TensorFlow

PYTORCH

深度学习-图像处理篇

作者：神秘的wz

MobileNet详解

传统卷积神经网络，**内存需求大、运算量大**
导致无法在移动设备以及嵌入式设备上运行

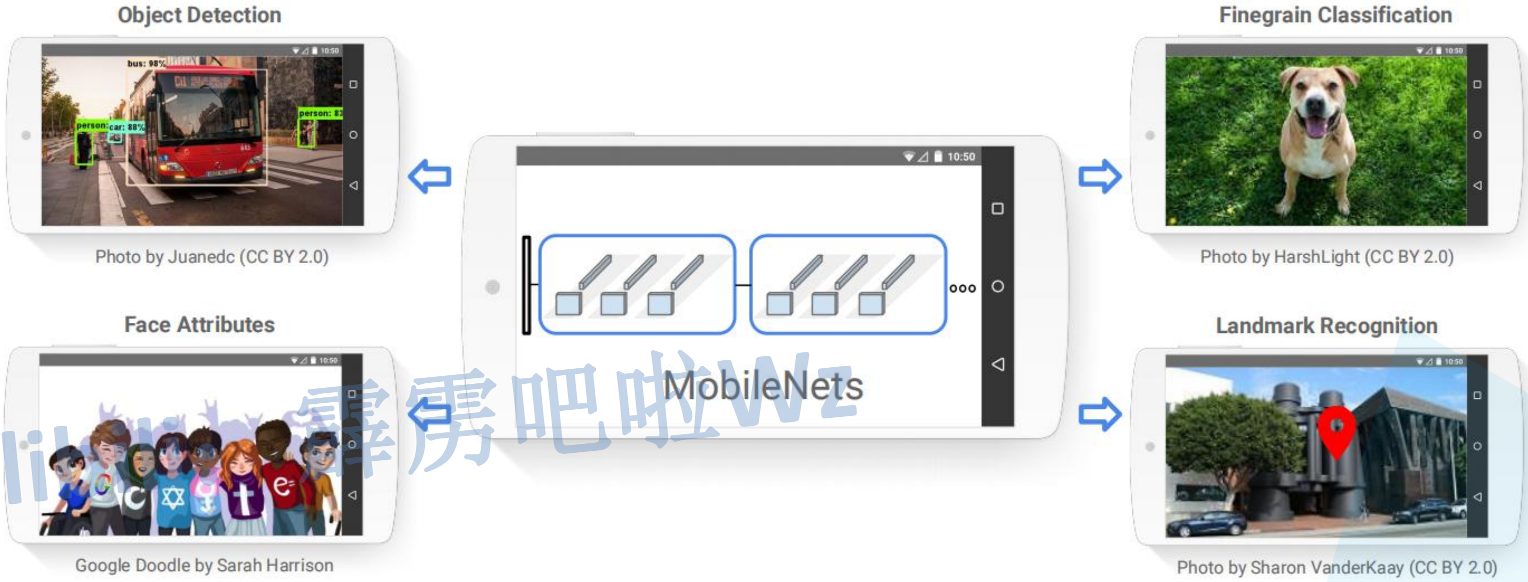
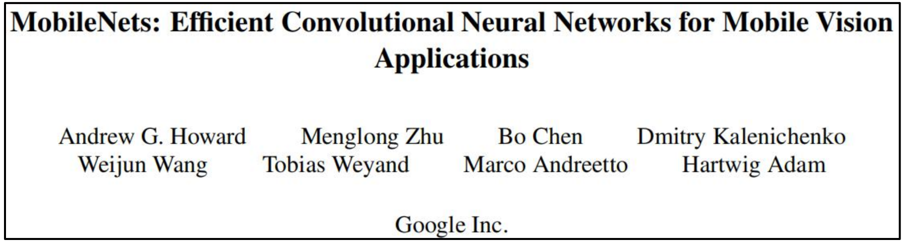


Figure 1. MobileNet models can be applied to various recognition tasks for efficient on device intelligence.

MobileNet详解

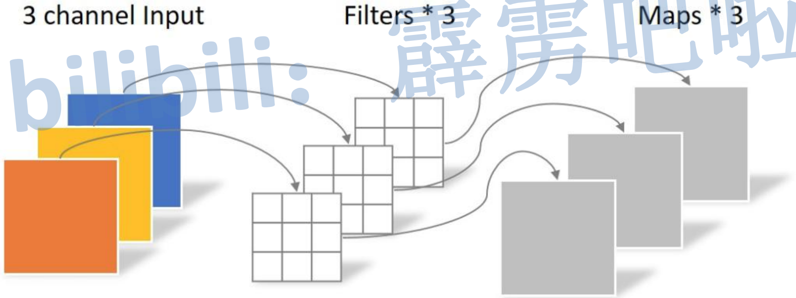
MobileNet网络是由google团队在2017年提出的，专注于移动端或者嵌入式设备中的轻量级CNN网络。相比传统卷积神经网络，在准确率小幅降低的前提下大大减少模型参数与运算量。(相比VGG16准确率减少了0.9%，但模型参数只有VGG的1/32)



MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

Andrew G. Howard　Menglong Zhu　Bo Chen　Dmitry Kalenichenko
Weijun Wang　Tobias Weyand　Marco Andreetto　Hartwig Adam

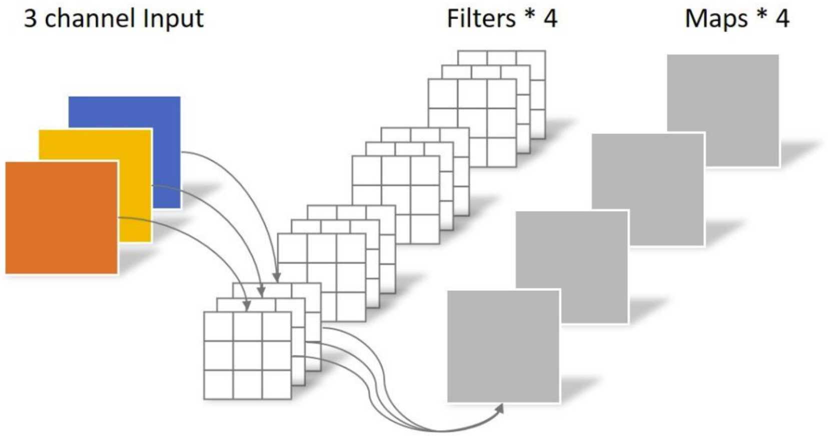
Google Inc.



网络中的亮点：

- **Depthwise Convolution(大大减少运算量和参数数量)**
- **增加超参数α、β**

MobileNet详解

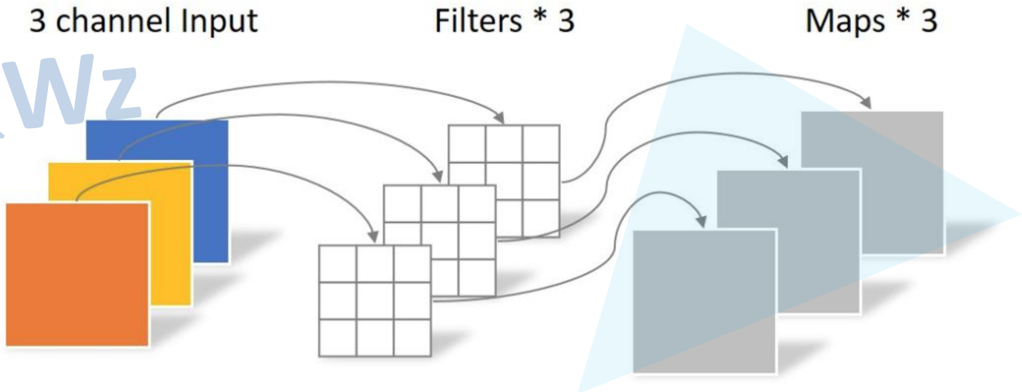


- 卷积核channel=输入特征矩阵channel
- 输出特征矩阵channel=卷积核个数

传统卷积

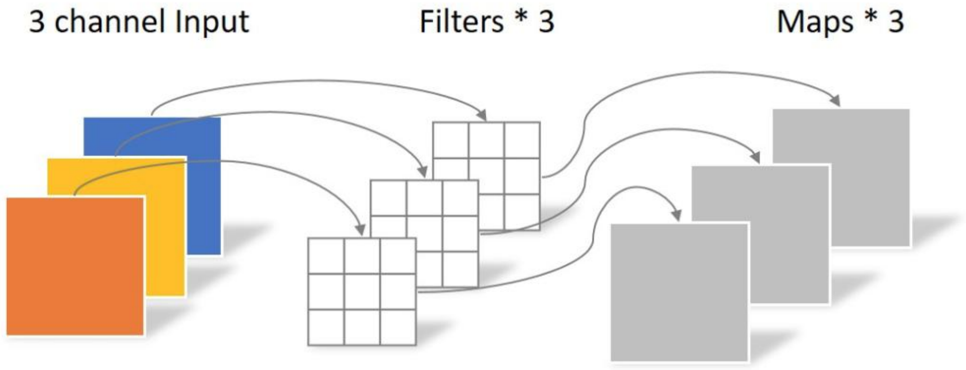
DW卷积

Depthwise Conv



- 卷积核channel=1
- 输入特征矩阵channel=卷积核个数=输出特征矩阵channel

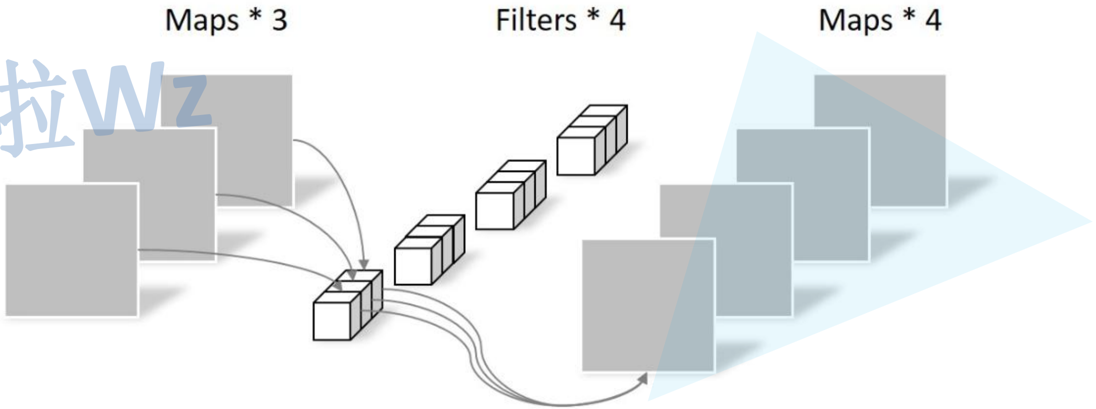
MobileNet详解



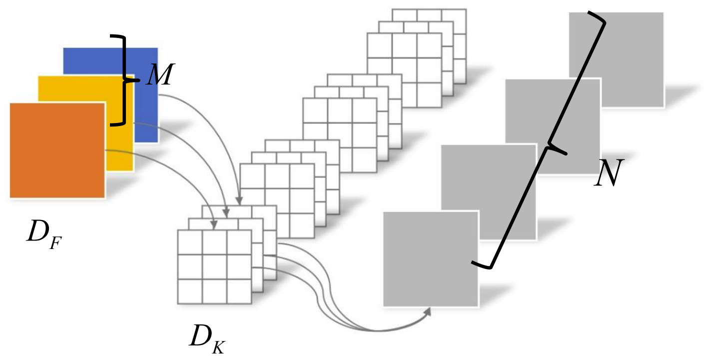
DW卷积
Depthwise Conv

Depthwise Separable Conv

PW卷积
Pointwise Conv



MobileNet详解



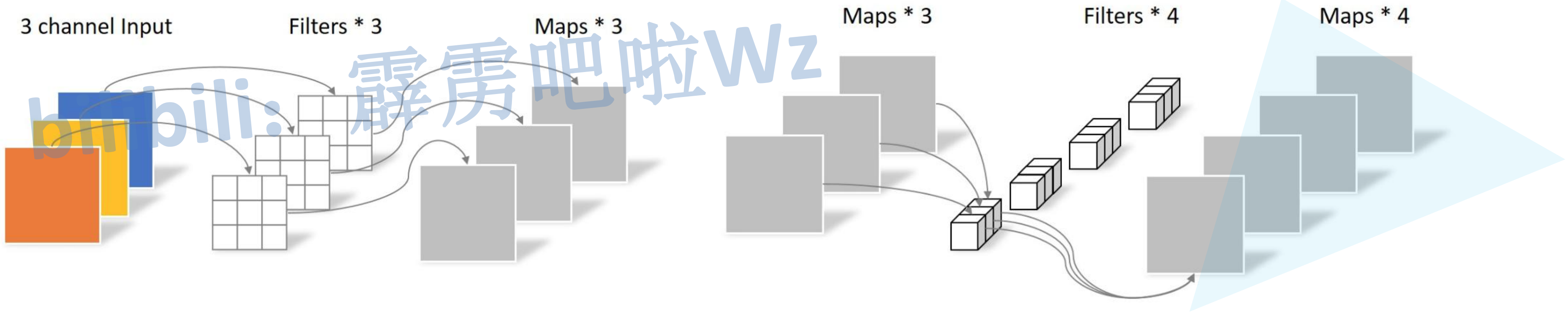
$$\frac{\boxed{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}}{\boxed{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F}}$$

DW + PW

普通卷积

$$= \frac{1}{N} + \frac{1}{D_K^2} = \frac{1}{N} + \frac{1}{9}$$

理论上普通卷积计算量是DW+PW的8到9倍



MobileNet详解

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× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
5× Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
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	$1 \times 1 \times 1000$

Table 8. MobileNet Comparison to Popular Models

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

Multiply-Add
计算量

Table 6. MobileNet Width Multiplier

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

α

Width Multiplier

Table 7. MobileNet Resolution

Resolution	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
1.0 MobileNet-192	69.1%	418	4.2
1.0 MobileNet-160	67.2%	290	4.2
1.0 MobileNet-128	64.4%	186	4.2

β

Resolution Multiplier

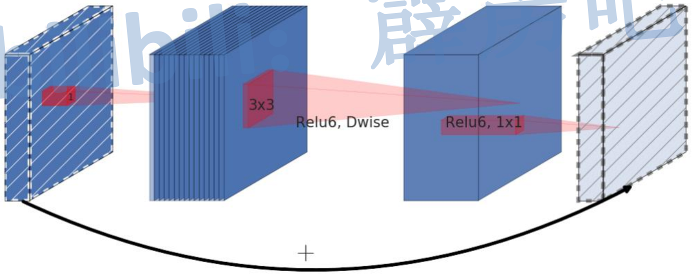
depthwise部分的卷积核容易费掉，即卷积核参数大部分为零。

MobileNet详解

MobileNet v2网络是由google团队在2018年提出的，相比MobileNet V1网络，准确率更高，模型更小。

MobileNetV2: Inverted Residuals and Linear Bottlenecks

Mark Sandler Andrew Howard Menglong Zhu Andrey Zhmoginov Liang-Chieh Chen
Google Inc.



网络中的亮点：

- **Inverted Residuals（倒残差结构）**
- **Linear Bottlenecks**

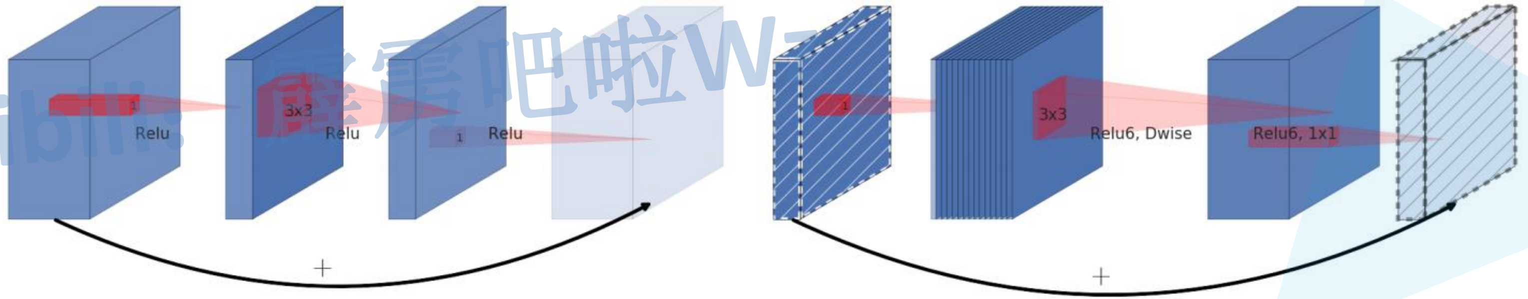
MobileNet详解

① 1x1 卷积降维
② 3x3 卷积
③ 1x1 卷积升维

① 1x1 卷积升维
② 3x3 卷积
③ 1x1 卷积降维

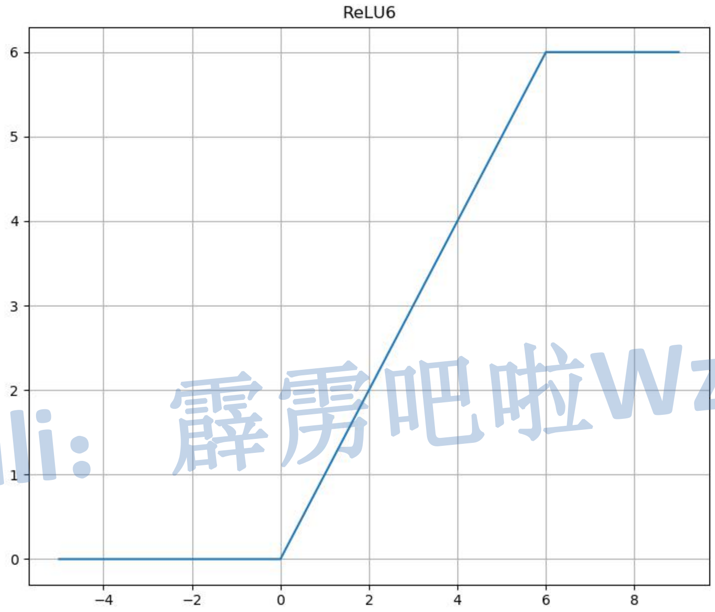
(a) Residual block

(b) Inverted residual block



MobileNet详解

$$y = \mathrm{ReLU6}(x) = \min(\max(x, 0), 6)$$



MobileNet详解

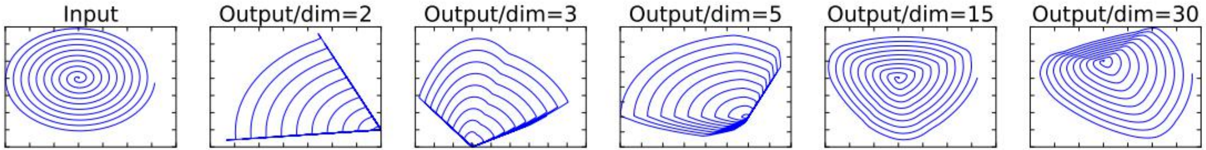
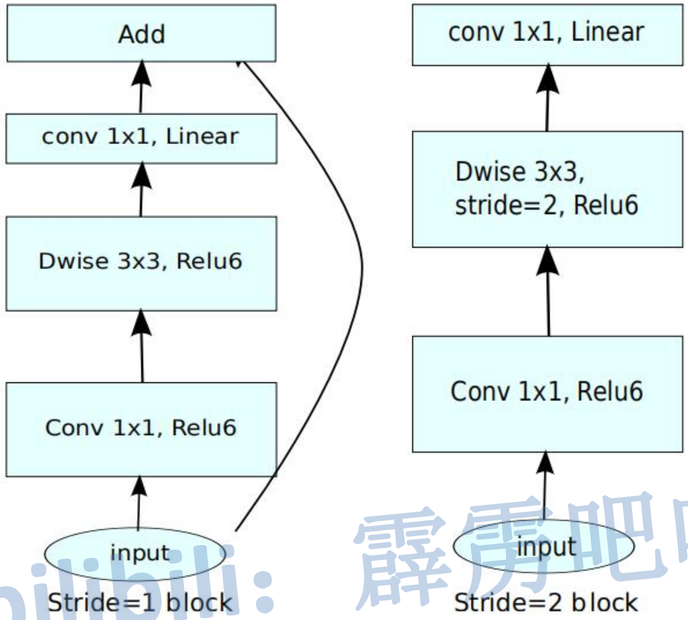


Figure 1: Examples of ReLU transformations of low-dimensional manifolds embedded in higher-dimensional spaces. In these examples the initial spiral is embedded into an n-dimensional space using random matrix T followed by ReLU, and then projected back to the 2D space using T^{-1}. In examples above $n = 2, 3$ result in information loss where certain points of the manifold collapse into each other, while for $n = 15$ to 30 the transformation is highly non-convex.

ReLU激活函数对低维特征信息照成大量损失

MobileNet详解



(d) Mobilenet V2

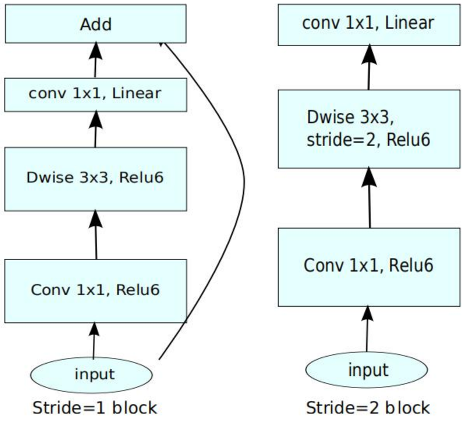
Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Table 1: *Bottleneck residual block* transforming from k to k' channels, with stride s, and expansion factor t.

当stride=1且输入特征矩阵与输出特征矩阵shape相同时才有shortcut连接

MobileNet详解

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	



- t是扩展因子
- c是输出特征矩阵深度channel
- n是bottleneck的重复次数
- s是步距（针对第一层，其他为1）

MobileNet详解

Classification

Network	Top 1	Params	MAdds	CPU
MobileNetV1	70.6	4.2M	575M	113ms
ShuffleNet (1.5)	71.5	**3.4M**	292M	-
ShuffleNet (x2)	73.7	5.4M	524M	-
NasNet-A	74.0	5.3M	564M	183ms
MobileNetV2	**72.0**	**3.4M**	**300M**	**75ms**
MobileNetV2 (1.4)	**74.7**	6.9M	585M	**143ms**

Object Detection

Network	mAP	Params	MAdd	CPU
SSD300[34]	23.2	36.1M	35.2B	-
SSD512[34]	26.8	36.1M	99.5B	-
YOLOv2[35]	21.6	50.7M	17.5B	-
MNet V1 + SSDLite	22.2	5.1M	1.3B	270ms
MNet V2 + SSDLite	22.1	**4.3M**	**0.8B**	200ms

沟通方式

1.github

https://github.com/WZMIAOMIAO/deep-learning-for-image-processing

2.CSDN

https://blog.csdn.net/qq_37541097/article/details/103482003

3.bilibili

https://space.bilibili.com/18161609/channel/index

尽可能每周更新

感谢各位的观看！