Efficient OD - WACV - 2020

coarse level: 隐变量

HR img: $x_H = (x_H^1, x_H^2, \cdots, x_H^{P_c})$ 、P_c: patch num of coarse

$Y = \{Y_1, Y_2, \cdots, Y_{P_c}\}$ 对应 bounding box (ground-truth)

$Y_i = \underline{\{y_1, y_2, \cdots y_{P_f}\}}$ 每个 Y_i 中划分 P_f 个框.

$(\underbrace{g_x, g_y}_{\text{centroid}}, \underline{w, h}, \underbrace{c}_{\text{object cls}})$

LR img

r.v. $x_L = (x_L^1, x_L^2, \cdots, x_L^{P_c})$

两步 MDP

代理: 观察 x_L、输出 a_c 策略

Action likelihood function of policy networks

$\pi_c(a_c | x_L; \theta_p^c) = \underline{P(a_c | x_L; \theta_p^c)}$ 映射到概率分布

$\{0,1\}^{P_c} \in \mathbb{R}^{10}$

第 i 个元素代表是否要 x_H^i $= \prod_{i=1}^{P_c} S_c^i (1-S_c^i)^{(1-a^i)}$ 伯努利分布

$S_c = f_p^c(x_L; \theta_p^c)$

$P(x_H, x_L, Y, a_c)$

$= P(x_H) \cdot P(Y | x_H) \cdot P(x_L | x_H) \cdot \underline{P(a_c | x_L; \theta_p^c)}$

代理: 目标检测策略.

$\pi_d(a_d | x_L^i; \theta_d^c) = P(a_d | x_L^i; \theta_d^c)$ coarse

$\pi_d(a_d | x_H^i; \theta_d^f) = P(a_d | x_H^i; \theta_d^f)$ fine

对不同精细度图像检测.

从 x_H^i, x_L^i 中选择 patch 得到 a_d.

$\hat{Y}_i = \{\hat{y}_1 \ldots \hat{y}_{P_f}\}$ 预测 bbox 值

目标

$\underbrace{J_c(\theta_p^c, \theta_d^f, \theta_d^c)}_{\text{三策略}} = \mathbb{E}_p[R_c(a_c, a_d, Y)]$

尽可能接受 patch

$\max_{\theta_p^c, \theta_d^f, \theta_d^c} J_c(\theta_p^c, \theta_d^c, \theta_d^f)$

KOKUYO

No. 相当于 Img patch 重新划分(?)

P_c = patch coarse num

P_f: patch fine num

fine-level: 重力 pixel. θ_p^f = FPNet 来检测.

从 $\boxed{x_H^i = (x_h^1, x_h^2, \dots, x_h^{P_f})}$ 采样 x_L^i

其中包括 P_f 个重叠、精细 subpatch 子块

$a_f = \{0,1\}^{P_f}$

相当于原先的 x_H^i 再细分成 P_f 个小块、然后随机采样

可重叠

$$p(x_H^i, x_L^i, Y^i, a_f) = p(x_H^i)\cdot p(Y^i \mid x_H^i)\cdot p(x_L^i \mid x_H^i)\cdot p(a_f \mid x_L^i;\theta_p^f)$$

事实上,我不懂为什么要写这个联合分布.感觉也没有用…

$$\max_{\theta_p^f,\theta_d^f,\theta_d^c} J_f(\theta_p^f,\theta_d^f,\theta_d^c)$$

$$= \mathbb{E}_p[R_f(a_f, a_d, Y_i)]$$

目标检测得到 a_d

x_L^i, x_h^i 来自 X_H^i、X_L^i 再次细分得到

观察 x_L^i,输出 a_f 策略

$$\boxed{\pi_f(a_f \mid x_L^i, \theta_p^f)} = \prod_{j=1}^{P_f} s_f^i (1-s_f^i)^{(1-a_f^i)}$$

Bernolli 分布.

$$s_f = f_p^f(x_L^i;\theta_p^f)$$

$\in [0,1]$

问下GPT: f_p^c、f_p^f 是什么?
哪里的网络?

Q: 对随是为了处理
何来引的指数?

训练过程: θ_p^{c*}、θ_p^{f*}

policy gradient
策略梯度

$$\nabla_{\theta_p^c} J_c = \mathbb{E}[R_c(a_c, a_d, Y)\nabla_{\theta_p^c} \log \boxed{\pi_{\theta_p^c}(a_c \mid x_L)}]$$

$$\pi_c(a_c \mid x_L;\theta_p^c)$$

$$\nabla_{\theta_p^f} J_f = \mathbb{E}[R_f(a_f, a_d, Y_i)\nabla_{\theta_p^f}\cdot \log \boxed{\pi_{\theta_p^f}(a_f \mid x_L^i)}]$$

$$s = as + (1-a)(1-s)$$

$$= \mathbb{E}[\underline{A}\sum_{i=1}^{P_c}\nabla_{\theta_p^c}\cdot \log(s_c^i a_c^i + (1-s_c^i)\cdot(1-a_c^i))]$$

$$\pi_f(a_f \mid x_L^i;\theta_p^f)$$

$$A(a_c, \hat{a}_c, a_d, \hat{a}_d) = R_c(a_c, a_d, Y) - R_c(\hat{a}_c, \hat{a}_d, Y)$$

Campus

避免训练过拟合震动.

No.

将 $R_c(a_c, a_d, Y)$ 替换为优势函数 A self-critical baseline

$$A(a_c, \hat{a}_c, a_d, \hat{a}_d) = R_c(a_c, a_d, Y) - R_c(\hat{a}_c, \hat{a}_d, Y) > 0.$$

优势函数
但当动作向量 a_c baseline action vectors
采样自(得到正值)
$\pi_f(a_f | x_L^i, \theta_f)$ 基线动作向量.
有很高奖励，比最大似然 a_c (自临界基线) 更高

$\hat{a}_c$: 策略网络提出最可能的动作 : 接受/拒绝

$$\begin{cases} a_c^i = 1 & s_c^i > 0.5 \\ a_c^i = 0 & 其他 \end{cases}$$

coarse/fine 检测器 : 观察 x_L^i / x_H^i、输出 $\hat{a}_d$ (预测 bbox)
f_d^c f_d^f
detector

时间和图像采集成本:
$R_{cost} = (\sigma + \lambda)(1 - |a_c|_1)$
$R_c \rightarrow P_c \downarrow$

$$R_c = \underline{R_{acc}(\hat{Y}^f, \hat{Y}^c, Y)} + R_{cost}(a_c)$$

$$R_{acc}(\hat{Y}^f, \hat{Y}^c, Y) = \sum_{i=1}^{P_c} \left(\boxed{Recall(\hat{Y}_i^f, Y_i) - \cdot (Recall(\hat{Y}_i^c, Y_i) + \beta)} \right) \cdot N_i$$

在有更多物体的区域加倍

>0 时
R_{acc} 鼓励 Zoom-in (放大)

$$Recall(\hat{Y}_i^f, Y_i) - (Recall(\hat{Y}_i^c, Y_i) + \beta)$$

LR img 优先 : 在两个 Recall 很相似时
更倾向 LR img

总结 : 写得一塌.很大坨.

Expr 实验细节.