

最近更新: 2019年9月20日

Learning to Segment via Cut-and-Paste

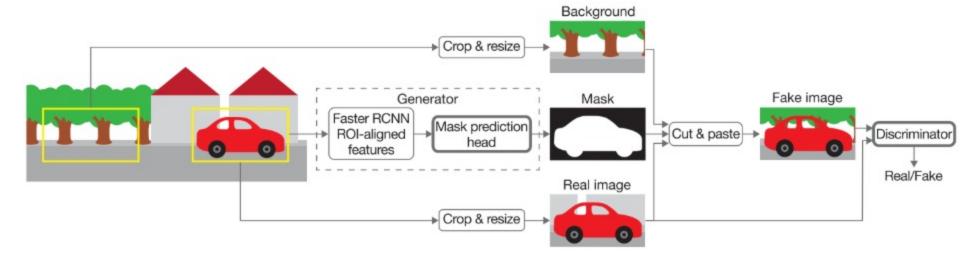
1. Introduction

同样是cut-paste方法,本文的落脚点不同,相比Cut, Paste and Learn的从realistic的角度探索增强,本文是将合成的图片通过生成器/判别器的训练得到mask的正确位置。

主要工作为:

- 提出一种基于cut-paste的对抗训练策略,在bbox的弱监督下实现实例分割任务
- 讨论了在何处放置paste的结果;提出了背景和物体之间的独立性,尽管实际物体和背景上下文之间是有关联的(基本是可以确定的),**这里却认为简单随机的位置放置也能有很好的效果**(应该就是实验没做到那,效果应该是有的)
- 弱监督分割模型的效果能达到全监督的90% (SDI号称95%, 但是是和DeepLabv1相比)

2. An Adversarial Formulation of the Cut and Paste Loss



思想

通过生成器生成更好mask来使得判别器更难判别,在对抗学习中提升两者的性能,逐渐获得更好的结果 [对抗学习的思想]

大致流程

采用GAN。分为两大部分:mask生成器和合成图像判别器。

- (1) 在图像上截取gt, 经过特征提取后预测gt的mask;
- (2) 在原图上随机cut一个背景图像,将mask提取出物体分割结果,然后paste到背景上去;
- (3) 合成的图像经过判别器进行判断

2.1 Where to Paste

尽管场景上下文约束一定是有影响的,作者认为似乎采用简单的随机paste也能达到不赖的效果。所以实际测试的 paste方法有两种: (1) 简单背景随机采样,只要不和bbox有重叠都可以 (2) 深度敏感的采样,一定程度地考虑合适 paste的位置

2.2 Avoiding Degenerate Solutions

作者确实心细,考虑到了会导致上述方法学习不收敛的两个极端情况,以及解决方案:

• 生成器选择0个像素作为mask

输出的mask为全不选,自然能在paste后得到具有极为realistic的patch,判别器根本分不清了,很容易致使收敛到这种局部极小导致网络根本学不到东西。解决办法也很直观,在上述生成器之后再加一个分类器,对mask内的物体进行分类判断。这样一来生成器生成的mask需要fool两个判别器。

• 生成器选择bbox内所有像素作为mask

一开始很懵,为什么选择bbox所有像素的paste结果会具有很大的误导性?答案是:通用目标检测的物体和box的IoU一般特别大,而且角度也不刁钻,将整个bbox进行paste后可以认为近似是物体segment的平移,也能取得不错的效果。

解决方法很简单:在进行cut&paste阶段时,crop的bbox额外加上10%宽度的上下文信息,之后paste到背景得到fake结果,这样一来更容易被判定fake;而正确的mask是不会受影响。

2.3 Adversarial Cut Loss

2.4 Overall Loss Function

3. Architecture

Output size	Layer
$7 \times 7 \times 2048$	Input, ROI-aligned features
$7 \times 7 \times 256$	Conv, $1 \times 1 \times 256$, stride 1
$7 \times 7 \times 256$	Conv, $3\times3\times256$, stride 1
$14 \times 14 \times 256$	Bilinear upsampling
$14 \times 14 \times 256$	Conv, $3\times3\times256$, stride 1
$28 \times 28 \times 256$	Bilinear upsampling
$28 \times 28 \times 256$	Conv, $3\times3\times256$, stride 1
$28 \times 28 \times 1$	Conv, $3\times3\times1$, stride 1
$28 \times 28 \times 1$	Sigmoid

Generator

Discriminator

Output size	Layer
$34 \times 34 \times 3$	Input image patch
$32 \times 32 \times 64$	Conv, 3×3×64, stride 1, valid
$15 \times 15 \times 128$	Conv, 3×3×128, stride 2, valid
$7 \times 7 \times 256$	Conv, 3×3×256, stride 2, valid
$3\times3\times512$	Conv, 3×3×512, stride 2, valid
4608	Flatten
2	Fully connected
2	Softmax

生成器和判別器的网络结构。生成器类似Mask RCNN,也有经过上采样再预测,因为生成mask需要精确的定位信息; 判別器就是个分类器。

4. Experiments

CityScapes

BOX是直接将bbox作为分割信息,可以作为比较的baseline。形状比较方正(物体与bboxloU大)的类别,Box的效果已经很好(如交通灯);对于比较复杂的person对象效果相差无几,倒是简单对象的差别大。可能是分割难度大,判别器过早无法区分达到收敛了,所以得分低。

其实这么比较有一点不太公平,Cut&Paste对每个类别都训练一个模型,这样必然是有优势的。

Method	Car	Person	Traffic- light	Traffic- sign
Box	0.62	0.49	0.76	0.79
GrabCut [I]	0.62	0.50	0.64	0.65
Simple Does It [2]	0.68	0.53	0.60	0.51
Cut&Paste (Ours)	0.67	0.54	0.77	0.79
FullySupervised	0.80	0.61	0.79	0.81

Table 2. mIOU performance on Cityscapes

• Effect of Pasting Strategy

关于paste方法作者论述和实验的不多。结论而言最后选择的是将bbox在其水平轴上以均值2W标准差W的范围(W为bbox的宽)进行滑动随机paste。同时还尝试将CityScapes粘贴到COCO上,结果不好,并且训练也更难收敛。(感觉是这个GAN的原因)

• Result

