# 1.Shift-net 网络

论文地址

<https://arxiv.org/abs/1801.09392>

github地址

<https://github.com/Zhaoyi-Yan/Shift-Net_pytorch>

## 论文介绍

https://blog.csdn.net/qq\_36406693/article/details/100176958?utm\_medium=distribute.pc\_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-5.nonecase&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-5.nonecase

### 创新点

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

总的结构还是unet网络。

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。

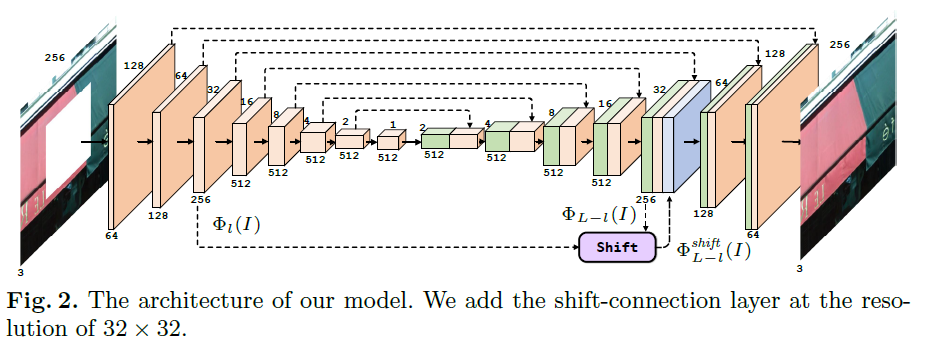
好出是方便。

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



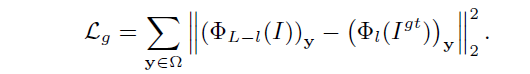
总而言之，三种贡献：

1. 引入了shift连接，能结合cnn方法和基于范例的优点（基于范例的方法是类似patch修复）。
2. The guidance, reconstruction, and adversarial losses are introduced to train our Shift-Net.
3. 细节纹路更好的修复结果。

详细介绍：

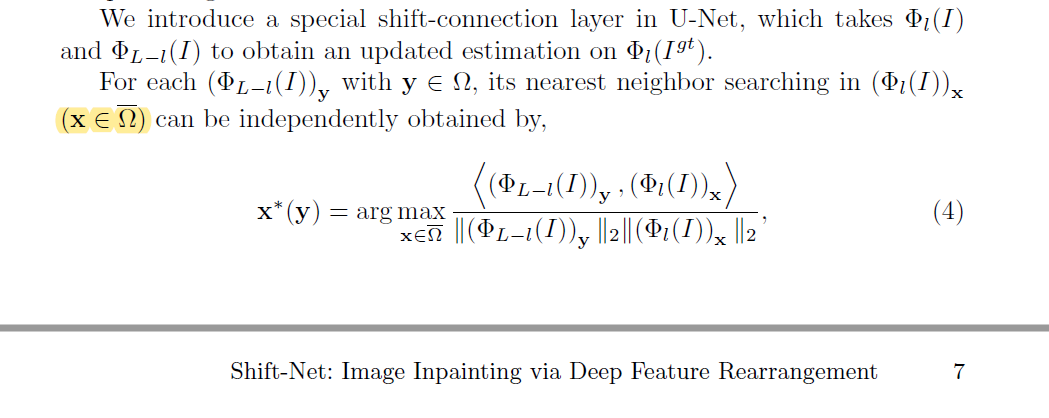
重点guide loss 和shift operation

Guidance loss采取的是l2损失



采取concatenating l层和L-l层，可以容易实现该目的。

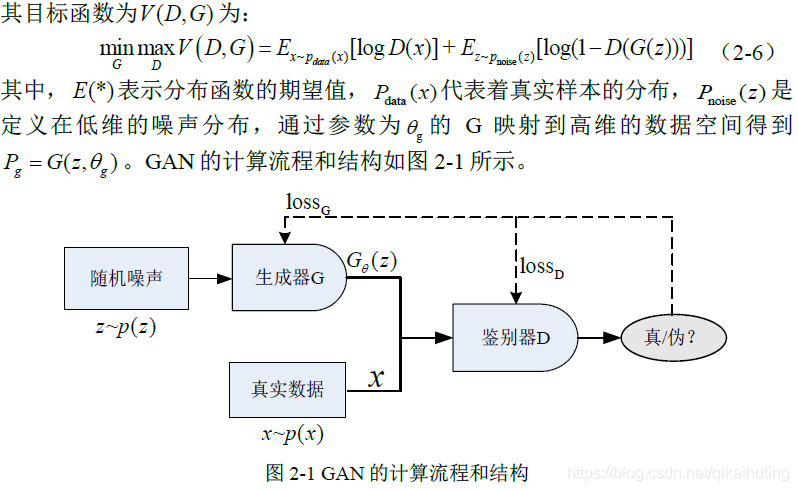
Shift connection



再利用l层和L-l层获得真实图片的l层估计，找到了转换关系（用卷积操作）

然后，我们将（l（Igt））y的估计值更新为编码器特征（l（I））x的空间重排，

总的loss是l1loss（最终的结果）+对抗损失+ gudience loss



Taking guidance, `1, and adversarial losses into account, the overall objective

of our Shift-Net is de\_ned as,

L = L`1 + \_gLg + \_advLadv;

# 使用说明

## 训练过程

安装pytorch1.3

Conda 进入新环境

安装必要的包

pip install visdom

pip install dominate

requirement.txt安装一下就好

几个包有坑，scipy安装1.1.0 才有imresize函数

训练的基础模型

参数详见base\_option.py

Input\_nc灰度图选1，--output\_nc灰度图选1

Loadsize

python train.py --batchsize=1 --use\_spectral\_norm\_D=1 --which\_model\_netD='basic' --mask\_type='center' --which\_model\_netG='unet\_shift\_triple' --model='shiftnet' --shift\_sz=1 --mask\_thred=1 --loadSize=256

python train.py --dataroot=’./datasets/gray\_origin’ --batchSize=1 --loadSize=256 --input\_nc=1 --output\_nc=1 --training\_mask\_folder='./datasets/masks' --offline\_loading\_mask=1

报错：RuntimeError: Given groups=1, weight of size [64, 2, 4, 4], expected input[1, 4, 256, 256] to have 2 channels, but got 4 channels instead

## 自己的mask训练

要设置

set --offline\_loading\_mask=1 when you want to train/test with your own prepared masks.

放在

The prepared masks should be put in the folder --training\_mask\_folder and --testing\_mask\_folder.

当采取固定的mask时候，推荐使用 --norm switchable选项

训练选项(loadsize要至少256…,网络里简单地resize一下了，采取bicubic方法)

python train.py --dataroot ./datasets/gray\_origin --batchSize 1 --loadSize 256 --input\_nc 1 --output\_nc 1 --training\_mask\_folder ./masks/training\_mask\_folder --offline\_loading\_mask 1 --nThreads 0

通道问题

# 通道问题啊啊啊啊 3 通道和单通道问题

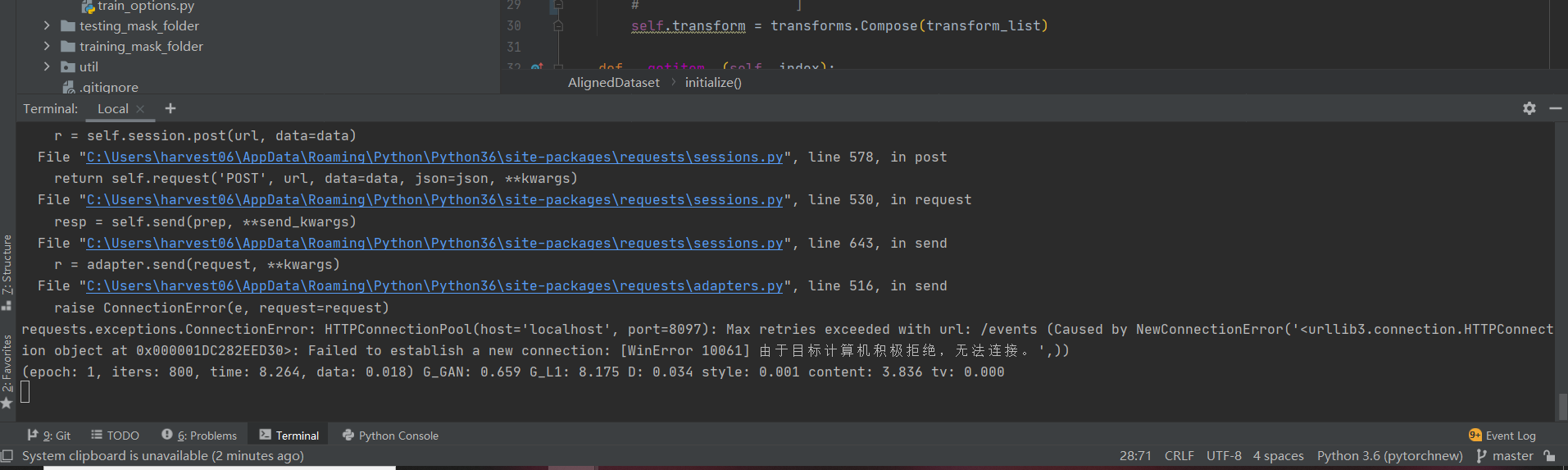
transform\_list = [transforms.ToTensor(),

transforms.Lambda(lambda x: x.repeat(3,1,1)),

transforms.Normalize((0.5, 0.5, 0.5),

(0.5, 0.5, 0.5))]

数据老是报错，最后修改



报错？？？（这个要先提前运行python -m visdom.server命令，然后去<http://localhost:8097>查看结果

Visdom不能正常启动问题python -m visdom.server提示Downloading

<https://zhuanlan.zhihu.com/p/105808225>

https://github.com/casuallyName/document-sharing

）

python train.py --dataroot ./datasets/gray\_origin --batchSize 1 --loadSize 256 --input\_nc 3 --output\_nc 1 --training\_mask\_folder ./masks/training\_mask\_folder --offline\_loading\_mask 1 --nThreads 0

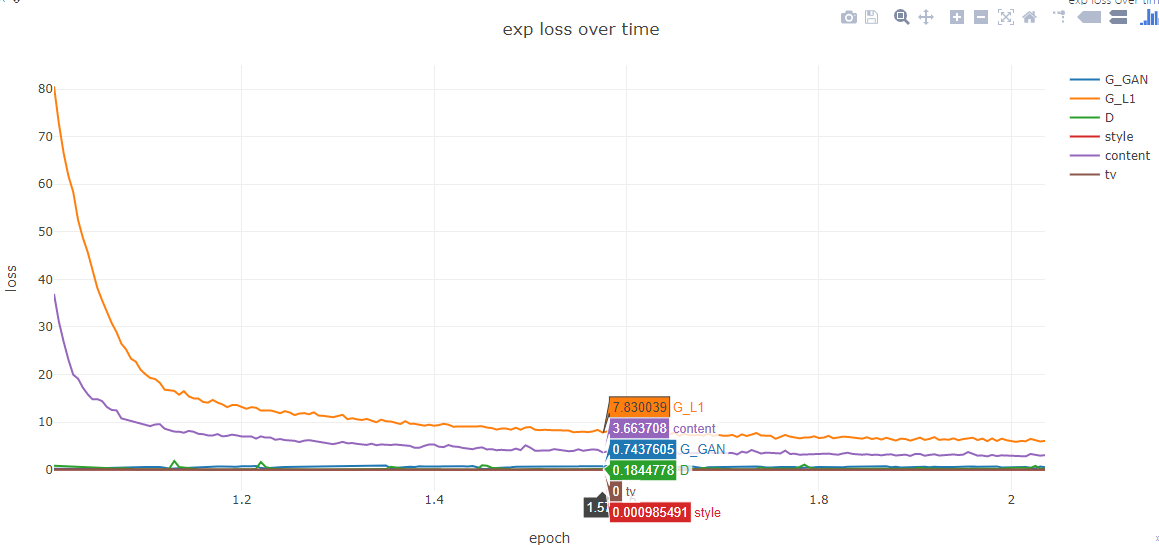
加大batchsize

python -m visdom.server

python train.py --dataroot ./datasets/gray\_origin --batchSize 25 --loadSize 256 --input\_nc 3 --output\_nc 3 --training\_mask\_folder ./masks/training\_mask\_folder --offline\_loading\_mask 1 --nThreads 0

python train.py --dataroot ./datasets/gray\_origin --batchSize 25 --loadSize 256 --input\_nc 1 --output\_nc 1 --training\_mask\_folder ./masks/training\_mask\_folder --offline\_loading\_mask 1 --nThreads 0

终于调通了。。（三通道改为单通道怎么改？？）



## 测试：

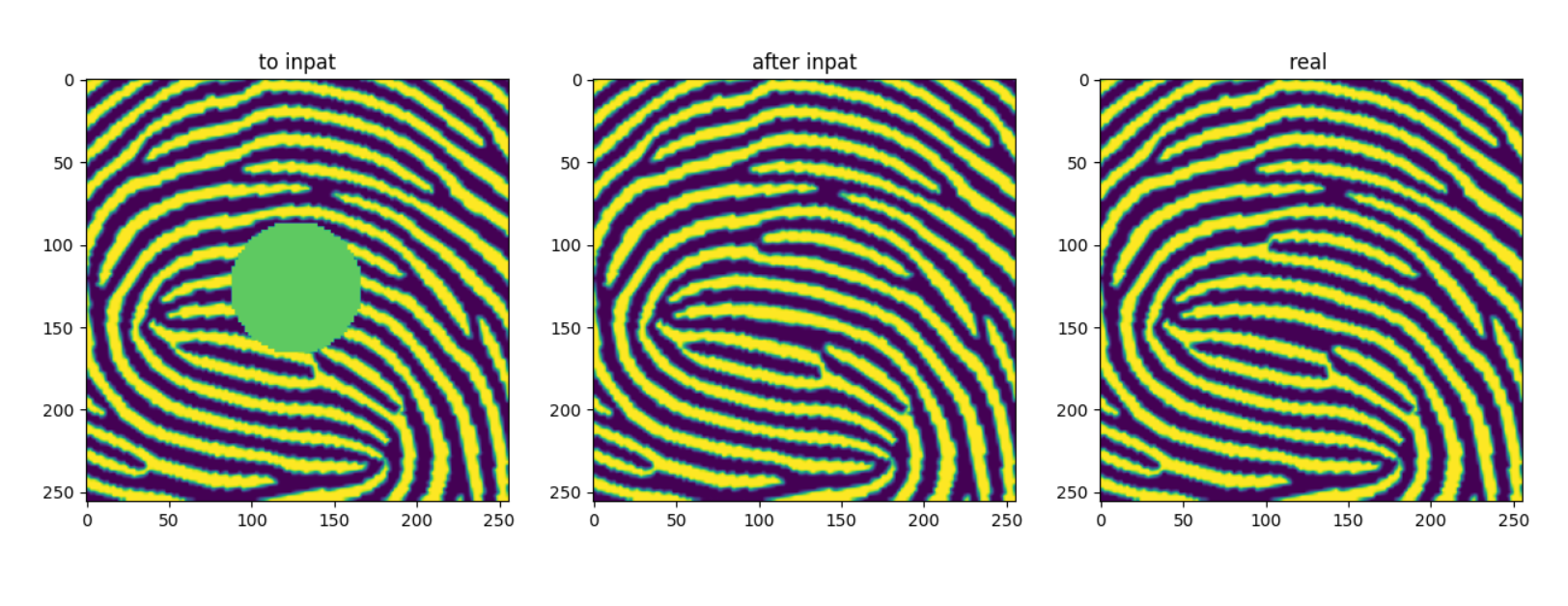
代码：

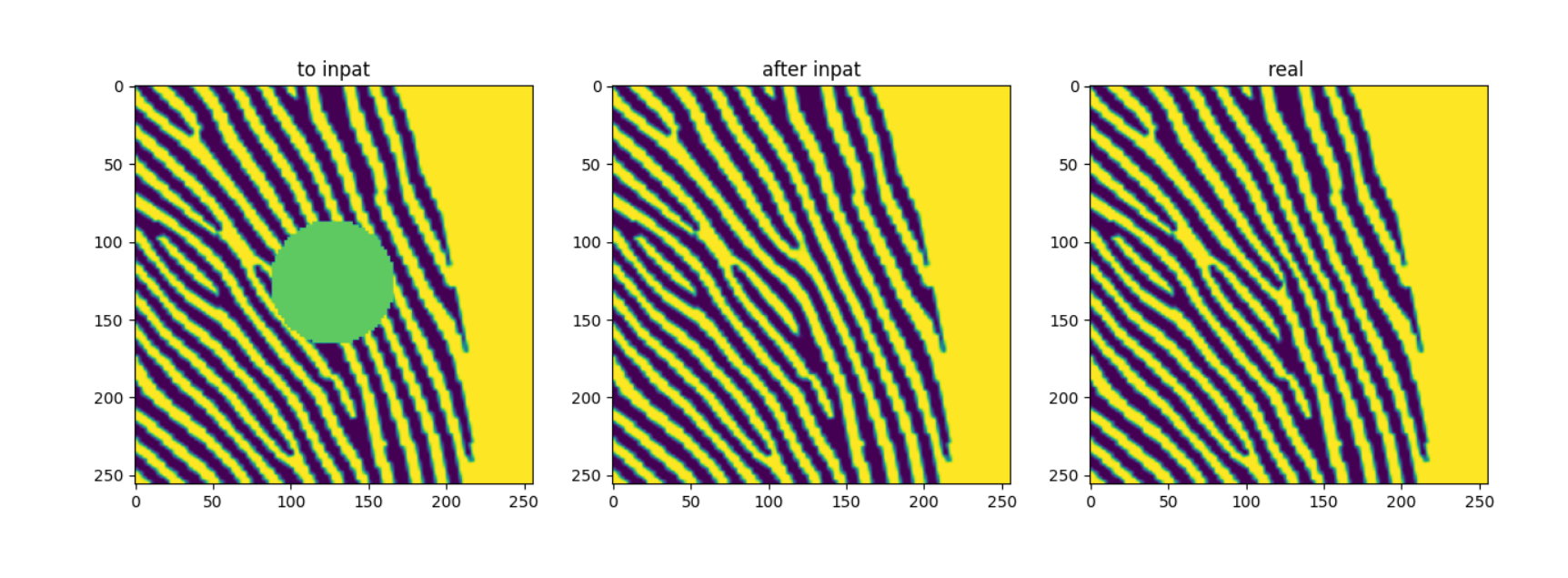
python test.py --dataroot ./datasets/200test --batchSize 25 --loadSize 256 --input\_nc 3 --output\_nc 3 --offline\_loading\_mask 1 --nThreads 0 --results\_dir ./results/ --which\_epoch 20 --testing\_mask\_folder ./masks/testing\_mask\_folder

测试要修改test.py的opt.nThreads = 0避免出现，测试还是

EOFError: Ran out of input

效果展示（也挺好了）只循环了20多epoch次数，效果和pconv差不多好了





修改后的代码（）

训练代码：

选择文件夹（shift-net\_rgb）

彩色图训练：（训练数据放在./datasets/rgb\_origin）

python train.py --dataroot ./datasets/rgb\_origin --batchSize 25 --loadSize 256 --input\_nc 3 --output\_nc 3 --training\_mask\_folder ./masks/training

\_mask\_folder --offline\_loading\_mask 1 --nThreads 0

选择文件夹（shift-net\_gray）

黑白图训练：（训练数据放在./datasets/gray\_origin）

python train.py --dataroot ./datasets/gray\_origin --batchSize 25 --loadSize 256 --input\_nc 3 --output\_nc 3 --training\_mask\_folder ./masks/training\_mask\_folder --offline\_loading\_mask 1 --nThreads 0

--whichepoch可选项，--continue-train可选