Supplementary materials for "SP-NAS: Serial-to-Parallel Backbone Search for Object Detection"

Chenhan Jiang*[1], Hang Xu*†[1],
Wei Zhang[1], Xiaodan Liang[2], Zhenguo Li[1]
[1]Huawei Noah's Ark Lab [2]Sun Yat-Sen University

Detailed Searched Architectures of SPNet Series

We describe our SPNet series backbone networks with 224×224 input size for ImageNet in Figure 1. The network starts from a stem that consists of two strided 3×3 convolutions decreasing the resolution to $\frac{1}{4}$ following [4][5].

Layer name	Output size	SPNet$_{VOC}$(BB)	SPNet$_{ECP}$	SPNet$_{BDD}$	SPNet$_{COCO}$	SPNet$_{COCO}$(XB)	ResNet101
Conv1	112×112	$3 \times 3, 32$, stride 2					$7 \times 7, 64$, stride 2
		$3 \times 3, 64$, stride 2					2×2, Max-pool
Stage1	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$ $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 5$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 10$ $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 4$ $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 256 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
Stage2	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 5$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 17$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$ $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 512 \end{bmatrix} \times 5$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
Stage3	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 10$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 2$ $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 18$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 1024 \end{bmatrix} \times 25$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$
Stage4	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 7$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 17$ $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
Stage5	4×4	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024 \\ 1 \times 1, 2048 \end{bmatrix} \times 1$	-
Stage6	2×2	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 4$	-	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 1$	-	-
Block Type		basic block	bottleneck block	bottleneck block	bottleneck block	ResNext block	bottleneck block
FLOPS (GB)		5.5	14.1	13.8	9.5	8.8	4.1
Top-1 Error (%)		23.7	20.3	20.0	20.9	20.0	22.8

Figure 1. Searched Architectures of SPNet. Detailed kernel size and output channel of blocks are shown in brackets, with the numbers of blocks stacked. Downsampling is performed by Stage 2-6 with a stride of 2.

*Both authors contributed equally to this work.
†Corresponding Author: xuhang33@huawei.com

Structrue of Blocks

Specific structure in each type of block is shown in Figure 2. Left: Basic Block with 2 convolution layers with the same input and output channel D. Middle: Bottleneck Block with 3 convolution layers following [3]. Right: A block of ResNext with groups = 32, width = 4, like in [8]. A layer is shown as (kernel size, # out channels).

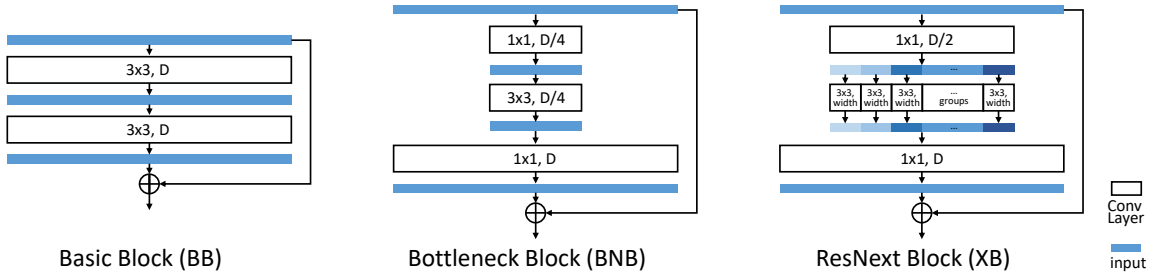

Figure 2. Illustration of specific structure in each type of block.

Detailed Comparison of Different Detector on COCO

We attempt to combine searched SPNet$_{COCO}$ with two different detectors, Faster RCNN with FPN [6] and Cascade RCNN [2] as in Table 1. The input size is 800×1333 and the training epoch is 24. It shows SPNet$_{COCO}$-BNB with FPN [6] detector improves 1.7% mAP to ResNext101 with faster inference speed. Replace FPN head with the cascade heads [2] can boost performance to 45.6 mAP with reducing inference speed by 11%. Furthermore, SPNet$_{COCO}$-XB with Cascade RCNN can reach 47.4% mAP.

%	Method	Backbone	AP	AP$_{50}$	AP$_S$	Time (fps)
COCO	FPN [6]	ResNet101 [3]	39.4	60.6	22.1	11.9
		ResNext101 [8]	40.1	62.0	23.4	10.3
		SPNet$_{COCO}$-BNB	41.8	63.4	25.8	**11.5**
		SPNet$_{COCO}$-XB	44.0	65.0	26.6	6.0
	Cascade RCNN [2]	ResNet101 [3]	42.8	62.1	23.7	10.2
		ResNext101 [8]	44.5	63.3	26.1	6.7
		SPNet$_{COCO}$-BNB	45.6	64.3	28.4	10.2
		SPNet$_{COCO}$-XB	**47.4**	**65.7**	**29.6**	5.6

Table 1. Comparison of mean Average Precision and inference time of ResNet/ResNext with different detection head. Inference time is tested on one V100 GPU. BNB and XB indicate bottleneck block and ResNext block respectively. The proposed SPNet has consistent improvement with stronger detection head.

Generalization ability of the searched architecture

To evaluate the domain transferability of our SP-NAS, we transfer the searched architecture to other datasets, as shown in Table 2. It can be found that directly searching on the target dataset performs best while transferring from other datasets will have a small performance drop. The better transferability can be found between similar datasets such as BDD [9] and ECP [1] (both street scenes).

Arch Searched From	COCO mAP	BDD mAP	ECP LAMR
COCO [7]	**41.8**	$37.4^{-1.3}$	$0.060^{+0.006}$
BDD [9]	$40.7^{-1.1}$	**38.7**	$0.059^{+0.005}$
ECP [1]	$40.6^{-1.2}$	$37.8^{-0.9}$	**0.054**

Table 2. Generalization ability of SP-NAS. We transfer the searched architecture with FPN from COCO, BDD and ECP to other datasets (smaller LAMR is better).

Qualitative Results.

Qualitative comparisons on EuroCity Person (ECP) [1] dataset between ResNet101 and our method SPNet$_{ECP}$ can be found in Figure 3. Both backbone networks are based on the Faster RCNN with FPN [6] detector. For the comparisons, our SPNet$_{ECP}$ considers more computation allocation on former stages for locating the tiny-size person and additional stage for

distinguishing objects with occlusion and ambiguity, while ResNet101 performs more miss prediction and false positives. The network searched by our SP-NAS is more accurate than the baseline ResNet series due to the data-oriented computation allocation.

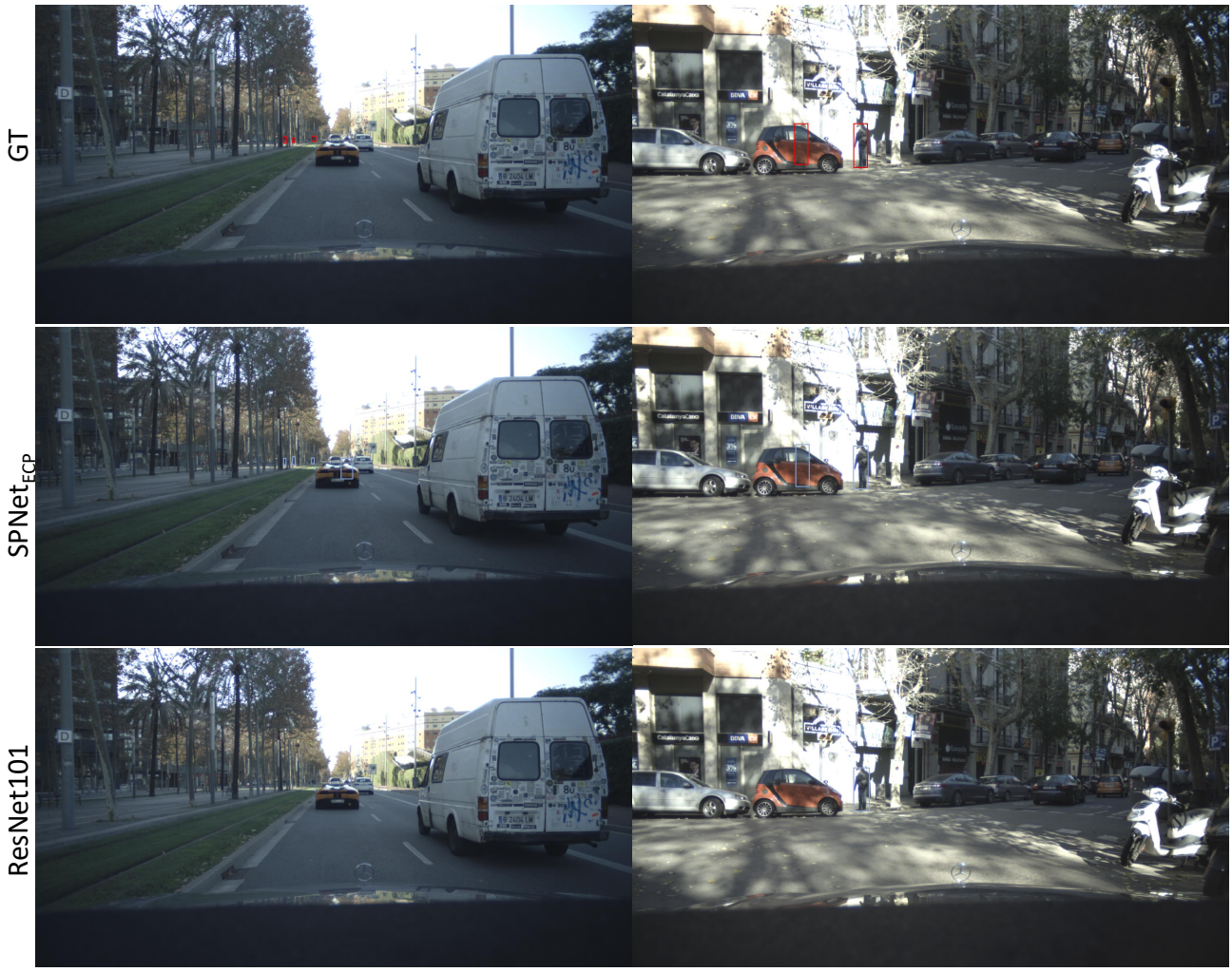

Figure 3. Qualitative results comparison on ECP between ResNet101 and SPNet$_{ECP}$. Our method can detect more tiny-size and serve occlusion pedestrians.

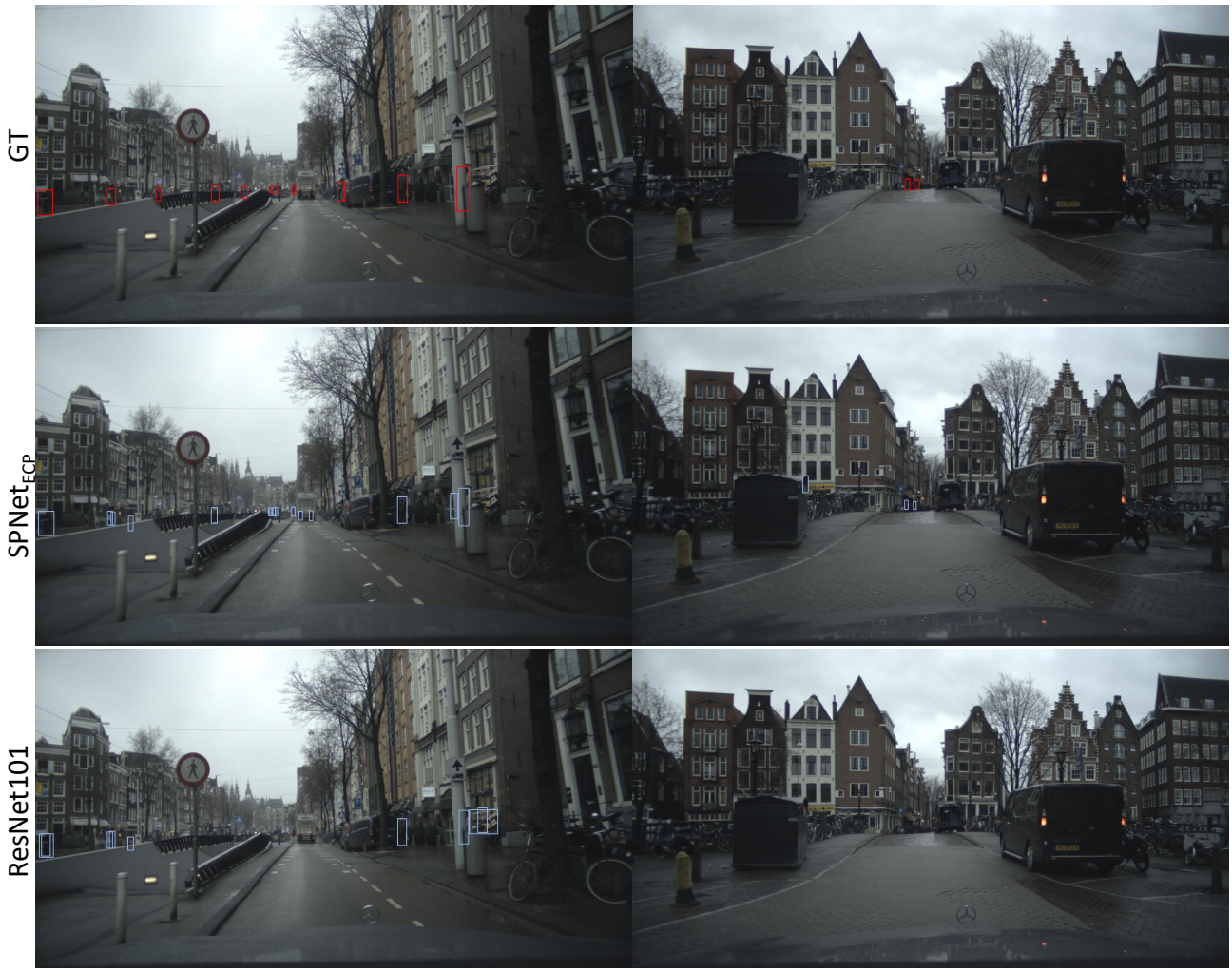

Figure 4. Qualitative results comparison on ECP between ResNet101 and SPNet$_{ECP}$. Our method is more accurate than the baseline ResNet101 due to the data-oriented computation allocation.

Figure 5. Qualitative results comparison on ECP between ResNet101 and SPNet$_{ECP}$.

References

[1] Markus Braun, Sebastian Krebs, Fabian B. Flohr, and Dariu M. Gavrila. Eurocity persons: A novel benchmark for person detection in traffic scenes. *TPAMI*, 2019.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[4] ingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2019.

[5] Xin Li, Yiming Zhou, Zheng Pan, and Jiashi Feng. Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In *CVPR*, pages 9145–9153, 2019.

[6] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[8] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[9] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.