

图像补全实验报告

2018011422 李响

运行环境

python 3.8.9

numpy 1.21.5

opencv-python 4.5.2.54

代码分析

读图并初始化

结构补全

在曲线与未知区域相交的部分确定锚点，确定其对应需要补全的范围，在曲线和一直区域的相交部分确定样本集，同样确认其所对应的范围。

之后计算最小能量差并补全， $energy[i][k]$ 表示锚点 i 与样本点 k 之间的能量差，用动态规划确定最小能量差，其中 E_1, E_2, E_S, E_I 与论文中公式一致。在部分计算上参考了https://github.com/grahambryan/img_completion。

结构补全的中间结果输出为，xx_stru.png。

纹理补全

为对比效果，直接使用cv2.inpaint输出为xx_text1.png，实现效果输出为xx_text2.png。

这里参考的方法是课堂上“基于像素的补全”中《Region Filling and Object Removal by Exemplar-Based Image Inpainting》。

其中主要流程是，确定mask边缘部分，计算待填充部分的优先级，由优先级最高的点确定所需要填充的块，寻找最相似的块，填补图片。

其中较为复杂的是计算待填充部分优先级，代码中confidence、data、priority分别对应论文中以下三个公式。

$$C(p) = \frac{\sum_{q \in \Psi_p \cap (I - \Phi)} C(q)}{|\Psi_p|}$$

$$D(p) = \frac{|\nabla I_{\frac{1}{p}} \bullet n_p|}{\alpha}$$

$$P(p) = C(p)D(p)$$

运行方式

注意路径不能有中文

```
PS E:\hw\img_completion> python main.py --help
usage: main.py [-h] [-i IMAGE_NAME] [-s SAMPLING_INTERVAL] [-p PATCH_SIZE] [-d DOWN_SAMPLE] [-t]

set option

optional arguments:
  -h, --help            show this help message and exit
  -i IMAGE_NAME, --image_name IMAGE_NAME
                        Image Name
  -s SAMPLING_INTERVAL, --sampling_interval SAMPLING_INTERVAL
                        Sampling Interval
  -p PATCH_SIZE, --patch_size PATCH_SIZE
                        Patch Size
  -d DOWN_SAMPLE, --down_sample DOWN_SAMPLE
                        Down Sample
  -t, --to_show          show detail by plt
```

```
# 默认运行样例南瓜图
python main.py
# -t可以在代码运行过程中逐步展示
python main.py -t
# 对应结果展示效果具体运行方式(纹理补全用时较长)
python main.py -i pumpkin -s 15 -p 15 -d 2
python main.py -i bed -s 20 -p 20 -d 2
python main.py -i tree -s 5 -p 5 -d 2
```

结果展示与分析

样例中的南瓜图



上一为原图，上二为补全区域，上三为补全结构

下一为仅结构补全的结果，下二为opencv自带补全结果，下三为实现的结果

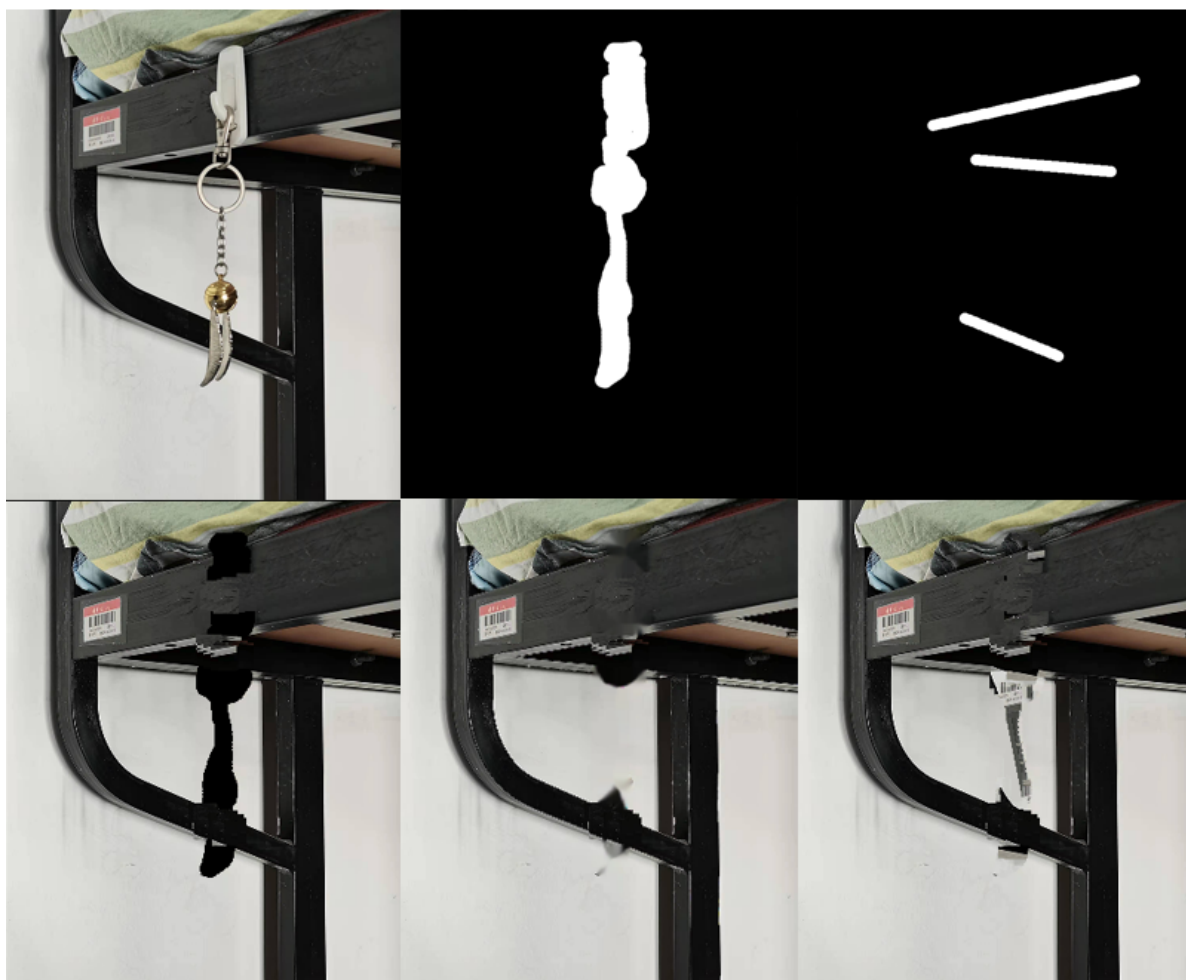
可见，除了窗框下有一些瑕疵歪，基本可以接受

无结构信息的纹理补全结果如下：



意外的是，居然补全了竖向的窗框，但南瓜部分结果不佳

随手拍的床部分框架（效果比较有特殊性，就选出来了）



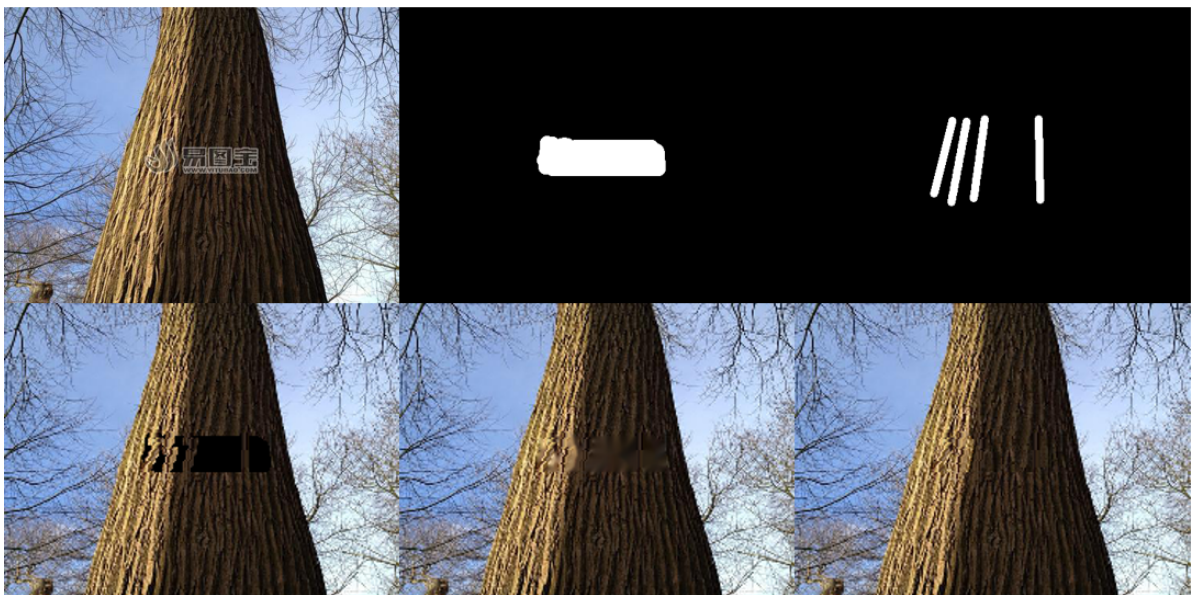
可以看到，这个补全结构的效果有部分效果较差，主要在于亮面和暗面差别较大，无法很好的区分边界，导致框架不能很好地补全，在这里opencv的效果就比实现效果要好，我认为是填补顺序的关系。

无结构信息的纹理补全结果如下：



这里直接用纹理补全甚至比结构补全还好，我认为是在结构的标注上，阴暗面不能在同一个标注线上，导致阴暗面分界线不明显，影响后续纹理补全，直接用纹理补全就好一些。于是在下面的水印图中，我在阴暗边界都标注了结构。

网上找到的带水印树干图



可以看到，在亮部和暗部的边界各标注一个结构，可以更好地区别边界，纹理也较为清晰，实现的效果比opencv填充效果更好。

无结构信息的纹理补全结果如下：



这里直接纹理补全效果不佳，我认为主要原因是填补顺序和图片大小的关系，这张图像素较小，但在纹理补全时patch_size固定，结果就可能会差一些。

小结

在这次作业中，实现了带结构标注的纹理补全办法，结构补全可以尽可能地保留原有结构，为纹理补全时提供更可能相似的部分，但在亮暗交界处表现不太好，在背景较为单一（如玻璃）或结构较为重复（如树干）上表现较好。