**1.简单介绍（What）**

Ovefeat是2013年ImageNet定位任务的冠军，同时在分类和检测任务也取得了不错的结果。

（1）它用一个共享的CNN来同时处理图像分类，定位，检测三个任务，可以提升三个任务的表现。

（2）它用CNN有效地实现了一个多尺度的，滑动窗口的方法，来处理任务。

（3）提出了一种方法，通过累积预测来求bounding boxes（而不是传统的非极大值抑制）。

2. **论文动机（Why）**

虽然ImageNet的数据包含一个大致充满图像的中心目标，但是目标在图像中的大小和位置有着显著差异。解决这个问题有几个做法。

（1）使用多个固定大小的滑动窗口移动，对每个扫过的窗口图像做CNN预测。该方法的缺点在于窗口没有包含整个目标，甚至中心也没有，只是包含了一部分（比如狗狗的头），虽然适合做分类，但是定位和检测效果很差。

（2）训练一个卷积网络，不仅产生分类的分布，还产生预测框bouding box（预测目标的大小和位置）。

（3）累积每个位置和尺寸对应类别的置信度。

AlexNet展示了CNN可在图像分类和定位任务上取得了优秀的表现，但是并没有公开描述他们的定位方法。  
这篇论文是第一次清晰地解释CNN如何用于定位和检测。

3. **视觉任务（How）**

论文探索了图像处理的三大任务，按难度上升的顺序分别是：

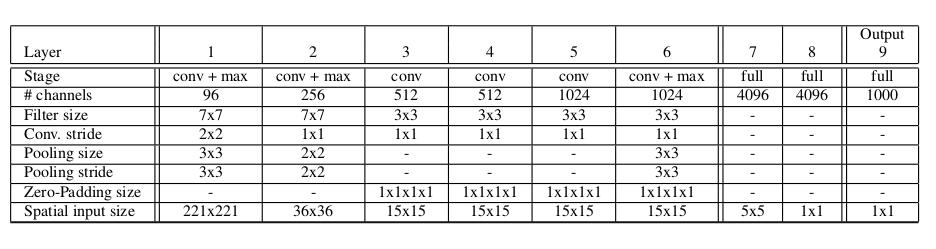
(1)分类（classification），给每一张图像打标签，表示是什么物体。只要概率最大的前5个中有一个是正确的就认为是正确（top5）。

(2)定位（localization），除了打标签，还需要给出目标的位置大小，而且bounding box和真实框的相似度必须达到阈值（比如交并比至少要为0.5）。也有top5指标，5个标签必须有一个标签，分类正确且边框符合条件，才算正确。

(3)检测（detection），一张图像有很多个目标物体，需要全部找出来（分类并定位）。分类和定位使用同一个数据集，而检测使用另外的数据集，里面的物体会更小一些。

(4)下面分为三个部分来讲讲论文是怎么做的（分类，定位和检测），重点主要是讲述分类任务，然后是定位任务，至于最后的检测任务论文没怎么提具体做法，就稍微提了一下。

4. **分类任务**

（1）论文的网络结构和alexNet很类似，在网络设计和测试阶段的做法上做了一些改进。论文的网络分为两个版本，一个快速版，一个精确版。下图是精确版的网络结构图。  


（2）该网络和alexNet类似，有几点不同，一是没有使用对比归一化，二是没有使用重叠的池化，三是stride的超参用2代替了4，大stride可以提升速度，减小精度。

（3）该网络和alexNet最大的不同之处在于测试阶段使用了不同的方法来预测。

（4）alexNet在测试阶段对256\*256的图像做裁剪（四个角落和中间）和水平翻转，得到5\*2也就是10张227\*227的图像，然后送进网络里面得到10个结果求平均来进行预测。这样的做法有两个问题，裁剪时可能忽略了图像的一些区域，以及10张图像有很多重叠部分导致了冗余计算。

**（5）该网络的测试阶段，用到了多尺度的，滑动窗口的方法（实验最多输入了6个不同尺度的图像）。这也是论文最大的创新点。**

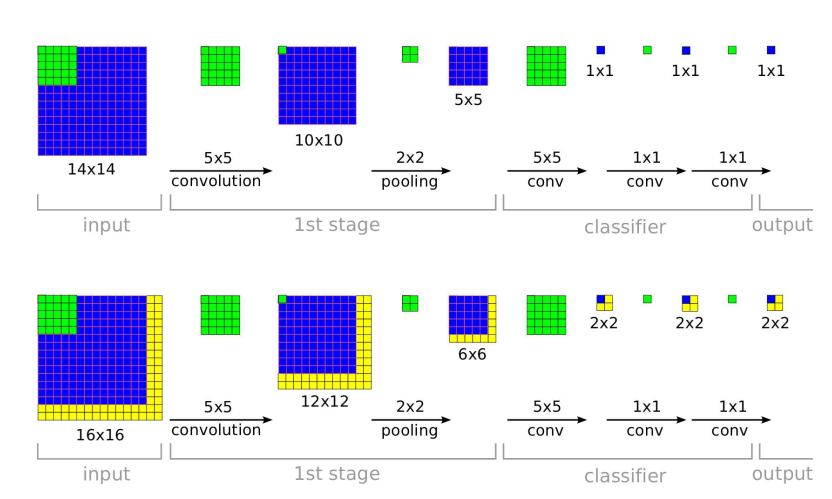
**5.多尺度分类——全卷积（全卷积意为全部都是卷积层）**

（1）上图中各层的输入大小是训练时的，由于在测试时会输入6张不同尺寸的图，所以大小肯定都不一样的。

**（2）全卷积是什么：**上图中后三层的全连接层实际上使用的是全卷积，全连接层是可以转为全卷积的，举例来说，全连接层的输入shape为5\*5\*1024的feature map，输出为4096的话，参数个数就是5\*5\*1024\*4096，这个时候转为全卷积层，那么卷积的参数就是，卷积核大小为5\*5\*1024，卷积核个数为4096，二者的参数量是一样的。

**（3）全卷积导致了什么：**如下图所示，对14\*14的图像进行卷积操作，在得到5\*5的feature map后的这一步，如果使用全连接，就会把它压平再全连接，这样就破坏了feature map的图像位置关系，直接转为一列特征。但是如果使用的是全卷积，最后会得到1\*1\*C的feature map，C是channel数，也是类别的大小。这个时候如果来了一张16\*16的图像，经过全卷积后就会得到2\*2\*C的feature map，这个时候可以对这个2\*2的4个值做一个取最大或平均，就会变成一个值了，以此类推，来了更大的图像，最后得到的feature map就是3\*3\*C，4\*4\*C，5\*5\*C的大小，输出的大小和输入的大小相关，但总是可以对这个输出map池化（取最大）来得到这个类别的值。

**（4）全卷积的好处：**下图中第一个图是训练时用14\*14的图像，最后产生一个输出，下面的图是测试时，可以用16\*16的图像产生了“2\*2”个输出，以此类推我们可以在测试时使用更大的图像（使用多scale），产生“更多”的输出进行（取最大）预测。这个做法相对于传统的滑动窗口（用14\*14大小，步长为2的滑动窗口在16\*16的图像上执行4次卷积操作进行分类）的优点是，只需要执行一次，保证了效率同时可以建模用各种不同尺度图像，不局限于固定的裁剪翻转方式（相对于alexNet测试阶段的做法），而且消除了很多冗余计算，提高了模型的鲁棒性又保证了效率。

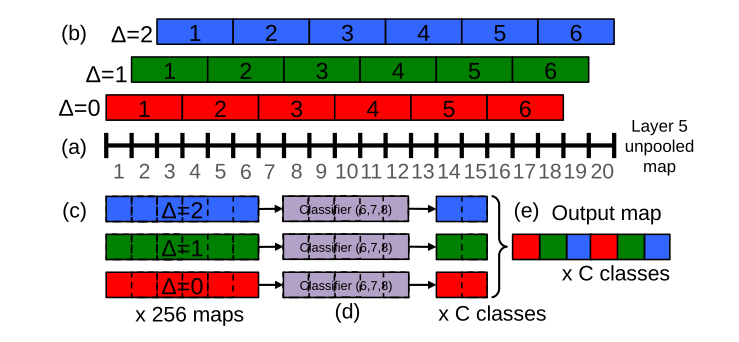


6.**多尺度分类——offset池化**

（1）为了解释结合offset池化后最后计算出来的输出，以下图为例，(a)是第5层得到的暂未池化的一张图的某一维，比如图的大小为20\*23，下图中画出的是20\*23中的20。（20\*23是后面的scale2在第5层得到的一个图的大小，后面我们会用到6个scale，这里以scale2的某一维为例子）。

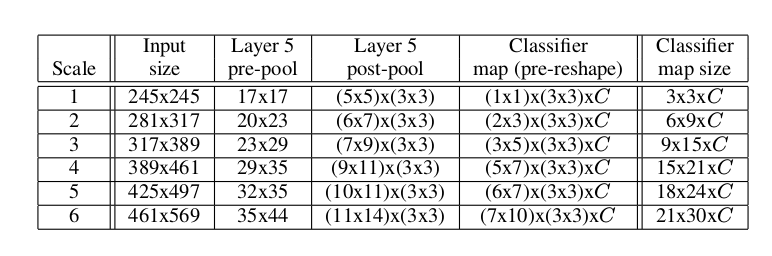
（2）传统的做法，对长度为20的序列进行3\*3的最大池化后会得到长度为6的序列，就是(b)中Δ=0Δ=0这样的池化

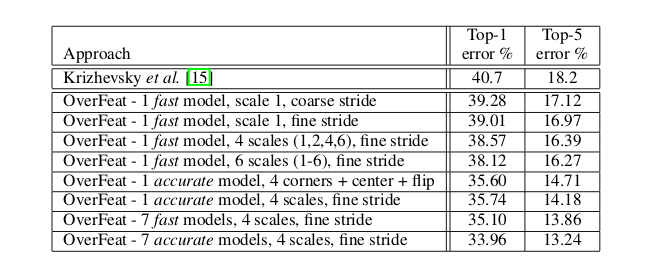
（3）offset池化就是移动一定的位置再池化，(b)中Δ=0,1,2Δ=0,1,2就可以表示可以做三种池化，得到三个结果，因为图像是二维的，所以最后会得到3\*3也就是9种池化结果，最后对于**每个类别就有9个结果**，可以对这些结果集成预测（下图的例子中只考虑一维的所以图中最后会得到三个结果，红蓝绿三种颜色表示三种池化后得到的结果）。

（4）(c)表示进行3\*3池化后得到6\*6的图（6个格子）。(d)表示经过5\*5的全卷积得到2\*2的图（2个格子）。e表示把位置信息（长度为2）和offset方式（3种）交错后得到的最后的输出图。  


（5）上述这个操作会对重复6\*2也就是12次，其中6代表6个scale，如下图所示的6个不同的scale，而2表示水平翻转后会得到两个图。

（6）在这12次里面的每一次，对位置信息取最大，以Scale2为例，最后大小为6x9xC，就在这6x9个值中取最大。

（7）那么就会得到12个长度为C的向量，12个向量加起来取平均，得到一个长度为C的向量，然后求Top1或Top5，得到最后的结果。  


**验证集上的分类结果**  


其中coarse stride表示Δ=0Δ=0，fine stride表示Δ=0,1,2Δ=0,1,2。

（1）使用fine stride可以提升模型表现，但是提升不大，说明实际上offset-pooling在这里的作用不大。

（2）使用多scale，增加scale可以提升模型表现。

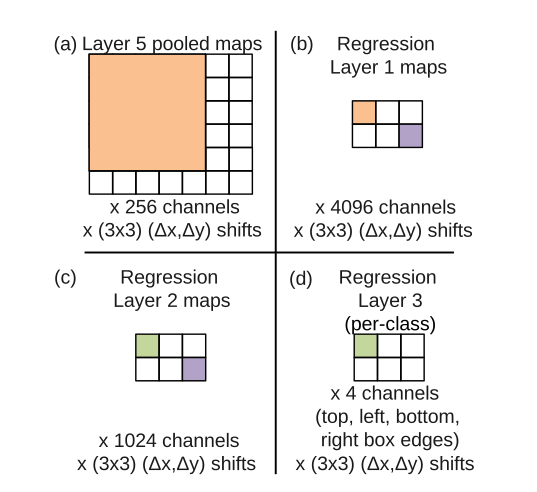
（3）最后多模型融合，又提升了表现

**定位任务**

前面提到的分类任务中，1到5层做特征提取网络，6到输出层作为分类网络，这个时候只要在5层（池化后的）后面接一个回归网络就可以来做定位了。

（1）训练的时候固定特征提取网络，根据box和真实box之间的l2损失进行训练。

（2）如下图所示，同样以scale2为例，第五层输出的是6\*7的图，通过回归网络的一系列卷积后，得到2\*3个位置信息（2\*3个box），4个channel表示box的四个边值（坐标）。

（3）回归层最后是1000个版本（类），下图中只是表示了一个类。  


**累积预测**

（1）对于回归网络得到的一系列bounding box，该论文不是通过传统的非极大值抑制，而是使用了累积预测的方法。

（2）首先对于每个scale计算出前k个类别，对每个类别计算出所有的bouding box。

（3）然后合并所有scale的bounding box得到集合BB，重复以下步骤

（4）(b∗1,b∗2)=argminb1≠b2Bmatch\_score(b1,b2)

（5）假如，match\_score(b1,b2)>tmatch\_score(b1,b2)>t，则停止

（6）否则，B←B \{b∗1,b∗2}∪box\_merge(b∗1,b∗2)

其中match\_score(b1,b2)match\_score(b1,b2)计算的是两个box中点的距离和交集区域的面积之和，当它大于某个阈值时算法停止box\_merge(b∗1,b∗2)box\_merge(b1∗,b2∗)计算的是两个box坐标的平均值。  
通过合并具有高置信度的box来得到最终预测。  
这种方法可以淘汰那些低置信度以及低连续（多个box相差很远）的类别，会更加鲁棒