

# Descripción General del Programa

Este proyecto de python procesa un set de imágenes aplicando una serie de filtros. Estos filtros permiten realizar tareas como cambiar brillo, contraste, desenfoque, detección de bordes, entre otros. Cada uno de los filtros es susceptible de ser configurado a través de parámetros definidos en el código, lo que permite la adaptación a las necesidades del usuario. El proyecto en general trabaja con imágenes en formato RGB (RED GREEN BLUE) y también realiza configuraciones de filtros para sets más complejos.

Cabe destacar que el programa no realiza giros de 90°, y tampoco ajusta sus etiquetas (siguiendo lineamientos/configuraciones de roboflow). Si se desea se lo puede modularizar e implementarlo en futuras versiones.

La mayoría de los filtros se basan en configuraciones sugeridas por Roboflow, aunque dicha plataforma ha presentado errores al procesar imágenes, por lo que se optó por una implementación personalizada para asegurar el correcto funcionamiento.

## Dependencias

Este proyecto utiliza las siguientes librerías y versiones:

Contourpy - 1.3.0  
Cycler - 0.12.1  
Fonttools - 4.53.1  
Imageio - 2.35.1  
Kiwisolver - 1.4.7  
lazy\_loader - 0.4  
Matplotlib - 3.9.2  
Networkx - 3.3  
Numpy - 2.1.1  
Opencv-python - 4.10.0.84  
Packaging - 24.1  
Pillow - 10.4.0  
Pyparsing - 3.1.4  
Python-dateutil - 2.9.0.post0  
Scikit-image - 0.24.0  
Scipy - 1.14.1  
Six - 1.16.0  
Tifffile - 2024.8.30

## Compilador

Este proyecto utiliza Python versión 3.10.12.

# Estructura del Proyecto

El proyecto tiene la siguiente estructura de carpetas:

```
amp_img/
├── ampliacion_lib/
│   ├── __init__.py
│   ├── blur_filter.py
│   ├── bordes_canny_f.py
│   ├── brightness_f.py
│   ├── contrast_f.py
│   ├── cropping_f.py
│   ├── cutout_f.py
│   ├── des_gauss_selec.py
│   ├── elastic_f.py
│   ├── exposure_f.py
│   ├── flipping_f.py
│   ├── gray_scale_f.py
│   ├── hue_f.py
│   ├── mix_blur_grayscale_f.py
│   ├── mix_contrast_brightness.py
│   ├── mix_saturation_brightness_f.py
│   ├── noise_f.py
│   ├── process_img.py
│   ├── rotate_filter.py
│   ├── rotation_f.py
│   ├── saturation_f.py
│   ├── sepia_f.py
│   ├── sobel_f.py
│   └── utils.py
├── data/
│   ├── input_images/
│   ├── processed_images/
│   └── process_changed/
├── README.md
├── requirements.txt
└── run_img_amp.py
```

## Ejecución

Para ejecutar el procesamiento de imágenes, simplemente corre el script principal **run\_img\_amp.py** desde la terminal. Este script aplicará los filtros configurados a las imágenes presentes en la carpeta `input_images` y almacenará los resultados en `processed_images`.

# Variables de inicialización:

Parámetros para los filtros a aplicar

## **Prefijo, nombre del set a tratar**

`prefix_name = 'Accordion'`

Prefijo utilizado para nombrar el conjunto de imágenes procesadas.

Este prefijo será parte del nombre de las imágenes renombradas al consolidar el set.

## **Índice de inicio para contador**

`cont_indice = 1`

Índice numérico de inicio utilizado para nombrar las imágenes.

Las imágenes serán renombradas secuencialmente comenzando con este valor.

## **Desenfoque (blur)**

`blur_sigma = 2.5`

Valor de sigma para el filtro de desenfoque gaussiano.

Este valor controla el grado de suavizado de la imagen, con un valor más alto produciendo un desenfoque más fuerte.

## **Canny (detector de bordes)**

`thres1 = 50`

`thres2 = 75`

Umbrales utilizados en el filtro de detección de bordes Canny.

`thres1` es el valor mínimo para la detección de bordes, mientras que `thres2` es el valor máximo.

Juntos controlan qué bordes se detectan en la imagen.

## **Brillo**

`brightness_percent = 50`

`mode = 'positive'` Modo de brillo, configuración base 'both'

Porcentaje de ajuste de brillo en la imagen.

'`brightness_percent`' ajusta la intensidad del brillo, y '`mode`' controla si se incrementa ('positive'), se reduce ('negative'), o ambos ('both').

## **Contraste**

`v_contrast_percentage = 40`

Porcentaje de ajuste de contraste. Este valor incrementa o reduce el contraste general de la imagen,

haciendo las áreas claras más claras y las oscuras más oscuras.

## **Cropping (recorte)**

`crop_perc = 0.5` Porcentaje de recorte, entre 0 y 0.5

Porcentaje de recorte aplicado alrededor de la imagen.

Este valor controla cuánto de la imagen se recorta, donde 0 representa ningún recorte y 0.5 el máximo recorte permitido.

## **Cutout**

`num_squ = 5`

cutout\_per = 15

El filtro "cutout" añade oclusiones (cuadrados negros) sobre la imagen.

'num\_squ' es el número de cuadrados, mientras que 'cutout\_per' es el tamaño de los cuadrados como porcentaje del área total de la imagen.

### **Desenfoque selectivo de Gauss**

threshold\_gauss = 0.9

Umbral que determina qué áreas de la imagen se desenfocan selectivamente.

Áreas con contraste por debajo de este umbral se desenfocarán.

### **Transformación elástica**

alph = 100

sigm = 10

Parámetros para la transformación elástica de la imagen.

'alph' controla la intensidad de la deformación, mientras que 'sigm' define la suavidad de la distorsión.

### **Exposición**

exp\_perc = 50

Porcentaje de ajuste de la exposición de la imagen.

Un valor más alto aumenta la luminosidad general de la imagen.

### **Flipping (volteo)**

flip\_option = "horizontal"    Puede ser "horizontal", "vertical" o "both"

Controla la dirección del volteo de la imagen.

'horizontal' voltea la imagen de izquierda a derecha, 'vertical' de arriba a abajo, y 'both' realiza ambas.

### **Escala de grises**

g\_scale = 100

Porcentaje de conversión a escala de grises.

Un valor de 100 convierte completamente la imagen a escala de grises, mientras que un valor menor mezcla color y escala de grises.

### **Filtro Hue**

hue\_val = 150

Valor de ajuste del tono de color de la imagen (Hue).

Este valor controla qué tan desplazado está el tono de color de la imagen, en un rango de 0 a 180 grados.

### **Filtro de ruido**

noise\_val = 0.25    Porcentaje de ruido, entre 0 y 0.25

Porcentaje de ruido tipo "sal y pimienta" que se agrega a la imagen.

Este valor simula ruido en la imagen para hacerla más robusta ante variaciones.

### **Rotación**

r\_angle = 15

Ángulo en grados para rotar la imagen.

Un valor positivo rota la imagen en sentido horario, mientras que un valor negativo la rota en sentido antihorario.

### Saturación

`saturation_percent = 80`

Porcentaje de ajuste de saturación de color en la imagen.

Un valor alto hace que los colores sean más vibrantes, mientras que un valor bajo los desatura.

### Filtro Sepia

```
custom_filter = np.array([[0.393, 0.769, 0.189],    Sepia para el canal Rojo  
                          [0.349, 0.686, 0.168],    Sepia para el canal Verde  
                          [0.272, 0.534, 0.131]])    Sepia para el canal Azul
```

`filter_name = "sepia"`

Filtro personalizado basado en una matriz de transformación que aplica un efecto Sepia a la imagen.

El filtro transforma los valores RGB para darle un aspecto cálido y envejecido.

### Sobel

`apply_sobel = True`

Activa o desactiva el filtro Sobel, que se utiliza para detectar bordes en la imagen calculando el gradiente de intensidad de los píxeles en las direcciones x e y.

### Filtros mixtos

```
mix1 = True    blur-grayscale  
mix2 = True    contrast-brightness  
mix3 = True    saturation-brightness
```

Filtros mixtos que combinan múltiples efectos.

'mix1' combina desenfoque y escala de grises, 'mix2' combina contraste y brillo, y 'mix3' ajusta saturación y brillo simultáneamente. Estos se los ocupa para poder llegar a los 20 filtros para sets ampliados de 4000 imágenes. Para agregar más se debe modificar el código.

## Filtros

### 1. `apply_blur_filter`

- Descripción: Aplica un desenfoque gaussiano a la imagen, suavizando detalles menores en función del valor de sigma.
- **Configuración actual:** `blur_sigma = 2.5`
- **Sugerencia:** Mantener valores entre 1.0 y 5.0 para un desenfoque efectivo sin perder demasiada información.

### 2. `apply_canny_filter`

- Descripción: Detecta bordes en la imagen utilizando el filtro Canny, basado en dos umbrales que determinan los contornos detectados.
- **Configuración actual:** `thres1 = 50, thres2 = 75`
- **Sugerencia:** Ajustar los umbrales entre 50 y 150 para optimizar la detección de bordes según el nivel de detalle deseado.

### 3. `apply_brightness_filter`

- Descripción: Ajusta el brillo de la imagen según un porcentaje dado, aumentando o disminuyendo la luminosidad de forma controlada.
- **Configuración actual:** `brightness_percent = 50`, `mode = 'positive'`
- **Sugerencia:** Usar valores entre 30 y 70 para ajustes sutiles. El modo puede ser 'positive', 'negative' o 'both'.

### 4. `crop_image`

- Descripción: Recorta la imagen en función de una caja delimitadora y añade un margen adicional definido por un porcentaje de recorte.
- **Configuración actual:** `crop_perc = 0.5`
- **Sugerencia:** Usar entre 0.1 y 0.5 para incluir suficiente área alrededor del objeto recortado.

### 5. `apply_cutout_filter`

- Descripción: Crea máscaras de cuadrados negros en áreas aleatorias de la imagen, eliminando detalles en función del porcentaje y el número de cuadrados.
- **Configuración actual:** `cutout_per = 15`, `num_squ = 5`
- **Sugerencia:** Aplicar entre 5 y 15 cuadrados con tamaños entre 10% y 20% de la imagen para una oclusión efectiva.

### 6. `apply_selective_gaussian_blur`

- Descripción: Aplica un desenfoque gaussiano selectivo en áreas de bajo contraste, definido por un umbral de contraste.
- **Configuración actual:** `threshold_gauss = 0.9`
- **Sugerencia:** Ajustar el umbral entre 0.5 y 1.0 para controlar el nivel de suavización en áreas de bajo contraste.

### 7. `elastic_transform`

- Descripción: Realiza una deformación elástica en la imagen, distorsionando su estructura con parámetros de intensidad y suavidad.
- **Configuración actual:** `alph = 100`, `sigm = 10`
- **Sugerencia:** Usar valores de alpha entre 50 y 200, y sigma entre 5 y 15 para obtener distorsiones naturales.

### 8. `apply_exposure_filter`

- Descripción: Ajusta la exposición de la imagen, haciéndola más brillante u oscura según el porcentaje dado.

- **Configuración actual:** exp\_perc = 50
- **Sugerencia:** Mantener entre 30% y 70% para lograr una exposición equilibrada sin pérdida excesiva de detalles.

## 9. apply\_flip\_image

- Descripción: Invierte la imagen en sentido horizontal, vertical, o en ambas direcciones.
- **Configuración actual:** flip\_option = "horizontal"
- **Sugerencia:** Usar "horizontal" o "both" para imágenes orientadas horizontalmente y "vertical" para imágenes alargadas.

## 10. first\_apply\_grayscale\_filter

- Descripción: Convierte la imagen en escala de grises según un porcentaje especificado, mezclando la versión original y la de grises.
- **Configuración actual:** g\_scale = 100
- **Sugerencia:** Mantener entre 50% y 100% para lograr una reducción del impacto del color mientras se preservan detalles relevantes.

## 11. apply\_hue\_filter

- Descripción: Ajusta el tono (hue) de la imagen desplazando aleatoriamente el valor de tono dentro del rango especificado en grados.
- **Configuración actual:** hue\_val = 150
- **Sugerencia:** Usar valores entre 50 y 150 para variaciones de color naturales sin afectar drásticamente el balance de colores.

## 12. apply\_noise\_filter

- Descripción: Agrega ruido de tipo "sal y pimienta" a la imagen en función del porcentaje de ruido especificado.
- **Configuración actual:** noise\_val = 0.25
- **Sugerencia:** Mantener el ruido entre 0.1 y 0.3 para crear imperfecciones leves sin afectar la legibilidad de la imagen.

## 13. first\_rotate\_image

- Descripción: Rota la imagen en un ángulo específico, manteniendo su tamaño original y ajustando el centro de rotación.
- **Configuración actual:** r\_angle = 15
- **Sugerencia:** Usar ángulos entre 10° y 45° para rotaciones efectivas sin alterar drásticamente la composición de la imagen.

#### 14. `apply_saturation_filter`

- Descripción: Ajusta la saturación de la imagen, incrementando o disminuyendo los niveles de color en función del porcentaje dado.
- **Configuración actual:** `saturation_percent = 80`
- **Sugerencia:** Usar valores entre 50% y 100% para mantener colores vibrantes pero controlados.

#### 15. `apply_custom_filter` (Sepia)

- Descripción: Aplica un filtro personalizado de transformación de color usando una matriz de transformación, en este caso, el filtro Sepia.
- **Configuración actual:** `filter_name = "sepia"`
- **Sugerencia:** Mantener la matriz de transformación para aplicar efectos específicos como Sepia, ideal para lograr un estilo cálido y antiguo.

#### 16. `apply_sobel_filter`

- Descripción: Utiliza el filtro Sobel para detectar bordes en la imagen, calculando gradientes en las direcciones x e y.
- **Configuración actual:** `apply_sobel = True`
- **Sugerencia:** Aplicar el filtro Sobel cuando se necesite identificar contornos claros en imágenes con detalles definidos.

#### 17. `apply_blur_grayscale_mix` (Mix1)

- Descripción: Aplica un desenfoque Gaussiano seguido de una conversión a escala de grises, reduciendo detalles y colores simultáneamente.
- **Configuración actual:** `mix1 = True`
- **Sugerencia:** Usar cuando se necesite suavizar y simplificar una imagen, ideal para análisis de bordes o texturas.

#### 18. `apply_contrast_brightness_mix` (Mix2)

- Descripción: Ajusta el contraste y el brillo de la imagen simultáneamente, mejorando tanto las áreas oscuras como las luminosas.
- **Configuración actual:** `mix2 = True`
- **Sugerencia:** Aplicar en imágenes con un rango dinámico bajo para mejorar la visibilidad general sin sobresaturar o deslucir detalles.

#### 19. `apply_saturation_brightness_mix` (Mix3)

- Descripción: Aumenta la saturación de los colores y ajusta el brillo de la imagen, mejorando la intensidad del color y la iluminación.
- **Configuración actual:** `mix3 = True`



- **Sugerencia:** Usar en imágenes que necesiten resaltar colores y mejorar la claridad visual, sin comprometer el balance de color.

## 20. `apply_exposure_filter`

- Descripción: Ajusta la exposición de la imagen, incrementando o disminuyendo la luminosidad según el porcentaje de exposición.
- **Configuración actual:** `exp_perc = 50`
- **Sugerencia:** Mantener valores entre 30% y 70% para obtener ajustes equilibrados sin sobreexponer ni subexponer la imagen.

# Uniando los sets separados `mixture.py`

## Parámetros

### `root_folder`

Especifica la ruta de la carpeta raíz que contiene todas las subcarpetas con imágenes y archivos `.json`.

Valores: Ruta absoluta o relativa al directorio donde se encuentran las subcarpetas.

### `output_image_dir` (diferente al output `.../data/processed_images/total_set`)

Define la carpeta donde se almacenarán todas las imágenes consolidadas después de haber sido copiadas y renombradas.

Valores: Ruta absoluta o relativa al directorio de destino de las imágenes.

### `output_json_file`

Ruta donde se guardará el archivo `.json` consolidado que contendrá las anotaciones actualizadas de todas las imágenes procesadas.

Valores: Ruta absoluta o relativa al archivo de salida.

## Ejecución del Script

Primero, el script crea la carpeta de salida si no existe. Luego, explora todas las subcarpetas en la carpeta raíz, busca archivos `.json` y sus respectivas imágenes. Las imágenes son copiadas a la carpeta de salida y renombradas secuencialmente. Las anotaciones en los archivos `.json` son actualizadas para reflejar los nuevos nombres de las imágenes. Finalmente, el archivo `.json` consolidado es guardado en la ruta especificada.

Para ejecutar este script, asegúrate de ajustar correctamente las rutas de los parámetros. Después, simplemente ejecuta el siguiente comando en la terminal:

```
bash
Copy code
python mixture.py
```

El script procesará todas las imágenes y creará el archivo .json consolidado en el directorio de salida especificado.

## **Requisitos**

Este script requiere Python 3.6 o superior y usa las bibliotecas estándar `os`, `shutil`, y `json` para la manipulación de archivos y .json.