

1. 这个 scale match 的从原理上来说就是很简单

出发点：首先这个 Tiny person 反馈对目标本身就在对质量就不多  
按照深度学习通用的做法，肯定是在比较  
通用的。有大量数据的数据库上进行一个训练  
然后再在我们这个特定的任务上进行一些微调

但是因为存在一个分布不匹配的问题，所以你这样  
做通常不能取得很好的效果

[预训练数据集的目标框都很大，实际  
任务数据集的目标框都很小]

所以他想了一个解决方法，应该是对于正常的  
的数据集，他都进行一个 Scale Match。  
把他们的 size 变成和目标数据集的 size  
一样。这样预训练以后再进行微调  
效果就会好很多。

scale match：scale match 的这个原理也很简单。  
就是以下三步

① 首先通过直方图的形式表示出  
Tiny person 中 object size 一个  
概率分布

② 其次对于预训练数据集里的每张  
图片，都会按照同一个分布采样一个  
 $\hat{S}$  size 出来

③ 进行实际缩放的步骤其实很简单

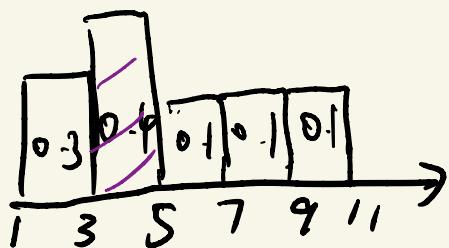
$$\text{ratio} = \frac{\text{采样出来的 size}}{\text{实际的 size}}$$

第①③步都很好说，第②步采样出来的这个 size 肯定要和 Tiny person 真正的分布越接近越好。

我理解的

- a. 就单次来说，采样出来的这个 size 应该还是不满足靠谱的，因为我的实际的 size 已经比较小了。但是你又采样出了一个更更小的 size，这样 scale match 以后我就把这个 size 变得特别特别小了。但是总的来说，从一大堆结果上去看，应该是和真实分布的结果很接近的。

从实际的代码上来看，逻辑应该是和下面一样的。



首先会在 0-1 之间进行一个采样，用那两个落在直方图的哪个 bin。比如随机采样，得到了 0.5。

在第一种概率分布中就落在了第二个 bin 上 (3-5)  
在第二种概率分布中就落在了第三个 bin 上 (5-7)

所以这种采样方法采样也是比较能贴近概率分布的。

⇒ 第二步就是在这个 bin 内，再通过均一采样，确定出一个更确切的 size

