从 RNN 到 Mask RCNN--简述 CNN 在图像分割应用中的发展历程

第一章 引言

Convolutional Neural Networks(CNNs) 不仅仅被用来分类，而且被用于很多其它方面。在这篇文章中，我们将看到 CNN 如何被用于提升图像实例分割任务中的结果。

自从 Alex Krizhevsky, Geoff Hinton, and Ilya Sutskever 在 2012 年赢得了 ImageNet，Convolutional Neural Networks(CNNs) 已经成为图像分类的黄金准则。事实上，从那时起，CNNs 已经提高到现在它们在 ImageNet 挑战上超越人类的地步！

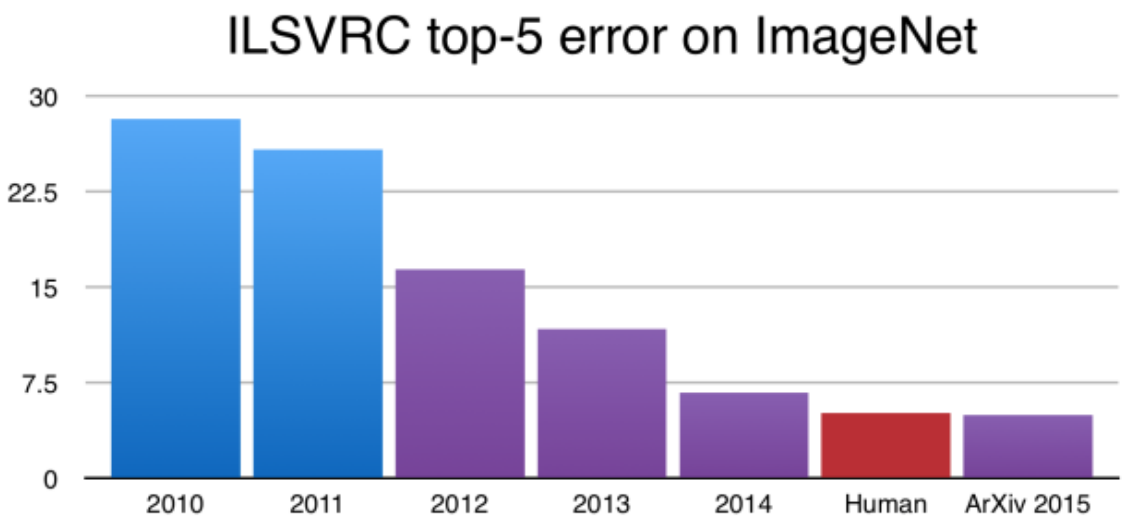

图 1：CNNs 现在已经在 ImageNet 挑战方面胜过人类。上图中的 y 轴是 ImageNet 上的错误率。

虽然这些结果令人印象深刻，但是比起真实的人类视觉理解的复杂性和多样性，图像分类要简单得多。

图 2：在分类挑战中使用的一个图像样本。注意图像是否是构成良好的并且只有一个目标。

在分类中，一个图像通常只有单一目标作为焦点，这个任务是简单地说出那个图像是什么（见上文）。但是，当我们看看我们周围的世界，我们执行比这更为复杂的任务。

图 3：现实生活中的场景通常由许多不同的，重叠的目标，背景和行为组成。

我们看到的复杂的场景通常带有多个重叠的目标和不同的背景，我们不仅分类这些不同的目标，而且识别它们的边界，差异和彼此之间的关系！

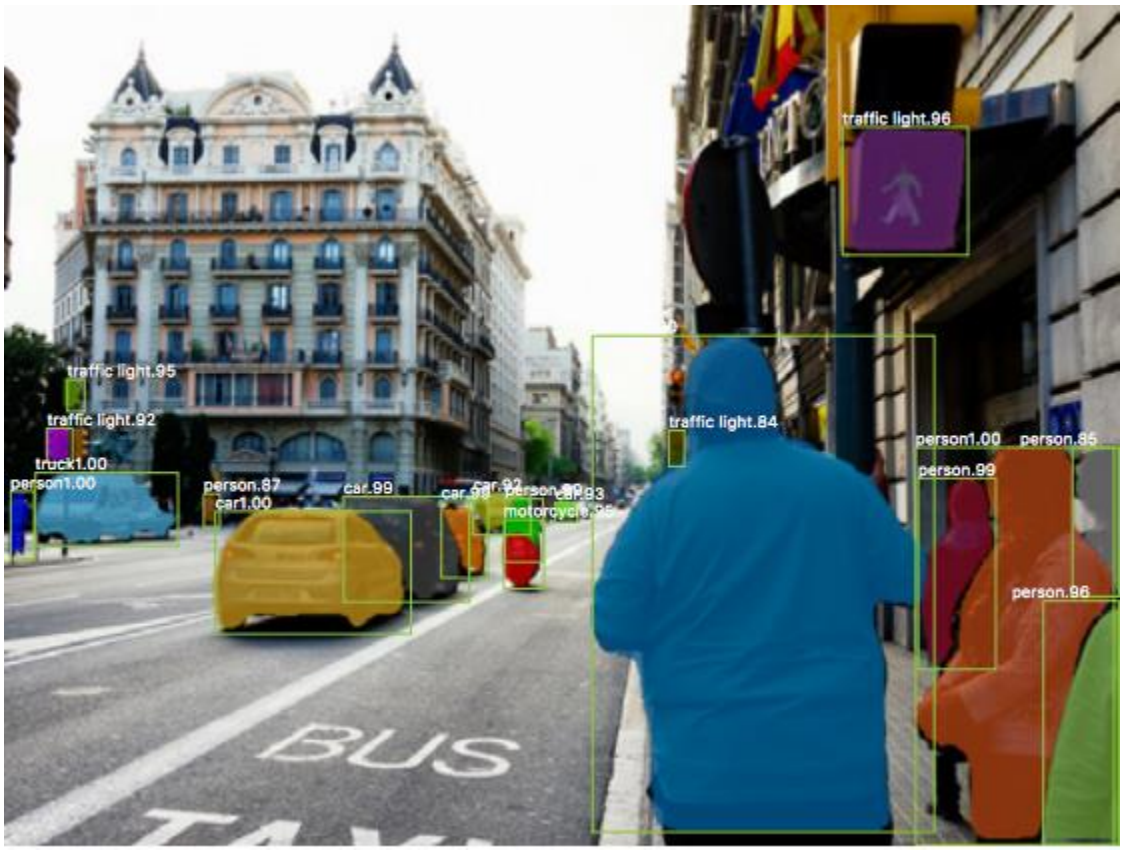

图 4：在图像分割中，我们的目标是对图像中的不同对象进行分类，并确定其边界。Source: Mask R-CNN paper.

CNNs 可以帮助我们实现这样复杂的任务吗？也就是说，给出更复杂的图像，我们可以使用 CNNs 来识别图像中的不同目标及其边界吗？事实上，正如 Ross Girshick 和他的同事在过去几年所展示的那样，答案是肯定的。

第二章 文章目标

通过这篇文章，我们将介绍一些用于目标检测和分割的主要技术背后的直觉思想，并了解它们是如何从一个实现演变到下一个实现的。具体来说，我们将介绍 R-CNN（区域 CNN），CNNs 的这个问题的最原始应用，以及其衍生 Fast R-CNN 和 Faster R-CNN。最后，我们将介绍 Mask R-CNN，Facebook Research 最近发布的一篇论文，扩展了这种目标检测技术，以提供像素级别分割。以下是这篇文章中引用的论文：

1. R-CNN: https://arxiv.org/abs/1311.2524
2. Fast R-CNN: https://arxiv.org/abs/1504.08083
3. Faster R-CNN: https://arxiv.org/abs/1506.01497
4. Mask R-CNN: https://arxiv.org/abs/1703.06870

第三章 综述

3.1 2014：R-CNN——将 CNNs 应用于目标检测的诞生

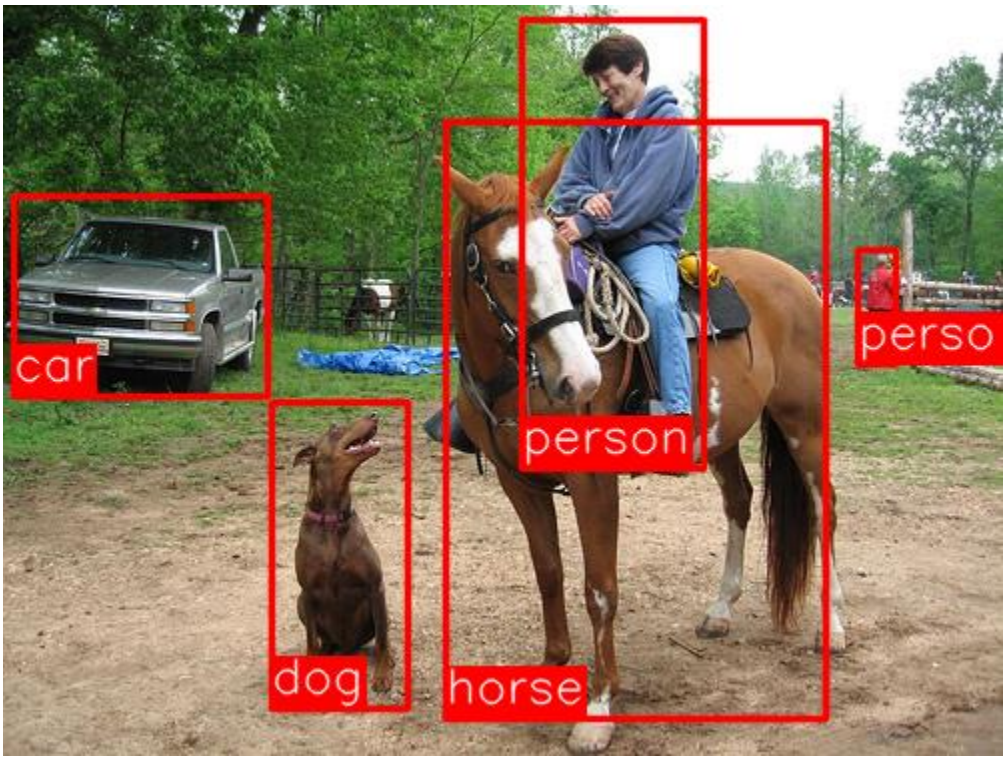

图 5：诸如 R-CNN 的目标检测算法应用在图像上，可以识别图像中主要目标的位置和类别。

受 University of Toronto 的 Hinton’s lab 的研究启发，UC Berkeley 里由 Jitendra Malik 教授领导的一个小团队，问自己一个即使是今天似乎也不可避免的问题：

在多大程度上[Krizhevsky et. al’s results]概括对象检测？

目标检测是找到图像中的不同目标并进行分类的任务（如上图所示）。由 Ross Girshick（我们再次看到的名字），Jeff Donahue 和 Trevor Darrel 组成的团队发现，通过测试 PASCAL VOC Challenge（一种类似于 ImageNet 的受欢迎的目标检测挑战），这个问题确实可以通过 Krizhevsky 的结果解决。他们写到：

This paper is the first to show that a CNN can lead to dramatically higher object detection performance on PASCAL VOC as compared to systems based on simpler HOG-like features.

（这篇论文首次显示，与基于更简单的 HOG 类特征的系统相比，CNN 可以显著提高 PASCAL VOC 上的目标检测性能。）

现在让我们花点时间来理解他们的架构，Regions With CNNs (R-CNN) 如何运作。

3.1.1 理解 R-CNN

R-CNN 的目标是摄取图像，并正确识别图像中主要目标（通过边框）的位置。

- Inputs:图像
- Outputs:边框和图像中每个目标的标签。

但是我们如何找出这些边框在哪里？R-CNN 可以像我们直观上做的那样好--在图像上提出一堆边框，看看它们中的任何一个实际上是否对应一个目标。

Figure 2: Two examples of our selective search showing the necessity of different scales. On the left we find many objects at different scales. On the right we necessarily find the objects at different scales as the girl is contained by the tv.

图 6：通过多个尺度的窗口查找进行选择性的搜索，并查找共享纹理，颜色或强度的相邻像素。Image source: https://www.koen.me/research/pub/uijlings-ijcv2013-draft.pdf

R-CNN 使用称为选择性搜索的过程（可以在此处阅读 here），来创建这些边界框或区域提案。在更高级别，选择性搜索（如上图所示）通过不同大小的窗口查看图像，并且对于每个大小，尝试通过纹理，颜色或强度将相邻像素分组在一起，以识别目标。

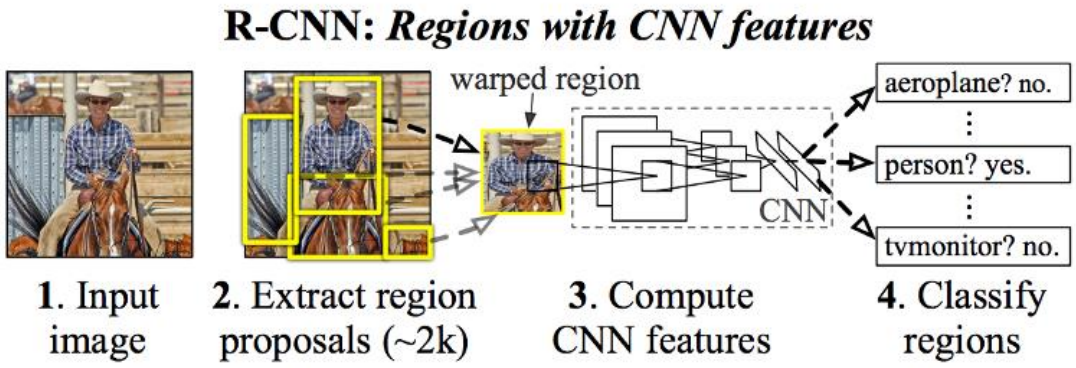

图 7：在创建一组区域提案后，R-CNN 只需将图像传递给 AlexNet 的修改版本，以确定它是否为有效区域。

一旦创建了这些提案，R-CNN 就会简单地将该区域扭曲到一个标准的平方尺寸，并将其传递给一个修改版本的 AlexNet（ImageNet 2012 获胜者提交，启发了 R-CNN）如图 7 所示。

在 CNN 的最后一层，R-CNN 添加了一个支持向量机（SVM），它简单地分类这是否是一个对象，并且是什么目标对象。这是上图中的第 4 步。

3.1.2 改善边界框

现在，在边界框里找到了对象，我们能够收紧盒子以适应对象的真实尺寸吗？的确，我们可以这样做，这也是 R-CNN 的最后一步。R-CNN 对区域提案进行简单的线性回归，以生成更紧密的边界框坐标并获得最终结果。这是回归模型的输入和输出：

- Inputs: 对应于对象的图像的子区域
- Outputs: 在子区域中对象的新边界框坐标。

3.1.3 R-CNN 算法流程

因此，总结一下，R-CNN 只是包含以下简单的步骤：

1. 生成一组边界框的提案。
2. 通过预先训练的 AlexNet 运行边框中的图像，最后通过 SVM 来查看框中图像包含的对象。
3. 一旦对象被分类，通过线性回归模型输出输入框中更紧密的坐标。

3.2 2015：Fast R-CNN——加速和简化 R-CNN

图 8：Ross Girshick 研究 R-CNN 和 Fast R-CNN。他继续推动 Facebook Research 的计算机视觉界限。

R-CNN 的工作效果非常好，但由于以下几个简单的原因，实际上相当慢：

1. 它需要 CNN（AlexNet）针对每个单个图像的每个区域提案前向传递（每个图像大约 2000 次前向传递）。
2. 它必须分别训练三种不同的模型--CNN 生成图像特征，预测类别的分类器和收紧边界框的回归模型。这使得管线非常难以训练。

在 2015 年，R-CNN 的第一作者 Ross Girshick，解决了这两个问题，导致了我们短期历史上的第二个算法 -- Fast R-CNN。现在我们来看看它的主要见解。

3.2.1 Fast R-CNN 见解 1：RoI (Region of Interest) 池化

对于 CNN 的前向传递，Girshick 意识到，对于每个图像，图像的许多提出的区域总是重叠，使得我们一次又一次地运行相同的 CNN 计算（~2000 次！）他的见解很简单 -- **为什么不每个图像只运行一次 CNN，然后在 ~2000 个提案中找到一种分享这种计算的方法？**

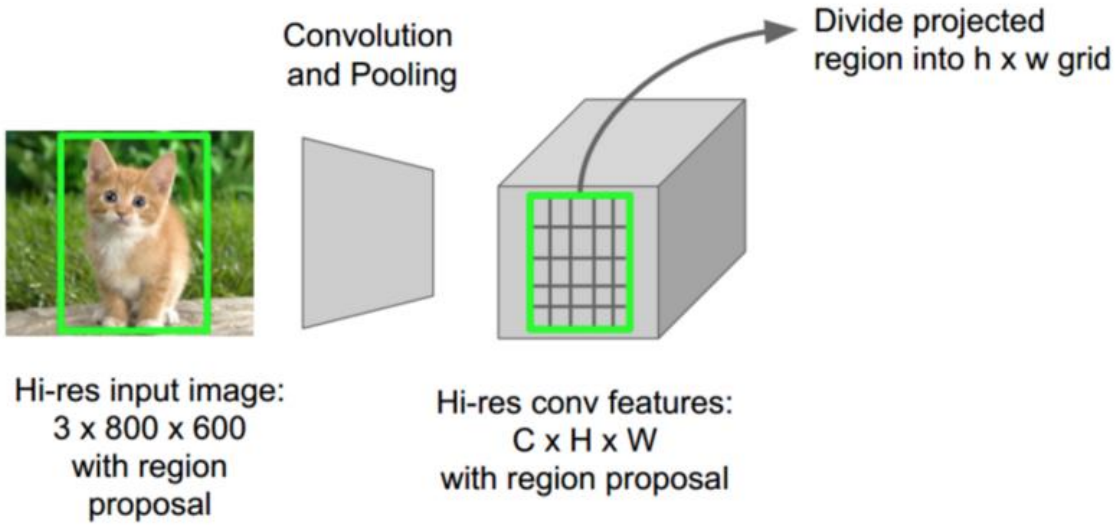

图 9：在 RoIPool 中，创建图像的完整前向传递，并从所得到的前向传递中提取每个感兴趣区域的卷积特征。Source: Stanford’s CS231N slides by Fei Fei Li, Andrei Karpathy, and Justin Johnson.

这正是 Fast R-CNN 使用被称为 RoIPool（Region of Interest Pooling）的技术。其核心在于，RoIPool 分享了 CNN 在图像所有子区域的向前传递。在上图中，注意如何通过从 CNN 的特征图选择相应的区域来获取每个区域的 CNN 特征。然后，每个区域的特征简单的池化（通常使用最大池）。所以它只花费我们对图像的一遍遍历，而不是相对的 ~2000 次！

3.2.2 Fast R-CNN 见解 2：将所有模型整合成一个网络

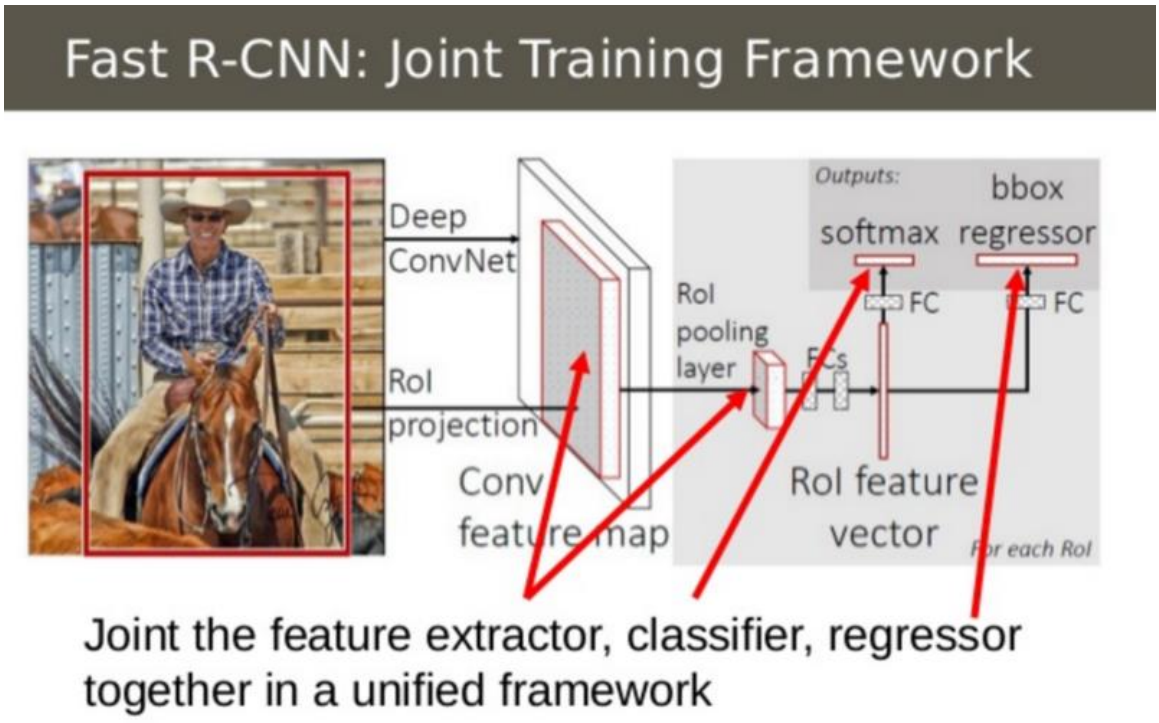

图 10：Fast R-CNN 结合 CNN，分类器和边界盒回归器，组合成一个简单的网络。

Fast R-CNN 的第二个见解是在单一模型中联合训练 CNN，分类器和边界盒回归器。早些时候，我们有不同的模型提取图像特征（CNN），分类（SVM）和收紧边界框（回归），**Fast R-CNN 使用单个网络来代替计算所有三个。**

你可以在上面的图像中看到它是如何实现的。Fast R-CNN 使用 CNN 顶部的简单 softmax 层代替了 SVM 分类器，以输出分类。它还添加了与 softmax 层平行的

线性回归层以输出边界框坐标。 这样，所有需要的输出来自一个单一的网络！这是整个模型的输入和输出：

- Inputs: 具有区域提案的图像。
- Outputs: 每个区域的对象分类以及更紧密的边界框。

3.3 2016: Faster R-CNN——加速区域提案

即使已经有了这些进步，Fast R-CNN 流程仍然存在一个瓶颈——区域提出者。正如我们所看到的那样，检测对象位置的第一步是产生一堆潜在的边界框或感兴趣区域去进行测试。在 Fast R-CNN 中，这些提案是使用选择性搜索创建的，这是一个相当缓慢的过程，被认为是整个流程的瓶颈。

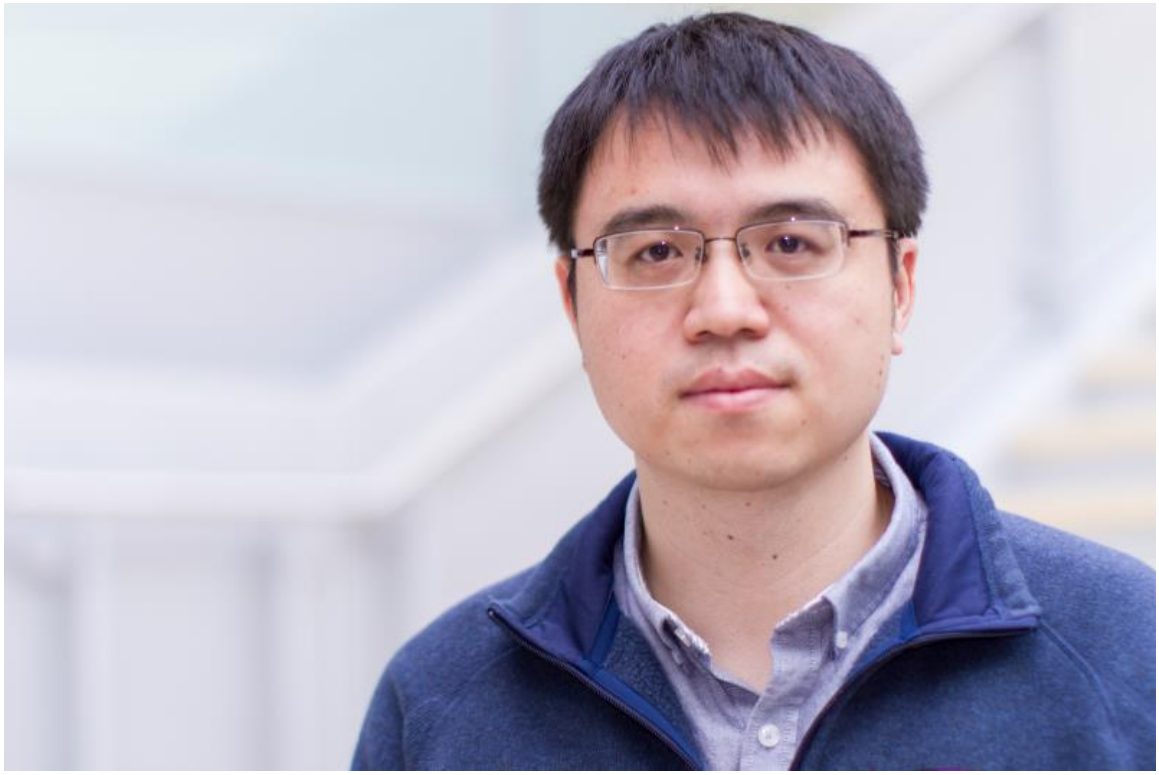

图 11：Jian Sun, Microsoft Research 的首席研究员，领导了 Faster R-CNN 的团队。Source: https://blogs.microsoft.com/next/2015/12/10/microsoft-

researchers-win-imagenet-computer-vision-challenge/#sm.00017fqnl1bz6fqf11amuo0d9ttdp

在 2015 年中期，Microsoft Research 中一个由 Shaoqing Ren, Kaiming He, Ross Girshick, 和 Jian Sun 组成的团队，找到了一种方法，通过一个他们命名为“Faster R-CNN”（创造性）的架构，使得区域的提案步骤花费几乎为 0。

Faster R-CNN 的见解是，区域提案取决于已经通过 CNN 的前向计算的图像的特征（第一步分类）。那么为什么不为区域提案重用那些相同的 CNN 结果，而不是运行单独的选择性搜索算法？

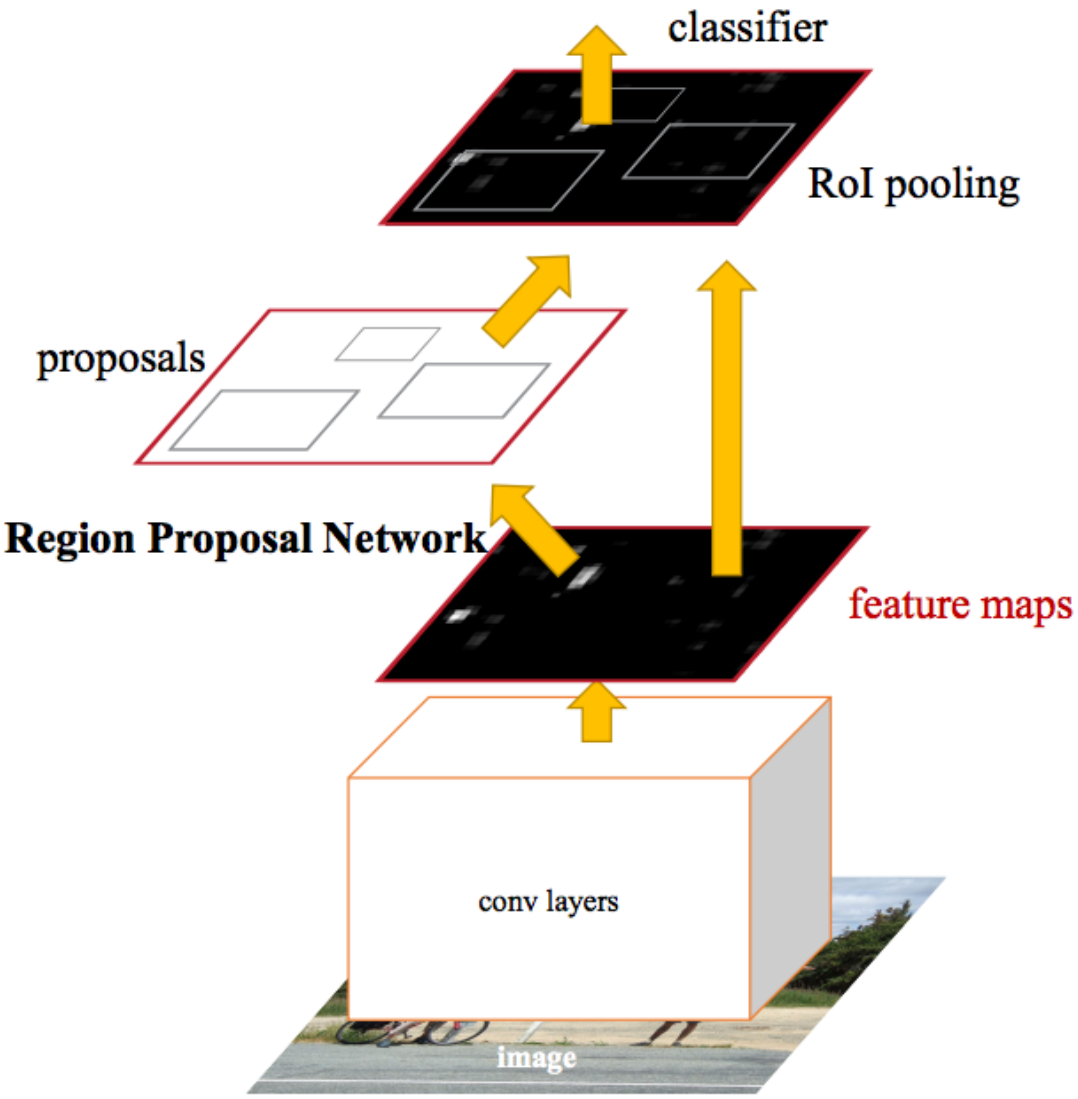

图 12：在 Faster R-CNN 中，单个 CNN 用于区域提案和分类。

的确，这正是 Faster R-CNN 团队所取得的成就。在上图中，您可以看到如何使用单个 CNN 来执行区域提案和分类。这样，只有一个 CNN 需要接受训练，我们几乎免费获得区域提案！作者写道：

Our observation is that the convolutional feature maps used by region-based detectors, like Fast R- CNN, can also be used for generating region proposals [thus enabling nearly cost-free region proposals].

（我们的观察结果是，基于区域的检测器（如 Fast R-CNN）使用的卷积特征图也可用于生成区域提案[从而实现几乎无成本区域提案]。）

以下是其模型的输入和输出：

- Inputs: 图像（注意不需要区域提案）。
- Outputs: 图像中目标的分类和边界框坐标。

3.3.1 如何生成区域

让我们花点时间看看 Faster R-CNN 如何从 CNN 特征中生成这些区域提案。Faster R-CNN 在 CNN 的特征之上添加了一个简单的完全卷积网络（Fully Convolutional Network），构建了所谓的区域提案网络。

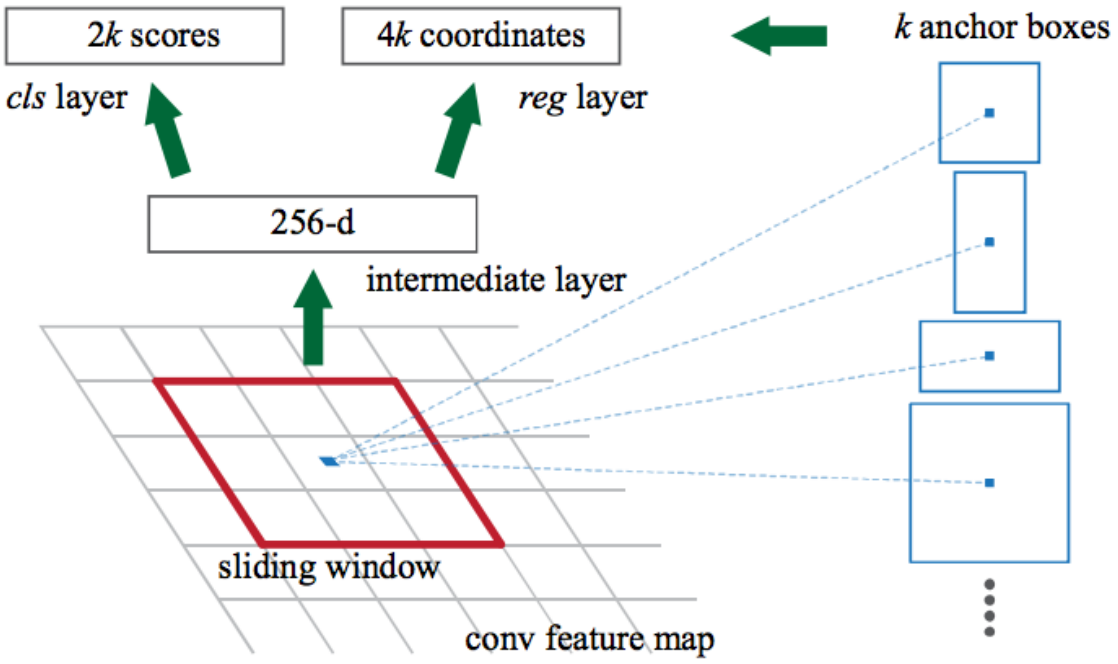

图 13：区域提案网络在 CNN 的特征上滑动一个窗口。在每个窗口位置，网络输出一个得分和一个边界框的每个锚点（因此，4k 个边框坐标，其中 k 是锚的数量）

区域提案网络通过在 CNN 特征图上滑动窗口，并在每个窗口中输出 k 个潜在的边界框和分数，以便预计这些框的选取效果。这些 k 盒代表什么？

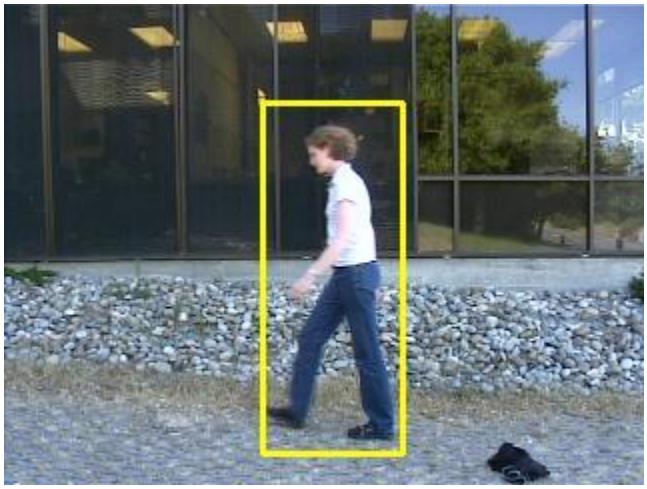

图 14：我们知道，人的边框往往是矩形和垂直的。 我们可以使用这种见解，通过创建这样的维度的锚点，来指导我们的区域提案网络。Image

Source: http://vlm1.uta.edu/~athitsos/courses/cse6367_spring2011/assignments/assignment1/bbox0062.jpg

直观地，我们知道图像中的对象应该符合某些常见的纵横比和大小。例如，我们知道我们想要一些类似于人类形状的矩形框。同样，我们知道我们不会看到很多盒子非常薄。以这种方式，我们创建 k 这样的通用长宽比，我们称之为锚盒。对于每个这样的锚盒，我们在图像中输出一个边界框和每个位置的得分。

考虑到这些锚盒，我们来看看这个区域提案网络的输入和输出：

- Inputs: CNN 特征图
- Outputs: 每个锚点的边框和表示该边界框中的图像作为对象的可能性的分数。

然后，我们简单地将每个可能成为对象的边界框传递到 Fast R-CNN 中，以生成分类和收紧边界框。

3.4 2017：Mask R-CNN——扩展 Faster R-CNN 进行像素级分割

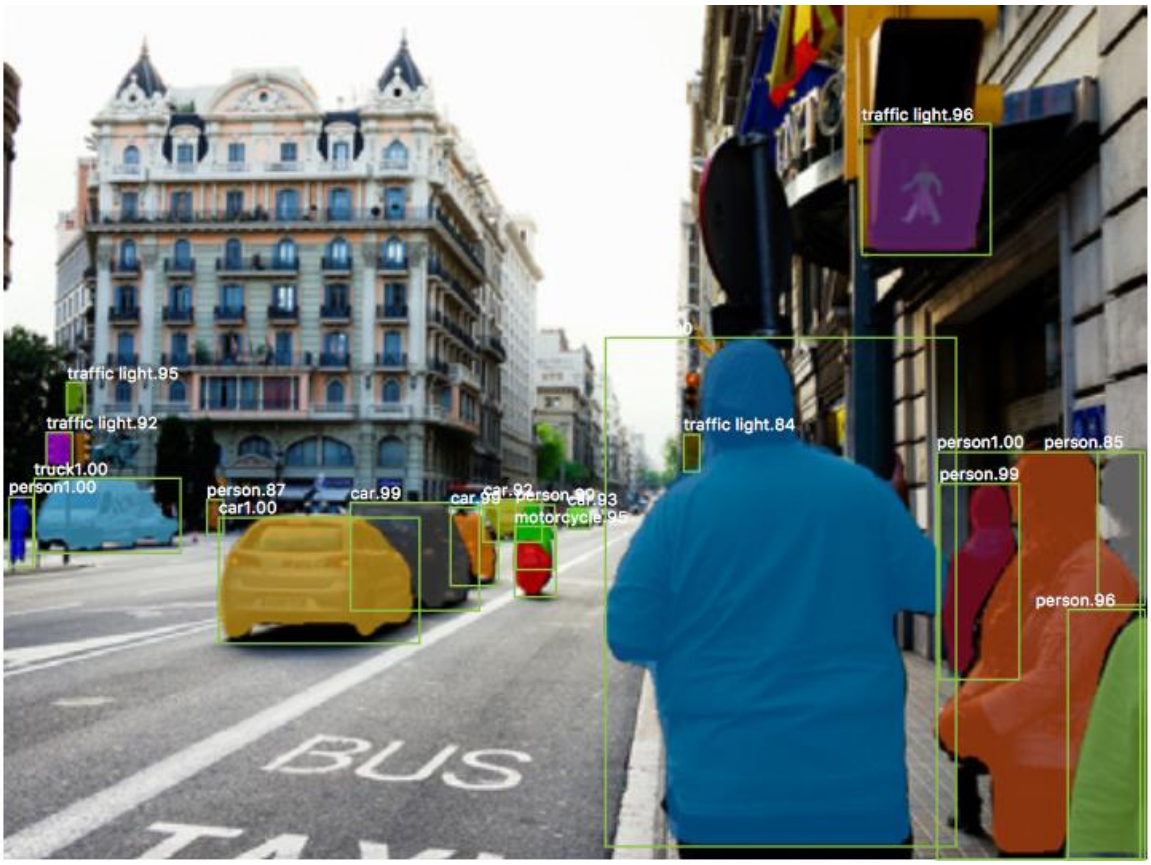

图 15：图像实例分割的目的是在像素级别识别场景中不同的对象。

到目前为止，我们已经看到我们如何能够以许多有趣的方式使用 CNN 特征，有效地使用边框来定位图像中的不同对象。

我们是否可以将这些技术进一步扩展，并定位每个对象的精确像素，而不是仅限于边框？这个称为图像分割的问题是 Kaiming He 和一组研究人员，包括 Girshick，在 Facebook AI 上使用被称为 Mask R-CNN 的架构进行了探索。

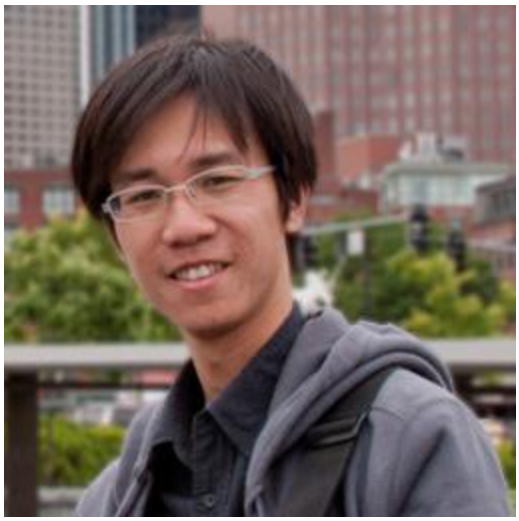

图 16：Kaiming He,一名 Facebook AI 的研究员，是 Mask R-CNN 的首席作者，也是 Faster R-CNN 的合着者。

就像 Fast R-CNN 和 Faster R-CNN，Mask R-CNN 的底层见解非常简单。鉴于 Faster R-CNN 对于物体检测的效果非常好，我们是否可以简单地将其扩展到能够进行像素级分割？

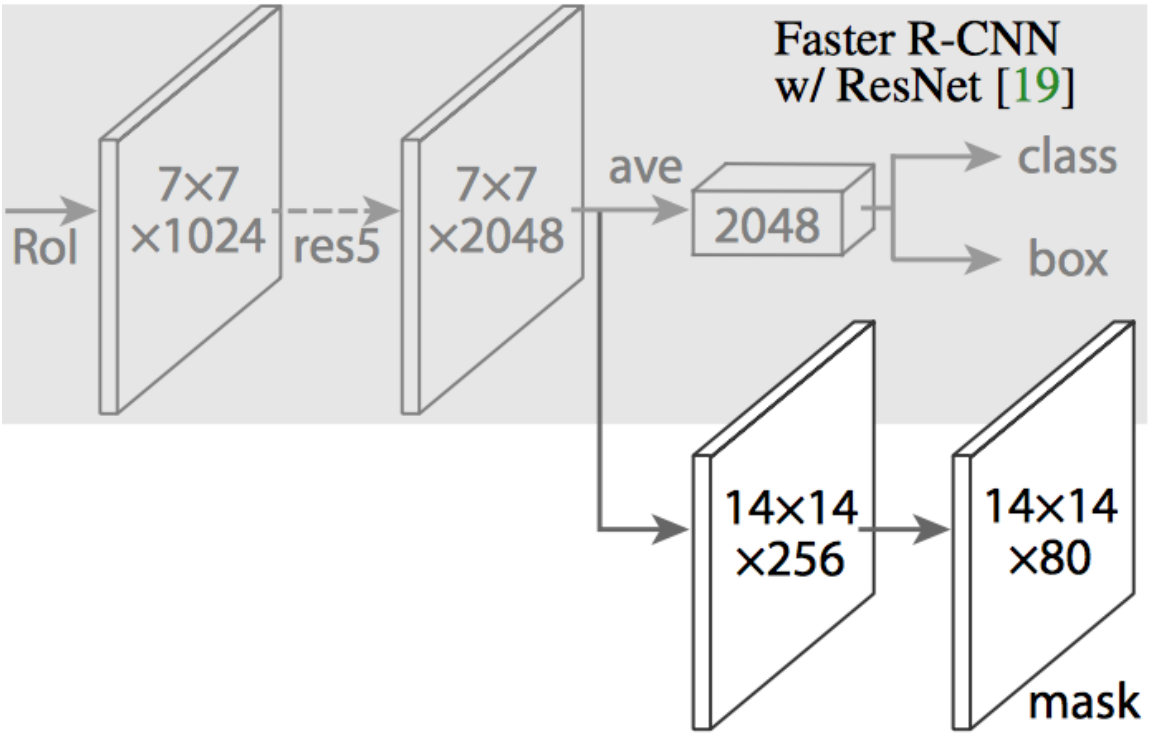

图 17：在 Mask R-CNN 里，在 Faster R-CNN 的 CNN 特征之上添加了一个简单的完全卷积网络（FCN），以生成掩码（分段输出）。请注意，这与 Faster R-CNN 的分类和边界框回归网络是如何并行的。

Mask R-CNN 通过简单地向 Faster R-CNN 添加一个分支来输出一个二进制掩码，来说明给定像素是否是对象的一部分。如上所述，分支（在上图中为白色）仅仅是基于 CNN 的特征图上的简单的完全卷积网络。 以下是其输入和输出：

- Inputs: CNN 特征图
- Outputs: 矩阵在像素属于目标的所有位置都有 1，其他位置为 0（这被称为二进制掩码（binary mask））。

但是 Mask R-CNN 不得不进行一个小的调整，使这个管线按预期工作。

3.4.1 RoiAlign -- 重新定位 RoIPool 使其更准确

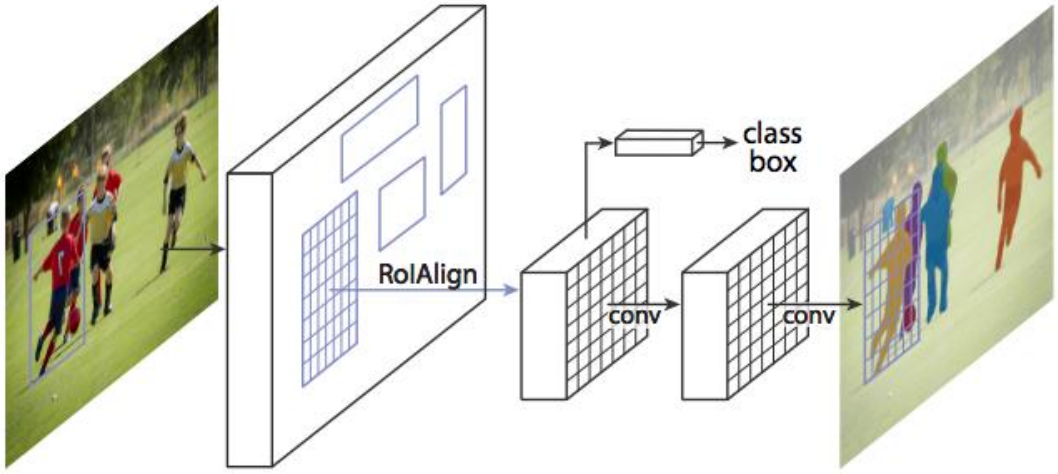

图 18：与 RoIPool 相对，图像通过 RoIAlign 传递，使得由 RoIPool 选择的特征图的区域更精确地对应于原始图像的区域。这是需要的，因为像素级分割需要比边界框更细粒度的对齐。

当运行原始的未经修改的 Faster R-CNN 架构时，Mask R-CNN 作者意识 RoIPool 选择的特征图的区域与原始图像的区域略有不对齐。由于图像分割需要像素级特异性，与边框不同，这自然导致不准确。

作者通过使用 RoIAlign 方法简单地调整 RoIPool 来更精确地对齐，从而解决了这个问题。

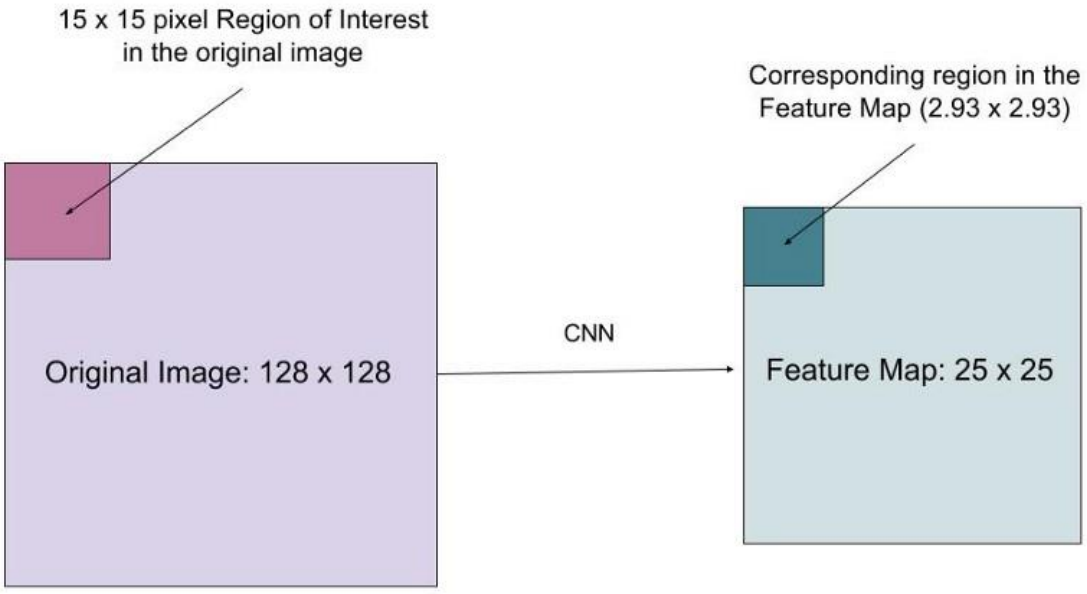

图 19：我们如何准确地将原始图像的兴趣区域映射到特征图上？

想象一下，我们有一个大小为 128x128 的图像和大小为 25x25 的特征图。让我们想象一下，我们想要的是与原始图像中左上角 15x15 像素对应的区域特征（见上文）。 我们如何从特征图中选择这些像素？

我们知道原始图像中的每个像素对应于原始图像中的～25/128 像素。 要从原始图像中选择 15 像素，我们只选择 15 * 25/128～= 2.93 像素。

在 RoIPool 中，我们会将其舍入，并选择 2 个像素，导致轻微的错位。然而，在 RoIAlign 中，我们避免了这种四舍五入。 相反，我们使用双线性插值（bilinear interpolation）来精确地构想像素 2.93 中的内容。这在很大程度上是让我们避免 RoIPool 造成的错位。

一旦这些掩模生成，Mask R-CNN 简单地将它们与来自 Faster R-CNN 的分类和边界框组合起来，以产生如此奇妙的精确分割：

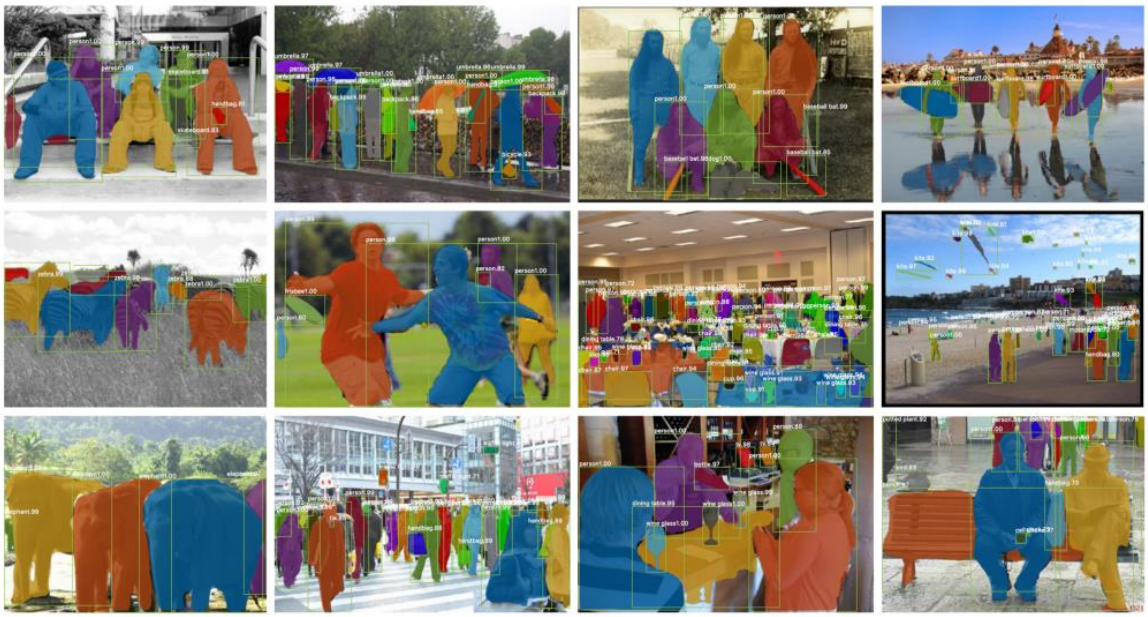

图 20：Mask R-CNN 能够对图像中的对象进行分段和分类。

第四章 展望

在短短 3 年的时间里，我们已经看到了研究界从 Krizhevsky 等进步，这是 R-CNN 的最初成果，最后一路成为 Mask R-CNN 的强大效果。孤立地看，Mask R-CNN 的结果看起来似乎是无法达到的无与伦比的天才飞跃。然而，通过这篇文章，我希望您已经看到，通过多年的辛勤工作和协作，这些进步实际上是直观，渐进的提升之路。R-CNN, Fast R-CNN, Faster R-CNN 和最终的 Mask R-CNN 提出的每个想法并不一定是跨越式发展，但是它们的总和结果却带来了非常显著的效果，使我们更接近于人类的理解视觉水平。特别令我兴奋的是，R-CNN 和 Mask R-CNN 之间的时间只有三年！随着资金，关注和支持的增多，计算机视觉在未来三年会有怎样进一步的进展？

参考文献

[1] https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation-from-r-cnn-to-mask-r-cnn-34ea83205de4

[2]word: https://github.com/BasicCoder/BriefHistoryOfCNNsInImageSegmentation/blob/master/CNN%E5%9C%A8%E5%9B%BE%E5%83%8F%E5%88%86%E5%89%B2%E4%B8%AD%E7%9A%84%E7%AE%80%E5%8F%B2%EF%BC%9A%E4%BB%8E%20R-CNN%20%E5%88%B0Mask%20R-CNN.docx

[3]PDF:

[4]有关论文：