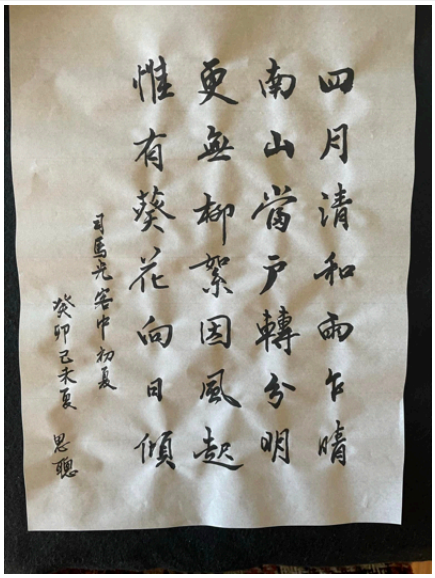
Digital Image Processing

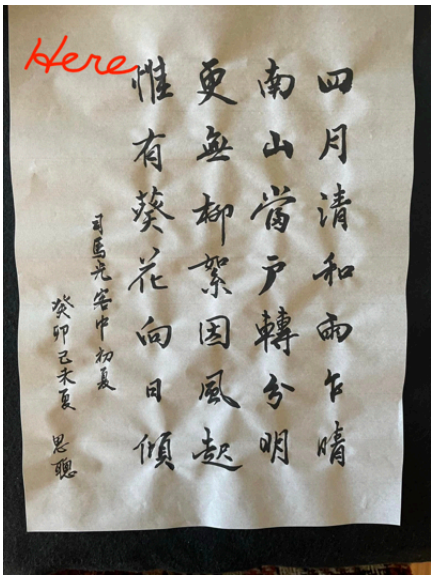
Goal:

Demo load digital image as data matrix

Matrix subtraction to get RGB intensity difference

save the difference into new image file

1. Requirement: Import PIL : Python Image Library
2. test1.jpg is opened and loaded into Numpy array with shape (2048, 1536, 3) representing 2048 by 1536 pixel with RGB intensity
3. code: `difference_matrix = np.abs(matrix1 - matrix2)`
4. test1.jpg 

5. test2.jpg 

6. different image 

In [ ]:
```python
from PIL import Image
import numpy as np

def load_image(image_path):
    img = Image.open(image_path)
    return img

def resize_image(img, target_size=(256, 256)):
    img_resized = img.resize(target_size, Image.BICUBIC)
    return img_resized

def image_to_matrix(img):
    img_array = np.array(img)
    return img_array

def save_difference_matrix_image(matrix, output_path='difference_image.jpg')
```

```
    normalized_matrix = (matrix / matrix.max() * 255).astype(np.uint8)
    difference_image = Image.fromarray(normalized_matrix)
    difference_image.save(output_path)

def main():
    image_path1 = './test1.jpg'
    target_size=(1536, 2048)
    img1 = Image.open(image_path1)
    img1_resized = resize_image(img1, target_size)

    matrix1 = np.array(img1_resized)
    #print(matrix1)
    print("Matrix 1 from Image has shape of ", matrix1.shape)

    image_path2 = './test2.jpg'
    img2 = Image.open(image_path2)
    img2_resized = resize_image(img2, target_size)

    matrix2 = np.array(img2_resized)
    print("Matrix 2 from Image has shape of ", matrix2.shape)

    # compute the different in two matrices
    difference_matrix = np.abs(matrix1 - matrix2)
    save_difference_matrix_image(difference_matrix)
    print("Saving different RGB pixel into ./difference_image.jpg . . .")

if __name__ == "__main__":
    main()
```

```
Matrix 1 from Image has shape of  (2048, 1536, 3)
Matrix 2 from Image has shape of  (2048, 1536, 3)
Saving different RGB pixel into ./difference_image.jpg . . .
```