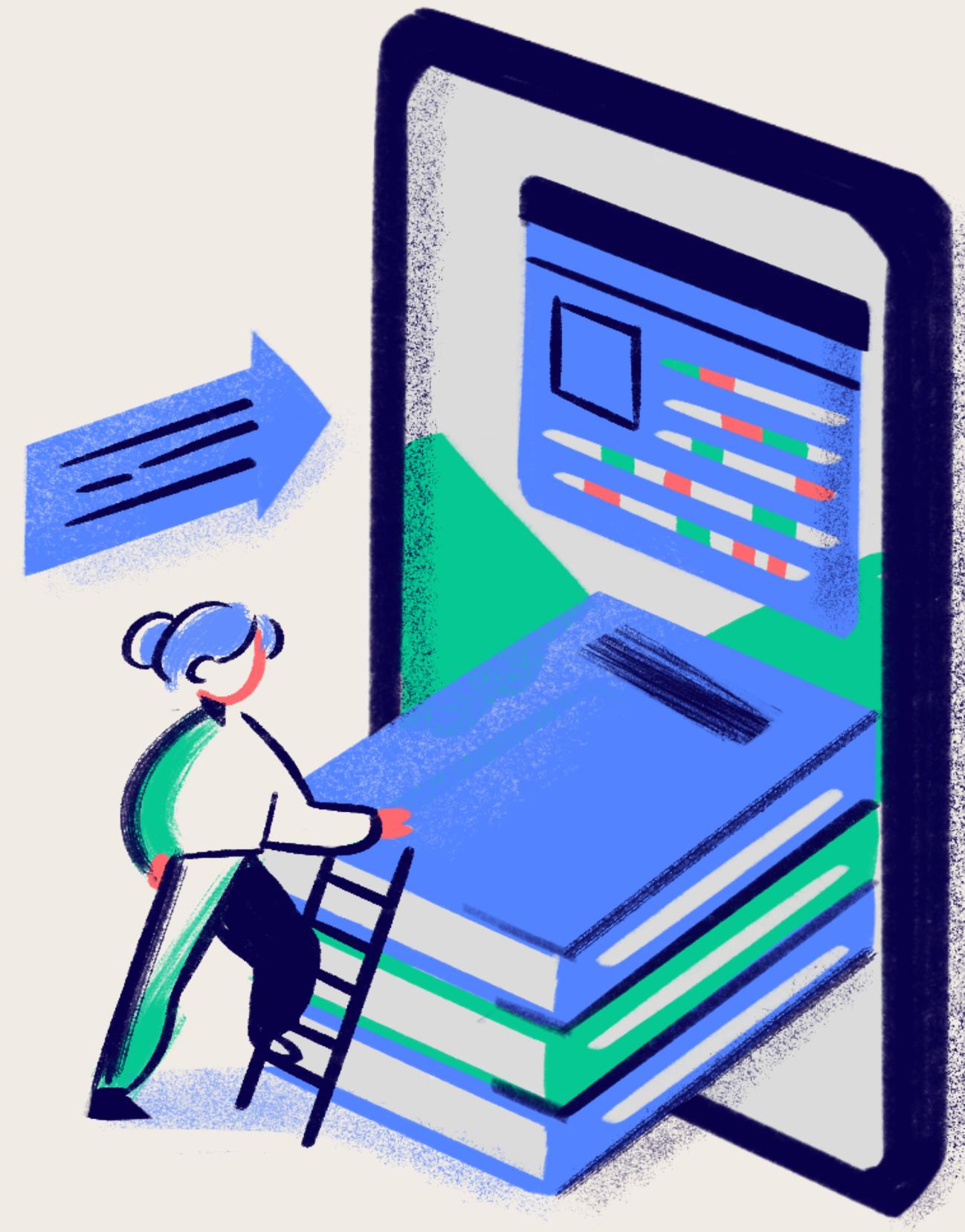


THUẬT TOÁN GIẤU TIN TRÊN MIỀN SVD

NHÓM 3:

- Lê Văn Minh
- Nguyễn Quỳnh Sơn
- Vũ Trung Kiên
- Đào Hoàng Hải
- Nguyễn Minh Quang



MỤC LỤC

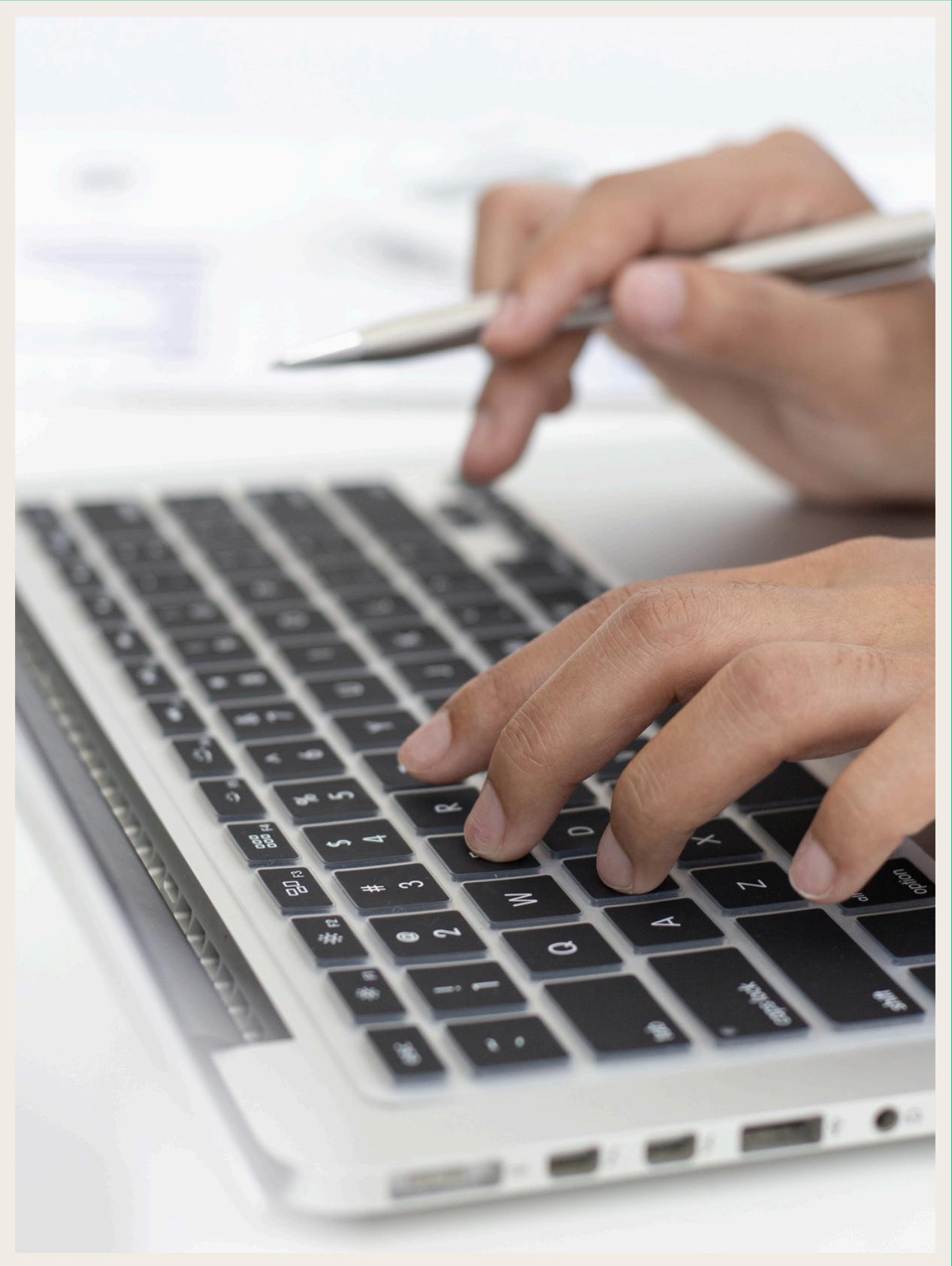
I. Tổng quan kỹ thuật giấu tin thuận nghịch

II. Tổng quan về SVD

III. Giấu tin thuận nghịch trên miền SVD



I. TỔNG QUAN KỸ THUẬT GIẤU TIN THUẬN NGHỊCH



Định nghĩa

Kỹ thuật giấu tin thuận nghịch (Reversible Data Hiding) là một lĩnh vực trong khoa học máy tính và mật mã học, cho phép chèn thông tin bí mật vào trong dữ liệu hình ảnh, âm thanh hoặc video mà không làm giảm quá nhiều chất lượng của dữ liệu gốc.

<insert>
<some_code>
<here>



Điểm nổi bật của phương pháp này là khả năng "thuận nghịch", tức là sau khi thông tin đã được giấu và dữ liệu được truyền hoặc lưu trữ, người nhận có khả năng khôi phục dữ liệu gốc mà không mất mát.

Các công nghệ và phương pháp thường gặp:

- Kỹ thuật LSB (Least Significant Bit): Thay đổi các bit ít quan trọng nhất trong dữ liệu để giấu thông tin mà không làm thay đổi đáng kể hình ảnh hoặc âm thanh.
- Kỹ thuật Mã hóa: Để bảo mật thông tin được giấu, kỹ thuật mã hóa có thể được áp dụng trước khi giấu.
- Phân tích Tần số: Sử dụng các kỹ thuật như DCT (Discrete Cosine Transform) hoặc DWT (Discrete Wavelet Transform) để ẩn thông tin trong các tần số cao, nơi mà sự biến đổi ít nhận thấy hơn.

Ứng dụng

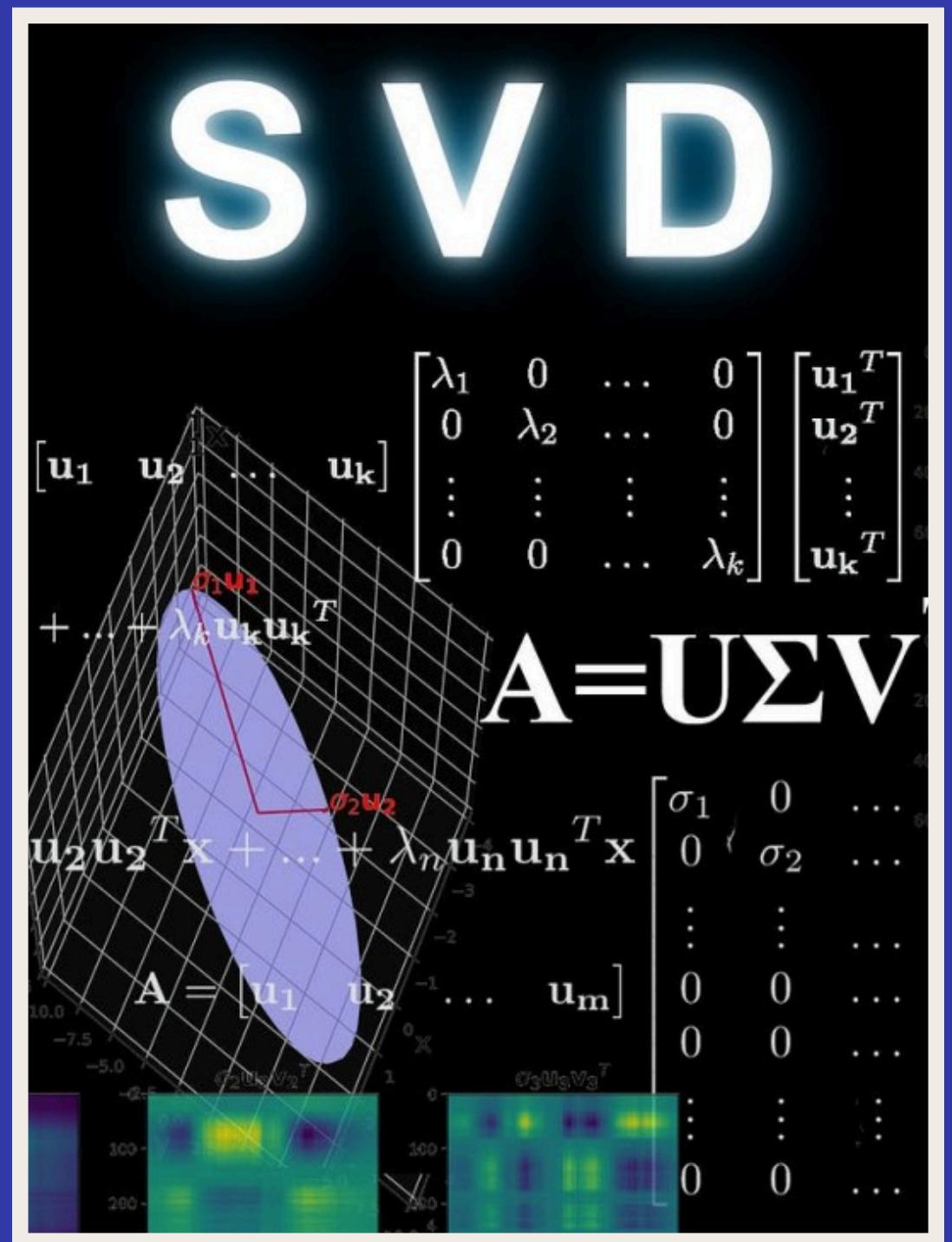
Ghi Nhận Bản Quyền: Thêm thông tin bản quyền vào hình ảnh hoặc video

Chống Sao Chép: Giấu thông tin nhận diện trong sản phẩm kỹ thuật số để theo dõi phân phối.

Giao Tiếp An Toàn: Truyền tải thông tin bí mật dưới dạng tín hiệu mà không rõ ràng cho bên ngoài.

Kết Luận

Kỹ thuật giấu tin thuận nghịch có thể bảo vệ sự riêng tư và an toàn thông tin, đồng thời mở ra nhiều khả năng mới cho việc bảo mật dữ liệu trong các hệ thống số ngày nay.



2. TỔNG QUAN VỀ SVD

KHÁI NIỆM

$$A = U.S.V^*$$

SVD hay Phân rã Giá trị Đặc (Singular Value Decomposition), là một phương pháp phân rã ma trận. Đối với một ma trận hình chữ nhật A có kích thước $m \times n$, SVD cho phép chúng ta phân rã ma trận này thành ba ma trận khác U, S, V

V^* : là ma trận chuyển vị của ma trận V

U: là ma trận trực giao với kích thước $M \times M$ có $U.U^* = I$ (ma trận đơn vị)

V: là ma trận trực giao với kích thước $N \times N$ có $V.V^* = I$

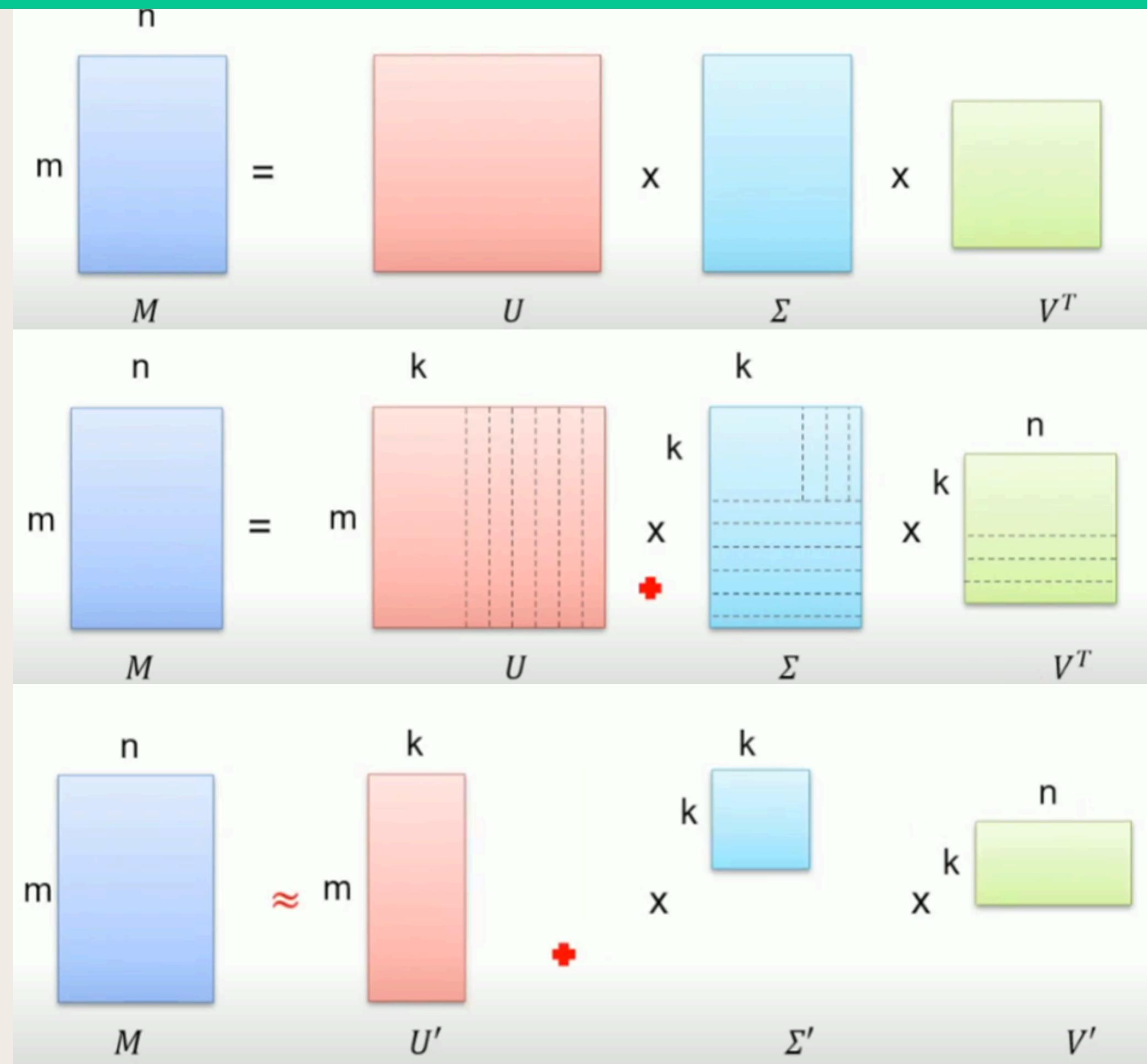
S: là ma trận đường chéo hình chữ nhật có kích thước $m \times n$ chứa các giá trị không âm (giá trị kỳ dị - singular values) và được sắp xếp theo thứ tự giảm dần.

$$M_{m \times n} = U_{m \times m} \Sigma_{m \times n} V^*_{n \times n}$$

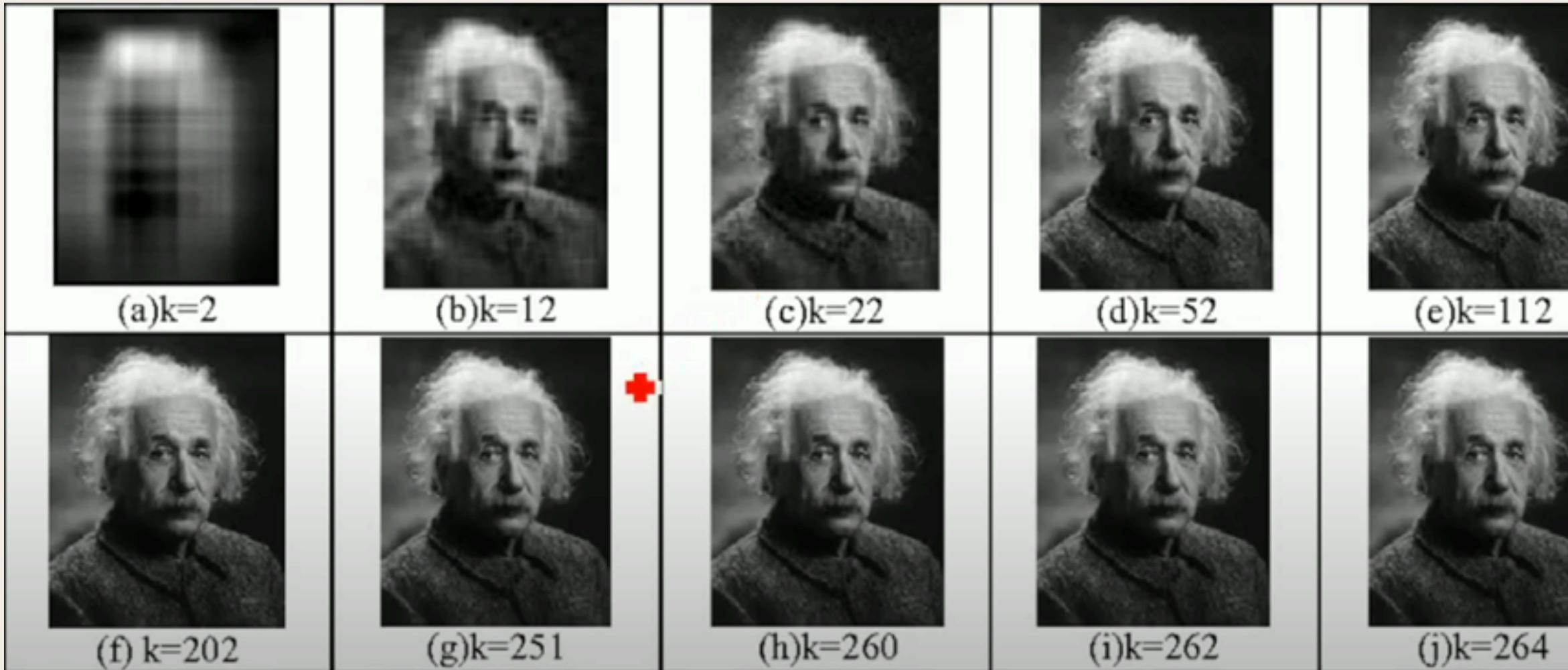
$$U \quad U^* = I_m$$

$$V \quad V^* = I_n$$

Phương pháp tìm SVD của một ma trận khá phức tạp, cần kết hợp rất nhiều kiến thức như vector riêng của ma trận, giá trị riêng của ma trận,... Hiện nay, có rất nhiều ngôn ngữ lập trình, phần mềm như Python, Matlab,... đã có thể tính toán nhanh SVD của một ma trận bất kỳ



CHẤT LƯỢNG ẢNH KHI THAY ĐỔI K



- SVD được áp dụng phổ biến vào trong xử lý ảnh, đặc biệt là trong các bài toán nén ảnh. Điều này giúp giảm lượng thông tin cần lưu trữ mà vẫn giữ được chất lượng ảnh ở mức chấp nhận được.
- Để có thể nén ảnh, ta sử dụng một tham số k là số lượng giá trị trên đường chéo chính của ma trận S được giữ lại. Khi đó, các phần tử còn lại của ma trận S được gán giá trị là 0.

TÍNH CHẤT

- Khi phân tích ma trận A thành U.S.V*, mỗi ma trận A chỉ cho ra một kết quả S duy nhất, nhưng có thể cho ra nhiều ma trận U và V

Ví dụ:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

Khi đó:

$$\mathbf{U} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & \sqrt{5} & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & \textcolor{red}{0} & 0 \end{bmatrix}$$

$$\mathbf{V}^* = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ -\sqrt{0.2} & 0 & 0 & 0 & -\sqrt{0.8} \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\sqrt{0.8} & 0 & 0 & 0 & \sqrt{0.2} \end{bmatrix}$$

```
Ma tran tai tao tu U, S, Vt (voi k = 4):  
[[1. 0. 0. 0. 2.]  
  
[0. 0. 3. 0. 0.]  
  
[0. 0. 0. 0. 0.]  
  
[0. 2. 0. 0. 0.]]
```

```
[[1. 0. 0. 0. 2.]  
  
[0. 0. 3. 0. 0.]  
  
[0. 0. 0. 0. 0.]  
  
[0. 2. 0. 0. 0.]]
```

- Với k=3, 2 kết quả cho ra tương tự

- Với K = 1:

```
Ma tran tai tao tu U, S, Vt (voi k = 1):  
[[0. 0. 0. 0. 0.]  
  
[0. 0. 3. 0. 0.]  
  
[0. 0. 0. 0. 0.]  
  
[0. 0. 0. 0. 0.]]
```

```
[[0. 0. 0. 0. 0.]  
  
[0. 0. 3. 0. 0.]  
  
[0. 0. 0. 0. 0.]  
  
[0. 0. 0. 0. 0.]]
```

ỨNG DỤNG

SVD CÓ NHIỀU ỨNG DỤNG TRONG CÁC LĨNH VỰC KHÁC NHAU, BAO GỒM

Giải quyết hệ phương trình tuyến tính

Giúp tìm nghiệm gần đúng cho các hệ phương trình không có nghiệm duy nhất

Hệ thống gợi ý

SVD có thể được sử dụng để phân tích ma trận tương tác giữa người dùng và sản phẩm nhằm đưa ra gợi ý

Giảm kích thước dữ liệu

Trong phân tích thành phần chính (PCA), SVD được sử dụng để giảm số chiều của dữ liệu trong khi vẫn giữ được các đặc trưng quan trọng

Nén hình ảnh

SVD giúp nén hình ảnh bằng cách giữ lại các thành phần chính trong ma trận pixel và loại bỏ những thành phần ít quan trọng hơn

ƯU ĐIỂM, NHƯỢC ĐIỂM

Ưu điểm:

- Cung cấp một cách tiếp cận mạnh mẽ để phân tích và xử lý dữ liệu.
- Dễ dàng thực hiện và có thể mở rộng cho các ma trận lớn.
- Giúp tiết kiệm bộ nhớ với khả năng nén mạnh mẽ

Nhược điểm:

- Tính toán SVD có thể tốn kém về mặt tài nguyên, đặc biệt với các ma trận lớn.
- Kết quả có thể khó giải thích trong một số ngữ cảnh cụ thể.

TRONG XỬ LÝ ẢNH

Ưu điểm:

- Giảm kích thước dữ liệu
- Khử nhiễu
- Phân tích cấu trúc ảnh
- Độ chính xác cao trong phân tích ma trận
- Đối với ảnh có kích thước lớn, SVD có thể rút gọn kích thước mà không mất quá nhiều thông tin

Nhược điểm:

- Thời gian phân tích lâu ($O(n^3)$) cho ma trận vuông kích thước $n \times n$
 - Không thích hợp cho ảnh có nhiễu phức tạp
- Yêu cầu bộ nhớ lớn để tích trữ các ma trận con (U, Σ, V)

3. GIẤU TIN THUẬN NGHỊCH TRÊN MIỀN SVD



A. THỬ NGHIỆM TÍNH CHẤT SVD

- Ta có ma trận A kích thước $m \times n$, tách thành $U_a.S_a.V_a^*$.
- Thay một vài giá trị kì dị trên S_a mà vẫn giữ nguyên tính chất ta thu được S_b .
- Tính $U_a.S_b.V_a^* = B$ (ma trận số thực B)
- Chuyển B về dạng ảnh (làm tròn) thu được ma trận C do bị làm tròn nên C sẽ không thể tách thành $U_a.S_b.V_a^*$ được mà là các giá trị mới $U_c.S_c.V_c^*$ có độ lệch khá cao so với giá trị ban đầu.

A. THỬ NGHIỆM TÍNH CHẤT SVD

Ma tran A:

```
[[100  10  240  70  160]
 [ 76   46  235 142   63]
 [182 198   58   54 124]
 [191 235     1   62 172]]
```

Ma tran SA:

```
[550.37853783 304.36843042 90.12760585 29.82177606]
```

Ma tran SB:

```
[550.37853783 304.36843042 90.12760585 0]
```

Ma tran C:

```
[[100. 10. 240. 71. 160.]
 [ 78. 46. 237. 139. 60.]
 [171. 196. 50. 69. 136.]
 [200. 236.    7. 51. 163.]]
```

A. THỬ NGHIỆM TÍNH CHẤT SVD

Từ đây ta có thể thấy ma trận C đã đánh mất các thông tin trên ma trận A mặc dù hình ảnh không bị thay đổi quá nhiều khi nhìn bằng mắt thường

Nhưng ở một khía cạnh khác, ta có thể kiểm soát độ lớn tương đối các giá trị trong ma trận Sc bằng cách tăng giảm một vài giá trị trong Sa, tạo ra Sc0 và Sc1, làm cho 2 ảnh c0 với c1 không thể phân biệt được bằng mắt thường nhưng có thể phân biệt mức độ tác động giá trị tăng giảm lên Sa một cách rõ rệt.

A. THỬ NGHIỆM TÍNH CHẤT SVD



Ảnh A



Ảnh C0



Ảnh C1

B. DỰ ÁN

Ví dụ một văn bản đơn giản là “Cat” và một bức ảnh xám hình con mèo có kích thước 318x159:



Một bức ảnh con mèo ở dạng rgb



Bức ảnh con mèo khi chuyển sang màu xám

B. DỰ ÁN

Ta có văn bản cần giấu “Cat” qua bảng ASCII được biểu diễn dưới dạng bit như sau: 0100 0011 0110 0001 0111 0100

Tiếp theo, chia hình ảnh đầu vào thành các khối 4×4 và tính toán SVD cho từng khối. Thông điệp được chèn từng bit một vào các khối này. Sau khi SVD được tính toán, bit thông điệp sẽ được chèn vào giá trị kỳ dị ít quan trọng nhất của ma trận kỳ dị (ma trận S), mục đích giảm lượng thông tin bị mất trong hình ảnh. Để trích xuất giá trị này, SVD sẽ được tính toán lại và các bit thông điệp được nhúng sẽ được trích xuất và lắp ráp lại.

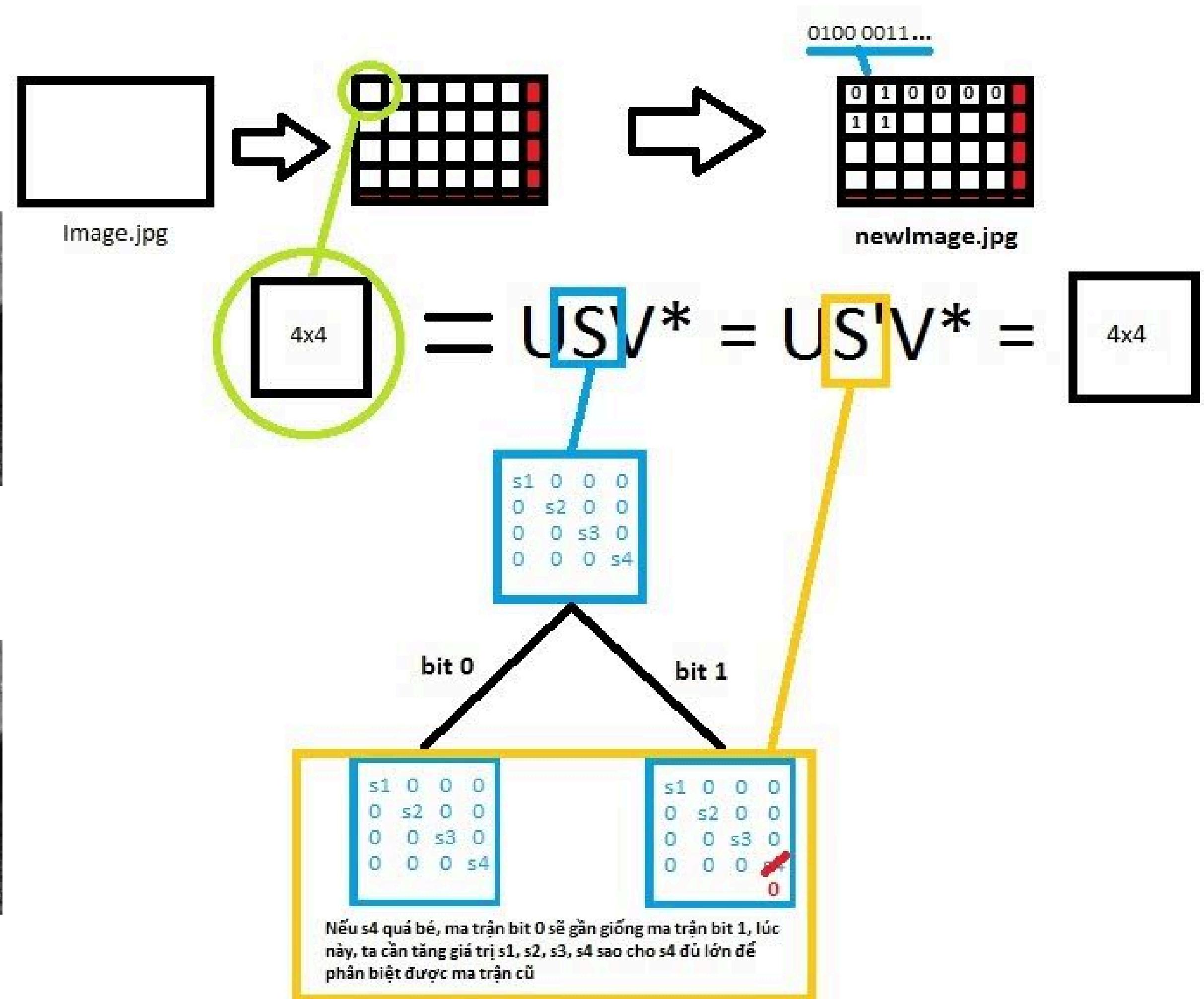
B. DỰ ÁN



Ảnh gốc



Ảnh sau khi mã hóa



C. NHẬN XÉT

Ưu điểm:

- Thuật toán mã hóa này khá đơn giản và tốc độ chạy khá nhanh, đồng thời ảnh sau khi mã hóa không có nhiều thay đổi đáng kể.
- Có thể giấu tin trong ảnh rgb bằng cách tách bức hình rgb ra thành các bức xám và giấu tin vào trong những bức xám đó
- Để gia tăng độ bảo mật của thuật toán, ta có thể chia bức hình thành các kích thước khác nhau ngoài 4×4 như 6×6 , 7×4 , ... Đây có thể coi là khóa bí mật của thuật toán, khiến tin tặc dù có biết được thuật toán cũng cần mất thêm thời gian để giải quyết mật mã.

C. NHẬN XÉT

Nhược điểm:

- Do đây là một phương pháp đơn giản, nếu tin tặc nắm vững thuật toán và có phương pháp tìm khóa bí mật nhanh dựa vào 8 bit đầu thì có thể giải mã một cách dễ dàng.
- Nên áp dụng thêm các thuật toán khác như DWT-SVD để mã hóa thông tin

**THANK
YOU VERY
MUCH!**

