

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**BÁO CÁO GIỮA KỲ
MÔN HỌC XỬ LÝ ẢNH**

Họ và tên:	Cao Việt Hoàng
Ngày sinh:	04/10/2001
Mã sinh viên:	19020819
Lớp học phần:	INT3404 1

Hà Nội, năm 2023.

Em dùng Google Colab (Colaboratory) để hoàn thành các tasks.

Task 1: Make “spot the difference” game data

Level 1: Thêm 1 vài quả bóng đặc biệt khác vào hình ảnh

1. Import packages cần thiết

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
```

2. Input một hình ảnh, và các hình ảnh muốn thêm vào

*3. Sử dụng phương thức **shape** trong thư viện **numpy** để trả về các kích thước của ảnh cần chèn*

```
rows,cols,channels=add_img.shape
```

4. Xác định vị trí cần chèn các ảnh

```
roi=img[50:(50+rows),50:(50+cols)]
```

5. Convert các ảnh cần chèn sang GRAY

```
add_Gray = cv2.cvtColor(add_img,cv2.COLOR_BGR2GRAY)
```

*6. Threshold các ảnh GRAY: sử dụng **THRESH_BINARY***

```
ret,mask = cv2.threshold(add_Gray,220,255,cv2.THRESH_BINARY)
```

7. Trích xuất bitwise các phần cần thiết (inverseMask, background, frontImage)

```
inverseMask1 = cv2.bitwise_not(mask1)
```

```
background1 = cv2.bitwise_and(roi1,roi1,mask=mask1)
```

```
frontImage1 = cv2.bitwise_and(add_img1,add_img1,mask=inverseMask1)
```

8. Thêm các phần đã trích xuất vào ảnh ban đầu đúng vị trí mong muốn

```
dest=cv2.add(background,frontImage)
```

```
img[50:(50+rows),50:(50+cols)]=dest
```

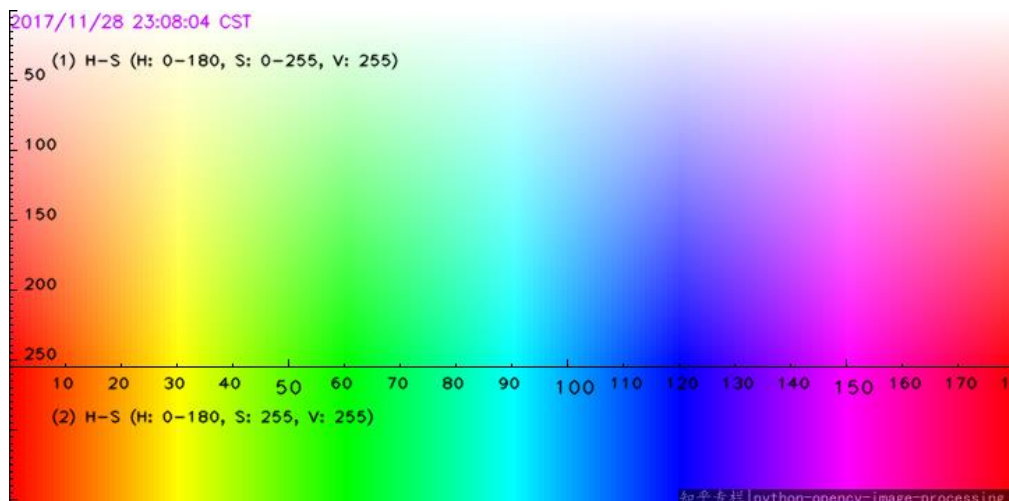
9. Output kết quả



Level 2: Thay đổi các quả bóng màu đỏ thành màu hồng

Thay đổi màu của một số đối tượng trong ảnh, sử dụng không gian màu của HSV (HSV Color Spaces)

HSV Color Spaces



1. Import các packages cần thiết

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
```

2. Input một hình ảnh

3. Chuyển đổi BGR sang HSV

```
hsv=cv2.cvtColor(image,cv2.COLOR_BGR2HSV)
```

4. Xác định phạm vi màu cần thay đổi

Phạm vi màu đỏ trong HSV:

```
red_lo=np.array([155,25,0])  
red_hi=np.array([179,255,255])  
mask=cv2.inRange(hsv,red_lo,red_hi)
```

5. Chuyển sang màu khác (hồng) trong HSV

```
image[mask>10]=(255, 155, 255)
```

6. Output kết quả



Level 3: Thay đổi các quả bóng màu xanh lá cây thành màu hồng nhũ (chính xác hơn Level 2), thêm 1 vài quả bóng đặc biệt vào ảnh

a. Thay đổi các quả bóng màu xanh lá cây thành màu hồng nhũ

Các bước tương tự Level 2, nhưng tính toán chi tiết hơn phạm vi màu, tính toán chuyển màu chi tiết hơn và trích xuất các phần quan trọng:

* Phạm vi màu xanh lá cây trong HSV:

```
green_lower = np.array([36,0,0])
```

```
green_upper = np.array([86,255,255])
mask = cv2.inRange(hsv, green_lower, green_upper)
inv_mask = cv2.bitwise_not(mask)
```

* Tính toán chuyển sang màu hồng nhũ trong HSV

```
h, s, v = cv2.split(hsv)
h = np.mod(h + 100, 180)
s = np.clip(s - 100, 0, 255)
v = np.clip(v, 0, 255)
hsv = cv2.merge([h, s, v])
```

* Trích xuất các phần cần thiết cho kết quả

```
result = cv2.bitwise_or(cv2.bitwise_and(img, img, mask=inv_mask), cv2.bitwise_and(bgr, bgr, mask=mask))
```

b. Thêm một vài quả bóng đặc biệt vào ảnh (Tương tự như Level1)

→ *Output kết quả:*



Task 2: Solve game Spot the differences

Các bước để phát hiện sự khác biệt của 2 hình ảnh:

1. Import các packages cần thiết

```
from skimage.metrics import structural_similarity
from google.colab.patches import cv2_imshow
import numpy as np
import imutils
import cv2
```

2. Input 2 hình ảnh khác nhau, convert 2 hình ảnh sang Gray Scale

```
image_one = cv2.imread("pic1.jpg")
image_two = cv2.imread("pic2.jpg")
gray1 = cv2.cvtColor(image_one, cv2.COLOR_BGR2GRAY)
gray2 = cv2.cvtColor(image_two, cv2.COLOR_BGR2GRAY)
```

3. Tính toán Structural Similarity Index Measurement (SSIM) giữa hai hình ảnh, trả về hình ảnh khác biệt

```
(score, diff) = structural_similarity(gray1, gray2, full=True)
diff = (diff*255).astype("uint8")
```

4. Threshold

```
thresh=cv2.threshold(diff, 0, 255, cv2.THRESH_BINARY_INV|cv2.THRESH_OTSU) [1]
```

5. Sử dụng findContours để tìm các khu vực

```
cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)
```

6. Nếu các contours được tìm thấy:

Tính toán bounding của contour; xác định tọa độ tâm, bán kính; sau đó vẽ vòng tròn để chỉ ra sự khác biệt.

```
for c in cnts:
    (x,y,w,h)=cv2.boundingRect(c)
    x1=int(x+w/2)
    y1=int(y+h/2)
    r=int(1/2*np.sqrt(w*w+h*h))
```



```
if cv2.contourArea(c) > 50:
    cv2.circle(image_two, (x1, y1), r, (0, 0, 255), 2)
```

7. Hiện thị các hình ảnh output

```
cv2.imshow(image_one)
cv2.imshow(image_two)
```

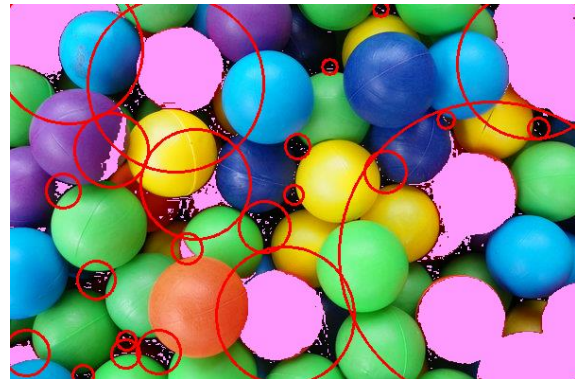
Origin:



Level 1



Level 2



Level 3

