

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**UIT**  
TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

# **BÁO CÁO ĐỒ ÁN**

**MÔN HỌC: XỬ LÝ ẢNH VÀ ỨNG DỤNG**

**ĐỀ TÀI**

## **CÁC PHƯƠNG PHÁP TĂNG CƯỜNG DỮ LIỆU**

**Giảng viên hướng dẫn:** Mai Tiến Dũng

**Sinh viên thực hiện:** Tô Thanh Hiền - 19521490

Trần Vĩ Hào - 19521482

Lê Đặng Đăng Huy - 19521612

**Lớp:**

CS406.N11.KHCL

Thành phố Hồ Chí Minh, ngày 30 tháng 12 năm 2022

## MỤC LỤC

I. GIỚI THIỆU CHUNG .....	4
1. Tăng cường dữ liệu là gì? .....	4
2. Tính ứng dụng .....	4
3. Tính phụ thuộc dữ liệu .....	5
4. Sự đa dạng của augmentation .....	5
5. Cách augmentation tốt nhất? .....	5
II. CÁC PHƯƠNG PHÁP TĂNG CƯỜNG DỮ LIỆU .....	6
1. Noise Injection .....	6
a. Gaussian Noise .....	6
b. Salt and Pepper Noise .....	6
c. Speckle Noise .....	7
2. Geometric Transformation .....	8
a. Scaling .....	8
b. Flipping .....	8
c. Rotation .....	9
d. Translation .....	10
e. Shearing .....	10
3. Color Space Transformation .....	11
a. Solarize .....	11
b. Posterize .....	11
c. Color .....	12
d. Contrast .....	12
e. Brightness .....	13
f. Sharpness .....	13
4. Cut Out .....	14
5. Mixing Images .....	15
a. MixUp .....	15
b. CutMix .....	15
6. RandAugment .....	16
a. N: Number of transformation .....	17
b. M: Magnitude of Transformation .....	17
III. THỰC NGHIỆM .....	18
1. Dataset .....	18
2. Extract feature .....	18
3. Model .....	18
4. Evaluate .....	18

a. Vertical translation .....	19
b. Horizontal translation .....	19
c. Rotation .....	19
d. Shearing .....	20
e. Scaling .....	20
f. Flip .....	21
g. Noise injection .....	21
h. Color space transformation .....	21
i. Các phương pháp khác .....	22
j. Combine các phương pháp .....	22
5. Kết luận .....	23
IV. DEMO .....	24
V. TÀI LIỆU THAM KHẢO .....	25

# I. GIỚI THIỆU CHUNG

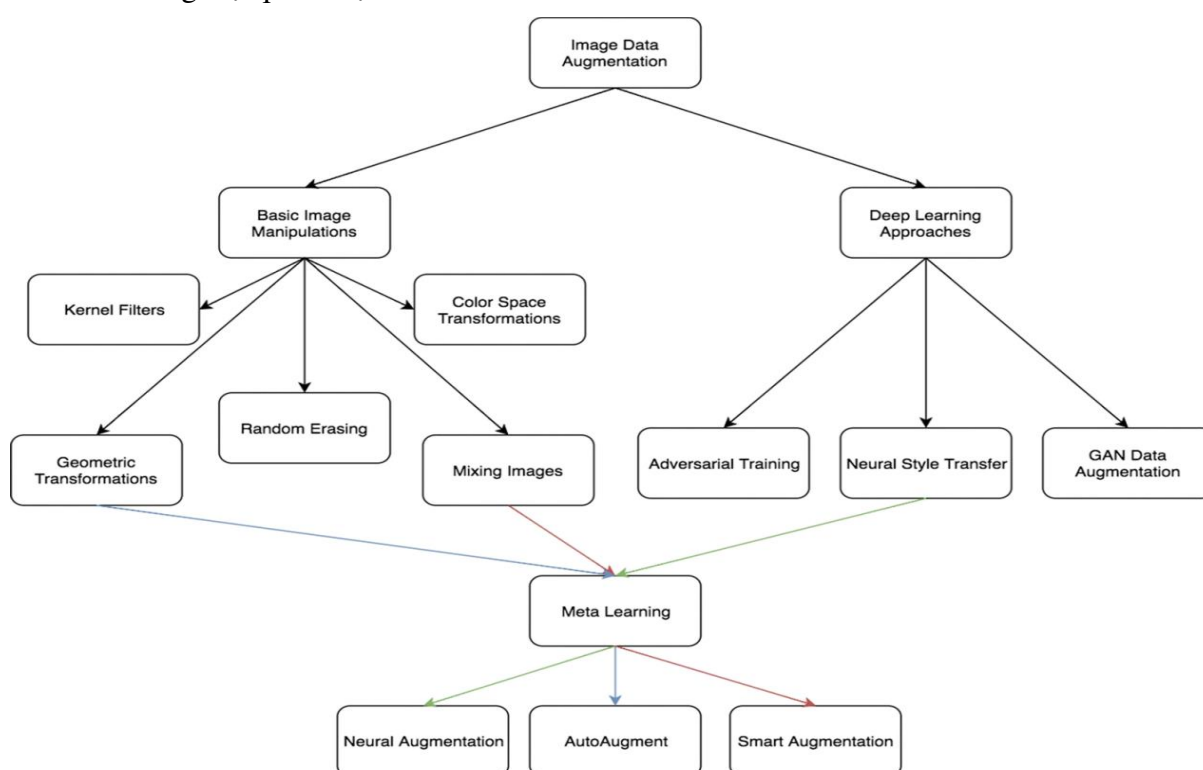
## 1. Tăng cường dữ liệu là gì?

Tăng cường dữ liệu là quá trình phát triển dữ liệu mới và đại diện từ dữ liệu hiện có. Chúng ta có thể thực hiện điều này bằng cách bao gồm các phiên bản đã sửa đổi của dữ liệu hiện có hoặc tổng hợp dữ liệu mới.

Bộ dữ liệu được tạo ra bằng phương pháp này sẽ cải thiện mô hình học máy hoặc học sâu của chúng ta bằng cách giảm thiểu nguy cơ trạng bị quá mức. Đây là quá trình thay đổi, hoặc "bổ sung", một tập dữ liệu với thông tin bổ sung.

Đầu vào bổ sung này có thể bao gồm từ hình ảnh đến văn bản và nó tăng cường hiệu suất của hệ thống học máy.

Giả sử chúng ta muốn xây dựng một mô hình để phân loại các giống chó và chúng ta có một số lượng lớn các bức ảnh về tất cả các giống ngoại trừ chó con. Kết quả là, mô hình sẽ gặp khó khăn trong việc phân loại các con chó con.



## 2. Tính ứng dụng

Ứng dụng cho học máy đang phát triển nhanh chóng và đa dạng, đặc biệt là trong lĩnh vực học sâu. Những thách thức mà ngành trí tuệ nhân tạo phải đối mặt có thể được khắc phục thông qua các kỹ thuật tăng cường dữ liệu.

Tăng cường dữ liệu có thể cải thiện hiệu suất và kết quả của các mô hình học máy bằng cách thêm các ví dụ mới và đa dạng vào tập dữ liệu đào tạo.

Khi tập dữ liệu đủ lớn và đủ, mô hình học máy hoạt động tốt hơn và chính xác hơn. Đối với các mô hình học máy, việc thu thập dữ liệu và ghi nhãn có thể tốn nhiều thời gian và tốn kém.

Các công ty có thể giảm chi phí hoạt động của họ bằng cách thay đổi bộ dữ liệu và sử dụng các chiến lược tăng cường dữ liệu.

Làm sạch dữ liệu là một trong những giai đoạn trong quá trình phát triển mô hình dữ liệu và nó là điều cần thiết đối với các mô hình có độ chính xác cao. Tuy nhiên, mô hình sẽ không thể dự đoán đầu vào phù hợp từ thế giới thực nếu việc làm sạch dữ liệu làm giảm khả năng đại diện.

Mô hình học máy có thể được củng cố bằng cách sử dụng các phương pháp tiếp cận nâng cao dữ liệu, tạo ra các phương sai mà mô hình có thể gặp phải trong thế giới thực.

### 3. Tính phụ thuộc dữ liệu

Vấn đề là “con vịt nào cũng béo vật lông con nào”, quá nhiều cách thức augmentation, chọn cách nào để cho chất lượng tốt nhất đây? Câu trả lời là - tùy thuộc vào dữ liệu (số lượng mẫu, tính balance/imbalance của mẫu, dữ liệu test, v.v. và ứng dụng tương ứng. Nghĩa là mỗi bộ dữ liệu sẽ có cách thức riêng để augmentation sao cho ra kết quả tốt nhất.

Điển hình là dữ liệu MNIST được cho là tốt với phương pháp elastic distortion, scale, translation, và rotation. Trong khi dữ liệu ảnh tự nhiên như CIFAR10 và ImageNet thì lại tốt với thuật toán random-cropping, image mirroring, color shifting/whitening [8]. Không chỉ có vậy mà một số augmentation method không tốt cho một số tập dữ liệu. Đơn cử là horizontal flipping tốt cho CIFAR10 nhưng không tốt cho MNIST (bởi vì flip là thành ra số khác rồi còn gì).

### 4. Sự đa dạng của augmentation

Với danh sách các phương thức augmentation kể trên thì cũng còn nhiều cách thức mình chưa liệt kê hết. Bản thân mỗi augmentation lại có các Yếu tố điều khiển riêng. Mình có thể phân loại thành:

- + Các phương thức augmentation: flip, rotation, random crop, v.v.
- + Các yếu tố điều khiển augmentation: mỗi augmentation sẽ có các yếu tố điều khiển riêng. Ví dụ rotation thì bao nhiêu độ, scaling thì scaling lên xuống bao nhiêu lần, crop random thì random trong khoảng bao nhiêu ...
- + Tần suất sử dụng từng phương thức augmentation?
- + Cách augmentation tốt nhất là một bộ các phương thức augmentation
- + Cách augmentation cho từng giai đoạn training/epoch có thể khác nhau. Epoch đầu ở learning rate lớn có thể khác với các epoch cuối ở learning rate nhỏ
- + Cách augmentation cho từng class, tần suất dùng augmentation có thể khác nhau
- + Cách augmentation cũng có thể bị ảnh hưởng bởi cấu trúc mạng. Tức là augmentation cho mức gain khác nhau tới từng network. Và không phải các yếu tố kể trên có tác động giống nhau tới mức độ cải thiện chất lượng của augmentation.

### 5. Cách augmentation tốt nhất?

Việc phụ thuộc vào dữ liệu và ứng dụng, kiến trúc mạng, kể trên đồng nghĩa với việc chúng ta cần phải thử khá, và chắc chắn sẽ tốn rất nhiều thời gian mà chưa chắc tìm ra cách augmentation tốt nhất.

## II. CÁC PHƯƠNG PHÁP TĂNG CƯỜNG DỮ LIỆU

### 1. Noise Injection

#### a. Gaussian Noise

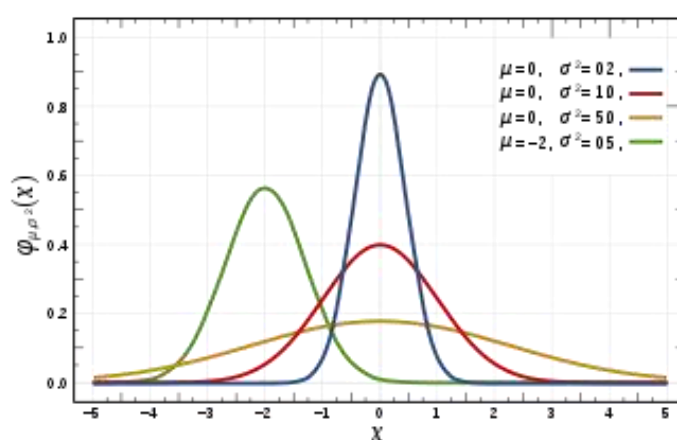
Nhiều Gaussian còn được gọi là nhiễu điện tử vì nó phát sinh trong bộ khuếch đại hoặc cảm biến gây ra.

Nhiều Gaussian là nhiễu có xác suất với hàm mật độ xác suất chính là hàm phân phối chuẩn (normal distribution) còn được biết đến như là phân phối Gaussian (Gaussian Distribution).

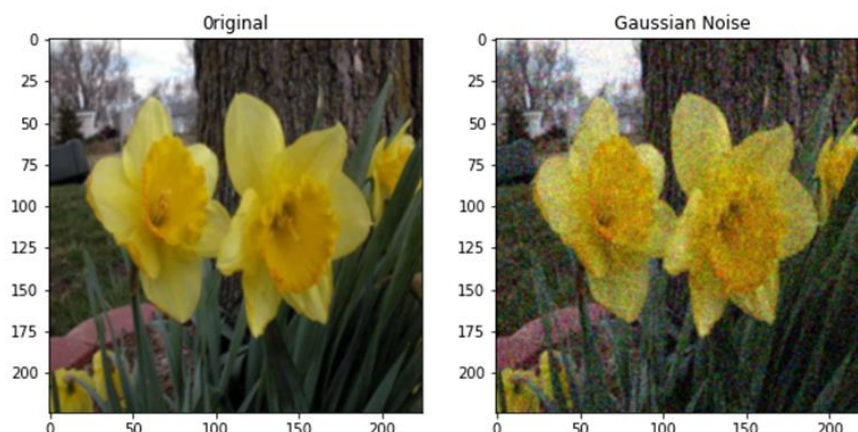
Hàm mật độ xác suất Gaussian là:

$$P(g) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(g-\mu)^2}{2\sigma^2}}$$

Với  $g$  = giá trị màu,  $\sigma^2$  là phương sai,  $\mu$  là số bình quân.



Đồ thị minh họa phân phối Gaussian



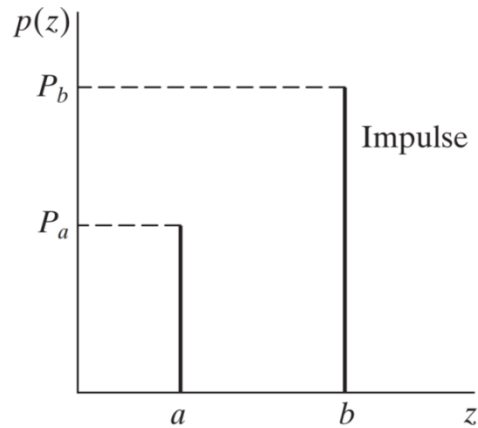
#### b. Salt and Pepper Noise

Nhiều muối tiêu (**Salt and Pepper Noise**) còn được gọi là drop data noise (nhiều loại bỏ thông tin gốc) vì nhiễu sẽ loại bỏ giá trị pixel của ảnh theo xác suất cho sẵn. Nhiều muối tiêu sẽ thay đổi bất kỳ giá trị pixel ngẫu nhiên của ảnh bằng giá trị pixel 255 (sáng) hay 0 (tối).

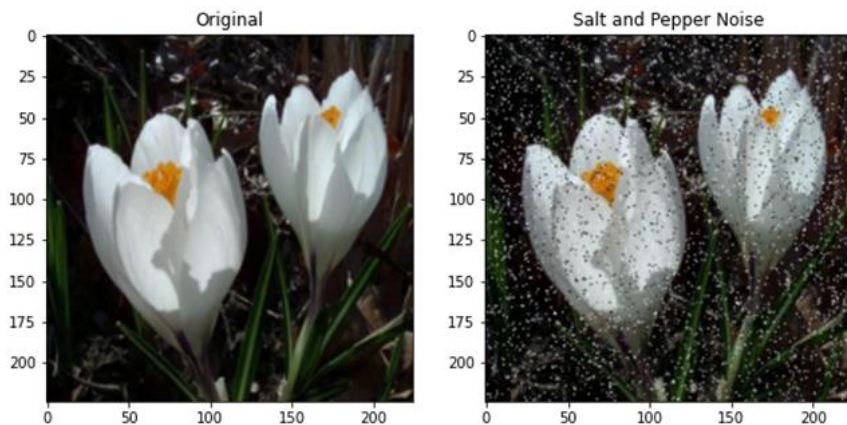
Lấy ma trận 3x3 cho hình minh họa bên dưới. Giá trị ở giữa đã bị thay thế bằng 0 bởi nhiễu muối tiêu.

254	207	210
97	212	32
62	106	20

254	207	210
97	0	32
62	106	20



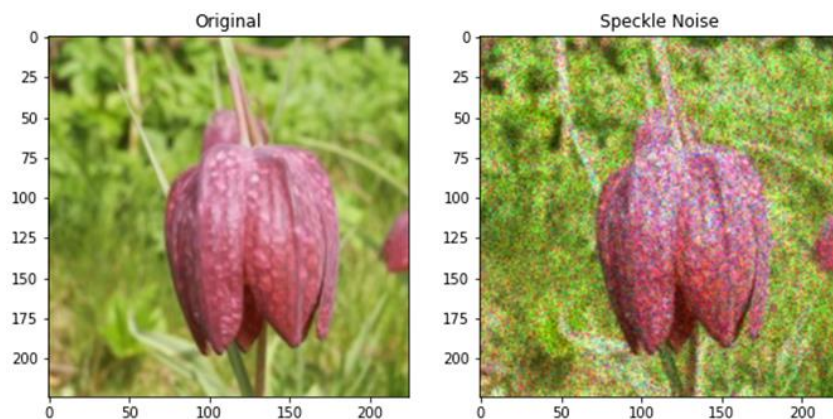
### Hàm phân phối mật độ xác suất của Salt and Pepper noise



### c. Speckle Noise

Xuất hiện ở hệ thống ảnh phân tán như radar, sóng siêu âm y tế, v.v... Nó được tạo ra do sự giao thoa của các thành phần tín hiệu phản xạ và có thể quan sát dưới hình dạng hạt trên ảnh siêu âm. Hàm phân phối mật độ xác suất của speckle noise tuân theo phân bố Gamma. Công thức hàm phân phối mật độ xác suất:

$$P(g) = \frac{g^{\alpha-1} e^{-\frac{g}{\alpha}}}{(\alpha-1)! \alpha^\alpha}$$





## 2. Geometric Transformation

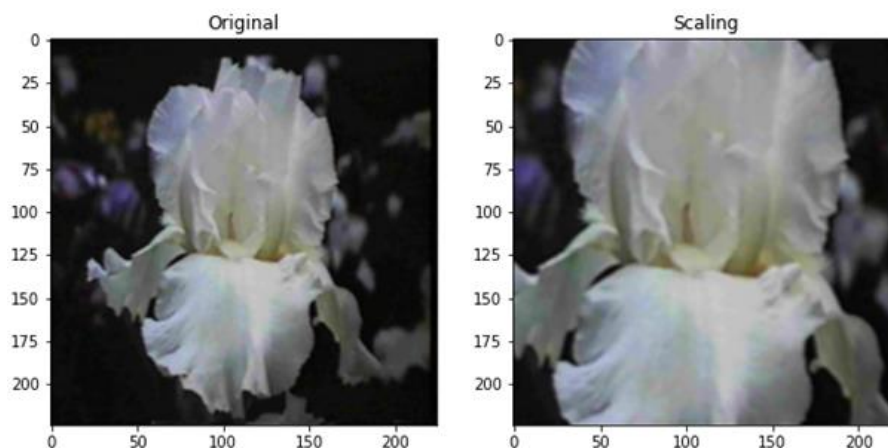
### a. Scaling

Phóng to hay thu nhỏ bức ảnh. Điều này giúp cho máy được cách con người nhìn ở góc xa góc gần.

Với phép tỉ lệ (nghĩa là giãn ra hay co lại), ta có:  $x' = s_x \cdot x$  và  $y' = s_y \cdot y$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Với  $(x, y)$  là tọa độ của điểm trước phép tỉ lệ và  $(x', y')$  là tọa độ của điểm sau phép tỉ lệ,  $s_x$  và  $s_y$  là scaling factor.



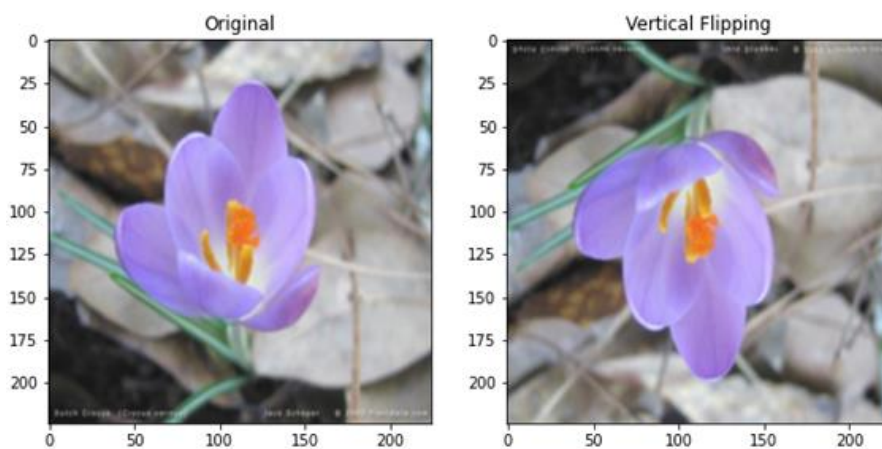
### b. Flipping

Flipping là phép lật nội dung của hình ảnh.

Flipping bao gồm hai loại:

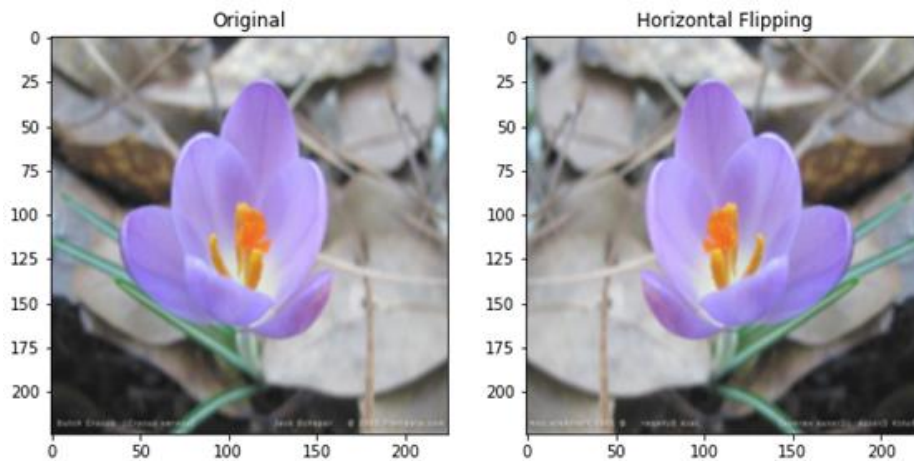
- + Lật theo chiều dọc (từ trên xuống dưới) dựa theo trục Ox.
- + Lật theo chiều ngang (từ dưới lên trên) dựa theo trục Oy.

Ta có thể lật ảnh tùy ý miễn sao vẫn giữ nguyên được ý nghĩa của ảnh. Hoặc ta có thể suy ra ảnh input ban đầu từ ảnh output mới có được. Trong phép Flipping, kích thước của đối tượng không thay đổi. Lật theo chiều ngang phổ biến hơn so với lật theo chiều dọc.



Lật theo chiều dọc





Lật theo chiều ngang

### c. Rotation

Quay bức ảnh sang trái hay sang phải từ 1 độ đến 359 độ. Phương pháp rotation augmentation hoàn toàn quyết định bằng tham số góc quay. Nếu góc quay quá lớn thì bức ảnh không còn giữ đúng thông tin với nhãn của mình.

Khi xoay tọa độ một góc  $\theta$  thuận chiều kim đồng hồ quanh điểm gốc, dạng hàm của nó là:

$$x' = x \cdot \cos \theta + y \cdot \sin \theta \text{ và } y' = -x \cdot \sin \theta + y \cdot \cos \theta$$

Viết dưới dạng ma trận:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

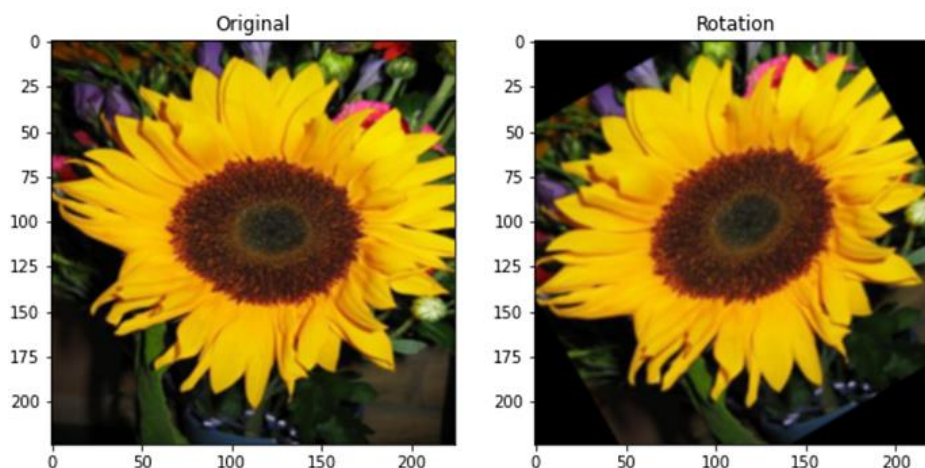
Khi xoay tọa độ một góc  $\theta$  ngược chiều kim đồng hồ quanh điểm gốc, dạng hàm của nó là:

$$x' = x \cdot \cos \theta - y \cdot \sin \theta \text{ và } y' = x \cdot \sin \theta + y \cdot \cos \theta$$

Viết dưới dạng ma trận:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Với  $(x, y)$  là tọa độ của điểm trước khi quay và  $(x', y')$  là tọa độ của điểm sau khi quay.



#### d. Translation

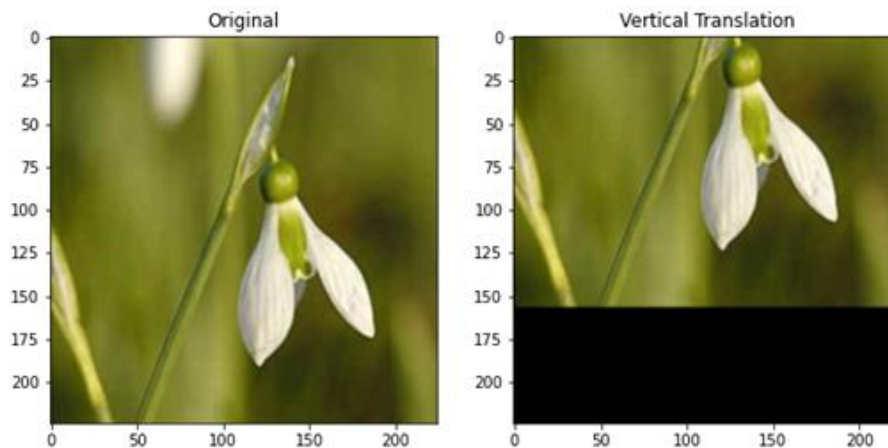
Dịch chuyển bức ảnh lên, xuống, trái, phải để tránh positional bias. Ví dụ nếu tất cả các chữ của các ảnh nằm ở trung tâm bức ảnh thì yêu cầu các chữ phải nằm ngay chính giữa ảnh để nhận diện tốt được.

Với vector tịnh tiến  $(t_x, t_y)$  ta có công thức  $x' = x + t_x$  và  $y' = y + t_y$

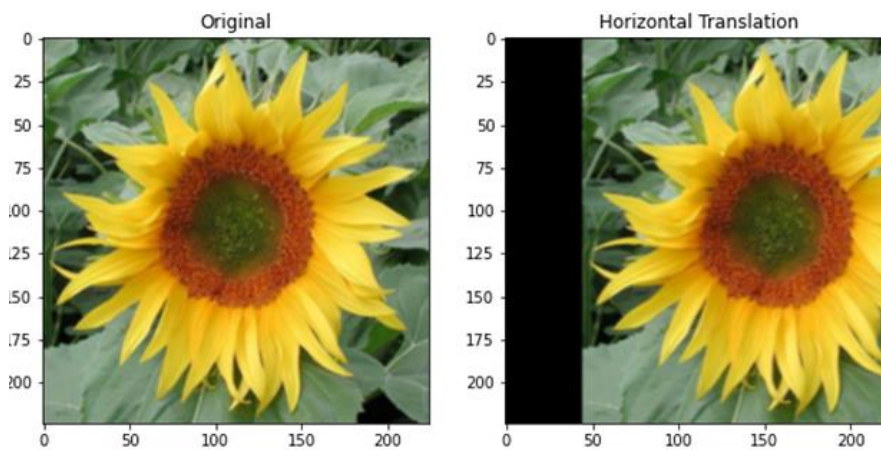
Viết dưới dạng ma trận là:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Với  $(x, y)$  là tọa độ của điểm trước khi tịnh tiến và  $(x', y')$  là tọa độ của điểm sau phép tịnh tiến.



Translation theo chiều dọc



Translation theo chiều ngang

#### e. Shearing

Làm méo mó ảnh theo các trục, tạo nên các góc nhìn khác nhau. Điều này giúp cho máy được cách con người nhìn ở nhiều góc nghiêng khác nhau.

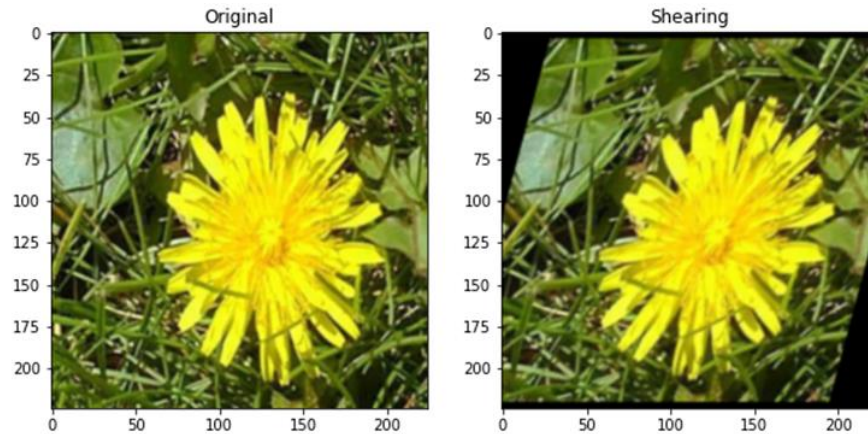
Shearing dọc theo trục hoành  $x$  có công thức  $x' = x + ky$  và  $y' = y$ ; dạng ma trận là:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shearing song song theo trục tung  $y$  có công thức  $x' = x$  và  $y' = y + kx$  [12]; dạng ma trận là:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ k & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Với  $(x, y)$  là tọa độ của điểm trước phép nghiêng,  $(x', y')$  là tọa độ của điểm sau phép nghiêng và  $k$  là shearing factor.



### 3. Color Space Transformation

#### a. Solarize

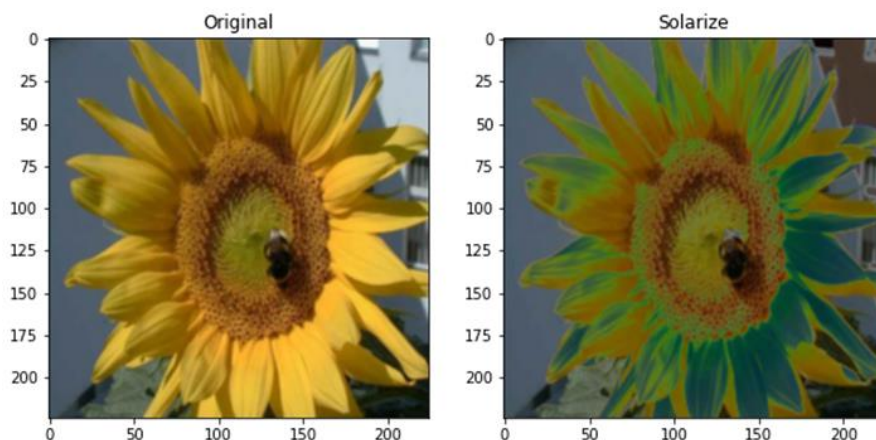
Solarize là phép đảo ngược các giá trị pixel trên một ngưỡng nhất định.

Solarize được sử dụng để mô tả hiệu ứng đảo ngược tông màu trong trường hợp hình ảnh bị chiếu sáng quá mức. Hiệu ứng này lần đầu tiên được nhìn thấy trong các bức ảnh phong cảnh bao gồm cả Mặt Trời. Lẽ ra, Mặt Trời là điểm trắng nhất trong hình nhưng lại chuyển qua màu đen hoặc màu xám. Solarize thường được sử dụng trong lĩnh vực nhiếp ảnh.

Cú pháp: PIL.ImageOps.solarize(image, threshold)

- image
- threshold

Với **image** là bức ảnh muốn đảo ngược và **threshold** là ngưỡng pixel do người dùng chọn (thường là 128), những pixel có giá trị lớn hơn ngưỡng này sẽ bị đảo ngược.



#### b. Posterize

Posterize là phép giảm số bit cho mỗi kênh màu.

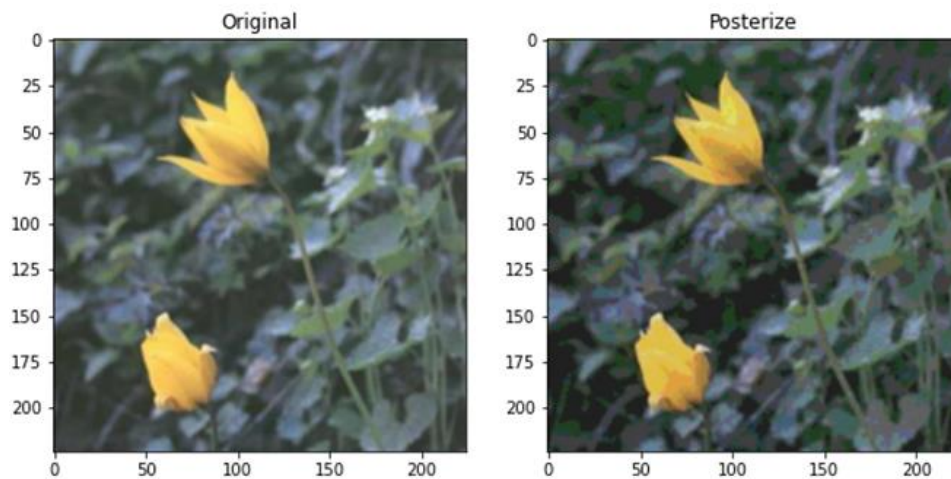


Posterize được sử dụng để in hoặc hiển thị một hình ảnh với một số tông màu hoặc màu sắc hạn chế theo cách phù hợp với người sử dụng. Posterize thường được thực hiện với các quy trình chụp ảnh để tạo áp phích.

Cú pháp: PIL.ImageOps.posterize(image, bits)

- image
- bits

Với **image** là bức ảnh muốn giảm bit và **bits** là số bit cần giữ cho mỗi kênh màu (1-8). Bit 8 là bit tối đa mà kênh có thể sử dụng.

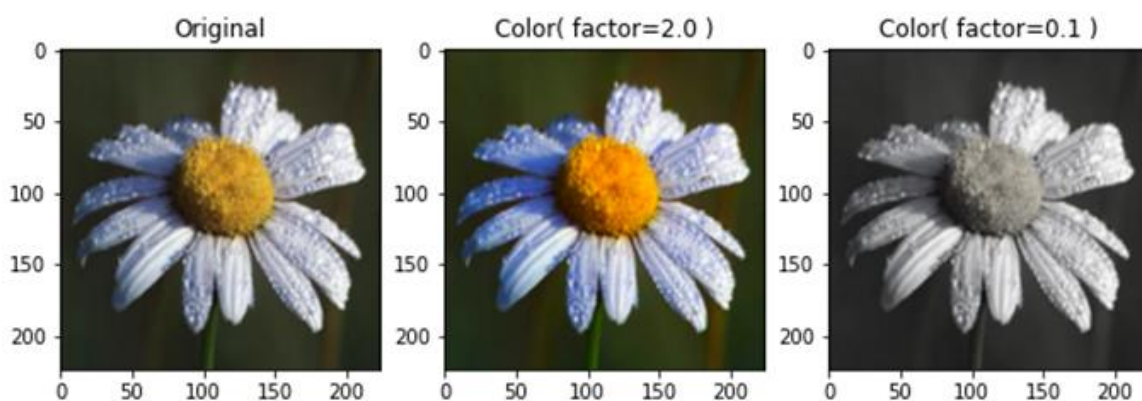


### c. Color

Color có thể được sử dụng để điều chỉnh độ cân bằng màu sắc của hình ảnh, theo cách tương tự như điều khiển trên TV màu.

Hệ số factor bằng 1 sẽ cho ra bức ảnh giống với ảnh gốc, hệ số nhỏ hơn 1 sẽ cho ra bức ảnh trắng đen, còn lớn hơn 1 sẽ cho ra bức ảnh sắc sỡ màu sắc hơn.

Cú pháp: PIL.ImageEnhance.Color(image)

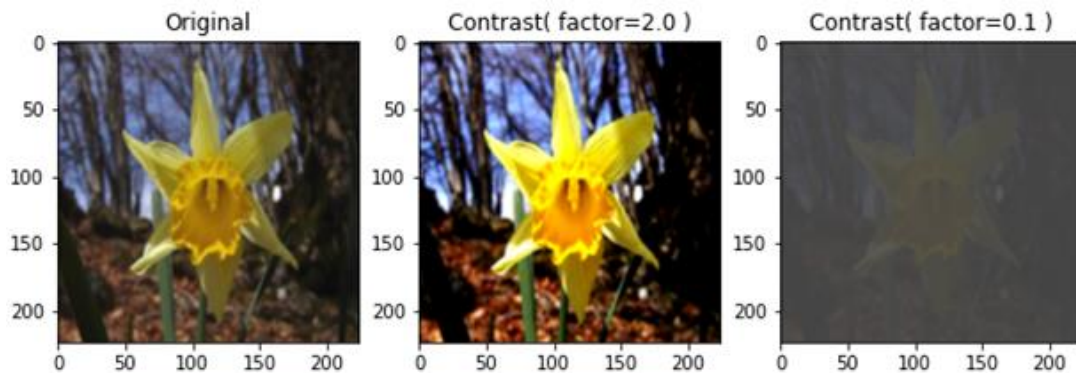


### d. Contrast

Color có thể được sử dụng để kiểm soát độ tương phản của hình ảnh, theo cách tương tự như kiểm soát độ tương phản trên TV.

Hệ số factor bằng 1 sẽ cho ra bức ảnh giống với ảnh gốc, hệ số nhỏ hơn 1 sẽ cho ra bức ảnh màu xám đặc, còn lớn hơn 1 sẽ cho ra bức ảnh mà trong đó đối tượng có thể dễ dàng nhận ra hơn.

Cú pháp: PIL.ImageEnhance.Contrast(image)

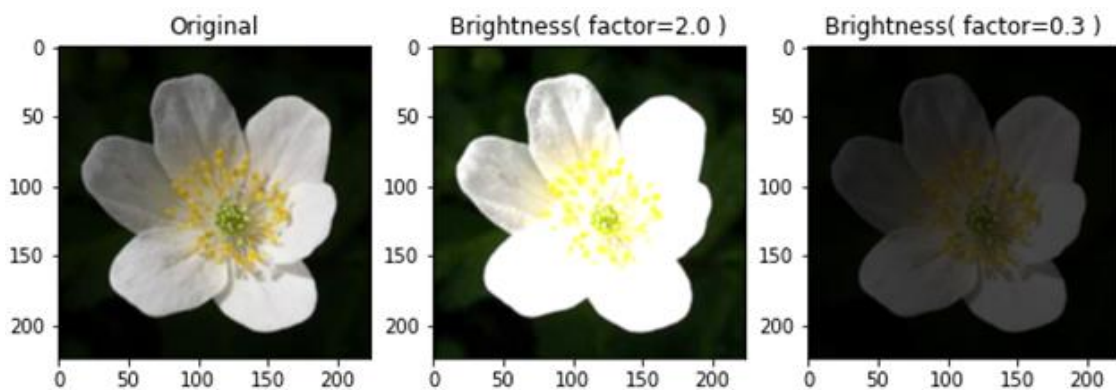


#### e. Brightness

Brightness có thể được sử dụng để kiểm soát độ sáng của hình ảnh.

Hệ số factor bằng 1 sẽ cho ra bức ảnh giống với ảnh gốc, hệ số nhỏ hơn 1 sẽ cho ra bức ảnh màu đen, còn lớn hơn 1 sẽ cho ra bức ảnh với độ sáng cao.

Cú pháp: PIL.ImageEnhance.Brightness(image)

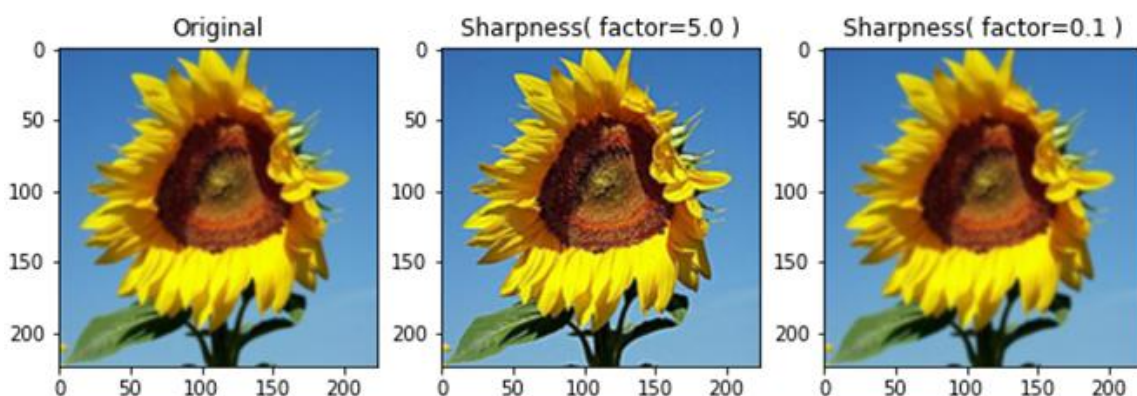


#### f. Sharpness

Sharpness có thể được sử dụng để điều chỉnh độ sắc nét của hình ảnh.

Hệ số factor bằng 1 sẽ cho ra bức ảnh giống với ảnh gốc, hệ số nhỏ hơn 1 sẽ cho ra tấm hình bị mờ, còn lớn hơn 1 sẽ cho ra bức ảnh với độ sắc nét cao.

Cú pháp: PIL.ImageEnhance.Sharpness(image)



## 4. Cut Out

CutOut là một kỹ thuật tăng cường dữ liệu thú vị khác Lấy cảm hứng từ các cơ chế dropout. CutOut có thể được coi là tương tự như dropout ngoại trừ việc CutOut được sử dụng trong không gian dữ liệu đầu vào thay vì được nhúng vào kiến trúc mạng. Kỹ thuật này được thiết kế đặc biệt để chống lại những thách thức về nhận dạng hình ảnh do occlusion (sự tắc nghẽn). Occlusion là khi đối tượng bị đối tượng khác che, khi đó một số phần của đối tượng không rõ ràng. CutOut sẽ ngăn chặn điều này bằng cách buộc mô hình học thêm về các đặc điểm mô tả khác của bức ảnh, ngăn mô hình overfitting với chỉ một đặc điểm trực quan (visual feature) nhất định trong hình ảnh. Đặc biệt, bên cạnh thách thức về vấn đề occlusion, CutOut là một kỹ thuật đầy hứa hẹn để đảm bảo mạng chú ý đến toàn bộ hình ảnh, thay vì chỉ một tập hợp con của nó.

CutOut hoạt động bằng cách chọn ngẫu nhiên các patch  $n \times m$  của một hình ảnh và che nó bằng giá 0 s, 255 s, giá trị pixel trung bình hoặc giá trị ngẫu nhiên. Các patch có giá trị các giá trị ngẫu nhiên cho kết quả tốt nhất.

CutOut là một phương pháp tăng cường dữ liệu tìm cách trực tiếp ngăn chặn overfitting bằng cách thay đổi không gian đầu vào. Bằng cách loại bỏ các input patches nhất định, mô hình buộc phải tìm các đặc điểm mô tả khác. Phương pháp tăng cường này cũng có thể sử dụng với các phương pháp tăng cường dữ liệu khác như horizontal flipping hay color filters.

Một bất lợi đối với phương pháp CutOut là nó không phải lúc nào cũng là một phép biến đổi bảo toàn nhãn. Ví dụ, trong nhận dạng chữ số viết tay nếu phần trên cùng của số '8' bị cắt ngẫu nhiên, thì nó không khác gì số '6'. Trong nhiều tác vụ chi tiết chẳng hạn như bộ dữ liệu Stanford Cars, việc xóa ngẫu nhiên các phần của hình ảnh (logo, v.v...) có thể khiến thương hiệu ô tô không thể nhận dạng được. Do đó, một số can thiệp thủ công có thể cần thiết tùy thuộc vào tập dữ liệu và nhiệm vụ.

### *Algorithm:*

Input: Input image I;

Image size W and H;

Area of image S;

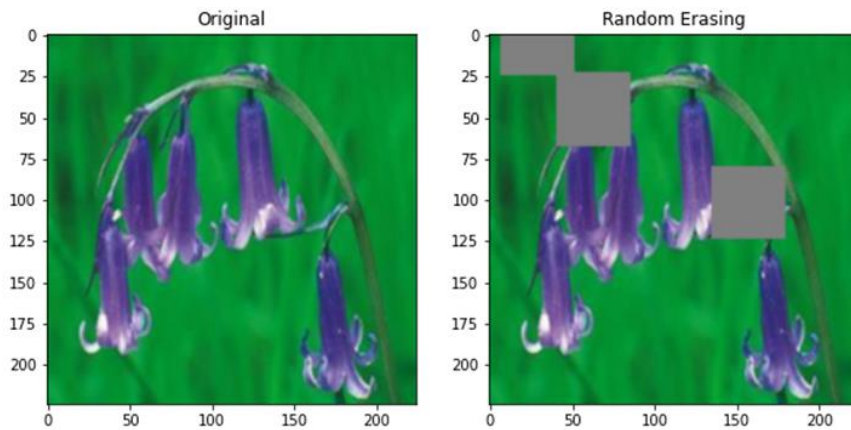
Erasing area ratio range sl and sh;

Erasing aspect ratio range r1 and r2.

Output: Erased image I

- 1 while True do
- 2  $Se \leftarrow \text{Rand}(sl, sh) \times S$ ;
- 3  $re \leftarrow \text{Rand}(r1, r2)$ ;
- 4  $He \leftarrow \sqrt{Se * re}$ ,  $We \leftarrow \sqrt{\frac{Se}{re}}$ ;
- 5  $xe \leftarrow \text{Rand}(0, W)$ ,  $ye \leftarrow \text{Rand}(0, H)$ ;
- 6 if  $xe + We \leq W$  and  $ye + He \leq H$  then
- 7  $Ie \leftarrow (xe, ye, xe + We, ye + He)$ ;
- 8  $I(Ie) \leftarrow \text{Rand}(0, 255)$ ;
- 9  $I * \leftarrow I$ ;
- 10 return  $I *$ .
- 11 end

- 13 end



## 5. Mixing Images

Mixing images là phương pháp tăng cường dữ liệu bằng cách lấy trung bình các giá trị pixel của chúng, là một cách tiếp cận khác thường đối với tăng cường dữ liệu. Các hình ảnh được tạo ra bằng cách này sẽ không giống như một phép biến đổi hữu ích đối với người thường quan sát.

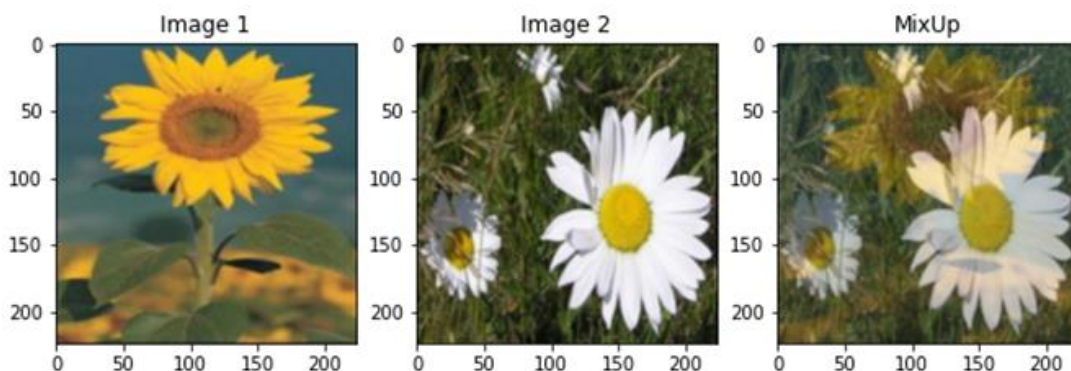
### a. MixUp

MixUp là một kỹ thuật tăng cường dữ liệu *domain-agnostic* được đề xuất trong MixUp: Beyond Empirical Risk Minimization của Zhang et al. Nó được thực hiện với các công thức sau:

$$\begin{aligned}\tilde{x} &= \lambda x_i + (1 - \lambda)x_j, & \text{where } x_i, x_j \text{ are raw input vectors} \\ \tilde{y} &= \lambda y_i + (1 - \lambda)y_j, & \text{where } y_i, y_j \text{ are one-hot label encodings}\end{aligned}$$

(Các giá trị lambda là các giá trị có phạm vi  $[0, 1]$  và được lấy mẫu từ bản phân phối Beta.)

Phương pháp này hiệu đơn giản đơn giản chính là mix up (trộn lẫn) đặc trưng ảnh và nhãn tương ứng của các ảnh. Mạng lưới thần kinh dễ bị ghi nhớ các nhãn bị hỏng (corrupted label). MixUp nói lỏng điều này bằng cách kết hợp các đặc trưng khác nhau với nhau (điều tương tự cũng xảy ra với các nhãn) để mạng không quá tự tin về mối quan hệ giữa các đặc trưng và nhãn của chúng dẫn đến overfitting.



### b. CutMix

Cụ thể, để ngăn CNN tập trung quá nhiều vào một tập hợp nhỏ các activation trung gian hoặc vào một vùng nhỏ trên hình ảnh đầu vào, các quy tắc loại bỏ đặc trưng ngẫu nhiên (random feature removal regularizations) đã được đề xuất. Các ví dụ bao gồm dropout để loại bỏ ngẫu nhiên **hidden activation** và **regional dropout** để xóa các vùng ngẫu nhiên trên đầu vào. Các nhà nghiên cứu đã chỉ ra rằng các chiến lược loại bỏ đặc trưng cải thiện khả năng generalization và



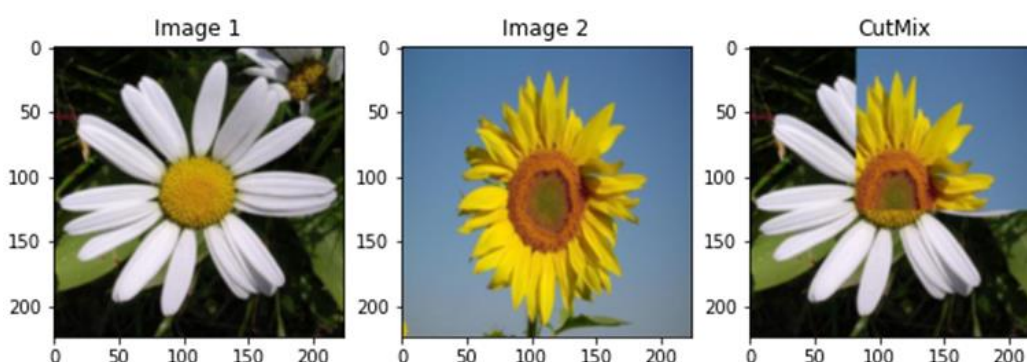
localization bằng cách cho phép một mô hình không chỉ tham gia vào các phần phân biệt một phần của đối tượng mà còn cho toàn bộ vùng đối tượng.

Mặc dù các chiến lược regional dropout đã cho thấy sự cải thiện về hiệu suất bài toán classification và localization ở một mức độ nhất định, nhưng các khu vực bị xóa thường bị bỏ trống hoặc chứa đầy nhiễu ngẫu nhiên, làm giảm đáng kể tỷ lệ pixel chứa thông tin trên tập ảnh huấn luyện. Đây là một hạn chế nghiêm trọng về mặt khái niệm vì CNN thường “đói” dữ liệu.

CutMix là một kỹ thuật tăng cường dữ liệu nhằm giải quyết vấn đề mất thông tin và không hiệu quả trong các chiến lược regional dropout. Thay vì loại bỏ các pixel và lấp đầy chúng bằng các pixel đen hoặc xám hoặc nhiễu Gaussian, thay thế các vùng đã loại bỏ bằng một patch từ một hình ảnh khác, trong khi các ground truth labels được trộn theo tỷ lệ với số lượng pixel của các hình ảnh được kết hợp.

CutMix hiện có đặc điểm là không có pixel nào không có thông tin trong quá trình đào tạo, giúp quá trình đào tạo hiệu quả, trong khi vẫn giữ được lợi thế của regional dropout. Các patches được thêm vào nâng cao hơn nữa khả năng localization bằng cách yêu cầu mô hình xác định đối tượng từ một phần góc nhìn.

CutMix có điểm tương đồng với MixUp trộn hai mẫu bằng cách nội suy cả ảnh và nhãn. Mặc dù chắc chắn cải thiện hiệu suất bài toán classification, các mẫu MixUp thường nhìn không tự nhiên. CutMix khắc phục sự cố bằng cách thay thế vùng hình ảnh bằng một bản vá từ ảnh huấn luyện khác.



## 6. RandAugment

Thật không may, các phương pháp tăng cường dữ liệu yêu cầu chuyên môn và công việc thủ công để thiết kế các chính sách thu thập kiến thức trước đó trong mỗi miền. Yêu cầu này gây khó khăn cho việc mở rộng các phương pháp tăng cường dữ liệu hiện có sang các ứng dụng và miền khác.

Các chính sách học tập để tăng cường dữ liệu gần đây đã nổi lên như một phương pháp tự động hóa việc thiết kế các chiến lược cho tăng cường dữ liệu và do đó có khả năng giải quyết một số điểm yếu của các phương pháp tăng cường dữ liệu truyền thống. Đào tạo mô hình học máy bằng chính sách tăng cường dữ liệu đã học có thể cải thiện đáng kể độ chính xác, độ **robustness** của mô hình và hiệu suất học bán giám sát cho tác vụ phân loại hình ảnh; tương tự như vậy, đối với các tác vụ phát hiện đối tượng trên dataset **COCO** và **PASCAL-VOC**. Đáng chú ý, không giống như kỹ thuật kiến trúc mạng tốt hơn, tất cả những cải tiến này trong hiệu suất dự đoán không phát sinh thêm chi phí tính toán tại thời điểm tính toán.

Bất chấp những lợi ích của các chính sách tăng cường dữ liệu tự học, các yêu cầu tính toán cũng như độ phức tạp gia tăng của hai quy trình tối ưu hóa riêng biệt có thể gây khó khăn.

Đối với các phương pháp tăng cường dữ liệu dựa trên NAS, chẳng hạn như AutoAugment (AA), cần có không gian tìm kiếm lớn để tìm một tập hợp các kỹ thuật tăng

cường. Giai đoạn tìm kiếm riêng biệt này làm phức tạp đáng kể quá trình đào tạo và tốn kém về mặt tính toán.

Trong RandAugment, một cách đơn giản nhưng hiệu quả được đề xuất là chọn ngẫu nhiên một tập hợp các kỹ thuật tăng cường.

```
transforms = [  
    'Identity', 'AutoContrast', 'Equalize',  
    'Rotate', 'Solarize', 'Color', 'Posterize',  
    'Contrast', 'Brightness', 'Sharpness',  
    'ShearX', 'ShearY', 'TranslateX', 'TranslateY']  
  
def randaugment(N, M):  
    """Generate a set of distortions.  
  
    Args:  
        N: Number of augmentation transformations to  
            apply sequentially.  
        M: Magnitude for all the transformations.  
    """  
  
    sampled_ops = np.random.choice(transforms, N)  
    return [(op, M) for op in sampled_ops]
```

#### a. N: Number of transformation

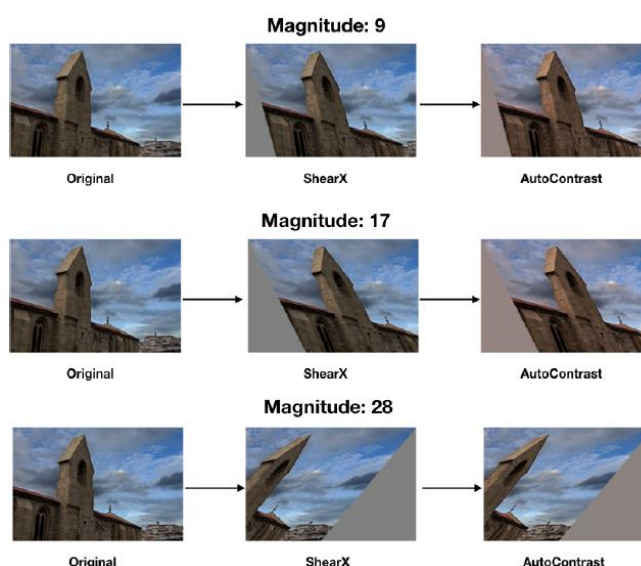
Mục tiêu chính của RandAugment là loại bỏ nhu cầu về giai đoạn tìm kiếm riêng trên proxy task.

Một quy trình không có tham số được đề xuất trong đó RandAugment luôn chọn một phép biến đổi có xác suất đồng nhất  $1/K$ .

Với N phép biến đổi cho một ảnh huấn luyện, do đó, RandAugment có thể thể hiện  $K^N$  chính sách tiềm năng.

Với  $N=14$ , số phép biến đổi là: *identity*, *autoContrast*, *equalize*, *rotate*, *solarize*, *color*, *posterize*, *contrast*, *brightness*, *sharpness*, *shear-x*, *shear-y*, *translate-x*, *translate-y*.

#### b. M: Magnitude of Transformation



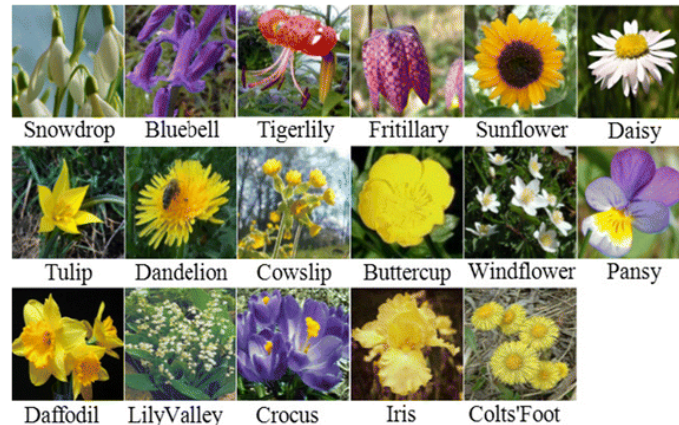
Tham số cuối cùng cần xem xét là độ lớn (magnitude) của mỗi biến dạng tăng cường dữ liệu.

Mỗi phép biến đổi nằm trên thang số nguyên từ 0 đến 10 trong đó giá trị 10 biểu thị thang tối đa cho một phép biến đổi đã cho nhất định.

### III. THỰC NGHIỆM

#### 1. Dataset

Nhóm sử dụng bộ dataset VGG FLOWERS 17 (A Visual Vocabulary for Flower Classification M. Nilsback, A. Zisserman).



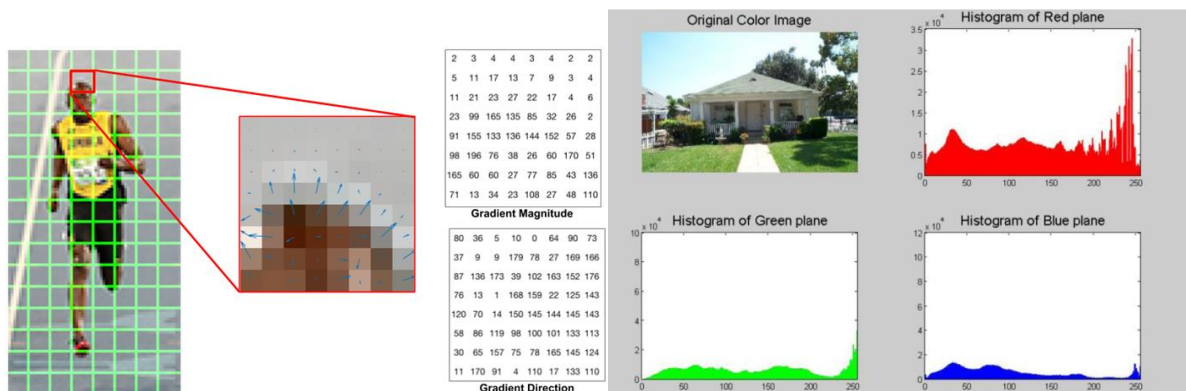
Dataset gồm 17 loại hoa khác nhau với 80 ảnh mỗi loại. Có những loài có vẻ ngoài rất độc đáo, ví dụ như hoa bách hợp (fritillaries) và hoa loa kèn (tigerlilies). Cũng như những loài có vẻ ngoài rất giống nhau, chẳng hạn như hoa bồ công anh (dandelions) và coltsfeet.

Dataset đa dạng về góc nhìn, tỷ lệ và chiếu sáng. Sự phương sai lớn trong lớp và phương sai nhỏ đa lớp khiến cho bộ dữ liệu vô cùng thách thức. Các loại hoa được chọn có chủ ý để có một số sự mơ hồ về từng khía cạnh. Ví dụ, một số loài không thể phân biệt chỉ dựa vào màu sắc (bồ công anh (dandelion) và mao lương (buttercup)), những loài khác không thể phân biệt chỉ dựa vào hình dạng (hoa thủy tiên vàng (daffodils) và hoa gió (windflower)). Các hình ảnh hoa được lấy từ các trang web khác nhau.

Dataset sẽ được chia thành tỉ lệ 8:2 cho training set và test set. Training set sẽ được tăng cường bằng các phương pháp nói trên. Những phần bị mất của ảnh trong quá trình tăng cường ví dụ như các góc của ảnh được tăng cường bằng phương pháp Rotation bị mất do đã di chuyển ra ngoài sẽ được bồi đắp giá trị pixel đen (255) hay background đen.

#### 2. Extract feature

Nhóm sử dụng Histogram of oriented gradient và histogram để extract feature của ảnh.



#### 3. Model

Nhóm dùng Support Vector Machine để đánh giá kết quả.

#### 4. Evaluate

**Raw: 53,81%**

#### a. Vertical translation

	Accuracy
Translation <b>30%</b> lên trên	+2,39%
Translation <b>20%</b> lên trên	+4,92%
Translation <b>10%</b> lên trên	+5,55%
Translation <b>10%</b> xuống dưới	+3,02%
Translation <b>20%</b> xuống dưới	+1,44%
Translation <b>30%</b> xuống dưới	+2,39%
Random vertical translation	+3,02%

##### Nhận xét:

Phương pháp tốt nhất khi tịnh tiến ảnh lên trên và đạt kết quả tốt nhất khi biến đổi trong khoảng 10%.

#### b. Horizontal translation

	Accuracy
Translation <b>30%</b> qua phải	+4.18%
Translation <b>20%</b> qua phải	+4,97%
Translation <b>10%</b> qua phải	+4,04%
Translation <b>10%</b> qua trái	+3,18 %
Translation <b>20%</b> qua trái	+6%
Translation <b>30%</b> qua trái	+2,64 %
Random horizontal translation	+3,74 %

##### Nhận xét:

Dù translation qua phải đạt kết quả khả quan hơn qua trái tuy nhiên translation lại đạt kết quả tốt nhất khi translation qua trái 20%.

#### c. Rotation

	Accuracy
Rotation <b>30</b> độ ngược chiều kim đồng hồ	+6,58%
Rotation <b>20</b> độ ngược chiều kim đồng hồ	+8,01%
Rotation <b>15</b> độ ngược chiều kim đồng hồ	+6,29%
Rotation <b>10</b> độ ngược chiều kim đồng hồ	+7.15%
Rotation <b>5</b> độ ngược chiều kim đồng hồ	+7.44%

Rotation <b>5</b> độ cùng chiều kim đồng hồ	+6,58%
Rotation <b>10</b> độ cùng chiều kim đồng hồ	+8,29%
Rotation <b>15</b> độ cùng chiều kim đồng hồ	+5,94%
Rotation <b>20</b> độ cùng chiều kim đồng hồ	+5,43%
Rotation <b>30</b> độ cùng chiều kim đồng hồ	+7,43%
Random rotation	+7,72%

Nhận xét:

Phương pháp đạt kết quả ổn định dù biến đổi ở góc nào khi chênh lệch cao nhất giữa các tham số chỉ gần 3%.

**d. Shearing**

	Accuracy
Shearing <b>30</b> độ ngược chiều kim đồng hồ	+3,34%
Shearing <b>20</b> độ ngược chiều kim đồng hồ	+5,57%
Shearing <b>15</b> độ ngược chiều kim đồng hồ	+2,7%
Shearing <b>10</b> độ ngược chiều kim đồng hồ	+4,63%
Shearing <b>5</b> độ ngược chiều kim đồng hồ	+7,44%
Shearing <b>5</b> độ cùng chiều kim đồng hồ	+4,95%
Shearing <b>10</b> độ cùng chiều kim đồng hồ	+6,19%
Shearing <b>15</b> độ cùng chiều kim đồng hồ	+5,88 %
Shearing <b>20</b> độ cùng chiều kim đồng hồ	+2,7%
Shearing <b>30</b> độ cùng chiều kim đồng hồ	+1,67%
Random shearing	+5,72%

Nhận xét:

Shearing chỉ tốt trong khoảng 10 độ. Tuy nhiên shearing lại tốt khi quay 20 độ ngược chiều kim đồng hồ.

**e. Scaling**

Scaling	Accuracy
0.5	+7,45%
0.6	+6,03%
0.7	+7,24%

0.8	+3,02%
0.9	+6,02%
1.1	+6,93%
1.2	+6,3%
1.3	+3,97%
1.4	+1,75%
1.5	+6,63%
Random	+6,12%

Nhận xét:

Scaling đạt kết quả tốt trong khoảng 0.5 và 1.2 và ngạc nhiên cả 1.4, 1.5 dù 1.3, 1.4 đạt kết quả không tốt.

#### **f. Flip**

	Accuracy
Vertical flip	+3,74%
Horizontal flip	+4,49%

Nhận xét:

Horizontal flipping đạt kết quả cao hơn vertical flip.

#### **g. Noise injection**

	Accuracy
Salt and pepper	+0,79%
Gaussian noise	-0,52%
Speckle noise	+1,44%

Nhận xét:

Các phương pháp noise injection đạt kết quả rất tệ. Gaussian noise không những tăng mà còn giảm.

#### **h. Color space transformation**

	Accuracy
Solorize	+3,88%
Color	+7,24%
Posterize	+4,79%
Contrast	+7,24%

Brightness	+5,4%
Sharpness	+6,93%

Nhận xét:

Các phương pháp enhance color tăng cường chất lượng mô hình cao hơn là các phương pháp color space transformation khác.

**i. Các phương pháp khác**

	Accuracy
CutOut	+1,12%
MixUp	+6,63%
CutMix	+7,75%
RandAugment	+8,46%
RandAugment (X3)	+9,38%
RandAugment (X4)	+8,46%

Nhận xét:

CutOut đạt kết quả không hề tốt. Trong khi MixUp, CutMix, RandAugment. RandAugment đạt kết quả tốt nhất khi tăng lên 3 lần dataset. Tuy nhiên, khi tăng lên 4 lần accuracy lại rơi xuống ngang 2 lần.

**j. Combine các phương pháp**

Ở đây nhóm sẽ kết hợp các phương pháp tốt nhất với nhau.

	Accuracy
Rotate + Scaling	+8,08%
Shear + Scaling	+7,45%
Translation (mixed)	+6,82%
Horizontal Translation + Scaling	+7,13%
Vertical Translation + Scaling	+7,13%
Sharpness + Scaling	+6,64%
Color + Scaling	+5,23%
Contrast + Scaling	+5,87%
Sharpness + Rotate	+5,55%
Color + Rotate	+4,29%
Contrast + Rotate	+5,86%



Sharpness + Shear	+5,23%
Color + Shear	+4,6%
Contrast + Shear	+7,76%
RandAugment + CutMix	+6,63%
RandAugment & CutMix (X3)	+5,4%

#### Nhận xét:

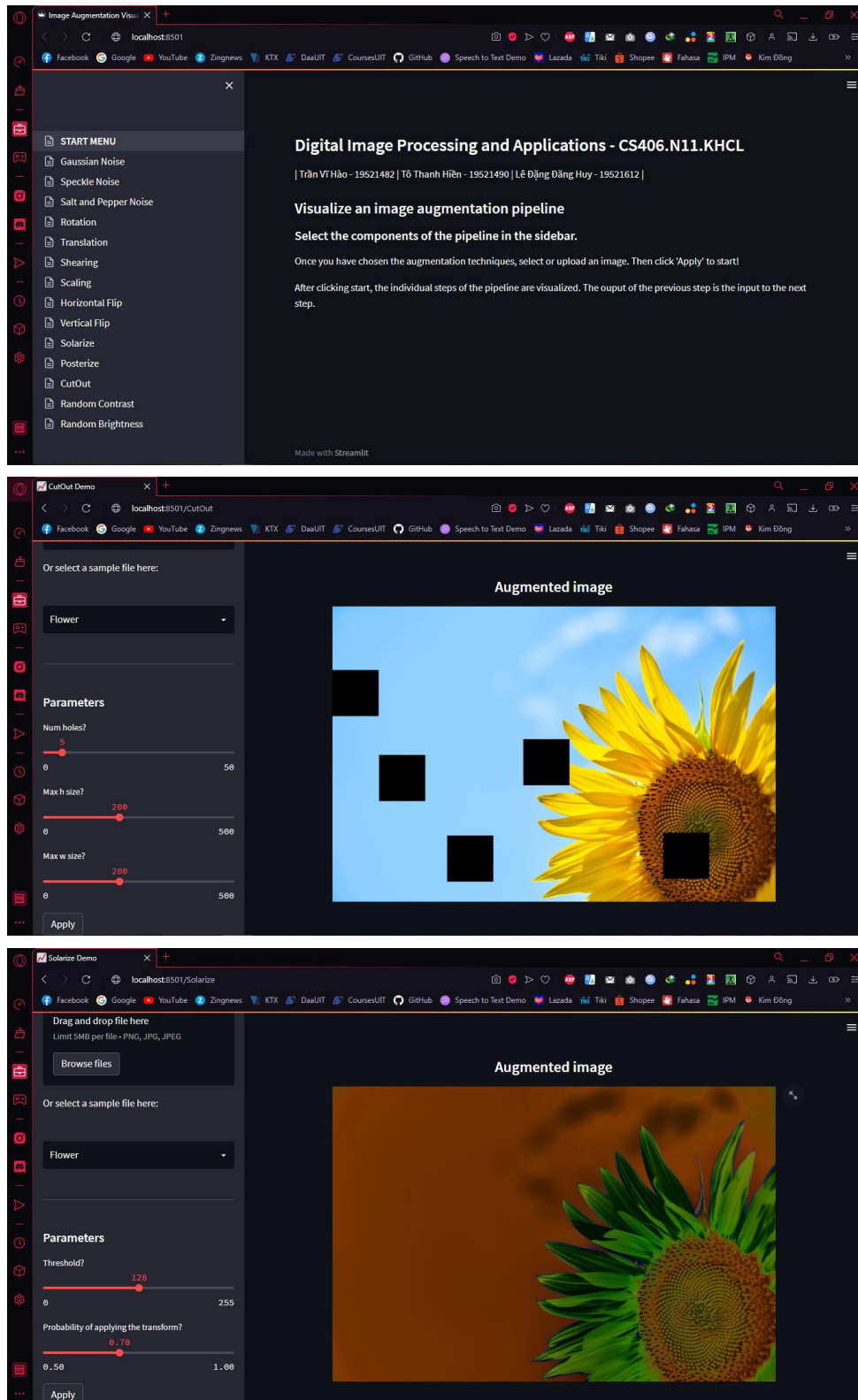
Các phương pháp Color đạt kết quả không tốt bằng khi combine các phương pháp khác ngoại trừ Contrast + Shear. RandAugment và CutMix không nên kết hợp với các phương pháp khác. Tuy nhiên các phương pháp geometrics lại đạt kết quả khả quan mặc dù không cao bằng lúc chưa kết hợp.

## 5. Kết luận

- Ngoại trừ Noise Injection và CutOut các phương pháp khác đều cho kết quả ổn nguyên nhân có thể do các phương pháp đó không phù hợp với dữ liệu và mô hình huấn luyện. Đối với cutOut phương pháp này dùng để đối phó với occlusion nhưng dataset lại không có. Trong khi noise injecton làm mất dữ liệu ảnh.
- RandAugment, CutMix, MixUp đạt kết quả rất tốt. Trong khi đó geometrics ngoại trừ flipping dễ dàng đạt kết quả trên 5%.
- Các phương pháp color space transformation, Contrast và Brightness đạt kết quả tốt nhất.
- Combine các phương pháp đạt kết quả không tốt khi chỉ sử dụng một phương pháp đặc biệt là các phương pháp Color Space Transformation, CutMix, RandAugment. Chỉ có kết hợp cả horizontal và vertical Translation đạt kết quả khả quan hơn khi sử dụng một mình.

## IV. DEMO

Nhóm có viết demo visualize các phương pháp tăng cường dữ liệu.



## V. TÀI LIỆU THAM KHẢO

- [1]. RandAugment: Practical automated data augmentation with a reduced search space
- [2]. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features
- [3]. Improved Regularization of Convolutional Neural Networks with Cutout
- [4]. mixup: BEYOND EMPIRICAL RISK MINIMIZATION
- [5]. A survey on Image Data Augmentation for Deep Learning
- [6]. A Visual Vocabulary for Flower Classification
- [7]. Thư viện Imgaug
- [8]. Thư viện Opencv
- [9]. Thư viện Sklearn
- [10]. <https://pillow.readthedocs.io/en/stable/reference/ImageOps.html#PIL.ImageOps.posterize>
- [11]. <https://pillow.readthedocs.io/en/stable/reference/ImageEnhance.html>
- [12]. <https://www.geeksforgeeks.org/python-pil-imageops-solarize-method/>
- [13]. <https://www.geeksforgeeks.org/python-pil-imageops-postarize-method/>
- [14]. <https://www.geeksforgeeks.org/python-pil-imageenhance-color-and-imageenhance-contrast-method/>
- [15]. <https://www.geeksforgeeks.org/python-pil-imageenhance-brightness-and-imageenhance-sharpness-method/>
- [16]. <https://viblo.asia/p/mot-so-phuong-phap-khu-nhieu-anh-cac-phuong-phap-co-dien-4dbZNGwnlYM>