Fuzzy Logic Based Enhanced Edge Detection Technique for Defence Imageries

Paramjeet Singh[1], M J Nigam[2]
[1, 2] Indian Institute of Technology, Roorkee, India
[1]ps.badial@gmail.com, [2]mkndnfec@gmail.com

Abstract— Edge detection is useful in feature detection, medical diagnosis and defence applications. In this paper, a hybridized edge detection algorithm has been proposed using Fuzzy Logic, image enlargement, image sharpening and contrast enhancement and results have been compared with the existing methods. The simulation results have been obtained using MATLAB R2013a and analyzed qualitatively. Comparison of results show that the proposed algorithm is an efficient and effective method for defence imageries.

Index Terms— Enhanced Edge Detection, Image Sharpening, Contrast Enhancement, Fuzzy Logic, Fuzzy Logic Edge Detection, Defence Imageries.

1. INTRODUCTION

An edge may be characterized by a change in intensity with some directional behavior contained within an image. It forms an essential step in many applications like segmentation, object detection and tracking, feature extraction, medical imaging, defence applications etc. The edge information performs a significant role while dealing with noise removal and deblurring of images [1]. Based on the levels of intensity variations we can broadly classify the edges into two categories: strong edges, which corresponds to large step variations and weak edges, which corresponds to small step variations. As far as detection of strong edges is concerned, there are many methods which may be applied however, most of them are unable to detect weak edges [2]. In this paper, a hybridized edge detection algorithm has been proposed that is capable of detecting both types of edges when applied to defence imageries.

In literature, many gradient based edge detectors like Prewitt, Sobel, Roberts, and LoG have been proposed [3] and these methods forms the basis of all gradient operators. Canny [4] proposed an optimal algorithm, where the image is smoothened using Gaussian filter and then directional gradients are applied to obtain edge information. Prewitt operator comprises of two separate horizontal and vertical gradient operators. Sobel operator results when we double the weight to central pixel of Prewitt operator. Method of computing diagonal gradients, i.e., cross gradients is used in case of Roberts detector. In the case of LoG edge detector, the second derivative for edge detection is approximated by applying Gaussian convolution and Laplacian. The edge detectors involving gradient operators and Gaussian convolution may be prone to the problems like improper localization of edges, missing and false edges [5].

Fuzzy theory advancement has lead to the growth of numerous edge detectors. Accordingly, the ambiguity present in edge information, i.e., location and evident clarity, may be dealt with fuzzy reasoning. Bezdek *et al.* [6] utilized fuzzy logic in modeling functions for edge detection based on geometric feature. The fuzzy reasoning also finds its importance when applied for edges in noisy images [7], it has also been integrated with morphological edge extraction method along with minimization of fuzzy entropy [8].

Structured decision forests have also been used for fast edge detection by P. Dollar, which results in a time efficient algorithm [9]. Gabor wavelet technique has also been utilized for edge detection in medical images [10]. Edge detector based on SUSAN principle has also been proposed [2]. All these edge detection techniques work fine for normal images however, when it comes to images with strong and weak edges there is high probability of incorrect edge information.

We have proposed an algorithm to handle defence imageries efficiently.

2. PROPOSED ALGORITHM

The proposed algorithm for enhanced edge detection comprises of various steps before applying actual edge detection procedure. The flowchart may be depicted as under.

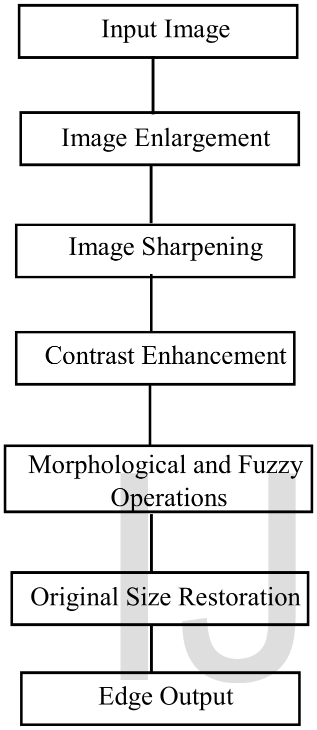

Figure 1: Proposed Algorithm Flowchart

The details about all the steps in Fig. 1, are discussed as follows.

2.1 Image Input

The image is read into the workspace of Matlab, and in case the input image happens to be a color image, conversion to a grayscale image is performed. The approach used here for color to grayscale conversion of image is the line projection shown as under.

$$Igray \quad {}_R\, R + \alpha_G\, G + \alpha_B\, B \qquad (1)$$

Where, $Igray$ is the required grayscale image, R G B are the color components and the non-negative coefficients α_R , α_G , and α_B satisfies.

$$\alpha_R + \alpha_G + \alpha_B = 1 \qquad (2)$$

The values for α_R, α_G and α_B are taken as, 0 .2126, 0.7152 and 0.0722 respectively as used in HDTV [11].

2.2 Image Enlargement

After converting the image to grayscale, enlargement of image is performed such that every pixel of input image stands represented by a patch of 3×3 pixels. The enlargement procedure may be depicted as under.

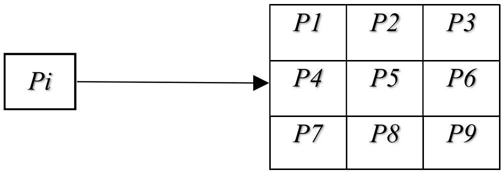

Figure 2: Pixel-wise Enlargement

Where, Pi is one of the pixels of original grayscale image and it is represented by 3×3 pixels for enlargement process as depicted in Fig.2. Nearest neighbor algorithm with box shaped kernel for interpolation has been used in this case as it is one of the simplest and faster algorithms available. The idea for this step has been taken from work done by B. Gardiner *et al.* [12] and has been changed accordingly. The process of enlargement enables to capture even smaller details in the image.

2.3 Image Sharpening

The enlarged image is then enhanced using sharpening process. Image sharpening is performed by employing unsharp masking algorithm [13]. Here enlarged image is blurred by passing through a Gaussian low pass filter and then this blurred image is subtracted from the original enlarged image as depicted in Fig.3.

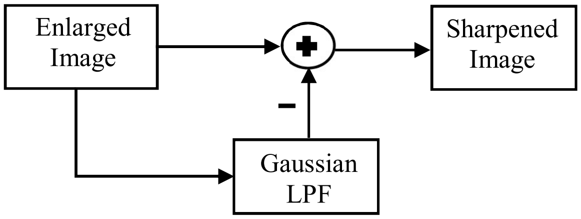

Figure 3: Sharpening Algorithm

Standard deviation of the Gaussian filter is selected as 1, which selects the extent of the region around the edge pixels to be sharpened. Large values sharpen wider region, whereas small values lead to narrow region sharpening around the edge pixels. Strength of the sharpening effect has been set at 0.8, higher the value larger is the increase in contrast of sharpened pixels. Minimum contrast which is essential for a pixel to be considered for sharpening varies between 0 and 1, in this case for ensuring that most of the details are covered we have selected it as 0.

2.4 Contrast Enhancement

The sharpened image is then enhanced in terms of contrast using the step of contrast enhancement [14]. Here the histogram of the image is obtained and by taking inference from the histogram the intensity value limits are determined. The low and high limits of intensity are specified as a fraction between 0.0 and 1.0, so that we can represent them as a [*low, high*] vector. We take *high* and *low* limits as the intensity figures that denote top 1% and bottom 1% of the range respectively. Now we define the output intensity limits as [*bottom, top*] as [0, 1] and map *low* to *bottom* and *high* to *top* with a correction factor, $\gamma = 0.8$, as depicted in Fig.4.

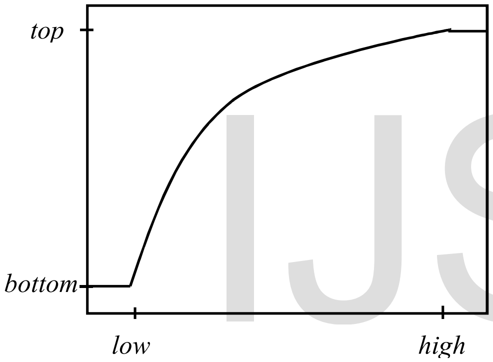

Figure 4: Contrast Mapping

The correction factor $\gamma = 1$, corresponds to linear relationship, $\gamma < 1$, will make the image brighter and $\gamma > 1$, will make it darker. Using this makes the image much clear than the original one.

2.5 Morphological and Fuzzy Logic Operations

2.5.1 Morphological Gradient

Morphological gradient for a gray-scale image can be termed as the variation among the intensity values of two adjoining pixels. The gradient operator ∇, which is the basis of gradient edge detection can be approximated by calculating the intensity change in i^{th} direction, Di. We can apply a 3 × 3 matrix for the coefficients Mi, as shown in Fig. 5, where the center coefficient is $M5 = f(x,y)$ and other coefficients as referenced to $M5$, like $M1 = f(x-1,y-1)$, $M2 = f(x,y-1)$, $M3 = f(x+1,y-1)$, $M4 = f(x-1,y)$, $M6 = f(x+1,y)$, $M7 = f(x-1,y+1)$, $M8 = f(x-1,y)$ and $M9 = f(x+1,y+1)$ [15].

M 1	*M* 2	*M* 3
M 4	*M* 5	*M* 6
M 7	*M* 8	*M* 9

Figure 5: Mask for Obtaining *Di* Coefficients

We can now obtain the edges in four different directions as under by using the above mask over the input image using the relations as described below.

$$D1 = \sqrt{(M5-M2)^2-(M5-M8)^2}$$
$$D2 = \sqrt{(M5-M4)^2-(M5-M6)^2}$$
$$D3 = \sqrt{(M5-M1)^2-(M5-M9)^2}$$
$$D4 = \sqrt{(M5-M3)^2-(M5-M7)^2} \qquad (3)$$

Now we can obtain the absolute gradient edges, S by adding the Di's as under.

$$S = D1 + D2 + D3 + D4 \qquad (4)$$

After contrast enhancement the image is operated upon by the mask given in Fig.5, and the coefficients Di's are obtained using equation (3).

2.5.2 Fuzzification

The process of fuzzification uses the Fuzzy Logic System (FLS) for mapping sharp input values into type-2 fuzzy sets. Here we have used FLS, which maps the sharp input, x'_p to a type-2 fuzzy singleton, whose membership function is μ_{Ap} (x_p) = 1/1 for $x_p = x'_p$ and μ_{Ap} (x_p) = 0 for all $x_p \neq x'_p$ for p = 1, 2, ..., *P,* where P represents the number of FLS inputs [16].

In this case, the inputs are taken to be the gradients, $Di's$ as obtained in previous step, and all of these will be given as input to the fuzzy logic system. The general fuzzy inference system used here may be depicted as shown in Fig. 6.

(a) Input variables: Here four inputs have been taken, and each of them has three membership functions which are Gaussian with uncertain mean.

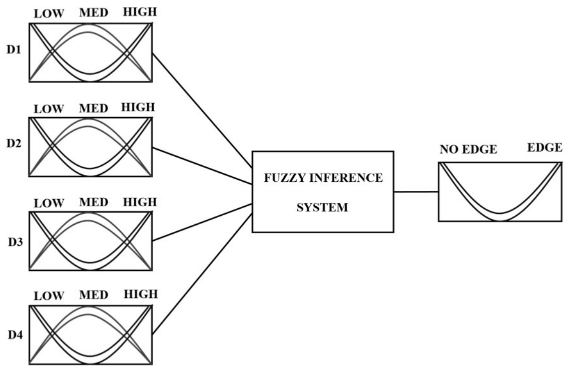

Figure 6: Fuzzy Inference System

The required variables in linguistic terms utilized for the inputs are: low, med, and high. To adjust the membership functions to the gray scale range, we obtain these values as under [17].

$$\begin{aligned} low_i &= mi\ \ (Di) \\ high_i &= max(\mathrm{Di}) \\ med_\mathrm{i} &= low_i + (high_i - low_i)/2 \end{aligned} \qquad (5)$$

These values can be used to get the mean for the membership functions, with adding various sizes of the Footprints of Uncertainty (*FoU*), taking the value of *FoU* = 0.2, as shown below.

$$\begin{aligned} m1 &= high_i \\ m2 &= m1 + (m1 \times FoU) \\ mx &= (m1 + m2)/2 \\ \sigma i &= high_i/5 \end{aligned} \qquad (6)$$

Where, $m1, m2$ are the means and σ_i is the variance. Using these the Gaussian membership functions may be obtained as under.

$$\mu(x,u) = exp\left[-\frac{1}{2}\left(\frac{x-px}{\sigma u}\right)^2\right] \qquad (7)$$

Where, px is given as under.

$$px = exp\left[-\frac{1}{2}\left(\frac{x-mx}{\sigma x}\right)^2\right] \qquad (8)$$

(b) Output Variables: The inference system gives one variable as output, *S* (edge) in the range [0,1] and we have used the linguistic values as: *edge* and *no edge*. The minimum and maximum values for the output are given as under.

$$\begin{aligned} no\ edge_i &= 0 \\ edge_i &= 1 \end{aligned} \qquad (9)$$

The Gaussian membership functions in respect of output are obtained using equations (7) and (8), after substituting the values of means, $mo1$ and $mo2$ of each function and the σy value which are given as under [17].

$$\begin{aligned} mo1 &= edge_i \\ mo2 &= mo1 + (mo1 \times FOU) \\ \sigma y &= edge_i/4 \end{aligned} \qquad (10)$$

(c) Defining rules: Here three rules have been considered that define the relationship among the various image gradients as obtained. These are summed up as.

i) If ($D1$ or $D2$ or $D3$ or $D4$ is HIGH), then (S is EDGE).
ii) If ($D1$ or $D2$ or $D3$ or $D4$ is MED), then (S is EDGE).
iii) If ($D1$ and $D2$ and $D3$ and $D4$ is LOW), then (S is NO EDGE).

(d) Type reduction: Here the conversion from type-2 to type-1 fuzzy system is carried out. We use the method of approximation which may be explained by the following theorem. If each of, $Pl \in [\,\underline{Pl}, \overline{Pl}\,]$ forms an interval type-1 set with center, cl and spreads sl, and if every, $Vl \in [\,\underline{\mu l}\,(y), \overline{\mu l}\,(y)\,]$ also forms an interval type-1 set with center hl, and spreads Δl, then *Y* may be approximated as an interval type-1 set, with center, C and spread S, which can be described as under [18].

$$[\hat{y}\ left, \hat{y}\ right] = Y(P1, \ldots\ldots Pm, V1, \ldots\ldots\ldots Vm)$$
$$= \left(\int_{P1} \cdots \int_{Pm} \int_{V1} \cdots \int_{Vm} 1\right) \Big/ \frac{\sum_{l=1}^{m} Pl\,Vl}{\sum_{l=1}^{m} Vl} \qquad (11)$$

$$C = \frac{\sum_{l=1}^{m} hl\,cl}{\sum_{l=1}^{m} hl} \qquad (12)$$

$$S = \frac{\sum_{l=1}^{m}[hl\,sl + |cl - C|\Delta l]}{\sum_{l=1}^{m} hl} \qquad (13)$$

Provided following condition is true.

$$\frac{\sum_{l=1}^{m} \Delta l}{\sum_{l=1}^{m} hl} \ll 1 \qquad (14)$$

Using above relations, we can calculate the values of output and these are found to be given by following relations.

$$\begin{aligned} \hat{y}\ left_j\,(x') &= C - S \\ \hat{y}\ right_j\,(x') &= C + S \end{aligned} \qquad (15)$$

(e) Defuzzification: After performing the reduction to type-1 and integration of all the results, defuzzification is done to obtain the final output by taking the average as under.

$$\hat{y}_j(x') = [\hat{y}\, left_j(x') + \hat{y}\, right_j(x')]/2 \qquad (16)$$

2.6 Original Size Restoration

The edges that are obtained in previous step are for the enlarged image, so it needs to be restored to its original image size. For achieving this we need to perform the reverse of image enlargement that is image shrinking. For achieving that we take a set of 3×3 pixels and represent them by a single value pixel as depicted below in Fig. 7.

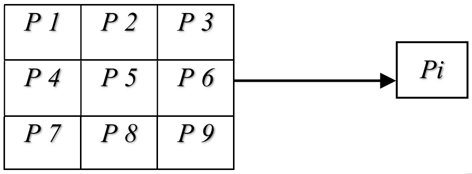

Figure 7: Pixel-wise Image Shrinking

For every set of 3×3 pixels we obtain a single value pixel, *Pi* by taking the weighted average of the neighbor pixels. Here bicubic interpolation technique along with antialiasing has been used to ensure non repetition of edges in output. In case the average of these pixels comes out to be equal or greater than 0.5, the value of *Pi* is set to be 1, else it is set to be 0. By performing this we get the edge image which is same as the size of the original image and a pixel value of 1 denotes edge pixel and a pixel value of 0 denotes a non-edge pixel. Thus the resultant image turns out to be a binary image.

2.7 Edge Display

The edge map that has been obtained now can be visually seen using Matlab. Pixel values which are equal to 0 are shown as black color and the pixel values which are equal to 1 are represented as white color. Therefore, we get the edge image as a black and white image with white color depicting the edges while black color depicts the homogenous regions of the image. The image output may be shown using inbuilt MATLAB function (imshow). In addition to image display using this function the copy of the output edges may also be saved on to the hard disk of the computer for intended future reference and use.

3. RESULTS

The results of the proposed algorithm, Sobel, LoG, Canny, Gabor Wavelet, SUSAN, and Structured Random Forests methods have been presented in table of Fig. 8.

Numerous defence images have been used for the experimentation purpose. However, for the ease of representation, edge outputs for three images viz. Fighter Plane (Sukhoi.jpg), Tank (T90.jpg), and Artillery Gun (Bofors.jpg) have been compared.

The resultant edges using Sobel, LoG and Canny techniques are quite comparable to each other and are prone to false edges and loss of edge information. The edges obtained using Structured Forests technique, Gabor Wavelet method and SUSAN method are found to be better than the results of Sobel, LoG and Canny methods but, by looking at the results it is clearly visible that even these methods are not effective for capturing the finer details of the input images. This leads to loss of information in terms of weak edges.

Original image, Sukhoi.jpg as shown in Fig. 8 has been considered and output data quality with respect to all the methods has been analyzed and it is observed that the proposed method gives all the edges whereas, in other methods edges obtained suffer from problems like missing edges, false edges, bifurcation, and comparatively lesser brightness. Similarly, it can be explained for the other images. Thus resultant edges obtained using the proposed algorithm detects nearly all the edge information including the weak edges which were not covered by the other techniques.

4. CONCLUSION

The proposed algorithm for edge detection is capable of detecting the strong as well as weak edges. Simulation results using the proposed algorithm, Sobel, LoG, Canny, Gabor Wavelet, Structured Forests method and SUSAN edge detection techniques when applied to various defence imageries are as shown in Fig. 8.

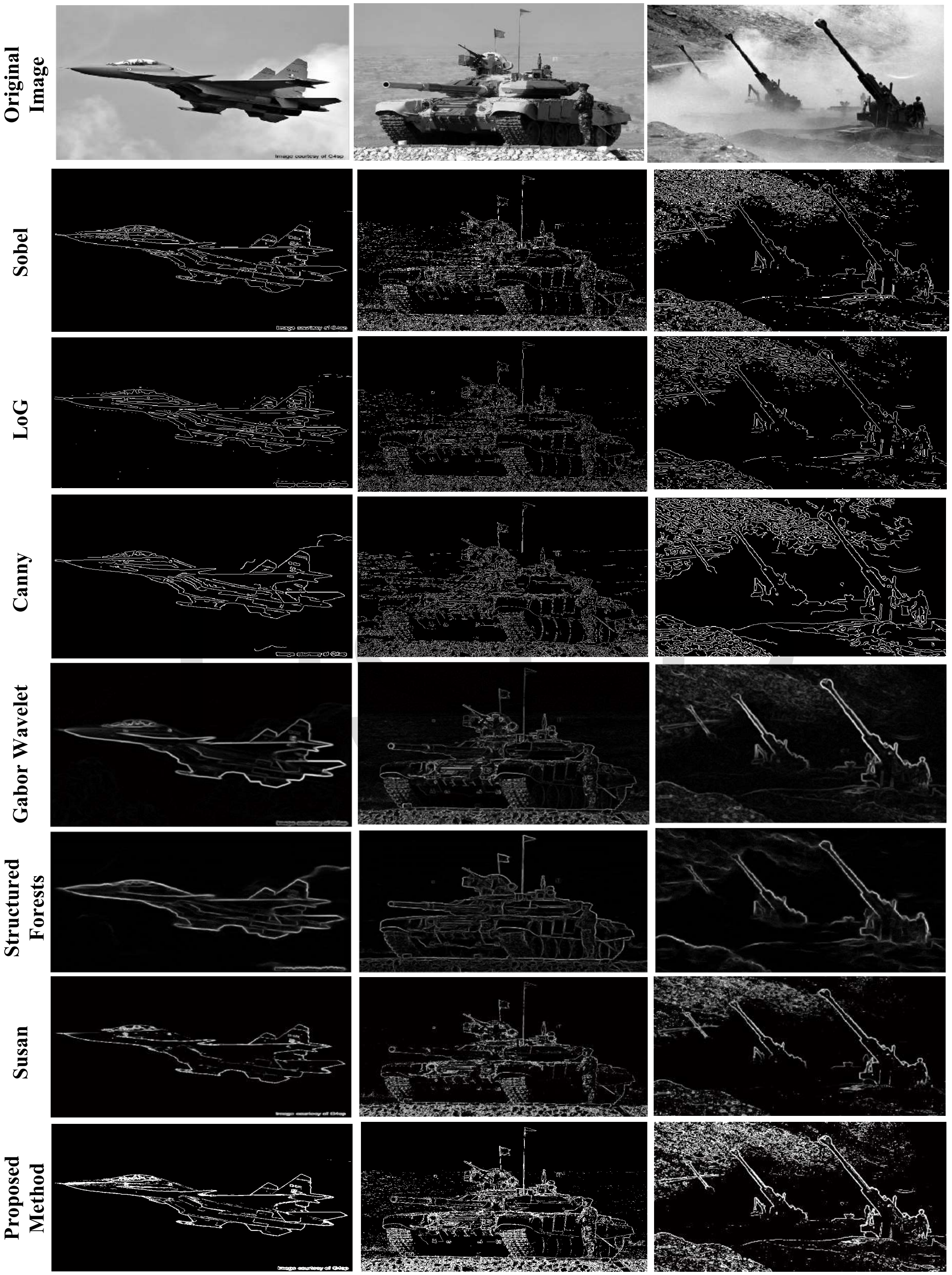

Figure 8: Results for Fighter Plane, Tank and Artillery Gun

Qualitative comparison of the results obtained clearly establish that the edges obtained by applying the proposed algorithm are the best in terms of missing edges, false edges, bifurcation, and clarity.

References

[1] A. A. Goshtasby, " 2-D and 3-D Image Registration: For Medical, Remote Sensing, and Industrial Applications, " Hoboken, NJ, USA: Wiley, pp. 34–39, 2005.

[2] Om Prakash Verma, Anil Singh Parihar, "An Optimal Fuzzy System for Edge Detection in Color Images Using Bacterial Foraging Algorithm," IEEE Transactions On Fuzzy Systems, vol. 25, No. 1, pp. 114-127, Feb 2017.

[3] R. C. Gonzalez and R. E.Woods, "Digital Image Processing," 3rd ed, Noida, India, Pearson.

[4] J. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[5] M.Basu, "Gaussian based edge detection methods: A survey," IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 32, no. 3, pp. 252–260, Aug 2002.

[6] C. L. Molina, B. De Baets, H. Bustince, and E. Barrenechea, "Multiscale edge detection based on Gaussian smoothing and edge tracking," Knowl.-Based Syst., vol. 44, pp. 101–111, 2013.

[7] F. Russo, "Edge detection in noisy images using fuzzy reasoning," IEEE Trans. Instrum. Meas., vol. 47, no. 5, pp. 1102–1105, Oct. 1998.

[8] I. Bloch, "Fuzzy sets in image processing," in Proc. ACM Symp. Appl. Comput., New York, NY, USA, pp. 175–179, Mar. 6–8, 1994.

[9] P. Dollar and C. Lawrence Zitnick, "Fast Edge Detection Using Structured Forests, "IEEE Transactions On Pattern Analysis And Machine Intelligence, vol. 37, no. 8, Aug 2015.

[10] B. Ergen, "A Fusion Method of Gabor Wavelet Transform and Unsupervised Clustering Algorithms for Tissue Edge Detection, "The Scientific World Journal, Hindawi, vol 2014, Mar 2014.

[11] Yi Wan, Qisong Xie, "A Novel Framework for Optimal RGB to Grayscale Image Conversion, " IEEE 8th International Conference on Intelligent Human-Machine Systems and Cybernetics, 2016.

[12] B. Gardiner, S. A. Coleman, and B. W. Scotney, " Multiscale Edge Detection Using a Finite Element Framework for Hexagonal Pixel-Based Images, " IEEE Transactions On Image Processing, Vol. 25, No. 4, April 2016.

[13] Guang Deng, " A Generalized Unsharp Masking Algorithm," IEEE Transactions On Image Processing, Vol. 20, No. 5, May 2011.

[14] Randeep Kaur, and Sandeep Kaur, "Comparison of Contrast Enhancement Techniques for Medical Images," IEEE International conference on Emerging Devices and Smart Systems (ICEDSS), Mar 2016.

[15] P. Melin, O. Mendoza, and O. Castillo, "An improved method for edge detection based on interval type-2 fuzzy logic," Expert Syst. Appl., vol. 37, no. 12, pp. 8527–8535, Dec. 2010.

[16] A. Bel, C. Wagner, and H. Hagras, "Multiobjective optimization and comparison of nonsingleton type-1 and singleton interval type-2 fuzzy logic systems," IEEE Trans. Fuzzy Syst., vol. 21, no. 3, pp. 459–476, Jun. 2013.

[17] P. Melin, C. I. Gonzalez, J. R. Castro, O. Mendoza, and O. Castillo, "Edge-Detection Method for Image Processing Based on Generalized Type-2 Fuzzy Logic," IEEE Transactions on Fuzzy Systems, vol. 22, no. 6, Dec 2014.

[18] F. Liu, "An efficient centroid type-reduction strategy for general type-2 fuzzy logic system," Inf. Sci., vol. 178, no. 9, pp. 2224–2236, May 2008.