# Performance Evaluation of Canny Edge Detector for Noisy Images

**[1] Priya Arora, [2]Dr. Lakhwinder Kaur**
[1]*Research Scholar, University College of Engineering, Punjabi University, Patiala*
prrr777@gmail.com
[2]*Professor, University College of Engineering, Punjabi University, Patiala*
mahal2k8@yahoo.com

***ABSTARCT-*** **Edge detection is one of the most important technique in the field of image processing and computer vision, particularly in the areas of feature detection and feature extraction. Detecting edges is a basic operation in image processing because an edge in an image usually refers to the boundaries between different regions and reflects a break in the continuation of any image characteristic. This paper analyses the effect of Canny Edge Detector for detection of edges in digital images corrupted with different kinds of noise. Different kinds of noise are studied in order to evaluate the performance of the Canny Edge Detector. Further, the various standard test images are examined to validate our results. The software is developed using MATLAB 7.0. It has been observed that the Canny Edge Detector performs effectively for digital images corrupted with poisson noise and performs poorly for other kinds of noise. The results of this study are quiet promising.**

***Keywords*: *Digital Image Processing,Edge Detector, Noise***

## 1. INTRODUCTION

Digital image processing is defined as the use of computer algorithms to perform different kinds of processing on digital images. Image processing is defined as a technique where data is collected from an image and converted into digitized form and various image processing related algorithms and operators are applied to the data, generally with the use of a digital computer, in order to create an enhanced image that is more useful or pleasing to a human observer[1]. Edge detection is a fundamental process for processing the image as it is use to localize the sharp discontinuities in an image. It could be defined as detecting the edges in an image rather than objects. Edge pixels are those whose intensity is much variable than any of its neighboring pixels and edges are the set of connected edge pixels[2]. Edge detection not only extracts structural information of objects in an image but also reduces the data to be processed. The main aim of the edge detection is to mark the points of abrupt discontinuity in intensity of pixels. Edge representation of an image drastically reduces the amount of data to be processed, yet it retains important information about the shapes of objects in the scene. This description of an Image is easy to integrate. This paper studies the effect of canny edge detector for Salt & Pepper, Gaussian, Poisson and Speckle noises. The paper is comprised of 3 sections. Section 2 introduces the various noises, section 3 discusses the canny edge detector and Section 4 deals with the results obtained by simulating the canny edge detector.

## 2. NOISE

In the Digital Image Processing field, enhancement and removing the noise from the image is the critical issue. Gaussian noise (White noise), Salt & Pepper noise and Speckle noise are the types of noises which are generally found in Images, and also denoising them with the help of some efficient technique is of main concern. Noise when get added to image destroy the details of it. So in order to preserve the real image, noise should get removed from it. And for the purpose of enhancement the contrast of the image should be improved.

### 2.1 GAUSSIAN NOISE

Gaussian noise is statistical noise that has a probability density function of the normal distribution (also known as Gaussian distribution)[3]. In other words, the values that the noise can take on are Gaussian-distributed. This type of noise can be described by an additive noise model, where the recorded image is the sum of the true image and the noise.

### 2.2 SALT & PEPPER NOISE

This type of noise is caused by errors in data transmission and is a special case of data dropout noise when in some cases single pixels are set alternatively to zero or to the maximum value giving the image a salt and pepper like appearance[4]. It generally appears as black and white dots in an image[5]. Unaffected pixels always remain unchanged. It represents itself as randomly occurring white and black pixels.

### 2.3 POISSON NOISE

This type of noise is caused by the nonlinear response of the image detectors and recorders. Here the image data dependent (signal dependent) term arises because detection and recording processes involve random electron emission having a Poisson distribution with a mean response value.

### 2.4 SPECKLE NOISE

Another common form of noise is data dropout noise commonly referred to as Speckle noise. This noise is, in fact, caused by errors in data transmission[6]. The corrupted pixels are either set to the maximum value, which is something like a snow in image or have single bits flipped over.

## 3. CANNY EDGE DETECTOR

A variety of Edge Detectors are available for detecting the edges in digital images. The basic idea behind edge detection is to find places in an image where the intensity changes rapidly. Edge detection is based on either first order derivative or second order derivative. A number of edge detectors have been proposed such as Sobel, Roberts, Prewitt, DoG (Difference of Gaussian) etc. which is either based on finding the first order derivative maxima or minima or zero crossings in the second order derivatives of pixel intensity in an image. The second order derivative based edge detectors are highly sensitive to noise and leads to error in edge detection in presence of noise. The Canny edge detection algorithm is known to many as the optimal edge detector. It works in a multi stage process. The canny edge detector works by first eliminating the noise by smoothing the image with the help of a Gaussian filter and then finds image gradient to highlight region with high spatial derivatives. It is then followed by then tracking all these regions and suppressing any pixel that is not at the maximum. Finally, it uses double thresholding and connectivity analysis to detect and link edges. The Canny's intentions were to enhance the many edge detectors already out at the time he started his work. He was very successful in achieving his goal and his ideas and methods can be found in his paper, "A Computational Approach to Edge Detection"[7]. His approach is based on the three criteria for an 'optimal' edge detector: a) low error rate, b) Edge points should be well localized and c) only on response to single edge point. In order to satisfy these criteria Canny suggested that a good approximation to the optimal edge detector is the first derivative of Gaussian. Based on these criteria, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (nonmaximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a nonedge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2.

The Canny edge detection algorithm uses a multi stage algorithm in order to detect a wide range of edges in images. The algorithm constitutes the following basic steps:

1. Filter the noise in the original image before trying to locate and detect any edges and smooth the input image through Gaussian filter. Gaussian smoothing can be performed using standard convolution methods. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise.

2. Find the edge strength by computing the gradient magnitude and angle of gradient vector for edge direction after smoothing the image and eliminating the noise. The Sobel operator performs a 2-D spatial gradient measurement on an image. Then, the approximate absolute gradient magnitude at each point can be found. The Sobel operator uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction and the other estimating the gradient in the y-direction. They are shown below:

| | | |
|---|---|---|
| -1 | -2 | -1 |
| 0 | 0 | 0 |
| +1 | +2 | +1 |

| | | |
|---|---|---|
| -1 | -2 | -1 |
| 0 | 0 | 0 |
| +1 | +2 | +1 |

Fig 1 (a, b): $G_x$, $G_y$ Canny mask

The magnitude or edge strength for the gradient is computed as

$$|G| = |Gx| + |Gy|$$

3. Finding the edge direction is trivial once the gradient in the x and y directions are known. However, you will generate an error whenever sum X is equal to zero. So in the code there has to be a restriction set whenever this takes place. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If GY has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. The formula for finding the edge direction is just:

$$\theta = \text{invtan}\,(Gy / Gx)$$

4. After the edge directions are known, apply non-maxima suppression[8] to the gradient magnitude to trace along the edge direction and suppress any pixel value that is not considered to be an edge and give a thin line to the input image.

5. Use double thresholding or hysteresis[9] and connectivity analysis to detect and connect edges. Hysteresis is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, T1 is applied to an image, and an edge has an average strength equal to T1, then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than T1 is presumed to be an edge pixel, and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than T2 are also selected as edge pixels[10].

## 4. RESULTS

In order to evaluate the performance of Canny edge detector, the standard images are used. Firstly, the edge is detected using the canny edge detector. Later the image is corrupted by adding various kind of noise to the original image using the MATLAB 7.0.Then the canny edge detector method is applied to the corrupted image to simulate their performance in presence of Salt and pepper, Gaussian noise, Speckle and Poisson noise. The edge image is used as a reference for comparison of images corrupted with different kind of noises. Then for each of the four noisy images, the performance of Canny Edge Detector is examined practically. PSNR values for different test images corrupted with different kind of noise are shown in Table-1. It has been observed that the Canny Edge Detector works well with the Poisson noise corrupted images as PSNR value is same for image corrupted with poison noise with original image by applying canny edge detection method. However, its performance decreases drastically for Gaussian, Salt & Pepper as well as Speckle noise corrupted images. In order to validate our results about the performance of Canny Edge Detector, three different standard test images, each corrupted with four kinds of noise are considered. The performance of Canny Edge Detector is examined both for the original as well as noise corrupted images. From the results, it has again been observed that the performance of the Canny Edge Detector is found to be satisfactory for all the test images corrupted with Poisson noise.

## 5. CONCLUSIONS

In this paper, an attempt is made to evaluate the performance of canny edge detector for noisy images. The canny edge detector whose performance has been evaluated with the salt and pepper, gaussian, poisson and speckle noises. It has been observed that the canny edge detector works well for poisson noise whereas its performance decreases for other kinds of noise. Hence, canny edge detector cannot be used in practical images which are generally corrupted with Gaussian noise, salt & pepper noise and speckle type of noise. Before applying the canny edge detector, these can be used successfully in conjunction with suitable digital filter to reduce the effect of noise. So, firstly noise is to be reduced by convolving the image with a suitable digital filter.

## 5. REFERENCES

[1] Namrata Puri, Manjeet Singh, Sumit Kaushik "Effect of Impulsive Noise on Edge Detectors for RGB Planar Images", (IJARCSSE ) Journal, Vol. 2, no. 10, Oct.2012

[2] Dr. Raman Maini, J.S.Sohal," Performance Evaluation of Prewitt Edge Detector for Noisy Images", GVIP Journal, Vol. 6, no. 3, 2006

[3] "Gaussian noise" from http:// en. wikipedia. org/ wiki/ Gaussian noise

[4] R.C. Gonzalez, R.E. Woods and S.L. Eddins, "Digital Image Processing using MATLAB", Pearson Education Inc., 2004.

[5] Z. Wang and D. Hang, "Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images," IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal processing, vol. 46, no. 1, pp. 78-80 Jan. 1999.

[6] L. Gagnon and F.D. Smaili: 'Speckle Noise Reduction of Airborne SAR Images with Symmetric Daubechies Wavelets', SPIE Proc.#2759, pp. 1424, 1996

[7]. J. F. Canny. "A computational approach to edge detection". IEEE Trans. Pattern Anal. Machine Intell, vol. PAMI-8, no. 6, pp. 679-697, 1986

[8] Yuan-Hui Yu, Chin-Chen Chang, "A new edge detection approach based on image context analysis" Image and Vision Computing 24 (2006) 1090–1102

[9] J. Canny. "Finding edges and lines in image". Master's thesis, MIT, 1983.

[10] E. Argyle. "Techniques for edge detection," Proc. IEEE, vol. 59, pp. 285-286, 1971

## RESULTS

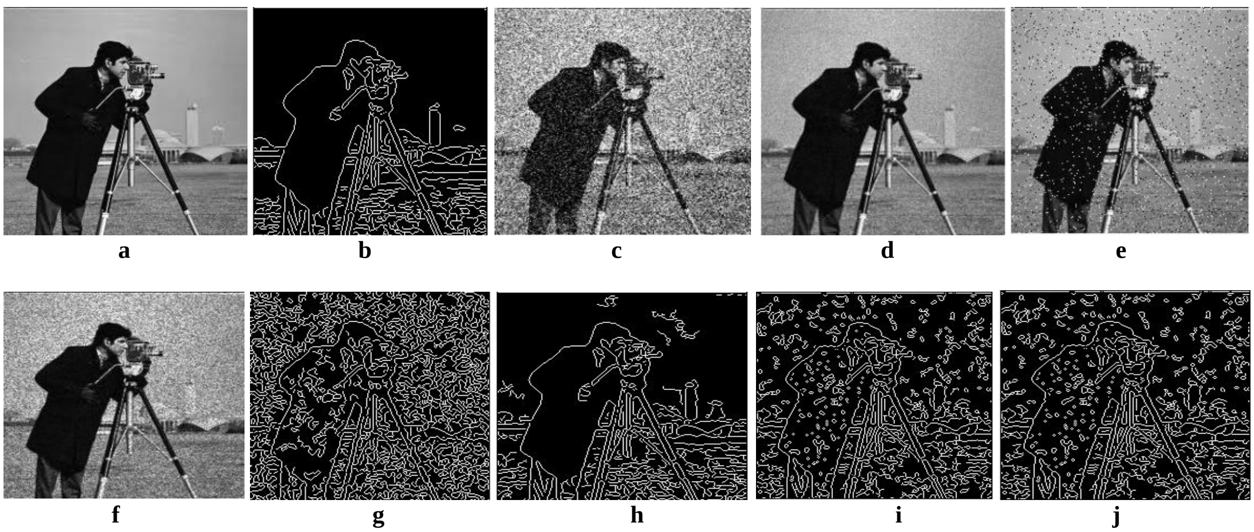



Fig 1(a,b,c,d,e,f,g,h,i,j): (a) Original Image (b) Canny edge detection method to original image (c) Gaussian noise corrupted image (d) Poisson noise corrupted image (e) Salt & pepper noise corrupted image (f) Speckle noise corrupted image (g) Canny edge detection method to gaussian noise corrupted image (h) Canny edge detection method to poisson noise corrupted image (i) Canny edge detection method to salt & pepper noise corrupted image (j) Canny edge detection method to speckle noise corrupted image

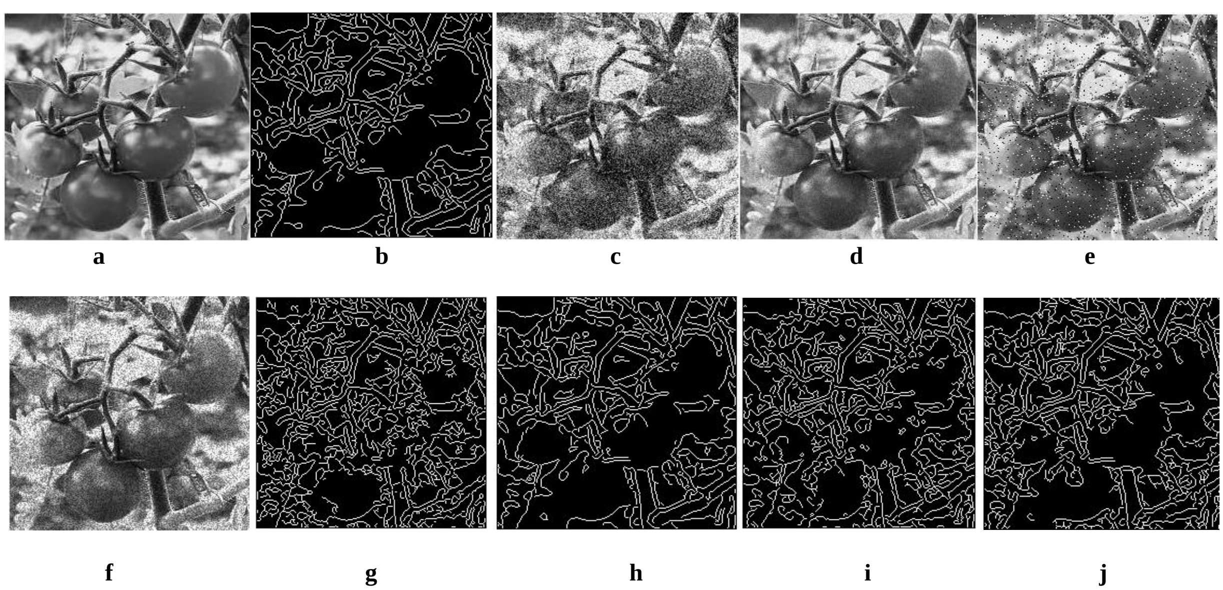
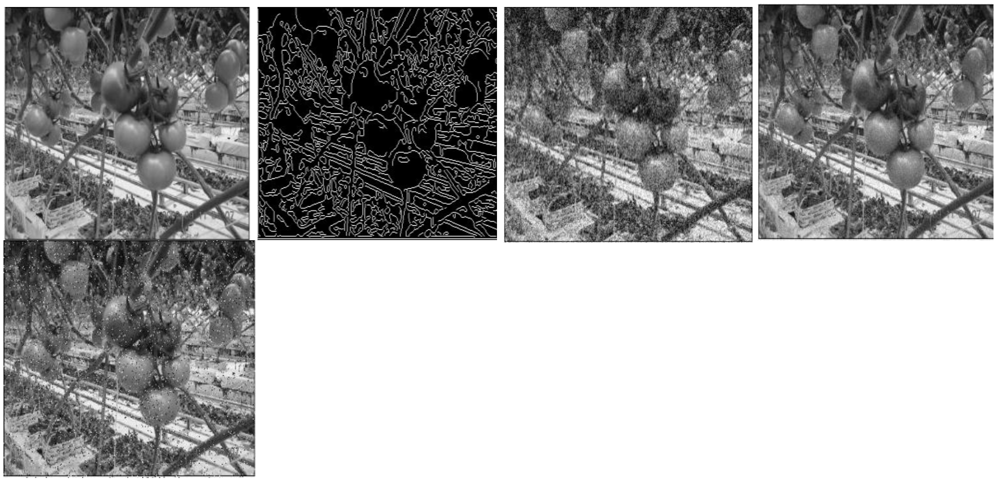
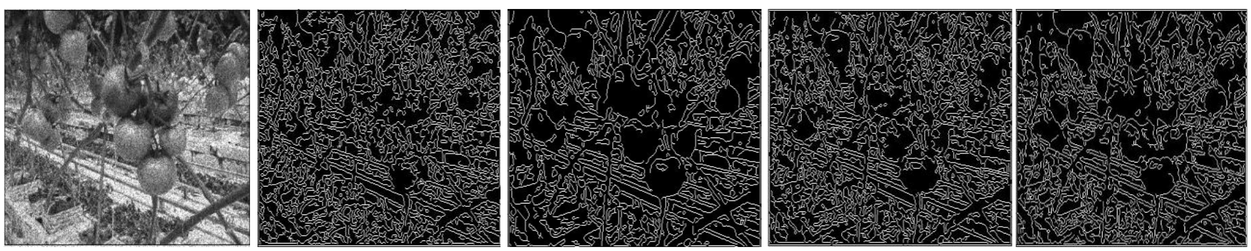



Fig 2(a,b,c,d,e,f,g,h,i,j): (a) Original Image (b) Canny edge detection method to original image (c) Gaussian noise corrupted image (d) Poisson noise corrupted image (e) Salt & pepper noise corrupted image (f) Speckle noise corrupted image (g) Canny edge detection method to gaussian noise corrupted image (h) Canny edge detection method to poisson noise corrupted image (i) Canny edge detection method to salt & pepper noise corrupted image (j) Canny edge detection method to speckle noise corrupted image



a b c d e



f g h i j

Fig 3(a,b,c,d,e,f,g,h,i,j): (a) Original Image (b) Canny edge detection method to original image (c) Gaussian noise corrupted image (d) Poisson noise corrupted image (e) Salt & pepper noise corrupted image (f) Speckle noise corrupted image (g) Canny edge detection method to gaussian noise corrupted image (h) Canny edge detection method to poisson noise corrupted image (i) Canny edge detection method to salt & pepper noise corrupted image (j) Canny edge detection method to speckle noise corrupted image

| S.No. | Image | PSNR value | PSNR value | PSNR value |
|---|---|---|---|---|
| | Original image | Cameraman | Tomatoes | Tomato plant |
| 1 | Gaussian Noise corrupted image | 16.52 | 16.48 | 16.40 |
| 2 | Pepper & Salt noise corrupted image | 17.95 | 17.99 | 18.25 |
| 3 | Speckle noise corrupted image | 18.66 | 18.93 | 19.67 |
| 4 | Poisson noise corrupted image | 123.26 | 123.05 | 123.61 |
| 5 | Canny edge detection method to original image | 5.45 | 5.31 | 6.04 |
| 6 | Canny edge detection method to Gaussian noise corrupted image | 5.78 | 5.23 | 5.91 |
| 7 | Canny edge detection method to salt & pepper noise corrupted image | 5.43 | 5.26 | 5.93 |
| 8 | Canny edge detection method to Speckle noise corrupted image | 5.93 | 5.36 | 6.07 |
| 9 | Canny edge detection method to Poisson | 5.45 | 5.31 | 6.04 |

| | | | | |
|---|---|---|---|---|
| | noise corrupted image | | | |

Table 1- Calculated PSNR values used by Canny Edge Detector for different test images corrupted with different kind of noise