University of Ibadan
Journal of Science and Logics in ICT Research

An Enhanced Data Encryption Standard Algorithm for Securing Medical Images

[1]✉Olumide, A. T., [2]Ajala, F. A. and [3]Fenwa, O. D.

Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria
atolumide@student.lautech.edu.ng; faajala@lautech.edu.ng; odfenwa@lautech.edu.ng;

Abstract

Securing medical images on the internet, in public or local networks is very important because they contain sensitive information about the well-being of patients and help health personnel to carry out diagnoses easily. Data Encryption Standard (DES) has been a popular algorithm used for image encryption due to its low memory consumption and good throughput but suffers from high computational time and lossy image problems after decryption. In this paper, the Ant Colony Optimization (ACO) technique was integrated into DES to enhance its performance for medical image encryption. The enhanced and existing DES's performance were measured in terms of memory utilisation, Computational time, output bytes, Mean Square Error (MSE), and Peak signal-to-noise ratio (PSNR) to evaluate differences and draw conclusions. Results showed that the enhanced DES performed better than the existing DES.

Keywords: Encryption algorithms, Medical Image Processing, Data Encryption Standard (DES) algorithm, Ant Colony Optimization (ACO)

1. INTRODUCTION

Information security has become extremely important to most organizations today. The security of any system and people remains a top priority for many institutions around the world [14]. Meanwhile, in today's health information systems, medical image data is a key part of diagnosis. However, transferring medical data, such as images or radiology results, from one medical data centre to another without security technologies means a low level of patient privacy [2]. In order not to violate medical images and ensure data integrity, one of the most common approaches is data encryption or cyphering, which uses methods that make readable data unreadable [4]. Encryption achieves irrefutability, reliability, availability, integrity, authentication, secrecy and confidentiality [5].

1.1 Medical Image Processing

An image is known to be a collection of measurements in two-dimensional (2-D) or three-dimensional (3-D) spaces [17]. Meanwhile, to get original information from an image dataset, it has to undergo various phases of processing. One of them is image processing, which is said to be a set of tools or techniques which involves converting an analogue or continuous image into digital form [19]. It helps to improve the quality and size of the obtained dataset. Image pre-processing procedures can be conducted to achieve the following goals: reducing the noise in the initial images, enhancing the quality through raising the contrast, and removing the high/low frequencies [12].

1.2 Encryption Algorithms

Encryption algorithms are broadly divided into two, namely: symmetric and asymmetric encryption algorithms. In symmetric algorithms, only one key is used to encrypt and decrypt data or files [10]. One of the most popular symmetric algorithms for image encryption was Data Encryption Standard (DES) due to its low memory consumption and good throughput but

Olumide, A. T., Ajala, F. A. and Fenwa, O. D. (2023). An Enhanced Data Encryption Standard Algorithm for Securing Medical Images, *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 10 No. 2, pp. 107 – 118.

suffers lossy image problem after decryption, brute force attacks due to low key strength and high computational time. Hence, the introduction of an optimization algorithm to improve its efficiency. Other symmetric algorithms used for image encryption are Triple Data Encryption Algorithm (that is, 3DES), Blowfish and Advance Encryption Standard (AES). While in asymmetric algorithms, the Public key is used for encryption and the private key is used for decryption [10]. A popular example is the Rivest Shamir and Alderman algorithm (RSA).

Data Encryption Standard (DES)

Data Encryption Standard was designed by International Business Machines (IBM) in 1974 and it was the first encryption standard to be published by NIST (National Institute of Standards and Technology) in America. Grabbe [8] illustrated the Data Encryption Standard technique as an algorithm that takes a 64-bit block of input as input and output same as cipher text as shown in Figure 1. In each round, data and key bits are shifted, permutated, XORed, and sent through, 8 s-boxes, a set of lookup tables that are essential to the DES algorithm [16].

1.3 Optimization Algorithms

Optimization, as the name suggests, is a way to solve a problem by tuning a set of parameters in order to achieve an optimal solution towards a defined goal. The purpose of using optimization techniques would be to tune certain parameters on which the profit and the expenses depend, so that the function maximization or minimization is possible. According to Ramson *et al.* [13], several natures inspired optimization algorithms have been developed and studied so far. They are, Genetic Algorithm (GA), Artificial Immune Systems (AIS), Ant Colony Optimization Algorithm (ACO), Particle Swarm Optimization (PSO) and many more.

1.3.1 Genetic Algorithm

Genetic algorithm is a search heuristic that was inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation [20].

1.3.2 Artificial Immune Systems (AIS)

Artificial Immune Systems (AIS) are adaptive systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving. This is a new and emerging soft computing method. In past decades, researchers have found various engineering and computational solutions through AIS. This algorithm was developed by Goodman *et al.* [7] and tested against Kohenon's learning vector quantization (LVQ) algorithm and the k-nearest neighbours (KNN) algorithm.

Step 1*: Create 16 subkeys -* C_n *and* $D_{n,}$ *each of which is 48 bits long.*

$$(1 <= \boldsymbol{n} <= 16) \qquad (1)$$

Where ***n*** *is the number of iterations in the subkeys*

Step 2: *Encode each 64-bit block of data by handing it over to an initial Permutation (IP) function which will act on the plain text. Each bit is moved to a new bit position called* ***M***

Step 3: *Divide the 64-bit permuted input into two 32-bit blocks, called left and right, respectively. The initial values of the left and right blocks are denoted* $\boldsymbol{L_0}$ *and* $\boldsymbol{R_0}$

$$IP/2 = \mathbf{L}_0 = \mathbf{R}_0 \qquad \mathbf{(}2)$$

That is**,** $\mathbf{L_0}$ = 32bits, $\mathbf{R_0}$ = 32bits

Step 4: E*ach* $\boldsymbol{L_n}$ *and* $\boldsymbol{R_n}$ *to go through 16 rounds of encryption process.*

$$Ln = R_n\text{-}1 \qquad (3)$$

$$R_n = L_n\text{-}1 + f(R_n\text{-}1, K_n) \qquad (4)$$

Step 5: L_0 *and* R_0 *are rejoined and a Final Permutation (FP) is performed on the combined block*

Step 6: *The result of this process produces 64-bit cipher text which is the encrypted image.*

Step 7: *Stop*

Figure 1: Algorithm for the Existing Data Encryption Standard Technique

1.3.3 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a population-based stochastic optimization algorithm motivated by intelligent collective behaviour of some animals such as flocks of birds or schools of fish. Since presented in 1995, it has experienced a multitude of enhancements. As researchers have learned about the technique, they derived new versions aiming to different demands, developed new applications in a host of areas, published theoretical studies of the effects of the various parameters and proposed many variants of the algorithm [21].

1.3.4 Ant Colony Optimization (ACO) Algorithm

ACO is an evolutionary algorithm inspired by an ant’s natural behavior. The bulk of the ant colony optimization algorithm is made up of only a few steps. First, each ant in the colony constructs a solution based on previously deposited pheromone trails. Next ants will lay pheromone trails on the components of their chosen solution, depending on the solution's quality. Finally, after all ants have finished constructing a solution and laying their pheromone trails, pheromone is evaporated from each component depending on the pheromone evaporation rate. Ant colony algorithm used by Ajala *et al.* [3] was used in conjunction with DES.

2. Related Works

Jamil and Rahma [9] proposed to use block chain technology with the Data Encryption Standard (DES) algorithm for the purpose of increasing the degree of security of the transmitted images by enhancing the key during the process of encrypting the transmitted images as well as increasing the degree of authentication between the sender and receiver. Experimental outcomes manifested that the security of encryption image that gained via the suggested algorithm is higher, performing the goal of protecting the information of medical image.

Abbas and Maisa’a [1] utilized the triple data encryption standard (3DES) encryption scheme with three chaotic maps namely logistic map, Arnold Cat’s map, and Baker’s map to build a digital image encryption strategy depending on a chaotic system. The results of the experiments revealed that the suggested digital image encryption technique is both efficient and secure, making it ideal for usage in insecure networks.

Singh *et al.* [18] examined existing studies on encryption algorithms including AES, 3DES, Blowfish, and DES. They discovered that DES key size was too little in contrast to other methods; 3DES was a sluggish and ineffective block cypher and AES was considered to be preferable than the original Blowfish algorithm. Furthermore, it was said that aside from the security issue, employing these cyphers directly to encrypt images takes a long time and is not suitable for real-time applications. To deal with these problems, they proposed that future research should modify them to create enhanced ones.

Chaudhary *et al.* [6] made a comparative study of cryptographic algorithms like DES, 3DES, AES, RSA, DH and hybrid techniques like dual RSA, AES-RSA, RSA-AES-DES on the basis of efficiency. The study concluded that efficiency of symmetric algorithms are more than asymmetric algorithms and efficiency of hybrid techniques is average but is more secure.

Kavitha and Saraswathi [11] surveyed various image encryption and decryption techniques. Many encryption techniques were studied and analysed to endorse the recital of the encryption methods. In all methods, original image was embedded and encrypted then send it to the receiver. Each algorithm, method and technique used was unique. They stated that every day latest encryption technique are evolving and more secure encryption techniques with high rate of security will work out forever. They proposed that further research should be done to provide an enhanced encryption technique that will be secured and have a reduced amount of noisy medical image.

It is noteworthy to say that several researchers have evaluated the use of DES algorithm to encrypt data independently or used in comparison with other encryption algorithms. However, this research enhanced the performance of DES algorithm in medical image encryption using the Ant Colony Optimization technique to solve its lossy image problem after encryption and high computational time.

3. METHODOLOGY

In this research, an enhanced DES algorithm was developed to secure medical images using the Ant Colony Optimization (ACO) technique. The methods adopted for the research are:

i) Image Acquisition: Ten human brain images of different sizes were obtained online from the Whole Brain Atlas (WBA) Web Archive.

ii) Preprocessing of the Dataset: The images obtained were preprocessed using the Gaussian filtering algorithm. This was to improve image quality and prepare it for further processing.

iii) Designing of an enhanced DES algorithm using the Ant Colony Optimization technique.

iv) Implementation of encryption and decryption of the processed images using the enhanced DES.

v) Evaluate the performance of the enhanced DES algorithm and the existing DES algorithm based on memory utilization, computational time, Output bytes, mean square error and peak signal-to-noise ratio.

3.1 Image Acquisition and Pre-processing

Ten medical images of human brains of different sizes were retrieved from the WBA online repository. Figure 2 shows the original image of different sizes used in the study. Acquired images were placed in a folder linked to the user interface via the encrypt button. It was designed using Python programming language version 2.1 as the software to carry out the implementation of the safe-to-use interface.

Image pre-processing was necessary before performing image analysis. This includes filtering, selection, randomization, grayscale, resizing and removal of items that may affect the proper processing of images. The main purpose of pre-processing was to improve image quality and prepare it for further processing by removing or reducing irrelevant and redundant parts of the image background. Noise and high-frequency components were removed using filters. In this work, Gaussian filtering was used for the image pre-processing.

3.2 Enhanced DES with ACO

This is a combination of the two algorithms to form a hybridization version. This is done to improve the standard DES algorithm. The steps of the combined algorithm can be seen in Figure 3. Ant Colony Optimization algorithm used by Ajala *et al.* [3] was used in conjunction with DES to implement the enhanced DES in this paper. The ACO algorithm was a novel algorithm developed by the aforementioned researchers.

The bold part of the enhanced DES algorithm shows where ACO was introduced. Feature extraction of acquired images was important in conducting this study. Therefore, the edges are extracted as features and optimized using the ant colony optimization algorithm. This is because the ACO can perform these two processes. The ant colony optimization parameters are initialized in this segment, which is the pheromone trail, its intensity and the cluster centre, which is calculated to obtain the optimized image.

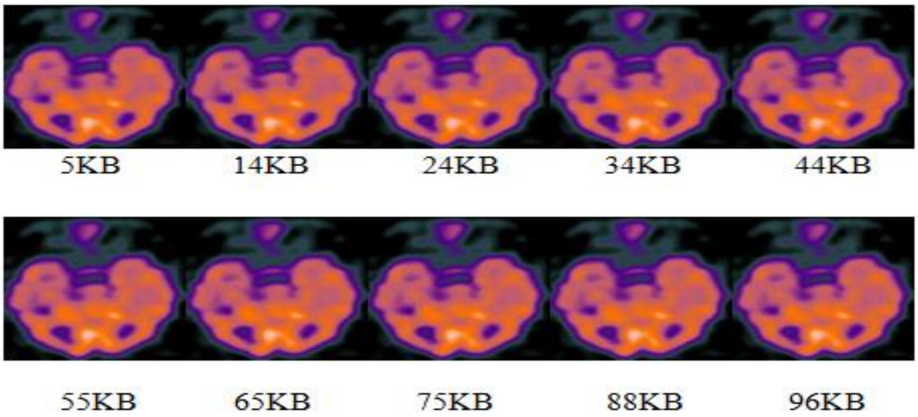


Figure 2: Image Sizes Used for Encryption

```
Step 1: Input the plain image (I)
Step 2: Initialize all parameters with a number of fault systems on the image and Compute the Pheromone trail, its intensity and cluster centre.
        $C_p = \int_I^n Pt, Pi, Cj$        (5)
Step 3: Compute the biased error and overall error to give the optimized image as the input to be encrypted.
        $I' = A_\infty - B_\infty$        (6)
Step 4: Create 16 subkeys for the optimized image (I') and encode to 64-bit permutated input.
        $IP = I'_{1 \le n \le 16}$        (7)
Step 5: Divide the 64 bits into half, making each left and right block 32-bit blocks.
        $\frac{IP}{2} = L_0R_0$        (8)
Step 6: Perform 16 rounds of encryption process on both blocks and combine the blocks before the final permutation to give the results.
        $FP = IP^{-1}$        (9)
Step 7: Output the decryption of the encrypted image
Step 8: Stop.
```

Figure 3: Pseudocode for the Enhanced Data Encryption Standard Algorithm

3.3 System Requirements

The algorithm was implemented using Python programming language and JavaScript (Nodejs) on a MacBook Pro environment software. It was implemented on a 16 Gigabytes RAM, 500GB SSD Hard Drive space of Intel Core i7 system running Mac Operating System (MacOS).

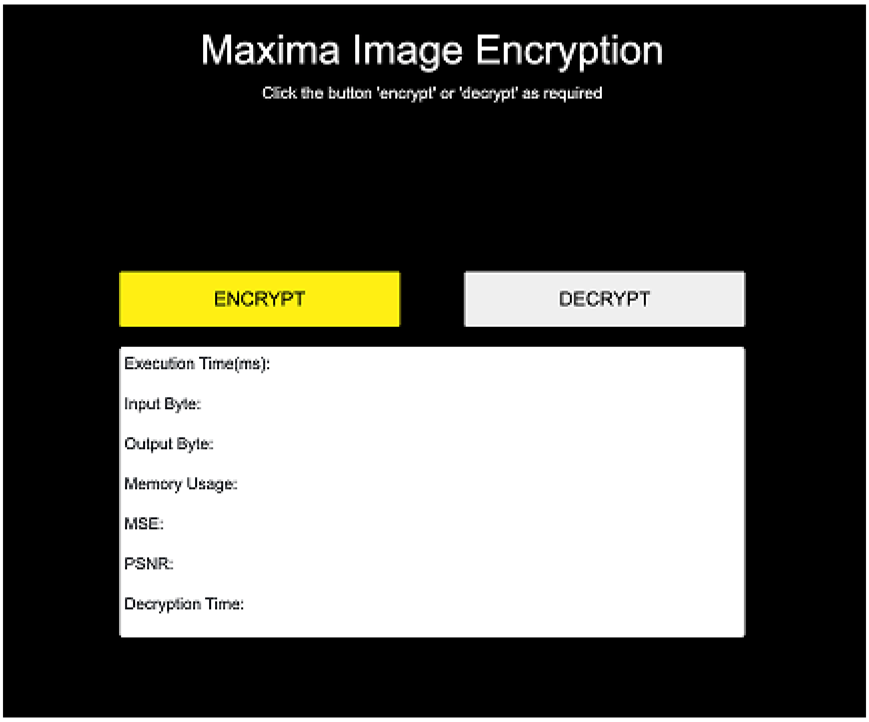
The system is meant to run on any Operating system that has Python software installed in it.

i) System: PC with Python software and JavaScript installed in it.
ii) RAM: 2GB or greater
iii) Operating System: Any Operating system with Python software and JavaScript installed.
iv) Free Hard Drive Space: Up to 500MB for Python installation, less than 100mb for use by the application and data.

4. RESULTS AND DISCUSSION

4.1 Summary of Results

The algorithm was implemented using a visual studio code as a code editor and two programming languages – python and JavaScript (Nodejs). Ten human brain images of different sizes from 5 KB to 96 KB were acquired and stored in a system folder. Figure 4 shows the user interface designed to input an image into the system. The user clicks the "encrypt" or "decrypt" button to complete the image encryption process.


Maxima Image Encryption

Click the button 'encrypt' or 'decrypt' as required

Execution Time(ms):

Input Byte:

Output Byte:

Memory Usage:

MSE:

PSNR:

Decryption Time:

Figure 4: The User Interface

Gaussian filtering was used to convert the image to grayscale using Python programming. Filtering gradually reduced the sensitivity of the images to noise. The result in Figure 5 shows the stages of image Preprocessing. Feature extraction and image optimization were performed with ACO before encryption with DES at the nodes and then decrypted to check the performance compared to conventional DES. The encryption key was not provided in the user interface for ease of use, protection against unauthorized users and ease of testing. The algorithm was implemented so that the image could be iteratively processed to better evaluate its effectiveness.

Figure 6a shows the features extracted from the images which are edges and Figure 6b shows the image optimization using the Ant Colony Optimization algorithm which was implemented in Python programming. This is part of the enhanced DES algorithm process. This distinguishes the enhanced DES from the existing DES (which performs feature extraction on the image itself before encrypting and decrypting the image).

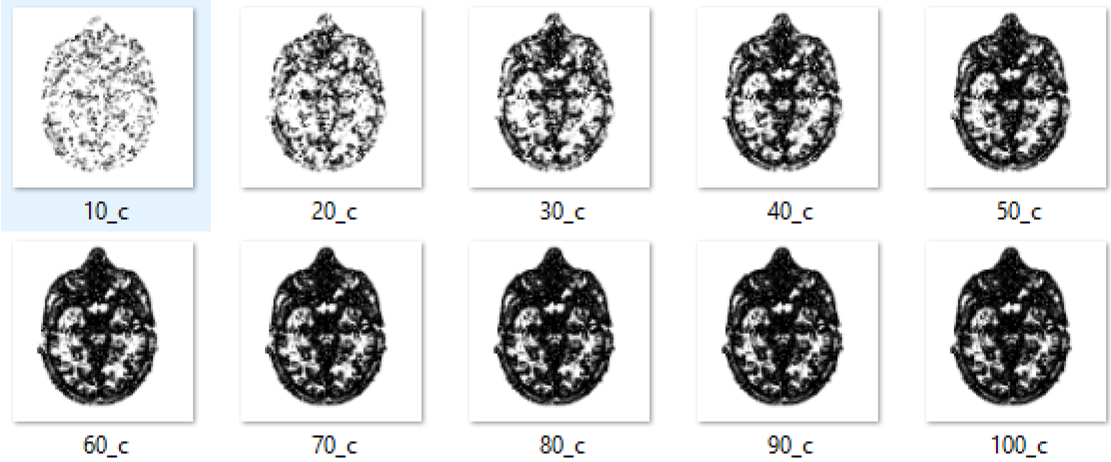


Figure 5: Image Pre-processing Results

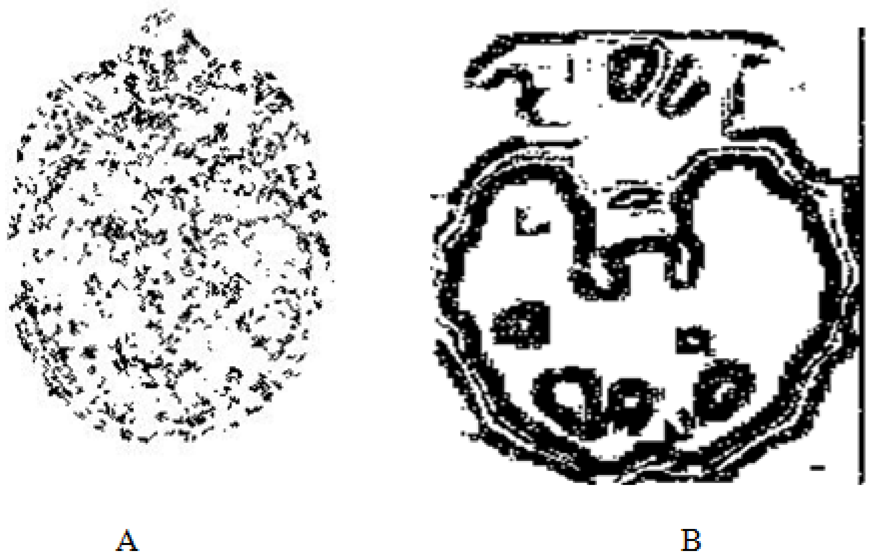


Figure 6: (A) Feature Extraction by ACO. (B) Image Optimization by ACO

The results of the enhanced Data Encryption Standard (DES) algorithm concluded that the computation time and decryption time increased as the size of processed images increased. Therefore, the greater the number and size of the image, the more time the computer takes to complete the implementation process.

4.2 Results of Evaluation of the Enhanced DES and Existing DES

Tables 1 and 2 show the performance evaluated in terms of computational time, input bytes, output bytes, memory usage (i.e. memory utilization), Mean Squared Error (MSE), and Peak Signal-to-Noise Ratio (PSNR) before iterations. Key size (also known as key strength) and decryption time were added to evaluate the effectiveness and efficiency of the improved algorithm. While the key size determines the strength of the algorithm against attacks, the decrypted time showed the time required to decrypt an encrypted image.

Results presented show that the Data Encryption Standard with Ant Colony Optimization was estimated to provide an improvement to the existing Data Encryption Standard. The ACO searches for the optimal feature in the image using an ant transversal behavior to look for the fastest way that enhances the computational time to carry out the entire process.

Table 1: Performance Evaluation Results of DES without ACO before Iteration

Images	Computational Time (Seconds)	Input Byte (Byte)	Output Byte (Byte)	Memory Usage (Byte)	Key Size (Byte)	MSE	PSNR (dB)	Decryption Time (Seconds)
A	32.0765	5967	5968	20988672	64	1.0	48.1308	63.8947
B	40.1714	14615	14616	22515712	64	1.0	48.1308	72.9435
C	48.1849	24656	24664	23900160	64	6.4	39.0999	104.0042
D	57.3483	34564	34568	24498176	64	1.6	42.1102	175.7900
E	69.8216	45960	45968	25382912	64	6.4	39.0999	239.9489
F	74.3653	55176	55184	24955776	64	6.4	39.0999	295.6315
G	86.7640	67440	67448	27710144	64	6.4	39.0999	327.8485
H	94.8045	77062	77064	25017344	64	4.0	45.1205	352.7974
I	103.9143	90243	90248	29132032	64	2.5	41.1411	396.7547
J	115.1734	98413	98416	28323040	64	9.0	43.3596	417.0644

Table 2: Performance Evaluation Results of DES with ACO before Iteration

Images	Computational Time (Seconds)	Input Byte (Byte)	Output Byte (Byte)	Memory Usage (Byte)	Key Size (Byte)	MSE	PSNR (dB)	Decryption Time (Seconds)
A	3.0506	5967	3224	41652800	64	0.036	63.3493	2.7378
B	3.0869	14615	4768	43769088	64	0.025	61.1411	2.7439
C	3.1289	24656	5080	44443904	64	0.064	69.0999	2.8740
D	3.1447	34564	5352	45067520	64	0.010	58.1308	2.9939
E	3.8133	45960	5532	45395200	64	0.090	73.3596	3.4370
F	3.9134	55176	5788	46878528	64	0.056	65.3493	3.4651
G	3.9341	67440	5920	41119744	64	0.092	73.5896	3.7929
H	4.0569	77062	6020	40701952	64	0.089	71.2496	4.2039
I	4.0872	90243	6232	41025923	64	0.013	59.1548	4.4419
J	4.2272	98413	6481	43312731	64	0.018	60.2708	5.1082

Also, the output bytes of the image produced by enhanced DES after decryption is lower when compared with the existing DES. Furthermore, although the root mean square error was lower, the peak signal-to-noise ratio was higher for both the existing DES and the enhanced DES. However, the enhanced DES had a lower root mean square error and a higher peak signal-to-noise ratio. This was an improvement over the current DES because the lower the MSE (i.e., closer to zero) and the higher the PSNR, the better the image quality. However, the memory usage of the enhanced DES was higher compared to the existing DES algorithm. This was due to the merger of the ACO with the existing DES. This resulted in a significant increase in usable memory in all iterations, as shown also in Tables 3 to 6.

.

Table 3: Performance Evaluation Results of DES without ACO at 50 Iteration

Images	Computational Time (Seconds)	Input Byte (Byte)	Output Byte (Byte)	Memory Usage (Byte)	Key Size (Byte)	MSE	PSNR (dB)	Decryption Time (Seconds)
A	43.9663	5967	5968	29682048	64	1.0	48.1308	64.2506
B	51.3546	14615	14616	31226240	64	1.0	48.1308	74.5357
C	57.1843	24656	24664	30833024	64	6.4	39.0999	114.7951
D	64.3254	34564	34568	31414656	64	1.6	42.1102	179.3677
E	75.8697	45960	45968	30620032	64	6.4	39.0999	267.2073
F	83.5605	55176	55184	30390656	64	6.4	39.0999	323.5999
G	94.8261	67440	67448	30599552	64	6.4	39.0999	344.8304
H	102.9778	77062	77064	30493056	64	4.0	45.1205	382.1176
I	121.4986	90243	90248	30906752	64	2.5	41.1411	395.2116
J	155.7328	98413	98416	30624128	64	9.0	43.3596	430.0781

Table 4: Performance Evaluation Results of DES with ACO at 50 Iteration

Images	Computational Time (Seconds)	Input Byte (Byte)	Output Byte (Byte)	Memory Usage (Byte)	Key Size (Byte)	MSE	PSNR (dB)	Decryption Time (Seconds)
A	3.6684	5967	3224	48967680	64	0.036	63.3493	3.4585
B	3.7469	14615	4768	50357888	64	0.025	61.1411	3.5406
C	3.9563	24656	5080	59936000	64	0.064	69.0999	3.5811
D	3.9731	34564	5352	51099264	64	0.010	58.1308	3.6308
E	4.2099	45960	5532	50927232	64	0.090	73.3596	3.7553
F	4.2102	55176	5788	51369600	64	0.056	65.3493	3.8054
G	4.2172	67440	5920	50583168	64	0.092	73.5896	3.8635
H	4.6312	77062	6020	50005632	64	0.089	71.2496	4.0705
I	4.7035	90243	6232	50751104	64	0.013	59.1548	4.1842
J	4.7271	98413	6481	50251392	64	0.018	60.2708	4.4581

Table 5: Performance Evaluation Results of DES without ACO at 100 Iteration

Images	Computational Time (Seconds)	Input Byte (Byte)	Output Byte (Byte)	Memory Usage (Byte)	Key Size (Byte)	MSE	PSNR (dB)	Decryption Time (Seconds)
A	54.2317	5967	5968	32482603	64	1.0	48.1308	70.7216
B	65.9583	14615	14616	34320656	64	1.0	48.1308	84.9422
C	78.0122	24656	24664	30772160	64	6.4	39.0999	116.4212
D	85.2712	34564	34568	32361361	64	1.6	42.1102	188.9281
E	105.5438	45960	45968	31853080	64	6.4	39.0999	280.4231
F	126.0233	55176	55184	34234552	64	6.4	39.0999	334.5432
G	146.0384	67440	67448	31360144	64	6.4	39.0999	390.3427
H	168.3245	77062	77064	30157344	64	4.0	45.1205	410.2315
I	187.3432	90243	90248	32131332	64	2.5	41.1411	459.8121
J	206.6544	98413	98416	35938352	64	9.0	43.3596	469.8734

Table 6: Performance Evaluation Results of DES with ACO at 100 Iteration

Images	Computational Time (Seconds)	Input Byte (Byte)	Output Byte (Byte)	Memory Usage (Byte)	Key Size (Byte)	MSE	PSNR (dB)	Decryption Time (Seconds)
A	3.7334	5967	3224	51342820	64	0.036	63.3493	3.5021
B	3.8912	14615	4768	52543203	64	0.025	61.1411	3.7424
C	4.0847	24656	5080	50443904	64	0.064	69.0999	3.9210
D	4.2938	34564	5352	52446105	64	0.010	58.1308	4.1210
E	4.4193	45960	5532	59332264	64	0.090	73.3596	4.3612
F	4.6823	55176	5788	52561326	64	0.056	65.3493	4.5341
G	4.8234	67440	5920	60319342	64	0.092	73.5896	4.7784
H	4.9252	77062	6020	62451351	64	0.089	71.2496	4.8617
I	5.0831	90243	6232	61025923	64	0.013	59.1548	5.0362
J	5.1123	98413	6481	63312731	64	0.018	60.2708	5.1008

4.3 Comparison of the Existing Data Encryption Standard and Enhanced Data Encryption Standard

Figures 7 - 9 show the computational time comparison between the existing Data Encryption Standard algorithm and the improved Data Encryption Standard algorithm.

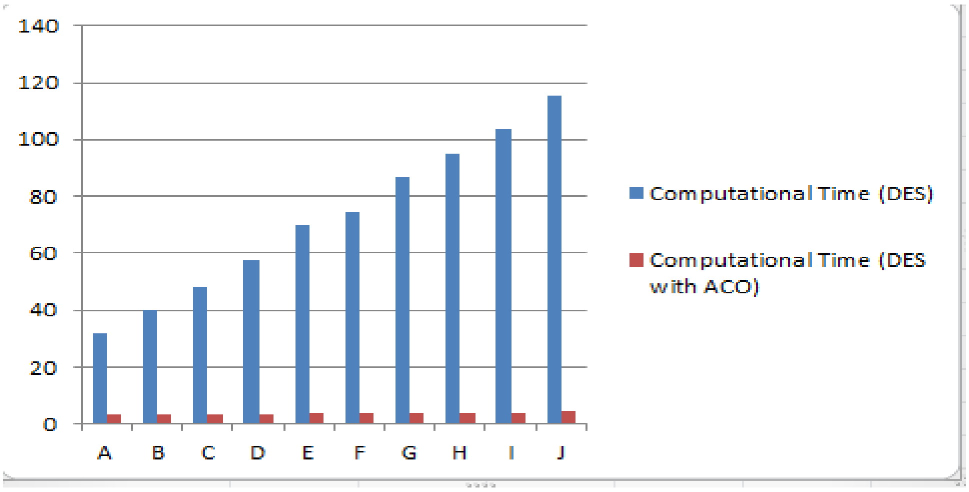


Figure 7: Computational time for DES and DES with ACO before iteration

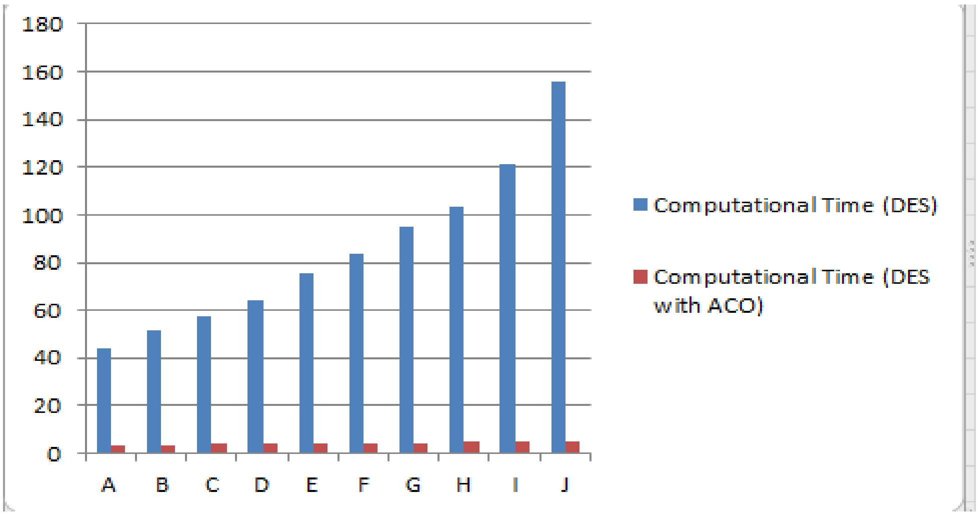


Figure 8: Computational time for DES and DES with ACO at 50 iterations

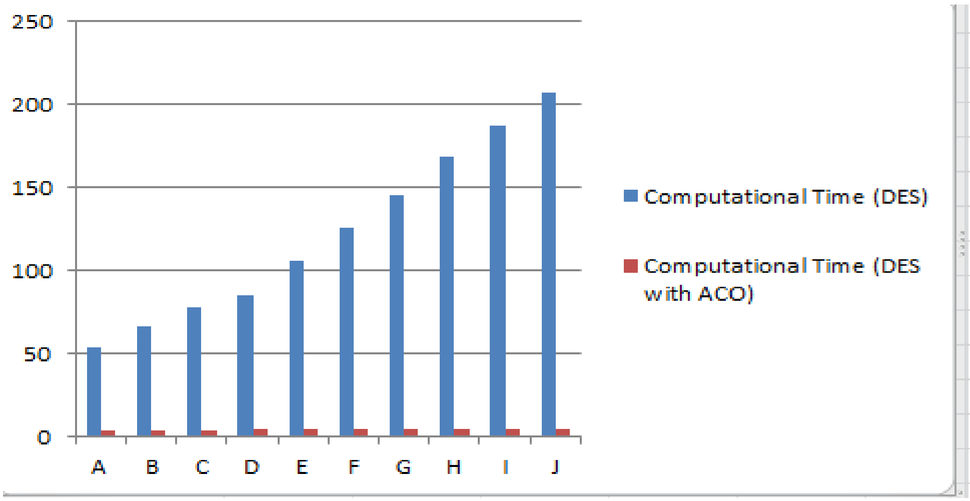


Figure 9: Computational time for DES and DES with ACO at 100 iterations

4.4 Discussion

The chart results obtained after comparing using the existing Data Encryption Standard and the enhanced Data Encryption Standard algorithm on ten different sizes of human brain images at before iteration and at 50 and 100 iterations show that the enhanced DES performed better at 94.95% in computational time compared to the existing DES.

Based on the results obtained before iterations, DES with ACO outperformed DES without ACO by reducing computation and decoding time. For example, with an image size of 5KB and 96 KB, the computation time of normal DES was 32.0765 seconds and 115.1734 seconds, respectively, while the computation time of DES WITH ACO was 3.0506 seconds and 4.2272 seconds for the same image sizes. Similar differences were observed at fifty (50) and one hundred (100) repetitions. Table 3 - 6 presented the evaluation results showing the differences, while Figure 7 - 9 presented graphs showing the performance evaluation differences between the existing DES and DES WITH ACO algorithms before and during the iterations.

Also, the output bytes of the image produced by DES WITH ACO after decryption is lower when compared with ordinary DES. At image size of 14KB before iteration, ordinary DES output byte was 14616 bytes (which is closer to the input byte) while DES WITH ACO was 4KB. The reduction in size was due to the image preprocessing, feature extraction and optimization that was done on the image by DES WITH ACO before encryption.

Furthermore, although the root mean square error was lower, the peak signal-to-noise ratio was higher for both the existing DES and the enhanced DES. However, the enhanced DES had a lower root mean square error and a higher peak signal-to-noise ratio. This was an improvement over the current DES because the lower the MSE (i.e., closer to zero) and the higher the PSNR, the better the image quality.

However, the memory usage of the enhanced DES was higher compared to the existing DES algorithm. This was due to the merger of the ACO with the existing DES. This resulted in a significant increase in usable memory in all iterations, as shown in all tables. Also, the key strength or size of existing DES and enhanced DES remains 64bit in all iterations. This means that the enhanced DES was still vulnerable to brute force attacks. This corresponds with the findings of Semwal and Sharma [15] which suggested that each encryption algorithm has its own strong and weak points.

5. CONCLUSION

In this paper, an enhanced Data Encryption Standard algorithm was implemented for securing medical images by introducing the Ant Colony Optimization technique with an existing DES. Also, the results obtained were compared after using the existing Data Encryption Standard and the enhanced Data Encryption Standard algorithm on ten different sizes of human brain images based on some performance evaluation metrics. The algorithm was implemented with the use of Visual Studio code as the code editor and two programming languages – python and JavaScript (nodejs). This has provided an improvement to the existing DES algorithm in solving image security problems faster. This work has scope for future application in cyber security, image security, pattern recognition and handling of big data. Based on the results, it can be concluded that the enhanced DES performed better than the existing DES. Hence, the importance of the study to improve the existing Data Encryption Standard (DES) has actually been achieved as a result of the better performance in securing brain images at lower computational time. Future research can improve the robustness of the algorithm's keys and reduce memory usage.

REFERENCES

[1] Abbas Z. H., Maisa’a A. A. K. (2023). Medical image encryption using multi chaotic maps. *TELKOMNIKA Telecommunication Computing Electronics and Control*, Vol. 21, No. 3, June 2023, pp. 556~565, ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v21i3.24324

[2] Ahmed M., Tarfa H., Charlie O., and Robert D., (2013). Improving the Security of the Medical Images. *International Journal of Advanced Computer Science and Applications*, Vol. ***4***, No. 9

[3] Ajala F. A., Fenwa O. D., Ojebamigbe V. I., (2018). Development of an Enhanced Fuzzy C-Means Algorithm for Medical Image Segmentation Using Ant Colony Optimization, *International Conference on Applied Research In Engineering, Science and Technology in Institute for European Studies (IES)*, Pleinlaan 5, Floor- B-1050, Brussels between 14th and 15th of September, 2018. Belgium.

[4] Ankita, V., Paramita, G., and Sunita, M. (2016). Comparative Study of Different Cryptographic Algorithms. *International Journal of Emerging Trends & Technology in Computer Science*, **5** (1): 58-63.

[5] Asmaa E. A. (2019). Encryption Algorithms for Data Security in Local Area Network, *Unpublished PhD Thesis*, Department of Computer Engineering and Science, Florida Institute of Technology.

[6] Chaudhary, S., Suthar, F. and Joshi, N. K. (2020). Comparative Study between Cryptographic and Hybrid Techniques for Implementation of Security in Cloud Computing. *In: Pant M., Sharma T., Basterrech S., Banerjee C. (eds) Performance Management of Integrated Systems and its Applications in Software Engineering. Springer, Singapore*. pp. 127–135.

[7] Goodman, D. E., Boggess, L., and Watkins, A. (2002). Artificial immune system classification of multiple-class problems. *Proceedings of Artificial Neural Network Engineering.* ANNIE 2, 179–183.

[8] Grabbe, O. (2006). The DES Algorithm Illustrated, *Laissez Faire City Times*, Vol 2, No. 28 Retrieved from https://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.html on 19th January, 2021

[9] Jamil, A. S. and Rahma, A. S. (2023). Cyber Security for Medical Image Encryption using Circular Blockchain Technology Based on Modify DES Algorithm. *International Journal of Online & Biomedical Engineering*, Vol. 19 Issue 3, p99-112.

[10] Kalyani, P. K., and Neha V. N. (2016). Comparative Analysis of Encryption

Algorithms for Various Types of Data Files for Data Security. *International Journal of Scientific Engineering and Applied Science (IJSEAS)* – Volume **2**, Issue 2.

[11] Kavitha P. K., and Saraswathi V. (2019). A Survey on Medical Image Encryption. *Conference: 1st International Conference on Applied Soft Computing Techniques At: Kalasalingam University, Krishnankoil* 10.32628/ICASCT2501.

[12] Muhammed Y. I., Rabab A. A., Mehrbakhsh N., Ashwaq A., Abdullah A., Ahmed O. A., Linah S.,Reem O., Shahla A., (2021). Medical image processing and COVID-19: A literature review and bibliometric analysis, *Journal of Infection and Public Health* 15 (2022) 75–93.

[13] Ramson, S.J., Raju, K.L., Vishnu, S., Anagnostopoulos, T. (2019). Nature inspired optimization techniques for image processing—A short review. *In Nature Inspired Optimization Techniques for Image Processing Applications; Springer*: Cham, Switzerland

[14] Rendell N. and Sheel M. (2022). Expert perspectives on priorities for supporting health security in the Pacific region through health systems strengthening. *PLOS Global Public Health.* 2(9): e0000529. 10.1371/journal.pgph.0000529.

[15] Semwal, P., and Sharma, M. K. (2017). Comparative study of different cryptographic algorithms for data security in cloud computing. *International Journal on Emerging Technologies (Special Issue NCETST-2017).* **8**(1): 746-750

[16] Shashi M. S. and Rajan M., (2011). Comparative Analysis of Encryption Algorithms For Data Communication, *International Journal of Computer Science and Technology, 0976-8491*

[17] Shruthishree S.H, and Harshvardhan T., (2017). A review paper on medical image processing. *International Journal of Research - Granthaalayah*, 5(4) RACSIT, 21-29. https://doi.org/10.29121/granthaalayah.v5.i4RACSIT.2017.3344.

[18] Singh H., Gupta C. and Shrivastava R. (2022). A Review on Various Image Encryption Technique and Challenges in the Current Scenario. *International Journal of Creative Research Thoughts (IJCRT),* Volume 10, Issue 9, ISSN: 2320-2882

[19] Sinha, G. R. (2013). *Medical Image Processing: An Overview and Research Scope* [powerpoint slides]. *Faculty of Engineering & Technology, Shri Shankaracharya Technical Campus, Bhilai, India*. 10.13140/RG.2.1.1101.5766.

[20] Vijini, M. (2017). Introduction to Genetic Algorithms — Including Example Code. Retrieved from https://towardsdatascience.com on 19th January, 2021

[21] Wang, D., Tan, D., and Liu, L. (2018). Particle swarm optimization algorithm: an overview. *Soft Comput* 22, 387–408, Retrieved from https://doi.org/10.1007/s00500-016-2474-6 on 19th January, 2021