Image Processing

Yousef Ahmed Bakier

GITHUB: REPO

Y-Baker/Image-Processing

Tech Stack

Python

Programming Languague

streamlit

GUI Python Lib (Web Based)

opencv

Lib bindings for computer vision

INFO

Image Processing Toolbox 🖼️

This application provides a comprehensive set of image processing tools organized into multiple categories for advanced image manipulation and analysis.

🔧 Available Filter Categories

- **Point Operations:** Brightness, contrast, gamma correction
- **Color Processing:** Grayscale conversion, color channel manipulation
- **Histogram Operations:** Equalization, adaptive methods
- **Neighborhood Processing:** Blur, sharpening, noise reduction
- **Image Restoration:** Noise addition/removal, inverse filtering
- **Segmentation:** Thresholding, clustering methods
- **Edge Detection:** Sobel, Canny, Laplacian
- **Mathematical Morphology:** Erosion, dilation, opening, closing
- **Advanced Filters:** Bilateral, non-local means, guided filter
- **Transformations:** Rotation, scaling, perspective

🚀 Getting Started

1. **Upload an image** using the file uploader in the sidebar.
2. **Select a filter category** and operation.
3. **Adjust parameters** as needed.
4. **Apply the filter** and see real-time results.
5. **Chain multiple filters** to achieve desired effects.
6. **Download** your processed image.

Installation

💻 Installation

1. Clone the repository:

```
git clone https://github.com/Y-Baker/Image-Processing.git
```

2. Install the required packages:

```
pip install -r requirements.txt
```

3. Run the application:

```
streamlit run app.py
```

4. Open your web browser and navigate to `http://localhost:8501` to access the application.
5. Upload an image and start processing!

📦 Dependencies

- `numpy`
- `opencv-python`
- `streamlit`
- `Pillow`

Available Filter

🧠 Available Services

Category	Functions	Parameters
Point Operation	Brightness adjustment	brightness_value (-100 to 100)
	Contrast adjustment	contrast_value (0.5 to 3.0)
	Complement	-
	Power law transformation	gamma (0.1 to 3.0)
	Log transformation	c (1 to 255)
Color Image Operation	Convert to Grayscale	-
	Enhance Contrast	alpha (1.0 to 3.0)
	Color Channel Split	channel (R, G, B)
	HSV Color Space	-
	Color Balance	red_gain, green_gain, blue_gain (0.5 to 2.0)
Image Histogram	Show Histogram	-
	Histogram Equalization	-
	Adaptive Histogram Equalization	clip_limit (1.0 to 4.0)
Neighborhood Processing	Mean Filter	kernel_size (3 to 15)
	Median Filter	kernel_size (3 to 15, step 2)
	Gaussian Blur	kernel_size (3 to 15, step 2), sigma (0.5 to 5.0, step 0.1)
	Sharpening Filter	strength (0.1 to 2.0)
	Unsharp Masking	radius (1 to 5), amount (0.5 to 2.0)

Available Filter

Image Restoration	Add Salt Pepper Noise	amount (0.01 to 0.2, step 0.01)
	Remove SP Noise (Average)	-
	Remove SP Noise (Median)	-
	Remove SP Noise (Outlier)	-
	Add Gaussian Noise	mean (-0.1 to 0.1, step 0.01), std (0.01 to 0.3, step 0.01)
	Remove Gaussian Noise (Average)	-
	Inverse Filtering	regularization (0.01 to 1.0)
	Motion Blur	length (1 to 20, step 1), angle (0 to 180, step 5)
Image Segmentation	Global Thresholding	threshold (0 to 255, step 5)
	Otsu's Method	-
	Adaptive Thresholding	block_size (3 to 15, step 2), C (2 to 10, step 2)
	K-Means Clustering	k (2 to 8)
Edge Detection	Sobel	-
	Prewitt	-
	Canny	low_threshold (50 to 150), high_threshold (100 to 200)
	Laplacian	kernel_size (1 to 7, step 2)
	Zero Crossing	threshold (0.1 to 1.0)
Mathematical Morphology	Erosion	kernel_size (3 to 15), iterations (1 to 5)
	Dilation	kernel_size (3 to 15), iterations (1 to 5)
	Opening	kernel_size (3 to 15)
	Closing	kernel_size (3 to 15)
	Gradient	kernel_size (3 to 15)
	Top Hat	kernel_size (3 to 15)
	Black Hat	kernel_size (3 to 15)

Available Filter

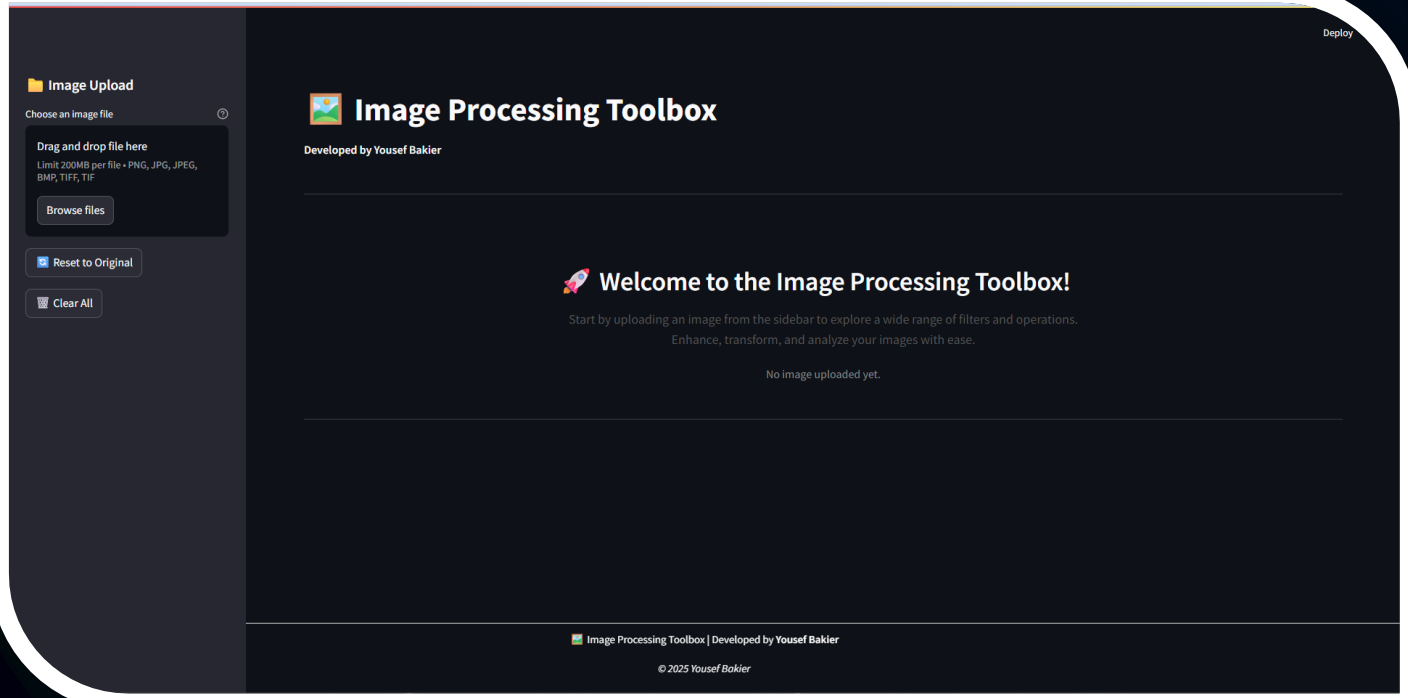
Advanced Filters	Bilateral Filter	d (5 to 15), sigma_color (10 to 100), sigma_space (10 to 100)
	Non-Local Means	h (3 to 15), template_window (7 to 21), search_window (21 to 35)
	Guided Filter	radius (1 to 8), eps (0.01 to 1.0)
	Anisotropic Diffusion	iterations (1 to 50), kappa (10 to 100)
Transformations	Rotate	angle (-180 to 180, step 5)
	Resize	width (50 to 2000), height (50 to 2000)
	Affine Transform	src_points, dst_points
	Perspective Transform	src_points, dst_points
	Translation	tx (-100 to 100), ty (-100 to 100)
	Scaling	sx (0.1 to 5.0), sy (0.1 to 5.0)

Authors:

- **Yousef Bakier** GitHub

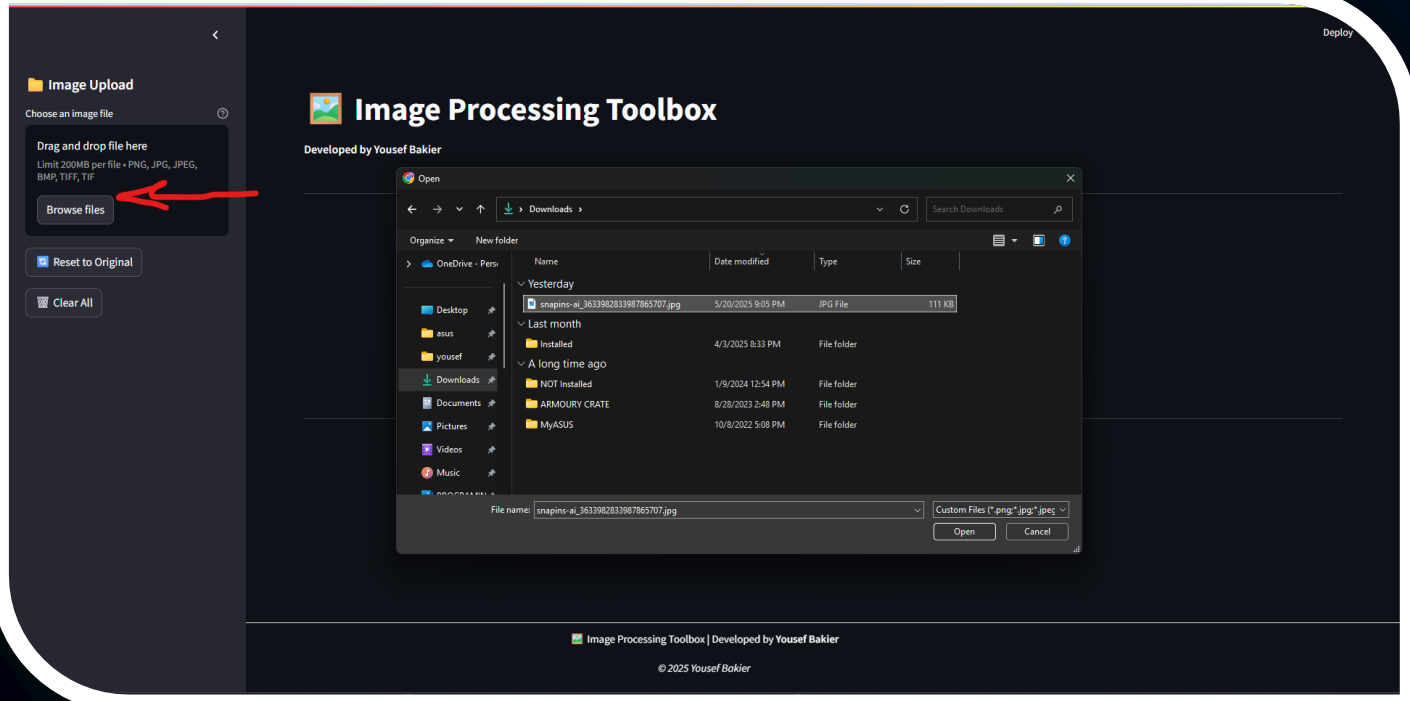
Main Screen

The start point of the project



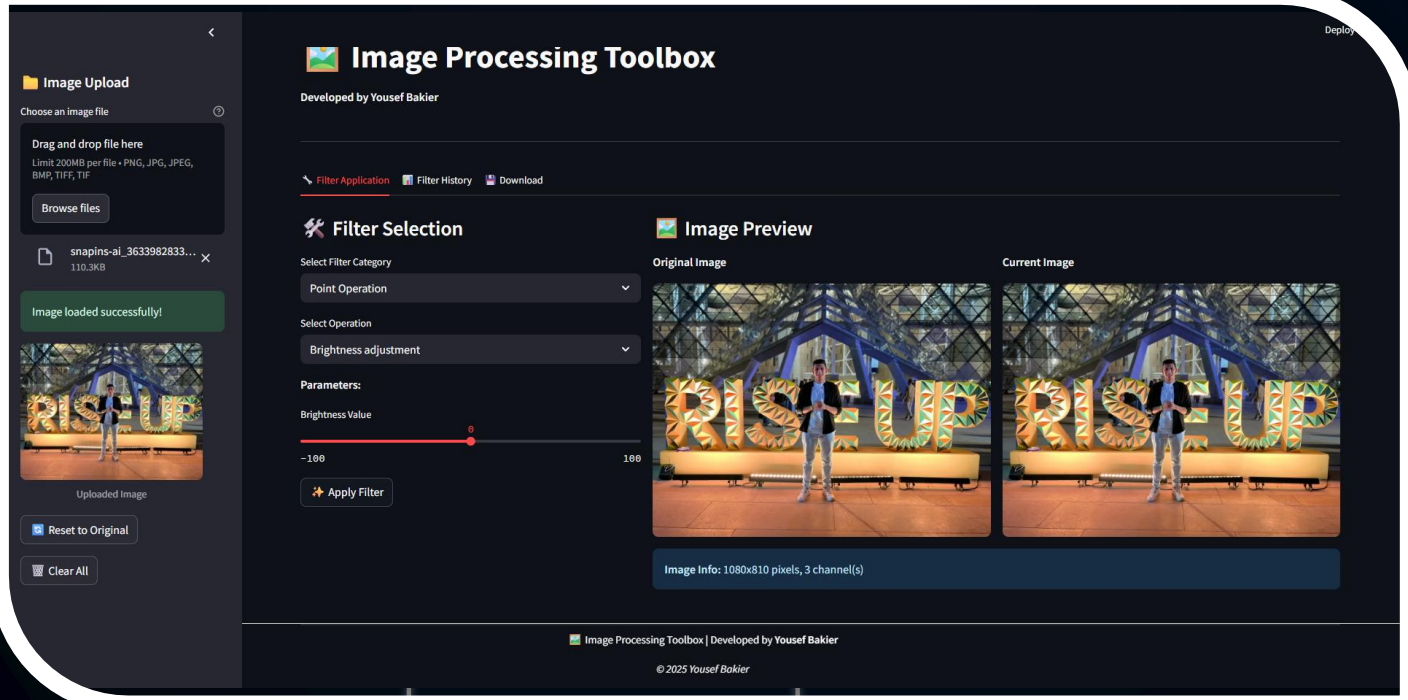
Upload Image

You can upload the image from the left slider



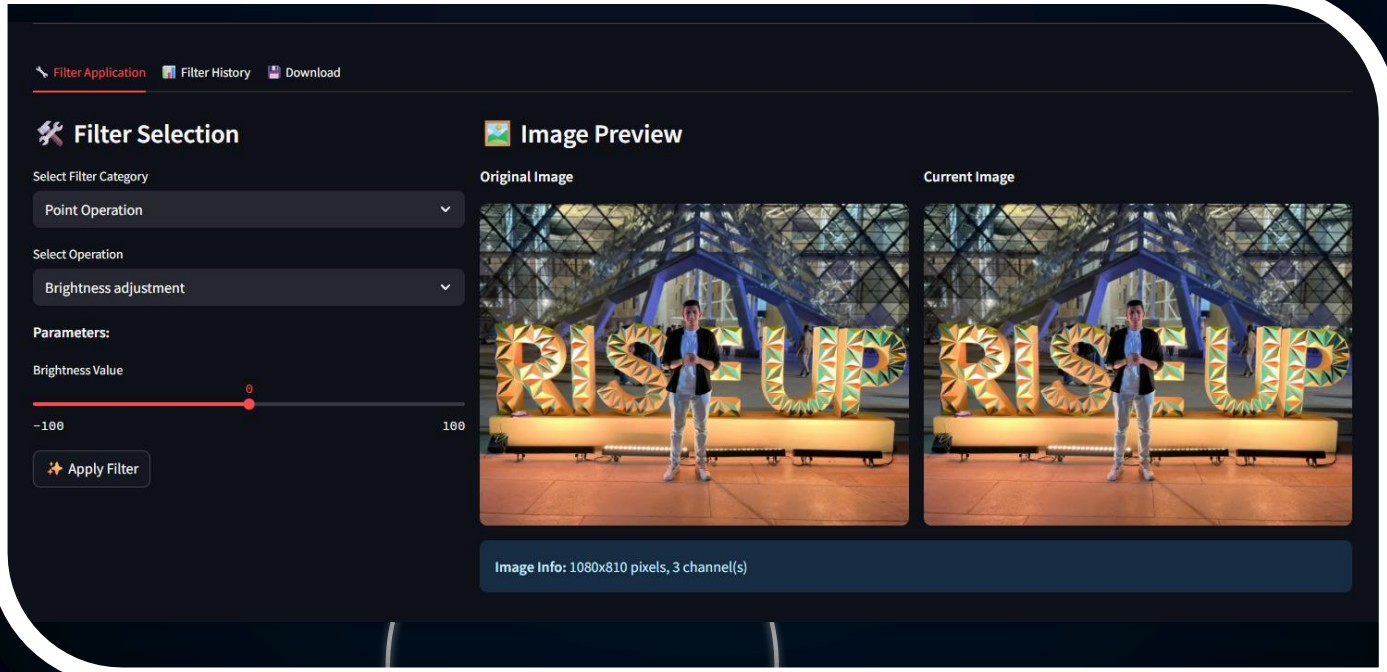
Edit Screen

After upload, you can select filter and operation and set parameters if needed to apply



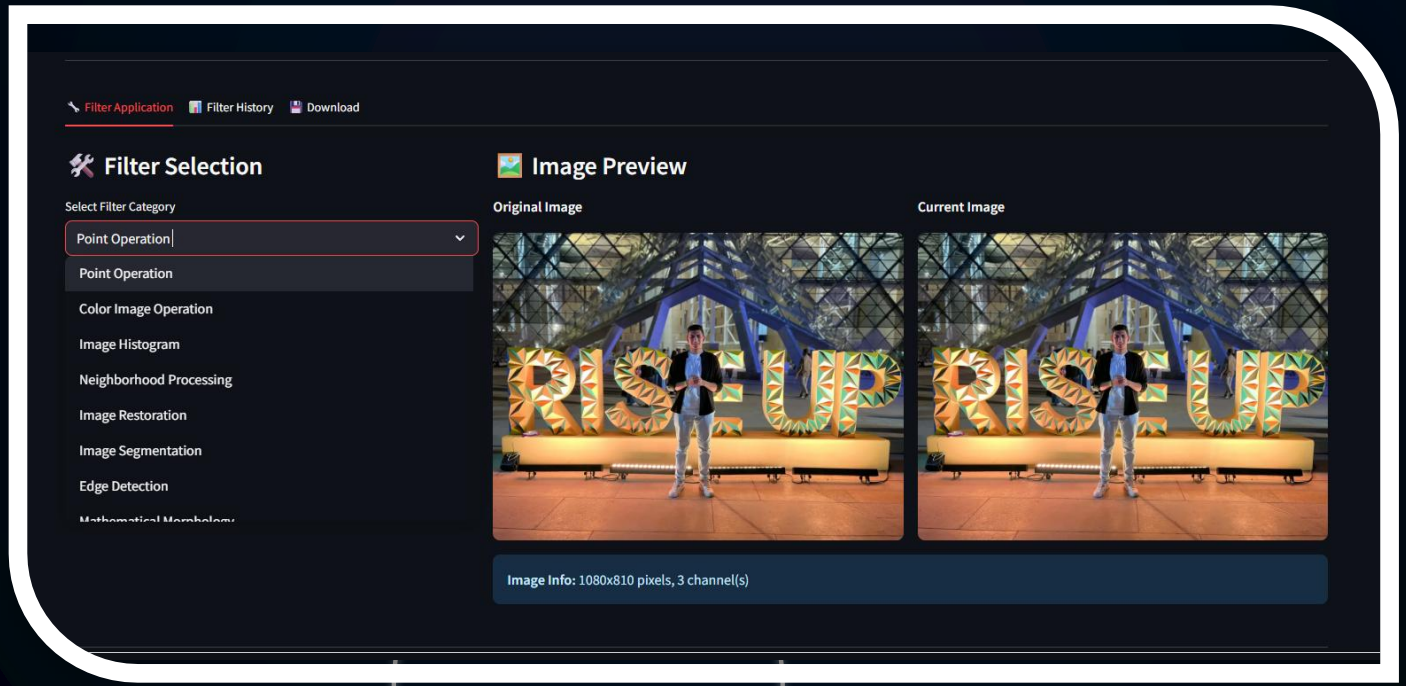
Upload Image

You can upload the image from the left slider



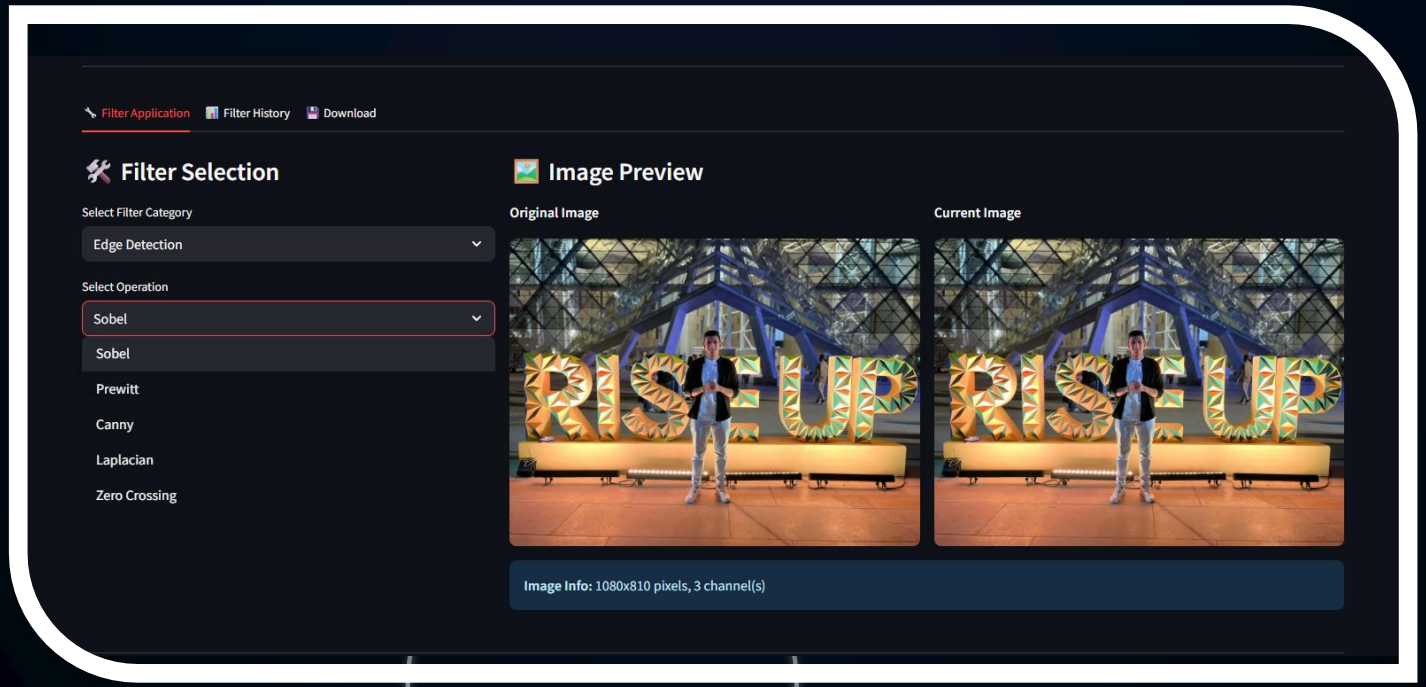
Filter Select

You can select the category for filters



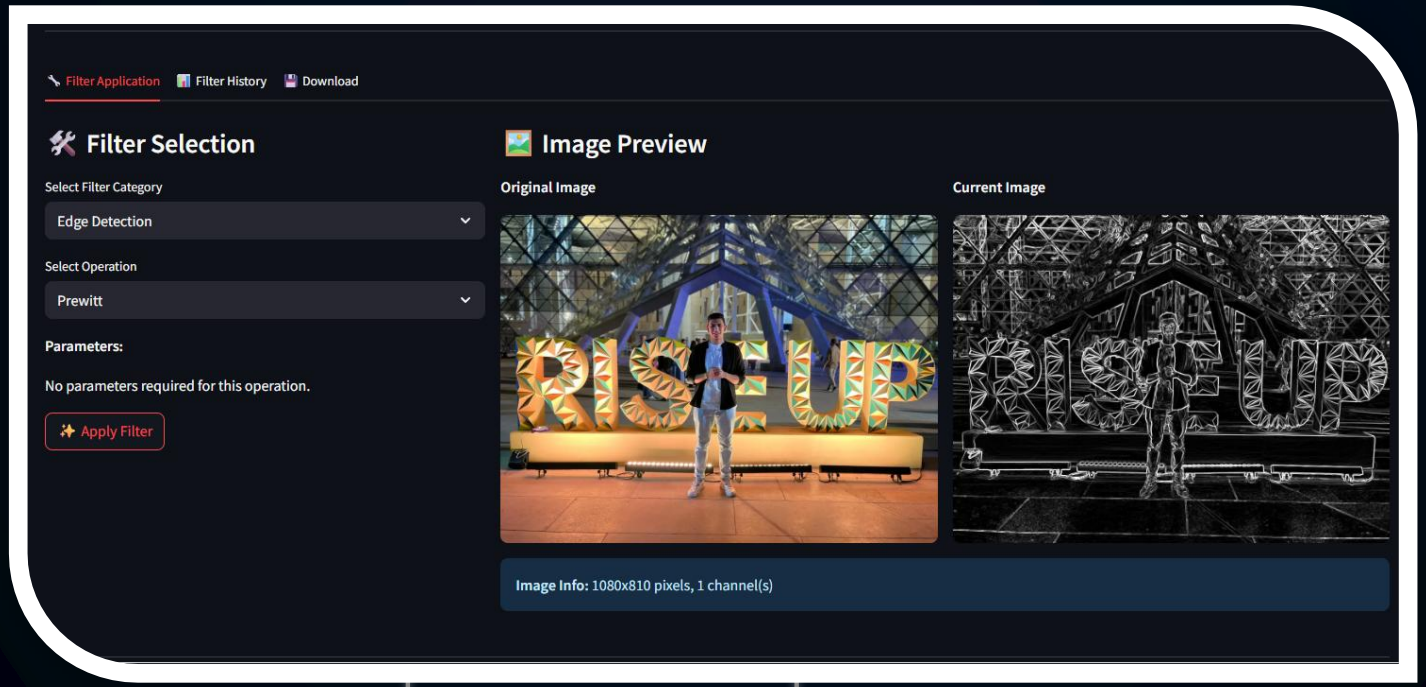
Operation Select

You can select the filter for the selected category



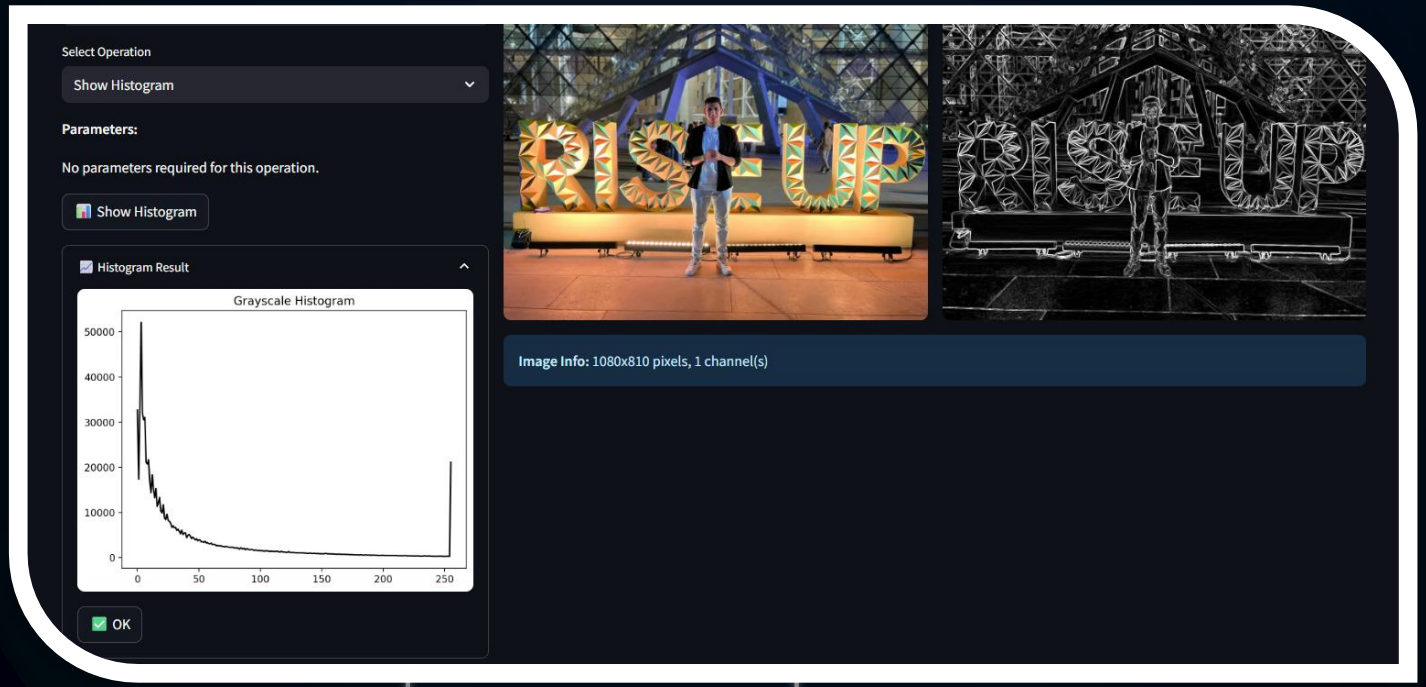
Apply Filter

Click the button Apply Filter to apply the filter

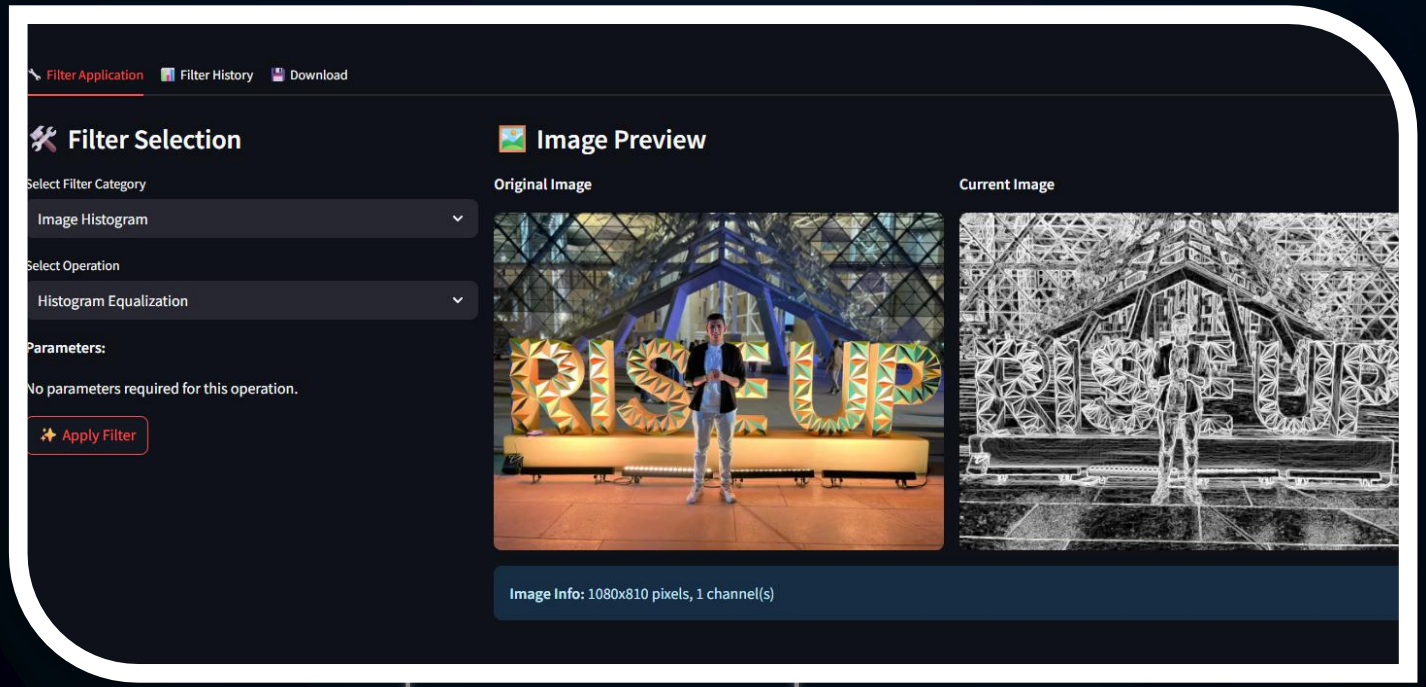


Show Histogram

You can show the histogram in Category (Image Histogram)

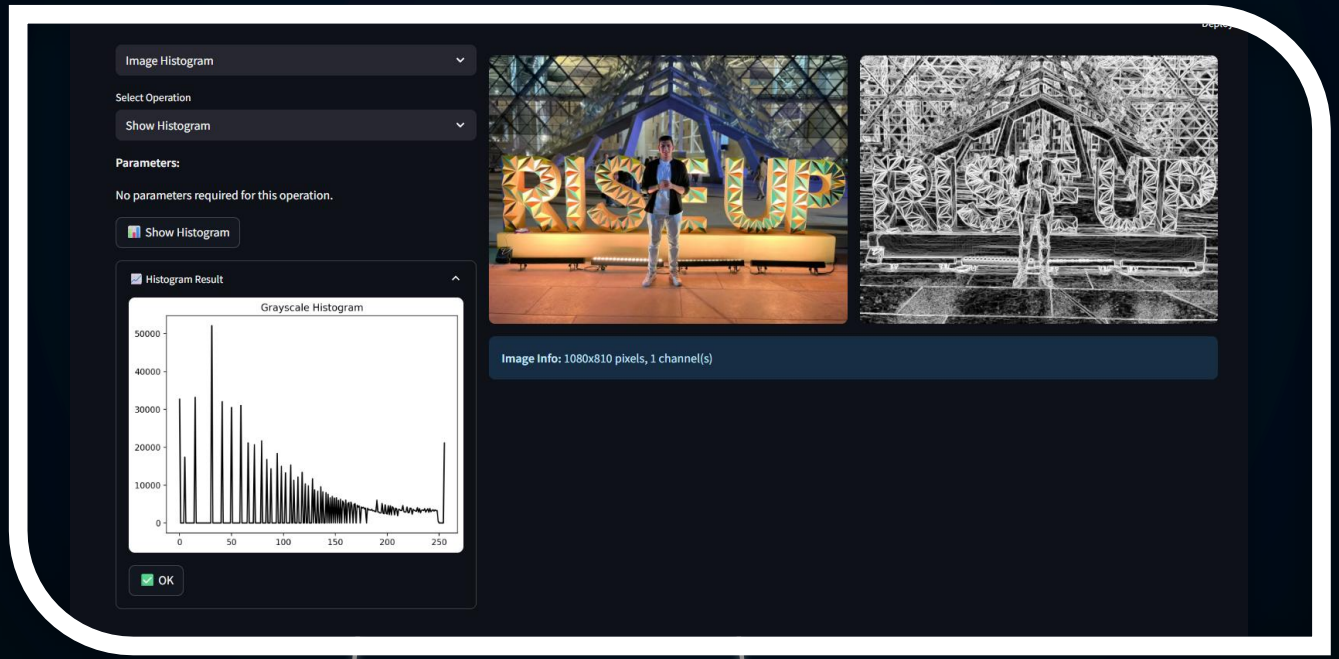


Histogram Equalization



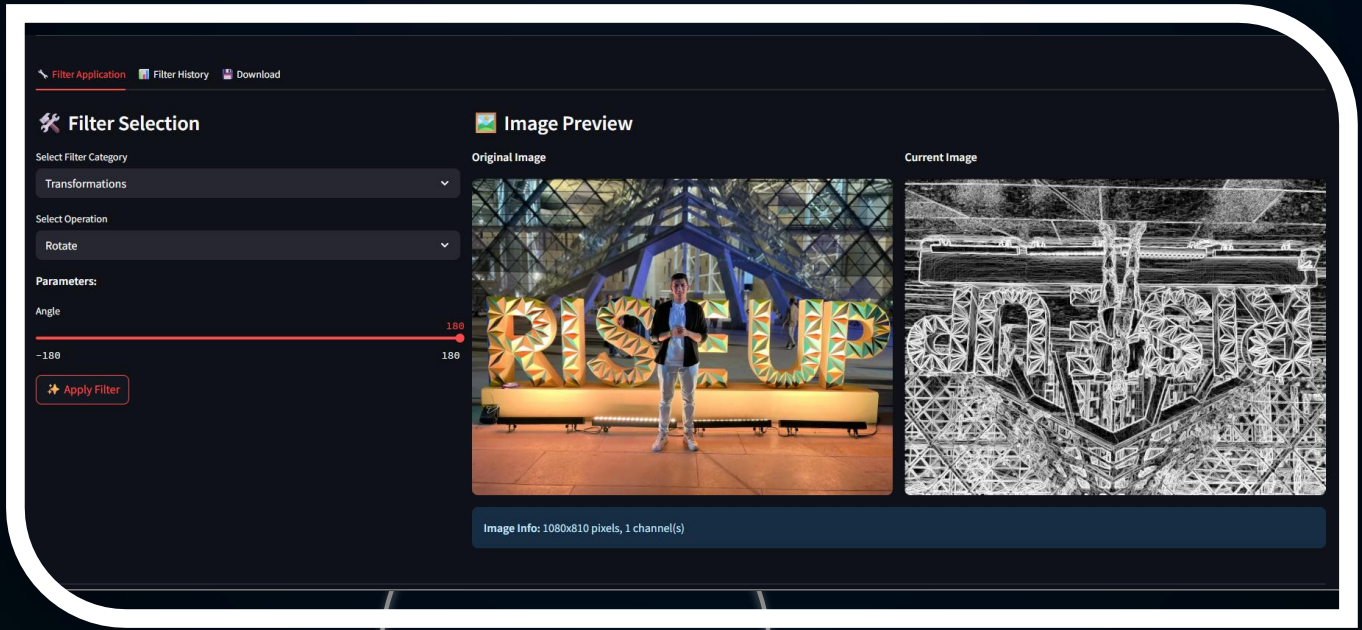
Histogram

Histogram after Histogram Equalization



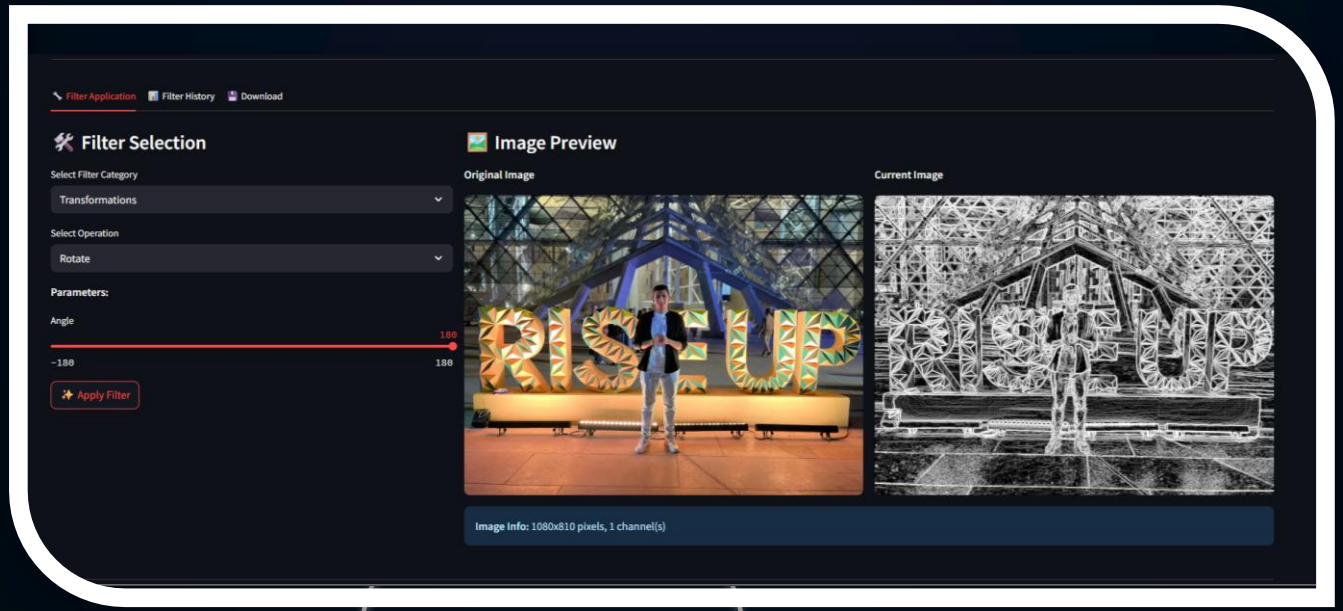
Rotate

You need to specify an angle (parameter) for rotate to apply



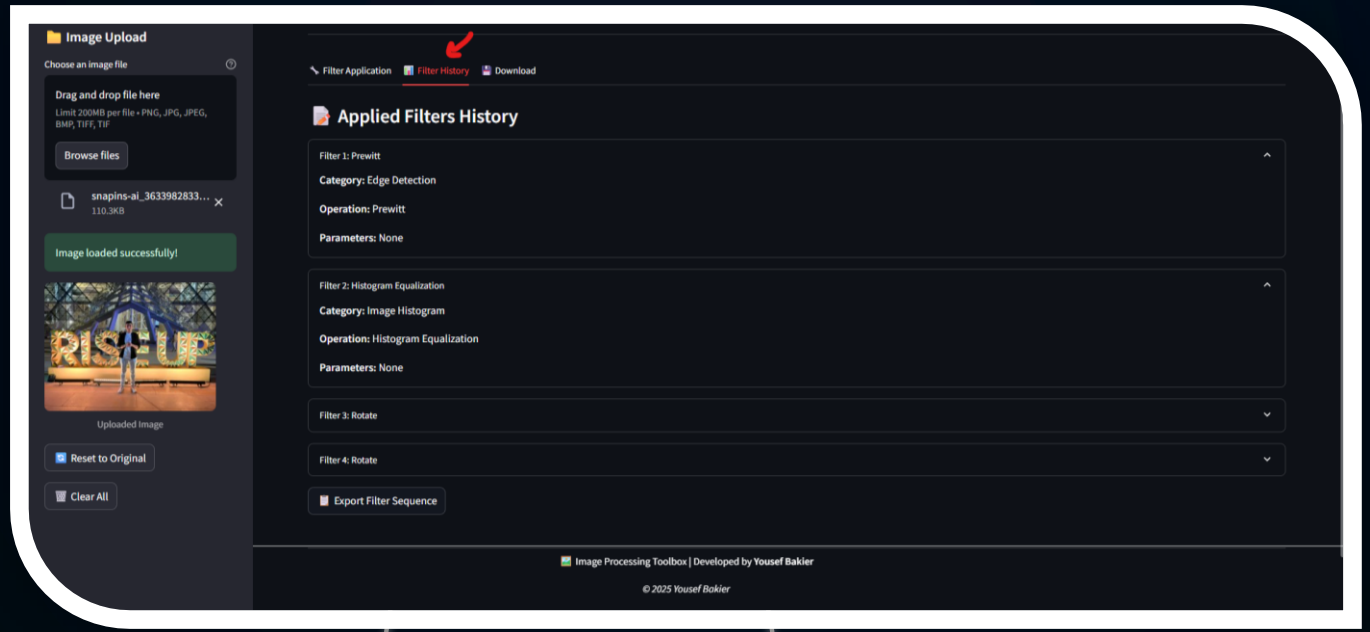
Filter Sequences

The new filter applies to the result of the previous filter



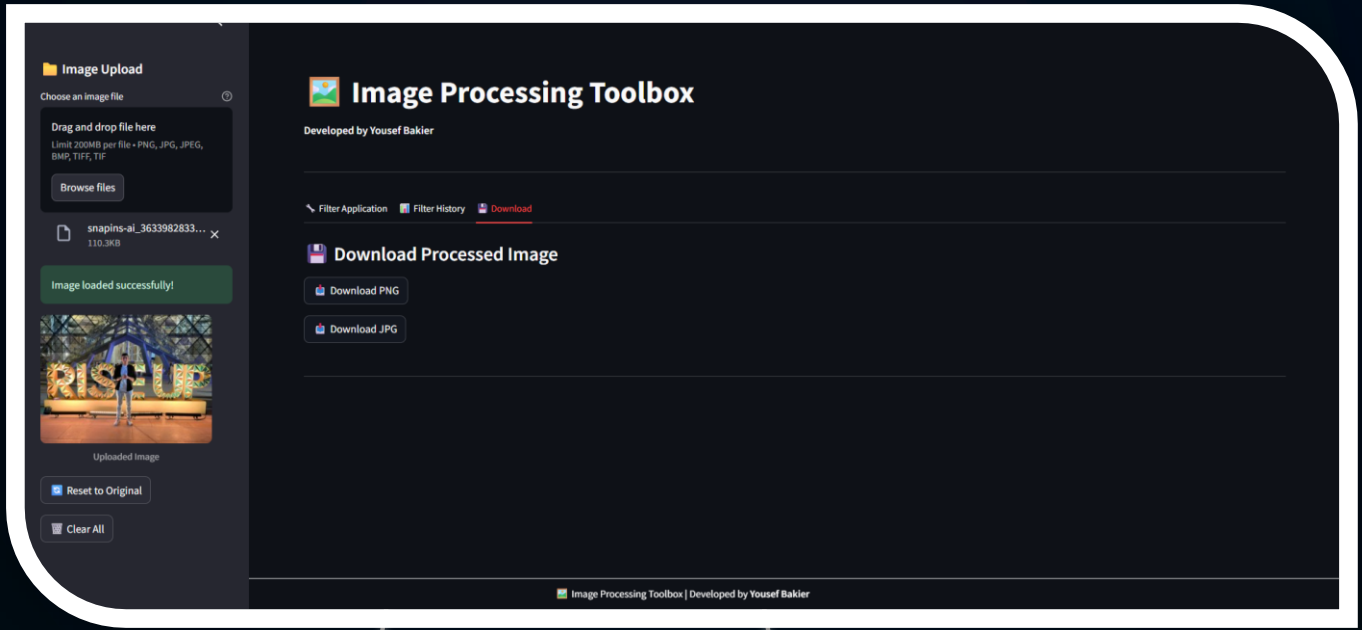
History

You can see a history of all applied filters as sequence, and you can export it to JSON object



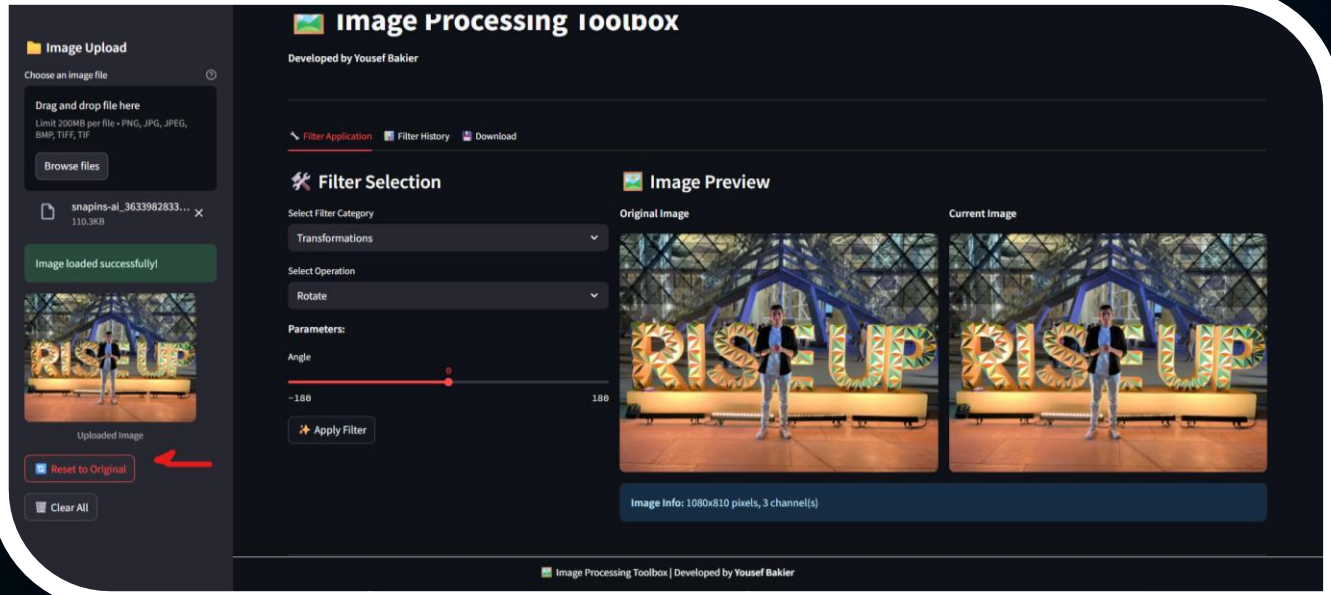
Download

You can download the image after edit as PNG or JPG



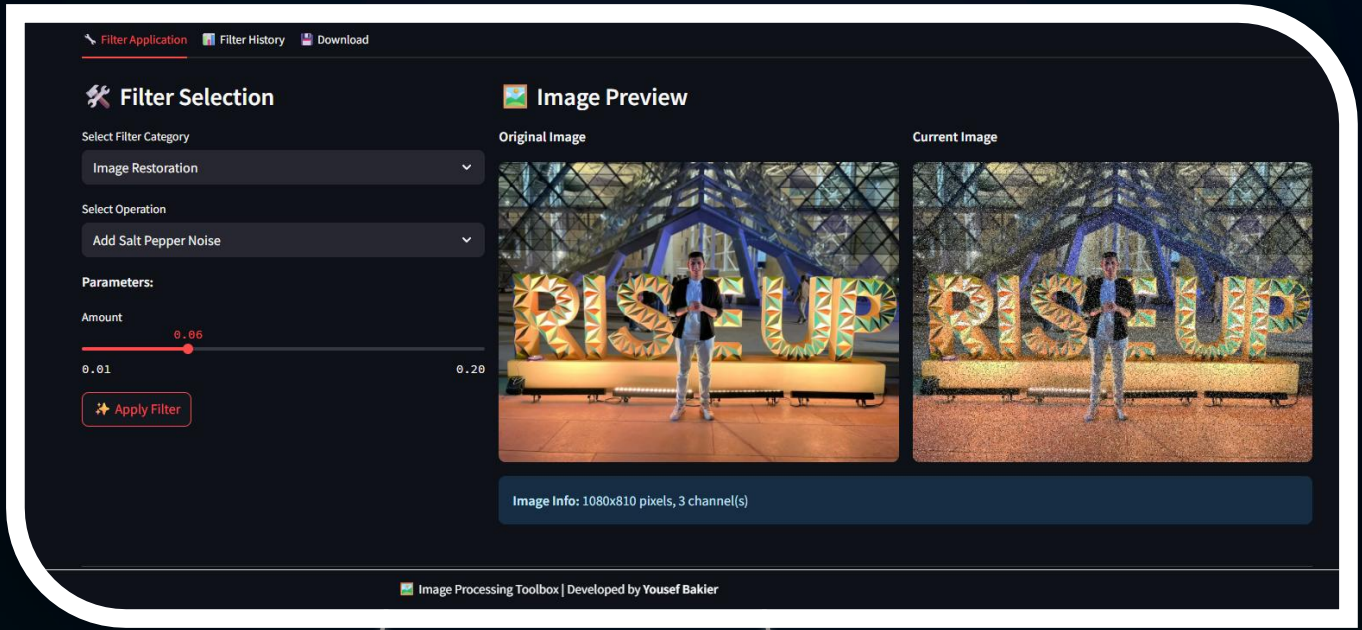
Reset Original

You can reset the image to original state in case things go wrong

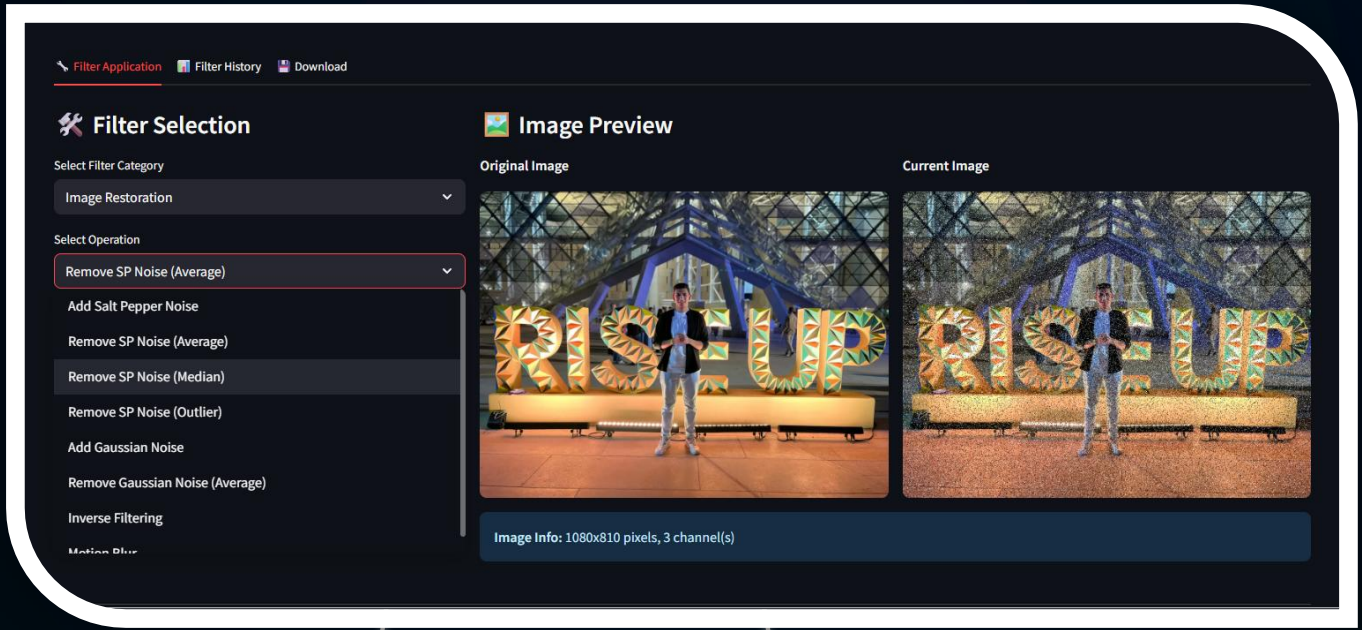


Restoration

You can apply a restoration to image for example (salt pepper)

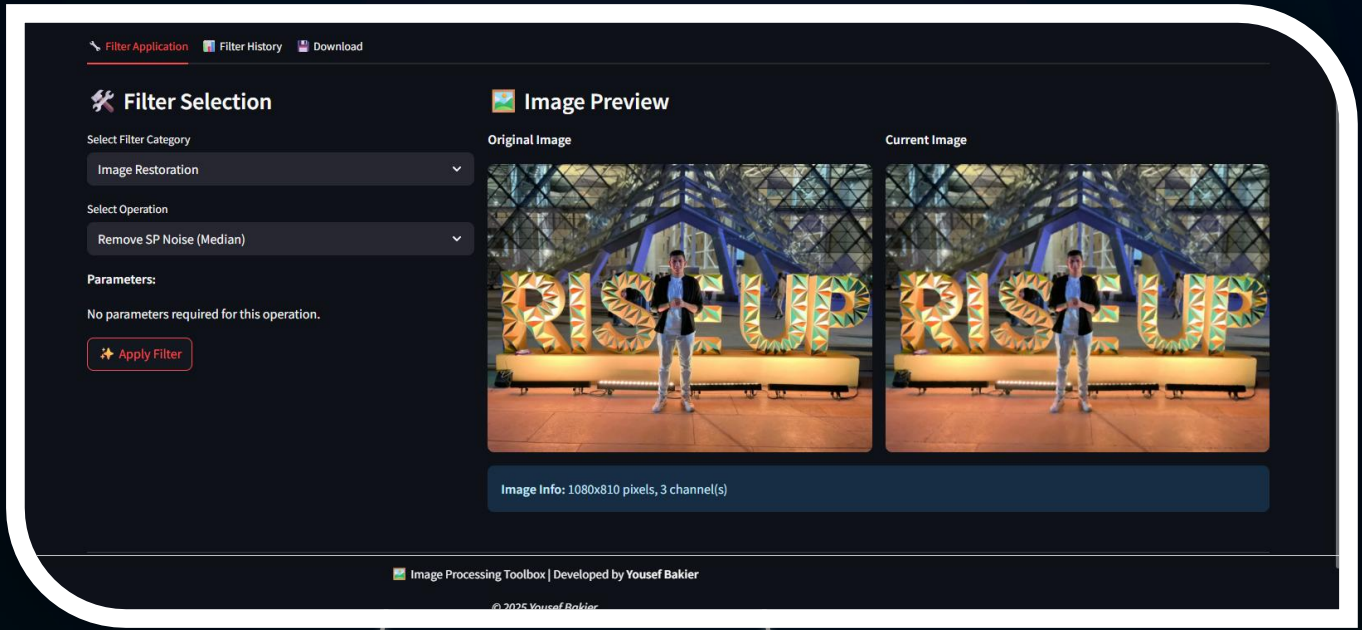


Restoration Options



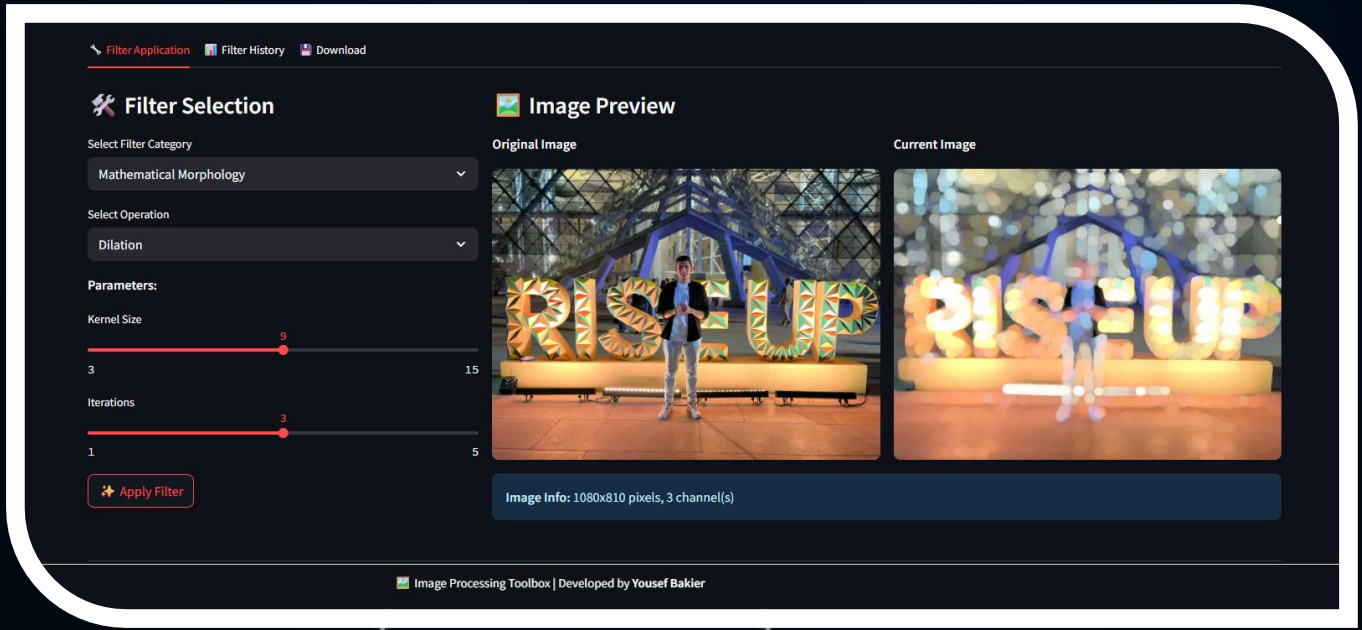
Remove Noise

You can after apply a restoration to image for example (salt pepper) use filters to remove the noise



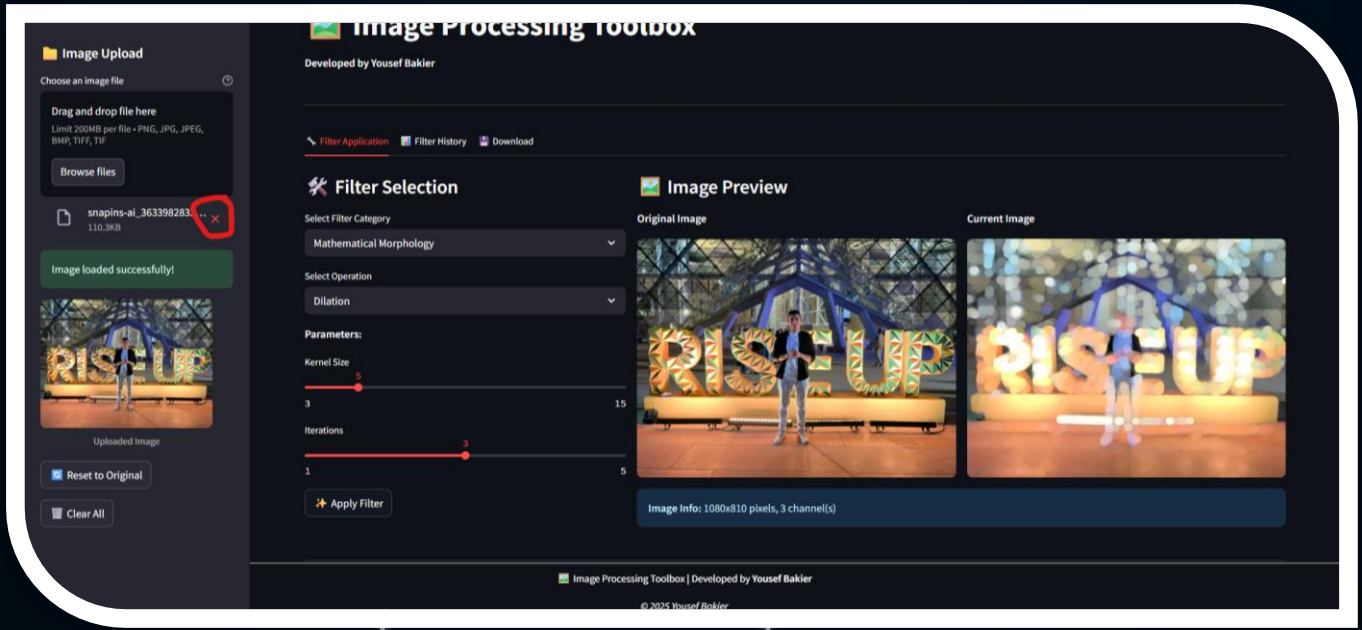
Dilation Morphology

Here you have two parameters (Kernel with step 2 and Iterations with step 1)



Remove Uploaded Image

Remove the uploaded image to upload another one



Thank You!