# Pconv 网络

论文地址

https://arxiv.org/pdf/1804.07723.pdf

论文介绍

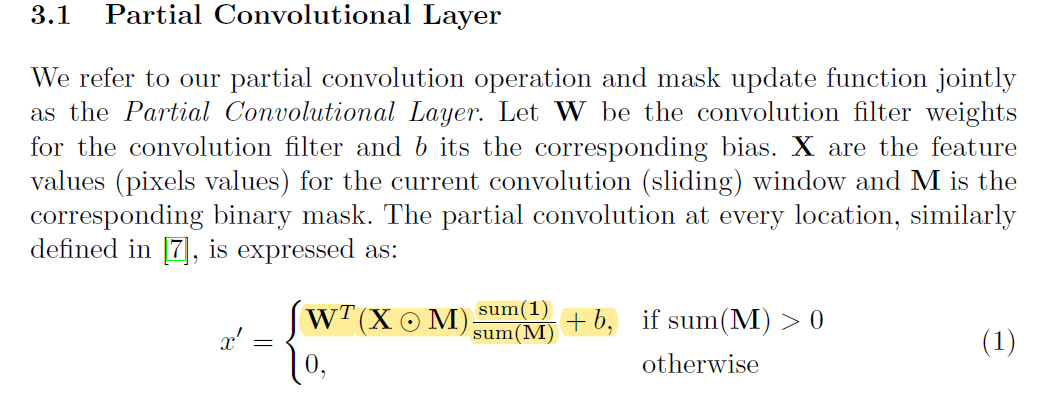
大意：就是解决普通卷积网络在训练时会利用掩膜中的不真实替代值的做法，容易出现伪影、颜色模糊、灰影等，后处理操作麻烦且代价昂贵。

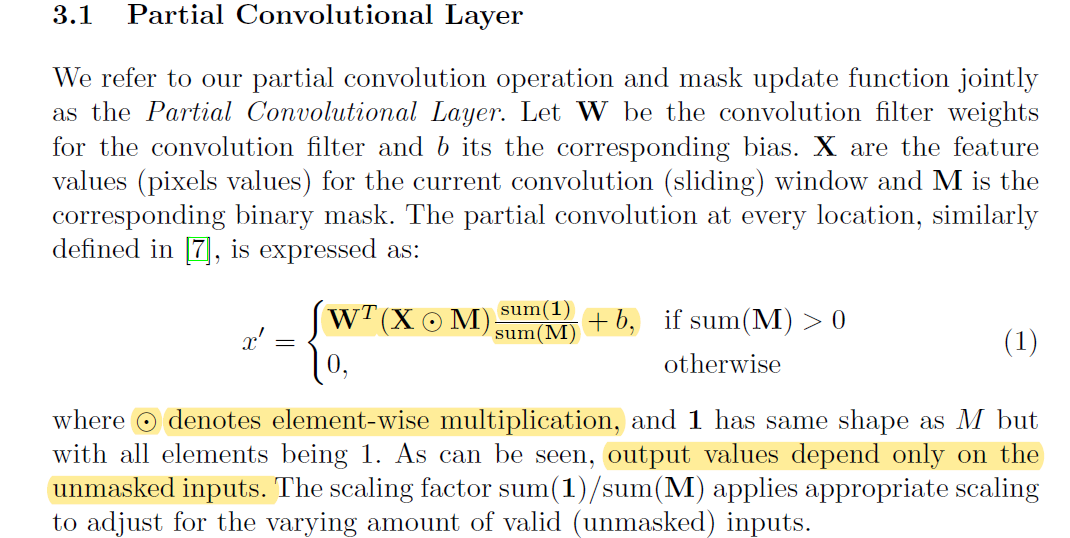
所以采取数据只在有效像素上进行卷积，自动化更新mask机制，下一层可以采取新的mask作为前向传播的一部分。方法在不规则mask上效果颇佳。

使用部分卷积的方法，其中卷积被掩蔽，并且仅基于有效像素进行重新归一化等处理。

在训练阶段，将空白或缺失的部分引入上述数据集的完整训练图像中，以使网络能够学习重建缺失的像素。

在测试阶段，另一批没有在训练期间使用的空白或缺失部分被引入数据集里的测试图像，以对重建的图像的精度进行无偏验证。

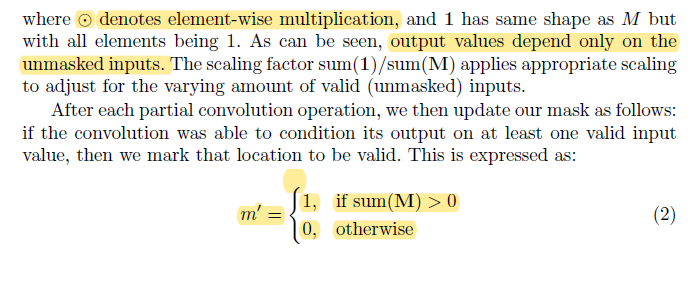




在当前滑动窗口上，选择有效的滑动值参与计算（选择方式是mask为判断标准），并加一个sum除法来规范化。

如果没有有效值，就直接置为0。

下一步更新mask



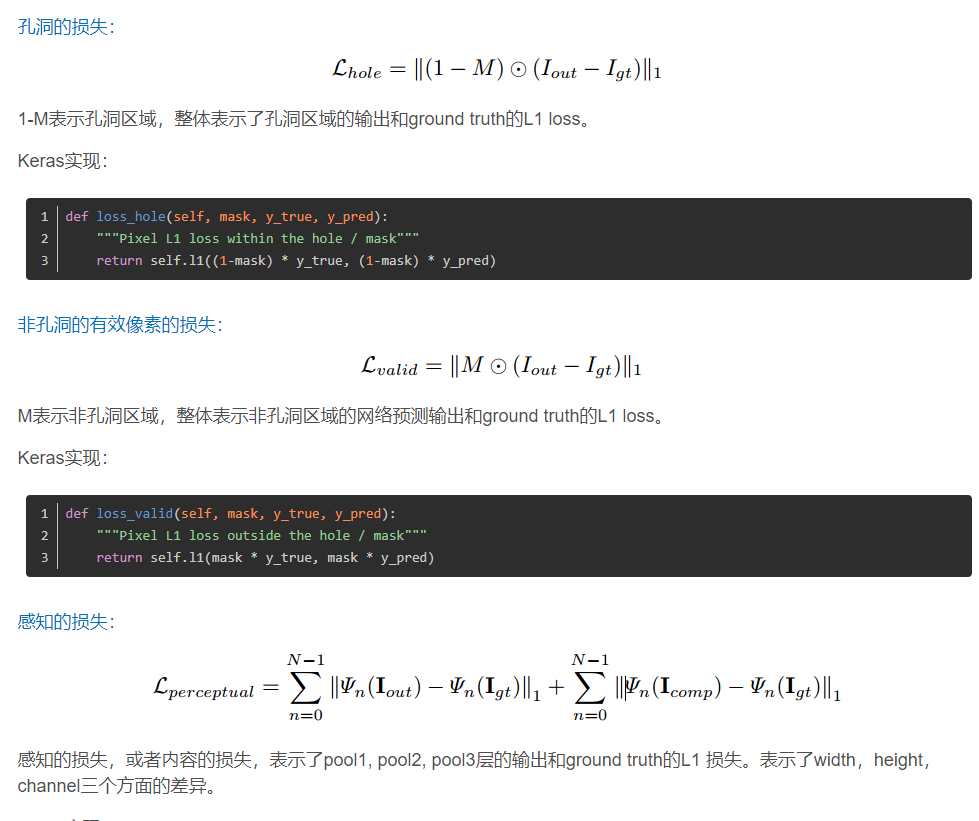
看上去很简单。。

Mask就是C\_H\_W的二进制mask

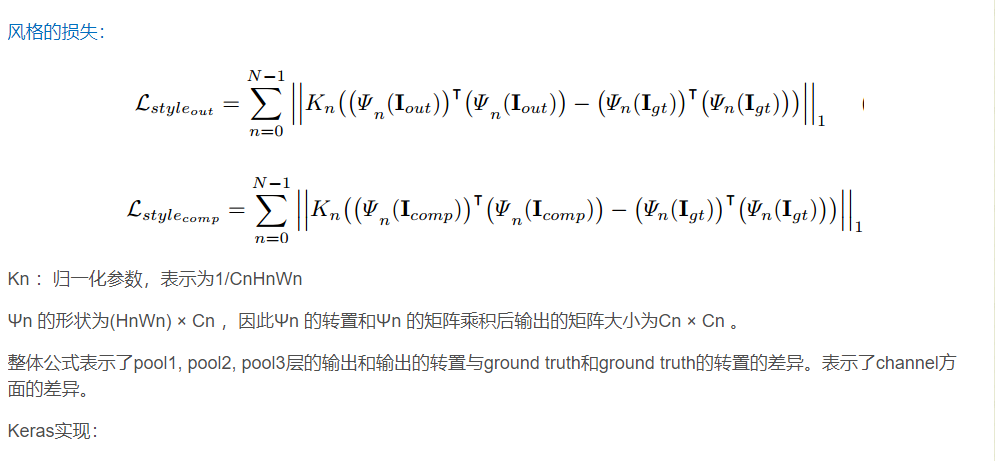
网络结构：

类似Unet-结构，pconv卷积层替换传统卷积层。

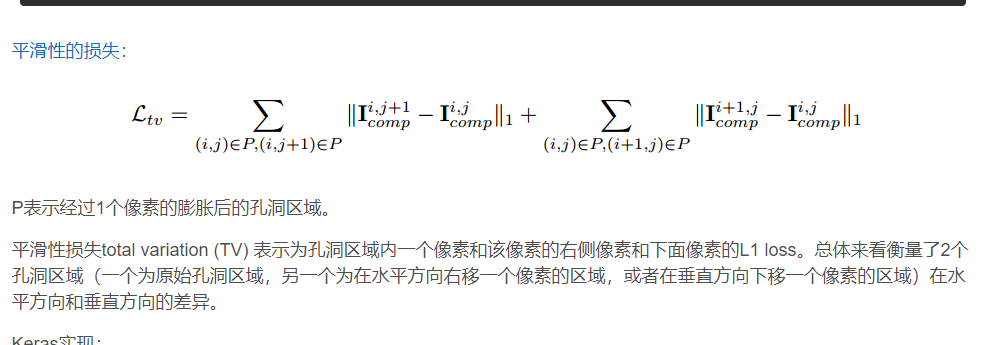
Loss函数：l1损失



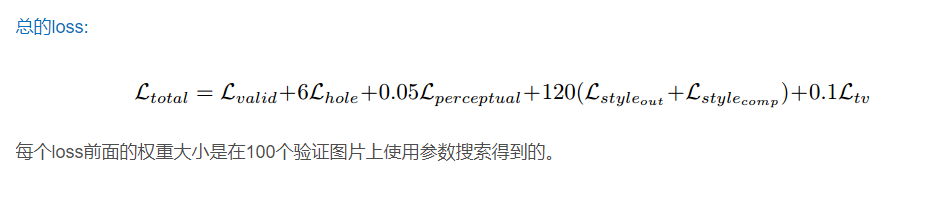
+风格损失（计算风格损失之前加了个矫正矩阵 Gram matrix）



平滑性损失



总损失



权重训练的？？

我们的损失函数的目标是每像素的重建精度和复合位置，即预测的孔值如何平滑地过渡到他们的左右舍入上下文。

数据文件自己新建

安装好环境

Python 3

PyTorch 1.0.0用的是

NVIDIA GPU + CUDA cuDNN

等一些包

pip install -r requirements.txt

训练

python train.py \

--image\_root [path to input image directory] \

--mask\_root [path to masks directory]

例如 python train.py --image\_root datasets\ --mask\_root masks\ --batch\_size 12 --pre\_trained ./checkpoint/ckpt/model\_100.pth

python train.py --image\_root datasets\ --mask\_root masks\ --batch\_size 16 --pre\_trained snapshots\PairsStreetView\ckpt\

其他的参数在train.py里面看

测试

python eval.py \

--pre\_trained [path to checkpoints] \

--image\_root [path to input image directory] \

--mask\_root [path to masks directory]

例如

python eval.py --pre\_trained snapshots\PairsStreetView\ckpt\model\_70.pth --image\_root datasets\ --mask\_root masks\ --result\_root result/time2020-10-13

python eval.py --pre\_trained checkpoint\ckpt\model\_70.pth --image\_root datasets\ --mask\_root masks\ --result\_root result/time2020-10-13

适用于

保存的测试图片依次为

masks[i]

damaged\_images[i]

outputs[i]

outputs\_comps[i]

ground\_truths[i]