- 背景

本篇论文的题目是 《Rich feature hierarchies for accurate oject detection and semantic segmentation》，翻译过来就是针对高准确度的目标检测与语义分割的多特征层级，通俗地来讲就是一个用来做目标检测和语义分割的神经网络。

这篇论文发布时间是 2014 年，它具有很多比较重要的意义。

在 Pascal VOC 2012 的数据集上，能够将目标检测的验证指标 mAP 提升到 53.3%,这相对于之前最好的结果提升了整整 30%。

这篇论文证明了可以讲神经网络应用在自底向上的候选区域,这样就可以进行目标分类和目标定位。

这篇论文也带来了一个观点，那就是当你缺乏大量的标注数据时，比较好的可行的手段是，进行神经网络的迁移学习，采用在其他大型数据集训练过后的神经网络，然后在小规模特定的数据集中进行 fine-tune 微调。

- 什么是目标检测

给定一张图片可以识别出类别就是，**对象识别**。而**目标检测**除了要识别类别外，还要找到他们的位置。显然，目标检测比对象识别更难。

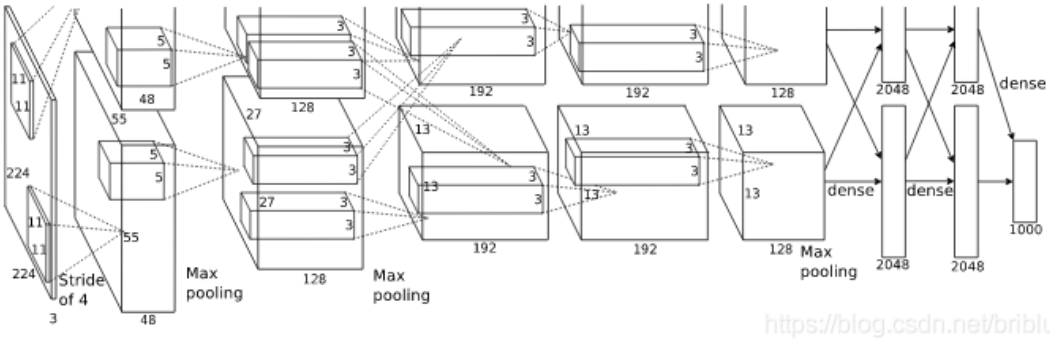
- R-CNN 在前人的肩膀上前行

在过去的十多年时间里，传统的机器视觉领域，通常采用特征描述子来应对目标识别任务，这些特征描述子最常见的就是 SIFT 和 HOG.而 OpenCV 有现成的 API 可供大家实现相关的操作。

SIFT 和 HOG 的王者地位最近被卷积神经网络撼动。

2012 年 Krizhevsky 等人在 ImageNet 举办的 ILSVRC 目标识别挑战大赛中一战成名，豪夺当年的第一名，Top5 错误率 15%,而他们团队提出来的网络结构以第一作者 Alex

Krizhevsky 名字命名，它就是 AlexNet。



它有 5 层卷积层,2 层全连接层。

因为 AlexNet 的出现，世人的目光重回神经网络领域，以此为契机，不断涌出各种各样的网络比如 VGG、GoogleNet、ResNet 等等。

受 AlexNet 启发，论文作者尝试将 AlexNet 在 ImageNet 目标识别的能力泛化到 PASCAL VOC 目标检测上面来。

但一切开始之前，需要解决两个主要的问题。

1. 如何利用深度的神经网络去做目标的定位？
2. 如何在一个小规模的数据集上训练能力强劲的网络模型？

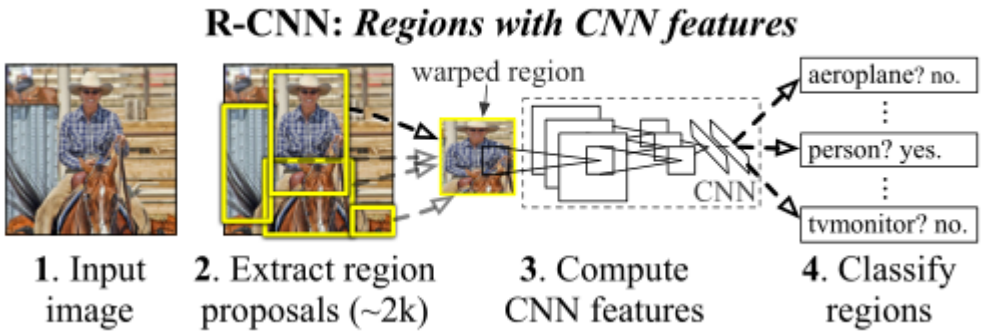
论文作者给出了思路。

- 利用候选区域与 CNN 结合做目标定位

借鉴了滑动窗口思想，R-CNN 采用对区域进行识别的方案。具体是：

1. 给定一张输入图片，从图片中提取 2000 个类别独立的候选区域。
2. 对于每个区域利用 CNN 抽取一个固定长度的特征向量。
3. 再对每个区域利用 SVM 进行目标分类。

下面的图像来自论文本身。



- **利用预训练与微调解决标注数据缺乏的问题**

采用在 ImageNet 上已经训练好的模型，然后在 PASCAL VOC 数据集上进行 fine-tune。因为 ImageNet 的图像高达几百万张，利用卷积神经网络充分学习浅层的特征，然后在小规模数据集做规模化训练，从而可以达到好的效果。现在，我们称之为迁移学习，是必不可少的一种技能。

- R-CNN 的目标识别之路

前面内容提到过，R-CNN 系统分为 3 个阶段，反应到架构上由 3 个模块完成。

1. 生产类别独立的候选区域，这些候选区域其中包含了 R-CNN 最终定位的结果。
2. 神经网络去针对每个候选区域提取固定长度的特征向量。
3. 一系列的 SVM 分类器。

- 候选区域

能够生成候选区域的方法很多，比如：

1. objectness
2. selective search
3. category-independen object proposals

4. constrained parametric min-cuts(CPMC)

5. multi-scale combinatorial grouping

6. Ciresan

R-CNN 采用的是 Selective Search 算法。

- 特征抽取

R-CNN 抽取了一个 4096 维的特征向量，采用的是 Alexnet，基于 Caffe 进行代码开发。需要注意的是 Alextnet 的输入图像大小是 227x227。

而通过 Selective Search 产生的候选区域大小不一，为了与 Alexnet 兼容，R-CNN 采用了非常暴力的手段,那就是无视候选区域的大小和形状,统一变换到 227*227 的尺寸。

有一个细节，在对 Region 进行变换的时候，首先对这些区域进行膨胀处理，在其 box 周围附加了 p 个像素，也就是人为添加了边框，在这里 $p=16$。

- **测试阶段的目标检测**

在测试阶段，R-CNN 在每张图片上抽取近 2000 个候选区域。然后将每个候选区域进行尺寸的修整变换，送进神经网络以读取特征，然后用 SVM 进行类别的识别，并产生分数。

候选区域有 2000 个，所以很多会进行重叠。

针对每个类，通过计算 IoU 指标，采取非极大性抑制，以最高分的区域为基础，剔除掉那些重叠位置的区域。

- **运行时分析**

两个因素可以让目标识别变得高效。

1. CNN 的参数是所有类别共享的。

2. R-CNN 生成的特征向量维度较少。论文拿应用在 UVA 采用的空间金字塔技术相比，它们生成的特征维度是 360k，而 R-CNN 就 4K 多。

也就是运行过程中，参数变少了，所以比传统的高效。体现在提取特征的时间，如果用 GPU ，13s/张，CPU 53s/张。R-CNN 能够处理 100k 种类别，在一个多核的 CPU 上只要花费 10 多秒。与 UVA 相比，如果处理 100k 个预测，需要 134GB 内存空间，而 R-CNN 只要 1.5GB。

- **训练**

前面已经提到过 R-CNN 采取迁移学习。提取在 ILSVRC 2012 的模型和权重，然后在 VOC 上进行 fine-tune。

需要注意的是，这里在 ImageNet 上训练的是模型识别物体类型的能力，而不是预测 bbox 位置的能力。

ImageNet 的训练当中需要预测 1000 个类别，而 R-CNN 在 VOC 上进行迁移学习时，神经网络只需要识别 21 个类别。这是 VOC 规定的 20 个类别加上背景这个类别。

R-CNN 将候选区域与 GroundTrue 中的 box 标签相比较，如果 IoU > 0.5，说明两个对象重叠的位置比较多，于是就可以认为这个候选区域是 Positive,否则就是 Negetive.

训练策略是：采用 SGD 训练，初始学习率为 0.001，mini-batch 大小为 128.

- **对象识别相关**

通常对待一个二值化的分类器，它的结果只要 2 中，Positive 和 Negetive。

比如，有一个汽车分类器，它可以轻松地确认，一个方框里面包含了一辆汽车，那么它肯定就是 Positive。也可以很清楚地确认，如果一个背景方框中没有包含汽车，那么它就是 Negetive。但是，比较难确认的是，如果一个方框，只有一部分与汽车重叠，那么如何标注这个方框呢？

R-CNN 采用的是 IoU 的阈值，这个 threshold 取 0.3，如果一个区域与 Ground tureth 的 IoU 值低于设定的阈值，那么可以讲它看成是 Negetive.

IoU 的 threshold 它不是作者胡乱取值的，而是来自 {0,0.1,0.2,0.3,0.4,0.5} 的数值组合的。而且，这个数值至关重要，如果 threshold 取值为 0.5,mAP 指标直接下降 5 个点，如果取值为 0，mAP 下降 4 个点。

一旦特征抽取成功，R-CNN 会用 SVM 去识别每个区域的类别，但这需要优化。因为训练的数据太大，不可能一下子填充到电脑内存当中，R-CNN 作者采取了一种叫做 Hard negetive mining 的手段。

- R-CNN 的在 PASCAL-VOC 2010-12 的表现

R-CNN 是在 PASCAL VOC 2012 进行最终的 fine-tune,也是在 VOC 2012 的训练集上优化 SVM.然后,还与当时 4 个强劲的对手,也就是 4 个不同的目标检测算法进行了比较。

VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [17]†	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [32]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [35]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [15]†	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN BB	**71.8**	**65.8**	**53.0**	**36.8**	**35.9**	**59.7**	**60.0**	**69.9**	**27.9**	**50.6**	**41.4**	**70.0**	**62.0**	**69.0**	**58.1**	**29.5**	**59.4**	**39.3**	**61.2**	**52.4**	**53.7**

值得关注的是，上面表格中 UVA 检测系统也采取了相同的候选区域算法，但 R-CNN 的表现要好于它。

- **可视化、框架精简和错误检测**

我们都知道，在卷积神经网络中，第一层可以直接用来显示，而且肉眼可视，通常他们是为了捕捉物体边缘，及突出的颜色信息，但越往后的卷积层越抽象，这个时候进行可视化就是一个挑战了。

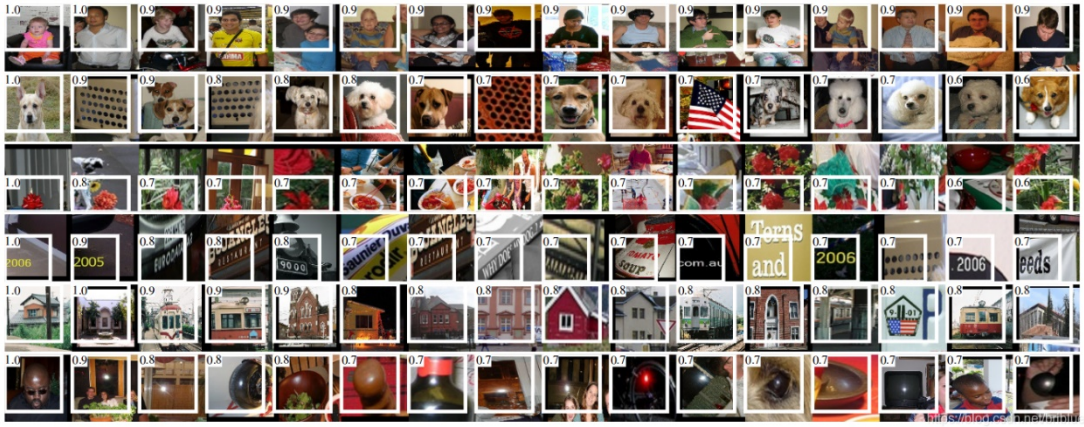
Zeiler 和 Fergus 提出了一种基于反卷积手段的可视化研究，但 R-CNN 的作者直接

提供了一个没有参数的方法，简单直接。

思路是挑选一个特征出来，把它直接当成一个物体分类器，然后计算它们处理不同的候选区域时,activation 的值,这个值代表了特征对这块区域的响应情况,然后将 activation 作为分数排名，取前几位，然后显示这些候选区域，自然也可以清楚明白，这个 feature 大概是什么。

R-CNN 作者将 pool5 作为可视化对象，它的 feature map 是 6x6x255 的规格，可以理解为有 256 个小方块，每个方块对应一个特征。

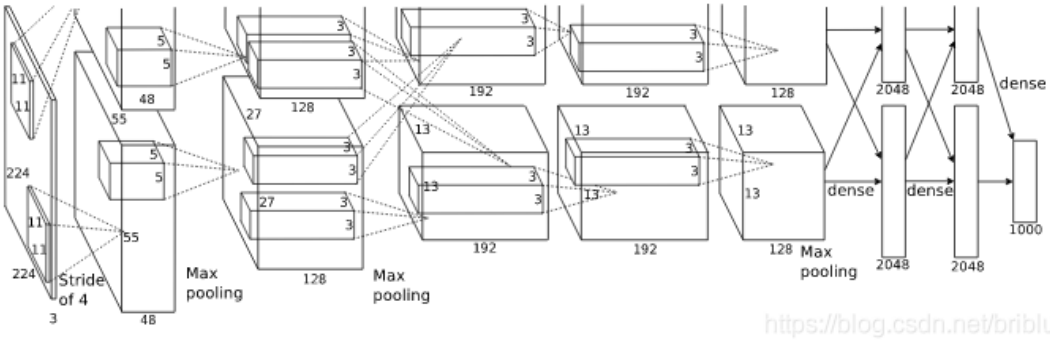
下面的图表中显示了这以可视化的效果，这里只显示了 256 个特征中的 6 个，每个特征取 activation 值最高的 16 个区域。



上图应该很明白了，对于同一类特征，activation 相差不大，这也是卷积神经网络能够准确识别物体的直观体现。

- **框架精简**

AlexNet 有 7 层，那么那些层是关键指标呢？哪些层可有可无呢？



pool5 在上一小节已经讨论过了，那么 fc6 和 f7 就成了研究的对象。fc6 与 pool5 构成全连接,为了计算 feature 它会乘以一个 4096x9216 的权重矩阵,然后在与一组 bias 相加，所以它有 3700 多万的参数。fc7 是最后一层，它的权重矩阵是 4096x409,它的参数有 1678 万多的参数。

但经过作者在 PASCAL 上不做 fine-tune 处理，直接测试，可以发现 fc7 的意义没有 fc6 大，甚至移除它之后，对于 mAP 结果指标没有影响。移除 fc7 就表示可以减少将近 1800 万个参数。

更惊喜的事情是，同时移除 fc6 和 fc7 并没有多大的损失，甚至结果还要好一点点。所以，神经网络最神奇的力量来自卷积层，而不是全连接层。

上面说的是没有 fine-tune 的情况，那么在 fine-tune 的情况是什么呢?

结果证明，fine-tune 后 fc6 与 fc7 提升的效果明显。

所以结论就是，pool5 从 ImageNet 训练集中学习了物体的泛化能力，而能力的提升则是通过特定领域的 fine-tune。

举个例子，神经网络在 ImageNet 数据集中学习到了 100 种猫的特征，而我自己的数据集只有两种猫，经过 fine-tune 训练后，这个神经网络可以更准确识别这两种猫了。

R-CNN 还与其他的特征方法进行了能力比较，作者选取了两种基于 DPM 的方法，DPM ST 和 DPM HSC,结果都证明，R-CNN 要好于它们。

- **目标检测错误分析**

R-CNN 作者采用了 Hoiem 提出的目标检测分析工具，能够直观地揭露错误的模型，作者通过这个工具针对性地进行 fine-tune。

- **bbox 回归**

bbox 的值其实就是物体方框的位置，预测它就是回归问题，而不是分类问题。受 DPM 的启发，作者训练了一个线性的回归模型，这个模型能够针对候选区域的 pool5 数据预测一个新的 box 位置。具体细节，作者放在补充材料当中。

- 语义分割

什么是语义分割？

区域分类技术是语义分割的标准做法，所以 R-CNN 也可以做语义分割，并且作者拿它跟 O2P 来比较。

R-CNN 进行语义分割分为 3 个阶段。

1. 利用 CPMC 生成候选区域，然后将这些区域调整大小为 227x227,送到神经网络当中，这是 full 阶段，区域中有背景也有前景。

2. 这个阶段只处理候选区域的前景，将背景用输入的平均值代替，然后背景就变成了 0 ，这个阶段称为 fg。
3. full + fg 阶段，将背景和前景简单拼接。

- 回顾
  1. R-CNN 采用 AlexNet
  2. R-CNN 采用 Selective Search 技术生成 Region Proposal.
  3. R-CNN 在 ImageNet 上先进行预训练，然后利用成熟的权重参数在 PASCAL VOC 数据集上进行 fine-tune
  4. R-CNN 用 CNN 抽取特征，然后用一系列的的 SVM 做类别预测。
  5. R-CNN 的 bbox 位置回归基于 DPM 的灵感，自己训练了一个线性回归模型。
  6. R-CNN 的语义分割采用 CPMC 生成 Region

R-CNN 灵活地运用了现有比较先进的工具和技术，并充分吸收，根据自己的逻辑改造，最终取得了很大的进步。

到 2018 年，R-CNN 已经不是最先进的目标检测模型，也不是最先进的语义分割模型，但这篇论文最大意义在于展示了作者在资源匮乏的情况下如何整合现有的先进技术去解决自己问题的手段。