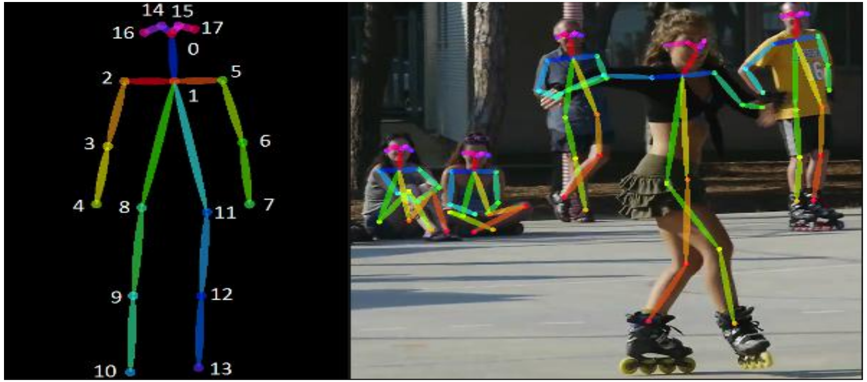
Lite-HRNet: A Lightweight High-Resolution Network

Changqian Yu[1,2] Bin Xiao[2] Changxin Gao[1] Lu Yuan[2] Lei Zhang[2] Nong Sang[1*] Jingdong Wang[2*]
[1]Key Laboratory of Image Processing and Intelligent Control
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2]Microsoft

Pose Estimation




```
BODY_PARTS = {"Nose": 0, "Neck": 1, "RShoulder": 2, "RElbow": 3, "RWrist": 4,
              "LShoulder": 5, "LElbow": 6, "LWrist": 7, "RHip": 8, "RKnee": 9,
              "RAnkle": 10, "LHip": 11, "LKnee": 12, "LAnkle": 13, "REye": 14,
              "LEye": 15, "REar": 16, "LEar": 17, "Background": 18}

POSE_PAIRS = [["Neck", "RShoulder"], ["Neck", "LShoulder"], ["RShoulder", "RElbow"],
              ["RElbow", "RWrist"], ["LShoulder", "LElbow"], ["LElbow", "LWrist"],
              ["Neck", "RHip"], ["RHip", "RKnee"], ["RKnee", "RAnkle"], ["Neck", "LHip"],
              ["LHip", "LKnee"], ["LKnee", "LAnkle"], ["Neck", "Nose"], ["Nose", "REye"],
              ["REye", "REar"], ["Nose", "LEye"], ["LEye", "LEar"]]
```

评价指标：

PCK： 关键点与其对应的groundtruth间的归一化距离小于设定阈值的比例。

OKS： 关键点与其对应的groundtruth间的相似度度量，[0,1]。

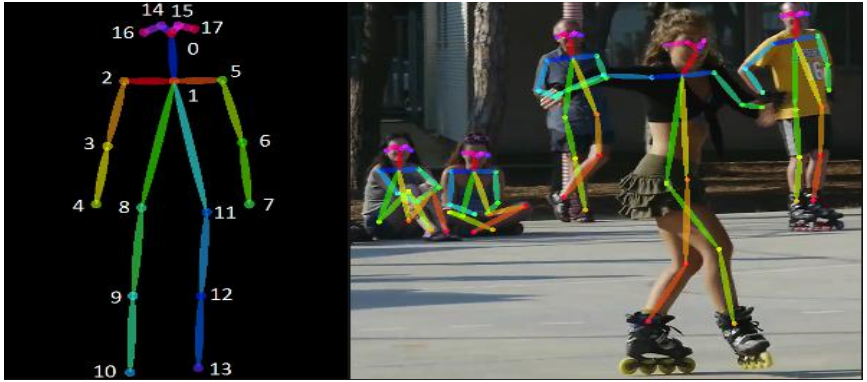
PCKh： 以头部长度(head length：) 作为归一化参考。

pckh@0.5(MPII)**：** 0.5表示以头部长度作为参考，如果归一化后的距离大于阈值0.5，则认为预测正确。最好计算检测正确的比例。

OKS m**AP**(coco)**：**

$$Precision = \frac{tp}{tp + fp}$$

Pose Estimation



评价指标：

PCK： 关键点与其对应的groundtruth间的归一化距离小于设定阈值的比例。

OKS： 关键点与其对应的groundtruth间的相似度度量，[0,1]。

PCKh： 以头部长度(head length：) 作为归一化参考。

pckh@0.5(MPII)**：** 0.5表示以头部长度作为参考，如果归一化后的距离大于阈值0.5，则认为预测正确。最好计算检测正确的比例。

OKS m**AP**(coco)**：**

$$Precision = \frac{tp}{tp + fp}$$

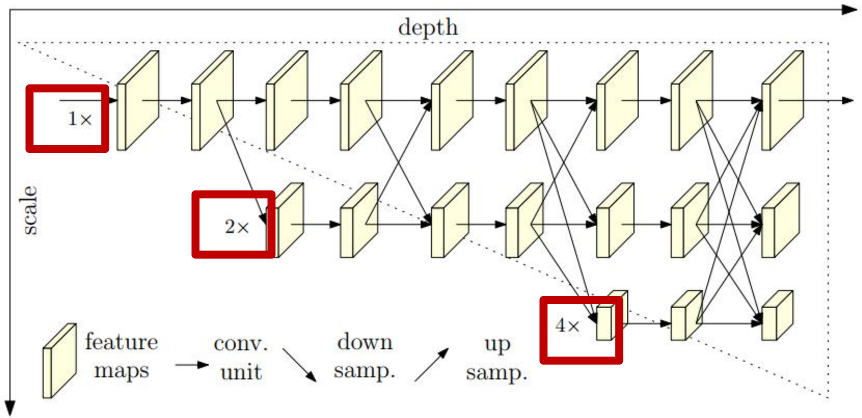
Lite-HRNet

- 1. Shuffle Block和Small HRNet简单融合，能够得到轻量化的HRNet
- 2. Naive Lite-HRNet中存在大量的1x1卷积操作，中使用conditional channel weighting模块替代卷积，以进一步提高网络的计算效率。

HRNet(High-Resoultion Net)

- Deep High-Resolution Representation Learning for Human Pose Estimation
- 微软亚洲研究院和中科大提出PAMI2019

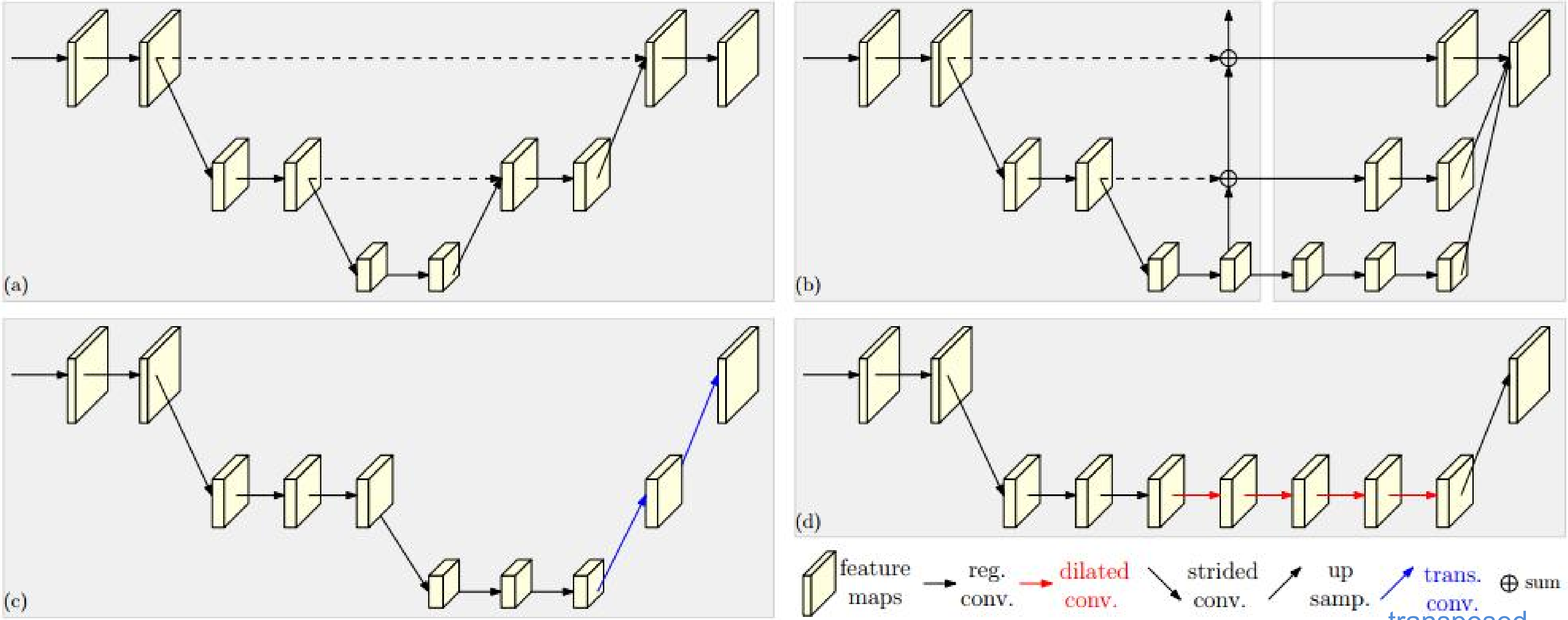
- **特点与优势：**
- （1） HRNet能够保持高分辨率， HRNet之前算法是通过：high-to-low and low-to-high framework：**将高分辨率特征图下采样至低分辨率，再从低分辨率特征图恢复至高分辨率的思路**（U-Net， encoder-decoder ）。



HRNet(High-Resoultion Net)

- Deep High-Resolution Representation Learning for Human Pose Estimation
- 微软亚洲研究院和中科大提出PAMI2019

high-to-low and low-to-high framework：

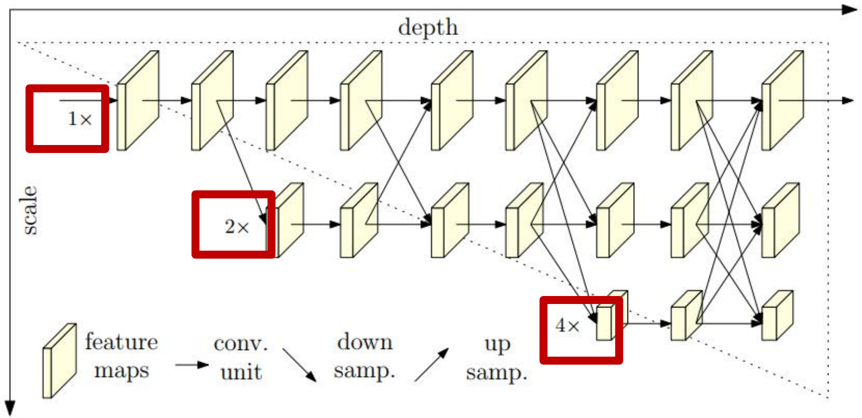


transposed convolution

HRNet(High-Resoultion Net)

- Deep High-Resolution Representation Learning for Human Pose Estimation
- 微软亚洲研究院和中科大提出PAMI2019

- **特点与优势：**
- （1） HRNet能够保持高分辨率， HRNet之前算法是通过：high-to-low and low-to-high framework：**将高分辨率特征图下采样至低分辨率，再从低分辨率特征图恢复至高分辨率的思路**（U-Net， encoder-decoder ）。（2）融合**相同深度和相似级别**的低分辨率特征图来提高高分辨率的特征图的表示效果，并进行重复的多尺度融合。



HRNet(High-Resoultion Net)

- Deep High-Resolution Representation Learning for Human Pose Estimation
- 微软亚洲研究院和中科大提出PAMI2019

contains four stages with
four parallel subnetworks

stages

$$
\begin{array}{ccccccc}
\mathcal{N}_{11} & \rightarrow & \mathcal{N}_{21} & \rightarrow & \mathcal{N}_{31} & \rightarrow & \mathcal{N}_{41} \\
 & \searrow & \mathcal{N}_{22} & \rightarrow & \mathcal{N}_{32} & \rightarrow & \mathcal{N}_{42} \\
 & & & \searrow & \mathcal{N}_{33} & \rightarrow & \mathcal{N}_{43} \\
 & & & & & \searrow & \mathcal{N}_{44}
\end{array}
$$

Stage内部没有交互，参数不共享。

parallel
subnetworks

$$
\begin{array}{ccccccc}
\mathcal{N}_{11} & \rightarrow & \mathcal{N}_{21} & \rightarrow & \mathcal{N}_{31} & \rightarrow & \mathcal{N}_{41} \\
 & \searrow & \mathcal{N}_{22} & \rightarrow & \mathcal{N}_{32} & \rightarrow & \mathcal{N}_{42} \\
 & & & \searrow & \mathcal{N}_{33} & \rightarrow & \mathcal{N}_{43} \\
 & & & & & \searrow & \mathcal{N}_{44}
\end{array}
$$

parallel subnetworks内部分辨率相同

HRNet(High-Resoultion Net)

- Deep High-Resolution Representation Learning for Human Pose Estimation
- 微软亚洲研究院和中科大提出PAMI2019

$$
\begin{array}{ccccccc}
\mathcal{N}_{11} & \rightarrow & \mathcal{N}_{21} & \rightarrow & \mathcal{N}_{31} & \rightarrow & \mathcal{N}_{41} \\
 & \searrow & \mathcal{N}_{22} & \rightarrow & \mathcal{N}_{32} & \rightarrow & \mathcal{N}_{42} \\
 & & & \searrow & \mathcal{N}_{33} & \rightarrow & \mathcal{N}_{43} \\
 & & & & & \searrow & \mathcal{N}_{44}
\end{array}
$$

big net HRNet-W48
small net HRNet-W32

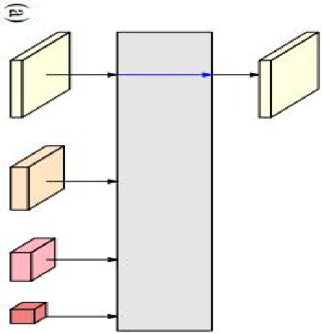
32 and 48 represent the NUM_CHANNELS of the high-resolution subnetworks in last three stages, respectively.

The widths of other three parallel subnetworks：
64; 128; 256 for HRNet-W32,
96; 192; 384 for HRNet-W48.

HRNet(High-Resoultion Net)

- Deep High-Resolution Representation Learning for Human Pose Estimation
- 微软亚洲研究院和中科大提出PAMI2019

$$
\begin{array}{ccccccc}
\mathcal{N}_{11} & \rightarrow & \mathcal{N}_{21} & \rightarrow & \mathcal{N}_{31} & \rightarrow & \mathcal{N}_{41} \\
 & \searrow & \mathcal{N}_{22} & \rightarrow & \mathcal{N}_{32} & \rightarrow & \mathcal{N}_{42} \\
 & & & \searrow & \mathcal{N}_{33} & \rightarrow & \mathcal{N}_{43} \\
 & & & & & \searrow & \mathcal{N}_{44}.
\end{array}
$$



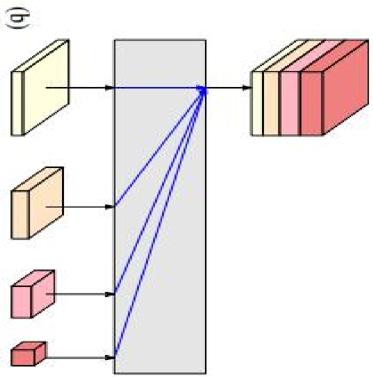
HRNetV1：只使用分辨率最高的特征图，人体姿态估计。

HRNetV2

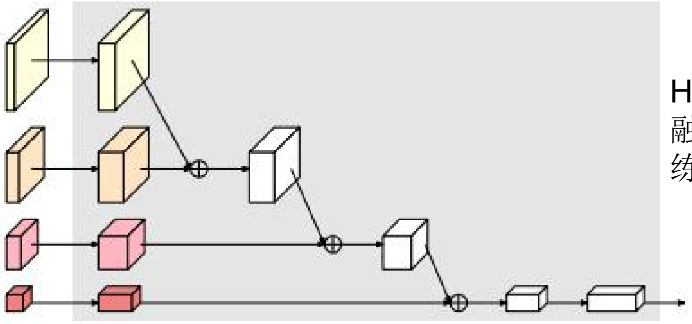
- High-Resolution Representations for Labeling Pixels and Regions，简称HRNet V2，发表于CVPR2019

HRNetV1和HRNetV2其实不是版本迭代
的过程,只是同一个网络用在不同任务上

$$
\begin{array}{ccccccc}
\mathcal{N}_{11} & \rightarrow & \mathcal{N}_{21} & \rightarrow & \mathcal{N}_{31} & \rightarrow & \mathcal{N}_{41} \\
 & \searrow & \mathcal{N}_{22} & \rightarrow & \mathcal{N}_{32} & \rightarrow & \mathcal{N}_{42} \\
 & & & \searrow & \mathcal{N}_{33} & \rightarrow & \mathcal{N}_{43} \\
 & & & & & \searrow & \mathcal{N}_{44}.
\end{array}
$$



HRNetV2：将所有分辨率的特征图进
行concate，主要用于语义分割和面部
关键点检测

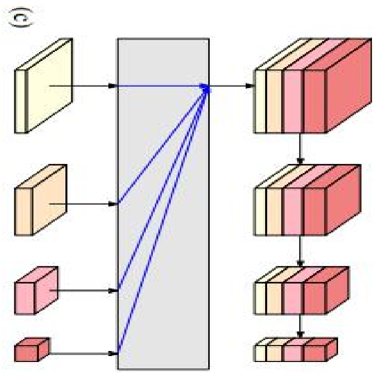


HRNetV2：采用上图的
融合方式，主要用于训
练分类网络。

HRNet(High-Resoultion Net)

- Deep High-Resolution Representation Learning for Human Pose Estimation
- 微软亚洲研究院和中科大提出CVPR2019

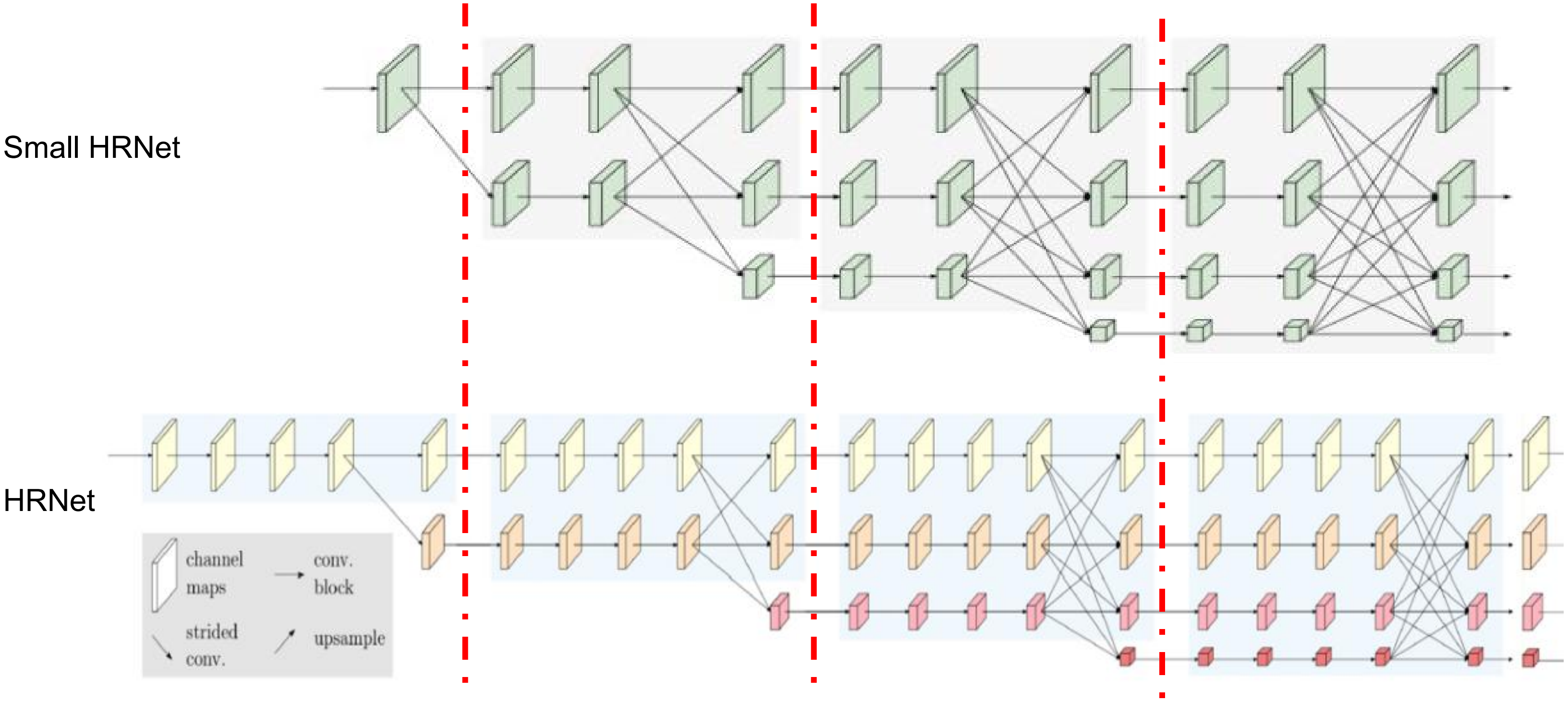
$$
\begin{array}{ccccccc}
\mathcal{N}_{11} & \rightarrow & \mathcal{N}_{21} & \rightarrow & \mathcal{N}_{31} & \rightarrow & \mathcal{N}_{41} \\
 & \searrow & \mathcal{N}_{22} & \rightarrow & \mathcal{N}_{32} & \rightarrow & \mathcal{N}_{42} \\
 & & & \searrow & \mathcal{N}_{33} & \rightarrow & \mathcal{N}_{43} \\
 & & & & & \searrow & \mathcal{N}_{44}.
\end{array}
$$



HRNetV2p：在HRNetV2的基础上，使用了一个特征金字塔，主要用于目标检测网络

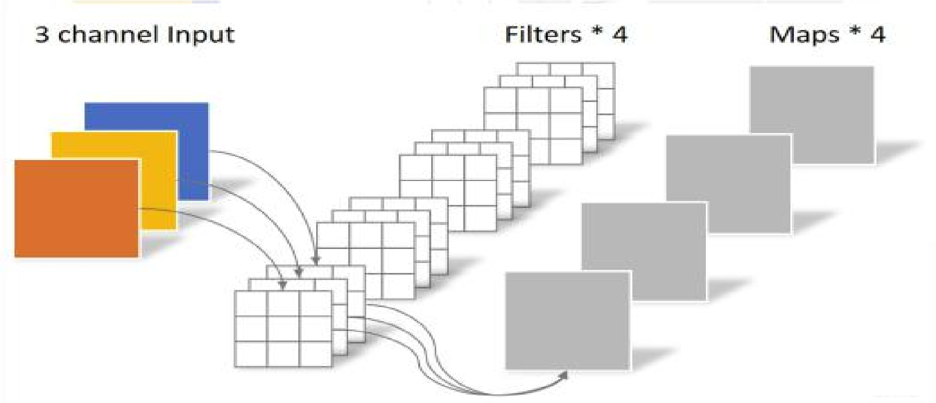
Small HRNet

- https://github.com/HRNet/HRNet-Semantic-Segmentation
- It simply reduces the depth and the width of the original HRNet.

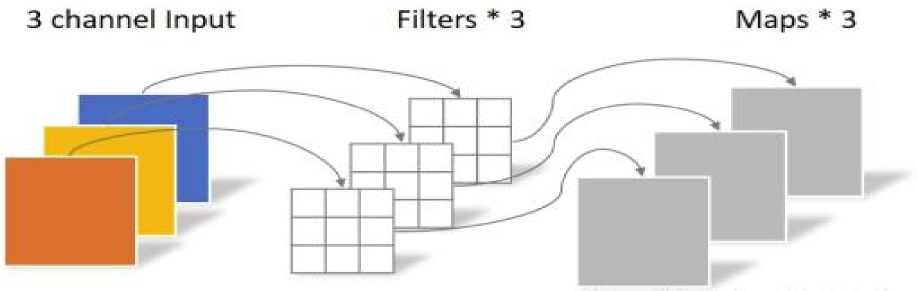


ShuffleNet

- ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices
- Face++ CVPR 2017



convolution

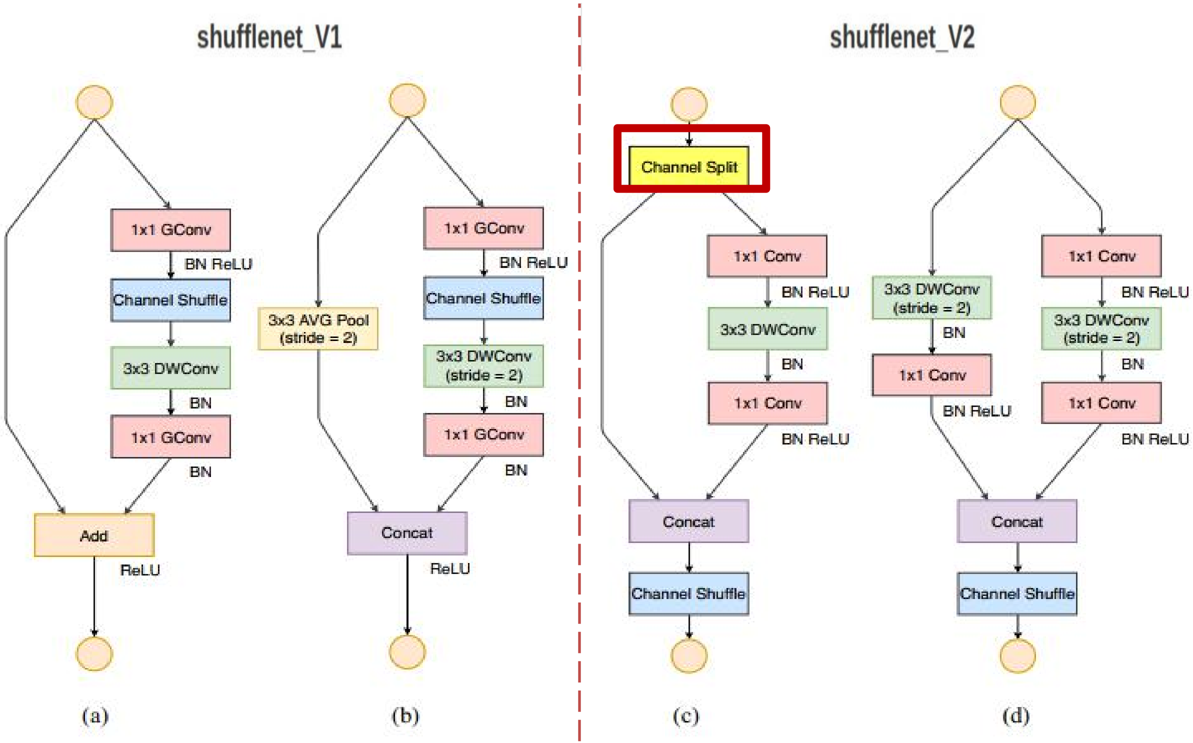


DWConv: depthwise convolution

ShuffleNetV2

- ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design
- Face++ & 清华大学 *ECCV* 2018

- 1. channel split: 将输入的feature maps分为两部分c’和c-c’ .



DWConv: depthwise convolution
Gconv: group convolution

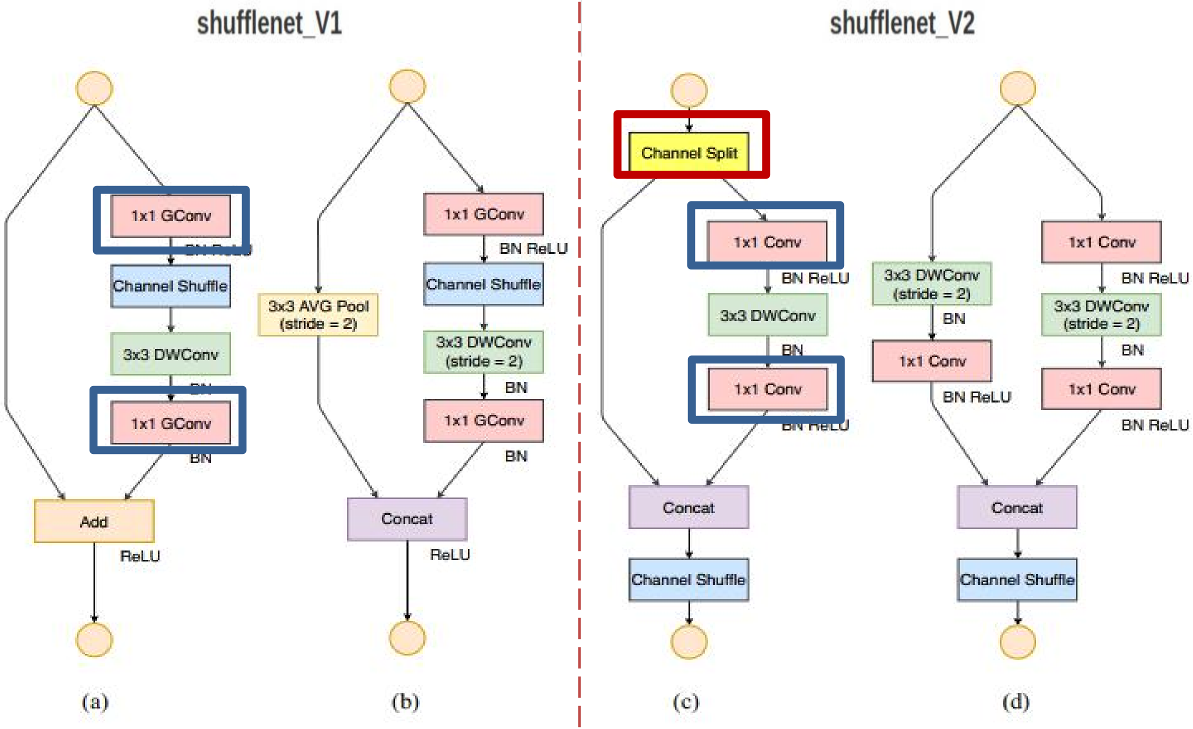
for spatial down sampling (2×)

for spatial down sampling (2×)

ShuffleNetV2

- ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design
- Face++ & 清华大学 *ECCV* 2018

- 1. channel split: 将输入的feature maps分为两部分c’和c-c’ .
- 2. GConv 替换成Conv



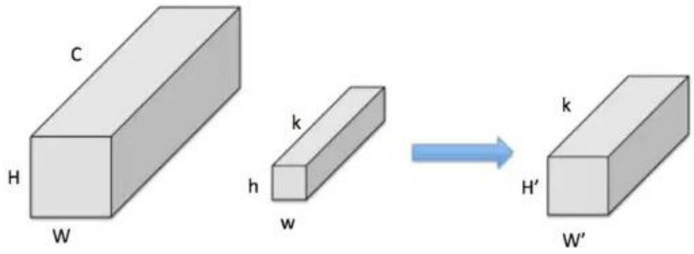
DWConv: depthwise convolution
Gconv: group convolution

for spatial down sampling (2×)

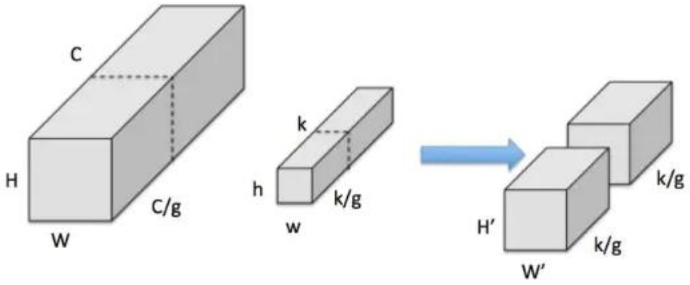
for spatial down sampling (2×)

ShuffleNetV2

- ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design
- Face++ & 清华大学 *ECCV* 2018



convolution

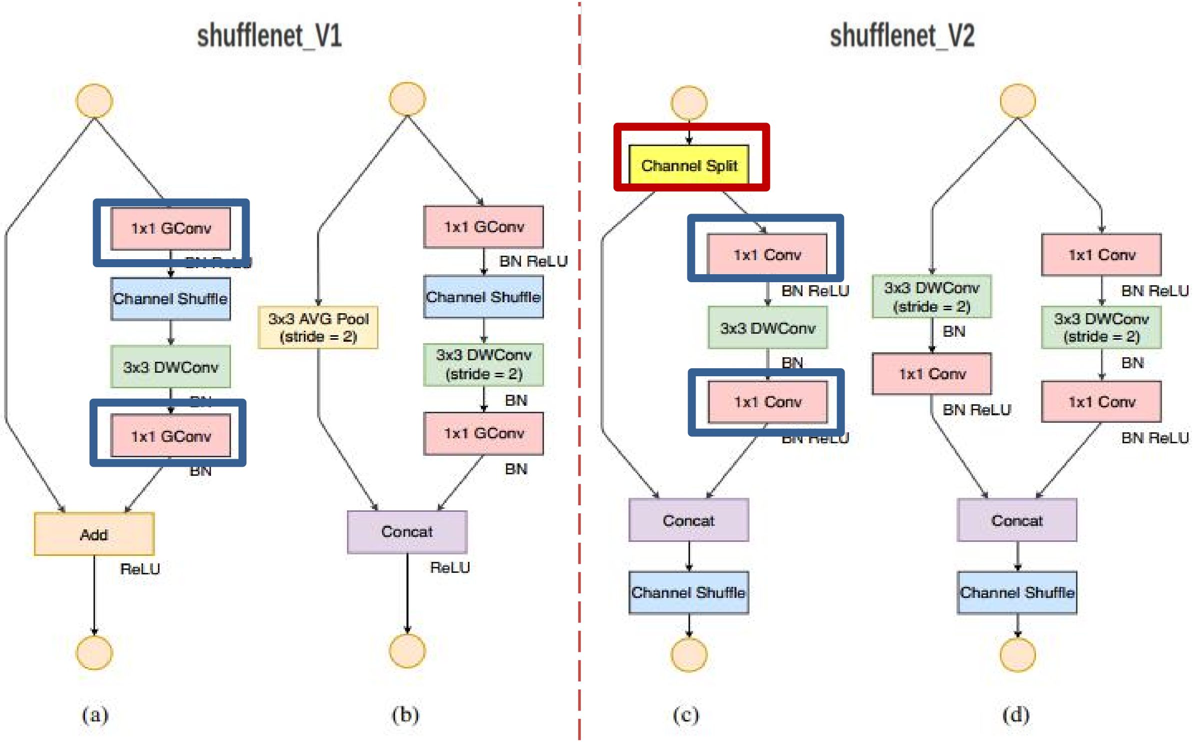


Gconv: group convolution

ShuffleNetV2

- ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design
- Face++ & 清华大学 *ECCV* 2018

- 1. channel split: 将输入的feature maps分为两部分c’和c-c’ .
- 2. GConv 替换成Conv，太多的组卷积会增加内存访问成本



DWConv: depthwise convolution
Gconv: group convolution

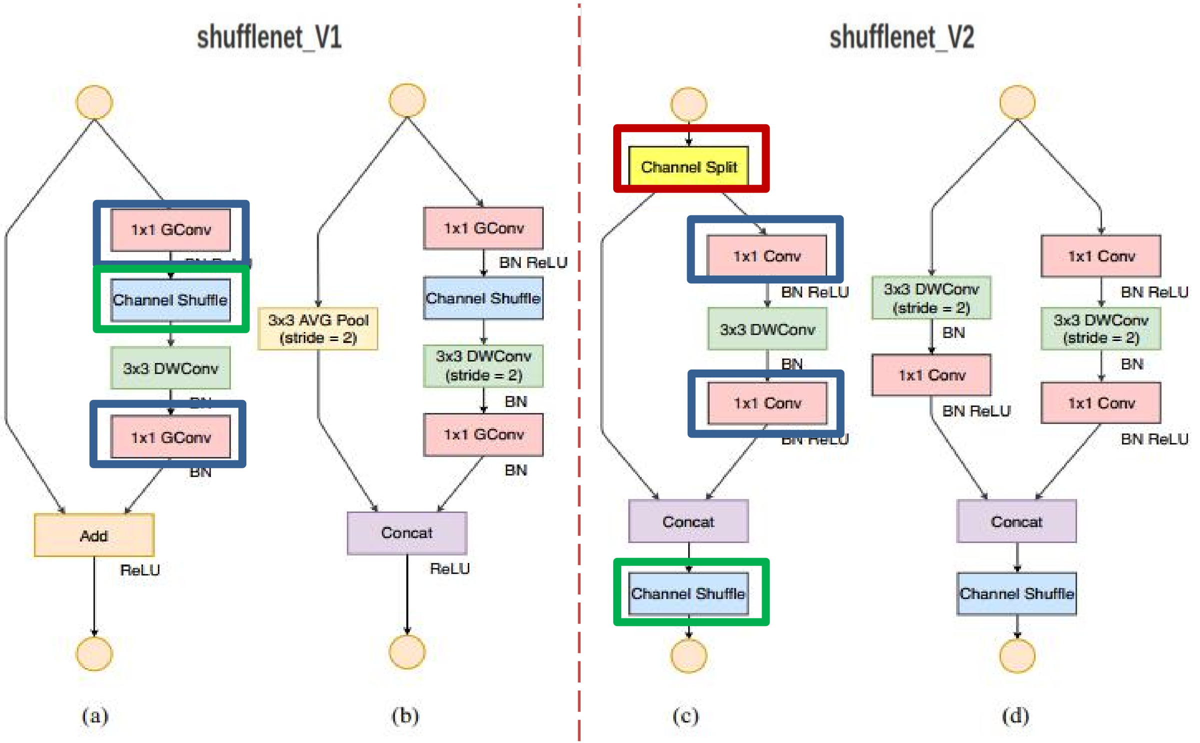
for spatial down sampling (2×)

for spatial down sampling (2×)

ShuffleNetV2

- ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design
- Face++ & 清华大学 *ECCV* 2018

- 1. channel split: 将输入的feature maps分为两部分c’和c-c’ .
- 2. GConv 替换成Conv
- 3. **channel shuffle**



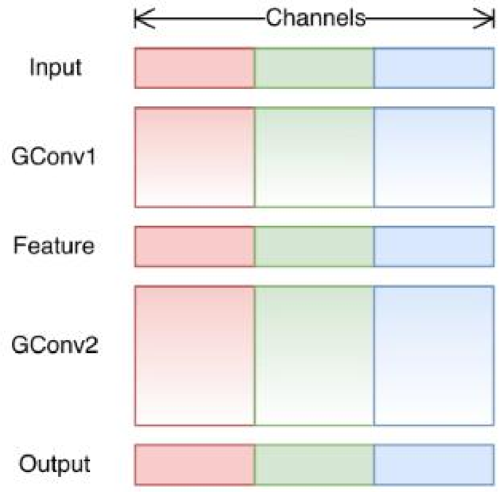
DWConv: depthwise convolution
Gconv: group convolution

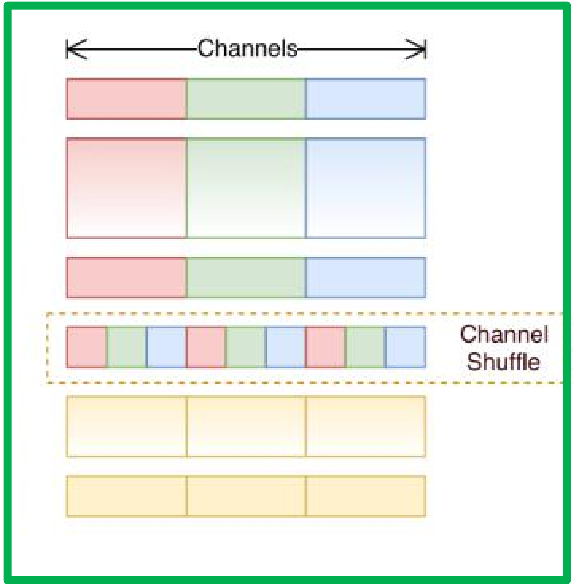
for spatial down sampling (2×)

for spatial down sampling (2×)

ShuffleNetV2

- ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design
- Face++ & 清华大学 *ECCV* 2018

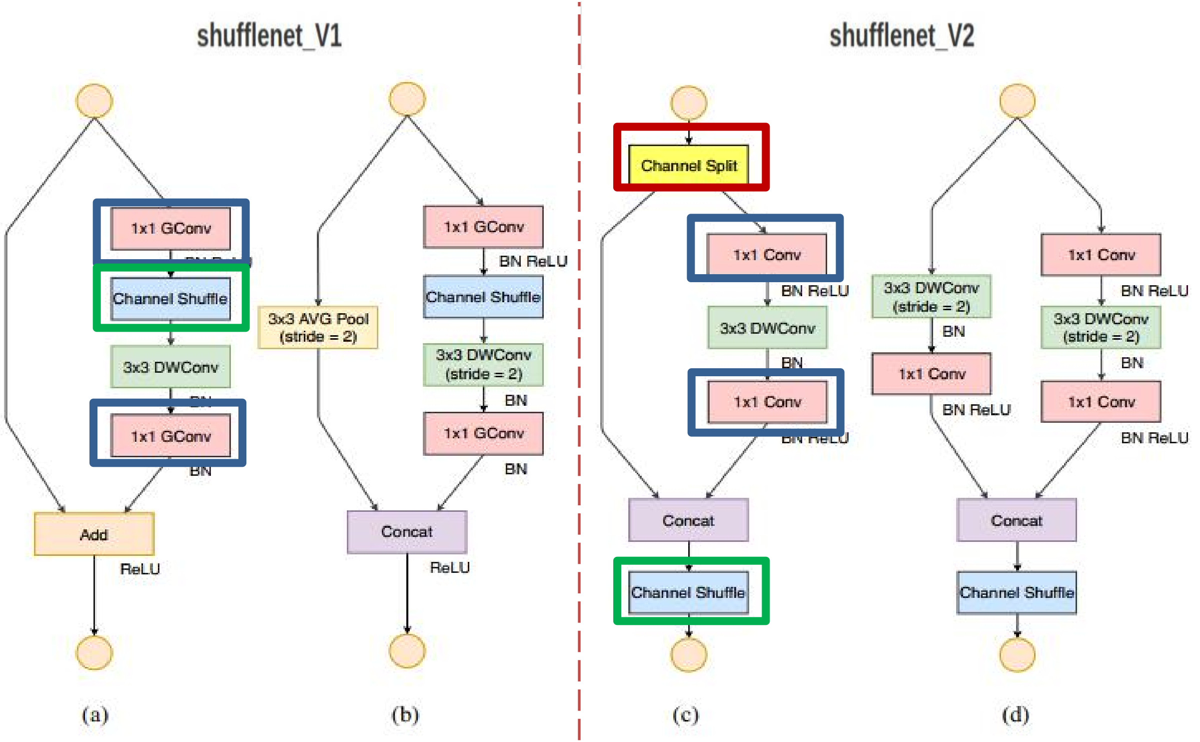




ShuffleNetV2

- ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design
- Face++ & 清华大学 *ECCV* 2018

- 1. channel split: 将输入的feature maps分为两部分c’和c-c’ .
- 2. GConv 替换成Conv
- 3. **channel shuffle**
- channel split已经分开了feature，如果channel shuffle继续使用会丢失另一半feature。



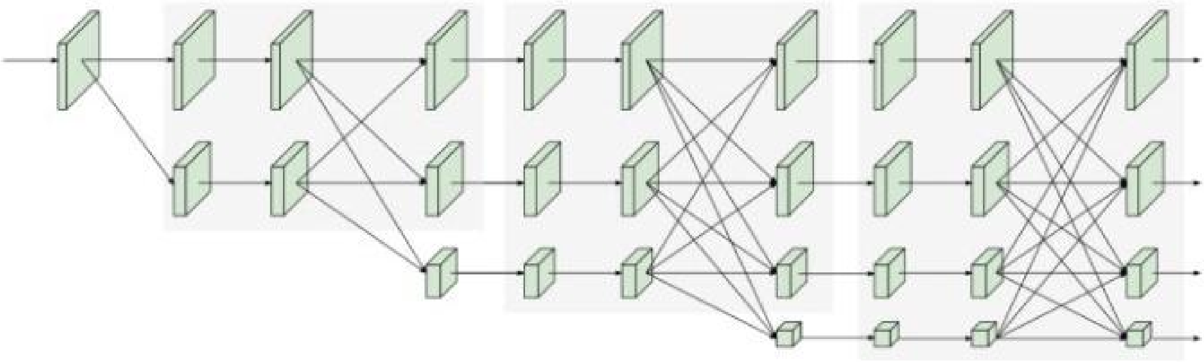
DWConv: depthwise convolution
Gconv: group convolution

for spatial down sampling (2×)

for spatial down sampling (2×)

Lite-HRNet

- **Naive Lite-HRNet：** Shuffle Block和Small HRNet简单融合，能够得到轻量化的HRNet
- **Lite-HRNet：** Naive Lite-HRNet中存在大量的1x1卷积操作，中使用conditional channel weighting模块替代卷积，以进一步提高网络的计算效率。



Naive Lite-HRNet

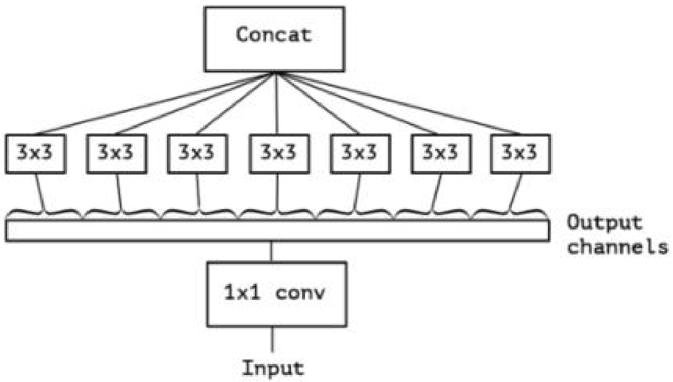
- We adopt the shuffle block to replace the second 3 $\times$ 3 convolution in the stem of Small HRNet， and replace all the normal residual blocks
- The normal convolutions in the multi-resolution fusion are replaced by the separable convolutions （**Xception**）

layer	output size	Small HRNet	Naive Lite-HRNet	resolution branch
image	256 × 256			1×
stem	64 × 64	conv2d	conv2d	2×
		conv2d	shuffle block	4×
$stage_2$	64 × 64	residual block	shuffle block	4× 8×
		fusion block	fusion block	4× 8×
$stage_3$	64 × 64	residual block	shuffle block	4× 8× 16×
		fusion block	fusion block	4× 8× 16×
$stage_4$	64 × 64	residual block	shuffle block	4× 8× 16× 32×
		fusion block	fusion block	4× 8× 16× 32×
FLOPs				
#Params				

Naive Lite-HRNet

- 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 .
- The normal convolutions in the multi-resolution fusion are replaced by the separable convolutions （**Xception**）

Figure 4. An “extreme” version of our Inception module, with one spatial convolution per output channel of the 1x1 convolution.



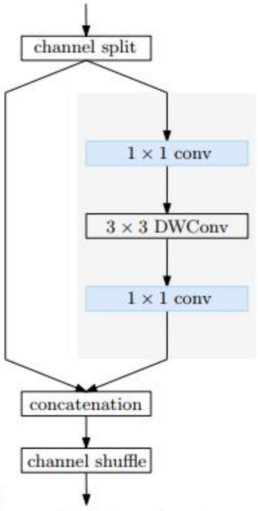
Xception: Deep learning with depthwise separable convolutions cvpr2017

- 1.顺序不同：depthwise separable convolution是先做channel-wise spatial convolution在再做1x1的conv，而Xception是相反的。
- 2. Xception每个操作的后面都跟了ReLU非线性激活，而depthwise separable convolution是没有的。

Lite-HRNet

- **Naive Lite-HRNet：** Shuffle Block和Small HRNet简单融合，能够得到轻量化的HRNet
- **Lite-HRNet：** Naive Lite-HRNet中存在大量的1x1卷积操作，中使用conditional channel weighting模块替代卷积，以进一步提高网络的计算效率。

  - 1x1卷积的时间复杂度: C^2=[1*1]*C*1*C
  - 3x3DW卷积: $9C$=[3*3]*1*1*C
  - 当C>5,Shuffle Block中2个1x1卷积复杂度大于1个3x3DW卷积

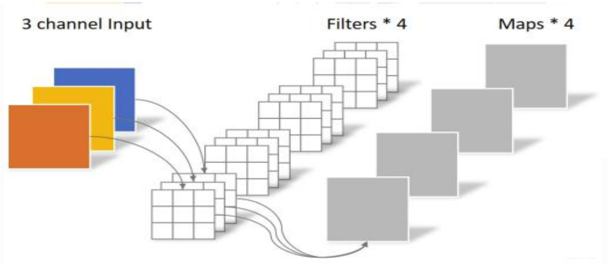


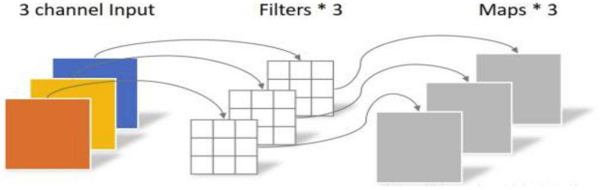
(a)

Lite-HRNet

- **Naive Lite-HRNet：** Shuffle Block和Small HRNet简单融合，能够得到轻量化的HRNet
- **Lite-HRNet：** Naive Lite-HRNet中存在大量的1x1卷积操作，中使用conditional channel weighting模块替代卷积，以进一步提高网络的计算效率。

  - 1x1卷积的时间复杂度: C^2=[1*1]*C*1*C
  - 3x3DW卷积: $9C$=[3*3]*1*1*C
  - 当C>5,Shuffle Block中2个1x1卷积复杂度大于1个3x3DW卷积

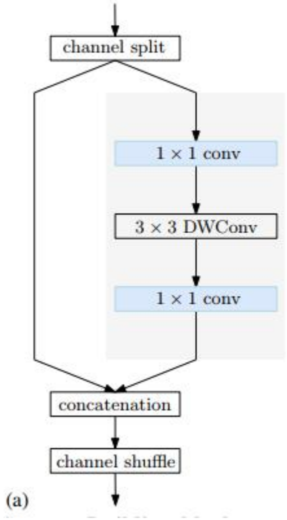




Lite-HRNet

- **Naive Lite-HRNet：** Shuffle Block和Small HRNet简单融合，能够得到轻量化的HRNet
- **Lite-HRNet：** Naive Lite-HRNet中存在大量的1x1卷积操作，中使用conditional channel weighting模块替代卷积，以进一步提高网络的计算效率。

  - 1x1卷积的时间复杂度: C^2=[1*1]*C*1*C
  - 3x3DW卷积: $9C$=[3*3]*1*1*C
  - 当C>5,Shuffle Block中2个1x1卷积复杂度大于1个3x3DW卷积

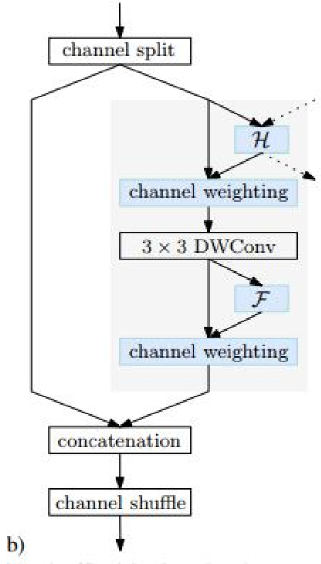


Lite-HRNet

- **Naive Lite-HRNet：** Shuffle Block和Small HRNet简单融合，能够得到轻量化的HRNet
- **Lite-HRNet：** Naive Lite-HRNet中存在大量的1x1卷积操作，中使用conditional channel weighting模块替代卷积，以进一步提高网络的计算效率。

  - 1x1卷积的时间复杂度: C^2=[1*1]*C*1*C
  - 3x3DW卷积: $9C$=[3*3]*1*1*C
  - 当C>5,Shuffle Block中2个1x1卷积复杂度大于1个3x3DW卷积

  - conditional channel weighting
  - CCW时间复杂度： C

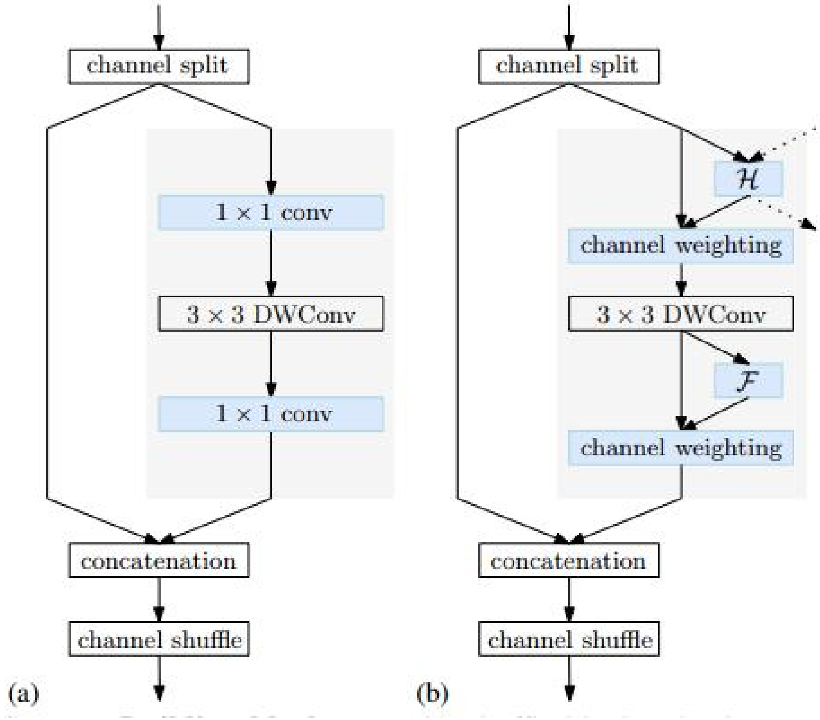


conditional channel weighting

- CCW： $\mathrm{Y}_s = \mathrm{W}_s \odot \mathrm{X}_s,$
- Conv： $\mathrm{Y} = \mathbf{W} \otimes \mathrm{X},$

W_s 是 $W_s \times H_s \times C_s$ 的矩阵，表示weight map； $\odot$ 表示元素乘法操作。

- w权重计算：
- H: Cross-resolution Weight Computation
- F: Spatial Weighting Computation



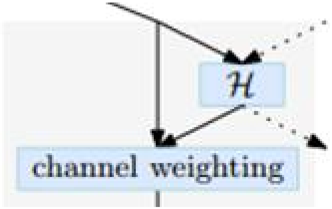
Shuffle Block CCW Block

conditional channel weighting

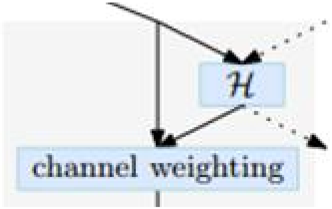
H: Cross-resolution Weight Computation

$$(\mathsf{W}_1, \mathsf{W}_2, \ldots, \mathsf{W}_s) = \mathcal{H}_s(\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_s),$$

- s-th stage has s parallel resolutions and s weight maps $\mathrm{W}_1, \mathrm{W}_2, \ldots, \mathrm{W}_s$



conditional channel weighting



H: Cross-resolution Weight Computation

$$(\mathrm{W}_1, \mathrm{W}_2, \ldots, \mathrm{W}_s) = \mathcal{H}_s(\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_s),$$

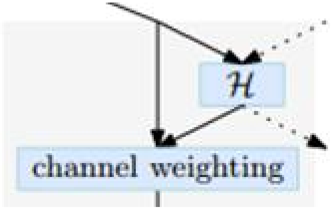
- *s*-th stage has *s* parallel resolutions and *s* weight maps $\mathrm{W}_1, \mathrm{W}_2, \ldots, \mathrm{W}_s$
- $\{\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_{s-1}\}$ —> adaptive average pooling (AAP) —> $\{\mathrm{X}'_1, \mathrm{X}'_2, \ldots, \mathrm{X}'_{s-1}\}$

$$\mathrm{X}'_1 = \mathrm{AAP}(\mathrm{X}_1), \mathrm{X}'_2 = \mathrm{AAP}(\mathrm{X}_2), \quad \ldots, \mathrm{X}'_{s-1} = \mathrm{AAP}(\mathrm{X}_{s-1}),$$

output size： $W_s \times H_s$.

conditional channel weighting

H: Cross-resolution Weight Computation



$$(\mathrm{W}_1, \mathrm{W}_2, \ldots, \mathrm{W}_s) = \mathcal{H}_s(\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_s),$$

- *s*-th stage has *s* parallel resolutions and *s* weight maps $\mathrm{W}_1, \mathrm{W}_2, \ldots, \mathrm{W}_s$

- $\{\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_{s-1}\}$ —> adaptive average pooling (AAP) —> $\{\mathrm{X}'_1, \mathrm{X}'_2, \ldots, \mathrm{X}'_{s-1}\}$

$$\mathrm{X}'_1 = \mathrm{AAP}(\mathrm{X}_1), \mathrm{X}'_2 = \mathrm{AAP}(\mathrm{X}_2), \ \ldots, \mathrm{X}'_{s-1} = \mathrm{AAP}(\mathrm{X}_{s-1}),$$

output size： $W_s \times H_s$.

concate $\{\mathrm{X}'_1, \mathrm{X}'_2, \ldots, \mathrm{X}'_{s-1}\}$ and X_s together,

$$(\mathrm{X}'_1, \mathrm{X}'_2, \ldots, \mathrm{X}_s) \rightarrow \mathrm{Conv.} \rightarrow \mathrm{ReLU} \rightarrow \mathrm{Conv.} \rightarrow \mathrm{sigmoid} \rightarrow (\mathrm{W}'_1, \mathrm{W}'_2, \ldots, \mathrm{W}'_s). \quad (4)$$

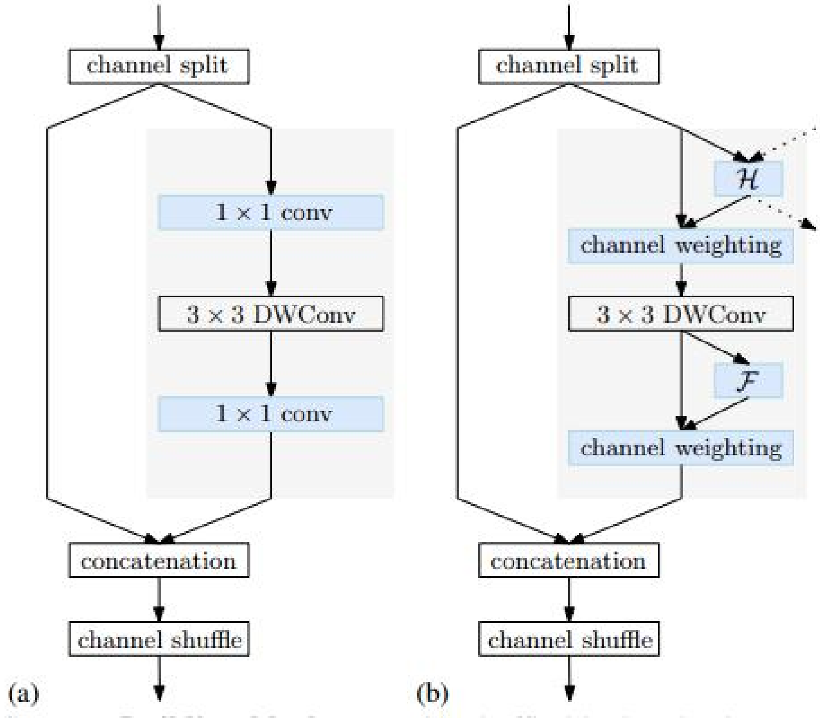
$\rightarrow$ upsampled to the corresponding resolutions,

conditional channel weighting

F: Spatial Weighting Computation

$$\mathbf{w}_s = \mathcal{F}_s(\mathsf{X}_s).$$

- GAP(global average pooling) + FC + ReLU + FC + sigmoid
- Gathering the spatial information from all the positions



Shuffle Block CCW Block

Conditional Channel Weighting

- 使用CCW代替卷积以减少网络的计算需求

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

Lite-HRNet结构

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 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
Large networks										
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 [49]	320 × 256	28.1M	26.7	72.3	89.2	79.1	68.0	78.6	–
SimpleBaseline [46]	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 [55]	HRNetV1-W48	384 × 288	63.6M	32.9	76.2	92.5	83.6	72.5	82.4	81.1
Small networks										
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

Table 5. **Comparisons on the MPII `val` 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

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*
Hand-crafted networks						
ICNet [59]	Y	–	28.3	1024×2048	67.7	69.5
BiSeNetV1 A [53]	Y	5.8M	14.8	768×1536	69.0	68.4
BiSeNetV1 B [53]	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 [62]	Y	–	36.9	1024×2048	–	74.8
BiSeNetV2 Small [50]	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	–
NAS-based networks						
CAS [58]	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

Thanks！