Open Access
Full Length Article

ORIGINAL ARTICLE

An Analysis of Edge Detection Methodologies in Spatial and Morphological Perspective

Qamar Nawaz[1], Mahzaib Younas[2], Isma Hamid[3], Syed Mushhad Mustuzhar Gilani[4]

[1]Lecturer, Department of Computer Science, University of Agriculture, Faisalabad Pakistan
[2]MS Student, Department of Computer Science, University of Agriculture, Faisalabad Pakistan
[3]Assistant Professor Department of Computer Science, National Textile University, Faisalabad Pakistan
[4]Assistant Professor, University Institute of Information Technology, PMAS, Arid Agriculture University, Rawalpindi Pakistan

ABSTRACT

Nowadays, edge detection through image processing is become core research due to its wide application in different fields such as medical and industries. In digital image processing, edge detection and feature extraction are considered active research areas. The different researcher has designed different methods (i.e.- Prewitt, Sobel, and Roberts) for edge detection in images for better visualization result. The basic purpose of the edge detection process is to interpret pixels values to find more significant, useful, and meaningful information about an image. The most important and strong feature in any image is called its edge, which can be detected by using different image processing algorithms. Therefore, a fast, simple, and robust method for edge detection become a basic need for edge detection and feature extraction. In this paper, we propose a morphologically based edge detection algorithm, which is based on three basic steps (i) Image Acquisition, (ii) Image enhancement, and (iii) Morphological edge detection filter. In the result, we compare the proposed algorithm with the existing spatial domain algorithm i.e.- Prewitt, Sobel, Laplacian of Gaussian, and kirsch to show the robustness and efficiency of the proposed morphological algorithm. Moreover, we did a quality-based comparison of all edge detection algorithms by using different quality matrix. After doing the comparison, it is clear that the performance of the proposed algorithm is better than the existing one in terms of time, efficiency, and quality.

Keywords: Image processing, Edge detection, Pattern Recognition, Spatial filter, Morphological operation

Author`s Contribution	*Address of Correspondence*	Article info.
[1,3] Data analysis, interpretation and manuscript writing, Active participation in data collection, [2]Conception, synthesis, planning of research, [4]Interpretation, and discussion	*Isma Hamid* *Email: ismahamid@ntu.edu.pk*	*Received: July 19, 2021* *Accepted: November 26, 2021* *Published: December 30, 2021*

Cite this article: *Nawaz Q, Younas M, hamid I, Gilani SM. Ana analysis of Edge Detection Methodologies in Spatial and Morphological Perspective. J. inf. commun. technol. robot. appl.2021; 12(2):10-11.*

Funding Source: *Nil*
Conflict of Interest: *Nil*

INTRODUCTION

Nowadays, image processing methods are used to analyze and manipulated digital images. Generally, images contain different types of information like size, orientation, color, and objects. In image processing, image edge detection is one of the important tasks that is used for quality enhancement and true edge detection. The edge detection process is very essential in image analysis because it is used for image segmentation, feature extraction, and texture extraction and these are their basic objectives. To extract the information from the image, it is mandatory to identify the strong low-level features of an image which are known as its edges[1]. The Features are the small piece of information that contain the image pixel values that represent low-level strong features of any image for detection and edges are one of the most important and meaningful features in an image. Therefore, detection of edge or small features is very difficult and sometimes impossible under various conditions like lightening, blurring, and noise. To understand the features of any image, an edge detection algorithm is used to trace the edges and boundaries of the image[2].

In simple, edge detection methods are used to transform the original image into an edge image by using different operations also known as algorithms [3]. In the transformation process, a digital image is used as input, and it can be processed by applying different operations to function to get the desired output. The output of the edge detection algorithm can be an image, set of characteristics, or simple pixel information. The basic goal of edge detections algorithms is to identify striking pixels of an image that may become the cause of an unknown change in image appearance.

To detect the true of edges of any digital image, in this paper, we propose a simple and efficient algorithm for edge detection, which is based on general image processing steps with a morphological filter to extract the edge information of edge location, meaningful and important information and pixel orientation[4]. The algorithm is based on four major steps, that are very easy to implement, require less computation time, and can save the information on true edges of different types of images i.e.- simple, low resolution, and noisy images. The rest of the paper is organized as follows: the existing method of edge detection is discussed in section 2, and the proposed algorithm is discussed in section 3. In section 4, we compare the proposed algorithm with the existing edge detection filter of the same domain to show the novelty of the algorithm. In section 5, we summarize the paper.

LITERATURE REVIEW

In the last decades, many researchers had tried to develop an edge detection algorithm by using image processing operations to detect and highlight the edges or boundaries of an image. That's why many researchers have been going on with developing a more efficient and faster algorithm for edge detection using existing filters[5]. The algorithms that are used for identifying the edges are completely based on discontinuous pixels. In this review, we discussed all the methods which are used for edge detection. A couple of years ago, simple gradients-based filters were used to detect the edges based on kernel [6]. The first gradient filter for edge detection was Robert designed by Lawrence Robert in 1963. The Robert edge detection filter was completely based on image gradients, and the gradient was calculated by adding the dissimilarities pixel between transversely adjacent pixels. This filter was very sensitive to noise, less reliable, and less accurate. By considering the drawbacks of the Robert filter, researchers were designed different gradient-based edge detection filters for better results i.e.- Sobel, Prewitt. The Sobel and Prewitt edge detection filter has used the kernel of 3 x 3 to provide better results and high detection accuracy. These filters were better than Robert, but not suitable of low- resolution and noisy images [7].

To maintain the accuracy of localization, the author [4] designed a log Sobel filter to analyze and detection of image edges. The log Sobel filter is related to the Prewitt, Sobel, and Robert filter. It processes the image with luminosity and logarithm and is suitable for fast detection. The log Sobel filter provides fast detection, but it is also sensitive to noise and has a high false detection rate. The authors' Al. had proposed a novel edge detection algorithm by using genetic programming to extract low-level feature information. This method was good for low resolution and less noisy images, but the false and missing edges detection is very high from highly noisy and blur images [8].

To reduce the risk of false detection, the author Anjum et al. used a canny filter for efficient and accurate edge detection. The canny filter operates the entire image and computes the gradient by using a second-order derivative. This filterable removes the dependency between the low-level pixels and blocks and processed the full image in a parallel way and it proved a good and better detection filter instead of the existing gradient filter. Edge detection through the canny filter is a little bit expensive in terms of speed and time [9]. In another paper, author Qinghang et al. designed an edge detection by using the Gaussian method to overcome noise sensitivity and false detection problems. By enhancing the gaussian, their algorithm has shown a better result with minimum processing time [10].

The proposed edge detection algorithm is based on three basic steps. (i) image acquisition (ii) image Enhancement (iii) proposed edge detection filter. In the first step, we get the input image for processing. In the second step, the Image enhancement steps are applied to the input image for grayscale conversion and noise removal to provide a better result. In the third step, the proposed edge detection filter is implemented on the enhanced image. After that, we applied image sharpening filer on the detected to get the fine detail of the edges. The complete procedure of proposed edge detection is shown in Fig 1, and detail of each step is given in subheadings 3.1, 3.2, and 3.3 respectively.

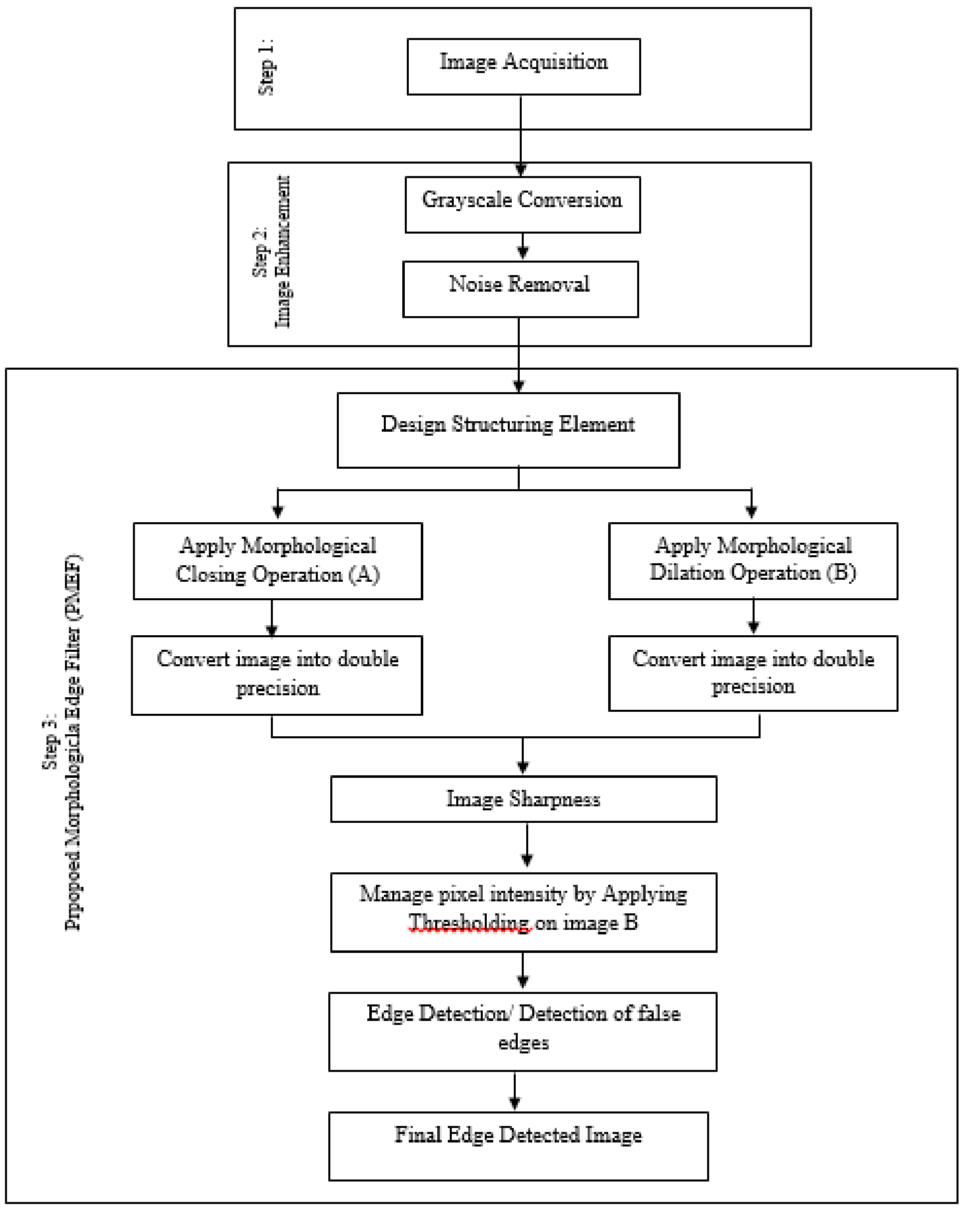


Figure 1: Proposed Edge Detection Algorithm

Image Acquisition

The image used in image processing is known as a digital image that consists of discrete picture elements called pixels. In a digital image, each pixel is associated with the digital number which contains the scene information or image information and is stored in the form of a matrix. In the image acquisition step, we acquired a color image which is used as input for algorithms. The color image is represented by three matrices, where each matrix specifies the value or pixel information of Red, Green, and Blue that make up the image. This color system is also called the RGB or 2D color image. The pixel values of an RGB image in a matrix between 0 and 255, determine the intensity of the pixel [11]. In digital image processing, any color image is an M * N * 3 array of the color pixel, where the spatial location of red, green, and blue is associated with three values. The pixel intensity of the red, green, and blue channels is described in equation 1.

$$I = R(i,j) + G(i,j) + B(i,j) \quad (1)$$

In the above equation I an image of (M * N * 3), R, G and B represent the color of image and (i,j) is their pixel values.

Image Enhancement

The image enhancement steps are used to improve the quality of the input image to provide a better result. We used general image processing steps to enhance the image which is (i) Grayscale Conversion and (ii) Noise removal. The input image is converted into the grayscale form because edge detection algorithms perform best with a grayscale image as compared to a color image[1]. The grayscale conversion is performed by using equation 2.

$$G(i,j) = 0.2989R(i,j) + 0.5870G(i,j) + 0.1140B(i,j) \quad (2)$$

In the above equation, G(i,j) is the grayscale intensity of the pixel (i,j), and R(i,j), B(i,j), and G(i,j) is the pixel intensity of red, green and blue channel of any color image. After converting the image into the grayscale form the next step is to remove the noise of the image by using a Bilateral filter. A bilateral filter is an edge-preserving, noise-reducing, and nonlinear smoothing filter, which is suitable for both spatial domain and frequency domain images. It replaces each pixel value of images with an average intensity value from neighborhood pixels or nearby pixels [10]. Mathematically, the pixel value of the image by using the bilateral filter at a specific pixel location is calculated by using Equation 3.

$$N(i,j) = \frac{1}{N_C} \sum_{y \epsilon B(x,y)} e^{\frac{-\|(y-i)-(x-j)\|^2}{2\sigma_d^2}} e^{\frac{-|I(x,y)-N(i,j)|^2}{2\sigma_r^2}} N(Y) \quad (3)$$

In the above equation, σd and σr represent the fall-out weights, B(x, y) are the spatial neighborhood pixel of image N (i,j), and N_C is the normalization constant.

Edge Detection Algorithm

Crack Highlight by using the Edge detection process is very essential in image analysis because it is used for image segmentation, crack extraction, and highlighting false edges [12]. Edge detection methods are used to transform the original image into edge images to provide the basic information of images like edges, size, and shape. In image processing, mostly edge detection Algorithms are applied to the grayscale image to get more accurate and prescribe information to the image [13]. This is a fundamental process to detect the boundaries among the objects. background in an image, and significant discontinuities of pixel and intensity values of the image. The morphologically based edge detection is performed by using morphological operations which are erosion, dilation, opening, and closing. Morphological operations can manage and remove noise from images without any other algorithm or operations. Typically, Morphological operations are used for binary image processing i.e. (black and white), but now these operations are also used for variations in a grayscale image. Morphological operations take two images as an input (i) the image which is to be processed and (ii) the structuring element. In morphological image processing operations, the grayscale image is treated as a three-dimensional object, the black color represents the lower areas or darker areas, and the white color shows higher areas or brighter areas of the image. The second input also called the structuring element is very mandatory to perform any morphological operation on an image. The structuring element is a set of coordinate points which is used to enhance and modify the input image [8]. Different types of structuring elements such as disk, square, rectangular, and line can be used with different sizes such as 1,2, 3, …., n to modify the image. The most common morphological operation that is used for crack detection is closing and dilation. These are the two basic and most widely used operations of morphological in spatial domain methods. The main advantage of morphological operation is that it automatically removes the noise and unwanted artifacts from the image. These operations work with the structuring element, so firstly we define the structuring element to process the image [14]. Different types of strutting elements with different radii can be used with it. In this proposed method, we used the square shape structuring element with a radius size of 3 as shown in figure 2. After performing the dilation and closing operation, convert both images in double precision to get a more accurate detection as shown in the figure. The Detail of closing and dilation is given in subheadings.

Edge Detection Algorithm

Crack Highlight by using the Edge detection process is very essential in image analysis because it is used for image segmentation, crack extraction, and highlighting false edges [12]. Edge detection methods are used to transform the original image into edge images to provide the basic information of images like edges, size, and shape. In image processing, mostly edge detection Algorithms are applied to the grayscale image to get more accurate and prescribe information to the image [13]. This is a fundamental process to detect the boundaries among the objects. background in an image, and significant discontinuities of pixel and intensity values of the image. The morphologically based edge detection is performed by using morphological operations which are erosion, dilation, opening, and closing. Morphological operations can manage and remove noise from images without any other algorithm or operations. Typically, Morphological operations are used for binary image processing i.e. (black and white), but now these operations are also used for variations in a grayscale image. Morphological operations take two images as an input (i) the image which is to be processed and (ii) the structuring element. In morphological image processing operations, the grayscale image is treated as a three-dimensional object, the black color represents the lower areas or darker areas, and the white color shows a higher area or brighter area of the image. The second input also called the structuring element is very mandatory to perform any morphological operation on an image. The structuring element is a set of coordinate points which is used to enhance and modify the input image [8]. Different types of structuring elements such as disk, square, rectangular, and line can be used with different sizes such as 1,2, 3, …., n to modify the image. The most common morphological operation that is used for crack detection is closing and dilation. These are the two basic and most widely used operations of morphological in spatial domain methods. The main advantage of morphological operation is that it automatically removes the noise and unwanted artifacts from the image. These operations work with the structuring element, so firstly we define the structuring element to process the image [14]. Different types of a strutting elements with different radius can be used with it. In this proposed method, we used the square shape structuring element with a radius size of 3 as shown in figure 2. After performing the dilation and closing operation, convert both images in double precision to get a more accurate detection as shown in the figure. The Detail of closing and dilation is given in subheadings.

$$\mathrm{SE} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Figure 2: square Shape Structuring element of size 3

After selecting the size and dimension of the structuring element, the algorithm performed the closing and dilation operation on the enhanced image which we get after step 2. The dilation operation is used to expand the image foreground by selecting the maximum neighborhood value instructing element. In simple, the dilation process is just like spatial convolution. The general equation of dilation operation is represented in equations (4) and (5).

$$(E \oplus SE)(x,y) = \max\{E(x-x`, y-y`) \mid (x`, y`) \epsilon S_{SE}\}$$

$$B = E \oplus SE \qquad (5)$$

where SE is the structuring element, S_{SE} is the domain of the structuring element, E is the enhanced image, $E(x,y)$ is supposed the $-\infty$ outside the image domain, $(x`, y`)$ is the domain of SE which belongs to S_{SE}, this symbol $\oplus$ represents the dilation operation and B is the eroded image. The closing operation is used to remove the internal noise of the image without affecting the background. In closing morphological operation, dilation is followed by erosion, which is a process with the structuring element. In simple, the closing operation eliminates the small holes in the image and fills gaps or missing pixel values [12]. The general equation of grayscale closing operation is represented in the equation.

$$A = (E \cdot SE) = (E \oplus SE) \ominus SE \qquad (6)$$

After applying the morphological operation, the Image sharpening step or technique is used to highlight the edges and fine detail of the image to manage the resolution and acutance of the image. The resolution is the size, pixel, and intensity of the image and acutance refers to the contrast of the image, and both factors handle with the help of the image sharpness technique. In the proposed algorithms, the unsharp method is used for image sharpness and edge enhancement on both dilated and closing images [15]. Then, the resulting image a(x, y) and B(x, y) contain a range of different pixels values which require scaling. To manage the intensity and contrast of the compressed image, scaling of the image is performed with the pixel threshold function [16]. The pixel threshold operation on image A(x, y) is performed by using equation (7)

$$A'(x,y) = A(x,y) * T1 \qquad (7)$$

In the above equation, A(x,y) are the points of dilated image pixel values and T1 is the threshold value which we calculated by using equation (8).

$$T1 = max(A(x,y)) + min(A(x,y)) \qquad (8)$$

In the proposed algorithm, we used 1.02 as the threshold value of the T1 square image. After performing threshold operations on a dilated image, then at the final stage subtract the closed image from the dilated image, to get the final image.

$$\text{Edge Highlight image(E)} = \text{A}'(\text{x,y}) - \text{B}(\text{x,y})$$

RESULT AND ANALYSIS

In this section, we compared the proposed algorithm with different spatial filters and morphological filters. The filter which we selected for comparison includes Prewitt, Sobel, Kirsch, and Laplacian of Gaussian [17]. The gradient of an image is obtained by first-order partial derivative which is also called the convolution operation, *f(x, y)* is assumed as the local maxima in the direction of the image. the gradient of an image *f(x,y)* is defined as a vector [18] shown in eq 10.

$$\nabla f = \begin{bmatrix} gx \\ gy \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \qquad (10)$$

The magnitude of the vector is obtained by using the formula that define in eq 11.

$$\nabla f = mag\sqrt{{g_x}^2 + {g_y}^2} \qquad (11)$$

The Prewitt edge detection method was used to evaluate the intensity of the image and provide the solution to increasing the intensity value from dark to light. The 3 x 3 kernel size used for the Prewitt filter to find the first derivative [19] which is given below

-1	-1	-1
0	0	0
1	1	1

P_x

-1	0	1
-1	0	1
-1	0	1

P_y

The second filter which we used for comparison was Sobel Operator which also works like the Prewitt method by giving some extra weight to manage the noise in the image [20]. The mask that is used in the Sobel method for convolution is following

-1	-2	-1
0	0	0
1	2	1

S_x

-1	0	1
-2	0	2
-1	0	1

S_y

The kirsch operator is a non-linear edge detection method that finds the maximum edge by using convolution kernels [21]. This method consists of 8 kernels which are the following.

-3	-3	5
-3	0	5
-3	-3	5

K1

-3	5	5
-3	0	5
-3	-3	-3

K2

5	5	5
-3	0	-3
-3	-3	-3

K3

5	-3	-3
5	0	-3
5	-3	-3

K4

5	5	5
-3	0	-3
-3	-3	-3

K5

-3	-3	-3
5	0	-3
5	5	-3

K6

-3	-3	-3
-3	0	5
-3	5	5

K7

-3	-3	-3
-3	0	-5
5	5	5

K8

Laplacian of Gaussian operator edge detection method based on the second-order derivative. In this method, a

Gaussian filter is used to remove the noisy artifact from the image Laplacian operator is used to take the second derivative of the image to detect the sharp edges [22]. The Gaussian operation is defined by using the following formula is given in eq 12.

$$LOG = \frac{\partial}{\partial x^2} G(x,y) \frac{\partial}{\partial y^2} G(x,y) \tag{12}$$

The mask that is used for the digital implementation of Laplacian of Gaussian filter [23] is shown below.

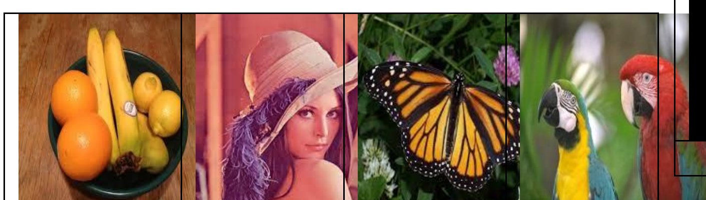
1	1	1
1	-8	1
1	1	1

LGx

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

LGy

This filter is mostly used to detect the pixel whether its value is light or dark [17]. The images which we used for comparison are shown in figure 3.



(a) (b) (c) (d)

Figure 3: Input image Dataset

Visual results of all existing filters with proposed algorithms are shown in fig 4, 5, 6, and 7 respectively. It can be observed that the final edge detected image after applying the proposed algorithm is better, clearer, and provides a noiseless and smoother image as compared to the existing method.

(a) (b) (c) (d) (e)

Figure 4: Visualization result of edge detection

(a) (b) (c) (d) (e)

Figure 5: Visualization result of edge detection on image 3(b)

(a) (b) (c) (d) (e)

Figure 6: Visualization result of edge detection on image 3(c)

(a) (b) (c) (d) (e)

Figure 7: Visualization result of edge detection on image 3(d)

Figure 4, 5, 6, and 7, (a) show the result of the Prewitt filter, (b) represent the result of the Sobel filter, (c) represent the result of the Laplacian filter, (d) shows the result of kirsch filter and (e) shows the result of proposed morphological edge detection filter (PMEF). Furthermore, to show the novelty of the proposed algorithm we also perform the quality comparison [24], we used the five most common and mostly used image quality matrix SE, PSNR [25], SSIM, FSIM, and DSSIM[26]. In Table MEF stands for Proposed Morphological Edge Filter. mathematical detail of all these quality matrices is n in Equations 13, 14, 15, 16, and 17 respectively.

$$MSE = \frac{\sum[n(x,y) - S(x,y)]}{x * y}$$

$$RMSE = \sqrt{MSE} \tag{13}$$

In equation 11, x and y represent the number of rows and columns of edge detected image, S is considered as the input image and S is the final output image which is also called the smooth compressed image. RMSE (Root Mean square Error) is the square root of MSE (Mean Square Error)[25].

$$PSNR = 10log_{10}\left(\frac{M^2}{MSE}\right) \qquad (14)$$

Where M is the maximal value of the reference image and the mean square error which we calculated by using the eq 8.

$$FSIM = \frac{\sum_{x \in \Omega} S_L(x).PC_m(x)}{\sum_{x \in \Omega} PC_m(x)} \qquad (15)$$

Equation 10, Ω containing the value of the whole image in the form of the spatial domain, $S_L(x)$ is the overall similarity index of both edge detected and the reference image and PC_m is determine phase contingency that effect due to light variation [27].

$$SSIM\ (x,y) = [l(x,y)]^{\alpha}.[c(x,y)]^{\beta}.[s(x,y)]^{\gamma}$$

Equation 8, $l, c\ and\ s$ used to compare the luminous brightness factor, contrast factor for the dark and bright region, and structure pattern with a positive constant factor $\alpha, \beta\ and\ \gamma$.

$$DSSIM(x,y) = \frac{1 - SSIM(x,y)}{2}$$

Where SSIM is the structural similarity index [25] which we calculated by using eq 15. DSSIM is used to deduce the value of SSIM. The quality-based comparison of the existing edge detection filter with PMEF is shown in Table 1, and the red highlighter value represents the best performance. The graphical representation of filter performance with quality assessment matrices is shown in Figures 8,9,10 and 11 respectively.

Table 1 – Comparison of Edge Detection Algorithm

Images	Filter Name	RMSE	PSNR	FSIM	SSIM	DSSIM
Figure 3(a)	Prewitt	85.5548	49.6351	0.3160	0.9625	0.0187
	Sobel	50.7416	54.9672	0.5661	0.9700	0.0150
	Kirsch	29.9082	55.4121	0.6059	0.9787	0.0107
	LoG	63.0968	54.5279	0.5286	0.9776	0.0112
	PMEF	**10.1341**	**56.0196**	**0.7232**	**0.9817**	**0.0091**
Figure 3(b)	Prewitt	114.764	46.9490	0.3023	0.9325	0.0337
	Sobel	72.4082	52.7758	0.4642	0.9618	0.0191
	Kirsch	34.9827	54.1076	0.5884	0.9750	0.0125
	LoG	96.2750	53.8535	0.4045	0.9724	0.0188
	PMEF	**9.9996**	**55.1153**	**0.7469**	**0.9778**	**0.0111**
Figure 3(c)	Prewitt	155.184	42.9972	0.2483	0.8642	0.0679
	Sobel	90.133	58.2373	0.3536	0.9873	0.0064
	Kirsch	74.7479	56.3626	0.5571	0.9916	0.0042
	LoG	96.2750	53.4696	0.3202	0.9835	0.0082
	PMEF	**14.7923**	**61.0116**	**0.6257**	**0.9956**	**0.0022**
Figure 3(d)	Prewitt	84.6407	49.9499	0.3401	0.9688	0.0156
	Sobel	81.8976	54.8173	0.5801	0.9691	0.0154
	Kirsch	23.4381	55.6878	0.6260	0.9779	0.0111
	LoG	55.1793	54.8251	0.5471	0.9804	0.0098
	PMEF	**7.3299**	**56.1566**	**0.7361**	**0.9806**	**0.0097**

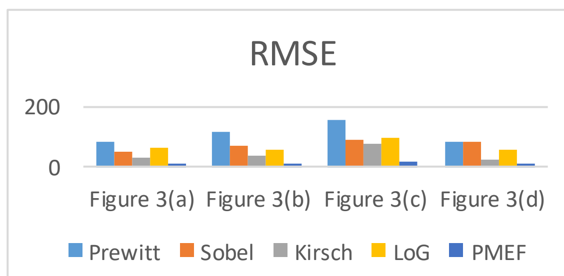
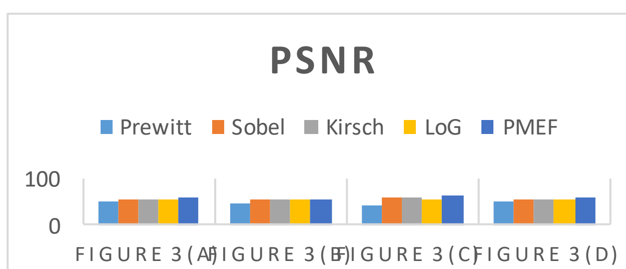


Figure 8: RMSE based Comparison Proposed method with existing



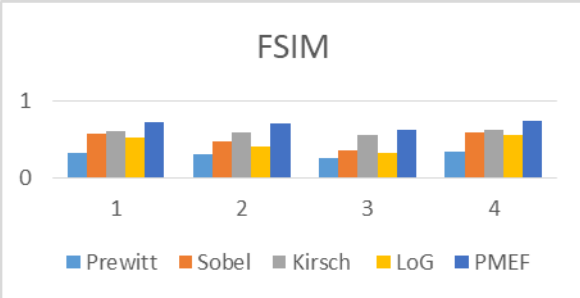


Figure 10: FSIM based Comparison Proposed method with existing

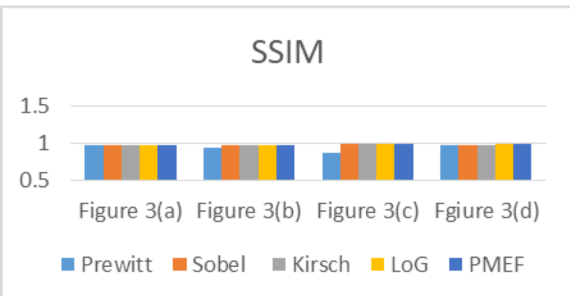


Figure 11: SSIM based Comparison Proposed method with existing

DISCUSSION

In this paper, we proposed a fast, efficient, and robust algorithm or filter for edge detection by using the morphological filter to find the fine details of edges. the proposed algorithm can get fine detail of edges from noisy, low-resolution, and blur images. To show the robustness and efficiency of the algorithm, we compared it with the different existing filters of edge detection such as Prewitt, Sobel, Laplacian of Gaussian (LoG), and Kirsch. A different Quality matrix such as RMSE, PSNR, SSIM, DSSIM, and FSIM is used to check the quality of all algorithms and then compare their result to find the best filter. After the result, we find that the proposed edge detection algorithm is better than the existing methods.

CONCLUSION

The proposed algorithm was designed to detect the edges of any image and get low-level feature information. The image enhancement steps were applied to remove the unwanted noise of the image and make it clear. The proposed algorithm achieved better results by comparing it with the existing algorithm with the help of different quality parameters or matrices i.e.- PSNR, SSIM, and RMSE. Various algorithm of edge detection was used for edge detection, but the error rate of the proposed algorithm was very less and lay within the accepted range. In the future, the proposed work or algorithm should be used with the deep learning method. Moreover, a morphologically based edge detection algorithm would also be used for real-time images or real-time applications i.e.- detection of a road defect, glass scratch, and walls

REFERENCES

1. M. Hagara, O. Ondracek, P. Kubinec, and R. Stojanovic, ‘Edge Detection in JPEG Grayscale Images’, in 2019 8th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, Jun. 2019, pp. 1–4. doi: 10.1109/MECO.2019.8760195.
2. T. Zhu et al., ‘Generalized Spatial Differentiation from the Spin Hall Effect of Light and Its Application in Image Processing of Edge Detection, Phys. Rev. Applied, vol. 11, no. 3, p. 034043, Mar. 2019, doi: 10.1103/PhysRevApplied.11.034043.
3. R. M. Yousaf, H. A. Habib, H. Dawood, and S. Shafiq, ‘A Comparative Study of Various Edge Detection Methods’, in 2018 14th International Conference on Computational Intelligence and Security (CIS), Hangzhou, China, Nov. 2018, pp. 96–99. doi: 10.1109/CIS2018.2018.00029.
4. G. Yang and F. Xu, ‘Research and analysis of Image edge detection algorithm based on the MATLAB’, Procedia Engineering, vol. 15, pp. 1313–1318, 2011, doi: 10.1016/j.proeng.2011.08.243.
5. E. M. Dogo, O. J. Afolabi, N. I. Nwulu, B. Twala, and C. O. Aigbavboa, ‘A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks, in 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), Belgaum, India, Dec. 2018, pp. 92–99. doi: 10.1109/CTEMS.2018.8769211.
6. N.-D. Hoang and Q.-L. Nguyen, ‘Metaheuristic Optimized Edge Detection for Recognition of Concrete Wall Cracks: A Comparative Study on the Performances of Roberts, Prewitt, Canny, and Sobel Algorithms’, Advances in Civil Engineering, vol. 2018, pp. 1–16, Nov. 2018, doi: 10.1155/2018/7163580.
7. G. N. Chaple, R. D. Daruwala, and M. S. Gofane, ‘Comparisons of Robert, Prewitt, Sobel operator based edge detection methods for real-time use on FPGA’, in 2015 International Conference on Technologies for Sustainable Development (ICTSD), Mumbai, India, Feb. 2015, pp. 1–4. doi: 10.1109/ICTSD.2015.7095920.
8. W. S. AlAzawee, I. Abdel-Qader, and J. Abdel-Qader, ‘Using morphological operations — Erosion based algorithm for edge detection, in 2015 IEEE International Conference on Electro/Information Technology (EIT), Dekalb, IL, USA, May 2015, pp. 521–525. doi: 10.1109/EIT.2015.7293391.
9. P. Selvakumar and S. Hariganesh, ‘The performance analysis of edge detection algorithms for image processing, in 2016 International Conference on Computing Technologies and

Intelligent Data Engineering (ICCTIDE'16), Kovilpatti, India, Jan. 2016, pp. 1–5. doi: 10.1109/ICCTIDE.2016.7725371.
10. M. Juneja and P. S. Sandhu, 'Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain', IJCTE, pp. 614–621, 2009, doi: 10.7763/IJCTE.2009.V1.100.
11. V. K. Mishra, S. Kumar, and N. Shukla, 'Image Acquisition and Techniques to Perform Image Acquisition', SMSJ, vol. 9, no. 01, pp. 21–24, Jun. 2017, doi: 10.18090/samriddhi.v9i01.8333.
12. P. P. Acharjya, R. Das, and D. Ghoshal, "A study on image edge detection using the gradients", Int. J. Sci. Res. Publ., vol. 2, no. 12, pp. 2–6, 2012
13. Rashmi, M. Kumar, and R. Saxena, 'Algorithm and Technique on Various Edge Detection : A Survey', SIPIJ, vol. 4, no. 3, pp. 65–75, Jun. 2013, doi: 10.5121/sipij.2013.4306.
14. R. Alamsyah, A. D. Wiranata, and R. -, 'Defect Detection of Ceramic Tiles using Median Filtering, Morphological Techniques, Gray Level Co-occurrence Matrix, and K-Nearest Neighbor Method', Scientific Research Journal, vol. VII, no. IV, Apr. 2019, doi: 10.31364/SCIRJ/v7.i4.2019.P0419632.
15. A. Zaafouri, M. Sayadi, and F. Fnaiech, 'A developed unsharp masking method for images contrast enhancement', in Eighth International Multi-Conference on Systems, Signals & Devices, Sousse, Mar. 2011, pp. 1–6. doi: 10.1109/SSD.2011.5767378.
16. Z. Othman, A. Abdullah, F. Kasmin, and S. S. S. Ahmad, 'Road crack detection using adaptive multi-resolution thresholding techniques', TELKOMNIKA, vol. 17, no. 4, p. 1874, Aug. 2019, doi: 10.12928/telkomnika.v17i4.12755.
17. Q. Li and X. Liu, 'Novel Approach to Pavement Image Segmentation Based on Neighboring Difference Histogram Method', in 2008 Congress on Image and Signal Processing, Sanya, China, 2008, pp. 792–796. doi: 10.1109/CISP.2008.13.
18. Owotogbe, J.S., Ibiyemi, T.S. and Adu, B.A., 2019. Edge detection techniques on digital images-a review. Int J Innov Sci Res Technol, 4, pp.329-332.
19. P. Ganesan and G. Sajiv, 'A comprehensive study of edge detection for image processing applications, in 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, Mar. 2017, pp. 1–6. doi: 10.1109/ICIIECS.2017.8275968.
20. L. Han, Y. Tian, and Q. Qi, 'Research on edge detection algorithm based on improved Sobel operator', MATEC Web Conf., vol. 309, p. 03031, 2020, doi: 10.1051/matecconf/202030903031.
21. A. Abdullah, M. Ali, and H. Jawad, "Evaluation of the performance of kirsch and Robinson methods on different color spaces", International Journal of Simulation: Systems, Science and Technology. vol.20, no.3, pp. 1-9, 2019.
22. Z. Zainal Abidin, S. A. Asmai, Z. Abal Abas, N. A. Zakaria, and S. N. Ibrahim, 'Development of Edge Detection for Image Segmentation, IOP Conf. Ser.: Mater. Sci. Eng., vol. 864, no. 1, p. 012058, May 2020, doi: 10.1088/1757-899X/864/1/012058.
23. A. Dogra and P. Bhalla, 'Image Sharpening By Gaussian And Butterworth High Pass Filter', Biomed. Pharmacol. J., vol. 7, no. 2, pp. 707–713, Dec. 2014, doi: 10.13005/bpj/545.
24. S. R. Joshi and R. Koju, 'Study and comparison of edge detection algorithms, in 2012 Third Asian Himalayas International Conference on Internet, Kathmundu, Nepal, Nov. 2012, pp. 1–5. doi: 10.1109/AHICI.2012.6408439.
25. U. Sara, M. Akter, and M. S. Uddin, 'Image Quality Assessment through FSIM, SSIM, MSE, and PSNR—A Comparative Study', JCC, vol. 07, no. 03, pp. 8–18, 2019, doi: 10.4236/jcc.2019.73002.
26. Guan-Hao Chen, Chun-Ling Yang, Lai-Man Po, and Sheng-Li Xie, 'Edge-Based Structural Similarity for Image Quality Assessment', in 2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings, Toulouse, France, 2006, vol. 2, p. II-933-II–936. doi: 10.1109/ICASSP.2006.1660497.
27. D. Sadykova and A. P. James, 'Quality assessment metrics for edge detection and edge-aware filtering: A tutorial review', in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, Sep. 2017, pp. 2366–2369. doi: 10.1109/ICACCI.2017.8126200.