Raheja, S and Kumar, A (2021) Edge detection based on type-1 fuzzy logic and guided smoothening. Evolving Systems, 12 (2). pp. 447-462. ISSN 1868-6486

Publisher: Springer Verlag

DOI: https://doi.org/10.1007/s12530-019-09304-6

Edge Detection Based on Type-1 Fuzzy Logic and Guided Smoothening

Abstract

Edge detection is an important phenomenon in computer vision. Edge detection is helpful in contour detection and thus helpful in obtaining the important information. Edge detection process heavily depends on chosen technique. Soft computing techniques are considered as powerful edge detection methods due to their adaptability. This paper presents a fuzzy logic based edge detection method where the quality of edges is controlled using sharpening guided filter and noise due to the sharpening is controlled using Gaussian filter. The accuracy of the method is judged using a variety of statistical measures. It has been found that by proper selecting the smoothening parameters a significant improvement in the detected edges can be obtained.

Keywords: Edge detection, Image enhancement, Fuzzy Logic, Guided Filtering

Introduction

In numerous computer vision systems, edge detectors are very important. The process of edge detection is valuable in the analysis of images by decreasing the processed data [1]. The technique of Edge detection is among prime methods applied in various field where processing of digital image is required. The purpose of using edge detection in the processing of an image is to get rid of useless data and perverse the important image characteristics. It includes certain methods like feature extraction, registration, interpretation and image segmentation which are very helpful in keeping the useful data only [2].

An edge could be defined as the object between boundary and image background [3, 4]. Soft Computing is a developing research area that comprises of integral components of machine learning techniques, for example, fuzzy logic, artificial neural networks, adaptive neuro-fuzzy inference system, Particle Swarm Optimization and Genetic algorithms etc... These strategies are generally utilized for edge detection and image segmentation in the greater part of the critical applications. Out of these methods, Fuzzy rule based classifiers have been used in various classes of engineering problems such as self localization and landmark recognition [5], data density estimation[6] , real time object detection [7] and human activity recognition [8] etc.

It is important to note that a universal definition for edges cannot be created as each human has its own way of judging the edges in images. Therefore, edge is a subjective phenomenon, and outcome depends on individual perspective. Therefore, in this paper, we have discussed image sharpening and Fuzzy Logic based edge detection and to control sharpening noise Gaussian kernel based filtering is proposed. The results are compared with notable statistical quality measures.

This paper is organized as follows, in section II of the paper related works is presented. Edge detection steps are detailed in section III of the paper. Proposed method is discussed in section

IV of the paper. In section V results of the paper are discussed, and finally in section VI of the paper major conclusions are discussed.

II. RELATED WORKS

In this part of the article, we have examined various edge detection strategies and concentrate on digital images' edge detection. Taking about the standard images, the edges are used to define the boundaries of the subject and helpful for division, enlistment, identification. It removes the irrelevant data and interestingly of protecting the essential properties of a picture. There are different conventional systems useful in edges detection mechanism. Some of the popular techniques are Laplacian Roberts, Sobel and gradient [4]. With the help of masks, edge detection is possible by linear operators. These masks are representing the ideal edge steps in different ways both for intensity and color. They utilized similar technique for identification. The comparison of various types of edge detection methods are detailed in Table 1.

Table 1: Comparison of Edge Detection Techniques

Edge-Detection Techniques	Methods
First–order derivative/ Gradient methods	Sobel, Roberts, Prewitt
Second–order derivative/ Zero crossing	Laplacian of Gaussian
Optimal image detection	Canny edge Detector
Soft Computing	ANN, PSO, SVM, GA, Ant colony Optimization

Roberts [9], Prewit and Sobel [10] introduced edge detection on the basis of the principle of gradient which demonstrate the impact of these filters on the images which is based on the first derivative. A progressively successful operator is the Laplacian, which utilizes the second derivative in locating the edge [10].

Alshennawy et al [11], Aborisade et al [12] and Begol et al [13] proposed Soft Computing based methodologies, for example, fuzzy logic. Hamed Mehrara et al proposed another methodology that works on the concept of Back Propagation Neural Network [14]. Lei Zhang et al introduced a hybrid technique works on the principle of Adaptive neuro- fuzzy inference system (ANFIS) for the detection of edge [15].

Table 2: Soft Computing based notable Techniques

Authors	Techniques
Borji and Hamidi	Fuzzy Logic
Alshennawy	Fuzzy Logic
Zhang et al	Adaptive Neuro-Fuzzy
Mehrara et al	Neural Network
Aborisade	Fuzzy Logic
Begol	Fuzzy Logic

Mathur et al. introduced a latest algorithm based on fuzzy relative pixel value [16]. In this methodology, the relative pixel values are examined and subsequently reduce the image

processing by the application of Artificial intelligence. Table 2 demonstrates edge detection methods work on the concepts of Fuzzy Logic, Neural Network and ANFIS.
The other recent notable mechanisms in edge detection are: deep convolutional neural network [17], Fuzzy cellular automata [18], particle swarm optimization [19], cuckoo search optimization [20], Anisotropic Gaussian Kernels [21], convolution neural networks (CNNs) [22, 23]. Edge detection based on single pixel imaging was proposed in [24]. 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 [25].
This paper, proposes a technique based on type 1 fuzzy logic and guided image filtering is used for the sharpening of the images. Using sharpening, strength of the edges can be enhanced to detect them easily. Even in case of perturbation sharpening can be used to control edges. The proposed method is effective as it considers the advantages of both fuzzy logic design and guided image smoothening.

III. EDGE DETECTION STEPS

The main aim of edge detection is to successfully locate and identify genuine edges. Edge detection is difficult in low resolution which is mostly found in colour images. In edge detection the main steps are Filtering, Enhancement, Detection and localization which are utilized for detection of correct edges (Figure 1).

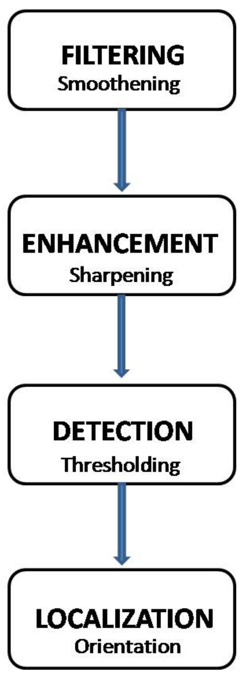


Figure 1: Edge Detection Notable Steps

We cannot use images straightway in light of the fact that it might be corrupted by arbitrary varieties in intensities, varieties in brightening, or uneven contrast. This irregular variety in intensity is known as noise. Image sharpening is needed to enhance the quality of image. In enhancing the quality of image, high pass filtering plays an important role. The sharpened image

(I_S) signal is proportional to the high pass filtering of original signal and mathematically written as

$$I_S = I + \gamma[I - \underbrace{I \otimes H}_{\text{Smoothening}}] \tag{1}$$

where,

I: original image, H: High pass filter and 'γ' is scaling factor.

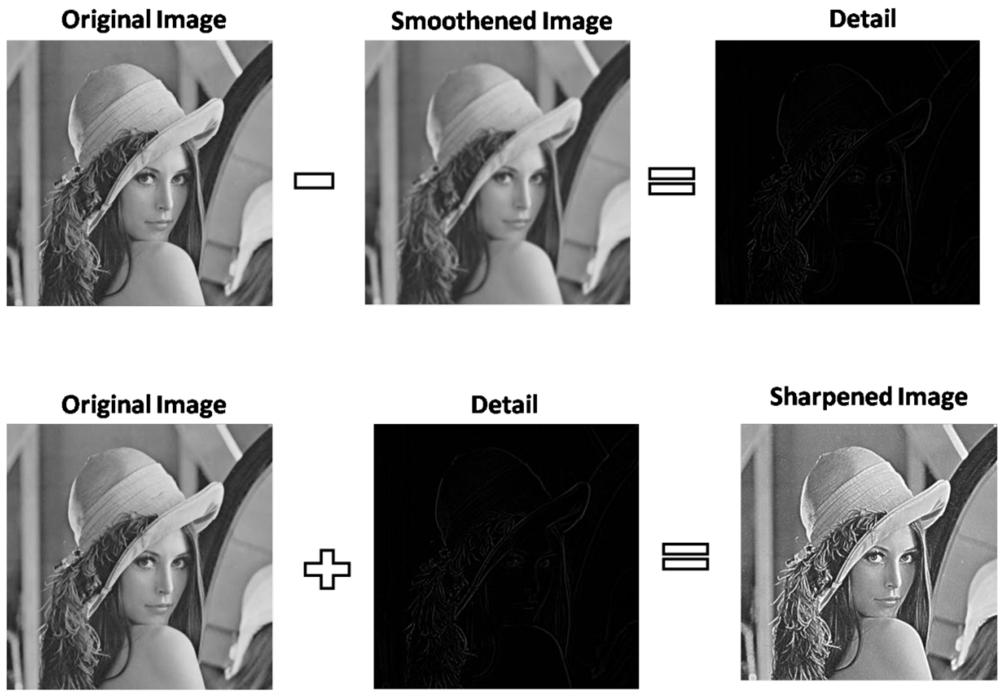


Figure 2: Demonstration of image sharpening (Lena)

Canny Edge Detection

Canny edge detection is a well known method in edge detection; it is a multi-step algorithm and can detect edges in noisy environment by suppressing noise [26]. The main steps of the algorithms are as under

1. Use Gaussian filter to reduce noise and other redundant information.

   $f(i,j) = H(i,j) \otimes I(i,j)$

   Where, $H(i,j) = \dfrac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\dfrac{i^2 + j^2}{2\sigma^2}\right)$ (2)

2. Compute gradient of $f(i,j)$ using any of the gradient operators (Roberts, Sobel, Prewitt, etc) to get:

   $M(i,j) = \sqrt{f_i^2(i,j) + f_j^2(i,j)}$ and $\theta = \tan^{-1} \dfrac{f_j\ (i,j)}{f_i\ (i,j)}$

   (3)

3. Threshold M:

$$M_T(i,j) = \begin{cases} M(i,j) & \text{if } M(i,j) > \gamma \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Where, γ is threshold and is chosen carefully such that that all edges are preserved while major portion of the noise is suppressed.

4. Use non-maximal suppression in the pixels edges in M_T obtained above to thin the edge ridges...
5. Using two different thresholds τ_1 and τ_2 (where $\tau_1 < \tau_2$) obtain two binary images I_1 and I_2. Note that I_2 with greater τ_2 has less noise and fewer false edges but greater gaps between edge segments, when compared to I_1 with smaller τ_1.
6. Link edge segments in I_2 to form continuous edges.

Figure 3: Edge Detected image using Canny method (Lena)

The canny edge detection is considered as one of the better methods in edge detection. However, without post processing of detected edges performance of canny edge detection is not good, and suffers from large number of falsely accepted edges.

IV PROPOSED METHOD

In this section proposed method is detailed. The various steps are detailed and utility of various processes are discussed.

Fuzzy Expert System

Figure 4 depicts the basic architecture layout of a fuzzy expert system. 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 sharpening filter and after this fuzzy logic is used to detect edges. In method M_3 the input image is first passes through the sharpening filter and then for Gaussian filter and finally 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 to obtain crisp values to obtain desired results. The detailed description of the fuzzy based system, image sharpening using guided filter and finally use of Gaussian kernel for noise removal is discussed.

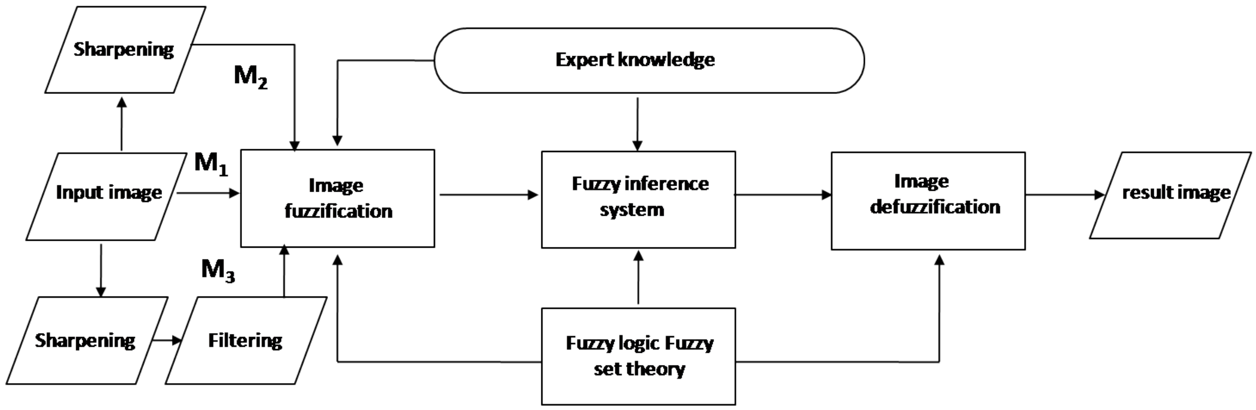


Figure 4: Schematic of proposed edge detection mechanism (M_1: Edge detection using fuzzy logic only, M_2: Edge detection using fuzzy logic and sharpening filter and M_3: Edge detection using fuzzy logic and sharpening filter)

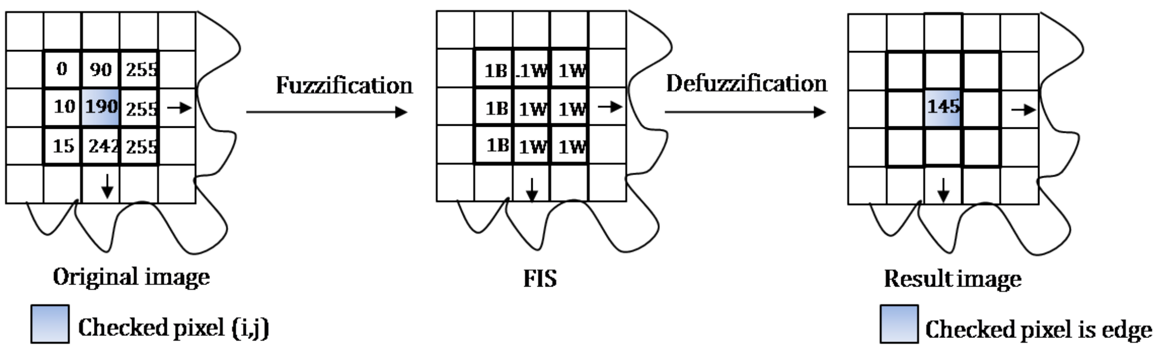


Figure 5: Schematic view of fuzzy logic based edge detection

In this work, we use triangular membership function for both input and output and defined as:

$$triangle(x;a,b,c) = \max\left(\min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right)$$

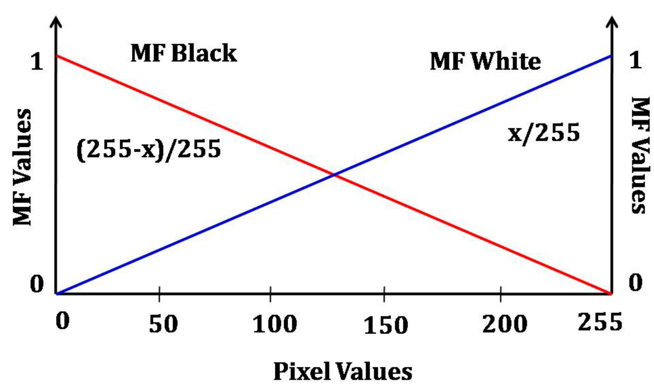


Figure 6: Membership function for black and white pixels values

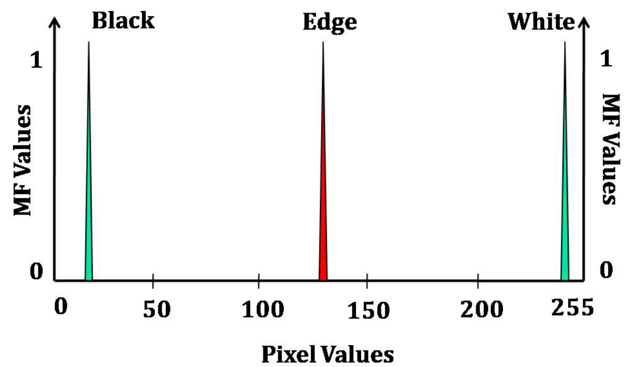
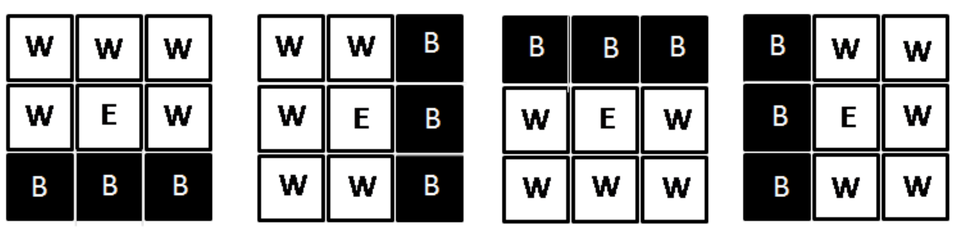


Figure 7: Output membership function for black, white and edge

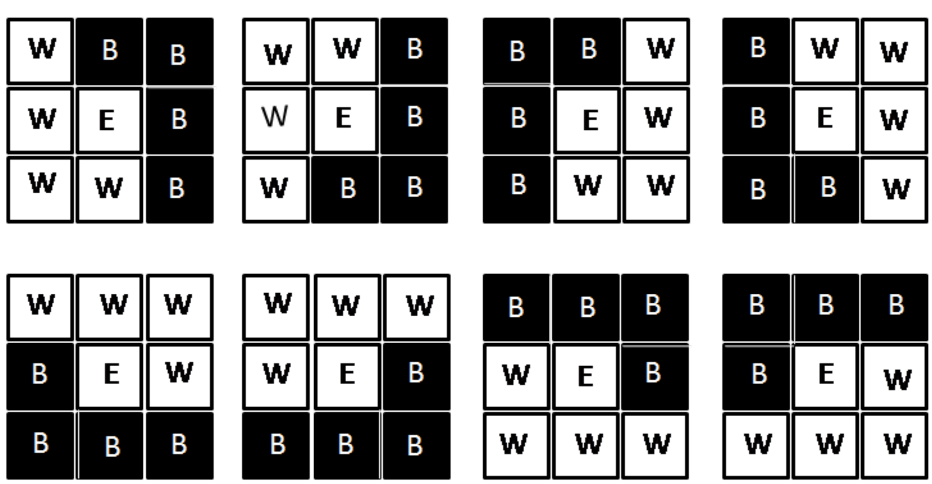
Fuzzy Rules

Fuzzy rules are developed using human expert system, and edge and non-edge pixels are decided on the basis of neighbourhood pixels. For the edge detection a total of 30 rules are defined. The rules are developed on a 3×3 mask as shown in Figure 1. In the rule designing Img (x, y) represents pixel position, '1' represents white pixel and '0' 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



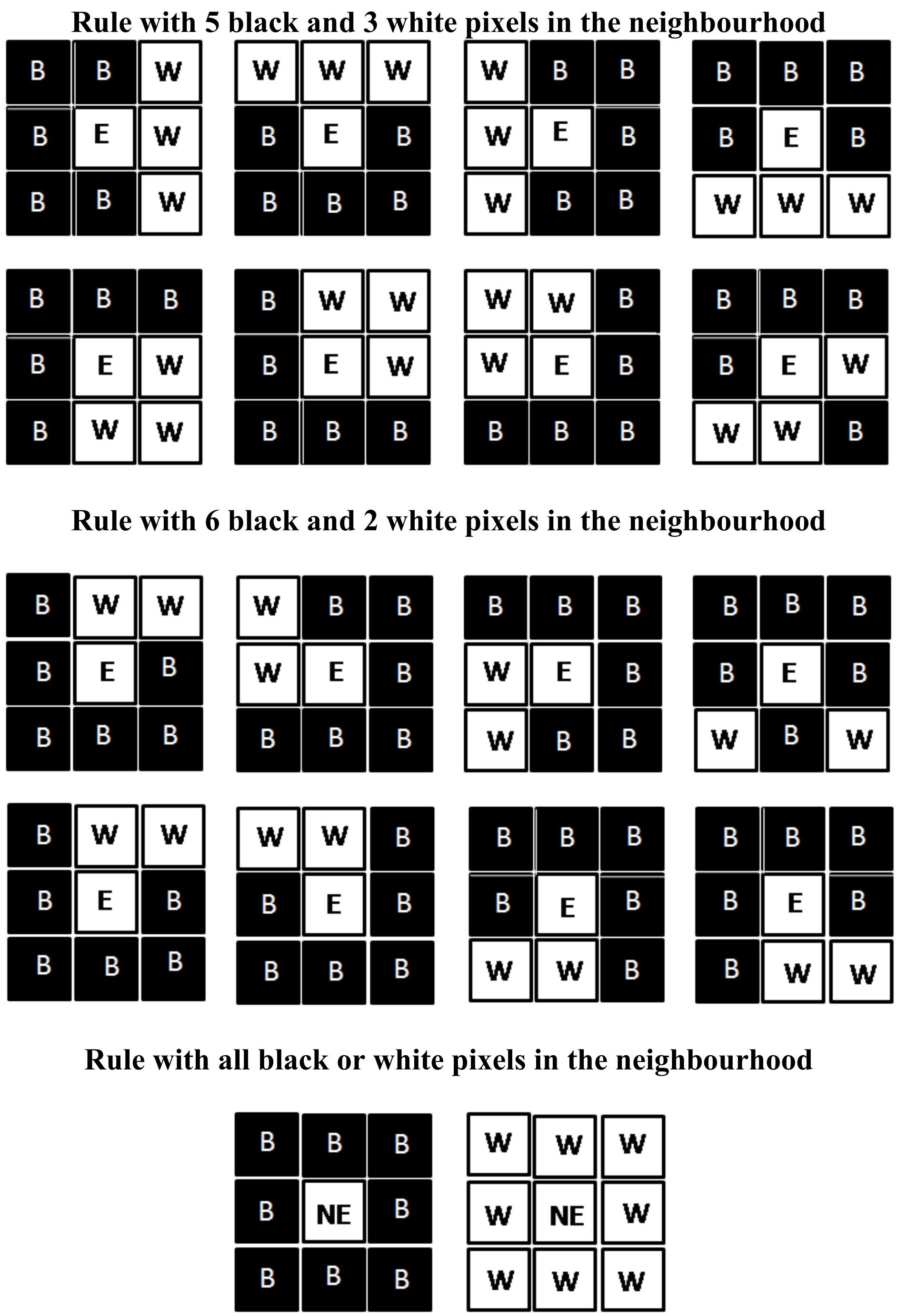


Figure 8: Fuzzy rules black (B), white (W) and edge (E) and non-edge (NE)

Guided Image Filtering and Sharpening

In guided image filtering, the output image is linear transformation of guided image 'G' and is expressed as [27]

$$\hat{I}_n = a_i G_n + b_i, \qquad \forall n \in w_i . \tag{5}$$

a_i and b_i are the co-efficient in window w_i. The guided image filtering problem can be formulated as the minimization of difference in input and output images, and ε is smoothness parameter and decides the degree of smoothness.

$$E\left(a_i, b_i\right) = \sum_{n \in w_i} \left(\left(a_i G_n + b_i - I_n \right)^2 + \varepsilon a_i^2 \right). \tag{6}$$

The co-efficient are evaluated as

$$a_i = \frac{\left(1/|w|\right) \sum_{p \in w_i} G_n I_n - \bar{G}_i \bar{I}_i}{\sigma_i^2 + \varepsilon} \tag{7}$$

$$b_i = \bar{I}_i - a_i \bar{G}_i$$

where, bar represents the mean values of corresponding parameter. $|w|$ denotes the total number of pixel w_i.

$$\hat{I}_n = \left(\frac{1}{|w|} \sum_{i \in w_n} a_i \right) G_n + \left(\frac{1}{|w|} \sum_{i \in w_n} b_i \right). \tag{8}$$

In [28] it is proved that GF can also be expressed as

$$W_{nm}^{GF}\left(G\right) = \frac{1}{|w|^2} \sum_{i:(n,m) \in w_i} \left(1 + \frac{\left(G_n - \bar{G}_i\right)\left(G_m - \bar{G}_i\right)}{\sigma_i^2 + \varepsilon} \right) \tag{9}$$

In general we have

$$\hat{I}_n = \sum_{m \in w_n} W_{nm}^{GF}\left(G\right) I_m . \tag{10}$$

The enhanced image can be written as

$$\hat{I}_s = \gamma\left(\hat{I}_n - G_n\right) + G_n \tag{11}$$

$$\hat{I}_s = \gamma[a_i G_n + b_i - G_n] + G_n \tag{12}$$

$$\hat{I}_s = [\gamma(a_i - 1) + 1]G_n + kb_i \tag{13}$$

The Gaussian 5×5 kernel is applied as filter for the removal of noise, with kernel as

$$\frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}. \tag{14}$$

The Gaussian kernel is used in method M_3 only.

V. RESULTS

This section presents the edge detection results under the considered method. The simulation is performed in MATLAB software. For performance evaluation various edge detection measures are considered as described below:

Edge Detection Performance Measures

It is not easy to define the general-purpose evaluation for edge detection, as some edges are missed, some are falsely accepted, and some alters their positions. Before going into detail of edge detection method, let us consider basic properties and concept of edge detection. Throughout the paper we have defined, ground truth image as (I_{gt}) and edge detected image as (I_{ed}). Let the quality measure is denoted by Q and property as (P) then following properties must be satisfied:

Symmetry (P_1) : $Q(I_{gt}, I_{ed}) = Q(I_{ed}, I_{gt})$

Ideal Solution (P_2) : $I_{gt} = I_{ed}$

Sensitivity to noise (P_3) : if any pixel (p) does not belong to ground truth (I_{gt}) or edge detected image (I_{ed}) i.e., $p \notin (I_{gt} \cup I_{ed})$, then

$Q(I_{gt}, I_{ed}) < Q(I_{ed} \cup \{p\}, I_{gt})$

Sensitivity to improvement (P_4): if $p \in I_{gt}$ and $p \notin I_{ed}$, then

$Q(I_{gt}, I_{ed}) < Q(I_{ed} \cup \{p\}, I_{gt})$

First property is self explanatory, second property state that there is only one optimal solution. Third and fourth properties indicate that inclusion of correct and in-correct pixels will decreases or increase error. However, these four properties are not good enough to describe edge detection as to include falsely excluded important edge sometime extra pixels are accepted. As considering $p \notin I_{gt}$ and to include p in I_{ed} will increase un-avoidable error. In edge detection the main requirements are the correct identification of edge pixels and also the correct location of edge pixels.

We classify the positives pixels as edges when the classification is binary. We can classify pixels edge into four unique classifications with the condition that ground truth is present. These four classes are: True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). Due to the fact that not a large portion of the pixels are edges, there will be an issue of imbalanced binary classification [29], where the dominating class will be the negative class.

The issues related with the position of pixel can be solved with the help of the spatial tolerance during the matching of the edge pixels as the factor, which decides whether the pixel classification is accurate or not, is the least alteration in the position of the pixel. Normally, we cannot assume an edge pixel as fully accurate or not, relying upon the way that it is found t or t+1 pixel from the true edge. According to the Liuand Haralick, a quality measure is exact in the event of small changes in value reflects small variations in the detect [30]. The measures such as F-score, and χ^2 test, are precise when pixels positions are exact however fall flat when pixels position changes. These issues prompted penalizing an edge pixel relying upon its separation to a true edge, which encourages the concept of distance based EMs.

Correlation Coefficient

The correlation coefficient is obtained by means of the equation [31]

$$|\rho| = \frac{\mathrm{cov}[I_{gt}, I_{ed}]}{\sqrt{Var[I_{gt}]Var[I_{ed}]}}. \tag{15}$$

Under perfect matching $|\rho|$ is one theoretically lowest value is zero. This metric satisfies this satisfies properties P_1 and P_2.

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 [32]

$$FoM = \frac{1}{\max(I_{gt}, I_{ed})} \sum_{i=1}^{I_{ed}} \frac{1}{1 + \mu d^2(p, I_{gt})} \tag{16}$$

Here,

$I_{gt} =$ ideal edge points (ground truth)

$I_{ed} =$ edge points detected

d = displacement of detected edges from ideal edges

$\mu =$ 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 [33]

$$SSIM(I_{gt}, I_{ed}) = \frac{\left(2\mu_{gt}\mu_{ed} + k_1\right)(2\sigma_{gted} + k_2)}{\left(\mu_{gt}^2 + \mu_{ed}^2 + k_1\right)\left(\sigma_{gt}^2 + \sigma_{gt}^2 + k_2\right)} \tag{17}$$

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 I_{gt}={a_1, ...,a_n} and I_{ed}={b_1,...,b_n}, the Hausdoff distance calculated as [34]:

$$H(I_{gt}, I_{ed}) = \max(d(I_{gt}, I_{ed}), d(I_{ed}, I_{gt}))$$

Where $d(I_{gt}, I_{ed}) = \max_{a \in I_{gt}} \min_{b \in I_{ed}} \| a - b \|$ (18)

The function $d(I_{gt}, I_{ed})$ is the directed Hausdorff distance from I_{gt} to I_{ed}. This method is based on distance among the points, and lesser distance means more closeness between the images.

Euclidiean Distance (E_D)

The Euclidian distance between two images is evaluated as [35]

$$E_D(I_{gt}, I_{ed}) = \frac{1}{ij}\sum_{m=0}^{i-1}\sum_{n=0}^{j-1}\left|I_{gt}(m,n) - I_{ed}(m,n)\right|^2 \tag{19}$$

Average point-to-set distances (D_K)

The average distance from the edge pixels in the image under consideration to those in the ground truth is obtained as [36]

$$D_K = \frac{1}{I_{ed}}\sqrt[K]{\sum_{p\in I_{ed}} d^K(p, I_{gt})} \tag{20}$$

We have considered only one image Lena along-with the ground truth image which contain ideal edge to cover vast varieties of results.

Baddeley's Delta metric (BDM)

Baddeley's Delta metric (BDM) is modified form of the Hausdorff distance [34]. It is based on the distance between the elements of each of the sets, and expressed as [37]

$$\mathrm{B}_f^K = \left[\frac{1}{|P|}\sum_{p\in P}\left|fd(p, I_{gt}) - fd(p, I_{ed})\right|^K\right]^{1/K}, \tag{21}$$

f is a function which is concave in nature, and modulates the point to point distance.

In figure 9(a) original Lena image is shown, while varying patch radius from r= 16,32,64 and 128, while keeping ε to a fixed level of 0.01 and γ equals 5 the obtained results are shown in figure 9(b) to 9(e). the size of local patch is considered to be $(2r+1)\times(2r+1)$ For smaller radius, block size is smaller, and sharpening takes place in local patches, therefore edges becomes sharper but discontinuity points increases thus sharpness quality is not good, as we increase the radius, the sharpness in image is relatively more uniform, and for r=64 and 128 the sharpness in images is of good quality.

(a)

Figure 9: Image Sharpening using Guided Filtering (*k* constant)

- **(a)** Original Image
- **(b)** r=16,ε=0.01, γ=5
- **(c)** r=32,ε=0.01, γ =5
- **(d)** r=64,ε=0.01, γ =5
- **(e)** r=128,ε=0.01, k=5
- **(f)** r=16,ε=0.001, γ =5
- **(g)** r=32,ε=0.001, γ =5
- **(h)** r=64,ε=0.001, γ =5
- **(i)** r=128,ε=0.001, γ =5

The above experiment is repeated again with ε=0.001, while keeping other parameters fixed, the effect of regularization parameter can be observed the mean variance of Lena image under consideration is 0.0256 therefore the effect of regularization parameter (ε) is negligible and image quality does not alter significantly.

In figure 10, image enhancement for various values of γ is shown, here as γ increases the sharpness increases, but for larger values of γ the quality of the images changes considerably as compared to original image. Moreover a significant variation in the colours is also observed. It can be observed (marked region in original image) from the figures that the first with the rise in γ edges becomes sharper thereafter they start to diminish, therefore it can also be concluded that both ε and γ have significant effect on the overall quality of the sharpened images.

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

(e) **(f)** **(g)** **(h)**

Figure 10: Image Sharpening using Guided Filtering (γ varying)

(a) Original Lena Image
(b) r=64,ε=0.01, γ =5
(c) r=64,ε=0.01, γ =10
(d) r=64,ε=0.01, γ =20
(e) r=64,ε=0.01, γ =30
(f) r=64,ε=0.01, γ =50
(g) r=64,ε=0.01, γ =75
(h) r=64,ε=0.01, γ =100

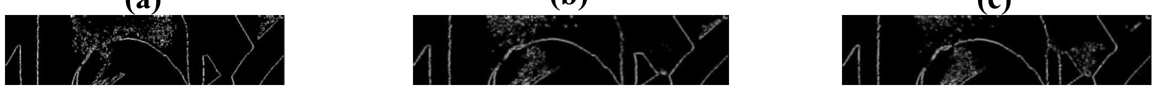
In figure 11, results are presented for methods M_1 and M_2. In Figure 11(a) edge detected image using Fuzzy logic is shown for Lena image. In figure 11(b) to 11(h) results are shown for various values of γ as in figure 10. Consider the mark region in Figure 10 (a), it is clear from 11 (a) that Fuzzy logic method in alone fail to capture edges in this region. Similarly around hat area noise is also added. As we increases the value of γ from 5 to 100, the noise around hat area diminishes but just above the mark area noise start to increases. These changes are shown in figure 12.

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

(e) (f) (g) (h)

Figure 11: Edge detected images (a) Fuzzy logic (b) M_2, γ=5 (c) M_2, γ=10 (d) M_2, γ=20 (e) M_2, γ=30 (f) M_2, γ=50 (g) M_2, γ=75 (h) M_2, γ=100

(a) (b) (c)



(d) (e) (f)

Figure 12: Image segment with visible variations

In figure 13, results are presented for methods M_3. It is clear from the figure that using filtering noise is suppressed significantly, and detected edges are much better as compared to shown in figure 11.

The discussion made above is based on human visualization. However, shifting in pixel positions cannot be identified using human vision system. Moreover, the regularization term also lead to the shifting of pixels. Therefore, distance based parameters for quality measurements provide the correctness of the chosen methods... In our results we have not done any further processing on detected image to clearly visualize the effect of each method.

Table 3: Comparison of parameters for methods M_1 and M_2

Methods	Performance Metrics						
	FoM	**SSIM**	**HoD (Avg./Max)**	E_D	**BDM**	D_K	**ρ**
Canny	0.297	0.451	6.69/9.21	**0.0048**	10.74	**0.06**	0.225
Fuzzy	0.398	0.519	5.31/8.12	0.0059	12.24	0.08	0.146
γ=5	**0.483**	0.559	4.67/7	0.0056	**9.40**	0.111	0.21
γ=10	0.472	0.557	4.59/6.78	0.0056	9.48	0.113	0.218
γ=20	0.469	**0.560**	4.47/6.86	0.0055	10.03	0.11	0.229
γ=30	0.467	0.559	4.45/7.42	0.0055	10.02	0.112	**0.232**
γ=50	0.471	0.556	**4.44/7**	0.0055	9.98	0.102	**0.232**
γ=75	0.467	0.557	4.46/7.42	0.0055	9.85	0.102	0.231
γ=100	0.469	0.555	4.47/7.21	0.0055	9.93	0.106	0.231

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

Figure 13: Edge detected images (a) M_3, γ=5 (b) M_3, γ=10 (c) M_3, γ=50 (d) M_3, γ=100

In table 3, listing of parameters results are shown, FoM is least for canny detection with value of 0.297 and it is best for γ=5, equals 0.483, SSIM for Canny is 0.451 and for γ=20, equals 0.560, while HoD is best for γ=50, and equals to 4.44/7, the Euclidian distance is minimum for canny method, which is an expected results as large number of edges are detected by canny method. BDM is least for γ=5, while it is at maximum for Fuzzy method. D_K is best for Canny method. Finally the co-relation co-efficient is best for γ=30, 50. Therefore it can be inferred from the table that the performance of the Fuzzy method in alone is poorest and among the chosen parameters for γ=5, the obtained results are better than other considered methods. In table 4, results are compared for methods M_1 and M_3. Obtained results have shown similar trend but obtain results are much better in comparison to the results in Table 3. In Table 4, FoM is maximum for γ=20, SSIM is maximum for γ=30, 50. HoD is maximum for γ=50. Still these performance measures are indicative, and do not provide clear information about exact edge detection mechanism. Therefore, in most of the recent research [13-19] human visual system (HVS) is used to characterize edge and non-edge pixels.

Table 4: Comparison of parameters for methods M_1 and M_3

Methods	Performance Metrics						
	FoM	**SSIM**	**HoD (Avg/Max)**	E_D	**BDM**	D_K	**ρ**
Canny	0.297	0.451	6.69/9.21	**0.0048**	10.74	**0.06**	**0.225**
Fuzzy	0.398	0.519	5.31/8.12	0.0059	12.24	0.08	0.146
γ=5	0.470	0.581	4.15/10.44	0.0056	9.92	0.120	0.186
γ=10	0.475	0.582	4.17/9.64	0.0056	9.69	0.110	0.198
γ=20	**0.497**	0.590	4.13/8.43	0.0055	9.05	0.093	0.211
γ=30	0.496	**0.595**	4.09/7.09	0.0055	8.81	0.088	0.217
γ=50	0.495	**0.595**	**4..09/6.86**	0.0057	8.59	0.079	0.216
γ=75	0.489	0.593	4.11/6.63	0.0057	8.65	0.087	0.217
γ=100	0.493	0.593	4.12/6.56	0.0057	**8.57**	0.081	0.219

The above results are obtained on Lena image, detected edges are compared with ground truth image and results are compared in terms of various performance measures. However, to prove usefulness of proposed method results are obtained other database images. Image shows in figure 14, is taken from Berkley Segmentation Database [38], while in image in figure 15, and is from USC-SIPI Image Database [39]. Considering, figure 14, fuzzy logic based method fail to detect edge on west-south corner of the image. However, from (b) to (e) it is clear that as we increase sharpness the edges can be detected more correctly. From figure 15, it can be visualized that as sharpness increases; image detail is more clearly visible. Same effect is also obtained in detected edges.

Original Image	**Sharpened Image** γ=5	**Sharpened Image** γ=10	**Sharpened Image** γ=15	**Sharpened Image** γ=20

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

Figure 14: BSD edge detected images (a) Fuzzy logic (b) M_3, γ=5 (c) M_3, γ=10 (d) M_3, γ=15 (e) M_3, γ=20

Original Image	**Sharpened Image** γ=5	**Sharpened Image** γ=10	**Sharpened Image** γ=15	**Sharpened Image** γ=20

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

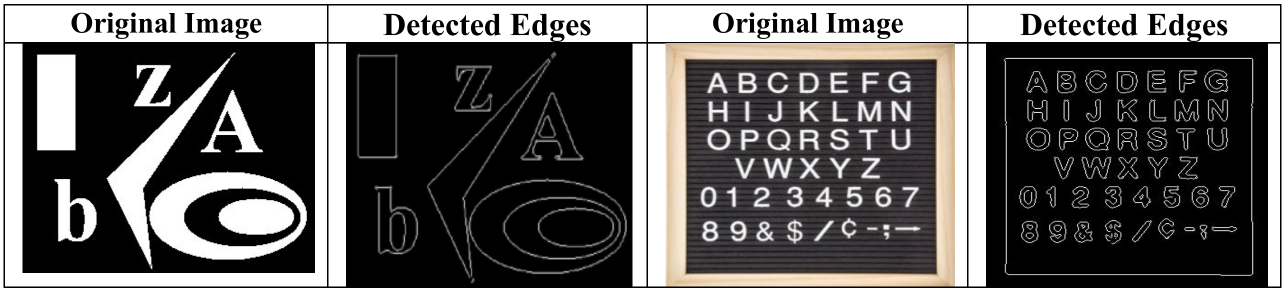
Figure 15: USC-SIPI edge detected images (a) Fuzzy logic (b) M_3, γ=5 (c) M_3, γ=10 (d) M_3, γ=15 (e) M_3, γ=20

Original Image	Sharpened Image γ=5	Sharpened Image γ=10	Sharpened Image γ=15	Sharpened Image γ=20

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

Figure 16: Tulip edge detected images (a) Fuzzy logic (b) M_3, γ=5 (c) M_3, γ=10 (d) M_3, γ=15 (e) M_3, γ=20

In figure 16, results are obtained for tulip images and obtained response show similar trend as in figure 14 and 15. It is also clear from figures 15 and 16, sometime more sharpness leads to the generation of noise; therefore image should be sharpened to a level where effect of noise is minimal. However, if this noise is dominant, than this additional noise can be suppressed using filters as used in other edge detection methods [1].

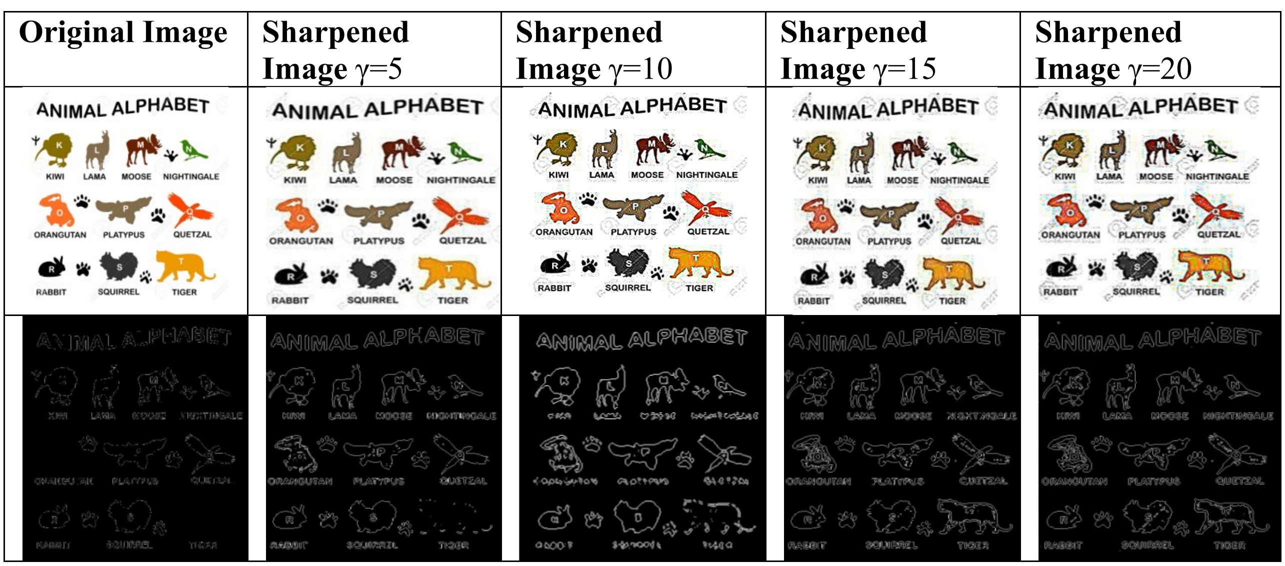


(a) (b) (c) (d)

Figure 17: Edge Detection using Fuzzy Logic (a) Original Image (b) Detected Edges (c) Original Image (d) Detected Edges

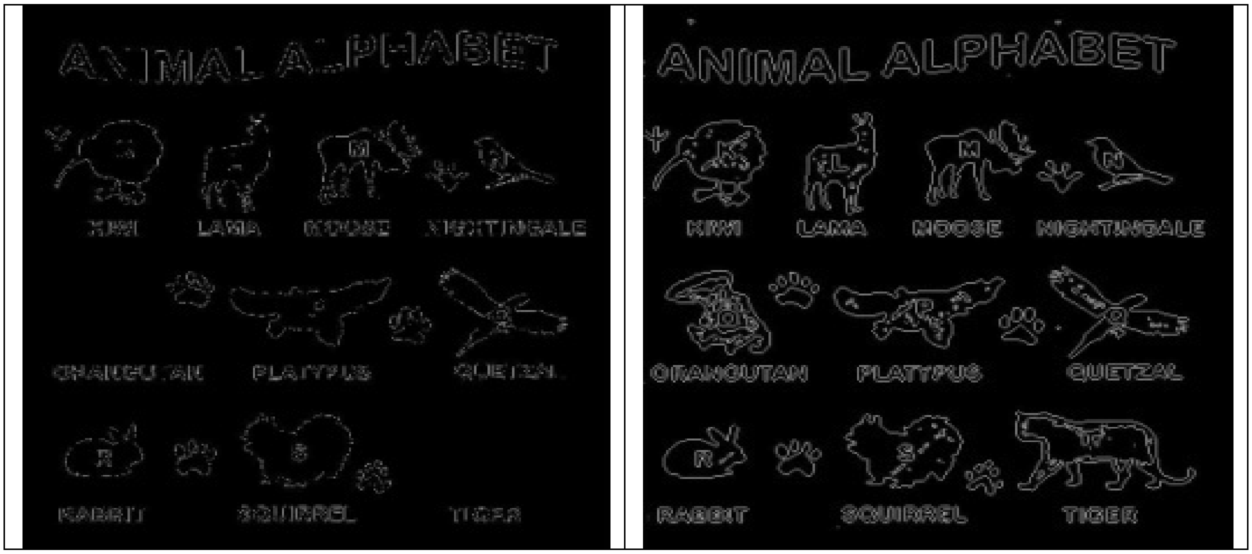
In the above discussed results it is found that Fuzzy Logic based method fails to detect some of the edges. Therefore, to check the validity of the considered fuzzy logic structure, in figure 17 results are generated for two images ((a) and (c)) where, edges are clearly visible, and it is found that considered fuzzy logic structure correctly detect all the edges in both the images. Thus, in case of clear edges images Fuzzy logic structure gives accurate results.

In figure 18, animal alphabet image is taken which has more complex edges as compared to images considered in figure 17. Fuzzy logic based method fails to detect 'Tiger' and 'Orangutan' shapes and also letter mark on the animals are not detected. As we increase sharpness edges and letters are detected more clearly. In figure 19, zoomed version of fuzzy logic and sharpened image γ=20 is shown, the name of animal is not clearly detected with Fuzzy design, while with proposed method edges of both animal name and letter mark on the animals are detected. With our design claws and paws edges are clearly visible.



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

Figure 18: Animal alphabet image (a) Fuzzy logic (b) M_3, γ=5 (c) M_3, γ=10 (d) M_3, γ=15 (e) M_3, γ=20



(a) **(b)**

Figure 19: Zoomed version of (a) figure 18(a) (b) figure 18(e)

Finally, we can conclude that the proposed method is superior to fuzzy logic based method, and it overcomes the limitations of fuzzy logic structure. The membership function considered in this work is same as in previous works [7-12]. However, to overcome the limitations of fuzzy logic method, further research can be carried out in developing fuzzy rules which are developed using human expert system and in choosing appropriate membership function.

VI. CONCLUSIONS

This paper presents an edge detection method based on fuzzy logic, sharpening and filtering. The main aim of the paper is to design an edge detection method where edges can be controlled without deteriorating the considered image. It has been found that Fuzzy logic method in alone falsely rejected some of the edges, noise is also added. To combat this we have shown that sharpening of image can be done, which improves the results significantly. It is also shown that sharpening itself depends on parameters r, ε and γ, and these parameters should be chosen efficiently to get desired results. It is also notable that the regularization parameter (ε) should be kept within the sub-range of image variance as regularization parameter shift the pixel positions. It is also shown that noise generated due to fuzzy process can be significantly brought down by using Gaussian filter. The obtained results are compared with various statistical measures and it has been found that proposed methods M_2 and M_3 performs better in comparison to method M_1.

REFERENCES

1. Shah, M., 1997. Fundamentals of computer vision. *University of Central Florida*.
2. Goswami, J.C. and Chan, A.K., 2011. *Fundamentals of wavelets: theory, algorithms, and applications* (Vol. 233). John Wiley & Sons.

3. Gevers, T., Gijsenij, A., Van de Weijer, J. and Geusebroek, J.M., 2012. *Color in computer vision: fundamentals and applications* (Vol. 23). John Wiley & Sons.
4. Shih, F.Y., 2010. *Image processing and pattern recognition: fundamentals and techniques*. John Wiley & Sons.
5. Zhou, X. and Angelov, P., 2007, April. Autonomous visual self-localization in completely unknown environment using evolving fuzzy rule-based classifier. In *2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications* (pp. 131-138). IEEE.
6. Angelov, P. and Zhou, X., 2008, June. On line learning fuzzy rule-based system structure from data streams. In *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)* (pp. 915-922). IEEE
7. Angelov, P., Sadeghi-Tehran, P. and Ramezani, R., 2011. An approach to automatic real-time novelty detection, object identification, and tracking in video streams based on recursive density estimation and evolving Takagi–Sugeno fuzzy systems. *International Journal of Intelligent Systems*, *26*(3), pp.189-205.
8. Sargano, A., Angelov, P. and Habib, Z., 2017. A comprehensive review on handcrafted and learning-based action representation approaches for human activity recognition. *applied sciences*, *7*(1), p.110.
9. Rogowska, J., 2000. Overview and fundamentals of medical image segmentation. *Handbook of medical imaging, processing and analysis*, pp.69-85.
10. Rangarajan, S., 2005. Algorithms for edge detection. *Stony Brook University. Web document link: www. ee. sunysb. edu/~ cvl/ese558/s2005/Reports/Srikanth%20Rangarajan/submission. doc*.
11. Alshennawy, A.A. and Aly, A.A., 2009. Edge detection in digital images using fuzzy logic technique. *World Academy of science, engineering and technology*, *51*, pp.178-186.
12. Aborisade, D.O., 2011. Novel fuzzy logic based edge detection technique. *International Journal of Advanced Science and Technology*, *29*(1), pp.75-82.
13. Begol, M. and Maghooli, K., 2011. Improving digital image edge detection by fuzzy systems. *World Academy of Science, Engineering and Technology*, *81*, pp.76-79.
14. Mehrara, H., Zahedinejad, M. and Pourmohammad, A., 2009, December. Novel edge detection using BP neural network based on threshold binarization. In *Computer and Electrical Engineering, 2009. ICCEE'09. Second International Conference on* (Vol. 2, pp. 408-412). IEEE.
15. Zhang, L., Xiao, M., Ma, J. and Song, H., 2009, October. Edge detection by adaptive neuro-fuzzy inference system. In *Image and Signal Processing, 2009. CISP'09. 2nd International Congress on* (pp. 1-4). IEEE.
16. Mathur, S. and Ahlawat, A., 2008. Application of fuzzy logic on image edge detection.
17. 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.
18. 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), p.013030.
19. 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.

20. Gonzalez, C.I., Melin, P., Castro, J.R., Castillo, O. and Mendoza, O., 2016. Optimization of interval type-2 fuzzy systems for image edge detection. *Applied Soft Computing*, *47*, pp.631-643.
21. 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.
22. 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.
23. Liu, Y. and Lew, M.S., 2016. Learning relaxed deep supervision for better edge detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 231-240).
24. Ren, H., Zhao, S. and Gruska, J., 2018. Edge detection based on single-pixel imaging. *Optics express*, *26*(5), pp.5501-5511.
25. Cosgrove, C. and Yuille, A.L., 2019. Adversarial Examples for Edge Detection: They Exist, and They Transfer. *arXiv preprint arXiv:1906.00335*.
26. Canny, J., 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), pp.679-698.
27. He, K., Sun, J. and Tang, X., 2013. Guided image filtering. *IEEE transactions on pattern analysis & machine intelligence*, (6), pp.1397-1409.
28. He, K., Sun, J. and Tang, X., 2010, September. Guided image filtering. In *European conference on computer vision* (pp. 1-14). Springer, Berlin, Heidelberg.
29. N.V. Chawla, N. Japkowicz, P. Drive, Editorial: special issue on learning from imbalanced data sets, ACM SIGKDD Explorations Newsletter 6 (1) (2004) 1–6.
30. G. Liu, R.M. Haralick, Optimal matching problem in detection and recognition performance evaluation, Pattern Recognition 35 (10) (2002) 2125–2139.
31. Van Heel, M., 1987. Similarity measures between images. *Ultramicroscopy*, *21*(1), pp.95-100.
32. Wesolkowski, S., Jernigan, M.E. and Dony, R.D., 2000, September. Comparison of color image edge detectors in multiple color spaces. In *Image Processing, 2000. Proceedings. 2000 International Conference on* (Vol. 2, pp. 796-799). IEEE.
33. Hore, A. and Ziou, D., 2010, August. Image quality metrics: PSNR vs. SSIM. In *Pattern recognition (icpr), 2010 20th international conference on* (pp. 2366-2369). IEEE.
34. Ma, X. and Grimson, W.E.L., 2005, October. Edge-based rich representation for vehicle classification. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (Vol. 2, pp. 1185-1192). IEEE.
35. Nadernejad, E., Sharifzadeh, S. and Hassanpour, H., 2008. Edge detection techniques: evaluations and comparisons. *Applied Mathematical Sciences*, *2*(31), pp.1507-1520.
36. T. Peli, D. Malah, A study of edge detection algorithms, Computer Graphics and Image Processing 20 (1) (1982) 1–21.
37. Lopez-Molina, C., Ayala-Martini, D., Lopez-Maestresalas, A. and Bustince, H., 2017. Baddeley's delta metric for local contrast computation in hyperspectral imagery. *Progress in Artificial Intelligence*, *6*(2), pp.121-132.
38. https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images.html
39. http://sipi.usc.edu/database/database.php