工作汇报

2019-10-30

Ying-Peng Tang

目录

Проблем сеттинг

- Training an effective detector often requires plentiful labeled data which is rather expensive and time consuming to obtain
- Can not require more data from target domain
- Find a source domain whose label space is identical with the target domain is difficult
- We assume a complicated and large enough unlabeled source domain to make sure its label space includes the target classes.
- Query from the source domain to boost the performance of target task.

Кеы идеа

- We use one-stage detector. It will predict for each anchor box, which is more favorable to active learning.
- To close the domain gap, we apply the Adversarial Domain Adaptation to the model.
- We query the specific box instead of the whole image.
- We propose a method to train detector from partly labeled image.

Адверсариал Домаин Адаптатион

$$= \frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} L_y \left(G_y \left(G_f \left(\mathbf{x}_i \right) \right), y_i \right)$$

$$- \frac{\lambda}{n_s + n_t} \sum_{\mathbf{x}_i \in \mathcal{D}_s \cup \mathcal{D}_t} L_d \left(G_d \left(G_f \left(\mathbf{x}_i \right) \right), d_i \right)$$

L: loss

G_f : Feature extractor

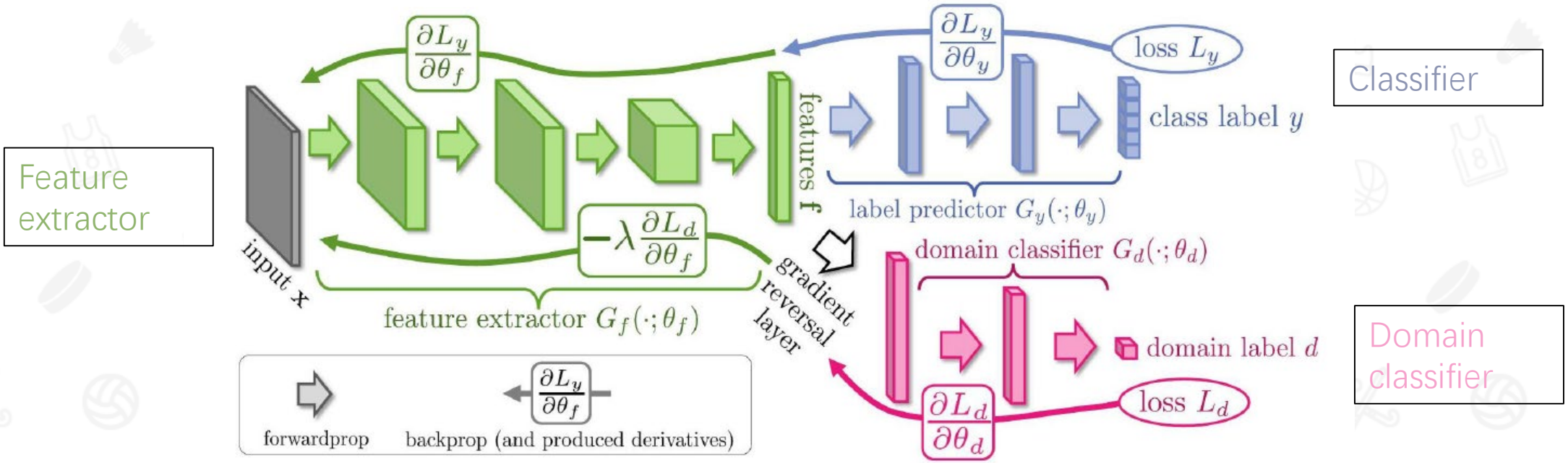
G_y : Classifier

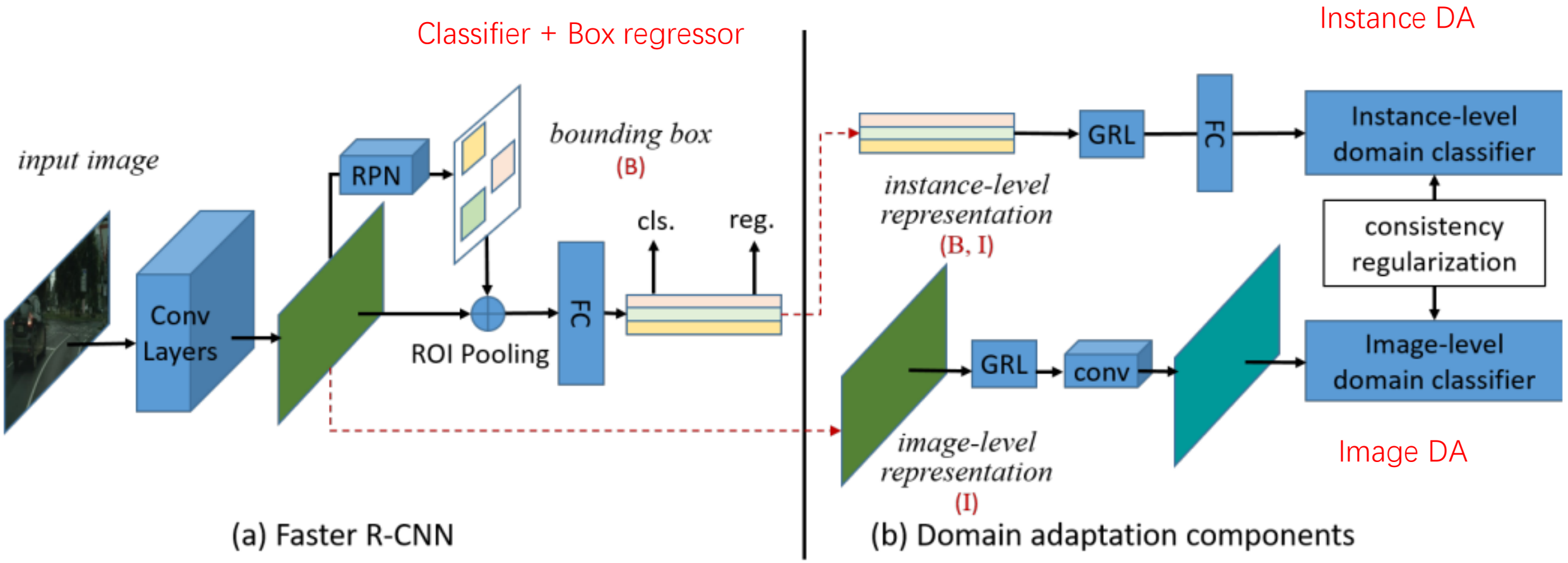
G_d : Domain classifier

x, y: example, label

d: domain label (source 1, target 0)

n: number of examples



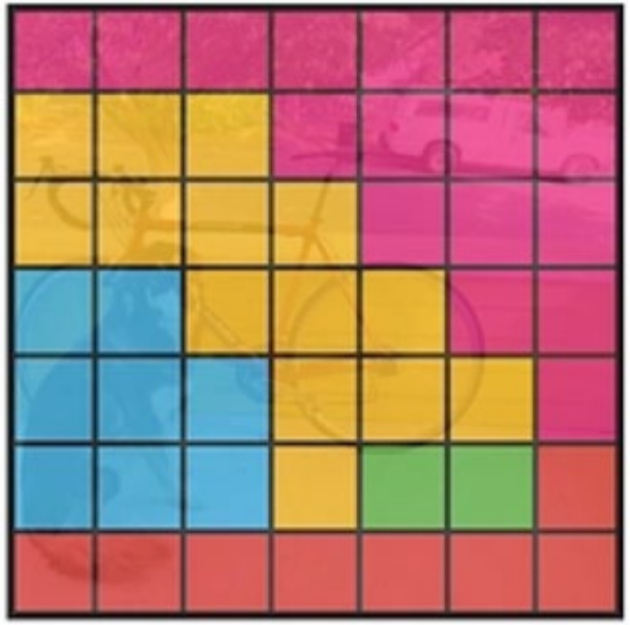


The model we used has 3 components: Domain Adapter, Classifier, Box Regressor.

We want to get a high performance model in target domain by qureying the least data from source domain.

Инстанце ДА фор ЫОЛО

S × S grid on input



Class probability map

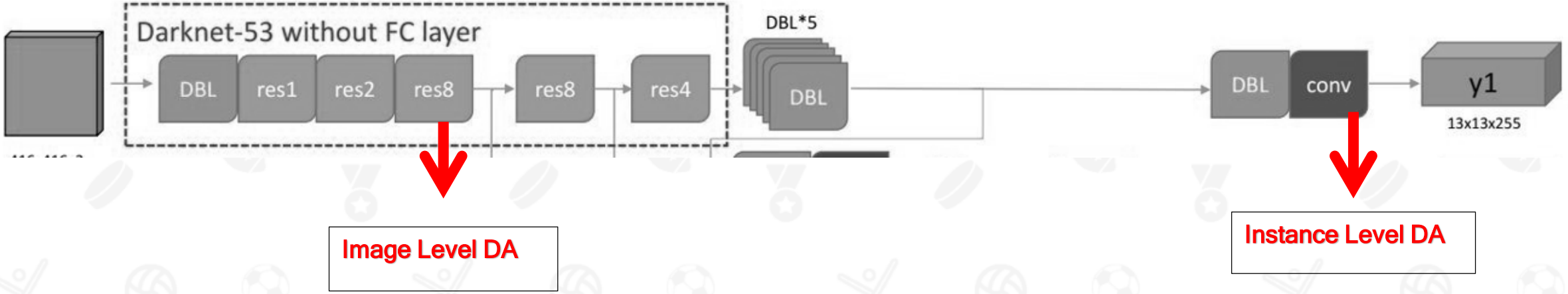
YOLO prediction: S x S x 256 → S x S x 3 x (4 + 1+ class)

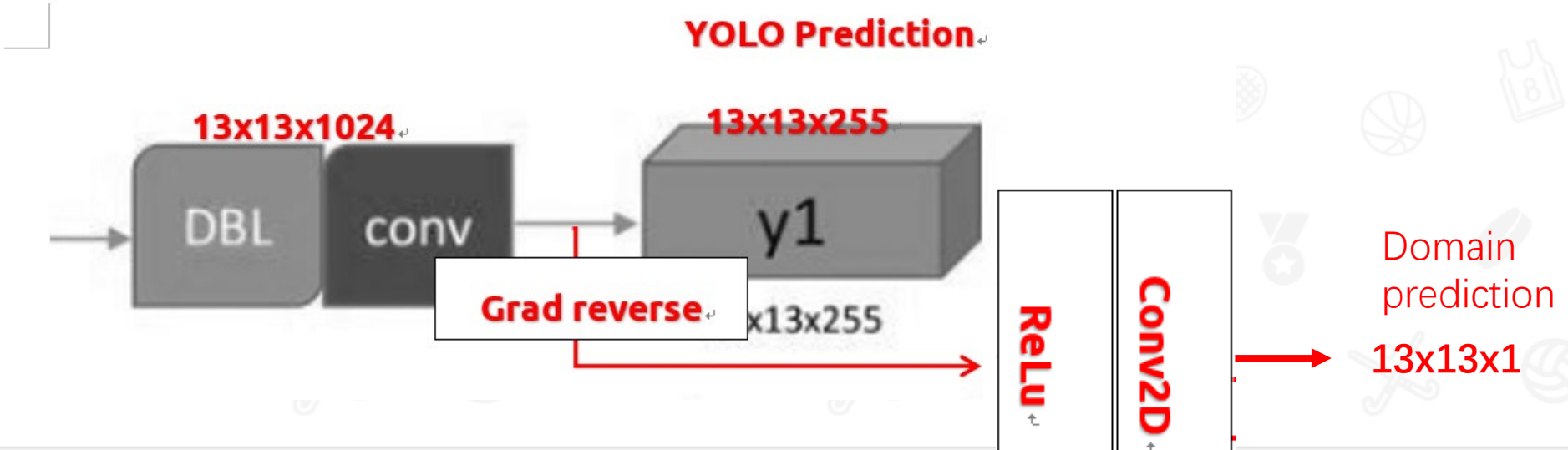
框坐标

前景的概率

每一类的概率

Инстанце ДА фор ЫОЛО





Валидате ДА

在传统的设置下（target无监督，source有监督）进行了多个对比实验，并调了一下DA的loss的权重。我认为instance DA只计算有物体的grid的损失，而一张图中有物体的grid非常少，这部分损失可能会被dominate，因此可以增大他的倍率，结果如下。（0x代表不参与训练，1x代表loss乘1倍）

Image DA	Instance DA	mAP (%)
0x	0x	14.297
0x	1x	14.375
1x	0x	14.679
1x	1x	17.897
0x	2x	16.126
1x	2x	17.573
0x	5x	16.890
1x	**5x**	**19.301**
5x	0x	15.887
5x	5x	17.340

Кеы идеа

- We use one-stage detector. It will predict for each anchor box, which is more favorable to active learning.
- To close the domain gap, we apply the Adversarial Domain Adaptation to the model.
- We query the specific box instead of the whole image.
- We propose a method to train detector from partly labeled image.

More 'active'

- A more 'active' way
- Only query the most informative instance, reduce the labeling cost.
- A more chanlenge setting (has to evaluate the informativeness and transferrability for each anchor box whose amount is extremely large)

Чуеры стратегы

主要考虑了instance的2个方面：

Transferability: 利用DA模块的输出选出与目标域分布接近的源域样本，以及避免选到Outlier classes.

Inconsistency: 利用object conf 和classification conf的不一致性选出信息量最高的框。

Transferability

Source domain label: 1

Target domain label: 0

For source domain example:

$if\ D^*(z) \approx 1$ highly likely come from the outlier classes

$if\ D^*(z) \approx 0$ more likely come from the shared classes

$$\tilde{w}(\mathbf{z}) = 1 - D^*(\mathbf{z})$$

Inconsistency

YOLO prediction: 4 + 1+ class

框坐标 前景的概率 每一类的概率

Objectness	Classification	
High	Unsure	Uncertain
Low	Confident	Hard negative

Инцонсистенцы

分类的不确定度：Best vs the 2nd Best

$$p_{cla}(x_{i,j}) = P(\hat{y_1}|x_{i,j}) - P(\hat{y_2}|x_{i,j})$$

不一致性：取两个概率的1-范数

$$D_{i,j} = |p_{cla}(x_{i,j}) - p_{obj}(x_{i,j})|$$

为了查到更多前景，提供更多信息， 对置信度低的框做一定的抑制，k=0.05

$$D_{i,j} = p_{obj}(x_{i,j})^{\frac{1}{k}} |p_{cla}(x_{i,j}) - p_{obj}(x_{i,j})|$$

Чуериед бошес

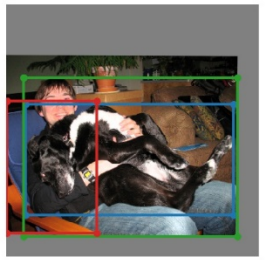
- 使用随机方法以图片为单位查询，大约能查 4K 张左右
- 我们的方法能查到 18376 张图片，其 中 16953 个 positive instances，6387 个 negtive instaces，7403 个 outlier instances

- 我们的方法还是能选出非常多的 target domain class 。
- outlier instances 的数量与 negtive 相当，说明还是能比较准的找到前景框，
- Positive instance 的数量远超outlier classes，说明transferability 对选 target class 的框还是有一些作用的。
- 我们的方法平均每张图片会查到1.67个框，说明模型确实只对某一些区域不确定，查询整张图片会浪费cost

查到的正样本确实是比较难学的样本(框为 GT 框)

COCO_train2014_000000530535.jpg


COCO_train2014_000000074947.jpg

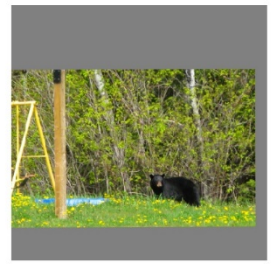

COCO_train2014_000000355778.jpg


COCO_train2014_000000390935.jpg


COCO_train2014_000000488747.jpg

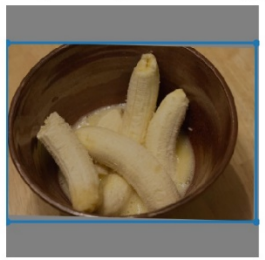

COCO_train2014_000000071970.jpg

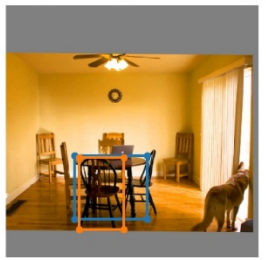

COCO_train2014_000000550338.jpg


COCO_train2014_000000003481.jpg


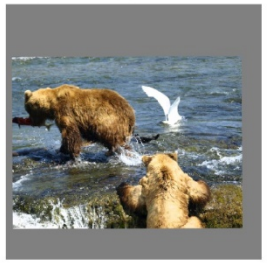
COCO_train2014_000000069480.jpg

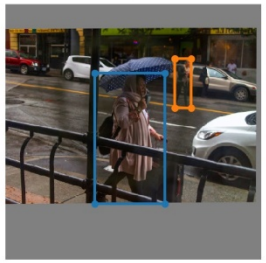

COCO_train2014_000000462968.jpg


COCO_train2014_000000271416.jpg


COCO_train2014_000000199923.jpg


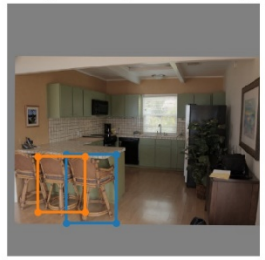
COCO_train2014_000000134502.jpg


COCO_train2014_000000346678.jpg


COCO_train2014_000000574562.jpg


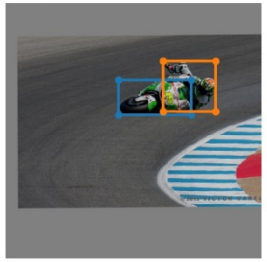
COCO_train2014_000000298199.jpg


COCO_train2014_000000570775.jpg

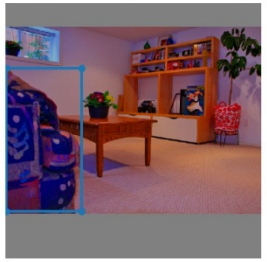

COCO_train2014_000000363190.jpg


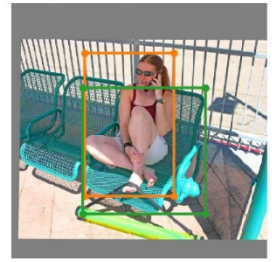
COCO_train2014_000000234350.jpg


COCO_train2014_000000561436.jpg


COCO_train2014_000000400734.jpg


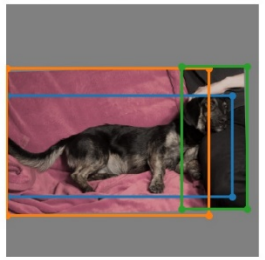
COCO_train2014_000000180139.jpg


COCO_train2014_000000199458.jpg


COCO_train2014_000000208612.jpg


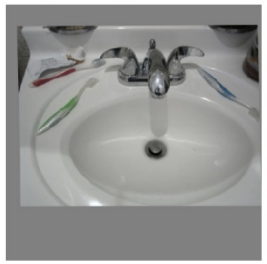
COCO_train2014_000000385379.jpg


COCO_train2014_000000511734.jpg


COCO_train2014_000000284568.jpg


COCO_train2014_000000138507.jpg


COCO_train2014_000000572362.jpg


COCO_train2014_000000565857.jpg


COCO_train2014_000000050563.jpg


COCO_train2014_000000235237.jpg


查到的负样本确实是比较又迷惑性的框（框为预测值）

COCO_train2014_000000530535.jpg

COCO_train2014_000000074947.jpg

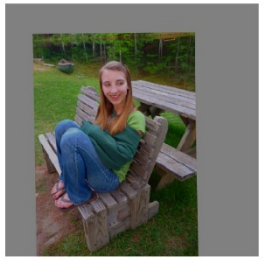
COCO_train2014_000000355778.jpg

COCO_train2014_000000390935.jpg

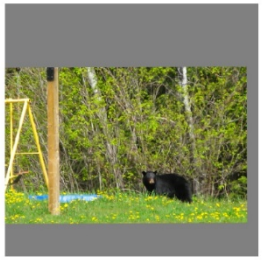
COCO_train2014_000000488747.jpg

COCO_train2014_000000071970.jpg

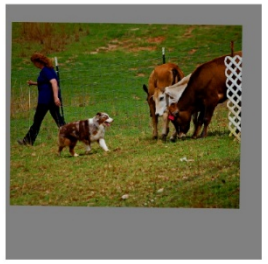
COCO_train2014_000000550338.jpg



COCO_train2014_000000003481.jpg



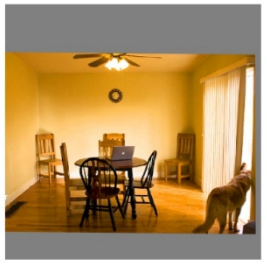
COCO_train2014_000000069480.jpg



COCO_train2014_000000462968.jpg

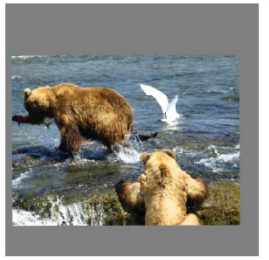
COCO_train2014_000000271416.jpg

COCO_train2014_000000199923.jpg

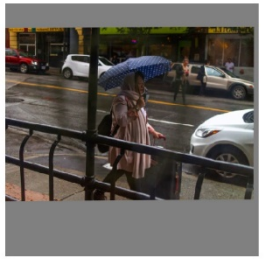


COCO_train2014_000000134502.jpg

COCO_train2014_000000346678.jpg

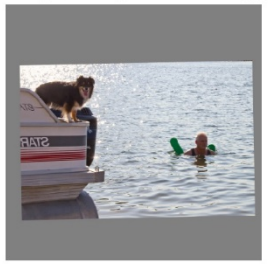


COCO_train2014_000000574562.jpg

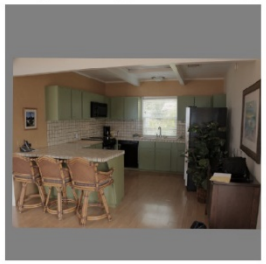


COCO_train2014_000000298199.jpg

COCO_train2014_000000570775.jpg



COCO_train2014_000000363190.jpg



COCO_train2014_000000234350.jpg

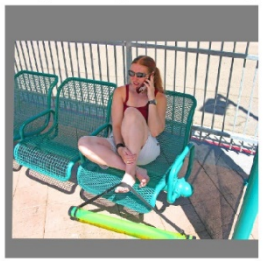
COCO_train2014_000000561436.jpg

COCO_train2014_000000400734.jpg

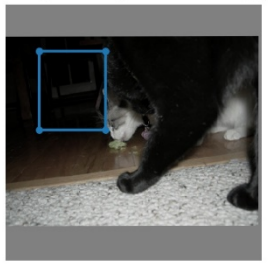
COCO_train2014_000000180139.jpg

COCO_train2014_000000199458.jpg

COCO_train2014_000000208612.jpg



COCO_train2014_000000385379.jpg



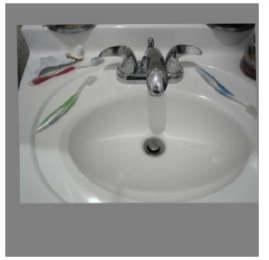
COCO_train2014_000000511734.jpg

COCO_train2014_000000284568.jpg

COCO_train2014_000000138507.jpg

COCO_train2014_000000572362.jpg

COCO_train2014_000000565857.jpg



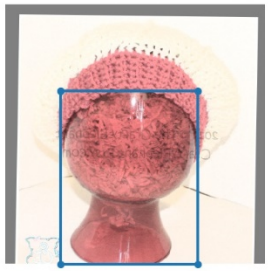
COCO_train2014_000000050563.jpg

COCO_train2014_000000235237.jpg

查到的outlier classes是一些类别与target classes相近的框 (框为 预测 框)

COCO_train2014_000000530535.jpg

COCO_train2014_000000074947.jpg



COCO_train2014_000000355778.jpg

COCO_train2014_000000390935.jp

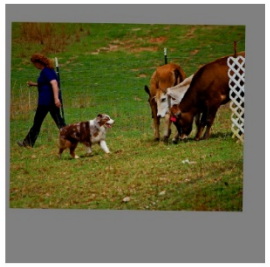
COCO_train2014_000000488747.jpg

COCO_train2014_000000071970.jpg

COCO_train2014_000000550338.jpg

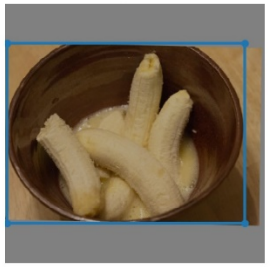
COCO_train2014_000000003481.jpg

COCO_train2014_000000069480.jpg

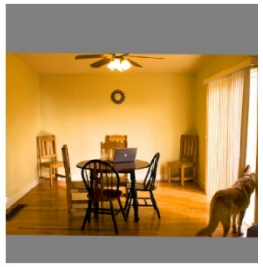


COCO_train2014_000000462968.jpg

COCO_train2014_000000271416.jpg

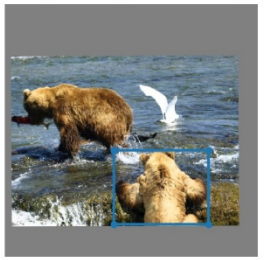


COCO_train2014_000000199923.jp

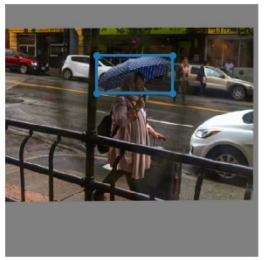


COCO_train2014_000000134502.jpg

COCO_train2014_000000346678.jpg



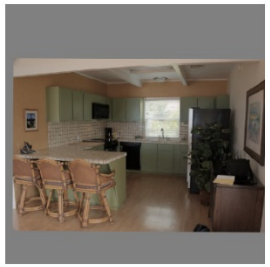
COCO_train2014_000000574562.jpg



COCO_train2014_000000298199.jpg

COCO_train2014_000000570775.jpg

COCO_train2014_000000363190.jpg



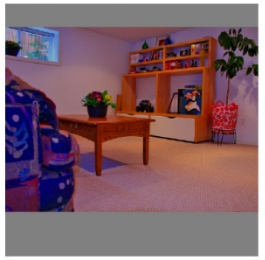
COCO_train2014_000000234350.jpg

COCO_train2014_000000561436.jp

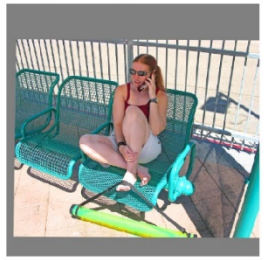
COCO_train2014_000000400734.jpg

COCO_train2014_000000180139.jpg

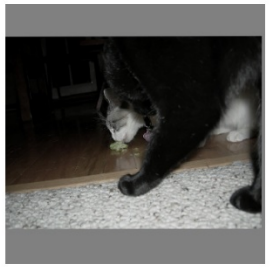
COCO_train2014_000000199458.jpg



COCO_train2014_000000208612.jpg

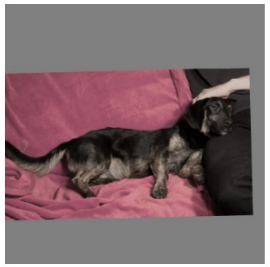


COCO_train2014_000000385379.jpg

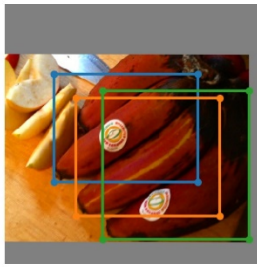


COCO_train2014_000000511734.jpg

COCO_train2014_000000284568.jpg



COCO_train2014_000000138507.jp



COCO_train2014_000000572362.jpg

COCO_train2014_000000565857.jpg

COCO_train2014_000000050563.jpg

COCO_train2014_000000235237.jpg

Траининг щитх партлы лабелед дата

Yolo的loss分为3部分:

1. 分类损失：使用交叉熵损失计算与前景框交叠率大于0.5的框的分类损失，标记向量为对应的类别那一维是1，其余是0
2. 前背景损失：计算所有框的前背景预测损失，与前景框交叠率大于0.5的框标记为1，否则认为是背景，标记为0
3. 框回归损失：计算与前景框交叠率大于0.5的框的框回归损失，使用平方损失

对于partly labeled data，我们正常使用查到的前景计算其分类损失和框回归损失，背景框参与计算前背景损失，未知部分不参与loss计算；Outlier classes的框认为是前景，计算前背景损失，但乘以系数0.3 （因为只知道这个框与一个outlier classes的GT框交叠率大于0.3）

训练方式：

在一开始源域和目标域样本都充分的时候，各取一个batch出来训练，然后更新一次梯度，等到一个域的样本用完，就只用另一个域的养样本训，直到所有样本都过一遍。

Проблем

- 使用partly labeled data 训练模型分类损失与前背景损失非常大，与正常训练产生的损失量纲不一致，训练出来的模型效果也很差，大约30%的mAP（初始模型为59.1%）

尝试

- 乘以一个权重（等价于调整学习率）使得这部分损失与正常训练量纲一致　mAP变化不大,Precision 很低
- 一上来loss太大，采用warm up的策略，使learning rate 慢慢增大　mAP变化不大,Precision 很低，只有0.08
- 去掉背景框的损失计算　mAP变化不大,Precision 更低，只有0.01
- 去掉所有框的objectness损失计算　mAP变化不大,F1正常，怀疑是类别不均衡
- 由于源域样本多于目标域，到后面会只用源域样本训练，导致模型过多的拟合这些样本，可以尝试当目标域样本用完之后，再从头开始迭代，使得目标域和源域样本总能组成pair来训练模型　mAP变化不大

性能差的原因所在

- Loss大到量纲不一致，等价于学习率过高
- 由于前景框居多，位置部分不参与loss计算，使得模型为了拟合前景框，将所有图片的objectness预测都提高了，从而引入了很多False positive，导致precision 很低
- 查到的类别不平衡，导致有的类AP很低，所有平均AP也很低

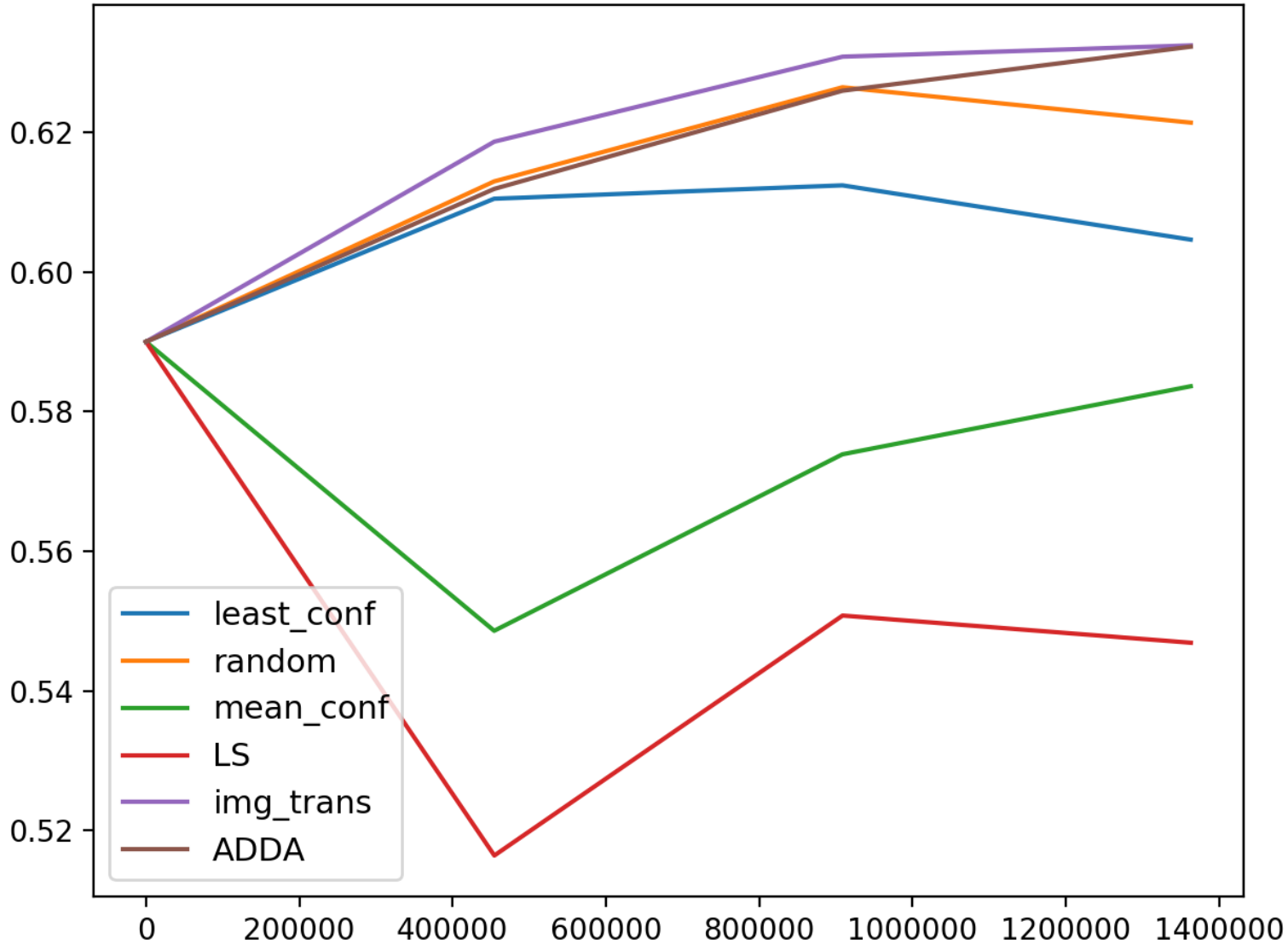
查到的instance类别统计

167 , 582, 264, 201, 66, 253, 1107, 370, 1667, 483, 769, 620, 768, 1266, 6824, 106, 203, 929, 133, 121

每一类的AP

0.45 0.68 0.42 0.32 0.25 0.65 0.66 0.73 0.35 0.57 0.51 0.65 0.72 0.71 0.67 0.27 0.53 0.58 0.65 0.56

Перформанце оф цомпаред метходс



- Uncertainty效果不太好，似乎源域中的不确定样本对于目标域的训练帮助不大，甚至有副作用

计划尝试

- 加一项正则化项，比如，让没查到的区域的值与上一轮模型的预测值接近
- 使用模型预测的非常确定的背景框，拿来当成GT参与优化
- 将查到的框，以及框外围一圈抠出来，当做一张正常的图片
- 对于类别不平衡的问题，尝试为少数类加一个比较大的权
- 调参：调整partly labeled data loss的权值，AL选样本的tradeoff等