《数据科学与工程算法》项目报告

报告题目：图像压缩

姓　　名：李芳

学　　号：10214602404

完成日期：2024.06.28

摘要 [中文]：

图像压缩属于图像处理中的重要研究领域之一，如何在减少图像存储和传输所需的空间和带宽的同时尽量保持图像质量是该领域研究重难点，PCA 是实现这一目标的有效方法，能够在压缩的同时保持图像的主要信息。本次实验实现基于主成分分析（PCA）的图像压缩算法，然后对压缩前后的图像进行多方面评价：重构误差（均方误差 MSE、峰值信噪比 PSNR）、空间节省、压缩比例、运行时间等。该测试运行基于农业、飞机和海滩图像类别，均能得出 PCA 在保持图像主要特征的同时，显著减少图像的存储空间的结论。该实验展示了 PCA 在图像压缩中的潜力，但仍有需要改进和优化的地方，比如并行计算技术等，后续研究可基于本次实验进一步深入。

Abstract [English]

Image compression is one of the important research fields in image processing. How to reduce the space and bandwidth required for image storage and transmission while maintaining image quality as much as possible is a key and difficult research area in this field. PCA is an effective method to achieve this goal, which can maintain the main information of the image while compressing. The aim of this experiment is to implement an image compression algorithm based on Principal Component Analysis (PCA), and then evaluating the images before and after compression from multiple aspects, including reconstruction error (mean square error MSE, peak signal-to-noise ratio PSNR), space saving, compression ratio, running time, etc. This test run is based on multiple image categories, including agricultural, aircraft, and beach images, and can conclude that PCA significantly reduces the storage space of images while maintaining the main features of the images. This experiment demonstrates the potential of combining the two in image compression, but there is still room for improvement and optimization, such as parallel computing technology. Further research can be conducted based on this experiment.

一、 项目概述

- **科学价值与相关研究工作**

图像压缩是图像处理中的重要研究领域，旨在减少图像存储和传输所需的空间和带宽的同时，尽量保持图像质量。传统的图像压缩方法，如 JPEG、PNG 等，在压缩效率和图像质量之间取得了一定的平衡，但面对越来越大的数据量和更高的压缩需求，我们需要探索新的算法来进一步提高压缩效率和保持图像质量。这不仅可以节省存储空间和传输带宽，还可以在大数据和人工智能时代下，为数据分析和机器学习提供更高效的支持。因此，开发和优化新的图像压缩算法成为当前研究的重点。

主成分分析（PCA）是一种常用的数据降维技术，主要用来减少数据集的维度，同时尽量保留原始数据中的重要信息。它通过找出数据中的主成分（数据变动最大的方向），将数据投影到这些方向上，从而达到降维的目的。主要步骤：首先，将数据标准化，即减去平均值并除以标准差，使得数据的每个维度都具有相同的尺度。接着，计算数据的协方差矩阵，它表示数据中每两个维度之间的相关性。然后，计算协方差矩阵的特征值和特征向量。特征值表示数据在特征向量方向上的方差大小，特征向量则表示新的坐标轴方向。选择前几个特征值最大的特征向量作为主成分。这些主成分包含了原始数据中大部分的信息。最后，将原始数据投影到选定的主成分上，从而得到降维后的数据。

- **项目主要内容**

本项目的主要内容包括实现基于 PCA 的图像压缩算法，并对压缩前后的图像进行多维度的评价。具体内容包括以下几个方面：

1. 实现图像通道的去中心化处理和 PCA 降维算法：

   (1) 针对每张 RGB 图像的每个通道（例如红色、绿色和蓝色通道）进行去中心化处理，通过减去每个通道的平均值，使数据更适合后续的 PCA 处理。

(2) 应用 PCA 算法对去中心化的通道数据进行降维处理，找出数据中变动最大的方向，将数据投影到这些方向上，从而减少数据维度。阈值选取 90%。

2. 加载多种类型的图像，并应用 PCA 进行压缩和重构：

(1) 基于农业、飞机和海滩图像类别，因为这些图像具有不同的特征和复杂性，能够全面测试 PCA 压缩算法的性能。

(2) 对每张图像应用 PCA 进行压缩和重构，生成压缩后的图像。

3. 计算重构误差（MSE、PSNR）、空间节省、压缩比例和运行时间等指标：

(1) 为全面评估 PCA 压缩算法的性能，计算重构误差（均方误差 MSE 和峰值信噪比 PSNR），以评估压缩后图像的质量。

(2) 计算空间节省和压缩比例，以评估压缩算法在减少存储空间方面的效果。

(3) 记录算法的运行时间，以评估其计算效率。

4. 分析各类图像在压缩前后的评价指标，并总结实验结果：

(1) 对每类图像的压缩效果进行了详细分析，包括各项评价指标的统计结果和对比，以及使用 excel 表格生成图表，将评估结果可视化。

(2) 总结实验结果，探讨 PCA 压缩算法在不同类型图像上的表现，并提出改进建议和未来研究方向。

二、 问题定义（提供问题定义的语言描述与数学形式）

- **语言描述**
  - 通过 PCA 算法对图像进行压缩，使得压缩后的图像在视觉上与原图像尽量接近，同时显著减少存储空间和传输带宽。通过对数据矩阵进行奇异值分解（SVD），找到数据变化最大的方向，将原始数据投影到这些方向上，从而实现降维和压缩。

- 定义一系列评价指标，以全面评估 PCA 压缩算法的性能。包括重构误差、峰值信噪比、空间节省、压缩比例和运行时间等。具体来说：重构误差（MSE）：重构误差用于衡量压缩后图像与原始图像之间的差异。
  - 均方误差：表示原始图像和重构图像之间的平均平方误差。MSE 值越小，表示重构图像与原始图像越接近，压缩质量越高。
  - 峰值信噪比：用于衡量图像质量，表示图像的最大像素值与重构误差之间的比值。PSNR 通常用分贝（dB）表示，值越大，表示重构图像与原始图像越接近，压缩质量越高。
  - 空间节省：表示压缩后图像相对于原始图像在存储空间上的减少比例。空间节省值越大，表示压缩效果越好。
  - 压缩比例：表示压缩后图像的存储大小与原始图像存储大小的比值。压缩比例越小，表示压缩效果越好。
  - 运行时间：表示算法从开始到完成所需的总时间。运行时间越短，表示算法的计算效率越高。对于大规模数据集，运行时间是一个关键的考虑因素，显示了是否显著提高数据处理的整体效率。

- **数学形式**
  - PCA：

给定一个图像矩阵 X，我们将其分解为若干通道，每个通道 X_c 经过PCA处理后得到降维后的矩阵 X_c' 和重构后的矩阵 $\hat{X}_c$。对于每个通道，我们有：

$$\hat{X}_c = U_c S_c V_c^T + \mu_c$$

其中 U_c, S_c, V_c 分别为奇异值分解（SVD）的结果，μ_c 为通道均值。

  - 评估指标：
    - MSE：

$$\mathrm{MSE}(X, \hat{X}) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2$$

其中，X 表示原始图像，$\hat{X}$ 表示重构图像，n 表示图像中像素的数量。MSE值越小，表示重构图像与原始图像越接近，压缩质量越高。

- PSNR：

$$\mathrm{PSNR}(X, \hat{X}) = 20\log_{10}\left(\frac{\mathrm{MAX}_X}{\sqrt{\mathrm{MSE}(X, \hat{X})}}\right)$$

其中，MAX_X 是图像像素值的最大值（通常为255）。PSNR值越大，表示重构图像与原始图像越接近，压缩质量越高。

- 空间节省：

$$\text{Space Saving} = \frac{\text{Original Size} - \text{Compressed Size}}{\text{Original Size}}$$

其中，Original Size 是原始图像的存储大小，Compressed Size 是压缩后图像的存储大小。空间节省值越大，表示压缩效果越好。

- 压缩比例：

$$\text{Compression Ratio} = \frac{\text{Compressed Size}}{\text{Original Size}}$$

压缩比例越小，表示压缩效果越好。

- 运行时间：

$$\text{Run Time} = t_{\text{end}} - t_{\text{start}}$$

其中，t_{start} 表示算法开始时间，t_{end} 表示算法结束时间。

三、 方法

1. 导入所需库：

遍历访问被处理文件夹以及保存文件夹的 os 库；支持数学计算以及图像矩阵暂存的

numpy 库；图像处理的 PIL 库；存放转化表格数据的 pandas 库；计时的 time 库。

```
import os
import numpy as np
from PIL import Image
import pandas as pd
import time
```

2. 通道矩阵去中心化：

接收一个参数 channel（二维数组，表示 RGB 图像的某一个颜色通道（红色、绿色或蓝色））；首先，计算 channel 矩阵每一行的平均值保存在 matrix_mean 中，保持结果的二维结构；然后，对 channel 矩阵的每个元素减去对应行的均值，得到去中心化后的矩阵 matrix_centered；最后，返回两个值：去中心化后的矩阵 matrix_centered 和每行的均值 matrix_mean。

```
# 通道矩阵求均值，去中心化
def process_channel(channel):  1 个用法
    matrix_mean = np.mean(channel, axis=1, keepdims=True)
    matrix_centered = channel - matrix_mean
    return matrix_centered, matrix_mean
```

3. 进行 PCA 计算降维：

接收两个参数：channel_data 作为输入数据，explained_variance_ratio 作为希望保留的解释方差比例；

首先，使用 np.linalg.svd 对输入数据 channel_data 进行奇异值分解（省略零填充），得到矩阵 U、奇异值 S 和矩阵 Vt；然后，计算每个奇异值对应的特征值以及解释方差的累积比例，找到累积解释方差比例达到设定阈值 explained_variance_ratio 的最小主成分数量；接着，保留前 n_feature 个主成分对应的 U 矩阵列、奇异值、Vt 矩阵行；

最后，将原始数据投影到前 n_feature 个主成分上，返回降维后的数据 pca_data、保留的特征向量 Vt_reduced 和保留的主成分数量 n_feature。

```python
# 实现PCA降维，保留特征向量数自行设置
def PCA_compute(channel_data, explained_variance_ratio):  1 个用法
    U, S, Vt = np.linalg.svd(channel_data, full_matrices=False)
    eigenvalues = S ** 2 / (len(S) - 1)
    ratio = np.cumsum(eigenvalues) / np.sum(eigenvalues)
    n_feature = np.argmax(ratio >= explained_variance_ratio) + 1
    U_reduced = U[:, :n_feature]
    S_reduced = np.diag(S[:n_feature])
    Vt_reduced = Vt[:n_feature, :]
    pca_data = np.dot(U_reduced, S_reduced)
    return pca_data, Vt_reduced, n_feature
```

4. 评估计算函数：

MSE&PSNR，根据第二部分公式计算即可，都是输入原始图像矩阵值以及压缩后的图像矩阵值。

```python
# 计算MSE
def calculate_mse(original, reconstructed):  2 个用法
    return np.mean((original - reconstructed) ** 2)
```

```python
# 计算PSNR
def calculate_psnr(original, reconstructed):  1 个用法
    mse = calculate_mse(original, reconstructed)
    if mse == 0:
        return 100
    max_pixel = 255.0
    return 20 * np.log10(max_pixel / np.sqrt(mse))
```

5. 遍历文件，执行 PCA 并保存图像函数：

接收三个参数：input_folder 输入图像文件夹、output_folder 输出图像文件夹、explained_variance_ratio 解释方差比例

先进行输出文件夹存在性，如果不存在，创建对应文件夹；初始化一个空列表 results，存储每个图像的评价指标。然后，使用 os.walk 遍历输入文件夹中的所有文件，如果文件以.tif 结尾，就进行 PCA 处理：

(1) 构建输入图像文件的完整路径 input_path 和输出图像文件的完整路径 output_path，使用 PIL 库打开图像文件，并将其转换为 NumPy 数组 image，获取图像的高度、宽度和通道数。

(2) 初始化一个与原图像大小相同的数组 reconstructed_image，用于存储重构后的图像。记录开始时间，用于计算运行时间。

(3) 遍历每个通道，依次对每个通道进行 PCA 处理：

① 提取通道的数据 channel_data，调用去中心化处理函数，得到 channel_centered 和 channel_mean。

② 调用 PC 计算函数，得到降维后的数据 pca_channel_data、特征向量 eigenvector_subset 和保留的主成分数量 n_feature。

③ 使用特征向量和去中心化数据重构通道数据 reconstructed_channel，并加上均值向量。

④ 将重构后的通道数据存储到 reconstructed_image 中。

(4) 记录结束时间。将重构后的图像数据限制在 0 到 255 之间，并转换为无符号 8 位整数类型，并使用 PIL 库保存重构后的图像到 output_path。

(5) 计算各种评价指标：

① mse：原始图像和重构图像之间的均方误差。

② psnr：原始图像和重构图像之间的峰值信噪比。

③ original_size：原始图像的文件大小。

④ compressed_size：重构图像的文件大小。

⑤ compression_ratio：压缩比例，压缩后图像的大小与原始图像大小的比值。

⑥ space_saving：空间节省比例，压缩后图像相比原始图像减少的存储空间比例。

⑦ run_time：算法运行时间。

(6) 将每个图像的评价指标存储到 results 列表中，将列表转换为 PandasDataFrame，

保存到 CSV 文件 pca_evaluation_results.csv 中，用于结果分析

(7) 返回 DataFrame results_df。

```python
# 加载图像并应用PCA，并保存重构后的图像
def apply_and_save_PCA_images(input_folder, output_folder, explained_variance_ratio):  1 个用法
    if not os.path.exists(output_folder):
        os.makedirs(output_folder)
    results = []
    for root, _, files in os.walk(input_folder):
        for file in files:
            if file.endswith('.tif'):
                input_path = os.path.join(root, file)
                output_path = os.path.join(output_folder, file)
                image = np.array(Image.open(input_path))

                height, width, channel = image.shape
                reconstructed_image = np.zeros_like(image, dtype=np.float64)

                start_time = time.time()

                for c in range(channel):
                    channel_data = image[:, :, c]
                    channel_centered, channel_mean = process_channel(channel_data)
                    pca_channel_data, eigenvector_subset, n_feature = PCA_compute(channel_centered,
                                                                                  explained_variance_ratio)
                    reconstructed_channel = np.dot(pca_channel_data, eigenvector_subset) + channel_mean
                    reconstructed_image[:, :, c] = reconstructed_channel

                end_time = time.time()
                reconstructed_image = np.clip(reconstructed_image, a_min: 0, a_max: 255).astype(np.uint8)
                Image.fromarray(reconstructed_image).save(output_path)

                mse = calculate_mse(image, reconstructed_image)
                psnr = calculate_psnr(image, reconstructed_image)
                original_size = os.path.getsize(input_path)
                compressed_size = os.path.getsize(output_path)
                compression_ratio = compressed_size / original_size
                space_saving = (original_size - compressed_size) / original_size
                run_time = end_time - start_time
```

```python
                results.append({
                    'Image': file,
                    'MSE': mse,
                    'PSNR': psnr,
                    'Original Size': original_size,
                    'Compressed Size': compressed_size,
                    'Compression Ratio': compression_ratio,
                    'Space Saving': space_saving,
                    'Run Time': run_time
                })

    results_df = pd.DataFrame(results)
    results_df.to_csv(os.path.join(output_folder, 'pca_evaluation_results.csv'), index=False)
    return results_df
```

6. 主函数：

对 0.05-0.9 的解释方差比例数组，进行三个文件夹所有图像的 PCA 压缩，结果输出到指定文件夹中，并将评估指标合并打印输出，方便接下来的评估和观测用。

```python
if __name__ == '__main__':
    input_base_folder = 'E:/pythonProject/DaAlhw2/Images'
    output_base_folder = 'E:/pythonProject/DaAlhw2'
    categories = ['agricultural', 'airplane', 'beach']

    explained_variance_ratios=[0.05,0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
    for category in categories:
        for ratio in explained_variance_ratios:
            input_folder = os.path.join(input_base_folder, category)
            output_folder = os.path.join(output_base_folder, f'Images_rev_{ratio}', category)
            apply_and_save_PCA_images(input_folder, output_folder, ratio)
            print(f"finish{ratio}")
```

四、 实验结果

1. 首先，对三类图像：农业、飞机和海滩都进行了压缩和评估。对每个阈值压缩的每类图像的评估取平均值，取 0.9 阈值的操作,展示：

- **农业图像**

97	agricultural95.	13.53468831	36.81632102	207192	196748	0.949592648	0.050407352	0.042435884
98	agricultural96.	27.13817342	33.79499747	207744	196748	0.94706947	0.05293053	0.046673536
99	agricultural97.	25.00721741	34.15014991	207996	196748	0.945922037	0.054077963	0.16332221
100	agricultural98.	48.59203084	31.26515311	209152	196748	0.940693849	0.059306151	0.071761847
101	agricultural99.	59.27294922	30.40223824	208872	196748	0.941954881	0.058045119	0.041569233
102								
103	Average	30.69035543	34.14461384	208287.8	196748	0.944646208	0.055353792	0.106192079
104								
105								
106								

  - MSE：农业图像在重构后的均方误差平均为 30.69035543，重构图像与原图像之间存在一定的差异。
  - PSNR：农业图像的峰值信噪比平均为 34.14461384 dB，图像质量最好。
  - 空间节省：压缩后，农业图像的存储空间减少了约 5.5%。
  - 压缩比例：平均压缩比例为 0.945，表明压缩后图像占用的存储空间约为原始图像的 94.5%。
  - 运行时间：每张农业图像的处理时间平均为 0.106 秒，三者之中最慢

- **飞机图像**

98	airplane96.tif	45.37090556	31.56302913	209108	196748	0.940891788	0.059108212	0.045491695
99	airplane97.tif	45.42281596	31.55806306	209080	196748	0.941017792	0.058982208	0.040261984
100	airplane98.tif	35.53245036	32.62455203	208068	196748	0.945594709	0.054405291	0.044331551
101	airplane99.tif	55.23736064	30.70847442	209276	196748	0.94013647	0.05986353	0.040397167
102								
103	**Average**	**55.32734806**	**30.87467266**	**208686.28**	**196678.88**	**0.942480596**	**0.057519404**	**0.045180261**
104								
105								

- MSE：飞机图像在重构后的均方误差平均为 55.32734806，重构质量三者之中较差。
- PSNR：飞机图像的峰值信噪比平均为 30.87467266 dB，图像质量三者之中较差
- 空间节省：压缩后，飞机图像的存储空间减少了约 5.8%。
- 压缩比例：平均压缩比例为 0.942，表明压缩后图像占用的存储空间约为原始图像的 94.2%。
- 运行时间：每张飞机图像的处理时间平均为 0.045 秒。

- **海滩图像**

94	beach92.tif	36.90684001	32.45973499	205648	196748	0.956722166	0.043277834	0.045342207
95	beach93.tif	56.02211507	30.6472086	207144	196748	0.949812691	0.050187309	0.043094873
96	beach94.tif	49.13360087	31.21701767	207652	196748	0.947489068	0.052510932	0.043331146
97	beach95.tif	48.8183492	31.24497271	207344	196748	0.94889652	0.05110348	0.04238534
98	beach96.tif	51.432724	31.01840835	207512	196748	0.948128301	0.051871699	0.049487352
99	beach97.tif	55.83385213	30.66182769	207724	196748	0.947160655	0.052839345	0.042020559
100	beach98.tif	50.73205566	31.07797901	207776	196748	0.94692361	0.05307639	0.050470114
101	beach99.tif	55.45212809	30.69162143	207588	196748	0.947781182	0.052218818	0.054630518
102								
103	**Average**	**36.36950511**	**32.96739876**	**206642.56**	**196717.28**	**0.951991358**	**0.048008642**	**0.052990184**
104								
105								
106								

- MSE：海滩图像在重构后的均方误差平均为 36.36950511，重构图像与原图像之间存在一定的差异。
- PSNR：海滩图像的峰值信噪比平均为 32.96739876 dB，图像质量较好。
- 空间节省：压缩后，海滩图像的存储空间减少了约 4.8%。
- 压缩比例：平均压缩比例为 0.952，表明压缩后图像占用的存储空间约为原始图像的 95.2%。
- 运行时间：每张海滩图像的处理时间平均约为 0.052 秒。

2. 然后，对三类图像，每个阈值对应的评估指标平均值汇总，均值汇总结果保存在 pca_evaluation_result.xlsx 中，以便进行分析：

- **农业图像**

explained_variance_ratios	MSE	PSNR	Original Size	Compressed Size	Compression Ratio	Space Saving	Run Time
0.05	72.8256956	29.66120713	208287.8	196748	0.944646208	0.055353792	0.042930441
0.1	72.68226135	29.6716878	208287.8	196748	0.944646208	0.055353792	0.0428283
0.2	71.57387655	29.75094725	208287.8	196748	0.944646208	0.055353792	0.044966552
0.3	69.28169439	29.91679776	208287.8	196748	0.944646208	0.055353792	0.043905313
0.4	65.9715564	30.16082872	208287.8	196748	0.944646208	0.055353792	0.045053616
0.5	61.03512054	30.56096517	208287.8	196748	0.944646208	0.055353792	0.063810821
0.6	55.14197652	31.04054976	208287.8	196748	0.944646208	0.055353792	0.048730495
0.7	50.0162384	31.5321785	208287.8	196748	0.944646208	0.055353792	0.045274167
0.8	42.95049586	32.33690144	208287.8	196748	0.944646208	0.055353792	0.046144617
0.9	30.69035543	34.14461384	208287.8	196748	0.944646208	0.055353792	0.047170208

- **飞机图像**

explained_variance_ratios	MSE	PSNR	Original Size	Compressed Size	Compression Ratio	Space Saving	Run Time
0.05	83.92575601	28.94571442	208686.28	196678.88	0.942480596	0.057519404	0.049431863
0.1	83.92575601	28.94571442	208686.28	196678.88	0.942480596	0.057519404	0.050511701
0.2	83.64970649	28.96263059	208686.28	196678.88	0.942480596	0.057519404	0.055205321
0.3	82.09626008	29.04823102	208686.28	196678.88	0.942480596	0.057519404	0.050482671
0.4	80.41827481	29.14425294	208686.28	196678.88	0.942480596	0.057519404	0.061203811
0.5	78.18947395	29.27333701	208686.28	196678.88	0.942480596	0.057519404	0.052767949
0.6	75.52679432	29.43127237	208686.28	196678.88	0.942480596	0.057519404	0.051899629
0.7	72.13142608	29.6433607	208686.28	196678.88	0.942480596	0.057519404	0.053196492
0.8	66.22220607	30.03844737	208686.28	196678.88	0.942480596	0.057519404	0.054614139
0.9	55.32734806	30.87467266	208686.28	196678.88	0.942480596	0.057519404	0.05514792

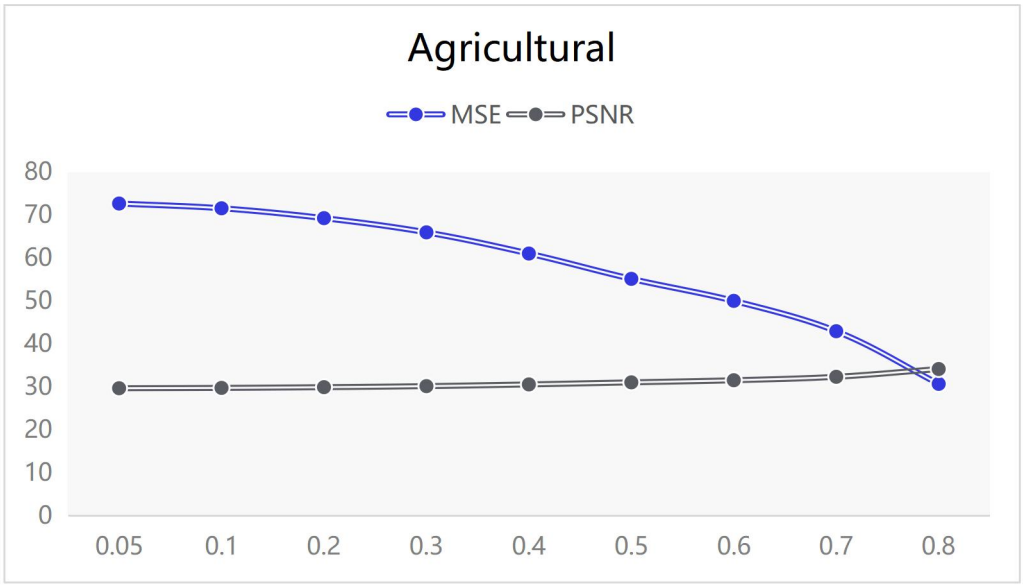
- **海滩图像**

explained_variance_ratios	MSE	PSNR	Original Size	Compressed Size	Compression Ratio	Space Saving	Run Time
0.05	64.67127129	30.36678946	206642.56	196717.28	0.951991358	0.048008642	0.052849915
0.1	64.67127129	30.36678946	206642.56	196717.28	0.951991358	0.048008642	0.054310358
0.2	64.63887271	30.3687077	206642.56	196717.28	0.951991358	0.048008642	0.052251856
0.3	64.36883013	30.38589577	206642.56	196717.28	0.951991358	0.048008642	0.051896284
0.4	63.06341581	30.46805543	206642.56	196717.28	0.951991358	0.048008642	0.053455145
0.5	61.37923765	30.58068693	206642.56	196717.28	0.951991358	0.048008642	0.052791576
0.6	58.55806694	30.79048622	206642.56	196717.28	0.951991358	0.048008642	0.053022768
0.7	54.59319431	31.10938961	206642.56	196717.28	0.951991358	0.048008642	0.052790589
0.8	47.32587052	31.74680165	206642.56	196717.28	0.951991358	0.048008642	0.053721914
0.9	36.36950511	32.96739876	206642.56	196717.28	0.951991358	0.048008642	0.054222431

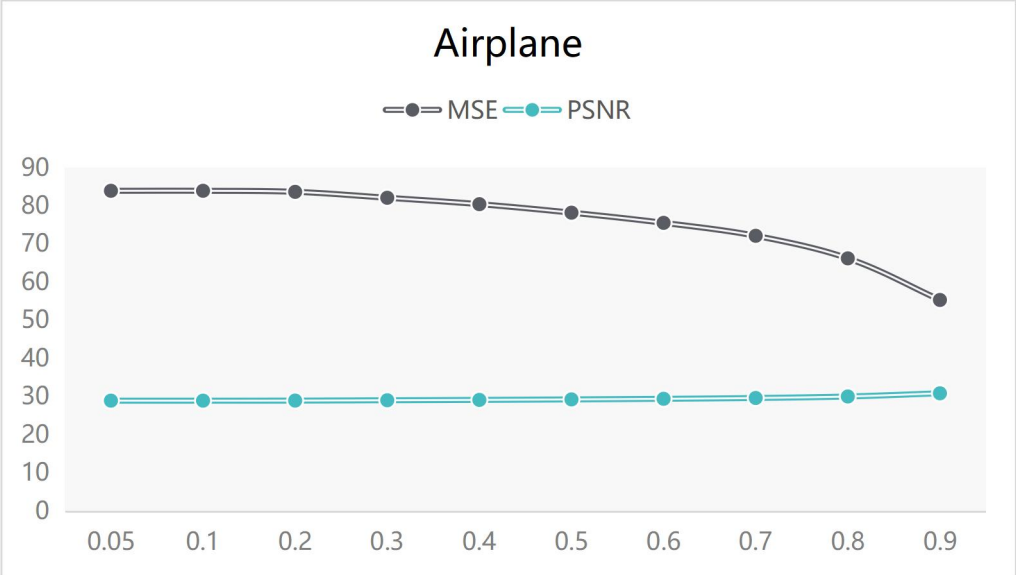
3. 评估指标均值分析：

   (1) MSE & PSNR

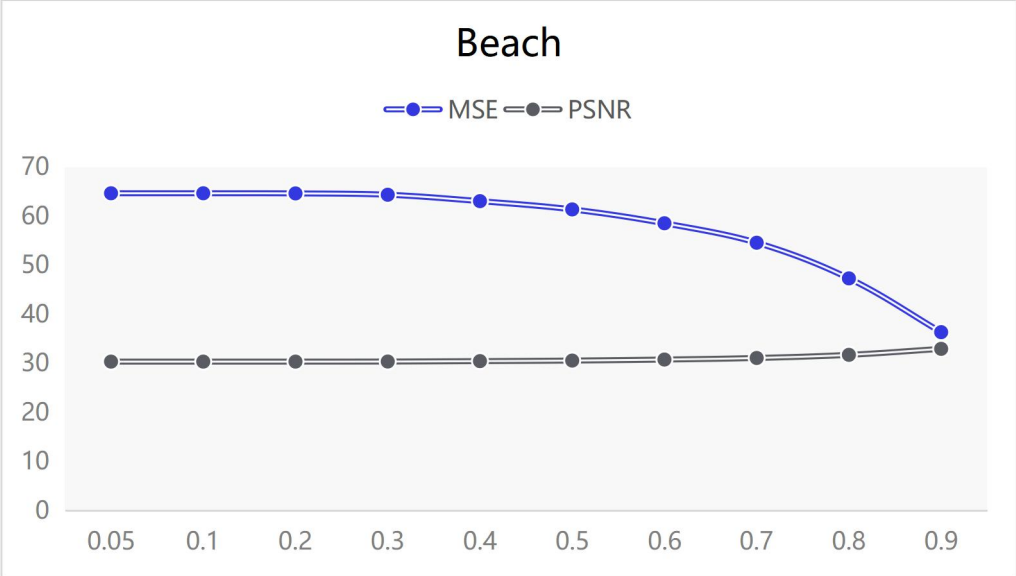
- **农业图像**



- **飞机图像**



- **海滩图像**



三类图像的均方误差和峰值信噪比随阈值变化的趋势大致相同：MSE 随阈值不断增大而逐渐下降，下降速度越来越快；PSNR 随阈值不断增大稍有上升。不同类别图像两指标范围略有差异。

(2) Compress_ratio & Space_Sparing

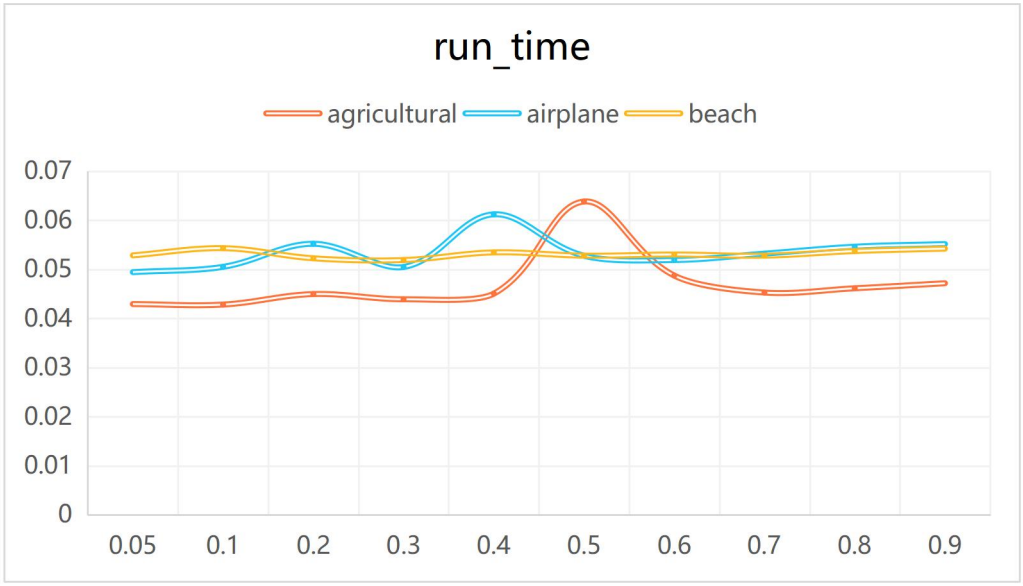
三类图像的压缩比和空间节省率不会随着阈值而变化，原因可能是文件格式的限制：图像文件的压缩比和存储空间不仅依赖于 PCA 算法，还与最终保存的文件格式有关。

某些文件格式可能对压缩效果有固定的存储开销，这使得不同阈值下的实际文件大小变化不大。

我此次实验保留了图像原本的格式属性，使用.tif 格式，所以会导致压缩后的文件大小变化不明显。

(3) Run_time

	A	B	C	D
1	explained_variance_ratios	agricultural	airplane	beach
2	0.05	0.042930441	0.049431863	0.052849915
3	0.1	0.0428283	0.050511701	0.054310358
4	0.2	0.044966552	0.055205321	0.052251856
5	0.3	0.043905313	0.050482671	0.051896284
6	0.4	0.045053616	0.061203811	0.053455145
7	0.5	0.063810821	0.052767949	0.052791576
8	0.6	0.048730495	0.051899629	0.053022768
9	0.7	0.045274167	0.053196492	0.052790589
10	0.8	0.046144617	0.054614139	0.053721914
11	0.9	0.047170208	0.05514792	0.054222431
12				

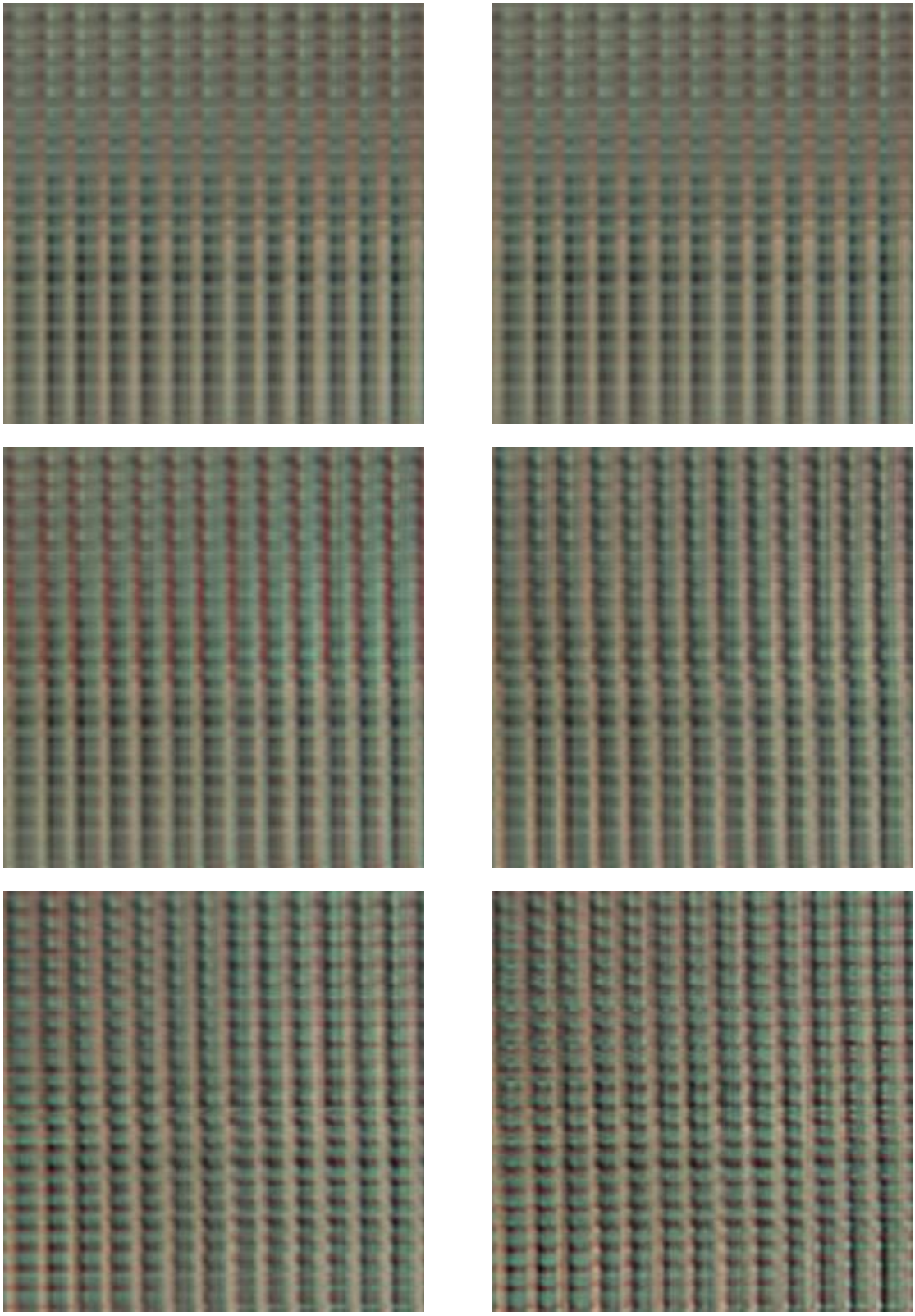


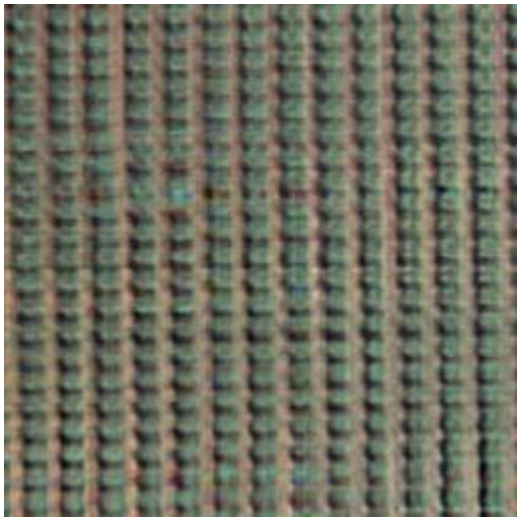
三类图像的运行时间都分别稳定在小区间内，该算法在效率上比较稳定，但不同类型图片处理上，效率会有一定差距。只有农田类型的图片有较大波动，原因可能是不同图像的内容复杂性可能差异较大：

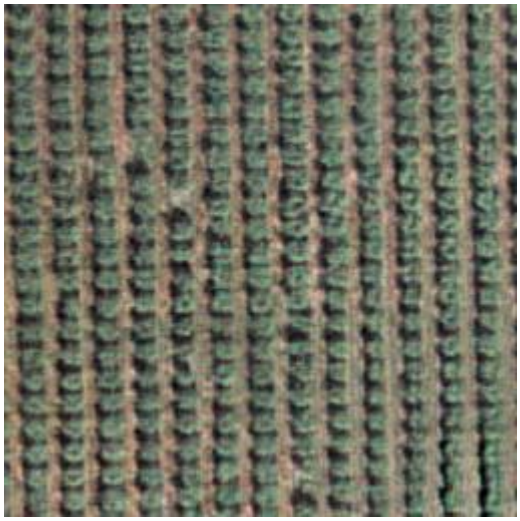
复杂的图像(例如包含很多细节和纹理的图像)在处理过程中可能需要更多的计算资源，导致运行时间较长。简单的图像（例如纯色或低细节的图像）则可能处理得更快。

(4) 实际视觉效果（从左到右，从上到下，阈值依次增大，最后为原图像）：

- **Agricultural21.tif**







从一系列图像中可以看出：在阈值 0.8 以上对图片的还原度已经很高了，采用课本做说

的 0.9 时，基本能得到和原图片一样的清晰度，肉眼基本看不出差别。

五、 结论

- **方法存在的不足**
  - 计算复杂度: 本次实验实现的 PCA 算法还不够完美，涉及大量的矩阵计算，处理大批量高分辨率图像时计算复杂度较高，运行时间会较长。
  - 重构误差: 尽管 PCA 保留了主要特征，但对于细节丰富的图像，重构后可能会出现细节丢失和失真现象，对压缩图像质量要求较高的情况下，压缩效率不高。
  - 存储需求: 除了压缩后的图像，PCA 还需要存储特征向量，这增加了额外的存储需求。
  - 压缩后格式转化：某些文件格式可能对压缩效果有固定的存储开销，这次实验，我没有对压缩后保存图像的格式进行调整，所以压缩效果不是非常理想。
- **未来研究方向**
  - 探究格式：探究并优化保存压缩后照片的格式，能做到占用空间小，压缩效果明显，并且图像还原效果高的双重目标。
  - 改进算法: 结合其他降维技术（如非负矩阵分解、独立成分分析）与 PCA，进一步提高压缩效果和图像质量，我在实验中原本进行了非负矩阵分解这一步优化，但它反而会让图像乱码，可能是我没有使用机器学习等一些自带的库，自己写的函数逻辑有些不对，因此，课后还会进一步研究这方面的问题。
  - 并行计算: 利用 GPU 等并行计算技术加速 PCA 计算，减少运行时间。因为我的电脑没有配置独显，所以压缩效率并不是非常优秀，往后可以深入研究一下，显卡在图像压缩领域的重要性。

- 自适应压缩：假设代码的一些输入参数能够根据图像内容自适应调整，那么压缩效果一定能够得到进一步提升。
- 多尺度分析：结合多尺度分析方法，对不同尺度的图像特征进行压缩和重构，进一步减少重构误差。

通过本次实验，我进一步实践掌握了 PCA 在图像压缩中的应用，让我把从课本上学习到的理论知识应用于实践当中，收获很大。希望未来的研究能够进一步改进和优化 PCA 算法，应用于更广泛的图像处理领域！