

Pytorch_demo

目的：

1. 使用pytorch尝试构筑 简单的图像融合网络
2. 尝试跑通简单的对比方法

网络要求：输入红外和可见光(RGB)图像对，输出融合图像(RGB)

网络结构：

Denseblock或Resblock为基本单元构筑

网络形式不要求 直链式/U-Net形式带上下采样

实际实现（三选一）：

- 红外图像和可见光图像 分开处理 不共享参数 提取特征后concat
- 红外图像和可见光图像 分开处理 共享参数 提取特征后concat
- 红外图像和可见光图像 一开始就concat合并处理

任务要求：

分别实现红外可见光的有&/无监督学习、

无监督学习：

损失函数可以使用l1、l2以及SSIM

SSIM可由python包kornia得到 其他为已封装基本函数

融合图像分别和红外、可见光图像求loss

有监督学习：

可用损失函数同上

融合图像和真值求loss （也可以加点和源图像对的约束 都可以）

该处所用真值由<https://github.com/JinyuanLiu-CV/TarDAL>代码跑通得到

其他：

- 实现训练过程的可视化 能够输出loss曲线
- 实际训练时需要用到成对图像块进行训练 将所给数据集37对进行图像裁剪扩充以增广训练数据 （参考：扩充到1k个64×64的patch）
- 测试时输入数据需预处理 RGB的可见光图像需转换为YCbCr通道提取Y通道和红外图像进行融合 最后融合得到的实际上是融合图像的Y通道 再将此Y通道配合可见光的CbCr通道还原色彩

训练用37对灰度图片数据集百度云链接

<https://pan.baidu.com/s/1gMiDAMbOblvCIDYQxfo7XQ?pwd=1234>

测试用221对RGB图片github链接 <https://github.com/hanna-xu/RoadScene>

主要代码分布参考：

dataset.py 写dataloader

model.py 搭建网络

loss.py 损失函数（该任务可以没有 三个基础loss均已被封装好）

train.py 训练

test.py 测试