Computer Vision Tasks

Name	Section	B.N Number
Ahmed Salah El-Dein	1	5
Ahmad Abdelmageed Ahmad	1	8
Ahmad Mahdy Mohammed	1	9
Abdullah Mohammed Sabry	2	7

This repository is created by

In this Repository we present a variety of Image processing Techniques implemented from scratch using `Python` programming language with Numpy and Pure Python. Each Category of Algorithms is presented in its tab in the UI which we will discover next.

Our UI present a tab for each category of the implemented algorithms. We first load our image and apply the selected algorithm.

Table of contents

1. Image Processing

1.1 Adding Noise To an Image

We implemented 3 types of noise: `Uniform`, `Gaussian` and `Salt & Pepper` Noise.

Implementations Added:

1. Noise Functions (Simulation of Different Noise Types): Uniform, Gaussian and Salt & Pepper.
2. Edge Detection Techniques: Prewitt, Sobel and Roberts.
3. Image Histogram Equalization and Normalization.
4. Local and Global Thresholding
5. Transformation to Gray Scale
6. Frequency Domain Filters: Low Pass and High Pass Filters

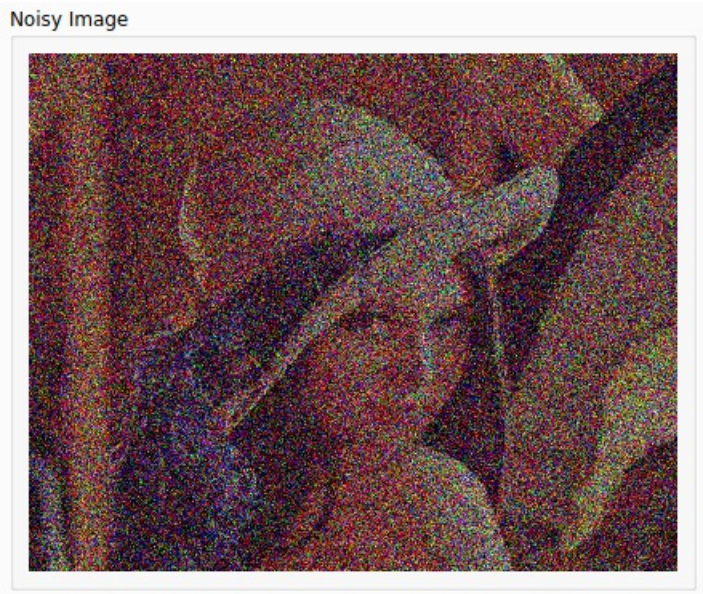
In addition to histogram and distribution curve drawing for the loaded image and the option to mix 2 input images.

Results:

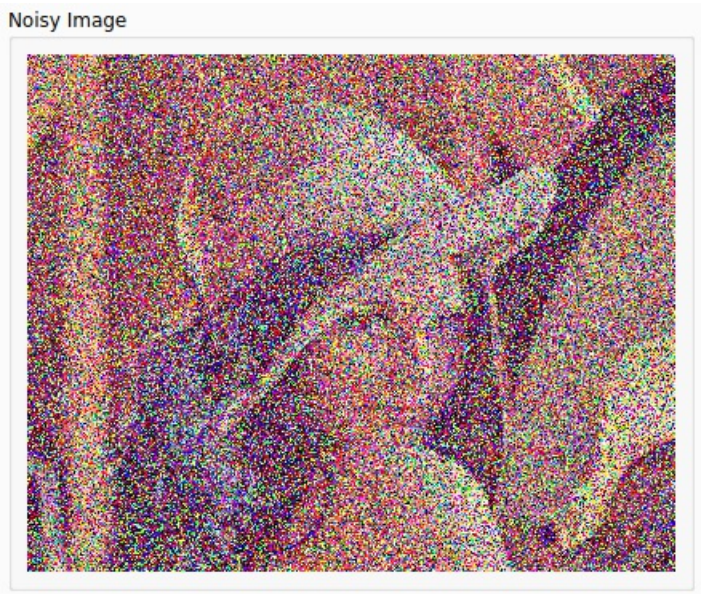
1. Noise Addition:

1.1 Uniform Noise

1.2 Gaussian Noise



1.3 Salt & Pepper Noise



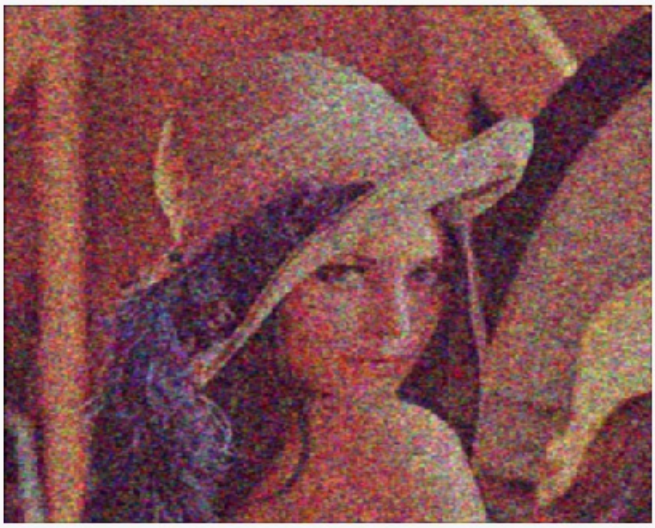
2. Noise Filtration:

2.1 Average Filter (Applied on Gaussian Noise)

Filtered Image

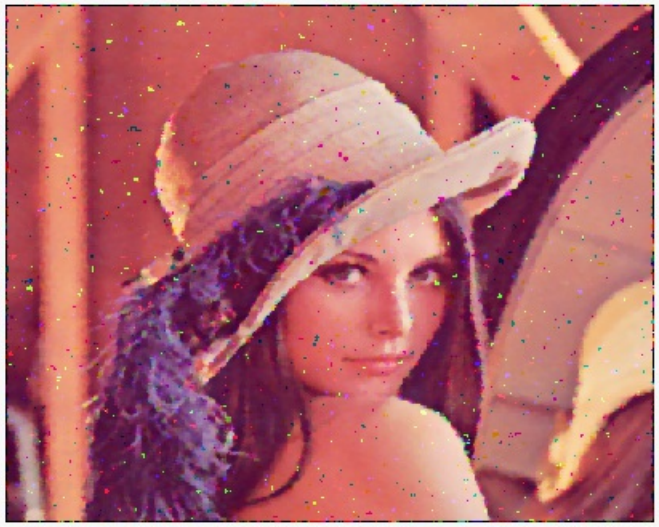
2.2 Gaussian Filter (Applied on Gaussian Noise)

Filtered Image



2.3. Median Filter (Applied on a Salt & Pepper Noisy Image)

Filtered Image



3. Edge Detection Techniques:

3.1 Sobel

Edge Detection Image

3.2 Prewitt

Edge Detection Image

3.3 Roberts

Edge Detection Image

3.4 Canny

MainWindow

Filters | Histograms | Hybrid | Hough | Active Contour

Load Image

Name: Bikesgray.jpg

Size: 640x480

Add Noise

SNR

0 0.5 1.0

Sigma

0 128 255

Apply Filter

Mask Size

3x3 5x5 7x7 9x9

Sigma

0 128 255

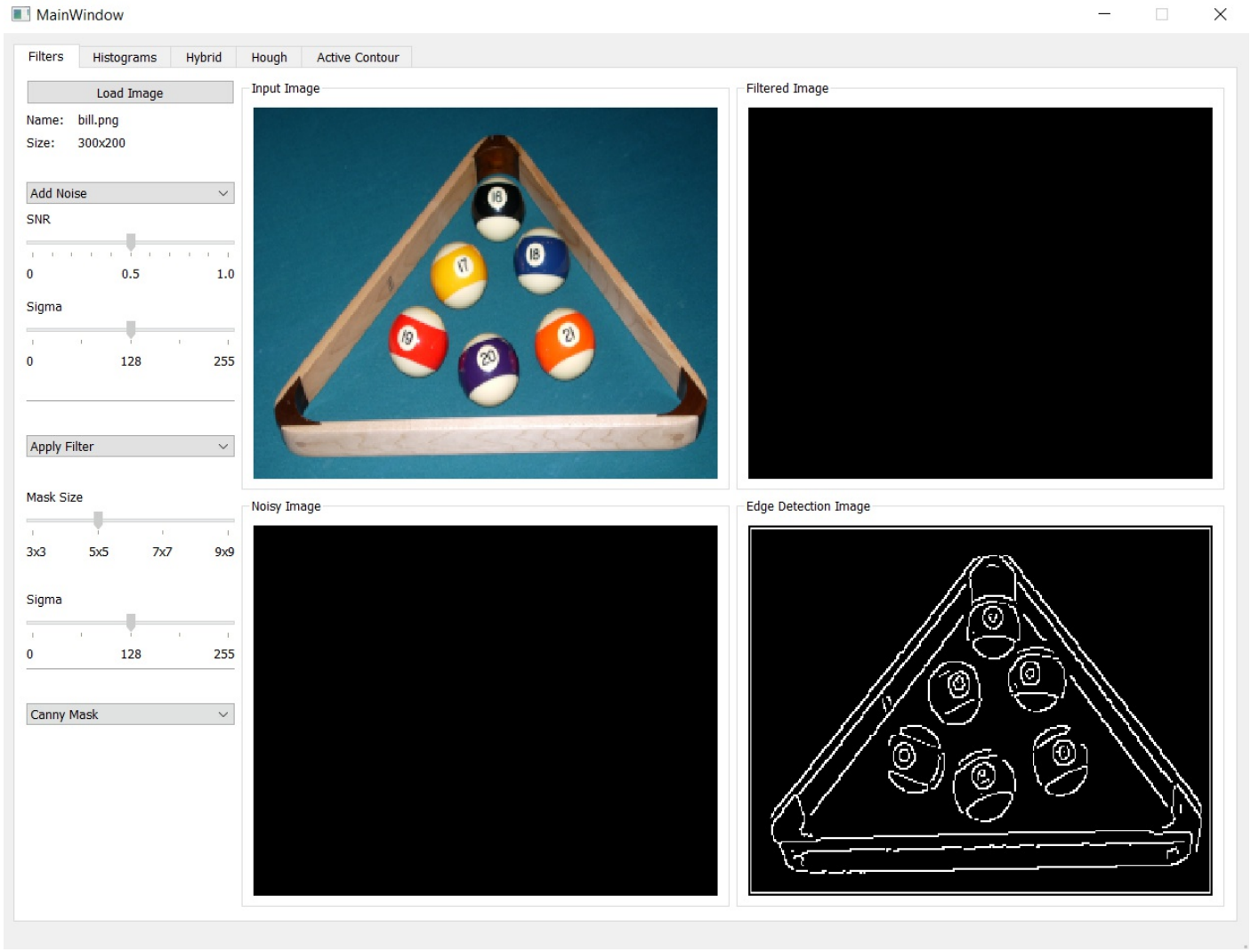
Canny Mask

Input Image

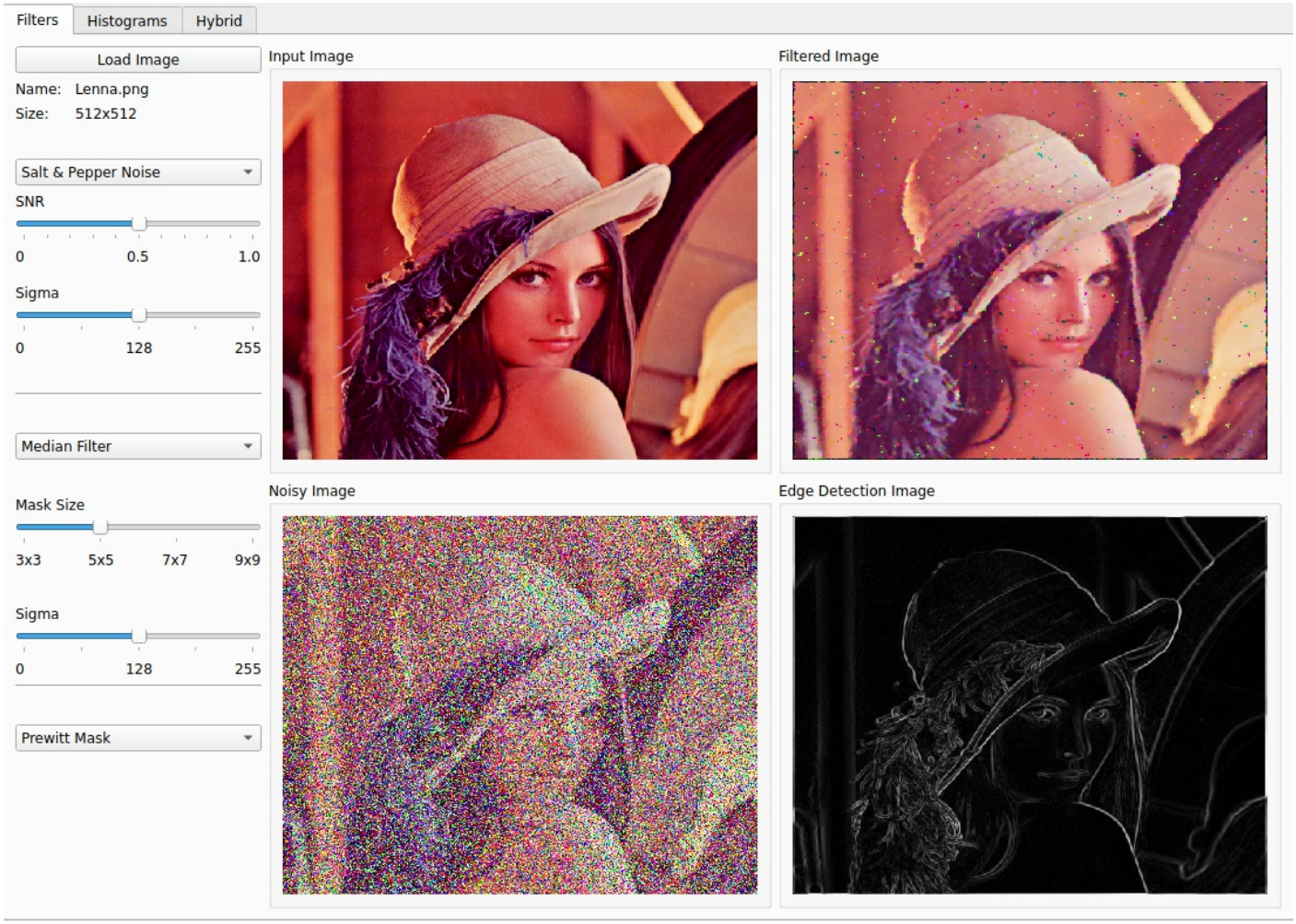
Filtered Image

Noisy Image

Edge Detection Image



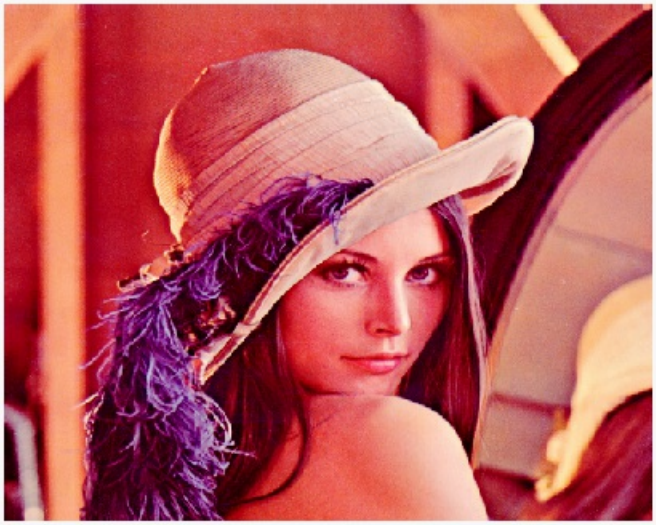
You can apply different SNR ratios and choose the Sigma of Each Algorithm implemented from the sliders added on the left, each cell is marked with its contents and the application of the change in the sliders is instant.



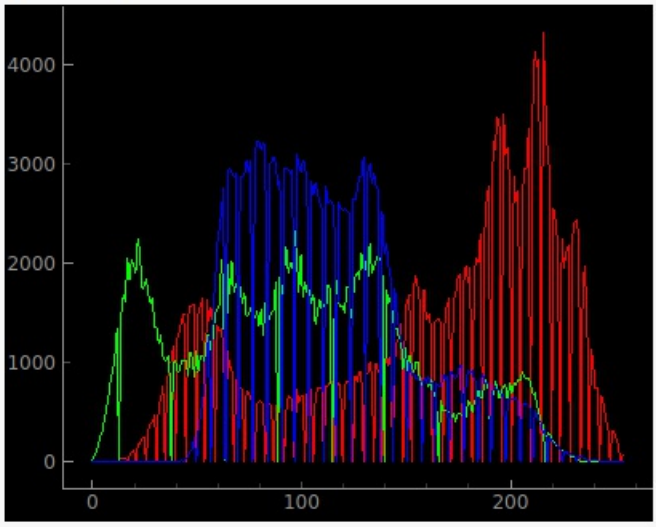
4. Histogram Equalization with input and output histograms

Input Image

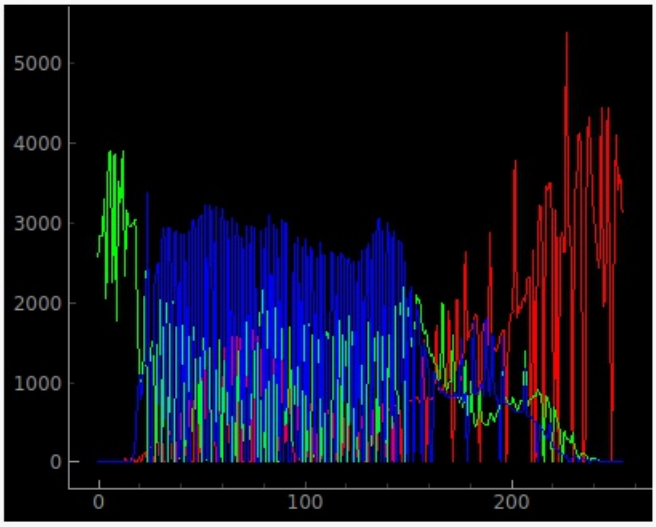
Output Image



Original Histogram



Output Histogram



5. Local and Global Thresholding

Output Image

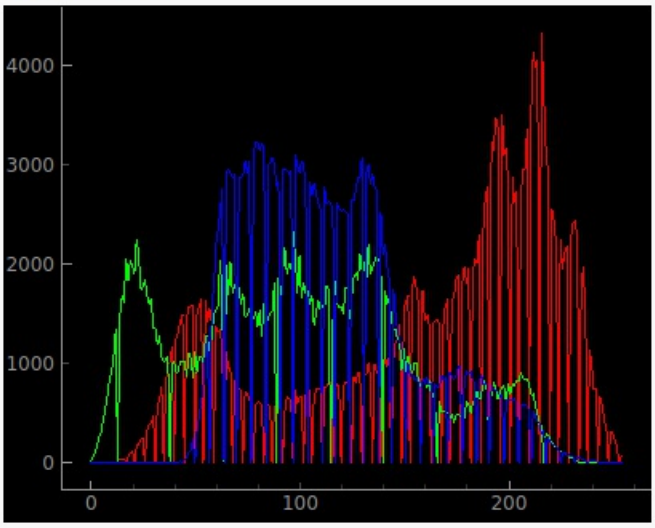
Output Image

6. Gray Scale Transformation

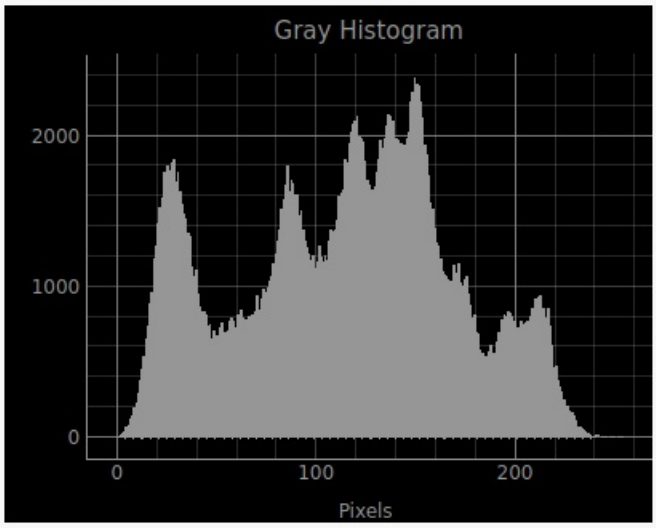
Input Image

Output Image

Original Histogram



Output Histogram



7. Frequency Domain Mixing

Input Image A

Input Image B

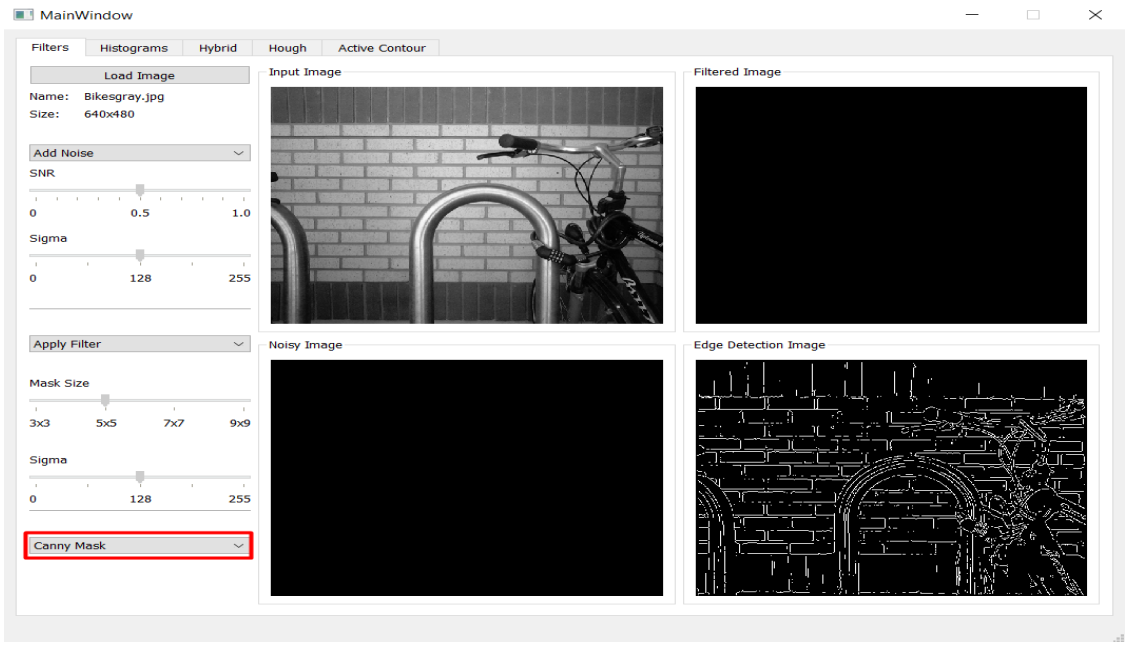
Output Image

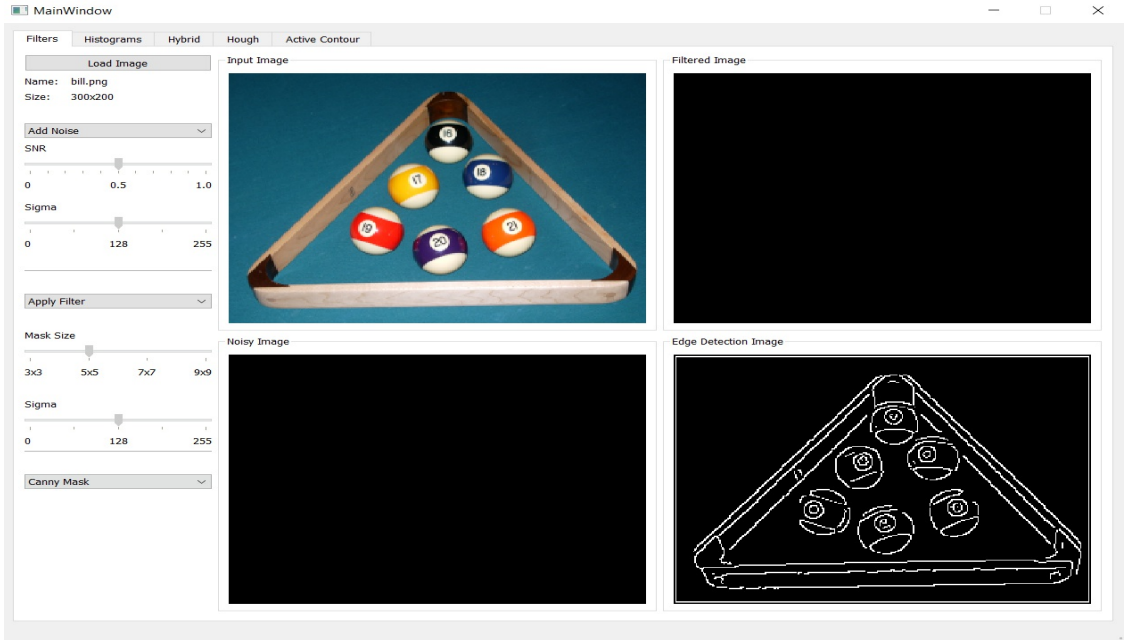
Edge and Boundary Detection

Table of content

Edge Detection Using Canny Edge Detector

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.

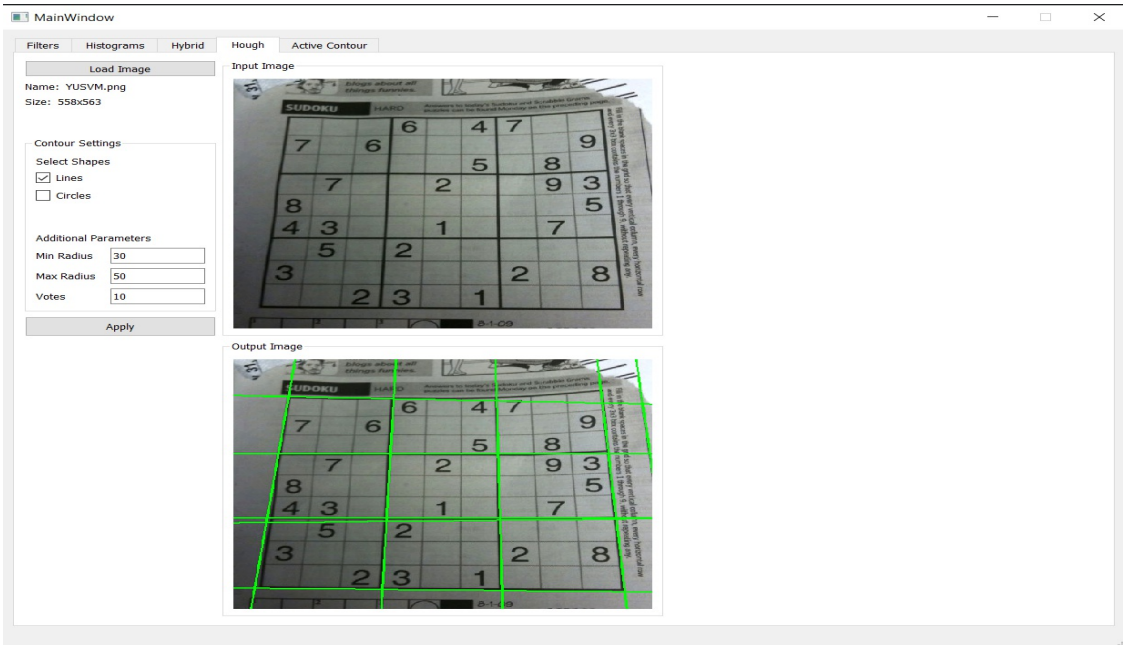



MainWindow
Filters
Histograms
Hybrid
Hough
Active Contour
Load Image
Name: bill.png
Size: 300x200
Add Noise
SNR
0
0.5
1.0
Sigma
0
128
255
Apply Filter
Mask Size
3x3
5x5
7x7
9x9
Sigma
0
128
255
Canny Mask
Input Image
Filtered Image
Noisy Image
Edge Detection Image

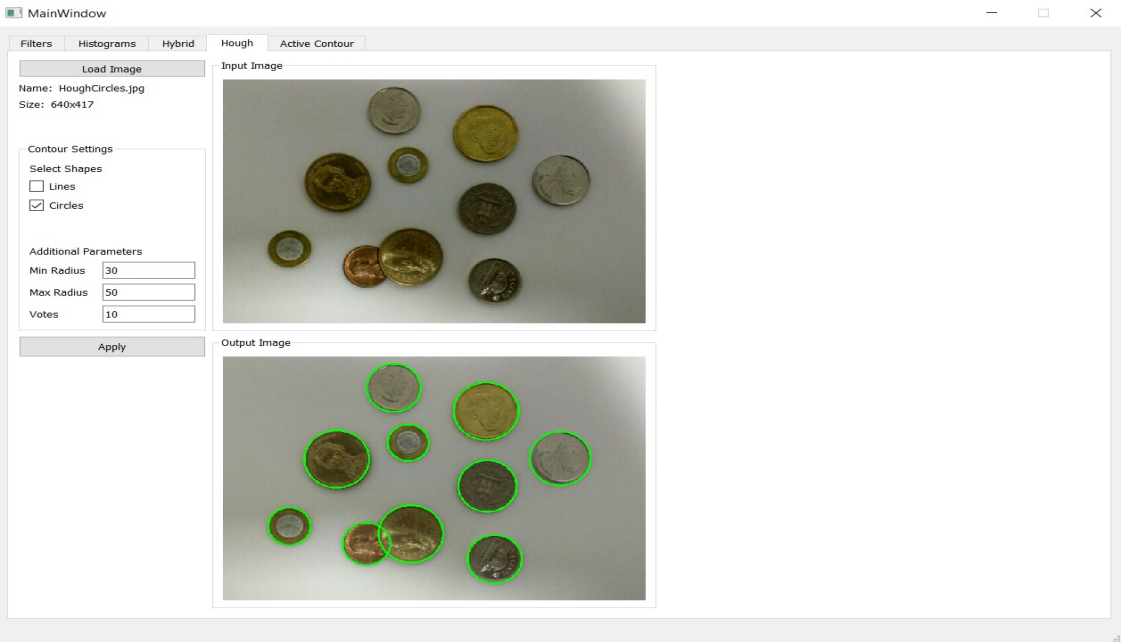
Hough Transformation

The Hough transform is a technique that locates shapes in images. In particular, it has been used to extract lines, circles and ellipses if you can represent that shape in mathematical form.

Line Detection



Circles Detection

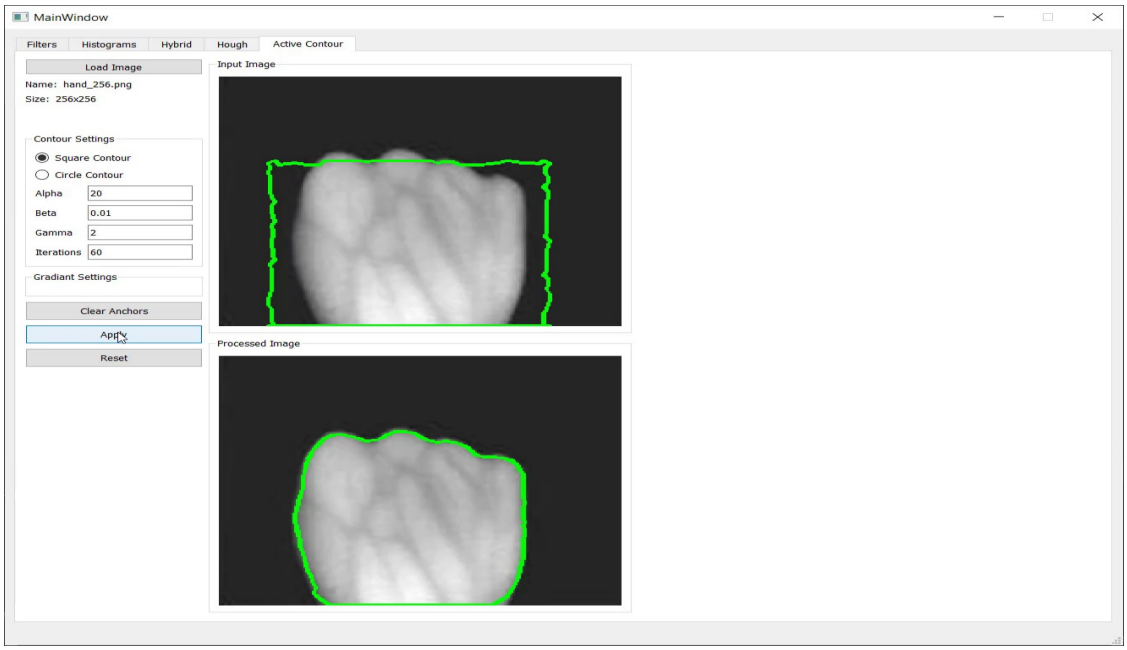


Active Contour Model (Using Greedy Algorithm)

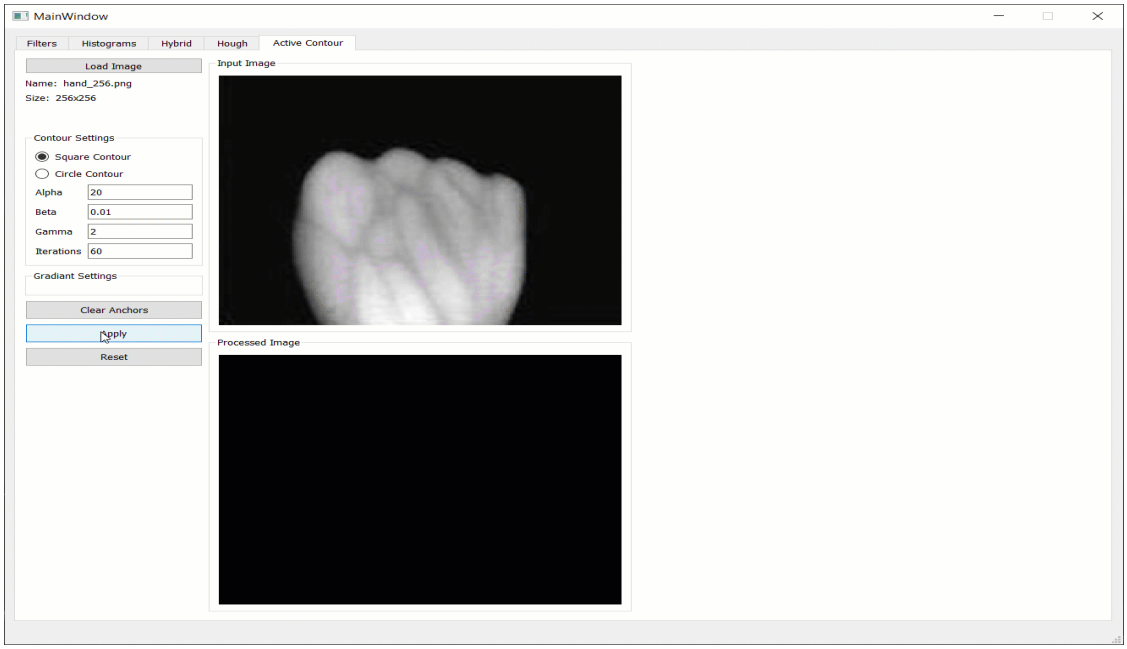
Active contour is one of the active models in segmentation techniques, which makes use of the energy constraints and forces in the image for separation of region of interest.

Active contour defines a separate boundary or curvature for the regions of target object for segmentation.

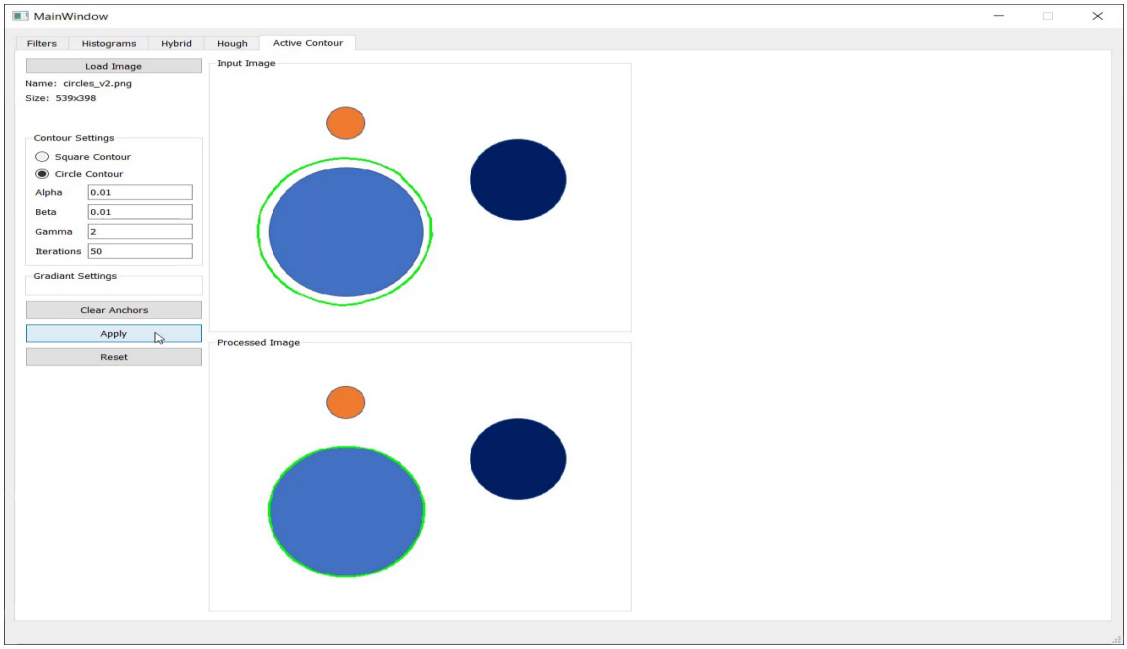
Result of applying the algorithm on hand image



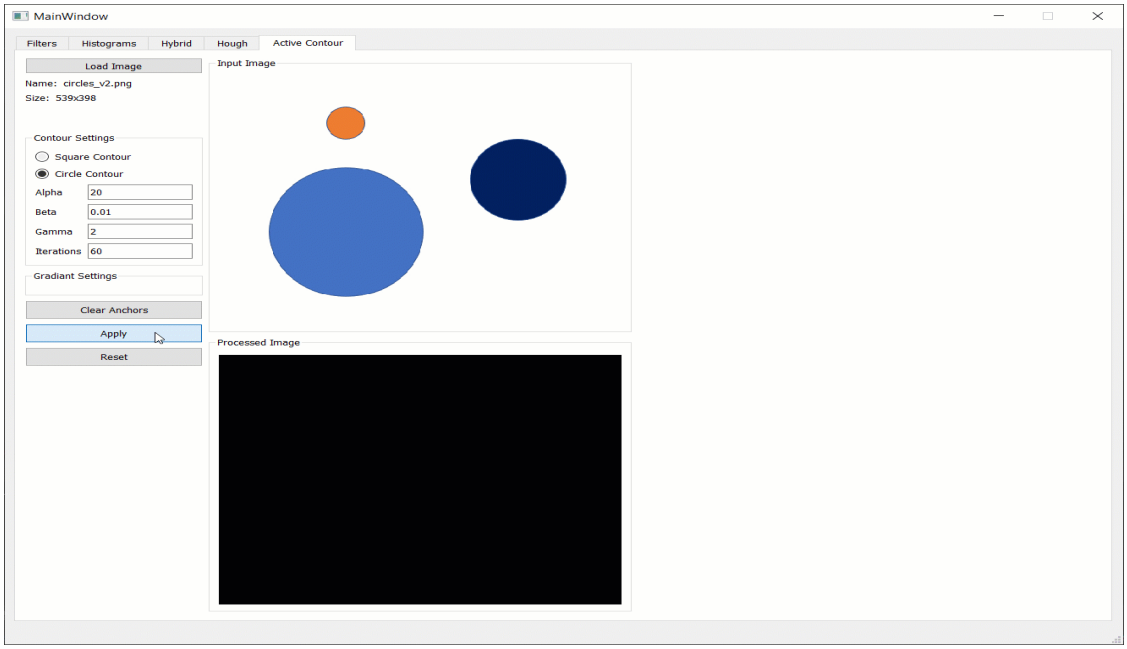
This GIF shows the process in a better way



Result of applying the algorithm on hand image



This GIF shows the process in a better way



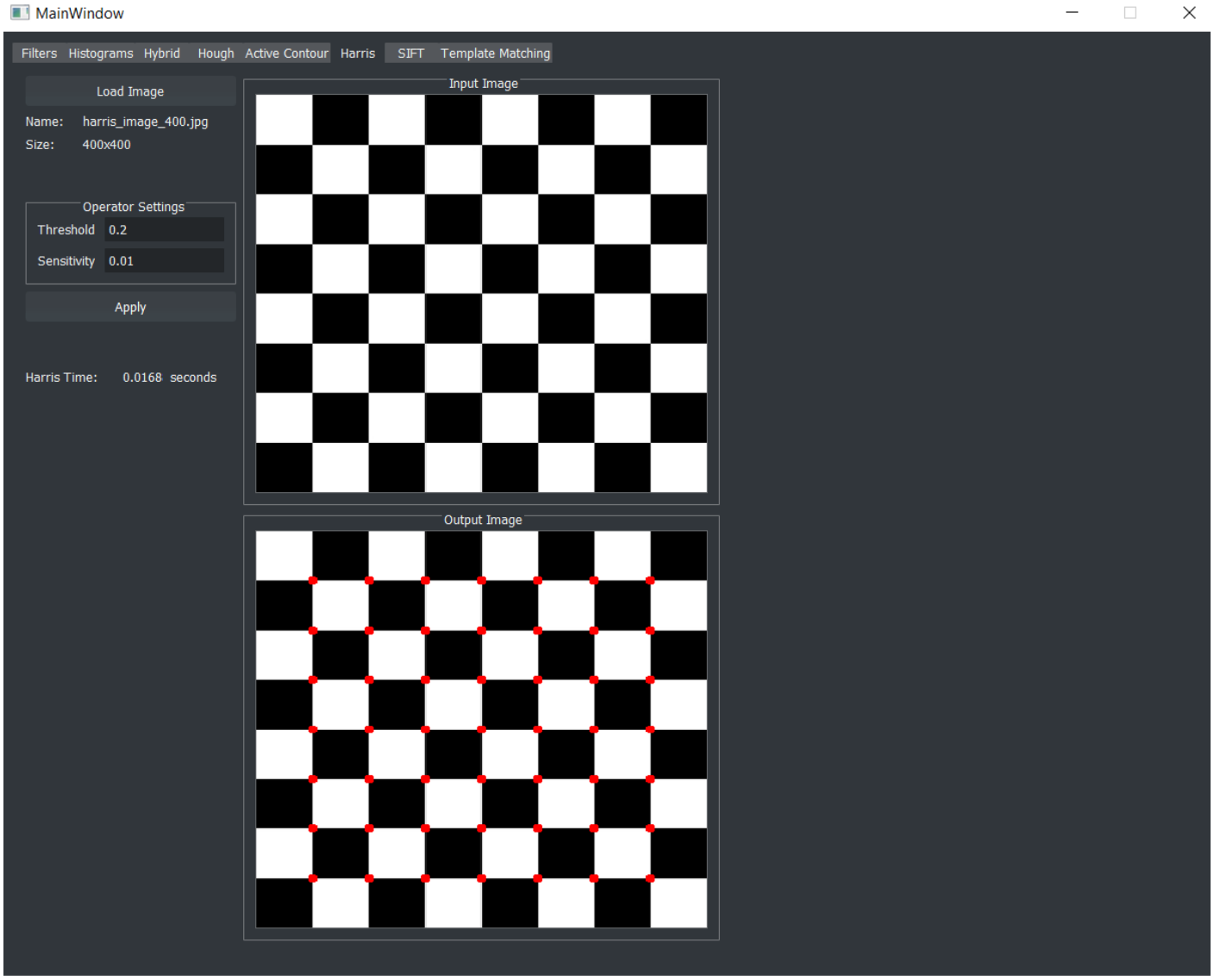
Features Detection and Image Matching

Table of content

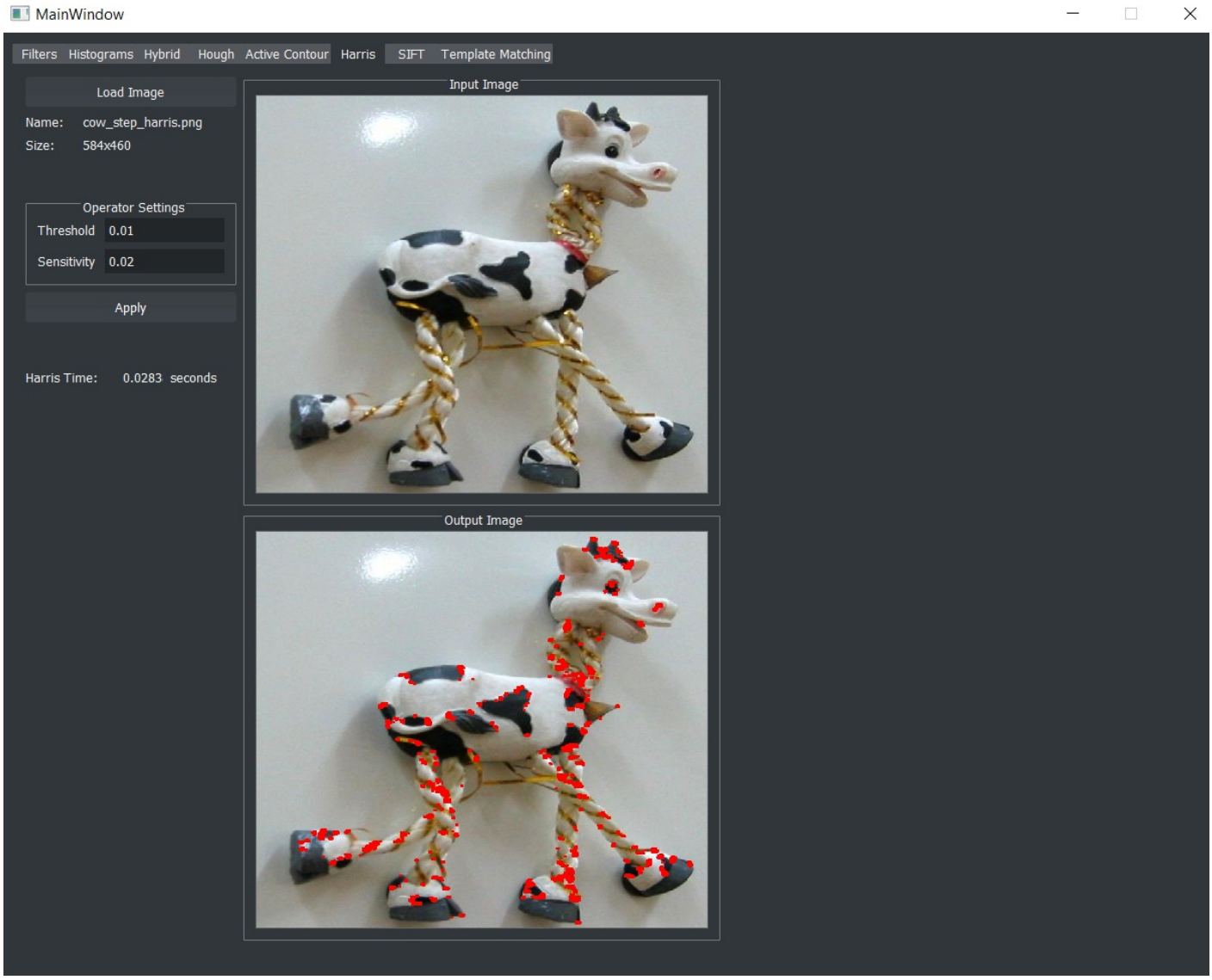
Each Algorithm Applied was thrown onto a thread for faster better experience

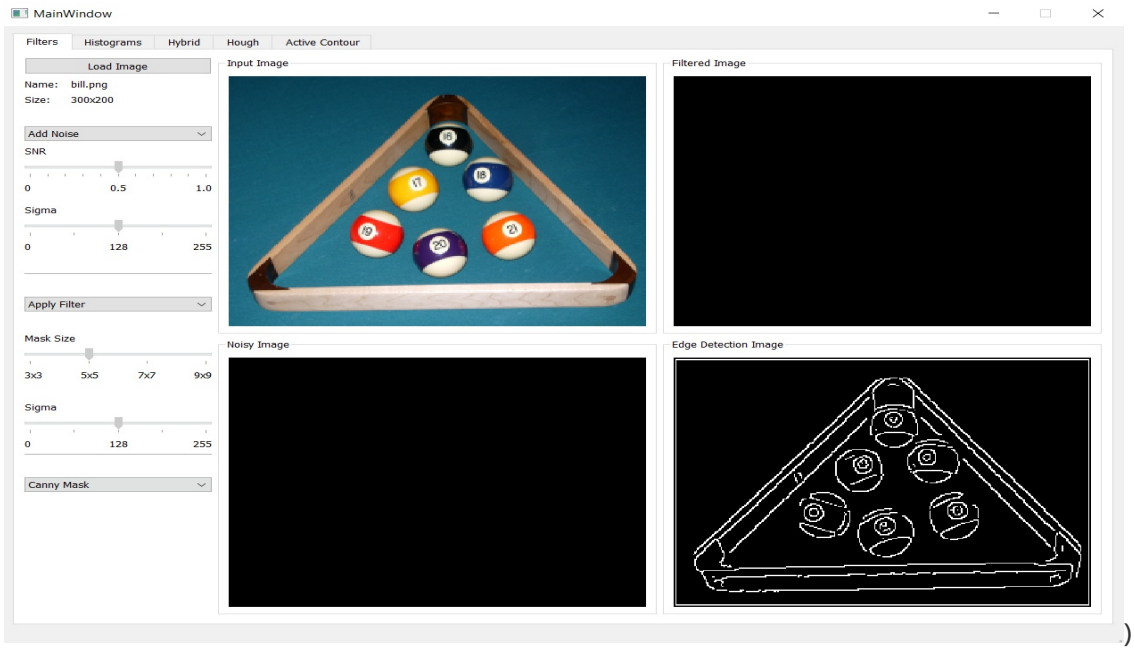
Extract the unique features in all images using Harris operator

Applying a threshold 0.2 it only took about 0.01 second to detect all Image Corners



We also applied the harris operator on a harder image with the same threshold this time it only took 0.02 seconds


MainWindow
Filters
Histograms
Hybrid
Hough
Active Contour
Harris
SIFT
Template Matching
Load Image
Name: cow_step_harris.png
Size: 584x460
Operator Settings
Threshold 0.01
Sensitivity 0.02
Apply
Harris Time: 0.0283 seconds
Input Image
Output Image


MainWindow
Filters
Histograms
Hybrid
Hough
Active Contour
Load Image
Name: bill.png
Size: 300x200
Add Noise
SNR
0
0.5
1.0
Sigma
0
128
255
Apply Filter
Mask Size
3x3
5x5
7x7
9x9
Sigma
0
128
255
Canny Mask
Input Image
Filtered Image
Noisy Image
Edge Detection Image

)

Using Sift Descriptors and Harris Operator to Match the image set features

Using Harris Operator to Detect Image Key Points and Applying the SIFT Algorithm to Generate each Feature Descriptor, Applying Two Matching Algorithms SSD And NCC

using sum of squared differences (SSD)

Using Normalized Cross Correlations (NCC)