Comparative Analysis of EDGE detection using Fuzzy set theory and Automata Theory with existing edge detection methods with CLA

Mauhik Thakkar, Prof.Pranav Lapsiwala
Department of Electronics and Communication
S.C.E.T
Surat, Gujarat, India

Abstract **- In this paper, traditional methods of edge detection and their problems are discussed. After that high performance method for edge detection that is fuzzy logic based image processing is used for accurate and noise free edge detection and Cellular Learning Automata (CLA) is used for enhance the previously-detected edges with the help of the repeatable and neighborhood-considering nature of CLA. In the end, we compare it with popular methods such as Sobel, Canny and Prewitts with enhanced by CLA. Edge is most important visual parameter so if the edges in an image can be identified accurately, all of the objects can be located and basic properties such as area, perimeter, and shape can be measured. In this paper, all the algorithms and result are prepared in MATLAB.**

Keywords **- Edge Detection, Fuzzy Logic, Learning Automata, Cellular Learning Automata, Canny, Sobel**

I. INTRODUCTION

Edge detection is one of the most important algorithms in image processing [4]. It plays a fundamental role in higher level processing. Edges potentially have important information about image content. Also in human visual system, in a preprocessing stage, image edges are detected. Regarding to importance of edges in image processing algorithms, such as object detection, capabilities and accuracy of edge detection algorithms are important factors.

Edge is defined as object border, and extracted by features such as gray, color or texture discontinuities. Luminance and geometrical features, lightening condition and noise volume has a great impact on shaping the edge. Edge contains important information of image and provides object's location. Many of edge detection algorithms such as Sobel, Prewitt and Robert are based on gradient value. In these algorithms, the estimated gradient pixel value is higher than a threshold and counted as an edge pixel. Because threshold value is often empirically determined, it is possible to lose some edges or over estimation occurs. Another important gradient based edge detection method is canny algorithm which solves an optimization problem to detect the edges. The tradeoff between detection and location of edge pixels make a problematic inaccuracy. By changing threshold values, edge detection rate increases, but the accuracy of edge locating decrease. Because of noise, low contrast, and some other factor edge detection methods that have been mentioned cannot give satisfactory results. For example in some cases over or under edge based segmentation estimation occurs, especially in natural complex image is more obvious.

There are some edge detection algorithms in frequency domain. The algorithm presented by Marr-Hildreth detects zero crossing second order of derivative function in Laplace domain as edge pixels [8]. Low sensitivity to noise is achieved regards to using the Gaussian filter. Some other edge detection algorithms process the image in Wavelet domain. In the all above mentioned approaches, after applying edge detection algorithm and then threshold, the thinning stage is followed till a single edge pixel is achieved.

A very important role is played in image analysis by what are termed feature points, pixels that are identified as having a special property. Feature points include edge pixels as determined by the well-known classic edge detectors of PreWitt, Sobel, Marr, and Canny [17:21]. Recently there has been much revived interest [22,23] in feature points determined by "corner" operators such as the Plessey, and interesting point operators such as that introduced by Moravec. [24,25] Classical operators identify a pixel as a particular class of feature point by carrying out some series of operations within a window centered on the pixel under scrutiny. The classic operators work well in circumstances where the area of the image under study is of high contrast. In fact, classic operators work very well within regions of an image that can be simply converted into a binary image by simple thresholding.

Traditional edge detection operator is based on first and second order directional derivative which is based on gray value changes of neighbors of every pixel but sometimes edges are still uncertain, step functions corrupted by Additive White Gaussian Noise (AWGN) [5]. The problem of existing edge detection operator is that the neighborhood of an edge is not involved in the edge detection process, that's why Combined method based on Fuzzy logic [1,2] and CLA [3,5,10,11] is used for edge detection technique is implemented and the edge detected output by fuzzy logic is applied to CLA where using the repeatable and neighborhood considering nature of CLA, the edge pixels are strengthened and the non-edge pixels are weakened and noises of the earlier phase are also removed .

II. PRAPOSED METHOD

A. The simple algorithm of edge detection using fuzzy logic

Fuzzy image processing is the collection of all approaches that understand, represent and process the images, their segments and features as fuzzy sets. Three main stages in fuzzy image processing are: Image fuzzification [2], modification of membership value (Fuzzy Inference system), Defuzzification [2].

Algorithm:

Step 1: Read input image
Step 2: Image fuzzification
Step 3: Fuzzy Inference rule , in that create if- then rules. And create fuzzy logic variables and functions.
Step 4: Defuzzification
Step 5: separate the image in 3x3 windows for window operation.
Step 6: now give penalty or reward on the basis of templates.
Step 7: all pattern checked? If no then go to step 5.
Step 8: Use Cellular Learning Automata for enhancement purpose.
Step 9: final output.

Step 1: Image fuzzification

The image that read is gray scale image and data might range from 0 to 255. The data 0 belongs to black pixel of the image and data 255 belongs to while pixel of the image. In order to apply the fuzzy algorithm, data should be in the range of 0 to 1 only. The image data are converted to this range that is known as membership plane, after the image data are transformed from gray-level plane to the membership plane (fuzzification); appropriate fuzzy techniques modify the membership values. This can be a fuzzy clustering, a fuzzy rule-based approach and a fuzzy integration approach.

Step 2: Fuzzy Inference system

The system implementation was carried out considering that the input image and the output image obtained after de-fuzzification are both 8-bit quantized; this way, their gray levels are always between 0 and 255. The fuzzy sets were created to represent each variable's intensities; these sets were associated to the linguistic variables "Black", Edge and "White". The adopted membership functions for the fuzzy sets associated to the input and to the output were Z-shaped, S-shaped and triangles shaped membership functions, as shown in Fig. 2.

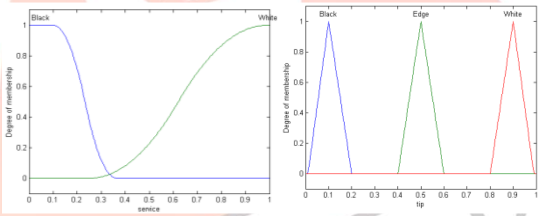


Figure 1. Input and output membership function

The functions adopted to implement the "and" and "or" operations were the minimum and maximum functions respectively. The Mamdani method was chosen as the de-fuzzification procedure [2], which means that the fuzzy sets obtained by applying each inference rule to the input data were joined through the add function; the output of the system was then computed as the low of the resulting membership function. The values of the three membership's function of the output are designed to separate the values of the blacks and whites and edges of the image.

In many image processing applications, expert knowledge is often used to work out the problems. Expert knowledge, in the form of fuzzy if-then rules, is used to deal with imprecise data in fuzzy set theory and fuzzy logic.

[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 2. Gray level Mask

Step 3: Fuzzy inference rule

The Inference rules [2] is depends on the weights of the eight neighbors gray level pixels, if the neighbor's weights are degree of blacks or degree of whites. The powerful of these rules is the ability of extract all edges in the processed image directly. This study is as saying all the pixels of the processed image by studying the situation of each neighbor of each pixel. The condition of each pixel is decided by using the floating 3x3 mask which can be scanning the all grays, is shown in Fig. 3. In this location, some of the desired rules are explained. The first four rules are dealing with the vertical and horizontal direction lines gray level values around the checked or centered pixel of the mask. If the grays represented in one line are black and the remains grays are white then, the checked pixel is edge Fig. 4. The second four rules are dealing with the eight neighbors also depending on the values of the gray level weights, if the weights of the four sequential pixels are degree of blacks and the weights of the remain fours neighbors are the degree of whites, then the center pixel represents the edge Fig. 5.

Rule1	If {(i-1, j-1) & (i-1, j) & (i-1, j+1) } are whites If {(i, j-1) & (i, j) & (i, j+1) } are whites If {(i+1, j-1) & (i+1, j) & (i+1, j+1)} are blacks	checked pixel is **Edge**
Rule2	If {(i-1, j-1) & (i-1, j) & (i-1, j+1) } are blacks If {(i, j-1) & (i, j) & (i, j+1) } are whites If {(i+1, j-1) & (i+1, j) & (i+1, j+1)} are whites	checked pixel is **Edge**
Rule3	If {(i-1, j-1) & (i, j-1) & (i+1, j-1) } are blacks If {(i-1, j) & (i, j) & (i+1, j) } are whites If {(i-1, j+1) & (i, j+1) & (i+1, j+1)} are whites	checked pixel is **Edge**
Rule4	If {(i-1, j-1) & (i, j-1) & (i+1, j-1) } are whites If {(i-1, j) & (i, j) & (i+1, j) } are whites If {(i-1, j+1) & (i, j+1) & (i+1, j+1)} are blacks	checked pixel is **Edge**

Figure 3. Fuzzy system rules

Rule5	If {(i-1, j) & (i-1, j-1) & (i, j-1) & (i+1, j-1)} are blacks If {(i-1, j+1) & [i, j+1] & (i+1, j+1) &(i+1, j)} are whites If (i, j) is white	checked pixel is **Edge**
Rule6	If {(i-1, j) & (i-1, j-1) & (i, j-1) & (i+1, j-1)} are whites If {(i-1, j+1) & [i, j+1] & (i+1, j+1) & (i+1, j)} are blacks If (i, j) is white	checked pixel is **Edge**
Rule7	If {(i-1, j-1) & (i, j-1) & (i+1, j-1) & (i+1, j)} are blacks If {(i-1, j) & (i-1, j+1) & (i, j+1) & (i+1, j+1)} are whites If (i, j) is white	checked pixel is **Edge**
Rule8	If {(i-1, j) & (i-1, j+1) & (i, j+1) & (i+1, j+1)} are blacks If {(i-1, j-1) & (i, j-1) & (i+1, j-1) & (i+1, j)} are whites	checked pixel is **Edge**

Figure 4. Fuzzy system rules

The introduced rules and another group of rules are detecting the edges, the white and the black pixels. The result images contribute the contours, the black and the white areas.

Step 3: Defuzzification

From the side of the fuzzy construction, the input grays is ranged from 0-255 gray intensity, and according to the desired rules the gray level is converted to the values of the membership functions . The Mamdani method was chosen as the defuzzification procedure [2], the output of the FIS according to the defuzzification is presented again to the values from 0-255 and then the black, white and edge are detected.

B. Edge enhancement using Cellular learning Automata

Basics criteria that are typically used for testing the quality of an edge detector is to find the probability of a false positive (marking something as an edge which isn't an edge) ,The probability of a false negative (failing to mark an edge which actually exists) and the mean square distance of the edge estimate from the true edge [12]. To find out that initially Learning Automata [6,11] are developed which is having infinite states Each selected state gets evaluated by a probabilistic environment and by means of a positive or negative signal the result of evaluation, which an automaton uses to determine the next state, is given to the automaton. The ultimate goal is teaching an automaton how to select the best state from all possible choices. The best state is a state that maximizes the reward received by the environment. Environment can be described by the triplet

$E=\{\alpha,\beta,c\}$ Where, $\alpha=\{\alpha_1,\alpha_2,\alpha_3...\alpha_r\}$ is a set of input for LA.

$\beta=\{\beta_1,\beta_2,\beta_3....\beta_m\}$ β is the set of outputs

$c=\{c_1,c_2,c_3....c_r\}$ is a set of penalty probabilities

Variable structure learning automaton is described by the quadruplet $\{\alpha,\beta,\rho,T\}$ where

$\alpha=\{\alpha_1,\alpha_2,\alpha_3...\alpha_r\}$ is a set of states,

$\beta=\{\beta_1,\beta_2,\beta_3....\beta_m\}$ is a set of inputs

$\rho=\{\rho_1,\rho_2,\rho_3...\rho_r\}$ is the probability vector of choosing

Each state by LA, and

$\rho(n+1)=T[\alpha(n),\beta(n),\rho(n)]$ is the learning algorithm.

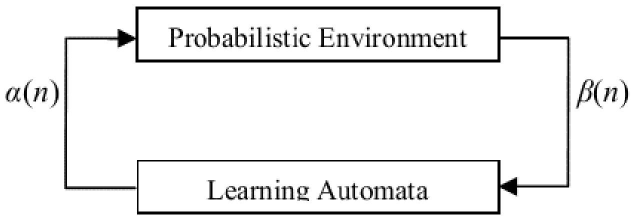


Figure 5. The interaction between probabilistic environment and learning automata

But, some cases LA can't do the learning task properly, in that case CLA uses different kinds of neighborhoods, the most common kinds of neighborhoods are Von Neumann, Moore, Smith, and Cole, which are known as "nearest neighbors" neighborhoods [2]. These neighborhoods are given as:

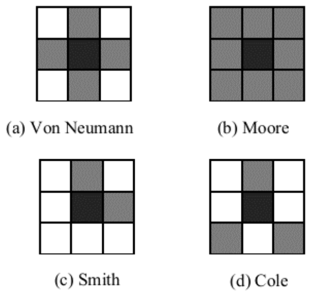


Figure 6. Types of neighborhood

Here in CLA we uses Moore neighborhood and each cell of the image is considered to be a variable structure learning automaton which has relations with its neighboring automata with radius 1, Each learning automaton has two states: edge and non-edge. Initially each learning automaton is determined by the final image X' of the pre-processing phase. Local rules of these CLA are defined in such a way that in continuous repetitions strengthen edge pixels and weaken non-edge pixels and noises. It should also be able to strengthen edge pixels that are in the middle of two edge pixels which are detected as a non-edge pixel or a weak edge and on the other hand be able to weaken non-edge pixels which are detected as strong edges. To improve the edges if two to four neighbors of a learning automaton and the central learning automaton decided that the pixel is an edge, the central pixel is rewarded and if not, it is penalized.

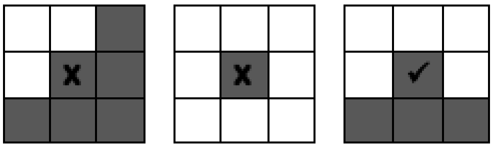


Figure 7. Strengthening and weakning with CLA

If more than four neighbors of the central learning automaton or none of them decided that the pixel is an edge but the central learning automaton decided that it is an edge, the pixel is penalized in order to weaken the edge. Fig. 8 explains the above statement. The black cells imply that the learning automaton of that cell has decided it is an edge.

If only two neighbors of the learning automaton consider their pixel as an edge and the central learning automaton does the same, it is rewarded. But if the central learning automaton doesn't consider its pixel as an edge then, it is penalized. This is done to improve or weaken the separated edges. Fig. 9 describes the point perfectly.

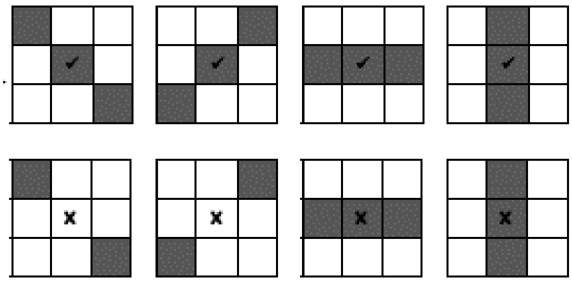


Figure 8. Strengthening and weakening of connected and separated edges

At each time, the rules for receives a reward or a penalty for selected state, with respect to neighboring cells are, all LA in a cellular learning automaton select a state from their set of states. This selection can be based upon either prior observations or random selection.

Patterns that generates penalty are shown in Fig. 10. In these patterns, white cells are edges and black cells are non-edges. These patterns create thick edges, noises and unwanted edges. Some patterns shown in Fig. 10 are representing other patterns that receive penalty, which are obtained by rotating or flipping these patterns. Due to the similarities, these patterns are not shown in Fig. 10. All the other patterns receive a reward. The process of updating cells and giving penalties and rewards continues until the system reaches a stable state or satisfies a predefined condition.

Penalties and rewards given to each cell of the CLA are respectively calculated by formulas (1) and (2):

$$\begin{aligned} \rho_i(n+1) &= (1-\beta)\rho_i(n) \\ \rho_j(n+1) &= \beta(255-\rho_j(n)) \end{aligned} \quad \forall_j \quad i \neq j \qquad (1)$$

$$\begin{aligned} \rho_i(n+1) &= \rho_i(n) + \alpha(255-\rho_i(n)) \\ \rho_j(n+1) &= (1+\alpha)\rho_j(n) \end{aligned} \quad \forall_j \quad i \neq j \qquad (2)$$

Where α is the probability increase coefficient and β is the probability reduction coefficient (α<<β).

C. Comparitive analysis

The proposed system was tested for different images, its performance being compared to that of the Sobel, Prewitt, Canny operator and the proposed FIS method. The firing order associated with each fuzzy rule were tuned to obtain good results while extracting edges of the image, where we used this image as comparative model for the classical Sobel, Prewitt, Canny operator and the FIS method.

The edge detection based on Sobel, Prewitt, and Canny operator using the image processing toolbox in MATLAB is illustrated. The white pixels on the map indicate there are edges, thus will be preserved from smoothing. There is obviously some noise left on the edge map and some of the edges are corrupted. By applying the new FIS on the image to detect its edges, it is found that the modified version of edge map has less noise and less edge corruption as shown in the Fig. 11.

For the segmentation task, a thin edge is better because only want to preserve the edge rather than the details in the neighborhood. The values of the edge map are normalized to the interval of 0 and 1 to represent the edginess membership values. The original captured image is shown in Fig.11, in part (b) Sobel, part (c) Prewitt, part (d) Canny operator with threshold automatically estimated from image's binary value does not allow edges to be detected in the regions of low contrast and enhanced all the images with CLA. The FIS system, in turn, allows edges to be detected even in the low contrast regions as illustrated in Fig. 11, part (e). This is due to the different treatment given by the fuzzy rules to the regions with different contrast levels, and to the rule established to avoid including in the output image pixels not belonging to continuous lines. Finally edge enhancement result shows in Fig. 11, part (f) (Edge using Fuzzy and CLA).

III. Conclusion

Image quality measurement plays an important role in various image processing applications In this paper different techniques are applied on different images & among these techniques fuzzy method is the best method because in this PSNR has maximum value and MSE has lower value and images are clearly visible.

The designed fuzzy rules with CLA are an attractive solution to improve the quality of edges as much as possible. One past drawback of this type of algorithm was that they required extensive computation.

References

[1] Hong Wang, Junna Jiang, Lihui Zhou and Liping Du, "Image Processing Based on Fuzzy Cellular Automata Model" , International Conference on Innovative Computing, Information and Control, pp. 954-957, 2009,

[2] Abdallah A. Alshennawy, and Ayman A. Aly, "Edge Detection in Digital Images Using Fuzzy Logic Technique" *World Academy of Science, Engineering and Technology 51*, pp. 178-186, 2009.

[3] Ehsan Nadernejad, Sara Sharifzadeh, Hamid Hassanpour, "Edge Detection Techniques: Evaluations and Comparisons" *Applied Mathematical Sciences*, Vol. 2, pp. 1507 – 1520,2008.

[4] Xiangtao Chen, Yujuan Chen, "An Improved Edge Detection in Noisy Image Using Fuzzy Enhancement" In proceeding of: Biomedical Engineering and Computer Science (ICBECS), 2010

[5] S. K. Pal, R. A. King, "Image Enhancement Using Fuzzy Sets," *Electronics Letters*, Vol. 16, pp. 376-378, 1980.

[6] M.C. Shin, D. Goldgof, and K.W. Bowyer ."Comparison of Edge Detector Performance through Use in an Object Recognition Task". Computer Vision and Image Understanding, vol. 84, no. 1, pp. 160-178, Oct.2001

[7] Gonzalez, Woods, "Digital Image Processing", Vol. 2nd Edition, Prentice Hall, 2001.

[8] Hong Wang, Junna Jiang, Lihui Zhou and Liping Du, "Image Processing Based on Fuzzy Cellular Automata Model" , International Conference on Innovative Computing, Information and Control, pp. 954-957, 2009,

[9] R.Enayatifar, M.R.Meybodi, "Adaptive Edge Detection via Image Statistic Features and Hybrid Model of Fuzzy Cellular Automata and Cellular Learning Automata", *International Conference on Information and Multimedia Technology ICIMT*, IEEE, pp. 273-278, 2009.

[10] Yong Yang, "An adaptive fuzzy-based edge detection algorithm", *IEEE proceedings of 2007 International Symposium on Intelligent Signal Processing and Communication Systems*, Nov.28-Dec.1, 2007.

[11] D. Marr and E. Hildreth, "Theory of Edge Detection" Proceedings of the Royal Society of London. Series B, Biological Sciences, Vol. 207, No.1167 (Feb. 29, 1980).

[12] P. Jebaraj Selvapeter, Wim Hordijk, "Cellular Automata for Image Noise Filtering" 978-1-4244-5612-3/09 IEEE 2009.

[13] Yan Ha, "Method of edge detection based on Non-linear cellular

[14]

[15] Automata", *Proceedings of the 7th World Congress on Intelligent Control and Automation*, June 25 - 27, 2008.

[16] Amir Hosein Fathy, Amir Bagheri, "Cellular Learning Automata" , (2013,May 8),http://www.intechopen.com/books/emerging-applications-of-cellular- automata/cellular-learning-automata-and-its-applications

[17] Cuevas, E.; Wario, F.; Zaldivar, D.; Pérez-Cisneros, M., "Circle detection on images using learning automata," Computer Vision, IET , vol.6, no.2, pp.121,132, March 2012

Part a: Original Image	Machine.jpg	Wheel.png	Obama.jpg	H1_gray.jpg

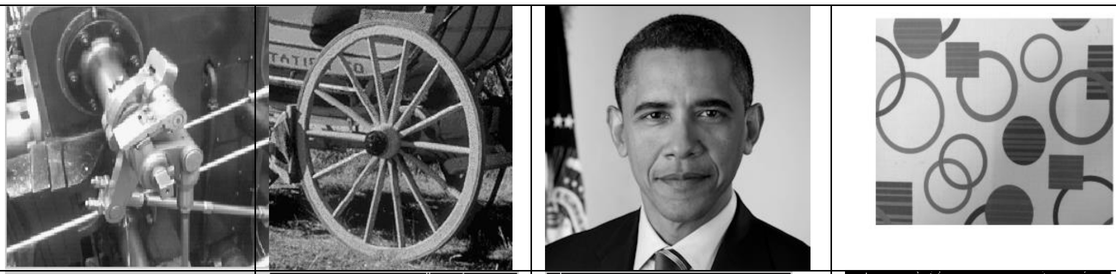
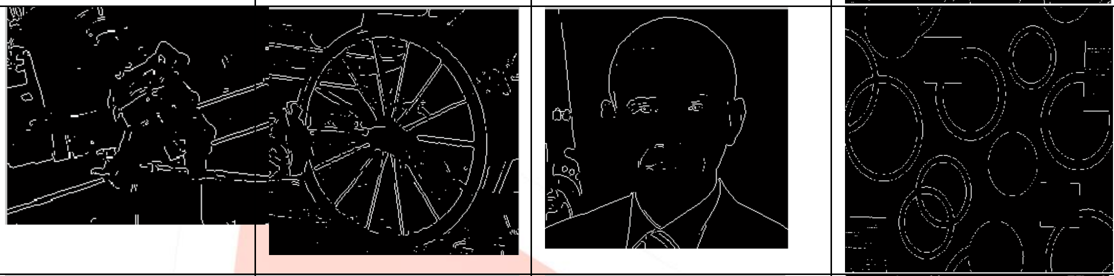
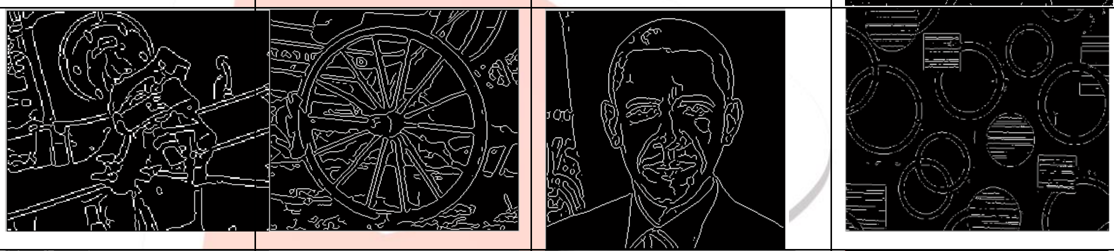
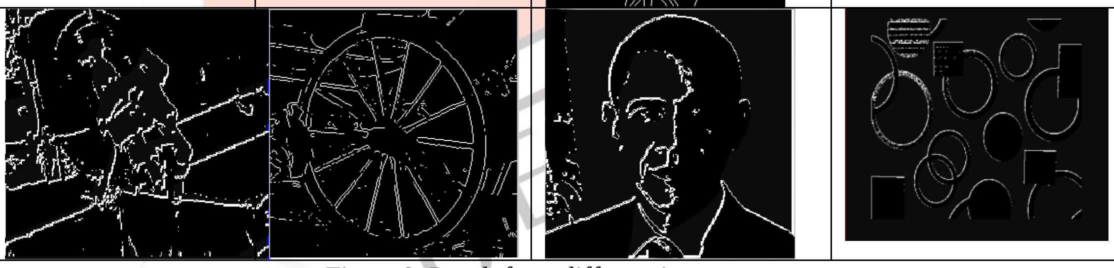
	
Part b: Sobel Edge + CLA	
Part c: Prewitt Edge + CLA	
Part d: Canny Edge + CLA	
Part e: Edge using Fuzzy and CLA	

Figure 9. Result from different images

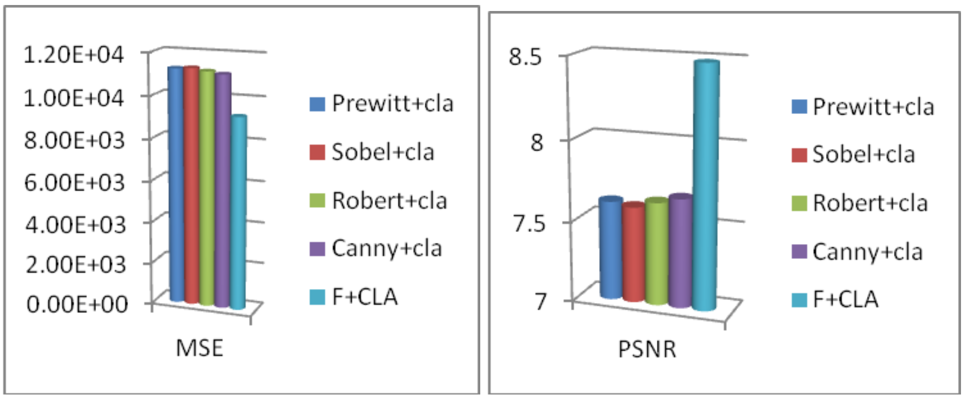


Figure 10. Analysis of wheel image

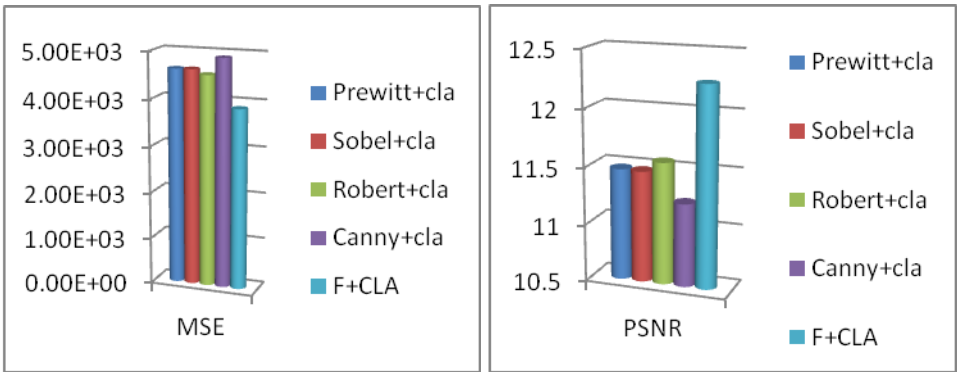


Figure 11. analysis of obama image

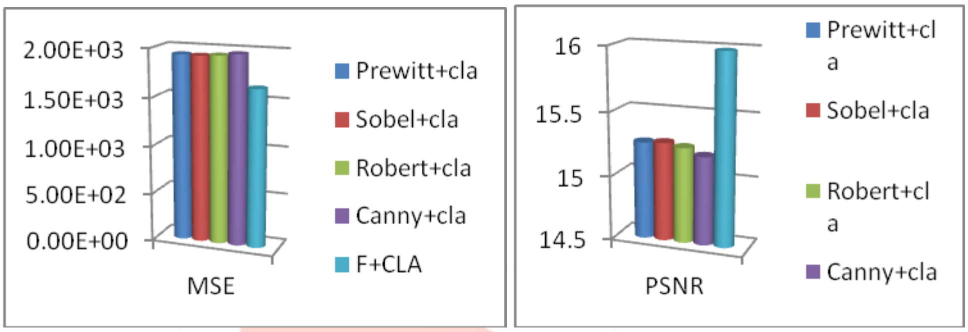


Figure 12. analysis of machine image

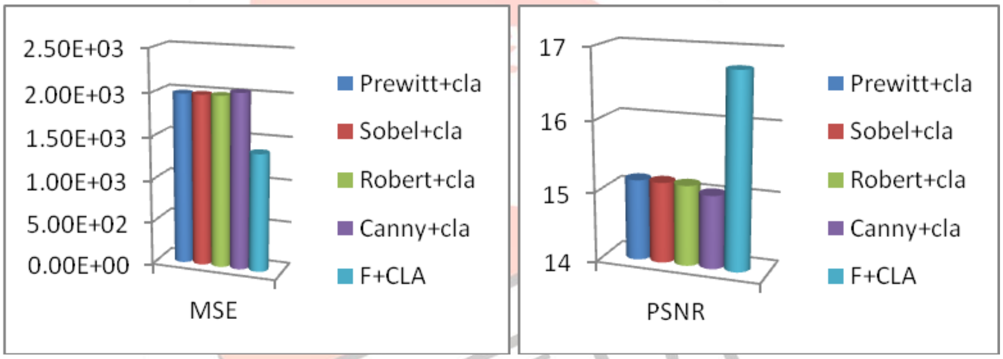


Figure 13. analysis of h1_gray image