基于卷积神经网络的空间不变无监督目标检测

Spatially Invariant Unsupervised Object Detection with Convolutional Neural Networks[[1]](#footnote-1)

# 原文翻译

摘要: 对任何智能系统来说, 获得解释物体能力都是非常必要的. 的确, 近期的机器学习会议上也有很多有关物体表示的应用的例子: 泛化和迁移到新任务上的能力, 可解释性等等. 然而, 为了解释物体, 系统需要一个在视觉世界里检测物体的办法: 无监督目标检测. 相比于这类方法的重要性来说, 其获得的关注实在是太少了, 尤其是在检测有许多物体的大图这样的任务上. 就我们的工作来说, 我们构建了一个能够处理多物体大型图片的神经网络架构. 实际上, 我们结合了AIR(Attend, Infer, Repeat)和近期的有监督目标检测的思路. AIR能够实现无监督目标检测, 但在物体的缩放上的处理不是很好. 我们把AIR的核心部分RNN替换成了CNN( 我们因此获得了空间不变性), 并且利用了一种由物体规范的模式来描述物体的位置, 使它以当前的网格为基准而非整幅图像. 经过一系列实验, 我们证实了我们这个架构的许多性质: 不同于AIR, 它能够在多物体大型图像上发现并检测物体; 它能够适应比训练图像拥有更多物体更大尺寸的图像; 他有足够的准确程度, 从而为下游任务创造良好的条件.

## 引言

物理世界能够被自然地分为若干个离散的物体. 因此, 要追求构建更智能的系统时, 设计解释和学习物体的方法应当十分重要. 确实, 近期的机器学习会议上也有很多有关物体表示的应用的例子. (Diuk, Cohen, and Littman 2008; Chang et al. 2016; Kansky et al. 2017; Santoro et al. 2017). 例如, 应用于物体的学习算法可以显著提高样本复杂度和泛化能力(Chang et al. 2016). 此外, 学习物体有关的知识有助于促进迁移学习(Kansky et al. 2017)因为世界正是由物体构成的, 新任务中往往包含了熟悉的物体. 在这样的背景下, 我们期望学习物体的算法能够有更高的迁移学习性能, 因为从原任务学到的物体级别的知识在目标任务上也很可能有用. 最后, 解释物体的机器学习模型的可解释性应当更好, 因为他们本就被设计为使用人类能够理解的表示方式.

对于直接接收像素作为输入的系统来说, 解释和学习物体需要把他们转换成更高级的对象表示. 以往的机器学习方法使用ad hoc, 特定任务或环境下的物体检测方法(Garnelo, Arulkumaran, and Shanahan 2016; Diuk, Cohen, and Littman 2008)或是直接使用环境提供的物体表示(Kansky et al. 2017). 或者, 如果能够使用标注了边界框的图像数据集, 人们也能够训练许多复杂精巧的有监督目标检测模型(Ren et al. 2015; Redmon et al. 2016). 然而, 这样的数据集往往是昂贵的. 因此, 我们对能够广泛通用的无监督目标检测方法非常感兴趣: 事实上就是没有标注边界框的数据集. 我们把这种发现数据集中有哪些共有的物体并且学习在图像中检测这些物体的任务称为无监督目标检测.

近期有名为AIR (Eslami et al. 2016)的方法显示出在无监督目标检测任务上的潜力. 然而(正如我们后文展示的那样)它处理多物体大图时表现不佳. 收到近期有监督目标检测的启发, 这篇论文中我们替代的结构来避免这种尺度问题. 这些工作主要依赖于空间不变计算, 尤其是卷积(LeCun et al. 1998; Long, Shelhamer, and Darrell 2015). 例如, YOLO使用了CNN来直接把输入图像映射到检测到的物体集合(Redmon et al. 2016).

如果一种计算对图像的不同区域应用完全相同的变换, 就称他为空间不变的. 无论是否是有监督的, 目标检测任务的许多特性使之适合空间不变计算. 一方面, 目标检测很大程度上只需要局部图像而非整张图片的信息. 其次, 在图像的指定子区域检测物体的能力应当也能在不同的区域上展现. 不利用空间不变性的方法, 比如训练一个直接从整幅图像到物体映射的MLP将因此丧失很大一部分性能.

在这些工作基础上, 我们提出了一种能处理多物体大图像的无监督目标检测方法SPAIR(**Sp**atially Invariant **A**ttend, **I**nfer, **R**epeat). 类似于YOLO, 它的核心是一个卷积物体检测器, 它负责把图像映射到物体. 为了无监督地训练这个物体检测器, 我们把它作为一个VAE (Kingma and Welling 2013)的编码器. 类似于AIR的策略, 我们通过重建图像来训练它. 大体上, SPAIR可以看作是一个有着特殊结构的VAE, 它能够潜在地表示物体, 且拥有一个能有效揭示物体结构的空间不变的卷积编码网络.

经过许多实验我们从经验上论证了, 应对包含较多对象的图像时, 同类的其他无监督目标检测方法效果较差, 而我们能够适应足够多的物体数目. 此外, 由于利用了空间不变性, 我们也发现了这种结构可以很好的应用于比我们的训练数据有着更多物体更大尺寸图片的数据集上. 最后, 我们也展示了我们的模型有着足够的可靠性给下游任务创造好的条件.

## 相关工作

此前的一种无监督物体检测的方法是AIR(Eslami et al. 2016). AIR设计了一种有着循环编码网络的VAE, 它也有着物体的潜在表示, 也有一个解码器来实现图像重建. 在每个处理步上, 循环编码器输入图像和隐藏状态, 输出物体位置大小和新的隐藏状态. 对每一张图片, 这个编码器都可以运行多个时间步; 显然, 每个时间步能输出一个物体, 而编码器通过隐藏状态来记录哪个物体已经被输出了. 若网络判断所有的物体都已经输出, 他就会停止计算. 最后由编码器接收这些物体信息并把它们重建回图像. 同时训练编/解码器使得重建图像接近原图. 通过这种训练过程, 编码器网络应当成为一个有效的物体检测器, 而这一切并不需要物体的边界标注.

然而, AIR有个缺点是他不能应对包含多物体的大图. 这是AIR的两方面实现造成的. 第一, 网络在每个时间步学习解释一个不同的物体, 但是假如有很多物体的话, 做到这一点就很难. Eslami et al. (2016)确实也没有在这方面测试AIR的适应性, 他们最多只试过图里有3个物体. 第二, 一个训练步(事实上是重建过程)的计算复杂度是, 其中是图像大小, 是检测到的物体个数. 假设物体密度固定[[2]](#footnote-2), 复杂度就达到了, 这样在大型图像上训练是不可行的.

近期的有关神经期望最大化(Greff, van Steenkiste, and Schmidhuber 2017; van Steenkiste et al. 2018)旨在完成与AIR类似的任务, 但也有着类似的问题. 计算机视觉领域的相关工作包括(Cao and Fei-Fei 2007; Kim and Torralba 2009; Lee and Grauman 2009; Russell et al. 2006; Karpathy, Miller, and Fei-Fei 2013), 以及共分割(Zhu et al. 2016)和无监督物体识别(Tuytelaars et al. 2010).

## 背景: VAE

VAE为我们的方法提供了理论基础. 尽管VAE可以用于各种数据形式, 我们主要关注用VAE建模图像.

假设图像从下面这样的通用模型中采样. 一个图像的潜在表示从先验分布中取样得到; 图像由一个似然分布产生. 这个似然分布通常是未知的, 于是我们用一个含参数的函数来近似它. 一张图像的概率由下式给出:

通常关注:

* 参数学习:
* 从一张图的隐变量里推测分布:

然而在许多场合, 计算是困难的. 这使得直接最大化难以实现, 因为它包含作为归一化常数. 在这样的条件下, 变分法提供了支持.

令为含参的, 将图像映射为隐变量z上的概率分布的函数; 我们使用来近似. 对任意的都有:

其中

()

由于KL散度总是正的, 就有. 称为ELBO. 接着在上最大化. 得到的记为. 此外, 由于使得我们指定的情况下最大化, 并且不会随着变化, 可得在上最小. 因此, 可以被作为的近似.

在VAE的框架下, 和都是神经网络, 且通过随机梯度上升最大化. 而(1)中的第二项通常能够近似计算, 第一项通常不能直接计算. 为此需要从中采样并估计. 采样操作下的反向传播可以由近年来的许多技术实现(Schulman et al. 2015).

VAE中的某些部分在标准的AE中有着对应. 可以被视为两项之和: 第一项促使图像精确重建(类似于标准AE中的重建训练), 第二项是KL散度正则化项. 此外, 和分别对应于标准AE中的编码器, 瓶颈层和解码器. 因此我们通常把称作编码器, 称作解码器.

## SPAIR

我们的模型是一个有着特殊结构的VAE, 它能够潜在地表示物体, 拥有一个卷积物体检测编码网络, 以及一个将识别到的物体重建回图像的解码器网络. 我们下面详细地解释这些组件.

### 物体表示

我们首先概述一下SPAIR在中间层描述物体的方式. 给定图像, 可以将它们分成个块. 其中, 和为表示块大小的固定常数. 我们使用一种允许每块一个物体的表示(改进为每块多个物体是简单的).

对块, 对应的物体表示为:

是一个维向量, 它存储物体的形貌. 是一个表示物体深度的实数: 在输出图像中, 更浅深度的物体叠在更深深度的物体之上. 是个二值量, 表示这个块中是否存在物体; , 其中由下式参数化物体的位置:

()

(3)

()

()

其中表示sigmoid函数, 为固定整数用于限制物体中心和块的距离. 图1对这些变量之间的关系作了说明. 由下式参数化物体的大小:

()

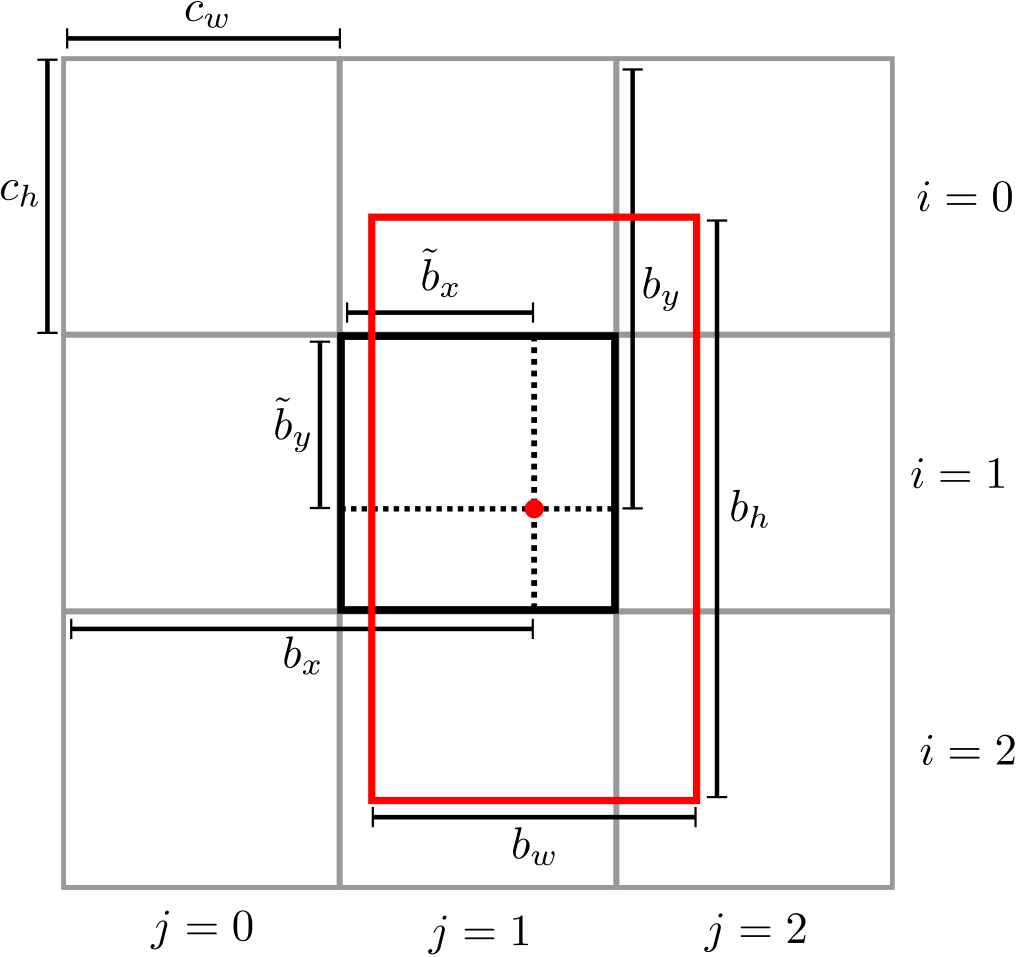


图 1 对SPAIR中物体边界框参数化的可视化解释. 其中黑框是我们所关注的块, . 红框表示物体边界框, 红点为中心. 式2~6表示了这些量之间的数量关系.

使用表示物体大小(而不是整幅图像的大小)的保证了无论处理何种大小的图像都是有意义的.

### 物体表示的先验分布

VAE的一个重要部分就是隐变量们的先验分布. 对所有实数量, 我们假设它们服从正态分布:

这里正态分布的参数是超参数.

对二值随机变量, 我们设计了一个先验, 它能迫使网络使用尽可能少的物体来重建网络(即尽可能少的), 这和AIR使用的先验类似. 这种策略对于网络从图像中提取高质量的物体表示是有用的; 如果没有它, 网络可以将所有的置1, 使用许多隐变量来解释图像中的每个物体.

我们首先把所有的整合为一个二值向量, 其维度为. 要生成, 我们首先从一个计数分布中采样一个非零正整数, 并依照均匀分布从含有个的所有向量中采样. 我们选择了在低值概率更大的分布; 事实上是个截断的几何分布. (的先验概率推导在原文附件, 对此的解读作为笔记, 见[2-1](#_截断几何分布先验概率推导))

### 编码器

我们的目的是训练具有空间不变性的编码器. 为此我们用一个CNN来把图像转化为特征向量, 其空间维度是, 也就是说每块一个特征向量. 接下来这个向量被从左上到右下按块处理来生成物体(见4.1).

上述处理是这样的: 首先, 一个MLP, 分别产生参数, 和, 这能够指示的分布. 接受当前快的特征向量和附近块已经采样的物体表示作为输入. 因此可以被视为包含了一组层内[[3]](#footnote-4)连接, 这促进了物体之间的相互调节. 在5.5中我们展示了这种连接对性能的影响.

接下来是采样:

其中采样值将被用于一个空域变换(Jaderberg et al. 2015)来从图像中提取一部分. 这一部分经过一个物体编码器, 产生分布的参数. 过程如下:

### 解码器

解码器将物体重建回图像. 首先, 一个物体解码网络为每个生成物体的形貌:

其中是的向量, 表示RGB的图像; 是的图像, 表示透明度[[4]](#footnote-5).

所有物体的形貌被合并到一起来形成最后的图像, 用经过空域变换来指定物体的形状和大小. 和相乘, 保证为0的物体不会出现在图像上. 是空间上重叠的物体的组合参数, 可以认为是一种对象相对深度的一种可微的近似. 重建的输出是图像, 相当于把参数化为一组独立的像素级的伯努利随机变量.

物体是在一个背景上重建的; 实际上可以使用另外一个神经网络来学习每张图的背景. 在我们的工作里, 我们使用了纯色背景; 可能是从训练集中采样出一部分统计出的颜色, 也可以用一个MLP来从每张图中映射出一个背景色出来.

### 训练

SPAIR通过ELBO(公式1)上的梯度下降来训练, 这其中有一些采样操作. 为了使得反向传播在采样操作上可行, 我们使用了重参数技巧(Kingma and Welling 2013). 对于正态分布变量, 这很简单. 而离散伯努利随机变量由伯努利的连续松弛[[5]](#footnote-6)变量代替, 在其上可以容易地实现重参数技巧(这个trick引于其他论文, 作为笔记见[2-2](#_重参数技巧)) (Maddison, Mnih, and Teh 2016; Jang, Gu, and Poole 2016). 在验证和测试阶段, 这些(连续松弛)变量由round操作离散化.

公式(1)也需要计算和之间的KL散度. 对实数域上的独立正态分布来说这是很简单的. 而伯努利/连续松弛变量稍有些困难. 详情见附件. (这个推导确实是复杂的, 而原文公开的代码在实现时使用的是一些更简单的办法)

### SPAIR的缩放特性

SPAIR设计上的某些方面对于缩放能力是很重要的. 考虑这样一种使得任何一种物体检测器都做到空间不变的简单方法: 让物体检测器处理固定的小尺寸图像, 当应对大尺寸图像时, 在适当大小且可重叠的局部小区域上多次运行检测器. 把各个位置检测到的物体汇总到一起就是整幅图片的物体集合. 假设物体都足够小能够处于一个小的局部区域内部, 那么这样就会比非空间不变方法企图一次处理整幅图像的做法要容易学习得多. 此外, 每次对局部物体的检测可以视作一个单独的例子; 我们可以认为这样物体检测器能有更多的训练数据.

在编码器中使用CNN基本上是一个实现这种简单物体检测的有效方法, 它输出层的每一个神经元都类似于一个局部物体的检测器.

正如2所讲述的那样, AIR在缩放能力上较差的原因是解码器将所有检测到的物体合并到一张图里时复杂度过高; SPAIR有一个类似的步骤, 因此并没有直接地避免这个问题. 然而, SPAIR的这一步在计算代价上的问题要小得多, 因为如同我们在5.2[[6]](#footnote-7)中描述的那样, 它能在大尺度图片上使用随机裁剪来训练, 而训练时仍然处理整幅图片.

# 原文及其他资料

* 原文<http://e2crawfo.github.io/pdfs/spair_aaai_2019.pdf>
* 论文中给出的代码实现<https://github.com/e2crawfo/auto_yolo>
* 附录<https://e2crawfo.github.io/misc/aaai_2019_supplementary.pdf>
* 离散重参数方法引文<https://arxiv.org/pdf/1611.00712.pdf>

1. AAAI- 19 [↑](#footnote-ref-1)
2. 考虑SPAIR的网格化特点, 这里”固定”指的数目可能是个. [↑](#footnote-ref-2)
3. 原文直译为”侧向连接” [↑](#footnote-ref-4)
4. 注: 这里大概是指可见度opacity, 原文使用的是transparency. [↑](#footnote-ref-5)
5. 原文为Concrete variables, 并没有找到合适的译法, 使用其解释(连续松弛)做代替 [↑](#footnote-ref-6)
6. 5及后续部分涉及具体实验流程、对比以及模型效果, 在理论上意义不如2~4, 而这里SPAIR为何能够在随机裁剪上训练其实说得很清楚了, 于是4之后的部分不再翻译 [↑](#footnote-ref-7)