

# 噪声干扰下的图像恢复重建

## 人工智能基础 —— 实践课（三）

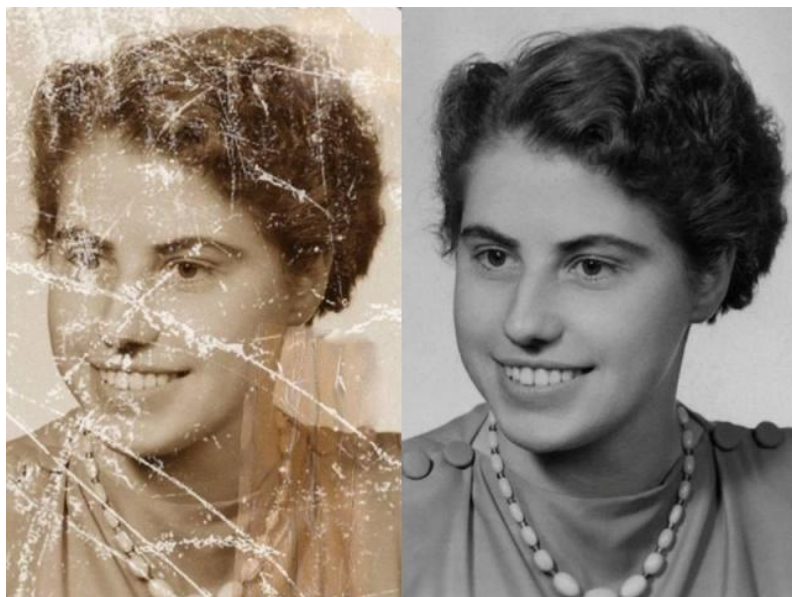


- 图像是一种非常常见的信息载体，但是在图像的获取、传输、存储的过程中可能由于各种原因使得图像受到噪声的影响。
- 如何去除噪声的影响，恢复图像原本的信息是计算机视觉中的重要研究问题。
- 常见的图像恢复算法有基于空间域的中值滤波、基于小波域的小波去噪、基于偏微分方程的非线性扩散滤波等，在本次实验中，我们要对图像添加噪声，并对添加噪声的图像进行基于模型的去噪。

## 图像修复

图像修复（英语：Inpainting）指重建的图像和视频中丢失或损坏的部分的过程。例如在博物馆中，这项工作常由经验丰富的博物馆管理员或者艺术品修复师来进行。数码世界中，图像修复又称图像插值或视频插值，指利用复杂的算法来替换已丢失、损坏的图像数据，主要替换一些小区域和瑕疵。

----- 百度百科



# 图像恢复重建技术

## 空间滤波器

- ◆ **均值滤波器**：算术均值滤波器、几何均值滤波器、谐波均值滤波器、逆谐波均值滤波器等
- ◆ **统计排序滤波器**：中值滤波器、最小值滤波器、最大值滤波器、中点滤波器、修正阿尔法均值滤波器等
- ◆ **自适应滤波器**：自适应中值滤波器等

## 频域滤波器（主要处理周期性噪声干扰）

- ◆ **带阻滤波器**：理想带阻滤波器、巴特沃思带阻滤波器、高斯带阻滤波器
- ◆ **带通滤波器、陷波滤波器等**

## 线性回归

对每个像素点做一次区域二元线性回归：将选定像素点周围指定窗宽大小 $d$ 区域内所有像素点的（横坐标，纵坐标，像素值）加入到训练集中，以像素值为监督信息对横纵坐标做回归训练，然后基于选定像素点的横纵坐标预测其像素值

## 神经网络...

## 空间滤波器

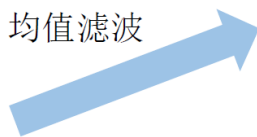
**均值滤波**是典型的线性滤波算法，它是指在图像上对目标像素给一个模板，该模板包括了其周围的临近像素（例：以目标像素为中心的周围8个像素，构成一个滤波模板，即包括目标像素本身），再用模板中的全体像素的平均值来代替原来像素值。

**中值滤波**是一种非线性平滑技术，它将每一像素点的灰度值设置为该点某邻域窗口内的所有像素点灰度值的中值。

---- 百度百科

10	9	4	7	2
4	31	12	23	30
5	43	1	32	8
54	7	3	43	4
23	76	56	45	9

均值滤波



10	9	4	7	2
4	31	12	23	30
5	43	22	32	8
54	7	3	43	4
23	76	56	45	9

中值滤波



10	9	4	7	2
4	31	12	23	30
5	43	23	32	8
54	7	3	43	4
23	76	56	45	9

# 图像恢复重建技术

## 空间滤波器

- ◆ 均值滤波器：算术均值滤波器、几何均值滤波器、谐波均值滤波器、逆谐波均值滤波器等

- ◆ 统计排序滤波器：中值滤波器、最小值滤波器、最大值滤波器、修正

算术均值： $M = \frac{x_1 + x_2 + \dots + x_n}{n}$

- ◆ 自适应滤波器：自适应中值滤波器等

几何均值： $G = \sqrt[n]{x_1 x_2 x_3 \dots x_n}$

- ◆ 带阻滤波器：理想带阻滤波器、巴特沃思带阻滤波器、高斯带阻滤波器

- ◆ 带通滤波器、陷波滤波器等

谐波均值：

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}$$

对每个像素点做一次区域二元线性回归：将选定像素点周围指定窗宽大小d区域内所有像素点的（横坐标，纵坐标，像素值）加入到训练集中，以像素值为监督信息对横纵坐标做回归训练，最后基于选定像素点的横纵坐标预测其像素值

逆谐波均值：

$$B = \frac{x_1^{Q+1} + x_2^{Q+1} + \dots + x_n^{Q+1}}{x_1^Q + x_2^Q + \dots + x_n^Q}$$

神经网络...



# 图像恢复重建技术

## 空间滤波器

- ◆ **均值滤波器**：算术均值滤波器、几何均值滤波器、谐波均值滤波器、逆谐波均值滤波器等
- ◆ **统计排序滤波器**：中值滤波器、最小值滤波器、最大值滤波器、修正阿尔法均值滤波器等
- ◆ **自适应滤波器**：自适应中值滤波器等

## 频域滤波器（主要处理周期性噪声干扰）

- ◆ **带阻滤波器**：理想带阻滤波器、巴特沃思带阻滤波器、高斯带阻滤波器

- ◆ **带通滤波器、陷波滤波器**等
- 修正阿尔法均值：
$$A = \frac{x_{d+1} + x_{d+2} + \dots + x_{n-d}}{n - 2d}$$

## 线性回归

对每个像素点做一次区域二元线性回归：将选定像素点周围指定窗宽大小d区域内所有像素点的（横坐标，纵坐标，像素值）加入到训练集中，以像素值为监督信息对横纵坐标做回归训练，然后基于选定像素点的横纵坐标预测其像素值

## 神经网络...

# 图像恢复重建技术

## 空间滤波器

- ◆ 均值滤波器：算术均值滤波器、几何均值滤波器、谐波均值滤波器、逆谐波均值滤波器等
- ◆ 统计排序滤波器：中值滤波器、最小值滤波器、最大值滤波器、中点滤波器、修正阿尔法均值滤波器等
- ◆ 自适应滤波器：自适应中值滤波器等

自适应中值滤波器：

1.  $x_{\min} = \text{MIN}\{x_1, x_2, \dots, x_n\}, x_{\max} = \text{MAX}\{x_1, x_2, \dots, x_n\}$

$x_{\text{med}} = \text{MED}\{x_1, x_2, \dots, x_n\}, x_{\text{ori}} : \text{初始}x\text{值}, d : \text{窗宽}, T : \text{最大窗宽}$

2.  $a_1 = x_{\text{med}} - x_{\min}, a_2 = x_{\text{med}} - x_{\max}$

if  $a_1 > 0$  and  $a_2 < 0$ : 进入步骤3

else:  $d = d + 1$

if  $d < T$ : 重复步骤2

else: 输出  $x_{\text{med}}$

3.  $b_1 = x_{\text{ori}} - x_{\min}, b_2 = x_{\text{ori}} - x_{\max}$

if  $b_1 > 0$  and  $b_2 < 0$ : 输出  $x_{\text{ori}}$

else: 输出  $x_{\text{med}}$

神经网络...

# 图像恢复重建技术

## 空间滤波器

- ◆ **均值滤波器**：算术均值滤波器、几何均值滤波器、谐波均值滤波器、逆谐波均值滤波器等
- ◆ **统计排序滤波器**：中值滤波器、最小值滤波器、最大值滤波器、中点滤波器、修正阿尔法均值滤波器等
- ◆ **自适应滤波器**：自适应中值滤波器等

## 频域滤波器（主要处理周期性噪声干扰）

- ◆ **带阻滤波器**：理想带阻滤波器、巴特沃思带阻滤波器、高斯带阻滤波器
- ◆ **带通滤波器、陷波滤波器等**

## 线性回归

对每个像素点做一次区域二元线性回归：将选定像素点周围指定窗宽大小 $d$ 区域内所有像素点的（横坐标，纵坐标，像素值）加入到训练集中，以像素值为监督信息对横纵坐标做回归训练，然后基于选定像素点的横纵坐标预测其像素值

## 神经网络...



## 频域滤波器

—— 某些在空间域表述困难的增强任务，  
在频率域中变得非常普通。

二维离散傅立叶变换: 
$$F(u, v) = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N f(x, y) e^{-i2\pi(ux/M + vy/N)}$$

二维离散傅立叶逆变换: 
$$f(x, y) = \sum_{u=1}^M \sum_{v=1}^N F(u, v) e^{i2\pi(ux/M + vy/N)}$$

卷积定理: 
$$f(x, y) * h(x, y) \Leftrightarrow F(u, v) H(u, v)$$

$$f(x, y) h(x, y) \Leftrightarrow F(u, v) * H(u, v)$$

$$f(x, y) * h(x, y) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N f(m, n) h(x-m, y-n)$$

等价于: 
$$F(u, v) H(u, v)$$

矩阵乘法

# 图像恢复重建技术

## 空间滤波器

- ◆ 均值滤波器：算术均值滤波器、几何均值滤波器、谐波均值滤波器、逆谐波均值滤波器等
- ◆ 统计排序滤波器：中值滤波器、最小值滤波器、最大值滤波器、中点滤波器、修正阿尔法均值滤波器等
- ◆ 自适应滤波器：自适应中值滤波器等

## 频域滤波器（主要处理周期性噪声干扰）

- ◆ 带阻滤波器：理想带阻滤波器、巴特沃思带阻滤波器、高斯带阻滤波器
- ◆ 带通滤波器、陷波滤波器等

## 线性回归

对每个像素点做一次区域二元线性回归：将选定像素点周围指定窗宽大小 $d$ 区域内所有像素点的（横坐标，纵坐标，像素值）加入到训练集中，以像素值为监督信息对横纵坐标做回归训练，然后基于选定像素点的横纵坐标预测其像素值

## 神经网络...

## 生成受损图像

- a) 受损图像 ( $X$ ) 是由原始图像 ( $I \in \mathcal{R}^{H*W*C}$ ) 添加了不同噪声遮罩 (noise masks) ( $M \in \mathcal{R}^{H*W*C}$ ) 得到的 ( $X = I \odot M$ )，其中  $\odot$  是逐元素相乘。
- b) 噪声遮罩仅包含  $\{0,1\}$  值。对原图的噪声遮罩的可以RGB通道分别以 0.8/0.4/0.6 的噪声概率产生的，即噪声遮罩RGB通道的像素值各有 80%/40%/60% 的概率值为0，否则为1。

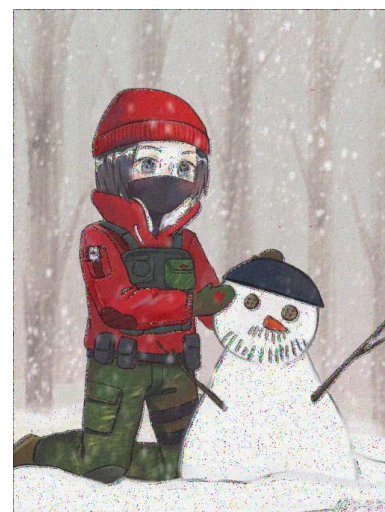
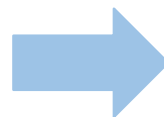
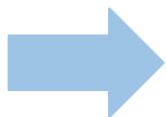


$$X = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}_{H*W*3} =$$



## 回归模型

对每个像素点做一次区域二元线性回归：将选定像素点周围指定窗宽大小  $d$  区域内所有像素点的（横坐标，纵坐标，像素值）加入到训练集中，以像素值为监督信息对横纵坐标做回归训练，然后基于选定像素点的横纵坐标预测其像素值。



## 示例

### 1. 将图像转为数值矩阵



$\frac{\text{img}}{255}$



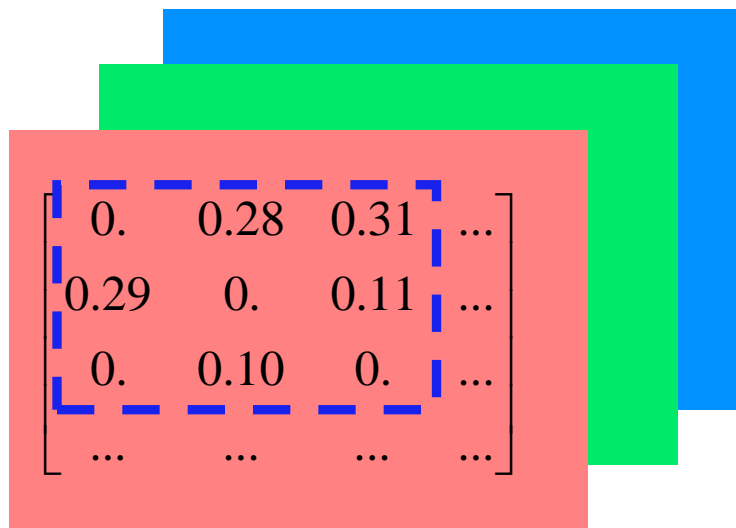
$$\begin{bmatrix} 0.44 & 0.28 & 0.31 & \dots \\ 0.29 & 0.10 & 0.11 & \dots \\ 0.31 & 0.10 & 0.13 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

### 2. 随机生成0-1遮罩

$$\begin{bmatrix} 0.44 & 0.28 & 0.31 & \dots \\ 0.29 & 0.10 & 0.11 & \dots \\ 0.31 & 0.10 & 0.13 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 & 1 & \dots \\ 1 & 0 & 1 & \dots \\ 0 & 1 & 0 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} = \begin{bmatrix} 0. & 0.28 & 0.31 & \dots \\ 0.29 & 0. & 0.11 & \dots \\ 0. & 0.10 & 0. & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

## 示例

3. 设定窗宽( $d = 3$ ), 选中区域



$$\begin{bmatrix} 0. & 0.28 & 0.31 & \dots \\ 0.29 & 0. & 0.11 & \dots \\ 0. & 0.10 & 0. & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

对应坐标:

$$\begin{bmatrix} (0,0) & (0,1) & (0,2) & \dots \\ (1,0) & (1,1) & (1,2) & \dots \\ (2,0) & (2,1) & (2,2) & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

4. 展平, 并组合成三元组对

$\{(0, 0, 0.), (0, 1, 0.28), (0, 2, 0.31), (1, 0, 0.29), (1, 1, 0.), (1, 2, 0.11), (2, 0, 0.), (2, 1, 0.10), (2, 2, 0.), \dots\}$

5. 提出遮罩噪声为0的位置的数据对

$\{(0, 1, 0.28), (0, 2, 0.31), (1, 0, 0.29), (1, 2, 0.11), (2, 1, 0.10), \dots\}$



## 示例

6. 用有效数据的 $(x, y)$ 去预测像素点值

$\{(0, 1, 0.28), (0, 2, 0.31), (1, 0, 0.29), (1, 2, 0.11), (2, 1, 0.10), \dots\}$

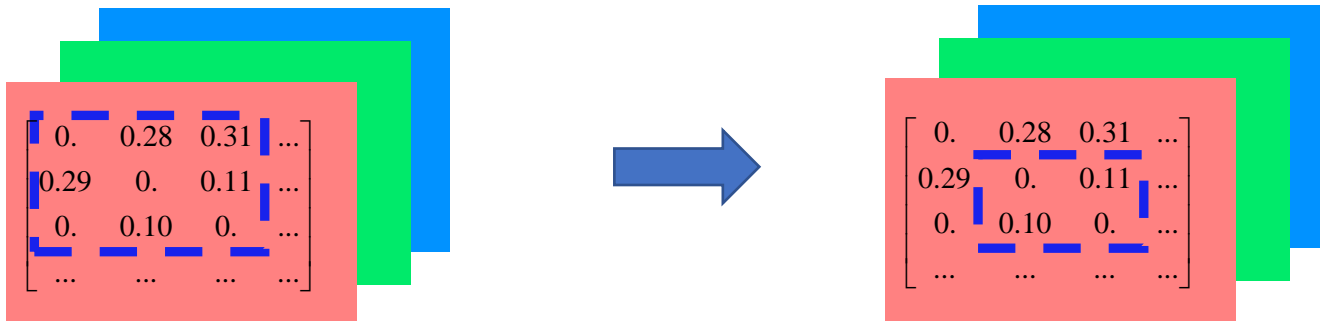


$(x, y, \text{values})$



$(3, 3, 0.21)$

7. 逐元素遍历拟合各个像素点窗宽区域内的 $(x, y, \text{value})$ ，并以预测值替换中心像素值



## 参考学习内容

### 回归模型:

1. 线性回归

`sklearn.linear_model.LinearRegression`

2. 岭回归

`sklearn.linear_model.Ridge`

3. Lasso回归

`sklearn.linear_model.Lasso`

### 损失函数:

Norm\_0、Norm\_1、Norm\_2、Norm\_∞

Cosine 相似度、SSIM 相似度