Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Ant Colony Optimization Based Edge Detection Algorithm

Fallah H. Najjar[1,2*], Ola N. Kadhim[3], Salman Abd Kadum[1]

[1]Technical Institute of Najaf, Al-Furat Al-Awsat Technical University, Najaf 54001, Iraq
[2]Department of Emerging Computing, Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia
[3] Technical Institute of Al-Mussaib, Al-Furat Al-Awsat Technical University, Najaf 54001, Iraq
[*]Corresponding author E-mail: fallahnajjar@atu.edu.iq

https://doi.org/10.46649/fjiece.v3.2.32a.11.6.2024

Abstract. *The problem of edge detection represents one of the most elementary assignments in image processing, providing an essential base for all further study and interpretation of the visual data analysis. This paper proposed an enhanced version of the Ant Colony Optimization (ACO) algorithm for edge detection. The following paper tries to compare the Proposed ACO method with the conventional techniques of edge detection like Canny, Prewitt, and Sobel, using various quantitative metrics like Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Entropy, Natural Image Quality Evaluator (NIQE), and Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) applied over different images. The datasets for this evaluation are considered as a standard Cameraman, a biological cell, and MRI image, with and without noise, considering the ranges of complexities and textures. The results of our study prove the competencies of ACO's algorithm. In some cases, it stands out against standard algorithms for MSE and PSNR values and maintains high Entropy values, suggesting the robustness of detail-keeping in an image. Further, the quality assessment of the images by using NIQE and BRISQUE shows the ability of ACO to maintain a natural appearance post-edge detection. In this regard, the study highlights that the proposed ACO is an effective method for edge detection in varying image conditions, and, in doing so, it even validates the effectiveness of bio-inspired algorithms in image processing domains.*

Keywords: *Image processing; edge detection; Ant colony optimization; ACO.*

1. INTRODUCTION

Edges are a primary feature of an image because they form the part in which transitions between different objects, regions, or primitives take place [1]. Most of the information in an image is contained at the edges [2]. Typically, an edge is characterized by a group of pixels where there is a noticeable shift in gray-level values [3]. Edge detection is fundamentally vital for crucial technologies in computer vision [4], which, in an image-based approach, separate different elements in an image depending upon those sudden changes or variations of gray levels or textures. In brief, this technique is essential in applications such as object tracking [5], segmentation [6], active contours [7], cryptography [8], and many others, very much so that it groups detected edges into contours or surfaces outlining object boundaries. Edge detection is designed to segment images by identifying and delineating these sharp discontinuities in gray levels [9]. It includes trendy methods like Canny, Sobel, Prewitt, and Log. All these methods in the traditional system have a colossal search space on which they carry out edge detection, hence making this process very memory-bound and slow [10]. Ant Colony Optimization (ACO) is an optimization algorithm

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

used for computation in ant-like agents that resemble the foraging food by real ants [11]. It is a technique of computational evolutionary computation in which ants communicate with each other by means of pheromone trails about the food sources found [12]. These pheromone markers evaporate in their turn and are attractive to other ants, thus positively reinforcing the discovery of the best paths [13]. This method has been adapted for edge detection, adding strategies to optimize and streamline the process [14].

For example, some combination methods of edge detection based on ACO have been proposed to address the build-in problems associated with incomplete, broken edges and huge computational loads [15]. Considering the ever-increasing significant data challenges, traditional edge detection is performing very poorly [16, 17]. It takes much processing time, especially for larger images, and when the complexity of the image is higher. Even the computational power and algorithms need to be improved to keep pace with such humongous datasets [18].

Our main aims in this study are to randomize the selection of the ant position and dynamic threshold updating. However, with this brief introduction, the rest of this paper is organized into five sections. The second section explains the review of the related literature. The third section elaborates on the architecture, flowchart, and algorithm of our proposed method. The fourth section will discuss the results obtained from applying this method, and the final section will provide conclusions.

2. RELATED WORKS

Chen [19] implements an ACO technique for edge detection in images using the distributed framework through Hadoop/MapReduce while doing parallel computing to gain performance and accuracy improvement. Chen's approach has the same computational intensity and memory challenges as the traditional method, which was customarily distributed over many nodes, thus enabling the processing of enormous image data sets. Chen's approach embraces a newly parallelized ACO technique where the images taken as input are ready for processing and converted into a form that can be computed. The approach does the edge detection in a mapped way, through segmenting and using ACO to find the edges; in a reduced way, the results are finally composited and stored. This process is highly distributed in nature from within the benefit of MapReduce, accelerating computation speeds and scaling higher levels of data effectively.

Reddy and Pandian [20] have proposed an edge detection of the image based on ACO. It is a new approach to edge detection of the image that closely relates to foraging behavior with which ants are heavily interconnected. Their framework utilized the natural behavior of ants, which is laying down pheromones leading to optimal paths between food sources and their colony, as a metaheuristic algorithm to resolve the problem of edge detection in digital images. They exploit the power of ACO in coming up with approximate solutions to complex optimization problems, especially adaptations that would handle subtleties of finding critical changes in image intensity, which describes the edges of images. What they follow is a method whereby they build a pheromone matrix that reflects changes in intensity at each pixel, led by a heuristic matrix that speeds its convergence toward optimal threshold edge detection values.

Liu and Pu [21] proposed an algorithm for the detection of edges in images using the fractional-order ACO approach coupled with fractional differential masks and the coefficient of variation. It is denoted as FACAFCV. Characteristics of fractional calculus—which are the ingredients of long memory, non-locality, and weak singularity—are premised on the method at hand to increase the precision of edge detection in digital images. They generalized that the simple method borrowed from traditional integer order ant-colony algorithms has been generalized in their framework with fractional order techniques, playing a better edge detection role than traditional integer order ant-colony algorithms, mainly in the presence of noise. The authors deploy the methodology where the ants are initialized on the image pixels while their motion and pheromone patterns are applied to fractional calculus principles, thus involving a finer search for the edge features in the images.

The results of the experimentation carried out show that FACAFCV yielded much better accuracy in edge detection than traditional methods, particularly in the images destroyed by multiplicative noise. The result concluded that fractional calculus not only helps enhance the edge but also reduces the effect of noise present in the image; therefore, it is robust for use in practical situations for edge detection. They proposed the development of adaptive algorithms in the future, where the parameters of the fractional-order model would change based on the specific characteristics of the input images. All these clearly point to the necessity for in-depth testing with all sorts of images at different noise levels to validate and tune further the effectiveness of the algorithm. Such progress can open the way for broader applications of fractional-order algorithms to real-world image-processing tasks.

3. MATERIALS AND METHODS

This paper proposes an approach to the problem of edge detection in grayscale images using ACO techniques. This present method imitates the behaviors of ant colonies that can enhance edges in a process over digital images. Here, we outline the methodology in four main steps: initialization, construction, update, and decision processes.

Step 1: Initialization Process

Initialize the positions of K ants that are assigned randomly on image (I) and initialize the pheromone matrix.

Step 2: Construction Process

For construction step n=1: N

For ants k=1: K

Sequentially move the ants for L steps, according to a probabilistic transition matrix $P^{(n)}$

$$P^{n}_{(l,m),(i,j)} = \frac{\left(\tau_{i,j}^{(n-1)}\right)^{\alpha}\left(\eta_{i,j}\right)^{\beta}}{\sum_{(i,j)\in\Omega_{(l,m)}}\left(\tau_{i,j}^{(n-1)}\right)^{\alpha}\left(\eta_{i,j}\right)^{\beta}} \tag{1}$$

Where: $\eta_{i,j}$, heuristic information, can be calculated using equation 2.

$$\eta_{i,j} = \frac{1}{Z} V_c\left(I_{i,j}\right) \tag{2}$$

Where: $V_c\left(I_{i,j}\right)$, represent the measure of local intensity variation, can be calculated using equation 3.

$$\begin{aligned} V_c\left(I_{i,j}\right) = f\big(&\left|I_{i-2,j-1} - I_{i+2,j+1}\right| + \left|I_{i-2,j+1} - I_{i+2,j-1}\right| + \left|I_{i-1,j-2} - I_{i+1,j+2}\right| \\ &+ \left|I_{i-1,j-1} - I_{i+1,j+1}\right| + \left|I_{i-1,j} - I_{i+1,j}\right| + \left|I_{i-1,j+1} - I_{i-1,j-1}\right| \\ &+ \left|I_{i-1,j+2} - I_{i-1,j-2}\right| + \left|I_{i,j-1} - I_{i,j+1}\right|\big) \end{aligned} \tag{3}$$

Step 3: Update Process

The pheromone matrix is updated in two stages:

Step 3.1: Local update:

After every move of an ant, the level of pheromone at every visited pixel is raised based on the gradient at that pixel modulated by an evaporation factor ρ. The local update is:

$$\tau_{i,j}^{(n-1)} = \begin{cases} (1-\rho).\tau_{i,j}^{(n-1)} + \rho.\Delta_{i,j}^{(k)}, & if\ (i,j)\ is\ visited\ by\ the\ current\ k^{th} ant \\ \tau_{i,j}^{(n-1)}, & otherwise \end{cases} \quad 4$$

Step 3.2: Global update:

After all ants have moved in a given iteration, the global update is applied to refine the paths in which many ants agree on significant edge features, which is computed as:

$$\tau^{(n)} = (1-\psi).\tau^{(n-1)} + \psi.\tau^{(0)} \quad 5$$

Step 4: Decision Process

The final decision if the pixel is part of an edge or not belongs to an applied threshold T and the pheromone matrix. A binary image is produced where each pixel is marked as an edge if its pheromone value exceeds this threshold. The current dynamic method of updating T iterates, step by step in each iteration, updates to separate the pheromone matrix into two classes with progressive adjustment based on the mean values of the two classes until convergence as the following:

Step 4.1: Initialize $\boldsymbol{T^{(0)}}$ $\boldsymbol{using\ equation}$ 6, then set the iteration index l to 0.

$$T^{(0)} = \frac{\sum_{i=1:M1}\sum_{j=1:M2}\tau_{i,j}^{(N)}}{M1M2} \quad 6$$

Step 4.2:

Separate the pheromone matrix $\tau^{(N)}$ into two classes using $T^{(l)}$. Furthermore, calculate the mean of each class using equations 7 and 8.

$$\mu_L^l = \frac{\sum_{i=1:M1}\sum_{j=1:M2} g_{T^{(l)}}^L(\tau_{i,j}^{(N)})}{\sum_{i=1:M1}\sum_{j=1:M2} h_{T^{(l)}}^L(\tau_{i,j}^{(N)})} \quad 7$$

$$\mu_U^l = \frac{\sum_{i=1:M1}\sum_{j=1:M2} g_{T^{(l)}}^U(\tau_{i,j}^{(N)})}{\sum_{i=1:M1}\sum_{j=1:M2} h_{T^{(l)}}^U(\tau_{i,j}^{(N)})} \quad 8$$

Where $g_{T^{(l)}}^L(x)$, $h_{T^{(l)}}^L(x)$, $g_{T^{(l)}}^U(x)$, and $h_{T^{(l)}}^U(x)$ can be calculated using the following equations.

$$g_{T^{(l)}}^L(x) = \begin{cases} x, & x \le T^{(l)} \\ 0, & otherwise \end{cases} \quad 9$$

$$h_{T^{(l)}}^L(x) = \begin{cases} 1, & x \le T^{(l)} \\ 0, & otherwise \end{cases} \quad 10$$

$$g_{T^{(l)}}^U(x) = \begin{cases} x, & x \ge T^{(l)} \\ 0, & otherwise \end{cases} \quad 11$$

$$h_{T^{(l)}}^U(x) = \begin{cases} 1, & x \ge T^{(l)} \\ 0, & otherwise \end{cases} \quad 12$$

Step 4.3: Update the threshold for the next iteration

$$T^{(l=1)} = \frac{\mu_L^l + \mu_U^l}{2} \qquad 13$$

Step 4.4: Check for convergence using equation 14, then repeat Step 4.2;

$$if \left|T^{(l=1)} - T^{(l)}\right| > \epsilon \qquad 14$$

Finally, finalize the edge detection:

$$E_{i,j} = \begin{cases} 1, & if\ \tau_{i,j}^{(n-1)} \geq T^{(l)} \\ 0, & otherwise \end{cases} \qquad 15$$

Furthermore, **Fig. 1** displays the flowchart of the proposed methodology.

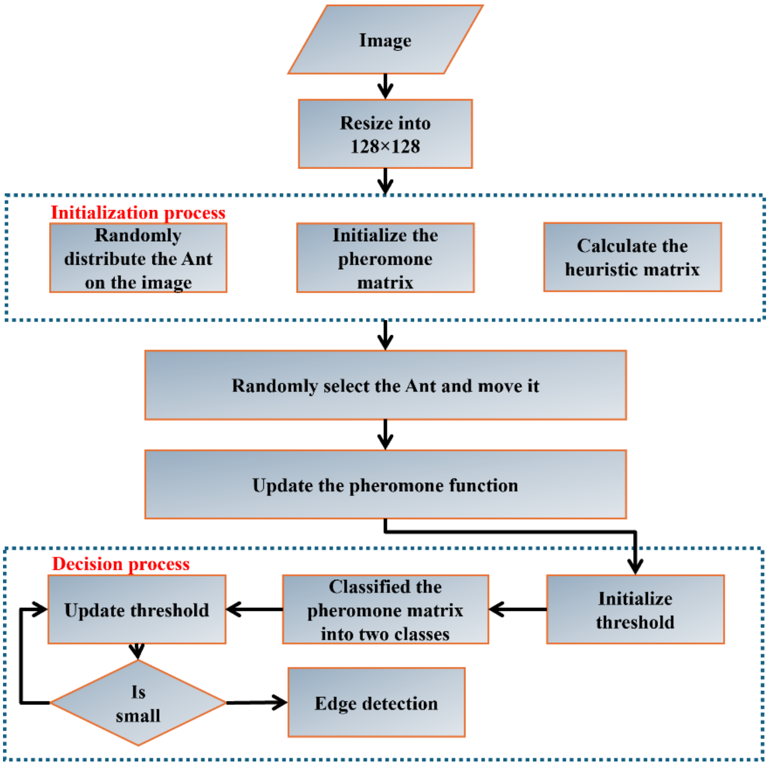

Fig. 1. Proposed Method Flowchart.

4. RESULTS AND DISCUSSION

In this study, we test our proposed ACO method using three standard images, see **Fig. 2**. **Table 1** provides a comparison of the ACO method against traditional methods (Canny, Sobel, Prewitt) across three images: Cameraman, Cell, and MRI, across various image quality metrics. Test results after the experiment showed that the ACO approach produced excellent performance with respect to all the basic metrics of image processing. The consistently lower MSE values indicate that ACO introduces fewer errors during the image processing task.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Cameraman Cell MRI

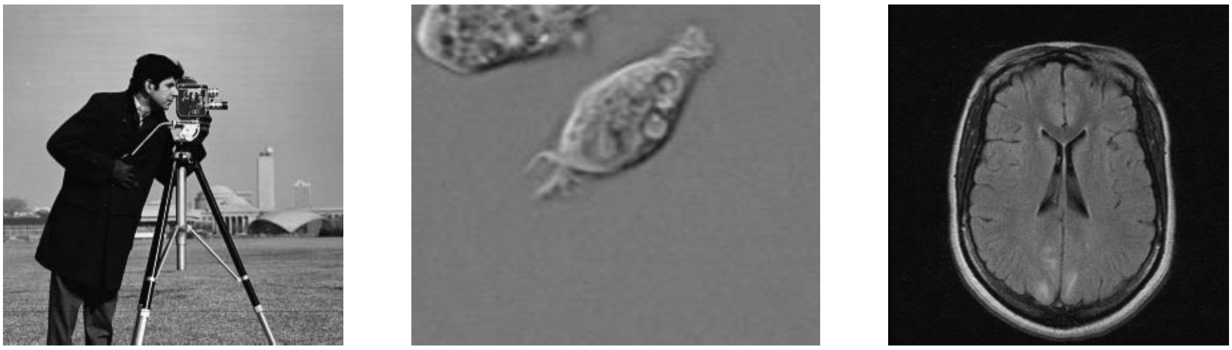

Fig. 2. Sample Images.

So, it means that by virtue of the fact that generally, these metrics were lower and added up to higher PSNR values across the dataset. ACO has a higher capacity to maintain a higher quality signal in the presence of potential noise. Wherever the actual quality of the image is required, as in medical imaging or the analysis of satellite images, this is indispensable in applications. ACO implies higher entropy values; therefore, it would be better for retaining or even improving the informational content of the image.

Table 1. Comparative analysis: best results are in bold.

Image	Metric	ACO	Canny	Prewitt	Sobel
Cameraman	MSE	**0.2712**	0.2918	0.2851	0.2847
	PSNR	**6.8588**	5.3489	5.4507	5.4566
	Entropy	0.7186	0.4685	0.2344	0.2339
	NIQE	20.647	25.074	26.0458	26.7872
	BRISQUE	46.2264	46.6264	47.8466	47.8234
Cell	MSE	0.2031	0.2251	0.2208	0.2209
	PSNR	8.0151	6.476	6.5598	6.5585
	Entropy	0.5592	0.4188	0.2138	0.2148
	NIQE	17.8456	18.8587	18.8541	18.8542
	BRISQUE	48.2442	48.8813	46.5195	46.5867
MRI	MSE	0.0625	0.0976	0.0755	0.0756
	PSNR	12.3363	10.1077	11.2199	11.217
	Entropy	0.4266	0.4187	0.1322	0.1328
	NIQE	17.913	19.1088	22.0799	22.2711
	BRISQUE	45.3039	47.6813	45.6402	45.6691

This will be an essential aspect in cases where the task at hand for a texture analysis has to be exhaustive or where the subtlety of the image carries essential information, as in the case of pattern recognition or forensic analysis. However, while it displays such competitive NIQE and BRISQUE scores, it showcases performance variance. All of these measures predict all practical purposes for human

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

perceptions, and so they will match the outcome predictably, regardless of an ACO's efficiency on technical measures. Finally, **Fig. 3** displays the detected edges across different edge detection methods.

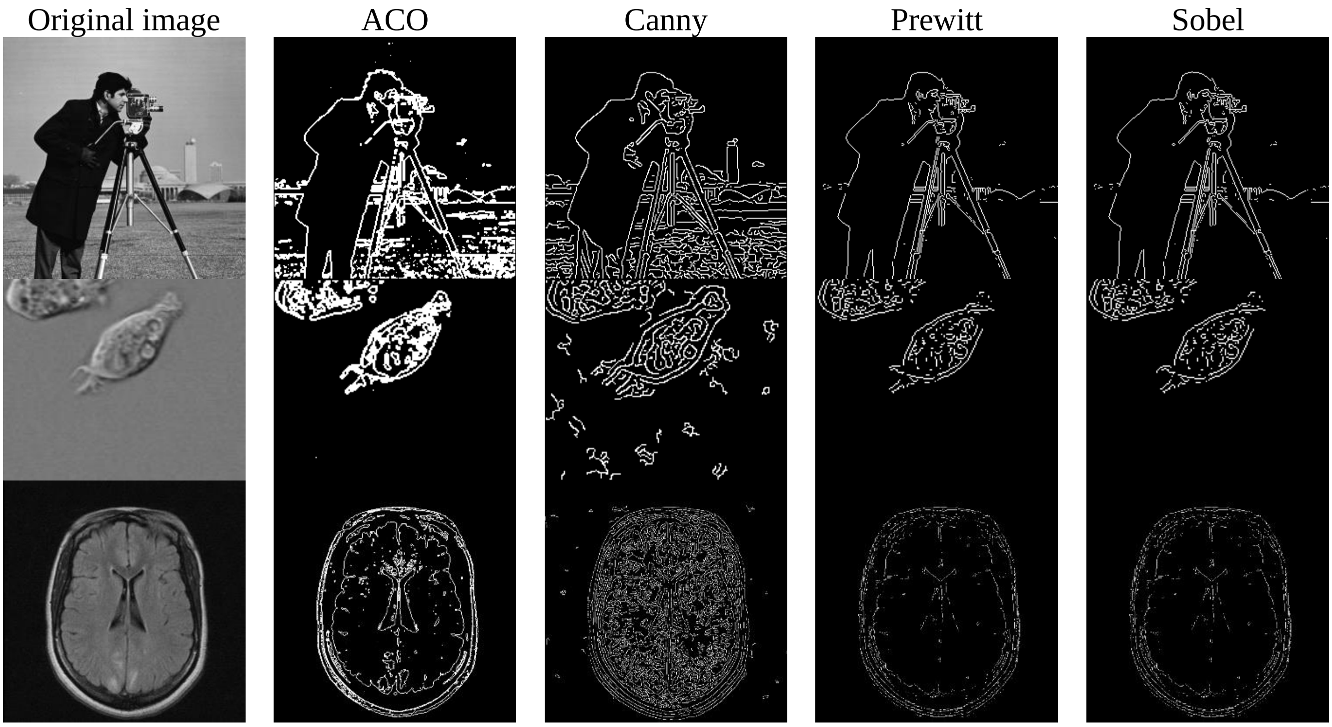

Fig. 3. Detected edges.

The reasons for this variance could be, for instance, that ACO treats image details and noise in ways that affect their visual aesthetic qualities and are not well represented by traditional metrics like MSE or PSNR. Of course, there is a need to compare traditional edge detection methods, such as Canny, Sobel, and Prewitt, to make a simple conclusion that ACO would be able to reduce errors and improve the quality of the signal effectively. However, there may still be some situations where there is a necessity for these traditional methods, especially when there is ease of computation and speed involved, being paramount to the smallest gain in reduction of error and preservation of detail facilitated by ACO.

5. CONCLUSIONS

In this paper, an enhanced ACO algorithm for edge detection was proposed, and its result was compared with traditional techniques of edge detection, such as Canny, Prewitt, and Sobel. Such comparison uses quantitative metrics such as MSE, SNR, Entropy, NIQE, BRISQUE, and a whole bunch more. This comparison is done across quite an array of different images, including Cameraman, cells, and MRI images, all differing in complexity and texture. From the experiment, we can conclude that the ACO algorithm works well, especially in cases where the algorithm works more efficiently than the conventional one with high MSE and PSNR, maintaining high values of entropy. It assures that the method can keep more detailed information in images. It is worth noting that quality assessment employing NIQE and BRISQUE algorithms reveals that ACO manages a look that is representative of a natural look after edge detection. The paper, therefore, applied the effectiveness of bio-inspired algorithms in the management of diverse imaging conditions in the edge detection framework.

REFERENCES

[1] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences,* vol. 207, no. 1167, pp. 187-217, 1980.

[2] N. You, L. Han, D. Zhu, and W. Song, "Research on image denoising in edge detection based on wavelet transform," *Applied Sciences,* vol. 13, no. 3, p. 1837, 2023.

[3] Y. Chi, W. Zhao, and B. Pan, "Gray level residual field: an effective metric for pixelwise matching quality evaluation in local digital image correlation," *Measurement Science and Technology,* vol. 34, no. 7, p. 075018, 2023.

[4] G. Cai, J. Qian, T. Song, Q. Zhang, and B. Liu, "A deep learning-based algorithm for crop Disease identification positioning using computer vision," *International Journal of Computer Science and Information Technology,* vol. 1, no. 1, pp. 85-92, 2023.

[5] D. Meimetis, I. Daramouskas, I. Perikos, and I. Hatzilygeroudis, "Real-time multiple object tracking using deep learning methods," *Neural Computing and Applications,* vol. 35, no. 1, pp. 89-118, 2023.

[6] Sakshi and V. Kukreja, "Image segmentation techniques: statistical, comprehensive, semi-automated analysis and an application perspective analysis of mathematical expressions," *Archives of Computational Methods in Engineering,* vol. 30, no. 1, pp. 457-495, 2023.

[7] Y. Chen, P. Ge, G. Wang, G. Weng, and H. Chen, "An overview of intelligent image segmentation using active contour models," *Intell. Robot,* vol. 3, no. 1, pp. 23-55, 2023.

[8] M. M. Msallam, "An approach to hide an audio file in image using LSB technique," *Al-Furat Journal of Innovations in Electronics and Computer Engineering,* 2023.

[9] Q. Wu and K. R. Castleman, "Image segmentation," in *Microscope Image Processing*: Elsevier, 2023, pp. 119-152.

[10] Y. Zhou, F. He, and Y. Qiu, "Optimization of parallel iterated local search algorithms on graphics processing unit," *The Journal of Supercomputing,* vol. 72, no. 6, pp. 2394-2416, 2016.

[11] I. Michelakos, N. Mallios, E. Papageorgiou, and M. Vassilakopoulos, "Ant colony optimization and data mining," in *Next Generation Data Technologies for Collective Computational Intelligence*: Springer, 2011, pp. 31-60.

[12] A. Nayyar and R. Singh, "Ant colony optimization—computational swarm intelligence technique," in *2016 3rd International conference on computing for sustainable global development (INDIACom)*, 2016: IEEE, pp. 1493-1499.

[13] J. Shang *et al.*, "A review of ant colony optimization based methods for detecting epistatic interactions," *IEEE access,* vol. 7, pp. 13497-13509, 2019.

[14] K. Raimond and J. Lovesum, "A novel approach for automatic remodularization of software systems using extended ant colony optimization algorithm," *Information and software technology,* vol. 114, pp. 107-120, 2019.

[15] G. Tato, "Lazy and locality-aware building blocks for fog middleware: a service discovery use case," Université de Rennes, 2019.

[16] F. Wang, M. Zhang, X. Wang, X. Ma, and J. Liu, "Deep learning for edge computing applications: A state-of-the-art survey," *IEEE Access,* vol. 8, pp. 58322-58336, 2020.

[17] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys & Tutorials,* vol. 22, no. 2, pp. 869-904, 2020.

[18] N. O'Mahony *et al.*, "Deep learning vs. traditional computer vision," in *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1 1*, 2020: Springer, pp. 128-144.

[19] M. Chen, "A Distributed Ant Colony Optimization in Edge Detection," 2021.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

[20] C. Gupta and S. Gupta, "Edge detection of an image based on ant colony optimization technique," *Int. J. Sci. Res.(IJSR)*, vol. 2, no. 6, pp. 1256-1260, 2013.
[21] X. Liu and Y.-F. Pu, "Image Edge Detection Based on Fractional-Order Ant Colony Algorithm," *Fractal and Fractional*, vol. 7, no. 6, p. 420, 2023.