method ensures the numbers of large object positive hooks have a limit and this method improves the performance of low-levels in detecting small objects.

Differentiate scale weight

In order to further differentiate the detection capabilities of the two detectors. If the target bounding box of an object occupies an area greater than 0.3 in the original image, the regression of our $detector_1$ will ignore this object, and it can be described as

$$L_{detector_1} = \sum_{i,j\in\boldsymbol{\Theta}} \boldsymbol{\lambda}_{\boldsymbol{bbox}}^{\boldsymbol{i,j}} L_{bbox}^{i,j} + \lambda_{cls} L_{cls} \tag{13}$$

where $\boldsymbol{\lambda}_{\boldsymbol{bbox}}^{\boldsymbol{i,j}}$ is zero if the target bbox of an object occupies an area greater than 0.3 in the original image, other situations are 1.

3.5. Data Augment and Sample Balance

We use several data augmentation strategies presented in [16] to construct a robust model for adapting the variations of objects. That is, we randomly expand and crop the original training images with additional random photo-metric distortion and flipping to generate the training samples[30]. In additional to these methods, we use a batch balance method.

Batch balance

We first traverse the entire data set to create a hash category table. For each image, we select the category of an

object as the primary attribute in turn, which means that if a picture has n objects, this picture will appear n times in the hash table. In training, we select a category picture from the header of the hash category table with equal probability to fill the current batch in training phase.

Positive and negative sample balance

To mitigate the imbalance issue, we design a positive and negative sample balance and online hard example mining (OHEM)[23]. Concretely, in the training phase, assuming the number of the positive hooks in Θ is N, we will select $3N$ negatives in the output map through sorting the negative hooks by the loss and select the top-$3N$ negatives. The rest of negatives are ignored. This strategy is only used in classification task.

4. Experiment

4.1. Implementation details

As a common practice, the backbone network of our Dubox method is initialized by the pretrained VGG-16[24] and ResNet-101[6] classifier trained from Imagenet[22], and the extra added convolution layers are randomly initialized by the $Xavier$ [5] method. During training, all parameters of the network is fine-tuned on the detection datasets. For simplicity, all architectures including the dual scale residual unit of our Dubox are trained in the end-to-end manner. We train the network on 8 Nvidia P40 GPUs using the synchronized SGD with gradient clipping value 10. The momentum is 0.9 and weight decay is 0.0005.

4.2. Ablation Experiments

4.2.1 Baseline

We experiment with several variants of Dubox to demonstrate how the key components of it affect the detection performance. VOC 2007 and VOC 2012 trainval datasets are used to train the models, and all hyper training parameters and input size (512×512) keep same among all models for fair comparison. VOC 2007 test set is used as the testing data. Smooth $l1$ loss[21] is widely used as the box regression method by the anchor-based one stage and two stage detection methods and we use the loss without anchors as our baseline. Because the anchor-based methods have the prior box shape and size. But no-prior box method don't have the prior knowledge. With the help of prior bix, the detector only get 54.3%mAP.

In ablation experiments, we use the IoU loss with batch balance and OHEM as the box regression loss, and Tab.1 shows that Dubox (73.4%mAP) can improve **9.84%** mAP using the IoU loss compare to the smooth $l1$ loss with batch balance and OHEM (63.56%mAP).

Table 1. Comparison of Dubox with different key components on pascal VOC 2007 test (512×512).

Component	Dubox							
Batch balance and OHEM?		✓	✓	✓	✓	✓	✓	✓
IoU loss?			✓	✓	✓	✓		
Hooks?				✓	✓	✓	✓	✓
Refine module?					✓	✓	✓	✓
Residual dual scale?						✓	✓	✓
CRPS loss?							✓	✓
Reducing redundancy strategy?								✓
mAP@0.5	54.32	63.56	73.4	74.2	75.7	77.01	79.43	81.02

4.2.2 No-prior box detection

We also experiment how the fixed hooks in classification and regression affect the detection performance. In Tab.1, it is clear that using fixed hooks to regress $(\Delta w_1, \Delta w_2, \Delta h_1, \Delta h_2)$ outperforms methods which predict (c_x, c_y, w, h) by **0.8%** mAP. This also confirms that the hook-based approach has better performance than traditional methods.

4.2.3 Importance of residual unit and why dual scales

We construct a baseline by cutting the box bridge and refine module in Dubox to isolate the direct communication between the detectors, and this proves that these two components have a performance improvement of **2.81%** mAP, as shown in Tab.1.

We also find that $detector_2$ has to be trained more iterations than $detector_1$ for achieving the optimal detection performance. Specifically, the loss of $detector_1$ is stable after $80k$ training iterations, but the mAP of $detector_2$ continues to increase until $140k$ iterations. We suppose that the optimization goal of $detector_2$ is based on the residual of $detector_1$. The convergence of $detector_2$ needs to be under the premise of $detector_1$. So the time to convergence for $detector_2$ is longer than $detector_1$. Under the limits of training time, we only trained Dubox with dual scale detectors which already achieves high detection performance, but we hypothesis more detectors may further improve the performance with our residual unit which can be studied in further work.

4.2.4 Importance of CRPS loss?

To demonstrate the effectiveness of the proposed CRPS loss, we remove the IoU gate unit between the classification loss and regression loss. In this setting, no ground truth box is filtered, and all ground truth boxes are used in computing loss for both detectors. Hooks corresponding to positive sample compute classification and regression loss, and others corresponding to negative samples only compute in classification loss. Tab.1 shows that this setting leads to drop **2.42%** mAP compared to the method with IoU gate unit. This significant performance decline indicates that the CRPS loss is effective for Dubox.

Table 2. The performance of dual branch on VOC2007 dataset (512×512).

detector	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
$Detector_1$	71.26	77.52	80.30	60.94	59.20	77.32	81.50	83.48	78.82	41.78	59.35	69.24	69.39	80.61	78.66	71.48	32.89	60.95	60.78	67.01	68.70
$Detector_2$	75.37	79.12	86.33	71.46	67.76	64.10	82.91	86.62	80.53	61.04	80.62	72.77	79.90	84.60	80.92	78.75	48.93	74.43	73.91	78.41	74.37
$Joint$	81.02	85.28	88.39	79.89	74.31	67.88	86.71	85.58	89.35	60.62	86.78	73.37	88.82	88.62	85.03	83.21	55.43	84.42	82.89	86.02	77.85

Table 3. Comparison with state-of-the-art detectors on VOC 2007 and 2012

Methods	Backbone	Input Size	FPS	mAP@0.5(VOC07)	mAP@0.5(VOC12)
two-stage detectors					
Faster R-CNN	VGG-16	1000×600	7	73.2	70.4
HyperNet	VGG-16	1000×600	0.88	76.3	71.4
DeRPN	VGG-16	1000×600	-	76.5	71.9
Faster R-CNN	ResNet-101	1000×600	2.4	76.4	73.8
R-FCN	ResNet-101	1000×600	9	80.5	77.6
CoupleNet	ResNet-101	1000×600	8.2	82.7	80.4
one-stage detectors					
YOLO	GoogleNet	448×448	45	63.4	57.9
SSD300	VGG-16	300×300	46	74.3	72.4
SSD512	VGG-16	512×512	19	76.8	74.9
YOLOv2	Darknet-19	544×544	40	78.6	73.4
DSSD321	ResNet-101	321×321	9.5	78.6	76.3
DSSD513	ResNet-101	513×513	5.5	81.5	80.0
YOLOv3	Darknet-53	416×416	47	80.25	-
RefineDet512	VGG-16	512×512	24.1	81.8	80.1
DFPR-Net512	VGG16	512×512	-	81.1	80.0
DR-Net512	ResNet-101	512×512	-	82.0	80.4
ours					
Dubox320	VGG-16	320×320	**50**	79.31	78.82
Dubox512	VGG-16	512×512	18	81.02	80.36
Dubox800	VGG-16	800×800	7	82.31	81.75
Dubox800(multi-scale)	VGG-16	800×800	-	**82.89**	**82.01**

4.2.5 Importance of reducing redundancy

We also compare the detection performance of dual scale detector on different object categories with the reducing redundancy strategy. As shown in Tab.2, we find that the mAP on small object, such as bottle, of $detector_1$ is always higher than $detector_2$, and the mAP on large object, such as airplane, of $detector_2$ is higher than $detector_1$. The performance gaps of the detectors is further enlarged by the mechanism of reducing redundancy strategy. This result indicates that the reducing redundancy strategy helps the two detectors to focus on detecting different scales objects.

4.3. Comparisons with state-of-the-art methods

4.3.1 VOC

VOC 2007 $trainval$, $test$ and VOC 2012 $trainval$ set are used as the training data. We set the batch size to 50(320×320), 32(512×512), 16(800×800) for each GPU in training, and train the model with 10^{-3} learning rate for the first 80k iterations, then reduce to 10^{-4} for another $40k$ iterations, respectively. We evaluate our detector on VOC07 and VOC12 $test$ set. As the performance show in Tab.3, the proposed Dubox detector gets **81.02%** on VOC07 and **80.36%** on VOC12 with input size 512×512, and get **82.31%** and **81.75%** when the input size is 800×800.

Even feeding with 320×320 input size, Dubox obtains the top **79.31%**mAP on VOC07 and **78.82%** on VOC12, which is even better than most of those two-stage methods using about 1000×600 input size (e.g., 70.4% of Faster R-CNN [21], 76.5% of DeRPN[28] and 77.6% of R-FCN [1]). With the input size 512×512 or 800×800, the performance is surpassing the one-stage methods(e.g., 63.4, 57.9 in YOLO[18], 78.6, 73.4 in YOLOv2[19], 80.25 in YOLOv3 [31], 78.6, 76.3 in DSSD[4],81.8, 80.1 in RefineDet[30], 81.2% in DFPR-Net[8], 82.0% in DR-Net[29]). Meanwhile, we experiment Dubox800 with multi-scale images, it gets **82.89**, **82.01%**mAP.

4.3.2 COCO

We also evaluate Dubox on MS COCO[14]. Unlike PASCAL VOC, we report the results of ResNet-101 based DuBox directly. Following the protocol in MS COCO, $trainval35k$ set [14] is used for training and evaluate

Table 4. Comparison with state-of-the-art detectors on MS COCO test-dev

Methods	Backbone	Data	AP	AP50	AP75	APS	APM	APL
two-stage detectors								
Faster R-CNN	VGG-16	trainval	21.9	42.7	-	-	-	-
DeRPN	VGG-16	trainval	25.5	47.3	25.4	9.2	26.9	38.3
R-FCN	ResNet-101	trainval	29.9	51.9	-	10.8	32.8	45.0
CoupleNet	ResNet-101	trainval	34.4	54.8	37.2	13.4	38.1	50.8
Deformable R-FCN	Aligned-Inception-ResNet	trainval	37.5	58.0	40.8	19.4	40.1	52.5
one-stage detectors								
YOLOv2	Darknet-19	trainval35k	21.6	44.0	19.2	5.0	22.4	35.5
DSSD321	ResNet-101	trainval35k	28.0	46.1	29.2	7.4	28.1	47.6
RFB-Net300	VGG-16	trainval35k	30.3	49.3	31.8	11.8	31.9	45.9
DSSD513	ResNet-101	trainval35k	33.2	53.3	35.2	13.0	35.4	51.1
YOLOv3 608	Darknet-53	trainval35k	33.0	57.9	34.4	18.3	35.4	41.9
RFB-Net512	VGG-16	trainval35k	33.8	54.2	35.9	16.2	37.1	47.4
RetinaNet500	ResNet-101	trainval35k	34.4	53.1	36.8	14.7	38.5	49.1
DFPR-Net512	ResNet-101	trainval35k	34.6	54.3	37.3	14.7	38.1	51.9
RefineDet512	ResNet-101	trainval35k	36.4	57.5	39.5	16.6	39.9	51.4
DR-Net512	ResNet-101	trainval35k	39.3	59.8	-	21.7	43.7	50.9
ours								
Dubox320	ResNet-101	trainval35k	31.8	53.36	34.01	17.79	35.89	42.87
Dubox512	ResNet-101	trainval35k	35.32	54.75	37.63	18.94	38.80	45.65
Dubox800	ResNet-101	trainval35k	38.03	56.31	41.7	19.01	40.60	49.23
Dubox800(multi-scale)	ResNet-101	trainval35k	**39.52**	**57.31**	**42.18**	**20.94**	**41.80**	**50.86**

the results on $test\text{-}dev$ set. We set the batch size to 16(320×320), 8(512×512), 4(800×800) for each GPU in training, and train the model with 10^{-3} learning rate for the first 280k iterations, then 10^{-4} and 10^{-5} for another $180k$ and $140k$ iterations, respectively.

As show in Tab.4 on MS COCO test set. Dubox320 with ResNet-101 produces **31.8%** AP that is better than other two-stage methods (e.g., Faster R-CNN[21], DeRPN[28]). The accuracy of Dubox can be improved to **35.32%** using 512×512 input size. When the input size is 800 × 800, Dubox gets **38.03%**AP which is much better than several one-stage object detectors (e.g., SSD[16] and YOLOv2[19], YOLOv3[20], RetinaNet[13], RFB-Net[15], DFPR-Net[8], DR-Net[29].). With multi-scale input, Dubox800 get **39.52%**AP.

4.4. Inference time Performance

With the help of no-prior box detection, Dubox only uses 4 channel feature maps to represent the target boxes, while the anchor-based methods have to use $4A$ (A is the number of anchors) feature maps to regress the boxes. The less output feature maps and less scale detectors accelerate the inference speed of Dubox. Furthermore, the architecture of proposed Dubox only contains convolution, deconvolution layer and element-wise activation functions which are highly optimized in common deep learning frameworks, making it easy to deploy in resource limited platforms, such as mobile phone and autopliot systems.

We present the inference speed of Dubox and the state-of-the-art methods in the Tab.3. The speed is evaluated with batch size 1 on a machine with NVIDIA Titan X, CUDA 8.0 and cuDNN v6. As shown in Tab.3, we find that Dubox processes an image in **20.0**ms (**50** FPS), **55.4**ms (**18** FPS) and **142**ms (**7** FPS) with input sizes 320×320, 512×512 and 800×800 respectively. To the best of our knowledge, Dubox is the first getting **50** FPS real-time method to achieve detection accuracy above **79.31%** mAP on PASCAL VOC 2007. In summary, Dubox achieves the best trade-off between accuracy and speed.

5. Conclusion

Anchor-based method is not the only choice in object detection. Dubox, as an no-prior box method, also can work effectively using the proper regression loss and network architecture. At the same time, Dubox further considered the problem of multi-scale detection to enhance the capacity of heuristic-guided feature selection. The proposed dual scale residual unit enables multi-scale detectors not to operate independently, but make the high-level learning from the low one. These strategies improve the performance of the detection while significantly reducing the redundancy of various scale detectors.

References

[1] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. 7

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *international Conference on computer vision & Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE Computer Society, 2005. 2

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 1, 2

[4] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017. 7

[5] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010. 6

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 3, 6

[7] L. Huang, Y. Yang, Y. Deng, and Y. Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015. 2

[8] T. Kong, F. Sun, C. Tan, H. Liu, and W. Huang. Deep feature pyramid reconfiguration for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 169–185, 2018. 7, 8

[9] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018. 2

[10] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5325–5334, 2015. 2

[11] X. Liang, T. Wang, L. Yang, and E. Xing. Cirl: Controllable imitative reinforcement learning for vision-based self-driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 584–599, 2018. 1

[12] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 1, 3

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 1, 2, 8

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 2, 7

[15] S. Liu, D. Huang, et al. Receptive field block net for accurate and fast object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 385–400, 2018. 8

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1, 3, 5, 8

[17] D. G. Lowe. Object recognition from local scale-invariant features. In *iccv*, page 1150. Ieee, 1999. 2

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2, 7

[19] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1, 7, 8

[20] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1, 2, 8

[21] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 3, 6, 7, 8

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 6

[23] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 761–769, 2016. 6

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 3, 6

[25] X. Tang, D. K. Du, Z. He, and J. Liu. Pyramidbox: A context-assisted single shot face detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 797–813, 2018. 1

[26] J. Wang, Y. Yuan, G. Yu, and S. Jian. Sface: An efficient network for face detection in large scale variations. *arXiv preprint arXiv:1804.06559*, 2018. 4

[27] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon. Cbam: Convolutional block attention module. In *The European Conference on Computer Vision (ECCV)*, September 2018. 3

[28] L. Xie, Y. Liu, L. Jin, and Z. Xie. Derpn: Taking a further step toward more general object detection. *arXiv preprint arXiv:1811.06700*, 2018. 7, 8

[29] H. Xu, X. Lv, X. Wang, Z. Ren, N. Bodla, and R. Chellappa. Deep regionlets for object detection. In *The European Conference on Computer Vision (ECCV)*, September 2018. 7, 8

[30] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4203–4212, 2018. 5, 7

[31] Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, and M. Li. Bag of freebies for training object detection neural networks. *arXiv preprint arXiv:1902.04103*, 2019. 7