最近在看RCNN和微软的SPP-net，其中涉及到Non-Maximum Suppression，论文中没具体展开，我就研究下了代码，这里做一个简单的总结，听这个名字感觉是一个很高深的算法，其实很简单，就是把找出score比较region，其中需要考虑不同region的一个重叠问题。

假设从一个图像中得到了2000region proposals，通过在RCNN和SPP-net之后我们会得到2000*4096的一个特征矩阵，然后通过N的SVM来判断每一个region属于N的类的scores。其中，SVM的权重矩阵大小为4096*N，最后得到2000*N的一个score矩阵（其中，N为类别的数量）。

Non-Maximum Suppression就是需要根据score矩阵和region的坐标信息，从中找到置信度比较高的bounding box。首先，NMS计算出每一个bounding box的面积，然后根据score进行排序，把score最大的bounding box作为队列中。接下来，计算其余bounding box与当前最大score与box的IoU，去除IoU大于设定的阈值的bounding box。然后重复上面的过程，直至候选bounding box为空。最终，检测了bounding box的过程中有两个阈值，一个就是IoU，另一个是在过程之后，从候选的bounding box中剔除score小于阈值的bounding box。需要注意的是：Non-Maximum Suppression一次处理一个类别，如果有N个类别，Non-Maximum Suppression就需要执行N次。

源代码：

```
function pick = nms(boxes, overlap)
% top = nms(boxes, overlap)
% Non-maximum suppression. (FAST VERSION)
% Greedily select high-scoring detections and skip detections
% that are significantly covered by a previously selected
% detection.
%
% NOTE: This is adapted from Pedro Felzenszwalb's version (nms.m),
% but an inner loop has been eliminated to significantly speed it
% up in the case of a large number of boxes


% Copyright (C) 2011-12 by Tomasz Malisiewicz
% All rights reserved.
%
% This file is part of the Exemplar-SVM library and is made
% available under the terms of the MIT license (see COPYING file).
% Project homepage: https://github.com/quantombone/exemplarsvm


if isempty(boxes)
  pick = [];
  return;
end


x1 = boxes(:,1);
y1 = boxes(:,2);
x2 = boxes(:,3);
y2 = boxes(:,4);
s = boxes(:,end);


area = (x2-x1+1) .* (y2-y1+1);    %计算出每一个bounding box的面积
[vals, I] = sort(s);                %根据score递增排序


pick = s*0;
counter = 1;
while ~isempty(I)
  last = length(I);
  i = I(last);
  pick(counter) = i;              %选择score最大bounding box加入到候选队列
  counter = counter + 1;

  xx1 = max(x1(i), x1(I(1:last-1)));
  yy1 = max(y1(i), y1(I(1:last-1)));
  xx2 = min(x2(i), x2(I(1:last-1)));
  yy2 = min(y2(i), y2(I(1:last-1)));

  w = max(0.0, xx2-xx1+1);
  h = max(0.0, yy2-yy1+1);

  inter = w.*h;        %计算出每一bounding box与当前score最大的box的交集面积
  o = inter ./ (area(i) + area(I(1:last-1)) - inter);  %IoU (intersection-over-union)

  I = I(find(o<=overlap));  %找出IoU小于overlap阈值的index
end


pick = pick(1:(counter-1));
```