Video Compression HW 2－Color Transform 多工所 313553024 蘇柏叡

1. Introduction

在本次Lab中，我們以範例RGB影像(lena.png)作為輸入，先轉灰階後實作離散餘弦轉換與逆轉換。我們分別實作2D-DCT與兩次1D-DCT(row-column)之等價算法，完成係數計算、影像重建與PSNR品質評估，並比較兩者的計算時間。結果顯示，兩方法在數學上等價且重建品質一致；同時在對數域視覺化DCT係數可清楚觀察到能量集中於低頻的特性。另依作業規範，本實作未使用任何現成DCT/IDCT函式，僅以自行產生之正交基底與矩陣運算完成實作。

2. Implementation Details

2.1 I/O Functions

在**I/O Functions**部分，我們分別撰寫了讀取檔案並轉灰階的load_image function、於對數域視覺化DCT係數並顯示的visualize_and_save_dct_coefficients function、存取結果圖的save_iamge function。

```
def load_image(image_path):
    img = Image.open(image_path).convert('L')
    return np.asarray(img, dtype=np.float64)

def visualize_and_save_dct_coefficients(coefficients, output_path):
    V = np.log1p(np.abs(coefficients))
    plt.figure(figsize=(12, 10))
    plt.imshow(V, cmap='gray')
    plt.colorbar(label='Log magnitude')
    plt.title("DCT Coefficients (Log Domain)")
    plt.tight_layout()
    plt.savefig(output_path); plt.close()

def save_image(image, path):
    Image.fromarray(np.uint8(np.clip(image, 0, 255))).save(path)
```

Fig 1. Functions of I/O.

2.2 Row-Column One-Dimensional DCT/IDCT

我們參考講義內容公式撰寫程式碼，考量**Two-Dimensional DCT/IDCT**可由列、行兩種**One-Dimensional DCT/IDCT**組成，因此我們針對DCT-based設計了_dct_basis function 以供後續撰寫程式2D-DCT/IDCT使用。至於在作列、行兩種**One-Dimensional DCT/IDCT**中，我們先以1D-DCT/IDCT功能進行撰寫，並且以 $\mathbf{X} = \mathbf{C}@\mathbf{x}$，$\mathbf{x} = \mathbf{C}^{\mathrm{T}}@\mathbf{X}$ 呈現如**Fig 3**所示，接著我們也會針對以列、行兩種**One-Dimensional DCT/IDCT**實作講義內容公式如**Fig 4**所示。

```python
# ------------ Orthonormal DCT basis ------------
@lru_cache(maxsize=None)
def _dct_basis(n: int) -> np.ndarray:
    n_idx = np.arange(n)[None, :]
    k_idx = np.arange(n)[:, None]
    C = np.sqrt(2.0 / n) * np.cos((2 * n_idx + 1) * k_idx * np.pi / (2 * n))
    C[0, :] *= 1 / np.sqrt(2.0)
    return C  # orthonormal: C @ C.T = I
```

Fig 2. Implementations of DCT-based Matrix.

```python
# ---------------- 1D DCT / IDCT ----------------
def dct_1d(signal):
    x = np.asarray(signal, dtype=np.float64)
    C = _dct_basis(x.shape[0])
    return C @ x

def idct_1d(coefficients):
    a = np.asarray(coefficients, dtype=np.float64)
    C = _dct_basis(a.shape[0])
    return C.T @ a
```

Fig 3. Implementations of One-Dimensional DCT/IDCT.

```python
# -------- 2D via two 1D (row pass + col pass) --------
def dct_2d_using_1d(image):
    X = np.asarray(image, dtype=np.float64)
    M, N = X.shape
    Cx, Cy = _dct_basis(M), _dct_basis(N)
    # row-wise 1D: Cx @ X，再 col-wise 1D: (·) @ Cy.T
    return (Cx @ X) @ Cy.T

def idct_2d_using_1d(coefficients):
    D = np.asarray(coefficients, dtype=np.float64)
    M, N = D.shape
    Cx, Cy = _dct_basis(M), _dct_basis(N)
    # row-wise inverse: Cx.T @ D，再 col-wise inverse: (·) @ Cy
    return (Cx.T @ D) @ Cy
```

Fig 4. Implementations of Row-Column One-Dimensional DCT/IDCT.

2.3 Two-Dimensional DCT/IDCT

本函式我們參考講義內容公式撰寫程式碼，並且使用**Fig 2.**先前寫好的DCT-based Matrix，並且以$D = C_m@X@C_n^T$, $X = C_m^T@D@C_n$呈現如Fig 5所示。

```
# ---------------- 2D DCT / IDCT ----------------
def dct_2d(image):
    X = np.asarray(image, dtype=np.float64)
    M, N = X.shape
    Cx, Cy = _dct_basis(M), _dct_basis(N)
    return Cx @ X @ Cy.T

def idct_2d(coefficients):
    D = np.asarray(coefficients, dtype=np.float64)
    M, N = D.shape
    Cx, Cy = _dct_basis(M), _dct_basis(N)
    return Cx.T @ D @ Cy
```

Fig 5. Implementations of Two-Dimensional DCT/IDCT.

2.4 Metrics Evaluation and main Function

我們依據PSNR的公式定義撰寫了psnr function，並且依據兩種方法比較各自的處理時間，並且存檔案，接著我們打印結果並且於主程式中呼叫各類函式以完成本次實作以及將結果可視化呈現。

```
def psnr(original, reconstructed):
    x = np.asarray(original, dtype=np.float64)
    y = np.asarray(reconstructed, dtype=np.float64)
    mse = float(np.mean((x - y) ** 2))
    if mse == 0:
        return float('inf')
    return 20 * np.log10(255.0 / np.sqrt(mse))

def process_image(image, dct_function, idct_function, method_name):
    start_time = time.time()
    coeff = dct_function(image)
    recon = idct_function(coeff)
    processing_time = time.time() - start_time
    psnr_value = psnr(image, recon)
    os.makedirs('output', exist_ok=True)
    visualize_and_save_dct_coefficients(coeff, os.path.join('output', f"dct_coefficients_{method_name}.png"))
    save_image(recon, os.path.join('output', f"reconstructed_{method_name}.png"))
    return psnr_value, processing_time
```

Fig 6. Metrics Evaluation and Processing the images.

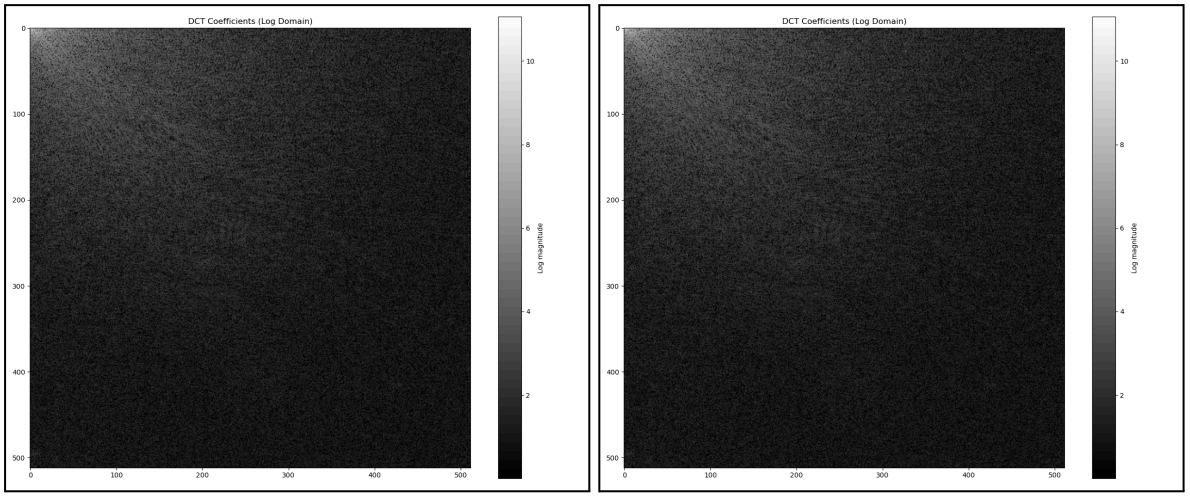
```python
def print_results(method_name, psnr_value, processing_time):
    print(f"Results for {method_name}:")
    print(f"  PSNR: {psnr_value:.2f} dB")
    print(f"  Processing time: {processing_time:.4f} seconds")
    print()

def main():
    if not os.path.exists('output'):
        os.makedirs('output')

    image = load_image('lena.png')

    psnr_2d, time_2d = process_image(image, dct_2d, idct_2d, "2d")
    print_results("2D-DCT", psnr_2d, time_2d)

    psnr_1d, time_1d = process_image(image, dct_2d_using_1d, idct_2d_using_1d, "1d")
    print_results("Two 1D-DCT", psnr_1d, time_1d)

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

Fig 6. Functions of printing the results and Main Functions

Note: You can key in python main.py to run this Lab's codes.

3. Experiment Results

3.1 Visualization of the DCT Coefficients



1D-DCT **2D-DCT**

3.2 Visualization of the Reconstructed Images

1D-DCT **2D-DCT**

3.3 Performance of Metrics and Comparison of Runtime

```
PS C:\Users\user\Desktop\Video Compression\Lab2> python main.py
Results for 2D-DCT:
  PSNR: 274.53 dB
  Processing time: 0.0203 seconds

Results for Two 1D-DCT:
  PSNR: 274.53 dB
  Processing time: 0.0070 seconds
```

Fig 7. Performance of Metrics and Comparison of Runtime

4. Conclusion

在本次Lab中，我們分別實作2D-DCT與兩次1D-DCT(row-column)方法，實驗結果表明兩種方法在PSNR均為274.53dB，不過在執行時間方面有著懸殊的差異(約3倍)，以時間複雜度而言2D-DCT為$\mathbf{O(N^4)}$，至於1D-DCT(row-column)則是為$\mathbf{O(2\times N^3)}$，因此在執行時間上1D-DCT(row-column)會比較佔優勢是可以預期的。

Supplementary

本次作業檔案路徑如**(Fig 8、9)**所示，output/為存放輸出結果圖的資料夾，main.py則是本次作業的程式檔案，lena.png則是本次作業提供之範例圖檔；至於requirements.txt則是記錄本次作業使用之環境所需之套件版本。

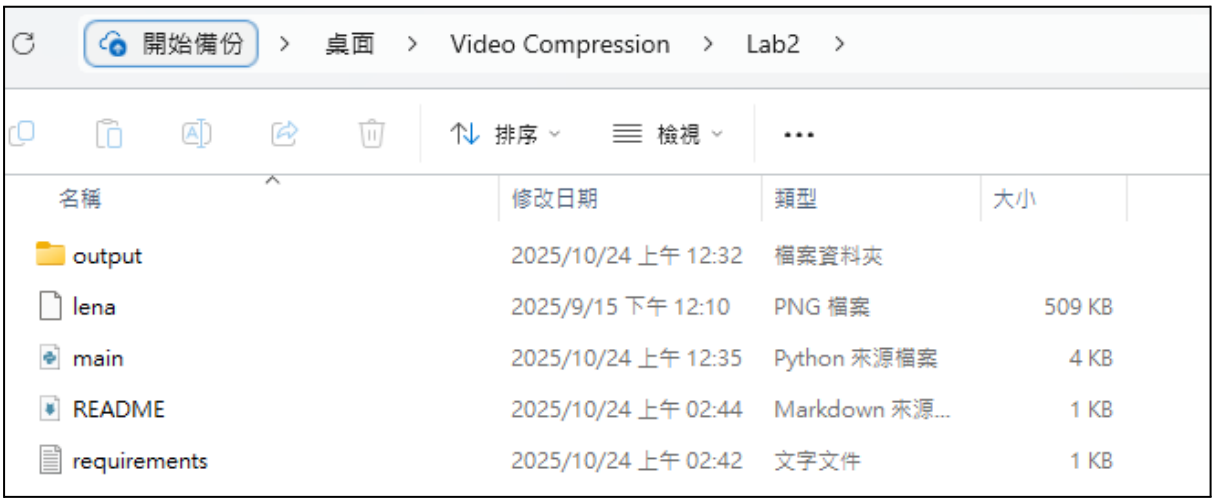


Fig 8. File paths of this Homework

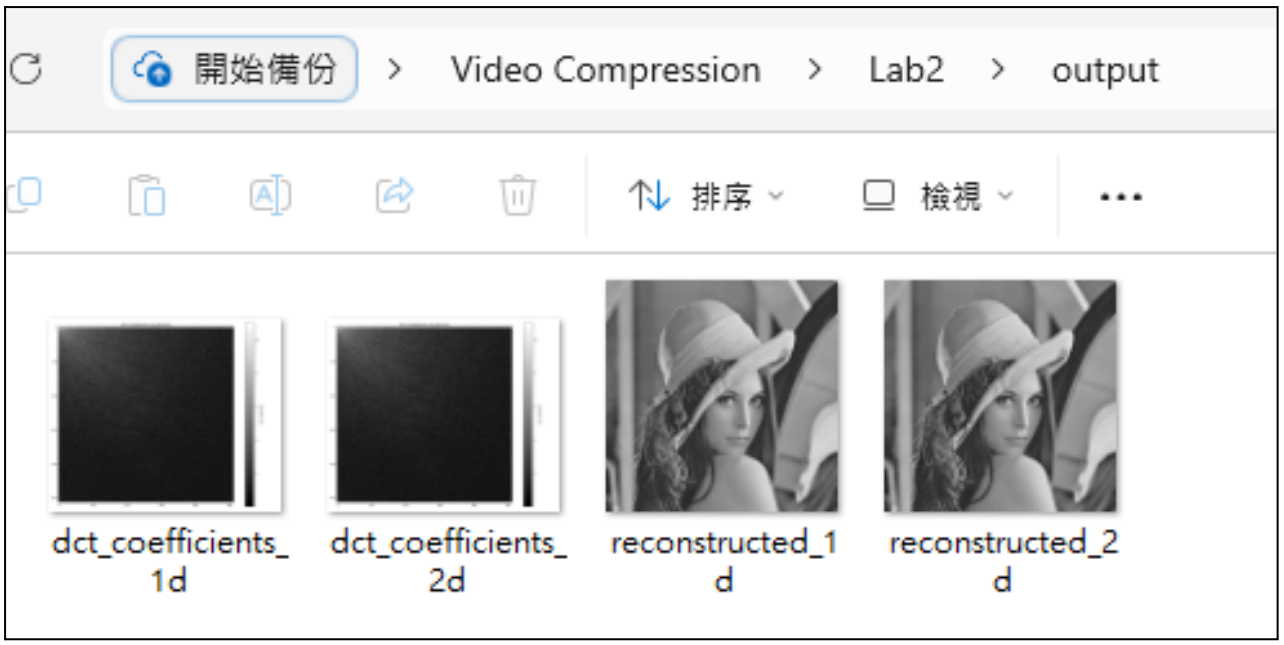


Fig 9. Output files in result folder

Github Link: https://github.com/rmd926/NYCU_VC/tree/main/Lab2