# 目标识别算法

目标识别：给定的场景图像中，找到目标所在的位置，并对有效目标进行识别。

衡量指标：对象识别精度(mAP)、识别效率(FPS)、对象位置准确性(MES)。

目前的常用方法：

RCNN,SPP-Net,Fast-RCNN,Faster-RCNN,R-FCN,YOLO,SSD,Mask-RCNN, PVANET

## 1.背景概述

### 1.1 2-stage 目标检测与识别

分两个步骤完成目标的检测与识别，典型的网络为Fast-RCNN,Faster-RCNN,R-FCN等在Faster-RCNN的基础上修改的网络。

1. 候选区域的生成

**传统方法：**Selective search、edgeBox 更多的依赖于对象的纹理和几何、颜色等特性，生成候选区域，生成候选区域多，速度慢。

**深度学习：**RPN网络，利用anchors的机制，对区域进行前景背景的分类，同时利用回归初步生成候选区域。

**这部分为了提高生成候选区域的速度，让ROI共享更多的计算量，即先让图像通过共享的CNN网络，仅对抽取的特征图进行ROI特征的提取。共享CNN网络结构的复杂程度可以对最终的识别与定位造成影响。**

1. 候选区域的分类与回归

对(1)中生成的候选区域，通过提取共享卷积层的ROI特征，并送入到后续识别网络，完成具体目标的识别与位置精挑。

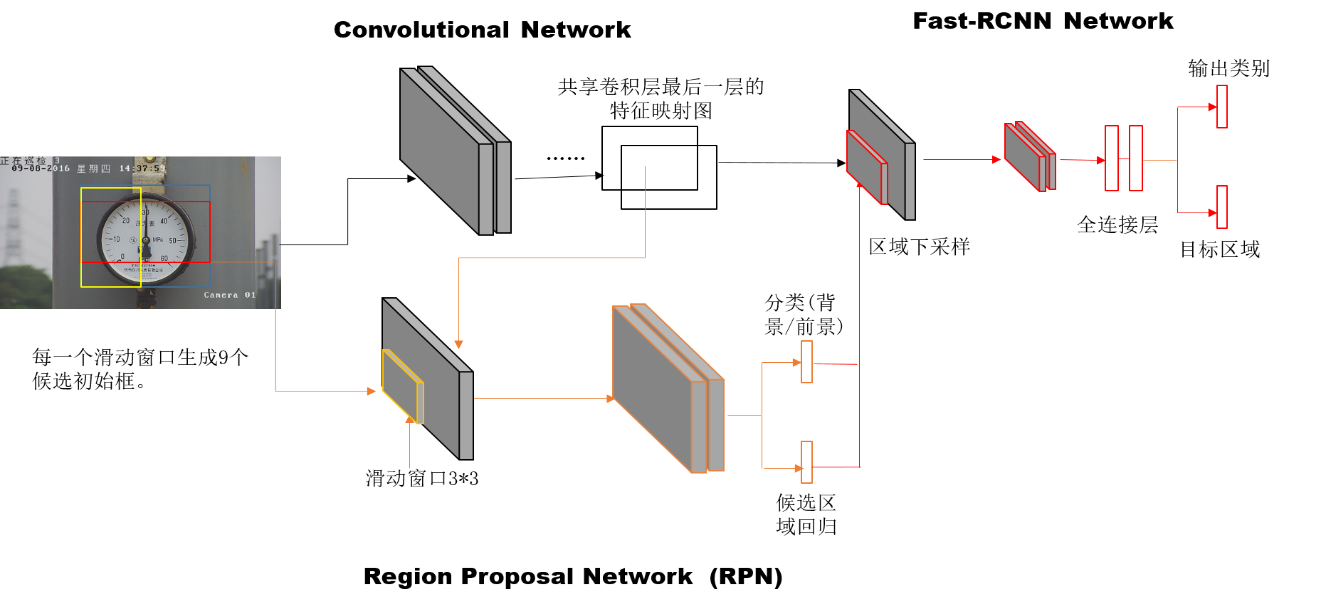
### 1.2 1-stage 目标检测与识别

通过端对端的网络一步回归到物体的具体位置，这类算法代表性的为SSD和YOLO以及各自的改进版本。

## 2.算法具体分析

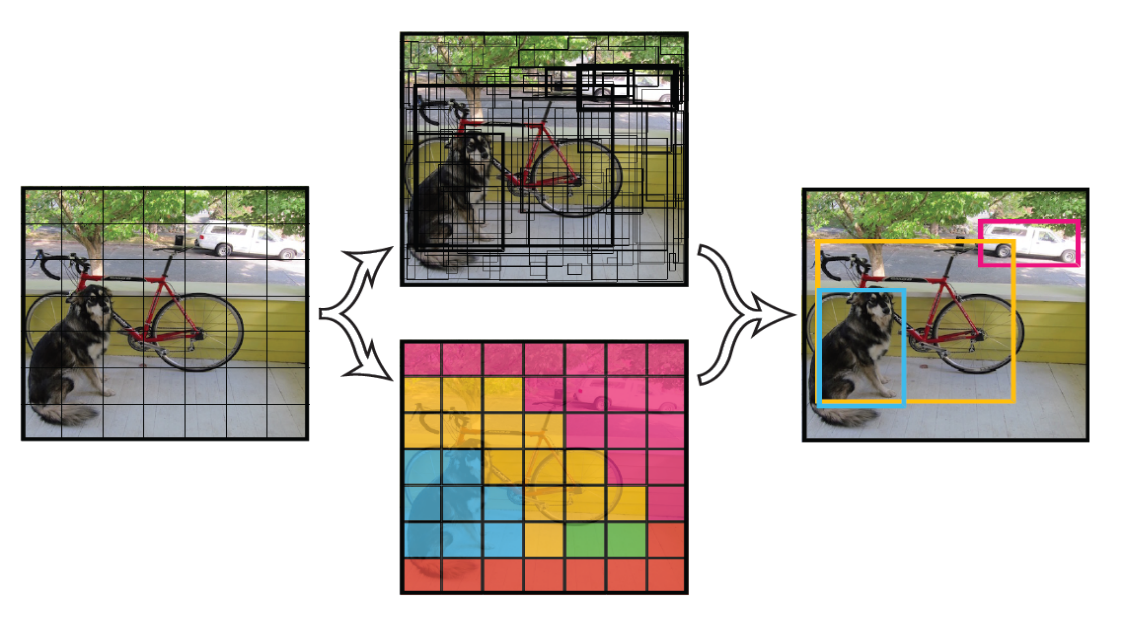
### 2.1 Faster-RCNN

Faster-RCNN中，目标识别分为两个阶段，第一个阶段为RPN网络，图像先送入共享卷积层，通过提取某一层(如conv)的特征图，并用以预测目标前景背景的分类概率和可能出现目标的候选区域，损失函数采取了分类和回归的多任务损失函数。在第二个阶段，一阶段生成的候选区域(300个)通过局部下采样层对第一阶段的特征图进行区域特征的提取，从而保证后续的网络输入特征图尺寸一致，后续的网络完成对目标类别的具体分类与位置精挑，第二阶段也为多任务损失函数，具体类别的损失和目标位置的回归损失。



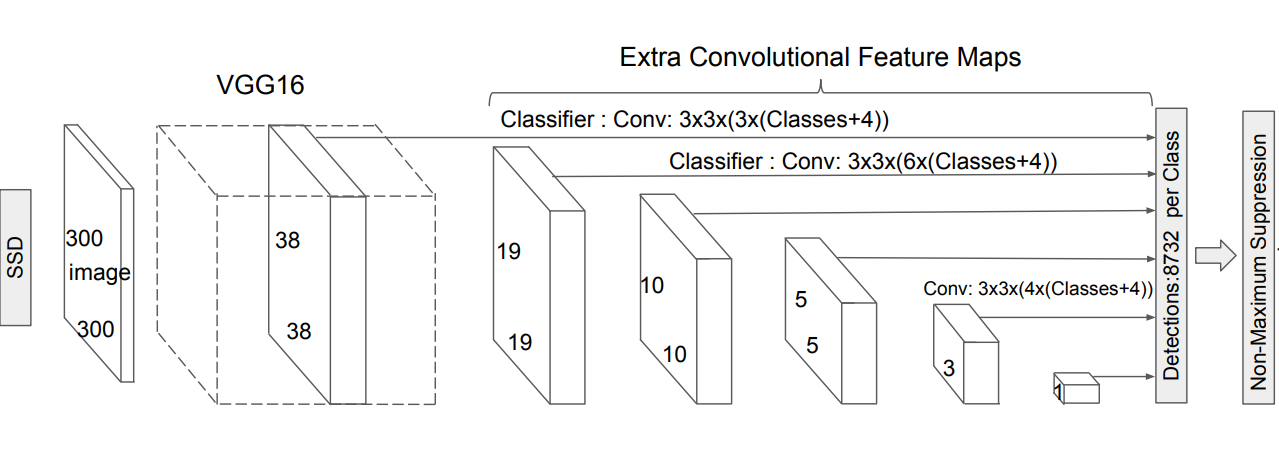
### 2.2 Yolo

通过对图像分为S\*S的网格，每个网格生成Num个候选Box，每个Box输出一个五维向量(x,y,w,h,confidence),同时每个网格输出C个类别概率，类别概率与每个box的confidence的乘积作为该box内包含目标的概率，从而完成目标的识别与定位。



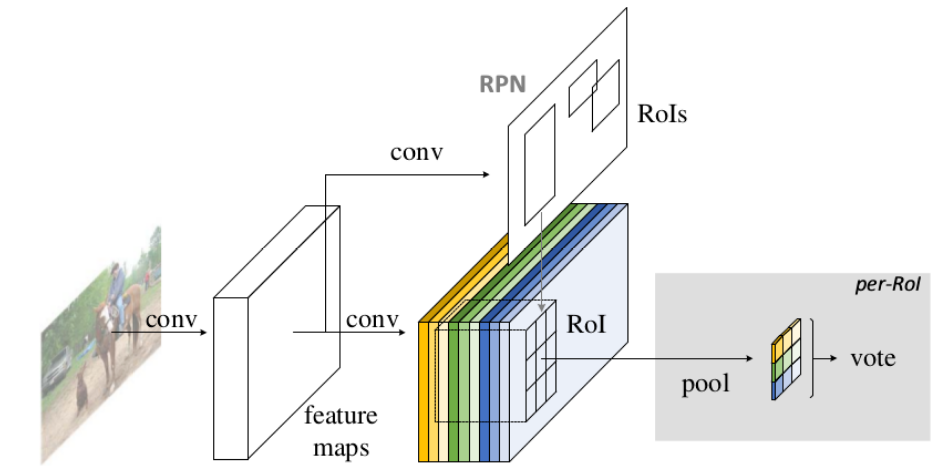
### 2.3 SSD

SSD 是基于一个前向传播的CNN网络，产生一系列固定大小bounding boxes，以及每一个box中包含物体实例的可能性，即score之后，进行一个[非极大值抑制](http://www.pyimagesearch.com/2014/11/17/non-maximum-suppression-object-detection-python/" \t "_blank)得到最终的predictions，在特征提取的过程中，采取了多特征图的预测。



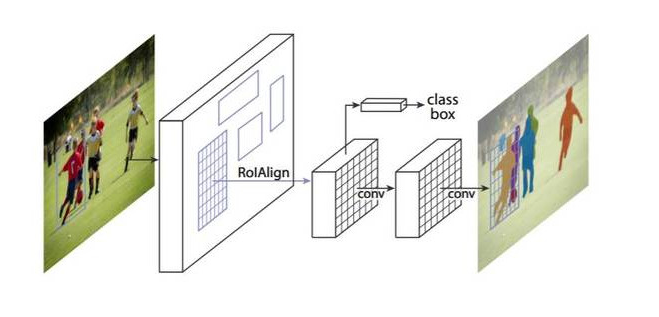
### 2.4 R-FCN

R-FCN网络在Faster-RCNN的基础上做了小幅度的修改，由于RPN生成的所有候选区域都要通过局部下采样层后输入到共同的网络进行后续识别，这种方法每个候选区域之间并没有共享相同的计算，从而候选区域的数量对最终的速度造成了影响，因此R-FCN网络提出，在FCN的顶层，添加了一个卷积层，整个网络中所有可学习的层次都是可卷积的并且共享的，在精度达到了Faster-RCNN效果的同时，加快了检测速度。



### 2.5 MASK-RCNN

Mask R-CNN是Faster R-CNN的扩展形式，在每个ROI上加一个用于预测分割掩码的分层，称为掩码层，使该层并行于已有边界层和分类层，同时候采用RoIAlign替代ROIPooling，更加准确的映射出原始图像与特征图的映射关系，在COCO上取得了与Faster-RCNN相当的结果。



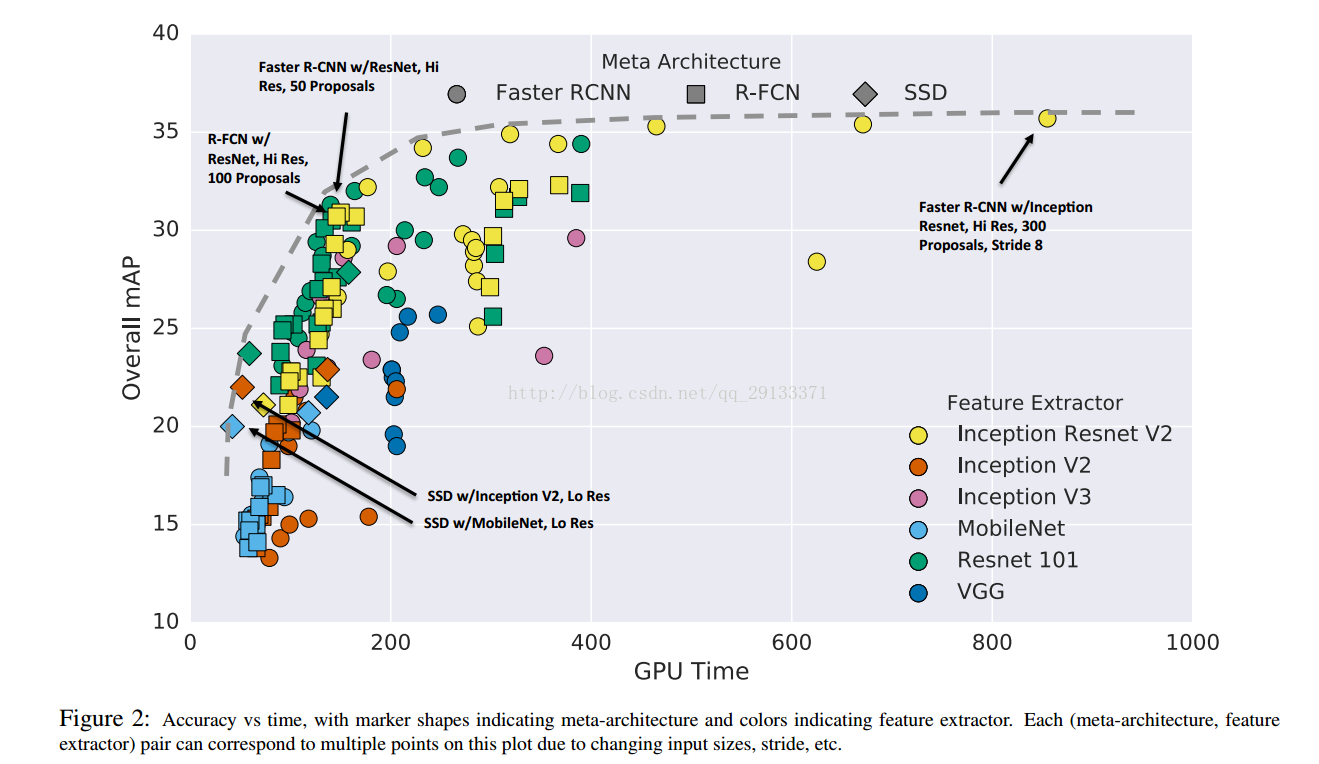
## 3.结果分析

### 3.1 硬件

电脑性能: 32G内存，GTX Titan X 。

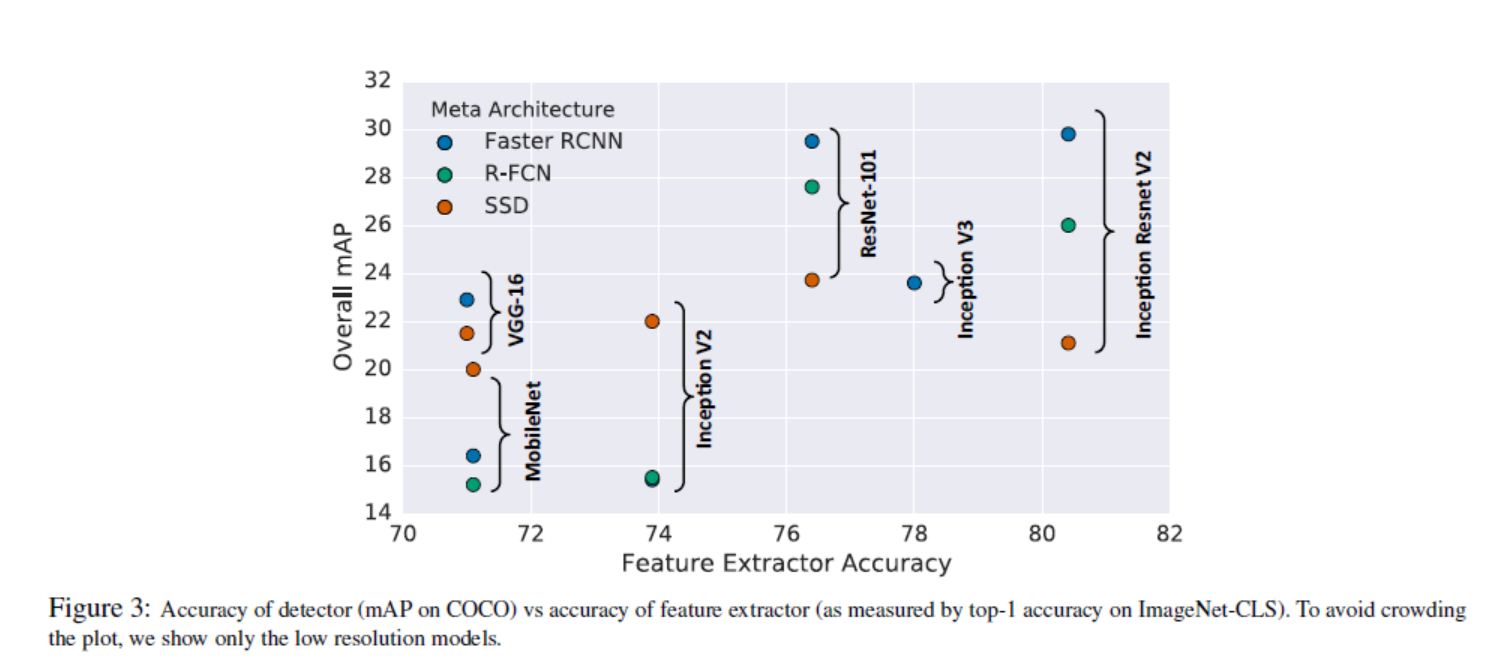
### 3.2 准确率以及计算时间

SSD，R-FCN在速度上要远远超过Faster R-CNN，但是在精度上Faster R-CNN领先，R-FCN紧随其后。但是Faster R-CNN可以通过设置region proposal来降低处理速度，如Faster R-CNN w/ResNet 50 proposal的速度比300 proposal的速度要快很多。



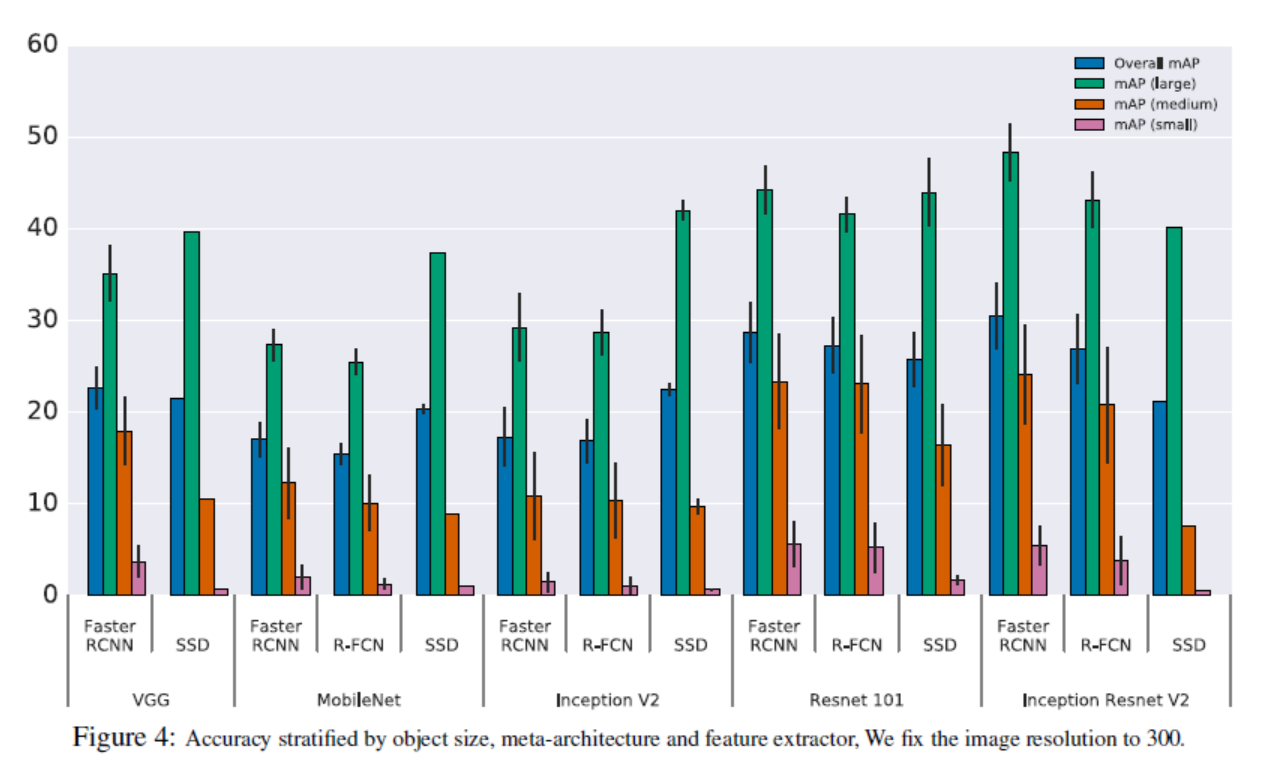
### 3.3 特征网络的影响

图像的对于SSD为300\*300,Faster和R-FCN min(w,h)为300，网络结果较为复杂的Residual和inception的特征网络会得到更好的mAP.



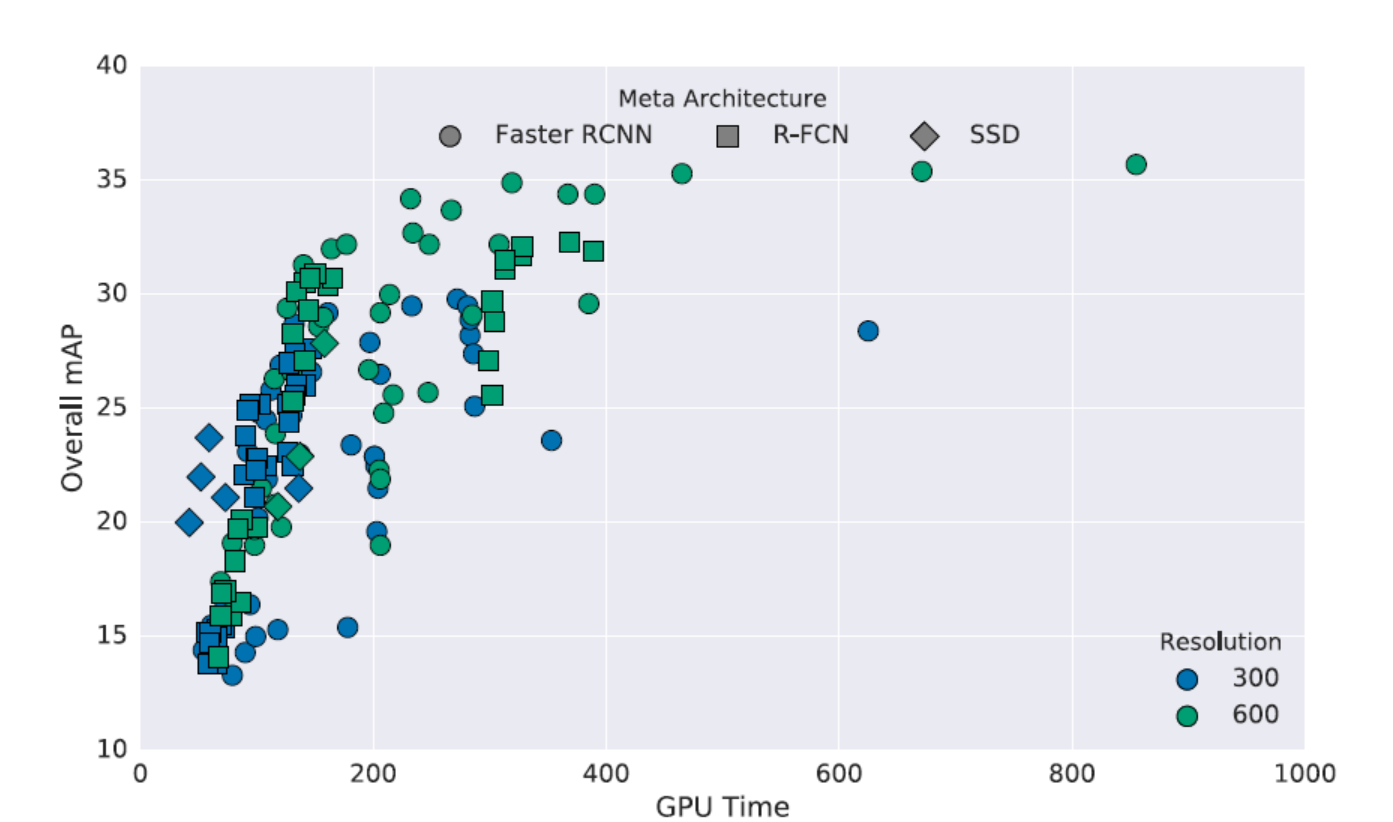
### 3.4 目标尺寸的影响

基本所有的网络在大目标的检测上会比小目标的效果要好，相比于SSD，Faster-RCNN和R-FCN在小目标的检测上效果也很出色。



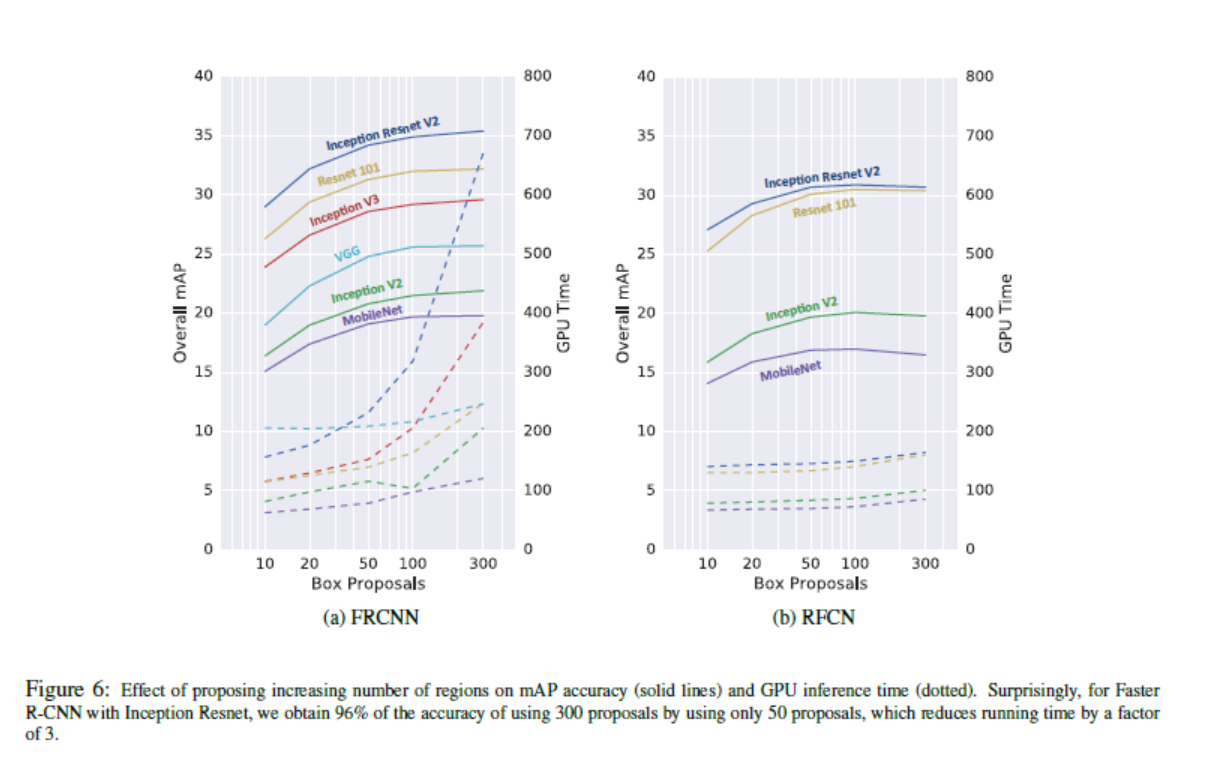
### 3.5 图像尺寸

图像尺寸较大时，相对的小目标识别问题也有了一定的解决，因此得到了更高的精确度，但是消耗的时间也会增加。



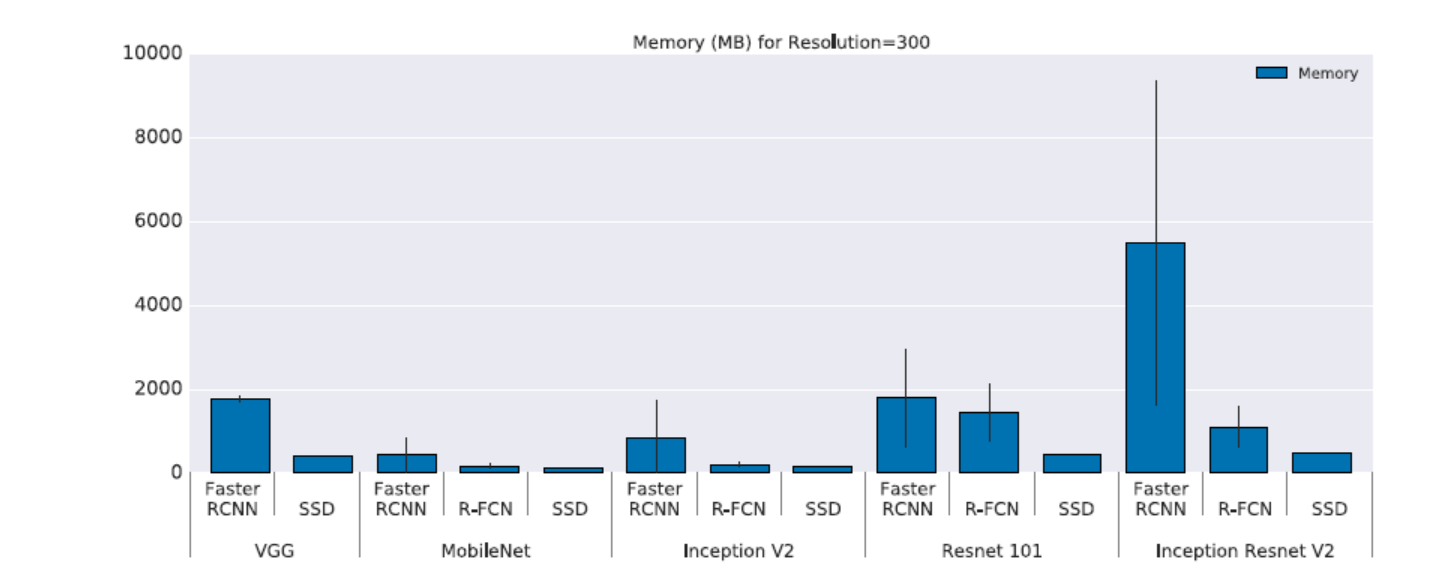
### 3.6 候选区域的个数

R-FCN和Faster-RCNN的默认候选框个数为300，但是适当的减少候选区域，并不会对最终的mAP 有很大的影响。50个候选区域的Faster-RCNN的inceptionResnetV2精度从35.4—39.8，但是fast网络速度提高3倍。R-FCN的提升效果没有Faster-RCNN明显，这是由于R-FCN共享了更多的卷积层计算。



### 3.7 内存分析

网络的复杂度越大，占用的显存越大，纵向对比中，Faster-RCNN占用更大的内存。



### 3.8 其他结果分析

其他的结果由于作者的实现环境和平台包括深度学习框架使用的不同，都会有点差异。列出的为作者论文中写出的结果。

|  |  |  |  |
| --- | --- | --- | --- |
| Detect Frameworks | Train(VOC) | mAP | FPS |
| Fast R-CNN | 2007+2012 | 70.0 | 0.5 |
| Faster R-CNN VGG-16 | 2007+2012 | 73.2 | 7 |
| Faster R-CNN ResNet | 2007+2012 | 76.4 | 5 |
| YOLO | 2007+2012 | 63.4 | 45 |
| SSD300 | 2007+2012 | 74.3 | 46 |
| SSD500 | 2007+2012 | 76.8 | 19 |
| YOLOv2 288X288 | 2007+2012 | 69.0 | 91 |
| YOLOv2 352X352 | 2007+2012 | 73.7 | 81 |
| YOLOv2 416X416 | 2007+2012 | 76.8 | 67 |
| YOLOv2 480X480 | 2007+2012 | 77.8 | 59 |
| YOLOv2 544X544 | 2007+2012 | 78.6 | 40 |

