简介

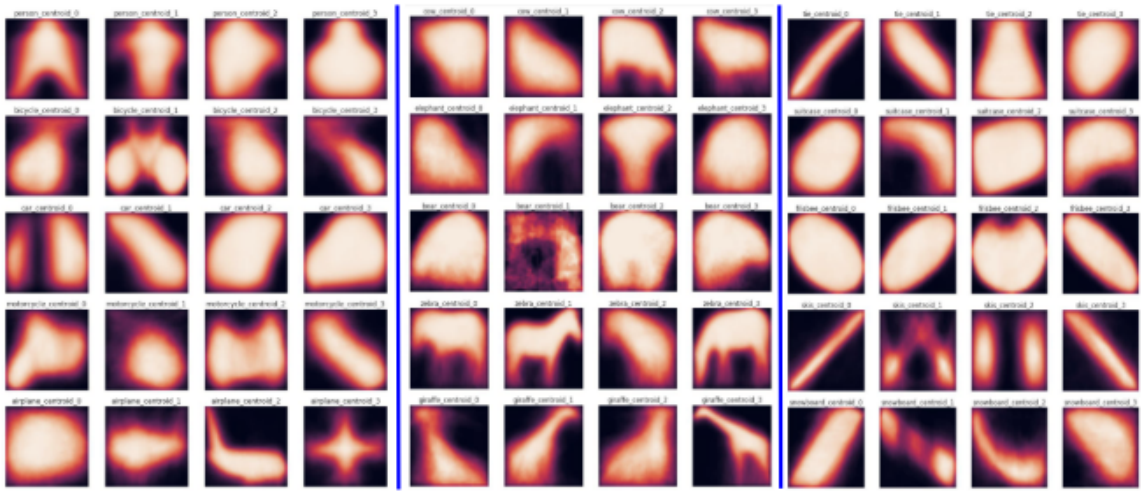
这篇文章用较短的时间实现了一种显著性分割策略，只要提供一个bbox就可以，加入了shape prior的先验信息，可以泛化到训练集当中不曾存在的类别。
shapemask可以产生bbox之外的mask，而且作者还设计了一些加速策略比如：

1. 不同于ROIAlign，直接采用simple cropping
2. 利用jitter ground truth去训练而不是detection的结果，这样就避免了训练时的NMS和Sorting
3. 利用RetinaNet来加速训练过程。

shape recognition

shape priors

由于很多类别虽然不同但是mask都是比较类似的，比如马和驴、骆驼等，首先作者对每个类别通过k-means找到k个centroids，对于class specific的任务，shape priors的总数为C x K，其中C是类别个数K是每个类别聚多少个类。而对于class agnostic任务，每个类别都聚一个类，总共聚K类（e.g. k=100），将所有的聚类结果放入一个data base H当中。最后在coco上的结果如下



shape estimation

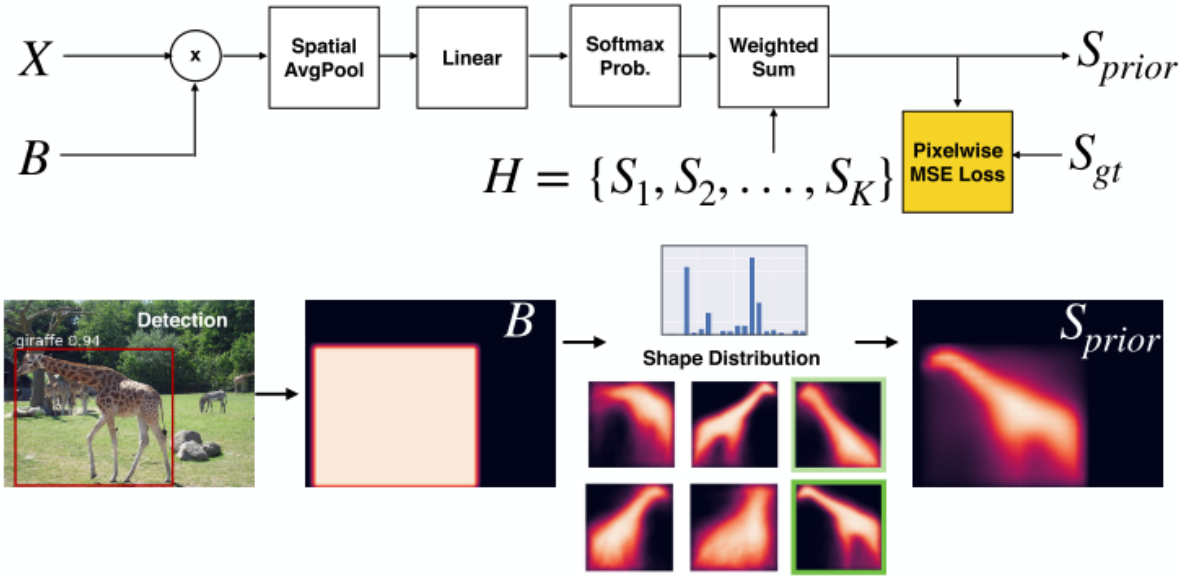
这里还蛮有意思，将原图像区域的一个roi拿出来，然后embedding成一个低维向量，这个向量就代表了每个shape prior的线性组合

$$x_{box} = \frac{1}{|B|} \sum_{(i,j) \in B} X_{(i,j)} \qquad (1)$$

首先这里的B其实是binary heatmap，其值只有0和1，取值是由box roi的区域决定的，

$$S = \sum_{k=1}^{K} w_k S_k \tag{2}$$

$$L_{prior} = MSE(S_{prior}, S_{gt}) \tag{3}$$



注意这里的X指的并不是图像，而是图像在进入网络后某一层的feature map。 作者认为这里加入了强先验知识，约束了解空间，避免其生成一些不可能存在的形状，而且这些约束对于没见过的形状其实是有更好的泛化能力的。

Coarse Mask Prediction

得到Sprior之后，利用1x1卷积将其和之前的feature相加，这样得到的结构就既拥有了image的信息和detection prior，因此可以利用这两个信息来生成出一个coarse mask。

$$X_{prior} = X + g(S_{prior}) \tag{4}$$

$$S_{coarse} = f(X_{prior}) \tag{5}$$

这里生成coarse mask的结构和maskRCNN的mask branch结构基本一样，用了四个卷积，不一样的地方在于用了detection prior Sprior来指导decoding的过程。最后求cross-entropy loss。

$$L_{coarse} = CE(S_{coarse}, S_{gt}) \tag{6}$$

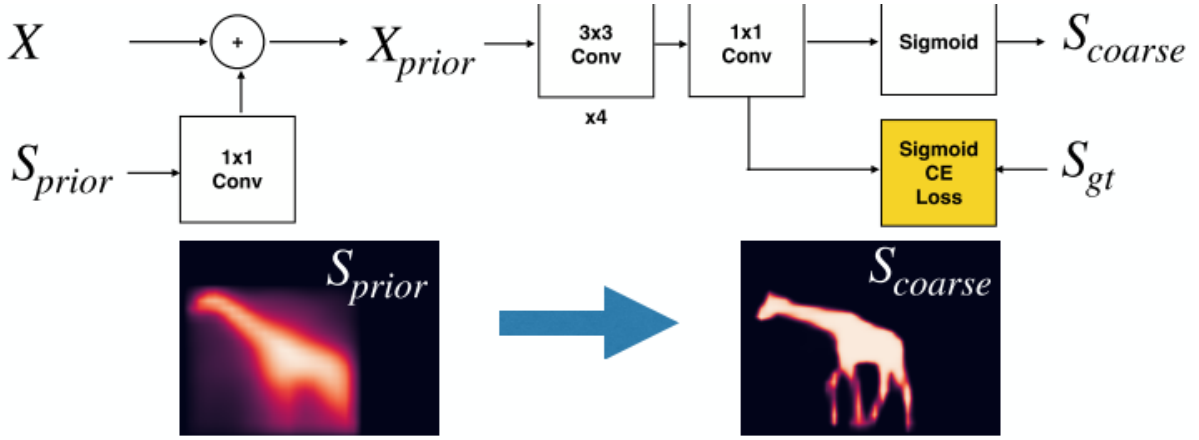


Figure 5: **Coarse Mask Prediction.** We fuse S_{prior} with the image features X to obtain prior conditioned features X_{prior}, from which we decode a coarse object shape S_{coarse}.

Shape Refinement by Instance Embedding Although

虽然生成的coarse mask已经包含了较强的物体形状信息，但一些instance-specific的细节信息还不充分，因此这一阶段的作用是对mask进行refine。 所以和之前的做法类似，同样利用soft coarse mask二值化然后计算得到1D的instance embedding x_{mask} 。计算方法为对coarse mask中的target feature做pooling。

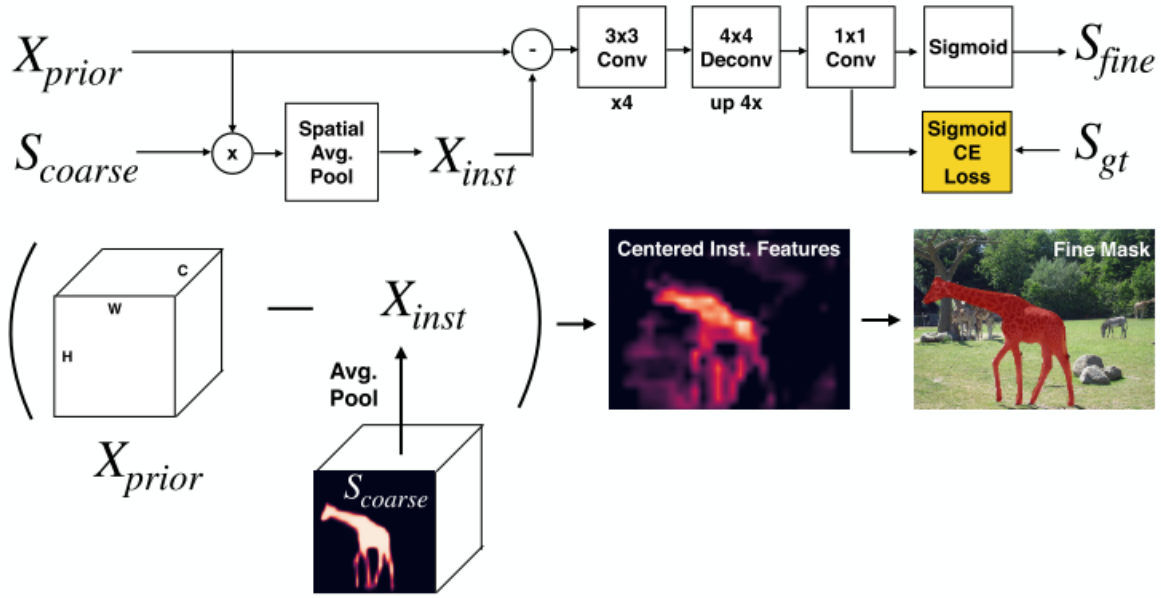
$$x_{mask} = \frac{1}{|S_{coarse}|} \sum_{(i,j) \in S_{coarse}} X_{prior(i,j)} \tag{7}$$

然后将上一步生成的 x_{priror} 减去mask。

$$X_{inst(i,j)} = X_{prior(i,j)} - x_{mask} \tag{8}$$

这样做可以看成将target instance作为image feature的一个条件。最后再进一个decoder的结构就行了，这里和之前docoder略有不同，多加了一个上采样层，生成更大的结果。

$$L_{fine} = CE(S_{fine}, S_{gt}) \tag{9}$$



实现细节

作者用了retinanet作为one stage detector,bbox做了Jitter处理，同时用FPN的P3到P5来进行roi的选取。
最后为了测试这个结果的泛化能力，划分了不同的训练集和测试集。
在训练过程中有一点值得注意，一般的instance segmentation网络的训练，会先经过一个region proposal的阶段，然后在相应的region上sample mask。这里作者为了实现one-stage training。在每张图片上随机sample 8个ground truth mask去训练集，这样就避免了region proposal的阶段，可以进行one stage训练。
再Roi feature的选取时，并没有采用roiAlign等类似的方法，而是先用公式计算出roi feature应该在哪一层:

$$k = m - \lfloor \log_2 \frac{L}{\max(box_h, box_w)} \rfloor, \tag{10}$$

其中L是原始图像的输入尺寸（文中为1024），m是特征金字塔的最高层(m=5)。在目标指定分配的特征层中，作者没有采取类似ROIAlign的crop and resize方法，而是直接取一个cxc的patch，为了保证每一层取的patch都能完全覆盖到目标物体，作者取 $c = L/2^m$ 。
然后利用1x1卷积将feature channel从256降到128，shapeMask算法的输入X其实就是这里的feature patch X。

结果分析

首先是半监督的实验中，本文提出的网络结构仅靠1/100的训练集就能比Mask rcnn的结果高出2ap。
在全监督的实验中，用Mask RCNN的backbone比MRCNN高出1.7AP

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Training (hrs)	Inference (X + Y ms)	GPU
FCIS+++ [27] +OHEM	ResNet-101-C5-dilate	33.6	54.5	-	-	-	-	24	240	K40
Mask R-CNN [18]	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4	44	195 + 15	P100
Detectron Mask R-CNN [14]	ResNet-101-FPN	36.4	-	-	-	-	-	50	126 + 17	P100
ShapeMask (ours)	ResNet-101-FPN	37.4	58.1	40.0	16.1	40.1	53.8	11*	125 + 24	V100
Mask R-CNN [18]	ResNext-101-FPN	37.1	60.0	39.4	16.9	39.9	53.5	-	-	-
MaskLab [5]	Dilated ResNet-101	37.3	59.8	39.6	19.1	40.5	50.6	-	-	-
PANet [33]	ResNext-101-PANet	42.0	65.1	45.7	22.4	44.7	58.1	-	-	-
ShapeMask (ours)	ResNet-101-NAS-FPN [12]	40.0	61.5	43.0	18.3	43.0	57.1	25*	180 + 24	V100

Table 2: ShapeMask Instance Segmentation Performance on COCO. Using the same backbone, ShapeMask outperforms Mask R-CNN by 1.7 AP. With a larger backbone, ShapeMask outperforms Mask R-CNN and MaskLab by 2.9 and 2.7 AP respectively. Compared to PANet, ShapeMask is only 2.0 AP behind without using any techniques reported in [33, 5]. This shows that ShapeMask is competitive in the fully supervised setting. Timings reported on TPUs are marked with star signs. Inference time is reported following the Detectron format: X for GPU time, Y for CPU time. All mask APs are single-model, and are reported on COCO test-dev2017 without test time augmentation except Detectron on val2017 (gray).

作者提出的两个模块，shape priorh和embeding策略，在半监督学习过程中作用非常明显。

		VOC → Non-VOC			Non-VOC → VOC		
Shape	Embed.	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
		13.7	28.0	12.0	24.8	45.6	23.5
	✓	26.2	44.6	27.1	29.4	51.7	29.0
✓		26.4	44.9	27.2	30.6	53.4	30.4
✓	✓	30.2	49.3	31.5	33.3	56.9	34.3

Table 4: Ablation results for the partially supervised model.