

Abstract

A novel Parsing Induced Learner to exploit parsing information to assist pose estimation, The proposed Parsing Induced Learner is composed of a parsing encoder and a pose model parameter adapter

所以重点是 Parsing Induced Learner

The Proposed Approach

3.1 The Formulation

Existing works suggest applying the parsing map S in pose processing to refine the pose estimation P . Differently, we propose a generic parsing induced pose estimation model $f_{[\theta, \theta']}: I \rightarrow P$, where $\theta' = g(I, S)$ (1) to fully leverage parsing information learned from the pair (I, S) in a flexible and effective way:

$$f_{[\theta, \theta']}: I \rightarrow P, \text{ where } \theta' = g(I, S) \quad (1)$$

上式中 I 是原图像输入, S 是 parsing map, P 是人体关节, 所以上式意味着

由 I 和 S 决定的 g 函数决定参数 θ' , 然后基于参数 θ' (该参数是动态的, 所以能很好的调整模型从而捕捉到更好的信息) 到 P 的模型

此外, 上式中的 g 函数由提取解析特征的解析编码器 $E_{\theta^S}^S(\cdot)$ 和学习动态参数 θ' 的参数适配器 $K_\phi(\cdot)$ 组成:

$$\theta' = g(I, S) := K_\phi(E_{\theta^S}^S(I)) \quad (2)$$

on top of these two encoders are pose and parsing classifiers $C_{w^P}^P(\cdot)$ and $C_{w^S}^S(\cdot)$ (猜测, K_ϕ 可能就是 $C_{w^S}^S$), 此外 w^P 和 θ^P 共同实例化了 (1) 式中的参数 θ . 最终的损失函数为:

$$\mathcal{L} := \mathcal{L}^P(C_{w^P}^P(E_{[\theta^P, \theta']}^P(I)), \hat{P}) + \beta \mathcal{L}^S(C_{w^S}^S(E_{\theta^S}^S(I)), \hat{S}).$$

3.2 The Network Architecture

Pose Encoder

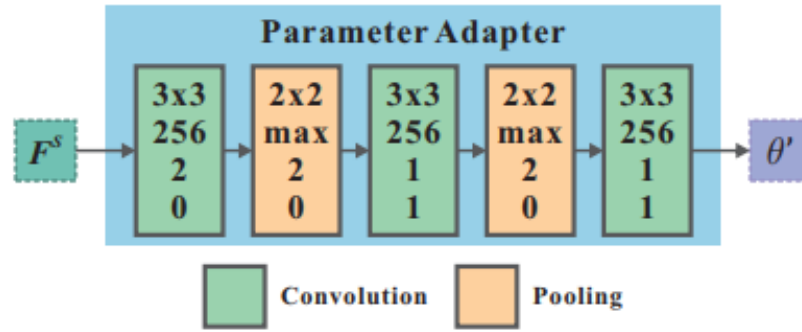
两种选择, 基于 VGG16 或者基于 Hourglass network 提取 F^P .

Parsing Encoder

提取特征 $F^S = E_{\theta^S}^S(\cdot)$, 是 PIL 的前半部分, 同样有两种实现方式, VGG16 和 Hourglass network.

Parameter Adapter

$K_\phi(\cdot)$ 是 PIL 的后半部分, 用来基于 F^S 预测 θ' , 其结构如下:



由于使用多层CNN预测 θ' 的参数空间过大，且由于参数较多，很容易造成过拟合以及浪费时间，所以，为了避免这个结果，我们使用一种类似于SVD的方式分解，减少自由参数：

$$\theta' = U * \tilde{\theta} * V$$

其中U和V是从adaptive convolution（下一节）中学习到的参数，而原先要预测的参数 θ' 变成了参数 $\tilde{\theta}$ 。

Adaptive Convolution

$$F^a = \theta' * F^P = U * \tilde{\theta} * V * F^P$$

Feature Fusion

$$F^{P*} = F^P + F^a$$

作为最后的pose estimation(此外，还有一个parsing estimation)

Classifiers

最后在 F^{P*} 上用一个1x1的convolution 实现一个线性分类器。同样，在 F^S 上也会实现一个线性分类器。

3.3 Training and Inference

针对3.1中的损失函数，我们使用均方误差函数训练pose部分的损失，使用交叉熵函数训练parsing部分的函数，此外，PIL也可以预训练，既是在不同的数据集上迁移信息。