# Shift-net网络尝试

全名为Shift-Net: Image Inpainting via Deep Feature Rearrangement（特征重排）

## 网络背景介绍

论文地址

<https://openaccess.thecvf.com/content_ECCV_2018/papers/Zhaoyi_Yan_Shift-Net_Image_Inpainting_ECCV_2018_paper.pdf>

介绍

https://blog.csdn.net/qq\_36406693/article/details/100176958?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522160307065419195246608059%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request\_id=160307065419195246608059&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~first\_rank\_v2~rank\_v28-1-100176958.pc\_first\_rank\_v2\_rank\_v28&utm\_term=Shift-Net%3A+Image+Inpainting+vi&spm=1018.2118.3001.4187

### 大体介绍：

结合了patchbatch的传统方法提取细节信息与cnn的提取全局信息的优点。

在unet基础模型上引入一个shift-connection连接层，以填充任何形状的缺失，以及细节。在解码器功能上引入了gudience loss，以最大程度地减小完全连接的层后的解码器功能与缺失部分的真实编码特征之间的距离。With such constraint, the decoder feature in missing region can be used to guide the shift of encoder feature in known region

#### 创新点3个：

1. 增加了一个shift-connection layer to the U-Net architecture。

2. 引入了一个guidance loss 为了监督 损失图的缺失部分经过编码，解码后的特征图，

更加接近真实原始图的该缺失位置所编码的特征图。

3. 端到端训练（就是搬砖工越来越懒了，都喜欢 raw data 丢进去，期望的输出吐出来，拿去训练就能得到好看的结果。

好处是方便。

坏处是越来越像一个黑箱。）

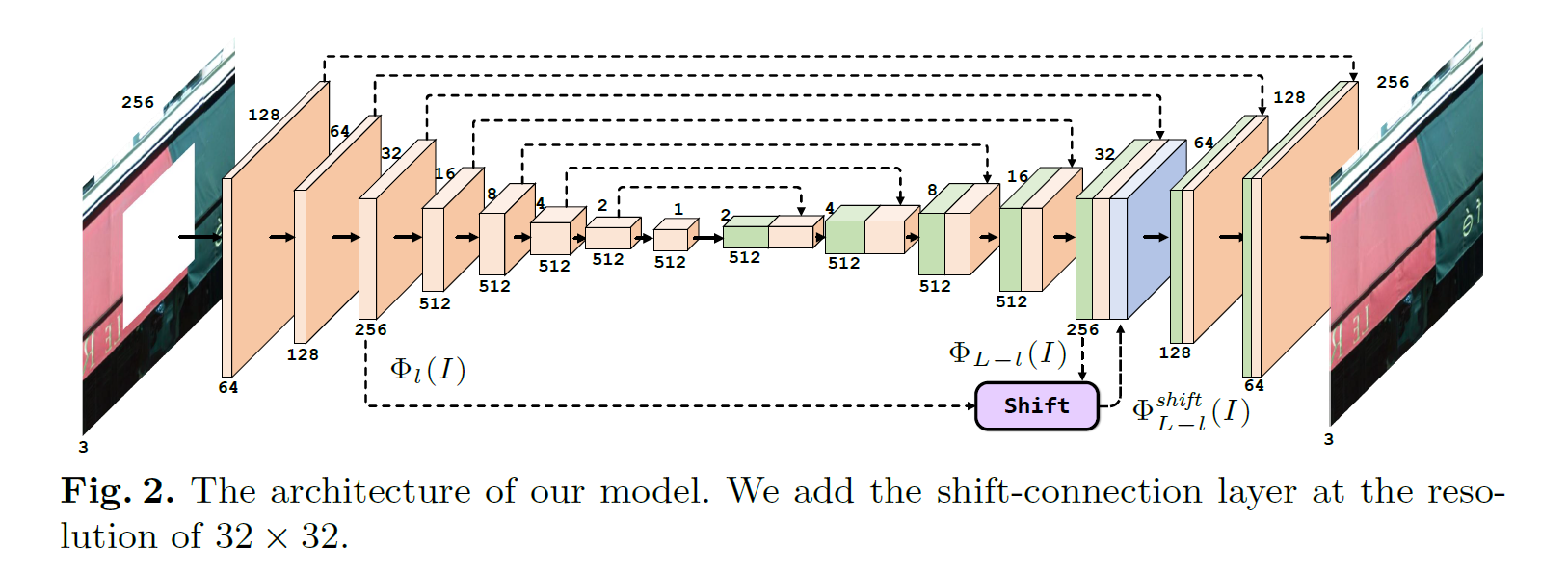
用已知区域编码得到的编码方式服务于未知区域的编码估计，

链接已知区域的编码特征

和缺失部分的解码特征

##### 详细介绍：

###### 新的shift-connect层：



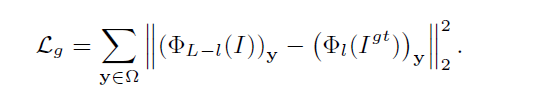
###### Loss函数：

Guidance loss 和 reconstruction loss 和对抗损失结合，指导shift层的参数和全局的model得到参数。

Guidance loss：

为了确保解码器功能可以用作良好的引导，引入了引导损失以将解码的特征强化为接近真实图片的编码特征。

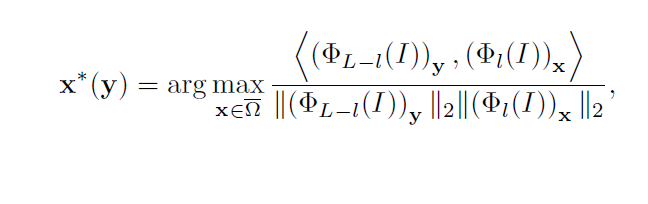
Guidance loss只是定义在缺失区域。



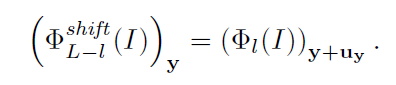
以此获得缺失区域的信息。

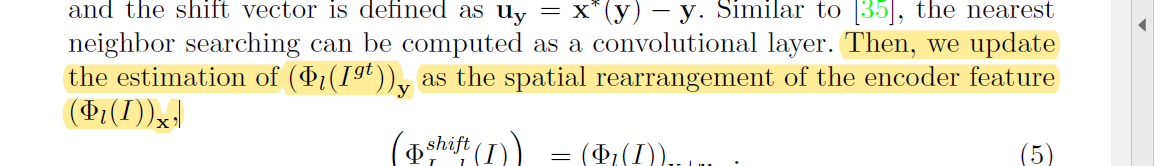
###### Shift-connection的方法：

对于L-l解码层上的未知区域上的点y，其在l编码层上的最相似搜索的某点x（x处于已知区域），采取min余弦距离，得到x



得到了转换向量，u=x\*y – y



最相似搜索可以用卷积层替代，其对未知区域的编码可看为损失区域的某个部位的空间特征重排。

以前的方法都是在同一层的已知区域中寻找点填充未知区域，而作者是从l层中的已知区域提取点来填充L-l层的未知区域。

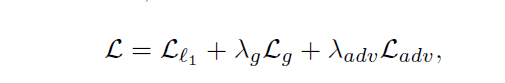
最后将 \Phi _{L-l}(I)，\Phi _{l}(I) 和\Phi _{shift}^{L-l}(I)三个concatenated(通道拼接)并且作为第（L-l+1）层的输入

L1损失和对抗损失为了获得更好的缺失部分更接近的纹理细节，在结果上使用。在图像修复中很常见。

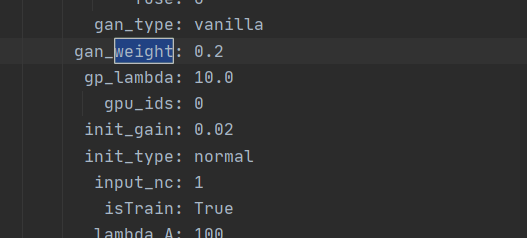
Shift 方法与传统的方法的区别：

1. shift方法基于深度编码的，端到端学习。
2. 传统的方法是解一个最优化问题，而shift connection是利用guidence loss平行计算多个loss
3. 传统方法不保留全局信息连贯性，只有局部连贯性

#### loss函数最终和：

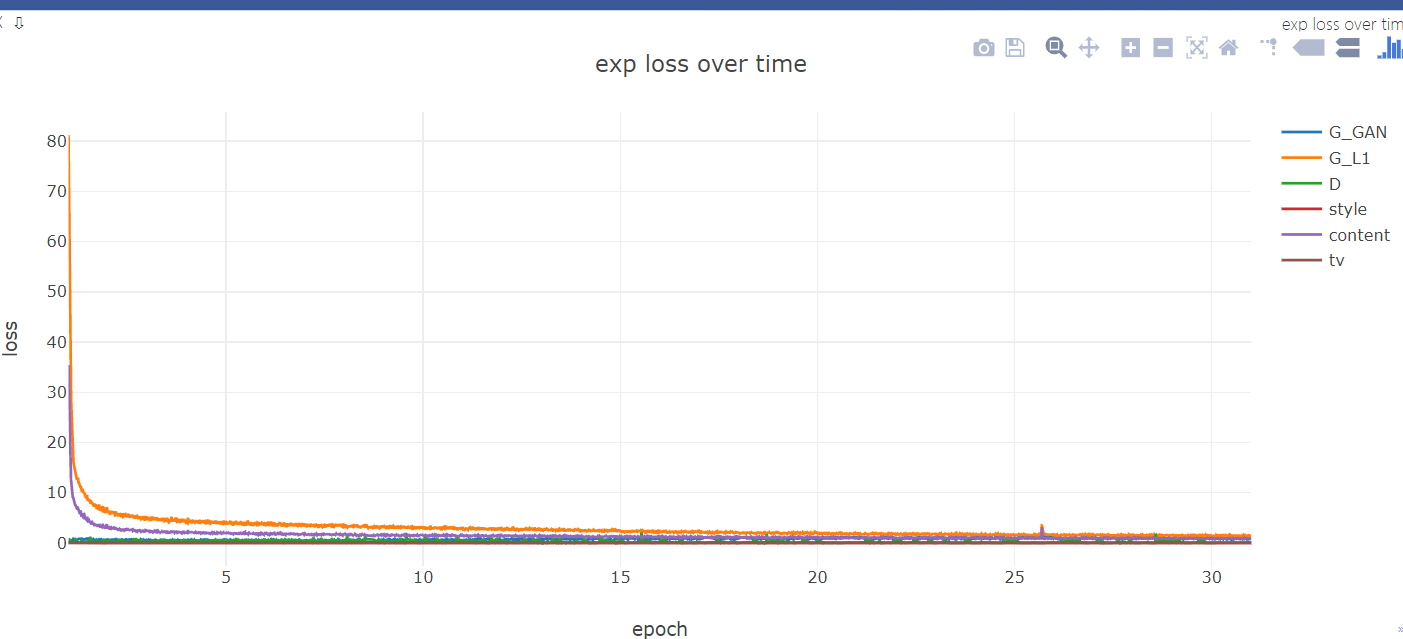


系数预先学习得到。

看到原始代码上给定的参数是固定值。

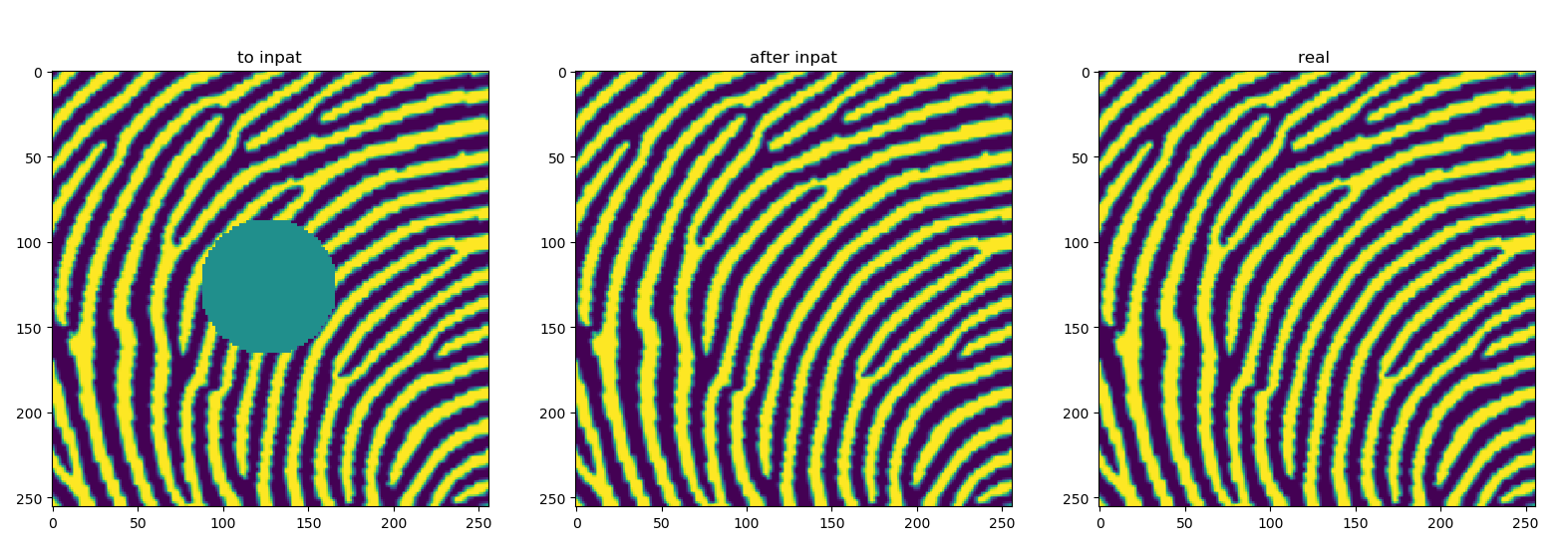
## 结果几个示例:

Loss函数下降趋势



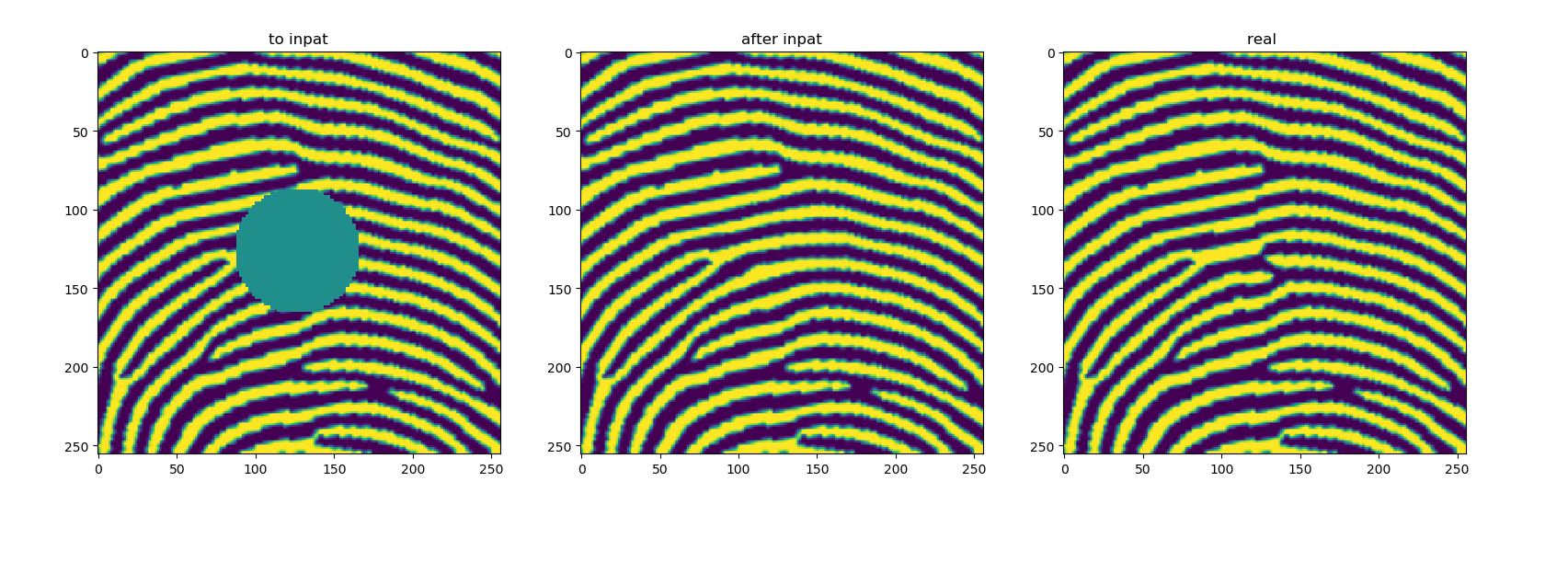
将1w张照片训练了20次的模型。测试。

缺失部分没有特征点的图像修复效果已经足够好了。（因为本来就等价于直线连起来。）



### 问题1：

右1是真实照片，右2是补全的照片。

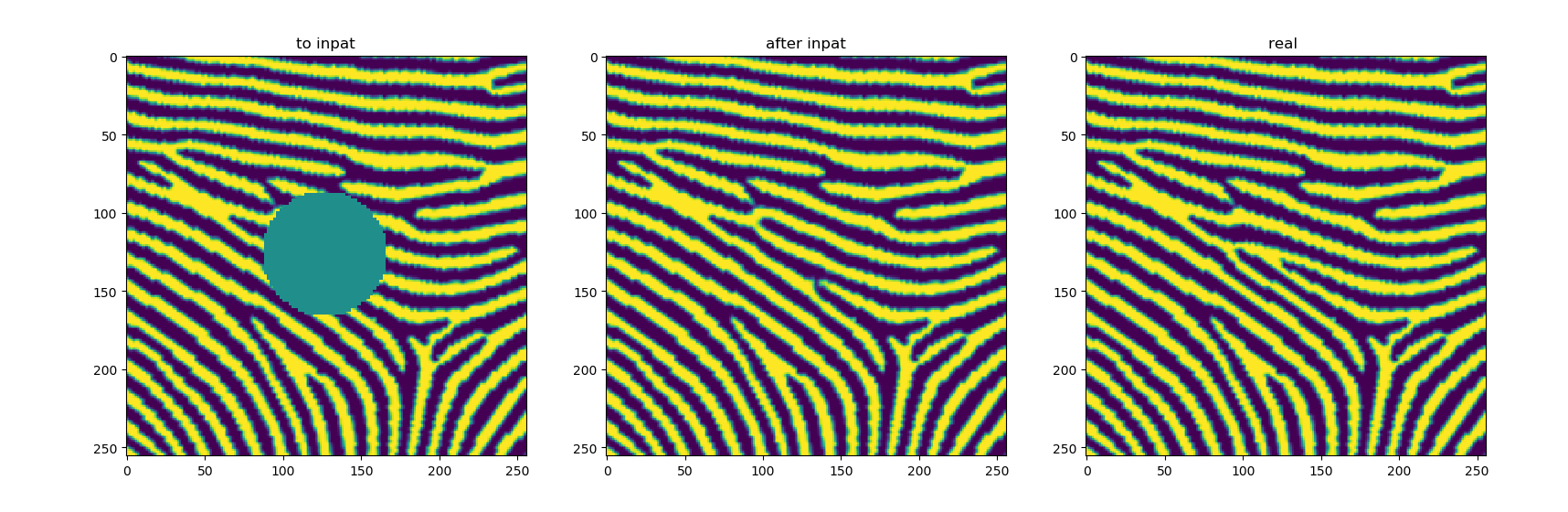


遗失了信息，是个特别的纹路？

拿来进行特征点预测也是可行的。

同理类似

### 例子2：

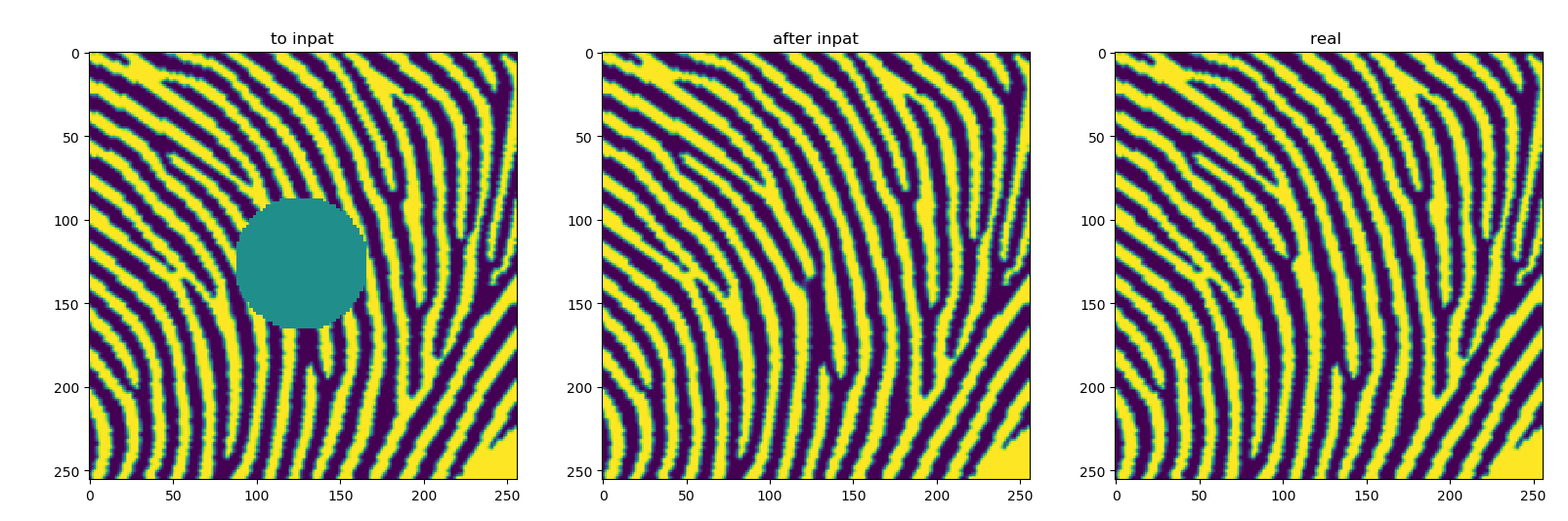


右1图有2个特征点，2个端点，

右2图有2个特征点，1个分叉点，+1个端点

特征点的数目上还是符合的。

### 例子3：



例3类似于例1，在靠近的地方正确地预测出了特征点是存在的，虽然位置上有一点偏差。

其他例子效果类似

## 总结

可以大概预测正确特征点的数目，只是有时会有一点位置上的偏差。

训练次数几十次就可以得到较好的结果。

论文里写的未来的研究方向：将进行更多的研究来提高shift操作中最近邻搜索的速度，可以引入多个shift连接层，并将移位连接扩展到其他低级视觉任务。

## 细化测试

同理，细化后的图进行修补测试