Edge Detection in Digital Images using Guided L_0 Smoothen Filter and Fuzzy Logic

Akshi Kumar[1] Sahil Raheja[2]

Dept. of Computer Science and Engineering, DTU, Delhi, India

[2]Research Scholar, Dept. of Information and Technology, Delhi Technological University, Delhi, India

Corresponding Author e mail and Contact Number: sahilrahejaphd@rediffmail.com ,

+91-9017080763

ABSTRACT

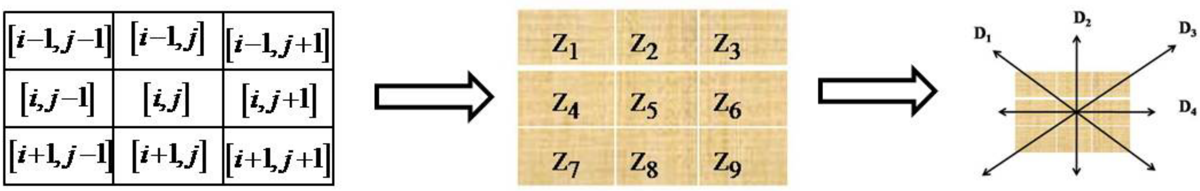
Image segmentation is an important process in computer vision. Recently fuzzy logic based edge detection is heavily investigated as by changing the number of rules edge detection can be improved. However, due to large colour variations in the images false edges are detected and even using fuzzy rules they cannot be reduced significantly. These falsely detected edges can be controlled by using smoothen filter while controlling the degree of smoothness. This paper, presents fuzzy logic based edge detection mechanism while using Guided L_0 smoothen filter for the smoothening of image under various degree of smoothens. Simulation results for edge detection is presented for Canny, Sobel, Fuzzy logic based edge detection and finally fuzzy logic edge detection with inclusion of L_0 smoothen filter. The results are compared with classical and modern methods. Simulation is performed on Berkley Segmentation Database (BSD) and USC-SIPI Image Database while considering more than 100 images. The obtained F-measure is as high as 0.848.

Keywords: Edge detection, Guided filter, Fuzzy Logic, Image processing, sparsity, L_0 smoothing filter

1. INTRODUCTION

Edge detection is a prominent area of interest in various fields of research and engineering. An edge can be characterized as a collection of associated pixels separating two distinct boundary regions [1]. An Edge could be defined as a local concept yet the boundary could be defined as a global concept. An ideal edge is a set of pixels whose intensity changes abruptly like step function. Blurry edges are likewise obtained by the elements such as issues or imperfections occurred during of sampling, optics, and image acquisition. In this way, we can closely observe that an edge possess a ramp-like profile [2]. The slope of the ramp is associated with the measure of blurriness. The length of the ramp is termed as the thickness of the edge. Sharp edges are thin while blurred edges are thick. If intensity is constant than first derivative is zero, while in ramp intensity profile derivative is constant. Additionally in the 2nd derivative, we can easily notice that it is negative along the light side of the edge while positive along the dark side of the edge such as delta functions. Moreover, we can observe it as zero along and outside the ramp [3].

Edge detection in a grey image is based greatest gradient. In discrete images gradient is defined as pixel intensity difference between any two pixels. Considering 3×3 mask as in figure 1(a), pixel position is denoted by (i,j) and around pixels are denoted by



(a) (b) (c)

Figure 1. Traditional Edge detection processes (a) 3×3 mask (b) z notation conversion (c) possible edge directions

Converting the pixels as in Figure 1(a) using mapping

$$z_1 = f(i-1, j-1), z_2 = f(i, j-1) \ldots \text{ and } z_9 = f(i+1, j+1) \quad (1)$$

The resultant image is shown in Figure 1(b).

The corresponding edge directions are shown in Figure 1(c). The absolute difference in 135^0 and 45^0 can be evaluated as

$$D_1 = |z_5 - z_1| + |z_9 - z_5| \text{ and } D_3 = |z_5 - z_3| + |z_7 - z_5| \quad (2)$$

Similarly in 0^0 and 90^0 the differences are

$$D_2 = |z_5 - z_2| + |z_8 - z_5| \text{ and } D_4 = |z_5 - z_4| + |z_6 - z_5| \quad (3)$$

Finally, edge can be calculated as

$$E_d = D_1 + D_2 + D_3 + D_4 \quad (4)$$

In binary images, it can likewise be characterized as the black pixels with one closest white neighbour [1].

The rest of the paper organizes as follows, in section 2, background and related work is presented. The proposed method is described in section 3 of the paper, in section 4, fuzzy expert system is described. In section 5 of the paper, results are presented; finally section 6 discusses the major conclusions of the paper.

2. BACKGROUND AND RELATED WORKS

Edge detection is a prominent area of interest in various fields of research and engineering [2]. Therefore, both edge enhancement and detection is important in various classes of engineering applications. A number of methodologies have been proposed for edge detection. Each one of these was introduced to deal with the limitations of the earlier techniques [3]. The typical strategies consolidate the application of linear time invariant filters. With the help of these kernel based filters, it is possible to recognize an edge due to the sudden variations in grey scale pixel intensities. The pioneer kernel based methods in edge detection are Canny [4], Sobel [5], Robert [6], Prewitt [7] etc. Genming and Bouzong [8] produced a 5×5 kernel to locate edges in a picture while considering a fixed threshold level. But results were not promising as fixed threshold was used. Ongoing researches fuses techniques based on artificial intelligent like artificial neural networks [9], ant colony optimization [10], particle swarm optimization and genetic algorithms etc [11]. Another technique is Fuzzy Set theory that has been utilized for edge detection [12-13]. Kim et al. [14] proposed an algorithm using a 3×3 kernel and a look up table. Kaur et al. discussed a method based on fuzzy rules; here sixteen fuzzy rules were used to characterize edge detection [15]. The outcomes for edge detection were quite accurate in images (with no noise) but fails in presence of noise. More experiments have been performed in higher type of fuzzy logic particularly fuzzy type-2 to oblige more noteworthy vulnerabilities [16, 17]. As of late, fuzzy based edge detection is contemplated by numerous analysts [18-20]. Recently, Convolutional Neural Networks based methods have gain popularity and some of the notable methods are Deep-Contour [21], DeepEdge [22], and CSCNN [23]. Holistically method have automatic learning capability are based on deep learning phenomenon. The other recent notable mechanisms in edge detection are: deep convolutional neural network [24], Fuzzy cellular automata [25], particle swarm optimization [26], cuckoo search optimization [27], Anisotropic Gaussian Kernels [28], convolution neural networks (CNNs) [29]. Edge detection based on single pixel imaging was proposed in [30]. In recently published work it is detailed artificial intelligence and CNN based edge detection methods have shown that these methods fail in presence of small perturbations [31]. In our recent work, we have shown that sharpening of image using guided filtering can improve edge detection significantly [32]. However, to correctly detect edges, sometime false edges need to be suppressed. This can be achieved using Guided L_0 Smoothen Filter. This paper, proposes a technique based on type 1 fuzzy logic and guided L_0 image filtering is used for the smoothening of the images. Using smoothening, strength of the unwanted edges can be suppressed. The proposed method is effective as it considers the advantages of both fuzzy logic design and guided image smoothening. The smoothening also very effective in case of small perturbation where deep learning based methods fails very easily. The fuzzy logic is very useful in edge detection because it decision making capability between partial true and partial false with respect to true and false edges.

3. PROPOSED METHOD

Smoothness of an image can change detected edges, as smoothness reduces amplitude variation therefore unwanted edges can be diminished. In the spirit of edge preserve smoothening, various methods are proposed over the period of time. In the similar context L_0 smoothing filter is proposed, in this method prominent edges are preserved by increasing the steepness of transition and diminishing the other edges, still maintaining the overall structure of an image. After smoothness of the image fuzzy logic based edge detection method is applied. For better understanding of the Guided L_0 smoothen filter we first discuss L_0 smoothing filter, thereafter L_0 gradiant minimization is detailed which sharpen the dominant edges. Finally, Guided L_0 smoothen filter is discussed which has the edge-preserving smoothing property. Therefore, Guided L_0 smoothen filter takes the advantages of L_0 gradiant minimization and guided filtering.

3.1 L_0 smoothing filter

In this part we discuss the basic concept of L_0 smoothing filter as discussed in [33]. In gradient based method we count intensity changes and can be defined as

$$c(I) = N\{i, j \| |I_{i,j} - I_{i-1,j}| \neq 0\} \tag{5}$$

where *i, j* and (*i* – 1), *j* represents neighboring samples (or pixels) indices. $|I_{i,j} - I_{i-1,j}|$ is the forward difference of the intensity also known as gradient w.r.t. *i*. The parameter $N\{\}$ is used to represent the counting operator which counts the number of *i* that satisfies $|I_{i,j} - I_{i-1,j}| \neq 0$. This is, the L_0 norm of gradient. We consider objective function as (omitting index *j*)

$$\min_I \sum_i (I_i - g_i)^2 \quad \text{s.t.,}\ c(I) = \alpha \tag{6}$$

The number of non-zero gradient is given by $c(I) = \alpha$ and the input discrete signal is denoted by variable g and its smoothed version is represented by variable I. Thus, more appropriate objective function which represents the constraint optimization would be

$$\min_{I} \sum_{i} (I_i - g_i)^2 + \lambda\, c(I) \tag{7}$$

This optimization is necessary to maintain the image structure as the value of α may be very large. The parameter λ is very important as it controls the sparsity of the image gradient and consequently the smoothness of the output image.

3.2 L_0 gradient minimization

With the help of L_0 gradient minimization, the L_0 norm could be directly optimized to have a piecewise steady output image [33]. It is quite useful in sharpening prominent edges through enhancing the steepness of transition. Therefore, following minimization problem (8) needs to be solved

$$\min_{I} \left\| I - I^* \right\|_2^2 + \lambda \left\| \nabla I \right\|_0 \tag{8}$$

In the above equation, I is output, ∇I represent the gradients of I, the parameter I^* is the observed image, and λ is a weight to control gradient. With the end purpose to overcome the issue of the objective function i.e., the discontinuity of the term $\left\| \nabla I \right\|_0$, auxiliary variable Ω is introduced to deal with ∇I , therefore (8) can be converted into following minimization problem:

$$\min_{I} \left\{ \left\| I - I^* \right\|_2^2 + \beta \left\| \Omega - \nabla I \right\|_0 \right\} + \lambda \left\| \Omega \right\|_0 \tag{9}$$

In the above equation, the parameter β controls the similarity between ΔI and Ω, and the degree of smoothness is handelled by λ.

3.3 Guided L_0 smoothing filter

Not very long time before Xu et al. introduced L_0 gradient minimization which sharpens the image while maintaining dominant edges [33]. Later, X. Ding et. al., propose a guided L_0 smoothing filter. It takes benefits of the properties of both L_0 gradient minimization and guided filter. This method is known as guided L_0 smoothing filter [34].

To begin, Using, I^k, the parameter Ω^k is optimized using

$$\min_{\delta^k} \beta^k \left\|\nabla I - \Omega^k\right\|_2^2 + \lambda \left\|\Omega^k\right\|_0 \quad (k = 1,2,3....) \tag{10}$$

We can solve the above equation as detailed in [34]

$$\Omega^k = \begin{cases} 0 & \nabla I^k \le \dfrac{\lambda}{\beta} \\ \nabla I^k & others \end{cases} \tag{11}$$

Now both Ω^k and I^k are known, we now evaluates I^{k+1} using equation (9) and (11):

$$\min_{I^{k+1}} \left\|I^{k+1} - I^*\right\|_2^2 + \beta^k \left\|\Omega^k - \nabla I^{k+1}\right\|_2^2 \tag{12}$$

The expression (12) while considering (eqn. 11) is equivalent to (13):

$$\begin{cases} \min\limits_{I^{k+1}} \left\|I^{k+1} - I^*\right\|_2^2 + \beta^k \left\|\nabla I^{k+1} - H.*\nabla I^k\right\|_2^2 \\ H = \begin{cases} 0 & \Omega^k = 0 \\ 1 & \Omega^k \ne 0 \end{cases} \qquad (k = 1,2,3........) \end{cases} \tag{13}$$

The objective function (eqn. 13) is quadratic therefore it has convex optimization issue. Thus, least square technique and Fourier transformation is used to solve it [34]:

The solution of eqn. 13 is

$$I^{k+1} = \mathrm{ifft}\left(\frac{\mathrm{fft}(I^*) + \beta\left(\mathrm{fft}(\partial_x^T)\,\mathrm{fft}(\Omega_x^k) + \mathrm{fft}(\partial_y^T)\,\mathrm{fft}(\Omega_y^k)\right)}{\mathrm{fft}(1) + \beta\left(\mathrm{fft}(\partial_x^T)\,\mathrm{fft}(\partial_x) + \mathrm{fft}(\partial_y^T)\,\mathrm{fft}(\partial_y)\right)} \right) \tag{14}$$

The parameter fft represent the fast Fourier transform operator and the parameter $ifft$ is used to represent its inverse. ∂x and ∂y implies difference operators in the horizontal and vertical directions, respectively.

Defining new variable 's' for smoothed image, then using, Ω^k and s^k we obtain s^{k+1} as

$$\begin{cases} \min\limits_{s^{k+1}} \left\| s^{k+1} - s^* \right\|_2^2 + \beta^k \left\| \nabla s^{k+1} - H.*\nabla s^k \right\|_2^2 \\ H = \begin{cases} 0 & \Omega^k = 0 \\ 1 & \Omega^k \neq 0 \end{cases} \qquad (k = 1,2,3........) \end{cases} \tag{15}$$

The solution of (15) is:

$$s^{k+1} = \mathrm{ifft}\left(\frac{\mathrm{fft}(s^*) + \beta\left(\mathrm{fft}(\partial_x^T)\mathrm{fft}(H.*\nabla s_x^k) + \mathrm{fft}(\partial_y^T)\mathrm{fft}(H.*\nabla s_y^k)\right)}{\mathrm{fft}(1) + \beta\left(\mathrm{fft}(\partial_x^T)\mathrm{fft}(\partial_x) + \mathrm{fft}(\partial_y^T)\mathrm{fft}(\partial_y)\right)} \right) \tag{16}$$

Algorithm:

Input: Image s^*, guided image I^*, parameters λ, $\beta_0, \beta_{\max}$,rate κ

Initialization: $I^1 \leftarrow I^*, s^1 \leftarrow s^*, \beta^1 \leftarrow \beta_0, k \leftarrow 1,$

repeat:

with I^k, solve Ω^k for in (13);

with I^k and Ω^k, solve for I^{k+1} in (14);

with s^k and Ω^k, solve for s^{k+1} in (16);

$\beta \leftarrow \kappa\beta, k++$;

Until $\beta > \beta_{\max}$

Output: s.

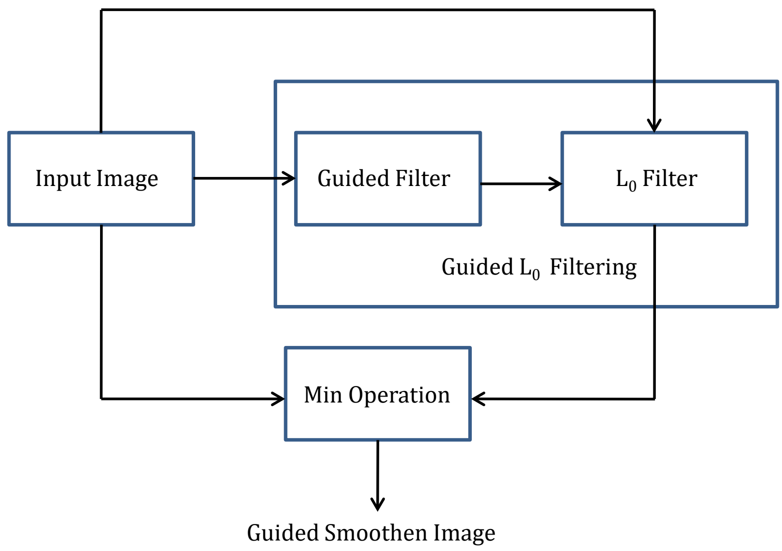


Figure 2. Guided L_0 smoothen filter

4. Fuzzy Expert System

Figure 3, depicts the basic architecture layout of proposed fuzzy expert system for edge detection. In this work three methods defined as M_1, M_2 and M_3 are presented. In method M_1 on the input image fuzzy logic is directly applied to detect edges. In method M_2 the input image is first passes through the L_0 smoothen filter and after this fuzzy logic is used to detect edges. In method M_3 the input image is first passes through the guided L_0 smoothen filter and then fuzzy logic is applied to detect edges. In Fuzzy logic based edge detection first input image is fuzzified using fuzzy input and output membership function and then apply IF-ELSE rules using Mamdani fuzzy inference engine and at the output de-fuzzification is done using centroid method to obtain crisp values to obtain desired results.

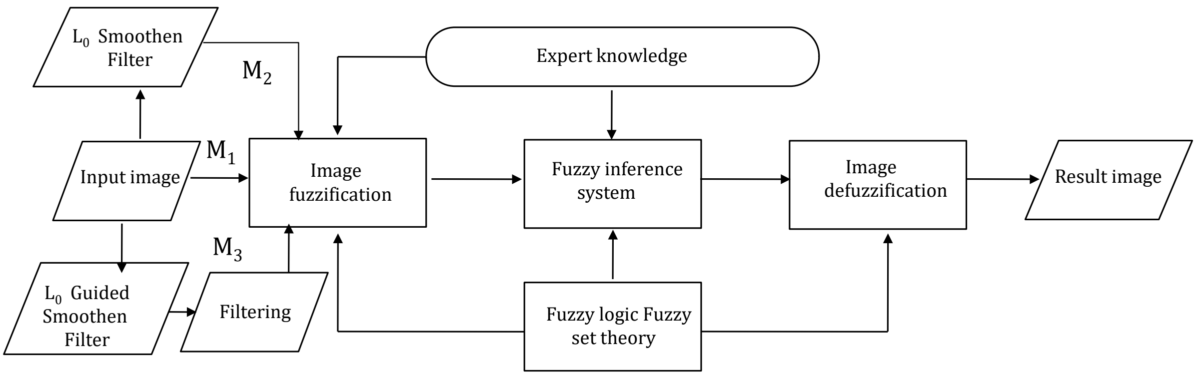


Figure 3. Schematic of proposed edge detection mechanism (M_1: Edge detection using fuzzy logic only, M_2: Edge detection using fuzzy logic and L_0 smoothen filter and M_3: Edge detection using fuzzy logic and Guided L_0 smoothen filter)

In this work, we use triangular membership function for both input and output, the input triangular function is defined as: $triangle(x;a,b,c) = \max\left(\min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right)$ as in figure 4. The output triangular membership functions for black, while and edge are very narrow as shown in figure 5.

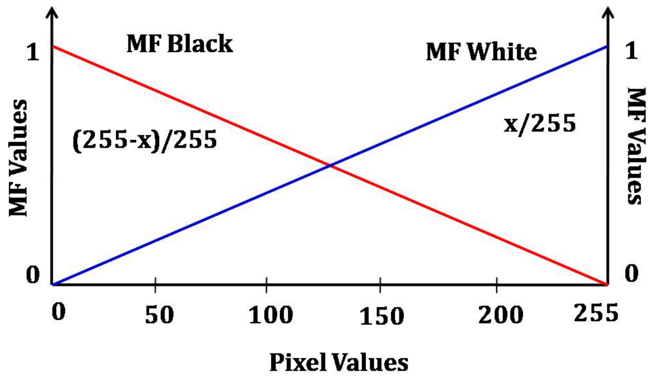


Figure 4. Membership function for black and white pixels values

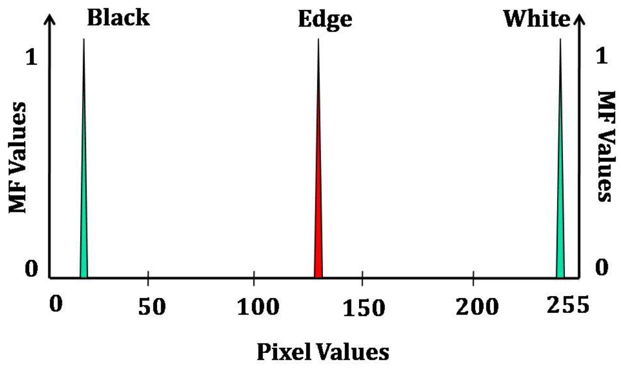


Figure 5. Output membership function for black, white and edge

Fuzzy Rules

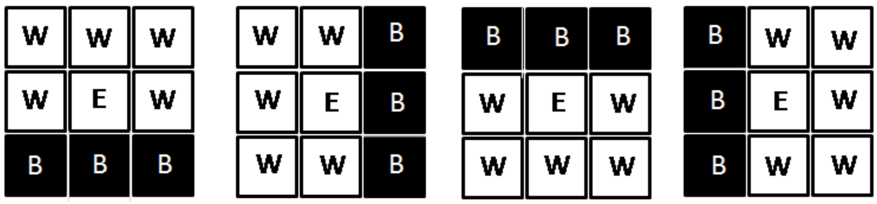
For the edge detection a total of 30 rules are defined. The rules are developed on a 3×3 mask as shown in figure 6.

$[i-1, j-1]$	$[i-1, j]$	$[i-1, j+1]$
$[i, j-1]$	$[i, j]$	$[i, j+1]$
$[i+1, j-1]$	$[i+1, j]$	$[i+1, j+1]$

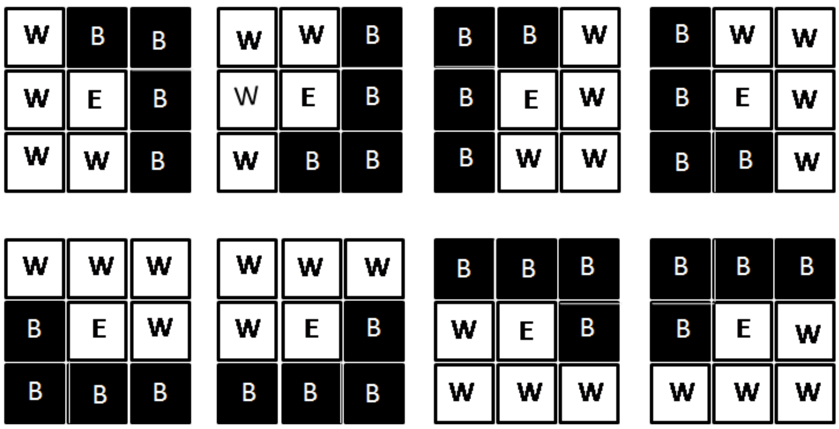
Figure 6. 3×3 mask for rule development

In the rule designing Img (i, j) represents pixel position, '*W*' represents white pixel and '*B*' represents black pixel value. For 3 black and 5 white pixels in neighbourhood 4 rules are defined, while for 4 black and 4 white pixels in neighbourhood, 8 rules are defined, for 5 black and 3 white pixels in the neighbourhood and for 6 black and 2 white pixels in the neighbourhood again for both the cases 8 rules are defined and finally for all black or white pixels in the neighbourhood 2 rules are defined. Various rules are detailed below:

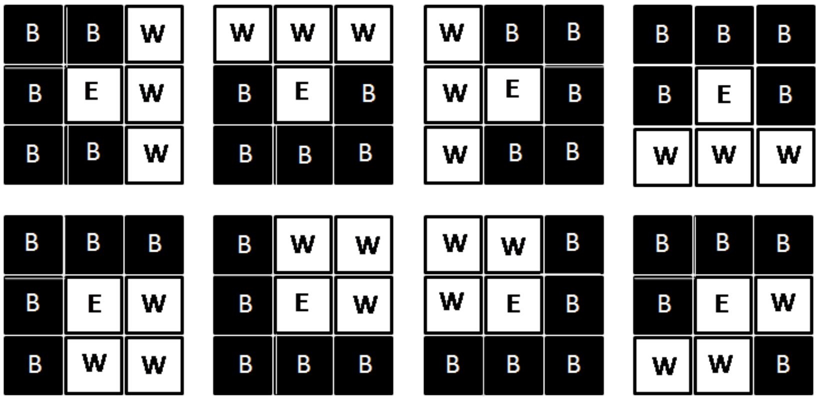
Rule with 3 black and 5 white pixels in neighbourhood



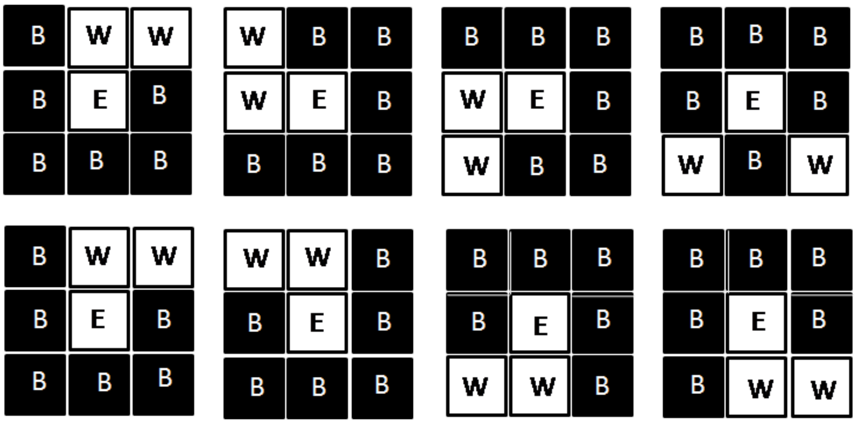
Rule with 4 black and 4 white pixels in neighbourhood



Rule with 5 black and 3 white pixels in the neighbourhood



Rule with 6 black and 2 white pixels in the neighbourhood



Rule with all black or white pixels in the neighbourhood

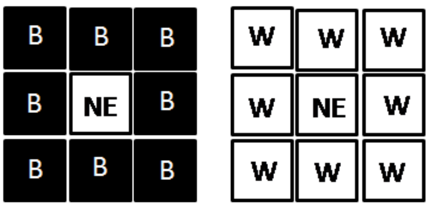


Figure 7. Fuzzy rules black (B), white (W) and edge (E) and non-edge (NE)

5. RESULTS

5.1 Performance Measures

It is not easy to define the general-purpose evaluation for edge detection, some edges are missed, some are falsely accepted, and some alters their positions. Therefore traditional measures like mean square error (MSE) and peak signal-to-noise ratio (PSNR) fails. In this paper four measures Pratt's Figure of Merit [34], Structure Similarity Image Metrics [35], F-Score [34] and Hausdorff Distance [34] is considered.

Pratt's Figure of Merit (FoM)

The Pratt's Figure of Merit evaluates edge location exactness in edge detected image in comparison to ground truth image, by measuring the displacement of edge points that are detected from an ideal edge. The Figure of Merit is characterized by

$$FoM = \frac{1}{\max(I_A, I_B)} \sum_{i=1}^{I_B} \frac{1}{1 + \mu d^2} \tag{19}$$

Here,

$I_A =$ ideal edge points (ground truth)

$I_B =$ edge points detected

d = displacement of detected edge from ideal edge

$\mu =$ scaling constant.

It is essential to note that these measurements binarize information before assessing images; this implies assessment is done over images that have lost data. The above mentioned metrics return esteems in the vicinity of 0 and 1, where 0 would imply that we have no similarity between detected image and reference image, and 1 implying that high closeness was detected, or in other words, each of the pixels present in one image edges are recognized at the same place in other image.

Structure Similarity Image Metrics (SSIM)

SSIM completes a greatly improved activity at measuring subjective image quality in comparison to MSE or PSNR. At a high state, SSIM endeavors to estimate the adjustment in luminance, contrast, and structure in a picture. The SSIM is given by

$$SSIM(A, B) = \frac{\left(2\mu_A \mu_B + k_1\right)(2\sigma_{AB} + k_2)}{\left(\mu_A^2 + \mu_B^2 + k_1\right)\left(\sigma_A^2 + \sigma_B^2 + k_2\right)} \tag{20}$$

Where,

μ is mean, σ^2 is variance, σ is cross-correlation term and rest terms are fixed constants.

Hausdorff Distance (HoD)

Considering two images $A=\{a_1, \ldots, a_n\}$ and $B=\{b_1, \ldots, b_n\}$, the Hausdorff distance calculated as:

$$H(A,B) = \max(d(A,B), d(B,A))$$

Where $d(A,B) = \max_{a \in A} \min_{b \in B} \| a - b \|$ (21)

The function *d(A, B)* is the directed Hausdorff distance from *A* to *B*. This method is based on distance among the points, and lesser distance means more closeness between the images.

F-Score

F-measure is a test of accuracy, in binary classification. It depends on both precession and recall to get test score. The maximum value of *F* is 1 with minimum as 0. In case of equal weightage, it is the harmonic mean of precession and recall.

	p	**n**
Y	True Positives	False Positives
N	True Negatives	False Negatives
Column Totals	**P**	**N**

Fig. 8. Characteristic matrix.

The important parameters are defined as:
Precision (P_r) $=TP/(TP+FP)$ and Recall (R_c) $= TP/(TP+FN)$

$$\text{F-score}=\frac{2}{\left(\frac{1}{P_r}+\frac{1}{R_c}\right)}$$

Where *TP* is true positive, *TN* is true negatives, *FP* is false positive and *FN* is false negative (Fig. 8).

5.2 Comparative Results (Proposed Methods)

However, to prove usefulness of proposed method and to cover wide varieties of experiments results are presented on a single image considered from Berkley Segmentation Database (BSD) [35]. Finally, in the comparative analysis results are presented using both BSD and USC-SIPI Image Database is considered [36].

Results M_1 method

(a) **(b)** **(c)**

Figure 9. (a) Original image, (b) ground truth (c) detected edges using fuzzy logic

In figure 9, results for edge detection are shown, using fuzzy logic only. It is clear from the figure that in edge detected images most of the edges are correctly detected with some more edges are falsely detected specially in 9(c) where at corners falsely detected edges can be easily seen.

Results M_2 method

The falsely detected edges can be suppressed using L_0 smoothen filtering, but it should be kept in mind that more and more smoothness may leads to false rejection of edges. Therefore degree of smoothness plays an important role in detected edges. The chosen parameters for smoothening are β_0=2λ, $\beta_{\max}$=100000, k=2 and λ varies (0.05-0.2).

(a) λ=0.05 **(b)** λ=0.1 **(c)** λ=0.2

Figure 10. Smoothen images using L_0 smoothing filter

In figure 10, on the top row smoothen images are shown, using various level of degree of smoothness. As L_0 is edge preserving filter therefore even in smoothened prominent edges are preserved. In bottom row results for edge detection using fuzzy logic is shown on the smoothen images. It is clear from the figure that as smoothness increases, the falsely detected edges are reduced while prominent edges are still preserved.

Results M_3 method

In figure 11, on the top row smoothen images are shown, using various level of degree of smoothness while considering guided filtering. As guided L_0 is edge preserving filter therefore even in smoothened prominent edges are preserved. On the bottom row edge detected images are shown. Therefore, by varying the degree of smoothness and applied methods (M_2 and M_3) the detected edges can be controlled. This is the main advantage of the proposed method where detected edges can be controlled.

(a) λ=0.05 **(b)** λ=0.1 **(c)** λ=0.2

Figure 11. Detected Edges in smoothen images using guided L_0 smoothing filter

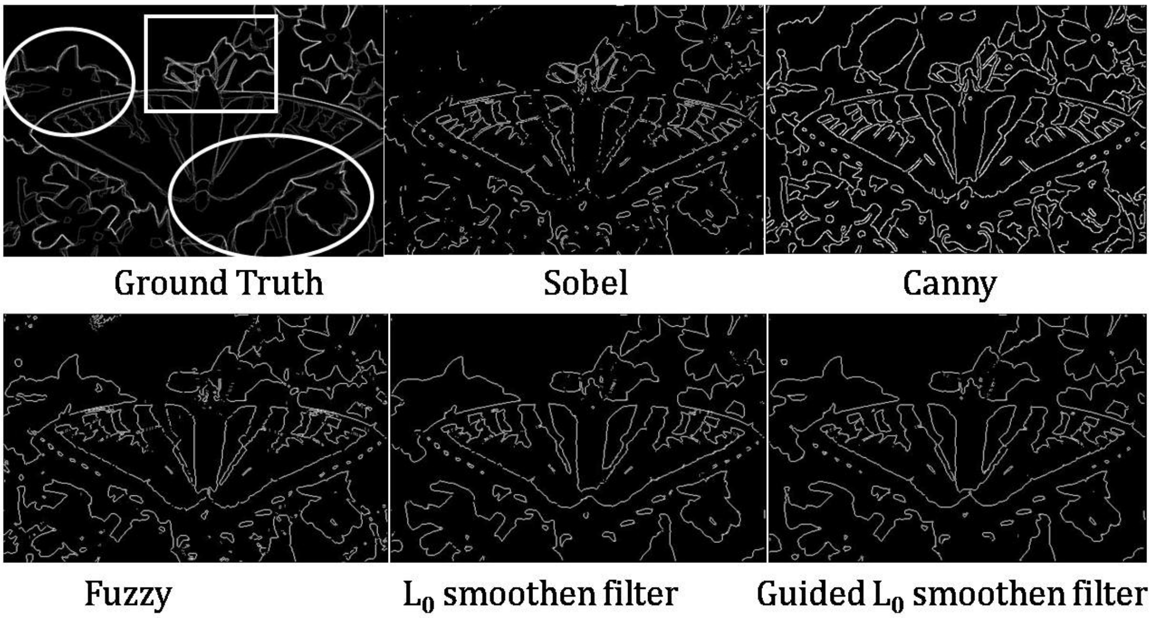


Figure 12. Comparison of various edge detection methods

In Figure 12, comparison of various edge detection methods is shown, where in Fig. 12 ground truth image is shown with marked white areas where notable changes takes place in different methods. In Sobel method, edges in circular mark region are not properly detected. In case of Canny method a large number of falsely detected edges are found. In fuzzy method obtained results are well in agreement with ground truth image, but variation in rectangular mark region can be seen, along with some more falsely accepted edges across the image. For

L_0 smoothen filter and fuzzy method obtained results are well in agreement with ground truth image and variation is small, and finally for guided L_0 smoothen filter and fuzzy method, here again results are very much similar to L_0 smoothen filter and fuzzy method with minor inclusion of false edges in circular region, however oval mark section is best detected. It is clear from the figures that the variation is so small it is difficult to judge from naked eyes, therefore to judge the performance of the method three performance measure as discussed above are used and obtained results are shown in figures 13 to 15.

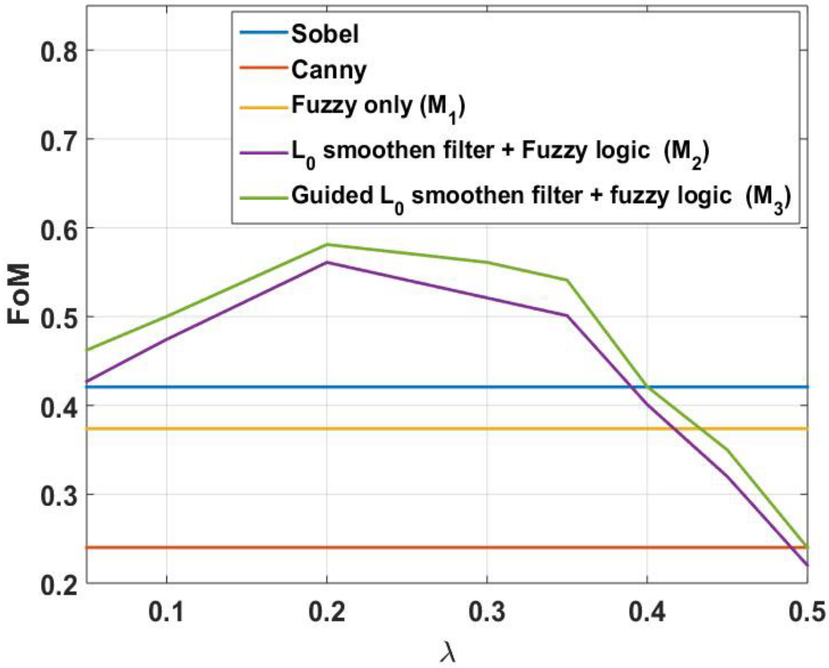


Figure 13. Comparison of various edge detection methods (FoM vs. λ)

In figures 13 to 15, results for various edge detection techniques under three performance metrics are shown. Ideally FoM, SSIM should be one and HoD should be equal to zero. It is clear from the tables that FoM is lowest for canny edge detection, i.e, more pixels shift their position in comparison to other considered technique. In figure 13, FoM vs. λ is shown, It is evident that best FoM is obtained under Guided L_0 smoothen filter + fuzzy logic case. It is also observable that as degree of smoothness increases FoM increases up to a limit thereafter it starts to decreases. The best value of FoM is obtained for $\lambda = 0.2$. In figure 14, SSIM vs. λ is shown, again SSIM is comparatively better, for Guided L_0 smoothen filter + fuzzy logic case, but first it increases with degree of smoothness, thereafter SSIM decreases. This is obvious more smoothening will lead to structural modifications. In the considered cases, best SSIM is obtained for degree of smoothness of 0.1. In the figure 15, Hausdorff Distance *is* plotted, for best case average minimum distance is 3.30. Again here with increases in λ, HoD first decreases and then increases. These results clearly reveal that Guided L_0 smoothen filter + fuzzy logic provides better results for edge detection.

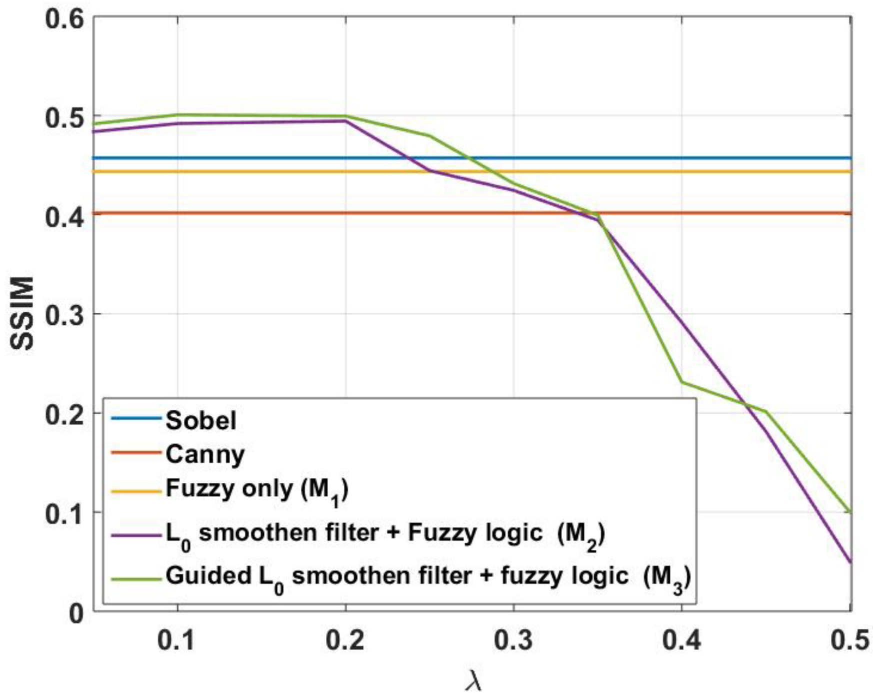


Figure 14. Comparison of various edge detection methods (SSIM vs. λ)

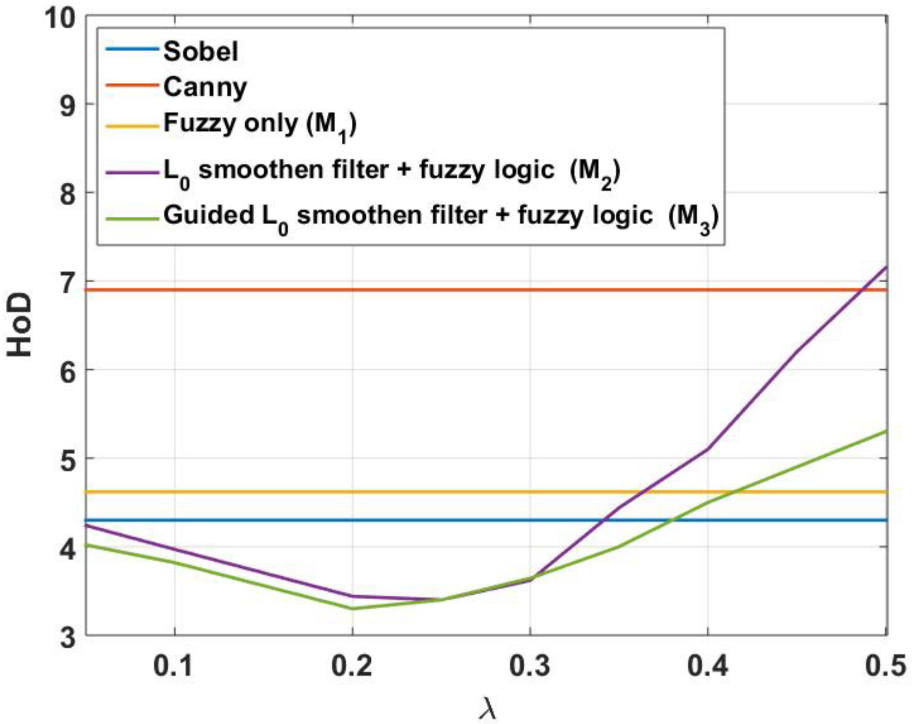


Figure 15. Comparison of various edge detection methods (HoD vs. λ)

5.3 Comparative with recent methods

In order to compare our method with more recent edge detection techniques, results are compared with Gonzalez, C. et. al, work where edges are detected using sobel and type-2 fuzzy logic method. Results were tested on more than 100 images, and a few reproduced results are shown in figure 14.

In figure 16, four rows and four columns are shown, first column shows original image, and in second column results are shown for canny edge detection. In column 3 and 4 results are shown for Gonzalez, C. et. al. and proposed method respectively. In Canny method large numbers of false edges are accepted leads to the lowering of F-score. It is also observable that when intensity difference is less Canny method fail to detect edges (fourth row second column). Our method is comparable to Gonzalez, C. methods with less number of falsely accepted edges (second row and second, third columns).

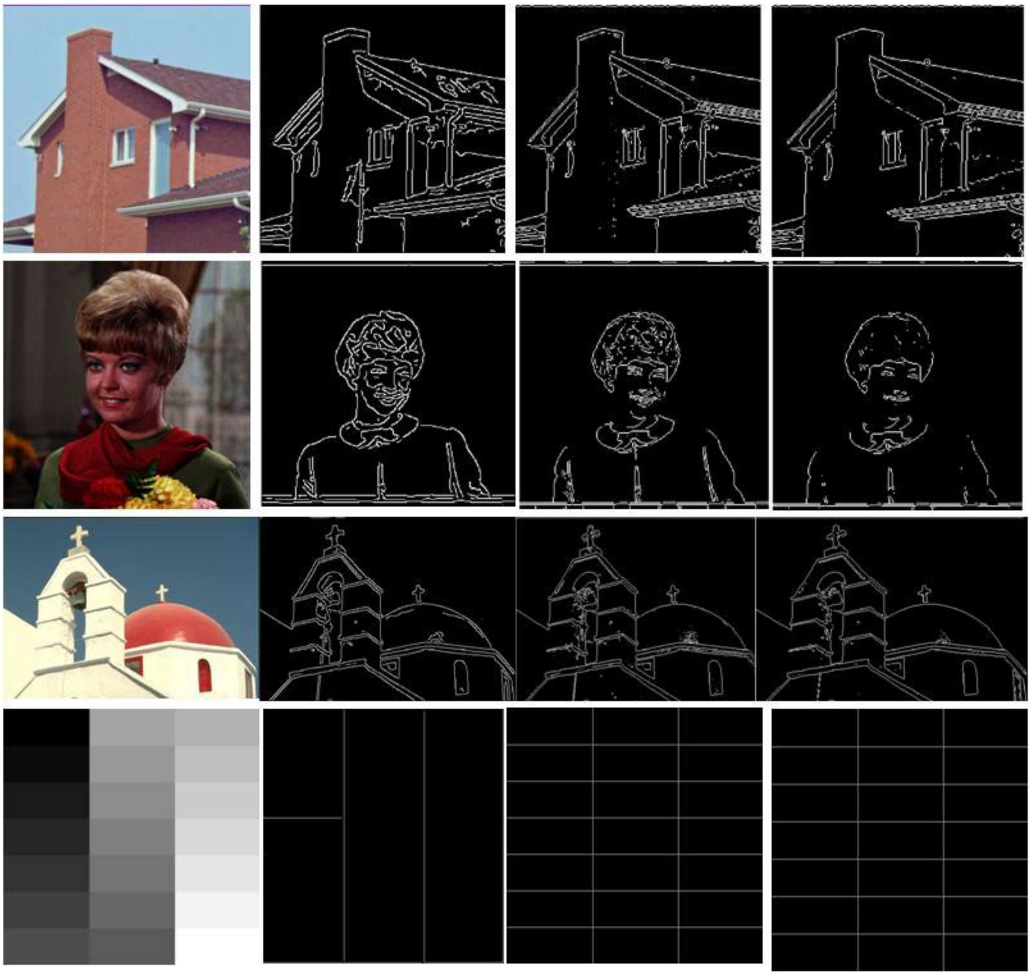


Figure 16. Each row left to right: Original, Canny, Gonzalez, C et.al., and proposed

In Table 1, notable and recently proposed methods are compared in terms of F-score. For classical methods canny and sobel F-measure is 0.49 and 0.40 respectively. The recently proposed learning based methods have F-score ranging from 0.63 to 0.78. A new kernel based method with singular value decomposition [43] has F-score as high as 0.83. Our proposed methods, L_0 smoothen filter + fuzzy logic (M_2) and Guided L_0 smoothen filter + fuzzy logic (M_3) attain F-score of 0.82 and 0.848 respectively.

Table 1: Comparison of notable edge detection methods

Methods	**Year**	**F-measure**
Canny [4]	[1996]	0.49
Sobel [37]	[2009]	0.40
BEL [38]	[2006]	0.63
gPb [39]	[2011]	0.71

Sketch Token [40]	[2013]	0.73
Structure Forest [41]	[2013]	0.71
Holistically-Nested Edge Detection [42]	[2015]	0.78
Gonzalez, C et.al. [43]	[2018]	0.83
Fuzzy only (M_1)	[2019]	0.77
L_0 smoothen filter + fuzzy logic (M_2)	[2019]	0.82
Guided L_0 smoothen filter + fuzzy logic (M_3)	[2019]	0.848

6. CONCLUSIONS

Edge detection has been an important area of research from past many years due to its utility in many fields like: image segmentation, medical, forensic and defense applications. In past years various methods based on kernels and soft computing based has been proposed, but they are all dependent on some kind of threshold mechanism and suffer from false acceptance/rejection. To deal such issues, in this paper a fuzzy based edge detection mechanism is proposed where edges are controlled using smoothen filters. In this paper two types of smoothen filters L_0 smoothen and guided L_0 smoothen filters are discussed, and using these filters prominent edges can be preserved and thus making edge detection more efficient. The performance of the edge detection methods are compared in terms of FoM, SSIM, F-measure and HoD, and it has been found that Guided L_0 smoothen filter + fuzzy logic produces better results. It is also found that smoothen should be done carefully and it should be within limit to obtain better results otherwise SSIM slips down and image quality goes down.

REFERENCES

1. Petrou M, & Petrou C.2010. Image processing: the fundamentals. John Wiley & Sons.
2. Bovik AC. Handbook of image and video processing. Academic press; 2010 Jul 21.
3. Poobathy D, & Chezian R.M. 2014.Edge detection operators: Peak signal to noise ratio based comparison. International Journal of Image, Graphics and Signal Processing. 1(10):55-61.
4. Canny J. 1986. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence. (6):679-98.
5. Matthews J. 2002. An introduction to edge detection: The sobel edge detector. Available at http://www.generation5.org/content/2002/im01.asp, 2002.
6. Roberts L.G. 1965. Machine perception of 3-D solids" ser. Optical and Electo-Optical Information Processing

7. Gonzalez R.C. & Woods RE. 2002. Digital image processing. 2nd ed. Prentice Hall.
8. Genming C, & Baozong Y. 1989. A new edge detector with thinning and noise resisting abilities. Journal of Electronics, 6(4):314-9.
9. Naumenko A, Lukin V, & Egiazarian K.2012. SAR-image edge detection using artificial neural network. Proc. International Conference on Mathematical Methods in Electromagnetic Theory (MMET), Pp. 508-512.
10. Verma O.P., & Sharma R.2011. An optimal edge detection using universal law of gravity and ant colony algorithm. Proc. Information and Communication Technologies (WICT), Pp. 507-511
11. Setayesh M, Zhang M, & Johnston M. 2012. Effects of static and dynamic topologies in particle swarm optimisation for edge detection in noisy images. Proc. Evolutionary Computation (CEC), Pp. 1-8.
12. Kuo Y.H., Lee C.S., & Liu C.C. 1997. A new fuzzy edge detection method for image enhancement. Proc. Sixth IEEE International Conference 2, Pp. 1069-1074.
13. El-Khamy S.E., Lotfy M., & El-Yamany N.2000. A modified fuzzy Sobel edge detector. Proc. Radio Science Conference, Pp. C32-1
14. Kim D.S., Lee W.H., & Kweon I.S. 2004. Automatic edge detection using 3× 3 ideal binary pixel patterns and fuzzy-based edge thresholding. Pattern Recognition Letters. 25(1):101-106.
15. Kaur E.K., Mutenja V, & Gill E.I. 2010. Fuzzy logic based image edge detection algorithm in MATLAB. International Journal of Computer Applications. 1(22):55-8.
16. Chen S.M., Chang Y.C., & Pan J.S. 2013. Fuzzy rules interpolation for sparse fuzzy rule-based systems based on interval type-2 Gaussian fuzzy sets and genetic algorithms. IEEE transactions on fuzzy systems. 21(3):412-25.
17. Hsu C.H., & Juang C.F. 2011. Evolutionary Robot Wall-Following Control Using Type-2 Fuzzy Controller With Species-DE-Activated Continuous ACO. IEEE Trans. Fuzzy Systems. 21(1):100-12.
18. Tab F.A., Shahryari O.K. 2009. Fuzzy edge detection based on pixel's gradient and standard deviation values. Proc. Computer Science and Information Technology, IMCSIT'09, Pp. 7-10.
19. Alshennawy A.A., & Aly A.A. 2009. Edge detection in digital images using fuzzy logic technique. World Academy of science, engineering and technology. 24(51):1781-86.
20. Russo F. 1998. Edge detection in noisy images using fuzzy reasoning. Proc. Instrumentation and Measurement Technology Conference, 1998. IMTC/98. 1 Pp. 369-372.
21. Shen, W., Wang, X., Wang, Y., Bai, X. and Zhang, Z., 2015. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3982-3991).
22. G. Bertasius, J. Shi, and L. Torresani. Deepedge: A multiscale bifurcated deep network for top-down contour detection. In CVPR, 2015.
23. J.-J. Hwang and T.-L. Liu. Pixel-wise deep learning for contour detection. In ICLR, 2015.
24. Marmanis, D., Schindler, K., Wegner, J.D., Galliani, S., Datcu, M. and Stilla, U., 2018. Classification with an edge: Improving semantic image segmentation with boundary detection. ISPRS Journal of Photogrammetry and Remote Sensing, 135, pp.158-172.

25. Farbod, M., Akbarizadeh, G., Kosarian, A. and Rangzan, K., 2018. Optimized fuzzy cellular automata for synthetic aperture radar image edge detection. Journal of Electronic Imaging, 27(1), Pp.013030.
26. Setayesh, M., Zhang, M. and Johnston, M., 2013. A novel particle swarm optimisation approach to detecting continuous, thin and smooth edges in noisy images. Information Sciences, 246, pp.28-51.
27. Gonzalez, C.I., Castro, J.R., Melin, P. and Castillo, O., 2015, May. Cuckoo search algorithm for the optimization of type-2 fuzzy image edge detection systems. In 2015 IEEE Congress on Evolutionary Computation (CEC) (pp. 449-455). IEEE.
28. Zhang, W., Zhao, Y., Breckon, T.P. and Chen, L., 2017. Noise robust image edge detection based upon the automatic anisotropic Gaussian kernels. Pattern Recognition, 63, pp.193-205.
29. Jeong, C., Yang, H.S. and Moon, K., 2019. A novel approach for detecting the horizon using a convolutional neural network and multi-scale edge detection. Multidimensional Systems and Signal Processing, 30(3), pp.1187-1204.
30. Ren, H., Zhao, S. and Gruska, J., 2018. Edge detection based on single-pixel imaging. Optics express, 26(5), pp.5501-5511.
31. Cosgrove, C. and Yuille, A.L., 2019. Adversarial Examples for Edge Detection: They Exist, and They Transfer. arXiv preprint arXiv:1906.0033.
32. Raheja, S. & Kumar, A. Evolving Systems (2019). https://doi.org/10.1007/s12530-019-09304-6.
33. Xu L., Lu C., Xu Y., & Jia J. 2011. Image smoothing via L_0 gradient minimization. ACM Transactions on Graphics.30(6):174.
34. Ding X, Chen L, Zheng X, Huang Y, & Zeng D. 2016. Single image rain and snow removal via guided L_0 smoothing filter. Multimedia Tools and Applications. 75(5):2697-71.
35. https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images.html
36. http://sipi.usc.edu/database/database.php
37. Vincent, O. R.; and Folorunso, O. (2009). A descriptive algorithm for sobel image edge detection. In Proceedings of Informing Science & IT Education Conference, 40, 97-107.
38. Dollar, P.; Tu, Z.; and Belongie, S. (2006). Supervised learning of edges and object boundaries. In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, 2, 1964-1971.
39. Arbelaez, P.; Maire, M.; Fowlkes, C.; and Malik, J. (2011). Contour detection and hierarchical image segmentation. IEEE transactions on pattern analysis and machine intelligence, 33(5), 898-916.
40. Lim, J. J.; Zitnick, C. L.; and Dollár, P. (2013). Sketch tokens: A learned mid-level representation for contour and object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3158-3165
41. Dollár, P.; and Zitnick, C. L. (2013). Structured forests for fast edge detection. In Proceedings of the IEEE International Conference on Computer Vision, 1841-1848.
42. Xie, S. and Tu, Z., 2015. Holistically-nested edge detection. In Proceedings of the IEEE international conference on computer vision (pp. 1395-1403).
43. Gonzalez, C. I., Melin, P., Castro, J. R., Mendoza, O., & Castillo, O. (2016). An improved sobel edge detection method based on generalized type-2 fuzzy logic. Soft Computing, 20(2), 773-784.