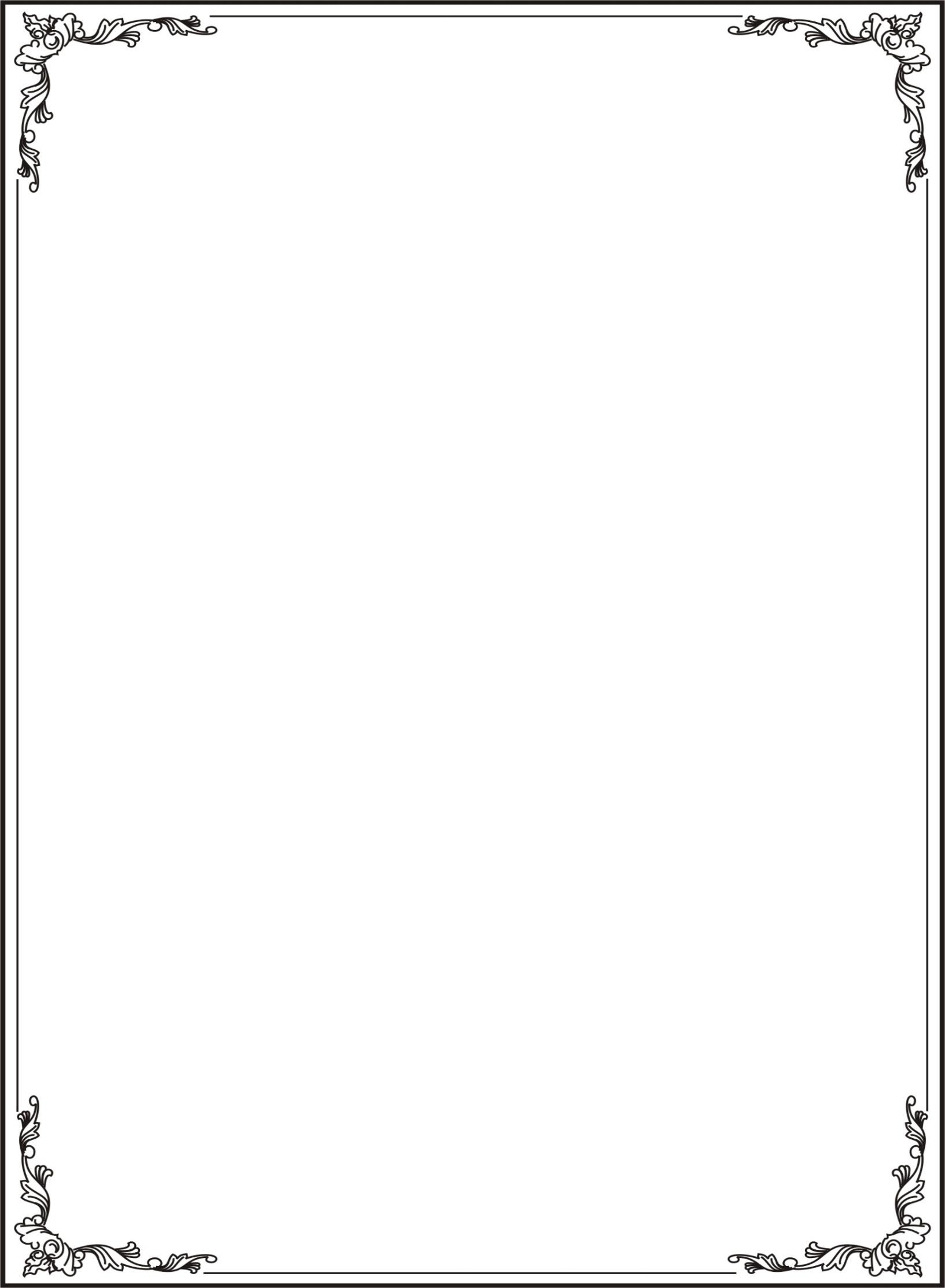
**ĐẠI HỌC QUỐC GIA TPHCM**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN MÔN HỌC**

**TÍNH TOÁN ĐA PHƯƠNG TIỆN**

**CS232.K21.KHTN**

**ĐỀ TÀI:**

**CÀI ĐẶT VÀ ĐÁNH GIÁ MỘT SỐ THUẬT TOÁN NÉN VÀ GIẢI NÉN DỮ LIỆU**

**SINH VIÊN THỰC HIỆN:**  **GIẢNG VIÊN HƯỚNG DẪN:**

**Trần Đình Khang - 18520896 TS. Ngô Đức Thành**

**Trần Thị Phương Thảo - 18521422**

**Nguyễn Văn Tiến - 18521489**

***TP.HCM, ngày 10 tháng 8 năm 2020***

*Trang này để trống*

# THÔNG TIN NHÓM

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Đóng góp |
| 1 | Trần Đình Khang | 18520896 | * Cài đặt các thuật toán nén và giải nén VLCs, bao gồm:   + Huffman Coding   + Adaptive Coding * Thu thập dữ liệu cho việc đánh giá hiệu quả của các thuật toán. * Viết báo cáo kết quả và đánh giá. |
| 2 | Trần Thị Phương Thảo | 18521422 | * Cài đặt các thuật toán nén:   + Run Length Coding (RLC).   + Shannon-Fano Coding   + Dictionary-based Coding (LZW). * Viết báo cáo kết quả và đánh giá. |
| 3 | Nguyễn Văn Tiến | 18521489 | * Cài đặt các thuật toán nén và giải nén, bao gồm:   + Arithmetic Coding.   + JPEG lossless.   + K-mean clustering. * Viết báo cáo kết quả và đánh giá. |

# MỤC LỤC

[THÔNG TIN NHÓM](#_heading=h.30j0zll) **3**

[MỤC LỤC](#_heading=h.1fob9te) **4**

[I. CÁC THUẬT TOÁN CÀI ĐẶT](#_heading=h.3znysh7) **5**

[1. Thuật toán nén và giải nén cho dữ liệu text](#_heading=h.2et92p0) 5

[Run Length Coding (RLC)](#_heading=h.tyjcwt) 5

[1.2 Variable Length Coding (VLC)](#_heading=h.3dy6vkm) 5

[1.2.1 Shannon Fano Coding](#_heading=h.37wovs6276ed) 5

[1.2.2 Huffman Coding](#_heading=h.vozbaqtwpnd3) 6

[1.2.3 Adaptive Huffman Coding](#_heading=h.depzvfii8th) 6

[1.3 Dictionary-based Coding (LZW)](#_heading=h.1t3h5sf) 6

[1.4 Arithmetic Coding](#_heading=h.4d34og8) 6

[2. Thuật toán nén và giải nén cho dữ liệu hình ảnh](#_heading=h.1kjb64z2rchy) 7

[2.1 JPEG lossless](#_heading=h.17dp8vu) 7

[2.2 K-mean clustering](#_heading=h.3rdcrjn) 7

[II. MÔ TẢ DỮ LIỆU](#_heading=h.26in1rg) **8**

[1. Dữ liệu text](#_heading=h.lnxbz9) 8

[2. Dữ liệu ảnh](#_heading=h.tc35f21cd1iv) 8

[III. KẾT QUẢ VÀ ĐÁNH GIÁ](#_heading=h.1ksv4uv) **9**

[1. Các thuật toán nén và giải nén dữ liệu text.](#_heading=h.44sinio) 9

[2. Các thuật toán nén và giải nén dữ liệu hình ảnh.](#_heading=h.8zobjdpa2d6g) 10

[3. Đánh giá](#_heading=h.z337ya) 10

[Các thuật toán nén và giải nén dữ liệu text.](#_heading=h.t0qxgjb08x0) 10

[Các thuật toán nén và giải nén dữ liệu hình ảnh.](#_heading=h.umdmebkb9qls) 11

[IV. TỔNG KẾT](#_heading=h.3j2qqm3) **11**

[TÀI LIỆU THAM KHẢO](#_heading=h.1y810tw) **12**

# I. CÁC THUẬT TOÁN CÀI ĐẶT

## 1. Thuật toán nén và giải nén cho dữ liệu text

### Run Length Coding (RLC)

Thuật toán Run length coding sử dụng cho dữ liệu có tính lặp lại liên tiếp nhau. Thay vì lưu trữ một chuỗi liên tiếp n ký tự giống hệt nhau, ta chỉ cần lưu trữ 1 ký tự duy nhất kèm theo đó là số lần xuất hiện liên tiếp của ký tự đó. Tuy nhiên, sự lặp lại liên tục của các ký hiệu trong mẫu thông tin cũng là vấn đề ở RLC.

Mô tả thuật toán:

* Input: mẫu thông tin cần mã hóa, ví dụ: một đoạn văn bản
* Output: một chuỗi mã hóa của đoạn văn bản có bản .
* Tỉ số nén được tính theo công thức (với *len* là hàm lấy chiều dài của chuỗi ký tự):

### 1.2 Variable Length Coding (VLC)

Thuật toán Variable length coding là nhóm thuật toán mã hóa ký tự trong một mẫu thông tin dựa trên tần suất xuất hiện của ký tự đó trong mẫu thông tin.

Nhóm thuật toán này hiệu quả hơn RLC vì giải quyết được vấn đề lặp lại không liên tục. Các thuật toán được cài đặt và đánh giá trong đồ án này gồm có:

* Shanon Fano
* Huffman Coding.
* Adaptive Huffman Coding.

Mô tả thuật toán:

* Input: mẫu thông tin cần mã hóa, ví dụ: một đoạn văn bản
* Output: chuỗi ký tự nhị phân biểu diễn cho đoạn văn bản và một từ điển để ánh xạ các chuỗi ký tự nhị phân thành ký tự để giúp cho quá trình giải mã.
* Tỉ số nén được tính theo công thức:

#### 1.2.1 Shannon Fano Coding

Shannon Fano Coding được đặt tên bởi Claude Shannon – một nhà toán học, tin học và Robert Fano – một nhà khoa học máy tính. Thuật toán này được xây dựng theo hướng tiếp cận top-down, từ một mẫu thông tin ban đầu đi chia nhỏ dần sau cho mỗi lần chia nhỏ ta đảm bảo sự chênh lệch về độ dài giữa hai mẫu thông tin là nhỏ nhất.

#### 1.2.2 Huffman Coding

Thuật toán Huffman sử dụng hướng tiếp cận bottom-up. Thuật toán xây dựng một câu nhị phân cân bằng với nút là là các ký tự mà mẫu thông tin chứa sao cho ký tự nào có số lần xuất hiện nhiều sẽ có độ sâu càng nhỏ, tức là càng ít bit để biểu diễn.

#### 1.2.3 Adaptive Huffman Coding

Thuật toán Adaptive Huffman được cải tiến từ Huffman Coding, dùng được cho dữ liệu dạng stream.

### 1.3 Dictionary-based Coding (LZW)

Thuật toán Dictionary-based Coding sử dụng một từ điển để mã hóa các ký tự cần mã hóa. Thuật toán dùng cho cả dữ liệu tĩnh và dữ liệu stream.

Mô tả thuật toán:

* Input: mẫu thông tin, ví dụ: văn bản .
* Output: chuỗi các ký hiệu mã hóa và một từ điển lưu giữ mối quan hệ giữa ký tự mã hóa và ký tự được giải mã.
* Tỉ số nén được tính theo công thức:

### 1.4 Arithmetic Coding

Thuật toán Arithmetic Coding sử dụng việc chia khoảng giá trị để mã hóa thông tin. Ký hiệu nào mang nhiều thông tin (tần số xuất hiện ít) sẽ được mã hóa bởi một đoạn giá trị có độ rộng càng nhỏ.

Mô tả thuật toán:

* Input: mẫu thông tin, ví dụ: văn bản .
* Output: giá trị và bảng phân phối xác suất của các ký hiệu.
* Tỉ số nén được tính theo công thức:

## 2. Thuật toán nén và giải nén cho dữ liệu hình ảnh

### 2.1 JPEG lossless

Thuật toán JPEG lossless mã hóa sự khác nhau giữa các pixel lân cận. Thay vì lưu trữ giá trị mỗi pixel ta chỉ cần lưu trữ sự khác nhau với số lượng bits ít hơn.

Mô tả thuật toán:

- Input: một ảnh RGB với kích thước .

- Output: một chuỗi bit nhị phân biểu diễn hình ảnh đầu vào , một từ điển (ngoài ra cần lưu giá trị W, H, predictor sử dụng () nhưng so với dung lượng của chuỗi bits và từ điển thì không đáng kể).

- Tỉ số nén được tính theo công thức:

### 2.2 K-mean clustering

Thuật toán K-mean clustering là một toán học không giám sát (unsupervised learning) trong máy học. Trong mã hóa hình ảnh, K-mean clustering là một thuật toán nén có mất mát. Tư tưởng của thuật toán chính là gom điểm dữ liệu hay cụ thể là các điểm ảnh thành các cụm và lưu trữ giá trị một centroid của một cụm để biểu diễn cho toàn bộ các điểm ảnh thuộc cụm đó.

Mô tả thuật toán:

* Input: một ảnh RGB với kích thước , số lượng cụm để gom nhóm .
* Output: nhãn của các điểm ảnh và giá trị của các centroids
* Tỉ số nén được tính theo công thức:

# II. MÔ TẢ DỮ LIỆU

## 1. Dữ liệu text

Dữ liệu text được thu tập từ các trang báo The New York Times, Daily Mail.

|  |  |  |
| --- | --- | --- |
| Nguồn | The New York Times | Daily Mail |
| Số lượng ký tự/mẫu (xấp xỉ) | 1336 | 5462 |
| Số lượng mẫu | 20 | 20 |

## 

## 2. Dữ liệu ảnh

Dữ liệu ảnh được thu thập có chọn lọc từ nguồn Internet gồm nhiều ảnh với nhiều chủ đề, kích thước và tỉ lệ khung hình khác nhau. Định dạng ảnh là ảnh màu RGB, loại tệp JPEG.

Tập dữ liệu được mô tả chi tiết như bảng bên dưới:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kích thước (WxH) | 1920x1080 | 1280x720 | 1080x1080 | 1200x900 | 1200x800 |
| Ratio | 16:9 | 16:9 | 1:1 | 4:3 | 3:2 |
| Số lượng | 20 | 20 | 20 | 20 | 20 |

# III. KẾT QUẢ VÀ ĐÁNH GIÁ

## 1. Các thuật toán nén và giải nén dữ liệu text.

|  |  |  |
| --- | --- | --- |
| **Dữ liệu** | **Thuật toán** | **Tỉ lệ nén trung bình** |
| Text 1→ 20 | RLC | 1.001 |
| Shanon Fano | 1.5708 |
| Huffman | 1.8005 |
| Adaptive Huffman |  |
| LZW | 1.606 |
| Arithmetic Coding | 10.3275 |
| Text 21 → 40 | RLC | 1.0004 |
| Shanon Fano | 1.5677 |
| Huffman | 1.8015 |
| Adaptive Huffman |  |
| LZW | 2.1465 |
| Arithmetic Coding | 29.4265 |

## 

## 2. Các thuật toán nén và giải nén dữ liệu hình ảnh.

|  |  |  |
| --- | --- | --- |
| **Dữ liệu** | **Thuật toán** | **Tỉ lệ nén trung bình** |
| 1080x1080 | JPEG lossless | 2.223 |
| K mean (k = 32) | 1.6 |
| K mean (k = 64) | 1.33 |
| 1200x800 | JPEG lossless | 2.1405 |
| K mean (k = 32) | 1.6 |
| K mean (k = 64) | 1.33 |
| 1200x900 | JPEG lossless | 1.9875 |
| K mean (k = 32) | 1.6 |
| K mean (k = 64) | 1.33 |
| 1280x720 | JPEG lossless | 2.361 |
| K mean (k = 32) | 1.6 |
| K mean (k = 64) | 1.33 |
| 1920x1080 | JPEG lossless | 1.9075 |
| K mean (k = 32) | 1.6 |
| K mean (k = 64) | 1.33 |

## 3. Đánh giá

Dựa vào 2 bảng kết quả trên:

## Các thuật toán nén và giải nén dữ liệu text.

Thuật toán cho tỉ lệ nén tốt nhất với tất cả các kiểu dữ liệu là Arithmetic Coding với tỉ lệ nén trung bình trên 2 tập dữ liệu text là 19.877 , thấp nhất là RLC với tỉ lệ nén trung bình trên 2 tập dữ liệu text là 1.0007

## Các thuật toán nén và giải nén dữ liệu hình ảnh.

Thuật toán cho tỉ lệ nén tốt nhất với tất cả các kiểu dữ liệu là JPEG lossless với tỉ lệ nén trung bình 2.1239 , thấp nhất là K mean (k = 64) với tỉ lệ nén trung bình 1.33.

K mean cho tỉ lệ nén trung bình như nhau 1.33 với tất cả các dữ liệu kiểm tra.

# IV. TỔNG KẾT

Trong đồ án này, nhóm đã cài đặt các thuật toán nén và giải nén dữ liệu cho hai loại dữ liệu là text và hình ảnh.

Các thuật toán cài đặt gồm có:

* Run length Coding.
* Variable Length Coding:
  + Shanon Fano.
  + Huffman Coding.
  + Adaptive Huffman Coding.
* Dictionary-based Coding.
* Arithmetic Coding.
* JPEG lossless.
* K-mean clustering.

# TÀI LIỆU THAM KHẢO

[1] Ze-Nian Li, Mark S. Drew. Fundamentals of Multimedia. Pearson/Prentice Hall,2004.

[2] [Run Length Encoding](https://www.geeksforgeeks.org/run-length-encoding/)

[3] [Shannon-Fano-Algorithm](https://github.com/Mohammed-Ashour/Shannon-Fano-Algorithm)

[4] [Huffman-Coding-Algorithm](https://github.com/bhrigu123/huffman-coding)

[5] [K-mean clustering](https://www.geeksforgeeks.org/image-compression-using-k-means-clustering/)