

Optimized Adaptive Thresholding based Edge Detection Method for MRI Brain Images

S. Jansi
Research Scholar

P. Subashini
Phd, Associate Professor

Avinashilingam Institute of Home Science and Higher Education for Women
Bharathi Park Road, Coimbatore, India.

ABSTRACT

Edge detection is one of the fundamental tool in image processing, machine vision and computer vision, which aim at identifying points in a digital image. It is an important tool for medical image segmentation and 3D reconstruction. Generally, edge has detected according to some early brought forward algorithms such as gradient-based algorithm and template-based algorithm, but they are not so good for noisy medical image edge detection. In order to overcome this problem, adaptive threshold using ACO has proposed. Ant colony optimization technique is used for computing an optimal threshold value used by adaptive threshold for edge detection. The various edge detection algorithms are compared with the proposed algorithm and their performance are evaluated using the evaluation metrics. From the experimental results, the proposed algorithm was better than the adaptive threshold method.

Keywords

Medical image processing, Ant Colony Optimization, Edge detection, Adaptive threshold, and MRI images.

1. INTRODUCTION

In computer vision, edge detection is a fundamental task that needs to point out the true edges to get the best results. That is why it is important to choose edge detectors that fit best to the application. Detection of edge is a fundamental step for most computer vision applications such as MRI feature extraction and remote sensing [1]. The image edge detection refers to extraction of the edges in a digital image. It is a process whose aim is to find points in an image where discontinuities or sharp changes in intensity occur. This process is crucial to understanding the content of an image and has its applications in image analysis and machine vision. It is usually applied in initial stages of computer vision applications. The edge detection aims to localize the boundaries of objects in an image and is a basis for many image analysis and machine vision applications. Conventional approaches to edge detection are computationally expensive because each set of operations has conducted for each pixel. In conventional approaches, the computation time quickly increases with the size of the image. An ACO-based approach has the potential of overcoming the limitations of conventional methods [2].

Ant colony optimization (ACO) takes inspiration from the foraging behavior of some ant species. These ants deposit pheromone on the ground to mark some favorable path that should be followed by other members of the colony. Ant colony optimization technique exploits a similar mechanism for solving optimization problems. ACO has been applied in a

variety of problem domains such as graph coloring, protein folding and the traveling salesman problem [3].

2. EDGE DETECTION METHODS

Edge detection is a fundamental tool used in most image processing applications to get information from images as a precursor step to feature extraction and object segmentation. This process detects boundaries between objects and the background in the image at which the image brightness changes sharply or more formally has discontinuities [4]. The image containing these boundaries is known as edge map. In this paper, the various edge detection techniques are applied for obtaining the optimized edges as follows:

- Canny edge detector
- Sobel edge detector
- Prewitt edge detector
- Laplacian of Gaussian
- Robert edge detector
- Adaptive threshold
- Proposed System

2.1 Canny Edge Detector

The Canny edge detection operator was developed by John F. Canny in 1986 and uses a multi-stage algorithm to detect a wide range of edges in images. An optimal edge detector is based on the following three criteria [5]:

Good detection: The algorithm should mark as many real edges in the image as possible.

Good localization: Marked edges should be close as possible as to the edge in the real scene.

Minimal response: A given edge in the image should only be marked once, and where possible, image noises should not create false edges. Based on these criteria, the canny edge detection process included the following stages:

1. Noise removal: The canny edge detector smoothes the image to eliminate noise.
2. Differentiation: It finds the image gradient in order to highlight regions with high spatial derivatives.
3. Non-maximum suppression: The algorithm tracks along these already highlighted regions and suppresses any pixel that is not at the maximum.

2.2 Sobel Edge Detector

The Sobel operator is particularly used for edge detection algorithms. Technically, it is a discrete differentiation operator, computing an image intensity function. At each point in the image, the result of the Sobel operator is either the

corresponding opposite of the gradient vector or the norm of this vector. The Sobel operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations [6]. On the other hand, the opposite of the gradient approximation that it produces is relatively crude, in particular for high frequency variations in the image.

The sobel edge operator calculate the image intensity at each point, giving the direction of the largest possible change from light to dark and the rate of change in that direction. Mathematically, the operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of horizontal and vertical changes of the derivatives. The sobel edge detection masks look for edges in both the horizontal and vertical directions and then combine this information into a single metric [1].The masks are as follows:

$$Row\ Mask = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (1)$$

$$Column\ Mask = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2)$$

The row mask and the column mask are convolved with the image. At each pixel location there are two numbers: s_1 corresponding to the result from the row mask, and s_2 from the column mask. These numbers are used to compute the edge magnitude which is equal to

$$\sqrt{S_1^2 + S_2^2} \quad (3)$$

2.3 Prewitt Edge Detector

The Prewitt operator is used for edge detection algorithms. This operator does not place any pixels which are closer to the center of the mask [7]. Technically, it is a discrete differentiation operator, computing an image intensity function. At each pixel in the image, the result of the Prewitt operator shows the corresponding gradient vector or the norm of this vector. The prewitt operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and in terms of computations. On the other hand, the gradient operator is relatively crude for high frequency variations in the image [8]. Prewitt operator is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images.

In simple terms, the operator calculates the gradient vector of the image intensity at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction. Mathematically, the operator uses two 3×3 kernels which are convolved with the original image to calculate approximations horizontal and vertical changes of the derivatives. Here A as the source image, the row mask and the column mask are two images which at each point contain the

horizontal and vertical derivative approximations, the latter are computed as:

$$RowMask = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * A \quad (4)$$

$$Column\ Mask = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * A \quad (5)$$

where $*$ here denotes the 2-dimensional convolution operation. The *row mask* is here defined as increasing in the "right"-direction, and the *column mask* is defined as increasing in the "down"-direction. At each pixel in the image, the result of the gradient magnitude, using the equation:

$$G = \sqrt{G_1^2 + G_2^2} \quad (6)$$

2.4 Laplacian of Gaussian

The Laplacian of Gaussian was invented by Marr and Hildreth (1980) who combined Gaussian filtering with the Laplacian. This algorithm is not used frequently in machine vision [9]. The Laplacian of an image highlighted the regions of rapid intensity change and is used for edge detection. Often, the Laplacian of an image is applied to an image with Gaussian smoothing filter in order to reduce its sensitivity to noise. This LOG operator normally takes a single gray level image as input and produces another gray level image as output. The Laplacian $L(x, y)$ of an image with pixel intensity values $I(x, y)$ is given by [10]:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (7)$$

Since the input image normally takes a set of discrete pixels to find the output as a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian. The commonly used 3×3 small kernels are shown in equation.

$$X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (8)$$

$$Y = \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix} \quad (9)$$

Because these kernels are approximating a second derivative measurement on the image, they are very sensitive to noise [11]. To counter this, the image is often Gaussian smoothed before applying the Laplacian filter [12]. This step reduces the high frequency noise components prior to the differentiation step.

2.5 Robert Cross Operator

The Roberts edge detection technique is the most basic of all the techniques discussed. It performs a simple, quick to compute, gradient measurement on an image. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point [13]. The operator consists of a pair of 2×2 convolution kernels as shown in equation. Extension to the higher image dimensions is not possible in this edge detection technique.

$$X = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (10)$$

$$Y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (11)$$

The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each matrix X, Y [14]. These can then be combined together to find the absolute magnitude of the gradient at each point and the matrix of that gradient. The gradient magnitude is given by [15]:

$$|G| = \sqrt{Gx^2 + Gy^2} \quad (12)$$

Although typically, an approximate magnitude is computed using [16]:

$$|G| = |Gx| + |Gy| \quad (13)$$

This equation is much faster to compute.

2.6 Adaptive Threshold

Thresholding is called adaptive threshold when a different type of threshold is used for different regions in the image. This is known as local or dynamic thresholding [17]. Thresholding assumes that the image have pixel values generally different from the background [18]. In the proposed approach, initially edges are extracted using an adaptive variation threshold. The connectivity of the edges is increased and then obtained using ACO. The threshold is used to segment an image by setting all pixels whose intensity values are above a threshold to a foreground value and all the remaining pixels to a background value. Whereas the conventional threshold operator uses a global threshold for all pixels, adaptive threshold changes the threshold dynamically over the image. This more sophisticated version of threshold can accommodate changing lighting conditions in the image, e.g. those occurring as a result of a strong illumination gradient or shadows [19]. The threshold value is calculated for each pixel in the image. If the pixel value

is below the threshold value it is set to the background value, otherwise it assumes the foreground value.

2.7 Proposed System

Adaptive threshold typically takes a gray scale or color image as input and, in the simplest implementation, outputs a binary image representing the edge information. For each pixel in the image, a threshold value is automatically calculated. If the pixel value is below the threshold, it is set to the background value; otherwise it assumes the foreground value. In order to obtain an optimal threshold value for the adaptive threshold method this can be viewed as an optimization problem. Hence, ACO is proposed to use along with adaptive threshold.

3. ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) is a pattern for designing metaheuristic algorithms for combinatorial optimization problems. In computer science and operation research, the Ant Colony Optimization algorithm (ACO) is a probability technique for solving computational problems which could be reduced to finding good paths through graphs. Initially, this algorithm was proposed by Marco Dorigo in 1992. The first algorithm was aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food [20].

The ACO-based image edge detection approach aims to use a number of ants to move on an image for constructing a pheromone matrix, each entry of which represents the edge information at each pixel place of the image. Furthermore, the movements of the ants have steered by the local variation of the image's intensity values [21]. The proposed approach starts from the initialization process, and then runs for N iterations to build the pheromone matrix by iterative performing both the construction process and the update process. Finally, the decision process has performed to find the edge. Each of these processes has presented in detail as follows, respectively.

ACO solution aims to find the optimal solution of the target problem through a guided search (i.e. the movements of a number of ants) over the solution space, by constructing the pheromone information. For better understanding, suppose that K ant have used to find a suitable solution in the space x that includes $M_1 \times M_2$ nodes. ACO could summarize as follows in:

1. Initialize the Position all K ants and the matrix pheromone $\tau^{(0)}$.
2. For the construction-step index $n=1: N$,
For the ant index [22] $k=1: K$,
Consecutively move k -th ant for L spaces based on a probabilistic transition matrix $p^{(n)}$ (with a size of $M_1 M_2 \times M_1 M_2$).
Update the matrix pheromone $\tau^{(0)}$.
3. Make the solution decision according to the last pheromone matrix $\tau^{(N)}$.

There are two fundamental issues in the above mentioned ACO stages; that is, the establishment of probabilistic transition matrix $p^{(n)}$ and the update pheromone matrix $\tau^{(n)}$, each of which has shown in detail. Normally, at the n -th construction-

step of ACO, k -th ant moves from the likely node i to node j based on a probabilistic action rule.

$$P_{i,j}^{(n)} = \frac{(\tau_{i,j}^{(n-1)})^\alpha (n_{i,j})^\beta}{\sum_{j \in \Omega_i} (\tau_{i,j}^{(n-1)})^\alpha (n_{i,j})^\beta}, \text{ if } j \in \Omega_i \quad (14)$$

where $\tau_{i,j}^{(n-1)}$ is the pheromone information value of the connective arc the node i to the node j . Ω_i is the neighborhood nodes for ant a_k given on node i . Fixed values α and β shown influence of pheromone information and heuristic information. $n_{i,j}$ show the heuristic information from node i to node j that is constant to be same for each construction-step. Then, the pheromone matrix in ACO algorithm needs to be updated twice [23]. The first update has performed after movement of each ant in each construction-step. To be more specific, after the move of k -th ant in the first n -th construction-step, pheromone matrix has updated.

$$\tau_{i,j}^{n-1} = \begin{cases} 1 - \rho \cdot \tau_{i,j}^{n-1} + \rho \cdot \Delta_{i,j}^{(k)} \\ \tau_{i,j}^{(n-1)} \end{cases} \quad (15)$$

which ρ is evaporation rate depends on the value of user choice. The second update has performed after movement of all K ants in each construction-step; and the pheromone matrix has updated.

$$\tau^{(n)} = (1 - \psi) \tau^{(n-1)} + \psi \cdot \tau^{(0)} \quad (16)$$

which ψ has the pheromone decay coefficient [24]. Note that ant colony system uses two operations for updating the pheromone matrix (i.e. equations (15) and (16)), while the ant colony system in only uses an operation (i.e. the equation (16)).

The proposed approach starts from the initialization process has performed at the beginning. In this step, it runs for N iterations to construct the pheromone matrix by iteratively performing both the construction process and the update process. Finally, the decision process is performed to determine the edge. Each of these processes is presented in detail as follows, respectively.

3.1 Initialization Process

For initializing the process, totally K ants are randomly assigned on the image I with size of $M_1 \times M_2$. Each pixel must consider as a node [22]. The first value of each part of pheromone matrix $\tau^{(0)}$ sets a fixed value τ_{init} .

3.2 Construction Process

At the n -th construction-step, one ant have chosen randomly from K mentioned ants and this ant for L Movement-steps moving on the image consecutively. This ant moves from node (l, m) to the neighboring node (i, j) based on chance of transition that defined as follows:

$$P_{(l,m),(i,j)}^{(n)} = \frac{(\tau_{i,j}^{(n-1)})^\alpha (\eta_{i,j})^\beta}{\sum_{(i,j) \in \Omega_{(l,m)}} (\tau_{i,j}^{(n-1)})^\alpha (\eta_{i,j})^\beta} \quad (17)$$

where $\tau_{i,j}^{(n-1)}$ is the pheromone value of node (i,j) , $\Omega_{(l,m)}$ is the neighborhood nodes of node (l,m) and $\eta_{i,j}$ shows the heuristic information in the node (i,j) (16). The fixed values are α and β shown influence of pheromone matrix and the heuristic matrix. There are two essential issues in the construction process. First issue which determination of the heuristic information $\eta_{i,j}$ in equation (17).

3.3 Update Process

The update process performs two operations for updating the pheromone matrix. The first operation has performed after movement of each ant in each construction-step. Each part of pheromone matrix have updated by using the following equation (18):

$$\tau_{i,j}^{(n-1)} = \begin{cases} (1 - \rho) \cdot \tau_{i,j}^{(n-1)} + \rho \cdot \Delta_{i,j}^{(k)} \\ \tau_{i,j}^{(n-1)} \end{cases} \quad (18)$$

where ρ have defined in equation (15), $\Delta_{i,j}^{(k)}$ which determined by the heuristic matrix, that is, $\Delta_{i,j}^{(k)} = \eta_{i,j}$. The second update operation has performed after movement of the entire ant in each construction-step as follows:

$$\tau^{(n)} = (1 - \psi) \cdot \tau^{(n-1)} + \psi \cdot \tau^{(0)} \quad (19)$$

which ψ have defined in equation (16).

3.4 Decision Process

Finally, the pheromone matrix is used to classify each pixel either as an edge or a non-edge. This decision is made by applying a threshold on the last pheromone matrix $\tau^{(N)}$. The threshold value computed is the ideal value [25]. This threshold value has compared with each pixel value. If the pixel value is below the threshold it sets the background value, otherwise it assumes the foreground value.

4. RESULTS AND DISCUSSIONS

This section presents the experimental results of the proposed techniques of edge detectors such as Canny, Sobel, Prewitt, Laplacian of Gaussian, Roberts cross operator, Adaptive threshold method, and Adaptive threshold using ACO. One major reason to design an objective quantitative evaluation method [18] is to compare the results from fundamentally different edge detection techniques. So here the performances of the algorithms are measured using the metrics such as SNR, RMSE, and MAE.

SNR has defined as ration of average signal power to average noise power and for an $M \times N$ image is

$$SNR(db) = 10 \log_{10} \left[\frac{\sum_{i,j} x(i,j)^2}{\sum_{i,j} (x(i,j) - y(i,j))^2} \right] \quad (20)$$

for $0 \leq i \leq M - 1$ and $0 \leq j \leq N - 1$. Let $x(i, j)$ represent the value of the i^{th} row and j^{th} column pixel in the original image x and let $y(i, j)$ represent the value of the corresponding pixel in the output image y . The local error is $e(i, j) = x(i, j) - y(i, j)$.

RMSE is often used measure of the differences between values predicted by a model or an estimator and the values actually observed. It is a good measure of accuracy. These individual differences are also called residuals.

$$RMSE(\hat{\theta}) = \sqrt{MSE(\hat{\theta})} = \sqrt{E\left(\left(\hat{\theta} - \theta\right)^2\right)} \quad (21)$$

The RMSE of an estimator $\hat{\theta}$ with respect to the estimated value θ has defined as the square root of the mean square error.

MAE measures the average size of the errors in a set of forecasts, without considering their direction. It measures accuracy for continuous variables.

$$E_i = \frac{1}{n} \sum_{j=1}^n |P_{(ij)} - T_j| \quad (22)$$

The MAE is the average over the verification sample of the absolute values of the differences between forecast and the corresponding observation. The MAE is a linear score which

means that all the differences have weighted equally in the average.

Figure1 shows the resultant images obtained by various edge detection methods. Table1 shows the performance metrics obtained for the various edge detection methods. From table1, it has been observed that the SNR value for adaptive threshold method is higher when compared to other methods. Hence, the adaptive threshold method would be the best method to reduce the redundant edge pixels, and to detect the correct edges. The MAE and RMSE value is higher than the other methods. So, to reduce the error value of adaptive threshold method the ACO is used.

Figure2 shows the resultant image obtained by adaptive threshold using ACO. Table2 shows the performance metrics obtained for adaptive threshold using ACO. From table2, it has been observed that the performance of the proposed adaptive threshold using ACO is better than the adaptive threshold method.

5. CONCLUSION

This paper has presented comparison of various edge detection algorithms by measuring their performance using evaluation metrics. Here, the adaptive threshold using ACO is proposed for edge detection. In the proposed system, the adaptive threshold method is used for edge detection and the ant colony optimization method is used for computing an optimal threshold value. From the experimental results, it is inferred that the proposed method shows good result when compared to other methods. The proposed ACO method had the better performance with SNR of 24.618 which is higher compared to other methods.

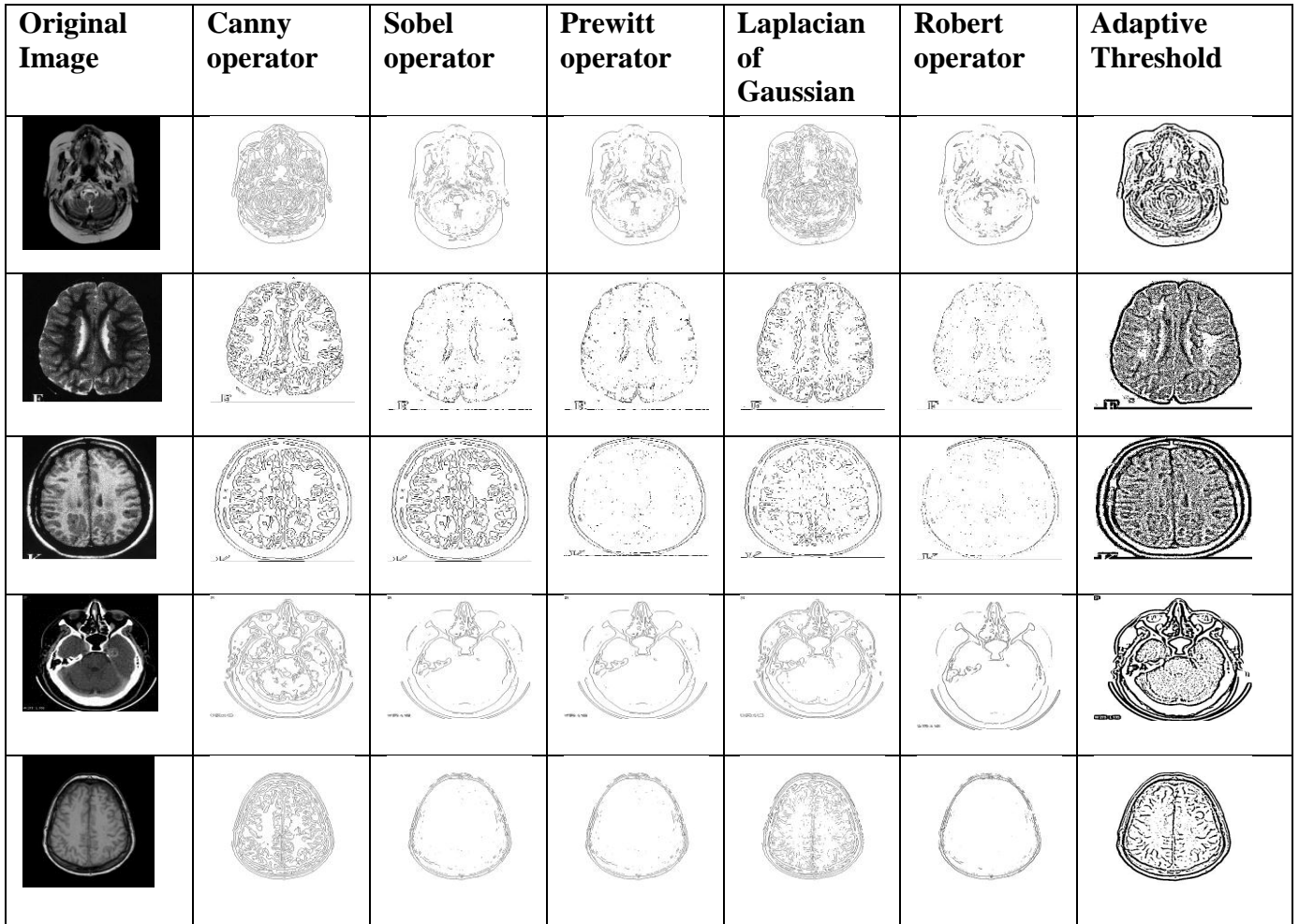


Figure1: Result of Edge Detection Methods

Table1: Performance Metrics for various edge detection methods

Edge Detection	MAE	RMSE	SNR
Canny Operator	6.72348	7.51066	23.144
Sobel Operator	2.82302	5.80772	23.032
Prewitt Operator	2.80244	5.80852	23.03
Laplacian of Gaussian	5.73694	6.41698	23.184
Robert operator	2.4477	5.5254	23.024
Adaptive Threshold	7.27906	11.69238	24.184

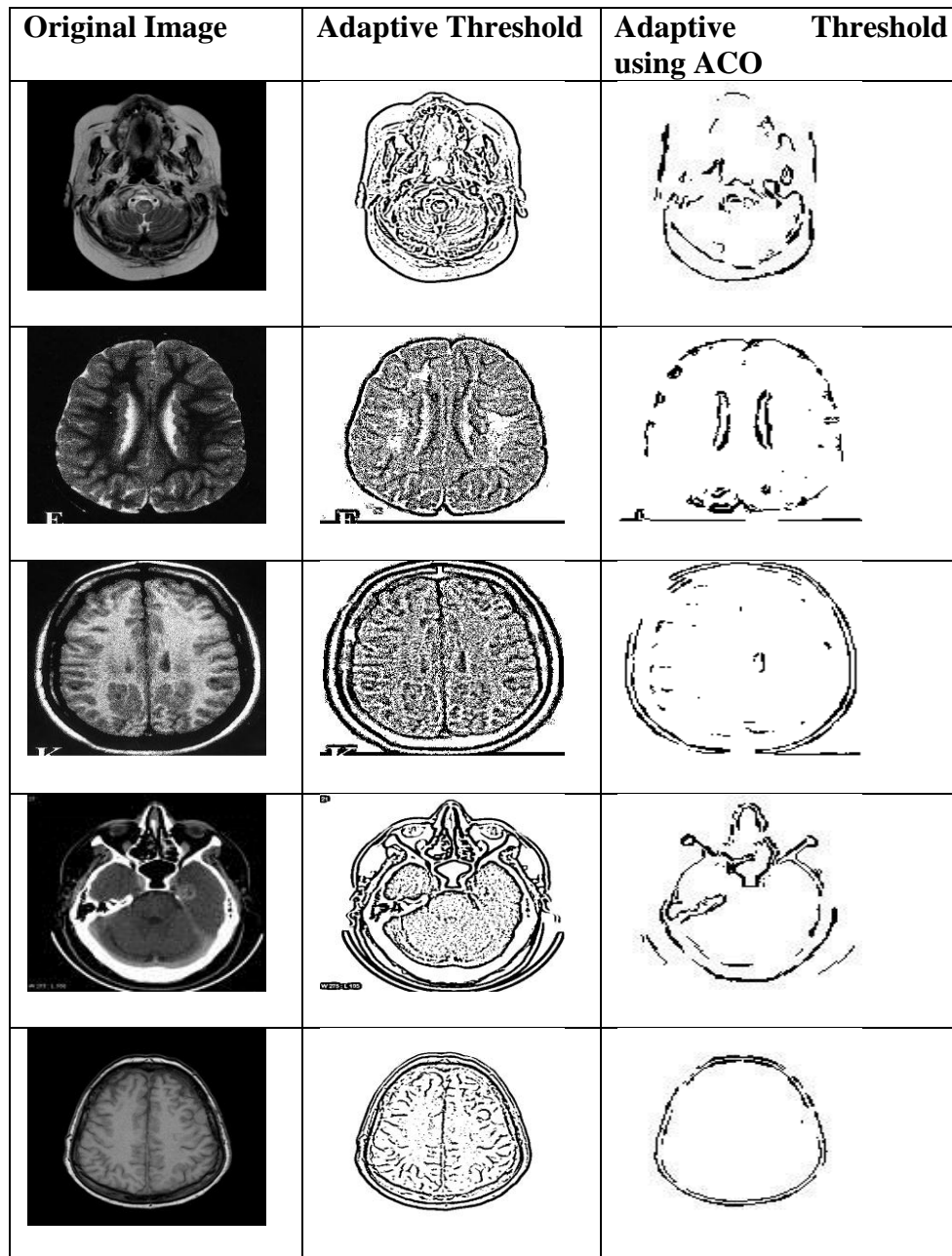


Figure2: Adaptive Threshold Method using ACO

Table1: Performance Metrics for Adaptive threshold using ACO

Edge Detection	MAE	RMSE	SNR
Adaptive Threshold	7.27906	11.69238	24.184
Adaptive Threshold using ACO	5.22508	7.75364	24.618

6. REFERENCES

- [1] Gholamali Rezai-Rad, Majid Aghababaie, "Comparison of SUSAN and Sobel Edge Detection in MRI Images for Feature Extraction", *IEEE transaction on Information and Communication Technologies*, volume1, October 2006, Pages 1103 - 1107.
- [2] Anna Veronica Baterina, Carlos Oppus, "Image Edge Detection using ACO", *WSEAS transaction on Signal Processing*, Volume6, Issue2, 2010, Pages: 2-5.
- [3] Rutger Claes, Tom Holvoet, "Ant Colony Optimization applied to Route Planning using Link Travel predictions", *IEEE International Parallel & Distributed Processing Symposium*, May 2011, Pages 353-360.
- [4] Taghizadeh, M.; Hajipoor, M., "Hybrid Algorithm for Segmentation of MRI Images Based on Edge Detection", *IEEE transactions on Soft Computing and Pattern Recognition*, Dec2011, Pages 107 - 111.
- [5] Moslem Taghizadeh, Mohammad Reza Mahzoun, "Bidirectional Image Thresholding algorithm using combined Edge Detection and P-Tile algorithms", *The Journal of Mathematics and Computer Science*, Volume2, 2011, Pages 255-261.
- [6] Guennadi (Henry) Levkine, "Prewitt, Sobel and Scharr gradient 5x5 convolution matrices", *Image Processing Articles*, Second draft, June 2012.
- [7] Harelimana Joyeuse, et.al, "Implementation of Edge Detection for a Digital Image", *KIST-AO Electronics and telecommunication engineering* 2008.
- [8] Vishwadeep Garg et.al., "Image Texture Enhancement through an Improved Grunwald-Letnikov Fractional Differential Mask", *International Journal of Advanced Computer Science and Applications*, Vol.3, No.3, Dec2011.
- [9] Sharifi, M.; Fathy, M.; Mahmoudi, M.T., "A Classified and Comparative Study of Edge Detection Algorithms", *IEEE, International conference on Information Technology: Coding and Computing*, 2002, Pages 117 - 120.
- [10] Raman Maini, Himanshu Aggarwal, "Study and Comparison of various image edge detection techniques", *International Journal of Image Processing (IJIP)*, Volume (3) : Issue (1), 2009.
- [11] Mamta Juneja, Parvinder Singh Sandhu, "Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain", *International Journal of Computer Theory and Engineering*, Vol1, Dec 2009.
- [12] <http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm>
- [13] Senthil Kumar N. Rajesh R., "Edge Detection techniques for Image Segmentation - A Survey of Soft Computing Approaches", *International Journal of Recent Trends in Engineering*, Vol.1, May 2009.
- [14] Yang Chao, "A comparison of medical image analysis algorithms for edge detection", *Bachelor Thesis*, June2010.
- [15] Salem Saleh Al-amri, Kalyankar and Khamitkar S.D., "Image Segmentation by using edge detection", *International Journal on Computer Science and Engineering*, Volume2, 2010, Pages: 804-807.
- [16] [www.cee.hw.ac.uk/roberts edge detector](http://www.cee.hw.ac.uk/roberts%20edge%20detector).
- [17] Tarek A. Mahmoud, Stephen Marshall, "Medical Image Enhancement Using Threshold Decomposition Driven Adaptive Morphological Filter", *16th European Signal Processing Conference (EUSIPCO 2008), EURASIP*, 2008.
- [18] P.Subashini, M.Krishnaveni, Suresh Kumar Thakur, "Quantitative Performance Evaluation on Segmentation Methods for SAR ship images", *Proceedings of the Third Annual ACM Bangalore Conference*, 2010.
- [19] Shapiro, Linda G. & Stockman, George C. (2002). "Computer Vision". Prentice Hall. ISBN 0-13-030796-3
- [20] <http://homepages.inf.ed.ac.uk/rbf/HIPR2/roberts.htm>
- [21] SenthilKumaran and Rajesh, "Brain Image Segmentation", *International Journal of wisdom based computing*, vol1 (3), Dec2011.
- [22] Jian Zhang, Kun He, et.al, "An Ant Colony Optimization Algorithm for Image Edge Detection", *International Conference on Artificial Intelligence and Computational Intelligence, AICI (2010)*.
- [23] Javad Rahebi et al., "Biomedical Image Edge Detection using an ACO Based on Artificial Neural Networks", *International Journal of Engineering Science and Technology (IJEST)*, 2011.
- [24] Vittorio Maniezzo, Luca Maria Gambardella, "Ant Colony Optimization", 2004.
- [25] Manisha Kaushal, Arjan Singh, Baljit Singh, "Adaptive Thresholding for Edge Detection in Gray Scale images", *International Journal of Engineering Science and Technology*, Volume2(6), 2010, Pages2077-2082.