# MobileNet

在之前的文章中讲的AlexNet、VGG、GoogLeNet以及ResNet网络，它们都是传统卷积神经网络（都是使用的传统卷积层），缺点在于内存需求大、运算量大导致无法在移动设备以及嵌入式设备上运行。而本文要讲的MobileNet网络就是专门为移动端，嵌入式端而设计。

优点：

* DW网络大大减少运算量和参数数量
* 增加超参数和（α参数是一个倍率因子，用来调整卷积核的个数，β是控制输入网络的图像尺寸参数）

## MobileNet V1

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)。

### 1.1传统卷积与DW卷积对比

要说MobileNet网络的优点，无疑是其中的Depthwise Convolution结构(大大减少运算量和参数数量)。下图展示了传统卷积与DW卷积的差异，在传统卷积中，每个卷积核的channel与输入特征矩阵的channel相等（每个卷积核都会与输入特征矩阵的每一个维度进行卷积运算）。而在DW卷积中，每个卷积核的channel都是等于1的（每个卷积核只负责输入特征矩阵的一个channel，故卷积核的个数必须等于输入特征矩阵的channel数，从而使得输出特征矩阵的channel数也等于输入特征矩阵的channel数），具体的说明可以看图1.1。

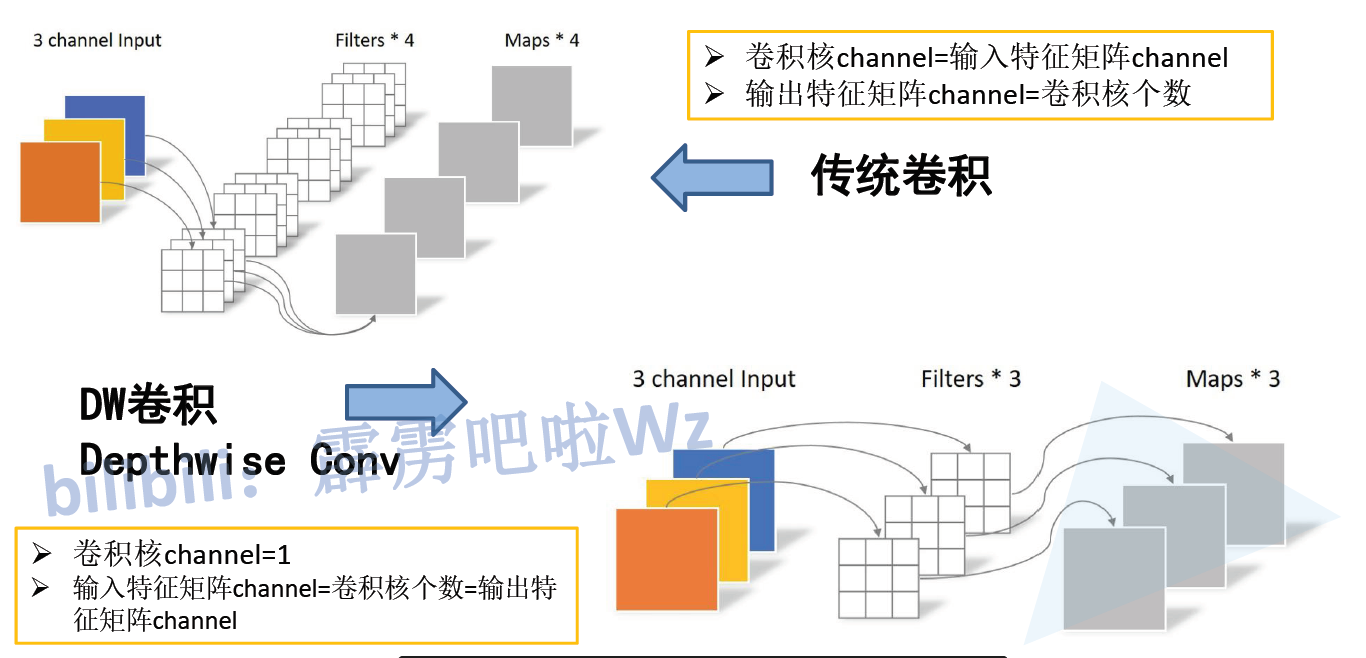


图1.1 传统卷积和DW卷积图示

### 1.2 V1网络（DP+PW）

刚刚说了使用DW卷积后输出特征矩阵的channel是与输入特征矩阵的channel相等的，如果想改变/自定义输出特征矩阵的channel，那只需要在DW卷积后接上一个PW卷积即可，如下图所示，其实PW卷积就是普通的卷积而已（只不过卷积核大小为1）。通常DW卷积和PW卷积是放在一起使用的，一起叫做Depthwise Separable Convolution（深度可分卷积）,卷积说明如图1.2如下：

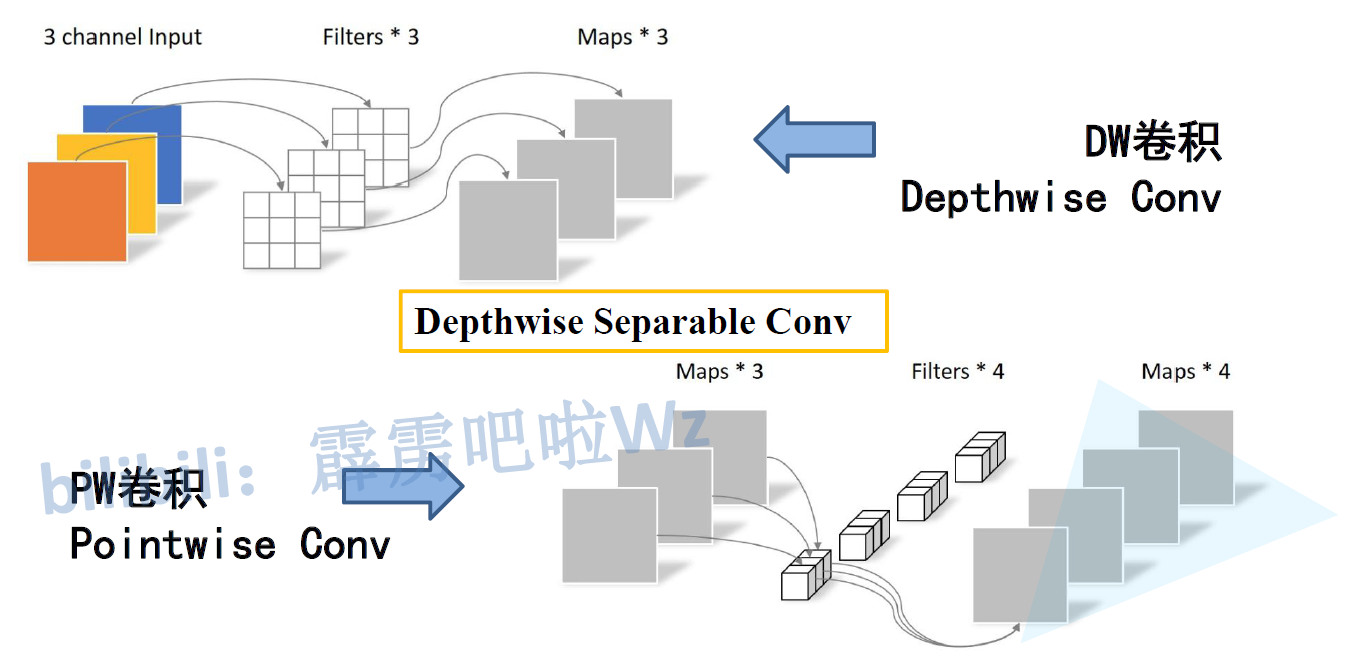
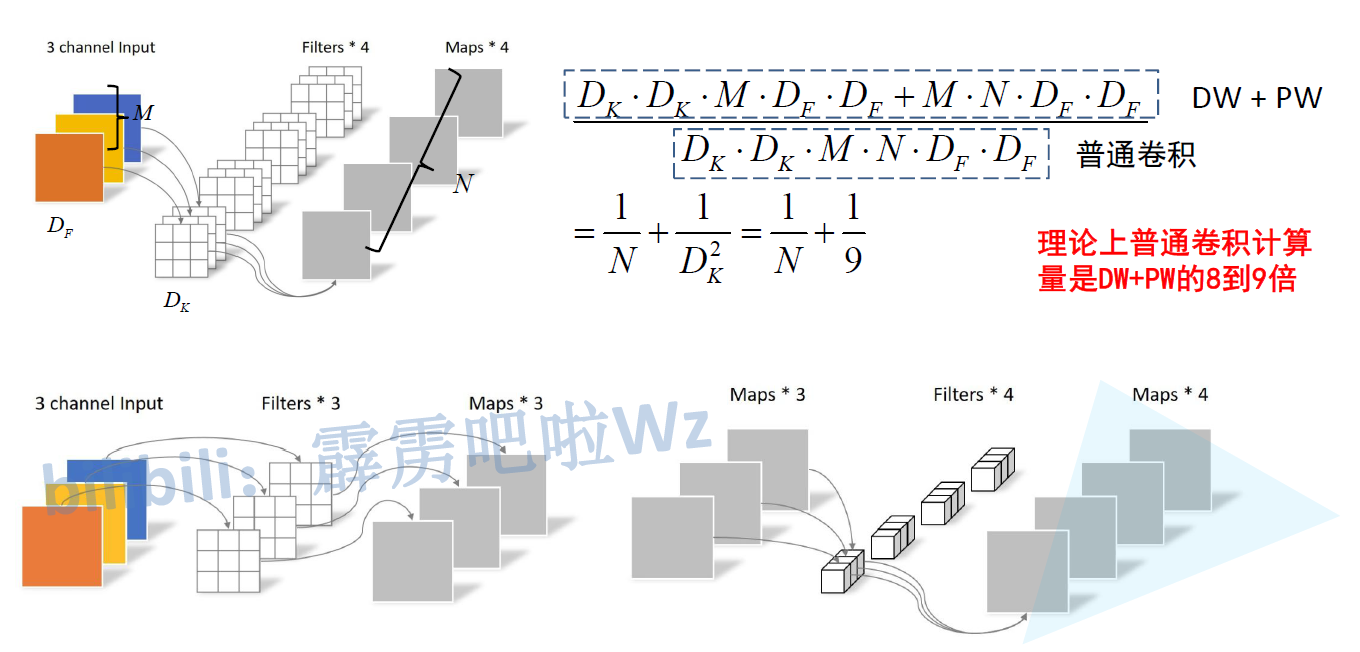


图1.2 DP卷积示意图

### 1.3计算量对比

那Depthwise Separable Convolution（深度可分卷积）与传统的卷积相比有到底能节省多少计算量呢，下图对比了这两个卷积方式的计算量，其中是输入特征矩阵的宽高（这里假设宽和高相等），是卷积核的大小，M是输入特征矩阵的channel，N是输出特征矩阵的channel，（这里假设stride等于1）,；传统卷积计算量，所以在我们mobilenet网络中DW卷积都是是使用3x3大小的卷积核。所以理论上普通卷积计算量是DW+PW卷积的8到9倍（公式来源于原论文）：



### 1.4 V1网络结构及参数（和）

在了解完Depthwise Separable Convolution（深度可分卷积）后在看下mobilenet v1的网络结构，图1.3中左侧的表格是mobileNetv1的网络结构，表中标Conv的表示普通卷积，Conv dw代表刚刚说的DW卷积，s表示步距，根据表格信息就能很容易的搭建出mobileNet v1网络。在mobilenetv1原论文中，还提出了两个超参数，一个是α一个是β。α参数是一个倍率因子，用来调整卷积核的个数，β是控制输入网络的图像尺寸参数，下图右侧给出了使用不同α和β网络的分类准确率，计算量以及模型参数：

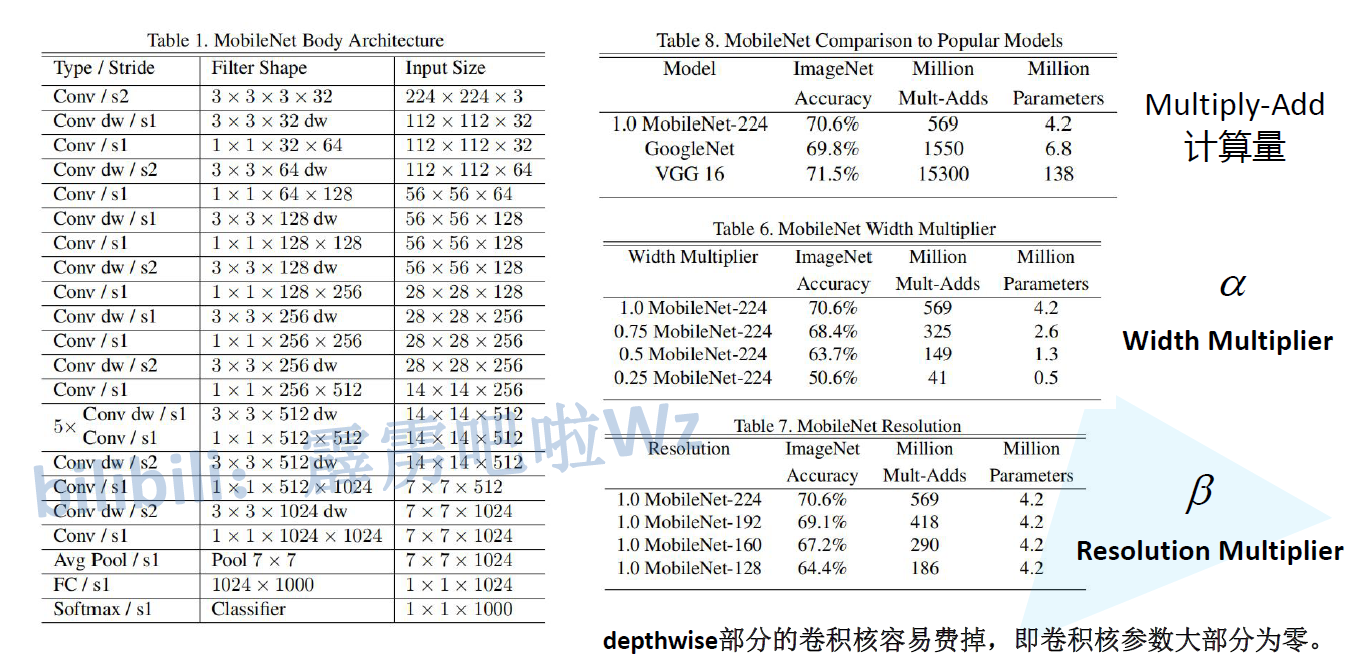


图1.3 网络结构与参数调整

从上图1.3的右上角的表格可以看出来，MobileNet-224（输入图像尺寸是224\*224），虽然准确率比VGG-16降低了0.9%，但是计算量却大约是它的1/3,参数量也大概是VGG-16的1/32，这样一来就大大减少了计算量。而当倍率因子逐渐减小时，准确率，计算量和参数数量也随之减少。分辨率因子逐渐减小时，准确率，计算量随之减少，但是参数数量并没有改变。

所以，综上所述，可以根据不同项目的需求，改变两个超参数，来进行网络的优化。

## MobileNet V2

在MobileNet v1的网络结构表中能够发现，网络的结构就像VGG一样是个直筒型的，不像ResNet网络有shorcut之类的连接方式。而且有人反映说MobileNet v1网络中的DW卷积很容易训练废掉，效果并没有那么理想。所以我们接着看下MobileNet v2网络。

优点：

* 倒残差结构（Inverted residual block）
* Linear Bottleneck

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。刚刚说了MobileNet v1网络中的亮点是DW卷积，那么在MobileNet v2中的亮点就是Inverted residual block（倒残差结构），如图2.1所示，左侧是ResNet网络中的残差结构，右侧就是MobileNet v2中的倒残差结构。在残差结构中是1x1卷积降维->3x3卷积->1x1卷积升维，在倒残差结构中正好相反，是1x1卷积升维->3x3DW卷积->1x1卷积降维。为什么要这样做，原文的解释是高维信息通过ReLU激活函数后丢失的信息更少（注意倒残差结构中基本使用的都是ReLU6激活函数，但是最后一个1x1的卷积层使用的是线性激活函数）。

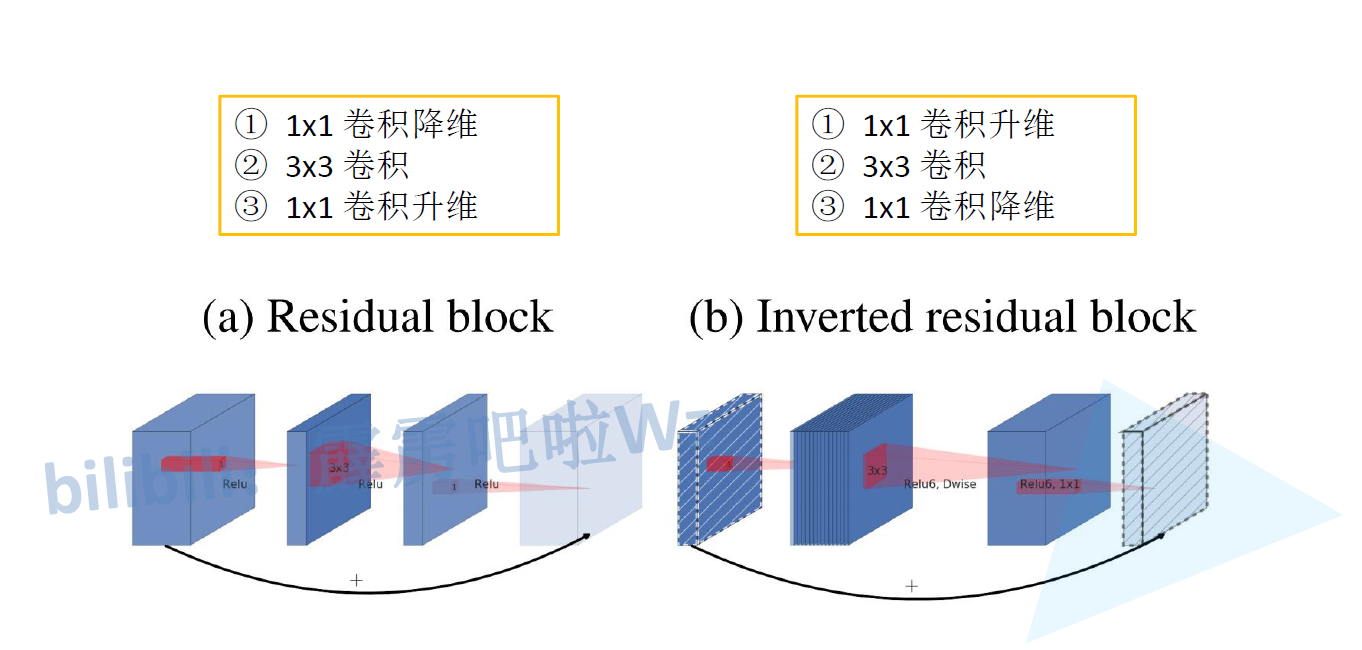
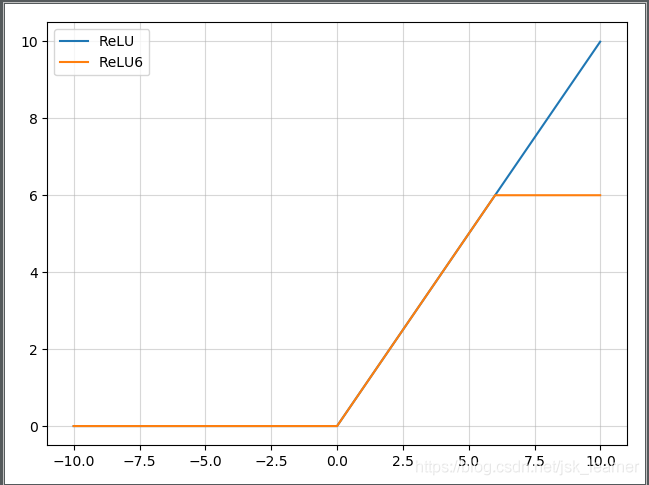
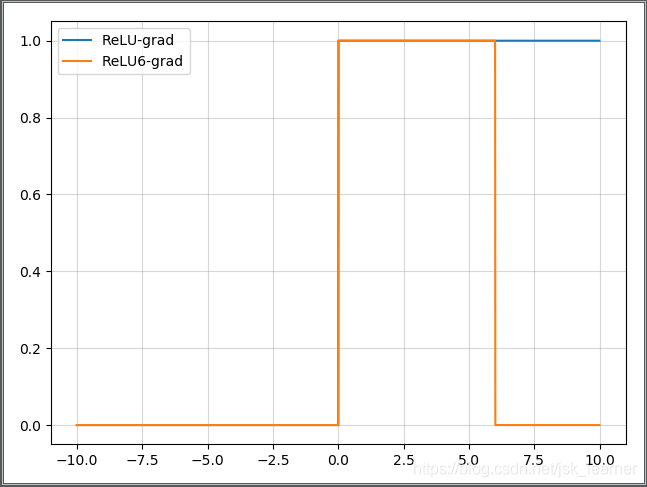


图2.1 残差结构和到倒残差结构

### 2.1．ReLU函数和ReLU6函数





2.2 ReLU函数和ReLU6函数图像和导函数图像对比

ReLU6函数与ReLU函数不同的地方在于，当x大于6以后，y的值不会不会一直增大，而是稳定在y=6。主要是为了在移动端float16的低精度的时候，也能有很好的数值分辨率，如果对ReLu的输出值不加限制，那么输出范围就是0到正无穷，而低精度的float16无法精确描述其数值，带来精度损失。

### 2.2 倒FUT残差结构说明

在使用倒残差结构时需要注意下，并不是所有的倒残差结构都有shortcut连接，只有当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（只有当shape相同时，两个矩阵才能做加法运算，当stride=1时并不能保证输入特征矩阵的channel与输出特征矩阵的channel相同）。

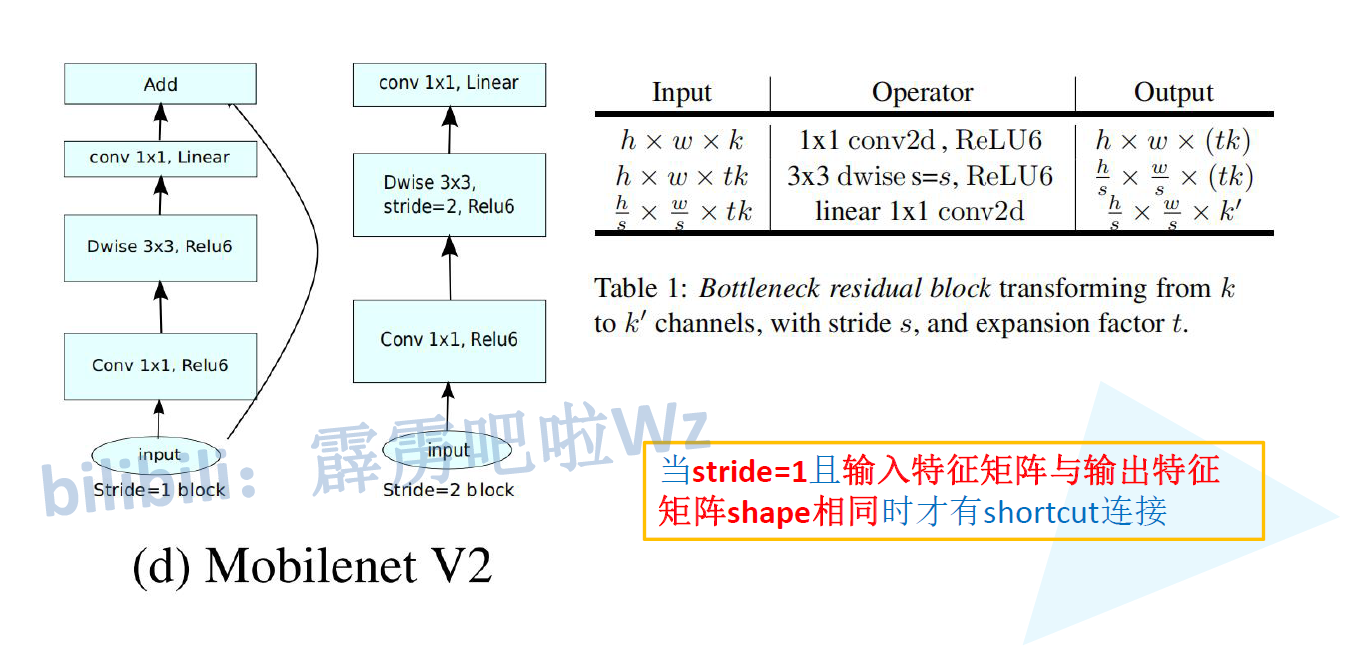


图2.3 残差结构说明

### 2.3 V2结构说明

下图是MobileNet v2网络的结构表，其中t代表的是扩展因子（倒残差结构中第一个1x1卷积的扩展因子），c代表输出特征矩阵的channel，n代表倒残差结构重复的次数，s代表步距（注意：这里的步距只是针对重复n次的第一层倒残差结构，后面的都默认为1）

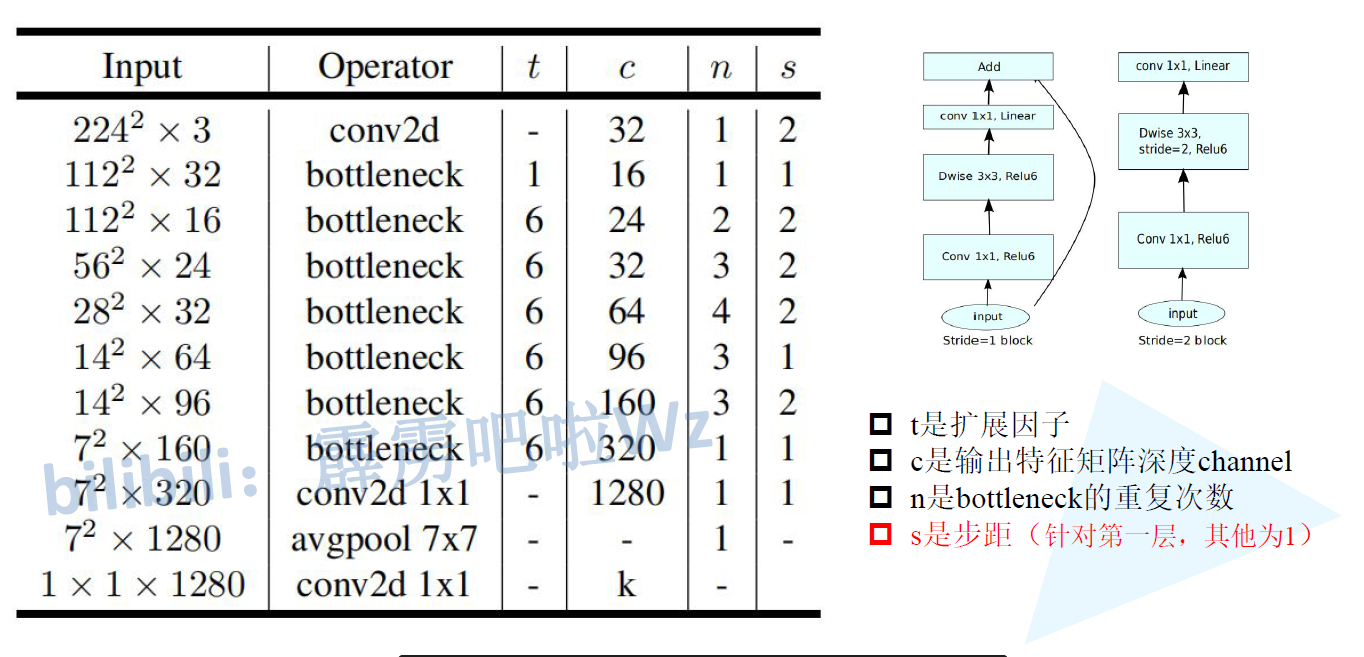


图2.4 v2网络结构

### 2.4 Mobilenet V2网络在分类和检测方面的标表现

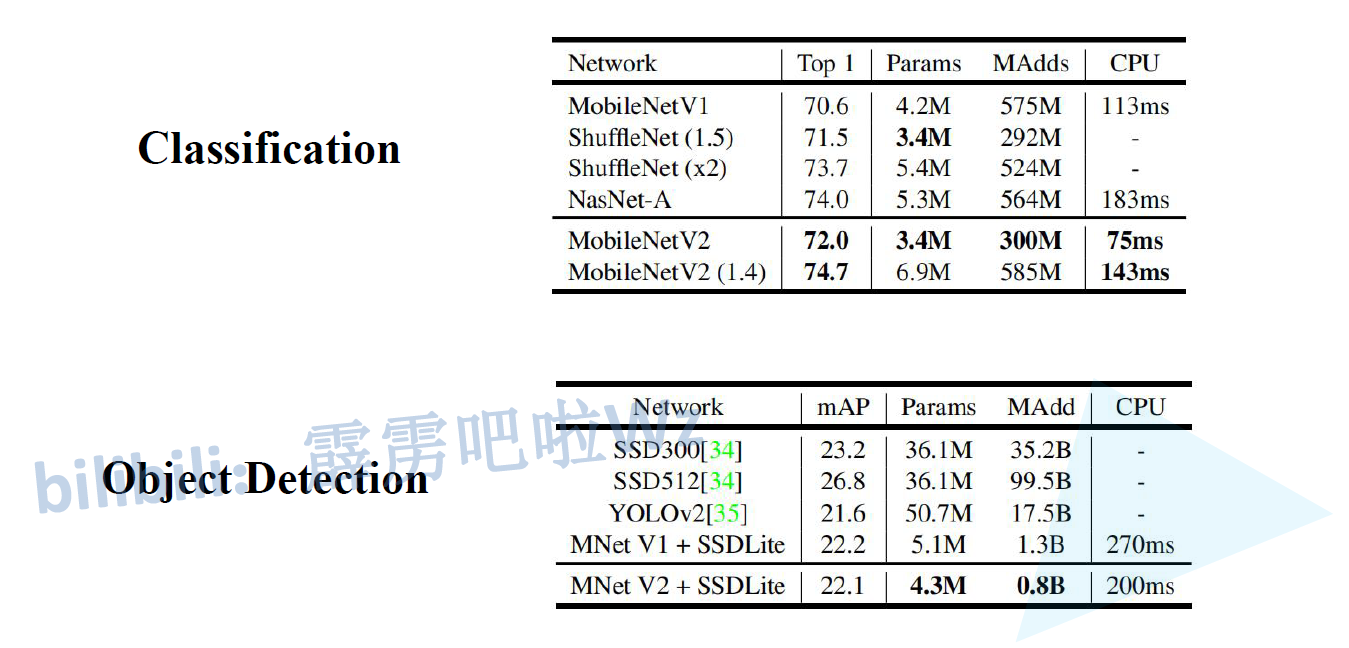


图2.5 分类与检测方面的表现