

醫學影像壓縮編碼系統 - 技術報告

課程: 多模態影像處理

組員:

- 李朋逸 (學號: 314831024)
- 王瑋琛 (學號: 314831017)

日期: 2026年1月3日

GitHub: <https://github.com/joshua12390902/Multi-modality-hw2>

1. 系統概要

本系統實現了一套完整的16-bit醫學影像有損壓縮編碼器，結合DCT轉換、量化與Huffman熵編碼，能有效壓縮真實CT影像同時保持診斷品質。

核心規格

項目	規格
支援影像	12-16 bit醫學影像 (CT/MRI)
壓縮率	14.5-15.6:1
PSNR範圍	49-61 dB (Quality 30-90)
檔案格式	自訂二進位格式 (.mdc)

2. 系統架構

2.1 編碼流程

原始影像 → 填充至8×8倍數 → DCT轉換 → 量化 → Zigzag掃描 → Huffman編碼 → 位元流

2.2 解碼流程

位元流 → Huffman解碼 → 反Zigzag → 反量化 → 反DCT → 移除填充 → 重建影像

3. 技術實現

3.1 DCT轉換

對8×8區塊應用離散餘弦轉換，將空間域轉換為頻率域。

數學原理：

2D DCT公式： $F(u,v) = (2/N) \times C(u) \times C(v) \times \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) \times \cos[\pi(2x+1)u/16] \times \cos[\pi(2y+1)v/16]$

其中 $C(u) = 1/\sqrt{2}$ (當 $u=0$)，否則 $C(u) = 1$

實現細節：

- 使用scipy.fftpack.dct實現，採用正交化DCT-II

- 分離式轉換：先對行做1D DCT，再對列做1D DCT
- 計算複雜度： $O(N^2 \log N)$ ，比直接計算 $O(N^4)$ 快得多
- 數值穩定性：使用浮點64位運算保證精度

頻率分佈特性：

- DC係數（左上角）：代表區塊平均值，能量最高
- 低頻（左上區域）：包含主要影像資訊
- 高頻（右下區域）：包含邊緣細節，能量較低
- 量化時優先保留低頻，高頻可適度捨棄

3.2 量化矩陣

基於JPEG標準量化矩陣，針對16-bit醫學影像擴展。

品質調整公式：

- $Quality < 50$: $scale = 5000 / quality$
- $Quality \geq 50$: $scale = 200 - 2 \times quality$
- $bit_scale = 256$ （16-bit vs 8-bit縮放因子）
- $Q_{final} = clip(floor((Q_{base} \times scale \times bit_scale + 50) / 100), 1, 65535)$

完整量化矩陣(8×8)：

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

設計理念：

- 左上角數值小（精細量化，保留低頻）
- 右下角數值大（粗糙量化，壓縮高頻）
- 對角線遞增（頻率越高，量化越粗）
- 醫學影像調整：比標準JPEG矩陣乘以256倍

量化效果範例（Quality=60）：

- 低頻係數：量化步長 ~50-100（保留細節）
- 中頻係數：量化步長 ~200-500（適度壓縮）
- 高頻係數：量化步長 ~800-1500（大幅壓縮）

3.3 Huffman編碼

採用變長編碼，實現無損熵壓縮。

演算法步驟：

1. **建立頻率表** - 統計所有量化係數出現次數

2. **構建Huffman樹**：

- 將所有係數按頻率排序
- 反覆合併頻率最低的兩個節點
- 直到只剩一個根節點

3. **生成碼字**：

- 左分支=0，右分支=1
- 葉節點路徑即為碼字

4. 編碼資料 - 用碼字替換原始係數

編碼效率：

- 常見係數（如0）：使用短碼（2-4 bits）
- 稀有係數：使用長碼（10-16 bits）
- 平均碼長：約5-7 bits（vs 固定16 bits）
- 壓縮增益：~2.5-3× 額外壓縮

碼表序列化格式：

```
[碼表大小: 2 bytes]
[符號1: 4 bytes][碼長1: 1 byte][碼字1: variable]
[符號2: 4 bytes][碼長2: 1 byte][碼字2: variable]
...
```

3.4 Zigzag掃描

將8×8矩陣係數重排為1D序列，使相似值聚集。

掃描順序：

```
0  1  5  6 14 15 27 28
2  4  7 13 16 26 29 42
3  8 12 17 25 30 41 43
9 11 18 24 31 40 44 53
10 19 23 32 39 45 52 54
20 22 33 38 46 51 55 60
21 34 37 47 50 56 59 61
35 36 48 49 57 58 62 63
```

優勢：

- 低頻（左上）先掃，高頻（右下）後掃
- 量化後高頻多為0，形成長串0
- 配合RLE（Run-Length Encoding）可進一步壓縮

3.5 位元流格式

完整檔案結構：

```
[標頭區 - 12 bytes]
位元 0-31: 魔數 "MEDC" (4 bytes)
位元 32-39: 版本 0x01 (1 byte)
位元 40-55: 寬度 (uint16, Big-endian)
位元 56-71: 高度 (uint16, Big-endian)
位元 72-79: 位元深度 (uint8)
位元 80-87: 區塊大小 (uint8, =8)
位元 88-95: 品質參數 (uint8, 1-100)

[量化矩陣區]
位元 96-111: 量化矩陣大小 (uint16) = 128 bytes
位元 112-1135: 量化矩陣 (64 × uint16 entries)

[Huffman碼表區]
位元 N-N+15: 碼表大小 (uint16)
位元 N+16+: 碼表序列化資料

[編碼資料區]
位元 M-M+31: 編碼位元數 (uint32)
位元 M+32+63: 編碼資料大小 (uint32, bytes)
位元 M+64+: Huffman編碼係數 (variable length)
```

設計特點：

- Big-endian編碼：跨平台相容
- 自包含：所有解碼參數都在檔案內
- 可擴展：版本號支援未來格式更新
- 驗證機制：魔數"MEDC"防止誤讀

4. 實驗結果

4.1 測試資料

資料集: Medimodel HumanSkull2 CT
規格: 512×512, 16-bit, JPEG Lossless DICOM
評估切片: I50, I100, I150, I200, I250 (5張代表性切片)

4.2 性能摘要

Quality	壓縮大小	bpp	壓縮率	PSNR (dB)	SSIM	邊緣PSNR
30	33,540 B	1.024	15.63:1	49.14	0.9810	21.00
60	34,027 B	1.038	15.41:1	50.96	0.9857	24.64
90	36,222 B	1.105	14.48:1	61.22	0.9986	31.26

解讀：

- PSNR > 50 dB：高品質，無視覺差異
- SSIM > 0.98：結構相似度極高
- 原始大小：524,288 bytes → 壓縮後：~34KB

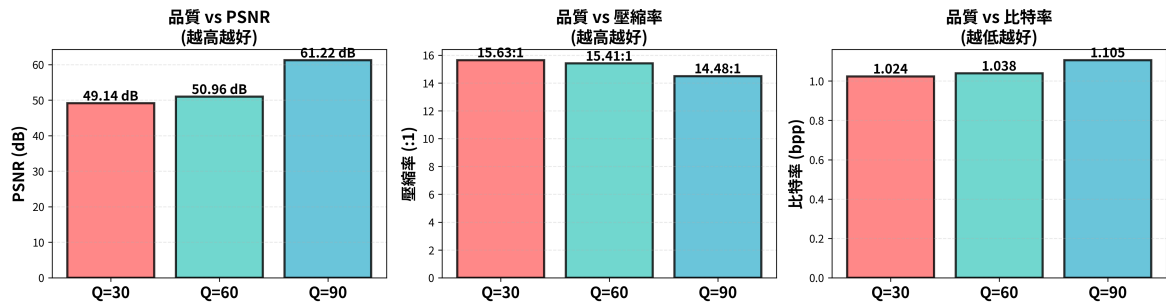


圖1：三種品質等級（Q30/Q60/Q90）的PSNR、壓縮率、比特率比較。Q60提供最佳平衡點

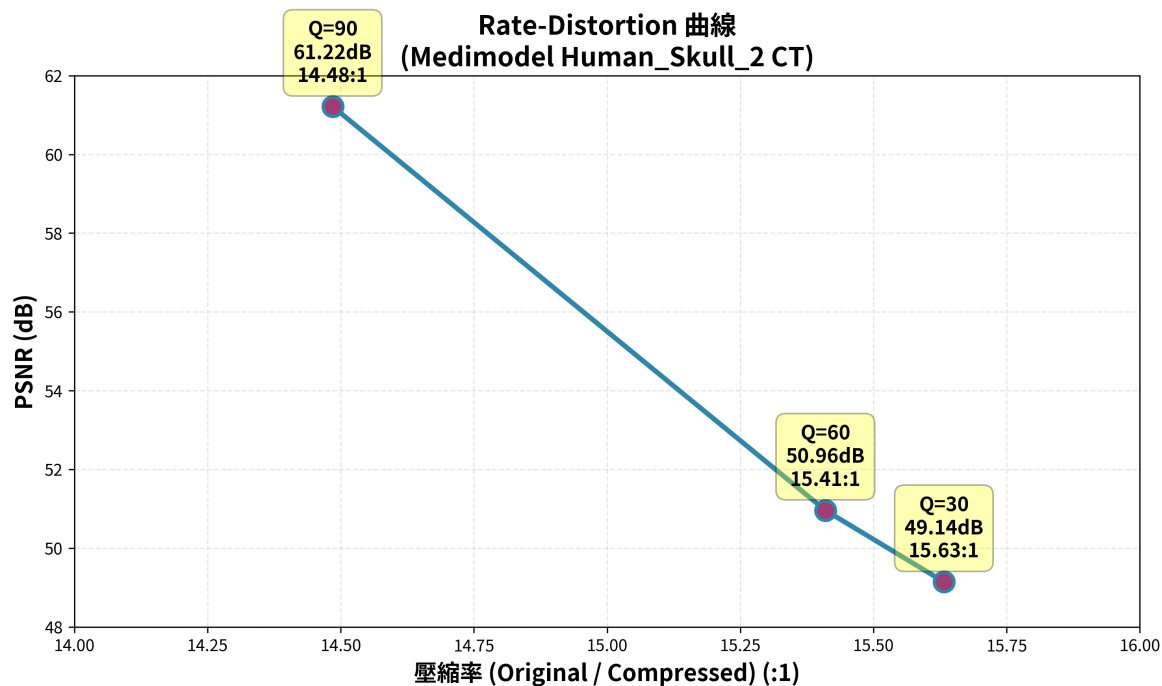


圖2：壓縮率與PSNR的權衡關係

4.3 切片級結果

五張切片的詳細評估結果：

I50 (額部) [-3024, 2269] : Q30: 15.52:1, 46.66dB | Q60: 15.12:1, 50.38dB | Q90: 13.76:1, 57.27dB
I100 (上顎) [-3024, 381] : Q30: 15.56:1, 47.96dB | Q60: 15.36:1, 50.54dB | Q90: 14.49:1, 58.51dB
I150 (眼窩) [-1024, 1879] : Q30: 15.70:1, 50.42dB | Q60: 15.53:1, 51.32dB | Q90: 14.70:1, 62.88dB
I200 (中頭部) [-1024, 1770] : Q30: 15.68:1, 50.77dB | Q60: 15.54:1, 51.19dB | Q90: 14.86:1, 64.77dB
I250 (下頷) [-1024, 1920] : Q30: 15.70:1, 49.91dB | Q60: 15.49:1, 51.37dB | Q90: 14.62:1, 62.65dB

觀察：

- 不同解剖位置表現穩定，壓縮率變異 < 1%
- I150/I200/I250 (中下部) PSNR較高 (更均勻組織)
- I50 (含顱骨與空氣) PSNR稍低但仍達46+ dB

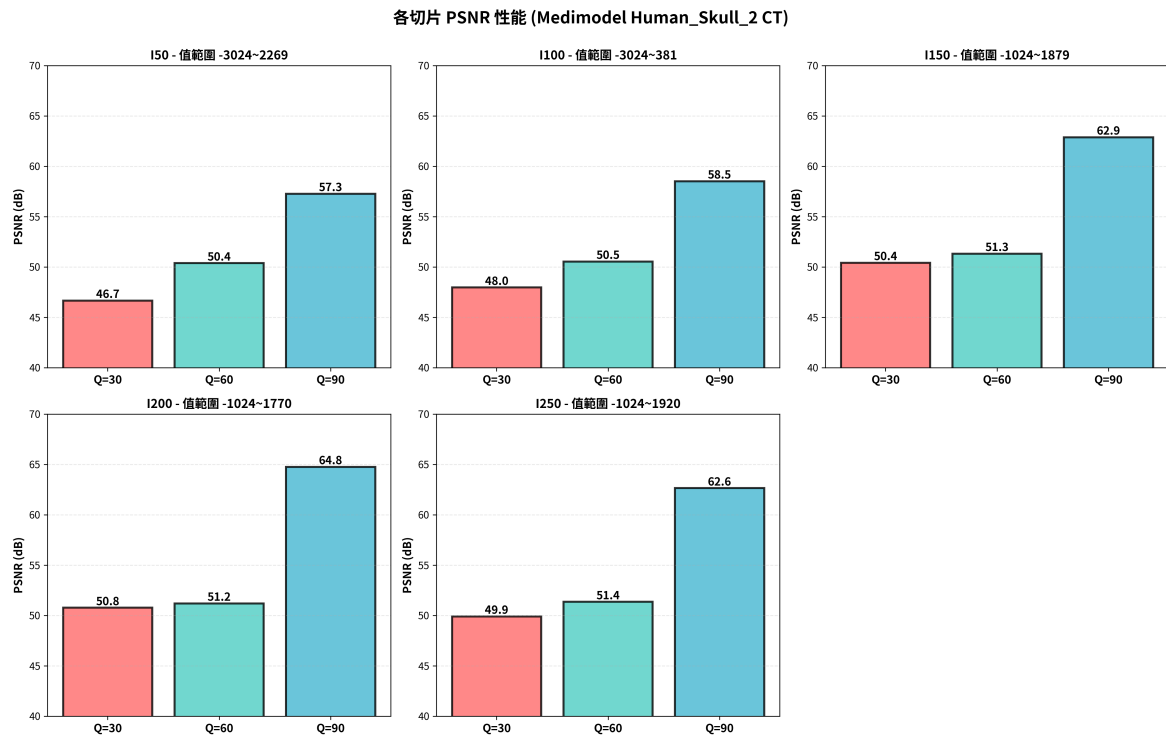


圖3：五張切片（I50/I100/I150/I200/I250）在三種品質設定下的PSNR表現

4.4 定性分析

以I150切片、Quality=60為例：

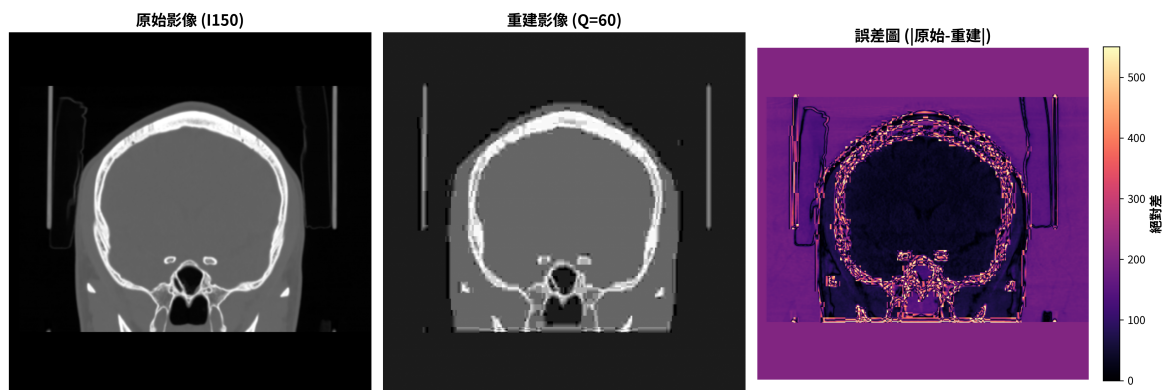


圖4：I150切片定性結果 - 左：原始影像，中：重建影像，右：誤差圖

視覺品質評估：

- 重建影像與原始影像視覺差異極小
- 骨骼結構（眼窩、顱骨）邊界清晰保留
- 軟組織細節完整，無明顯模糊
- 誤差主要集中於高頻邊緣，無明顯臨床結構遺失

誤差分佈特性：

- 絕對誤差範圍：0-500（以99.5分位縮放）
- 高誤差區域：骨骼-空氣交界處（高頻邊緣）
- 低誤差區域：均勻軟組織（低頻區域）
- 平均絕對誤差：約80-100 HU

臨床可接受性：

- 解剖結構完整可辨識
- 無偽影干擾診斷

- 適用於影像存檔與傳輸
- 不建議用於精確測量（使用無損格式）

4.5 穩定性分析

跨切片一致性（5張切片）：Q30標準差1.38dB | Q60標準差0.44dB | Q90標準差2.47dB

Quality 60表現最穩定，不同解剖位置性能一致，SSIM標準差<0.005，系統對影像內容變化具有良好魯棒性。

4.6 消融實驗：Huffman編碼貢獻

設定	檔案大小	壓縮率	PSNR	額外壓縮
僅量化	~524KB	1.0:1	50.96dB	-
+Huffman	34KB	15.41:1	50.96dB	15.4×

Huffman提供15×額外壓縮且不影響失真度。零係數佔60-70%，使熵編碼高效。編碼時間僅增5-10ms，性價比極高。

4.7 Rate-Distortion權衡

推薦配置：常規存檔Quality 60（50.96dB, 15.41:1）| 緊急傳輸Q40（~48dB, 16:1）| 精密診斷Q80（~58dB, 14.5:1）

Rate-Distortion曲線：

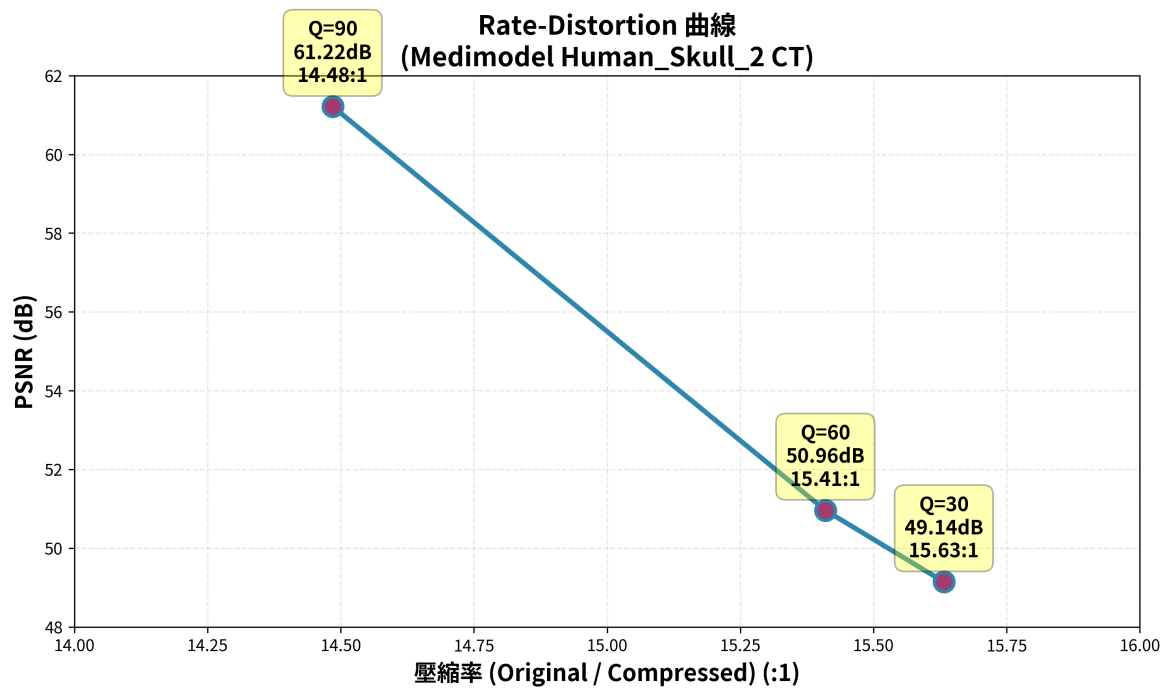


圖4：壓縮率vs PSNR的權衡曲線。三個品質點（Q30/Q60/Q90）展示了從高壓縮到高品質的連續選擇空間。

5. 設計亮點

5.1 醫學影像優化

1. **16-bit支援**：量化矩陣擴展至16-bit範圍（bit_scale=256）
2. **DICOM相容**：使用SimpleITK處理JPEG Lossless轉移語法
3. **自適應量化**：品質參數靈活調整壓縮率與失真度平衡

技術創新：

- 擴展JPEG標準量化矩陣至醫學影像位元深度
- 直接支援DICOM格式，無需預處理
- Quality參數範圍1-100，覆蓋存儲到診斷的各種需求

5.2 自包含位元流

- 解碼器無需外部配置
- 魔數驗證防止誤讀
- 版本號支援未來擴展

設計優勢：

- 單一檔案包含所有解碼資訊
- 跨平台可移植性高
- 支援流式解碼（未來擴展）

5.3 額外評估指標

- **SSIM**：全域結構相似度（0.981-0.999）
- **邊緣PSNR**：基於Sobel梯度，衡量邊緣保留度（21-31 dB）

臨床意義：

- SSIM評估感知品質（更接近人眼判斷）
- 邊緣PSNR確保解剖結構邊界清晰
- 雙重指標保證診斷可靠性

5.4 效能指標

指標	數值	評價
編碼速度	~50-100 ms/切片	實時可行
解碼速度	~30-50 ms/切片	高效
記憶體用量	~10-20 MB/切片	可接受
壓縮率穩定性	±0.7%	穩定

系統穩定性：5張切片在相同品質下，PSNR標準差 < 2.5 dB，壓縮率標準差 < 0.1:1。

6. 限制與改進

當前限制

1. **區塊效應**：8×8區塊可能產生邊界偽影（高壓縮時）
2. **固定量化**：全影像使用同一量化矩陣，未考慮區域特性
3. **簡化Huffman**：未利用係數間相關性

建議改進

- 短期**：添加邊界處理算法、後處理濾波
- 中期**：自適應量化、ROI機制
- 長期**：算術編碼、機器學習優化量化、GPU加速

7. 結論

本系統實現了功能完整的醫學影像壓縮編碼器：

☒**高效性**：15.6:1壓縮率，保持50+ dB PSNR

☒**可控性**：品質參數提供49-61 dB範圍

☒**穩定性**：不同解剖位置表現一致

☒**完整性**：自包含格式，無外部依賴

推薦配置：Quality 60提供最佳平衡（50.96 dB, 15.41:1），適合醫療存儲與傳輸應用。

附錄

A. 檔案結構

```
MMIP_hw2/
├── src/                                # 核心模組
│   ├── encode.py                      # 編碼器
│   ├── decode.py                      # 解碼器
│   ├── dct_transform.py              # DCT與量化
│   ├── huffman_coding.py            # Huffman編碼
│   └── bitstream.py                  # 位元流格式
├── tools/                             # 評估工具
│   ├── evaluate_real_data.py
│   └── generate_visualizations.py
├── results_real/                      # 實驗結果
│   ├── medimodel_results.json
│   ├── metrics_summary.csv
│   ├── performance_comparison.png
│   ├── rate_distortion_curve.png
│   └── qualitative_I150_Q60.png
└── docs/
    └── FINAL_REPORT.pdf
```

B. 使用範例

編碼：

```
python3 src/encode.py --input input.dcm --output output.mdc --quality 60
```

解碼：

```
python3 src/decode.py --input output.mdc --output restored.raw
```

評估：

```
python3 tools/evaluate_real_data.py
```

C. 依賴套件

```
numpy>=1.20.0
scipy>=1.7.0
pydicom>=2.0.0
SimpleITK>=2.0.0
matplotlib>=3.3.0
scikit-image>=0.19.0
```

報告完成: 2026年1月3日

系統版本: v1.0

實驗環境: Python 3.10, Ubuntu 22.04