Enabling Deep Residual Networks for Weakly Supervised Object Detection

Yunhang Shen[1], Rongrong Ji[1(✉)], Yan Wang[2], Zhiwei Chen[1], Feng Zheng[3], Feiyue Huang[4], and Yunsheng Wu[4]

[1] Media Analytics and Computing Lab, Department of Artificial Intelligence, School of Informatics, Xiamen University, Xiamen 361005, China
shenyunhang01@gmail.com, rrji@xmu.edu.cn, zhiweichen@stu.xmu.edu.cn
[2] Pinterest, San Francisco, USA
yanw@pinterest.com
[3] CSE, Southern University of Science and Technology, Shenzhen, China
zhengf@sustech.edu.cn
[4] Tencent Youtu Lab, Tencent Technology (Shanghai) Co., Ltd., Shanghai, China
garyhuang@tencent.com, wuyunsheng@gmail.com

Abstract. Weakly supervised object detection (WSOD) has attracted extensive research attention due to its great flexibility of exploiting large-scale image-level annotation for detector training. Whilst deep residual networks such as ResNet and DenseNet have become the standard backbones for many computer vision tasks, the cutting-edge WSOD methods still rely on plain networks, *e.g.*, VGG, as backbones. It is indeed not trivial to employ deep residual networks for WSOD, which even shows significant deterioration of detection accuracy and non-convergence. In this paper, we discover the intrinsic root with sophisticated analysis and propose a sequence of design principles to take full advantages of deep residual learning for WSOD from the perspectives of adding redundancy, improving robustness and aligning features. First, a redundant adaptation neck is key for effective object instance localization and discriminative feature learning. Second, small-kernel convolutions and MaxPool down-samplings help improve the robustness of information flow, which gives finer object boundaries and make the detector more sensitivity to small objects. Third, dilated convolution is essential to align the proposal features and exploit diverse local information by extracting high-resolution feature maps. Extensive experiments show that the proposed principles enable deep residual networks to establishes new state-of-the-arts on PASCAL VOC and MS COCO.

1 Introduction

Different from fully supervised object detection (FSOD) [19,39,41,42] that requires bounding-box-level annotations, weakly supervised object detection

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58598-3_8) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12353, pp. 118–136, 2020.
https://doi.org/10.1007/978-3-030-58598-3_8

Table 1. Comparisons of different backbones for WSDDN [7] on VOC 2007 [15].

Arch.	Backbone	Combination	Depth	Stride	CorLoc (%)	mAP (%)
Plain	AlexNet[1]	C5 [19]	8	16	53.8	32.6
	VGG F[28]		8	16	54.2	34.5
	VGG M[28]		8	16	56.1	34.9
	VGG S[28]		8	12	56.0	34.2
	VGG 16[52]		16	16	53.5	34.8
Residual	ResNet[26]	C4 [26]	18	16	56.8	31.5
			50	16	55.6	30.3
		FPN[36]	18	4/8/16/32	52.3	30.3
			50	4/8/16/32	50.1	30.1
			101	4/8/16/32	46.9	27.7
		C5 [19]	18	32	49.7	28.4
			50	32	50.5	26.5
			101	32	50.9	25.7
	DenseNet[23]	C5 [19]	121	32	55.3	29.7
			161	32	53.0	28.5
	ResNet-WS	C5 [19]	22	8	63.1	43.4
			54	8	63.6	44.0
			105	8	64.0	44.1
	DenseNet-WS	C5 [19]	125	8	**66.3**	**44.8**
			173	8	66.1	44.3

(WSOD) only needs image-level labels. Such relaxation significantly saves the labelling cost and brings large flexibility to many real-world applications.

In a standard pipeline, state-of-the-art WSOD methods first crop region proposals using methods such as RoIPool [19] from backbone networks. Then task-specific heads, *i.e.*, WSOD heads, are built on top of the backbones to localize object instances and learn proposal features jointly. Despite the promising progress made in recent years, there is still a large performance gap from WSOD to FSOD. Prevailing methods generally focus on designing WSOD heads and seldom touch the design of backbone networks, and most state-of-the-art WSOD methods are still built on plain network architectures, *e.g.*, VGG16 [52], VGG-F (M, S) [28] and AlexNet [1], leaving deep residual networks under-explored.

In contrast, it is well known that backbones are important for FSOD in both detection accuracy and inference speed. For accuracy, by simple replacing VGG16 with ResNet [26], Faster R-CNN [42] can increase the mAP@0.5 from 41.5%/75.9% (VGG16) to 48.4%/83.8% (ResNet-101) on COCO and PASCAL VOC 2012, respectively. For speed, light-weight backbones [44,62] significantly reduce the model size and computational complexity. And backbones proposed in [49,77] also enable training detectors from scratch.

However, the direct replacement of residual network to plain networks in WSOD has led to significant performance drop. As an investigation, we first

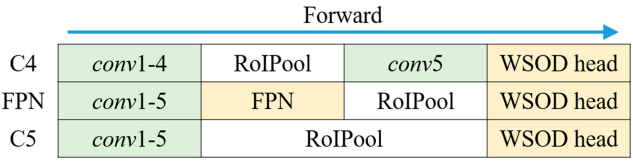

Fig. 1. Various schemes to adapt deep residual networks for WSOD.

quantize the performance of deep residual networks to WSOD under various combinational schemes, as shown in Fig. 1. We build WSDDN [7] head on various plain and residual backbones and evaluate them on PASCAL VOC 2007 [15]. As shown in Table 1, ResNet [26] and DenseNet [23] deteriorate detection performance, which is even inferior to AlexNet [1] in terms of mAP. Moreover, some state-of-the-art methods [27,55,59] are unable to converge as shown in Table 4.

In this paper, we investigate the intrinsic nature towards enabling residual networks to be workable in WSOD. The underlying problem is that WSOD heads are sensitive to model initialization [5,9,11,32] and suffer from instability [38], which may back-propagate uncertain and erroneous gradient to backbones and deteriorate the visual representation learning. Specifically, we propose a sequence of design principles to take full advantage of deep residual networks in three perspectives, *i.e.*, adding redundancy, improving robustness and aligning features.

1. Redundant adaptation neck. Directly employing ResNet backbones to train WSOD deteriorates the discriminability of proposal features, which also fails to localize object instances accurately. The shortcut connections in residual blocks also enlarge the uncertain and erroneous gradient, which overwhelms the direction of optimization steps. Therefore, our first principle is proposing a redundancy adaptation neck with high-dimension proposal representation between deep residual backbones and WSOD heads, which serves as the key to localize object instances and learn discriminative features jointly.

2. Robust information flow. We have also found that ResNet suffers from uncertainty around object boundaries and imperceptibility of small instances under weak supervision. This is mainly caused by the large-kernel (7×7) convolution and non-maximum down-sampling, *i.e.*, 2×2 strided convolution and AveragePool, which lose highly informative features from the raw images. We show that small-kernel convolutions and MaxPool down-samplings provide finer object boundaries and preserve the information of small instances, which enhances the robustness of information flow through the networks.

3. Proposal feature alignment. Modern residual networks commonly achieve large receptive fields by applying an overall stride with $32\times$ sub-sampling. However, such coarse feature maps lead to feature misalignment due to the quantizations in RoIPool [19] layer, which introduces confusing context and lacking diversity. By exploiting dilated convolution to extracts high-resolution feature maps for WSOD, we are able to support the efficient alignment of proposal features and exploit diverse local information, as well as to detect small objects.

We implement two instantiations of the proposed principles: ResNet-WS and DenseNet-WS. Extensive experiments are conducted on PASCAL VOC [15] and MS COCO [37]. We show that the proposed principles enable deep residual networks to achieve significant improvement compared with plain networks for various WSOD methods, which also establishes new state-of-the-arts.

2 Related Work

2.1 Weakly Supervised Object Detection

Prevailing WSOD work generally focuses on two successive stages, object discovery and instance refinement.

Object discovery stage combines multiple instance learning (MIL) and CNNs to implicitly model latent object locations with image-level labels. Several different strategies to train the MIL model had been proposed in the literature [6,8,17,51,61,63]. Bilen *et al.* [7] selected proposals by parallel detection and classification branches. Contextual information [27], attention mechanism [58], saliency map [31,46,48] and semantic segmentation [64] are leveraged to learn outstanding proposals. High-precision object proposals for WSOD are generated in [30,57]. Some methods focused on proposal-free paradigms with deep feature maps [3,4,78], class activation maps [12,21,69,70,75] and generative adversarial learning [13,45]. Some work also used additional information to improve the performance, *e.g.*, object-size estimation [51], instance-count annotations [16], video-motion cue [30,53] and human verification [40]. Knowledge transfer has also been exploited for cross-domain adaptation w.r.t. data [50] and task [24].

Instance refinement stage aims at explicitly learning the object location by making use of the predictions from the object discovery stage. The top-scoring proposals generated from the object discovery stage are used as supervision to train the instance refinement classifier [16,25,32,56,65]. Other different strategies [29,43,55,71] are also proposed to generate pseudo-ground-truth boxes and label proposals. Some methods exploit to improve the optimization of the overall framework that jointly learn the two-stage modules with min-entropy prior [34,60], multi-view learning [72] and continuation MIL [59]. Collaboration mechanism between segmentation and detection is proposed to take advantages of the complementary interpretations of weakly supervised tasks [33,47].

With the output of the above two stages, a fully-supervised detector can also be trained. Many efforts [18,74] have been made to mine high-quality bounding boxes. Zhang *et al.* [73] proposed a self-directed optimization to propagate object priors of the reliable instances to unreliable ones.

2.2 Network Architectures for Object Detection

Significant efforts have been devoted to the design of network architectures for the task of FSOD. DSOD [49] and Root-ResNet [77] exploit to train single-shot detectors, *i.e.*, SSD [39], from scratch, whilst PeleeNet [62] is proposed to train

Table 2. Result of freezing different number of stages in ResNet for WSDDN [7] on VOC 2007 [15]. "NAN" indicates that the training is non-convergent.

Backbone	ResNet18										ResNet50									
Learning Rate	0.001					0.01					0.001					0.01				
#Frozen stages	0	2	3	4	5	0	2	3	4	5	0	2	3	4	5	0	2	3	4	5
mAP (%)	25.2	25.6	26.2	**27.5**	14.4	NAN	NAN	26.7	**28.4**	NAN	23.0	24.9	**25.5**	24.9	NAN	21.0	26.3	**26.5**	26.0	NAN
CorLoc (%)	**46.7**	45.0	42.3	44.6	24.8	NAN	NAN	**50.4**	49.7	NAN	**43.3**	41.8	40.2	37.1	NAN	43.8	**51.3**	50.5	45.9	NAN

SSD for mobile devices. Li *et al.* [35] proposed DetNet backbone for FSOD. Fine feature maps are also useful for detecting small objects as observed in FPN [36].

In conclusion, most traditional backbone networks are usually designed for image classification or FSOD. We have not found one that explores the backbone networks for WSOD. Moreover, the cutting-edge WSOD methods follow the pipeline of ImageNet pre-trained plain networks, *i.e.*, VGG-style networks. Undoubtedly, the advanced modules in recent deep residual architectures have not been explored in WSOD.

3 Baseline WSOD

Without loss of generality, we consider building WSOD models on the pre-trained backbones and fine-tuning its parameters on the target data. We use the popular WSDDN [7] method as baseline WSOD head, which is also a basic module in many state-of-the-art approaches [27,55,56,59,65].

We first investigate several common combination schemes in FSOD to build WSOD heads on ResNet and DenseNet, which are widely used for Faster R-CNN [42], as illustrated in Fig. 1. The C4 [42] combination performs RoIPool [19] on the full-image feature maps from previous 4 stages. All layers in *conv*5 stage and WSOD heads are stacked sequentially on the RoIPooled features. The FPN [36] combination learns full-image feature pyramids from backbones. Then RoIPool is performed to extract 7×7 proposal features followed by two hidden $1,024$-d fully-connected (FC) layers before the WSOD heads. Besides, we also consider a solution, termed C5 [19] combination. C5 combination computes full-image feature maps using all convolutional layers (all 5 stages), followed by a RoIPool layer and later layers.

As shown in Table 1, directly employing ResNet and DenseNet for WSOD task reduces the performance dramatically in various combinations. The best performance of 31.5 mAP is obtained from C4 combination, which is still inferior to the shallow AlexNet backbone in terms of mAP. Moreover, some state-of-the-art methods [27,55,59] are unable to converge according to further experiments in Table 4. We focus on the C5 combination in the rest of the paper, as C4 and FPN combinations have their drawbacks in WSOD setting. C4 combination computes entire *conv*5 stage for each proposal. Thus, it will is cost additional $10\times$ training time and 100% memory usage compared with the C5 combination when each image has about $2,000$ proposals. FPN combination imposes

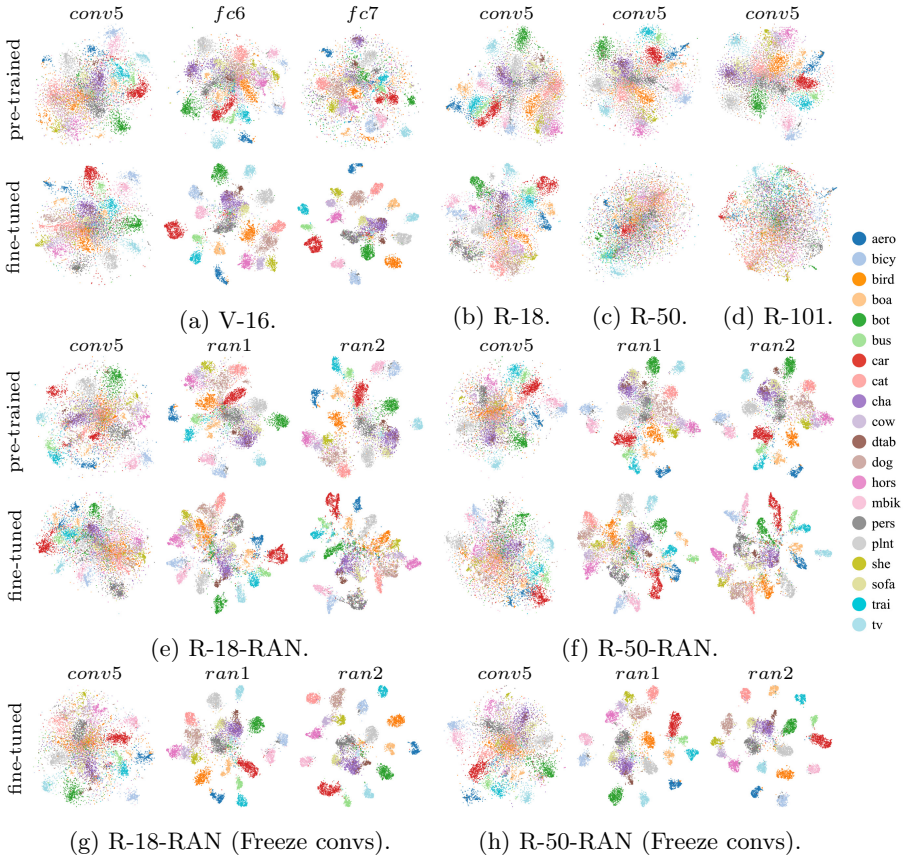

Fig. 2. Visualization of proposal features on PASCAL VOC using t-SNE [20].

an extra burden of learning top-down full-image feature pyramids with lateral connections.

Different from FSOD, WSOD has insufficient supervision and is often formulated via multiple instance learning (MIL) [14], which is sensitive to model initialization [5,9,11,32] and suffers from instability [38]. In this sense, WSOD heads may back-propagate uncertain and erroneous gradient to backbones, whilst deep residual networks enlarge the erroneous information and deteriorate the visual representation learning, which results in dramatically reduced detection performance. To further verify the above analysis, we freeze different number of stages in ResNet, and show the results in Table 2. We summarize: 1) The detection performance mAP is improved progressively by freezing pre-trained layers up to 4 stages, because it prevents convolutional layers in backbones from receiving the erroneous information from WSOD heads. 2) When freezing entire backbones, *i.e.*, all 5 frozen stages, the models has not enough capacity for representation learning (mAP drops dramatically) and even fails to converge.

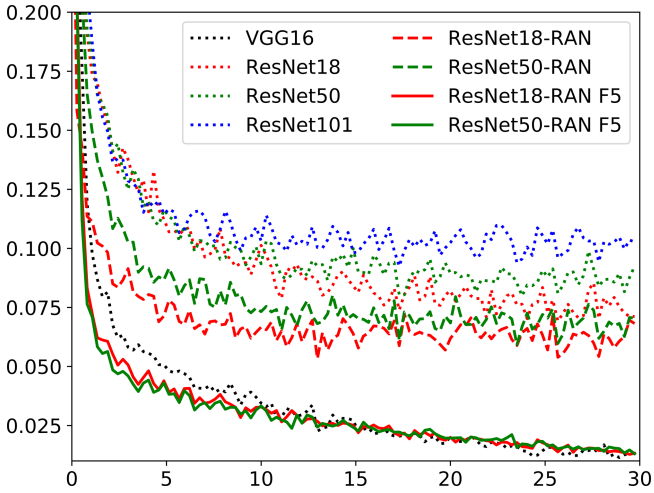

Fig. 3. Optimization landscape analysis of WSDDN with different backbones.

3) Larger learning rate, *i.e.*, 0.01, improves the performance of models with 3 and 4 frozen stages. However, such a large learning rate also enlarges the erroneous information, which results in non-convergent models with 0 and 2 frozen stages. 4) In contrast to mAP of *test* set, the localization performance $CorLoc$ that evaluated in *trainval* set becomes worse as more stages are frozen, which is mainly due to the overfitting. In the following sections, we propose a sequence of design principles to take full advantages of deep residual learning for WSOD.

4 Redundant Adaptation Neck

We visualize the distribution of proposal features uniformly sampled from the PASCAL VOC 2007 trainval set [15] using t-SNE [20] in Fig. 2. We compared VGG16 (V-16) with ResNet of 18 (R-18), 50 (R-50) and 101 (R-101) layers. Proposal features from RoIPool and subsequent layers, *i.e.*, $conv5$, $fc6$ and $fc7$ for VGG16 and $conv5$ for ResNet, are shown. In Fig. 2a, we observe that the proposal features of FC layers from fine-tuned VGG16 are more discriminative than that of the pre-trained ones, whilst the distribution of the features from $conv5$ only changes slightly. However, Fig. 2b, 2c and 2d show that the proposal features of ResNet are not discriminative enough to distinguish different categories. Even more, the proposal features of ResNet50 and ResNet101 are deteriorated compared with the pre-trained counterparts. To further explore the training procedure, we also draw the optimization landscape analysis curves for different backbones in Fig. 3. Generally, optimization loss indicates how well the models reason the relationship of proposals to satisfied the imposed constraints in WSOD. VGG16 demonstrates faster convergence and has lower loss than ResNet backbones, which converge to undesirable local minimums.

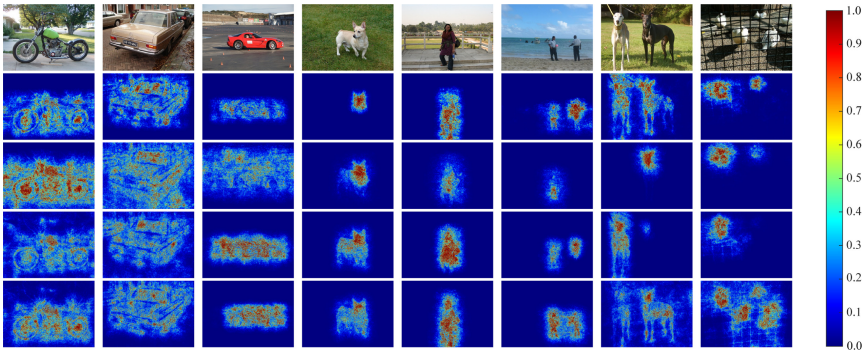

Fig. 4. The first row shows input images. The rest rows show gradient maps of VGG16, R-18-RAN, R-18-RAN-SK and R-18-RAN-SK-MP, respectively.

In conclusion, we observe indiscriminative proposal representation and poor convergence when directly employing ResNet backbones in WSOD task, which cause deteriorated detection performance. As WSOD is required to localize object instances and learn proposal feature jointly with only image-level labels. Therefore, directly stacking WSOD heads on top of residual networks has a large negative impact on the convolutional feature learning. And shortcut connections in residual blocks also enlarge the uncertain and erroneous gradient from WSOD heads throughout the backbones during back-propagation, which overwhelms the direction of optimization steps and fails to infer the proposal-level classifier.

From the perspective of adding redundancy, we propose the first principle that a Redundant Adaptation Neck (RAN), which learns high-dimension visual representation of proposals between deep residual network backbones and the WSOD heads, is the key to localize object instances and learn discriminative features jointly. Our intuition is that the redundant feature representation ensures various WSOD constraints under weak supervision and decreases the negative impact of uncertain and erroneous gradient from WSOD heads, whilst the convolutional layers focus on full-image feature learning. We implement and visualize this principle for ResNet (ResNet-RAN). Instead of instantiating the RAN by stacking convolutional layers, we use multiple perception layers, which are memory-feasible to extract high-dimension features for about $2,000$ proposals. Specifically, the last global pooling layer in ResNet is replaced by two FC layers with high dimension 2,048–4096 before the WSDDN heads. We show the proposal features from $conv5$ and two FC layers from the RAN, *i.e.*, $ran1$ and $ran2$, in Fig. 2e and 2f. ResNet18-RAN and ResNet50-RAN obtain discriminative proposal features in $ran1$ and $ran2$ layers. Figure 3 shows that ResNet-RAN also converges to better minimum. It demonstrates that the entangled tasks of localizing object instances and learning proposal features are optimized jointly.

To further explore the limit of the RAN, we freeze all convolutional layers in the backbones, which completely removes the effect of WSOD heads to the convolutional layers. Figure 2g and 2h show that the proposal features are even

Fig. 5. The two rows show sampling locations of 7^2 discrete bins with maximum values along channels in RoIPool for R-18-RAN and R-18-RAN-DC, respectively.

Table 3. Comparison of various proposal feature extractors for WSDDN [7] on PASCAL VOC 2007 [15] *test* in terms of *m*AP (%).

Extractor	VGG16	ResNet18	ResNet50	ResNet101
RoIPool [19]	**34.8**	**28.4**	**26.5**	**25.7**
RoIAlign [22]	29.4	24.2	23.7	24.8

more discriminative. Meanwhile, the optimization landscape in Fig. 3 is also improved (ResNet-RAN F5). This interesting observation shows that RAN has elastic capacity to accommodate the entangled tasks.

5 Robust Information Flow

Residual learning greatly alleviates the problem of vanishing gradient in deep networks by enhancing information flow with the skip connections. However, there still exist two main drawbacks in deep residual networks, *i.e.*, ResNet and DenseNet, that hinder the robustness of information flow to uncertain and erroneous gradient under weak supervision. First, the large-kernel (7×7) convolutions in the stem block weaken the information of object boundaries, resulting in uncertainty around the object boundaries. Second, non-maximum down-sampling, *i.e.*, 2×2 strided convolutions and AveragePool, may also hurt the flow of information, which makes small instances imperceptible, as the non-maximum down-sampling may not preserve the informative activations and gradient flowing through the network under weak supervision.

From the perspective of improving robustness, we propose a principle that using small-kernel (SK) convolution and MaxPool (MP) down-sampling in the backbones to improve the robustness of information flow, which give finer object boundaries and more sensitivity on small objects. Specifically, we replace the original stem block with three conservative 3×3 convolutions, with the first and third convolutions followed by 2×2 MaxPool layers. For down-sampling, we change the strided convolution or AveragePool operation with MaxPool, which is set to 2×2 with 2×2 stride to avoid the overlapping between input activations.

We utilize the gradient maps of input images to observe how information flows through the networks. In the second and third rows of Fig. 4, we observe that

Table 4. Ablation study on PASCAL VOC 2007 *test*.

	Backbone	Method	RAN	SM	MP	DC	CorLoc (%)	*m*AP (%)
a	ResNet18	WSDDN					49.7	28.4
b			√				57.7	37.5
c				√			53.7	32.9
d					√		54.4	33.7
e						√	51.9	30.9
f			√	√			60.2	40.2
g			√		√		59.2	39.1
h			√	√	√		62.3	42.5
i			√	√		√	61.9	42.1
j			√	√	√	√	**63.1**	**43.4**
k		ContextLocNet					NAN	NAN
l			√	√	√	√	**64.7**	**45.4**
m		OICR					55.3	34.7
n			√	√	√	√	**68.7**	**51.0**
o		PCL					NAN	NAN
p			√	√	√	√	**67.1**	**50.2**
q		C-MIL					NAN	NAN
r			√	√	√	√	**68.5**	**52.6**
s	ResNet50	WSDDN					50.5	26.5
t			√	√	√	√	**63.6**	**44.0**
u	ResNet101	WSDDN					50.9	25.7
v			√	√	√	√	**64.0**	**44.1**
w	DenseNet121	WSDDN					55.5	29.7
x			√	√	√	√	**66.3**	**44.8**
y	DenseNet169	WSDDN					53.0	28.5
z			√	√	√	√	**66.1**	**44.3**

the gradients of object boundaries in R-18-RAN are more blurry than that of VGG16. And the gradients of some object parts and small instances are missed in R-18-RAN. However, gradient maps of R-18-RAN-SK provide finer object boundaries, and R-18-RAN-SK-MP responses to multiple small objects.

6 Proposal Feature Alignment

Modern deep residual networks commonly use 5 stages to extract full-image feature maps with 32× sub-sampling. This brings large effective receptive fields, which are critical for high classification accuracy. However, the large stride may cause misalignment between region proposals and pooled features from the RoIPool [19] layer. The feature misalignment is caused by two quantization operations: coordinate rounding after being divided by the stride, and projected proposals segmentation into discrete bins. Although the misalignment has little negative impact on FSOD, it introduces serious features ambiguity in WSOD, which further raises the instability problem.

Table 5. Comparison with SotAs on PASCAL VOC 2007 *test* in terms of AP.

Method	Backbone	aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	plnt	she	sofa	trai	tv	Av.
Object Discovery																						
WCCN[12]	VGG16	49.5	60.6	38.6	29.2	16.2	70.8	56.9	42.5	10.9	44.1	29.9	42.2	47.9	64.1	13.8	23.5	45.9	**54.1**	60.8	54.5	42.8
Jie *et al.* [25]	VGG16	52.2	47.1	35.0	26.7	15.4	61.3	**66.0**	54.3	3.0	**53.6**	24.7	43.6	48.4	**65.8**	6.6	18.8	**51.9**	43.6	53.6	**62.4**	41.7
SGWSOD[31]	VGG16	48.4	61.5	33.3	**30.0**	15.3	**72.4**	62.4	59.1	10.9	42.3	34.3	53.1	48.4	65.0	20.5	16.6	40.6	46.5	54.6	55.1	43.5
TS^2C[64]	VGG16	**59.3**	57.5	43.7	27.3	13.5	63.9	61.7	**59.9**	**24.1**	46.9	36.7	45.6	39.9	62.6	10.3	23.6	41.7	52.4	58.7	56.6	44.3
CSC C5[48]	VGG16	51.4	62.0	35.2	18.7	**27.9**	66.7	53.5	51.4	16.2	43.6	43.0	46.7	20.0	58.4	**31.1**	23.8	43.6	48.8	65.4	53.5	43.0
WS-JDS[47]	VGG16	52.0	**64.5**	**45.5**	26.7	**27.9**	60.5	47.8	59.7	13.0	50.4	**46.4**	**56.3**	**49.6**	60.7	25.4	**28.2**	50.0	51.4	**66.5**	29.7	**45.6**
WSDDN[7]	VGG16	39.4	50.1	31.5	16.3	12.6	64.5	42.8	42.6	10.1	35.7	24.9	38.2	34.4	55.6	9.4	14.7	30.2	40.7	54.7	46.9	34.8
	ResNet18-WS	47.9	56.8	40.2	17.6	**29.9**	67.2	54.6	**49.6**	8.7	46.6	**47.0**	34.8	**52.0**	**61.4**	**17.0**	24.3	42.2	49.3	60.5	**59.9**	43.4
	ResNet50-WS	**50.4**	56.7	**41.8**	**24.9**	**29.9**	64.0	55.8	47.8	**21.5**	**50.3**	35.0	49.5	49.5	58.1	13.9	**24.5**	44.7	40.7	**65.3**	55.8	44.0
	ResNet101-WS	47.0	**58.6**	40.4	21.1	28.4	**68.4**	**57.1**	46.5	20.1	49.5	35.5	**51.8**	48.1	55.8	12.2	19.6	**45.4**	**53.8**	63.2	58.1	**44.1**
ContextLocNet[27]	VGG-F	57.1	52.0	31.5	7.6	11.5	55.0	53.1	34.1	1.7	33.1	**49.2**	42.0	47.3	56.6	15.3	12.8	24.8	48.9	44.4	47.8	36.3
	ResNet18-WS	58.1	56.3	41.6	**31.4**	22.9	66.1	**57.3**	64.1	11.0	34.2	45.0	**59.7**	**58.9**	60.2	12.9	22.0	30.2	**56.6**	68.5	50.6	45.4
	ResNet50-WS	54.9	**62.8**	41.5	19.1	**28.5**	**67.3**	55.1	52.4	17.9	**48.3**	39.4	45.7	55.3	61.2	**31.1**	22.2	**44.4**	46.7	64.6	45.9	45.3
	ResNet101-WS	**60.6**	53.5	**50.3**	26.1	26.4	66.9	55.8	**73.1**	**18.0**	35.7	19.2	54.7	56.0	**65.6**	25.5	**24.3**	30.3	51.9	**69.4**	**54.4**	**45.9**
Object Discovery + Instance Refinement																						
MELM[60]	VGG16	55.6	66.9	34.2	29.1	16.4	68.8	68.1	43.0	25.0	**65.6**	45.3	53.2	49.6	68.6	2.0	25.4	52.5	56.8	62.1	57.1	47.3
ZLDN[71]	VGG16	55.4	68.5	50.1	16.8	20.8	62.7	66.8	56.5	2.1	57.8	47.5	40.1	69.7	68.2	21.6	27.2	53.4	56.1	52.5	58.2	47.6
GAL-fWSD512[45]	VGG16	58.4	63.8	45.8	24.0	22.7	67.7	65.7	58.9	15.0	58.1	47.0	53.7	23.8	64.3	**36.2**	22.3	46.7	50.3	**70.8**	55.1	47.5
ML-LocNet[72]	VGG16	59.3	68.9	45.7	29.0	24.5	64.8	68.4	59.3	18.6	49.1	50.2	43.1	65.8	70.2	19.9	24.3	48.1	54.2	62.8	41.8	48.4
WSRPN[57]	VGG16	57.9	**70.5**	37.8	5.7	21.0	66.1	69.2	59.4	3.4	57.1	**57.3**	35.2	64.2	68.6	32.8	**28.6**	50.8	49.5	41.1	30.0	45.3
Kosugi *et al.* [29]	VGG16	61.5	64.8	43.7	26.4	17.1	67.4	62.4	67.8	25.4	51.0	33.7	47.6	51.2	65.2	19.3	24.4	44.6	54.1	65.6	59.5	47.6
Pred Net[2]	VGG16	**66.7**	69.5	52.8	31.4	**24.7**	**74.5**	**74.1**	67.3	14.6	53.0	46.1	52.9	69.9	**70.8**	18.5	28.4	54.6	**60.7**	67.1	60.4	52.9
OICR W-RPN[30]	VGG16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	46.9
SDCN[33]	VGG16	59.8	67.1	32.0	34.7	22.8	67.1	63.8	**67.9**	22.5	48.9	47.8	**60.5**	51.7	65.2	11.8	20.6	42.1	54.7	60.8	64.3	48.3
$WSOD^2$[68]	VGG16	65.1	64.8	**57.2**	**39.2**	24.3	69.8	66.2	61.0	**29.8**	64.6	42.5	60.1	**71.2**	70.7	21.9	28.1	**58.6**	59.7	52.2	**64.8**	**53.6**
OICR[56]	VGG16	58.0	**62.4**	31.1	19.4	13.0	65.1	62.2	28.4	**24.8**	44.7	**30.6**	25.3	37.8	**65.5**	15.7	**24.1**	41.7	46.9	64.3	62.6	41.2
	ResNet18-WS	61.3	54.5	52.4	30.1	34.9	**68.9**	65.0	75.0	22.5	57.4	19.7	66.6	64.8	64.9	**16.8**	22.3	**53.2**	54.9	69.9	64.8	51.0
	ResNet50-WS	61.2	50.9	**55.0**	33.2	**36.2**	68.6	**65.7**	**79.2**	17.3	58.1	19.3	69.1	65.7	64.8	15.1	18.9	50.1	55.1	69.8	64.4	50.9
	ResNet101-WS	**63.2**	51.1	51.9	**33.7**	32.4	67.9	65.0	78.9	19.0	**59.4**	21.9	**70.6**	**68.3**	64.4	15.2	20.8	49.3	**55.3**	**72.5**	**66.6**	**51.4**
PCL[55]	VGG16	54.4	69.0	39.3	19.2	15.7	62.9	64.4	30.0	**25.1**	52.5	44.4	19.6	39.3	67.7	17.8	22.9	46.6	**57.5**	58.6	**63.0**	43.5
	ResNet18-WS	54.4	**69.5**	48.7	29.7	**33.2**	70.7	69.7	57.2	11.5	62.4	37.2	39.3	66.3	67.5	23.7	**30.9**	**60.1**	52.0	65.3	55.3	50.2
	ResNet18-WS F2	54.5	67.6	48.1	**31.6**	32.6	**71.5**	**72.3**	67.7	3.3	**64.2**	**58.7**	45.4	67.3	**68.4**	27.7	30.8	56.7	50.6	67.4	51.1	51.9
	ResNet50-WS	55.4	60.7	50.8	30.1	31.0	69.8	69.0	66.6	9.6	62.0	25.0	56.4	68.2	65.5	35.7	28.1	57.2	52.9	67.0	54.2	50.8
	ResNet101-WS	**56.5**	65.4	**54.2**	27.8	30.2	70.8	67.5	**74.8**	3.2	60.4	56.0	**68.0**	**70.6**	65.4	**35.8**	23.1	53.1	53.0	**70.7**	60.4	**53.3**
C-MIL[59]	VGG16	62.5	**58.4**	49.5	32.1	19.8	70.5	66.1	63.4	20.0	**60.5**	**52.9**	53.5	57.4	68.9	8.4	24.6	51.8	58.7	66.7	63.5	50.5
	ResNet18-WS	57.0	54.9	43.6	**39.9**	32.2	70.9	**69.8**	75.2	14.2	59.9	28.5	66.3	67.5	65.3	**37.6**	21.8	**56.7**	49.8	71.1	**68.9**	52.6
	ResNet50-WS	**67.5**	45.2	62.9	33.4	41.6	**73.9**	66.7	76.2	**26.4**	54.8	11.6	**71.4**	71.9	72.9	20.6	31.9	42.5	58.8	**77.1**	61.3	53.4
	ResNet101-WS	66.7	41.4	**64.7**	35.5	**42.2**	73.7	67.3	**76.3**	23.4	56.0	12.1	68.7	**74.5**	**75.1**	22.6	**34.1**	43.6	**60.5**	76.2	64.2	**53.9**
OICR+REG[65]	VGG16	55.2	66.5	40.1	31.1	16.9	69.8	64.3	67.8	**27.8**	52.9	**47.0**	33.0	60.8	64.4	13.8	**26.0**	44.0	**55.7**	68.9	**65.5**	48.6
	ResNet101-WS	**67.3**	**72.1**	**55.8**	**31.8**	**31.3**	**71.6**	**70.0**	**76.7**	19.4	**58.7**	21.1	**68.5**	**74.6**	**69.9**	**19.1**	18.8	**48.4**	55.1	**71.9**	53.2	**52.8**

To address the misalignment of proposal features, we exploit dilated convolution (DC) [66,76] to extract high-resolution full-image feature maps for WSOD. Specifically, we fix the spatial size after stage 3 and use dilated convolution with a rate of 2 in the subsequent stages, which results in only 8× sub-sampling. We visualize the sampling locations of RoIPool in Fig. 5. In the first three columns, the sampling locations of RoIPool in R-18-RAN may exceed the border of proposals, due to the rounded coordinates, whilst R-18-RAN-DC constraints the regions of the sampling inside the proposals. Quantizing proposals into discrete bins in low-resolution feature maps also causes less diversity of sampling locations, as shown in the last five columns of Fig. 5, while high-resolution feature maps from dilated convolution provide more diverse information.

It is worth noting that RoIAlign [22] uses bilinear interpolation to compute the exact values at sampled locations in discrete bins, which aims to address the quantization errors. However, RoIAlign samples activation in a fixed position, which results in inferior performance as shown in Table 3.

7 Quantitative Results

Datasets. We evaluate the proposed design principles on PASCAL VOC 2007, 2012 [15] and MS COCO [37], which are widely-used benchmark datasets.

Evaluation Protocols. The CorLoc indicates the percentage of images in which a method correctly localizes an object of the target category according to the

Table 6. Comparison with SotAs on VOC 2007 *trainval* in terms of CorLoc.

Method	Backbone	aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	plnt	she	sofa	trai	tv	Av.
Object Discovery																						
WCCN[12]	VGG16	83.9	72.8	64.5	44.1	40.1	65.7	82.5	58.9	33.7	72.5	25.6	53.7	67.4	77.4	26.8	49.1	68.1	27.9	64.5	55.7	56.7
Jie *et al.*'17[25]	VGG16	72.7	55.3	53.0	27.8	35.2	68.6	81.9	60.7	11.6	71.6	29.7	54.3	64.3	88.2	22.2	53.7	72.2	52.6	68.9	75.5	56.1
SP-VGGNet[78]	VGG16	**85.3**	64.2	67.0	42.0	16.4	71.0	64.7	**88.7**	20.7	63.8	**58.0**	**84.1**	**84.7**	80.0	**60.0**	29.4	56.3	**68.1**	77.4	30.5	60.6
TST[50]	AlexNet	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	59.5
SGWSOD[31]	VGG16	71.0	**76.5**	54.9	49.7	54.1	78.0	**87.4**	68.8	32.4	**75.2**	29.5	58.0	67.3	84.5	41.5	49.0	**78.1**	60.3	62.8	**78.9**	62.9
TS^2C[64]	VGG16	84.2	74.1	61.3	**52.1**	32.1	76.7	82.9	66.6	**42.3**	70.6	39.5	57.0	61.2	**88.4**	9.3	**54.6**	72.2	60.0	65.0	70.3	61.0
CSC C5[48]	VGG16	76.1	75.3	61.8	42.0	54.1	74.7	78.8	67.4	32.8	73.1	46.5	59.9	37.6	78.0	56.0	42.5	71.9	67.3	82.4	65.6	62.2
WS-JDS[47]	VGG16	82.9	74.0	**73.4**	47.1	**60.9**	**80.4**	77.5	78.8	18.6	70.0	56.7	67.0	64.5	84.0	47.0	50.1	71.9	57.6	**83.3**	43.5	**64.5**
WSDDN[7]	VGG16	65.1	58.8	58.5	33.1	39.8	68.3	60.2	59.6	34.8	64.5	30.5	43.0	56.8	82.4	25.5	41.6	61.5	55.9	65.9	63.7	53.5
	ResNet18-WS	**75.0**	63.7	65.0	37.0	60.4	**80.4**	81.1	**70.5**	21.8	60.8	**65.9**	44.3	**74.8**	85.0	**37.5**	56.3	**68.7**	63.9	74.0	75.0	63.1
	ResNet50-WS	74.1	**68.9**	**69.4**	39.5	**64.0**	79.3	84.3	66.2	**42.4**	**73.9**	38.1	**52.7**	69.7	83.3	27.2	54.8	**68.7**	57.6	**81.8**	75.7	63.6
	ResNet101-WS	72.3	63.7	67.7	**49.3**	61.8	77.3	**85.1**	63.8	36.1	68.1	45.3	52.2	71.2	**87.5**	27.9	**58.6**	66.6	**66.6**	76.3	**81.2**	**64.0**
ContextLocNet[27]	VGG-F	**83.3**	68.6	54.7	23.4	18.3	73.6	74.1	54.1	8.6	65.1	47.1	59.5	67.0	83.5	35.3	39.9	67.0	49.7	63.5	65.2	55.1
	ResNet18-WS	82.1	62.9	**66.6**	44.4	53.2	80.4	**84.5**	**82.2**	22.7	60.8	**60.8**	**68.4**	**76.2**	**84.1**	29.5	55.6	64.5	**68.4**	**76.3**	**70.3**	64.7
	ResNet50-WS	74.0	77.8	63.5	43.6	**58.0**	**81.6**	79.6	68.0	25.2	79.5	59.5	60.0	61.3	81.1	54.5	47.3	**82.5**	62.3	75.7	65.9	65.1
	ResNet101-WS	81.7	**78.7**	65.3	**56.0**	56.2	77.5	82.2	73.0	**32.1**	**81.9**	36.9	62.8	67.6	84.0	**55.2**	**57.1**	70.8	64.4	731.	57.0	**65.7**
Object Discovery + Instance Refinement																						
ZLDN[71]	VGG16	74.0	77.8	65.2	37.0	46.7	75.8	83.7	58.8	17.5	73.1	49.0	51.3	76.7	87.4	30.6	47.8	75.0	62.5	64.8	68.8	61.2
GAL-fWSD512[45]	VGG16	78.6	81.9	63.6	40.3	48.8	80.7	85.3	76.3	30.3	78.0	54.5	65.3	48.4	86.5	56.3	46.9	76.0	68.1	**83.9**	73.1	66.1
ML-LocNet[72]	VGG16	78.6	82.3	68.2	42.0	**53.3**	78.5	88.5	70.3	36.4	70.2	60.5	58.0	80.5	88.2	38.8	59.2	75.0	**69.0**	78.2	64.5	67.0
WSRPN[57]	VGG16	77.5	81.2	55.3	19.7	44.3	80.2	86.6	69.5	10.1	87.7	**68.4**	52.1	84.4	91.6	**57.4**	**63.4**	77.3	58.1	57.0	53.8	63.8
Kosugi *et al.* [29]	VGG16	85.5	79.6	68.1	55.1	33.6	83.5	83.1	**78.5**	42.7	79.8	37.8	61.5	74.4	88.6	32.6	55.7	77.9	63.7	78.4	74.1	66.7
Pred Net VGG16[2]	VGG16	**88.6**	**86.3**	71.8	53.4	51.2	**87.6**	**89.0**	65.3	33.2	86.6	58.8	65.9	**87.7**	**93.3**	30.9	58.9	**83.4**	67.8	78.7	**80.2**	**70.9**
OICR W-RPN[30]	VGG16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	66.5
SDCN[33]	VGG16	85.8	83.1	56.2	58.5	44.7	80.2	85.0	77.9	29.6	78.8	53.6	74.2	73.1	88.4	18.2	57.5	74.2	60.8	76.1	79.2	66.8
$WSOD^2$[68]	VGG16	87.1	80.0	**74.8**	**60.1**	36.6	79.2	83.8	70.6	**43.5**	**88.4**	46.0	**74.7**	87.4	90.8	44.2	52.4	81.4	61.8	67.7	79.9	69.5
OICR[56]	VGG16	81.7	**80.4**	48.7	49.5	32.8	81.7	85.4	40.1	**40.6**	79.5	**35.7**	33.7	60.5	**88.8**	21.8	**57.9**	76.3	59.9	75.3	81.4	60.6
	ResNet18-WS	82.1	60.3	**81.1**	49.3	**67.6**	81.4	87.2	84.0	33.4	76.8	21.6	78.8	**87.0**	87.5	**30.8**	52.6	81.2	66.6	**81.8**	82.8	68.7
	ResNet50-WS	75.9	65.6	70.9	**56.9**	50.0	81.6	86.8	83.8	33.0	79.5	27.3	79.9	81.7	81.0	30.4	45.0	**85.5**	72.2	79.1	81.2	67.4
	ResNet101-WS	**83.3**	68.6	71.3	53.5	54.7	**83.3**	**88.6**	**87.3**	33.8	**80.3**	31.6	**82.5**	85.8	83.8	26.7	42.0	82.5	**73.5**	80.3	**84.7**	**68.9**
PCL[55]	VGG16	79.6	85.5	62.2	47.9	37.0	83.8	83.4	43.0	**38.3**	80.1	50.6	30.9	57.8	**90.8**	27.0	58.2	75.3	68.5	75.7	**78.9**	62.7
	ResNet18-WS	76.7	81.9	74.4	48.1	53.9	84.5	87.7	86.5	25.4	68.1	36.0	67.4	84.8	86.6	52.5	51.1	81.2	54.9	78.7	62.5	67.1
	ResNet18-WS F2	79.4	**86.2**	**75.0**	54.3	53.2	87.6	**88.8**	80.9	10.2	81.1	**68.0**	59.6	**89.2**	87.5	41.7	**59.4**	**83.3**	62.1	80.3	74.2	70.1
	ResNet50-WS	75.8	82.7	73.3	48.1	**60.4**	**88.6**	88.5	74.2	28.1	71.0	46.3	55.6	88.4	88.3	29.3	56.3	81.2	69.3	79.5	71.8	67.8
	ResNet101-WS	**84.9**	77.2	71.3	**60.0**	44.8	76.4	86.4	**87.9**	16.7	**86.1**	67.0	**84.4**	86.5	88.8	**53.1**	50.0	81.3	**72.9**	**85.8**	**78.9**	**72.0**
C-MIL[59]	VGG16	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	65.0
	ResNet18-WS	80.3	64.6	68.3	53.0	**56.8**	84.5	**89.1**	**86.5**	28.1	72.4	28.8	77.3	**84.1**	79.1	**56.8**	51.8	85.4	62.1	**81.1**	**80.4**	68.5
	ResNet50-WS	**80.8**	70.7	**74.4**	53.4	56.6	**85.6**	88.0	85.4	35.2	**84.2**	27.8	78.4	82.4	79.7	31.0	50.0	**89.6**	**73.0**	79.1	80.3	69.3
	ResNet101-WS	78.0	**75.3**	69.6	**63.4**	52.2	85.3	83.3	81.1	**37.8**	79.4	**44.5**	**79.9**	78.3	**85.1**	51.8	**55.7**	85.5	68.9	74.9	79.8	**70.4**
OICR+REG[65]	VGG16	81.7	81.2	58.9	54.3	37.8	**83.2**	86.2	77.0	**42.1**	83.6	**51.3**	44.9	78.2	**90.8**	20.5	**56.8**	74.2	66.1	81.0	**86.0**	66.8
	ResNet101-WS	**88.8**	**86.6**	**66.6**	**57.0**	**48.5**	78.6	**91.1**	**91.3**	34.3	**88.8**	29.1	**78.9**	**90.5**	89.6	**34.1**	41.0	**77.0**	**74.5**	**87.3**	66.4	**70.1**

PASCAL criterion. The mAP follows standard PASCAL VOC protocol to report the mAP at 50% Intersection-over-Union (IoU) of the detected boxes with the ground-truth ones. For MS COCO data, we report the standard COCO metrics, including AP at different IoU thresholds and instance scales.

Implementation Details. All backbone networks are initialized with the weights pre-trained on ImageNet ILSVRC [10]. We use synchronized SGD training on 4 GPUs. A mini-batch involves 1 images per GPU. In the multi-scale setting, we use scales of $\{480, 576, 688, 864, 1200\}$. We set the maximum number of proposals in an image to be $2,000$. We freeze all pre-trained convolutional layers in backbones unless specified otherwise. The test scores are the average of all scales and flips. Detection results are post-processed by non-maximum suppression using a threshold of 0.3.

7.1 Ablation Study

We validate the contribution of each design principle on PASCAL VOC 2007 in Table 4. For rows (b–e), we report the results of applying each principle to ResNet18, which show consistent improvements over the original backbone (a). Especially, RAN (b) provides the largest performance gain among all principles. It demonstrates that RAN is key to localize object instances and learn proposal features jointly. Rows (f–j) show integrating different principles further improve

Table 7. Comparison with SotAs on VOC 2012 in terms of mAP and CorLoc.

Method	Backbone	*m*AP (%)	CorLoc (%)
Object Discovery			
WCCN[12]	VGG16	37.9	–
Jie *et al.* [25]	VGG16	38.3	58.8
SGWSOD[31]	VGG16	39.6	62.9
TS^2C[64]	VGG16	**40.0**	**64.4**
CSC[48]	VGG16	37.1	61.4
WS-JDS[47]	VGG16	39.1	63.5
ContextLocNet[27]	VGG-F	35.3	54.8
	ResNet18-WS	**42.0**	**66.7**
Object Discovery + Instance Refinement			
MELM[60]	VGG16	42.4	–
ZLDN[71]	VGG16	42.9	61.5
WSRPN[57]	VGG16	40.8	64.9
GAL-fWSD300[45]	VGG16	43.1	67.2
Kosugi *et al.* [29]	VGG16	43.4	66.7
ML-LocNet[72]	VGG16	42.2	66.3
Pred Net VGG16[2]	VGG16	**48.4**	69.5
OICR + W-RPN[30]	VGG16	43.2	67.5
SDCN[33]	VGG16	43.5	67.9
$WSOD^2$[68]	VGG16	47.2	**71.9**
OICR[56]	VGG16	37.9	62.1
	ResNet101-WS	**50.4**	**72.5**
	DenseNet121-WS	48.6	70.3
PCL[55]	VGG16	40.6	63.2
	ResNet101-WS	**51.2**	**73.5**
C-MIL[59]	VGG16	46.7	67.4
	ResNet18-WS	**50.6**	**73.0**
OICR+REG[65]	VGG16	46.8	69.5
	ResNet101-WS	**51.1**	**73.2**

detection performance. Compared with the baseline (a), the best performances are improved by 15.0% *m*AP significantly. It demonstrates that the proposed principles are orthogonal to each other. Rows (k–r) show that more state-of-the-art WSOD methods [27,55,56,59] also have significant performance boost. Thus, the proposed principles for backbones are orthogonal to WSOD methods. Finally, rows (s–z) show that with different deep residual backbones, our models also outperform corresponding baselines, with ResNet50 and ResNet101 having more gains compared with ResNet18 (15.0% *vs*. 17.5% *vs*. 18.4% *m*AP).

7.2 Comparison with State of the Arts

To fully compare with other backbones, we separately report the detection results for two successive stages, *i.e.*, object discovery and instance refinement. Table 5 and Table 6 show the results on VOC 2007 in terms of *m*AP and CorLoc, respectively. For object discovery methods, our models with ResNet-WS obtains 43.4–44.1% *m*AP and 63.1–64.0% CorLoc for WSDDN [7], which significantly outperform the previous result with VGG16 by 8.6–9.3% mAP and 9.6–11.5% CorLoc. The improvements of ResNet101-WS for ContextLocNet [27] are 9.6% *m*AP and 10.6% CorLoc. For the instance refinement methods, replacing the backbones of

Table 8. Comparison with the state-of-the-art methods on COCO minival set.

Method	Bakcbone	Avg. Precision, IoU:			Avg. Precision, Area:		
		0.5:0.95	0.5	0.75	S	M	L
WSDDN[7]	VGG-M	8.1	16.0	7.3	1.0	7.8	14.3
	VGG16	9.5	19.2	8.2	2.1	10.4	17.2
	ResNet18-WS	10.7	21.9	**9.1**	2.6	**10.9**	**19.7**
	ResNet101-WS	**10.8**	**22.0**	9.0	**2.7**	10.8	19.6

OICR [56], PCL [55] and C-MIL [59] with ResNet-WS sets the new state-of-the-art results with improvements of 10.8–5.4 mAP.

For CorLoc, our ResNet101-WS backbone surpasses all single-model detectors with improvements of 8.3%, 7.3% and 7.4%, respectively. It is noted that we freeze all convolutional layers in our backbones when fine-tuning on target data. When only freezing the first two stages (ResNet18-WS F2) during training, the performances of PCL achieve further gains with 1.7% mAP and 3.0% CorLoc. Table 7 shows the results on VOC 2012. It can be observed that ResNet-WS models outperform all counterparts with different WSOD methods and achieve new state-of-the-art results. The superiority of ResNet-WS mainly benefits from successfully optimizing the entangled tasks of jointly localizing object instances and learning discriminative features. Table 8 shows the result on MS COCO. We find that ResNet18-WS backbone surpasses existing models on all metrics. For $AP_{0.5:0.95}$, our models outperforms compared works by at least 1.8%. The performance are significantly improved for small instances (44.8% relative improvement for ContextLocNet [27]). This also indicates the efficiency of improving robustness and aligning features.

8 Conclusion

In this paper, we propose a sequence of design principles to take full advantages of deep residual learning for WSOD task. Extensive experiments show that the proposed principles enable deep residual networks to achieve significant performance improvements compared with plain networks for various WSOD methods, which also establishes new state-of-the-arts. Note that our contributions are not specific to ResNet or DenseNet – other backbones (*e.g.*, GoogLeNet [54], WideResNet [67]) can also benefit from the proposed principles for WSOD task.

Acknowledgment. This work is supported by the Nature Science Foundation of China (No. U1705262, No. 61772443, No. 61572410, No. 61802324 and No. 61702136), National Key R&D Program (No. 2017YFC0113000, and No. 2016YFB1001503), Key R&D Program of Jiangxi Province (No. 20171ACH80022) and Natural Science Foundation of Guangdong Province in China (No. 2019B1515120049).

References

1. Alex, K., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Conference on Neural Information Processing Systems (NeurIPS) (2012)
2. Arun, A., Jawahar, C.V., Kumar, M.P.: Dissimilarity coefficient based weakly supervised object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
3. Bazzani, L., Bergamo, A., Anguelov, D., Torresani, L.: Self-taught object localization with deep networks. In: WACV (2016)
4. Bency, A.J., Kwon, H., Lee, H., Karthikeyan, S., Manjunath, B.S.: Weakly supervised localization using deep feature maps. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 714–731. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_43
5. Bilen, H., Pedersoli, M., Tuytelaars, T.: Weakly supervised object detection with posterior regularization. In: The British Machine Vision Conference (BMVC) (2014)
6. Bilen, H., Pedersoli, M., Tuytelaars, T.: Weakly supervised object detection with convex clustering. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
7. Bilen, H., Vedaldi, A.: Weakly supervised deep detection networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
8. Cinbis, R.G., Verbeek, J., Schmid, C.: Multi-fold MIL training for weakly supervised object localization. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
9. Cinbis, R.G., Verbeek, J., Schmid, C.: Weakly supervised object localization with multi-fold multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **39**, 189–203 (2015)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
11. Deselaers, T., Alexe, B., Ferrari, V.: Localizing objects while learning their appearance. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 452–466. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_33
12. Diba, A., Sharma, V., Pazandeh, A., Pirsiavash, H., Van Gool, L.: Weakly supervised cascaded convolutional networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
13. Diba, A., Sharma, V., Stiefelhagen, R., Van Gool, L.: Weakly supervised object discovery by generative adversarial and ranking networks. In: CVPR Workshop (2019)
14. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell. (AI) **89**, 31–71 (1997)
15. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. (IJCV) **88**, 303–338 (2010). https://doi.org/10.1007/s11263-009-0275-4
16. Gao, M., Li, A., Yu, R., Morariu, V.I., Davis, L.S.: C-WSL: count-guided weakly supervised localization. In: European Conference on Computer Vision (ECCV) (2018)

17. Ge, C., Wang, J.: Fewer is more : image segmentation based weakly supervised object detection with partial aggregation. In: The British Machine Vision Conference (BMVC) (2018)
18. Ge, W., Yang, S., Yu, Y.: Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
19. Girshick, R.: Fast R-CNN. In: IEEE International Conference on Computer Vision (ICCV) (2015)
20. Graham-Rowe, D.: Visualizing data using t-SNE. JMLR **9**, 2579–2605 (2008)
21. Gudi, A., van Rosmalen, N., Loog, M., van Gemert, J.: Object-extent pooling for weakly supervised single-shot localization. In: The British Machine Vision Conference (BMVC) (2017)
22. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: IEEE International Conference on Computer Vision (ICCV) (2017)
23. Huang, G., Liu, Z., Weinberger, K.Q.: Densely connected convolutional networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
24. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
25. Jie, Z., Wei, Y., Jin, X., Feng, J., Liu, W.: Deep self-taught learning for weakly supervised object localization. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
26. Kaiming He, Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
27. Kantorov, V., Oquab, M., Cho, M., Laptev, I.: ContextLocNet: context-aware deep network models for weakly supervised localization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 350–365. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_22
28. Ken, C., Karen, S., Andrea, V., Andrew, Z.: Return of the devil in the details delving deep into convolutional nets. In: The British Machine Vision Conference (BMVC) (2014)
29. Kosugi, S., Yamasaki, T., Aizawa, K.: Object-aware instance labeling for weakly supervised object detection. In: IEEE International Conference on Computer Vision (ICCV) (2019)
30. Kumar Singh, K., Jae Lee, Y., Singh, K.K., Lee, Y.J.: You reap what you sow: using videos to generate high precision object proposals for weakly-supervised object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
31. Lai, B., Gong, X.: Saliency guided end-to-end learning for weakly supervised object detection. In: International Joint Conferences on Artificial Intelligence (IJCAI) (2017)
32. Li, D., Huang, J.B., Li, Y., Wang, S., Yang, M.H.: Weakly supervised object localization with progressive domain adaptation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
33. Li, X., Kan, M., Shan, S., Chen, X.: Weakly supervised object detection with segmentation collaboration. In: IEEE International Conference on Computer Vision (ICCV) (2019)

34. Li, Y., Liu, L., Shen, C., van den Hengel, A.: Image co-localization by mimicking a good detector's confidence score distribution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 19–34. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_2
35. Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., Sun, J.: DetNet: a backbone network for object detection. In: European Conference on Computer Vision (ECCV) (2018)
36. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
37. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
38. Liu, B., Gao, Y., Guo, N., Ye, X., You, H., Fan, D.: Utilizing the instability in weakly supervised object detection. In: CVPR Workshop (2019)
39. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
40. Papadopoulos, D.P., Uijlings, J.R.R., Keller, F., Ferrari, V.: We don't need no bounding-boxes: training object class detectors using only human verification. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
41. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
42. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Conference on Neural Information Processing Systems (NeurIPS) (2015)
43. Ren, Z., et al.: instance-aware, context-focused, and memory-efficient weakly supervised object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
44. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: inverted residuals and linear bottlenecks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
45. Shen, Y., Ji, R., Zhang, S., Zuo, W., Wang, Y.: Generative adversarial learning towards fast weakly supervised detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
46. Shen, Y., Ji, R., Wang, C., Li, X., Li, X.: Weakly supervised object detection via object-specific pixel gradient. IEEE Trans. Neural Netw. Learn. Syst. (TNNLS) **29**, 5960–5970 (2018)
47. Shen, Y., Ji, R., Wang, Y., Wu, Y., Cao, L.: Cyclic guidance for weakly supervised joint detection and segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
48. Shen, Y., Ji, R., Yang, K., Deng, C., Wang, C.: Category-aware spatial constraint for weakly supervised detection. IEEE Trans. Image Process. (TIP) **29**, 843–858 (2019)
49. Shen, Z., Liu, Z., Li, J., Jiang, Y.G., Chen, Y., Xue, X.: DSOD: learning deeply supervised object detectors from scratch. In: IEEE International Conference on Computer Vision (ICCV) (2017)
50. Shi, M., Caesar, H., Ferrari, V.: Weakly supervised object localization using things and stuff transfer. In: IEEE International Conference on Computer Vision (ICCV) (2017)

51. Shi, M., Ferrari, V.: Weakly supervised object localization using size estimates. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 105–121. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_7
52. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: The International Conference on Learning Representations (ICLR) (2015)
53. Singh, K.K., Xiao, F., Lee, Y.J.: Track and transfer: watching videos to simulate strong human supervision for weakly-supervised object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
54. Szegedy, C., et al.: Going deeper with convolutions. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
55. Tang, P., et al.: PCL: proposal cluster learning for weakly supervised object detection. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **42**, 176–91 (2018)
56. Tang, P., Wang, X., Bai, X., Liu, W.: Multiple instance detection network with online instance classifier refinement. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
57. Tang, P., et al.: Weakly supervised region proposal network and object detection. In: European Conference on Computer Vision (ECCV) (2018)
58. Teh, E.W., Wang, Y.: Attention networks for weakly supervised object localization. In: The British Machine Vision Conference (BMVC) (2016)
59. Wan, F., Liu, C., Ke, W., Ji, X., Jiao, J., Ye, Q.: C-MIL: continuation multiple instance learning for weakly supervised object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
60. Wan, F., Wei, P., Jiao, J., Han, Z., Ye, Q.: Min-entropy latent model for weakly supervised object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
61. Wang, C., Ren, W., Huang, K., Tan, T.: Weakly supervised object localization with latent category learning. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 431–445. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_28
62. Wang, R.J., Li, X., Ao, S., Ling, C.X.: Pelee: a real-time object detection system on mobile devices. In: Conference on Neural Information Processing Systems (NeurIPS) (2018)
63. Wang, X., Zhu, Z., Yao, C., Bai, X.: Relaxed multiple-instance SVM with application to object discovery. In: IEEE International Conference on Computer Vision (ICCV) (2015)
64. Wei, Y., et al.: TS2C: tight box mining with surrounding segmentation context for weakly supervised object detection. In: European Conference on Computer Vision (ECCV) (2018)
65. Yang, K., Li, D., Dou, Y.: Towards precise end-to-end weakly supervised object detection network. In: IEEE International Conference on Computer Vision (ICCV) (2019)
66. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
67. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: The British Machine Vision Conference (BMVC) (2016)
68. Zeng, Z., Liu, B., Fu, J., Chao, H., Zhang, L.: WSOD∧2: learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In: IEEE International Conference on Computer Vision (ICCV) (2019)

69. Zhang, X., Wei, Y., Feng, J., Yang, Y., Huang, T.: Adversarial complementary learning for weakly supervised object localization. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
70. Zhang, X., Wei, Y., Kang, G., Yang, Y., Huang, T.: Self-produced guidance for weakly-supervised object localization. In: European Conference on Computer Vision (ECCV) (2018)
71. Zhang, X., Feng, J., Xiong, H., Tian, Q.: Zigzag learning for weakly supervised object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
72. Zhang, X., Yang, Y., Feng, J.: ML-LocNet: improving object localization with multi-view learning network. In: European Conference on Computer Vision (ECCV) (2018)
73. Zhang, X., Yang, Y., Feng, J.: Learning to localize objects with noisy labeled instances. In: AAAI Conference on Artificial Intelligence (AAAI) (2019)
74. Zhang, Y., Li, Y., Ghanem, B.: W2F : a weakly-supervised to fully-supervised framework for object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
75. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
76. Zhou, B., et al.: Semantic understanding of scenes through the ADE20K dataset. Int. J. Comput. Vis. (IJCV) **127**, 302–321 (2019). https://doi.org/10.1007/s11263-018-1140-0
77. Zhu, R., et al.: ScratchDet: exploring to train single-shot object detectors from scratch. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
78. Zhu, Y., Zhou, Y., Ye, Q., Qiu, Q., Jiao, J.: Soft proposal networks for weakly supervised object localization. In: IEEE International Conference on Computer Vision (ICCV) (2017)