An Edge Detection System for Polluted Images by Gaussian, Salt and Pepper, Poisson and Speckle Noises

- S.Muhammad Hossein Mousavi, Department of Computer Engineering, Bu Ali Sina University
- Marwa Kharazi, Department of Computer Engineering, Bu Ali Sina University

Sections:

- New Edge Detection System for Noisy Images (Pipeline)
- Median filter
- Kernel and Convolution
- Traditional edge detection operators and methods
- THE FOUR PROPOSED FILTER FOR EDGE DETECTION
- Hit-or-Miss Transform
- Erosion
- The visual procedure of the edge detection system
- Validation Measures

New Edge Detection System for Noisy Images (Pipeline):

Acquisiting RGB image from the input

→ Applying Median filter to the three decomposed channels of R, G, B for initial smoothing of the image

→ Applying Sharpening method to sharpen the edges, and converting the coloured image (smoothed from the inside and sharpen from the outside of the object) to gray in order to apply the proposed edge detection filter

→ Applying the proposed edge detection filter for edge detection of images which is polluted by different noises, from four sides

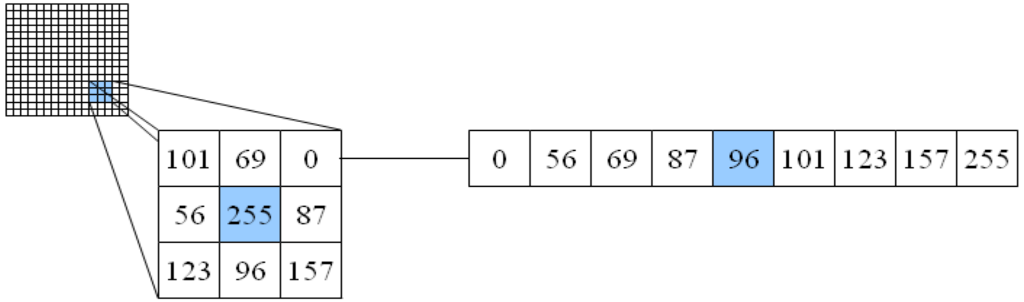
→ Applying a kind of Post-Processing to eliminate the possible spots in the binary images

→ Applying Hit or Miss morphology operations on the output, to thin the edges

→ Validation using MSE, PSNR, SSIM factors for comparing final edge detected images (with noise and without) and also comparing the proposed method with classic operators

Median filter:

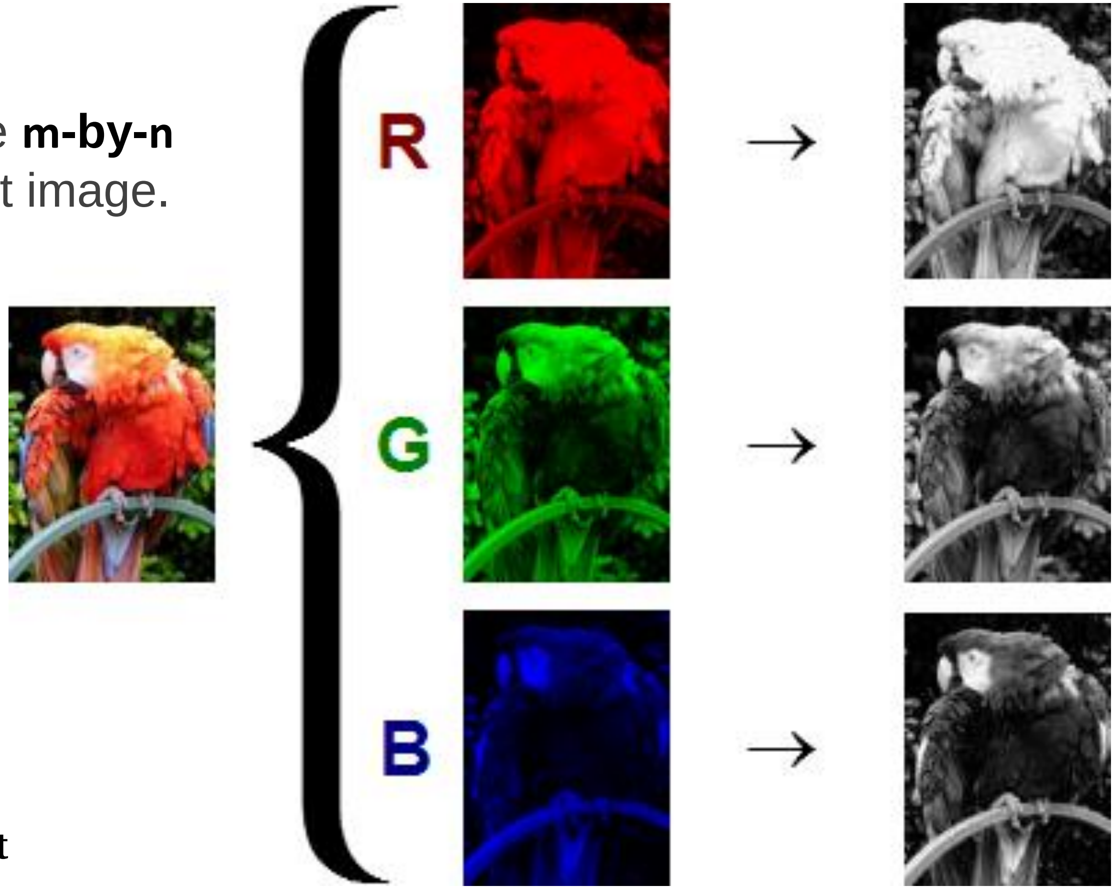
where each output pixel contains the median value in the **m-by-n** neighborhood around the corresponding pixel in the input image.



How selecting pixels by the Median filter

0	**1**	**0**
1	**-4**	**1**
0	**1**	**0**

SHARPENING FILTER, USED TO SHARPEN EDGES (**High Pass Filt**



Kernel and Convolution:

In image processing, a **kernel**, **convolution matrix**, or **mask** is a small matrix.

It is used for **blurring, sharpening, edge detection**, and more.

Convolution is the process of adding each element of the image to its local neighbors, weighted by the **kernel**.

Gaussian blur kernel

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Sobel Operator:

The operator uses two 3×3 kernels which are **convolved** with the original image

Convolution sign

Image

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

Roberts Cross:

It was one of the first edge detectors and was initially proposed by **Lawrence Roberts in 1963**.

$$\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$$

Prewitt operator:

$$\mathbf{G_x} = \begin{bmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G_y} = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * \mathbf{A}$$

The Laplacian of Gaussian:

One of the first and also most common **blob detectors** is based on the Laplacian of the Gaussian (LoG).

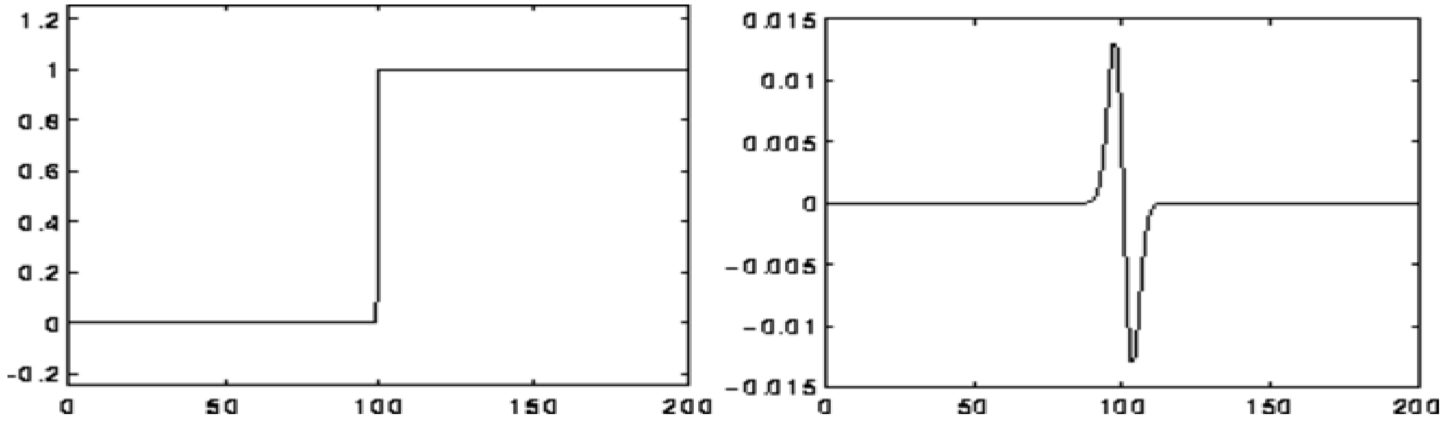
Given an input image f(x,y) , this image is convolved by a **Gaussian kernel**

$$g(x,y,t) = \frac{1}{2\pi t} e^{-\frac{x^2+y^2}{2t}}$$

Zero Crossing Detector:

The zero crossing detector looks for places in the Laplacian of an image where the value of the **Laplacian passes through zero** points where the Laplacian changes sign. Such points often occur at ‘edges’ in images.

The core of the zero crossing detector is the **Laplacian of Gaussian** filter.



Response of 1-D LoG filter to a step edge. The left hand graph shows a 1-D image, 200 pixels long, containing a step edge. The right hand graph shows the response of a 1-D LoG filter with Gaussian standard deviation 3 pixels.

Canny Edge Detector:

It was developed by **John F. Canny in 1986**.

The Process of Canny edge detection algorithm can be broken down to 5 different steps:

- Apply **Gaussian filter** to smooth the image in order to remove the noise.
- Find the **intensity gradients** of the image.(**Edges Directions Degree**)
- Apply **non-maximum suppression** to get rid of spurious response to edge detection.(**Edge Thinning Technique**)
  indicates location with the sharpest change of intensity value
- Apply **double threshold** to determine potential edges.
  some edge pixels remain that are caused by noise and color variation
- Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

THE FOUR PROPOSED FILTER FOR EDGE DETECTION:

-1.2	-0.8	-0.6		1.2	0.8	0.6
0	0	0		0	0	0
1.2	0.8	0.6		-1.2	-0.8	-0.6
1.2	0	-1.2		-1.2	0	1.2
0.8	0	-0.8		-0.8	0	0.8
0.6	0	-0.6		-0.6	0	0.6
-0.6	-0.8	-0.1		0.6	0.8	0.1
0	0	0		0	0	0
0.6	0.8	0.1		-0.6	-0.8	-0.1
0.1	0	-0.1		-0.1	0	0.1
0.8	0	-0.8		-0.8	0	0.8
0.6	0	-0.6		-0.6	0	0.6

AUXILIARY TABLE FOR IMPLEMENTATION OF (1)

Z1	Z2	Z3
Z4	Z5	Z6
Z7	Z8	Z9

$$|\nabla f(x,y)| = |(0.6Z7+0.8Z8+1.2Z9)-(0.6Z1-0.8Z2-1.2Z3)| + |(1.2Z3+0.8Z6+0.6Z9)-(1.2Z1-0.8Z4-0.6Z7)| \quad (1)$$

Hit-or-Miss Transform:

- In mathematical morphology, hit-or-miss transform is an operation that detects a **given pattern** in a **binary image**
- using the morphological **erosion** operator and a **pair of disjoint structuring elements**.
- The result of the hit-or-miss transform is the **set of positions.**
- the **first** structuring element **fits** in the foreground of the input image.
- **second** structuring element **misses** it completely.

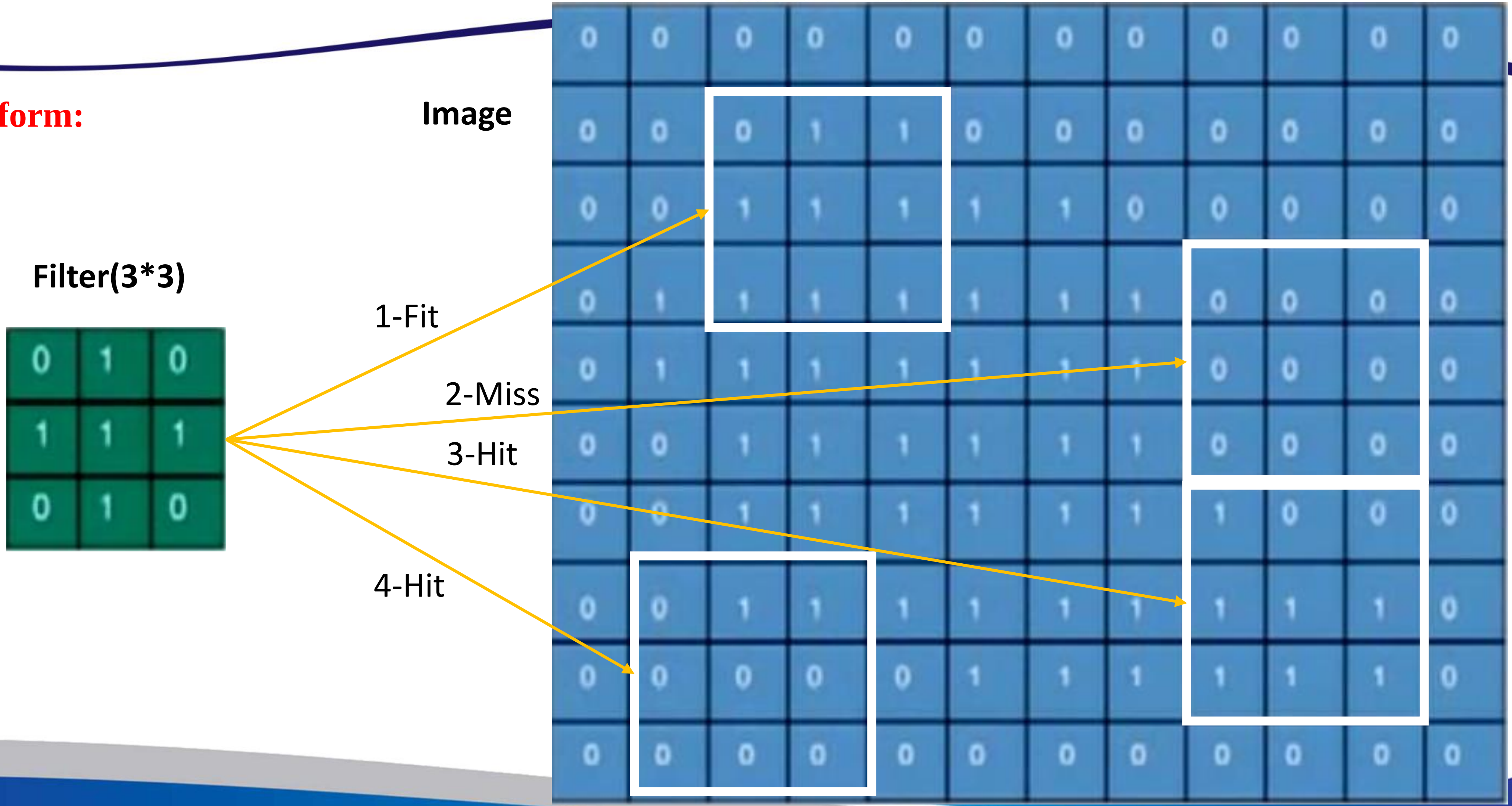
1	1	1
1	1	1
1	1	1

Structuring Element 1

0	1	0
1	1	1
0	1	0

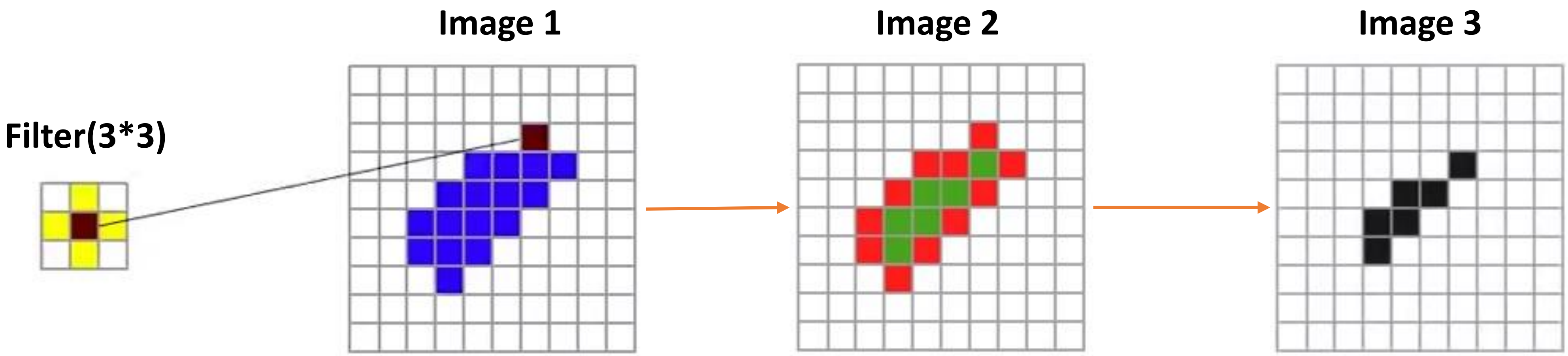
Structuring Element 2

Hit-or-Miss Transform:

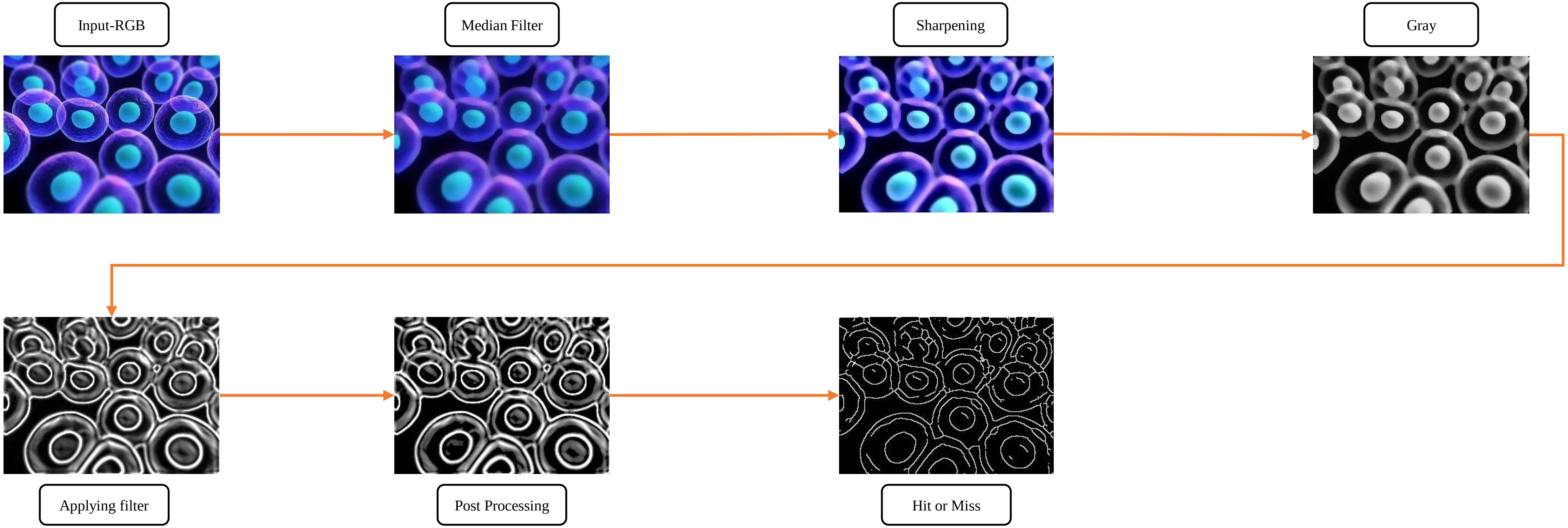


Erosion:

The basic idea in binary morphology is to **probe** an image with a simple, **pre-defined shape,**
drawing conclusions on how this shape fits or misses the shapes in the image.
This simple "**probe**" is called **structuring element**, and is itself a **binary image** (i.e., a subset of the space or grid).

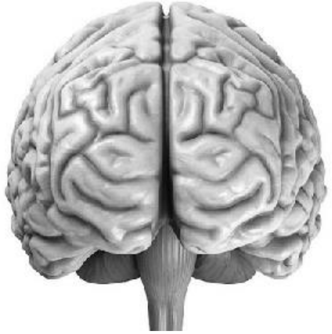


The visual procedure of the edge detection system (from left to right):

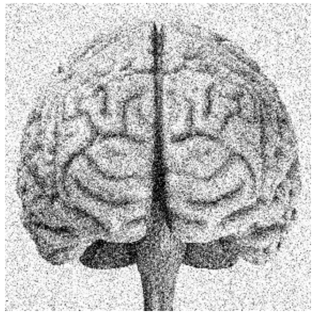


Applying different noise to test image:

RGB

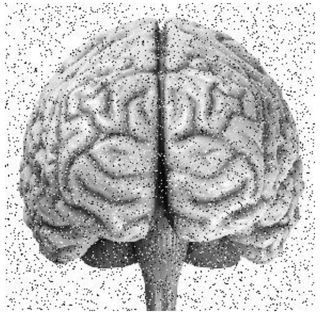


Gray



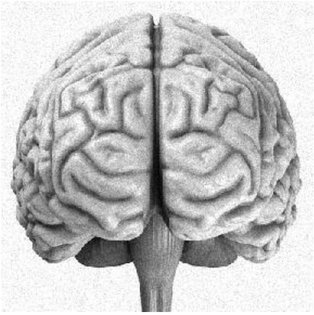
Guassian

Mean= 0 ,STD =0.05



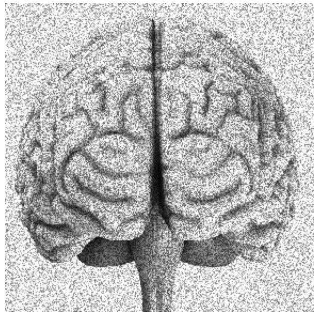
S & P

Noise Density=0.1



Poisson

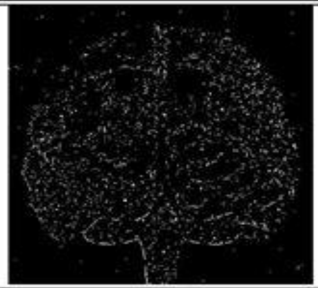
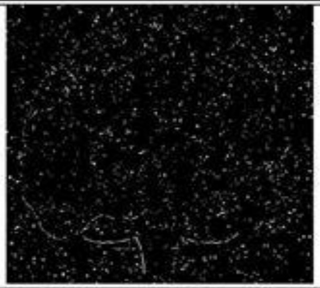
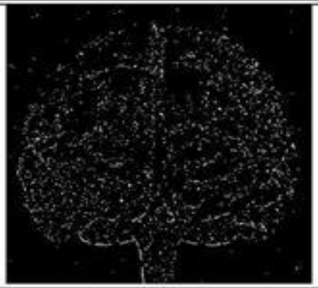
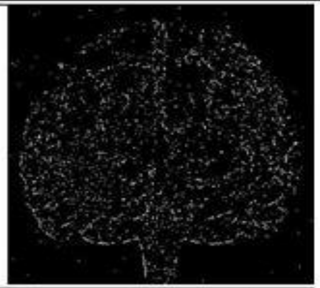
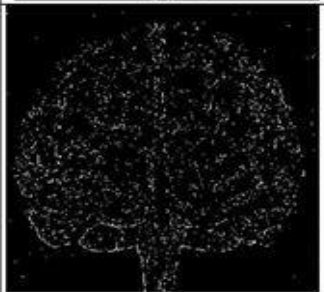
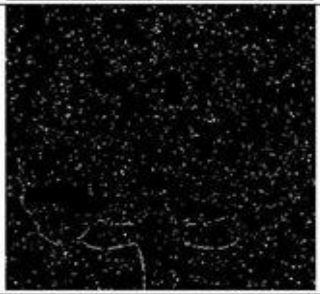
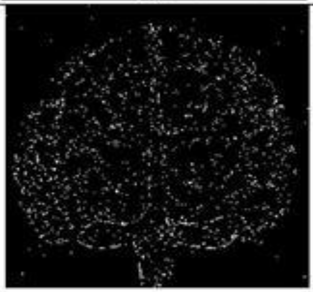
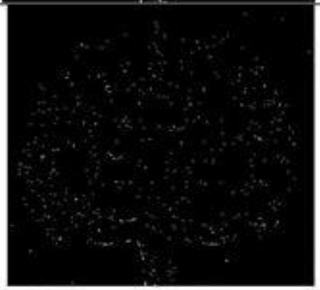
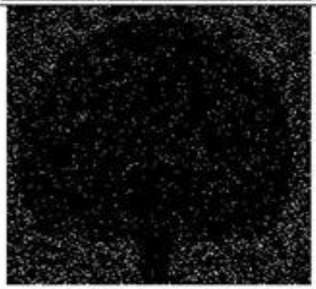
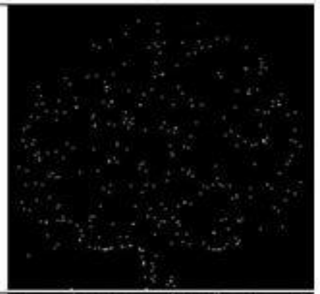
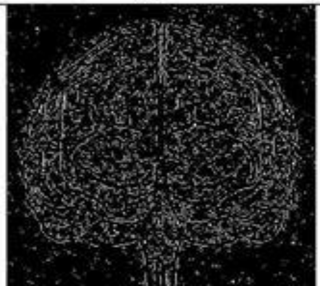
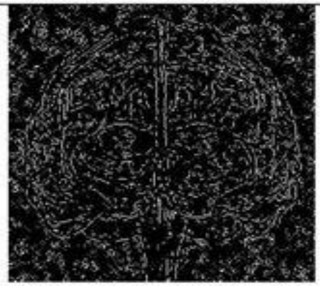
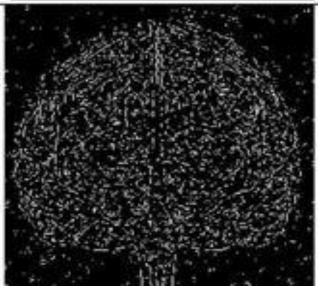
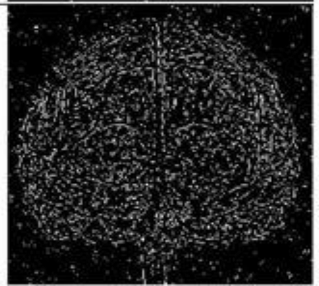
-



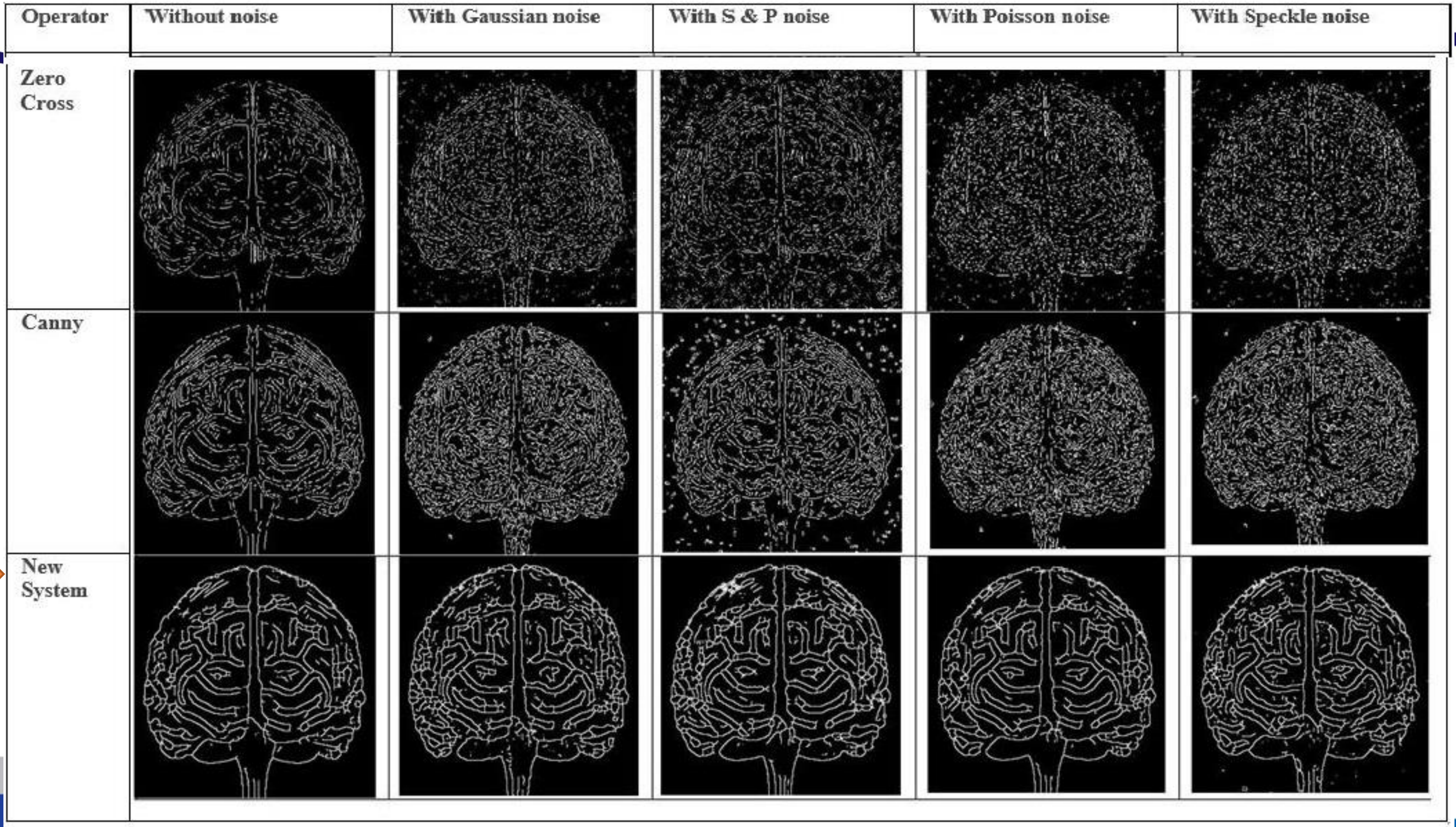
speckle

Variance=0.1

Results:

Operator	Without noise	With Gaussian noise	With S & P noise	With Poisson noise	With Speckle noise
Sobel					
Prewitt					
Roberts					
Log					

Results:



Proposed system results comparing with other traditional conventional operators, on different noises

Validation Measures:

$$\text{MSE} = \frac{1}{M \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [X(i,j) - Y(i,j)]^2$$

Mean squared error = The value of MSE must be less and the **value 0 is the best**, meaning there is no error.
In which, X and Y are two arrays with the size of M*N.

$$\text{PSNR} = 10\log_{10} \frac{L^2}{\text{MSE}}$$

Peak signal-to-noise ratio = usually between **5 to 50**. The **more this value is, the better**.
In which L determines the range of value, which a pixel could have. Its unit is DB, and has a limit of 50.

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C1)(2\sigma_{xy} + C2)}{(\mu_x^2 + \mu_y^2 + C1)(\sigma_x^2 + \sigma_y^2 + C2)}$$

Structural similarity index = is a number between 0 and 1. **The higher this value** is, the better result exists.
SSIM is designed to improve on traditional methods like peak signal-to-noise ratio (PSNR) and mean squared error (MSE)

MSE error, related to the mutuality of the proposed system and prior methods to the different noises:

MSE / Operators	Noise type	Gaussian noise	S & P noise	Poisson noise	Speckle noise
Sobel		0.0819	0.0516	0.0466	0.0994
Prewitt		0.0816	0.0511	0.0468	0.0985
Roberts		0.0956	0.0733	0.0637	0.1151
Log		0.1384	0.1612	0.0313	0.1562
Zerocross		0.1381	0.1609	0.0320	0.1573
Canny		0.1936	0.2080	0.0538	0.2543
New System		0.0732	0.0190	0.0244	0.0849

PSNR related to the mutuality of the proposed system and prior methods to the different noises:

PSNR / Operators	Noise type	Gaussian noise	S & P noise	Poisson noise	Speckle noise
Sobel		13.77	12.87	17.81	14.04
Prewitt		13.81	12.91	17.75	14.14
Roberts		11.28	11.96	10.16	10.09
Log		8.58	7.92	15.4	8.06
Zerocross		8.59	7.93	14.95	8.03
Canny		7.13	6.81	12.69	5.94
New System		14.53	23.88	23.56	14.32

SSIM, related to the mutuality of the proposed system and prior methods to the different noises:

SSIM / Operators	Noise type	Gaussian noise	S & P noise	Poisson noise	Speckle noise
Sobel		0.4196	0.3482	0.8230	0.4784
Prewitt		0.4235	0.3481	0.8196	0.4869
Roberts		0.3006	0.3995	0.6537	0.5211
Log		0.1400	0.1131	0.7487	0.1069
Zerocross		0.1432	0.1162	0.7443	0.1050
Canny		0.2649	0.1721	0.7542	0.1233
New System		0.7786	0.9576	0.9615	0.6038

Conclusion:

- The proposed system is not only a **new edge detection** approach, but is able to deal with different noises in digital images, and indicate the **least of sensitivity**.
- This system does not change the **real position of the edges**.
- Presenting **promising results**, in validated visual and statistical results against other **conventional edge detection** methods
- With changing the proposed system, we are able to create a sort of image **feature extraction method.**

با تشکر از توجه شما

Thank You

؟