

下采样、误差分析展示和可视化都已完成。

上采样

1. 最近邻上采样
2. 双线性插值法 (Bilinear Interpolation Upsampling 作业一里面好像是有的)
3. 三次插值上采样 (Bicubic Interpolation Upsampling 用三结合函数拟合16x16)
4. 反卷积、转置卷积上采样 (Deconvolution / Transposed Convolution / Fractionally-Strided Convolution) (可学习)
5. 深度学习上采样 (尽量实现包括基于超分辨率网络、Pixel Shuffle/Subpixel Convolution, 深度残差网络)

1. 用 PIL.Image 读取图像(uint8) 并转成浮点数 (/255)
2. 函数的返回值也是 np.ndarray
3. 要可以处理长方形图像和彩色图像
4. 非深度学习的的方法要可以实现非整数倍率 (例如2.5)

代码风格说明

加载图像以及可视化 (可视化传入的是 numpy 数组的列表)

```
1 def load_image(image_path):
2     image = Image.open(image_path)
3     return np.array(image) / 255
4
5 def show(images, titles=None, cmap='gray'):
6     if isinstance(images, np.ndarray):
7         images = [images]
8     num_img = len(images)
9     if isinstance(titles, str):
10        titles = [titles] * num_img
11    if titles is not None:
12        assert len(titles) == len(images), 'Titles length must match images
length'
13
14    fig, axes = plt.subplots(1, num_img, figsize=(3 * num_img, 4))
15    if num_img == 1:
16        axes = [axes]
17
18    for i, (ax, img) in enumerate(zip(axes, images)):
19        if len(img.shape) == 2:
20            ax.imshow(img, cmap='gray')
21        else:
22            ax.imshow(img)
23
```

```

24         if titles is not None:
25             ax.set_title(titles[i])
26     plt.tight_layout()
27     plt.show()
28
29     img = load_image("./asserts/VanGogh.jpg")
30     show(img, 'van gogh.png')
31     scale_factor = 1.5

```

具体函数风格，非深度学习方法的开头如下

```

1  def upsample(image, scale_factor):
2      assert scale_factor > 1 , "scale factor must be greater than 1"
3      # width, height, channels
4      h, w = image.shape[:2]
5      c = image.shape[2] if len(image.shape) == 3 else 1
6      new_w, new_h = int(w * scale_factor), int(h * scale_factor)
7      print(f"{h}x{w} -> {new_h}x{new_w}")
8
9      if c == 1:
10         new_image = np.zeros((new_h, new_w), dtype=image.dtype)
11     else:
12         new_image = np.zeros((new_h, new_w, c), dtype=image.dtype)
13
14     for new_y in range(new_h):
15         for new_x in range(new_w):
16             # 像素操作
17     return new_image

```