



**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-  
TPHCM  
KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO**

**Môn: Toán ứng dụng và thống kê cho công nghệ  
thông tin**

**Chủ đề:** Image processing.

**Họ tên:** Kuo Yung Sheng

**MSSV:** 21127684

**Lớp:** 21CLC07

*Thành phố Hồ Chí Minh, ngày 19 tháng 6 năm 202*

## 1. Các chứng năng hoàn thành

Chức năng	Hoàn thành
Thay đổi độ sáng cho ảnh	100%
Chuyển đổi ảnh RGB thành ảnh xám/sepia	100%
Lật ảnh (ngang - dọc)	100%
Làm mờ	100%
Làm sắc nét	100%
Cắt ảnh theo kích thước	100%
Cắt ảnh theo khung tròn	100%

## 2. Mô tả các hàm.

def increaseBrightness(image, amount): hàm tăng độ sáng.

Ý tưởng: mỗi điểm ảnh RGB + thêm số lượng amount, nếu >255 thì = 255.

def changeContrast(image, contrast\_factor): hàm thay đổi độ tương phản

Ý tưởng:

1. Nếu contrast\_factor = 0 thì trả về ảnh luôn khỏi xử lý
2. Ép kiểu dữ liệu ảnh về float
3. Tính theo công thức điều chỉnh tương phản:  $(\text{image\_normalized} - 128.0) * \text{contrast\_factor} + 128.0$
4. Ép về khoảng dữ liệu [0,255] bằng hàm np.clip
5. Chuyển ảnh trở lại dữ liệu uint8 và trả ảnh.

def scroll(image, type=0): hàm lật ngang/dọc.

Ý tưởng:

1. Tạo array image có kích thước như array image ban đầu.
2. Nếu type là 0 xoay ảnh theo chiều dọc, các vị trí điểm ảnh được gán RGB tại vị trí điểm ảnh đối xứng.
3. Nếu type là 1 thì xoay ảnh theo chiều ngang, các vị trí điểm ảnh được gán RGB với vị trí theo tọa độ chiều cao không đổi, tọa độ theo chiều ngang đối xứng

def convert\_to\_gray(image): hàm chuyển ảnh sang màu xám.

Ý tưởng:

Tại mỗi điểm ảnh nhân giá trị RGB với hệ số như công thức sau

$$\text{tr} = 0.393 * R + 0.769 * G + 0.189 * B$$

$$\text{tg} = 0.349 * R + 0.686 * G + 0.168 * B$$

$$\text{tb} = 0.272 * R + 0.534 * G + 0.131 * B$$

def blur(image): hàm làm mờ

Ý tưởng:

1. Tạo array image có kích thước như array image ban đầu với kiểu dữ liệu float.
2. Tại mỗi vị trí điểm ảnh gốc sẽ chia cho 9 và + giá trị đó cho các điểm ảnh xung quanh (lấy giá trị array image gốc để thực hiện cho array image copy)
3. Ép về kiểu dữ liệu uint8. Và trả về ảnh

def sharpen(image): hàm làm rõ ảnh

Ý tưởng:

1. Tại vị trí điểm ảnh sẽ bằng giá trị tại vị trí đó \*5 và trừ đi giá trị tại trái phải trên dưới của nó.
2. Kiểm tra xem có giá trị nào ngoài range [0, 255] không, nếu có thì ép nó về trong khoảng đó bằng cách  $>255$  thì  $= 255$ ,  $<0$  thì  $= 0$ .
3. Ép về kiểu dữ liệu uint8 và trả về ảnh.

def cut(image,size): hàm cắt

Ý tưởng:

1. Tính vị trí trung tâm của ảnh bằng cách lấy, dài chia 2, rộng chia 2.
2. Tìm vị trí bắt đầu cắt của ảnh bằng cách lấy tọa độ tâm của ảnh trừ đi chiều dài chiều rộng cần cắt.
3. Cắt tại vị trí bắt đầu cắt đến vị trí tọa độ bắt đầu + chiều dài chiều rộng cần cắt.

def circel(image): hàm cắt khung tròn

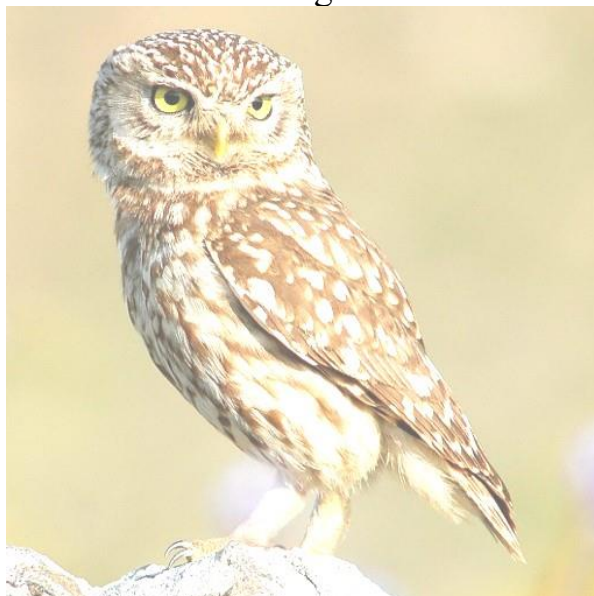
Ý tưởng:

Xét từng tọa độ trên điểm ảnh, nếu tọa độ  $x^2 + y^2 > (\text{chiều dài ảnh} / 2)^2$  (công thức bán kính hình tròn) thì gán cho vị trí đó RGB [0,0,0].

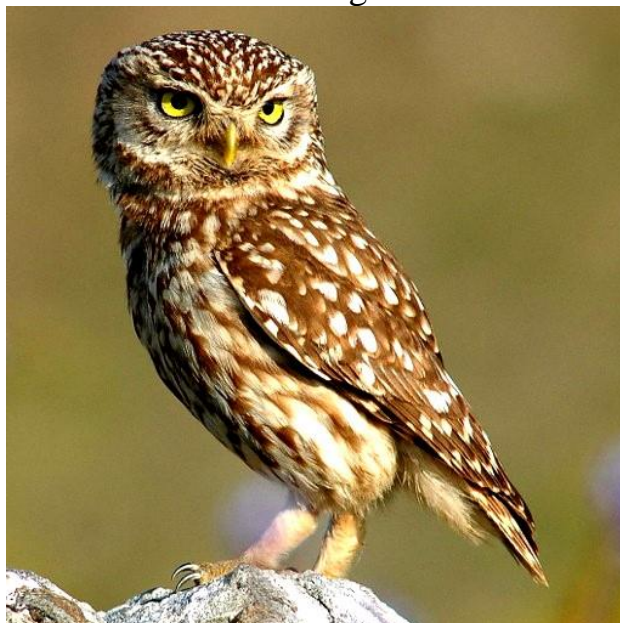
### 3. Kết quả



Ảnh gốc



Increase brightness



Contrast



Gray



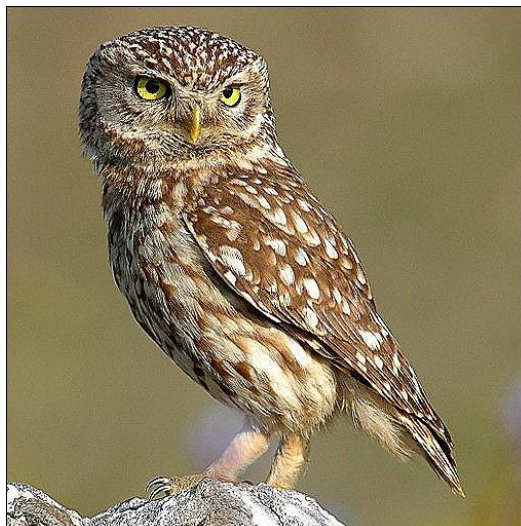
Rotate dọc



Rotate ngang



Blur





Sharp



Cut



Circle

#### **4. Tài liệu tham khảo**

[https://en.wikipedia.org/wiki/Kernel\\_\(image\\_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))