Academic Journal of Nawroz University (AJNU), Vol.11, No.2, 2022

https://doi.org/10.25007/ajnu.v11n2a1320

Hybridizing Ant Colony Optimization Algorithm for Optimizing Edge-Detector Techniques

Saman M. Almufti
Department of Computer Science, Nawroz University, Duhok Kurdistan Region of Iraq

ABSTRACT

Ant colony optimization is a swarm intelligent algorithm that mimics the ant behaviors to optimize solutions for hard optimization problems. Over years Ant-based algorithms have been used in solving different problems including: Traveling Salesman Problem (TSP), Wireless Sensors Network (WSN), Benchmark Problem, and it has been used in various image processing applications. In the image processing fields various techniques have been used to detect edges in a digital image such as Canny and Sobel edge detectors. This Study, proposed a hybridized Ant Colony Optimization algorithm for optimizing the edge detector quality. The proposed method initializes its attribute matrix and the information at each pixel routed by ants on the input image. Experimental results show the results of the proposed algorithm and compare the results with the original built-in MATLAB edge detection method called Canny and the results of basic Aco edge detector. All three algorithms tested in different images and the MSE and PNSR are calculated before and after applying Gaussian noise. Based on the Experimental results obtained by the three used methods (Canny Edge Detector, Ant Colony Optimization, and Hybrid Aco-Canny), the proposed Hybrid ACO-CANNY methods was the best method for detecting edges.

KEYWORDS: Edge Detection, Ant Colony Optimization, ACO, Canny Edge Detection, MSE, PSNR.

1. Introduction

In the image processing and pattern recognition field Image Edge Detection (IED) represents one of the most-common strategies used for the detection of important and meaningful information in gray-scale images. And it represents the most important step in all image segmentation, object detection, and feature extraction applications. Traditional image edge detection algorithms such as Sobel, Erosion Residue, Prewitt, Dilation Residue, and Canny operators typically use linear filtering operations for removing noise from the image (Rana & Dalai, 2014). With the increasing number of patients with various illnesses, that cannot be identified and diagnosed without using imaging modalities such as MRI and CT Scan (Ihsan, Almufti, & Marqas, 2020), various applications have been developed to enhance the Edge Detections techniques (Rana & Dalai, 2014).

Swarm Intelligence stands for computational Metaheuristic based algorithms that use the intelligent collective behavior in decentralized collections of simple individuals' agents interacting with each other to achieve a simple task in the environment around themselves (Almufti S. M., 2017). In computer science, various algorithms inspired the behavior of living animals and insects for solving different problems of different fields of studies.

Ant Colony Optimization (ACO) which is a swarm-based algorithm that inspires the behaviors of social ants, has been used in various image processing techniques and applications.

In this paper ACO is hybridized with Canny edge detector and have been developed to extract Edges and important features in different types of Images, and the results were compared with traditional image edge detection techniques.

The proposed Hybrid ACO-Canny algorithms results shows stretches reasonable better details and thinner edges in both original (clean) and noisy images, than using other edge detector techniques as shown in table1 and table 2.

EDGE DETECTION

In image processing, The classifying and discovering sharp breaks in an image is called edge detection. The discontinuities are immediate variations in pixels concentration (brightness or sharpness) that distinguish the edges of objects in a scene. This process is critical for understanding the content of an image, it meaningly decreases the amount of data of a scene in an image to be processed, thus far it retains critical information that regards the shape of objects in the scene (Kanugo & Mekala, 2016). In computer science, IDE has various applications in image processing, image analyzing, computer vision, Remote Monitoring, and especially in Medical Image enhancements and reconstruction which requires finding the continuity of the edges. In image processing fields, there are numerous edge detection techniques available for the image processing of digital images, such as Sobel, Canny, Prewitt, and Roberts. Where each of those techniques is developed to be sensitive to a specific types of edges (Acharjya, Das, & ρ, 2012).

1.1 Traditional edge detectors

Because of the necessity of Edge Detection Techniques, many techniques have been designed for detecting edges in digital images, such as Canny Edge Detector, Sobel Edge Detector, Roberts Cross Edge Detector, Compass Edge Detector, Zero Crossing Detector, Line Detector, and various other techniques. And are illustrated as a standard for edge detection in MATLAB (Rana & Dalai, 2014).

1.2 Canny Edge Detector

The Canny edge detector is a widely used standard's edge detection technique. It was initial designed in 1983 by John Canny (Canny, 1986). Until nowadays Canny performs better results than many of the newer developed algorithms for extracting edges in digital images. as an initial step, Canny Edge Detection separates noise from the image before finding edges of the image.

Generally, Canny edge detector procedures shown below (Ding & Goshtasby, 2001).

1. Smoothing images with appropriate Gaussian filter for reducing desired image details.
2. Find the gradient magnitude and gradient direction for all pixels.
3. In case of finding a pixel with larger gradient magnitude than the gradient magnitude of its two neighbors, sign the pixel with longer gradient magnitude as an Image edge. Otherwise, sign pixels as backgrounds normal pixel.
4. Using hysteresis thresholding for removing weak edges.

In MATLAB can be easily extracted from an image by Canny edge simply by using the 'Edge' function with attribute 'canny' as shown in (1).

Image_edge = edge(original_image , 'canny') (1)

2. GAUSSIAN NOISE

It is an ideal type of white-noise, it caused by random variations in a signal. Usually, gaussian noise is normally distributed such as for β image, γ is a Gaussian-noise mask, so the gaussian noisy image $\beta' = \beta + \gamma$. Gaussian noise mean and variance parameters default values take mean parameter $\mu = 0$ and standard deviation parameter $\alpha = 0.01$ (Boyat & Joshi, 2015).

Gaussian noise generally applies on gray values of images, therefore it is designed and characterized by its Probability Density Functions (PDF) or normalizing histograms according to gray values. And can be calculated by Eq.(2) (Boyat & Joshi, 2015).

$$P(g) = \sqrt{\frac{1}{2\pi\theta^2}} e^{-\frac{(g-\mu)^2}{2\theta^2}} \qquad (2)$$

Where g represents the gray value, θ represents standard deviation and μ represents the mean. Gaussian noise can be used simply in MATLAB by using the 'imnoise' MATLAB function with attribute 'gaussian' as shown below in (3).

noisy_image = imnoise(original_image , 'gaussian') (3)

In this paper the algorithms are tested to detect edges in a sample o imaged before and after adding 10% gaussian white noise.

3. SWARM INTELLIGENT ALGORITHM

In artificial intelligence, Swarm Intelligence (SI) represents a collection of algorithms that are concerned with developing intelligent multi-agent systems that interactively cooperate to achieve a certain task. SI is defined by Marko as "The emergent collective intelligence of groups of simple agents" (Li, 2010). Generally, Swarm-based algorithms are stimulated by behaviors of social living beings (insects, animals, and bacteria) in their social groups, such as ants colony, cats wilds, bats colony, bees swarm, Lions pride, and fishes school (Almufti S. M., 2021). The most remarkable features of swarm systems are Self-organization and decentralized control that naturally leads to local interaction between multi-agents in a system for achieving a required mission. In computer science, various algorithms are designed by inspiring real collective behaviors in nature (Almufti S. M., 2019). some well-known SI algorithms are described in table 1.

TABLE 1

well-known Swarm Intelligent Algorithms

#	Title	ABB.	INSPIRATION	REF.
1.	Ant Colony Optimization	ACO	Ant Behavior	(Dorigo, 2004)
2.	Particle Swarm Optimization	PSO	Bird flocking Behavior	(Eberhart & Kennedy, 1995)
3.	Artificial Bee Colony	ABC	Bee Behavior	(Karaboga, 2005)
4.	Glowworm Swarm Optimization	GWO	Glowworm Behavior	(Krishnanand & Ghose, 2009)
5.	Bat Algorithm	BA	Bat Behavior	(Yang, 2010)
6.	Cat Swarm Optimization	CWO	Cat Behavior	(Bahrami, Haddad, & Chu, 2017)
7.	Grey Wolf Optimizer	GWO	Gray wolf Behavior	(Mirjalili, Mirjalili, & Lewis, 2014)
8.	Social Spider Optimization	SSO	Spider Behavior	(Almufti S. M., 2021)
9.	Elephant Herding Optimization	EHO	Elephant Behavior	(Saeed, Almufti, & Marqas, 2019)
10.	Lion Optimization Algorithm	LOA	Loin Behavior	(Yazdani & Jolai, 2016)

3.1 Ant Colony Optimization (ACO)

ACO represents a swarm-based meta-heuristics algorithm that inspires the behaviors of ants in discovering the best bath to a food source (Dorigo, 2004). real ants indirectly communicate by depositing an aromatic chemical compound that can be detected by other ants in the swarm called pheromone trails. In ACO, each ant represents an agent in the algorithm. All the agents are required to lay down a pheromone trail on their way searching for the best near-optimal path, in case of significantly increasing pheromone deposit in a specific path all ants of the colony use the chosen (Almufti S. M., 2017).

In the image processing, ACO agents artificial ants uses pheromone-trail matrix in the process of updating their route over image-edges. Ant Colony Optimization iteratively discovery the optimal or near-optimal edge by movements of ants and the process of pheromone-trail depositing and evaporating over the digital-image pixels. The increases in pheromone trail on a pixel increase the probability for the ant's to move through it in transition matrix (Lu & Chen, 2008).

ACO follows the following steps for detecting the image edges

Step 1.Converting image to gray

Step 2.Randomly initialize Ant position and pheromone-trail matrix.

Step 3. Choosing next pixel by probability function

Step 4. Update Pheromone matrix

Step 5. Pheromone evaporation

3.2 Proposed Hybrid ACO-Canny Algorithm

In this paper, Ant Colony Optimization algorithm has been Hybridized by canny for detecting edges in digital images, the pheromone trail matrix is initialized based on the results of canny edge detection algorithm, this step effects on the process of choosing next pixel as the ant moves through the image pixels. The proposed algorithm follows the steps bellow:

1. Converting input image to gray-scale.
2. Randomly Position Ants in Image-Matrix.

   The algorithms starts with random allocations of ants on image pixels.
3. End point detection process

   After locating ants the algorithm checks for end points on the image matrix. End points are the points of immediate changes in pixel concentration brightness or sharpness. Figure (1) shows the end points on image matrix.

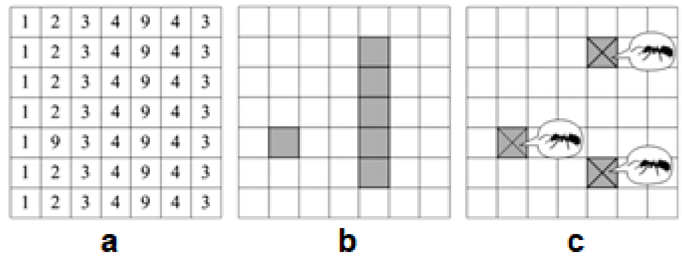


Fig.1. (a) image matrix, (b) end points, and (c) locating ants

4. Applying Canny edge detection to determine the possibility that end points is edge and initializing the pheromone trail matrix.
5. Process of choosing next pixel by ants
6. Every Ant chose the next pixel upon a probability-based function based on heuristics and pheromone-trail values, called random proportional rule. Eq.(4), shows the probability of Ant (A) that locate at the pixel (i,j) to move to its neighboring pixel (i_{new}, j_{new}) in the image matrix.

$$P^{A}_{(i,j)(i_{new},j_{new})} = \frac{\left[\tau_{(i_{new},j_{new})}\right]^{\alpha} \cdot \left[\eta_{(i_{new},j_{new})}\right]^{\beta}}{\sum_{(i_{new},j_{new}) \in \mathcal{N}^{A}_{(i,j)}} \left[\tau_{(i_{new},j_{new})}\right]^{\alpha} \cdot \left[\eta_{(i_{new},j_{new})}\right]^{\beta}} \quad (4)$$

$$\eta_{(i_{new},j_{new})} = \frac{1}{z} v_c\left(\gamma_{(i_{new},j_{new})}\right) \quad (5)$$

$$z = \sum_{(i,j)}^{M1} * \sum_{(i_{new},j_{new})}^{M2} * v_c\left(\gamma_{(i_{new},j_{new})}\right) \quad (6)$$

$$v_c\left(\gamma_{(i_{new},j_{new})}\right) = f\left(\left|\gamma_{(i-1,j-1)} - \gamma_{(i+1,j+1)}\right| + \left|\gamma_{(i-1,j)} - \gamma_{(i+1,j)}\right| + \left|\gamma_{(i+1,j-1)} - \gamma_{(i-1,j+1)}\right| + \left|\gamma_{(i,j-1)} - \gamma_{(i,j+1)}\right|\right) \quad (7)$$

$$f(x) = \begin{cases} \sin\left(\frac{\pi x}{2\delta}\right) & , \text{ where } 0 \le x \le \delta \\ 0 & , \text{ else} \end{cases} \quad (8)$$

In Eq.(4) the $\boldsymbol{\eta}_{(i_{new},j_{new})}$ is a present heuristic rate, and $\tau_{(i_{new},j_{new})}$ represents the pheromone-trail matrix, α is a parameter to control the value of $\tau_{(i_{new},j_{new})}$, β is a parameter to control the value of $\eta_{(i_{new},j_{new})}$, and $(\mathcal{N}^{A}_{(i,j)})$ represents the set of neighborhood-pixels that ant (A) require passed through them.

$\boldsymbol{\eta}_{(i_{new},j_{new})}$ is a present heuristic can be calculated by Eq.(5).

In Eq.(6) Z represents the normalization factor, and $(\gamma_{(i_{new},j_{new})})$ represents the intensity value of the pixel (i_{new}, j_{new}) in $\boldsymbol{\gamma}$ image, the function $v_c(\gamma_{(i_{new},j_{new})})$ which is used in both Eq. (5 and 6) represents a local group function of pixels c, and it depends on the difference in intensity values of c neighborhoods as shown in Fig. (2) can be calculated by Eq.(7). Finally f(x) is calculated by Eq.(8) (Rafsanjani & Varzaneh, 2015).

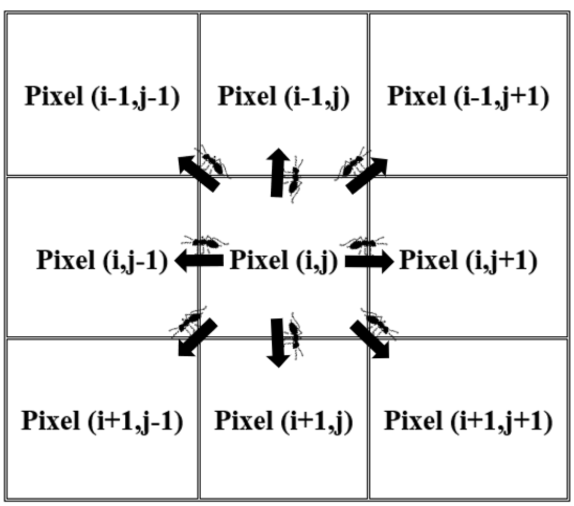


Fig. 2. pixel(i,j) neighborhoods

7. Update pheromone-trail matrix.

All Ants after each move it adds a new Pixel to its tour, and makes a local update on pheromone-trail Matrix values as shown in (9).

$$\tau_{ij}new=\tau_{ij}old+\Delta\tau_{ij} \qquad (9)$$

Figure (3), shows the movements of Ants depending on pheromone-trail matrix.

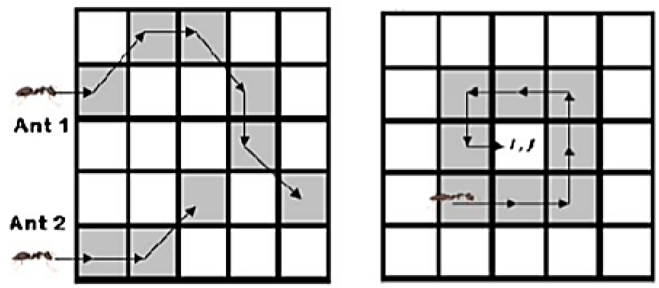


Fig. 3. Ant moves

8. Pheromone Evaporation

Final step for ants is the global pheromone evaporates in the path.

$$\tau_{ij}\boldsymbol{new} = (\mathbf{1} - \boldsymbol{\psi}).\tau_{ij}old \qquad (10)$$

Where (0<ψ<1) is a parameter used for reducing pheromone trail rate in the unvisited image pixels .

4. PERFORMANCE EVALUATIONS

The performance evaluation of edge detections algorithms are accomplished by detection of true edges and comparing the MSE and PSNR in the used image samples before and after adding gaussian noise.

4.1 Mean Square Error (MSE)

It calculates the deviation between the pixel values of reference image and edge detector image (Pawar

& Patil, 2014). A lesser value shows good results.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m}\sum_{j=0}^{n}([ori(i,j) - edg(i,j)])^2 \qquad (9)$$

Where M and N are dimensions pixels of image, ori represents the original image and edg represents the edges image. The lowest mean square error represents best quality image.

4.2 Peak Signal to Noise Ratio (PSNR)

It determines the degree of similarity between reference and edge detected image. A bigger value shows good results (Pawar & Patil, 2014).

$$PSNR = 20log_{10}\left[\frac{[l]^2}{\frac{1}{mn}\sum_{i=0}^{m}\sum_{j=0}^{n}([ori(i,j) - edg(i,j)])^2}\right]$$

Where, L denotes number of gray levels in image.

5. EXPERIMENTAL RESULTS

In this section the experimental results for detecting image edges are shown. In the experiment three different image has been used :

- Lena Image: which is the most-known image in the field of Image Processing, the used image was a RGB image with size 512*512 with extension PNG.
- Nawroz University Logo: which is a logo that contains Victors, Text, and shapes, the image was an RGB image with size 512*512 and extension PNG.
- Personal Image: which is an image that contains different color ranges, the image was an RGB image with size 512*512 and extension BMP.

In this paper, three algorithms have been used to extract the edges in an image

- Canny edge detection: which is a well-known edge detection method that has been a standard used by MATLAB in image edge detection process.
- Ant Colony Optimization (ACO): which a widely used swarm-based algorithm in the field of image processing.
- Hybrid ACO-Canny: this paper hybridize ACO with Canny to detect image edges, and the results are compared with the two other techniques.

The algorithm evaluations proceeded by testing the three techniques in two phases:

Phase 1.Original Images: the original images (Lena Image, Nawroz University Logo, and Personal Image) and the results are compared in table 2.

Phase 2.Noise Images: by adding 10% Gaussian Noise to the original images (Lena Image, Nawroz University Logo, and Personal Image) and the results are compared in table 3.

TABLE 2

Experimental results of (Canny, ACO, and Hybrid ACO-Canny) on Original Image

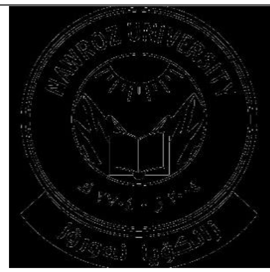
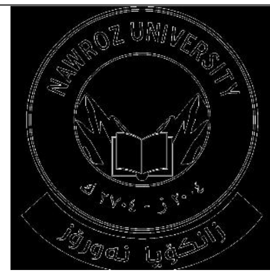
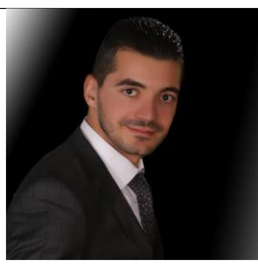
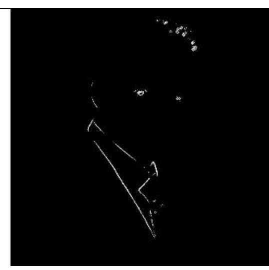
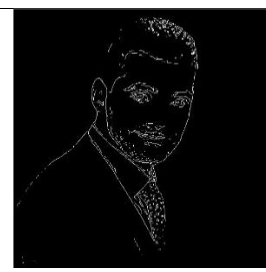
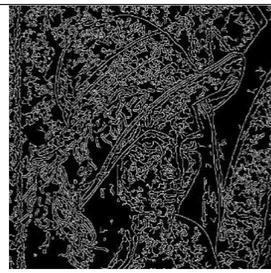
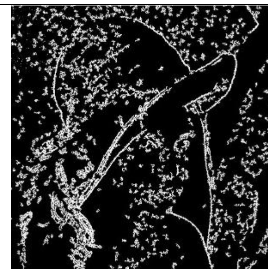
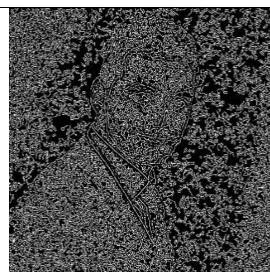
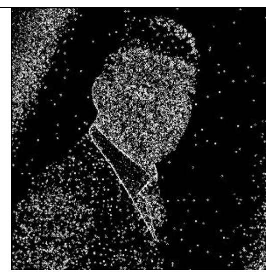
Id	Original image	Canny	ACO	Hybrid ACO-Canny
1				
	PSNR	7.2909	7.3073	7.3307
	MSE	12252.8495	12256.1329	12245.9853
2				
	PSNR	4.5220	4.3656	4.5729
	MSE	23954.9818	23796.6671	23756.9590
3				
	PSNR	13.1911	14.4507	14.6787
	MSE	3118.6497	2334.1884	2333.5171

TABLE 2

Experimental results of (Canny, ACO, and Hybrid ACO-Canny) on Noise Image

Id	Gaussian Noise image	Canny	ACO	Hybrid ACO-Canny

1				
	PSNR	**8.4862**	**8.3511**	**8.7727**
	MSE	**9626.0315**	**9505.3313**	**9214.2769**
2				
	PSNR	**5.3100**	**5.2911**	**5.9962**
	MSE	**19347.6644**	**19229.4930**	**19193.5894**
3				
	PSNR	**8.5248**	**11.2507**	**13.3366**
	MSE	**9132.7847**	**4875.3752**	**3015.9528**

From table 2 and 3, of the resulted edges image,

And from the calculated value of MSE and PSNR it clearly show that MSE values for the proposed hybrid ACO-Canny are much smaller than MSE values of the other used algorithms. Also PSNR values for the proposed hybrid ACO-Canny are higher than PSNR values of the other used algorithms after and before adding noise to the image.

6. CONCLUSION

This paper introduces a hybridized Ant Colony Optimization with Canny Edge Detector method for detection of Edges in digital images. The proposed Hybrid ACO-Canny initialize the Pheromone-Trail matrix bas on result of canny which enhance the result of algorithm.

By considering Table 2 and 3 When comparing the results MSE and PSNR of proposed edge detector (hybrid ACO-Canny edge detector) with other ant-based approaches and canny method. It can be concluded that the hybrid ACO-Canny edge detector stretches reasonable better details and thinner edges in both original (clean) and noisy images.

REFERENCES

Acharjya, P. P., Das, R., & ρ, D. G. (2012). Study and Comparison of Different Edge Detectors for Image Segmentation. *Global Journal of Computer Science and Technology Graphics & Vision, 12*(13), 29-32.

Almufti, S. M. (2017). Using Swarm Intelligence for solving NP-Hard Problems. *Academic Journal of Nawroz University, 6*(3), 46-50. doi:https://doi.org/10.25007/ajnu.v6n3a78

Almufti, S. M. (2019). Historical survey on metaheuristics algorithms. *International Journal of Scientific World*, 1-12. doi:10.14419/ijsw.v7i1.29497

Almufti, S. M. (2021). The novel social spider optimization algorithm: overview, modifications, and applications. *ICONTECH international journal of surveys, engineering, technology, 5*(2), 35-51. doi:10.46291/ICONTECHvol5iss2pp32-51

Bahrami, M., Haddad, O. B., & Chu, X. (2017). Cat Swarm Optimization (CSO) Algorithm. In *Advanced Optimization by Nature-Inspired Algorithms* (pp. 9-18). Springer, Singapore. doi:https://doi.org/10.1007/978-981-10-5221-7_2

Boyat, A. K., & Joshi, K. (2015). A REVIEW PAPER: NOISE MODELS IN DIGITAL IMAGE PROCESSING. *Signal & Image Processing : An International Journal (SIPIJ)*, 63-75. doi:10.5121/sipij.2015.6206

Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 8*(9).

Ding, L., & Goshtasby, A. (2001). On the Canny edge detector. *Pattern Recognition*, 721}725.

Dorigo, M. (2004). *Ant Colony Optimization.* Thomas Stützle. MIT Press. doi:ISBN 0-262-04219-3

Eberhart, R., & Kennedy, J. (1995). A New Optimizer Using Particle Swarm Theory. *Sixth International Symposium on Micro Machine and Human Science* (pp. 39-43). IEEE.

Ihsan, R. R., Almufti, S. M., & Marqas, R. B. (2020). A Median Filter With Evaluating of Temporal Ultrasound Image for Impulse Noise Removal for Kidney Diagnosis. *Journal of Applied Science and Technology Trends*, 71-77. doi:10.38094/jastt1217

Kanugo, S., & Mekala, A. M. (2016). Particle Swarm Optimization based Edge Detection Algorithms for Computer Tomography Images. *Indian Journal of Science and Technology, 9*(37), 1-8. doi:10.17485/ijst/2016/v9i37/102133

Karaboga, D. (2005). AN IDEA BASED ON HONEY BEE SWARM FOR NUMERICAL OPTIMIZATION., (pp. 1-10).

Krishnanand, K., & Ghose, D. (2009). Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions.

Swarm Intell, 87–124. doi:10.1007/s11721-008-0021-5

Li, Y. (2010). Solving TSP by an ACO-and-BOA-based hybrid algorithm. 22-24. doi:10.1109/ICCASM.2010.5622108

Lu, D. S., & Chen, C. C. (2008). Edge detection improvement by ant colony optimization. *Pattern Recognition Letters*, 416-425.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 46-61. doi:https://doi.org/10.1016/j.advengsoft.2013.12.007

Pawar, P. H., & Patil, R. P. (2014). Image Edge Detection Techniques using MATLAB Simulink. *International Journal of Engineering Research & Technology, 3*(6), 2149-2153.

Rafsanjani, M. K., & Varzaneh, Z. A. (2015). Edge detection in digital images using Ant Colony Optimization. *Computer Science Journal of Moldova, 33*(3(69)), 343-359.

Rana, D., & Dalai, S. (2014). Review on Traditional Methods of Edge Detection to Morphological based Techniques. *International Journal of Computer Science and Information Technologies*, 5915-5920.

Saeed, V. A., Almufti, S. M., & Marqas, R. B. (2019). Taxonomy of bio-inspired optimization algorithms. *Journal of Advanced Computer Science & Technology*, 23-31. doi:DOI: 10.14419/jacst.v8i2.29402

Yang, X.-S. (2010). A New Metaheuristic Bat-Inspired Algorithm. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, (pp. 1-10). Retrieved from https://arxiv.org/abs/1004.4170

Yazdani, M., & Jolai, F. (2016). Lion Optimization Algorithm (LOA): A nature-inspiredmetaheuristic algorithm. *Journal of Computational Design and Engineering, 3*, 24-36. doi:10.1016/j.jcde.2015.06.003