

# **BÁO CÁO ĐỒ ÁN CUỐI KỲ**

## **Image Compression use Singular Value Decomposition and Kmeans Clustering**

GVHD: Đỗ Văn Tiến

Sinh viên thực hiện: Hồ Hồng Hà - 20520480

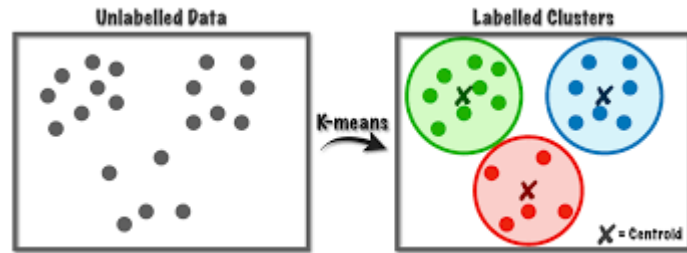
# Nội Dung

1. Giới thiệu đề tài
2. Nội dung thực hiện
3. Kết quả thực nghiệm và đánh giá

# Giới thiệu đề tài

- Nén hình ảnh là một kiểu nén dữ liệu được áp dụng cho hình ảnh kỹ thuật số mà làm giảm chất lượng của hình ảnh đến mức có thể chấp nhận được.
- Làm giảm chi phí lưu trữ và truyền tải chúng.
- Phương pháp sử dụng:
  - + K-means clustering
  - + Singular Value Decomposition(SVD)

# K-mean Clustering

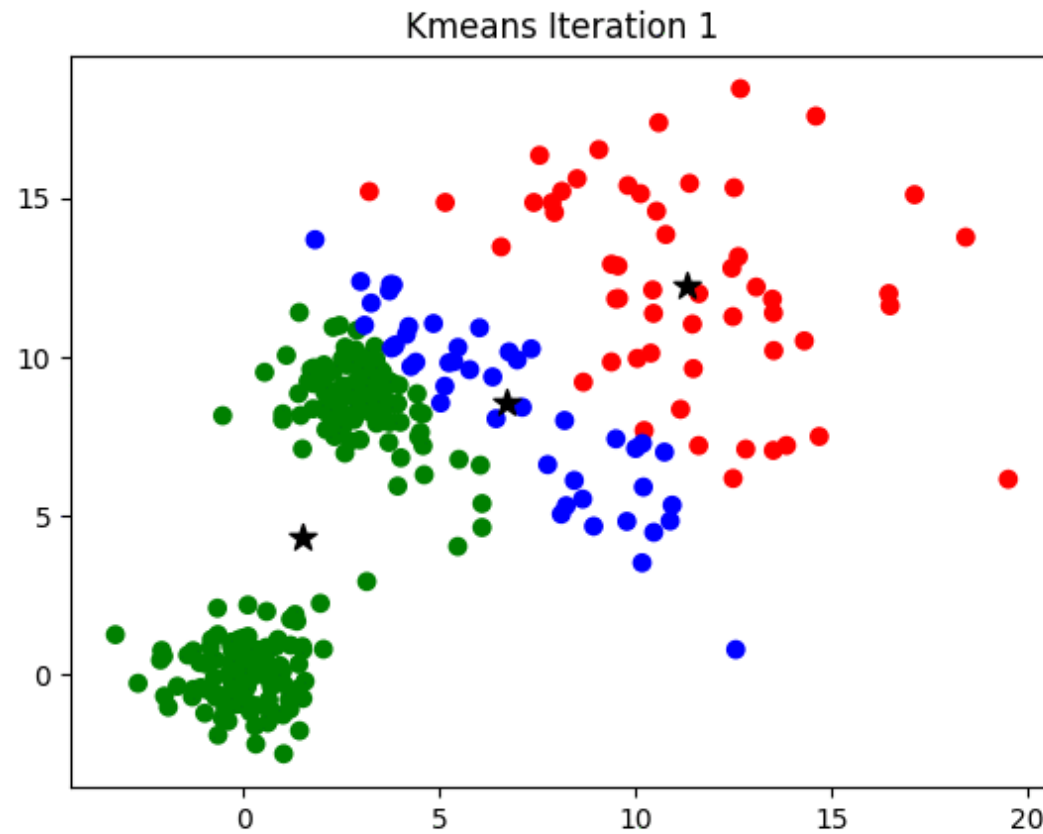


- K-means clustering (phân cụm K-means) là thuật toán cơ bản trong Unsupervised learning.
- Tâm là điểm đại diện nhất cho một cụm và có giá trị bằng trung bình của toàn bộ các quan sát nằm trong cụm.
- Dựa vào khoảng cách từ mỗi quan sát tới các tâm để xác định nhãn cho chúng.

# K-mean Clustering

- Cụ thể các bước của thuật toán k-Means được tóm tắt như sau:
  1. Chọn K điểm bất kỳ làm các center ban đầu.
  2. Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất.
  3. Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất cả các điểm dữ liệu đã được gán vào cluster đó sau bước 2.
  4. Nếu center không thay đổi so với trước đó thì dừng thuật toán. Ngược lại thì quay lại bước 2.

# K-mean Clustering

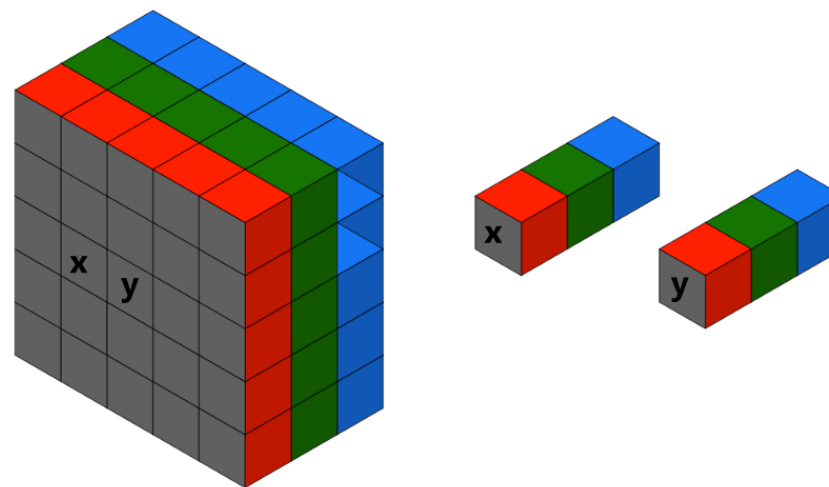


# K-mean Clustering trong nén ảnh

- ❑ Bước 1: Load ảnh input
- ❑ Bước 2: Làm phẳng hình ảnh
- ❑ Bước 3: Thực hiện K-means
- ❑ Bước 4: Thay thế các giá trị bằng các giá trị Centroid tương ứng
- ❑ Bước 5: Dựng lại ảnh từ mảng đã thực hiện qua K-means

# K-mean Clustering trong nén ảnh

- Mỗi pixel sử dụng chế độ ba màu RGB. Mỗi giá trị RGB cần 8 bit và  $3 \times 8 = 24$  bit lưu trữ cho mỗi pixel





# K-mean Clustering trong nén ảnh

- Sau khi dùng K-means Clustering thì chỉ số của mỗi pixel cần  $\log_2 K$  bit để lưu trữ.
  - + Kích thước ảnh trước khi nén:  $24 \cdot N$
  - + Kích thước ảnh sau khi nén:  $24 \cdot K + N \cdot \log_2 K$ 
    - Với  $k$  là số nhóm phân cụm K-means,  $N$  là số pixel

# Singular Value Decomposition

— Một ma trận  $A_{m \times n}$  bất kỳ đều có thể phân tích thành dạng:

$$A_{m \times n} = U_{m \times m} \times \Sigma_{m \times n} \times (V_{n \times n})^T$$

— Trong đó:

- $U$  và  $V$  là các ma trận trực giao.
- $\Sigma$  là ma trận đường chéo không vuông có giá trị giảm dần trên đường chéo và là các giá trị không âm.

# Singular Value Decomposition

$$\mathbf{A}_{m \times n} = \mathbf{U}_{m \times m} \times \Sigma_{m \times n} \times \mathbf{V}_{n \times n}^T$$

$(m < n)$

$$\mathbf{A}_{m \times n} = \mathbf{U}_{m \times m} \times \Sigma_{m \times n} \times \mathbf{V}_{n \times n}^T$$

$(m > n)$

Trong Python, để tính SVD của một ma trận, sử dụng module linalg của numpy

# Truncated SVD

$$A \approx U_k \times \Sigma_k \times (V_k)^T$$

$$\begin{array}{c}
 \begin{array}{c} \boxed{A} \\ m \times n \end{array} = \begin{array}{c} \boxed{U} \\ m \times m \end{array} \cdot \begin{array}{c} \boxed{\Sigma} \\ m \times n \end{array} \cdot \begin{array}{c} \boxed{V^T} \\ n \times n \end{array} \\
 \Downarrow \\
 \begin{array}{c} \boxed{\hat{A}} \\ m \times n \end{array} = \begin{array}{c} \boxed{U} \\ m \times k \end{array} \cdot \begin{array}{c} \boxed{\Sigma} \\ k \times k \end{array} \cdot \begin{array}{c} \boxed{V^T} \\ k \times n \end{array}
 \end{array}$$

# Truncated SVD trong nén ảnh

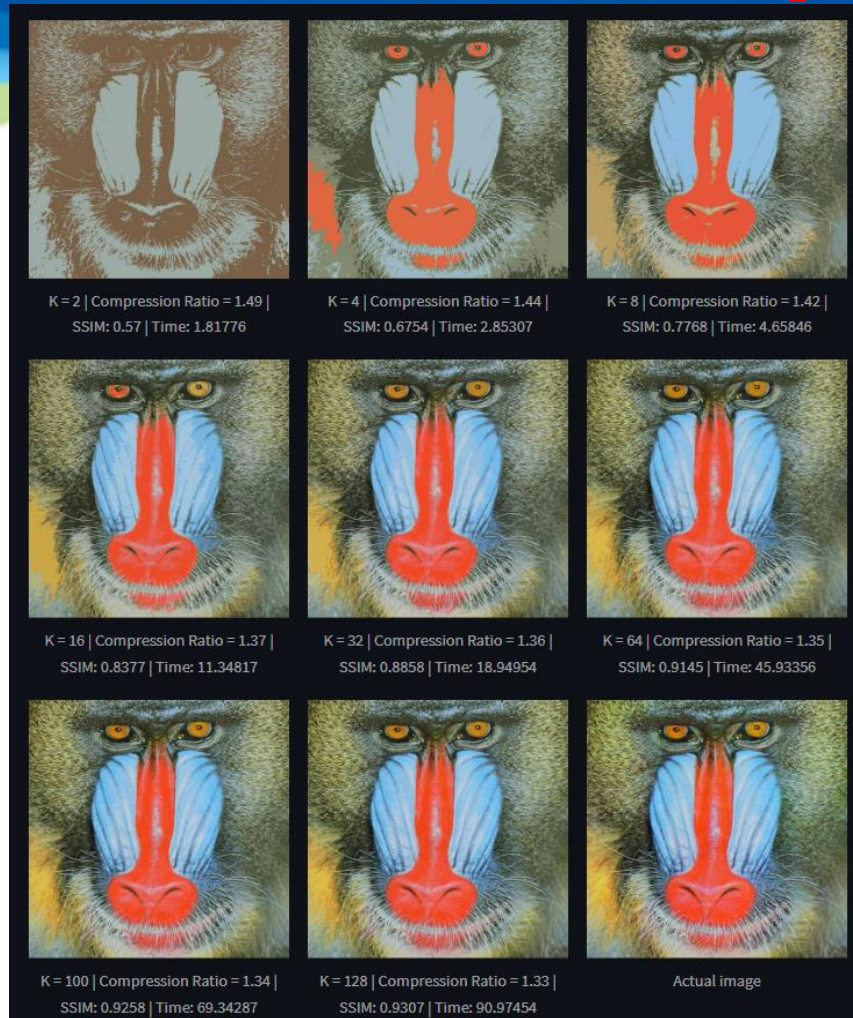
- ❑ Bước 1: Load ảnh input
- ❑ Bước 2: Tách từng kênh màu Red, Green, Blue
- ❑ Bước 3: Thực hiện tính SVD cho từng kênh màu
- ❑ Bước 4: Áp dụng Truncated SVD với từng kênh màu
- ❑ Bước 5: Gộp các kênh màu lại để tái tạo hình ảnh

# Truncated SVD trong nén ảnh

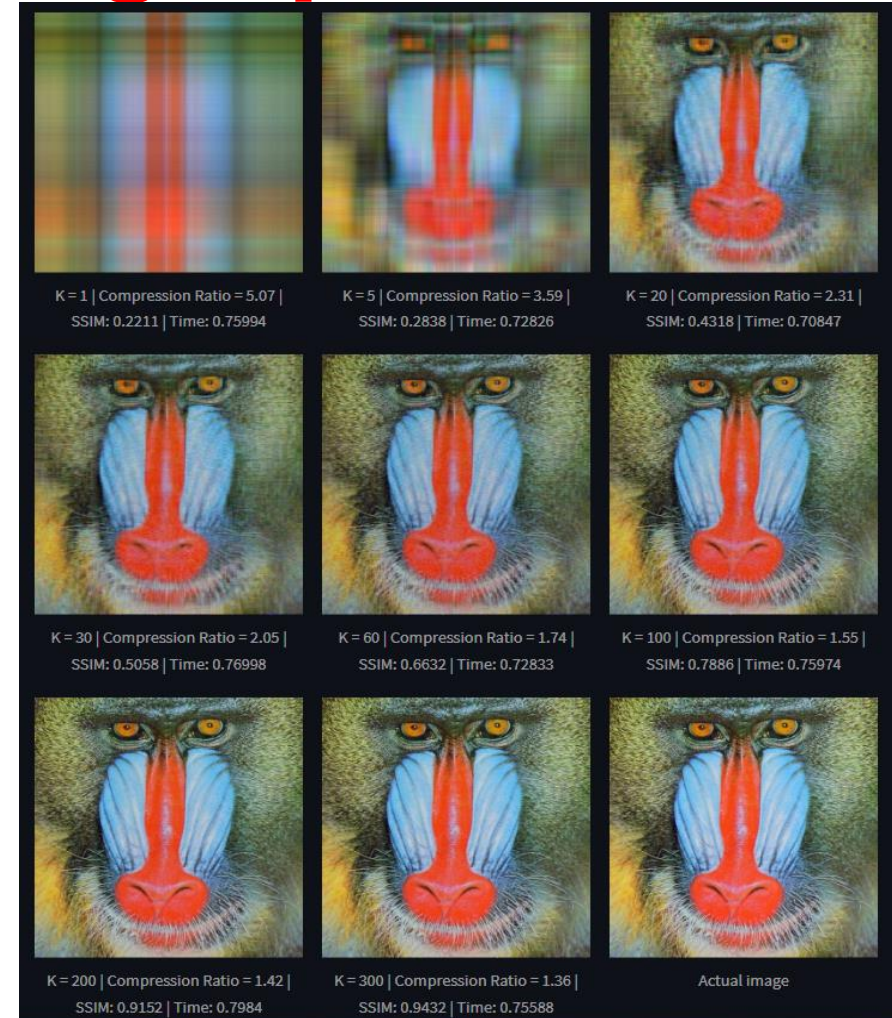
- Để lưu trữ ảnh sử dụng Truncated SVD chỉ cần lưu trữ các ma trận  $U_k$ ,  $\Sigma_k$ ,  $(V_k)^T$ . Số phần tử phải lưu là  $k \times (m+1+n)$ .
- Tỷ lệ nén:  $\frac{mn}{k(m+1+n)}$



# Kết quả thực nghiệm



Kmeans



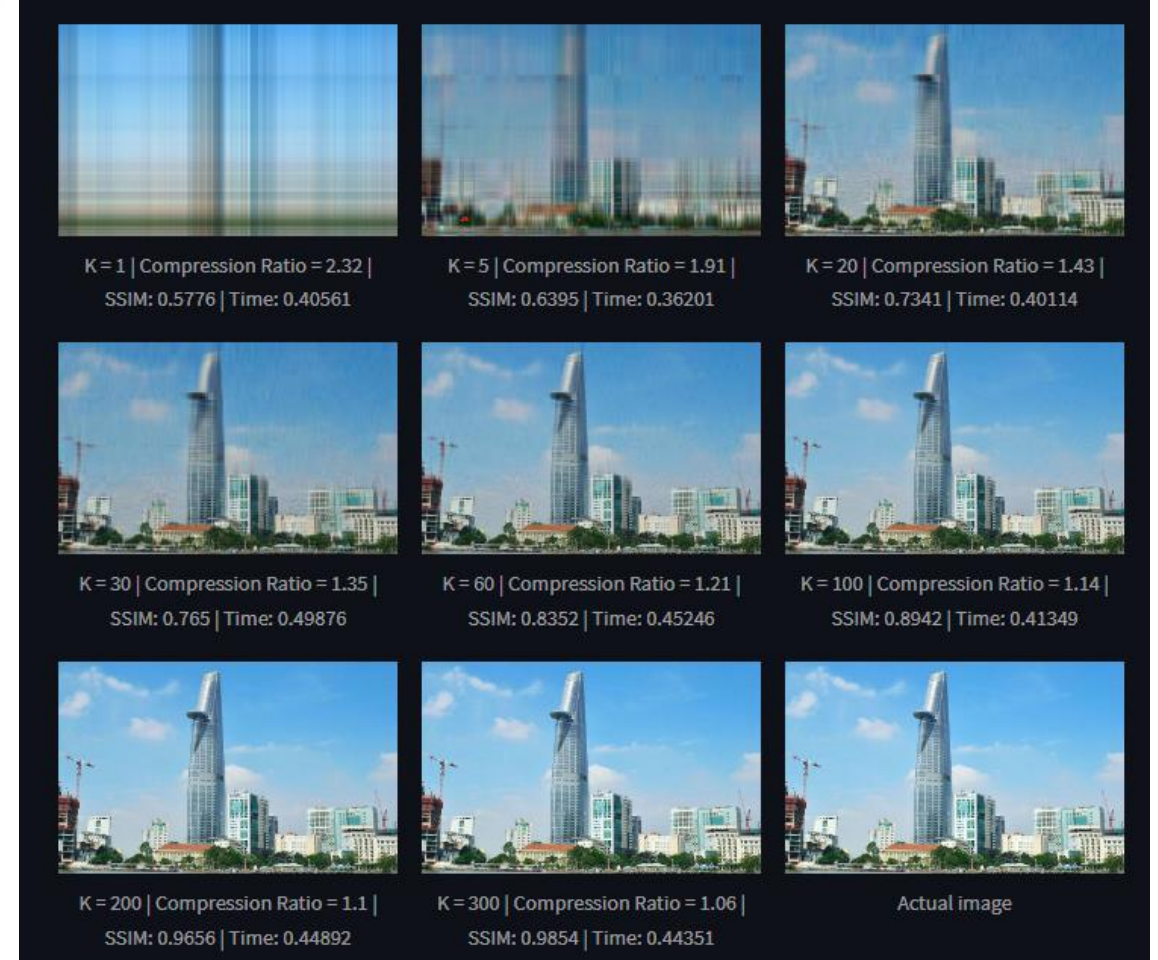
SVD



# Kết quả thực nghiệm






Kmeans



SVD



# Kết quả thực nghiệm

Ảnh gốc	After Compression with SVD	After Compression with Kmeans
		
	SSIM: 0.6957	SSIM: 0.907
	Ratio: 1.47	Ratio: 1.37
	Time: 1.48471	Time: 67.73427

# Kết quả thực nghiệm

Ảnh gốc	After Compression with SVD	After Compression with Kmeans
		
	SSIM: 0.3744	SSIM: 0.8562
	Ratio: 1.05	Ratio: 2.16
	Time: 2.63915	Time: 85.82003