Supplementary Materials for "PSConv: Squeezing Feature Pyramid into One Compact Poly-Scale Convolutional Layer"

Duo Li[1,2⋆], Anbang Yao[2✉], and Qifeng Chen[1✉]

[1] The Hong Kong University of Science and Technology
[2] Intel Labs China
duo.li@connect.ust.hk anbang.yao@intel.com cqf@ust.hk

A PSConv Based on Depthwise Convolution

Regarding the variant of PSConv based on depthwise convolution (DWConv), we do not consider it in our main method because applying PSConv directly to DWConv is non-trivial. Each group of DWConv contains only one channel, thus the cyclic pattern cannot be accommodated inside one group. However, adapting our cyclic pattern to external groups is possible. Specifically, one pattern is arranged across t groups, where t is the original cyclic interval. The illustration diagram is depicted in Fig. 1. It is also noted that such a DWConv-based variant is akin to the MixConv+dilated accompanied with channel shuffling.

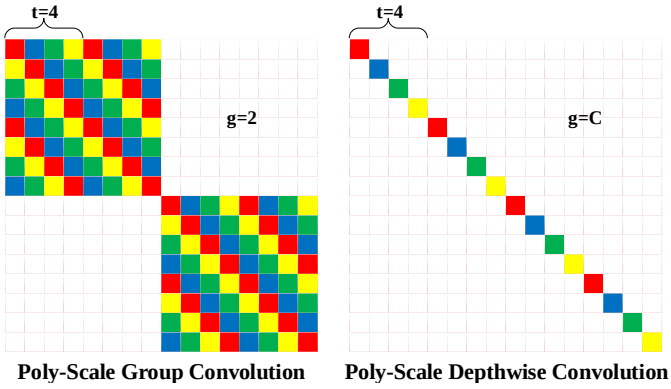

Fig. 1. Comparison between dilation space of kernel lattice in Poly-Scale group convolution (group number $g = 2$ and cyclic interval $t = 4$) and Poly-Scale depthwise convolution (group number $g = C$ and cyclic interval $t = 4$), where $C = 16$ represents the number of channels. Similar to Fig. 2 in the main paper, each color indicates one specific type of dilation rate, the same hereinafter. Best viewed in color.

⋆ indicates intern at Intel Labs China. ✉ indicates corresponding authors.

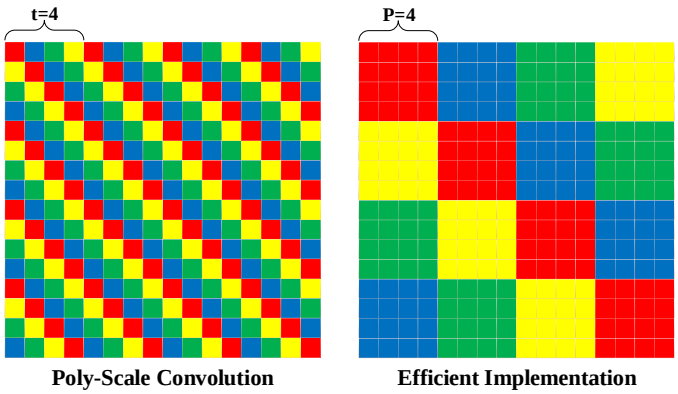

Fig. 2. Comparison between the dilation space of the original PSConv (cyclic interval $t = 4$) and its rearranged dilation space for efficient implementation.

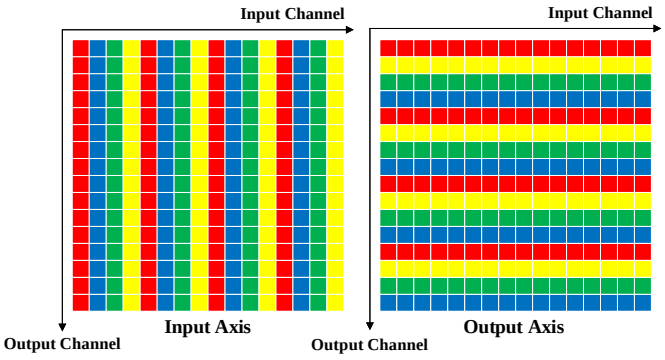

Fig. 3. The dilation space of two simplified cases of PSConv, which only vary dilation rates along the input (*left*) or output (*right*) channel axis.

B Efficient Implementation

In view of the interchangeability of channel indices, we provide an equivalent but efficient implementation of the original PSConv by grouping channels with the same dilation rate together. For each row in the re-arranged dilation space of kernel lattice, the same dilation rates in each partition (P partitions in total) are assembled, shaping a group with P channels. When rearranging the input channel indices, the output channel indices are rearranged accordingly, since the input channels of the current layer are output channels of the precedent layer. The original and rearranged PSConv are comparatively illustrated in Fig. 2. Reminiscent of the definition in the main paper, the dilation rate matrix D is a block matrix after rearrangement, which serves the purpose of efficient matrix operations.

C Ablation of Dilation Patterns

To validate the effectiveness of our design principle, we develop two simplified cases for ablation studies, as shown in Fig. 3. The first one merely varies dilation rates along the input channel axis, which means removing the shift operation from PSConv. Actually it can be interpreted as splitting the incoming features into groups along the channel dimension, transforming these features with one dilation rate per group and aggregating the output features through summation. The second one merely varies dilation rates along the output channel axis. It can be interpreted as transforming the incoming features with different dilation rates in parallel and concatenating the output features along the channel dimension. Therefore, both of these two cases can reduce to the multi-scale network design from the filter space. In contrast, the original PSConv is a more granular design in the kernel space. The corresponding ablative experiments are discussed in the Section 4.2 of the main paper and the comparison in Table 5 of the main paper also demonstrates the superiority of the original PSConv compared to these two simplified design.

D Visualization of Scale Allocation

With curiosity about the learned distribution of scale-relevant features, we dissect the weight proportions with respect to different dilation rates in each PSConv layer, as illustrated in Fig. 4. For each dilation rate in a PSConv layer, we compute the mean of absolute values in each 3×3 kernel and take the maximum across all corresponding kernels as the proxy. These layer-wise proxies can be representative of the importance of different dilation rates. They are finally normalized inside each layer for inter-layer comparison.

As for PS-ResNet-50 on the ImageNet, it is observed that in the first residual block of stage 3-5 (`conv3_x`, `conv4_x`, and `conv5_x`), where feature maps are processed with stride 2, PSConv is prone to overlook convolutional kernels with large dilation rates and emphasize those without dilation rates, as the downsampling

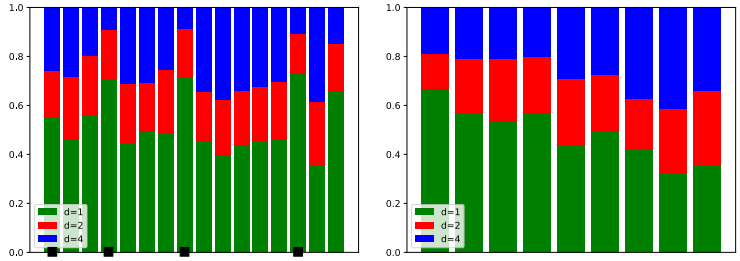

Fig. 4. Visualization of the automated selection mechanism concerning multi-scale features. The left panel reveals the result of PS-ResNet-50 on the ImageNet, where the horizontal axis corresponds to indices of residual blocks, ■ indicates the starting block of stage 2-5. The right panel displays the result of PS-ResNeXt-29 (16×64d) on the CIFAR-100. Best viewed in color.

operation already offers sufficient amplification of the receptive fields at these points. This trend is not obvious for PS-ResNeXt-29 (16×64d) on CIFAR-100, partially due to its few down-sampling operations. Nevertheless, there exists a clear tendency that convolutional kernels with large dilation rates will occupy a larger proportion in the deeper layers, implying the necessity of allocating more resources to semantic features in the high-level layers. The visual analysis also helps understand the quantitative performance improvement with a better coarse-to-fine feature generation process compared to standard convolutions.

E Object Detection

We perform experiments with Faster R-CNN on the MS COCO object detection track and report the results in Table 1. Compared to the detectors with vanilla convolutions, PSConv also achieves obviously higher AP based on different backbone architectures. The comparison of performance gains across three backbone networks shows a similar trend as Mask R-CNN in the main paper.

Table 1. Bounding-box Average Precision (AP) comparison on the COCO 2017 validation set for the bounding-box detection track with different backbones.

Detector	Architecture	Conv Type	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN	R50	standard	36.4	58.4	39.1	21.5	40.0	46.6
		PSConv	38.4$_{(+2.0)}$	60.6	41.6	22.9	42.4	49.9
	R101	standard	38.5	60.3	41.6	22.3	43.0	49.8
		PSConv	40.9$_{(+2.4)}$	63.0	44.3	23.8	45.3	53.5
	X101-32x4d	standard	40.1	62.0	43.8	23.4	44.6	51.7
		PSConv	41.3$_{(+1.2)}$	63.6	45.1	24.7	45.5	53.8
Cascade R-CNN	R50	standard	40.4	58.5	43.9	21.5	43.7	53.8
		PSConv	41.9$_{(+1.5)}$	60.8	45.5	24.2	45.3	55.6
	R101	standard	42.0	60.3	45.9	23.2	45.9	56.3
		PSConv	43.8$_{(+1.8)}$	62.6	47.7	25.6	47.5	57.9
	X101-32x4d	standard	43.6	62.2	47.4	25.0	47.7	57.4
		PSConv	44.4$_{(+0.8)}$	63.6	48.4	26.6	48.3	59.2

F Visualization of Predictions on MS COCO

We select Faster R-CNN and Mask R-CNN with ResNet-101 for visualization in view of the large margins between our PSConv-based detectors and the standard ones under this setting, as indicated by experimental results in the Section 4.3 of the main paper.

The result comparisons of Faster R-CNN are presented in Fig. 5, 6 and 7. Regarding the bounding-box results, detectors based on PSConv could reduce false alarms of large-sized objects and precisely perceive small-sized instances. For example, the potted plant in the second row of Fig. 5, the bear in the last row of Fig. 6, the refrigerator in the first row of Fig. 7 are obvious false alarms that are rejected in the predictions of our model. Furthermore, referring to the middle row in Fig. 6, the bounding box of the umbrella is more compact and the bench below the person is detected with confidence. It validates the superiority of our PSConv-based detector to capture objects with diverse shapes and sizes.

The result comparisons of Mask R-CNN are presented in Fig. 8, 9 and 10. For example, the traffic light in the third row of Fig. 8, the person in the first row and the sink in the last row of Fig. 10 are false alarms in the standard detector but omitted in our PSConv-based detector. A skiing person on the snow mountain is missed by the standard detector possibly due to its tiny size, but successfully detected by the PSConv-based model, as shown in the first row of Fig. 9. As demonstrated in the last row of Fig. 9, distinct instances of the bench are distinguished together with the detected small bird, thanks to the robustness of our PSConv to scale variation.

G Speed Optimization

Based on our preliminary GPU speed benchmark in the main paper, the speed gap is primarily due to dilated convolution inside our PSConv. However, such a gap can be largely bridged using a specialized implementation of Dilated-Winograd Convolution (DWC). Compared to GEMM implementation in cuDNN, the average speedup by DWC for dilated convolutional layers with a dilation rate of 2/4 is 2.14×/1.53×, on a single TITAN X GPU (similar results are also reported in [2]). By adopting the TVM compiler [1], the speedup can be further increased to 2.86×/2.01×. After combining the latest version of Intel OneDNN tool (achieving an additional speedup of +0.2), the inference time of a PSConv layer would be roughly 1.42× of the standard convolution. Furthermore, the above optimization procedure could yield a better speedup ratio on CPU inference, tested on Dual Intel Xeon Platinum 8280 @ 2.70GHz. Since we only apply PSConv to the 3×3 convolutional layers of a residual network, the slow-down effect will be diluted on a whole network compared to a single convolution layer. Specifically, the inference time of a PSConv-based ResNet-50/101 becomes very similar to the standard ResNet-50/101 (1.066× on GPU and 1.051× on CPU). As a consequence, our PSConv can be comfortably put into practical usage.

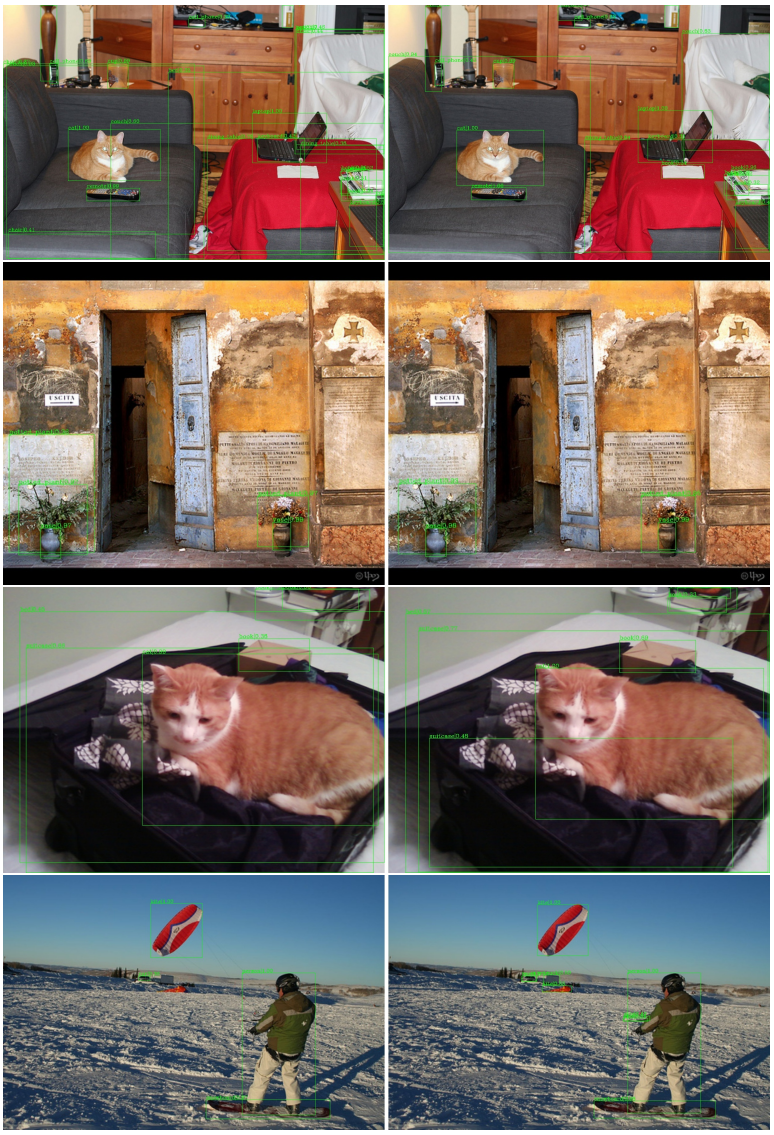

Fig. 5. Some bounding-box detection results of Faster R-CNN with ResNet-101 on the COCO 2017 validation set. The left panel shows predictions from the standard detector while the right panel shows the detector equipped with our PSConv, the same hereinafter.

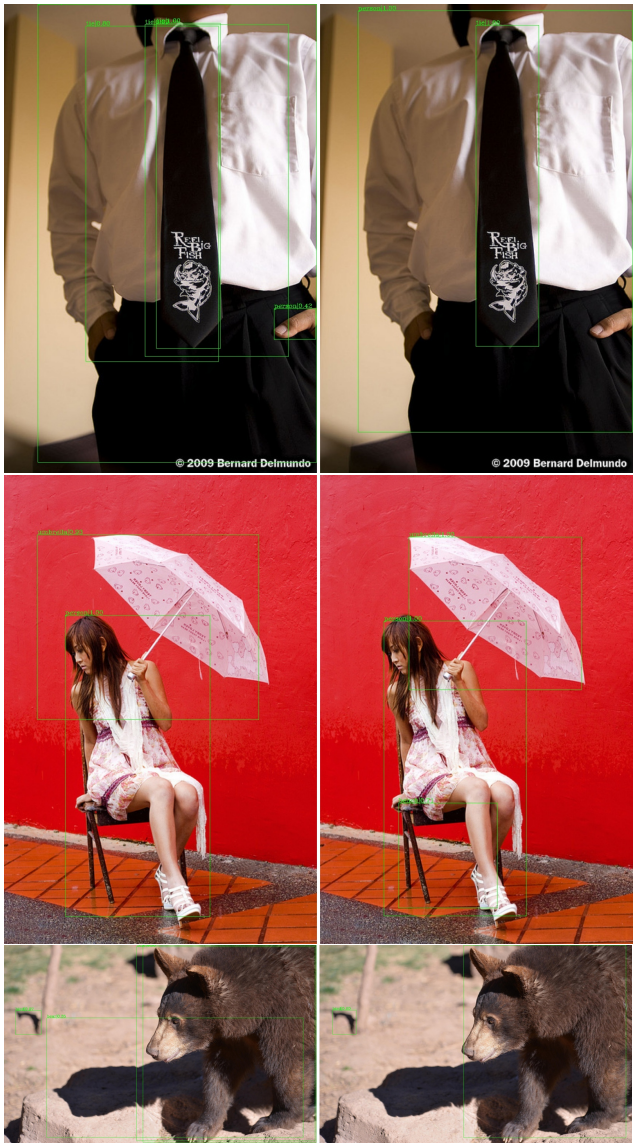

Fig. 6. Some bounding-box detection results of Faster R-CNN with ResNet-101 on the COCO 2017 validation set.

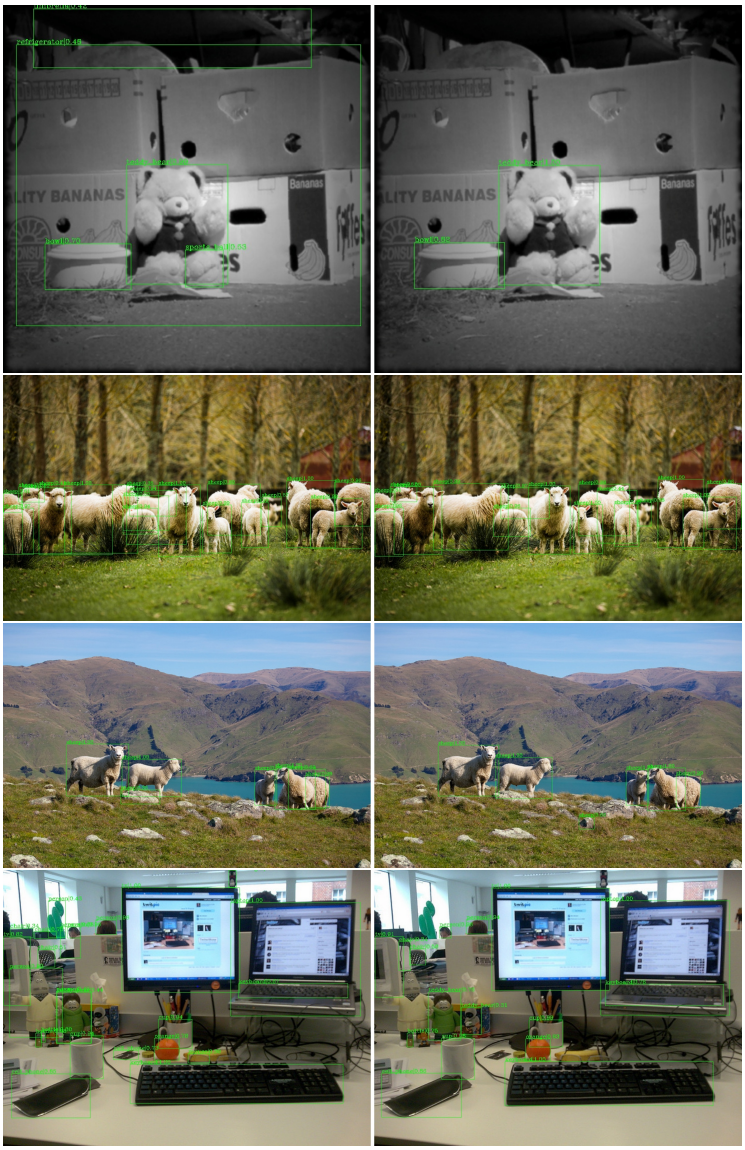

Fig. 7. Some bounding-box detection results of Faster R-CNN with ResNet-101 on the COCO 2017 validation set.

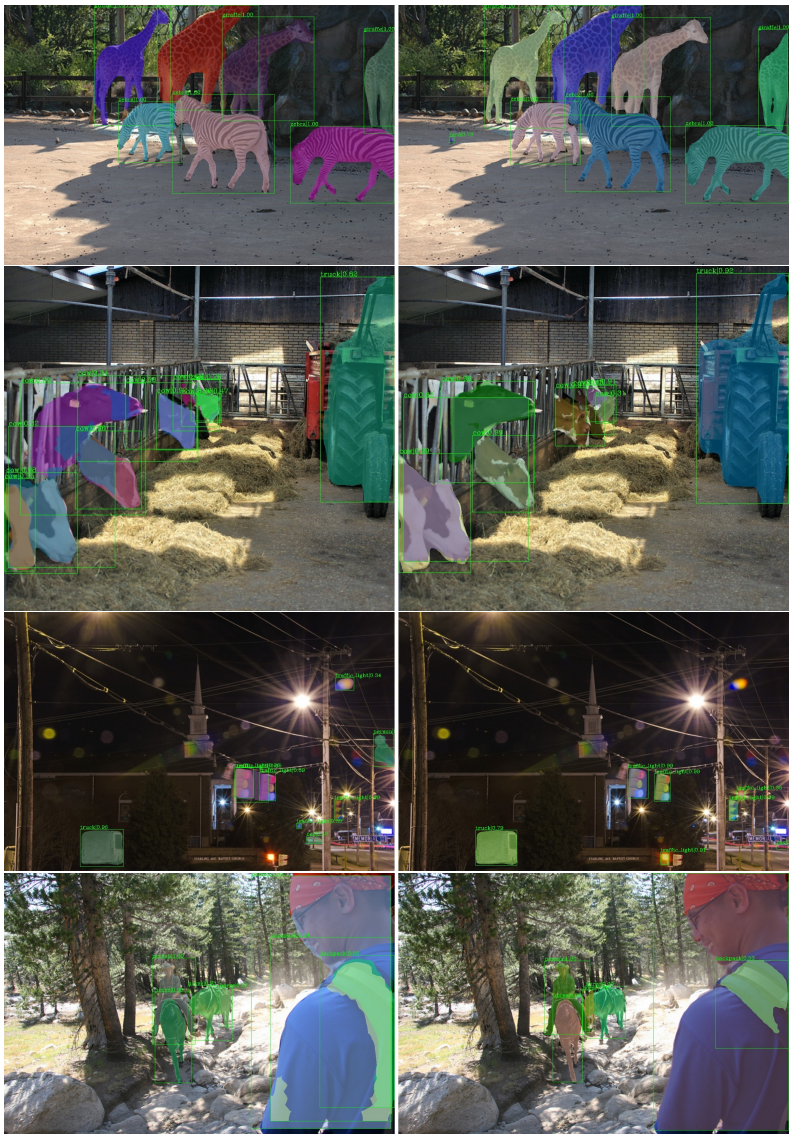

Fig. 8. Some bounding-box detection and instance segmentation results of Mask R-CNN with ResNet-101 on the COCO 2017 validation set.

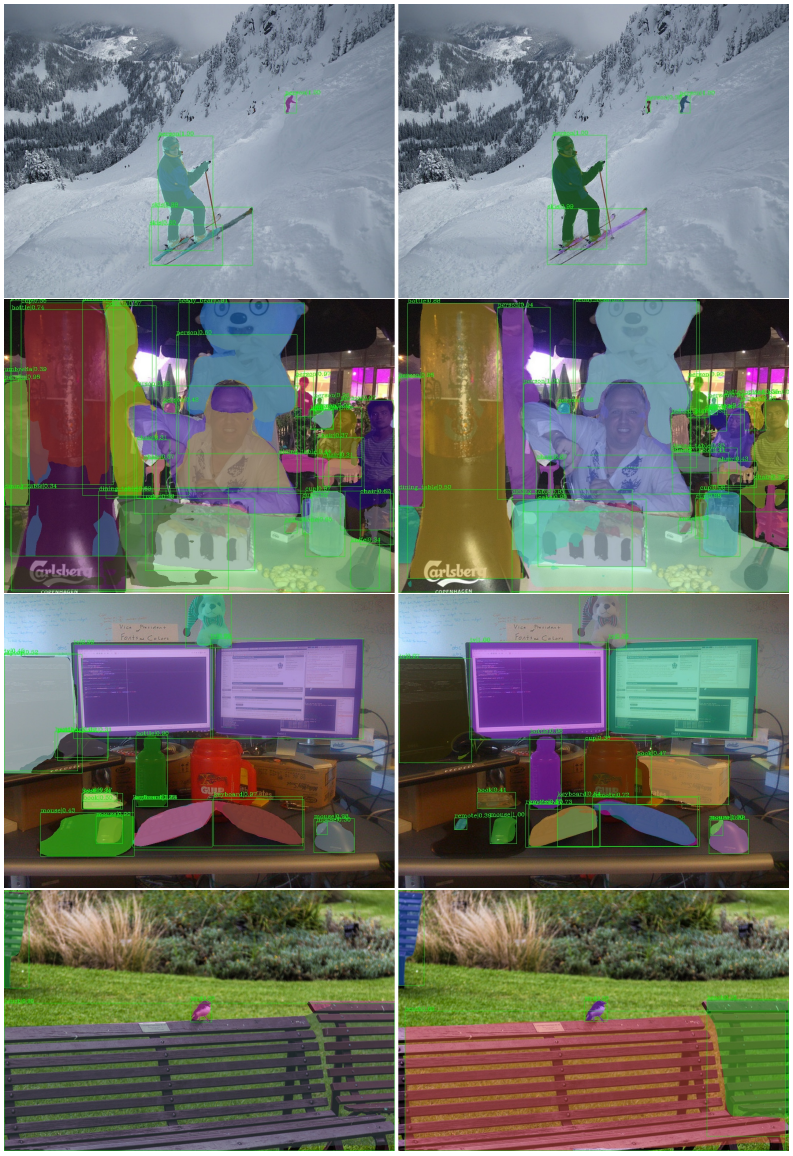

Fig. 9. Some bounding-box detection and instance segmentation results of Mask R-CNN with ResNet-101 on the COCO 2017 validation set.

Fig. 10. Some bounding-box detection and instance segmentation results of Mask R-CNN with ResNet-101 on the COCO 2017 validation set.

References

1. Chen, T., Moreau, T., Jiang, Z., Zheng, L., Yan, E., Shen, H., Cowan, M., Wang, L., Hu, Y., Ceze, L., Guestrin, C., Krishnamurthy, A.: TVM: An automated end-to-end optimizing compiler for deep learning. In: OSDI (2018)
2. Kim, M., Park, C., Kim, S., Hong, T., Ro, W.W.: Efficient dilated-winograd convolutional neural networks. In: ICIP (2019)