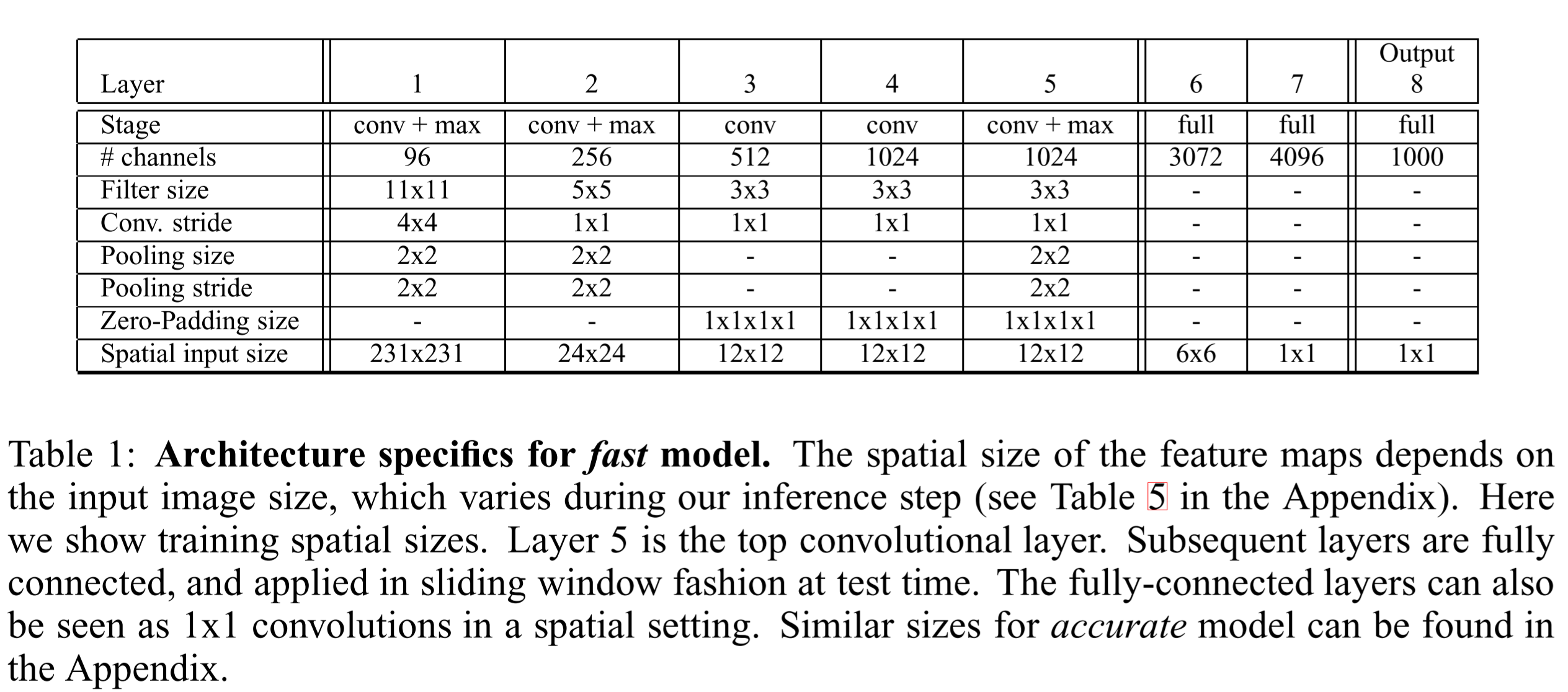
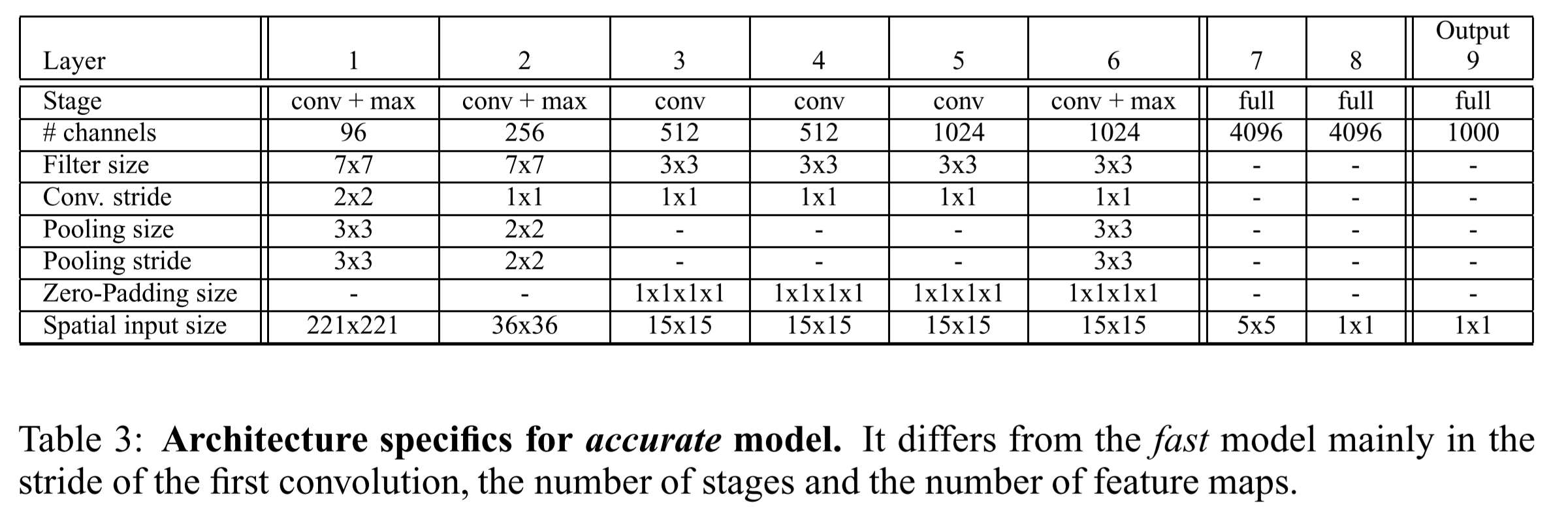
OverFeat Notes

1. 两种结构的模型





1. Alexnet图片分类回顾

* **训练阶段**

每张训练图片256\*256，然后我们随机裁剪出224\*224大小的图片，作为CNN的输入进行训练。

* **测试阶段**

输入256\*256大小的图片，我们从图片的5个指定的方位(上下左右+中间)进行裁剪出5张224\*224大小的图片，然后水平镜像一下再裁剪5张，这样总共有10张；然后我们把这10张裁剪图片分别送入已经训练好的CNN中，分别预测结果，最后用这10个结果的平均作为最后的输出。

1. FCN（Fully Convolutional Network­­）

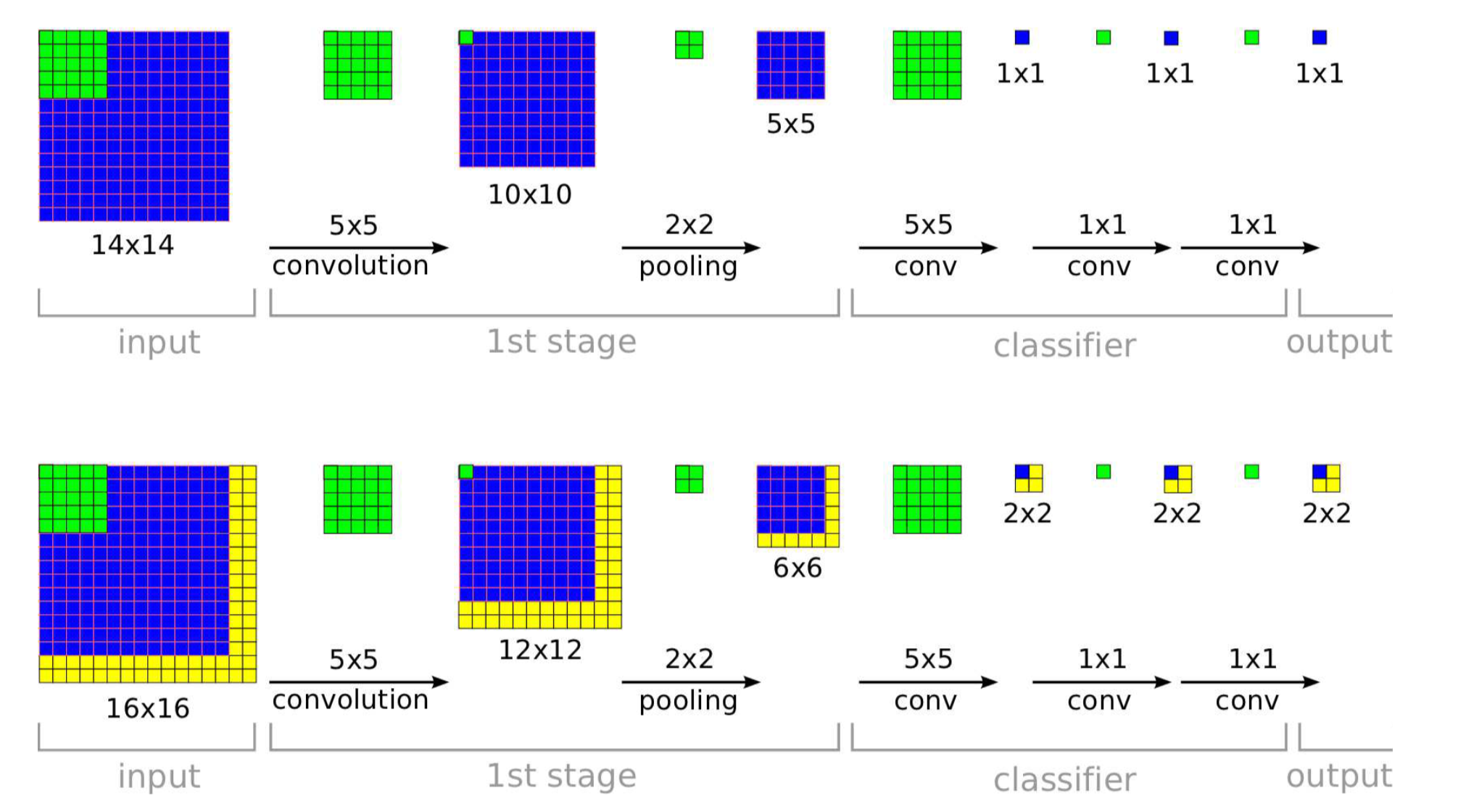
* **卷积层→全连接层**

直接对feature map进行卷积，如输入大小是5\*5\*4096，传统做法是将其拉成一个一维向量，然后通过矩阵乘法直接做全连接操作（W的大小为4096 by 5\*5\*4096），FCN的做法是使用5\*5大小的卷积核直接对特征图做卷积操作，最后都能得到1\*1\*4096大小的输出。

* **全连接层→全连接层**

把它看成是用1\*1大小的卷积核进行卷积操作。

* 示意图



如上图所示，上面图中绿色部分表示卷积核大小。假设我们设计了一个CNN模型，输入图片大小是14\*14，通过第一层卷积后我们得到10\*10大小的图片，然后通过池化得到了5\*5大小的图片。OK，关键部分来了，接着要从5\*5大小的图片→1\*1大小的图片：

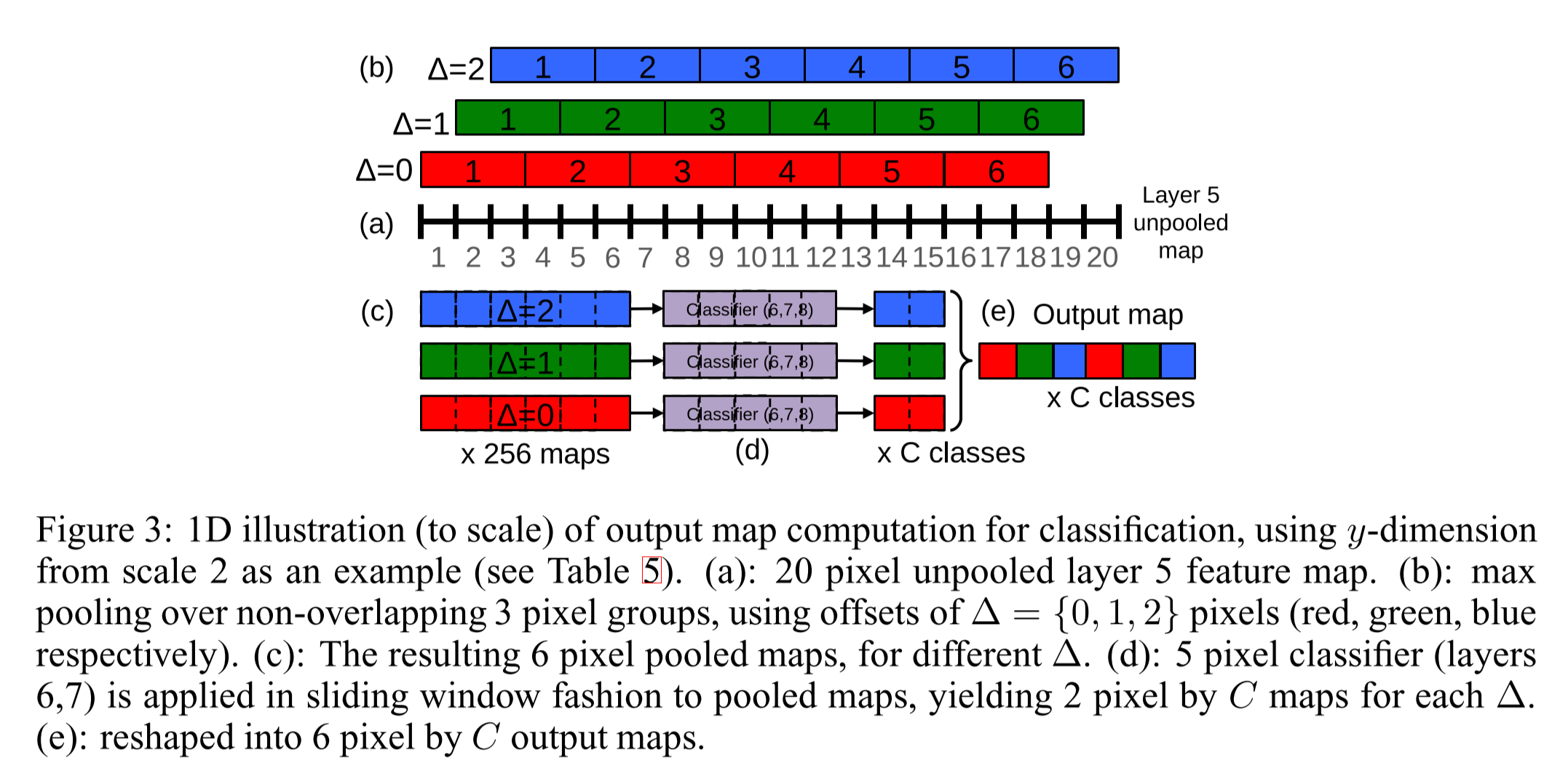
（1）传统的CNN：如果从以前的角度进行理解的话，那么这个过程就是全连接层，我们会把这个5\*5大小的图片，展平成为一个一维的向量，进行计算(写CNN代码的时候，经常会在这里加一个flatten函数，就是为了展平成一维向量)。

（2）FCN：FCN并不是把5\*5的图片展平成一维向量，再进行计算，而是直接采用5\*5的卷积核，对一整张图片进行卷积运算。

AlexNet在测试阶段采用了对输入图片的四个角落进行裁剪、预测，分别得到结果，最后的结果就是类似对应于上面2\*2的预测图。这个2\*2的每个像素点，就类似于对应于一个角落裁剪下来的图片预测分类结果。只不过AlexNet把这4个像素点相加在一起，求取平均值，作为该类别的概率值。

1. Offset max-pooling

* 示意图



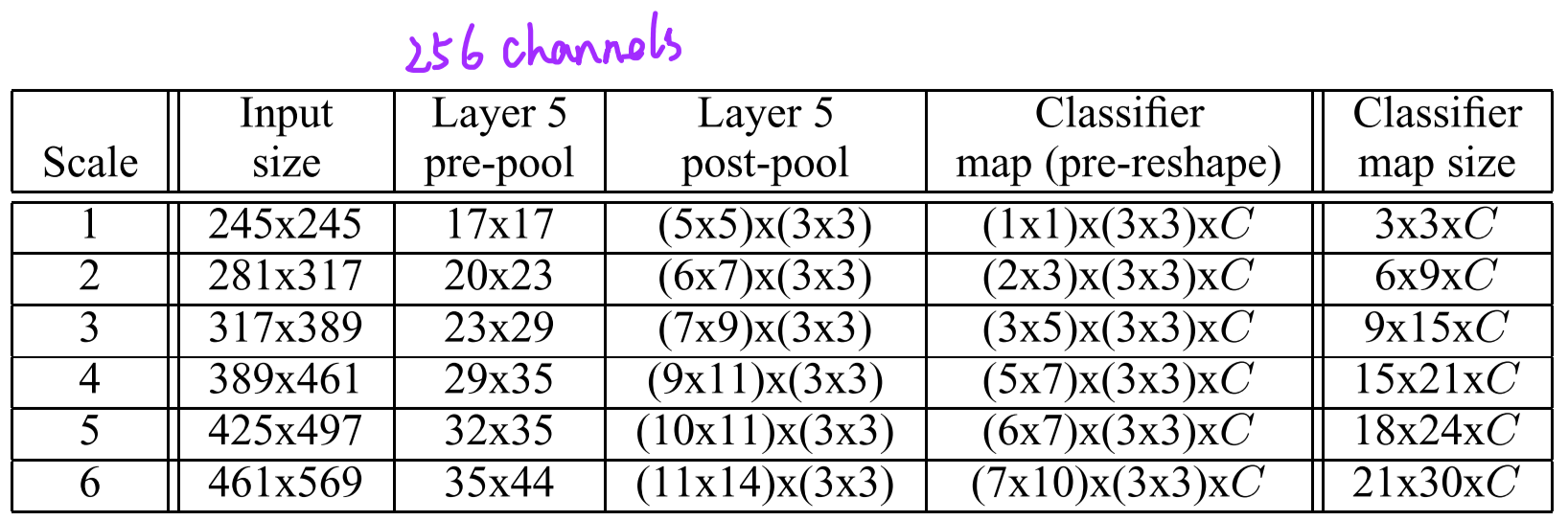
以往的CNN中，一般我们只用了△=0得到池化结果后，就送入一层。文献的方法是，把上面的△=0、△=1、△=2的三种组合方式的池化结果，分别送入网络的下一层。这样的话，我们网络在最后输出的时候，就会出现3种预测结果了。

我们前面说的是一维的情况，如果是2维图片的话，那么(△x, △y)就会有9种取值情况(3\*3)；如果我们在做图片分类的时候，在网络的某一个池化层加入了这种offset 池化方法，然后把这9种池化结果，分别送入后面的网络层，最后我们的图片分类输出结果就可以得到9个预测结果(每个类别都可以得到9种概率值，然后我们对每个类别的9种概率，取其最大值，作为此类别的预测概率值)。

1. OverFeat算法

* 训练阶段与AlexNet基本相同。
* 测试阶段

在测试阶段，我们不再是用一张221\*221大小的图片作为网络的输入，而是用了6张大小都不相同的图片，也就是所谓的多尺度输入预测，如下表格所示：



从Layer-5 pre-pool到Layer-5 post-pool：这一步的实现是通过池化大小为3\*3进行池化，然后△x=0、1、2,△y=0、1、2，这样对于每一张特征图，我们都可以得到9幅池化结果图。以上面表格中的sacle1为例，Layer-5 pre-pool大小是17\*17，经过池化后，大小就是5\*5，然后有3\*3张结果图(不同offset得到的结果)。

从Layer-5 post-pool到Classifier map(pre-reshape)：我们知道在训练的时候，从卷积层到全连接层，输入的大小是4096\*(5\*5)，然后进行全连接，得到4096\*(1\*1)。但是我们现在输入的是各种不同大小的图片，因此接着就采用FCN的招式，让网络继续前向传导。我们从Layer-5 post-pool到第六层的时候，如果把全连接看成是卷积，那么其实这个时候卷积核的大小为5\*5，因为训练的时候，Layer-5 post-pool得到的结果是5\*5。因此在预测分类的时候，假设Layer-5 post-pool 得到的是7\*9(上面表格中的scale 3)，经过5\*5的卷积核进行卷积后，它将得到(7-5+1)\*(9-5+1)=3\*5的输出。

然后我们就只需要在后面把它们拉成一维向量摆放就ok了，这样在一个尺度上，我们可以得到一个C\*N的预测值矩阵，每一列就表示图片属于某一类别的概率值，然后我们求取每一列的最大值，作为本尺度的每个类别的概率值。我们一共用了6种不同尺度做了预测，然后把这六种尺度结果再做一个平均，作为最后的结果。

从上面过程，我们可以看到整个网络分成两部分：layer 1~5这五层我们把它称之为特征提取层；layer 6~output我们把它们称之为分类层。