SELECTIVE CONVOLUTIONAL NETWORK: AN EFFICIENT OBJECT DETECTOR WITH IGNORING BACKGROUND

Hefei Ling[*][†], Yangyang Qin[*], Li Zhang, Yuxuan Shi, Ping Li

Department of Computer Science and Technology, Huazhong University of Science and Technology, CN

ABSTRACT

It is well known that attention mechanisms can effectively improve the performance of many CNNs including object detectors. Instead of refining feature maps prevalently, we reduce the prohibitive computational complexity by a novel attempt at attention. Therefore, we introduce an efficient object detector called *Selective Convolutional Network* (SCN), which selectively calculates only on the locations that contain meaningful and conducive information. The basic idea is to exclude the insignificant background areas, which effectively reduces the computational cost especially during the feature extraction. To solve it, we design an elaborate structure with negligible overheads to guide the network where to look next. It's end-to-end trainable and easy-embedding. Without additional segmentation datasets, we explores two different train strategies including direct supervision and indirect supervision. Extensive experiments assess the performance on PASCAL VOC2007 and MS COCO detection datasets. Results show that SSD and Pelee integrated with our method averagely reduce the calculations in a range of 1/5 and 1/3 with slight loss of accuracy, demonstrating the feasibility of SCN.

Index Terms— Object detection, Efficient convolutional neural network, Object saliency, Attention

1. INTRODUCTION

With the development of deep learning, CNN-based detectors have occupied the dominant position in object detection. We prefer one-stage to two-stage detectors as they are fast and efficient in deploying on the ordinary computer even mobile devices, such as SSD [1]. Even though, developers are often troubled by the expense of massive computational budget that results from the deep and wide architecture. To solve this challenging task, some innovative lightweight CNN models have been proposed in recent years, such as MobileNet [2], Pelee [3], which can run inference on ordinary device in real time. What's more, there are many methods to compress

(a) A bicycle on the street   (b) Erase the surrounding

Fig. 1. Erasing the surrounding information can't confuse us to seek out bicycles and should not hurt the performance of locating bicycles for detectors if their design is reasonable.

models, similar to our purpose, such as structured pruning [4, 5], knowledge distillation [6, 7], adaptive inference[8, 9].

Existing object detection models always perform convolution calculation on the whole spatial information, which results in a large amount of computation. An important thing that has been overlooked in the past researches is that people will not look closely at all the goals in front of them, due to the efficient strategy that the human visual system has learned imperceptibly through the superior colliculus (SC) structure [10, 11]. Taking the street scene in Fig. 1(b) as an example, the process of searching for bicycles will only speed up under the influence of SC structure, rather than lose targets just because of the missing surrounding information. Guided by this illuminating observation, it is possible to detect specified objects only relying on the partial but significant information mostly from the foreground. Though DCN [12] uses the deformable convolution with spatial domain offsets to focus on the specific objects instead of the adjacent background, dispensable activations on the background locations will be also calculated. And many works [13, 14, 15] only use attention mechanisms to enhance certain features, which violates the original intention to decrease the size of search spaces.

Our motivation is to avoid the generation of spatial information redundancy and it resembles a sophisticated spatial pruning more in line with the efficient human visual system to some extent. We make two major contributions as follows:

- We propose a new efficient method called SCN to selectively perform convolution according to the generated foreground mask, which is specially tailored to object detection. Our method has the following merits: i) Easy to embed. ii) Almost no accuracy loss.

- We explore some effortless strategies to get the foreground mask, including a hand-crafted extended branch called Selective Module, and two entirely different strategies to train the module without requiring use of segmentation datasets, as described in Section 2.3.

2. SELECTIVE CONVOLUTIONAL NETWORK
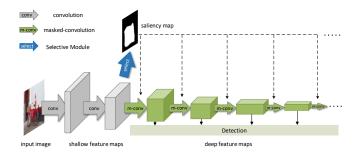
2.1. Overview

Fig. 2. An overview of our SCN. The added saliency map is inferred from the former shallow features, and then instruct the subsequent layers to efficiently extract the specified features.

SCN is designed to bring down the cost of computation on dispensable spatial locations. Applied to the popular SSD [1], an overview of our network structure can be found in Fig. 2. Few changes have been made to the original framework except an extended branch called Selective Module for predicting saliency maps. Saliency is to filter the visual information and select interesting ones for further processing [10]. In our overall architecture, saliency maps are rapidly generated from the former shallow feature maps and guide the latter layers where to do calculation through masked-convolutions [16], where we extend saliency as a binary location-guided mask. As an extended branch, the Selective Module is a tiny architecture and shares features with the shallower trunk branch, inspired by Mask R-CNN [17]. We cautiously design this branch so that it won't put the brakes on the execution of the trunk branch. Module details are described in Section 2.2. After capturing the saliency map, subsequent layers adopt the masked-convolution instead of the vanilla convolution to reduce computation cost extremely effectively as outlined in Fig. 3, which would lose little information on key locations. Moreover, ignored spatial locations will not predict the detection results for decreasing false positives. For a clearer explanation, the process of masked-convolution can be formulated as:

$$x'_{input} = \mathcal{F}(im2col(x_{input}), m_s) \quad (1)$$

$$x'_{output} = wx'_{input} \quad (2)$$

$$x_{output} = col2im(\mathcal{G}(x'_{output}, m_s)) \quad (3)$$

where x_{input} and x_{output} are input and output respectively, w stands for the filter matrix, and m_s is the corresponding
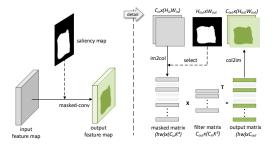

Fig. 3. After $im2col$, the elements in $H_{in} \times W_{in}$ rows of the feature matrix represent the features at corresponding spatial location, and then are selected into $h \times w$ ones to join calculation by saliency map, as $h \times w$ is the number of non-zero entries in saliency map. Finally, uncalculated locations (light olive) in output matrix will be filled into 0 for restoring shape.

saliency map. $im2col$ [18] converts the feature maps into matrix and $col2im$ is the opposite. $\mathcal{F}(\cdot)$ and $\mathcal{G}(\cdot)$ indicate the selective and the scatter function, respectively. What's more, different size masks need to be generated corresponding to different sizes of feature maps, and our experiments show that using downsampling with stride-convolution from the original size mask is better than simple pooling to perfect adaption because of the training strategy seen in Section 2.3.

2.2. Selective module

Selective Module is an essential component of SCN for guiding the network to do lightweight computing. Although it is not a simple matter to get the saliency map, we argue that our elaborate structure with less computational cost can achieve satisfactory results, as illustrated in Fig. 4. Motivated by [19, 20, 21], we adopt an encoder-decoder structure as the transformer for pixel-wise segmentation and use deconvolution as the manner of decoder upsampling, attaching skip architecture. Instead of using preprocessing subnetwork, we propose a plug-and-play network branch to generate foreground masks. The new branch is attached to the shallower trunk branch and shares the bottom-up structure with the trunk to exceedingly reduce the additionally introduced computation, which can almost eliminate the encoder part. However, the semantics of shared feature is insufficient, and the receptive field is not big enough because of fewer strides. To address this problem, we adopt dilated convolution [22] and non-local [14] to expand the receptive field and aggregation context information for better semantics. The number of channels in the Selective Module is quite small compared with the trunk and the convolutions can be replaced by the depthwise separable unit.

Selective Module will generate saliency maps from given feature maps, and which feature to be chosen as input is critical. As discussed above, the selected location determines the capability of the encoder network. Embedded too shallow will make it hard to get desired saliency maps. Meanwhile,
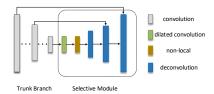
Fig. 4. Illustration of Selective Module.

too deep location hardly reduces computation. Regard to the VGG-16, a backbone with 300×300 pixels resolution, we use the 75×75 and the 38×38 size output feature maps from the Pool2 and the Pool3 layer for embedded location comparison.

2.3. Direct supervision or indirect supervision

Direct Supervision. An obvious way to train our proposed Selective Module is to supervise the mask results directly. Of course, we won't use any segmentation datasets for the sake of fairness. All the ground-truth masks are generated from the bounding boxes. Specifically, we regard the areas inside the bounding boxes as foreground mask 1 and the others as background mask 0. Therefore every ground-truth foreground mask is a square block. The mask value denotes the need for further inference. Note that we experimentally expand one stride size around the ground-truth foreground masks, which can mitigate the harm from the deviation of predicted masks and retain some contextual information. In terms of the loss function, in addition to the common classification loss L_c and localization loss L_l for detection, we regard saliency as foreground segmentation and define a dedicated per-pixel sigmoid cross entropy as saliency mask loss L_m for pixel-wise classification to achieve our goals. Then use a threshold function ($\psi = 0.5$) for converting the probability map to the binary saliency we need. And these three loss functions compose of a multi-task loss function as $L = L_c + \lambda_1 L_l + \lambda_2 L_m$ to jointly optimize parameters. L_c and L_l are identical as the softmax loss and the smooth L1 loss defined in SSD [1], and the saliency mask loss L_m is defined as: $L_m = \frac{1}{N} \sum_{k=1}^{N} L_{ce}(m_k, m_k^*)$ where N denotes the total number of the coordinates in all saliency masks. L_{ce} is the binary cross-entropy loss. m_k is the activation of each coordinate in saliency map, and m_k^* is the corresponding ground-truth mask described above. Like other multi-task loss, λ_1 and λ_2 commonly aim to balance the three terms. Here we simply set $\lambda_1 = \lambda_2 = 1$. Different from the general way, ignored locations won't calculate classification and regression loss to focus the training target on useful foreground locations.

Indirect Supervision. We can also train the Selective Module in an unsupervised fashion [13, 14]. Predicting detection results on each spatial location of specified feature maps is a key characteristic of one-stage detectors [1, 12], and different gradients flow at different locations during training. Therefore, it's possible to train and make the mask generation most beneficial to the prediction results without explicit supervision.

The supervision signal of the Selective Module comes entirely from the final classification and detection loss. We think this training strategy leads to better accuracy. To prevent the gradient explosion of the crowded connection to the backbone when training, only the gradient in the guided layers close to detection heads will flow back. And multiplying mask with output feature maps instead of masked-convolution is more conducive to gradient flow during the training phase.

3. EXPERIMENTS

3.1. Implementation details

We implement all models on the PyTorch framework with a uniform input resolution of 300×300. For a fair comparison, the experiment settings and training strategies are all similar to the original SSD [1] and no additional tricks except adding Batch Normalization for the training convenience. We use the SGD solver with a weight decay of 0.0005 and a momentum of 0.9. First, the models adopt the warm-up strategy for the first 5 epochs. Subsequently, we set the learning rate to 0.01 initially and then use a step decay strategy. The batch size is set to 32. Comparing with the original model, we double the number of iterations to guarantee sufficient training for the introduction of the additional saliency task.

3.2. Results on PASCAL VOC

PASCAL VOC dataset consists of natural images with 20 classes. We train all models on the union set of VOC 2007 trainval and VOC 2012 trainval, and test on VOC 2007 test set.

3.2.1. Design choice

Table 1 shows the performance of various embedded locations and supervision strategies, including reduced computational effort, measured in *floating point operations* (FLOPs), and accuracy indicator. Since different images have different proportions of background, we take the average of lightweight indicators. We can see that indirect supervision leads to higher accuracy, but they tend to be more conservative and don't dare to ignore too much complex background. While the direct supervision strategy is exactly the opposite. It encourages models to boldly ignore the background under the guidance. From another perspective, 38×38 location embedding shows exciting results with just 0.1% to 0.2% degradation. While deeper embedded location can reduce more calculations, observed from the comparison of the embedded location.

For more intuitive, some illustrative examples of inferred saliency map can be found in Fig. 5. The upper two rows show the conservatism of the indirect supervision strategy. Without guidance like direct supervision, the network hardly ignores the complicated background, because small objects are possible to hide in them, such as a cow hidden behind the man in the second row of Fig. 5.

Table 1. Evaluation results of various **design choices** on VOC. *: The baseline is the pre-trained SSD under ImageNet.

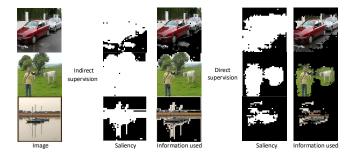| Design		GFLOPs	Reduced	mAP (%)
Location	Supervision			
Baseline*		31.78	-	79.1
75×75	Indirect	25.09	6.69(21.0%)	77.6
	Direct	21.84	**9.94(31.3%)**	77.4
38×38	Indirect	28.55	3.23(10.2%)	**79.0**
	Direct	25.71	6.07(19.1%)	78.9

Fig. 5. Visualization results during processing. For each row, we show an input image, two sets of saliency maps and corresponding information scope, where the maps corresponding from indirect supervision and direct supervision, respectively.

3.2.2. Ablation study

As shown in Table 2, we use 38×38 location embedding models as a baseline for ablation study. Results show that models with indirect supervision are more sensitive to the ablation of complicated subcomponents. But contrary to our intuition, they don't pay off under direct supervision. We consider its a confusing effect caused by the redundant and coarse mask supervision generated by the bounding box. As a matter of fact, the coarse ground-truth masks covered a lot of wrong locations, which make the network hesitate at the edge of objects. In other words, there is a contradiction between good structural performance and unreliable supervision so that we can't design an overly complicated module.

Table 2. **Ablation study** of Selective Module of SCN.

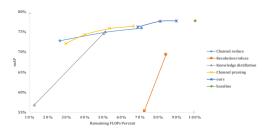|	Supervision	Selective Module of SCN			
Deconv upsample			✓	✓	✓	✓
Skip connection				✓	✓	✓
Non-local		✓			✓	✓
Dilated-conv		✓				✓
mAP(%)	Indirect	61.2	73.1	76.5	78.9	**79.0**
	Direct	77.5	78.9	**78.9**	77.2	78.6

Fig. 6. Comparison with other compression strategies on SSD.

Table 3. Results of SCN on COCO dataset and Pelee detector.

Method	Data	GFLOPs	Reduced	AP (%), IOU		
				0.5:0.95	0.5	0.75
Pelee	VOC	1.18	-	-	71.3	-
Pelee+SCN	VOC	0.79	**0.39(33.1%)**	-	70.1	-
SSD	COCO	35.58	-	26.9	45.3	28.1
SSD+SCN	COCO	27.45	**8.13(22.9%)**	26.1	44.3	27.0
Pelee	COCO	1.25	-	22.6	38.7	23.1
Pelee+SCN	COCO	0.85	**0.40(31.9%)**	20.7	36.2	20.8

3.2.3. Performance comparison

The comparative methods are as follows: i) Channel reduce. ii) Resolution reduce. iii) Knowledge distillation with mask guided [7]. iv) Channel pruning with LASSO-based channel selection[4, 23]. Fig. 6 shows our accuracy degradation is very slight though the compression degree is general, which demonstrates the potential of our method. All these strategies are not contradictory and can complement each other in practice.

3.3. More challenging experiments

In addition to the experiments of SSD on VOC, We further test on Pelee [3], a SOTA lightweight detector, and on more challenging MS COCO dataset, which contains more images and smaller objects. For the sake of simplicity, we only show the results of direct supervision in Table 3 and the indirect supervision situation is similar. The results are not much worse than SSD on VOC, which demonstrates the robustness of our method to handle more complicated examples.

4. CONCLUSION

In this paper, we propose an efficient object detection method to eliminate redundant information processing on useless background locations. Experimental results show that our SCN can reduce the computational cost of SSD and Pelee in a range of 1/5 and 1/3 with little accuracy degradation within 2%, including reducing the computational cost of SSD by about 20% only with 0.2% degradation. All these validate the feasibility of our novel method, and it can be easily integrated in not only ordinary detectors but also lightweight detectors.

5. REFERENCES

[1] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "Ssd: Single shot multibox detector," in *ECCV*. Springer, 2016, pp. 21–37.

[2] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[3] Robert J Wang, Xiang Li, and Charles X Ling, "Pelee: A real-time object detection system on mobile devices," in *Advances in Neural Information Processing Systems*, 2018, pp. 1963–1972.

[4] Yihui He, Xiangyu Zhang, and Jian Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE ICCV*, 2017, pp. 1389–1397.

[5] Kohei Yamamoto and Kurato Maeno, "Pcas: Pruning channels with attention statistics for deep network compression," *arXiv preprint arXiv:1806.05382*, 2018.

[6] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[7] Yousong Zhu, Chaoyang Zhao, Chenxia Han, Jinqiao Wang, and Hanqing Lu, "Mask guided knowledge distillation for single shot detector," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019, pp. 1732–1737.

[8] Andreas Veit and Serge Belongie, "Convolutional networks with adaptive inference graphs," in *Proceedings of ECCV*, 2018, pp. 3–18.

[9] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez, "Skipnet: Learning dynamic routing in convolutional networks," in *Proceedings of ECCV*, 2018, pp. 409–424.

[10] Mingwei Guo, Yuzhou Zhao, Chenbin Zhang, and Zonghai Chen, "Fast object detection based on selective visual attention," *Neurocomputing*, vol. 144, pp. 184–197, 2014.

[11] Brian J White, David J Berg, Janis Y Kan, Robert A Marino, Laurent Itti, and Douglas P Munoz, "Superior colliculus neurons encode a visual saliency map during free viewing of natural dynamic video," *Nature Communications*, vol. 8, pp. 14263, 2017.

[12] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei, "Deformable convolutional networks," in *Proceedings of the IEEE ICCV*, 2017, pp. 764–773.

[13] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon, "Cbam: Convolutional block attention module," in *Proceedings of ECCV*, 2018, pp. 3–19.

[14] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[15] Jianfeng Wang, Ye Yuan, and Gang Yu, "Face attention network: an effective face detector for the occluded faces," *arXiv preprint arXiv:1711.07246*, 2017.

[16] Guanglu Song, Yu Liu, Ming Jiang, Yujie Wang, Junjie Yan, and Biao Leng, "Beyond trade-off: Accelerate fcn-based face detector with higher accuracy," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7756–7764.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," in *Proceedings of the IEEE ICCV*, 2017, pp. 2961–2969.

[18] Kumar Chellapilla, Sidd Puri, and Patrice Simard, "High performance convolutional neural networks for document processing," 10 2006.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[20] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[21] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE ICCV*, 2015, pp. 1520–1528.

[22] Fisher Yu and Vladlen Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[23] Jiaxiang Wu, Yao Zhang, Haoli Bai, Huasong Zhong, Jinlong Hou, Wei Liu, and Junzhou Huang, "Pocketflow: An automated framework for compressing and accelerating deep neural networks," in *NIPS, Workshop*. 2018.