

## Bài tập 5 - Ứng dụng của truncated SVD trong việc nén ảnh

Viết chương trình cho phép nén **ảnh** theo phương pháp truncated SVD và xuất ảnh đã bị nén với 50, 100, 250 và 300 lần lượt là số  $\lambda$  lớn nhất để lưu giữ thông tin. Sau đó, xuất ra lượng thông tin còn giữ lại tương ứng.



Hình 1: Hình gốc

## Lưu trữ ảnh dưới dạng mảng

Để thao tác với ảnh trong chương trình, chúng ta cần phải lưu trữ hình ảnh có kích thước  $m \times n$  sang dạng mảng bằng câu lệnh:

```
A = numpy.array(Image.open('file ảnh'))
```

Lúc này,  $A$  là một mảng gồm 3 ma trận cấp  $m \times n$  đại diện cho các cường độ khác nhau của 3 màu đỏ, xanh lá và xanh dương. Mảng  $A$  được gọi là ảnh RGB. Trong đó, màu sắc của mỗi pixel của ảnh được tạo ra từ sự kết hợp bởi các cường độ khác nhau của các màu đỏ, xanh dương và xanh lá. Các cường độ này có giá trị nằm trong khoảng từ 0 đến 255.

Ví dụ: Nhấp vào đường [link](#) sau để chọn màu RGB.

## Làm xám ảnh

Để nén ảnh, mình cần tìm truncated SVD của mảng  $A$  dùng để lưu trữ ảnh.

Tuy nhiên, hàm `svd` của Scipy chỉ làm việc với một ma trận nên chúng ta cần phải chuyển  $A$  từ 3 ma trận sang 1 ma trận, tức là biến ảnh màu RGB thành ảnh xám (grayscale image) với màu sắc của mỗi pixel chỉ được tạo thành từ cường độ của ánh sáng, hay nói cách khác là chỉ các sắc thái của màu xám.

Cách chuyển đổi:

- ▶ **Dùng công thức:**

$$A = A.\text{dot}([0.299, 0.5870, 0.114])$$

- ▶ **Dùng OpenCV:** Sinh viên khai báo thư viện `cv2` và áp dụng

$$A = \text{cv2.cvtColor}(A, \text{cv2.COLOR\_RGB2GRAY})$$

## Lưu ý:

- ▶ Khi sử dụng công thức trên và vẽ ảnh bằng `imshow()`, ảnh sẽ có màu xanh. Thế nhưng, đó vẫn là ảnh xám. Để vẽ ảnh ra màu xám đẹp, khi vẽ ảnh, ta làm như sau:

```
plt.imshow(A, cmap = 'gray')
```

- ▶ Ngoài những cách này, còn rất nhiều cách khác giúp làm xám ảnh. Ví dụ như dùng Pillow,... Sinh viên có thể dùng bất cứ phương pháp nào tùy ý.

## Thuật toán

- ▶ Đầu vào: Hình ảnh chùa Một Cột.
- ▶ Đầu ra: Hình ảnh chùa đã bị nén lần lượt với số giá trị  $\lambda$  được giữ lại theo đề bài và phần trăm lượng thông tin giữ lại tương ứng.

Các bước làm bài:

1. Khai báo các thư viện cần thiết.
2. Dùng hàm `numpy.array(Image.open('file ảnh'))` để đọc hình ảnh và biến hình ảnh thành dạng mảng, đặt là  $A$ .
3. Nhân  $A$  với mảng  $(0.299, 0.5870, 0.1140)$  để tạo ảnh xám.
4. Tìm truncated SVD của  $A$  như trên với 50 số  $\lambda$  lớn nhất được giữ lại.
5. Tạo lại ma trận  $A$  từ 3 ma trận của truncated SVD. Đây chính là hình ảnh sau khi đã nén.
6. Áp dụng công thức (6) để tính phần trăm lượng thông tin giữ lại và xuất kết quả.
7. Vẽ ảnh sau khi đã nén và trước khi nén để so sánh. (dùng `imshow`)
8. Làm tương tự với số  $\lambda$  còn lại.