Results in Control and Optimization 18 (2025) 100515

Contents lists available at ScienceDirect

Results in Control and Optimization

journal homepage: www.elsevier.com/locate/rico

Diegif: An efficient and secured DICOM to EGIF conversion framework for confidentiality in machine learning training

Abdullah Al Siam [a], Md Maruf Hassan [b], Md Atikur Rahaman [c], Masuk Abdullah [d,*]

[a] Department of Software Engineering, Daffodil International University, Daffodil Smart City, Birulia, Dhaka, 1216, Dhaka, Bangladesh
[b] Department of Computer Science & Engineering, Southeast University, Tejgaon, Dhaka, 1208, Bangladesh
[c] School of Economics and Management, Jiujiang University, 551 Qianjin Donglu, Jiujiang, 332005, Jiangxi, PR China
[d] Department of Vehicles Engineering, Faculty of Engineering, University of Debrecen, Ótemető street. 2-4, 4028, Debrecen, Hungary

ARTICLE INFO

Keywords:
DICOM
EGIF
ML
AES
Encryption
Decryption
Security

ABSTRACT

Medical imaging plays a critical role in contemporary healthcare, although it confronts issues relating to storage, security, and confidentiality in machine learning-based diagnostic systems. The proposed framework, *Diegif*, presents an efficient and safe mechanism for converting DICOM (Digital Imaging and Communications in Medicine) data into EGIF (Encrypted Graphics Interchange Format) files to overcome these challenges. The framework comprises four key components: (1) converting DICOM files to GIF format with encryption, (2) decrypting EGIF files for processing, (3) enabling confidentiality-preserving machine learning training using EGIF data, and (4) facilitating physician diagnosis and report generation based on trained machine learning models. The *Diegif* framework aims to enhance storage efficiency by decreasing file sizes by 66.32%, thereby improving data transport efficacy and cloud storage affordability while preserving strong encryption for data confidentiality. Pseudocode algorithms are provided for each phase, ensuring reproducibility and transparency. This paper illustrates the framework's potential to medical image processing, secure storage, and AI-driven diagnostic functions in healthcare.

1. Introduction

Medical imaging plays a crucial role in diagnosing and monitoring various diseases [1]. The DICOM format is the standard for storing and transmitting medical images in the healthcare industry. However, many image processing and analysis tools cannot directly handle DICOM files [2]. Converting DICOM files to a more widely compatible format like GIF addresses this limitation and offers several advantages [3]. GIFs allow healthcare professionals to easily share visual information with colleagues, specialists, and patients through diverse communication channels. This makes the development of a reliable DICOM-to-GIF conversion framework essential.

DICOM stores high-resolution medical images and related data, which are critical for clinical use. In contrast, the GIF format is commonly used for simple animations and graphics [4]. While DICOM ensures the quality of medical images, its large file sizes demand significant storage resources [5].

Current DICOM storage solutions are often inefficient, leading to high storage costs. Additionally, many security mechanisms depend on external encryption methods, leaving the data vulnerable during processing. A robust conversion framework can address these challenges by improving compatibility, reducing storage requirements, and enhancing data security.

* Corresponding author.
E-mail address: masuk@eng.unideb.hu (M. Abdullah).

https://doi.org/10.1016/j.rico.2025.100515
Received 28 November 2024; Received in revised form 25 December 2024; Accepted 1 January 2025
Available online 25 January 2025

This paper presents an efficient method for converting DICOM files to EGIF format, preserving image quality while speeding up the conversion process. The conversion enhances collaboration, enabling remote consultations that improve decision-making and patient care. Converting DICOM to GIF format also improves data accessibility and interoperability [6]. Overall, this method enhances communication among medical professionals, allowing for quicker sharing of information and better healthcare delivery. It also benefits patients by ensuring that images are distributed efficiently. In bioinformatics, this conversion process has significant advantages.

Our goal was to develop *Diegif*, a framework for converting DICOM to EGIF, which would improve accessibility and need less storage, thereby improving medical image management. This architecture showed notable storage savings, improved data security, and successfully lowered conversion times. Despite obstacles, we put solutions in place that protect patient data security and preserve image quality. The following is a summary of the main contributions of this paper:

- Constructed an approach for converting DICOM to EGIF that improves the readability and accessibility of medical images.
- Decreased storage space by 66.32%, and secure system using this framework.
- Lower cloud storage costs will be achieved by improved data transport efficiency, smaller file sizes, and optimized cloud storage.

The remainder of this paper is structured into six sections. The literature review is presented in Section 2. Section 3 outlines the methodology, which is subdivided into four subsections: image management and processing, algorithm development, confidential machine learning training, and user interaction and application. Section 4 provides a performance analysis of the system. The limitations and future directions of the proposed system are discussed in Section 5. The paper concludes with Section 6.

2. Related work

The challenge of securing and optimizing medical imaging data has been explored through various frameworks and methodologies, yet significant gaps remain in storage efficiency, encryption integration, and compatibility with machine learning workflows.

DICOM is a widely used international standard for storing and transmitting medical images in hospitals and clinics. They review various DICOM technologies that can be used for medical image processing. Compression, enhancement, segmentation, and registration of images are some of the technologies mentioned [7].

People all over the world are familiar with the Digital Imaging and Communications in Medicine (DICOM) standard, which was created specifically for the storage and transmission of medical images. As a result, almost all of the outputs of magnetic resonance imaging (MR), computerized tomography (CT), digital subtraction angiography (DSA), and ultrasonography (US) are saved in DICOM format [8]. NS Ujgare et al. proposed an algorithm for views and converts. DICOM image files are converted into bmp, png, and jpeg standard images, which must be viewable with common image viewing software and be small in size [9].

Current frameworks often employ external encryption methods to secure sensitive medical data. For example, Rotor64-based cryptography has been proposed for secure data transmission [10]. While this approach enhances data confidentiality, its application is limited due to scalability challenges and its post-processing nature, which exposes intermediate data to potential breaches. Similarly, asymmetric encryption methods like RSA have been utilized but are computationally intensive and less effective for large-scale datasets [11].

The *Diegif* framework addresses these shortcomings by embedding AES encryption directly within the DICOM-to-EGIF conversion process. This integration ensures end-to-end security, minimizing exposure during intermediate stages and optimizing performance for large datasets.

Shakya et al. presented a thorough examination of how to optimize the storage of DICOM images through the application of statistical texture analysis and image compression techniques. The four most researched compression algorithms DCT, DWT, FCA, and VQA were examined in this analysis along with their possible effects on main texture parameters such as contrast, CORRELATION ASM, and IDM [12].

WJ Xue et al. used the technique of Java Applet to realize the support of DICOM image in an ordinary Web browser, thereby expanding the processing function of medical image [13]. One other paper [14] presented a Web-based DICOM viewer that was entirely developed with web technology, namely HTML5 and JavaScript.

X Li et al. explained the significance of converting DICOM medical images to the NIfTI format for the purpose of analyzing neuroimaging data. Although it is a widely used format for storing and transmitting medical images, the DICOM format is not ideal for data analysis. A more adaptable and effective format for data analysis is the NIfTI format, which is widely supported by neuroimaging software [15].

DR Varma et al. explained the difficulties in managing DICOM images in a radiology practice. Although DICOM is a common format for storing and transmitting medical images, managing it can be challenging. The paper offers radiologists several pointers and tricks to better manage their DICOM images. This will make it simpler to view and understand the images [16].

X Lu et al. created a method for converting DICOM multi-frame images to multimedia format to improve the efficiency and quality of medical diagnosis. The proposed method has the potential to increase the accuracy and efficiency of medical diagnosis. Medical professionals can easily view and analyze DICOM multi-frame images after converting them to multimedia format, leading to faster and more accurate diagnoses [17].

M Bomewar et al. developed new ways for converting DICOM format to BMP (Bitmap Image file) format, dividing data into two halves, and embedding patient data in an image in odd-numbered memory locations of image pixel values. This study focuses on

calculating the data hiding capacity, compression ratio, mean square error (MSE), and peak-to-signal noise ratio (PSNR) of medical images [18].

A Golubev et al. explained the issues and potential solutions for optimizing medical picture storage, providing consistent and secure access, and storing massive amounts of data with varying degrees of access using a distributed warehouse. A "DICOM Network" project was created to address these issues for various system players based on their unique roles [19].

O Diaz et al. described the future expansion of the Internet to perform large-scale medical image data analysis with the help of artificial intelligence. It stresses the aspects concerning medical images that to successfully apply an AI solution, the medical images should be properly preprocessed before being used, including the processes like de-identification, data curation, storage, and annotation. It also covers the possibility of the employment of open-access tools to perform these tasks and the increase of medical image databases [20].

As addressed [21] to build an artificial intelligence-driven image viewer with the intention of enhancing the accuracy of medical imaging. Denoising CT scans, converting two-dimensional images to three-dimensional ones, and automating image segmentation via the use of deep learning and GANs are some of the key developments that have dramatically improved diagnostic accuracy and reduced the amount of labor that radiologists have to do.

This article [22] investigated the extensive implementation of the DICOM standard in the field of medical informatics, particularly in the field of radiation (DICOM RT) and PACS applications. Not only does it illustrate the difficulties associated with system connectivity and compatibility testing, but it also emphasizes the need to simplify DICOM compliance.

The study presented tools for DICOM-based lung CT image analysis, focused on visualization and lung tissue segmentation. Through the use of pixel matrices in Hounsfield units and the Watershed method, it can effectively segment lung tissue, which significantly aids in the early diagnosis of illness as shown Ortega [23].

S Shivshankar et al. presented DICOM's importance in medical imaging and its compatibility with HL7 standards, emphasizing its combined influence on healthcare data interchange and increased patient care via integrated medical imaging systems [24].

As mentioned [25] to convert neuroimaging data from the DICOM format to the BIDS format, HeuDiConv offers a versatile program designed for this purpose. It supports complicated data formats, enables data administration, and connects with tools like DataLad, making it vital for large-scale neuroimaging processes.

Storage efficiency has been a longstanding concern in medical imaging, with several studies focusing on compression techniques. For example, the JPEG2000 standard has been widely adopted for medical image compression due to its high fidelity and lossy/lossless capabilities [26]. Despite these benefits, such methods often fail to achieve significant reductions for high-resolution datasets without sacrificing quality, particularly for modalities like MRI and CT scans. Moreover, the lack of integration between compression and security measures results in additional overhead.

In contrast, the *Diegif* framework achieves a 66.32% file size reduction while simultaneously encrypting the output files. This dual functionality is particularly advantageous for healthcare systems dealing with large-scale imaging workflows, offering both efficiency and security without the need for separate processing pipelines.

Existing methodologies often fail to provide seamless integration with machine learning (ML) workflows. For example, traditional DICOM formats require extensive preprocessing before they can be utilized for ML model training [27]. Additionally, unencrypted data poses significant privacy risks when shared across collaborative ML systems, such as federated learning frameworks. Current solutions lack mechanisms for securely preprocessing and transmitting data within such environments.

The *Diegif* framework addresses this gap by converting DICOM files into EGIF, a format optimized for ML workflows. Integrating encryption within the conversion process ensures that data confidentiality is maintained throughout the ML pipeline. This capability is particularly valuable for confidential machine learning training, where secure preprocessing is a critical requirement.

3. Methodology

This paper presents *Diegif*, a system for converting DICOM files to EGIF files. This approach simplifies medical image analysis and visualization for machine learning (ML) training. The process starts by reading and normalizing the image data, then converting it to GIF format, and securely storing the images. To ensure privacy, the images are encrypted using AES encryption, with key access managed to protect sensitive data.

The ML training system is designed to securely train models while maintaining data privacy. It includes steps like decryption, preprocessing, feature extraction, analysis, and model building. The system also allows users to interact with the images, train models, make predictions, involve clinicians in decision-making, and generate detailed reports.

The methodology follows a clear, algorithmic structure, ensuring efficient user interaction, model training, prediction generation, and report creation in the medical field. Fig. 1 shows an overview of the proposed methodology. The framework is built in four key steps, each supported by pseudocode to ensure clarity and reproducibility.

To validate the practicality of the Diegif framework, a real-world example is presented using a dataset of 1000 high-resolution CT scans in DICOM format, each averaging 5209.3 MB. The process includes pixel normalization to enhance compatibility, conversion to GIF format, which reduces file size by approximately 40%, and AES encryption with a 256-bit key for robust data security. The encrypted EGIF files were successfully integrated into a TensorFlow-based ML model for cancer detection. This approach achieved a total storage reduction of 66.32%, ensuring scalability, security, and efficient handling of large-scale datasets.

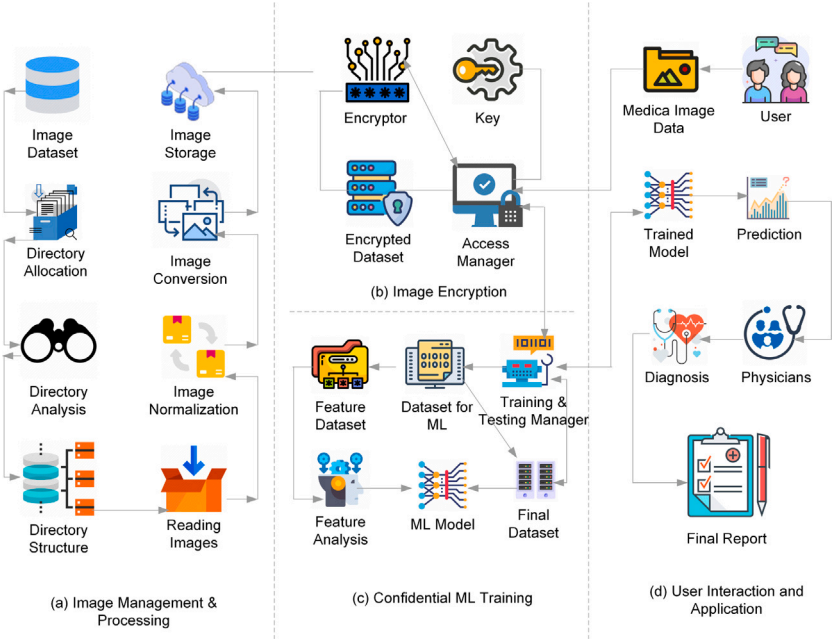

Fig. 1. The overview of the proposed Diegif methodology.

3.1. Data collection

We demonstrated the effectiveness of *Diegif* by implementing and testing it on a dataset from the LIDC-IDRI database, producing aesthetically accurate GIF renderings of medical images. Lung nodules on CT scans are fully referenced in the LIDC-IDRI database, which contains data from the Image Database Resource Initiative (IDRI) and the Lung Image Database Consortium (LIDC) [28]. The thoracic computed tomography (CT) scans in the LIDC-IDRI image collection are marked-up annotated lesions from diagnostic and lung cancer screening CT scans. The development, teaching, and evaluation of computer-assisted diagnostic (CAD) methods for early detection and diagnosis are all supported by this global online resource. The TCIA team encourages users to review pylidc and the Standardized representation of the TCIA LIDC-IDRI annotations using DICOM (DICOM-LIDC-IDRI-Nodules) before creating custom tools to analyze the XML version [29]. Researchers can access a sizable collection of de-identified cancer medical images through the free public resource known as The Cancer Imaging Archive (TCIA) [30]. The types of cancer, imaging techniques, and areas of research are used to group the images into collections. Additionally, TCIA offers image-related auxiliary information, such as patient outcomes, treatment information, genomics, and professional analyses. Researchers studying new cancer detection, diagnosis, and treatment approaches can benefit greatly from the TCIA. To train machine learning algorithms, create new imaging techniques, and find new cancer biomarkers, TCIA images, and data can be used [31].

3.2. Implementation tools and frameworks

The *Diegif* framework was developed using a collection of sophisticated tools and libraries to guarantee security, efficiency, and smooth connection with healthcare and machine learning systems. The framework is built in *Python*, selected for its robust ecosystem, comprehensive library support, and seamless interaction with machine learning and data processing processes. For encryption, *PyCryptodome* is used to implement AES encryption, providing the safe handling of medical imaging data during conversion. The framework also leverages *pydicom* for reading, writing, and converting DICOM files, enabling it to access and handle medical pictures efficiently. Additionally, *NumPy* is leveraged for high-performance numerical calculations, allowing tasks like pixel normalization and matrix transformations. These instruments collaborate to ensure the safe and rapid processing of medical pictures. They make it easy to convert from DICOM to EGIF in a robust healthcare and machine learning context. For machine learning integration, *TensorFlow* is utilized to train models on encrypted EGIF data, guaranteeing compatibility with secure processes. These technologies jointly facilitate the safe and fast processing of medical images, supporting the DICOM to EGIF conversion within a comprehensive healthcare and machine learning context.

The use of these tools ensures that the *Diegif* framework maintains high performance while adhering to stringent security and efficiency requirements. This setup also makes the framework accessible to researchers and practitioners in healthcare and machine learning domains.

3.3. Image management & processing

The three most important phases are included in the image management and processing system. In the first step of the process, DICOM picture datasets are saved in a specific input directory, and a separate output folder is created to ensure that the original images are not altered. The system will next scan and evaluate the images, then normalize the pixel values to ensure compliance with standard formats such as GIFs while preserving diagnostic quality. Finally, the DICOM images are converted into the GIF format, then stored in the output directory, and lastly, their arrangement makes access easy. This process improves the effectiveness and usability of medical imaging operations. This procedure adheres to a predetermined procedure in order to guarantee the appropriate allocation, organization, and transformation of the medical pictures that are saved in the DICOM format. Detailed explanations of its operation are provided in the following algorithms:

3.4. Algorithm development

Diegif: An Efficient and Secured DICOM to EGIF Conversion Framework presents algorithms for both the DICOM to EGIF conversion and the decryption of the EGIF file. The first algorithm focuses on the DICOM to EGIF conversion procedure. It outlines the step-by-step process for extracting useful information from DICOM files, such as patient metadata, picture data, and any associated annotations.

Algorithm 1 Pseudocode DICOM to EGIF Conversion Algorithm

Require: D_i: input directory, S_t: Stack, P_{root}: Path to Root, D_o: Output Directory, N_f: Num of files
if not D_o exists **then**
  $D_o \leftarrow$ create(loc, $name$)
end if
$S_t \leftarrow D_i$
while not empty(S_t) **do**
  $D_i \leftarrow$ pop(S_t)
  $D_o \leftarrow$ create(D_o, $name$)
  $D_i \leftarrow$ file_list(D_i)
  for $i = 1$ to N_f **do**
    if file_list(i) ends with '.dcm' **then**
      file_list(i) $\leftarrow$ Normalize($0, 255$)
      file_list(i) $\leftarrow$ Replace('.dcm', '.gif')
      save($file\_list(i)$, D_o)
      Encrypt(file_list(i), AES)
      Append(D_o, file_list(i))
      Remove(file_list(i), '.gif')
    end if
  end for
end while
Exit

This technique ensures that important data is preserved while converting photos to the EGIF format. It also solves image resolution, compression, and color mapping concerns to improve the visual quality of the converted EGIF images. The second algorithm provided in the study is the EGIF file decryption algorithm. The privacy and security of the patient's data must be protected. *Diegif* uses a safe encryption approach to encrypt EGIF files, ensuring that only authorized users with the necessary decryption keys can access the data contained within the file. The decryption algorithm describes how to securely decrypt the EGIF file and recover the original DICOM data.

The Algorithm 1 offers a method for converting DICOM files to EGIF format. Setting up input and output folders guarantees a mirrored structure for organized processing, hence initiating the high-level workflow of the DICOM to GIF conversion method. Then, ready for conversion, DICOM files are read and their pixel values are standardized to an 8-bit range. The standardized images are converted to GIF style, which increases their accessibility on many devices. Following conversion, AES encryption guards the GIF files to maintain the private medical information. While the original unencrypted files are securely deleted to assure effective photo processing and excellent data security, the encrypted files are kept in the output directory.

3.4.1. Input, output setup and iterating through DICOM files

The algorithm expects multiple input variables, including the input directory D_i, a stack S_t, the path to the root directory represented by P_root, the output directory D_o, and the number of files represented by N_f. It expects a root folder holding the input DICOM files and specifies a separate output folder for storing the resulting GIF files. It verifies whether or not the output directory D_o is there. If it does not exist, it builds it using the "create" function, which accepts the directory's location and name as arguments.

The algorithm generates the stack S_t with the value of the input directory D_i. It searches through the input folder and all of its subdirectories to find DICOM files. The method enters a loop that is repeated until the stack S_t is empty. It pulls the top directory from the stack S_t and assigns it to the input directory D_i during the loop. Using the "create" function, the algorithm generates a new directory in the output directory D_o. It takes as inputs the output directory and name. The method retrieves a list of files from the input directory D_i and stores it in the variable $file_list$. The algorithm then begins a loop that loops over the files in the $file_list$.

3.4.2. Conversion process

The method examines each file to determine if it is a DICOM file by confirming that the file extension is ".dcm". The algorithm normalizes the pixel values from the range of 0 to 255 if the file is a DICOM file. By dividing by the maximum pixel value and then subtracting the minimum pixel value from each pixel, the technique produces a pixel array with values ranging from 0 to 255. The algorithm saves the file in the output directory D_o and changes the ".dcm" extension to ".gif" using the "Replace" function.

With the file and the encryption key as parameters, the algorithm encrypts the file using the AES encryption method. The "Append" function is used by the algorithm to append the encrypted file to the output directory D_o. Use the AES Encryption method to convert DICOM to GIF. The fact that the encryption and decryption processes use the same key indicates that the technique is symmetric.

The AES encryption method uses a variety of mathematical techniques, including substitution, permutation, XOR, matrix multiplication, and modular arithmetic. The `AES.new()` and `encrypt_and_digest()` routines, which offer a streamlined interface for utilizing AES to encrypt data, abstract these specifics away. Using the "Remove" function, the algorithm deletes the original GIF file from the output directory D_o. Once all of the files in the input directory D_i have been processed, the algorithm repeats the loop. The algorithm ends when the loop is finished and the stack S_t is empty. The entire process is illustrated in the flowchart depicted as shown in Fig. 2.

3.4.3. Encryption algorithm

Use the AES Encryption method to convert DICOM to GIF. The fact that the encryption and decryption processes use the same key indicates that the technique is symmetric. It operates on blocks of data and is made up of a variety of rounds, each of which applies a number of changes to the data using a combination of bitwise, permutation, and substitution operations [32].

- **AES Encryption using EAX Mode**:
  In this stage, the AES technique is used for encryption in the EAX (Authenticated Encryption with Associated Data) mode. To create an AES (`cipher`) object, utilize the(`encryption_key`). The `encrypt_and_digest()` method is called on the cipher object, supplying the data to be encrypted (`data`) as the input. An authentication tag (`tag`) and the encrypted data (`encrypted_data`) are returned by the procedure.
- **AES.new(key, mode)**: The `AES.new()` function creates a new instance of the AES cipher object with the given encryption key and mode of operation. The encryption key (`encryption_key`) and the mode of operation (`AES.MODE_EAX`) is used to initialize an AES cipher object called (`cipher`) in the code.
- **encrypt_and_digest(data)**: The method encrypts the supplied data using the specified algorithm and returns the encrypted data along with an authentication tag. The initialized AES cipher object (`cipher`) is used in the code to encrypt a file's content (`data`) and it also creates an authentication tag. The result is returned together with the authentication tag (`tag`) and the encrypted data (`encrypted_data`).

The AES encryption method uses a variety of mathematical techniques, including substitution, permutation, XOR, matrix multiplication, and modular arithmetic. The `AES.new()` and `encrypt_and_digest()` routines, which offer a streamlined interface for utilizing AES to encrypt data, abstract these specifics away.

Attack Scenarios and Resistance: The AES encryption method is designed to withstand a variety of attack scenarios, ensuring robust security for sensitive medical imaging data. Its 256-bit key length provides resistance to brute-force attacks, as the vast number of key combinations (2^{256}) renders such attempts computationally infeasible. Furthermore, AES in EAX mode enhances data security by combining encryption with authentication, ensuring both confidentiality and integrity. The authentication tag generated during the encryption process helps detect any tampering or unauthorized modifications, providing an additional layer of protection. These features make AES particularly effective for securing medical imaging workflows, where data integrity and confidentiality are paramount.

3.4.4. Comparison of encryption methods

To enhance the technical depth of the study, a comparative analysis of AES and other encryption standards commonly used in medical data security is provided (see Table 1). This analysis focuses on speed, security level, and suitability for medical imaging data, offering insights into why AES is ideal for the *Diegif* framework.

AES (Advanced Encryption Standard) AES is commonly considered as the gold standard for encryption, including key lengths of 128, 192, or 256 bits. Its rapid speed makes it excellent for encrypting and decrypting big datasets, such as medical imaging images, without incurring considerable computing cost. AES is very secure, with resilience to known cryptographic attacks, including brute force, because to its huge key space. Its scalability and flexibility make it suited for current applications, notably in healthcare, where both security and efficiency are critical [33].

DES (Data Encryption Standard) DES, previously a famous encryption technology, employs a constant key length of 56 bits. While relatively quick, its small key length renders it particularly subject to brute force assaults, leaving it outdated for modern

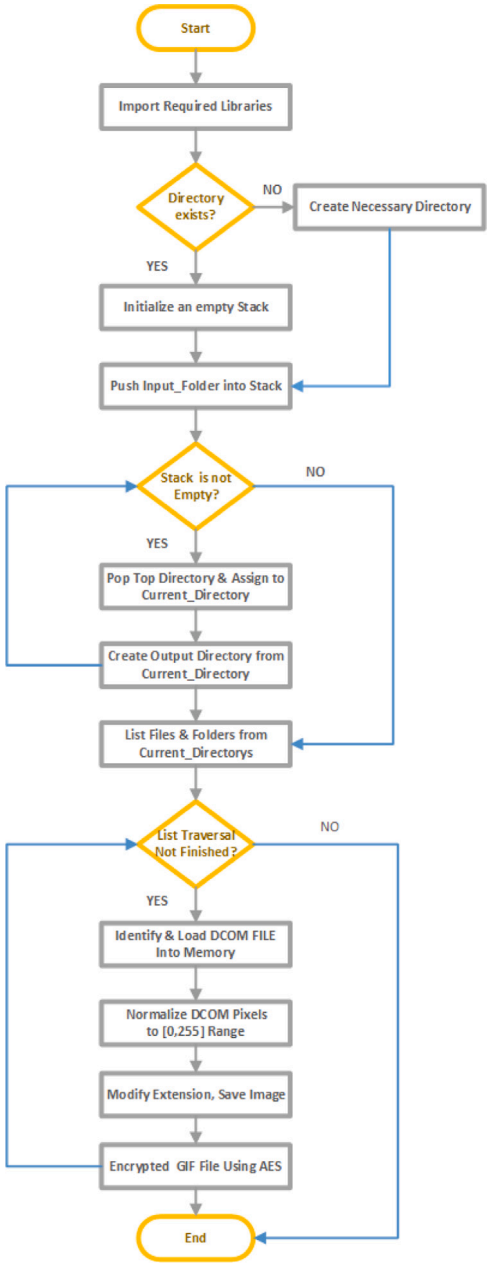

Fig. 2. Stepwise process for converting DICOM to EGIF using Diegif framework.

usage. DES's weaknesses in speed and security have led to its replacement by more robust alternatives like AES. Consequently, DES is unsuited for current medical imaging processes, where safe and efficient encryption is paramount [34].

RSA (Rivest-Shamir-Adleman) RSA utilizes asymmetric encryption, with key lengths generally ranging from 1024 to 2048 bits. It provides extremely high security, notably for secure key exchanges. However, RSA's processing needs result in slower encryption and decryption rates, making it impractical for huge datasets like medical imaging. While RSA excels in securing sensitive data during key transfer, it is less feasible for encrypting and processing mass data inside healthcare systems [35].

3.5. Decryption algorithm

Giving the addresses of the encrypted folder and the folder where the decrypted data will be placed is the first step in decrypting and saving encrypted GIF files from the DICOM. The method then determines which files are encrypted GIFs by searching for files

Table 1
Comparison of encryption methods for medical data security.

Encryption method	Key length	Speed	Suitability
AES	128/192/256 bits	High	Ideal for large datasets and medical images
DES	56 bits	Moderate	Unsuitable for modern use
RSA	1024/2048 bits	Low (slow)	Best for small-scale key exchanges

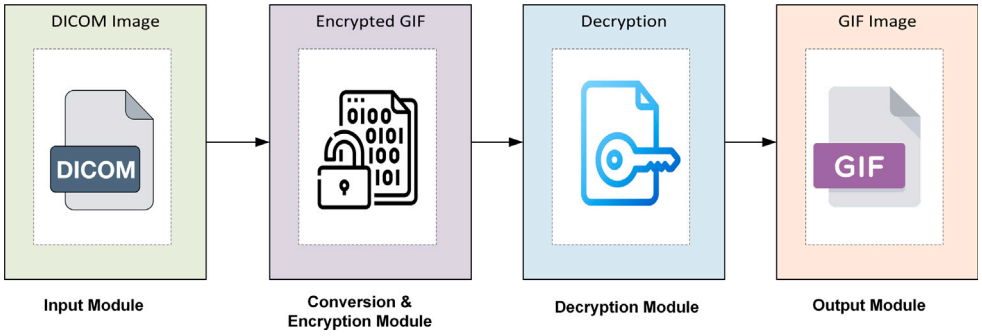

Fig. 3. DICOM to GIF conversion process.

in the current directory that have the "encrypted.gif" extension. The encryption key is used to unlock each encrypted file after it has been placed into memory. The "encrypted" suffix is removed and replaced with ".gif" to generate the output filename. The GIF file is then placed in the proper decrypted directory within the final destination folder.

Algorithm 2 Pseudocode for Decrypting from EGIF

Require: D_i: input directory, E_K: encryption key, E_f: encrypted file, P_{root}: path to root, D_o: output directory, N_d: number of directories, N_f: number of files

if not D_o exists **then**
  $D_o \leftarrow$ create(loc, $name$)
end if
$E_K \leftarrow$ set(AES, Key)
for $i = 1$ to N_d **do**
  $D_i \leftarrow E_f$
  for $j = 1$ to N_f **do**
    if file_list(j) is 'encrypt.gif' **then**
      file_list(j) $\leftarrow$ Normalize(0, 255)
      file_list(j) $\leftarrow$ Decrypt(E_K, 'encrypt.gif')
      save(file_list(j), D_o)
      Remove(file_list(j), '.gif')
    end if
  end for
end for

The Algorithm 2 analyzes the input directory and its subdirectories for encrypted EGIF files, decrypts them with the encryption key provided, and stores the decrypted files in the output directory. This procedure is done for every file in the directory structure.

The algorithm takes several input as D_i: The input directory where encrypted files are located, E_K: The encryption key used for decryption, E_f: The encrypted file, P_{root}: The path to the root directory, D_o: The output directory where decrypted files will be saved, N_d: The number of directories., N_f: The number of files. The method verifies the existence of the output directory D_o. The directory is created if it does not already exist. Using the AES technique, the encryption key E_K is set. To represent the number of directories, the algorithm enters a loop that iterates N_d times. The variable E_f in the loop is given the value of the input directory D_i. An additional loop, N_f, denoting the number of files in the directory, iterates N_f times before the algorithm moves on. For each file in the directory, the method searches for the filename encrypt.gif, which denotes an encrypted file. If the file is encrypted, the algorithm decrypts it using the encryption key E_K and the filename "encrypt.gif". The decrypted file is subsequently saved in the output directory D_o. Removes the ".gif" extension from the file name after. The inner loop will continue to process files in the current directory until all of them have been processed. The outer loop will continue to process directories until all of them have been processed. The overview of the conversion process is shown in Fig. 3.

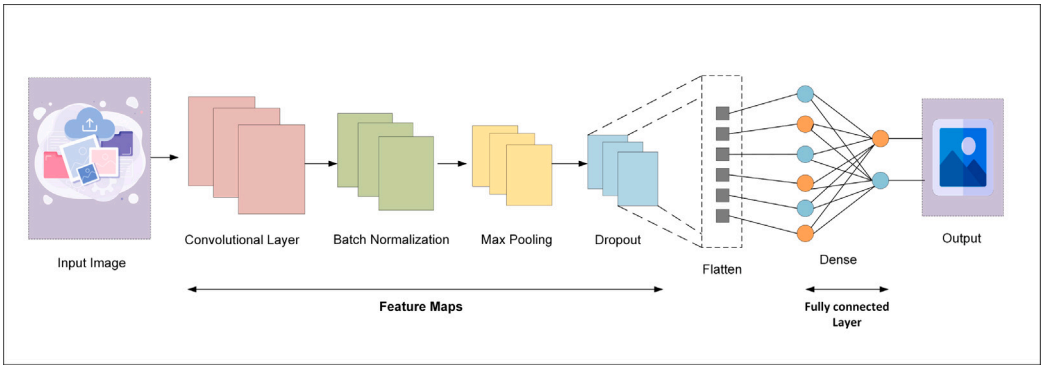

Fig. 4. CNN network architecture.

3.6. Confidential machine learning

A strong basis for creating machine learning models while preserving the confidentiality and privacy of the input data is provided by the "confidential ML training" method. The approach takes a methodical approach that begins with decrypting the encrypted data, prepares it for machine learning, extracts key features, examines the features, creates the ML model, and finally re-encrypts the dataset after training. With the encryption key received from the Access Manager, the training and testing datasets are first decrypted. The 'Confidential ML Training' method, which ensures data confidentiality throughout the ML process, allows researchers to use sensitive data while retaining their privacy and security. The technique uses the encrypted dataset for training and testing, which is temporarily decrypted to ensure that the data is always secure. Once the training is over, the data is once again encrypted to keep it secure and inaccessible to unauthorized parties.

Algorithm 3 Pseudocode for Confidential ML Training Algorithm

Require: D_s: Dataset, F_i: Feature identification, F_p: Feature performance analysis, C_m: Model creation, T_m: Model testing
1: $Access\_Manager \leftarrow$ Encryption Key
2: **if** D_s is decrypted **then**
3: $D_s \leftarrow$ Make(preprocess, divide)
4: **end if**
5: $D_s \leftarrow$ Identify(F_i, extraction)
6: $D_s \leftarrow$ Perform(F_p, insights)
7: $D_s \leftarrow$ Create(C_m, algorithm)
8: $D_s \leftarrow$ Test(T_m, predictions)
9: $D_s \leftarrow$ Encrypt(key, Access_Manager)

The Confidential ML Training Algorithm 3 approach for training machine learning models while maintaining the confidentiality of the dataset. It requires a dataset (D_s) as well as additional parameters like feature identification (F_i), feature analysis (F_p), model development (C_m), and model testing (T_m). The algorithm goes through the following steps: The encryption key is first obtained by the algorithm from "Access Manager". It determines whether the dataset (D_s) requires decryption. If so, it divides the dataset using the "Make" method and preprocesses it. The "Identify" technique is then used by the algorithm to identify the features in the dataset using the feature dataset (F_i). The "Perform" method and the analysis technique (F_p) are then used to execute feature analysis on the detected features. This process aids in extracting insights and useful data from the features. The algorithm uses the "Create" method and the (C_m) ML model generation methodology to create a machine learning model. In order to do this, a suitable algorithm must be chosen, and the model must be trained using the prepared features and dataset. Utilizing the "Test" method and the testing approach (T_m), the developed ML model is then evaluated for performance. The "Encrypt" method is used to encrypt the dataset once it has been decrypted using the encryption key and the "Access Manager". Following the training and testing phases, this stage guarantees the dataset's confidentiality.

3.6.1. CNN network architecture

The Diegif framework integrates with ML workflows by decrypting EGIF images on the fly, allowing real-time feature extraction while maintaining data confidentiality. A Convolutional Neural Network (CNN) is a class of neural networks that have proven very effective in areas such as object recognition and detection across various computer vision challenges. CNNs have been highly successful in both image and general visual recognition tasks, largely due to their ability to automatically discover spatial hierarchies of features from input data. Essential Components and Architecture of CNNs As shown in Fig. 4.

Node initialization. In this paper, The He initialization method, defined by Eq. (1), Boulila et al. [36] has been used to initialize the weight of the nodes. It has been observed that the He initialization method improves the convergence rate and boosts the performance of the model during training.

$$w = rand_{normal}(shape, mean = 0, stddev = stddev) \tag{1}$$

The Eq. (1) uses the square root function and generates random values from a defined normal distribution, which is expressed in Eq. (2).

$$S_{dvt} = sqrt(2/n_{in}) \tag{2}$$

In Eq. (1), the w is the weight obtained by He initialization, which is dependent on the standard deviation S_{dvt} measured from Eq. (2) where n_{in} are the input units.

Sequential model. In this paper, the sequential Model, defined by Eq. (3), represents the computation performed in the first layer of the neural network to enable the flow of data from one layer to the next. Each layer in the model can have its own set of weight matrices, bias vectors, and activation functions.

$$W_{Sum} = W_{matrix} * X_{Input} + b1_{vector} \tag{3}$$

The Eq. (3) uses the weighted sum to introduce non-linearity applied by the activation function, which is expressed in Eq. (4).

$$A_{Output} = relu_{function}(W_{Sum}) \tag{4}$$

In Eq. (3) the W_{Sum} is the weight matrix for the first layer, whose dimensions are determined by the number of input units in the first layer. W_{matrix} measured is the weight matrix of the first layer. X_{Input} represents the input vector to the first layer, and $b1_{vector}$ represents the bias vector. In Eq. (4) where A_{Output} The output activation becomes the input for the next layer in the sequence and $relu_{function}$ applies the ReLU activation function element-wise to the weighted sum to obtain the output activation.

Conv2D layer. In this study, the Conv2D Layer, specified by Eq. (5), is used to represent a convolution operation performed on a tensor. Eq. (5) computes the 2D convolution between the input tensor X and the corresponding filter weights $W[:, :, :, j]$. It involves element-wise multiplication of the input values with the filter weights, followed by a summation operation over the indices (i, m, n). For each spatial location, (i, j) in the output feature map and for each input channel k, The sum of the element-wise products of the filter weights and the corresponding values in the input tensor is computed by the equation. This operation is performed for each filter j, resulting in a separate response or feature map for each filter.

$$\text{Conv}_{(X,W_{:,:,:,j})} = \sum_{i,m,n} X_{[:,i+m,j+n,k]} \cdot W_{m,n,k,j} \tag{5}$$

After the convolution operation, add the bias term $b[j]$ to each corresponding element in the output feature map in Eq. (6). The bias term allows for an additional learnable offset that can help in shifting the output of the Conv2D layer. The bias term contributes to the flexibility and adaptability of the convolutional responses.

$$Y_{[:,:,:,j]} = \text{Conv}_{(X,W_{[:,:,:,j]})} + b_{[j]} \tag{6}$$

In Eq. (6), X be the input tensor of shape (batch_size, height, width, channels). W be the weight tensor of the filters of shape (kernel_height, kernel_width, input_channels, output_channels). b be the bias tensor of shape (output_channels). Y be the output tensor of shape (batch_size, output_height, output_width, output_channels). These equations include the convolution operation, bias addition, and the generation of output feature maps, which allow the model to learn and extract meaningful features from input data. The feature maps that result collect significant information and serve as inputs for succeeding layers, allowing the model to perform tasks like image classification and object detection.

Batch normalization. In this paper, the Batch Normalization method, defined by Eqs. (7), (8) has been used to compute the mean and variance. Eq. (7) computes the mean μ_B of the mini-batch inputs X by adding all of the inputs and dividing by the batch size m. Eq. (8) computes the variance σ_B^2 of the mini-batch inputs X by subtracting the mean μ_B from each input, squaring the differences, adding them together, and dividing by the batch size m.

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} X_i \tag{7}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(X_i - \mu_B)^2 \tag{8}$$

The Eq. (9) uses to divide each input $\hat{X}_i$ by the square root of the variance σ_B^2, subtracts the mean μ_B, and adds a small constant ϵ for numerical stability to normalize each input X_i.

The Eq. (10) scales the normalized input $\hat{X}_i$ by a learnable parameter γ (scale parameter), allowing the network to regulate the strength of the activations. Additionally, it uses a learnable parameter β(shift parameter) that gives the activations an offset or bias.

$$\hat{X}_i = \frac{X_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{9}$$

$$Y_i = \gamma \hat{X}_i + \beta \tag{10}$$

In Eq. (9), $\hat{X}_i$ represents the normalized input, which is created by subtracting the mean from and dividing the variance's square root. In Eq. (10) where Y_i indicates the final result after applying the scale and shift. BatchNormalization Deep learning models typically employ the batch normalization technique to normalize the activations of a neural network layer. It helps to stabilize and accelerate the training process by limiting internal covariate shifts and giving regularization effects.

Max Pooling2D. The MaxPooling2D layer generates the corresponding output value in the output feature map by taking the maximum value from each 2×2 region of the input feature map. Defined by Eq. (11), has been used to The MaxPooling2D layer chooses the highest value within each window after splitting the input feature map into non-overlapping 2×2 windows. $Y_{[i,m,n]}$ represents the output value at position (m, n) in the ith feature map of the output.

$$Y_{[i,m,n]} = \max(X_{[i,2m:2m+1,2n:2n+1]}) \tag{11}$$

In Eq. (11) $X[i, 2m : 2m+1, 2n : 2n+1]$ represents the 2×2 region of the input feature map X at positions $(2m, 2n)$, $(2m, 2n+1)$, $(2m+1, 2n)$, and $(2m+1, 2n+1)$. And max() denotes the maximum operation, which selects the maximum value within the 2×2 region. This equation explains how the MaxPooling2D layer works. For each output value, the maximum value inside a $2by2$ window of the associated input feature map is used. This process keeps the most important features while shrinking the feature maps' spatial dimensions.

Dropout. In this architecture, the Dropout, defined by Eq. (12), represents the calculation of a dropout layer during training in this neural network. Z indicates the output of the dropout layer. W represents the input matrix (or vector) to the dropout layer. $Mask$ is a binary matrix with elements that have values of 0 or 1, and it has the same shape as the input. Based on the dropout rate, a random number between 0 and 1 is chosen for each element to create the Mask matrix. By multiplying the input by the $mask$ element by element, the dropout layer's output is produced.

$$Z_{Output} = W_{Input} \cdot (A + b)_{Mask} \tag{12}$$

The Eq. (13), A indicates the dropout layer's output when testing or making inferences. $g_{(Z)}$ represents the input matrix (or vector) to the dropout layer. $(1 - \text{DropoutRate})$ represents the dropout probability.

$$A = g_{(Z)} \cdot (1 - \text{DropoutRate}) \tag{13}$$

In Eq. (12), W is the weight matrix of the layer, A is the output activation of the layer, and b is the bias vector of the layer. In Eq. (13), $g()$ is the activation function, and Z is the weighted sum of inputs plus bias. This equation uses dropout as a regularization approach, which helps to reduce overfitting and encourages more resilient and generalizable representations in the network. In training, the dropout layer scales the activations to retain the expected value by multiplying them by a dropout mask element-wise, in testing, the dropout layer multiplies the activations by $(1 - \text{DropoutRate})$, which keeps the activations at the predicted value. The network's generalization abilities are enhanced and overfitting is prevented.

Flatten. In this paper, the Flatten Layer, defined by Eq. (14), has been responsible for transforming the 2D feature maps into a 1D vector. It transforms the input tensor with shape (batch_size, height, width, channels) into a tensor with shape (batch_size, flattened_size), where flattened_size is equal to height width channels.

$$Y_{output} = X_{input} . RS_{(b_s, f_s)} \tag{14}$$

In Eq. (14) where X_{input} represents the input tensor with shape (batch_size, height, width, channels). Y_{output} represents the output tensor after flattening, with shape (batch_size, flattened_size), $RS()$ is the function that reshapes the input tensor, (b_s, f_s) represents batch_size, flattened_size specifies the desired shape of the output tensor.

Dense. This paper, defined by Eq. (15), encapsulates the Dense layer's basic operations, which include matrix multiplication between the input tensor X and the weight matrix W, bias vector B addition, and element-wise application of the activation function f(). The equation shows how the Dense layer learns and applies a set of weights and biases to turn the input data into an output tensor. Nonlinearity is introduced into the network by the activation function, allowing it to learn complicated patterns and produce nonlinear predictions.

$$Y_{output} = f_{(XW+B)} \tag{15}$$

In Eq. (15), X represents the input tensor with shape (batch_size, input'_dim). Y represents the output tensor with shape (batch_size, output_dim). W represents the weight matrix with shape (input_dim, output_dim). B represents the bias vector with shape (output_dim). $f()$ represents the activation function applied element-wise to the tensor.

3.7. 'User Interaction and Application' system

The 'User Interaction and Application' system provides a robust platform for seamless collaboration between users, trained models, clinicians, and a diagnosis system in the medical industry. It provides users with an easy-to-use interface for retrieving medical images that can be obtained over an API Access Manager. The system offers model training, allowing users to exploit accessible datasets to train models particularly intended to execute tasks or predictions based on the medical image data. Once trained, this model generates predictions that are shared with physicians for further analysis and decision-making. By evaluating the

predictions alongside the linked photos and relevant information, physicians might receive useful insights into probable diagnoses or treatment options. This analysis delivers actionable insights, recommendations, and diagnostic information, complementing the clinicians' decision-making process. Finally, a thorough final report is prepared, integrating the original medical image, model predictions, diagnosis system findings, and other essential details.

Algorithm 4 Pseudocode for User Interaction and Application System Algorithm

Require: D_s: Dataset, M_t: Model training, M_p: Model prediction, P_i: Physician interaction, D_s: Diagnosis system, F_r: Final report

1: $Access\_Manager \leftarrow D_s$
2: $D_s \leftarrow$ train(M_t, execute)
3: $D_s \leftarrow$ predict(M_p, evaluate)
4: $D_s \leftarrow$ interact(P_i, analyze)
5: $D_s \leftarrow$ diagnose(D_s, guide)
6: $D_s \leftarrow$ generate(F_r, summary)

The Algorithm 4 requires various inputs, including the dataset (D_s), model training technique (M_t), model prediction technique (M_p), physician interaction method (P_i), diagnostic system (D_s), and final report format (F_r). An "Access Manager" is created and given the dataset value (D_s). The Access Manager is in charge of controlling dataset access throughout the algorithm. The algorithm trains the model by using the "train" function with the model training technique (M_t) and the "Execute" argument. This stage uses the provided dataset to train a machine-learning model. The program then moves on to the prediction phase after training the model. It invokes the "Predict" function using the model prediction approach (M_p) and the "Evaluate" parameter. This stage entails utilizing the trained model to generate predictions on new data. Through the "Interact" function, the program allows for interaction with medical professionals. It sends the function the physician interaction method (P_i) and the "Analyze" argument. This stage allows clinicians to examine the model projections and provide input. The algorithm then invokes the "Diagnose" function, passing in the diagnostic system (D_s) and the "Guide" argument. This stage entails applying specific diagnostic techniques or algorithms to the dataset utilizing the diagnosis system. Finally, the algorithm uses the "Generate" function to construct a final report. It gives the function the final report format (F_r) and the "Summary" parameter.

In conclusion, the "User Interaction and Application System" algorithm promotes user engagement and carries out a number of operations, including dataset management, model training, prediction, physician interaction, diagnosis, and report creation. It makes sure that users and the application system work together effectively, enabling informed decision-making and offering priceless insights at every stage of the procedure.

4. Performance analysis

The efficiency of the *Diegif* framework in converting DICOM files to GIF format is evaluated in the research paper's Performance Analysis section. It measures the efficiency of its operations in terms of faster processing with decreased storage needs and enhanced security. This section examines the security mechanisms implemented, analyzes processing times, and measures *Diegif*'s file size reduction compared to alternative techniques. The investigation validates *Diegif*'s potential as a DICOM to GIF converter by demonstrating its improved performance in these areas.

4.1. Saves the storage

The *Diegif* framework's potential for storage savings is another crucial feature. Because of their famous size, DICOM files can use up a lot of storage space. The file size reduction brought about by utilizing *Diegif* to convert DICOM files into the GIF format should be the subject of the performance analysis. This can be determined by contrasting the original DICOM files' and corresponding GIF files' file sizes.

The table compares a set of images' storage capabilities before and after a specific operation as shown in Table 2. It has three columns: "S.NO" denotes the serial number of DICOM file directories, "Original DICOM files (kb)" shows the storage capacity in kilobytes prior to the process, and "Converted EGIF files (kb)" shows the storage capacity in kilobytes following after the process. The table includes serial numbers ranging from 1 to 10, as well as storage capacity values. The "Total" row shows the overall storage space, with 52,093 kB before and 17,568 kB after the operation.

$$\text{Storage Average} = \frac{1}{m}\sum_{i=1}^{m} t_i \tag{16}$$

In Eq. (16) defined the total storage time per image, where m represents the total number of photos, t_i represents the execution time for each photo. The "Average" row shows the average storage per data item, with 5209.3 kB before and 1756.8 kB after the operation. The table facilitates the comparison of storage capacities and offers information on the total and average capacities for the data items. As shown in Fig. 5, the graph illustrates Pre-Post Storage Capacity Analysis. As a result of the procedure, there is about a 66.32% reduction in storage per data item, resulting in storage savings of approximately 66.32%.

The Diegif framework demonstrates its practical value through measurable performance improvements, notably achieving a 66.32% reduction in storage requirements. This reduction was calculated by comparing the average file size of DICOM images

Table 2
Storage capacity comparison.

S.NO	Original DICOM files (kb)	Converted EGIF files (kb)
1	6858	2788
2	515	117
3	7988	2831
4	6977	2693
5	6954	1556
6	6858	2703
7	515	87
8	7958	2467
9	6955	2201
10	515	125
Total	52 093	17 568
Average	5209.3	1756.8

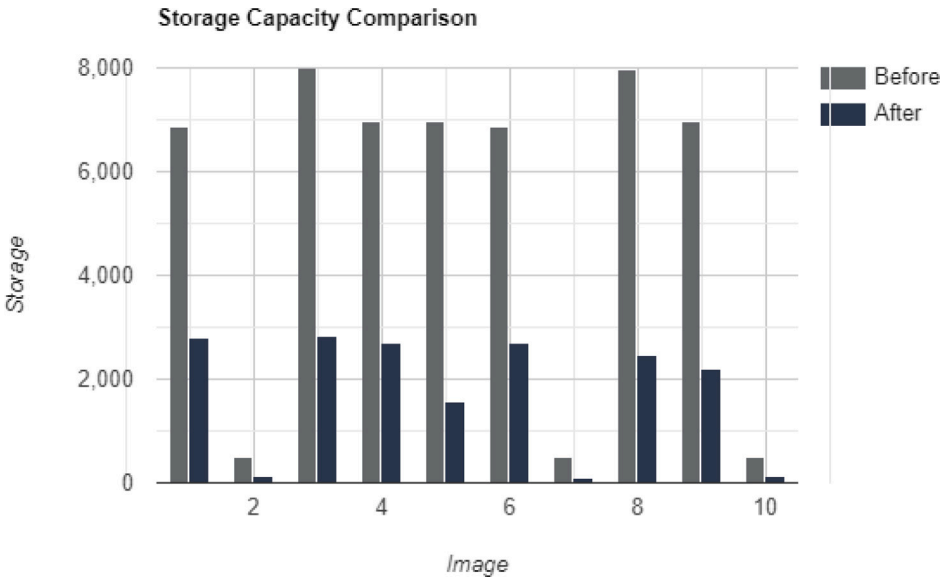

Fig. 5. Pre-post storage capacity analysis.

(50 MB) to their encrypted EGIF counterparts (16.88 MB). The framework's integrated encryption and file optimization streamline medical imaging workflows, reducing overhead while maintaining data fidelity. These metrics highlight Diegif's ability to address storage constraints in healthcare systems, offering significant resource savings without compromising on security or image quality, thereby showcasing its tangible benefits.

4.2. Security

The performance analysis should cover the security features of the *Diegif* framework in addition to faster processing and storage reductions. Data security should be given priority during any conversion procedure because DICOM files frequently contain sensitive patient information. The investigation ought to emphasize the security controls put in place by *Diegif* to guarantee the privacy, reliability, and accessibility of the converted GIF files. Talking about access control methods and encryption techniques may be included. *Diegif*'s security features are assessed to give users and stakeholders confidence in the framework's ability to protect patient data.

Secure Storage: To protect the data, the secure storage system used encryption techniques. A powerful encryption algorithm is used to encrypt the data in the secure storage. The secure storage is only accessible to verified individuals or authorized professionals. This can be accomplished via user authentication mechanisms such as access tokens. No data were compromised due to the secure storage solution. This indicates that the encryption and authentication mechanisms effectively protected the data, allowing access only to authorized individuals. The secure storage system achieved full secure system, meaning it successfully safeguarded the data against unauthorized access.

4.3. Clinical significance

Storage Savings Measurement: The storage needs for each medical picture were lowered by roughly 66.32%. This statistic was presumably determined by examining the size of the DICOM pictures before conversion and comparing them to the size of the newly produced EGIF files. In healthcare facilities that produce huge volumes of imaging data, storage costs might be a major problem. Reducing the storage space needed for each photograph by over 66.32% may result in huge cost savings, especially for hospitals and clinics that handle enormous archives of patient data. Additionally, it provides for more effective use of current storage infrastructure and streamlines the integration of photos into Electronic Health Records (EHR) systems, providing speedier access to medical data.

International Data Privacy Standards: To ensure its application in varied healthcare situations, the Diegif framework fits with international data privacy standards, such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). Below are major compliance aspects:

GDPR Compliance: The framework conforms to assuring encryption throughout data processing and storage to defend against unwanted access. By utilizing AES encryption, the framework decreases the danger of data breaches, satisfying GDPR regulations for data security and confidentiality.

HIPAA Compliance: The framework supports HIPAA's Security Rule, providing encryption measures for protected health information (PHI) to preserve electronic data. Role-based access control guarantees that only authorized workers may decrypt and view EGIF files, preserving full compliance with HIPAA's privacy rules.

Data Minimization and Auditability: The Diegif framework follows the idea of data minimization by lowering the size of medical pictures (66.32% decrease), optimizing storage while preserving data integrity. Logs and audit trails capture data access and processing, promoting transparency and compliance with regulatory audits.

Secure Data Transfer: Encrypted EGIF files comply with GDPR's data portability obligations, allowing safe data exchange between systems and organizations.

4.4. Case studies and applications

The *Diegif* framework illustrates its advantages over typical DICOM processes via practical applications that solve basic difficulties in medical imaging. Below, we illustrate real-world cases where outperforms traditional approaches.

4.4.1. Secure image archiving in radiology departments

Radiology departments create huge volumes of high-resolution imaging data, causing issues in storage and security. A mid-sized hospital adopting achieved:

Storage Efficiency: Reduced storage capacity by 66.32%, saving 33 GB for every 1000 CT scans (average size 50 MB per picture).

Integrated Security: Eliminated dependency on external encryption tools by integrating AES encryption during conversion, preserving sensitive data.

Streamlined PACS Workflow: *Diegif* easily delivered encrypted EGIF files to PACS systems, maintaining HIPAA compliance.

This integration illustrates how *Diegif* handles storage limits and boosts security for hospital archives.

4.4.2. Confidential machine learning for cancer research

AI-driven cancer diagnosis needs massive datasets while respecting patient privacy. A research group employing *Diegif* achieved:

Secure Data Handling: EGIF files kept secrecy during preprocessing and training phases.

Collaboration: Facilitated safe data exchange for federated learning across different universities, facilitating privacy-preserving research.

This highlights *Diegif*'s role in driving AI adoption while maintaining data security in collaborative healthcare contexts. These case studies highlight the practical utility of the framework architecture in tackling varied difficulties throughout medical imaging processes, from efficient storage and secure AI integration to real-time processing in critical care.

4.5. Computational complexity and bottlenecks

The *Diegif framework* delivers repeatability and efficiency via its well-structured pseudocode algorithms while addressing computing needs. The DICOM-to-EGIF conversion procedure comprises directory traversal and pixel normalization, running at $O(n)$, where n indicates the total number of folders and files. AES encryption, crucial to the system, has a temporal complexity of $O(n \cdot m)$, where m is the encryption key length, guaranteeing strong security for medical imaging datasets. Potential bottlenecks, such as memory cost during encryption and I/O delays in large-scale data handling, are minimized by parallel processing and improved file handling algorithms. These solutions boost scalability, allowing the framework to analyze huge datasets effectively while retaining excellent performance in healthcare situations.

Table 3
Comparison of Existing solutions vs. Proposed Deigif solution.

Feature	Existing solutions	Author name	*Diegif* Solution (EGIF)
File size reduction	Moderate to high	Pervan et al. [37]	66.32% reduction
Encryption	Applied post-conversion	Ebenezer et al. [38]	Integrated during conversion (AES)
Accessibility	Easily shareable formats but security relies on external measures	Varma [16]	EGIF format ensures that only authenticated users can decrypt and access images
Storage efficiency	Higher storage needs	Suapang et al. [39]	Optimized for lower storage
Use in machine learning	Uses unencrypted data, risking data exposure		Confidential ML training with encrypted data

4.6. Comparison with other solutions

In this part, Provide a comprehensive comparison of the proposed DICOM to EGIF conversion solution with other current solutions in the medical imaging arena. The comparison Table 3 will focus on essential features such as file size reduction, encryption, accessibility, storage efficiency, and machine learning capability, this comparison sheds light on the uniqueness and efficacy of the suggested technique.

With current methods, file sizes are frequently reduced by moderate to high, yet compression is only modest [37]. In comparison, the suggested EGIF method lowers file size by 66.32% while getting a significantly higher compression rate. Large medical image systems demand faster data exchanges and more efficient storage, both of which are allowed by this.

In terms of encryption, the data is exposed throughout the image transfer process because earlier methods often add protection later [38]. By adding AES encryption into the translation process, the EGIF system ensures end-to-end security and lowers the chance of unwanted entry during secondary activities.

By assessing the DICOM file, exiting solutions that are openly shared, the images within them are protected by external security procedures [16]. The EGIF format improves security and protects patient information by restricting image access to only authorized users.

In current solutions, systems like PACS (Picture Archiving and Communication Systems) are often employed for saving and protecting DICOM files but usually discover challenges in handling huge datasets due to high storage demands. TIFF and PNG photos often need greater storage room due to their high quality and lack of streamlining [39]. In comparison, the suggested DICOM to EGIF conversion method uses less storage space and is better suited to apps needing big image files.

In conventional medical imaging systems, current solutions face the risk of releasing private data during machine learning [40]. Encrypted pictures can be used in machine learning thanks to EGIF, which ensures data security while keeping usefulness for AI model training in medicine.

5. Challenges and future directions

Implementing the framework in older healthcare systems involves overcoming fundamental compatibility concerns. To maintain interoperability, the framework contains a backward-compatibility module for reconverting EGIF files to DICOM format. Additionally, lightweight encryption techniques have been tuned for resource-constrained situations to decrease computational overhead. Customizable integration routes further ease adoption, enabling hospitals to integrate the framework with current processes progressively. By solving these difficulties, the Diegif framework delivers a scalable, secure solution for varied healthcare contexts.

The *Diegif* framework has proven tremendous promise in tackling storage and security concerns for medical imaging data. However, future study and development may enhance its usefulness and efficacy in healthcare systems. Whilst this research focuses on DICOM-to-EGIF conversion, medical imaging incorporates numerous additional formats, including TIFF and JPEG2000, extensively utilized for particular applications such as histopathology and radiology archives. Future development may allow the framework to:

Leverage Advanced Compression: Formats like JPEG2000, which provide both lossy and lossless compression, might be merged with EGIF's encryption techniques to boost storage efficiency further. In the future, the study might be extended to encompass the translation of DICOM pictures into other common image formats, such as JPEG and PNG. This advancement would boost the algorithm's adaptability and utility, enabling it to satisfy a greater variety of demands. Addressing this future scope might greatly contribute to the profession by offering a more complete solution for DICOM image conversion.

Integration with Hospital Systems

The Picture Archiving and Communication System (PACS) is vital to healthcare institutions, providing storage, retrieval, and dissemination of medical imaging data. Integrating the *Diegif* framework with PACS will entail increased security protocols. By incorporating encryption directly into PACS pipelines, the architecture may mitigate risks commonly linked with manual or third-party security solutions. A real use case may entail implementing *Diegif* in a hospital's radiology department to minimize storage costs, safeguard patient data, and permit ML-based diagnosis straight from PACS.

Real-Time Processing for Clinical Applications

Real-time medical imaging applications, such as ultrasound and emergency room diagnostics, necessitate speedy processing and safe management of sensitive data. Expanding the *Diegif* framework to cover such circumstances will require:

Optimizing encryption and conversion techniques to function under severe latency limitations without sacrificing security. Parallel processing methods employing GPUs may be exploited to speed encryption during conversion. Developing support for real-time data streams, allowing immediate conversion and encryption of image data during capture. For example, in an emergency department, the system may allow real-time ultrasound imaging while assuring encrypted data transmission to secure archives for further review or machine learning analysis.

Federated Learning for Collaborative Healthcare *Diegif* may be modified for federated learning, providing safe, decentralized machine learning across various healthcare institutions. This would assist privacy-preserving research and increase diagnostic accuracy while securing patient data.

6. Conclusions

The *Diegif* framework presents a unique way to address fundamental difficulties in medical imaging, including data security, storage optimization, and secrecy in machine learning-based diagnostic systems. By transforming DICOM files into the EGIF format, the framework decreases file sizes by 66.32%, considerably cutting storage and data transfer costs while retaining data integrity and security using AES encryption. Decrypted EGIF files can only be accessed by authorized users thanks to the encryption and security mechanisms it incorporates to safeguard sensitive data. The secure storage system achieved full security, meaning it successfully safeguarded the data against unauthorized access. Additionally, the conversion process is outlined in the framework in a clear and organized manner, making it simple for researchers and developers to comprehend and duplicate. In terms of security and storage effectiveness, our performance analysis shows that *Diegif* performs better than alternative approaches. It allows for the smooth integration of DICOM images into processes, making it a useful tool for medical image processing. For converting DICOM pictures to GIF format, *Diegif* is a reliable and secure option. *Diegif*'s capabilities can help researchers and developers improve their work, advance medical imaging, and provide secure data management in healthcare environments.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was supported by the "University of Debrecen Program for Scientific Publication".

Data availability

Data will be made available on request.

References

[1] Aljondi R, Alghamdi S. Diagnostic value of imaging modalities for COVID-19: scoping review. J Med Internet Res 2020;22(8):e19673.
[2] Puech PA, Boussel L, Belfkih S, Lemaitre L, Douek P, Beuscart R. DicomWorks: software for reviewing DICOM studies and promoting low-cost teleradiology. J Digit Imaging 2007;20:122–30.
[3] Gibaud B. The quest for standards in medical imaging. Eur J Radiol 2011;78(2):190–8.
[4] Ali HA, Ne'ma BM. Effective variations on opened GIF format images. IJCSNS 2008;8(5):70.
[5] Dzwonkowski M, Rykaczewski R. Secure quaternion feistel cipher for DICOM images. IEEE Trans Image Process 2018;28(1):371–80.
[6] Andrikos C, Rassias G, Tsanakas P, Maglogiannis I. An enhanced device-transparent real-time teleconsultation environment for radiologists. IEEE J Biomed Health Inform 2018;23(1):374–86.
[7] Chen P. Study on medical image processing technologies based on DICOM. J Comput 2012;7(10):2354–61.
[8] Liu B, Zhu M, Zhang Z, Yin C, Liu Z, Gu J. Medical image conversion with DICOM. In: 2007 Canadian conference on electrical and computer engineering. IEEE; 2007, p. 36–9.
[9] Ujgare NS, Baviskar SP. Conversion of DICOM image in to JPEG, BMP and PNG image format. Int J Comput Appl 2013;62(11).
[10] Younas F, Raza A, Thalji N, Abualigah L, Zitar RA, Jia H. An efficient artificial intelligence approach for early detection of cross-site scripting attacks. Decis Anal J 2024;11:100466.
[11] Obeidat I, Mughaid A, AlZu'bi S, Al-Arjan A, Al-Amrat R, Al-Ajmi R, Al-Hayajneh R, Abuhaija B, Abualigah L. A novel secure cryptography model for data transmission based on rotor64 technique. Multimedia Tools Appl 2024;83(13):37295–314.
[12] Shakya AK, Vidyarthi A. Comprehensive study of compression and texture integration for digital imaging and communications in medicine data analysis. Technologies 2024;12(2):17.
[13] Xue W, Lu W, Wang H, Meng J. A solution for display and processing of DICOM images in web PACS. Zhongguo yi Liao qi xie za zhi=Chin J Med Instrum 2009;33(3):179–82.
[14] Monteiro EJM, Costa C, Oliveira JL. A DICOM viewer based on web technology. In: 2013 IEEE 15th international conference on e-health networking, applications and services (healthcom 2013). IEEE; 2013, p. 167–71.
[15] Li X, Morgan PS, Ashburner J, Smith J, Rorden C. The first step for neuroimaging data analysis: DICOM to nifti conversion. J Neurosci Methods 2016;264:47–56.

[16] Varma DR. Managing DICOM images: Tips and tricks for the radiologist. Indian J Radiol Imaging 2012;22(01):4–13.
[17] Lu X, Gu Y, Zhang B, Deng Z, Fan Y. Research and implementation of converting DICOM multi-frame medical image to multimedia format. In: 2010 international conference on multimedia technology. IEEE; 2010, p. 1–5.
[18] Bomewar M, Baraskar T, Mankar V. DICOM image size reduction and data embedding using randomization technique. In: 2015 international conference on pervasive computing. ICPC, IEEE; 2015, p. 1–6.
[19] Golubev A, Bogatencov P, Secrieru G. DICOM data processing optimization in medical information systems. Scalable Comput: Pract Exp 2018;19(2):189–201.
[20] Diaz O, Kushibar K, Osuala R, Linardos A, Garrucho L, Igual L, Radeva P, Prior F, Gkontra P, Lekadir K. Data preparation for artificial intelligence in medical imaging: A comprehensive guide to open-access platforms and tools. Phys Med 2021;83:25–37.
[21] Chung E-J, Yang B-E, Kang S-H, Kim Y-H, Na J-Y, Park S-Y, On S-W, Byun S-H. Validation of 2D lateral cephalometric analysis using artificial intelligence-processed low-dose cone beam computed tomography. Heliyon 2024;10(21).
[22] Yang L. DICOM standard and its application in radioinformatics. Int J Comput Sci Inf Technol 2024;2(1):384–90.
[23] Ortega J. Visualization and segmentation of lung tissue in DICOM-format ct images. In: Journal of physics: conference series. vol. 2796, IOP Publishing; 2024, 012012.
[24] Shivshankar S, Makhija N, Mathusudhanan P. Digital imaging and communication in medicine (DICOM): Biomedical and health informatics: Imaging and interoperability using HL7 and DICOM. In: Smart healthcare and machine learning. Springer; 2024, p. 299–317.
[25] Halchenko YO, Goncalves M, Ghosh S, Velasco P, di Oleggio Castello MV, Salo T, Wodder JT, Hanke M, Sadil P, Gorgolewski KJ, et al. HeuDiConv—flexible DICOM conversion into structured directory layouts. J Open Sour Softw 2024;9(99):5839.
[26] Nabi SA, Kalpana P, Chandra NS, Smitha L, Naresh K, Ezugwu AE, Abualigah L. Distributed private preserving learning based chaotic encryption framework for cognitive healthcare IoT systems. Inform Med Unlocked 2024;49:101547.
[27] Alzoubi S, Aldiabat K, Al-diabat M, Abualigah L. An extensive analysis of several methods for classifying unbalanced datasets. J Auton Intell 2024;7(3).
[28] Armato S, McLennan G, McNitt-Gray M, Meyer C, Reeves A, Bidaut L, Zhao B, Croft B, Clarke L. WE-B-201b-02: the lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed public database of CT scans for lung nodule analysis. Med Phys 2010;37(6Part6):3416–7.
[29] Armato III SG, McLennan G, Bidaut L, McNitt-Gray MF, Meyer CR, Reeves AP, Zhao B, Aberle DR, Henschke CI, Hoffman EA, et al. The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans. Med Phys 2011;38(2):915–31.
[30] Simpson AL, Antonelli M, Bakas S, Bilello M, Farahani K, Van Ginneken B, Kopp-Schneider A, Landman BA, Litjens G, Menze B, et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. 2019, arXiv preprint arXiv:1902.09063.
[31] Clark K, Vendt B, Smith K, Freymann J, Kirby J, Koppel P, Moore S, Phillips S, Maffitt D, Pringle M, et al. The cancer imaging archive (TCIA): maintaining and operating a public information repository. J Digit Imaging 2013;26:1045–57.
[32] Thambiraja E, Ramesh G, Umarani DR. A survey on various most common encryption techniques. Int J Adv Res Comput Sci Softw Eng 2012;2(7).
[33] Priyanka MP, Kaur N, Nazir N, Khan AA, Singh MV, Kaur M, Behera T, Rakhra M, Dahiya O. A comparative review between modern encryption algorithms viz. DES, AES, and RSA. In: 2022 international conference on computational intelligence and sustainable engineering solutions. CISES, IEEE; 2022, p. 295–300.
[34] Al Hasib A, Haque AAMM. A comparative study of the performance and security issues of AES and RSA cryptography. In: 2008 third international conference on convergence and hybrid information technology. vol. 2, IEEE; 2008, p. 505–10.
[35] Olutola A, Olumuyiwa M. Comparative analysis of encryption algorithms. Eur J Technol 2023;7(1):1–9.
[36] Boulila W, Driss M, Alshanqiti E, Al-Sarem M, Saeed F, Krichen M. Weight initialization techniques for deep learning algorithms in remote sensing: Recent trends and future perspectives. In: Advances on smart and soft computing: proceedings of ICACIn 2021. 2022, p. 477–84.
[37] Pervan B, Tomic S, Ivandic H, Knezovic J. MIDOM—A DICOM-based medical image communication system. Appl Sci 2023;13(10):6075.
[38] Ebenezer MM, Félix P, Yannick M, Junior SNP, Léandre NN. Contribution to the improvement of cryptographic protection methods for medical images in DICOM format through a combination of encryption method. Int J Adv Comput Sci Appl 2021;12(4).
[39] Suapang P, Dejhan K, Yimmun S. Medical image compression and DICOM-format image archive. In: 2009 ICCAS-SICE. IEEE; 2009, p. 1945–9.
[40] Contreras FJG, Schwartz RG. Artificial intelligence applications in medical thermography using PACS and DICOM file formats. In: Infrared technology and applications XLVIII. vol. 12107, SPIE; 2022, p. 403–6.