

Lite-HRNet: A Lightweight High-Resolution Network

Changqian Yu^{1,2} Bin Xiao² Changxin Gao¹ Lu Yuan² Lei Zhang² Nong Sang^{1*} Jingdong Wang^{2*}

¹Key Laboratory of Image Processing and Intelligent Control

School of Artificial Intelligence and Automation, Huazhong University of Science and Technology

²Microsoft

{changqian-yu, cgao, nsang}@hust.edu.cn, {bixi, luyuan, leizhang, jingdw}@microsoft.com

Abstract

We present an efficient high-resolution network, *Lite-HRNet*, for human pose estimation. We start by simply applying the efficient shuffle block in *ShuffleNet* to *HRNet* (high-resolution network), yielding stronger performance over popular lightweight networks, such as *MobileNet*, *ShuffleNet*, and *Small HRNet*.

We find that the heavily-used pointwise (1×1) convolutions in shuffle blocks become the computational bottleneck. We introduce a lightweight unit, conditional channel weighting, to replace costly pointwise (1×1) convolutions in shuffle blocks. The complexity of channel weighting is linear w.r.t the number of channels and lower than the quadratic time complexity for pointwise convolutions. Our solution learns the weights from all the channels and over multiple resolutions that are readily available in the parallel branches in *HRNet*. It uses the weights as the bridge to exchange information across channels and resolutions, compensating the role played by the pointwise (1×1) convolution. *Lite-HRNet* demonstrates superior results on human pose estimation over popular lightweight networks. Moreover, *Lite-HRNet* can be easily applied to semantic segmentation task in the same lightweight manner. The code and models have been publicly available at <https://github.com/HRNet/Lite-HRNet>.

1. Introduction

Human pose estimation requires high-resolution representation [3, 2, 26, 41] to achieve high performance. Motivated by the increasing demand for model efficiency, this paper studies the problem of developing efficient high-resolution models under computation-limited resources.

Existing efficient networks [5, 6, 50] are mainly designed from two perspectives. One is to borrow the design from

classification networks, such as *MobileNet* [17, 16] and *ShuffleNet* [28, 53], to reduce the redundancy in matrix-vector multiplication, where convolution operations dominate the cost. The other is to mediate the spatial information loss with various tricks, such as encoder-decoder architectures [2, 26], and multi-branch architectures [50, 55].

We first study a naive lightweight network by simply combining the shuffle block in *ShuffleNet* and the high-resolution design pattern in *HRNet* [41]. *HRNet* has shown a stronger capability among large models in position-sensitive problems, e.g., semantic segmentation, human pose estimation, and object detection. It remains unclear whether high resolution helps for small models. We empirically show that the direct combination outperforms *ShuffleNet*, *MobileNet*, and *Small HRNet*¹.

To further achieve higher efficiency, we introduce an efficient unit, named *conditional channel weighting*, performing information exchange across channels, to replace the costly pointwise (1×1) convolution in a shuffle block. The channel weighting scheme is very efficient: the complexity is linear w.r.t the number of channels and lower than the quadratic time complexity for the pointwise convolution. For example, with the multi-resolution features of $64 \times 64 \times 40$ and $32 \times 32 \times 80$, the conditional channel weighting unit can reduce the shuffle block's whole computation complexity by 80%.

Unlike the regular convolutional kernel weights learned as model parameters, the proposed scheme weights are conditioned on the input maps and computed across channels through a lightweight unit. Thus, they contain the information in all the channel maps and serve as a bridge to exchange information through channel weighting. Furthermore, we compute the weights from parallel multi-resolution channel maps that are readily available in *HRNet* so that the weights contain richer information and are strengthened. We call the resulting network, *Lite-HRNet*.

¹Small *HRNet* is available at <https://github.com/HRNet/HRNet-Semantic-Segmentation>. It simply reduces the depth and the width of the original *HRNet*.

*Corresponding author. This work is done when C. Yu was an intern at Microsoft Research, Beijing, P.R. China.

The experimental results show that Lite-HRNet outperforms the simple combination of shuffle blocks and HRNet (which we call naive Lite-HRNet). We believe that the superiority is because the computational complexity reduction is more significant than the loss of information exchange in the proposed conditional channel weighting scheme.

Our main contributions include:

- We simply apply the shuffle blocks to HRNet, leading a lightweight network naive Lite-HRNet. We empirically show superior performance over MobileNet, ShuffleNet, and Small HRNet.
- We present an improved efficient network, Lite-HRNet. The key point is that we introduce an efficient conditional channel weighting unit to replace the costly 1×1 convolution in shuffle blocks, and the weights are computed across channels and resolutions.
- Lite-HRNet is the state-of-the-art in terms of complexity and accuracy trade-off on COCO and MPII human pose estimation and easily generalized to semantic segmentation task.

2. Related Work

Efficient blocks for classification. Separable convolutions and group convolutions have been increasingly popular in lightweight networks, such as MobileNet [17, 36, 16], IGCv3 [37], and ShuffleNet [53, 28]. Xception [9] and MobileNetV1 [17] disentangle one normal convolution into depthwise convolution and pointwise convolution. MobileNetV2 and IGCv3 [37] further combine linear bottlenecks that are about low-rank kernels. MixNet [39] applies mixed kernels on the depthwise convolutions. EfficientHRNet [30] introduces the mobile convolutions into HigherHRNet [8].

The information across channels are blocked in group convolutions and depthwise convolutions. The pointwise convolutions are heavily used to address it but are very costly in lightweight network design. To reduce the complexity, grouping 1×1 convolutions with channel shuffling [53, 28] or interleaving [52, 46, 37] are used to keep information exchange across channels. Our proposed solution is a lightweight manner performing information exchange across channels to replace costly 1×1 convolutions.

Mediating spatial information loss. The computation complexity is positively related to spatial resolution. Reducing the spatial resolution with mediating spatial information loss is another way to improve efficiency. Encoder-decoder architecture is used to recover the spatial resolution, such as ENet [34] and SegNet [2]. ICNet [56] applies different computations to different resolution inputs to reduce the whole complexity. BiSeNet [50, 49] decouples the detail information and context information with different lightweight sub-networks. Our solution follows the high-

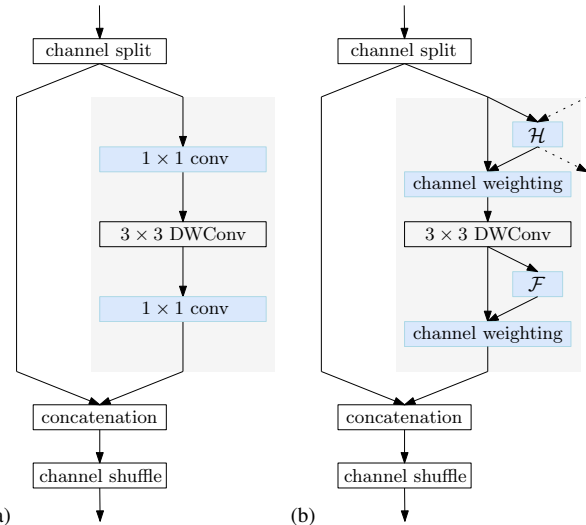


Figure 1. **Building block.** (a) The shuffle block. (b) Our conditional channel weighting block. The dotted line indicates the representation from other resolutions and the weights distributed to other resolutions. \mathcal{H} = cross-resolution weighting function. \mathcal{F} =spatial weighting function.

resolution pattern in HRNet to maintain the high-resolution representation through the whole process.

Convolutional weight generation and mixing. Dynamic filter networks [21] dynamically generates the convolution filters conditioned on the input. Meta-Network [29] adopts a meta-learner to generate weights to learn cross-task knowledge. CondINS [40] and SOLOV2 [43] apply this design to the instance segmentation task, generating the parameters of the mask sub-network for each instance. CondConv [47] and Dynamic Convolution [5] learn a series of weights to mix the corresponding convolution kernels for each sample, increasing the model capacity.

Attention mechanism [19, 18, 44] can be regarded as a kind of conditional weight generation. SENet [19] uses global information to learn the weights to excite or suppress the channel maps. GENet [18] expands on this by gathering local information to exploit the contextual dependencies. CBAM [44] exploits the channel and spatial attention to refine the features.

The proposed conditional channel weighting scheme can be, in some sense, regarded as a conditional channel-wise 1×1 convolution. Besides its cheap computation, we exploit an extra effect and use the conditional weights as the bridge to exchange information across channels.

Conditional architecture. Different from normal networks, conditional architecture can achieve dynamic width, depth, or kernels. SkipNet [42] uses a gated network to skip some convolutional blocks to reduce complexity selectively. Spatial Transform Networks [20] learn to warp the

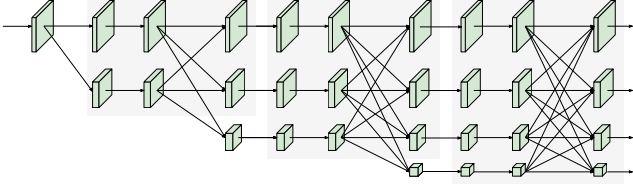


Figure 2. **Illustration of the Small HRNet architecture.** It consists of a high-resolution stem as the first stage, gradually adding high-to-low resolution streams as the main body. The main body has a sequence of stages, each containing parallel multi-resolution streams and repeated multi-resolution fusions. The details are given in Section 3.

feature map conditioned on the input. Deformable Convolution [11, 57] learns the offsets for the convolution kernels conditioned on each spatial location.

3. Approach

3.1. Naive Lite-HRNet

Shuffle blocks. The shuffle block in ShuffleNet V2 [28] first splits the channels into two partitions. One partition passes through a sequence of 1×1 convolution, 3×3 depthwise convolution, and 1×1 convolution, and the output is concatenated with the other partition. Finally, the concatenated channels are shuffled, as illustrated in Figure 1 (a).

HRNet. The HRNet [41] starts from a high-resolution convolution stem as the first stage, gradually adding high-to-low resolution streams one by one as new stages. The multi-resolution streams are connected in parallel. The main body consists of a sequence of stages. In each stage, the information across resolutions is exchanged repeatedly. We follow the Small HRNet design² and use fewer layers and smaller width to form our network. The stem of Small HRNet consists of two 3×3 convolutions with stride 2. Each stage in the main body contains a sequence of residual blocks and one multi-resolution fusion. Figure 2 illustrates the structure of Small HRNet.

Simple combination. We adopt the shuffle block to replace the second 3×3 convolution in the stem of Small HRNet, and replace all the normal residual blocks (formed with two 3×3 convolutions). The normal convolutions in the multi-resolution fusion are replaced by the separable convolutions [9], resulting in a naive Lite-HRNet.

3.2. Lite-HRNet

1×1 convolution is costly. The 1×1 convolution performs a matrix-vector multiplication at each position:

$$Y = W \otimes X, \quad (1)$$

²<https://github.com/HRNet/HRNet-Semantic-Segmentation>

where X and Y are input and output maps, and W is the 1×1 convolutional kernel. It serves a critical role of exchanging information across channels as the shuffle operation and the depthwise convolution have no effect on information exchange across channels.

The 1×1 convolution is of quadratic time complexity ($\Theta(C^2)$) with respect to the number (C) of channels. The 3×3 depthwise convolution is of linear time complexity ($\Theta(9C)$). In the shuffle block, the complexity of two 1×1 convolutions is much higher than that of the depthwise convolution: $\Theta(2C^2) > \Theta(9C)$, for the usual case $C > 5$. Table 2 shows an example of the complexity comparison between 1×1 convolutions and depthwise convolutions.

Conditional channel weighting. We propose to use the element-wise weighting operation to replace the 1×1 convolution in naive Lite-HRNet, which has s branches in the s th stage. The element-wise weighting operation for the s th resolution branch is written as,

$$Y_s = W_s \odot X_s, \quad (2)$$

where W_s is a weight map, a 3-d tensor of size $W_s \times H_s \times C_s$, and \odot is the element-wise multiplication operator.

The complexity is linear with respect to the channel number $\Theta(C)$, and much lower than 1×1 convolution in the shuffle block.

We compute the weights by using the channels for a single resolution and the channels across all the resolutions, as shown in Figure 1 (b), and show that the weights play a role of exchanging information across channels and resolutions.

Cross-resolution weight computation. Considering the s -th stage, there are s parallel resolutions, and s weight maps W_1, W_2, \dots, W_s , each for the corresponding resolution. We compute the s weight maps from all the channels across resolutions using a lightweight function $\mathcal{H}_s(\cdot)$,

$$(W_1, W_2, \dots, W_s) = \mathcal{H}_s(X_1, X_2, \dots, X_s), \quad (3)$$

where $\{X_1, \dots, X_s\}$ are the input maps for the s resolutions. X_1 corresponds to the highest resolution, and X_s corresponds to the s -th highest resolution.

We implement the lightweight function $\mathcal{H}_s(\cdot)$ as following. We perform adaptive average pooling (AAP) on $\{X_1, X_2, \dots, X_{s-1}\}$: $X'_1 = \text{AAP}(X_1)$, $X'_2 = \text{AAP}(X_2)$, \dots , $X'_{s-1} = \text{AAP}(X_{s-1})$, in which the AAP pools any input size to a given output size $W_s \times H_s$. Then we concatenate $\{X'_1, X'_2, \dots, X'_{s-1}\}$ and X_s together, followed by a sequence of 1×1 convolution, ReLU, 1×1 convolution, and sigmoid, generating weight maps consisting of s partitions, W'_1, W'_2, \dots, W'_s (each for one resolution):

$$\begin{aligned} (X'_1, X'_2, \dots, X_s) &\rightarrow \text{Conv.} \rightarrow \text{ReLU} \rightarrow \text{Conv.} \rightarrow \text{sigmoid} \\ &\rightarrow (W'_1, W'_2, \dots, W'_s). \end{aligned} \quad (4)$$

³In terms of time complexity, the constant 9 should be ignored. We keep it for analysis convenience.

Table 1. **Structure of Lite-HRNet.** The stem contains one stride 2×3 convolution and one shuffle block. The main body has three stages, each of which has a sequence of modules. Each module consists of two conditional channel weight blocks and one fusion block. N in Lite-HRNet- N indicates the number of layers. *resolution branch* indicates this stage contains the feature stream of the corresponding resolution. ccw = conditional channel weight.

layer	output size	operator	resolution branch	#output_channels	repeat	#modules	
						Lite-HRNet-18	Lite-HRNet-30
image	256×256		$1 \times$	3			
stem	64×64	conv2d	$2 \times$	32	1	1	1
		shuffle block	$4 \times$	32	1		
stage ₂	64×64	ccw block	$4 \times 8 \times$	40, 80	2	2	3
		fusion block	$4 \times 8 \times$	40, 80	1		
stage ₃	64×64	ccw block	$4 \times 8 \times 16 \times$	40, 80, 160	2	4	8
		fusion block	$4 \times 8 \times 16 \times$	40, 80, 160	1		
stage ₄	64×64	ccw block	$4 \times 8 \times 16 \times 32 \times$	40, 80, 160, 320	2	2	3
		fusion block	$4 \times 8 \times 16 \times 32 \times$	40, 80, 160, 320	1		
FLOPs						273.4M	425.3M
#Params						1.1M	1.8M

Table 2. **Computational complexity comparison: 1×1 convolution vs. conditional channel weight.** $X_s \in \mathcal{R}^{H_s \times W_s \times C_s}$ are the input channel maps for the s resolution, X_1 corresponds to the highest resolution. $N_s = H_s W_s$. For example, the shape of X_1 and X_2 are $64 \times 64 \times 40$ and $32 \times 32 \times 80$, respectively. single/cross-resolution=information exchange.

model	single-resolution	cross-resolution	Theory Complexity	Example FLOPs
1×1 convolution	✓		$\sum_1^s N_s C_s^2$	12.5M
3×3 depthwise convolution			$\sum_1^s 9 N_s C_s$	2.1M
CCW w/ spatial weights	✓		$\sum_1^s (2C_s^2 + N_s C_s)$	0.25M
CCW w/ multi-resolution weights		✓	$2(\sum_1^s C_s)^2 + \sum_1^s N_s C_s$	0.26M
CCW	✓	✓	$2(\sum_1^s C_s)^2 + 2\sum_1^s (C_s^2 + N_s C_s)$	0.51M

Here, the weights at each position for each resolution depend on the channel feature at the same position from the average-pooled multi-resolution channel maps. This is why we call the scheme as cross-resolution weight computation. The $s - 1$ weight maps, $W'_1, W'_2, \dots, W'_{s-1}$, are upsampled to the corresponding resolutions, outputting W_1, W_2, \dots, W_{s-1} , for the subsequent element-wise channel weighting.

We show that the weight maps serves as a bridge for information exchange across channels and resolutions. Each element of the weight vector w_{si} at the position i (from the weight map W_s) receives the information from all the input channels of all the s resolutions at the same pooling region, which is easily verified from the operations in Equation 4. Through such a weight vector, each of the output channels at this position,

$$y_{si} = w_{si} \odot x_{si}, \quad (5)$$

receives the information from all the input channels at the same position across all the resolutions. In other words, the channel weighting scheme plays the role as well as the 1×1 convolution in terms of exchanging information.

On the other hand, the function $\mathcal{H}_s(\cdot)$ is applied on the small resolution, and thus the computation complexity is

very light. Table 2 illustrates that the whole unit has much lower complexity than 1×1 convolution.

Spatial weight computation. For each resolution, we also compute the spatial weights which are homogeneous to spatial positions: the weight vector w_{si} at all positions are the same. The weights depend on all the pixels of the input channels in a single resolution:

$$w_s = \mathcal{F}_s(X_s). \quad (6)$$

Here, the function $\mathcal{F}_s(\cdot)$ is implemented as: $X_s \rightarrow \text{GAP} \rightarrow \text{FC} \rightarrow \text{ReLU} \rightarrow \text{FC} \rightarrow \text{sigmoid} \rightarrow w_s$. The global average pooling (GAP) operator serves as a role of gathering the spatial information from all the positions.

By weighting the channels with the spatial weights, $y_{si} = w_s \odot x_{si}$, each element in the output channels receives the contribution from all the positions of all the input channels. We compare the complexity between 1×1 convolutions and conditional channel weighting unit in Table 2.

Instantiation. The Lite-HRNet consists of a high-resolution stem and the main body to maintain the high-resolution representation. The stem has one 3×3 convolution with stride 2 and a shuffle block, as the first stage. The main body has a sequence of modularized modules.



Figure 3. **Example qualitative results on COCO pose estimation:** containing viewpoint change, occlusion, and multiple persons.

Table 3. **Comparisons on the COCO val set.** pretrain = pretrain the backbone on ImageNet. #Params and FLOPs are calculated for the pose estimation network, and those for human detection and keypoint grouping are not included.

model	backbone	pretrain	input size	#Params	GFLOPs	AP	AP ⁵⁰	AP ⁷⁵	AP ^M	AP ^L	AR
<i>Large networks</i>											
8-stage Hourglass [31]	8-stage Hourglass	N	256 × 192	25.1M	14.3	66.9	—	—	—	—	—
CPN [7]	ResNet-50 [15]	Y	256 × 192	27.0M	6.20	68.6	—	—	—	—	—
SimpleBaseline [45]	ResNet-50	Y	256 × 192	34.0M	8.90	70.4	88.6	78.3	67.1	77.2	76.3
HRNetV1 [41]	HRNetV1-W32	N	256 × 192	28.5M	7.10	73.4	89.5	80.7	70.2	80.1	78.9
DARK [51]	HRNetV1-W48	Y	128 × 96	63.6M	3.6	71.9	89.1	79.6	69.2	78.0	77.9
<i>Small networks</i>											
MobileNetV2 1 × [36]	MobileNetV2	Y	256 × 192	9.6M	1.48	64.6	87.4	72.3	61.1	71.2	70.7
MobileNetV2 1 ×	MobileNetV2	Y	384 × 288	9.6M	3.33	67.3	87.9	74.3	62.8	74.7	72.9
ShuffleNetV2 1 × [28]	ShuffleNetV2	Y	256 × 192	7.6M	1.28	59.9	85.4	66.3	56.6	66.2	66.4
ShuffleNetV2 1 ×	ShuffleNetV2	Y	384 × 288	7.6M	2.87	63.6	86.5	70.5	59.5	70.7	69.7
Small HRNet	HRNet-W16	N	256 × 192	1.3M	0.54	55.2	83.7	62.4	52.3	61.0	62.1
Small HRNet	HRNet-W16	N	384 × 288	1.3M	1.21	56.0	83.8	63.0	52.4	62.6	62.6
DY-MobileNetV2 1 × [5]	DY-MobileNetV2	Y	256 × 192	16.1M	1.01	68.2	88.4	76.0	65.0	74.7	74.2
DY-ReLU 1 × [6]	MobileNetV2	Y	256 × 192	9.0M	1.03	68.1	88.5	76.2	64.8	74.3	—
Lite-HRNet	Lite-HRNet-18	N	256 × 192	1.1M	0.20	64.8	86.7	73.0	62.1	70.5	71.2
Lite-HRNet	Lite-HRNet-18	N	384 × 288	1.1M	0.45	67.6	87.8	75.0	64.5	73.7	73.7
Lite-HRNet	Lite-HRNet-30	N	256 × 192	1.8M	0.31	67.2	88.0	75.0	64.3	73.1	73.3
Lite-HRNet	Lite-HRNet-30	N	384 × 288	1.8M	0.70	70.4	88.7	77.7	67.5	76.3	76.2

Each module consists of two conditional channel weighting blocks and one multi-resolution fusion. Each resolution branch’s channel dimensions are C , $2C$, $4C$, and $8C$, respectively. Table 1 describes the detailed structures.

Connection. The conditional channel weighting scheme shares the same philosophy to the conditional convolutions [47], dynamic filters [21], and squeeze-excite-network [19]. Those works learn the convolution kernels or the mixture weights by sub-network conditioned on the input features for increasing the model capacity. Our method instead exploits an extra effect and uses the weights learned from all the channels as a bridge to exchange information across channels and resolutions. It can replace costly 1×1 convolutions in lightweight networks. Besides, we introduce multi-resolution information to boost weight learning.

4. Experiments

We evaluate our approach on two human pose estimation datasets, COCO [27] and MPII [1]. Following the state-

of-the-art top-down framework, our approach estimates K heatmaps to indicate the keypoint location confidence. We perform a comprehensive ablation on COCO and report the comparisons with other methods on both datasets.

4.1. Setting

Datasets. COCO [27] has over 200K images and 250K person instances with 17 keypoints. Our models are trained on `train2017` dataset (includes 57K images and 150K person instances) and validated on `val2017` (includes 5K images) and `test-dev2017` (includes 20K images).

The MPII Human Pose dataset [1] contains around 25K images with full-body pose annotations taken from real-world activities. There are over 40K person instances, split 12K instances for testing, and others for training.

Training. The network is trained on 8 NVIDIA V100 GPUs with mini-batch size 32 per GPU. We adopt Adam optimizer with an initial learning rate of $2e^{-3}$.

The human detection boxes are expanded to have a fixed

Table 4. **Comparisons on the COCO test-dev set.** #Params and FLOPs are calculated for the pose estimation network, and those for human detection and keypoint grouping are not included.

model	backbone	input size	#Params	GFLOPs	AP	AP ⁵⁰	AP ⁷⁵	AP ^M	AP ^L	AR
<i>Large networks</i>										
Mask-RCNN [14]	ResNet-50-FPN	—	—	—	63.1	87.3	68.7	57.8	71.4	—
G-RMI [33]	ResNet-101	353 × 257	42.6M	57.0	64.9	85.5	71.3	62.3	70.0	69.7
Integral Pose Regression [38]	ResNet-101	256 × 256	45.0M	11.0	67.8	88.2	74.8	63.9	74.0	—
CPN [7]	ResNet-Inception	384 × 288	—	—	72.1	91.4	80.0	68.7	77.2	78.5
RMPE [13]	PyraNet [48]	320 × 256	28.1M	26.7	72.3	89.2	79.1	68.0	78.6	—
SimpleBaseline [45]	ResNet-152	384 × 288	68.6M	35.6	73.7	91.9	81.1	70.3	80.0	79.0
HRNetV1 [41]	HRNetV1-W32	384 × 288	28.5M	16.0	74.9	92.5	82.8	71.3	80.9	80.1
HRNetV1 [41]	HRNetV1-W48	384 × 288	63.6M	32.9	75.5	92.5	83.3	71.9	81.5	80.5
DARK [51]	HRNetV1-W48	384 × 288	63.6M	32.9	76.2	92.5	83.6	72.5	82.4	81.1
<i>Small networks</i>										
MobileNetV2 1 ×	MobileNetV2	384 × 288	9.8M	3.33	66.8	90.0	74.0	62.6	73.3	72.3
ShuffleNetV2 1 ×	ShuffleNetV2	384 × 288	7.6M	2.87	62.9	88.5	69.4	58.9	69.3	68.9
Small HRNet	HRNet-W16	384 × 288	1.3M	1.21	55.2	85.8	61.4	51.7	61.2	61.5
Lite-HRNet	Lite-HRNet-18	384 × 288	1.1M	0.45	66.9	89.4	74.4	64.0	72.2	72.6
Lite-HRNet	Lite-HRNet-30	384 × 288	1.8M	0.70	69.7	90.7	77.5	66.9	75.0	75.4

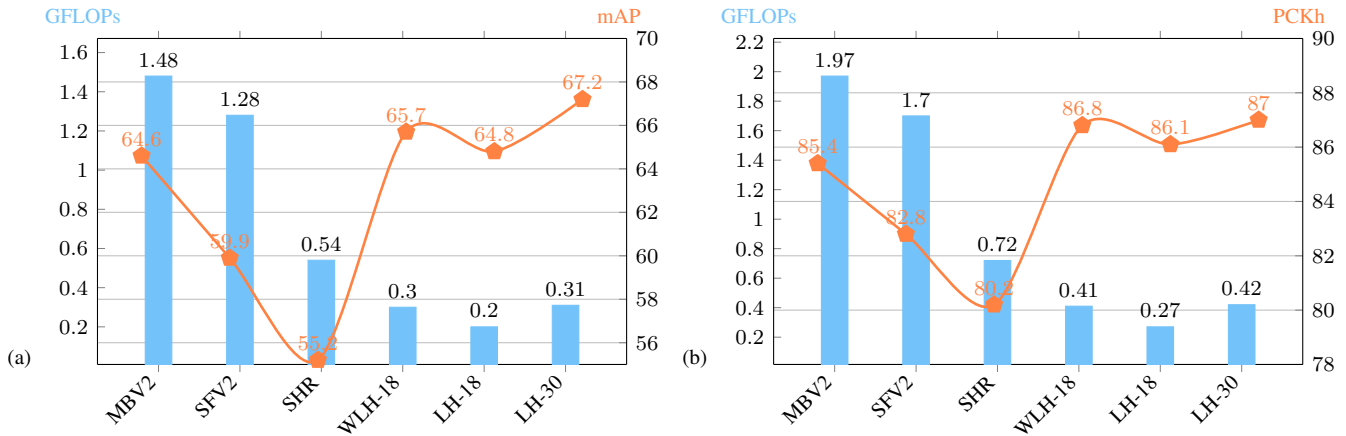


Figure 4. **Illustration of the complexity and accuracy comparison on the COCO val1 and MPII val1 sets.** (a) Comparison on COCO val1 with 256×192 input size. (b) Comparison on MPII val1 with 256×256 input size. MBV2= MobileNet V2. SFV2= ShuffleNet V2. SHR= Small HRNet-W16. (W)LH= (Wider Naive) Lite-HRNet.

Table 5. **Comparisons on the MPII val1 set.** The FLOPs is computed with the input size 256×256 .

model	#Params	GFLOPs	PCKh
MobileNetV2 1 ×	9.6M	1.97	85.4
MobileNetV3 1 ×	8.7M	1.82	84.3
ShuffleNetV2 1 ×	7.6M	1.70	82.8
Small HRNet-W16	1.3M	0.72	80.2
Lite-HRNet-18	1.1M	0.27	86.1
Lite-HRNet-30	1.8M	0.42	87.0

aspect ratio of 4: 3, and then crop the box from the images. The image size is resized to 256×192 or 384×288 for COCO, and 256×256 for MPII. Each image will go through

a series of data augmentation operations, containing random rotation ($[-30^\circ, 30^\circ]$), random scale ($[0.75, 1.25]$), and random flipping for both datasets and additional half body data augmentation for COCO.

Testing. For COCO, following [45, 7, 33], we adopt the two-stage top-down paradigm (detect the person instance via a person detector and predict keypoints) with the person detectors provided by SimpleBaseline [45]. For MPII, we adopt the standard testing strategy to use the provided person boxes. We estimate the heat maps via a post-gaussian filter and average the original and flipped images’ predicted heat maps. A quarter offset in the direction from the highest response to the second-highest response is applied to obtain each keypoint location.

Table 6. **Ablation about conditional channel weight vs. 1×1 convolutions** on the COCO `val` and MPII `val` sets. The input size of COCO is 256×192 , while 256×256 for MPII. Wider NLite-NRNet = wider naive Lite-HRNet.

model	#Params	COCO					MPII	
		MFLOPs	AP	AP ⁵⁰	AP ⁷⁵	AR	MFLOPs	PCKh
Small HRNet-W16	1.3M	551.7	55.2	83.7	62.4	62.1	735.5	80.2
Naive Lite-HRNet-18	0.7M	194.8	62.5	85.4	69.6	68.8	259.6	85.3
Wider Naive Lite-HRNet-18	1.3M	311.1	65.7	87	73.3	71.8	418.7	86.8
Wider NLite-HRNet-18 (<i>one 1×1 conv. dropped</i>)	1.1M	248.4	63.6	86.1	70.7	69.8	331.0	86.0
Wider NLite-HRNet-18 (<i>two 1×1 conv. dropped</i>)	0.9M	188.9	61.3	85.3	68.7	67.7	251.7	85.3
Lite-HRNet-18	1.1M	205.2	64.8	86.7	73.0	71.2	273.4	86.1
Lite-HRNet-30	1.8M	319.2	67.2	88.0	75.0	73.3	425.3	87.0

Evaluation. We adopt the OKS-based mAP metric on COCO, where OKS (Object Keypoint Similarity) defines the similarity between different human poses. We report standard average precision and recall scores: AP (the mean of AP scores at 10 positions, OKS = 0.50, 0.55, ..., 0.90, 0.95), AP⁵⁰ (AP at OKS = 0.50), AP⁷⁵, AR and AR⁵⁰. For MPII, we use the standard metric PCKH@0.5 (head-normalized probability of correct keypoint) to evaluate the performance.

4.2. Results

COCO `val`. The results of our method and other state-of-the-art methods are reported in Table 3. Our Lite-HRNet-30, trained from scratch with the 256×192 input size, achieves 67.2 AP score, outperforming other light-weight methods. Compared to MobileNetV2, Lite-HRNet improves AP by 2.6 points with only 20% GFLOPs and parameters. Compared to ShuffleNetV2, our Lite-HRNet-18 and Lite-HRNet-30 achieve 4.9 and 7.3 points gain, respectively. The complexity of our network is much lower than ShuffleNetV2. Compared to Small HRNet-W16, Lite-HRNet improves over 10 AP points. Compared to large networks, e.g., Hourglass and CPN, our networks achieve comparable AP score with far lower complexity.

With the input size 384×288 , our Lite-HRNet-18 and Lite-HRNet-30 achieve 67.6 and 70.4 AP, respectively. Due to the efficient conditional channel weighting, Lite-HRNet achieves a better balance between accuracy and computational complexity, as shown in Figure 4 (a). Figure 3 shows the visual results on COCO from Lite-HRNet-30.

COCO `test-dev`. Table 4 reports the comparison results of our networks and other state-of-the-art methods. Our Lite-HRNet-30 achieves 69.7 AP score. It is significantly better than the small networks, and is more efficient in terms of GFLOPs and parameters. Compared to the large networks, our Lite-HRNet-30 outperforms Mask-RCNN [14], G-RMI [33], and Integral Pose Regression [38]. Although there is a performance gap with some large networks, our networks have far lower GFLOPs and parameters.

MPII `val`. Table 5 reports the results of our net-

works and other lightweight networks. Our Lite-HRNet-18 achieves better accuracy with much lower GFLOPs than MobileNetV2, MobileNetV3, ShuffleNetV2, Small HRNet-W16. With increasing the model size, as Lite-HRNet-30, the improvement gap becomes larger. Our Lite-HRNet-30 achieves 87.0 PCKh@0.5, improving MobileNetV2, MobileNetV3, ShuffleNetV2 and Small HRNet-W16 by 1.6, 2.7, 4.2, and 6.8 points, respectively. Figure 4 (b) shows the comparison of accuracy and complexity.

4.3. Ablation Study

We perform ablations on two datasets: COCO and MPII, and report the results on the validation sets. The input size is 256×192 for COCO, and 256×256 for MPII.

Naive Lite-HRNet vs. Small HRNet. We empirically study that the shuffle blocks combined into HRNet improve performance. Figure 4 shows the comparison to Small HRNet-W16⁴. We can see that naive Lite-HRNet achieves higher AP scores with lower computation complexity. On COCO `val`, naive Lite-HRNet improves AP over the Small HRNet-W16 by 7.3 points, and the GFLOPs and parameters are less than half. When increasing to similar parameters as wider naive Lite-HRNet, the improvement is enlarged to 10.5 points, as shown in Figure 4 (a). On MPII `val`, naive Lite-HRNet outperforms the Small HRNet-W16 by 5.1 points, while the wider network outperforms 6.6 points, as illustrated in Figure 4 (b).

Conditional channel weighting vs. 1×1 convolution. We compare the performance between 1×1 convolution (wider naive Lite-HRNet) and conditional channel weighting (Lite-HRNet). We simply remove one or two 1×1 convolutions in the shuffle blocks in wider naive Lite-HRNet.

Table 6 shows the studies on the COCO `val` and MPII `val` sets. 1×1 convolutions can exchange the information across channels, important to representation learning. On COCO `val`, dropping two 1×1 convolutions leads to 4.4 AP points decrease for wider naive Lite-HRNet, and also reduces almost 40% FLOPs.

⁴Available from <https://github.com/HRNet/HRNet-Semantic-Segmentation>

Table 7. **Ablation about spatial and multi-resolution weights.** on the COCO val and MII val sets. The input size of COCO is 256×192 , while 256×256 for MII. CCW=conditional channel weight computation, Wider NLite-NRNet = wider naive Lite-HRNet.

model	#Params	COCO					MII	
		MFLOPs	AP	AP ⁵⁰	AP ⁷⁵	AR	MFLOPs	PCKh
Wider NLite-HRNet-18 (<i>two 1×1 conv. dropped</i>)	0.9M	188.9	61.3	85.3	68.7	67.7	251.7	85.3
Lite-HRNet-18 (<i>CCW only w/ spatial weights</i>)	0.9M	190.6	62.6	85.8	69.8	69.1	254.0	85.4
Lite-HRNet-18 (<i>CCW only w/ multi-resolution weights</i>)	0.9M	203.5	63.0	85.7	70.5	69.4	271.1	85.8
Lite-HRNet-18	1.1M	205.2	64.8	86.7	73.0	71.2	273.4	86.1
Lite-HRNet-30	1.8M	319.2	67.2	88.0	75.0	73.3	425.3	87.0

Our conditional channel weighting improves by 3.5 AP points over dropping two 1×1 convolutions with only increasing 16M FLOPs. The AP score is comparable with the wider naive Lite-HRNet by using only 65% FLOPs. Increasing the depth of Lite-HRNet leads to 1.5 AP improvements with similar FLOPs as wider naive Lite-HRNet and slightly larger #parameters than wider naive Lite-HRNet. The observations on MII val are consistent (see Table 6). The AP improvement is because that our lightweight weighting operations can make the network capacity improved, by exploring the multi-resolution information using cross-resolution channel weighting and deepening the network, if taking similar FLOPs with naive version.

Spatial and multi-resolution weights. We empirically study how spatial weights and multi-resolution weights influence the performance, as shown in Table 7.

On COCO val, the spatial weights achieve 1.3 AP increase, and the multi-resolution weights obtain 1.7 point gain. The FLOPs of both operations are cheap. With both spatial and cross-resolution weights, our network improves by 3.5 points. Table 7 reports the consistent improvements on MII val. These studies validate the efficiency and effectiveness of the spatial and cross-resolution weights.

We conduct the experiments by changing the arrangement order of the spatial weighting and cross-resolution weighting, which achieves similar performance. The experiments with only two spatial weights or two cross-resolution weights, lead to an almost 0.3 drop.

4.4. Application to Semantic Segmentation

Dataset. Cityscapes [10] includes 30 classes and 19 of them are used for semantic segmentation task. The dataset contains 2,975, 500, and 1,525 finely-annotated images for training, validation, and test sets, respectively. In our experiments, we only use the fine annotated images.

Training. Our models are trained from scratch with the SGD algorithm [22]. The initial rate is set to $1e^{-2}$ with a ‘‘poly’’ learning rate strategy [50, 49] with a multiplier of $(1 - \frac{iter}{max.iter})^{0.9}$ each iteration. The total iterations are 160K with 16 batch size, and the weight decay is $5e^{-4}$. We randomly horizontally flip, scale ([0.5, 2]), and crop the input images to a fixed size (512×1024) for training.

Table 8. **Segmentation results on Cityscapes.** P = pretrain the backbone on ImageNet. * indicates the complexity is estimated from the original paper.

model	P	#Params	GFLOPs	resolution	val	test
<i>Hand-crafted networks</i>						
ICNet [55]	Y	–	28.3	1024×2048	67.7	69.5
BiSeNetV1 A [50]	Y	5.8M	14.8	768×1536	69.0	68.4
BiSeNetV1 B [50]	Y	49.0M	55.3	768×1536	74.8	74.7
DFANet A’ [23]	Y	7.8M	1.7	512×1024	–	70.3
SwiftNet [32]	Y	11.8M	26.0	512×1024	70.2	–
SwiftNet [32]	Y	11.8M	104	1024×2048	75.4	75.5
Fast-SCNN [35]	N	–	–	1024×2048	68.6	68.0
ShelfNet [58]	Y	–	36.9	1024×2048	–	74.8
BiSeNetV2 Small [49]	N	–	21.15	512×1024	73.4	72.6
MoibleNeXt [12]	Y	4.5M	10.1*	1024×2048	75.5	–
MobileNet V2 0.5 [36]	Y	0.3M	3.73	512×1024	68.6	–
HRNet-W16 [41]	Y	2.0M	7.8	512×1024	68.6	–
<i>NAS-based networks</i>						
CAS [54]	Y	–	–	768×1536	71.6	70.5
DF1-Seg-d8 [24]	Y	–	–	1024×2048	72.4	71.4
FasterSeg [4]	Y	4.4M	28.2	1024×2048	73.1	71.5
GAS [25]	Y	–	–	769×1537	–	71.8
MobileNetV3 [16]	Y	1.5M	9.1	1024×2048	72.4	72.6
MobileNet V3-Small	Y	0.5M	2.7	512×1024	68.4	69.4
Lite-HRNet-18	N	1.1M	1.95	512×1024	73.8	72.8
Lite-HRNet-30	N	1.8M	3.02	512×1024	76.0	75.3

Results. We do not adopt testing tricks, e.g., sliding-window and multi-scale evaluation, beneficial to performance improvement but time-consuming. Table 8 shows that Lite-HRNet-18 achieves 72.8% mIoU with only 1.95 GFLOPs and Lite-HRNet-30 achieves 75.3% mIoU with 3.02 GFLOPs, outperforming the hand-crafted methods [34, 55, 50, 23, 58, 36, 41, 12] and NAS-based methods [54, 25, 16, 24], and comparable with SwiftNetRN-18 [32] that is far computationally intensive (104 GFLOPs).

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61433007 and 61876210).

References

- [1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3686–3693, 2014. 5
- [2] 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*, 39(12):2481–2495, 2017. 1, 2
- [3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *Proc. International Conference on Learning Representations (ICLR)*, 2015. 1
- [4] Wuyang Chen, Xinyu Gong, Xianming Liu, Qian Zhang, Yuan Li, and Zhangyang Wang. Fasterseg: Searching for faster real-time semantic segmentation. *Proc. International Conference on Learning Representations (ICLR)*, 2020. 8
- [5] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11030–11039, 2020. 1, 2, 5
- [6] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic relu. In *Proc. European Conference on Computer Vision (ECCV)*, August 2020. 1, 5
- [7] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7103–7112, 2018. 5, 6
- [8] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5386–5395, 2020. 2
- [9] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3
- [10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8
- [11] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017. 3
- [12] Zhou Daquan, Qibin Hou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Rethinking bottleneck structure for efficient mobile network design. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 8
- [13] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 2334–2343, 2017. 6
- [14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 6, 7
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5
- [16] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 8
- [17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv*, 2017. 1, 2
- [18] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 9401–9411, 2018. 2
- [19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 5
- [20] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 2017–2025, 2015. 2
- [21] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 667–675, 2016. 2, 5
- [22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2012. 8
- [23] Hanchao Li, Pengfei Xiong, Haoqiang Fan, and Jian Sun. Dfanet: Deep feature aggregation for real-time semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 8
- [24] Xin Li, Yiming Zhou, Zheng Pan, and Jiashi Feng. Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9145–9153, 2019. 8
- [25] Peiwen Lin, Peng Sun, Guangliang Cheng, Sirui Xie, Xi Li, and Jianping Shi. Graph-guided architecture search for real-time semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4203–4212, 2020. 8
- [26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid

- networks for object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017. 1
- [27] 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 *Proc. European Conference on Computer Vision (ECCV)*, 2014. 5
- [28] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proc. European Conference on Computer Vision (ECCV)*, pages 116–131, 2018. 1, 2, 3, 5
- [29] Tsendsuren Munkhdalai and Hong Yu. Meta networks. *Proceedings of machine learning research*, 70:2554, 2017. 2
- [30] Christopher Neff, Aneri Sheth, Steven Furgurson, and Hamed Tabkhi. Efficienthrnet: Efficient scaling for lightweight high-resolution multi-person pose estimation. *arXiv*, 2020. 2
- [31] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proc. European Conference on Computer Vision (ECCV)*, pages 483–499. Springer, 2016. 5
- [32] Marin Orsic, Ivan Kreso, Petra Bevandic, and Sinisa Segvic. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12607–12616, 2019. 8
- [33] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4903–4911, 2017. 6, 7
- [34] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv*, 2016. 2, 8
- [35] Rudra PK Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-scnn: fast semantic segmentation network. *arXiv*, 2019. 8
- [36] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1801, 2018. 2, 5, 8
- [37] Ke Sun, Mingjie Li, Dong Liu, and Jingdong Wang. IGCv3: interleaved low-rank group convolutions for efficient deep neural networks. In *Proc. the British Machine Vision Conference (BMVC)*, 2018. 2
- [38] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proc. European Conference on Computer Vision (ECCV)*, pages 529–545, 2018. 6, 7
- [39] Mingxing Tan and Quoc V. Le. Mixconv: Mixed depthwise convolutional kernels. *arXiv*, 2019. 2
- [40] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2
- [41] Jingdong 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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1, 3, 5, 6, 8
- [42] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *Proc. European Conference on Computer Vision (ECCV)*, pages 409–424, 2018. 2
- [43] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. 2
- [44] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proc. European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 2
- [45] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proc. European Conference on Computer Vision (ECCV)*, pages 466–481, 2018. 5, 6
- [46] Guotian Xie, Jingdong Wang, Ting Zhang, Jianhuang Lai, Richang Hong, and Guo-Jun Qi. Interleaved structured sparse convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [47] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 1307–1318, 2019. 2, 5
- [48] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1281–1290, 2017. 6
- [49] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *arXiv*, 2020. 2, 8
- [50] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proc. European Conference on Computer Vision (ECCV)*, pages 325–341, 2018. 1, 2, 8
- [51] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7093–7102, 2020. 5, 6
- [52] Ting Zhang, Guo-Jun Qi, Bin Xiao, and Jingdong Wang. Interleaved group convolutions. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017. 2
- [53] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6848–6856, 2018. 1, 2

- [54] Yiheng Zhang, Zhaofan Qiu, Jingen Liu, Ting Yao, Dong Liu, and Tao Mei. Customizable architecture search for semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [8](#)
- [55] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proc. European Conference on Computer Vision (ECCV)*, pages 405–420, 2018. [1](#), [8](#)
- [56] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [2](#)
- [57] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9308–9316, 2019. [3](#)
- [58] Juntang Zhuang and Junlin Yang. Shelfnet for real-time semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [8](#)