

DRPN: Making CNN Dynamically Handle Scale Variation

Jingchao Peng, Haitao Zhao, Zhengwei Hu, Kaijie Zhao, and Zhongze Wang

East China University of Science and Technology,
Automation Department, School of Information Science and Engineering

Abstract. Based on our observations of infrared targets, serious scale variation along within sequence frames has high-frequently occurred. In this paper, we propose a dynamic re-parameterization network (DRPN) to deal with the scale variation and balance the detection precision between small targets and large targets in infrared datasets. DRPN adopts the multiple branches with different sizes of convolution kernels and the dynamic convolution strategy. Multiple branches with different sizes of convolution kernels have different sizes of receptive fields. Dynamic convolution strategy makes DRPN adaptively weight multiple branches. DRPN can dynamically adjust the receptive field according to the scale variation of the target. Besides, in order to maintain effective inference in the test phase, the multi-branch structure is further converted to a single-branch structure via the re-parameterization technique after training. Extensive experiments on FLIR, KAIST, and InfraPlane datasets demonstrate the effectiveness of our proposed DRPN. The experimental results show that detectors using the proposed DRPN as the basic structure rather than SKNet or TridentNet obtained the best performances.

Keywords: Object detection, Infrared target detection, Scale variation, Dynamic convolution, Re-parameterization technique.

1 Introduction

Visual target detection is an essential task for many intelligent systems. Over the past decade, with the development of deep learning, target detection has achieved great success. Among them, the detection at degraded visibility conditions or at a great distance is a challenging problem [1,2,3,4,5]. On the contrary, thermal infrared cameras detect the thermal radiation of targets and convert it into images, which are not influenced by the changing of the visibility conditions. Furthermore, thermal infrared cameras have the ultra-long-distance detection capability. Very distant targets can still be recorded in infrared images. Due to the good properties of thermal infrared imaging, infrared target detection methods have been widely studied [6,7,3,8,4,9,5,10,11].

The existing infrared target detection methods usually focus on the detection of small targets, which mostly rely on the fine-resolution prediction map [6,7]. For instance, drop the low-resolution layers and enhance the high-resolution layer

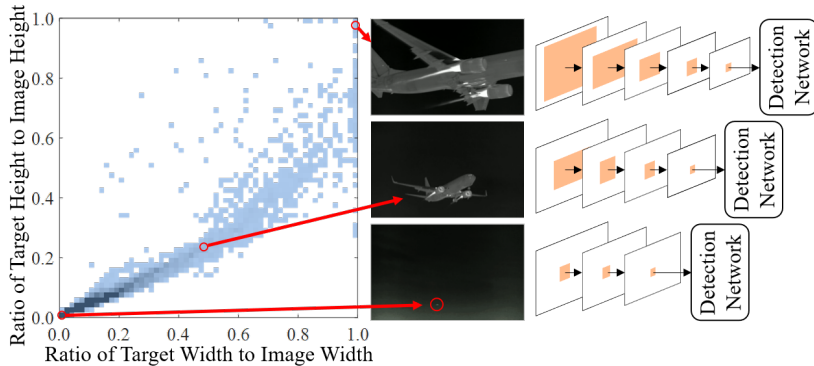


Fig. 1: The statistical observation of scale variation in infrared airplane datasets. The area proportion of the small target is less than 0.1%; The large target is almost full of the whole image. When detecting large targets, the receptive field needs to be large, while detecting small targets, the receptive field needs to be small.

to make networks more suitable for the detection task of infrared small targets [3,8]; use the dense connection to keep the high-resolution feature map in the previous channels [4,9]; use multimodal feature maps like combining with the visible spectrum sensor to increase resolution [5,10,11]. However, these methods pay less attention to scale variation which has high-frequently occurred in infrared datasets.

When detecting large-size targets, the feature map of the detection networks usually needs to contain semantic information to enhance the detection precision. a large receptive field after deep convolutions often needs to match the size of the target [12]. However, when detecting small targets, the feature map needs to have detailed information for positioning, a small receptive field should be utilized to prevent target information loss [13,14]. While the large receptive field (and high-level semantic understanding) is often in conflict with the small receptive field (and fine-resolution prediction maps) since deep networks learn more semantic representations by gradually attenuating the size of the feature maps [15]. Besides, as thermal infrared images have ultra-long-distance detection capabilities, this contradiction is further exacerbated. For instance, in the visible light target detection datasets such as COCO, the size of small targets is usually less than 32×32 , the size of large targets is greater than 96×96 , and the size of medium targets is between them. While in the infrared datasets, the small target is usually less than 9×9 , and the large target is more than 256×256 . The scale variation in infrared images is much more serious than that of visible images. To this end, we conduct statistical analysis about scale variation on a large infrared target detection dataset.

We count infrared targets according to the ratio of target size to image size, as shown in Fig. 1. Where the horizontal and vertical coordinates in the figure represent the ratio of target width to image width and the ratio of target height

to image height, respectively. Each blue point represents a target. When the image size is 300×300 , the size of the smallest target is 9×5 , the area proportion is less than 0.1%. The size of the biggest target is 296×292 , which is almost full of the whole image. It can be seen from the figure that the scale variation of infrared targets along with sequence frames is very serious. Large scale variation brings the challenge to infrared target detection. When detecting large targets, the network needs to be deep to obtain large receptive fields and high-level semantic information; when detecting small targets, the network needs to be shallow, so as not to reduce the resolution of the prediction map and lose the target information.

Just motivated by the above observations, in this paper, we propose a dynamic re-parameterization network (DRPN) to deal with the scale variation and balance the detection precision of small targets and large targets. DRPN adopts a multi-branch structure, in which every branch has different receptive fields. Besides, DRPN uses the strategy of dynamic convolution, and automatically generates weights according to the input, so that the DRPN can automatically weight different branches to automatically adjust the receptive field according to the changing of the target size. In this way, DRPN solves the problem of network design contradiction caused by scale variation in infrared target detection. Multi-branch structure brings more memory size and computation. In order to maintain efficient inference after training, DRPN adopts re-parameterization technology, which converts the architecture from multi-branch to single-branch via transforming convolution kernels of different sizes to convolution kernels of the same size in the inference phase. Extensive experiments demonstrate that our DRPN is superior to the baseline as well as other methods.

In summary, our contributions are summarized below:

1. By combining multi-branch structure and dynamic convolution strategy, DRPN can automatically adjust the receptive field according to the scale variation of targets.
2. DRPN adopts multi-branch structure to increase the model capacity while maintaining effective inference through re-parameterization technology.
3. The effectiveness of DRPN is demonstrated on FLIR, KAIST, and our In-fraPlane dataset. Detectors using the proposed DRPN as the basic structure rather than SKNet or TridentNet to replace standard convolution obtained the best results.

2 Related Work

2.1 Visible and Infrared Target Detection

Deep-learning-based visible target detection methods can be classified as anchor-based methods and anchor-free methods. The anchor-based methods regard the target detection problem as the classification and regression of candidate regions. The candidate region here is anchor, which is generated by sliding windows in

one-stage methods, such as SSD [16] and YOLO [17]; or generated by region proposal network (RPN) in two-stage methods, such as Faster-RCNN [18], Cascade-RCNN [19], and so on. In contrast to anchor-based architectures that rely on anchors for the localization of targets by introducing a lot of hyperparameters in the model, anchor-free architectures predict the key points or borders of the target directly, including CornerNet [20], FCOS [21], CenterNet [22], to name a few. However, these methods are designed mainly for medium and large targets, have no special design for large scale variation in infrared target detection.

Deep-learning-based infrared target detection methods mostly aim at dim-small target detection. Ding et al. [8] improved the network architecture of SSD for infrared small target detection. Since high-level semantic layers can hardly extract any feature of dim-small targets, and these layers make no contribution to the detection of infrared small targets, Ding et al. first remove high-level semantic layers, and then enhance the low-level layer by dilated convolution. Dilated convolution has the advantage of improving the receptive field without reducing the resolution of the feature maps. Dai et al. [23] propose an asymmetric contextual modulation, with particular emphasis on exchanging high-level semantics and subtle low-level details. They adopt the global channel attention modulation to propagate high-level semantic information down to shallow layers, whereas utilize pixel-wise channel attention modulation to preserve and highlight infrared small targets in high-level features. However, both of these two methods are designed for infrared small target detection and pay less attention to the scale variation problem.

2.2 Methods for Handling Scale Variation

There have also been methods considering the scale variation problem. The multi-scale image pyramid [24] is a common scheme, but it will bring a heavy computation burden. Instead of taking multiple images as input, SSD [16] utilizes multi-level features of different spatial resolutions to alleviate scale variation. FPN [25] further introduces a top-down pathway and lateral connections to enhance the semantic representation of low-level features at the bottom layers. SEPC [12] captures the inter-scale interactions through an explicit convolution in the scale dimension, forming a 3-D convolution in the feature pyramid. These methods study how to make better use of high-level semantic and fine-resolution features from the way of information flow. Experiments demonstrate the effectiveness of these methods.

Other methods [26,27] try to change the receptive field of the neural networks by the idea of multi-branch, which can also deal with the scale variation problem. SKNet [26] uses SoftMax attention to fuse multiple branches with different kernel sizes. Different attentions on these branches yield different sizes of the receptive fields. However, multiple branches represent multiple convolutions, which largely increases the computational burden. ACNet [28] and RepVGG [29] use the re-parameterization technique to reduce inference time. Specifically, the training-time model has a multi-branch structure, while the test-time model converts the architecture from multi-branch to single-branch via re-parameterization. But

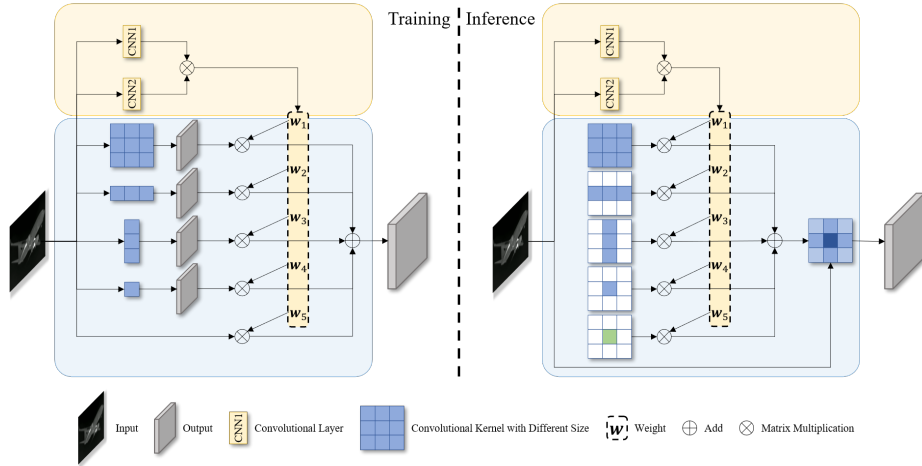


Fig. 2: Illustration of the proposed DRPN. DRPN consists of two modules, one is the weight generation module, which is shown in the yellow box; the other is the multi-branch module, which is shown in the blue box. The training phase of the multi-branch module is different from the inference phase. In the training phase, the input tensor is convoluted first, and then the outputs is weighted and summed; In the inference phase, the convolution kernel is weighted and summed first, and then the input tensor is convoluted.

they use the same weights when convert multi-branch to single-branch, which is not suitable to scale variation problem. TridentNet [27] generates multiple parallel branches with different dilation rates, thus endowing the same representational power for targets of different scales. However, these methods have no consideration of the dynamic design for the multiple branches. During the test time, these methods cannot adjust adaptively to handle the significant scale variation in infrared target detection.

3 Methodology

In this section, we first overview the architecture of dynamic re-parameterization network (DRPN) for infrared target detection. Then we will introduce the two modules of DRPN: the weight generation module and the multi-branch module.

3.1 Overall Structure

The overall network architecture of our proposed DRPN is shown in Fig. 2. DRPN aims to solve the scale variation problem in infrared target detection. DRPN adopts a multi-branch structure, including four convolution branches whose kernels with different sizes and a shortcut branch. The convolution kernels with different sizes have different receptive fields. Meanwhile, the multi-layer structure of the CNN can further enhance the difference. In order to enable

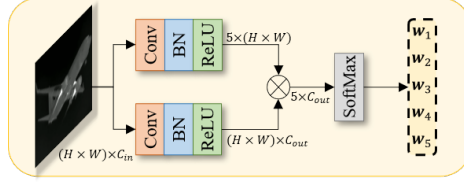


Fig. 3: Illustration of weight generation module, which is used to calculate attention matrices suitable for the size of the target in the infrared image.

CNNs to dynamically adjust the receptive field according to the variation of the target size, DRPN uses the structure of dynamic convolution, and adaptively generates weights according to the target size of the input.

DRPN consists of two modules, one is the weight generation module, the other is the multi-branch module. The weight generation module generates five weights from the input, corresponding to five branches of the multi-branch module. During the training phase, each branch of the multi-branch module convolves the input tensor respectively, and the obtained five output tensors are weighted and summed according to the generated weights to obtain the final output tensor. In the inference phase, the convolution kernels of five branches, including four convolution branches and a shortcut branch, are unified into the size of 3×3 . Then these five convolution kernels weigh and sum according to generated weights to obtain the final convolution kernel. Finally, the final convolution kernel is convoluted with the input tensor to obtain the output tensor.

3.2 Weight Generation Module

Given an input tensor $X_{in} \in \mathbb{R}^{H \times W \times C_{in}}$, and an output tensor $X_{out} \in \mathbb{R}^{H \times W \times C_{out}}$, the weight generation module first uses two convolution layers to encode the input tensor into query tensor $Q \in \mathbb{R}^{(H \times W) \times 5}$ and key tensor $K \in \mathbb{R}^{(H \times W) \times C_{out}}$:

$$Q = f_1(X_{in}) \quad (1)$$

$$K = f_2(X_{in}), \quad (2)$$

where f_1 and f_2 are two different convolutional layers. To conveniently define the matrix computation, we stack spatial positions in a row-by-row way: $[X] : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{N \times C}$, where $N = H \times W$ is the number of vertices. Then the attention matrix $W = [\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4, \mathbf{w}_5]^T \in \mathbb{R}^{5 \times C_{out}}$ is computed by the inner product of the query tensor and the key tensor, which can be formulated as:

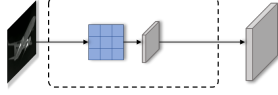
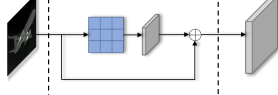
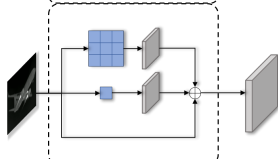

$$W = [\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4, \mathbf{w}_5]^T = S([Q]^T \times [K]) \quad (3)$$

where \times denote matrix multiplication, and $S(\cdot)$ is the SoftMax operation. The weight generation module is shown in Fig. 3.

3.3 Multi-branch Module

Inspired by the re-parameterization technique [28,29] and dynamic convolution [30,31,32,33], we construct the multi-branch module by simply replacing every

Table 1: Classic networks and corresponding special cases of DRPN.

The Convolutional Layer of Classic Network	Weights of DRPN	Structure of DRPN
VGG	$\mathbf{w}_1 = \mathbf{1},$ $\mathbf{w}_2 = \mathbf{w}_3 = \mathbf{w}_4 = \mathbf{w}_5 = \mathbf{0}$	
ResNet	$\mathbf{w}_1 = \mathbf{w}_5 = \frac{1}{2} \times \mathbf{1},$ $\mathbf{w}_2 = \mathbf{w}_3 = \mathbf{w}_4 = \mathbf{0}$	
RepVGG	$\mathbf{w}_1 = \mathbf{w}_4 = \mathbf{w}_5 = \frac{1}{3} \times \mathbf{1},$ $\mathbf{w}_2 = \mathbf{w}_3 = \mathbf{0}$	
Lightweight Network	$\mathbf{w}_5 = \mathbf{1},$ $\mathbf{w}_1 = \mathbf{w}_2 = \mathbf{w}_3 = \mathbf{w}_4 = \mathbf{0}$	

3×3 layer with five parallel layers with kernel size 3×3 , 1×3 , 3×1 , 1×1 , and a shortcut branch (only if the dimensions of input tensor and output tensor match) to model the information flow respectively. Since multi-branch architecture makes one convolution an implicit ensemble of five convolutions, it is beneficial to training [34] and has more representation power [31]. In addition, as will be shown in Sec. 4.5, because convolution kernels of different sizes have different receptive fields, the module can adjust the receptive field according to the changing of the target size. For the reason that the high computational density of 3×3 convolution [29] and the receptive field of existing networks is large enough to detect large targets, we do not use 5×5 and 7×7 convolutions which are used in SKNet [26] instead.

At the training phase, first, each branch convolutes the input tensor respectively, and then the obtained five output tensors are weighted and summed:

$$\begin{aligned}
 X_{out} = & \mathbf{w}_1 \times_3 \mathcal{F}_{3 \times 3}(X_{in}) + \mathbf{w}_2 \times_3 \mathcal{F}_{1 \times 3}(X_{in}) \\
 & + \mathbf{w}_3 \times_3 \mathcal{F}_{3 \times 1}(X_{in}) + \mathbf{w}_4 \times_3 \mathcal{F}_{1 \times 1}(X_{in}) + \mathbf{w}_5 \times_3 X_{in}, \quad (4)
 \end{aligned}$$

where $\mathcal{F}_{3 \times 3}$, $\mathcal{F}_{1 \times 3}$, $\mathcal{F}_{3 \times 1}$, and $\mathcal{F}_{1 \times 1}$ are the convolution layers of the corresponding kernel with the size of 3×3 , 1×3 , 3×1 , and 1×1 , respectively; \times_n refers to n -mode multiplication [35], e.g. $\mathbb{R}^{i \times j} \times_3 \mathbb{R}^{k \times h \times j} \rightarrow \mathbb{R}^{k \times h \times i}$. We train the multi-branch module using the same configurations as the original convolution layers without tuning any hyper-parameters.

When the training is completed, we convert the multi-branch to a single convolutional layer through padding zeros outside the 3×1 , 1×3 , and 1×1 convolution kernels and making them 3×3 kernels, respectively. This transformation also applies to the shortcut branch because it can be viewed as a 1×1 convolution layer with an identity matrix as the kernel, more details can be referred in [29]. Then we obtain the final convolution kernel k_{final} by adding up

the five convolution kernels:

$$k_{final} = \mathbf{w}_1 \times_1 k_{3 \times 3} + \mathbf{w}_2 \times_1 k_{1 \times 3} + \mathbf{w}_3 \times_1 k_{3 \times 1} + \mathbf{w}_4 \times_1 k_{1 \times 1} + \mathbf{w}_5 \times_1 k_{id}, \quad (5)$$

where $k_{3 \times 3}$, $k_{1 \times 3}$, $k_{3 \times 1}$, $k_{1 \times 1}$, and k_{id} are the kernels of the corresponding filter at the 3×3 , 1×3 , 3×1 , 1×1 convolution layers, and shortcut layers, respectively. Finally, the output tensor is obtained by convolution with the final kernel:

$$\tilde{X}_{out} = \mathcal{F}_{final}(X_{in}), \quad (6)$$

where \mathcal{F}_{final} is the convolution layer of the final convolution kernel k_{final} . The principle of this conversion is that the output from convolution with several convolution kernels which have different sizes is equal to the output obtained from an equivalent kernel convoluted with the same input, i.e., $X_{out} = \tilde{X}_{out}$.

With the weight generation module and the multi-branch module, DRPN can evolve many classical convolutional layers through the combination of different weights and different branches, as shown in Tab. 1. Let $\mathbf{1} \in \mathbb{R}^{C_{out}}$ which denote a column vector where each element is equal to one, and $\mathbf{0} \in \mathbb{R}^{C_{out}}$ which denote a column vector where each element is equal to zero. When $\mathbf{w}_1 = \mathbf{1}$, $\mathbf{w}_2 = \mathbf{w}_3 = \mathbf{w}_4 = \mathbf{w}_5 = \mathbf{0}$, DRPN can be regarded as the convolutional layer of VGG [36]; when $\mathbf{w}_1 = \mathbf{w}_5 = 1/2 \times \mathbf{1}$, $\mathbf{w}_2 = \mathbf{w}_3 = \mathbf{w}_4 = \mathbf{0}$, DRPN can be regarded as the convolutional layer of ResNet [15]; when $\mathbf{w}_1 = \mathbf{w}_4 = \mathbf{w}_5 = 1/3 \times \mathbf{1}$, $\mathbf{w}_2 = \mathbf{w}_3 = \mathbf{0}$, DRPN can be regarded as the convolutional layer of RepVGG [29]; when $\mathbf{w}_5 = \mathbf{1}$, $\mathbf{w}_1 = \mathbf{w}_2 = \mathbf{w}_3 = \mathbf{w}_4 = \mathbf{0}$, DRPN can be regarded as a simplified lightweight convolutional layer mainly designed to detect small targets. If the weight parameters \mathbf{w}_1 , \mathbf{w}_2 , \mathbf{w}_3 , \mathbf{w}_4 , and \mathbf{w}_5 are dynamically computed according to the changing of the target size, the multi-branch module should be adaptable to the scale variation of the target.

4 Experiments

In this section, we first describe the datasets and implementation details of DRPN and training settings in Sec. 4.1. Next, Sec. 4.2 compares the results of DRPN with other methods on three datasets, and Sec. 4.3 show the portability and generalizability of DRPN. Then, in Sec. 4.4 We conduct thorough ablation experiments to demonstrate the effectiveness of the proposed method. Finally, qualitative performances give the visual comparisons of our proposed DRPN in Sec. 4.5.

4.1 Datasets and Implementation Details

The methods compared in this section are all pre-trained on the COCO dataset [37]. COCO is a large and rich visible image dataset for object detection and segmentation, which has 330K images, 80 object categories, and 5 labels per image. FLIR and KAIST [38] datasets are commonly used open-source datasets for infrared target detection. The FLIR dataset contains 14K images with 10K from short video segments and random image samples, plus 4K images from

Table 2: Comparisons of different target detection methods which were evaluated on FLIR and KAIST datasets.

Dataset	Method	AP	AP50	APs	APm	APl
FLIR	SSD	21.1	49.8	11.0	27.9	36.4
	SKNet	22.3	51.8	10.9	29.2	39.8
	TridentNet	21.6	51.2	10.7	27.2	40.1
	DRPN(Ours)	22.9	52.2	11.4	29.1	38.2
KAIST	SSD	29.8	63.5	-	29.5	42.3
	SKNet	25.8	57.8	-	25.4	35.9
	TridentNet	29.7	60.6	-	28.7	43.1
	DRPN(Ours)	32.0	63.4	-	31.0	45.8
InfraPlane	SSD	37.4	79.6	33.6	63.5	81.2
	SKNet	39.1	81.6	35.0	66.1	83.5
	TridentNet	37.1	80.1	33.3	62.0	85.0
	DRPN(Ours)	40.0	82.7	36.2	64.3	85.2

a 140-second video. The KAIST dataset contains a total of 90K images, each containing both RGB color and infrared versions. It contains a total of 100K dense annotations. The dataset captured various conventional traffic scenes including campus, streets, and rural areas during the daytime and at night. The InfraPlane dataset contains 30,211 infrared images taken by infrared and hyper-spectral cameras, in which airplanes take off and land near the airport during the day and night. There are 20,177 images in the daytime and 10,034 images in the night. The resolution of the image is 480×300 . The examples of the InfraPlane dataset are shown in Sec. 4.5.

We implement our proposed DRPN on the PyTorch platform with I5-10700K @5.0GHz CPU and NVIDIA TITAN RTX GPU. We adopt SSD [16] as our baseline method in mmdetection [39]. We replace the corresponding convolution in conv1_1 ~ conv5_3 with our proposed DRPN. The input images are resized to 300×300 . Random crop and random flip are adopted during training. By default, the batch size is 8, and 24 epochs are trained with the learning rate starting from 0.01 and decreasing by a factor of 0.1 after the 16th and 22th epoch. For the evaluation, we report the standard COCO evaluation metric of Average Precision (AP) as well as AP₅₀, AP_s, AP_m, and AP_l.

4.2 Comparison with Other Methods

We evaluate DRPN on three datasets and compare DRPN with baseline (SSD [16]), SKNet [26], and TridentNet [27]. DRPN improves significantly over other methods. Here we report the results in Tab. 2. It can be found that DRPN achieves the best performance in each dataset. Specifically, DRPN achieves 22.9, 32.0, and 40.0 AP, which is 1.8, 2.2, and 2.6 higher than the baseline (SSD). To compare with SKNet and TridentNet, we adopt the same training settings and network architecture with DRPN, just the corresponding DRPN is replaced by SKNet or TridentNet. DRPN improves significantly over other methods, which improves 1.8, 2.2, and 0.9 AP than the second-best method on three datasets,

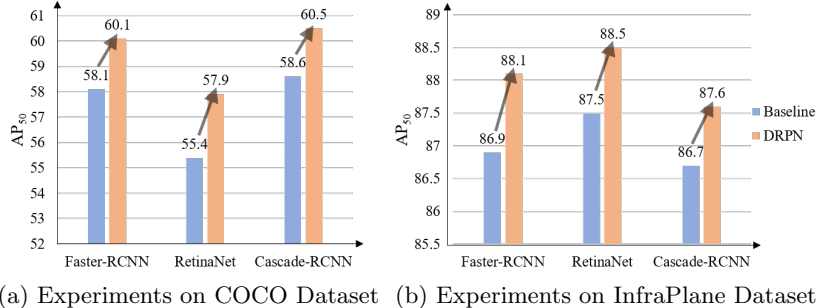


Fig. 4: Experiments on COCO and InfraPlane Datasets using other detection frameworks.

respectively. It shows the effectiveness of dynamic network architecture generated by DRPN with the same set of parameters.

Furthermore, DRPN achieves the performance with 59.55 GFLOPs computation. By contrast, the computations of SSD, SKNet, and TridentNet are 34.42, 263.41, and 34.42 GFLOPs, respectively. The computation of DRPN is 25.13 GFLOPs higher than that of baseline (SSD). This computation cost comes mainly from the weight generation module, which calculates the weights of different convolution kernels according to the size of the target in the input images. Nevertheless, given the performance of DRPN, this computation cost is acceptable.

Compared with the KAIST dataset, the FLIR and the InfraPlane dataset have the problem of serious scale variation. SKNet adopts the multi-branch structure both in the training phase and the test phase. Although the SKNet is more complex, the performance is better than TridentNet. On the contrary, the target size in the KAIST dataset is relatively large compared with the other two datasets, and there is no small target. The dilation rate and the receptive field of TridentNet are larger than SKNet, therefore TridentNet is more suitable for the KAIST dataset and has a better performance. While our proposed DRPN outperforms than SKNet and TridentNet on three datasets, indicating the effectiveness of the proposed DRPN.

4.3 Portability and Generalizability

DRPN has a good portability and good Generalizability. That says DRPN can easily be implemented on other detection frameworks and adapted to other datasets. Here we implement DRPN on three different detection frameworks including Faster-RCNN, RetinaNet, and Cascade-RCNN as an illustration to demonstrate the good portability, and use two datasets containing COCO and InfraPlane datasets to show the generalization ability of DRPN to visible and infrared images. We all use ResNet-50 as the backbone. The results are shown in Fig. 4, which can be shown from the figure that in different frameworks and

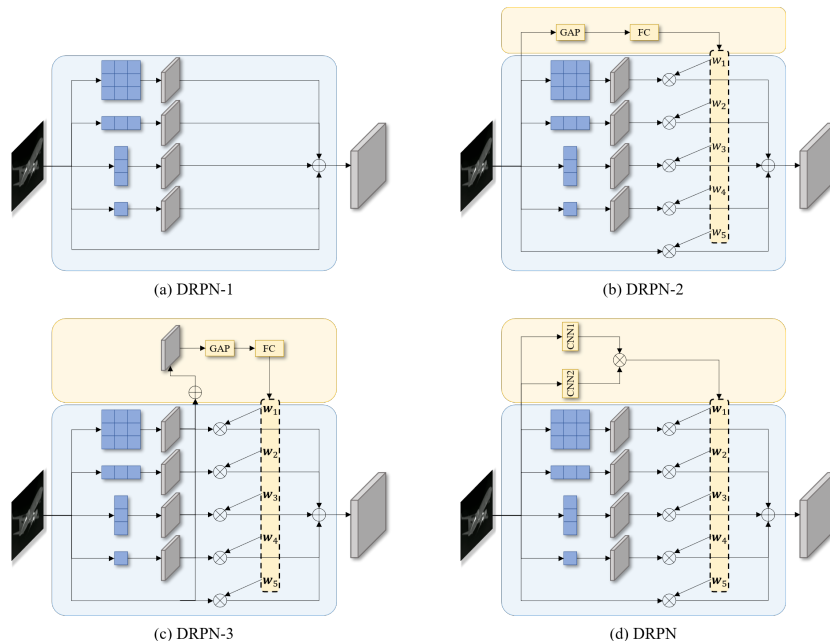


Fig. 5: Network structure diagram in ablation studies. The difference is mainly that the weight generation modules are different. DRPN-1 has no weight generation module, DRPN-2 adopts the weight generation module of dynamic convolution, and DRPN-3 adopts the weight generation module of SKNet.

datasets, the performances of DRPN are better than those of the baselines (origin Faster-RCNN, origin RetinaNet, or origin Cascade-RCNN). In specific, when using Faster-RCNN as the detection framework, DRPN acquires 1.9 and 1.2 AP higher than baseline (origin Faster-RCNN) in COCO and InfraPlane datasets, respectively. When using RetinaNet as the detection framework, the increased detection rates are 2.5 and 1.0 AP, and when using Cascade-RCNN as the detection framework, the two numbers are 1.9 and 0.9 AP.

4.4 Ablation Studies

In order to validate the effectiveness of our major contributions, we implement two variations, including 1) pruning multi-branch module; 2) pruning weight generation module, or changing the weight generation module. The comparison results on InfraPlane dataset are shown in Tab. 3. Where SSD-vanilla represents the original SSD [16] detector without bells and whistles. DRPN-1 represents an SSD detector with the multi-branch module, which is shown in Fig. 5(a). DRPN-2 represents SSD detector with multi-branch module and weight generation module, and the weight generation module uses the corresponding module in dynamic convolution [31], which is shown in Fig. 5(b). DRPN-3 represents SSD detector with multi-branch module and weight generation module, but the weight generation module uses the corresponding module in SKNet [26], which

Table 3: Results on the infrared airplane dataset. Starting from the baseline (SSD-vanilla), we gradually add the multi-branch module, add and change the weight generation module in our SSD for ablation studies.

Method	Multi-branch Module	Weight Generation Module	AP	AP50	APs	APm	API
SSD-vanilla	-	-	37.4	79.6	33.6	63.5	81.2
DRPN-1	✓	-	37.8	77.4	32.9	62.9	82.5
DRPN-2	✓	Dynamic Conv	35.6	75.2	31.5	63.1	88.4
DRPN-3	✓	SKNet	37.2	77.6	33.6	61.0	80.4
DRPN	✓	uncompressed attention	39.5	81.5	35.8	65.1	83.3

Table 4: The differences between weight generation methods.

Weight Generation Module	Compression Method	Weight Dimension	Computation Order	Number of Convolution
dynamic convolution [31]	global average pooling	$W^{DC} \in \mathbb{R}^{5 \times 1}$	First generate weights, then preform convolution.	1
SKNet [26]	global average pooling	$W^{SK} \in \mathbb{R}^{5 \times C_{out}}$	First preform convolution, then generate weights.	5
DRPN (Ours)	uncompressed attention	$W \in \mathbb{R}^{5 \times C_{out}}$	First generate weights, then preform convolution.	3

is shown in Fig. 5(c). DRPN represents our proposed DRPN introduced in Sec. 3, as shown in Fig. 5(d).

The first, second, and fifth rows in Tab. 3 show that both modules of DRPN can improve detection performance. Specifically, the multi-branch module can increase 0.2 AP compared with SSD-vanilla. Furthermore, the weight generation module can improve 1.7 AP on the basis of the multi-branch module. The third, fourth, and fifth rows of Tab. 3 show the detection results when using different weight generation module methods. Our DRPN obtained the best detection results.

In both DRPN-2 and DRPN-3, the three-dimensional tensor is compressed into a one-dimensional vector by global average pooling, and then the weights are generated by full connection layers. In DRPN-2, dynamic convolution is adopted to compress the input features, while in DRPN-3, SKNet is utilized to compress the output features. The other difference is that the weights generated by dynamic convolution be $W^{DC} = [w_1^{DC}, w_2^{DC}, w_3^{DC}, w_4^{DC}, w_5^{DC}]^T \in \mathbb{R}^{5 \times 1}$, i.e., each weight $w_i^{DC} \in \mathbb{R} (i \in 1, 2, 3, 4, 5)$ is a scalar, while the weights generated by SKNet be $W^{SK} = [w_1^{SK}, w_2^{SK}, w_3^{SK}, w_4^{SK}, w_5^{SK}]^T \in \mathbb{R}^{5 \times C_{out}}$, i.e., each weight $w_i^{SK} \in \mathbb{R}^{C_{out}} (i \in 1, 2, 3, 4, 5)$ is a column vector whose dimension is equal to the number of output channels C_{out} . These two modules are essentially for channel weighting, which focus on the contribution of different channels. The weight generation module of our DRPN is better than that of the two methods because the weight generation module of DRPN is essentially an uncompressed attention method, in which the feature map is not operated by global average pooling. Therefore the weight generation module of our DRPN focuses on the relationship between pixels, which has the ability to dynamically compute the weights for different convolutional kernels according to the target size.

In terms of computation order, SKNet first performs convolution, then calculates the weight based on the output tensor, and finally weights the out-

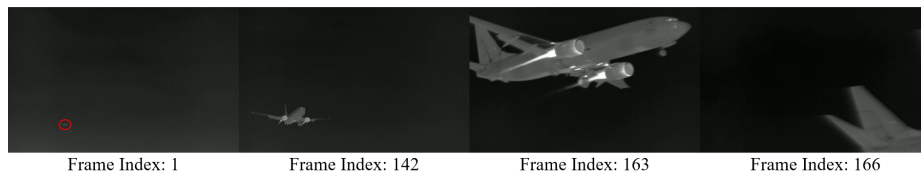


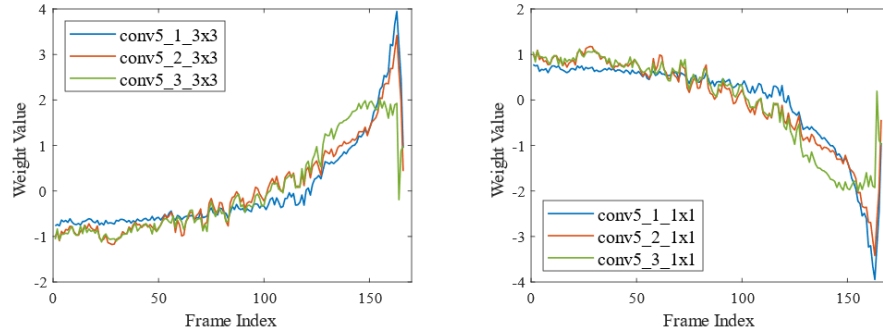
Fig. 6: Samples of the infrared image sequence in InfraPlane datasets. In the sequence, the airplane is from far to near, and the size of the target (airplane) is from small to large.

put tensor, whose computational burden is large. Both dynamic convolution and our DRPN first calculate the weight according to the input tensor, then weight the convolution kernels, and finally perform convolution. When in the inference phase, DRPN-2 adopted dynamic convolution because the weight is calculated first, the multi-branch can be transformed into a single-branch by re-parameterization technique, it only needs one convolution calculation. DRPN-3 adopted SKNet because convolution is performed first, the multi-branch cannot be transformed into a single-branch, it needs five convolution calculations whose computational burden is large. DRPN needs three convolution calculations (two from attention, one from single-branch), which is a compromise between DRPN-2 and DRPN-3, it can balance detection precision and inference speed. In this way DRPN can achieve effective inference. In summary, their differences are shown in Tab. 4.

4.5 Qualitative Performances

In order to show the effectiveness of DRPN in dynamically adjusting the receptive field under the scale variation, we selected a sequence of infrared images that contains an airplane flying from far to near, then disappearing outside the image as shown in Fig. 6. The size of this airplane changes from small to large in this sequence. The image size is 480×300 , the smallest target size is 5×15 , and the largest target size is 461×155 .

According to the experiments of Yang et al. [30], the distribution of the weights is very similar across classes at early layers in the network, and become more and more different at later layers. Therefore, we select the weights of conv5.1 ~ conv5.3 in the SSD. Fig. 7 to show the changing of the weights of different branches in the inference of this sequence. It can be found from Fig. 7(a) that the weights of the large convolution kernels, such as 3×3 convolution kernel, become larger with the increase of target size. The reason is that the 3×3 convolution kernel has a relatively bigger receptive field, which is more suitable for detecting large targets. While in Fig. 7(b) the weights of the small convolution kernels decrease, that is because the small convolution kernels, such as 1×1 or shortcut branch, are more suitable for small target detection. When the airplane finally disappears out of the image, all the weights of all big convolution kernels and small convolution kernels are close to zero. It means that the output should approach zero if the target disappears.



(a) The mean weights of 3×3 convolution kernel. (b) The mean weights of 1×1 convolution kernel.

Fig. 7: The weight changes of convolution kernel branches. The weights of the large convolution kernel increase with the increase of target size; while the weights of the small convolution kernel decrease with the increase of target size.

Based on all the experiments performed in this section, we conclude that:

1. The multi-branch module has different receptive fields for detecting targets at different scales in infrared images.
2. The weight generation module can adaptively weight different branches according to target scale changes. Both two modules can improve detection precision.
3. Compared with the channel-level weight generation module adopted by dynamic convolution and SKNet, the pixel-level weight generation module utilized by DRPN can focus on the changing of the target size and is more suitable for the serious scale variation problem in infrared target detection.

5 Conclusions

In this paper, we proposed a dynamic re-parameterization network (DRPN) to deal with the problem of scale variation in infrared target detection. DRPN consists of a weight generation module and a multi-branch module. The multi-branch module has multiple branches with different receptive fields. The weight generation module makes DRPN automatically weights multiple branches. In virtue of multiple branches with different receptive fields, the multi-branch module makes DRPN dynamically adjust its receptive field. The multi-branch structure is used in the training phase and then converted to a single-branch structure in the inference phase to maintain effective inference. In the weight generation module, dynamic convolution aggregates multiple convolution kernels to make DRPN dynamically adjust its receptive field to adapt to each input based on target sizes. We conducted extensive experiments on FLIR, KAIST, and our InfraPlane datasets, and validated the effectiveness of our method as well as the main modules therein. In the future, we will try to combine DRPN with other scale variation methods to further improve the detection performance.

References

1. Guang Chen, Haitao Wang, Kai Chen, Zhijun Li, Zida Song, Yinlong Liu, Wenkai Chen, and Alois Knoll. A survey of the four pillars for small object detection: Multi-scale representation, contextual information, super-resolution, and region proposal. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–18, 2020.
2. Kang Tong, Yiquan Wu, and Fei Zhou. Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing*, 97:103910, 2020.
3. Xin Xu, Yawen Sun, Lianghui Ding, and Feng Yang. A novel infrared small target detection algorithm based on deep learning. In *2020 4th International Conference on Advances in Image Processing, ICAIP 2020*, page 8b 14, New York, NY, USA, 2020. Association for Computing Machinery.
4. Kaidi Wang, Shaoyi Li, Saisai Niu, and Kai Zhang. Detection of infrared small targets using feature fusion convolutional network. *IEEE Access*, 7:146081–146092, 2019.
5. Dongfang Yang, Xing Liu, Hao He, and Yongfei Li. Air-to-ground multimodal object detection algorithm based on feature association learning. *International Journal of Advanced Robotic Systems*, 16(3):1729881419842995, 2019.
6. Moran Ju, Jiangning Luo, Guangqi Liu, and Haibo Luo. Istdet: An efficient end-to-end neural network for infrared small target detection. *Infrared Physics & Technology*, 114:103659, 2021.
7. Bruce McIntosh, Shashanka Venkataramanan, and Abhijit Mahalanobis. Infrared target detection in cluttered environments by maximization of a target to clutter ratio (tcr) metric using a convolutional neural network. *IEEE Transactions on Aerospace and Electronic Systems*, 57(1):485–496, 2021.
8. Lianghui Ding, Xin Xu, Yuan Cao, Guangtao Zhai, Feng Yang, and Liang Qian. Detection and tracking of infrared small target by jointly using ssd and pipeline filter. *Digital Signal Processing*, 110:102949, 2021.
9. Yu Zhang, Yan Zhang, Zhiguang Shi, Jinghua Zhang, and Ming Wei. Design and training of deep cnn-based fast detector in infrared suav surveillance system. *IEEE Access*, 7:137365–137377, 2019.
10. Vinicius G. Goecks, Grayson Woods, and John Valasek. Combining visible and infrared spectrum imagery using machine learning for small unmanned aerial system detection. In Riad I. Hammoud, Timothy L. Overman, and Abhijit Mahalanobis, editors, *Automatic Target Recognition XXX*, volume 11394, pages 198 – 207. International Society for Optics and Photonics, SPIE, 2020.
11. Manish Sharma, Mayur Dhanaraj, Srivallabha Karnam, Dimitris G. Chachlakis, Raymond Ptucha, Panos P. Markopoulos, and Eli Saber. Yolors: Object detection in multimodal remote sensing imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:1497–1508, 2021.
12. Xinjiang Wang, Shilong Zhang, Zhuoran Yu, Litong Feng, and Wayne Zhang. Scale-equalizing pyramid convolution for object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13356–13365, 2020.
13. Kang Tong, Yiquan Wu, and Fei Zhou. Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing*, 97:103910, 2020.
14. Guang Chen, Haitao Wang, Kai Chen, Zhijun Li, Zida Song, Yinlong Liu, Wenkai Chen, and Alois Knoll. A survey of the four pillars for small object detection: Multi-scale representation, contextual information, super-resolution, and region proposal. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–18, 2020.

- 16 Jingchao Peng, Haitao Zhao, Zhengwei Hu, Kaijie Zhao, and Zhongze Wang
15. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
 16. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing.
 17. Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017.
 18. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.
 19. Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018.
 20. Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 765–781, Cham, 2018. Springer International Publishing.
 21. Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9626–9635, 2019.
 22. Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6568–6577, 2019.
 23. Yimian Dai, Yiquan Wu, Fei Zhou, and Kobus Barnard. Asymmetric contextual modulation for infrared small target detection, 2020.
 24. E. H. Ad Elson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *rca engineer*, 1983.
 25. Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017.
 26. Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 510–519, 2019.
 27. Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhao-Xiang Zhang. Scale-aware trident networks for object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6053–6062, 2019.
 28. Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1911–1920, 2019.
 29. Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *2021 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13733–13742, 2021.
 30. Brandon Yang, Gabriel Bender, Quoc V. Le, and Jiquan Ngiam. *CondConv: Conditionally Parameterized Convolutions for Efficient Inference*. Curran Associates Inc., Red Hook, NY, USA, 2019.

31. Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11027–11036, 2020.
32. Ningning Ma, Xiangyu Zhang, Jiawei Huang, and Jian Sun. Weightnet: Revisiting the design space of weight networks. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 776–792, Cham, 2020. Springer International Publishing.
33. Yikang Zhang, Jian Zhang, Qiang Wang, and Zhao Zhong. Dynet: Dynamic convolution for accelerating convolutional neural networks, 2020.
34. Andreas Veit, Michael Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS’16*, page 550–558, Red Hook, NY, USA, 2016. Curran Associates Inc.
35. L Lathauwer, Bart De Moor, and Joos Vandewalle. Multilinear singular value tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 24, 01 2000.
36. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR 2015 : International Conference on Learning Representations 2015*, 2015.
37. 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. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.
38. Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1037–1045, 2015.
39. Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.