深蓝学院
shenlanxueyuan.com

模型压缩

主讲人 王晓波

中国科学院自动化研究所生物安全中心实验室博士
京东第二届博士技术管培生

人脸识别-模型压缩

1、小网络的设计

2、低秩分解和二值化

3、知识蒸馏

4、整体总结

人脸识别-模型压缩

1、小网络的设计

2、低秩分解和二值化

3、知识蒸馏

4、整体总结

1、小网络设计

- **MegaFace竞赛**

Algorithm	Details	Set1	Data Size
Sogou AIGROUP	multi model + combined margin loss	99.939%	Large
SRC-Beijing	Sphereface+large training set	99.888%	Large
ST-PureFace	attention-56+A-Softmax	99.801%	Large
EI Networks	resnet-150 + resnet-100	99.414%	Large
ICARE_FACE_V1	ResNet101	99.319%	Large

几乎都是大模型，大训练数据

https://megaface.cs.washington.edu/results/facescrub.html

1、小网络设计

- **LFR竞赛**

layer name	124-layer	output size
Input Image Crop		$112\times112\times3$
	3×3, 64, stride 1	$112\times112\times64$
Conv2_x	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$56\times56\times64$
Conv3_x	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times13$	$28\times28\times128$
Conv4_x	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times40$	$14\times14\times256$
Conv5_x	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times5$	$7\times7\times512$
FC		$1\times1\times512$

- ☐ **29.70 Gflops**
- ☐ **297 MB**

Lightweight Face Recognition Challenge

1、小网络设计

- **LFR竞赛**

Lightweight Face Recognition Challenge		
DeepGlint-Large	No.1	SEResNet-152
	No.2	AttentionIRSE-156
	No.3	ResNet-100
IQIYI-Large	No.1	xxx
	No.2	DenseNet-290
	No.3	ResNetSE-152

Lightweight Face Recognition Challenge

1、小网络设计

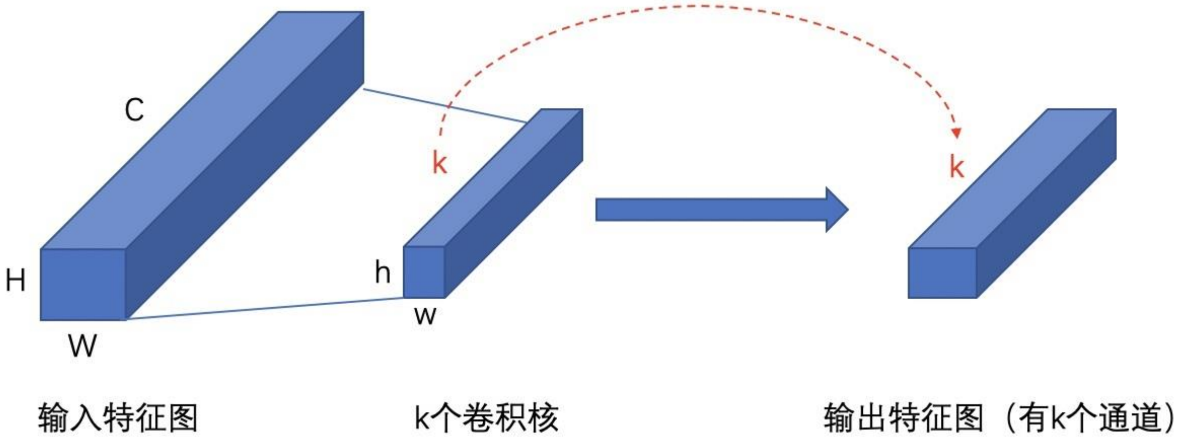
- **MobileNet**

目标就是获取到小网络

MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

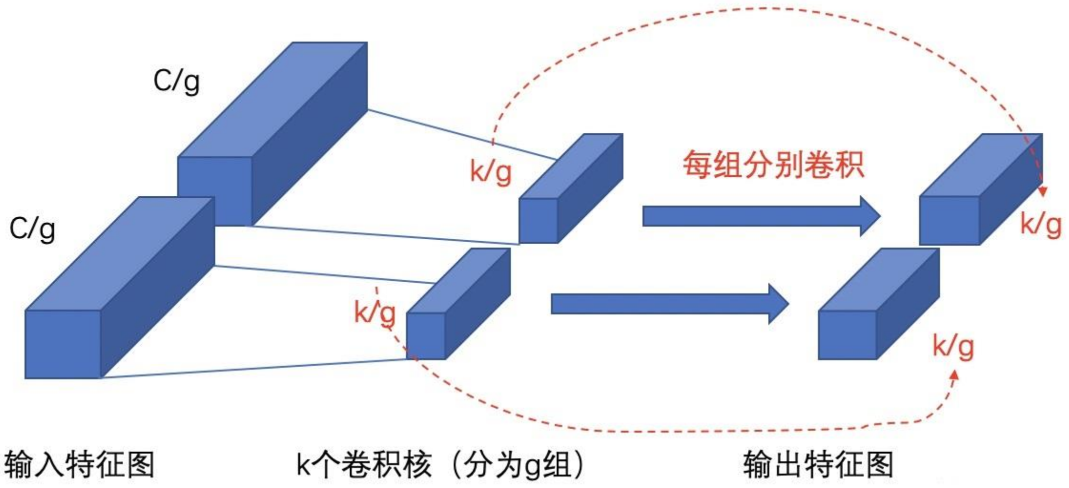
- **GroupConv**



MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

- **GroupConv**

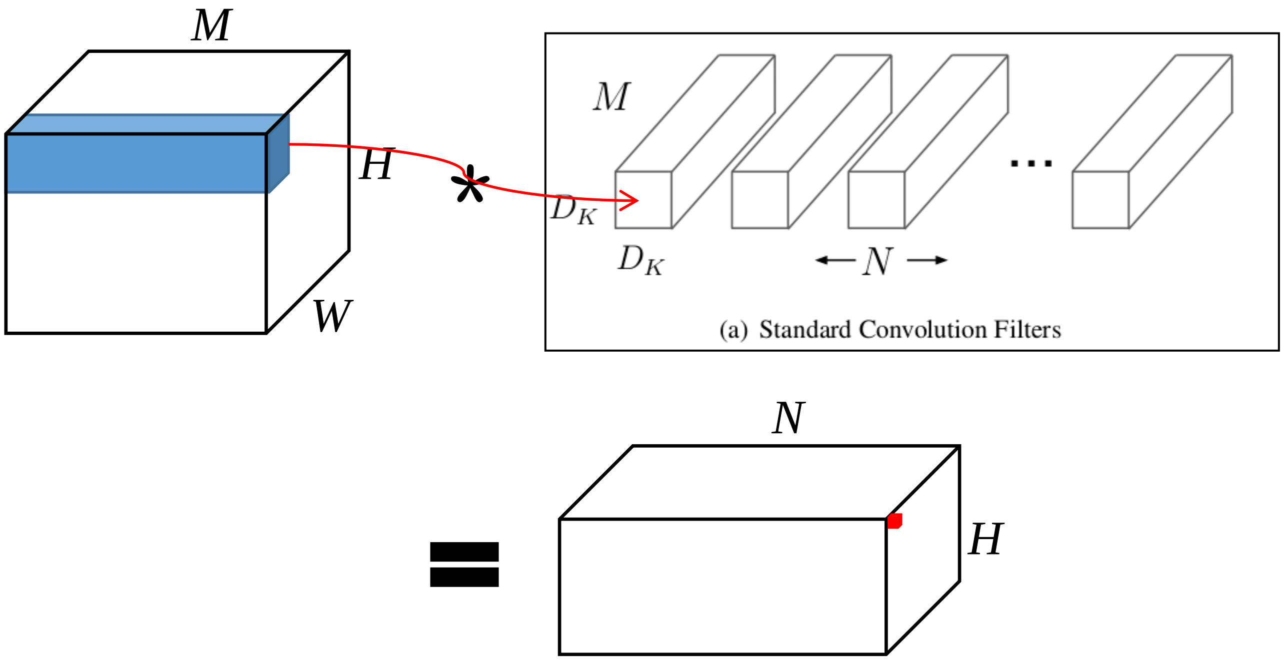



```
layer {
  name: "conv2"
  ......
  convolution_param {
    num_output: 256
    pad: 2
    kernel_size: 5
    group: 2
  }
}
```

MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

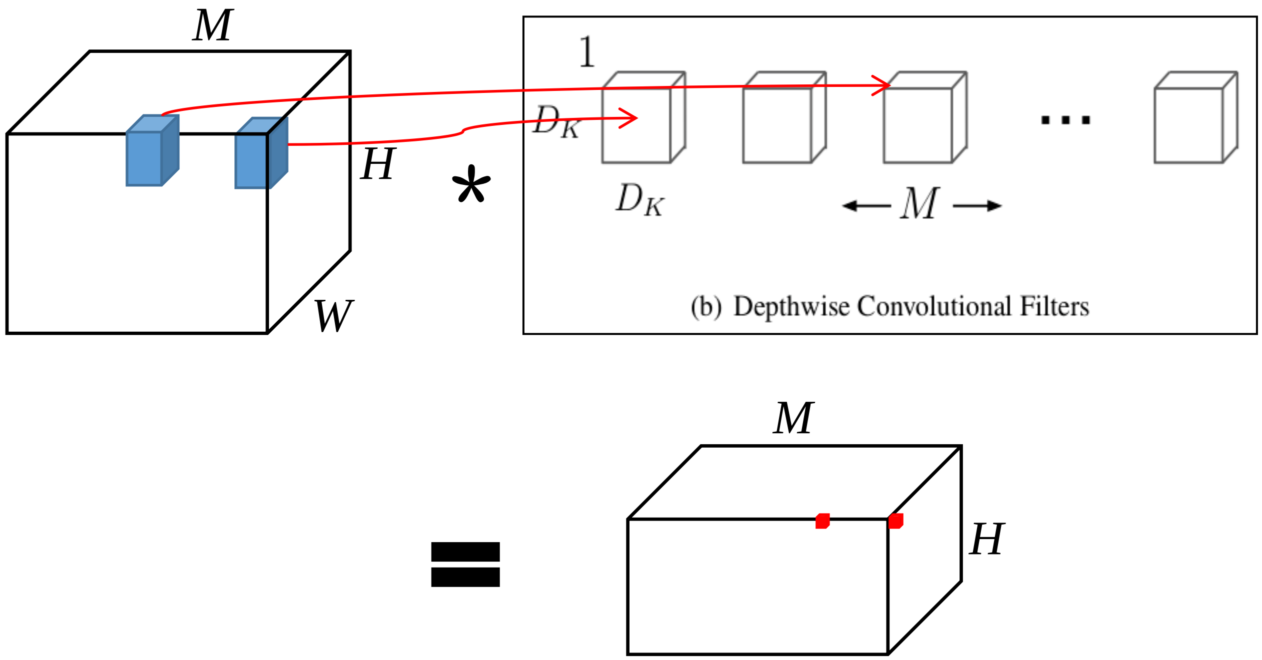
- **MobileNet**



MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

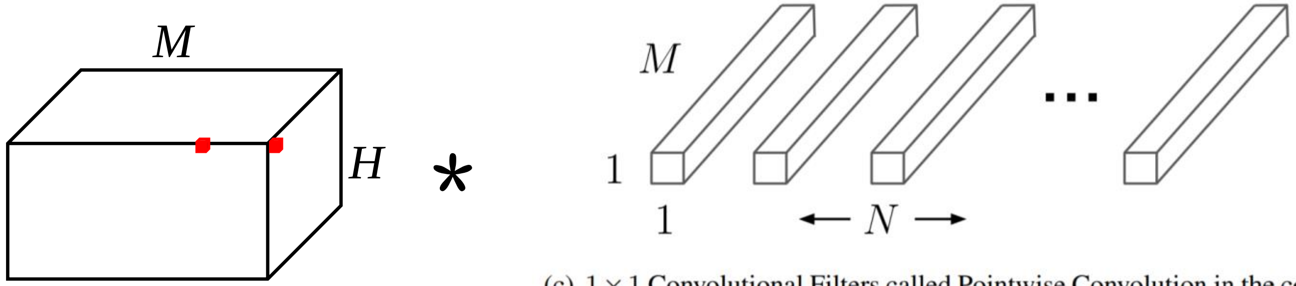
- **MobileNet**



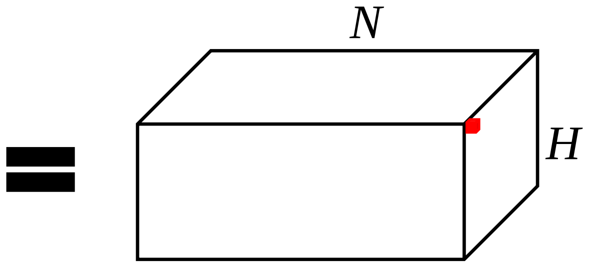
MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

- **MobileNet**



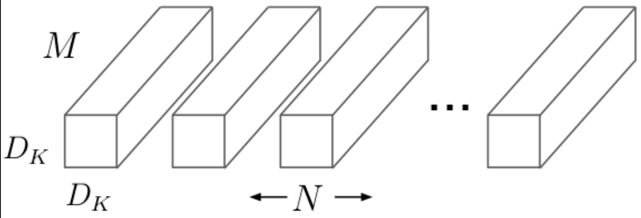
(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution



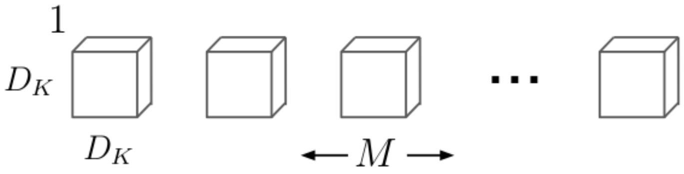
MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

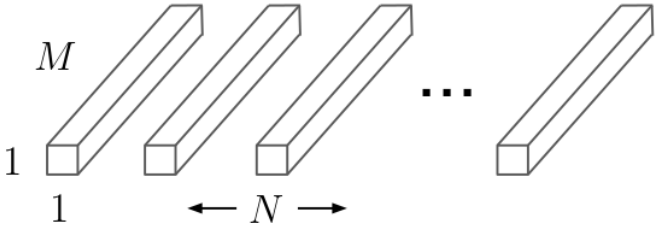
- **MobileNet**



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters

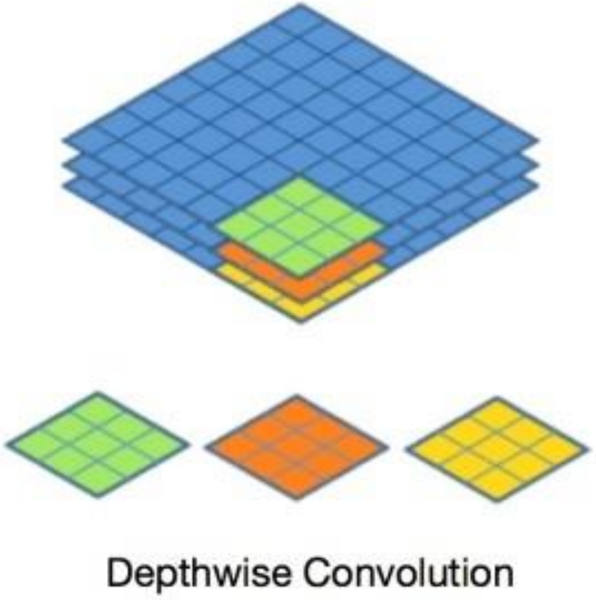


(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

[MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications]

1、小网络设计

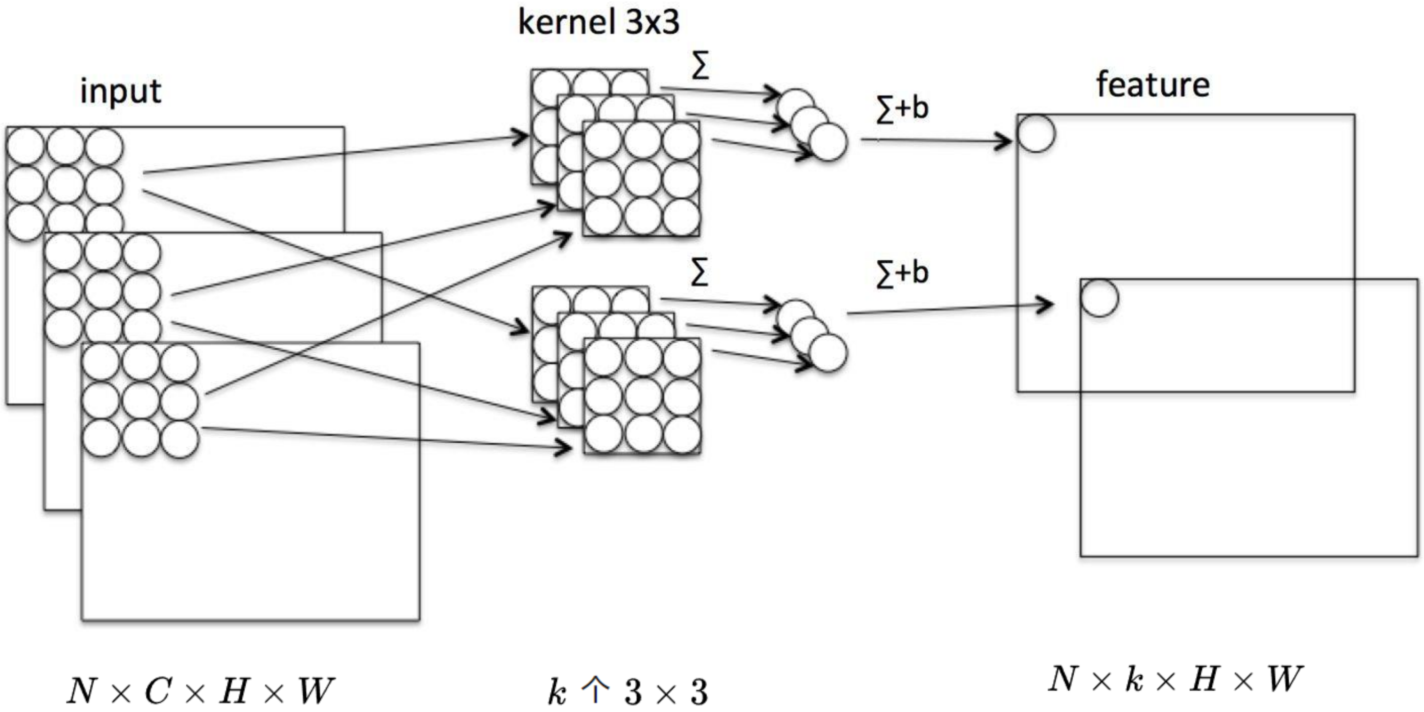
- **MobileNet**



MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

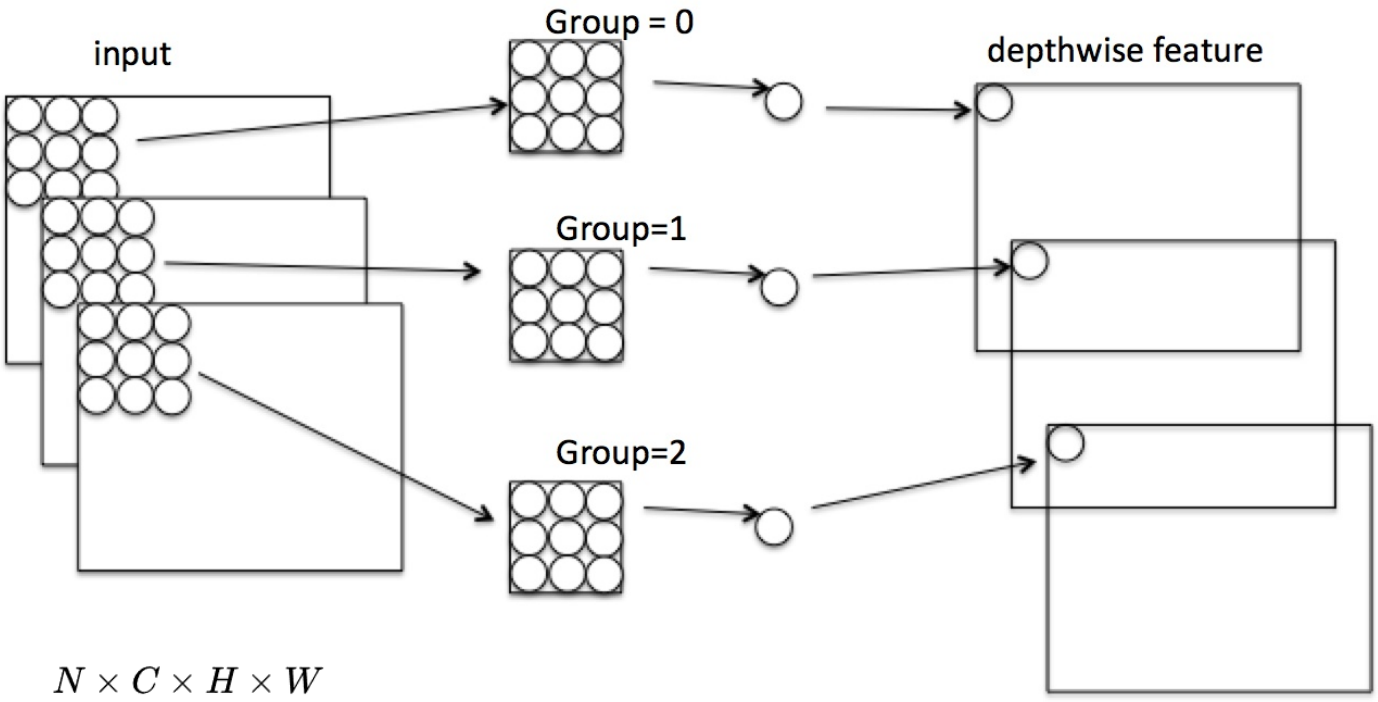
- **MobileNet**



MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

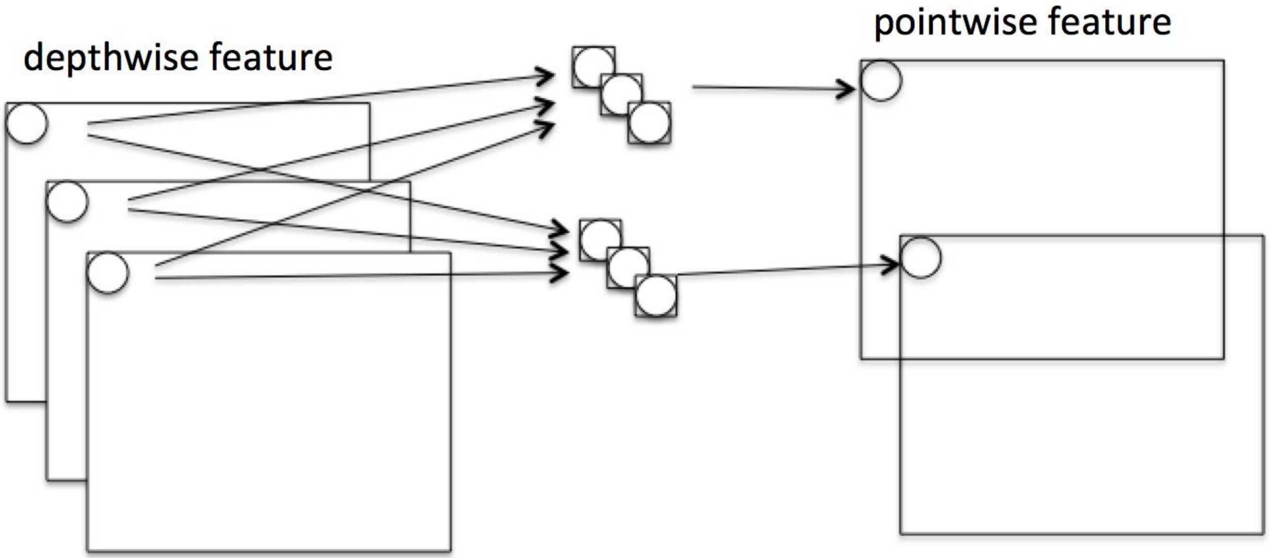
- **MobileNet**



MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

- **MobileNet**



$$N \times C \times H \times W$$

MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

- **MobileNet**

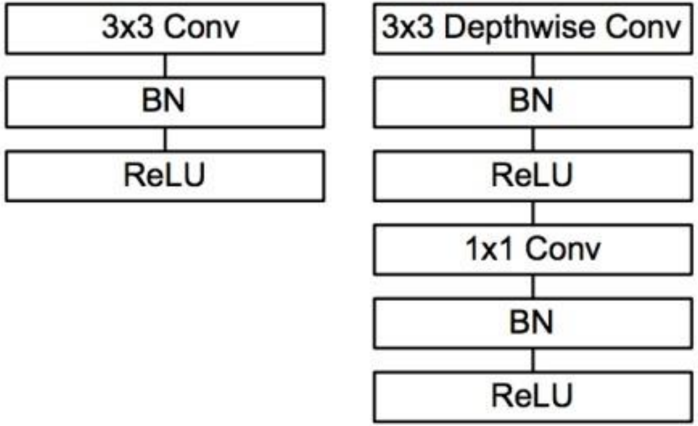


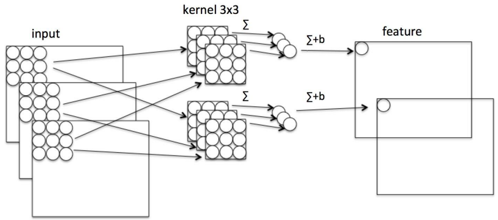
Figure 3. Left: Standard convolutional layer with batchnorm and ReLU. Right: Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batchnorm and ReLU.

- Depthwise通道间信息隔离
- Pointwise通道间信息交换
- .....

MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

- **MobileNet**

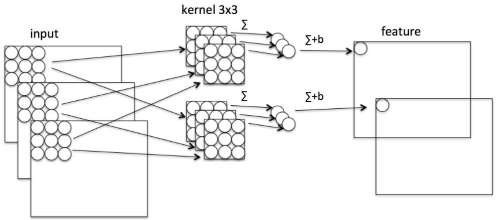


普通卷积计算量 $H \times W \times C \times k \times 3 \times 3$

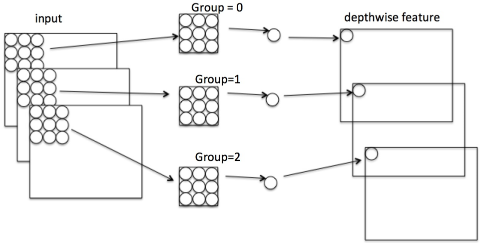
MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

- **MobileNet**



普通卷积计算量 $H \times W \times C \times k \times 3 \times 3$

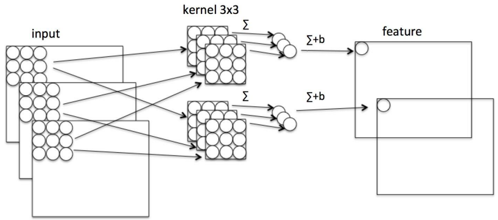


Depthwise计算量 $H \times W \times C \times 3 \times 3$

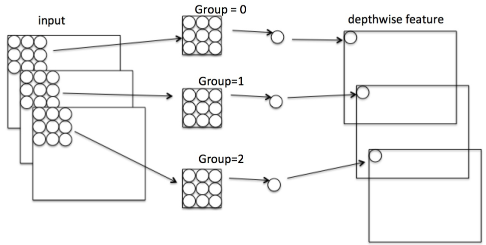
MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

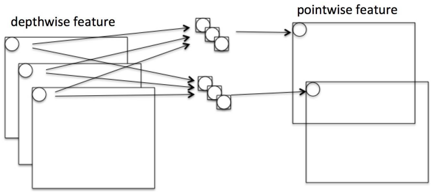
- **MobileNet**



普通卷积计算量 $H \times W \times C \times k \times 3 \times 3$



Depthwise计算量 $H \times W \times C \times 3 \times 3$

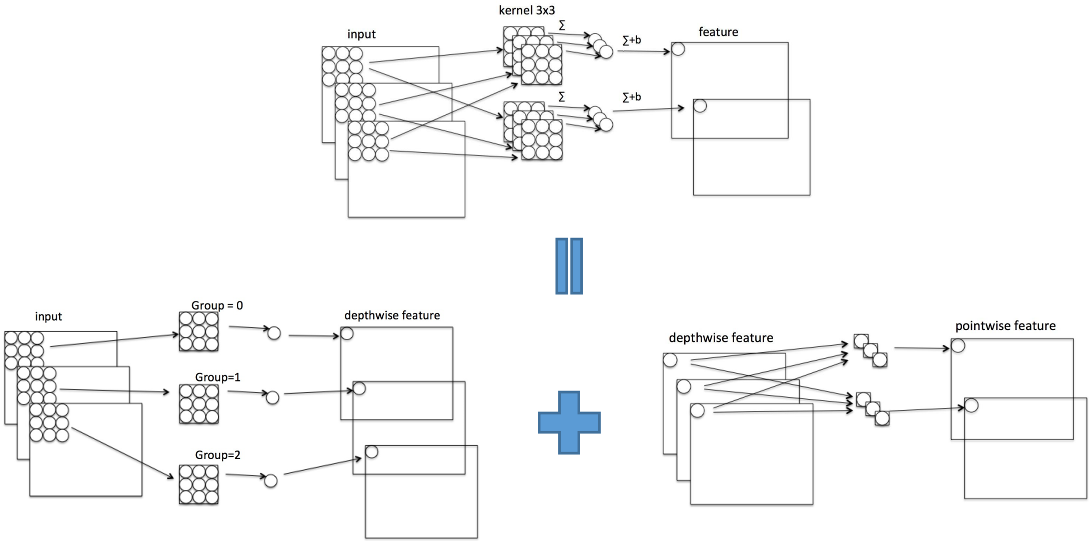


Pointwise计算量 $H \times W \times C \times k$

[MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications]

1、小网络设计

- **MobileNet**



$$\frac{depthwise + pointwise}{conv} = \frac{H \times W \times C \times 3 \times 3 + H \times W \times C \times k}{H \times W \times C \times k \times 3 \times 3} = \frac{1}{k} + \frac{1}{3 \times 3}$$

MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

- **MobileNet**

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1 Conv / s1	$3 \times 3 \times 512$ dw $1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$ $14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Width Multiplier	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
0.75 MobileNet-224	68.4%	325	2.6
0.5 MobileNet-224	63.7%	149	1.3
0.25 MobileNet-224	50.6%	41	0.5

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1、小网络设计

- **MobileFaceNet**

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1 Conv / s1	$3 \times 3 \times 512$ dw $1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$ $14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Global Depthwise Convolution

- MobileV2, V3
- ShuffleNet etc

MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devicess

人脸识别-模型压缩

1、小网络的设计

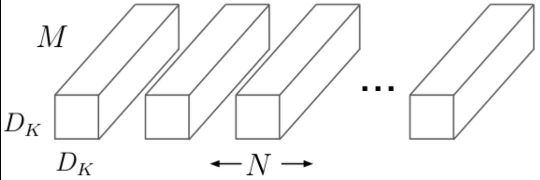
2、低秩分解和二值化

3、知识蒸馏

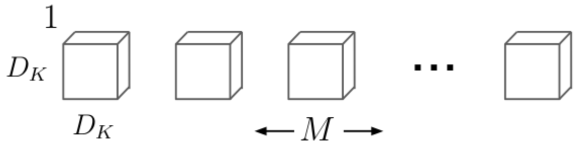
4、整体总结

2、低秩分解

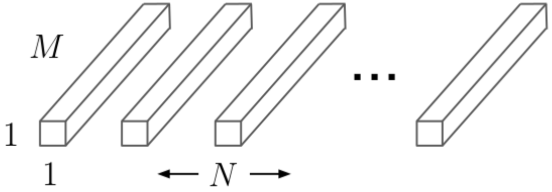
- **MobileNet**



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters

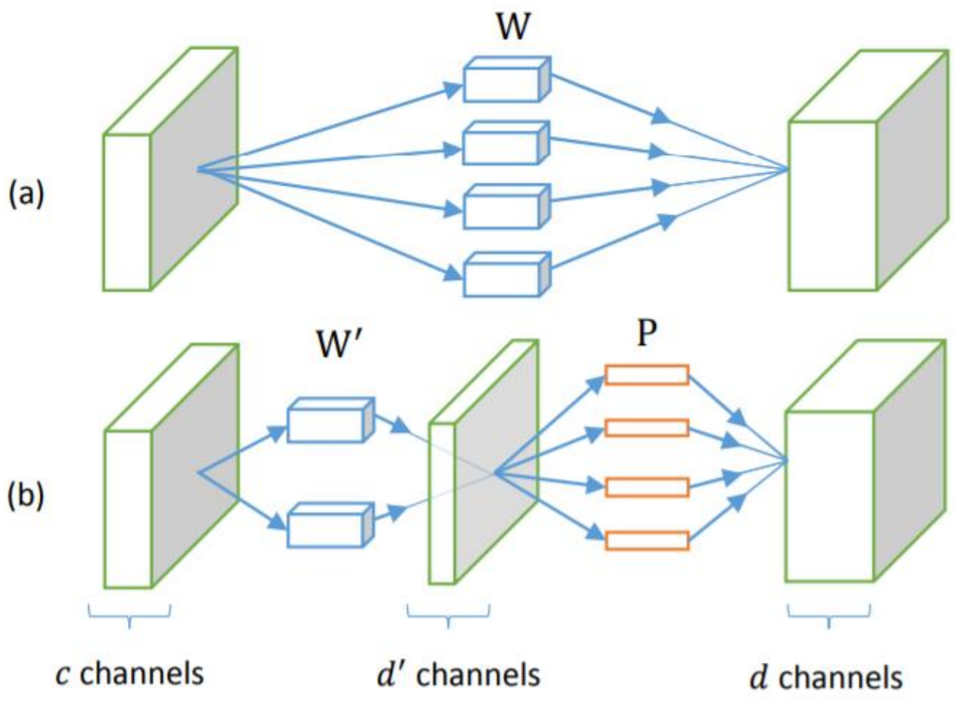


(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

2、低秩分解

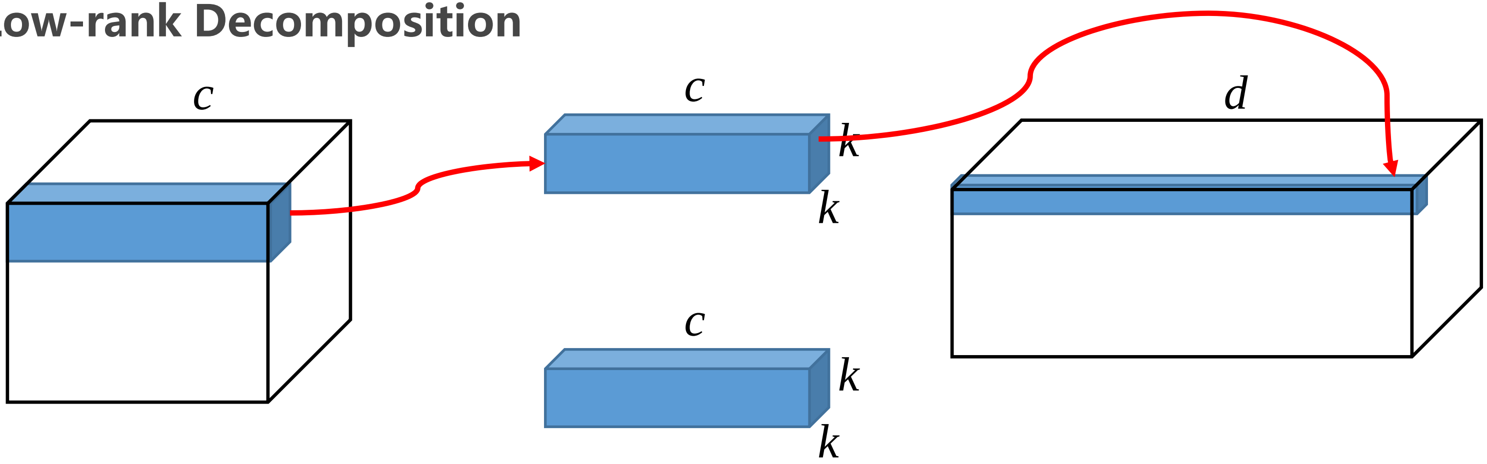
- **Low-rank Decomposition**



Accelerating Very Deep Convolutional Networks for Classification and Detection

2、低秩分解

- **Low-rank Decomposition**



$\boldsymbol{x} \in \mathbb{R}^{k^2c+1}$ $\boldsymbol{W} \in \mathbb{R}^{d\times(k^2c+1)}$ $\boldsymbol{y} \in \mathbb{R}^{d}$

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$$

Accelerating Very Deep Convolutional Networks for Classification and Detection

2、低秩分解

- **Low-rank Decomposition**

$$\boldsymbol{y} = \boldsymbol{M}(\boldsymbol{y} - \overline{\boldsymbol{y}}) + \overline{\boldsymbol{y}}, \quad \boldsymbol{M} \in \mathbb{R}^{d \times d}$$

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}, \qquad \boldsymbol{b} = \overline{\boldsymbol{y}} - \boldsymbol{M}\overline{\boldsymbol{y}}$$

$$\boldsymbol{M} = \boldsymbol{P}\boldsymbol{Q}^{T}, \quad \boldsymbol{W}' = \boldsymbol{Q}^{T}\boldsymbol{W}$$

$$\boldsymbol{y} = \boldsymbol{P}\boldsymbol{W}'\boldsymbol{x} + \boldsymbol{b}$$

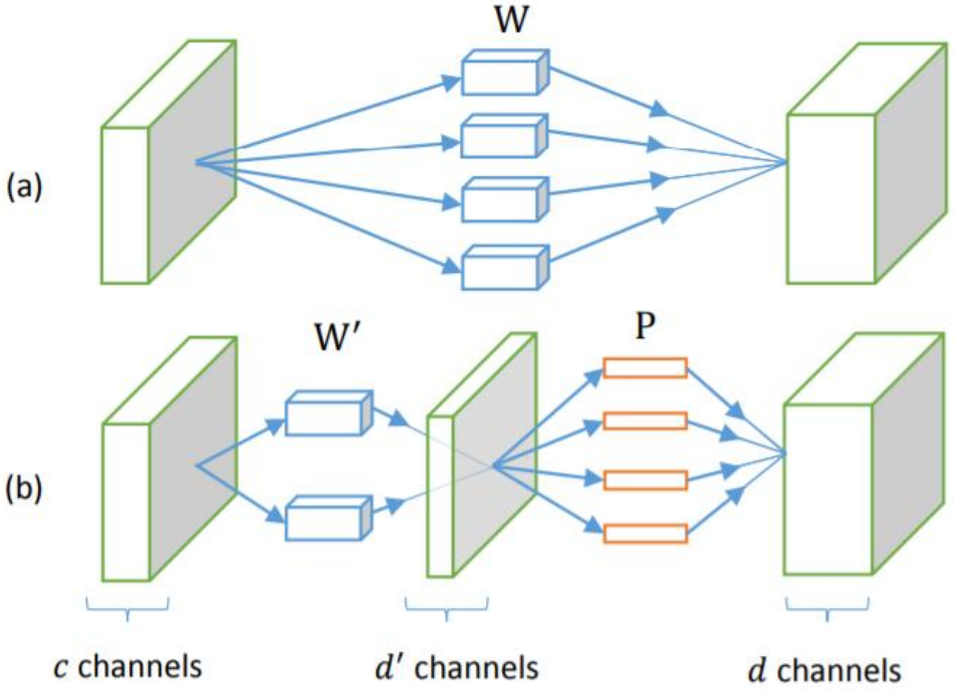
Accelerating Very Deep Convolutional Networks for Classification and Detection

2、低秩分解

- **Low-rank Decomposition**

$$y = Wx$$

$$y = PW'x + b$$



Accelerating Very Deep Convolutional Networks for Classification and Detection

2、低秩分解

- **Low-rank Decomposition**

$$\min_{\mathrm{M}} \sum_i \|(\mathbf{y}_i - \bar{\mathbf{y}}) - \mathrm{M}(\mathbf{y}_i - \bar{\mathbf{y}})\|_2^2, \qquad s.t. \quad rank(\mathrm{M}) \le d'.$$

$$\boldsymbol{y} = \boldsymbol{M}(\boldsymbol{y} - \overline{\boldsymbol{y}}) + \overline{\boldsymbol{y}}, \quad \boldsymbol{M} \in \mathbb{R}^{d \times d}$$

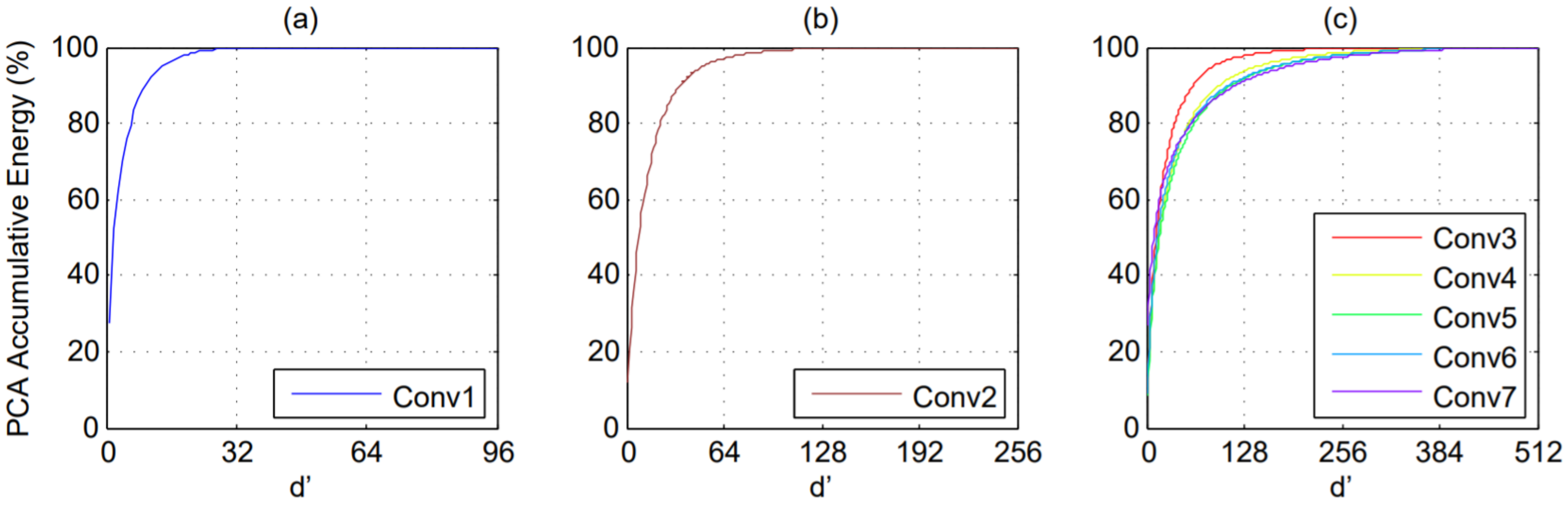
$\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}$进行特征值分解， $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} = \boldsymbol{USU^T}$,其中$\boldsymbol{U}$是正交矩阵，$\boldsymbol{S}$是对角线矩阵。$\boldsymbol{M} = \boldsymbol{U_{d'}U_{d'}^T}$，其中$\boldsymbol{U_{d'}}$就是$\boldsymbol{U}$的前$\boldsymbol{d'}$个特征向量。可设定$\boldsymbol{P} = \boldsymbol{Q} = \boldsymbol{U_{d'}}$

$$\boldsymbol{W'} = \boldsymbol{Q^T W} \qquad \boldsymbol{y} = \boldsymbol{PW'x} + \boldsymbol{b}$$

Accelerating Very Deep Convolutional Networks for Classification and Detection

2、低秩分解

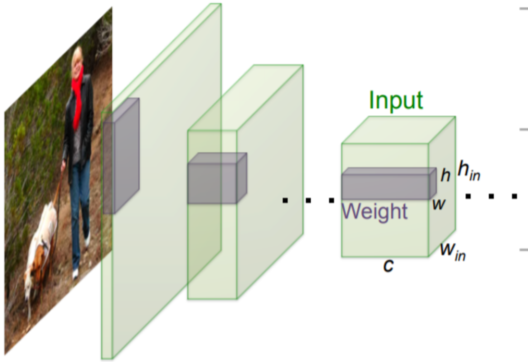
- **Low-rank Decomposition**



Accelerating Very Deep Convolutional Networks for Classification and Detection

2、二值化

- **XNOR-Net**

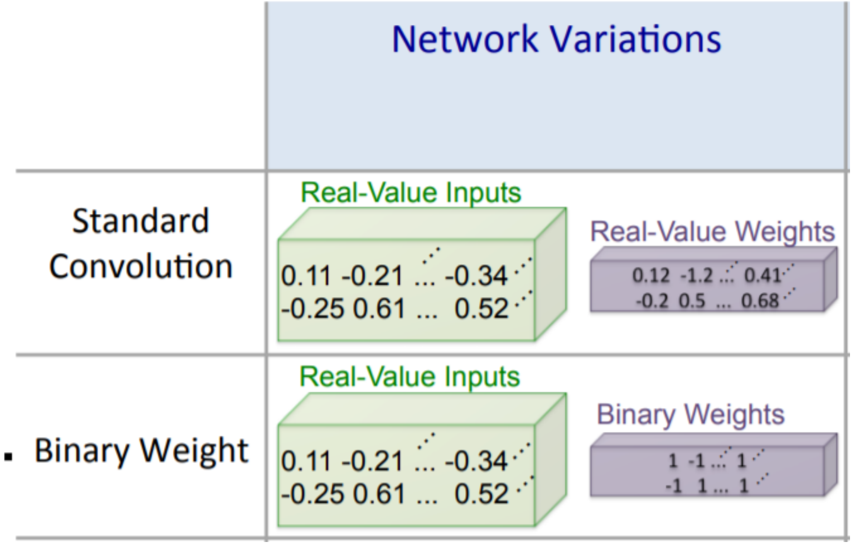


	Network Variations	Operations used in Convolution	Memory Saving (Inference)	Computation Saving (Inference)	Accuracy on ImageNet (AlexNet)
Standard Convolution	Real-Value Inputs 0.11 -0.21 ... -0.34 -0.25 0.61 ... 0.52 Real-Value Weights 0.12 -1.2 ... 0.41 -0.2 0.5 ... 0.68	+ , − , ×	1x	1x	%56.7
Binary Weight	Real-Value Inputs 0.11 -0.21 ... -0.34 -0.25 0.61 ... 0.52 Binary Weights 1 -1 ... 1 -1 1 ... 1	+ , −	~32x	~2x	%56.8
BinaryWeight Binary Input (**XNOR-Net**)	Binary Inputs 1 -1 ... -1 -1 1 ... 1 Binary Weights 1 -1 ... 1 -1 1 ... 1	XNOR , bitcount	~32x	~58x	%44.2

[XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks]

2、二值化

- **XNOR-Net**



real-value weight filter $\mathbf{W} \in \mathcal{W}$

binary filter $\mathbf{B} \in \{+1, -1\}^{c \times w \times h}$

$$\mathbf{W} \approx \alpha \mathbf{B}$$

a scaling factor $\alpha \in \mathbb{R}^+$

XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

2、二值化

- **XNOR-Net**

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

$$\mathbf{I} * \mathbf{W} \approx (\mathbf{I} \oplus \mathbf{B})\,\alpha$$

$\oplus$ indicates a convolution without any multiplication

XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

2、二值化

- **XNOR-Net**

$$J(\mathbf{B},\alpha) = \|\mathbf{W} - \alpha\mathbf{B}\|^2$$

$$\alpha^*, \mathbf{B}^* = \underset{\alpha,\mathbf{B}}{\text{argmin}} J(\mathbf{B},\alpha)$$

$$J(\mathbf{B},\alpha) = \alpha^2\mathbf{B}^\intercal\mathbf{B} - 2\alpha\mathbf{W}^\intercal\mathbf{B} + \mathbf{W}^\intercal\mathbf{W}$$

since $\mathbf{B} \in \{+1,-1\}^n$, $\mathbf{B}^\intercal\mathbf{B} = n$ is a constant $n = c \times w \times h$

$$\mathbf{c} = \mathbf{W}^\intercal\mathbf{W}$$

$$\mathbf{B}^* = \underset{\mathbf{B}}{\text{argmax}}\{\mathbf{W}^\intercal\mathbf{B}\} \quad s.t. \;\; \mathbf{B} \in \{+1,-1\}^n$$

XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

2、二值化

- **XNOR-Net**

$$J(\mathbf{B},\alpha) = \alpha^2\mathbf{B}^\intercal\mathbf{B} - 2\alpha\mathbf{W}^\intercal\mathbf{B} + \mathbf{W}^\intercal\mathbf{W}$$

$$\Downarrow$$

$$\mathbf{B}^* = \underset{\mathbf{B}}{\operatorname{argmax}}\{\mathbf{W}^\intercal\mathbf{B}\} \quad s.t.\ \ \mathbf{B} \in \{+1,-1\}^n$$

$$\Downarrow$$

最优的解：当$\boldsymbol{W}_i \geq 0$时，$\boldsymbol{B}_i = 1$；当$\boldsymbol{W}_i < 0$时，$\boldsymbol{B}_i = -1$

$$\boldsymbol{B}^* = sign(\boldsymbol{W})$$

XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

2、二值化

- **XNOR-Net**

$$J(\mathbf{B},\alpha) = \alpha^2\mathbf{B}^\intercal\mathbf{B} - 2\alpha\mathbf{W}^\intercal\mathbf{B} + \mathbf{W}^\intercal\mathbf{W}$$

$$J(B,\alpha) = \alpha^2 n - 2\alpha \boldsymbol{W}^T\boldsymbol{B} + \boldsymbol{c}$$

$$\alpha = \frac{\boldsymbol{W}^T\boldsymbol{B}^*}{n} = \frac{\boldsymbol{W}^T sign(\boldsymbol{W})}{n} = \frac{\sum |\boldsymbol{W}_i|}{n} = \frac{1}{n}\|\boldsymbol{W}\|_1$$

XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

2、二值化

- **XNOR-Net**

$$\mathbf{X}^{\intercal}\mathbf{W} \approx \beta\mathbf{H}^{\intercal}\alpha\mathbf{B}, \text{ where } \mathbf{H}, \mathbf{B} \in \{+1, -1\}^n \text{ and } \beta, \alpha \in \mathbb{R}^+$$

$$\alpha^*, \mathbf{B}^*, \beta^*, \mathbf{H}* = \underset{\alpha, \mathbf{B}, \beta, \mathbf{H}}{\operatorname{argmin}} \|\mathbf{X} \odot \mathbf{W} - \beta\alpha\mathbf{H} \odot \mathbf{B}\|$$

$$\Downarrow$$

$$\gamma^*, \mathbf{C}^* = \underset{\gamma, \mathbf{C}}{\operatorname{argmin}} \|\mathbf{Y} - \gamma\mathbf{C}\|$$

$$\mathbf{Y}_i = \mathbf{X}_i\mathbf{W}_i$$
$$\mathbf{C}_i = \mathbf{H}_i\mathbf{B}_i$$
$$\gamma = \beta\alpha$$

XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

2、二值化

- **XNOR-Net**

$$\gamma^*, \mathbf{C}^* = \underset{\gamma, \mathbf{C}}{\operatorname{argmin}} \|\mathbf{Y} - \gamma\mathbf{C}\|$$

$$\mathbf{C}^* = \operatorname{sign}(\mathbf{Y}) = \operatorname{sign}(\mathbf{X}) \odot \operatorname{sign}(\mathbf{W}) = \mathbf{H}^* \odot \mathbf{B}^*$$

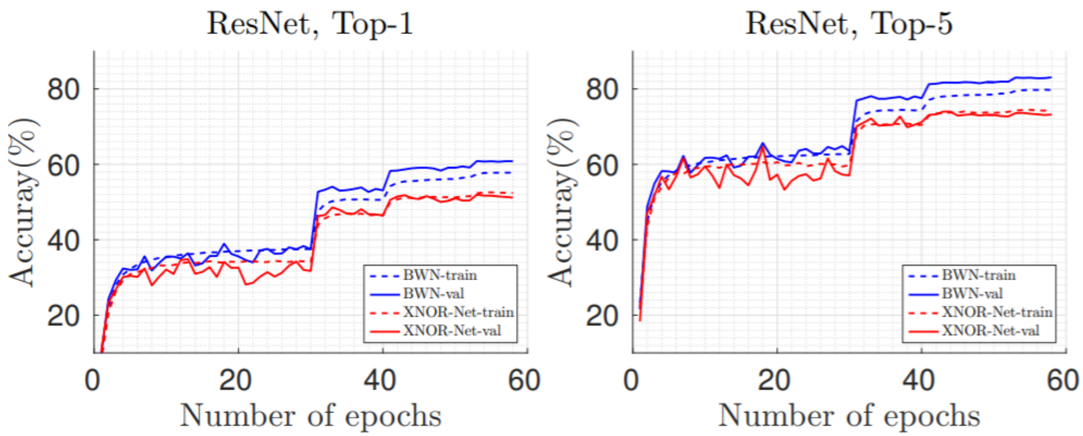
$$\gamma^* = \frac{\sum |\mathbf{Y}_i|}{n} = \frac{\sum |\mathbf{X}_i||\mathbf{W}_i|}{n} \approx \left(\frac{1}{n}\|\mathbf{X}\|_{\ell 1}\right)\left(\frac{1}{n}\|\mathbf{W}\|_{\ell 1}\right) = \beta^* \alpha^*$$

XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

2、二值化

- **XNOR-Net**

Classification Accuracy(%)									
Binary-Weight				Binary-Input-Binary-Weight				Full-Precision	
BWN		BC[11]		XNOR-Net		BNN[11]		AlexNet[1]	
Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
56.8	**79.4**	35.4	61.0	**44.2**	**69.2**	27.9	50.42	56.6	80.2



XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

人脸识别-模型压缩

1、小网络的设计

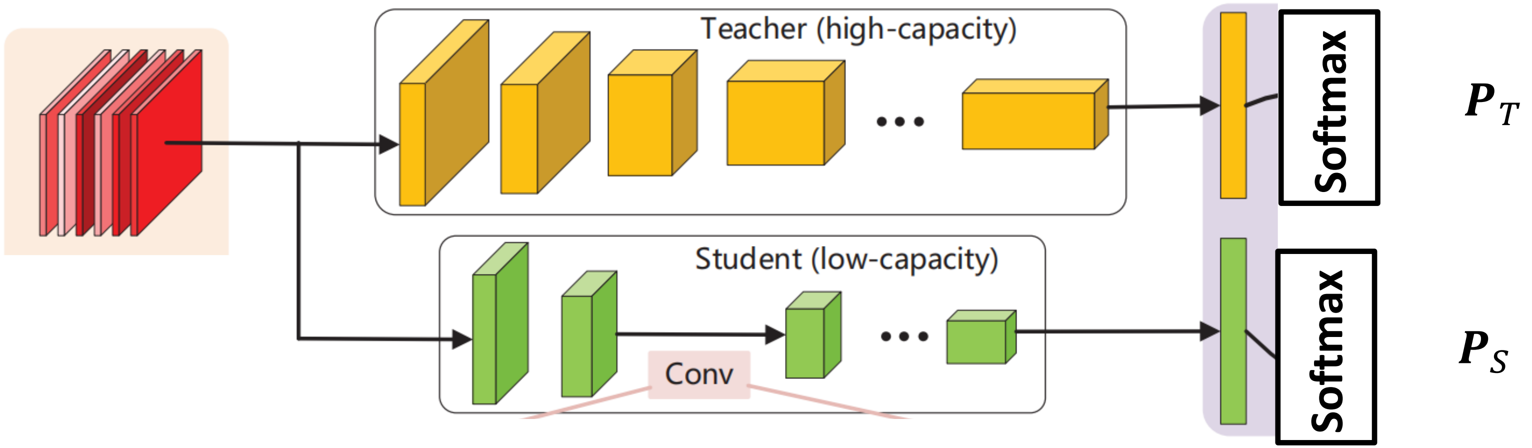
2、低秩分解和二值化

3、知识蒸馏

4、整体总结

3、知识蒸馏

- **Knowledge Distillation**

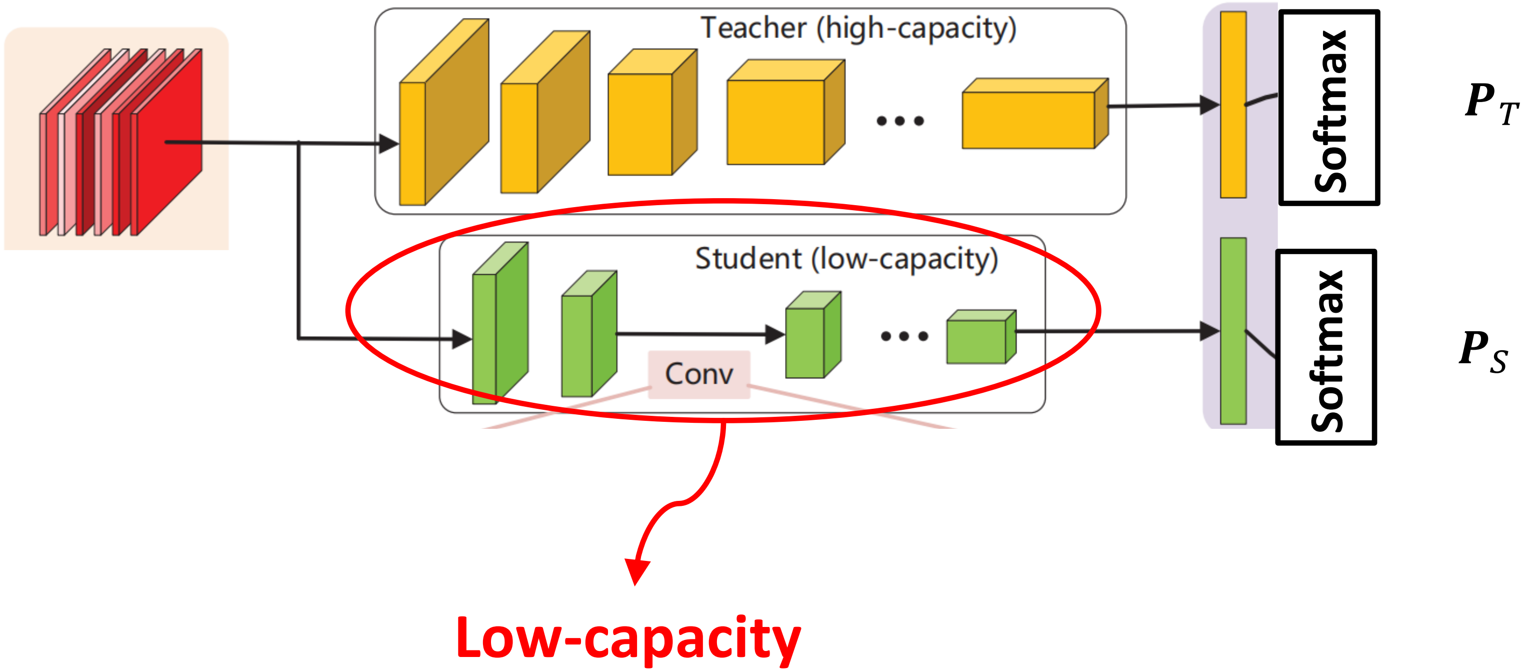


$$\mathcal{L}_{\mathrm{PC}} := \mathcal{L}(P_{\mathrm{T}}^{\tau}, P_{\mathrm{S}}^{\tau}) = \mathcal{L}((z_{\mathrm{T}}/\tau), (z_{\mathrm{S}}/\tau))$$

Distilling the Knowledge in a Neural Network

3、知识蒸馏

- EC-KD



Exclusivity-Consistency Regularized Knowledge Distillation for Face Recognition

3、知识蒸馏

- **EC-KD**

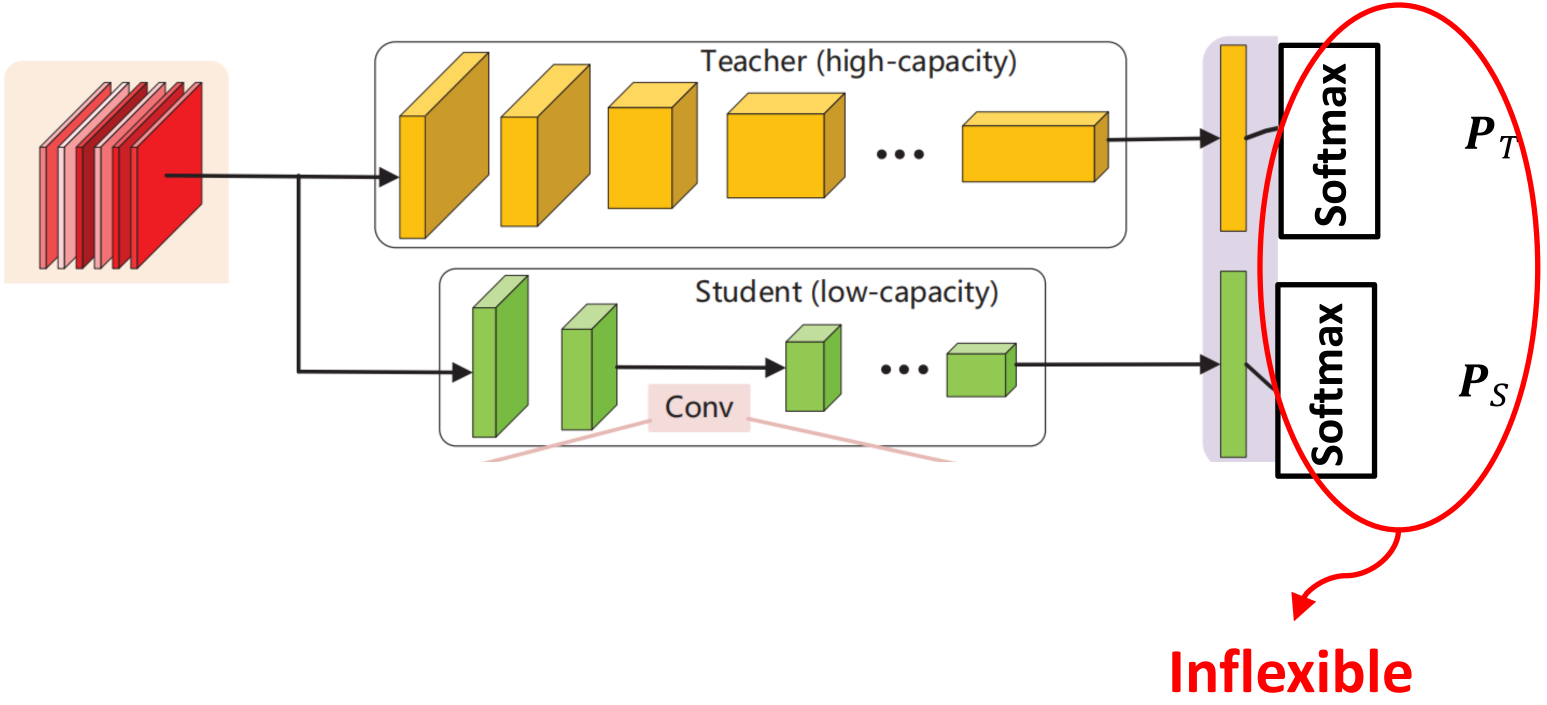
#Filters	Model Size	Flops	Infer Time	LFW	MF-Id.	MF-Veri.
(2×)128	16MB	1.44G	158ms	99.34	87.19	90.82
(Orig.)64	4.8MB	0.38G	84ms	99.11	83.96	87.57
(1/2)32	1.7MB	0.11G	49ms	98.55	74.32	78.71
(1/4)16	648KB	0.03G	34ms	97.60	52.60	58.69
(1/8)8	304KB	0.01G	28ms	94.29	25.32	27.04

Student 每一层的kernel数目越少，模型越小，性能则一般越低

Exclusivity-Consistency Regularized Knowledge Distillation for Face Recognition

3、知识蒸馏

- **EC-KD**



Exclusivity-Consistency Regularized Knowledge Distillation for Face Recognition

3、知识蒸馏

- **EC-KD**

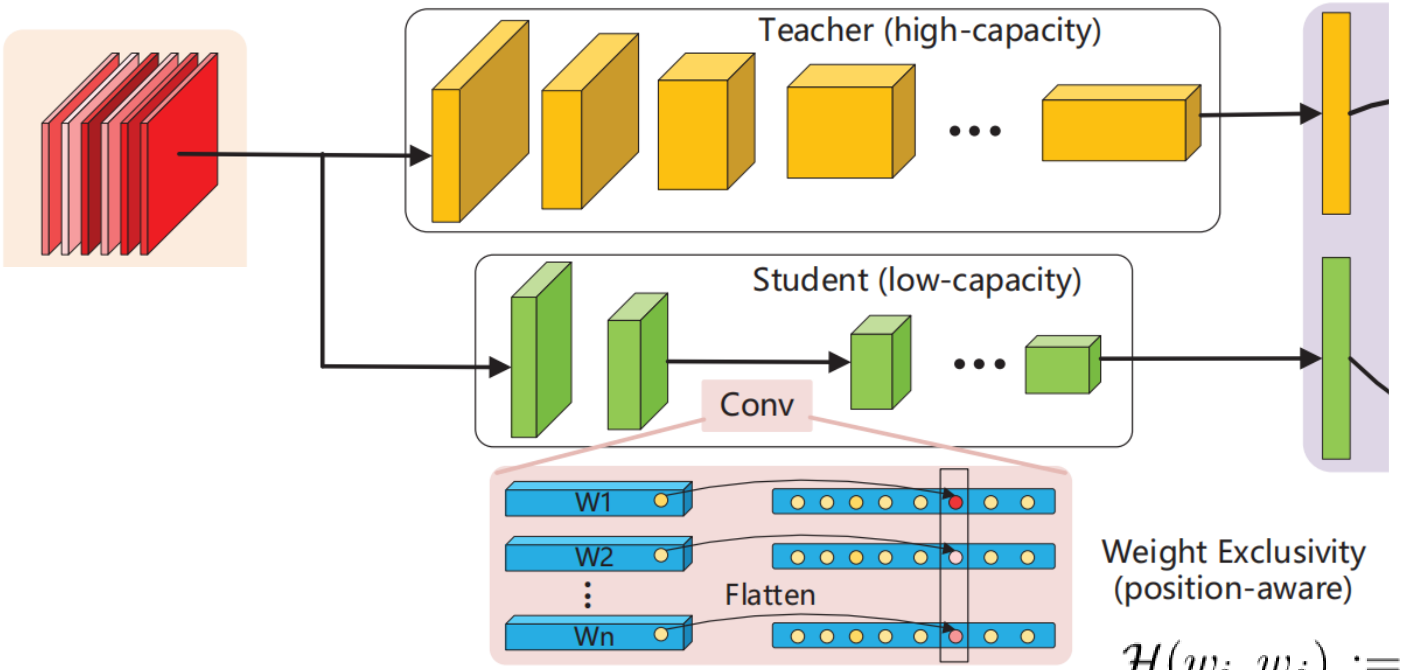
$$\mathcal{L}_{\mathrm{PC}} := \mathcal{L}(P_{\mathrm{T}}^{\tau}, P_{\mathrm{S}}^{\tau}) = \mathcal{L}((z_{\mathrm{T}}/\tau), (z_{\mathrm{S}}/\tau))$$

- Teacher的训练类别和Student的训练类别不一致，或者Teacher模型由Contrastive Loss等训练时，上述公式的Softmax没法计算。
- Student的训练数据包含噪声标签时，性能没有办法保证。
- Student训练类别数较多时，训练时间长，收敛速度比较慢。

Exclusivity-Consistency Regularized Knowledge Distillation for Face Recognition

3、知识蒸馏

- **EC-KD**



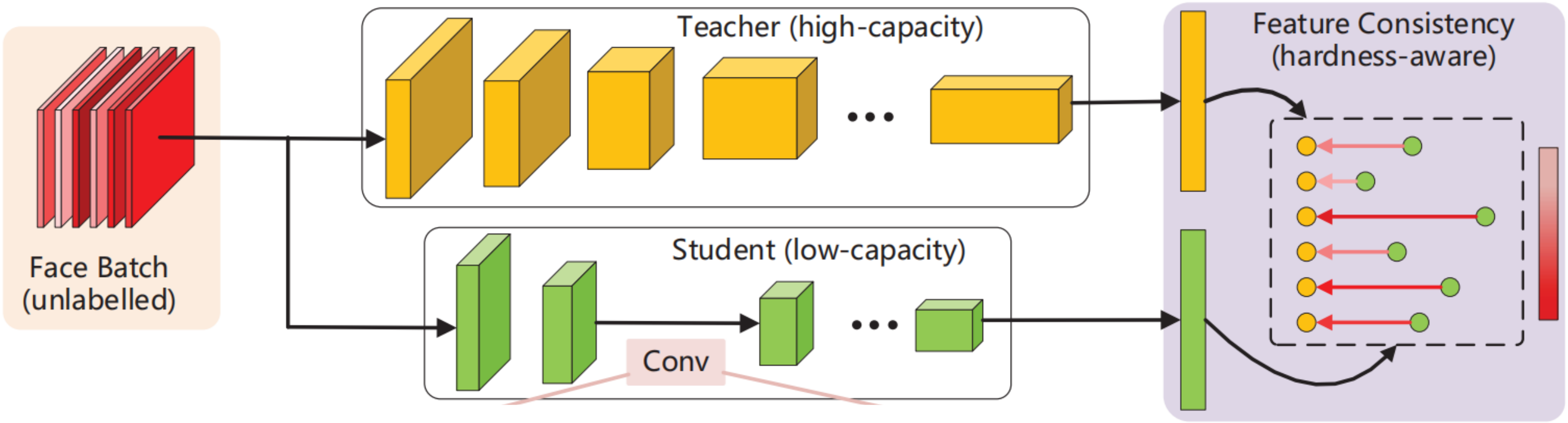
Weight Exclusivity (position-aware)

$$\mathcal{H}(w_i, w_j) := \|w_i \odot w_j\|_0$$

Exclusivity-Consistency Regularized Knowledge Distillation for Face Recognition

3、知识蒸馏

- **EC-KD**

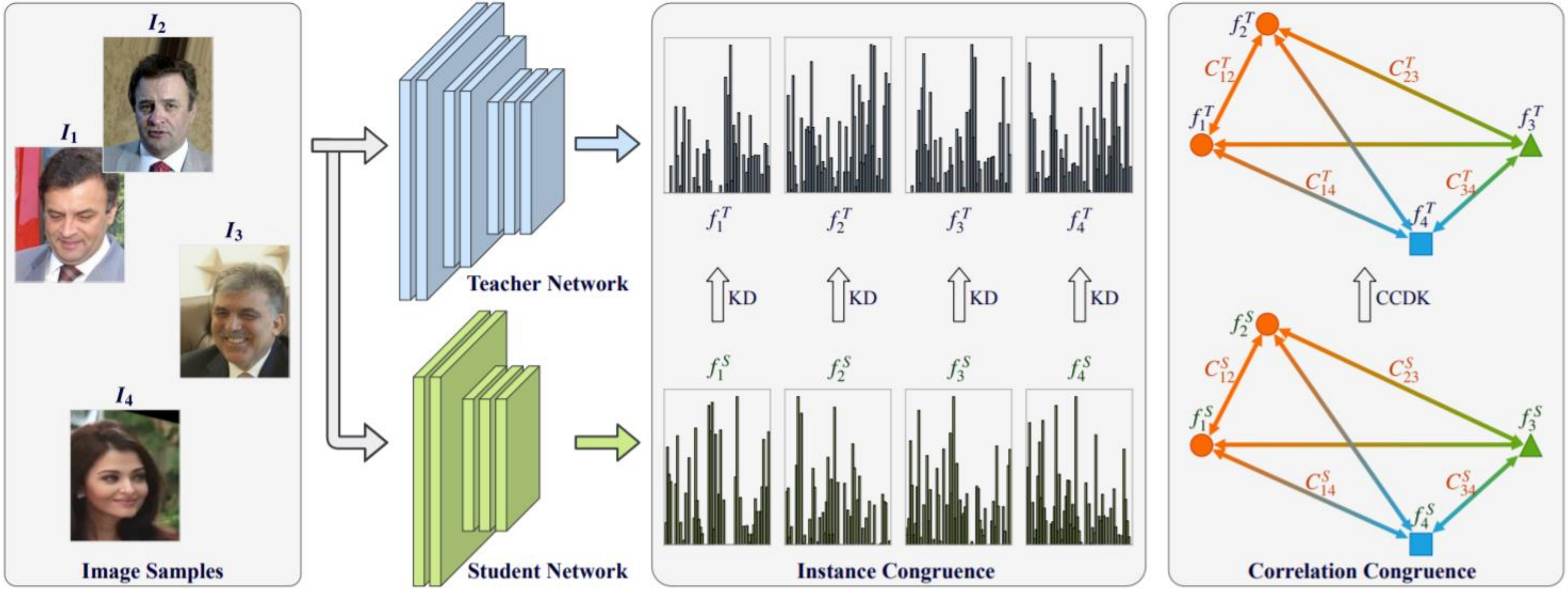


$$\mathcal{L}_{\mathrm{FC}} := \mathcal{H}(F_{\mathrm{S}}, F_{\mathrm{T}}) = ||F_{\mathrm{S}} - F_{\mathrm{T}}||$$

Exclusivity-Consistency Regularized Knowledge Distillation for Face Recognition

3、知识蒸馏

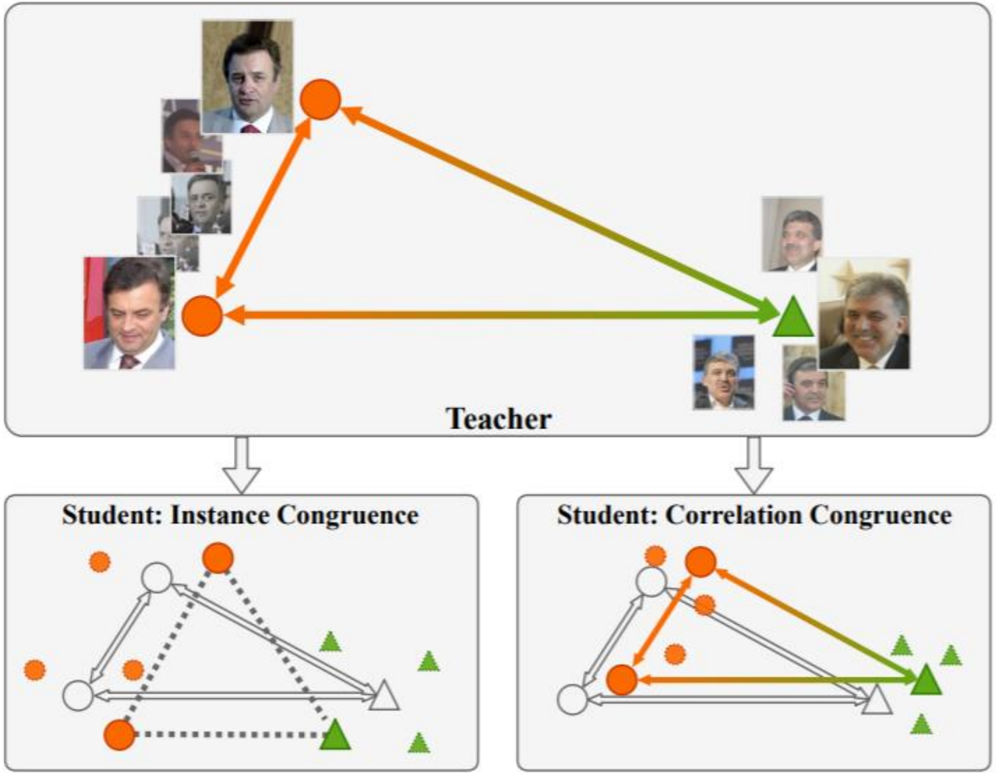
- **CC-KD**



Correlation Congruence for Knowledge Distillation

3、知识蒸馏

- **CC-KD**



Correlation Congruence for Knowledge Distillation

3、知识蒸馏

- **CC-KD**

$$\boldsymbol{F}_t = matrix(\boldsymbol{f}_1^t, \boldsymbol{f}_2^t, ..., \boldsymbol{f}_n^t),$$
$$\boldsymbol{F}_s = matrix(\boldsymbol{f}_1^s, \boldsymbol{f}_2^s, ..., \boldsymbol{f}_n^s).$$

$$\psi : \boldsymbol{F} \rightarrow \boldsymbol{C} \in \mathbb{R}^{n \times n} \qquad \boldsymbol{C}_{ij} = \varphi(\boldsymbol{f}_i, \boldsymbol{f}_j), \quad \boldsymbol{C}_{ij} \in \mathbb{R}$$

$$\begin{aligned} L_{CC} &= \frac{1}{n^2} \|\psi(\boldsymbol{F}_t) - \psi(\boldsymbol{F}_s)\|_2^2 \\ &= \frac{1}{n^2} \sum_{i,j} (\varphi(\boldsymbol{f}_i^s, \boldsymbol{f}_j^s) - \varphi(\boldsymbol{f}_i^t, \boldsymbol{f}_j^t))^2. \end{aligned}$$

Correlation Congruence for Knowledge Distillation

人脸识别-模型压缩

1、小网络的设计

2、低秩分解和二值化

3、知识蒸馏

4、整体总结

4、整体总结

数据	特征提取	分类损失
第一章：常见训练测试数据库	第二章：传统手工特征	第二章：传统分类器
第六章：常见数据分布	第三和第四章：深度学习模型	第四章：深度学习损失函数
	第七章：模型压缩	

4、整体总结

数据

第一章：常见训练测试数据库，比如CASIA-WebFace和LFW，各种测试协议的理解；

第六章：常见数据分布，比如噪声、长尾、无标签数据、监控人脸等。很多特定场景的数据，现有算法处理不是很好，发论文相对比较好。

4、整体总结

特征提取

第二章 传统手工特征: LBP，HOG；PCA，LDA；字典学习等；

第三和第四章 深度学习模型：DeepID系列定义了深度学习人脸识别的基本流程，分类网络架构和人脸特征的网络架构；

第七章 模型压缩：小网络模型的设计，网络结构搜索。

4、整体总结

分类损失

第二章 传统分类器：卡阈值，Joint Bayesian，SVM等；

第四章 深度学习损失函数：Softmax，Margin-based Softmax (SphereFace, CosFace, ArcFace，？？), Mining-Softmax (Hard Example Mining, FocalLoss), Metric Learning (Contrastive Loss, Triplet Loss)。

课程作业

作业：复现知识蒸馏算法

1. 从零训练MobileFaceNet；

2. 从零训练(1/2)MobileFaceNet (所有卷积层kernel数目减半)

3. 知识蒸馏方式训(1/2)MobileFaceNet，拟合概率和拟合特征两种方式做LFW结果对比

感谢聆听！

Thanks for Listening