



# CS231. Nhập môn Thị giác máy tính

**Đặc trưng màu sắc**



# Mục tiêu

---

- Sinh viên hiểu và áp dụng các phương pháp trích xuất đặc trưng dựa trên màu sắc để phân tích và nhận diện đối tượng trong ảnh.

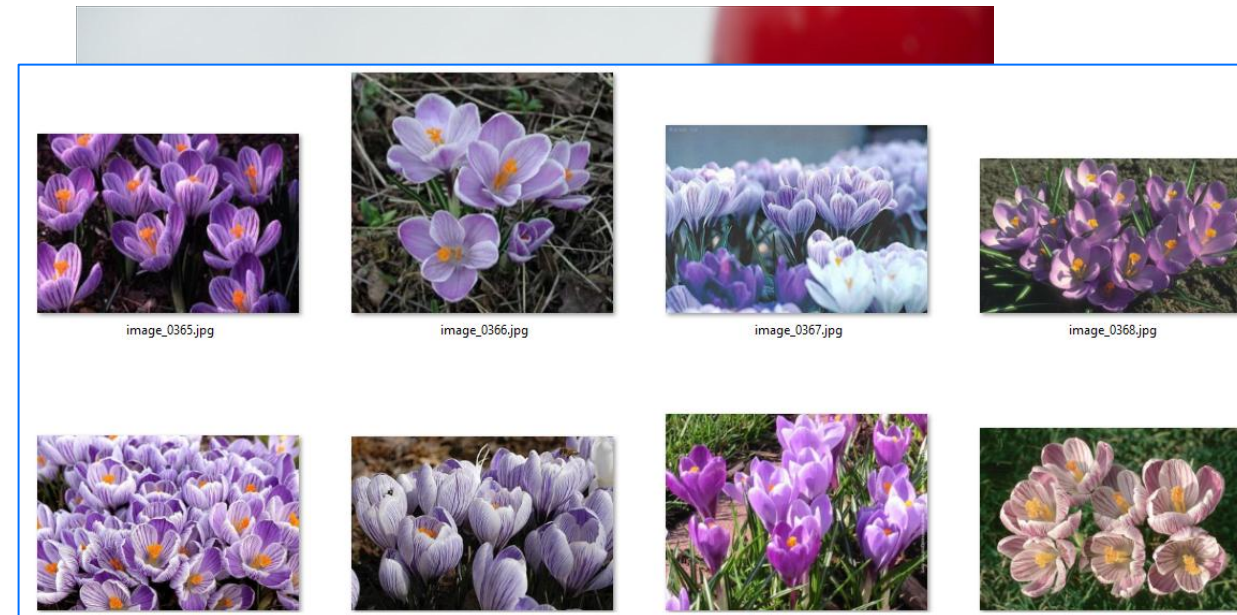
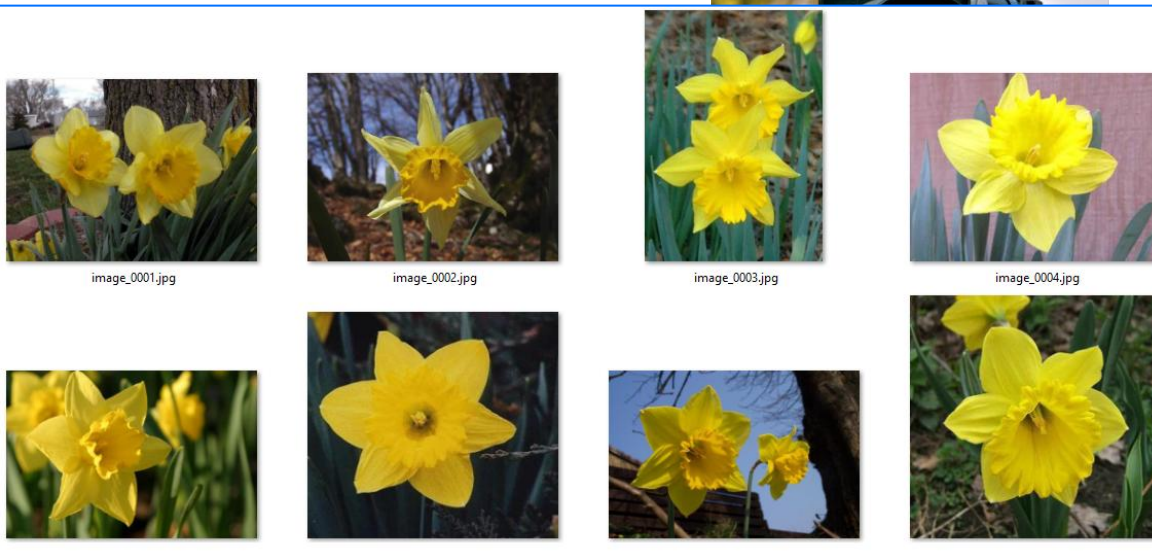
# Dẫn nhập

- Màu sắc đóng vai trò quan trọng trong cuộc sống.



# Dẫn nhập

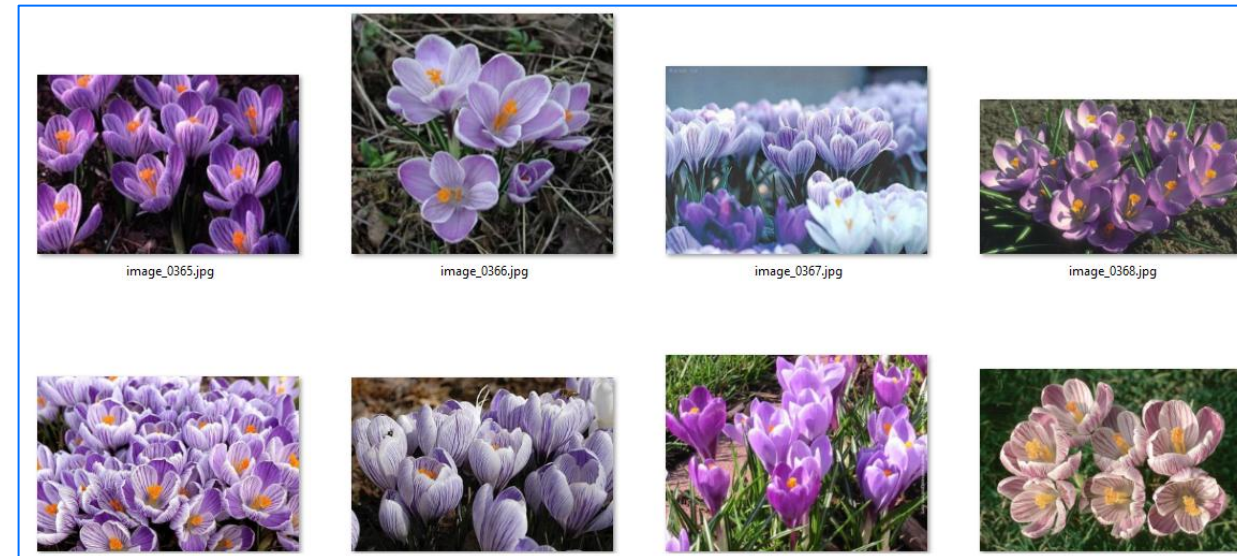
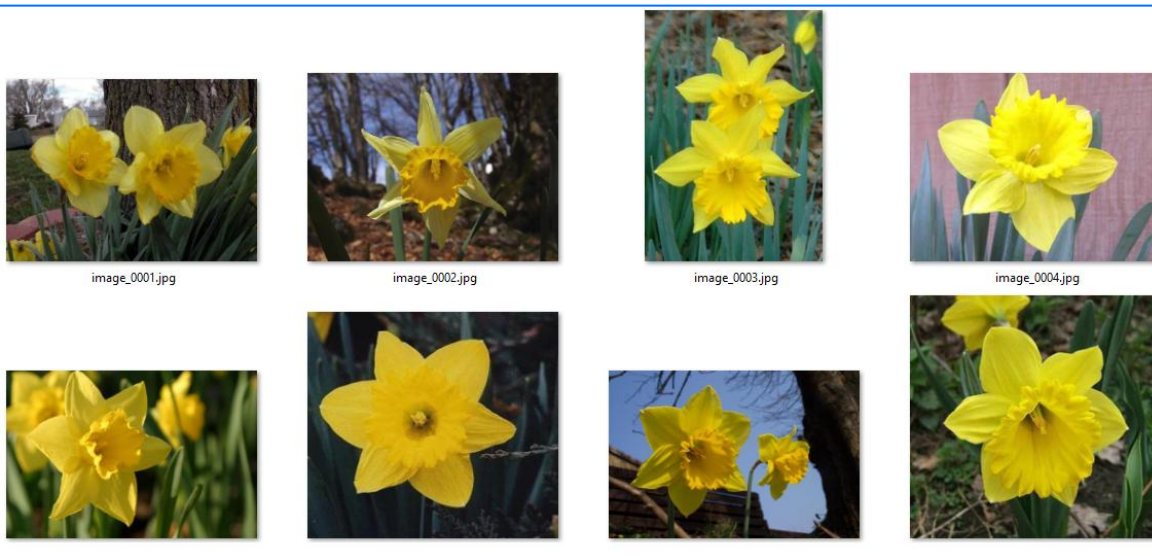
- Màu sắc đóng vai trò quan trọng trong cuộc sống.
- Có thể phân biệt các đối tượng, vật thể khác nhau dựa trên màu sắc





# Dẫn nhập

- Màu sắc đóng vai trò quan trọng trong cuộc sống.
- Có thể phân biệt các đối tượng, vật thể khác nhau dựa trên màu sắc





# Các đặc trưng màu sắc phổ biến

---

- Histogram màu
- Đặc trưng moment màu
- Màu sắc chính của ảnh
- Color Correlogram



# Histogram màu

---

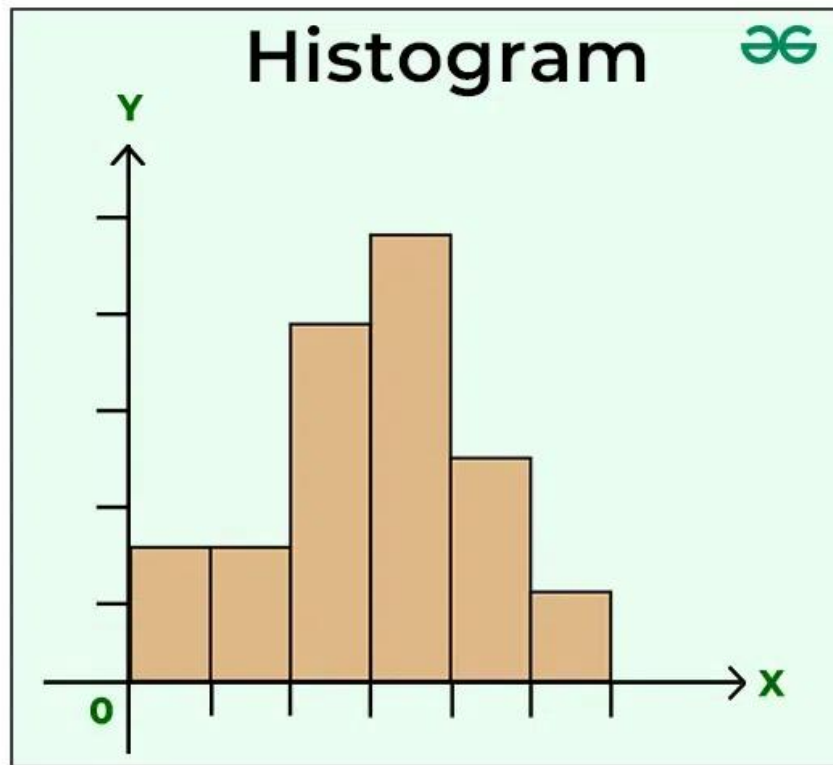
## Color Histogram

# Định nghĩa Histogram

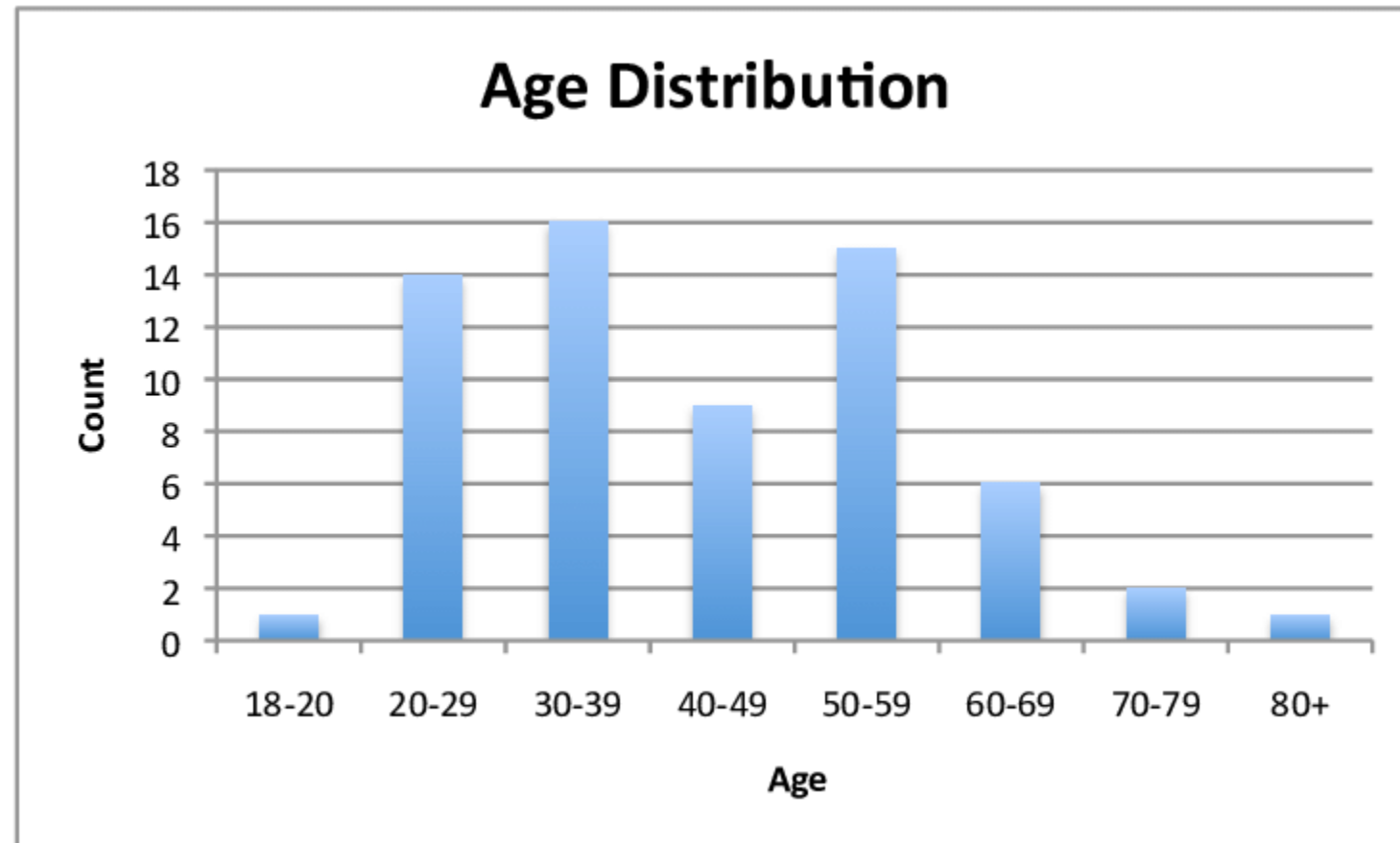
- **Histogram** là một phương pháp thống kê để biểu diễn **phân bố tần suất** của dữ liệu.
- Histogram, hay còn gọi là **biểu đồ tần suất**, là một dạng **biểu đồ cột** dùng để biểu diễn phân phối tần suất của một tập dữ liệu liên tục. **Mỗi cột** trong biểu đồ đại diện cho một **khoảng giá trị (bin)**, và **chiều cao** của cột thể hiện **số lượng** điểm dữ liệu nằm trong khoảng đó.



# Định nghĩa Histogram

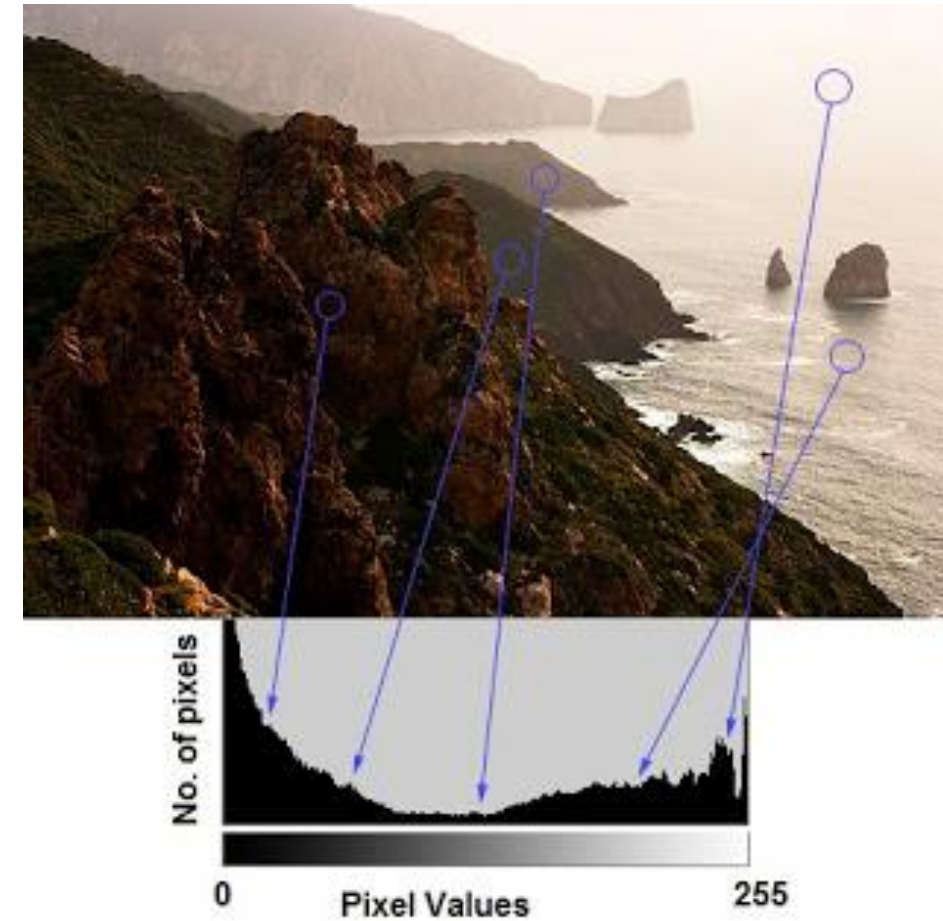


<https://www.geeksforgeeks.org/histogram/>



# Histogram màu trong ảnh

- **Histogram màu** thể hiện **tần suất xuất hiện** của **từng mức màu** trong một không gian màu cụ thể, chẳng hạn như:
  - Gray scale
  - RGB (Red, Green, Blue)
  - HSV (Hue, Saturation, Value)
  - Lab (Luminance, a, b)



[https://docs.opencv.org/4.5.1/d1/db7/tutorial\\_py\\_histogram\\_begins.html](https://docs.opencv.org/4.5.1/d1/db7/tutorial_py_histogram_begins.html)



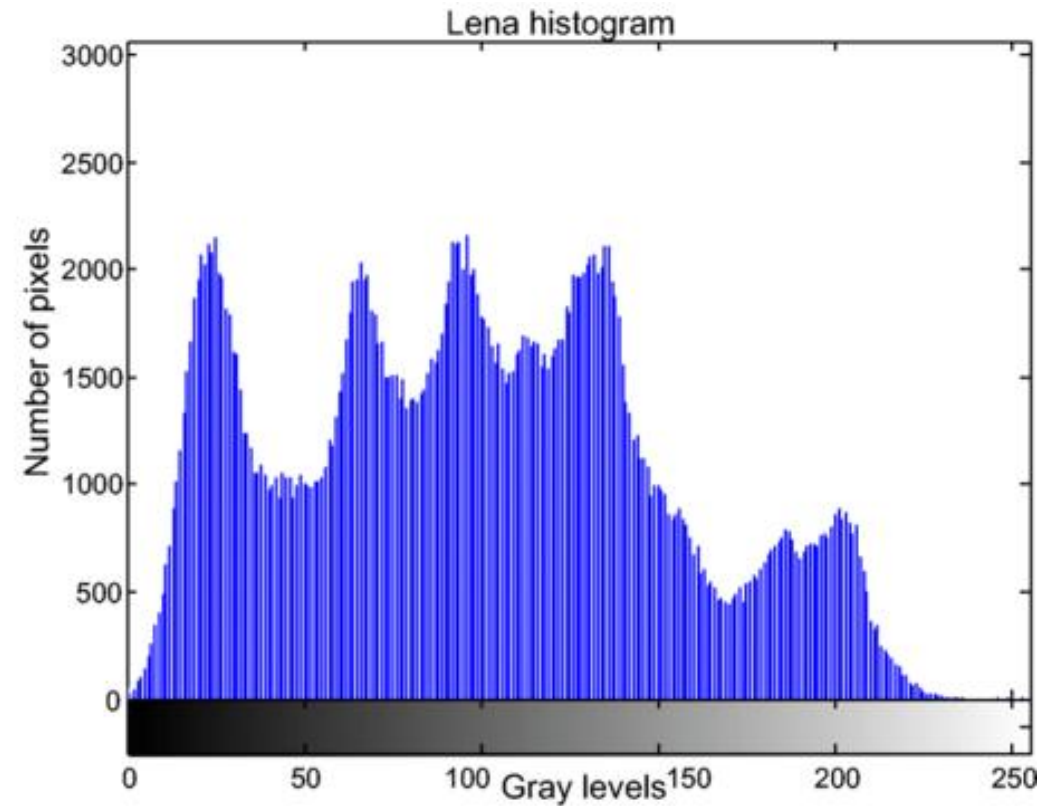
# Cách tính Histogram màu

- 1. Chia ảnh thành các kênh màu (Ví dụ: R, G, B hoặc H, S, V).**
- 2. Chia mỗi kênh thành các khoảng (bins) – ví dụ, 256 mức từ 0 đến 255.**
- 3. Đếm số lượng pixel trong mỗi bin để xác định phân bố màu sắc**

# Cách tính Histogram màu



(a)



(b)

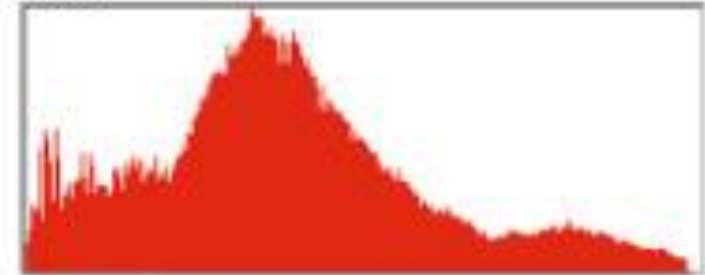
[https://www.researchgate.net/figure/a-Original-lena-image-b-Histogram-of-lena-image\\_fig1\\_335591569](https://www.researchgate.net/figure/a-Original-lena-image-b-Histogram-of-lena-image_fig1_335591569)



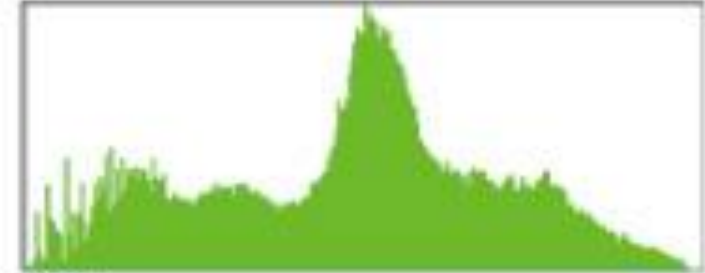
# Cách tính Histogram màu



RED



GREEN



BLUE



<https://vn.canon/vi/support/8200098700>

# Cách tính Histogram màu





# Code hiển thị Histogram màu

```
import cv2
import matplotlib.pyplot as plt

def display_histogram_RGB(image_path, bins=256):
    # Đọc ảnh
    image = cv2.imread(image_path)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # Chuyển từ BGR sang RGB

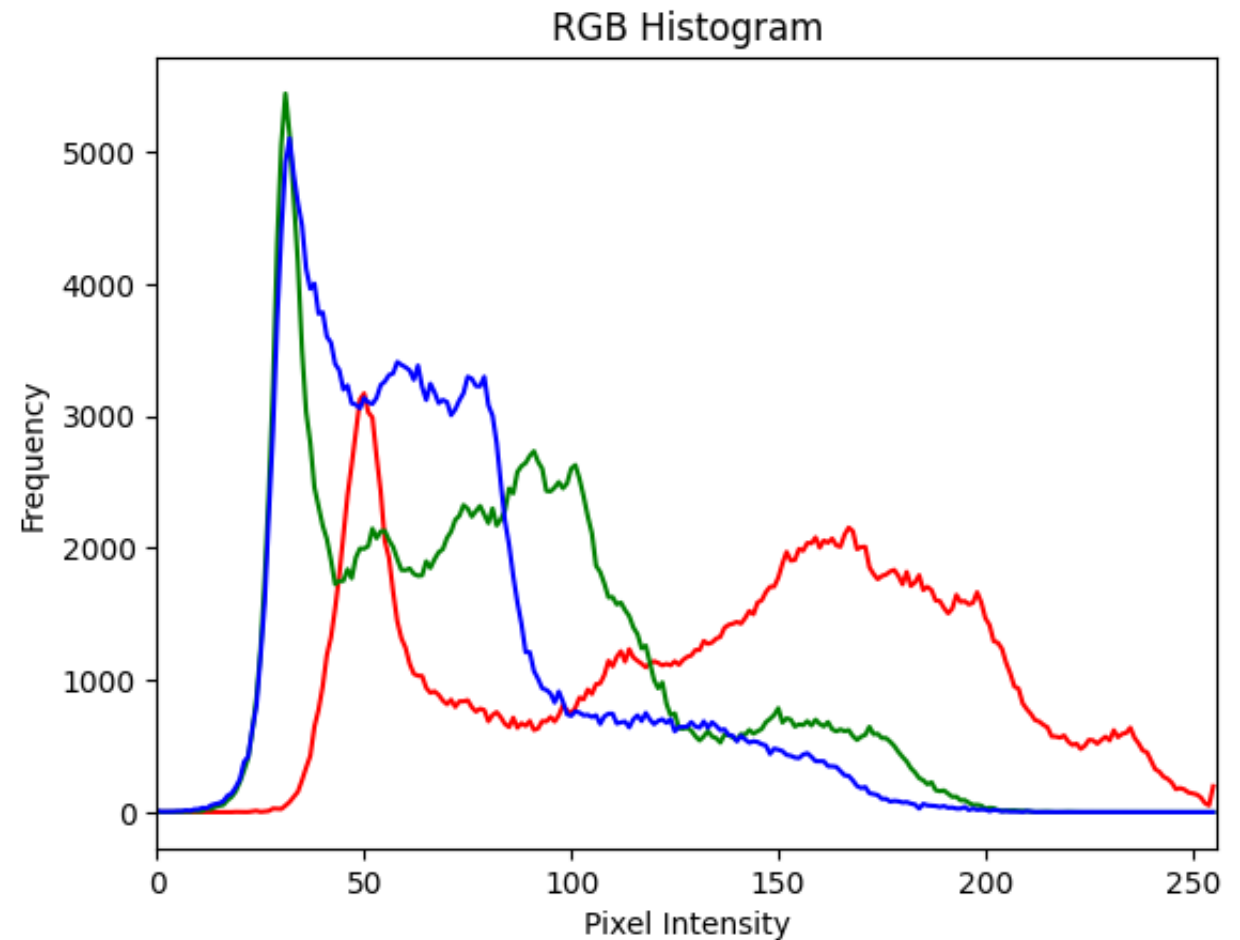
    # Tạo histogram cho từng kênh màu
    colors = ('r', 'g', 'b')
    plt.figure()
    plt.title('RGB Histogram')
    plt.xlabel('Pixel Intensity')
    plt.ylabel('Frequency')

    for i, color in enumerate(colors):
        hist = cv2.calcHist([image], [i], None, [bins], [0, 256])
        plt.plot(hist, color=color)

    plt.xlim([0, 256])
    plt.show()
```

# Code hiển thị Histogram màu

```
# Thay thế 'image_path' bằng đường dẫn đến ảnh của bạn  
image_path = "/content/lena.jpg"  
display_histogram_RGB(image_path)
```







# Sử dụng Histogram làm vector đặc trưng

```
"""Trích xuất histogram màu trong không gian RGB."""
```

```
def extract_color_histogram_rgb(image_path, bins=256):
```

```
    image = cv2.imread(image_path)    # Đọc ảnh
```

```
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # Chuyển từ BGR sang RGB
```

```
    # Tính histogram và chuẩn hóa thành feature vector
```

```
    feature_vector = []
```

```
    for i in range(3): # 3 kênh màu R, G, B
```

```
        hist = cv2.calcHist([image], [i], None, [bins], [0, 256])
```

```
        hist = cv2.normalize(hist, hist).flatten() # Chuẩn hóa và làm phẳng
```

```
        feature_vector.extend(hist)
```

```
    # Chuẩn hóa và làm phẳng
```

```
    feature_vector = np.array(feature_vector)
```

```
    feature_vector = cv2.normalize(feature_vector, feature_vector).flatten()
```

```
    return feature_vector # Kích thước 256+256+256 = 768
```



# Sử dụng Histogram làm vector đặc trưng

```
def extract_color_moments(image_path):  
    """Tính toán moment màu (mean, variance, skewness) trên kênh HSV."""  
    image = cv2.imread(image_path) # Đọc ảnh  
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)  
    mean = np.mean(hsv, axis=(0, 1))  
    std = np.std(hsv, axis=(0, 1))  
    skewness = np.mean(((hsv - mean) / (std + 1e-6)) ** 3, axis=(0, 1))  
    return np.concatenate([mean, std, skewness])
```



# Histogram màu

---

- **Ưu điểm:** Đơn giản, dễ tính toán
- **Nhược điểm:** không biểu thị được thông tin không gian (vị trí) của màu sắc trong ảnh.



# Đặc trưng moment màu

---

## Color Moments

[https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/AV0405/KEEN/av\\_as2\\_nkeen.pdf](https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV0405/KEEN/av_as2_nkeen.pdf)



# Moment màu

---

- Là một phương pháp mô tả phân phối màu sắc bằng cách sử dụng các giá trị thống kê:
  - Trung bình (mean),
  - Độ lệch chuẩn (standard deviation),
  - Độ nghiêng (skewness).

# Ý nghĩa của từng đặc trưng:

- **Trung bình (mean):** Thể hiện màu trung bình trong ảnh.
- Giá trị trung bình của kênh màu thứ  $i$  trong ảnh được tính như sau:

$$E_i = \sum_N^{j=1} \frac{1}{N} p_{ij}$$

$p_{ij}$  là giá trị màu của pixel thứ  $j$  trong kênh màu  $i$ .  
 $N$  là tổng số pixel

# Ý nghĩa của từng đặc trưng:

- **Độ lệch chuẩn (standard deviation):** Độ lan truyền của các giá trị màu xung quanh giá trị trung bình.

$$\sigma_i = \sqrt{\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^2}$$

$p_{ij}$  là giá trị màu của pixel thứ  $j$  trong kênh màu  $i$ .  
 $N$  là tổng số pixel

# Ý nghĩa của từng đặc trưng:

- **Độ nghiêng (skewness):** Độ bất đối xứng của phân phối màu

$$s_i = \sqrt[3]{\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^3}$$

$p_{ij}$  là giá trị màu của pixel thứ  $j$  trong kênh màu  $i$ .  
 $N$  là tổng số pixel





# Rút trích đặc trưng Moment màu

```
def extract_color_moments(image_path):  
    """Tính toán moment màu (mean, variance, skewness) trên kênh HSV."""  
    image = cv2.imread(image_path) # Đọc ảnh  
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)  
    mean = np.mean(hsv, axis=(0, 1))  
    std = np.std(hsv, axis=(0, 1))  
    skewness = np.mean(((hsv - mean) / (std + 1e-6)) ** 3, axis=(0, 1))  
    return np.concatenate([mean, std, skewness])
```

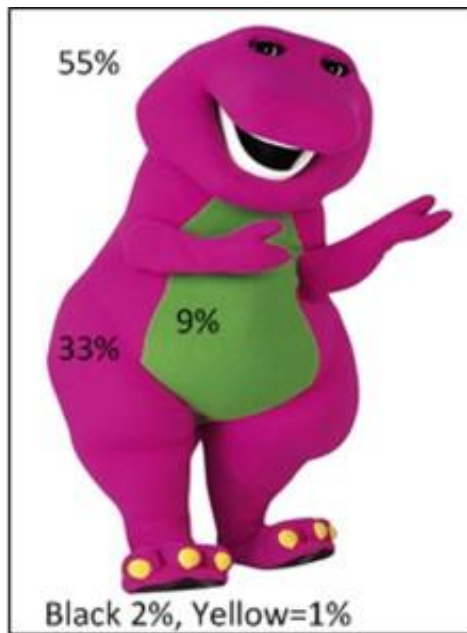


# Màu sắc chính của ảnh

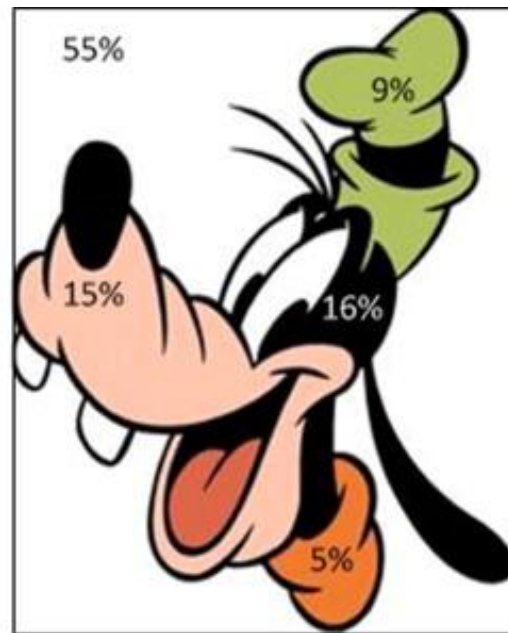
**Dominant Color Descriptor (DCD)**

# Màu sắc chính của ảnh

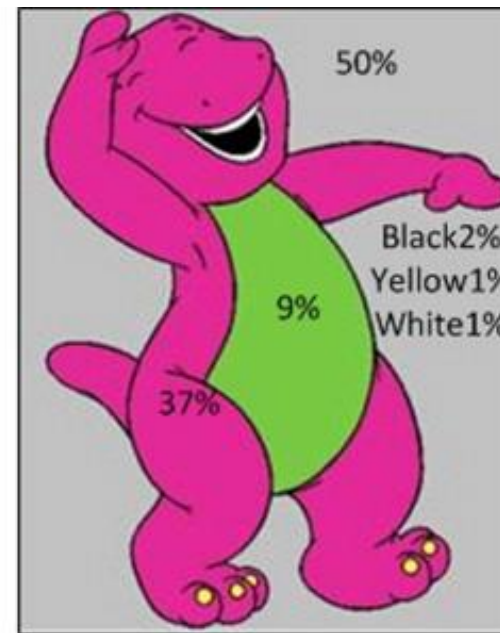
- Mô tả các **màu sắc nổi bật nhất** (chủ đạo)



(A) Image 1



(B) Image 2

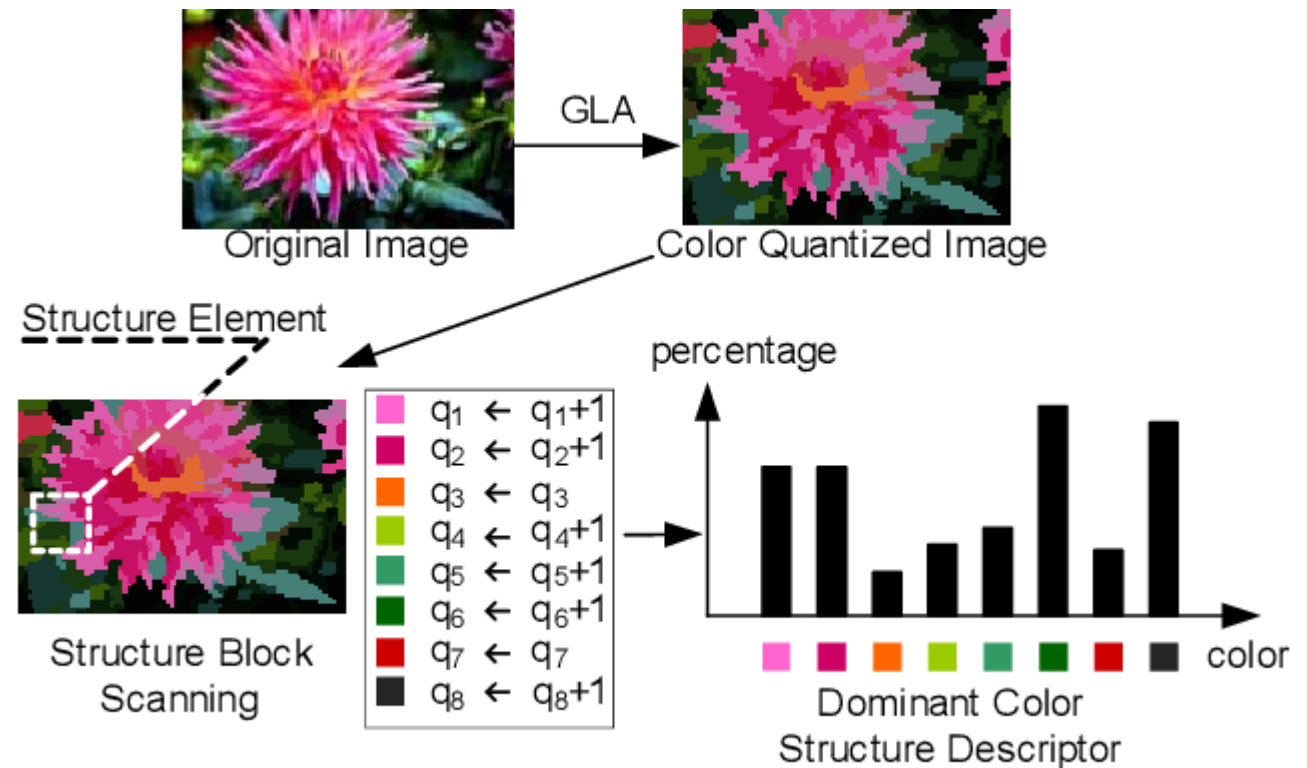


(C) Image 3

[A weighted dominant color descriptor for content-based image retrieval - ScienceDirect](#)

# Màu sắc chính của ảnh

- **Phương pháp xác định:** Dựa trên việc chia nhỏ các vùng **màu chủ đạo** và xác định **tỷ lệ xuất hiện** của chúng.



[\[PDF\] Dominant Color Structure Descriptor for Image Retrieval | Semantic Scholar](#)

**Fig. 1 Dominant Color Structure Descriptor extraction**



# Code trích xuất đặc trưng

```
def extract_dominant_color(image_path, k=3):  
    """Trích xuất màu sắc chủ đạo bằng K-means clustering."""  
    image = cv2.imread(image_path) # Đọc ảnh  
    pixels = image.reshape(-1, 3)  
    kmeans = cv2.KMEANS_RANDOM_CENTERS  
    _, labels, centers = cv2.kmeans(np.float32(pixels), k, None,  
                                    (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0),  
                                    10, kmeans)  
    hist = np.bincount(labels.flatten(), minlength=k) / labels.size  
    return np.hstack([centers.flatten(), hist])
```





# Color Coherence Vector (CCV)



# Color Coherence Vector (CCV)

- Biểu diễn mối quan hệ giữa các màu sắc
- CCV không chỉ lưu trữ thông tin về màu sắc mà còn lưu trữ các thông tin về tính liên tục của các vùng màu trong ảnh. CCV giúp phân biệt được các đối tượng có màu sắc giống nhau nhưng có phân bố khác nhau trong không gian ảnh.



# Color Coherence Vector (CCV)

- **Ưu điểm:** CCV tốt hơn Color Histogram ở chỗ nó có thêm thông tin về độ gắn kết không gian, giúp nhận diện đối tượng chính xác hơn.



# Tham khảo thêm

---

- <https://www.pinecone.io/learn/series/image-search/color-histograms/>
- <https://www.youtube.com/watch?app=desktop&v=I3na13AESjw>
- <https://slideplayer.com/slide/4904745/>