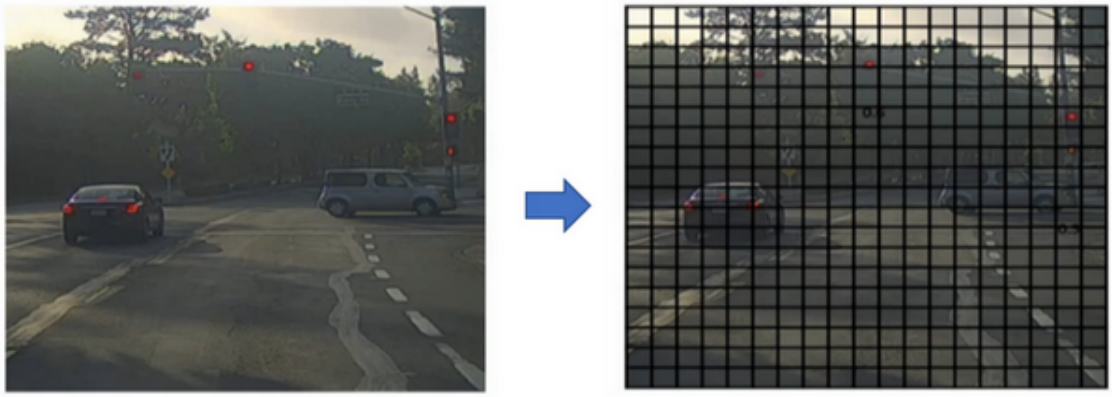
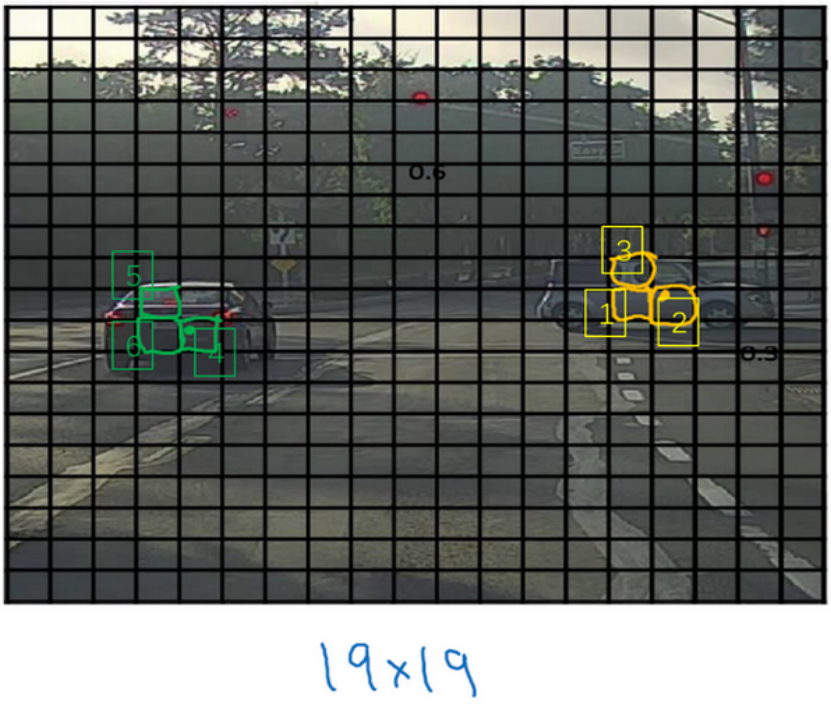
3.7 非极大值抑制（Non-max suppression）

到目前为止你们学到的对象检测中的一个问题是，你的算法可能对同一个对象做出多次检测，所以算法不是对某个对象检测出一次，而是检测出多次。**非极大值抑制这个方法可以确保你的算法对每个对象只检测一次，**我们讲一个例子。



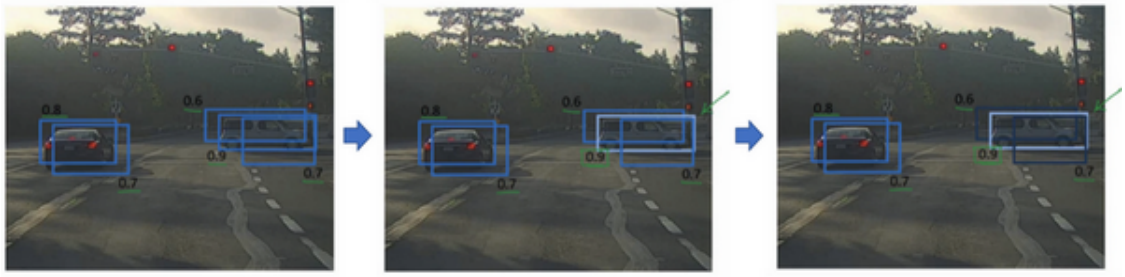
假设你需要在这张图片里检测行人和汽车，你可能会在上面放个 19×19 网格，理论上这辆车只有一个中点，所以它应该只被分配到一个格子里，左边的车子也只有一个中点，所以理论上应该只有一个格子做出有车的预测。



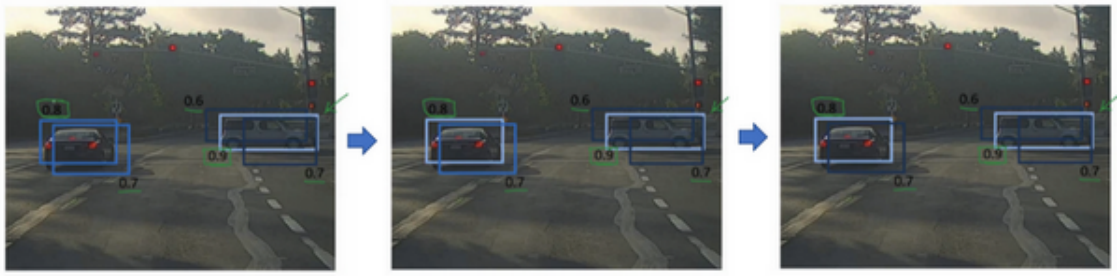
实践中当你运行对象分类和定位算法时，对于每个格子都运行一次，所以这个格子（编号 1）可能会认为这辆车中点应该在格子内部，这几个格子（编号 2、3）也会这么认为。对

于左边的车子也一样，所以不仅仅是这个格子，如果这是你们以前见过的图像，不仅这个格（编号 4）子会认为它里面有车，也许这个格子（编号 5）和这个格子（编号 6）也会，也许其他格子也会这么认为，觉得它们格子内有车。

我们分步介绍一下非极大抑制是怎么起效的，**因为你要在 361 个格子上都运行一次图像检测和定位算法，那么可能很多格子都会举手说我的p_c，我这个格子里有车的概率很高，而不是 361 个格子中仅有两个格子会报告它们检测出一个对象。所以当你运行算法的时候，最后可能会对同一个对象做出多次检测，所以非极大值抑制做的就是清理这些检测结果。这样一辆车只检测一次，而不是每辆车都触发多次检测。**



所以具体上，这个算法做的是，首先看看每次报告每个检测结果相关的概率p_c，在本周的编程练习中有更多细节，实际上是p_c乘以c_1、c_2或c_3。现在我们就说，这个p_c检测概率，首先看概率最大的那个，这个例子（右边车辆）中是 0.9，然后就说这是最可靠的检测，所以我们就用高亮标记，就说我这里找到了一辆车。**这么做之后，非极大值抑制就会逐一审视剩下的矩形，所有和这个最大的边框有很高交并比，高度重叠的其他边界框，那么这些输出就会被抑制。所以这两个矩形p_c分别是 0.6 和 0.7，这两个矩形和淡蓝色矩形重叠程度很高，所以会被抑制，变暗，表示它们被抑制了。**

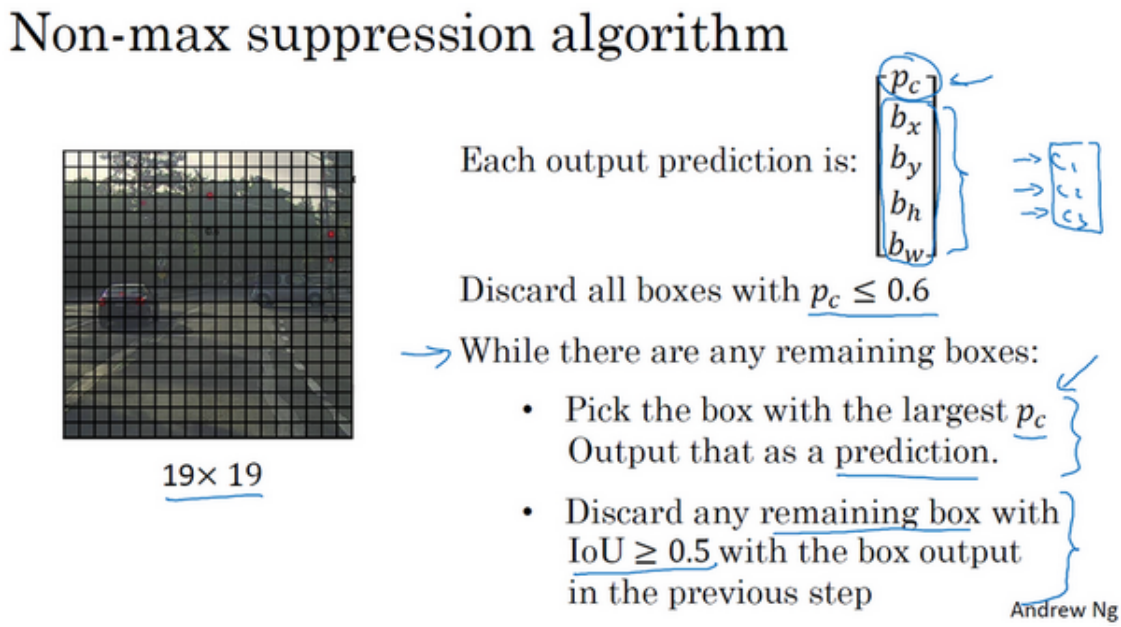


接下来，逐一审视剩下的矩形，找出概率最高，p_c最高的一个，在这种情况下是 0.8，我们就认为这里检测出一辆车（左边车辆），然后非极大值抑制算法就会去掉其他 IoU 值很高的矩形。所以现在每个矩形都会被高亮显示或者变暗，如果你直接抛弃变暗的矩形，那就剩下高亮显示的那些，这就是最后得到的两个预测结果。

所以这就是非极大值抑制，非最大值意味着你只输出概率最大的分类结果，但抑制很接

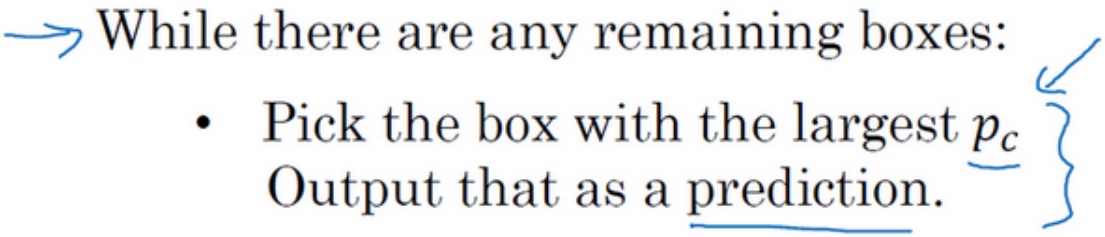
近，但不是最大的其他预测结果，所以这方法叫做非极大值抑制。

我们来看看算法的细节，首先这个 19×19 网格上执行一下算法，你会得到 19×19×8 的输出尺寸。不过对于这个例子来说，我们简化一下，就说你只做汽车检测，我们就去掉c_1、c_2和c_3，然后假设这条线对于 19×19 的每一个输出，对于 361 个格子的每个输出，你会得到这样的输出预测，就是格子中有对象的概率（p_c），然后是边界框参数（b_x、b_y、b_h和b_w）。如果你只检测一种对象，那么就没有c_1、c_2和c_3这些预测分量。多个对象处于同一个格子中的情况，我会放到编程练习中，你们可以在本周末之前做做。



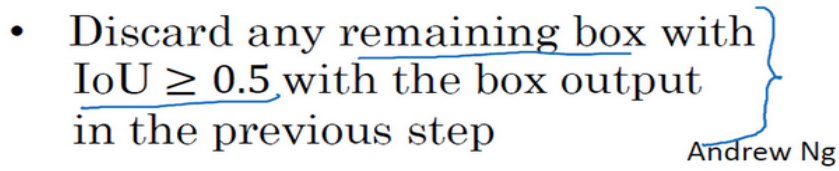
现在要实现非极大值抑制，你可以做的第一件事是，去掉所有边界框，我们就将所有的预测值，所有的边界框p_c小于或等于某个阈值，比如$p_c \leq 0.6$的边界框去掉。

我们就这样说，除非算法认为这里存在对象的概率至少有 0.6，否则就抛弃，所以这就抛弃了所有概率比较低的输出边界框。所以思路是对于这 361 个位置，你输出一个边界框，还有那个最好边界框所对应的概率，所以我们只是抛弃所有低概率的边界框。



接下来剩下的边界框，没有抛弃没有处理过的，你就一直选择概率p_c最高的边界框，然后把它输出成预测结果，这个过程就是上一张幻灯片，取一个边界框，让它高亮显示，这样

你就可以确定输出做出有一辆车的预测。



接下来去掉所有剩下的边界框，任何没有达到输出标准的边界框，之前没有抛弃的边界框，把这些和输出边界框有高重叠面积和上一步输出边界框有很高交并比的边界框全部抛弃。所以 **while** 循环的第二步是上一张幻灯片变暗的那些边界框，和高亮标记的边界重叠面积很高的那些边界框抛弃掉。在还有剩下边界框的时候，一直这么做，把没处理的都处理完，直到每个边界框都判断过了，它们有的作为输出结果，剩下的会被抛弃，它们和输出结果重叠面积太高，和输出结果交并比太高，和你刚刚输出这里存在对象结果的重叠程度过高。

在这张幻灯片中，我只介绍了算法检测单个对象的情况，如果你尝试同时检测三个对象，比如说行人、汽车、摩托，那么输出向量就会有三个额外的分量。事实证明，正确的做法是独立进行三次非极大值抑制，对每个输出类别都做一次，但这个细节就留给本周的编程练习吧，其中你可以自己尝试实现，我们可以自己试试在多个对象类别检测时做非极大值抑制。

这就是非极大值抑制，如果你能实现我们说过的对象检测算法，你其实可以得到相当不错的结果。但结束我们对 **YOLO** 算法的介绍之前，最后我还有一个细节想给大家分享，可以进一步改善算法效果，就是 **anchor box** 的思路，我们下一个视频再介绍。