

# Part1

## 任务一

### 基础问题

我认为是残差学习比较好，整个图形作为标准考虑，像素太多，没必要考虑那么多，专注雨层能减小计算量。

### 附加问题

- 1.雨丝像素的周围是曲线，且雨丝形状相差不大，或许可以获取曲线型的像素
- 2.有可能吧，模型依据其他地方的细节，模拟修复被遮挡的地方。但是如果遮挡面积过大，那么修复出来的和原图像则可能会出现大规模的错误

## 任务二

### 目标函数

1. L1范数定义：向量元素的绝对值之和 几何意义：原点到某点横纵移动的总长度  
比如 (0, 0) 到 (3, 4) 要走3横4纵，就为7  
L2范数定义：向量元素的平方和的平方根 几何意义：原点到某点的直线距离  
例子如上，这个直线距离求得为5
2. L1损失是计算两张图片每个像素差的绝对值然后求平均，L2损失是计算平方差然后求平均。  
把模型输出的去雨图和真实无雨图像按照素位置对齐。然后，对每个相同位置的像素，用它的像素值相减，对差值取绝对值（L1）或平方（L2）。接着，把图片中所有像素的这种惩罚值加起来。最后，除以总像素数，就得到了一个代表两图差异程度的损失值

评估函数

1.PSNR

2.SSIM

- 1.从数学公式上来看是 $[-1, +1]$ ,但一般会调整为 $[0, +1]$
- 2.从亮度 (Luminance) , 对比度 (Contrast) , 结构 (Structure)三个角度
- 3.SSIM每一次的评估对象是一个个小窗口，一个局部特征，然后对其求平均

## 任务三

# Name	Version	Build	Channel
bzip2	1.0.8	h2bbff1b_6	
ca-certificates	2025.9.9	haa95532_0	
colorama	0.4.6	pypi_0	pypi
expat	2.7.1	h8ddb27b_0	
filelock	3.20.0	pypi_0	pypi
fsspec	2025.9.0	pypi_0	pypi
imageio	2.37.0	pypi_0	pypi
jinja2	3.1.6	pypi_0	pypi
lazy-loader	0.4	pypi_0	pypi
libffi	3.4.4	hd77b12b_1	
libzlib	1.3.1	h02ab6af_0	
lpips	0.1.4	pypi_0	pypi
markupsafe	3.0.3	pypi_0	pypi
mpmath	1.3.0	pypi_0	pypi
networkx	3.5	pypi_0	pypi
numpy	2.3.4	pypi_0	pypi
openssl	3.0.18	h543e019_0	
packaging	25.0	pypi_0	pypi
pillow	12.0.0	pypi_0	pypi
pip	25.2	pyhc872135_1	
python	3.11.14	h981015d_0	
scikit-image	0.25.2	pypi_0	pypi
scipy	1.16.2	pypi_0	pypi
setuptools	80.9.0	py311haa95532_0	
sqlite	3.50.2	hda9a48d_1	
sympy	1.14.0	pypi_0	pypi
tifffile	2025.10.16	pypi_0	pypi
tk	8.6.15	hf199647_0	
torch	2.9.0	pypi_0	pypi
torch	2.9.0	pypi_0	pypi
torchvision	0.24.0	pypi_0	pypi
tqdm	4.67.1	pypi_0	pypi
typing-extensions	4.15.0	pypi_0	pypi
tzdata	2025b	h04d1e81_0	
ucrt	10.0.22621.0	haa95532_0	
vc	14.3	h2df5915_10	
vc14_runtime	14.44.35208	h4927774_10	
vs2015_runtime	14.44.35208	ha6b5a95_10	
wheel	0.45.1	py311haa95532_0	
xz	5.6.4	h4754444_1	
zlib	1.3.1	h02ab6af_0	

(derain) C:\Users\ROG>

## 2.2

关于深层网络带来梯度消失、训练困难，而Resnet网络就是在解决这个问题，这个网络通过增加Residual Block，其中最关键的一步是在Block中增加一个之前的输入（可以直接是X），这个可以显著减小梯度的下降，同时减少退化问题的发生。

## 3.1

呃，跑的太晚跑不完了

```
-- Epoch 3/100 ---  
100%|██████████████████████████████████████████████████████████████████████████████| 225/225 [01:53<00:00, 1.98it/s, pixel_loss=0.0686]  
EVAL => Avg PSNR: 26.9089 | Avg SSIM: 0.8595 | Avg LPIPS: 0.2114  
=> Saving checkpoint  
  
-- Epoch 4/100 ---  
100%|██████████████████████████████████████████████████████████████████████████████| 225/225 [01:56<00:00, 1.93it/s, pixel_loss=0.0745]  
EVAL => Avg PSNR: 26.8520 | Avg SSIM: 0.8610 | Avg LPIPS: 0.1623  
  
-- Epoch 5/100 ---  
48%|██████████████████████████████████████████████████████████████████████████████| 109/225 [01:10<00:39, 2.91it/s, pixel_loss=0.104]
```

### 3.2

学习到一个torch中一个叫functional的模块，可以不设置参数，对ReLU这样的激活函数很方便。还有Batch Normalization，可以对参数进行批量的归一化，对网络有着积极影响