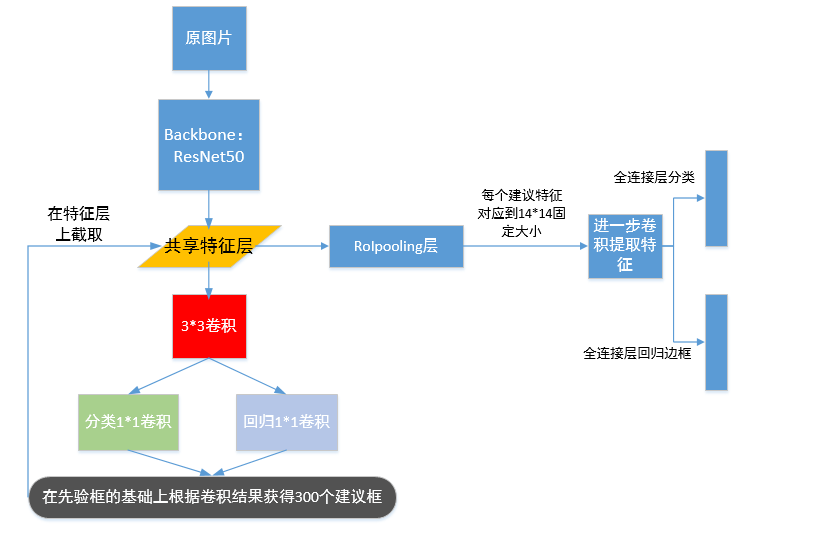
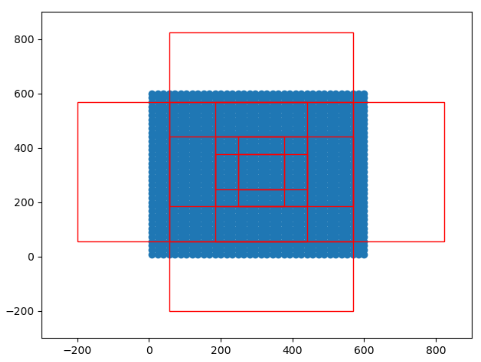
首先，看到以下是fasterRCNN的结构图，下面进行详细的过程梳理：



当我们输入一张图片，首先传达给backbone网络进行特征提取，backbone的选择不是唯一的，可以使用resnet50,101等等，其目的就是为了得到图像的大小较小，但却很深的特征层

举个例子，当输入图像的形状是600\*600\*3时，经过backbone会得到一个特征层形状为38\*38\*1024，注意，输入的图像应进行归一化处理

使用这个特征层进行3\*3卷积，滤波器个数为512，得到一个38\*38\*512的特征层，以38\*38为大小，获得38\*38=1444个锚点，每个锚点会有9个框，注意现在这个框是对于共享特征层而言的，下图表示在特征层中心位置这个锚点对应的9个先验框



RPN在第一张图中由3\*3卷积，两个1\*1卷积组成，其目的是进行粗略筛选出建议框，类似fastRCNN中使用图像处理算法提起建议框的作用

一般来说RPN默认做的一步就是把特征图上的所有先验框38\*38\*9个全部表示出来；之后，把38\*38\*512这个特征层上的每一个位置点（总共38\*38个）分别进行两个1\*1的卷积，第一个卷积有9个滤波器，代表了9个先验框中包含物体的概率（分类），第二个卷积有9\*4个滤波器，代表9个先验框，每一个框进行调整的参数（左上角坐标x,y和右下角坐标x,y（回归））

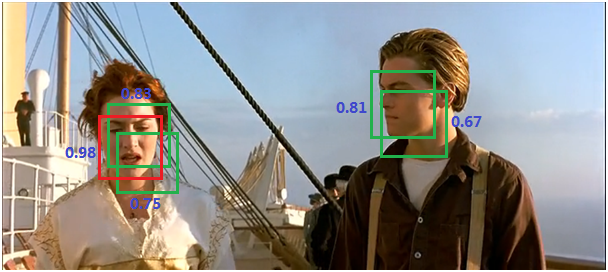
因此，目前RPN输出有两个数组，第一个是(38\*38,9)，代表先验框包含物体概率，第二个是（38\*38,9\*4），代表先验框调整的位置信息，注意，目前没有排除任何一个先验框，每个都参与了计算

下一步就需要进行删除没有用的先验框了，留下的叫建议框一般设置为300个，我们已知每个先验框都含有一个概率大小，我们将这个值与一个设置的置信度进行比较，包含物体的置信度一般取0.5，当概率小于0.5，删除这个先验框；下一步化简继续，考虑到框的重叠，进行一下非极大抑制，即NMS

插入一个说明：

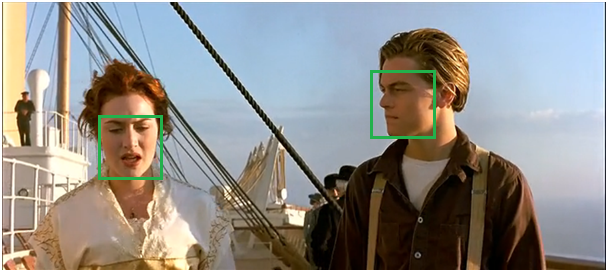
NMS：去除重复的边框

将所有候选框的得分进行排序，选中最高分及其所对应的BBox：



遍历其余的框，如果它和当前最高得分框的重叠面积大于一定的阈值，我们将其删除

从没有处理的框中继续选择一个得分最高的，重复上述过程最后得到：



现在继续；

然后把剩下的先验框结合（38\*38,9\*4）中的位置调整信息，获得初步调整位置的框，映射回到原图，删除超出边界的框，此时还没结束，将剩余的框按照置信度即它的概率大小进行排序，选择最高的前300个，目前便得到了初步筛选出的300个建议框，还是注意，这300个建议框的大小是针对共享特征层而言的；

将这300个建议框与共享特征层求交集，获得300个小特征层，即形状为（300,m\*n,1024），即300个m\*n大小，深度或者通道数为1024的特征层集合，m,n代表了各个建议框的宽和高，每个建议框的宽和高都是不一样的，但是我们可以使用RoIpooling层进行池化，获得300个固定大小为14\*14，1024的特征层

此处插入RoIpooling的过程：将一个宽和高为m\*n的特征层（深度为x），等分成设置的（14\*14份，通道数x不变，为了满足等分时取整，需要进行量化），然后在每个小区域内进行池化即可得到（14\*14,x）的固定特征层

举个例子：

1)Conv layers使用的是VGG16，feat\_stride=32(即表示，经过网络层后图片缩小为原图的1/32),原图800\*800,最后一层特征图feature map大小:25\*25

2)假定原图中有一region proposal，大小为665\*665，这样，映射到特征图中的大小：665/32=20.78,即20.78\*20.78，如果你看过Caffe的Roi Pooling的C++源码，在计算的时候会进行取整操作，于是，进行所谓的第一次量化，即映射的特征图大小为20\*20

3)假定pooled\_w=7,pooled\_h=7,即pooling后固定成7\*7大小的特征图，所以，将上面在 feature map上映射的20\*20的 region  proposal划分成49个同等大小的小区域，每个小区域的大小20/7=2.86,即2.86\*2.86，此时，进行第二次量化，故小区域大小变成2\*2

4)每个2\*2的小区域里，取出其中最大的像素值，作为这一个区域的‘代表’，这样，49个小区域就输出49个像素值，组成7\*7大小的feature map

现在得到了固定大小的特征层，且总计300个，因此整体看来这个张量形状为（300,14,14,1024），在后面计算上，可以并行地计算每一个特征层，我们现在为了方便理解，就只从300个特征层中选出1个进行计算过程上的浏览：

现在选出了一个特征层：（14,14,1024），对这个特征层进行卷积，使用到了残差网络50的最后一个stage，即conv\_block->identity\_block-> identity\_block，输出特征层形状为（7,7,2048），进行一次平均池化得到特征层（1,1,2048），将这个特征层reshape展开成一个向量，此时便可以直接进行全连接层处理，这里的全连接层计算也分成了2个分支，一个是预测类别，一个回归得到边框进一步调整的4个参数信息

对于分类分支，如果有20类物体，就要对应21个神经元，多出的这一个属于背景类别

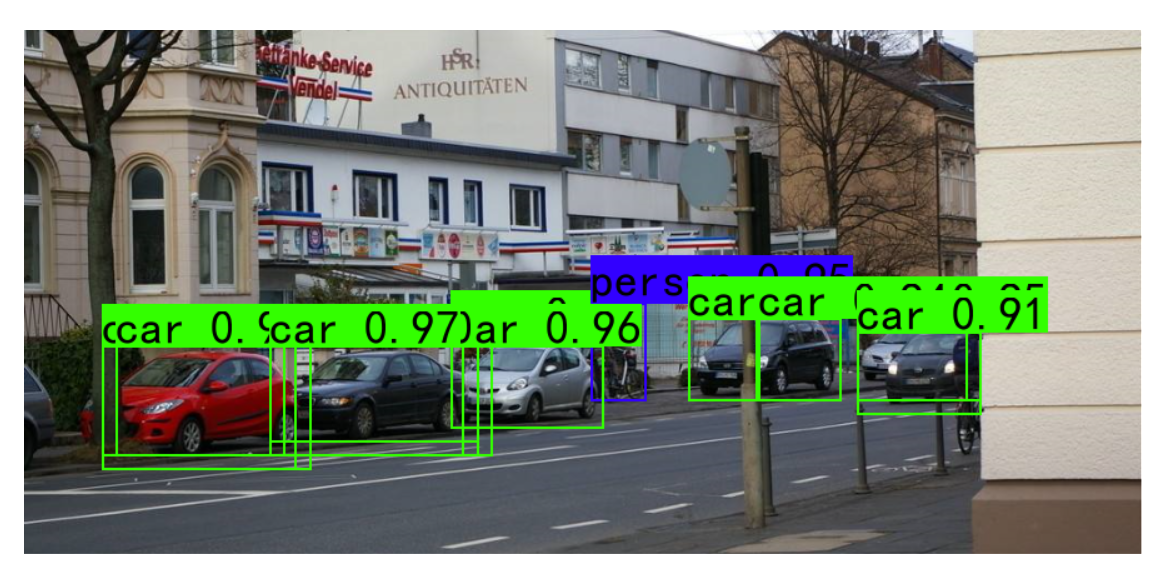
对于回归分支，就有20\*4个神经元，分别对应每一类下边框调整的4个位置参数

我们绘制边框时，只需要索引分类分支中得到概率最大的那个类别，选出这个类别对应的4个参数信息就可以了，这里注意，背景类别不用绘制

到这里还没有结束，经过一波计算，我们得到了num个（num<300, 因为背景不算）经过精细调整的边框，以及对应的num个物体

对剩下num个物体，进行一次置信度筛选，因为它对应的概率可能仅为0.6，这会导致结果判断出现错误，所以从num个物体中选择置信度大于0.9的物体，到这里还没结束

因为实验证明，到目前还是避免不了有边框重叠现象，比如：



所以对此，再次进行NMS，得到最终结果：

