Bài giảng môn học:

Thị giác máy tính (7080518)

CHƯƠNG 1:
TỔNG QUAN VỀ THỊ GIÁC MÁY TÍNH
(Phần 2)

Đặng Văn Nam

dangvannam@humg.edu.vn

Nội dung chương 1 – Phần 2

2.1 Thu thập và biểu diễn ảnh trong máy tính

2.2 Các loại ảnh cơ bản

2.3 Không gian màu và Cách chuyển đổi

2.4 Bài tập thực hành

Phần 2: Biểu diễn ảnh trong máy tính

- **Học xong module này người học có thể:**
  - Trả lời được câu hỏi: pixel – điểm ảnh là gì? Đọc và hiển thị các thông số của ảnh.
  - Nắm được 3 loại ảnh cơ bản: Ảnh nhị phân – Ảnh xám – Ảnh màu
  - Hiểu được Hệ màu (Không gian màu) là gì?
  - Nắm được một số hệ màu cơ bản và cách chuyển đổi các hệ màu trong OpenCV

1. Biểu diễn ảnh trong máy tính

Thu nhận và biểu diễn ảnh trong máy tính

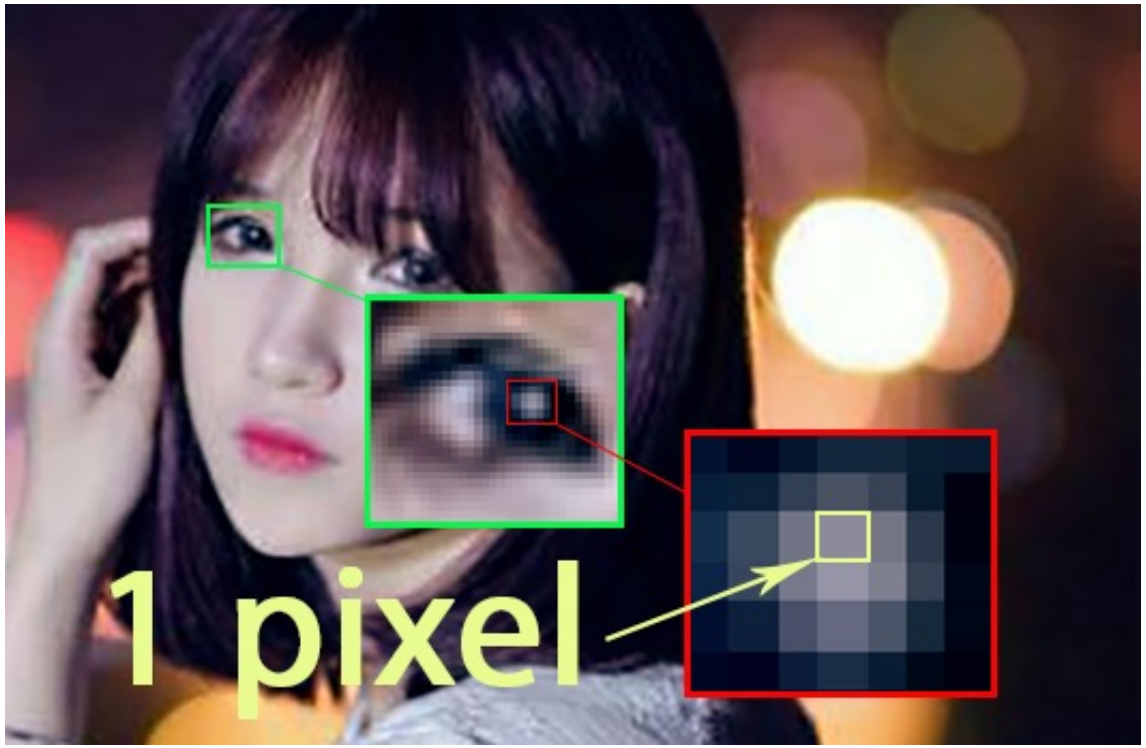
Zoom x 1000

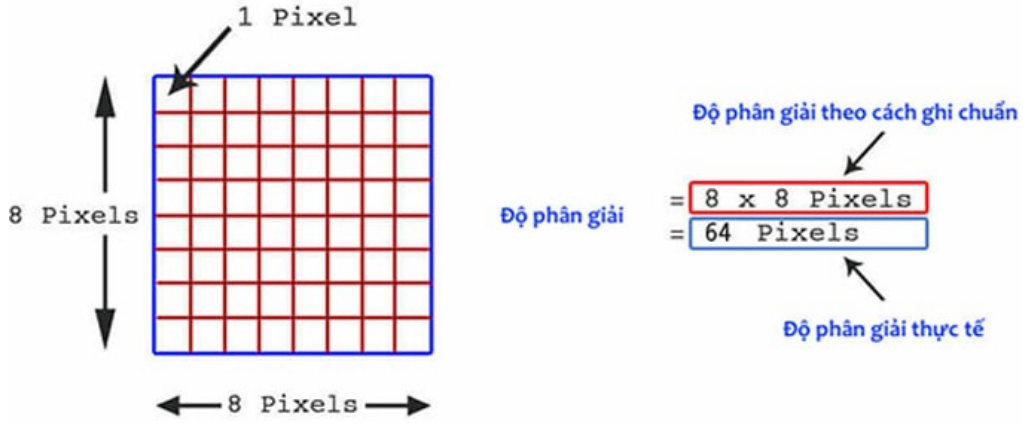
Điểm ảnh (Pixel)

Ảnh được xem là một tập hợp các điểm ảnh, mỗi điểm ảnh là đặc trưng cường độ sáng trong không gian ảnh.

Thu nhận và biểu diễn ảnh trong máy tính

Pixel (Picture Element) – có nghĩa là một điểm vật lý trên bức ảnh. Nó là đơn vị cơ bản nhất để tạo nên 1 bức ảnh mà chúng ta có thể nhìn thấy hàng ngày.





Độ phân giải ảnh càng cao, số lượng pixel điểm ảnh càng lớn → Ảnh càng chi tiết, sắc nét.

Thu nhận và biểu diễn ảnh trong máy tính

Con người nhìn thấy!

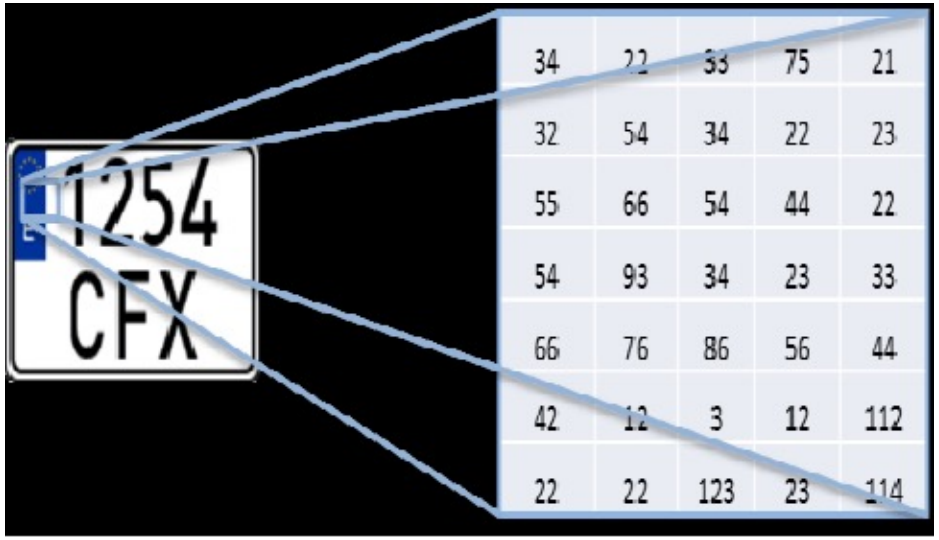
Máy tính nhìn thấy!

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

Mỗi ảnh là một ma trận, với một pixel là 1 giá trị trong ma trận.

Biểu diễn ảnh trong máy tính

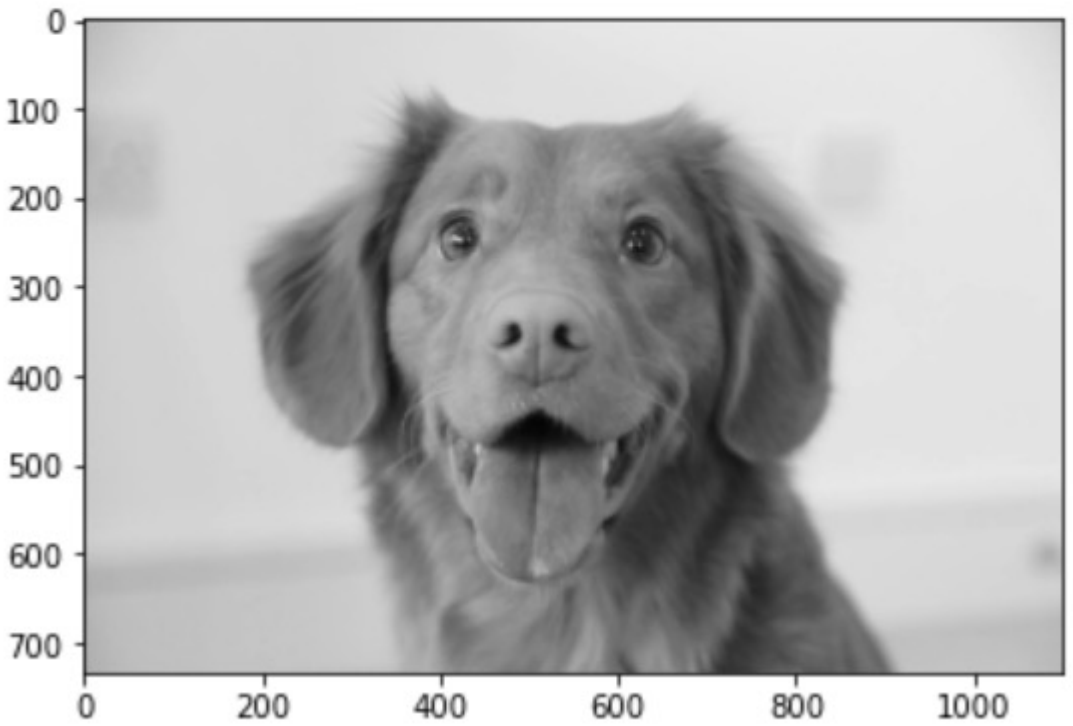
Mỗi ảnh là một ma trận, với một pixel là 1 giá trị trong ma trận.



Với ma trận có thể cộng trừ nhân chia, hoán vị,... để biến đổi ma trận theo ý muốn. Và từ ma trận có thể chuyển đổi thành vector và ngược lại (đại số tuyến tính và hình học).

Biểu diễn ảnh trong máy tính

```
1  #Đọc ảnh xám (gray)
2  img_gray = cv2.imread('images/5.jpg',0)
3  plt.imshow(img_gray,cmap='gray')
4  plt.show()
```




```
1  #Ảnh có kiểu dữ liệu là ndarray (mảng nhiều chiều)
2  type(img_gray)
```

```
numpy.ndarray
```

```
1  #Kích thước ảnh
2  img_gray.shape
```

```
(734, 1100)
```

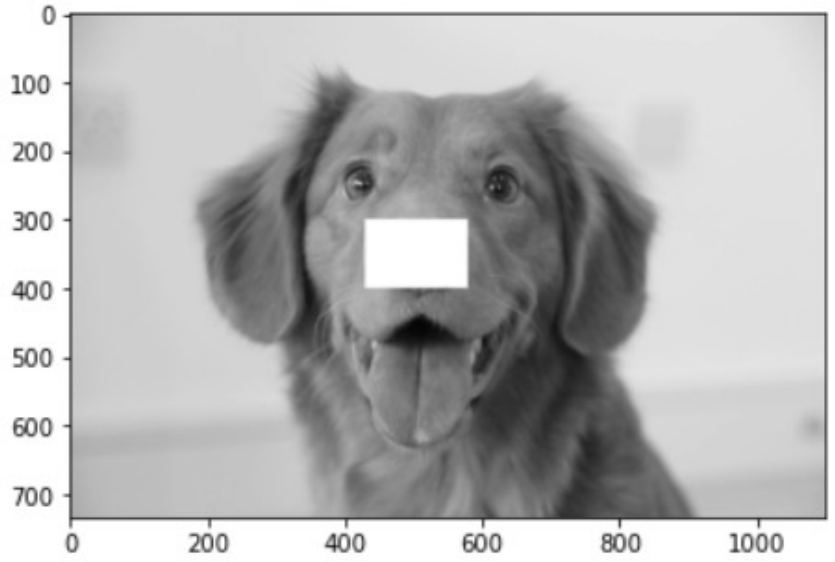
```
1  #Hiển thị ảnh ở dạng ma trận điểm ảnh
2  img_gray
```

```
array([[177, 178, 179, ..., 205, 205, 205],
       [177, 178, 179, ..., 205, 205, 205],
       [178, 178, 179, ..., 205, 205, 205],
       ...,
       [171, 173, 174, ..., 183, 183, 183],
       [171, 172, 174, ..., 183, 183, 183],
       [170, 172, 173, ..., 183, 183, 183]], dtype=uint8)
```

Biểu diễn ảnh trong máy tính

OpenCV chủ yếu xử lý trên các ảnh và video, mà ảnh và video lại được lưu dưới dạng các ma trận số có thể xử lý bằng thư viện numpy. Do đó việc thành thạo 2 thư viện này là yêu cầu bắt buộc.

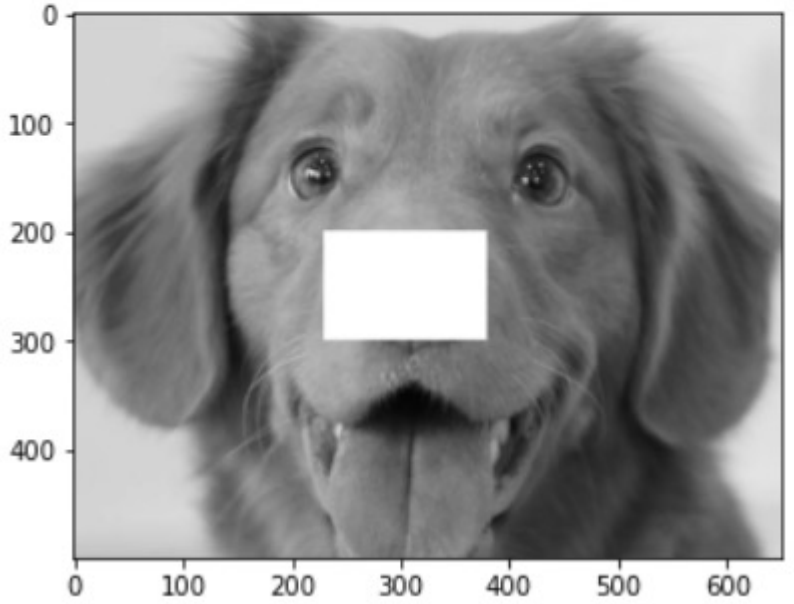
```python
#Thay đổi các giá trị trong ma trận
#--> Thay đổi màu sắc ảnh (0 đen --> 255 trắng)
img_gray[300:400,430:580] = 255

#Ảnh với các giá trị đã thay đổi
plt.imshow(img_gray,cmap='gray')
plt.show()
```



OpenCV

NumPy

```python
#Lấy một phần của ma trận
#--> tạo ra một ảnh mới
img1 = img_gray[100:600,200:850]

#Ảnh mới lấy một phần của ảnh ban đầu
plt.imshow(img1, cmap='gray')
plt.show()
```

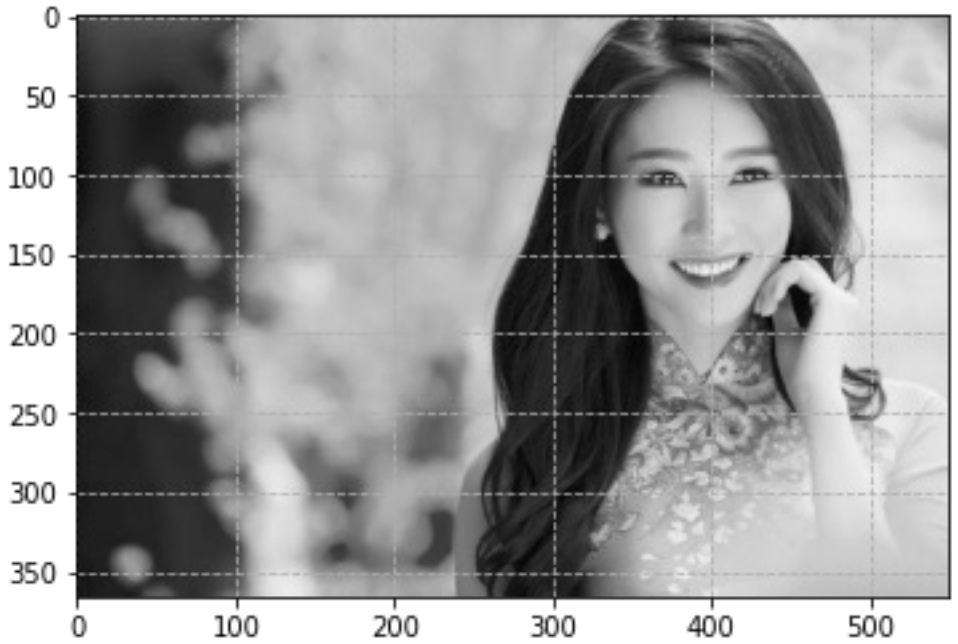


Thực hành số 1.4

Thực hành 1.4

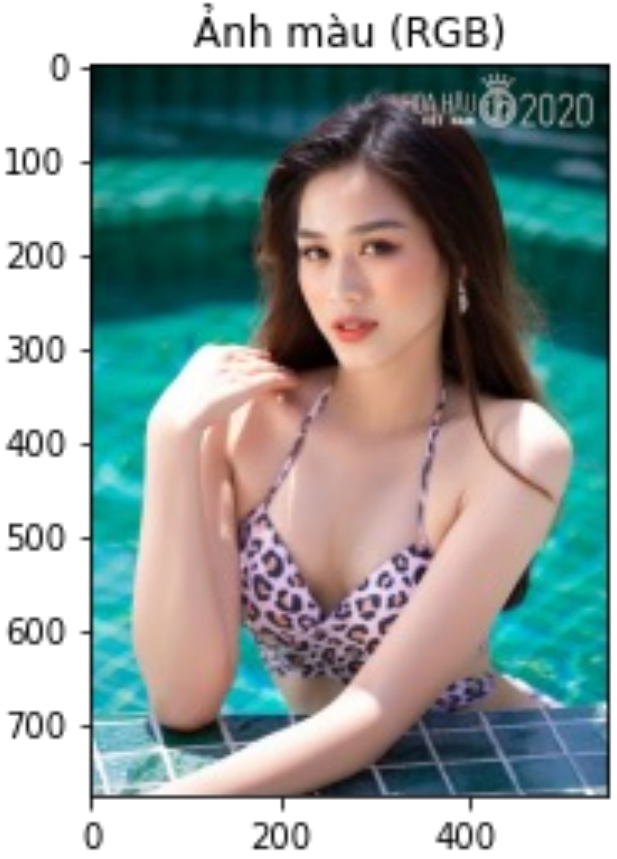
Yêu cầu:

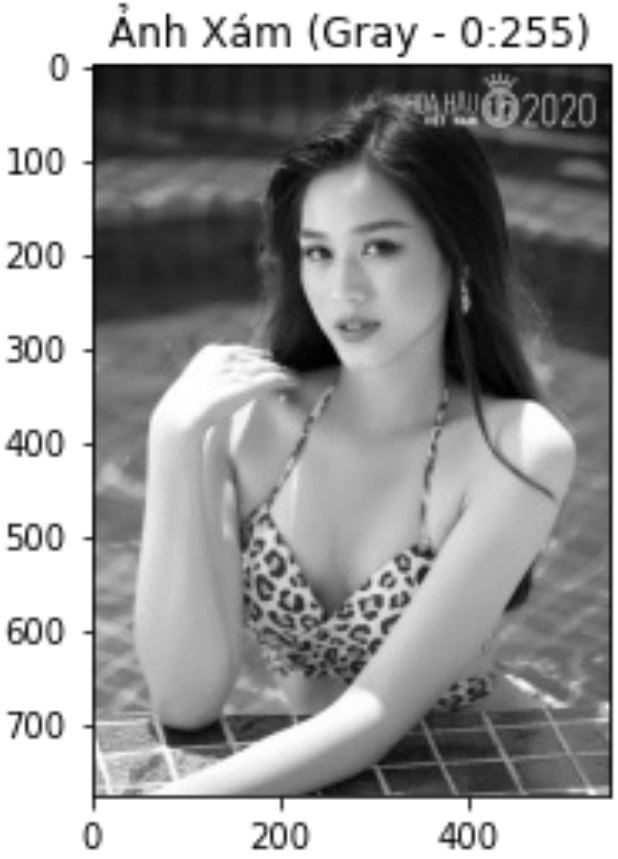
1. Học viên đọc ảnh images/Thuchanh_1.jpg dạng và hiển thị ảnh
2. Cho biết kích thước của ảnh và hiển thị ma trận pixel điểm ảnh
3. Tách lấy ảnh đôi mắt của cô gái và ghi ảnh đó vào thư mục images/Saves đặt tên file MSV_eyes.png

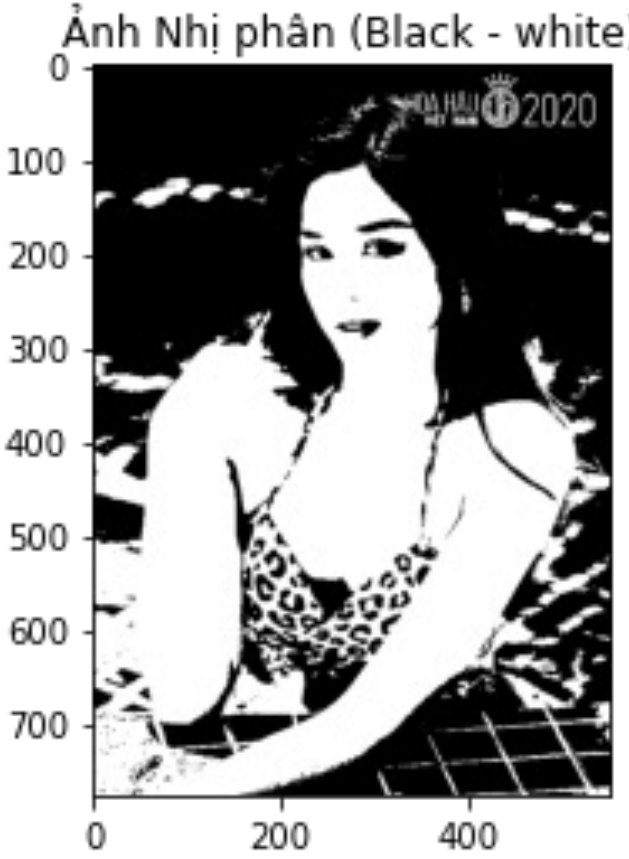


2. Các loại ảnh phổ biến

Các loại ảnh phổ biến

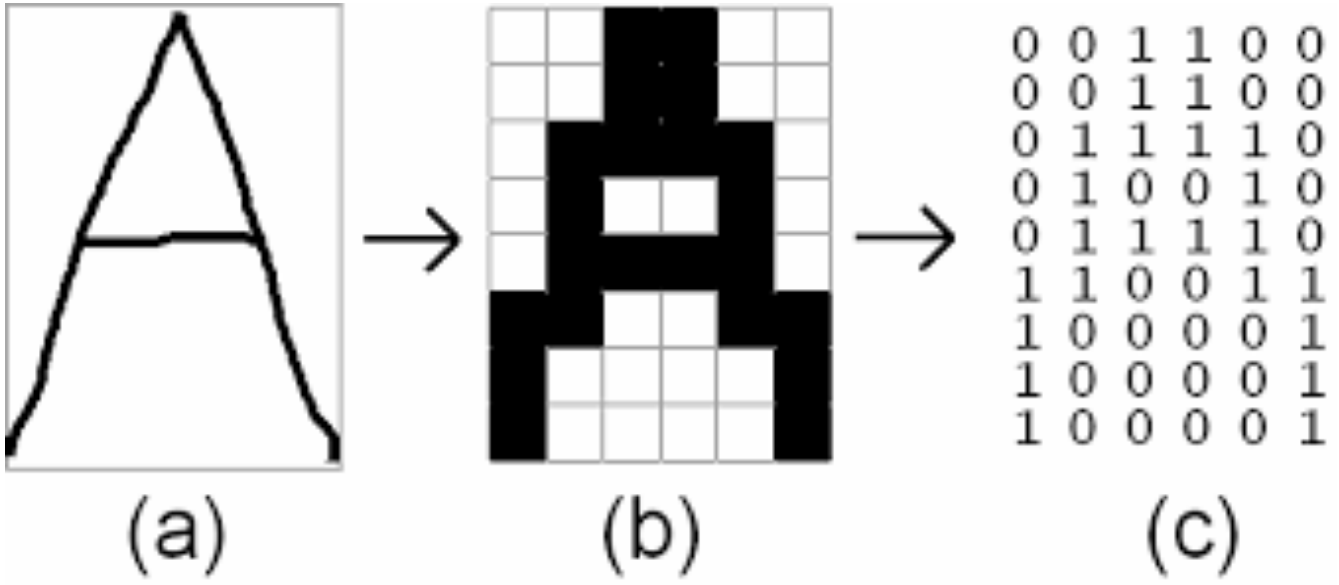


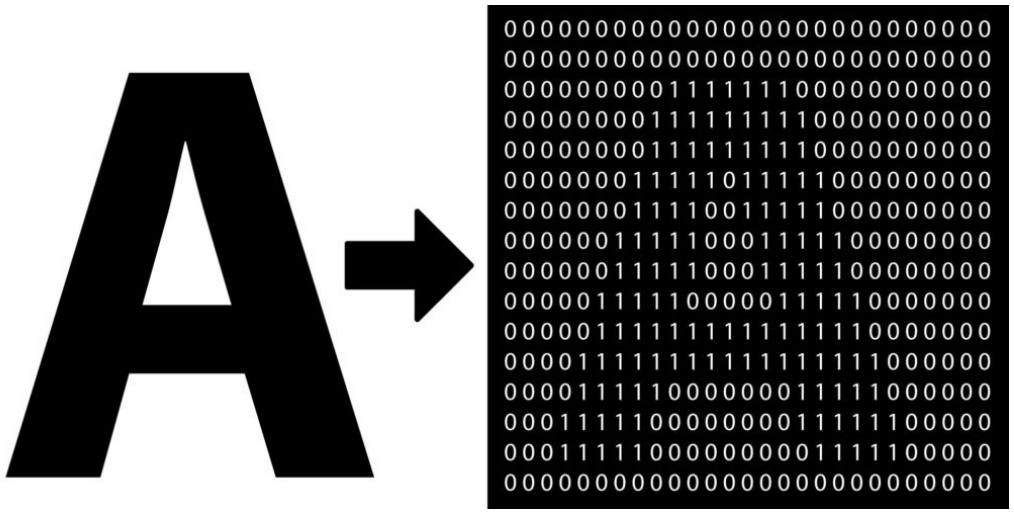




Ảnh nhị phân (Ảnh đen trắng)

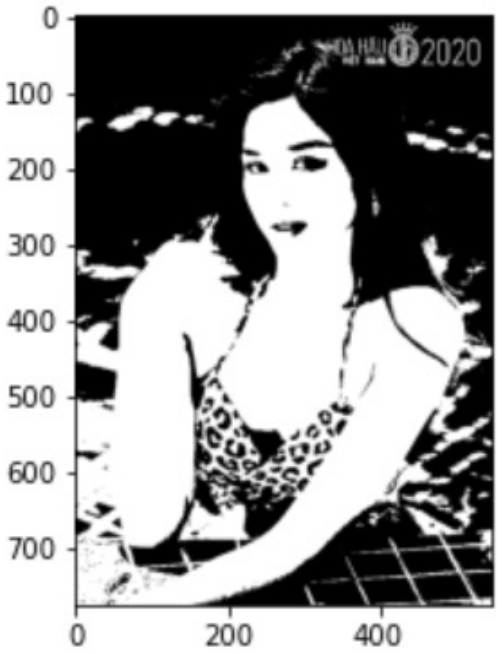
Ảnh nhị phân (trắng đen): mỗi điểm ảnh chỉ mang 1 trong 2 giá trị là: 0 - black hoặc 255- white. (0 – Black | 1- White)





Ảnh nhị phân (Ảnh đen trắng)

```
1  #Đọc ảnh nhị phân:
2  img_gray = cv2.imread('images/pic_binary.png',0)
3
4  #Hiển thị ảnh:
5  plt.imshow(img_binary,cmap='gray')
6  plt.show()
```




```
1  #Ảnh trắng đen: mỗi điểm ảnh chỉ mang 2 giá trị là
2  #0: black hoặc 255: white.
3  print('Kích thước ảnh:',img_binary.shape)
4  img_binary
```

```
Kích thước ảnh: (776, 550)

array([[  0,   0,   0, ...,   0,   0,   0],
       [  0,   0,   0, ...,   0,   0,   0],
       [  0,   0,   0, ...,   0,   0,   0],
       ...,
       [255, 255, 255, ..., 255, 255, 255],
       [255, 255, 255, ..., 255, 255, 255],
       [255, 255, 255, ..., 255, 255, 255]], dtype=uint8)
```

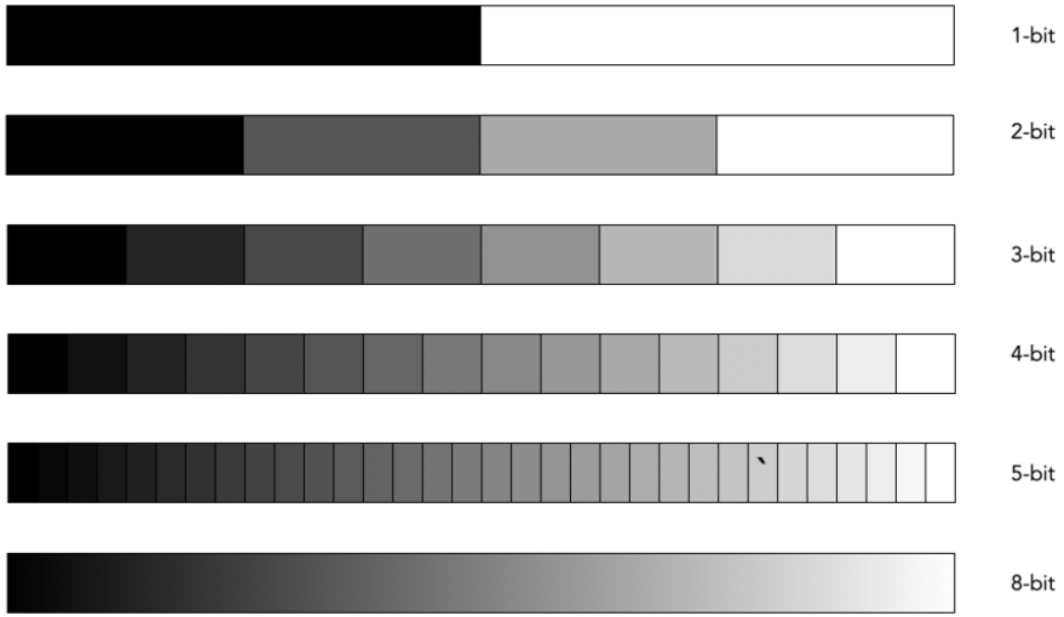
```
1  #Số pixel điểm ảnh
2  img_binary.size
```

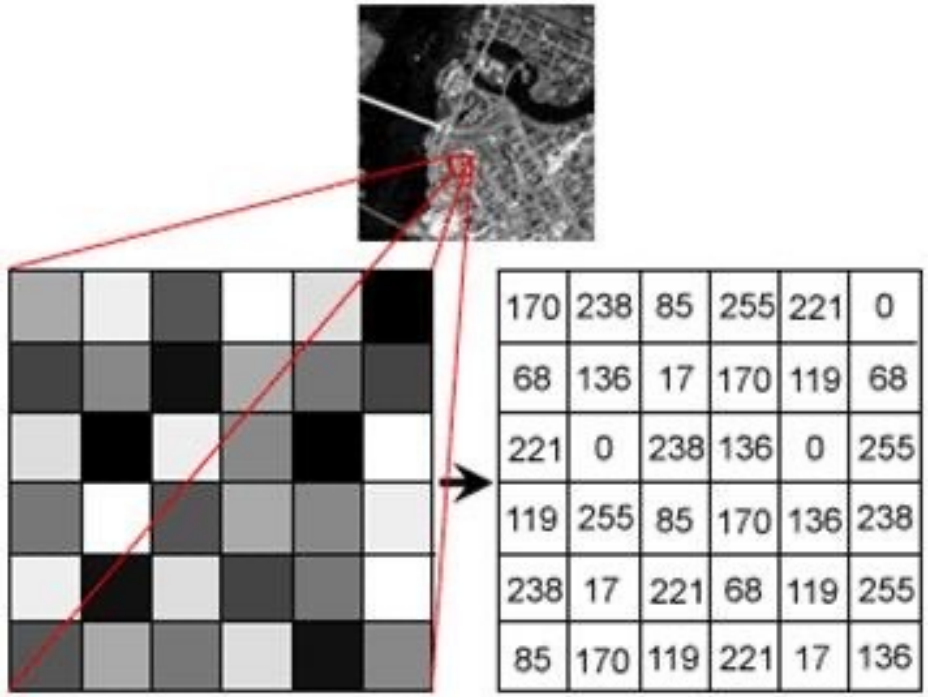
```
426800
```

Ảnh xám (gray)

Ảnh xám: mỗi điểm ảnh có 1 giá trị trong khoảng [0-n], n tùy vào độ sâu màu. Ảnh này chỉ đem lại cảm nhận về hình dạng vật thể chứ không mô tả được màu sắc.

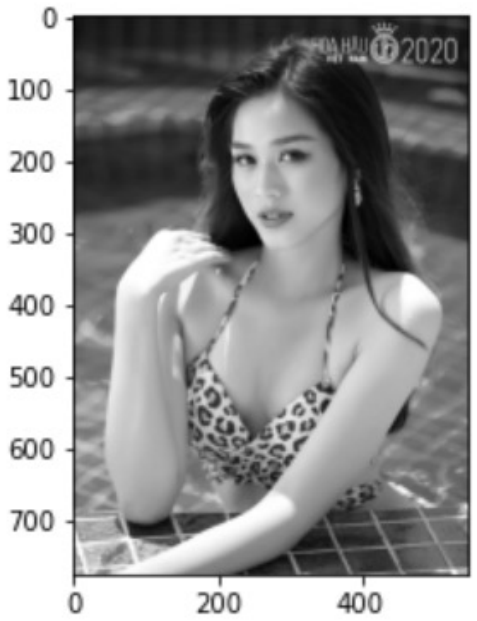
- Còn gọi là ảnh đơn sắc (monochromatic). Ảnh xám 3 bits mỗi điểm ảnh sẽ có giá trị nằm trong đoạn [0-7], ảnh mức xám 8 bits mỗi điểm ảnh sẽ có giá trị nằm trong đoạn [0-255].
- Giá trị của điểm ảnh bằng 0 đại diện cho điểm ảnh tối (đen), giá trị điểm ảnh lớn nhất đại diện cho điểm ảnh sáng (trắng).





Ảnh xám (gray)

```
#Đọc ảnh xám
img_gray = cv2.imread('images/pic_gray.png',0)
#Hiển thị ảnh xám
plt.imshow(img_gray,cmap='gray')
plt.show()
```




```
print('Kích thước ảnh:',img_gray.shape)
img_gray
```

```
Kích thước ảnh: (776, 550)

array([[ 22,  22,  22, ...,  20,  20,  20],
       [ 21,  21,  21, ...,  19,  18,  18],
       [ 22,  22,  22, ...,  19,  19,  18],
       ...,
       [195, 185, 179, ..., 154, 157, 143],
       [176, 187, 188, ..., 183, 164, 168],
       [192, 186, 179, ..., 201, 180, 179]], dtype=uint8)
```

```
#Số pixel điểm ảnh:
img_gray.size
```

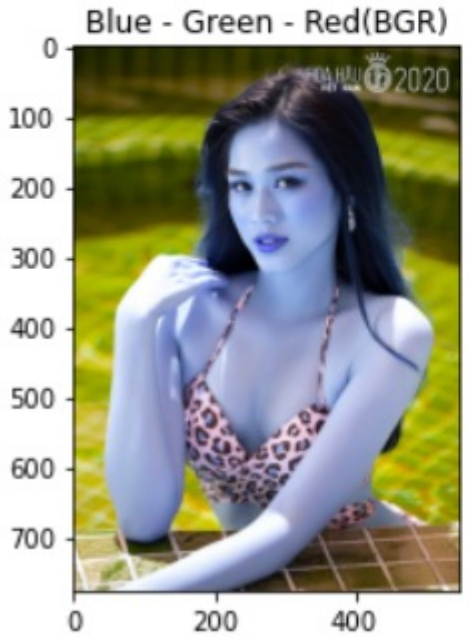
```
426800
```

Ảnh nhị phân và ảnh xám được tạo bởi 1 kênh màu, mỗi pixel được tạo bởi 1 màu duy nhất.

Ảnh màu

• **Ảnh màu:** mỗi điểm ảnh được tạo bởi m giá trị trong khoảng [0-n] riêng biệt tùy theo số kênh màu của ảnh. Ảnh 3 kênh RGB là phổ biến nhất.

```
#Đọc ảnh màu, OpenCV đọc ảnh theo thứ tự BGR:
img_color = cv2.imread('images/pic_0.jpg',1)
#Hiển thị ảnh màu BRG
plt.imshow(img_color)
plt.title('Blue - Green - Red(BGR)')
plt.show()
```




```
#Số chiều và pixel điểm ảnh:
img_color.shape
```

```
(776, 550, 3)
```

```
#pixel điểm ảnh đầu tiên:
print('pixcel đầu tiên (BGR):',img_color[0,0])

#pixel điểm ảnh cuối cùng:
print('pixcel cuối cùng (BGR):',img_color[-1,-1])
```

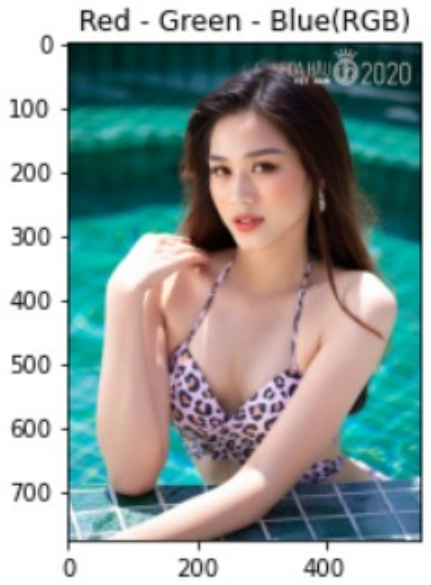
```
pixcel đầu tiên (BGR): [29 31  1]
pixcel cuối cùng (BGR): [206 188 151]
```

• **Mỗi một pixel điểm ảnh bao gồm 3 giá trị ứng với 3 kênh màu Blue - Green - Red.**

Ảnh màu

• OpenCV xử lý ảnh theo thứ tự kênh màu BGR blue - green - red, Matplotlib xử lý ảnh theo thứ tự kênh màu RGB red - green - blue. Cần convert ảnh sang RGB trước khi dùng matplotlib hiển thị.

```
#Chuyển đổi sang hệ màu RGB để hiển thị trên máy tính:
img_color_rgb = cv2.cvtColor(img_color,cv2.COLOR_BGR2RGB)
#Hiển thị ảnh màu RGB:
plt.imshow(img_color_rgb)
plt.title('Red - Green - Blue(RGB)')
plt.show()
```



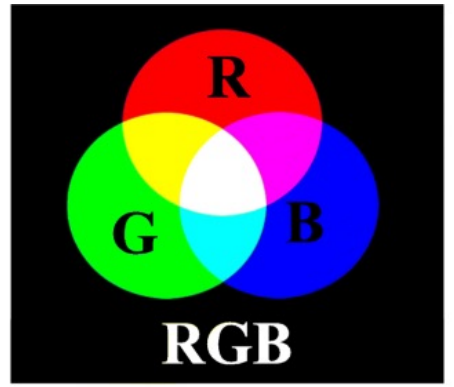

```
#Màu RGB:
#pixel điểm ảnh đầu tiên:
print('pixcel đầu tiên (RGB):',img_color_rgb[0,0])

#pixel điểm ảnh cuối cùng:
print('pixcel cuối cùng (RGB):',img_color_rgb[-1,-1])
```

```
pixcel đầu tiên (RGB): [ 1 31 29]
pixcel cuối cùng (RGB): [151 188 206]
```

3. Một số hệ màu cơ bản

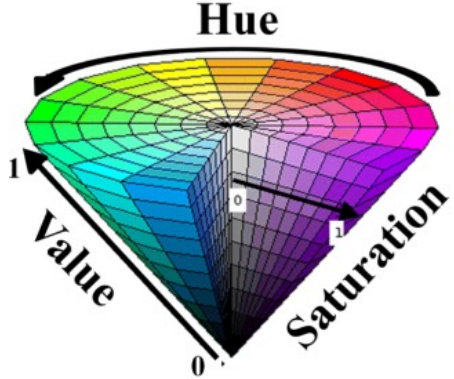
Một số hệ màu cơ bản

Hệ màu - Không gian màu (Colour Space): được hiểu là các mô hình toán để miêu tả màu sắc. Hệ thống không gian màu cho phép mỗi màu được xác định theo số học. Mỗi không gian màu đều có một tác dụng và ứng dụng trong các bài toán khác nhau.

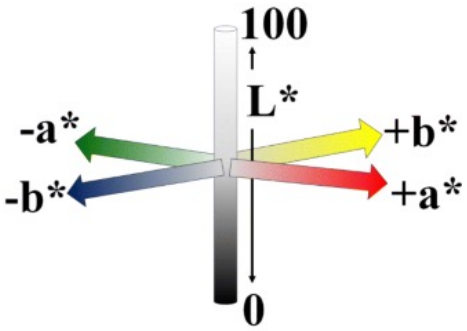


(a)

(b)



(c)

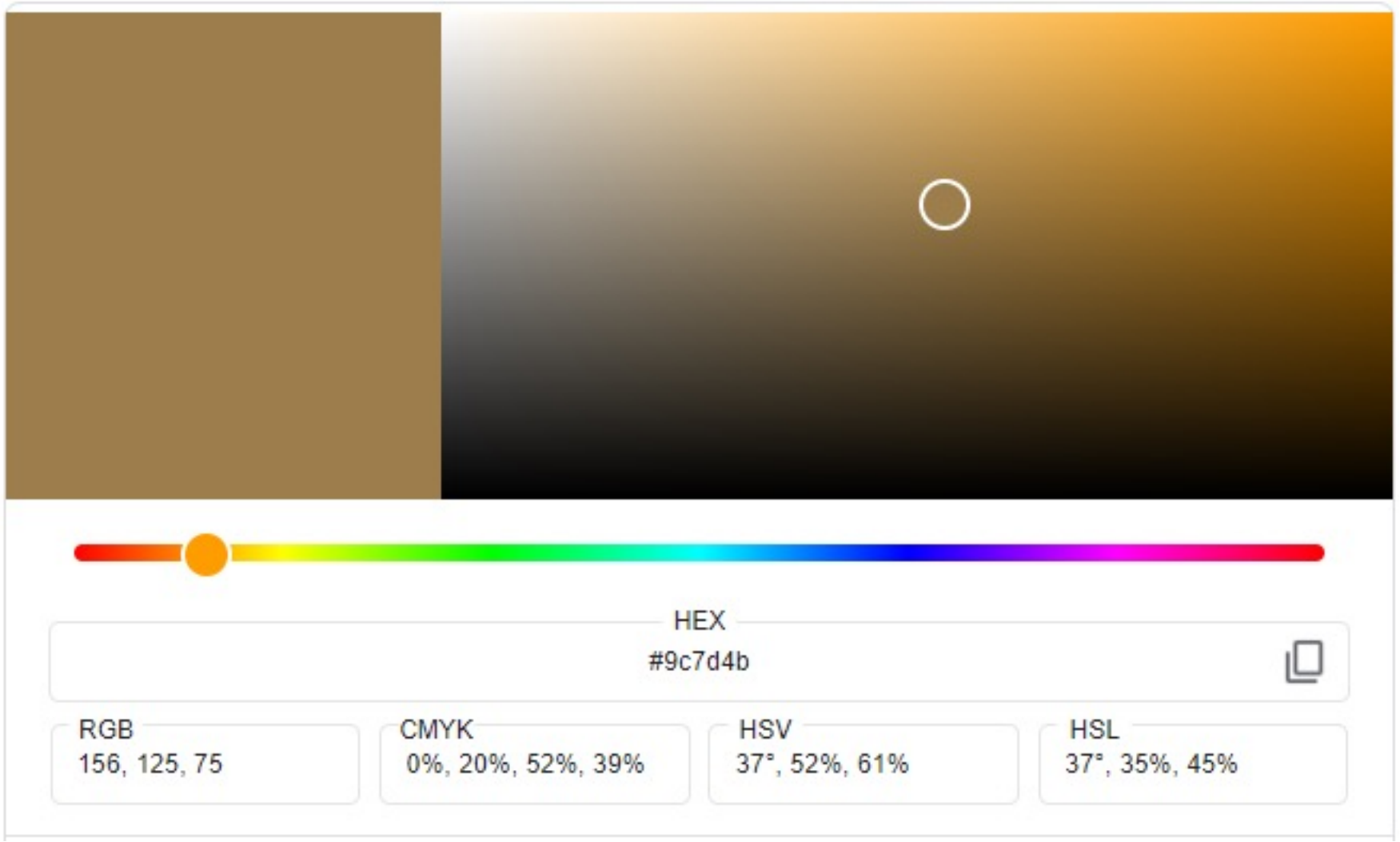


(d)

- **Hệ màu RGB (Red – Green – Blue)**
- **Hệ màu RGBA**
- **Hệ màu YUV**
- **Hệ màu CMYK**
- **Hệ màu HSV (Hue, Saturation, Value)**
- **Hệ màu HSL (Hue, Staturation, Luminance)**
- **…**

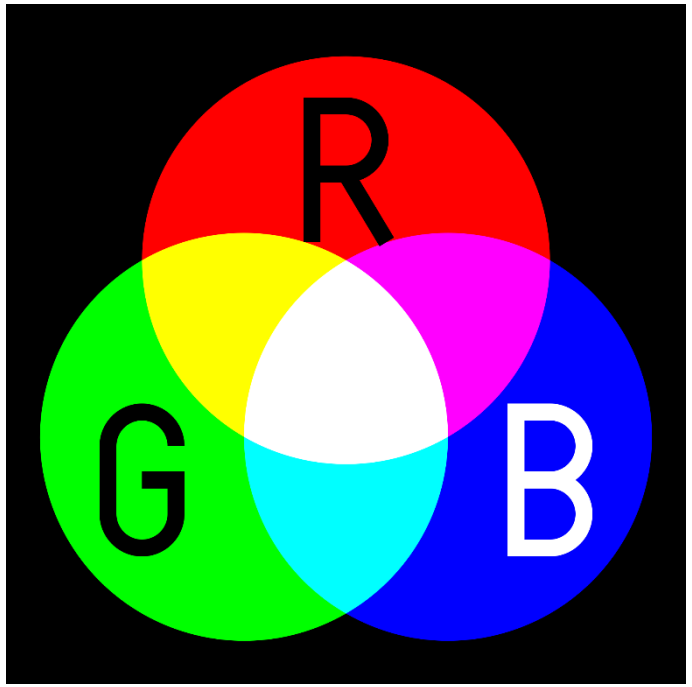
Một số hệ màu cơ bản

RGB color picker

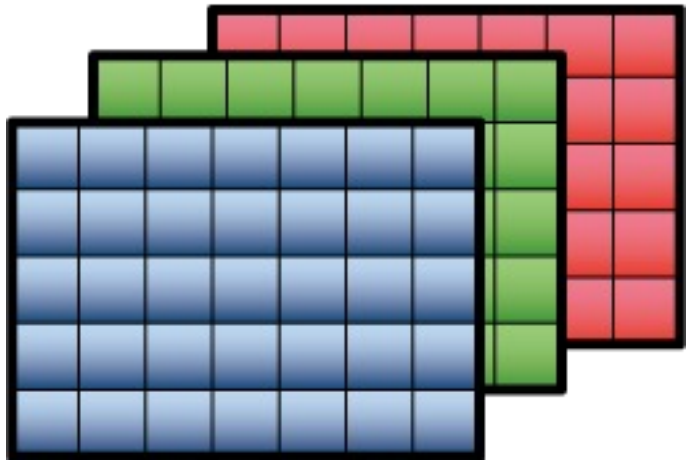


Hệ màu RGB

- RGB là không gian màu phổ biến dùng trong máy tính, máy ảnh, điện thoại và nhiều thiết bị kĩ thuật số khác nhau.
- Không gian màu này khá gần với cách mắt người tổng hợp màu sắc.
- Nguyên lý cơ bản là sử dụng 3 màu sắc cơ bản R (red - đỏ), G (green - xanh lục) và B (blue - xanh lam) để biểu diễn tất cả các màu sắc.

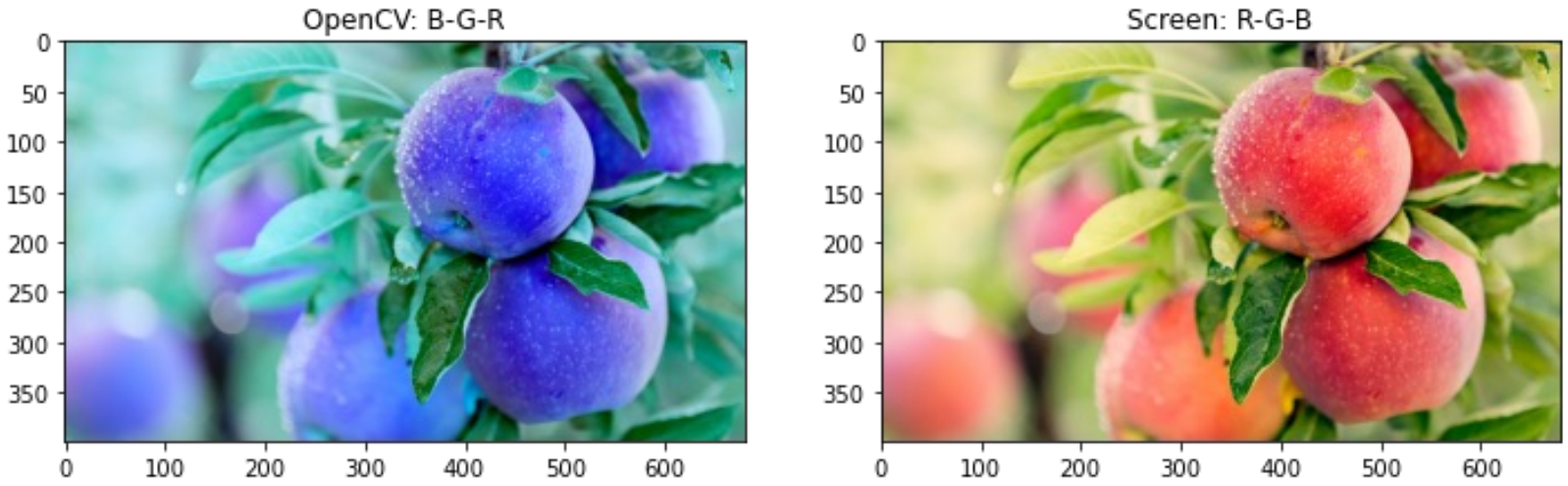


- Thông thường, trong mô hình 24 bit mỗi kênh màu sẽ sử dụng 8bit để biểu diễn, tức là giá trị R, G, B nằm trong khoảng 0 - 255. Bộ 3 số này biểu diễn cho từng điểm ảnh, mỗi số biểu diễn cho cường độ của một màu. Với mô hình màu 24bit thì số màu tối đa có thể tạo ra là *255 x 255 x 255 = 16581375* màu.



Hệ màu RGB

```python
#Đọc ảnh màu:
img_bgr = cv2.imread('images/apples.jpg')

# OpenCV lưu trữ ảnh theo thứ tự kênh màu B, G, R nên khi hiển thị lên màn hình ảnh sẽ khác
plt.figure(figsize=(12,6))
plt.subplot(1, 2, 1)
plt.imshow(img_bgr)
plt.title('OpenCV: B-G-R')

#Chuyển về từ hệ màu BGR ---> RGB và hiển thị
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

plt.subplot(1, 2, 2)
plt.imshow(img_rgb)
plt.title('Screen: R-G-B')
plt.show()
```



Hệ màu RGB

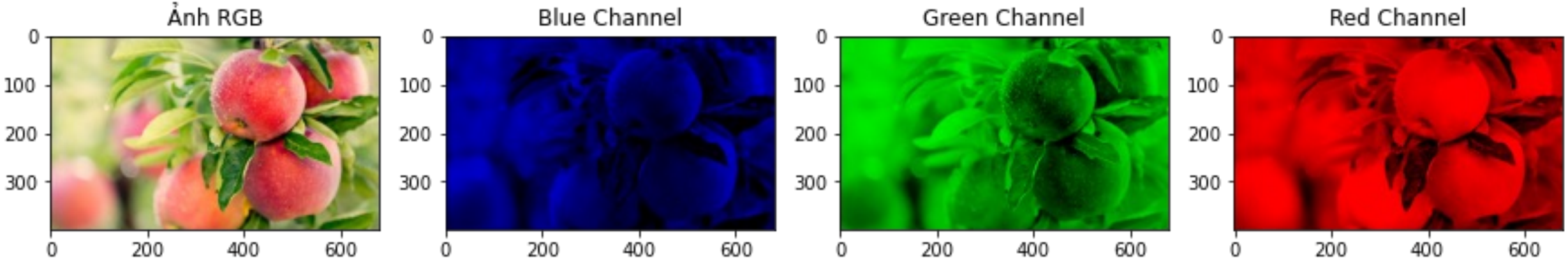
- **Tách các kênh màu RGB (Red – Green – Blue)**

```python
#Đọc ảnh màu:
img_bgr = cv2.imread('images/apples.jpg')

# CÁCH 1:
#Tách các kênh màu B, G, R từ ảnh màu
b_channel = None
g_channel = None
r_channel = None

#LẤY DỮ LIỆU CÁC KÊNH MÀU:
b_channel = img_bgr.copy()
b_channel[:, :, [1,2]] = 0

g_channel = img_bgr.copy()
g_channel[:, :, [0, 2]] = 0

r_channel = img_bgr.copy()
r_channel[:, :, [0, 1]] = 0
```

```python
#CÁCH 2: Tách màu BGR
#Lấy kích thước ảnh:
height = img_bgr.shape[0]
width = img_bgr.shape[1]
print('Chiều cao ảnh:',height)
print('Chiều rộng ảnh:',width)
#----------------------------------------
#Khai báo 2 biến chứa 3 kênh màu G-R-B:
blue=np.zeros((height,width,3),np.uint8)
green=np.zeros((height,width,3),np.uint8)
red=np.zeros((height,width,3),np.uint8)
#----------------------------------------
#Lấy màu từng pixel tương ứng với mỗi kênh:
for x in range(height):
    for y in range(width):
        B=img_bgr[x,y,0]
        G=img_bgr[x,y,1]
        R=img_bgr[x,y,2]

        blue[x,y,0]=B
        green[x,y,1]=G
        red[x,y,2]=R
```

Hệ màu RGB

- **Tách các kênh màu RGB (Red – Green – Blue)**



Thực hành số 1.5

Thực hành 1.5

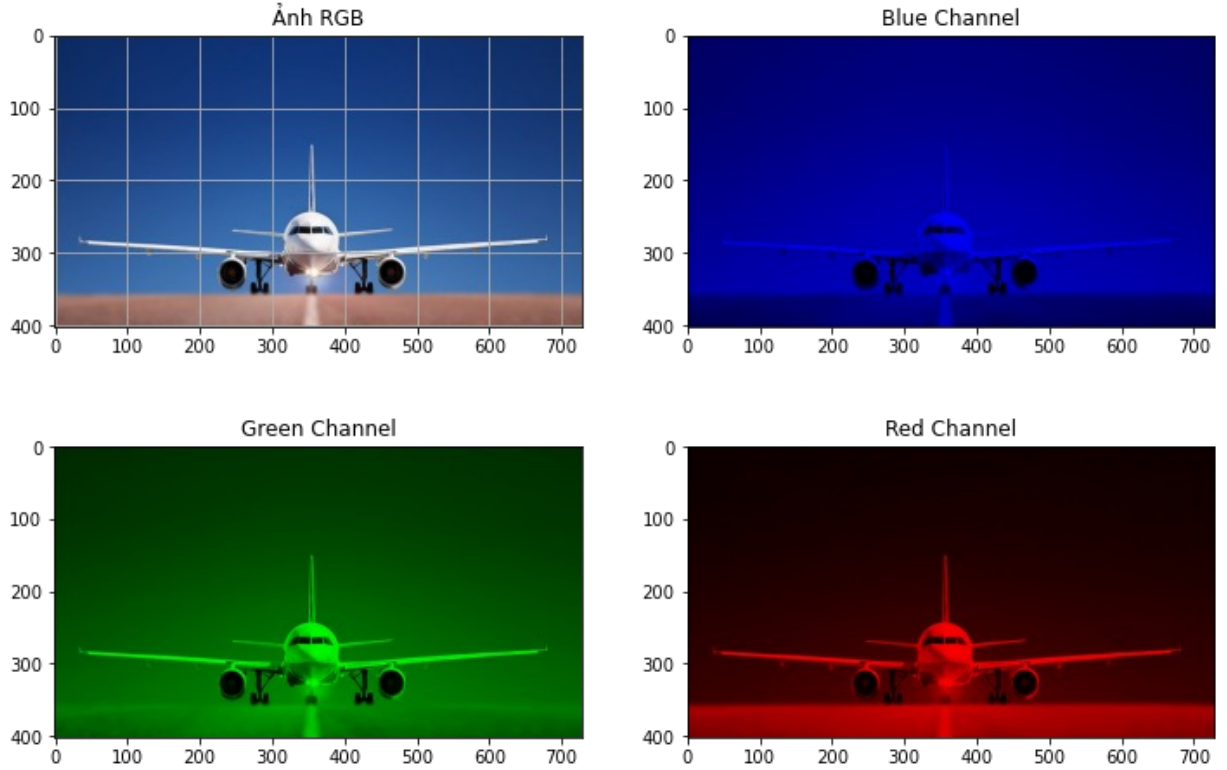
Yêu cầu:

1. Học viên đọc ảnh màu images/Thuchanh2.jpg

2. Cho biết chiều cao, chiều rộng và số pixel điểm ảnh.

3. Tách thành các kênh màu B-G-R tương ứng và hiển thị như yêu cầu trong slide

4. Ghi từng kênh màu thành các ảnh riêng vào thư mục images/Saves đặt tên như sau: MSV_blue.jpg; MSV_green.jpg; MSV_red.jpg



4. Chuyển đổi các hệ màu với OpenCV

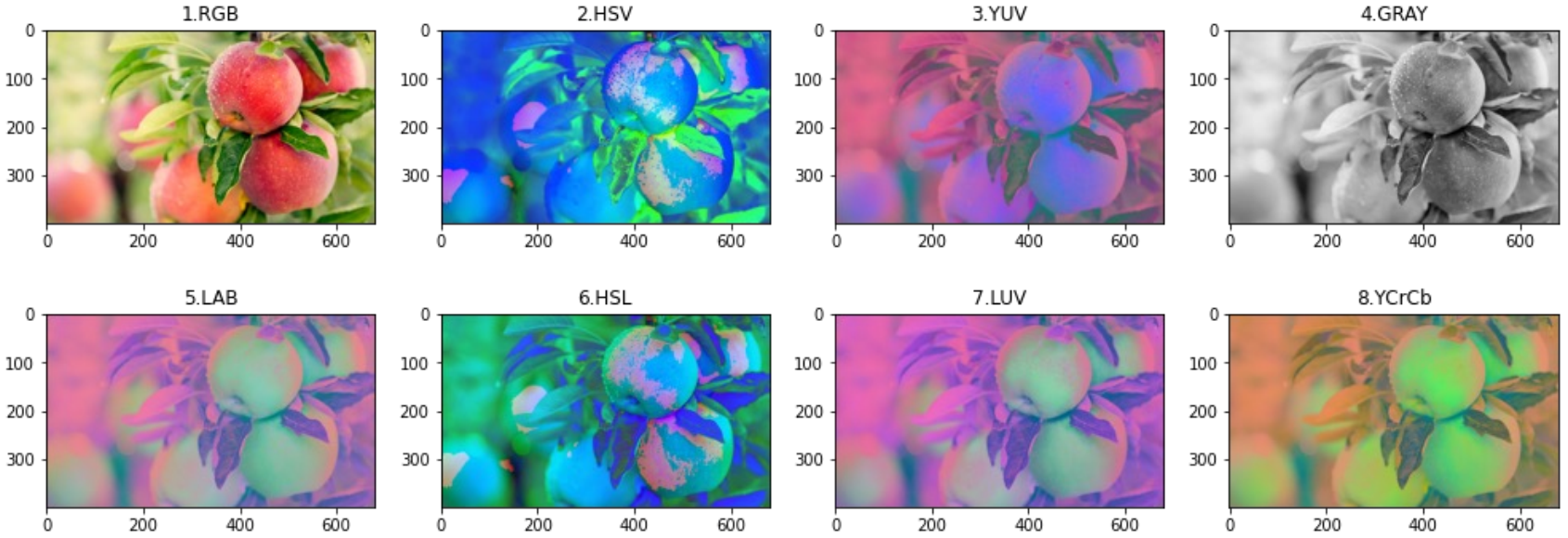
Chuyển đôi Hệ màu với OpenCV

Thay đổi kênh màu của ảnh sử dụng:

Hàm cv2.cvtColor(img,convert_colorspace)

Có 150 phương thức chuyển đổi không gian màu trong OpenCV, Phổ biến:

- COLOR_BGR2RGB
- COLOR_RGB2BGR
- COLOR_RGB2HSV
- COLOR_HSV2RGB
- COLOR_RGB2YUV
- COLOR_YUV2RGB
- COLOR_BGRA2RGBA
- COLOR_RGB2GRAY
- COLOR_BGR2GRAY
- COLOR_HSV2GRAY
- COLOR_FGBA2GRAY




QUESTIONS
Q&A
ANSWERS