目标检测与图像分割算法 - Dec 15'th, 2018

C-RNN, Fast R-CNN, Mask R-CNN, Faster R-CNN, YOLO etc.

Agenda

- 概述
- 目标检测算法
  - R-CNN
  - Fast R-CNN
  - Mask R-CNN
  - Faster R-CNN
  - YOLO
- 图像分割应用

Reference

- `https://arxiv.org/pdf/1311.2524.pdf`
- `https://arxiv.org/pdf/1504.08083.pdf`
- `https://arxiv.org/pdf/1506.01497.pdf`
- `https://arxiv.org/pdf/1506.02640v5.pdf`
- `http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf`

Introduction

- 目标检测的应用：
  - 自动驾驶领域
  - （工业）机器人领域
  - 姿态估计
  - 室外/内监控
- 分类算法 vs 目标检测算法
  - 目标检测算法不仅仅在做分类的事情，而且在检测的对象周围画出一个边界，并且目标检测算法事先是不知道在这个场景中有多少感兴趣区域（RoI），这是目标检测比传统的分类算法难的地方。
  - 那么目标检测算法除了能够确定目标的类别还要能够确定目标的位置以及如何确定目标的大小。

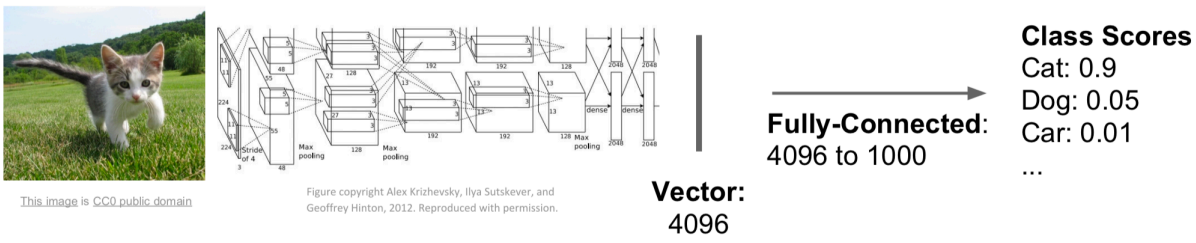

图 1：分类问题

- 在计算机视觉领域，往往讲目标检测定为最 top 的目标，因为 Segmentation → Localization→ Detec-

tion

- 下面举出几个比较常见的计算机视觉的任务

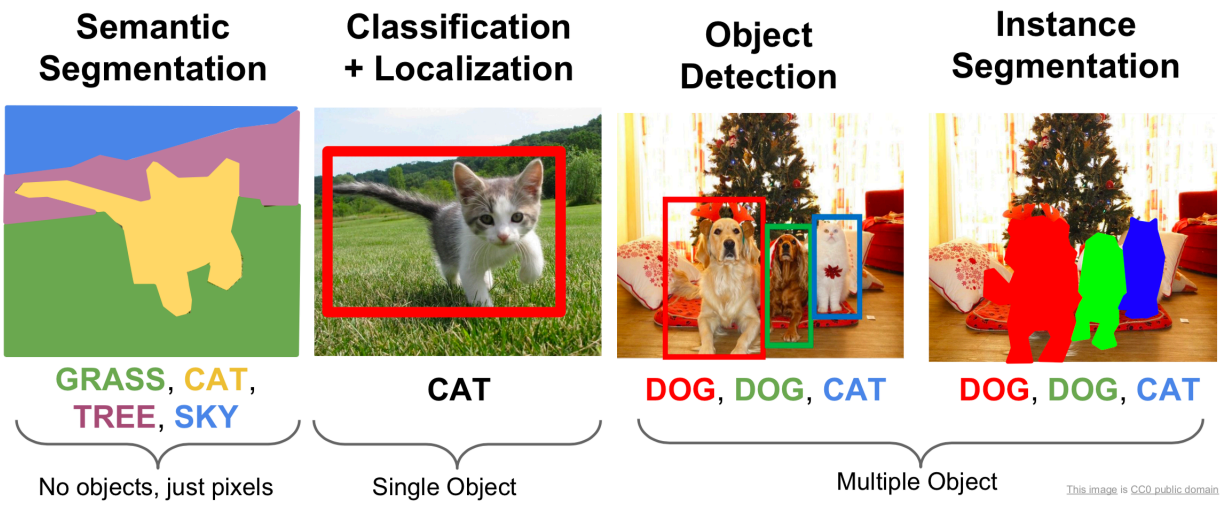

图 2：计算机视觉的其他任务

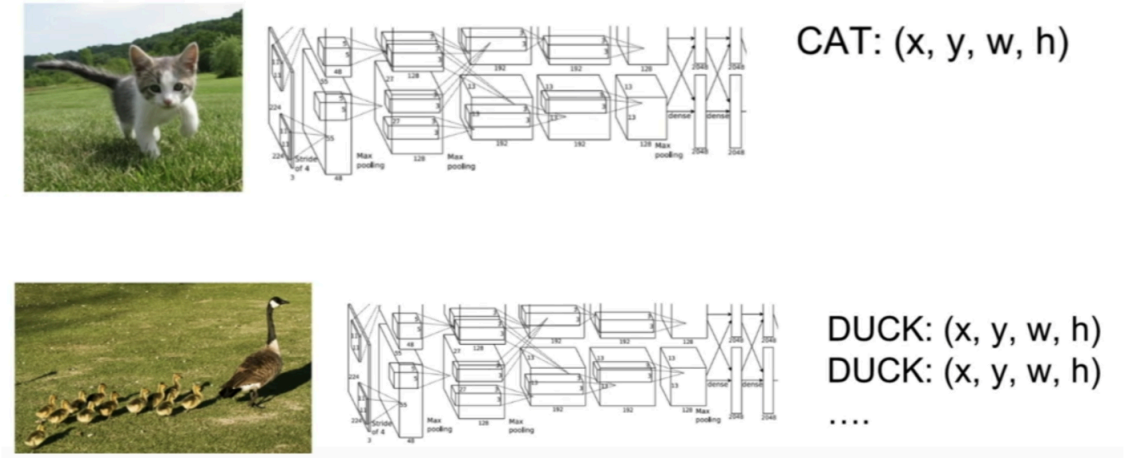

图 3：目标检测实例

- 为什么不能使用 CNN 来做目标检测?
  - 通过构建标准卷积神经网络后接一个全连接层无法解决此问题的主要原因：
    - 输出层的长度是可变的，不是常数，这是因为感兴趣对象的出现次数是不固定的。
  - 解决此问题的一种简单方法是从图像中获取不同的感兴趣区域，并使用 CNN 对该区域存在的对象进行分类。
  - 该方法的问题在于感兴趣的对象可能在图像内具有不同的空间位置和大小。因此必须选择大量的区域，这可能会带来计算爆炸问题。
  - 目前已经开发出诸如 R-CNN，YOLO 等算法来发现这些感兴趣对象并快速找到它们。

R-CNN

- 为了绕过“选择大量区域”的问题，Ross Girshick., 提出了一种方法，使用选择性搜索从图像中提取 2000 个区域，并将其称为区域提议 (region proposal)。因此，现在可以只使用 2000 个区域，而不是尝试对大量区域进行分类。使用下面写出的选择性搜索算法生成 2000 个区域提议 (region proposal)。
  1. Generate initial sub-segmentation, we generate many candidate regions
  2. Use greedy algorithm to recursively combine similar regions into larger ones

3. Use the generated regions to produce the final candidate region proposals

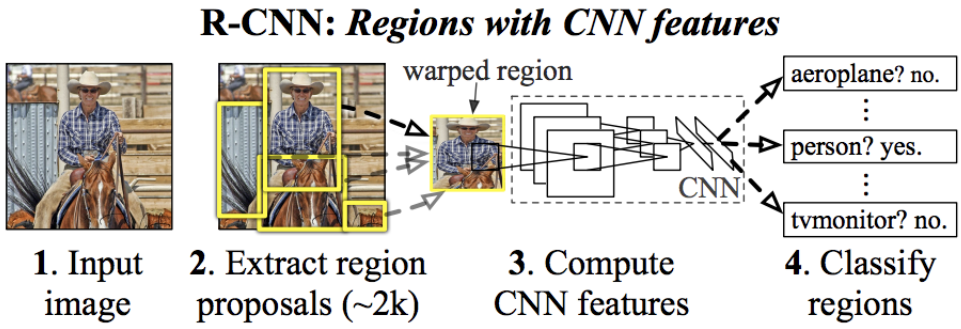

图 4：R-CNN Procedure

- 将这 2000 个候选区域提议扭曲成正方形喂给卷积神经网络，该网络产生 4096 维特征向量作为输出。CNN 在这里充当特征提取器，输出到 dense 层由从图像提取的特征组成，并且提取的特征被输送到 SVM 分类器以对该候选区域提议（region proposal）内的对象进行分类。除了预测区域提议中存在的对象的之外，该算法还预测四个值，这些值是偏移值用来增加边界的精度。例如，给定一个区域提议，该算法可以预测一个人的存在，但该区域提案中该人的面部可以减少一半。因此，偏移值有助于调整区域提议的边界框。

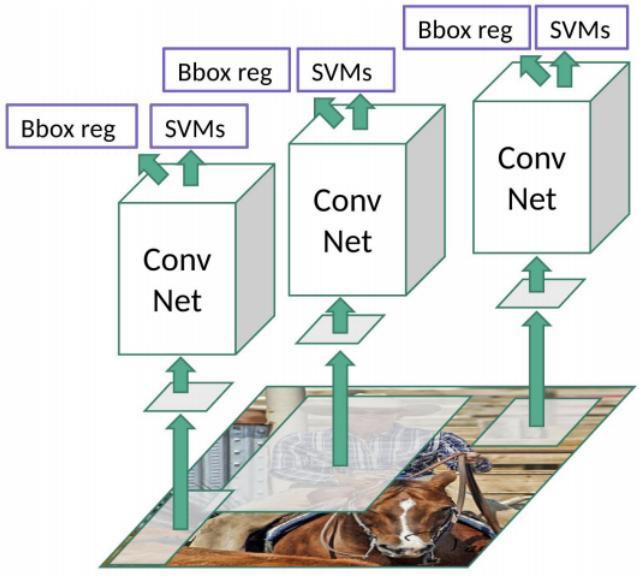

图 5：R-CNN Architecture

- R-CNN 存在的问题
  - 当你对每张图片的 2000 个区域提议进行分类，仍然需要耗费大量的训练时间.
  - 它不能实时实现，因为每个测试图像大约需要 47 秒。
  - 选择性搜索算法是固定算法，不是学习算法，因此，在那个阶段（第一阶段）没有学习。这可能导致产生不良候选区域提议。

Fast R-CNN

- R-CNN 的作者解决了 R-CNN 物体检测算法的一些缺点，该算法被称为 Fast R-CNN。
- 该方法类似于 R-CNN 算法。但是，我们不是将区域提议提供给 CNN，而是将输入图像提供给 CNN 以生成卷积特征图。从卷积特征图中，我们识别出建议的区域并将它们扭曲成正方形，并且通过使用

RoI 池化层，将它们重新整形 (Reshape) 为固定大小，以便可以将其馈送到全连接层中。从 RoI 特征向量中，我们使用 softmax 层来预测建议区域的类别以及边界框的偏移值。

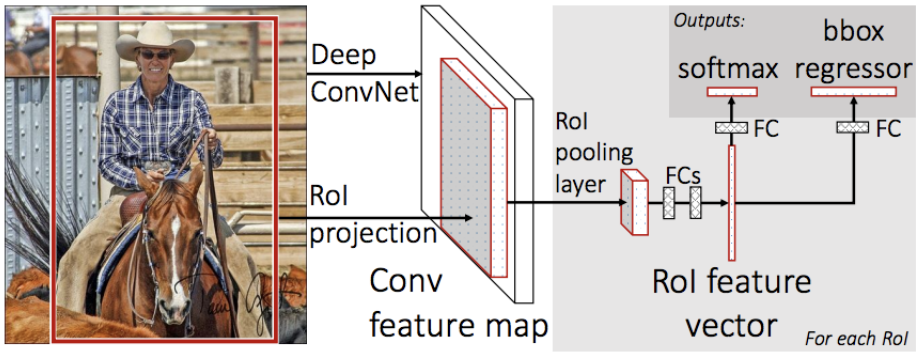

图 6：Fast R-CNN Architecture

- Why Fast R-CNN More Faster?
  - “快速 R-CNN”比 R-CNN 更快的原因是因为不必每次都向卷积神经网络提供 2000 个区域提议。相反，每个图像只进行一次卷积操作，并从中生成特征图。

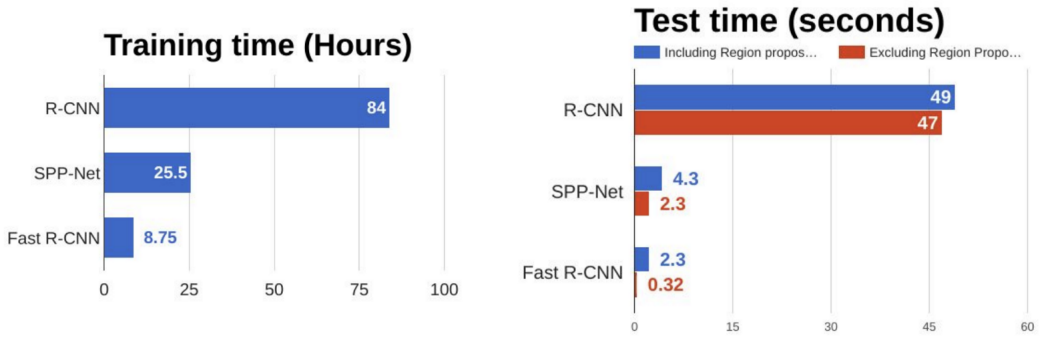

图 7：Comparison of object detection algorithms

- 从上面的图表中，可以看出 Fast R-CNN 在训练和测试上比 R-CNN 明显更快。通过右图 → 测试时间对比图可以看出，Fast R-CNN 使用区域提议与不使用区域提议相比，使用区域提议算法会显着降低算法速度。因此，**区域提议成为影响 Fast R-CNN 算法性能的的瓶颈**。

Faster R-CNN

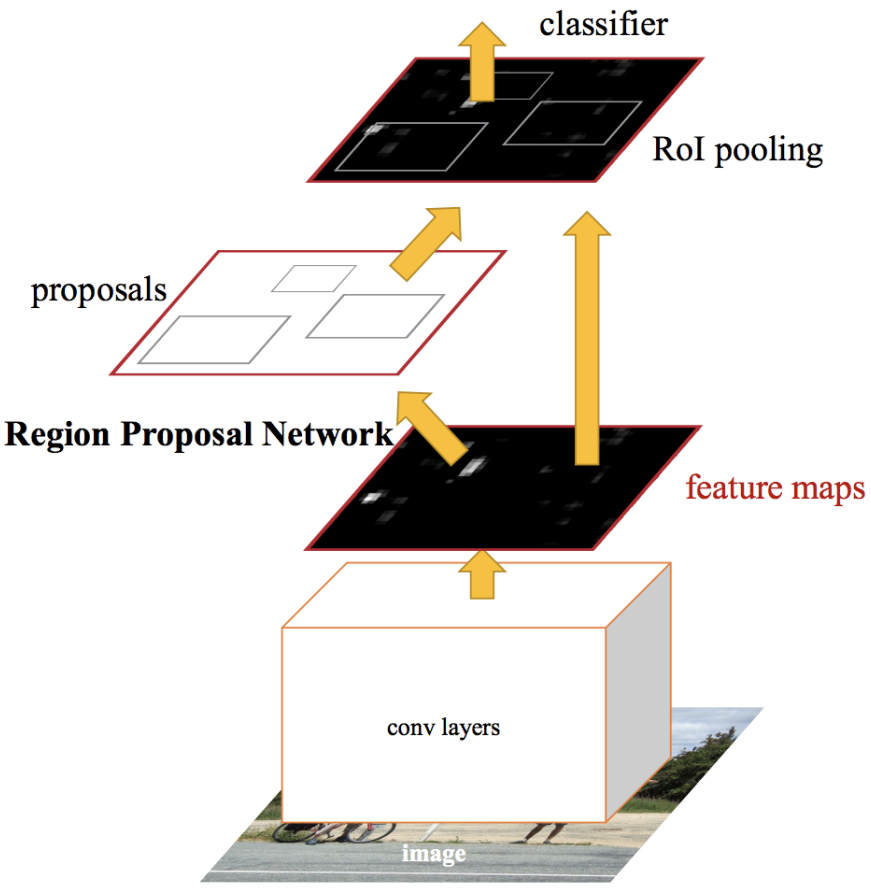

图 8：Faster R-CNN

- R-CNN 和 Fast R-CNN 都使用选择性搜索算法来找出区域提议。选择性搜索是一个影响网络性能的缓慢且耗时的过程。因此，Shaoqing Ren etal. 提出了一种物体检测算法，该算法移除了选择性搜索算法，并让网络自己学习区域提案。
- 类似于 Fast R-CNN，图像被作为卷积神经网络的输入，卷积神经网络输出则为特征图这时不再使用特征图上的选择性搜索算法来提供区域提议，而是使用单独的网络来预测区域提议。然后使用 RoI 池化层对预测的区域提议进行整形（Reshape），然后使用该池化层对所提议的区域内的图像进行分类并预测边界框的偏移值。

YOLO—You Only Look Once

- 所有先前的对象检测算法使用区域 (region) 来定位图像内的对象。网络不会查看完整的图像。相反，图像的一部分具有包含对象的高概率。YOLO 或 You Only Look Once 物体检测算法与上面提到的基于区域的算法有很大不同。在 YOLO 中，单个卷积神经网络预测这些边界框偏移量和类别概率。

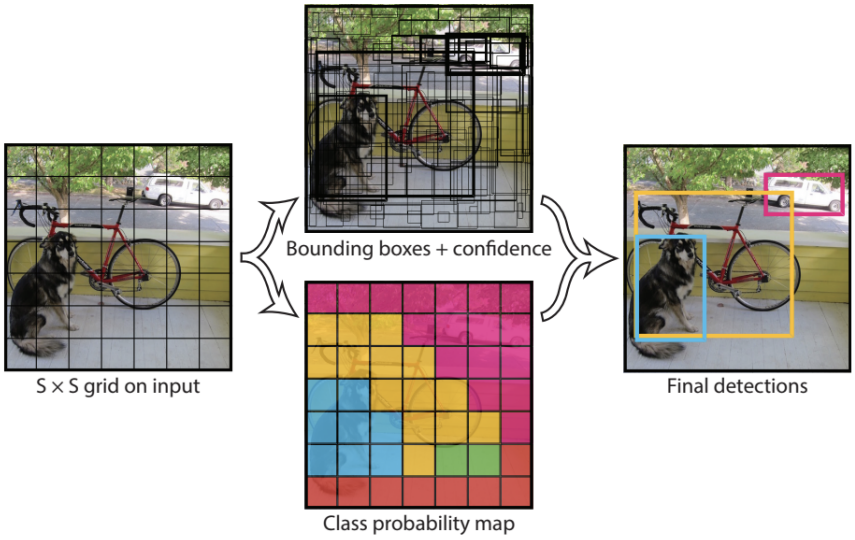

图 8：You Only Look Once

- YOLO 的工作原理是将我们拍摄一张图像分割成一个 SxS 的网格，在每个网格中我们选择了边界框。对于每个边界框，网络输出边界框的类概率和偏移值。选择具有高于阈值的类概率的边界框并用于定位图像内的对象。

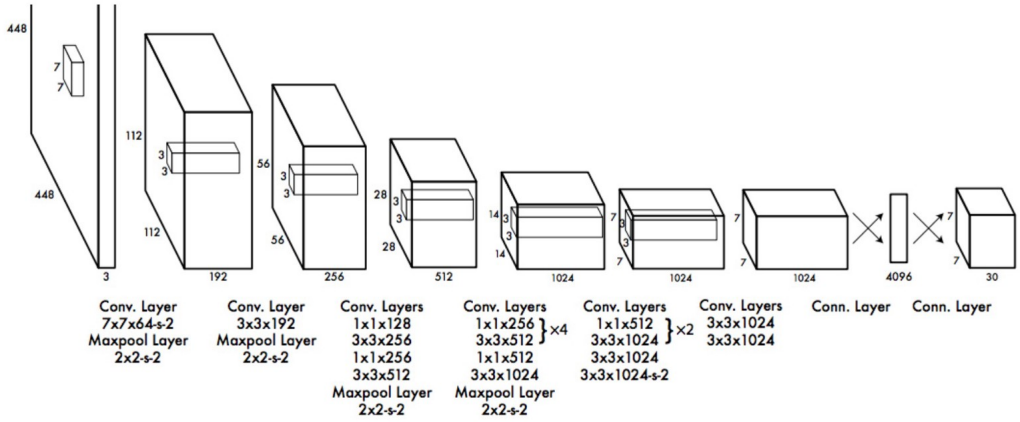

Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

图 8：YOLO Architecture