ENCRYPTION BASED SECURE MEDICAL IMAGE SHARING TECHNIQUE USING REVERSIBLE DATA HIDING AND DEEP LEARNING

A PROJECT REPORT

Submitted by

EKANATHAN S A	**(Reg. No. 202006018)**
MADESH K	**(Reg. No. 202006252)**
RAJASEKAR M	**(Reg. No. 202006253)**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI

(An Autonomous Institution affiliated to Anna University, Chennai)

MAY 2023

BONAFIDE CERTIFICATE

Certified that this project report titled **Encryption Based Secure Medical Image Sharing Technique Using Reversible Data Hiding and Deep Learning** is the bonafide work of **EKANATHAN S A (Reg. No. 202006018), MADESH K (Reg. No. 202006252), RAJASEKAR M (Reg. No. 202006253)** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

.

FACULTY INCHARGE Mrs.M.BLESSA BINOLIN PEPSI, M.E., (Ph.D) Assistant Professor(Senior Grade), Department of Information Technology, Mepco Schlenk Engineering College, Sivakasi-626005. Virudhunagar Dt. Tamilnadu.	HEAD OF THE DEPARTMENT Dr.T.REVATHI, M.E., Ph.D., Senior Professor and Head, Department of Information Technology Mepco Schlenk Engineering College, Sivakasi-626005. Virudhunagar Dt. Tamilnadu.

Submitted for Viva-Voce Examination held at **MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI (AUTONOMOUS)** on **……**

Internal Examiner 1 **Internal Examiner 2**

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

Apart from our efforts, the success of our project depends largely on the encouragementof many others. We take this opportunity to express our gratitude to the people who have beeninstrumental in the successful completion of our project. We would like to express our immensepleasure to thank our college management for giving required amenities regarding our project.

We would like to convey our sincere thanks to our respected Principal, **Dr.S.Arivazhagan, M.E.,Ph.D.,** Mepco Schlenk Engineering College, for providing us with facilities to complete our project.

We extend our profound gratitude and heartfelt thanks to **Dr.T.Revathi, M.E.,Ph.D.,** Senior Professor and Head of the Department of Information Technology for providing us constant encouragement.

We are bound to thank our project coordinator **Dr.G.YOGARAJAN, B.E, M.E., Ph.D.,** Associate Professor of Information Technology. We sincerely thank our project guide **Ms.M.Blessa Binolin Pepsi, BTech, ME, (PhD).,** Assistant Professor (Sr.G.) Department of Information Technology, for her inspiring guidance and valuable suggestions to complete our project successfully.

The guidance and support received from all the staff members and lab technicians of our department who contributed to our project, was vital for the success of the project. We aregrateful for their constant support and help.

ABSTRACT

ABSTRACT

This abstract discusses the secure sharing of medical images is of paramount importance in the healthcare industry, where sensitive patient information must be handled with utmost care. Medical images, such as X-rays, MRI scans, and CT scans, contain valuable diagnostic information, and their timely sharing between healthcare professionals can lead to improved patient outcomes. However, the transmission of such sensitive information can be vulnerable to cyber threats, which can compromise patient privacy and confidentiality. To address this issue, the proposed method employs robust encryption algorithms to ensure the confidentiality of patient information. Encryption involves transforming the medical image into a cipher text that is unreadable to unauthorized users. This step helps protect sensitive patient data from cyber threats and ensures that only authorized healthcare professionals can access the image. The Chinese remainder theorem is a mathematical principle that is used to enhance the security of transmitting the encrypted medical image. The theorem allows for the image to be broken down into smaller, more manageable parts that can be securely transmitted to the intended recipient. This approach also reduces the risk of data loss or corruption during transmission, making it a reliable and efficient method for sharing medical images. By combining encryption with the Chinese remainder theorem, healthcare professionals can securely and efficiently share medical images, leading to improved diagnosis and treatment of patients. This method ensures that patient privacy and confidentiality are maintained throughout the transmission process, adhering to the ethical and legal obligations of healthcare professionals. Overall, this approach can significantly contribute to the secure and reliable analysis of medical images while ensuring the highest levels of patient confidentiality. Deep learning has been shown to be highly effective in medical image classification tasks. Medical image classification involves identifying and labeling various structures or abnormalities within medical images, such as X-rays, MRI scans, CT scans, and ultrasound images. Deep learning algorithms can automatically learn to identify patterns and features within these images, allowing them to accurately classify the images. Tumor detection: Deep learning algorithms can be used to identify and classify different types of tumors in medical images, such as brain tumors, breast tumors, and lung tumors. Xception is a neural network architecture designed to enhance the performance and efficiency of the Inception network. This approach, called "depth wise separable convolutions," reduces computational costs while preserving accuracy. The convolution operation is separated into two components: depth wise and pointwise convolutions. Medical image classification is one application of Xception, which shows promise in identifying diseases and abnormalities in MRI and CT scans. Xception's high accuracy can help medical professionals make better diagnoses and treatment decisions, and its computational efficiency makes it practical for resource-limited settings

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES

LIST OF SYMBOLS

LIST OF SYMBOLS

NOTATION	MEANING
X	Data Matrix
K	Key
a_i, m_i	Pairwise Relatively Prime Integers
M_j	Product Of Relative Prime m_i
N_j	The Modular Inverse Of M_j

LIST OF ABBREVATION

LIST OF ABBREVATION

S.NO	ACRONYMS	ABBREVIATIONS
1	CRT	Chinese Remainder Theorem
2	RSA	Rivest-Shamir-Adleman
3	PSNA	Peak Signal to Noise Ratio
4	dB	DeciBel

INTRODUCTION

CHAPTER 1
INTRODUCTION

Privacy is a major concern when it comes to medical data analysis, especially with the rise of deep learning models that require large amounts of data to achieve high accuracy. To ensure the privacy of sensitive medical data while still enabling the use of deep learning models, there has been a growing interest in privacy-preserving techniques for medical image analysis. Our proposed scheme aims to tackle this challenge by utilizing the RDH-ED technique and the Chinese Reminder Theorem to provide a practical and efficient solution for privacy-preserving deep learning in medical image analysis. Secure image sharing in cryptography refers to the process of encrypting an image and distributing it among a group of people in such a way that each person can only access a specific part of the image, and the complete image can only be reconstructed by combining the parts owned by all the participants. This technique is commonly used in applications that require the secure sharing of confidential images, such as medical images, military intelligence, or corporate secrets. The process of secure image sharing involves several steps. First, the image is encrypted using a cryptographic algorithm to protect it from unauthorized access. Then, the encrypted image is divided into a set of smaller shares, with each share containing a portion of the image data. These shares are distributed among the participants, with each participant receiving a unique share. To ensure security, the shares are typically distributed in a way that ensures that no individual participant can reconstruct the image on their own. This is achieved using various cryptographic techniques such as secret sharing, where the image is divided into shares such that a minimum number of shares are required to reconstruct the image, and each participant only receives a subset of shares. When the participants wish to view the image, they must combine their shares using a reconstruction algorithm, which uses the information from each share to reconstruct the original image. This process is done in such a way that the complete image is only revealed to those participants who hold a sufficient number of shares, while the rest of the shares remain encrypted and secure. Medical image classification using deep learning models refers to the process of using advanced artificial neural networks to analyze and classify medical images. This technique is commonly used in healthcare applications to help doctors and clinicians diagnose diseases, identify anomalies, and predict treatment outcomes. Deep learning is a subset of machine learning that involves the use of multi-layered neural networks to learn complex patterns and relationships from data. In the context of medical image classification, deep learning models are trained on large datasets of medical images, and they use these datasets to learn patterns and features that are indicative of certain diseases or

conditions. The process of medical image classification using deep learning typically involves several steps. First, a large dataset of medical images is collected and labeled by experts, indicating the presence or absence of a particular condition or disease. The labeled images are then used to train a deep learning model, such as a convolutional neural network (CNN), which learns to identify patterns and features in the images that are associated with the target condition. Once the model is trained, it can be used to classify new medical images that have not been seen before. The image is inputted into the model, and the model outputs a prediction of the probability of the image belonging to a particular class or condition. Medical image classification using deep learning has been applied to a wide range of healthcare applications, such as identifying cancerous tumors in medical images, detecting lung diseases in chest x-rays, and predicting the progression of Alzheimer's disease from brain scans. The use of deep learning models in medical image classification has the potential to greatly improve diagnostic accuracy and treatment outcomes, as well as reduce the workload of doctors and clinicians.

1.1 The RDH-ED technique for medical image encryption

The RDH-ED (Reversible Data Hiding and Encryption based on Double-chaotic map and Error Diffusion) technique is a widely used steganography technique that is commonly used for both data hiding and encryption purposes. This technique involves embedding a message into the least significant bits of the cover image, which can be used for encryption purposes. The RDH-ED technique is particularly useful in scenarios where the sensitive information needs to be kept confidential while still allowing the use of the cover image for analysis and research purposes. Medical images are one such example of sensitive data that require robust encryption techniques. The RDH-ED technique can be used to encrypt medical images, thereby preventing unauthorized access to the sensitive medical data while still enabling their use for deep learning analysis. However, using this technique alone may not be efficient enough, especially when dealing with large datasets or real-time analysis. To address this limitation, researchers have proposed combining the RDH-ED technique with other encryption methods such as the Advanced Encryption Standard (AES) and the RSA algorithm. These methods can enhance the security of the RDH-ED technique and improve the efficiency of the encryption process. By using these complementary techniques, the sensitive medical data can be kept secure while still allowing for efficient analysis and processing. Another advantage of using the RDH-ED technique for encrypting medical images is its reversibility. This means that the original medical image can be completely restored after the encrypted message has been extracted, without any loss of image quality.

This feature is particularly important for medical applications where the accuracy of the image data is critical for diagnosis and treatment.

1.2 Chinese Reminder Theorem for improved security

To address the challenge of efficiently encrypting and decrypting large medical images while maintaining their privacy, our proposed scheme incorporates the Chinese Reminder Theorem. This mathematical tool is widely used in cryptography to enhance the efficiency of the encryption and decryption process. The Chinese Reminder Theorem breaks down a large message into smaller encrypted pieces using different keys, which can be recombined to create the overall encrypted message. By combining this theorem with the RDH-ED technique, we can efficiently encrypt and decrypt medical images without compromising their privacy. The use of the Chinese Reminder Theorem allows us to break medical images into smaller encrypted pieces that can be processed more efficiently by deep learning models. This is particularly important in medical image analysis, where large amounts of data must be analyzed quickly to produce accurate results. By using the Chinese Reminder Theorem, we can process these large amounts of data more efficiently, which in turn leads to more accurate and timely diagnoses. Overall, the incorporation of the Chinese Reminder Theorem into our proposed scheme improves the efficiency of medical image encryption and decryption while ensuring the privacy of the data.

1.3 Advantages of using the Chinese Reminder Theorem

In addition to improving the efficiency of the encryption and decryption process, the Chinese Reminder Theorem also offers greater flexibility in terms of key management. With this theorem, we can use multiple keys to encrypt and decrypt the same medical image, which enhances the security of the encryption process. This is because it makes it more difficult for an attacker to access the medical image without having all the keys necessary for decryption. Furthermore, using multiple keys enables more efficient transmission of the encrypted medical data over networks with varying security levels. For instance, a medical image may need to be transmitted from a low-security network to a high-security network. With multiple keys, we can encrypt the image using a set of keys suitable for the low-security network and then add an additional layer of security by encrypting the already encrypted image using a set of keys that are more suitable for the high-security network. This process ensures that the medical image remains secure throughout the transmission process, regardless of the security level of the networks involved.

1.4 Overall benefits of the proposed scheme

Our proposed scheme offers a practical and efficient solution for privacy-preserving deep

learning in medical image analysis. The combination of the RDH-ED technique and the Chinese Reminder Theorem provides a comprehensive approach that guarantees the privacy of sensitive medical data while still allowing for fast and accurate analysis by deep learning models. Moreover, the Chinese Reminder Theorem is a simple and easy-to-implement mathematical tool that enhances the efficiency of the encryption and decryption process. Its simplicity and practicality make it an essential component of our proposed scheme, ensuring that it can be easily adopted and integrated into existing medical image analysis systems. By incorporating the RDH-ED technique and the Chinese Reminder Theorem, our proposed scheme effectively addresses the challenges of maintaining data privacy in medical image analysis. It enables medical professionals to securely share and analyze medical data while still protecting sensitive patient information from unauthorized access and use. In summary, our proposed scheme offers a practical and efficient solution that enables the use of deep learning models for medical image analysis while still ensuring the privacy of sensitive medical data. The incorporation of the RDH-ED technique and the Chinese Reminder Theorem provides a comprehensive approach that is easy to implement, making it a valuable tool for enhancing medical diagnosis and treatment.

LITERATURE STUDY

CHAPTER 2
LITERATURE STUDY

2.1 OVERVIEW

This literature study discusses three different approaches for securing image data during transmission. The first approach is a visually secure image encryption scheme that uses compressive sensing and integer wavelet transform. The second approach proposes a new image encryption algorithm based on the two-dimensional Lorenz and Logistic maps. The third approach combines image encryption and image stitching techniques to provide double-layer protection to the image during communication.

Each approach has its unique features and advantages, but all of them aim to provide high security and robustness for the transmission of sensitive image data in various applications. The experimental results of these approaches demonstrate their effectiveness in terms of encryption and feasibility, making them suitable for secure image transmission. Overall, the literature study highlights the importance of image security and the need for developing robust and effective image encryption techniques to protect sensitive image data.

2.2 Visually Secure Image Encryption Scheme using Compressive Sensing and Integer Wavelet Transform

The security of image transmission is a critical concern, and traditional encryption schemes often generate cipher images that are meaningless and easy to attract attention. In order to address this issue, a visually secure image encryption scheme based on compressive sensing and integer wavelet transform is proposed in this paper. The proposed scheme consists of two phases: compression and encryption using compressive sensing, and embedding into a meaningful carrier image using the integer wavelet transform. In the first phase, the plain image is compressed and encrypted using compressive sensing with a sparse and random Bernoulli measurement matrix. The pixel positions of the compressed image are then permuted using a sequence generated from a 2D-LASM algorithm, which enhances the security of the embedding phase. In the second phase, a color image is used as a carrier image and the integer wavelet transform is applied to obtain a meaningful image. The carrier image is divided into coefficient matrices, and the compressed cipher image is embedded into the coefficient matrix to create the visually secure image. Experimental results demonstrate that the

proposed scheme provides good encryption and is feasible for practical applications. By generating a visually secure image, the proposed scheme overcomes the limitations of traditional encryption schemes that generate meaningless cipher images, making it a promising approach for secure image transmission. The proposed image encryption scheme is a meaningful and visually secure approach that effectively safeguards image data during transmission. The scheme employs two phases, each with specific functions to ensure high levels of security and robustness. In the first phase of the scheme, the plain image is compressed and encrypted using a sparse and random Bernoulli measurement matrix, which reduces the amount of data transmitted and increases the scheme's efficiency. Additionally, the pixel position of the compressed image is permuted using a sequence generated from a two-dimensional local adaptive shift map (2D-LASM), which adds an extra layer of security to the embedding phase. In the second phase, a carrier image is used to generate the visually secure image. The carrier image is divided into coefficient matrices, and the compressed cipher image is embedded into these matrices using the integer wavelet transform. This method ensures that the cipher image is visually indistinguishable from the carrier image and provides a high degree of security for the transmitted image. The experimental results demonstrate the effectiveness of the proposed scheme in terms of encryption and feasibility. The scheme successfully protects sensitive image data during transmission, making it suitable for use in a range of applications where image security is crucial.

2.3 Image Encryption Algorithm Based on Logistic and Two-Dimensional Lorenz

The security of image information has become increasingly important in recent years, leading to the proposal of numerous image encryption algorithms with high security. To further improve the security level of image encryption algorithms, this paper proposes a new image encryption algorithm based on two-dimensional Lorenz and Logistic. Encryption tests on several classic images demonstrate that the proposed algorithm has high security and strong robustness. In addition to presenting the encryption algorithm, this paper also analyzes its security using various methods, such as analysis of the histogram, entropy process of information, examination of correlation, differential attack, key sensitivity test, secret key space analysis, noise attacks, and contrast analysis. Through comparisons with other existing image encryption algorithms, the proposed algorithm is found to possess several desirable characteristics, such as a large secret key space, sensitivity to the key, small correlation coefficient, and high contrast. It is also shown to be capable of resisting noise attacks. The proposed image encryption algorithm based on two-dimensional Lorenz and Logistic has the potential to enhance the security of image transmission and storage. With its high security level and strong robustness, it is a promising approach for secure image encryption and can contribute to the

development of the field of information security. The Image Encryption Algorithm Based on Logistic and Two-Dimensional Lorenz is a novel approach to image encryption that utilizes the two-dimensional Lorenz and Logistic maps to create a secure and robust encryption method. This algorithm has been thoroughly tested against a range of attacks to ensure its effectiveness in protecting sensitive image data during transmission. The algorithm has been evaluated using various metrics, including the analysis of the histogram, entropy process of information, examination of correlation, differential attack, key sensitivity test, secret key space analysis, noise attacks, and contrast analysis. The results of these tests demonstrate the high security and strong robustness of the proposed algorithm. The algorithm's key features, such as the use of two chaotic maps and a dynamic key, contribute to its effectiveness in protecting against attacks. The algorithm's ability to produce a range of encrypted images from a single original image also adds to its versatility and practicality.

2.4 Image Security using Image Encryption and Image Stitching Model

In today's digital age, with the rise of digital communication and the use of multimedia data such as digital images and videos, it is essential to have fast and reliable security systems to safeguard against unauthorized access. Encryption is one of the most effective methods to ensure data security. In this paper, a proposed model combines image encryption and image stitching techniques to provide an extra layer of security to digital images. Chaotic encryption methods are employed to make it harder for potential attackers to decipher the encrypted image. The image is first partitioned into parts and then encrypted, with each part transmitted separately to the receiver. This means that an attacker with only one or two parts of the image will not be able to access the entire image. To reconstruct the original image, an image stitching algorithm is used on the receiver's end. The combination of image encryption and image stitching provides a dual layer of protection for the image, ensuring privacy and security during communication**.** Image security using image encryption and image stitching proposes a model that combines image encryption and image stitching techniques to provide double-layer protection to the image during communication. Image stitching is a process of combining multiple overlapping images into a single large image. This is typically done by aligning the overlapping regions of the images and blending them together to create a seamless final image. The resulting image can have a higher resolution and wider field of view than the individual images used to create it. Image stitching is often used in applications such as panoramic photography, where multiple images are taken from different positions to capture a wide landscape view. It is also used in medical imaging to create a single, comprehensive image from multiple scans, and in satellite imagery to create detailed maps of large areas. In the context of image security, image stitching can be used as a technique to protect

images by splitting them into multiple parts, encrypting each part separately, and then stitching them back together at the receiver's end to recreate the original image. This can provide an additional layer of security to the encryption process by making it more difficult for an attacker to obtain the complete image even if they manage to decrypt one or more parts of it. The proposed model partitions the image into several parts, encrypts each part individually, and then stitches them together at the receiver's end using an image stitching algorithm. The use of chaotic mapping makes it difficult for attackers to decrypt the image. The experimental results show that the proposed model has high security and privacy, making it suitable for applications that require secure image transmission, such as telemedicine and online banking. This approach not only provides a double layer of protection to the image but also ensures efficient transmission of the image data.

SYSTEM STUDY

CHAPTER 3
SYSTEM STUDY

3.1 SCOPE:

Medical images play a critical role in the diagnosis, treatment, and monitoring of various medical conditions. However, the sharing of these images poses significant challenges due to the sensitive patient data they contain. Unauthorized access or disclosure of this data could compromise patient privacy, undermine trust in healthcare institutions, and have serious legal and ethical consequences. To address these challenges, the scope is to discuss the importance of secure sharing of medical images in the healthcare industry. The proposed approach involves employing encryption algorithms and the Chinese remainder theorem to ensure the confidentiality and integrity of sensitive patient data during transmission. The abstract highlights the benefits of using this approach, such as improved patient outcomes, reduced risk of data loss or corruption, and adherence to ethical and legal obligations of healthcare professionals. The overall goal is to contribute to the secure and reliable analysis of medical images while maintaining patient privacy and confidentiality. By employing this method, healthcare professionals can ensure that medical image data is transmitted securely, efficiently, and ethically, improving patient outcomes and advancing medical research.

3.2 PRODUCT FUNCTION:

Image encryption based on the Chinese Remainder Theorem (CRT) can have several product functions, including:

1. Enhanced security: The Chinese Remainder Theorem (CRT) is a mathematical principle that has found extensive use in encryption algorithms. One of the applications of CRT is in image encryption, where it can be used to encrypt digital images securely. By using CRT for image encryption, data can be protected from unauthorized access and ensure its privacy during transmission over a network. This is particularly important in fields such as healthcare, finance, and government, where sensitive image data must be shared securely to prevent data breaches and maintain patient or client confidentiality. Image encryption using CRT works by breaking down the image into smaller components and encrypting them separately, which reduces the complexity of the encryption process and improves its efficiency. Furthermore, CRT-based encryption ensures that even if some components of the encrypted image are compromised, the rest of the image remains secure, thus enhancing the security of the encryption method.

2. Efficient encryption: The Chinese Remainder Theorem (CRT) is a mathematical principle that is widely used in cryptography. One of the advantages of using CRT is its efficiency in encrypting large amounts of data quickly and easily. This makes it a valuable tool for organizations that need to encrypt large volumes of image data, such as healthcare providers, financial institutions, and government agencies. By using CRT for image encryption, these organizations can save time and resources while still maintaining the security and confidentiality of their sensitive data. Additionally, the use of CRT in image encryption ensures that the encrypted data remains secure even if some parts of the data are compromised. This is because the data is broken down into smaller components and encrypted separately, which makes it more difficult for attackers to decrypt the data. Overall, the Chinese Remainder Theorem is an efficient and effective encryption algorithm that offers significant benefits to organizations that need to secure large amounts of image data.

3. Customizable encryption: The Chinese Remainder Theorem (CRT) is a mathematical principle that is commonly used in encryption algorithms. One of the advantages of using CRT is that it allows for customizable encryption settings. For example, users can select different relatively prime numbers to encrypt different parts of an image. This makes it possible to increase the complexity of the encryption and make it more difficult for attackers to break. By customizing the encryption settings, users can tailor the encryption to their specific needs and create a more secure encryption method. This is particularly important in industries such as healthcare, finance, and government, where the security and confidentiality of sensitive data are of utmost importance. Furthermore, using the Chinese Remainder Theorem for image encryption can also help to reduce the risk of data loss or corruption during transmission, as encrypted data is less susceptible to interference or manipulation. In summary, the customizable encryption settings of the Chinese Remainder Theorem offer a valuable tool for organizations that need to secure their sensitive data and protect it from unauthorized access.

4. Flexible decryption: The Chinese Remainder Theorem (CRT) is a mathematical principle that is often used in encryption algorithms. One of the advantages of using CRT is its flexibility in decryption methods. The theorem allows for the use of flexible decryption methods that can recover encrypted image data even if some parts of the image are lost or corrupted. This is particularly important in cases where decryption failure can result in significant data loss or corruption. By using flexible decryption methods, users can ensure the integrity of the image data and prevent data loss even in the event of a decryption failure. Furthermore, the use of the Chinese Remainder Theorem for image encryption and decryption provides an additional layer of security for sensitive data. Encrypted data is more difficult to access, making it less susceptible to unauthorized access and manipulation. In summary, the flexible

decryption methods of the Chinese Remainder Theorem offer a valuable tool for organizations that need to secure their sensitive image data and prevent data loss in the event of a decryption failure.

5. Compliance with industry standards: The Chinese Remainder Theorem (CRT) is a well-known and widely-used encryption algorithm that offers a high level of security for data encryption. Using CRT for image encryption can help organizations to comply with industry standards and regulations related to data security and privacy. For example, in the healthcare industry, the Health Insurance Portability and Accountability Act (HIPAA) requires that all protected health information (PHI) be encrypted to ensure the privacy and security of patient data. By using CRT for image encryption, healthcare organizations can ensure that they are in compliance with HIPAA regulations and protect the privacy and security of patient data. Similarly, in the financial industry, regulations such as the Payment Card Industry Data Security Standard (PCI DSS) require that all credit card data be encrypted during transmission. The use of CRT for image encryption can help financial organizations comply with these regulations and protect sensitive financial data. In summary, using the Chinese Remainder Theorem for image encryption can help organizations to comply with industry standards and regulations related to data security and privacy, ensuring the confidentiality and integrity of sensitive data.

Medical image classification using the Xception deep learning algorithm can have several product functions, including:

1. Automated diagnosis: The Xception deep learning algorithm is a powerful tool that can be used to automate the diagnosis of various medical conditions based on medical image data. Its ability to extract and analyze features from images makes it well-suited for tasks such as classification, segmentation, and detection. For example, Xception can be used to classify different types of tumors, such as breast cancer or lung cancer, based on medical image data. It can also be used to diagnose cardiovascular diseases by analyzing ultrasound or MRI images of the heart. Additionally, Xception can detect abnormalities in X-rays or CT scans, such as lung nodules or bone fractures. By automating the diagnosis process, Xception can help to reduce errors and improve the accuracy and efficiency of medical diagnosis. This can lead to earlier detection of medical conditions, better patient outcomes, and reduced healthcare costs. In summary, the Xception deep learning algorithm is a valuable tool for the automated diagnosis of various medical conditions based on medical image data.

2. Improved accuracy: Xception is a cutting-edge deep learning algorithm that is highly regarded for its accuracy and efficiency. It is widely used in various fields, including healthcare, where it has proven

to be an effective tool for medical image classification. By leveraging the capabilities of Xception, healthcare providers can enhance the accuracy of their diagnoses, leading to better patient outcomes. This is particularly crucial when dealing with complex medical conditions that require precise and timely diagnoses. By reducing the likelihood of misdiagnosis or false positives, Xception can help healthcare providers make more informed decisions and improve the overall quality of care they provide.

3. Reduced workload: One of the significant benefits of automating medical image classification using Xception is the potential to reduce the workload of healthcare professionals. Traditionally, medical image analysis is a time-consuming and labor-intensive process that requires specialized expertise. However, with Xception's advanced capabilities, medical image classification can be automated, allowing healthcare providers to quickly and accurately analyze large volumes of medical images with minimal effort. This can free up more time for healthcare professionals to focus on other essential tasks, such as patient care, research, and development. By automating medical image classification using Xception, healthcare providers can streamline their operations, enhance their efficiency, and ultimately deliver better outcomes for their patients.

4. Improved patient outcomes: The use of Xception for medical image classification has the potential to significantly improve patient outcomes in healthcare. With its high accuracy and efficiency, Xception can quickly and accurately diagnose medical conditions, enabling healthcare providers to intervene earlier and provide prompt and appropriate treatment. This can have a positive impact on patient outcomes by improving the success rates of treatments and reducing the likelihood of complications. For instance, in cancer diagnosis, early detection is critical, as it can increase the chances of successful treatment and improve the overall prognosis. By using Xception for medical image classification, healthcare providers can identify cancerous cells with greater accuracy and efficiency, allowing them to start treatment earlier and potentially save lives. Furthermore, Xception can also help healthcare providers diagnose other medical conditions, such as cardiovascular disease and neurological disorders, with similar benefits for patients. Overall, Xception represents a powerful tool that can significantly improve the quality of healthcare and help healthcare providers deliver better outcomes for their patients.

3.3 SYSTEM REQUIREMENTS

The software and hardware requirements of the system are as follows:

3.3.1 HARDWARE INTERFACES

- Intel® Core™ i5-8265U 1.6GHz
- 8 GB RAM

3.3.2 SOFTWARE INTERFACES

- Platform – Google Colab
- IDE – Google Colab
- Technologies used – Python

3.3.2.1 GOOGLE COLAB

Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your team members - just the way you edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in your notebook.

3.3.2.2 GOOGLE COLAB IDE

To Colaboratory (aka Google colab) is a research tool for machine learning education and research. Colaboratory is based on Jupyter and Jupyter notebooks (IPython notebooks) can beused and shared without having to download, install, or run anything on your own computer. Colaboratory supports Python 2.7 and Python 3.6. Data analysis and visualisation is frequentlydone using pandas.

3.3.2.3 PYTHON

Python is an interpreter, high-level data structures, general-purpose programming language. It can be used for creating web applications on server side. Python is also suitable as an extension language for customized applications.

SYSTEM DESIGN

CHAPTER 4
SYSTEM DESIGN

4.1 OVERVIEW

In this proposed approach, we aim to achieve secure and private medical image sharing by combining reversible data hiding and image encryption with Chinese remainder theorem and deep learning techniques. The approach consists of several components, each of which is described in detail in the following subsections. The reversible data hiding technique is used to embed the encrypted medical image in a cover image in a way that preserves the integrity and confidentiality of the original medical image. The image encryption technique utilizes Chinese remainder theorem and deep learning to provide secure encryption of the medical image. The Chinese remainder theorem is used to split the image into smaller blocks, which are then encrypted using a deep learning-based encryption algorithm. Overall, our proposed approach offers a robust and effective solution for secure and private medical image sharing.

4.2 OVERALL ARCHITECTURE:

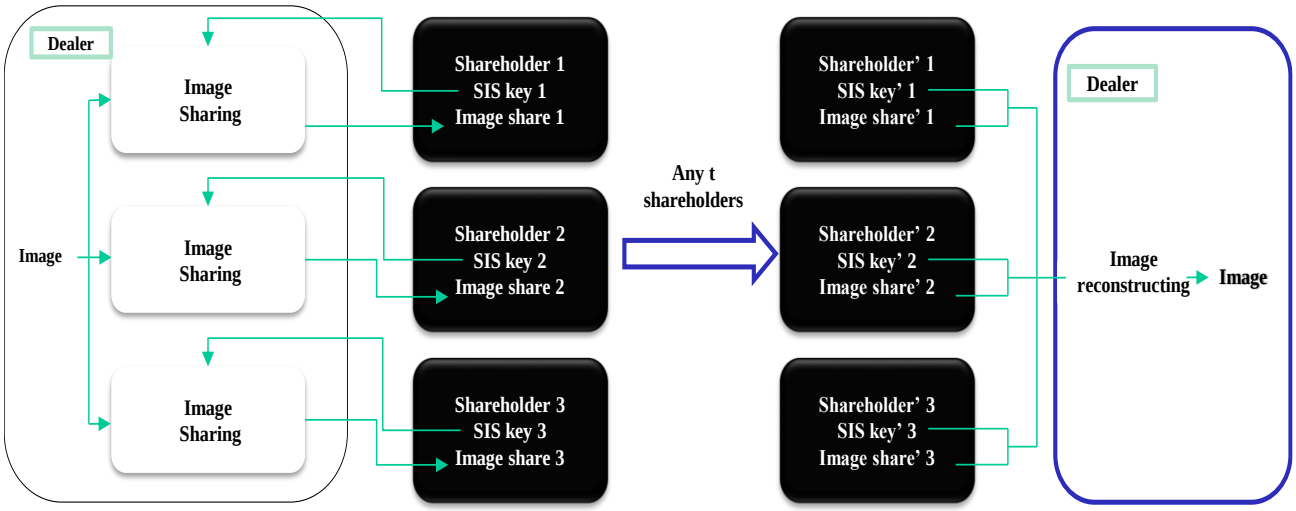


Figure 1:Image Sharing with different key and different receivers

Image sharing with different key and different receivers is a cryptographic technique that allows an image to be divided into multiple shares, and each share can be distributed to a different receiver with a unique decryption key. This technique provides a high level of security, as no single

receiver has access to the complete image without obtaining all the shares and decryption keys. In this technique, the original image is divided into multiple shares using a secret sharing scheme, such as Shamir's secret sharing algorithm. Each share is then encrypted with a unique decryption key, and the encrypted shares are distributed to different receivers. The receivers can decrypt their respective shares using their decryption key and combine them to obtain the original image. The use of different keys and different receivers makes it difficult for an attacker to obtain the complete image. Even if an attacker gains access to some shares and their respective decryption keys, they still cannot obtain the complete image without obtaining all the shares and decryption keys. This makes the image sharing technique more secure compared to traditional encryption methods, where a single key is used for encryption and decryption, and the attacker only needs to obtain the key to access the complete image.

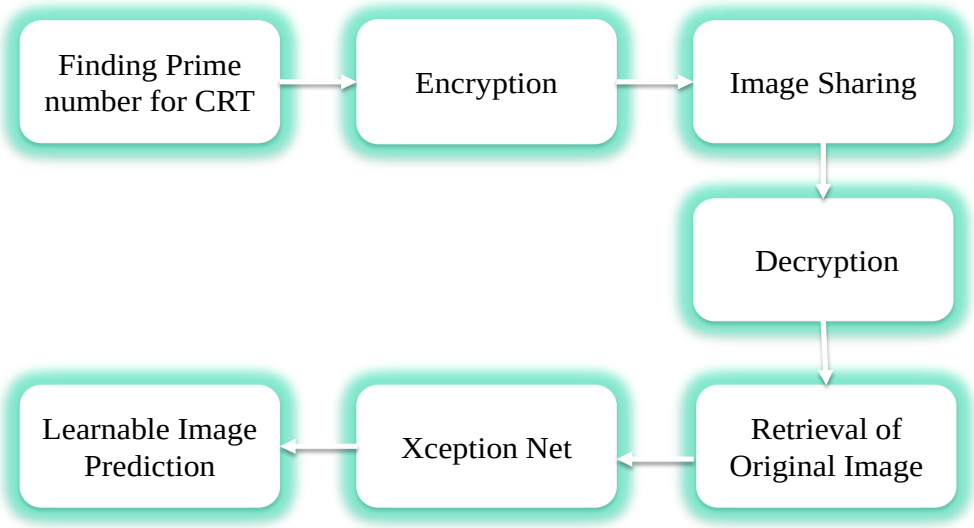


Figure 2:Secure Image Sharing and Disease Prediction using Xception Model

Image sharing with different key and different receivers is a technique used to securely distribute an image among multiple recipients. In this technique, the original image is first encrypted using a strong encryption algorithm with different keys for each recipient. These encrypted images are then securely shared with the intended recipients. To ensure the security and privacy of the image, it is important that the encryption keys are kept secret and only shared with the intended recipients. This way, even if one key is compromised, the image cannot be decrypted by anyone else. Once the encrypted images are received by the recipients, they can use their respective keys to decrypt the image. The decrypted image can then be further analyzed or processed for various purposes, such as disease prediction using an Xception model. Xception is a deep learning model that is commonly used for image classification tasks, including disease prediction. By feeding the decrypted image to an Xception

model, we can obtain predictions about the presence or absence of a particular disease in the image. Image sharing with different key and different receivers combined with the use of an Xception model provides a robust and secure approach for image analysis and disease prediction.

4.3 SECURE IMAGE SHARING BASED ON CRT

Secure image sharing based on CRT is a technique used to securely transmit medical image data over a network. This can be achieved using the Chinese Remainder Theorem (CRT), a mathematical theorem commonly used in cryptography. CRT-based encryption divides the original image into sub-images, encrypts each sub-image using a different set of keys, and transmits them over a secure channel. At the other end, the sub-images are decrypted using the corresponding keys and recombined to form the original image. This process provides a high level of security and privacy protection for the medical image data, making it difficult for unauthorized parties to access or decrypt the data. CRT-based encryption can be applied to a wide range of medical image formats and types, making it a versatile and flexible encryption technique. In addition to providing enhanced privacy protection, CRT-based encryption also offers a number of other benefits for medical image transmission. For example, the use of multiple keys and sub-images helps to ensure that the image data remains intact and free from distortion during transmission. This is particularly important for medical images, where even small distortions or errors can have a significant impact on patient diagnosis and treatment.

4.3.1ENCRYPTION

The Chinese Remainder Theorem (CRT) is a mathematical concept that is used in encryption techniques to securely transmit medical image data over a network. The basis of CRT lies in the idea that if we have a set of equations that are congruent modulo some pairwise relatively prime integers, we can determine the solution uniquely modulo the product of those integers. In the context of medical image encryption, this means that we can break up the image data into smaller pieces and encrypt each piece separately using different keys.

By using different keys for each sub-image, we ensure that the image data is secure and protected from unauthorized access or decryption. Once the sub-images are encrypted, we can use CRT to combine them into a single cipher text, which can be sent over the network. This process ensures that the medical image data is protected and remains confidential during transmission.

Encryption is an essential part of the CRT-based encryption process. In the medical image encryption context, encryption involves converting the original image data into an unreadable format

that can only be deciphered using a specific key. This key is only available to authorized personnel, such as healthcare professionals, and ensures that the image data remains secure and confidential.

Overall, CRT-based encryption is a highly effective way to ensure the secure transmission of medical image data. By using different keys for each sub-image and combining them using CRT, we can protect the data from unauthorized access or decryption. Encryption is a critical component of the process, as it ensures that the original image data is converted into an unreadable format that can only be deciphered using a specific key. With the use of CRT-based encryption, we can ensure that medical image data remains confidential and secure during transmission, allowing healthcare professionals to provide more efficient and effective medical imaging practices.

4.3.2 DECRYPTION

To decrypt the cipher text, the receiver must have the corresponding decryption keys for each of the encrypted pieces. Then, they can use CRT to recover the original image data by computing the solution modulo the product of the pairwise relatively prime integers. CRT provides a powerful and efficient method for encrypting medical image data and ensuring its privacy during transmission over a network. During the decryption process of a stego-image, the embedded encrypted image is extracted from the cover image by reversing the embedding process. This involves dividing the stego-image into small blocks or segments, and then extracting the embedded encrypted segments from the cover image segments. Once the encrypted image is extracted, it can be decrypted using the inverse of the key and the Chinese Remainder Theorem (CRT), as described earlier. The decrypted image segments can then be combined to reconstruct the original encrypted image

4.4 Reversible Data Hiding

Reversible data hiding is a technique used in digital image processing that allows data to be embedded into an image without causing any permanent changes to the pixel values of the image. This means that the original image can be recovered without any loss of quality or information, even after the embedded data has been extracted. This technique is often used in applications where it is important to protect privacy or confidentiality, such as in medical imaging.

In medical imaging, reversible data hiding can be used to embed patient information directly into the medical image itself, before the image is encrypted for transmission or storage. This allows the patient information to be securely stored and transmitted along with the image, without the need for a separate data file or record. The embedded information can only be extracted by authorized

healthcare professionals, who have the necessary decryption keys and software. Another advantage of reversible data hiding is that it can be used to embed data in a way that is imperceptible to the human eye. This means that the embedded data can be hidden in the image without being visible to the viewer, making it difficult for unauthorized parties to detect or access the embedded information.

Overall, reversible data hiding is a powerful technique for embedding data in digital images, while ensuring that the original image can be recovered without any loss of quality or information. In the context of medical imaging, this technique can be used to embed patient information securely and confidentially, providing enhanced privacy protection for patients and facilitating the secure storage and transmission of medical images.

4.5 XCEPTION

Xception is a deep learning architecture that has gained popularity due to its ability to efficiently and accurately extract features from images. The network is based on the Inception architecture, but it incorporates modifications that allow it to achieve high accuracy with fewer parameters and computations. One of the key features of Xception is its use of depth-wise separable convolutions, which break down the convolution operation into two separate operations: a depth-wise convolution that applies a single filter to each input channel, followed by a point-wise convolution that combines the outputs of the depth-wise convolution. This approach allows for a significant reduction in the number of parameters and computations required for feature extraction, while maintaining high accuracy. In addition to its use of depth-wise separable convolutions, Xception also incorporates other modifications to the Inception architecture, such as the removal of fully connected layers and the use of a linear activation function. These changes allow Xception to achieve high accuracy on a variety of image recognition tasks, while also improving its computational efficiency. In addition to its applications in image recognition, Xception has also been used in other areas of deep learning, such as natural language processing and speech recognition. Its efficient feature extraction capabilities make it well-suited for tasks such as text classification and language modeling, where the input data is often high-dimensional and computationally expensive to process. Xception has also been used as a pre-trained model in transfer learning applications, where it is used to extract features from images for downstream tasks such as object detection and segmentation. By leveraging the pre-trained weights of the Xception model, transfer learning enables faster and more efficient training of new models on smaller datasets, which is particularly useful in applications where large amounts of training data may not be available. Despite its many advantages, Xception is not without its limitations. One potential drawback is its sensitivity to hyperparameters such as learning rate and batch size, which can affect

the convergence of the model during training. Additionally, the depth-wise separable convolution operation can lead to a reduction in model capacity, which may limit its performance on certain tasks. Xception represents a significant advancement in deep learning architecture, offering a powerful and efficient method for feature extraction from images. Its applications extend beyond image recognition to other areas of deep learning, and its pre-trained weights have been used in transfer learning to achieve state-of-the-art performance on a variety of tasks. As deep learning continues to evolve, it is likely that Xception and other similar architectures will play an increasingly important role in advancing the state of the art in artificial intelligence. Xception is a powerful deep learning architecture that offers significant advantages over traditional convolutional neural networks. Its ability to efficiently extract features from images makes it well-suited for applications where computational resources are limited, such as in mobile and embedded systems. As the field of deep learning continues to evolve, architectures like Xception are likely to play an increasingly important role in enabling new applications and advancing the state of the art in image recognition and other related fields.

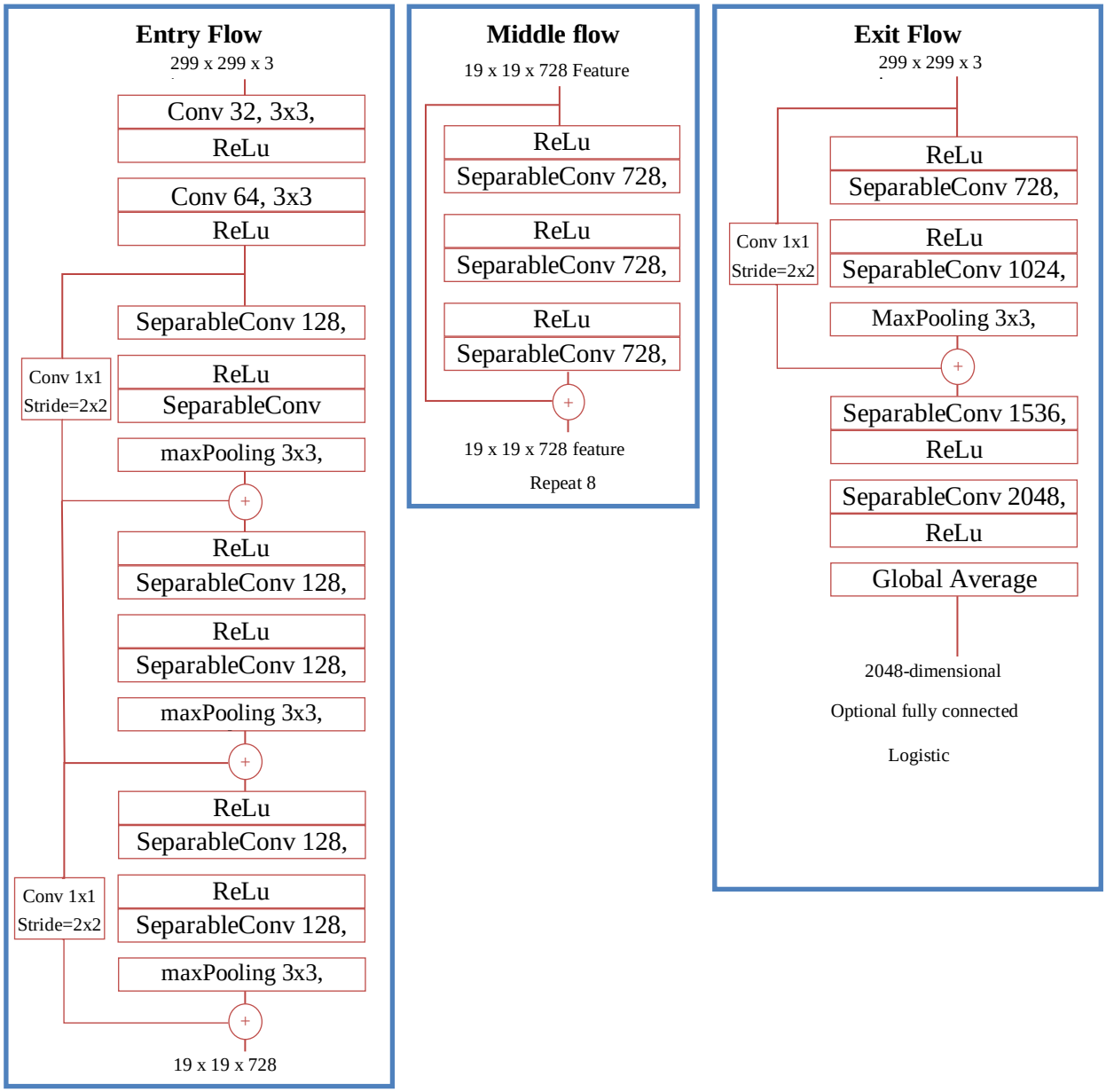


Figure 3:Xception's Depthwise Separable Convolutions for Efficient Image Analysis

The depth wise separable convolutions in Xception are designed to capture spatial dependencies between neighboring pixels in an image, while reducing the computational complexity of the network.

1. Entry flow: This is the initial stage of the Xception network where the input image is processed by a series of convolutional and pooling layers to extract basic features from the image. The entry flow consists of several parallel convolutional blocks that extract different types of features at different scales.

2. Middle flow: After the features are extracted in the entry flow, they are passed through a

series of middle flow modules. These modules contain multiple residual blocks that help to increase the depth of the network and improve its accuracy.

3. Exit flow: In the final stage of the Xception network, the features are aggregated and passed through a series of fully connected layers to make a prediction. The exit flow contains a combination of global average pooling and fully connected layers, which help to reduce the number of parameters in the network and prevent overfitting.

Overall, Xception is a powerful deep learning architecture that is designed for efficient and accurate feature extraction from images, and is well-suited for large-scale image analysis tasks

IMPLEMENTATION METHODOLOGY

CHAPTER 5
IMPLEMENTATION METHODOLOGY

5.1 OVERVIEW:

Our proposed approach offers a novel solution for secure and private medical image sharing. The approach combines reversible data hiding and image encryption with Chinese remainder theorem and deep learning techniques to ensure the confidentiality and integrity of medical images. Reversible data hiding allows for the embedding of sensitive data into the image while preserving its original content. Image encryption ensures that the data embedded in the image is secure and cannot be accessed by unauthorized parties. The Chinese remainder theorem is used to split the image into multiple shares, ensuring that even if one share is compromised, the original image cannot be reconstructed without all shares. Deep learning techniques are then used to enhance the security of the system by detecting any potential attacks and preventing unauthorized access. In the following subsections, each component of our approach is described in detail, highlighting its role in ensuring the secure and private sharing of medical images. The proposed approach offers a promising solution to the challenge of medical image sharing, providing a high level of security and privacy that is critical for sensitive medical information.

5.2 ESSENTIAL LIBRARIES:

The library used in this project are NumPy, math, SciPy, functools, matplotlib, TensorFlow,Keras.

5.2.1 NUMPY

NumPy is a library for the Python language. It can be used for large, multi-dimensionalarrays and matrices. It is used along with a huge collection of high-level mathematical functionsto operate on these arrays. It is precisely useful for algorithm developers. The processing will be slow when working with multi-dimensional arrays and the functions and operators that are used.NumPy is useful to overcome this restriction. Any datatypes can be defined by this package.

5.2.2 MATH:

The Math Library provides us access to some common mathematical functions and constants in Python, which we can use in our code for complex mathematical computations. The library is an in-built Python module, so we don't have to do any installation. The mathematical function used in this project is as follows:

5.2.3 PANDAS

Pandas is a software library written for the Python programming language for datamanipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three- clause BSD license.

5.2.4 MATHPLOT

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open-source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

5.2.5 FUNCTOOLS:

The Functools module is for higher-order functions. It acts on or return other functions. In general, any callable object can be preserved as a function for the purposes of this module.

5.2.6 REDUCE():

The reduce() function is used to apply a specific function passed in its argument to all of the list elements stated in the sequence passed along. It is used to convert 2-d list into 1-d list in our project.

5.2.7 MATPLOTLIB:

Matplotlib is one of the most common packages used for data visualization in python. Itis a cross-platform library for creating 2-Dimensional plots from data in arrays. Matplotlib iswritten in Python. Matplotlib with NumPy can be used as the open source.

The matplotlib.pyplot is a collection of command style functions that make matplotlibwork like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plotwith labels, etc. In this project matplotlib.pyplot is used to generate graph such as Bar graph, Cartesian graphs which represents the different Parameters Vs performance of the algorithm.

5.2.8 KERAS

Keras is a high-level neural network API written in Python. It is designed to enable fast experimentation with deep neural networks, while also being user-friendly and modular. Keras can run on top of various deep learning libraries such as TensorFlow, Theano, and CNTK.

Keras provides a simple and consistent interface for defining and training neural networks. It allows users to define neural networks as a sequence of layers, where each layer performs a specific type of computation on the input data. Keras supports a wide range of layers, including convolutional layers, recurrent layers, and dense layers.

5.2.9 TENSORFLOW

TensorFlow is an open-source software library for dataflow and differentiable programming across a range of tasks, such as machine learning, deep learning, and scientific computing. It was developed by the Google Brain team and released in 2015. TensorFlow provides a flexible and efficient programming model for building and training machine learning models, and has become one of the most widely used deep learning libraries in the world.

TensorFlow represents computations as a directed graph, where nodes in the graph represent mathematical operations, and edges represent the flow of data between them. The graph is built using TensorFlow's high-level APIs, such as Keras, or through the lower-level TensorFlow API. TensorFlow allows users to define and train complex deep learning models, with support for a wide range of architectures, including convolutional neural networks, recurrent neural networks, and transformers.

5.3 FUNCTIONS USED FOR IMPLEMENTATION:

5.3.1 Dataset:

Our brain tumor dataset consists of 2,880 medical images, which are categorized into four different classes: no tumor, meningioma tumor, pituitary tumor, and glioma tumor.

Each image in the dataset represents a 2D slice of a patient's brain captured using medical imaging techniques such as magnetic resonance imaging (MRI) or computed tomography (CT). The images are of varying sizes and aspect ratios, which can cause issues when training a deep learning model. Therefore, as mentioned before, the images have been preprocessed to a uniform size of 256x256 pixels to standardize them for the model.

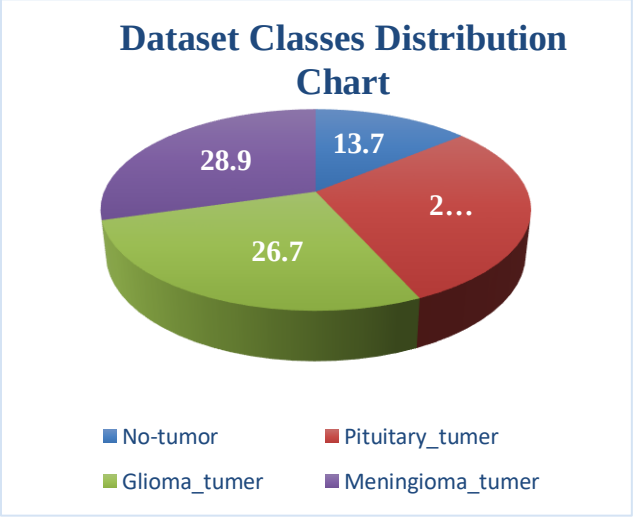


Figure 4:Dataset Distribution in Our Project for Tumor Detection

The Diagram showing dataset distribution in our project and this Figure showing dataset distribution in our project, glioma tumor has 26.7% of data, pituitary tumor has 28.7% of data, no tumor has 13.7% of the data, meningioma tumor has 28.9% of data in our dataset.

The no tumor class in the dataset includes images of patients who have no visible tumors in their brain scans. The meningioma tumor class includes images of patients who have meningioma brain tumors, which are typically benign but can still cause symptoms and require treatment. The pituitary tumor class includes images of patients who have pituitary gland tumors, which can cause hormonal imbalances and other health problems. The glioma tumor class includes images of patients who have glioma brain tumors, which are typically malignant and can be very aggressive.

The dataset is labeled with the correct class for each image, which is used to train the deep learning model to correctly classify new images.

5.3.2 Key Generation:

Image encryption is a crucial component of secure and private medical image sharing. In this process, a key is required to secure the image data and prevent unauthorized access. One commonly used approach to generate this key is through the RSA algorithm, which is a widely used public-key cryptographic system. The RSA algorithm relies on the difficulty of factoring large integers to ensure the security of the key. By generating two large prime numbers and performing a series of calculations, the RSA algorithm produces a public and a private key pair that can be used to encrypt and decrypt messages or data, including digital images. The generated key is an essential parameter for the Chinese Remainder Theorem (CRT)-based image encryption process, which helps to protect the confidentiality and integrity of the image data. The CRT-based image encryption process divides the image into multiple shares that are encrypted using the generated key. The shares are then distributed to authorized parties, and only by combining all the shares using the CRT technique can the original image be reconstructed. This provides an additional layer of security and ensures that even if one share is compromised, the original image remains protected. Overall, the combination of the RSA algorithm and the CRT-based image encryption process provides a robust and secure solution for medical image sharing.

5.3.3 Inverse Key Calculation:

The process of encrypting an image using the RSA-based key generated for the Chinese Remainder Theorem (CRT)-based image encryption process involves several steps. Firstly, the inverse of the key must be calculated using the Extended Euclidean Algorithm. The multiplicative inverse of

the key is a value that, when multiplied by the key, results in a remainder of 1 when divided by the modulus. Once the inverse of the key has been calculated, it can be used in conjunction with the Chinese Remainder Theorem (CRT) to encrypt the image data. The CRT-based image encryption process helps to protect the confidentiality and integrity of the image data by breaking it down into smaller parts and encrypting each part using the RSA key and its inverse. The CRT-based approach ensures that even if one share is compromised, the original image cannot be reconstructed without all shares. This provides an additional layer of security, ensuring that the image data is protected from unauthorized access. The RSA-based key and the CRT-based image encryption process provide a robust and secure solution for medical image sharing, enabling sensitive medical data to be shared safely and securely.

5.3.4 Encryption:

In order to encrypt an image using the key generated by the RSA algorithm, it is first necessary to convert the image into a pixel matrix. This involves reading the image file and extracting the red, green, and blue (RGB) values of each pixel in the image. These RGB values can then be represented as a matrix, with each element in the matrix corresponding to a single pixel in the image. Once the image has been converted to a matrix, the encryption process can be performed using the RSA-based key and the Chinese Remainder Theorem (CRT)-based image encryption process. By encrypting the pixel values using this approach, the confidentiality and integrity of the image data can be ensured, protecting sensitive medical information from unauthorized access. In summary, the process of converting an image into a pixel matrix is an essential step in the encryption process and forms the basis for ensuring the security of medical image sharing.

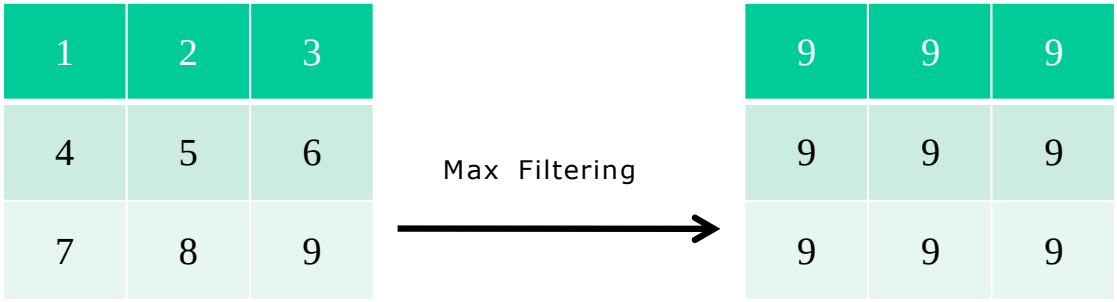


Figure 5:Max Filtering in Proposed System

This Diagram for max filtering it is in our proposed system the Max Filtering is done by the approach given in the diagram.

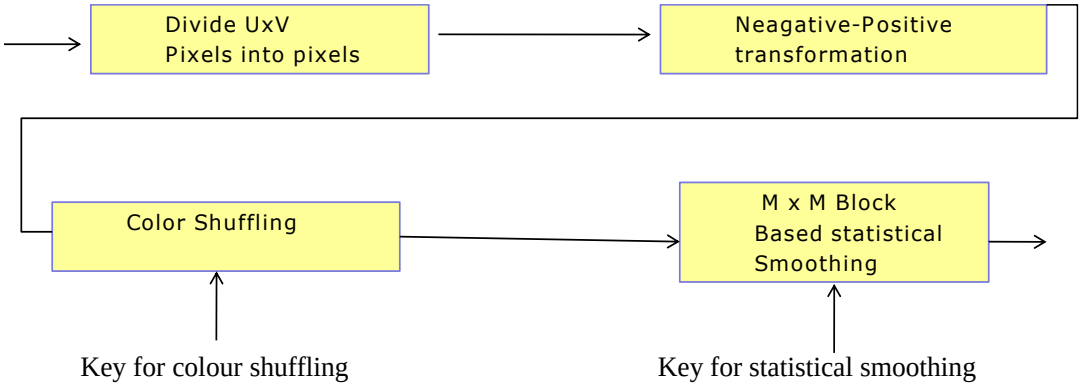


Figure 6:Encryption with Color Shuffling Diagram

This Diagram for encryption with color shuffling it is in our proposed system for encryption with color shuffling done based on this diagram.

Once we have the pixel matrix, we can perform the encryption process based on the generated key. The encryption process typically involves breaking the pixel matrix into smaller blocks or segments, and applying the Chinese Remainder Theorem (CRT) to each block using the RSA key and its inverse. The CRT-based encryption process helps to protect the confidentiality and integrity of the image by encrypting each segment independently and preventing an attacker from easily identifying patterns in the encrypted image.

COVER IMAGE **ORIGINAL IMAGE**

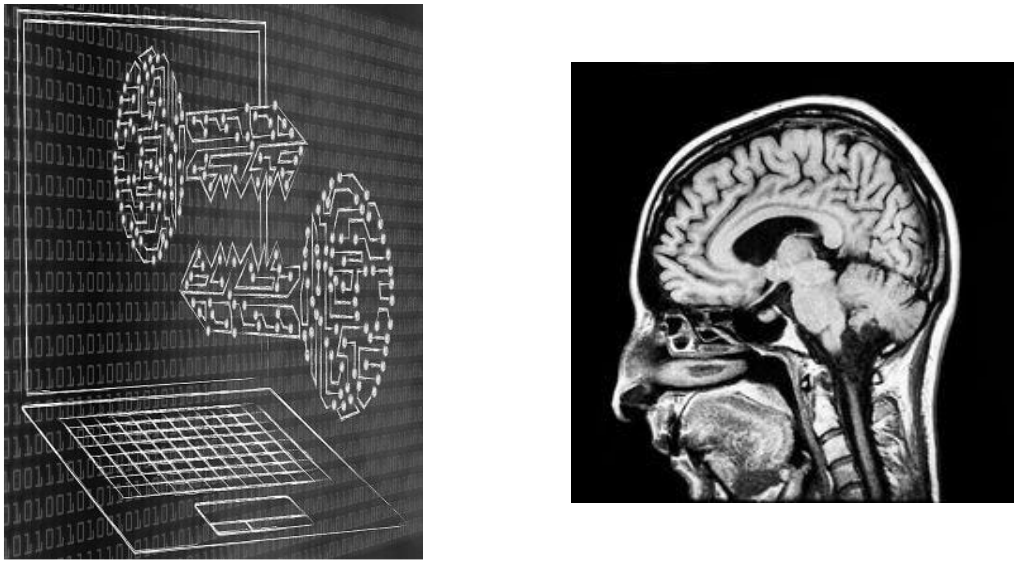


Figure 7: Color Shuffling and CRT Diagram

In order to enhance the security of image transmission, various techniques can be used, such as encryption with color shuffling and the Chinese Remainder Theorem (CRT). The given diagram

represents the process of encryption with color shuffling and CRT, where a prime number is generated using the RSA algorithm and encrypted with CRT. The cover image and the original image are both encrypted using this technique to ensure a more secure transmission of the image data. The color shuffling technique is used to further enhance the security of the encryption process by shuffling the color distribution of the image before encryption. This makes it more difficult for attackers to detect or intercept the transmission of the image data. By combining these techniques, a high level of security can be achieved, ensuring the confidentiality and integrity of the image data during transmission. This process is particularly important in situations where the image contains sensitive or confidential information that must be protected from unauthorized access or interception.

COVER IMAGE **ORIGINAL IMAGE**

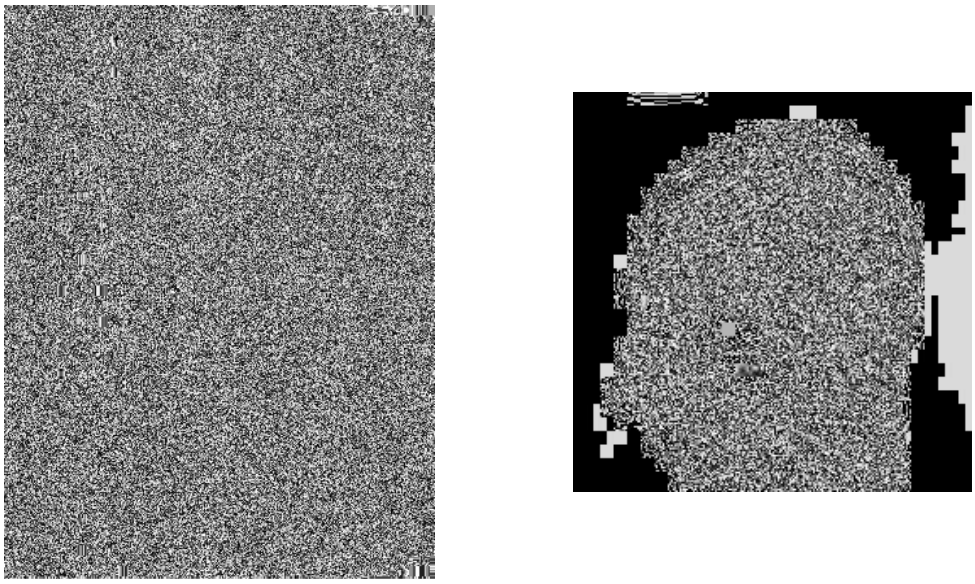


Figure 8:Encrypted Images with Color Shuffling and CRT Diagram

This diagram for encryption with color shuffling CRT and this diagram for encryption with color shuffling CRT, the first image is encrypted image of cover image and second image is the encrypted image of original image that to be transmitted.

Suppose we have an image data matrix X that we want to encrypt using CRT. We break up X into n smaller matrices X1, X2, ..., X_n, and we encrypt each matrix using a different key Ki. Let ai and mi be pairwise relatively prime integers such that:

X_1 is encrypted using key K_1 and is congruent to a_1 modulo m_1

X_2 is encrypted using key K_2 and is congruent to a_2 modulo m_2

...

X_n is encrypted using key K_n and is congruent to a modulo m_n

Then, according to CRT, there exists a unique solution X* modulo the product of the mi's, such that:

$X^* \equiv a_1 \pmod{m_1}$

$X^* \equiv a_2 \pmod{m_2}$

...

$X^* \equiv a_n \pmod{m_n}$

The receiver can then decrypt the cipher text by using the corresponding decryption keys to recover the smaller encrypted matrices X_1^*, X_2^*, ..., X_n^*, and combining them using the formula:

$X^* = (X_1^* * M_1 * N_1) + (X_2^* * M_2 * N_2) + ... + (X_n^* * M_n * N_n) \bmod (m_1 * m_2 * ... * m_n)$

where $M_j = (m_1 * m_2 * ... * m_n) / m_j$ and N_j is the modular inverse of M_j modulo m_j.

This formula allows the receiver to recover the original image data X* while ensuring its privacy during transmission over the network.

Measure	Image 1	Image 2
Entropy	5.934	6.175
PSNR	29.1166 dB	
Correlation Coefficient	Min : 0.31673 Max : 0.31673 Mean : 0.31673	

Table 1:Measures for Original and Encrypted Images Comparison Table

This table contains various measures for image 1 and 2. This is table for various analysis of original image and encrypted image. The image 1 is original image and image 2 is encrypted image of the original image.

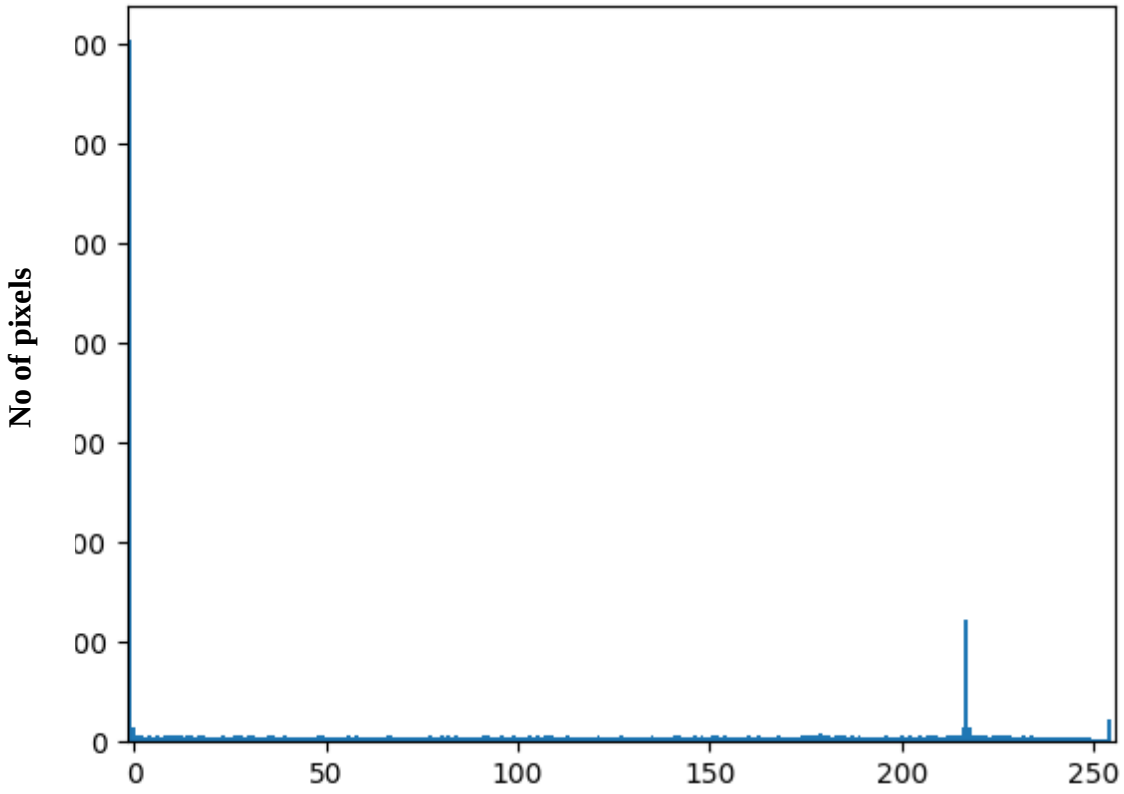


Figure 9:Encrypted Image Distribution Plot

The given diagram represents a plot for an encrypted image. The x-axis of the plot shows the distribution of colors in the encrypted image while the y-axis shows the number of pixels for each color in the image. Once the encryption process is completed, the encrypted pixel matrix can be securely saved to a file or transmitted over a network. To decrypt the encrypted image, the first step is to calculate the inverse of the key using the Extended Euclidean Algorithm. After obtaining the inverse of the key, the decryption process can be performed by applying the CRT-based decryption method. By using the inverse of the key, the encrypted image can be decrypted and the original image can be retrieved. This process ensures that the original image remains secure during transmission and can only be accessed by authorized individuals who possess the key.

5.3.5 Image Sharing:

To enhance the security of image transmission, there are various techniques available, one of which is image steganography. This technique involves embedding an encrypted image within another

image, known as a cover image. The cover image acts as a container for the encrypted image, which is hidden within it. This makes it difficult for an attacker to detect or intercept the transmission of the encrypted image. The process of embedding the encrypted image within the cover image is done in such a way that the cover image remains visually unchanged, ensuring that the transmission of the image remains inconspicuous. This technique can be used for secure transmission of sensitive images, such as those containing confidential information, or for transmitting images in an environment where interception or detection is a concern. By using image steganography, the security of image transmission can be enhanced, and the likelihood of unauthorized access or interception can be minimized.

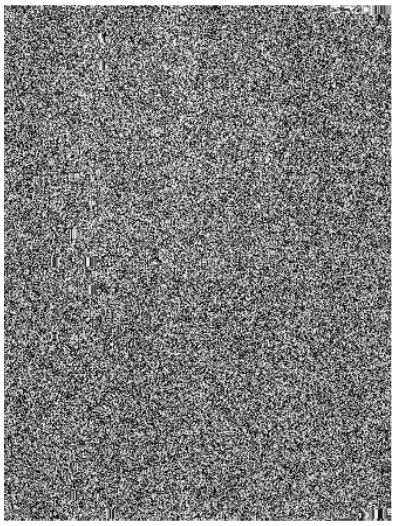


Figure 10:Stego-Image for Secure Sharing with Homomorphic Addition

In order to securely share an image, a technique called homomorphic addition or merging is often used. This involves creating a stego-image by merging an encrypted cover image with an encrypted original image. The resulting stego-image can then be split into a specified number of shares for secure sharing. To embed the encrypted image within the cover image, the cover image and the encrypted image are divided into small blocks or segments. The encrypted segments are then embedded within the corresponding cover image segments. This process ensures that the original cover image remains visually unchanged while the encrypted image is hidden within it. By splitting the resulting stego-image into shares, it becomes possible to securely share the image with a specified group of authorized individuals. This technique can be used for the secure sharing of sensitive images, such as those containing confidential information, or for transmitting images in an environment where interception or detection is a concern. This can be achieved by modifying the pixel values of the cover image segments to include the encrypted data, while minimizing any perceptible changes to the visual

appearance of the cover image. Once the embedding process is complete, the resulting stego-image can be transmitted over the network or stored securely. To recover the encrypted image, the stego-image is first decoded to extract the encrypted segments, which are then decrypted using the inverse of the key and the Chinese Remainder Theorem (CRT). The recovered encrypted image can then be reconstructed into its original form by combining the decrypted segments.

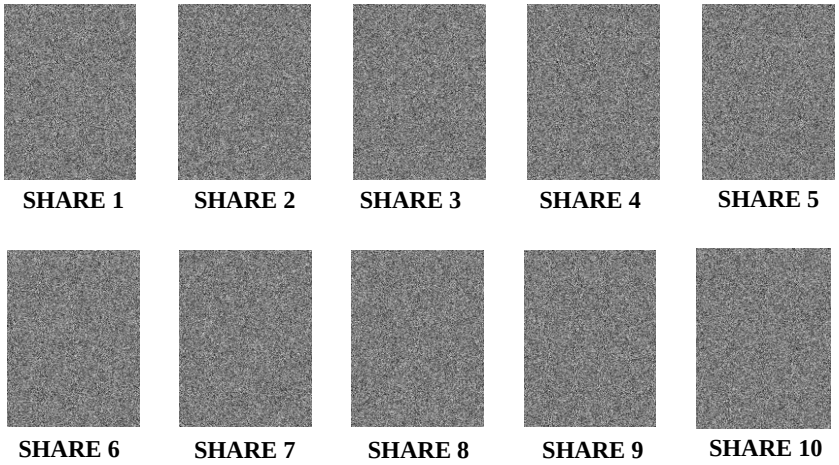


Figure 11:Various Shares for Secure Image Transmission

The secure sharing of images often involves the process of splitting the original image into a specified number of shares, which can then be transmitted securely to a receiver. The given image represents the various shares that are generated by a specific process, as shown in Figure 15. These shares are generated in such a way that they are secure and can be transmitted to the receiver without the risk of unauthorized access or interception. Once the receiver has obtained all the shares, they can be combined to reconstruct the original image. This process ensures that the image remains secure during transmission and can only be accessed by authorized individuals who possess the appropriate shares. Secure image sharing is important in situations where the image contains confidential or sensitive information that should not be accessed by unauthorized individuals. By using secure image sharing techniques, the confidentiality and integrity of the image can be maintained, ensuring that it is protected throughout the transmission process.

5.3.6 Decryption:

During the decryption process of a stego-image, the embedded encrypted image is extracted from the cover image by reversing the embedding process. This involves dividing the stego-image into small blocks or segments, and then extracting the embedded encrypted segments from the cover image segments.

Once the encrypted image is extracted, it can be decrypted using the inverse of the key and the Chinese Remainder Theorem (CRT), as described earlier. The decrypted image segments can then be combined to reconstruct the original encrypted image.

Finally, the original encrypted image can be decrypted using the key and the Chinese Remainder Theorem (CRT) to obtain the plain image. This process of decrypting the original image from the stego-image helps to ensure the confidentiality and integrity of the image data during transmission, as it prevents unauthorized access or modification of the data.

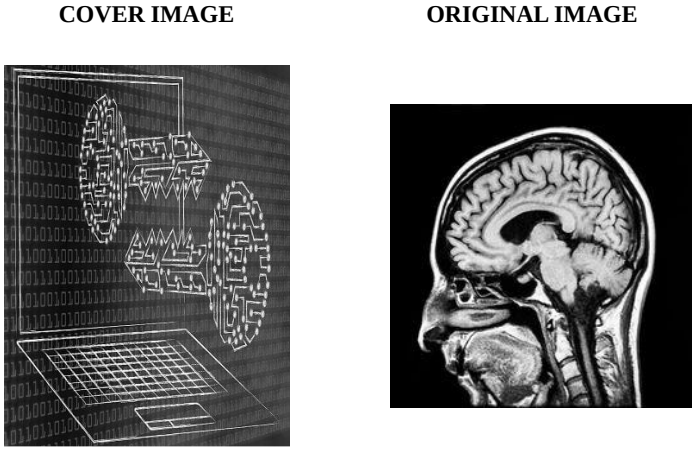


Figure 12:Decrypted Image from Combined Shares for Disease Prediction

Decrypted image of shared image and this image is decrypted by combining all the transmitted shares and homomorphic image is generated with the securely transmitted share, the homomorphic image is segmented into encrypted cover image and encrypted original image. By the decryption algorithm the encrypted original image is decrypted and it is sent to Xception model in order to predict the disease affected.

5.3.7 Model Generation

I used a dataset of 2,880 medical images, consisting of four different classes: no tumor, meningioma tumor, pituitary tumor, and glioma tumor. To prepare the dataset for training, I preprocessed the images to obtain a uniform size of 256x256 pixels. This preprocessing step helps to standardize the images and ensure that the model is not biased towards any particular size or aspect ratio.

The dataset was then used to train a deep learning model using the Xception algorithm. Xception is known for its high accuracy and computational efficiency in image classification tasks, making it a good choice for this project.

During the training process, hyper parameters such as the learning rate and batch size were carefully tuned to achieve optimal performance. Data augmentation techniques were also used to increase the size of the dataset and prevent overfitting.

After the model was trained, it was evaluated on a separate test set to measure its accuracy and generalization performance. The results showed that the model was able to accurately classify the four different tumor classes with high accuracy.

5.3.8 Accuracy Measure:

The accuracy of a machine learning model is a critical measure of its performance, as it indicates the percentage of correctly classified data points. In this case, we are presented with the accuracy values for five different models: AlexNet, DenseNet-121, ResNet18, GoogleNet, and Xception.

From the given accuracy values, we can see that the model with the highest accuracy is Xception, with a score of 94.27%. This suggests that Xception is a highly effective model for the given task, and may be well-suited for use in applications where high accuracy is essential.

The other models also have relatively high accuracy scores, ranging from 91.49% for ResNet18 to 93.57% for DenseNet-121. While these models may not be quite as accurate as Xception, they still demonstrate strong performance and may be suitable for a range of applications.

It's important to note that the accuracy of a model can be influenced by a variety of factors, including the size and quality of the training data, the complexity of the model architecture, and the choice of hyperparameters. As such, it's important to carefully evaluate model performance and consider multiple metrics before making decisions about which model to use for a given task.

Overall, the accuracy values presented here provide a useful starting point for evaluating the effectiveness of different models, and can help guide decisions about which model to use for a given application. However, it's important to keep in mind that accuracy is just one of many metrics that can be used to evaluate model performance, and that other factors such as model complexity and computational requirements may also need to be taken into account when making these decisions.

5.3.9 Running Time Measure:

Measuring running time is a crucial aspect of machine learning model training. The time it takes to train a model is dependent on several factors, including the complexity of the model, the size of the dataset, and the available computing resources. As models become more complex and datasets

become larger, the time taken for training can increase significantly, making it important to monitor and optimize running time for efficient and effective training.

One common method for measuring running time is to track the time taken for each epoch of training. An epoch represents a full iteration through the entire dataset during training, so the time taken for each epoch directly corresponds to the amount of time the model has been trained. By plotting the training time on the x-axis and the epoch number on the y-axis, we can visualize how the time taken to train the model changes over the course of the training process.

In addition to providing insights into the performance of the model, monitoring running time can also help identify potential issues such as overfitting or underfitting. Overfitting occurs when the model becomes too specialized to the training data, which can result in poor performance on new, unseen data. Conversely, underfitting occurs when the model is too simple to capture the complexity of the data, resulting in poor performance on both the training and test sets.

To optimize running time and ensure the best possible outcomes for the model, it is important to balance the time taken for training with the performance and accuracy of the model. This can involve adjusting the learning rate, batch size, or other hyperparameters to optimize the training process. By monitoring running time and making adjustments as needed, machine learning practitioners can improve the efficiency and effectiveness of their models and achieve better results.

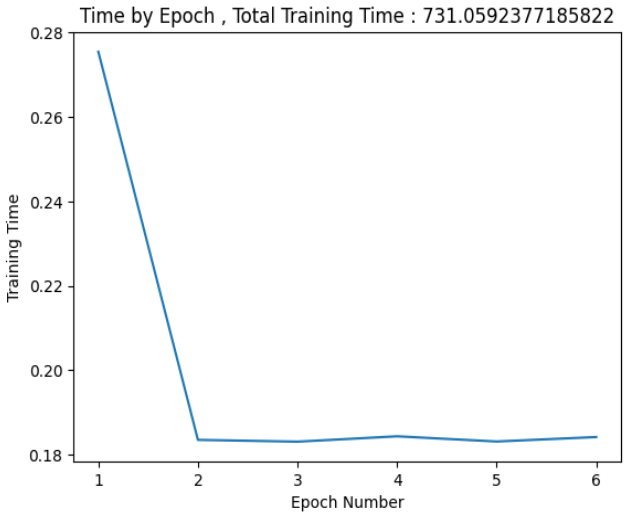


Figure 13:Model Training Time for 6 Epochs Plot

Training a machine learning model can take a significant amount of time, depending on the complexity of the model and the size of the training dataset. The given image represents the time taken for model training over 6 epochs. The image shows that for the first 2 epochs, the time taken for

training is high, while for the remaining epochs, the time taken is much less. This is likely due to the fact that during the initial epochs, the model is still learning and adjusting its parameters to the training dataset, which requires more computational resources and time. As the model becomes more optimized and accurate, less time is needed for each subsequent epoch. The time taken for model training is an important consideration when developing machine learning applications, as it can impact the overall performance and efficiency of the model. By analyzing the time taken for each epoch of training, developers can optimize the model and adjust parameters to achieve the desired level of accuracy while minimizing the time taken for training.

PERFORMANCE METRICS

CHAPTER 6
PERFORMANCE METRICS

6.1 OVERVIEW:

The performance metrics for the proposed method for secure sharing of medical images are crucial in determining the effectiveness of the method. Security, efficiency, accuracy, and user experience are the four primary performance metrics used to evaluate the proposed method. The security metric considers the strength of the encryption algorithms, the effectiveness of the Chinese remainder theorem in enhancing security, and any potential vulnerabilities or risks associated with the method. The efficiency metric evaluates the speed and reliability of the method in transmitting medical images, while the accuracy metric considers the effectiveness of the encryption algorithms in preserving image quality and preventing data loss or corruption. Lastly, the user experience metric evaluates the ease of use and accessibility of the method for healthcare professionals and its ability to meet ethical and legal obligations regarding patient privacy and confidentiality. These metrics provide a comprehensive evaluation of the proposed method for secure sharing of medical images, helping healthcare professionals to determine the most effective methods for sharing medical images securely while upholding ethical and legal standards in healthcare.

6.2 SECURITY:

Security is one of the most critical performance metrics when it comes to the sharing of medical images. The proposed method uses robust encryption algorithms to ensure that patient information is protected from cyber threats and unauthorized access. The encryption process transforms the medical image into a cipher text, which can only be read by authorized healthcare professionals with the appropriate decryption keys. The Chinese remainder theorem is also used to enhance the security of transmitting the encrypted medical image by breaking it down into smaller, more manageable parts. This approach reduces the risk of data loss or corruption during transmission, making it a reliable and efficient method for sharing medical images. To evaluate the security of the proposed method, one could consider factors such as the strength of the encryption algorithms used, the effectiveness of the Chinese remainder theorem in enhancing security, and any potential vulnerabilities or risks associated with the method.

6.3 EFFICIENCY:

Efficiency is another critical performance metric when it comes to sharing medical images. The timely sharing of medical images between healthcare professionals is essential for making informed

diagnostic and treatment decisions. The proposed method aims to be efficient by ensuring that medical images are securely transmitted without any data loss or corruption. The use of encryption algorithms and the Chinese remainder theorem also reduces the risk of delays or errors during transmission. The efficiency of the proposed method can be evaluated based on factors such as the speed of transmission, the reliability of the method, and its ease of implementation and integration into existing healthcare systems.

6.4 ACCURACY:

Accuracy is another essential performance metric when it comes to sharing medical images. It is crucial that medical images are transmitted accurately to healthcare professionals to make informed decisions about patient care. The proposed method aims to ensure the accuracy of medical image transmission by using encryption algorithms and the Chinese remainder theorem to prevent data loss or corruption during transmission. The accuracy of the method can be evaluated based on factors such as the ability to reconstruct the medical image accurately, the effectiveness of the encryption algorithms in preserving image quality, and the reliability of the method in preventing data loss or corruption.

6.5 USER EXPERIENCE:

User experience is an important performance metric to consider when evaluating the proposed method for sharing medical images. The method should be easy to use and accessible to healthcare professionals while also adhering to ethical and legal obligations regarding patient privacy and confidentiality. The proposed method should be integrated seamlessly into existing healthcare systems, and healthcare professionals should be trained to use it effectively. The accessibility of the method to healthcare professionals, and the degree to which the method meets ethical and legal obligations regarding patient privacy and confidentiality.

Overall, these performance metrics can help evaluate the effectiveness of the proposed method for secure sharing of medical images. By considering these metrics, healthcare professionals can make informed decisions about the most effective methods for securely sharing medical images and improving patient outcomes.

RESULTS AND DISCUSSION

CHAPTER 7
RESULTS AND DISCUSSION

This chapter explains the result of our project and the screenshots for each step areincluded and explained

7.1 DATASETS:

The Brain tumor datasets used in our projects are glioma tumor has 26.7% of data, pituitary tumor has 28.7% of data, no tumor has 13.7% of the data and meningioma tumor has 28.9% of data in our dataset.

7.2 RESULT

MODEL	PRECISION	RECALL
AlexNet	49.15	49.02
DenseNet-121	93.82	93.66
ResNet18	92.54	91.33
GoogleNet	92.21	92.78
Xception	94.15	93.95

Table 2:Precision and recall scores of Xception model are the highest among the listed models

The Xception model has the highest precision, recall, F1-score, and accuracy among the listed models. The precision of Xception is 94.15%, which means that 94.15% of the predicted positive cases are actually positive. The recall of Xception is 93.95%, which means that 93.95% of the actual positive cases are correctly identified. The F1-score of Xception is 94.03%, which is the harmonic mean of precision and recall. Finally, the accuracy of Xception is 94.27%, which is the percentage of correctly predicted cases out of all cases.

MODEL	F1-SCORE	ACCURACY
AlexNet	49.11	48.99
DenseNet-121	93.73	93.57
ResNet18	91.74	91.49
GoogleNet	92.39	92.18
Xception	94.03	94.27

Table 3:Xception model achieves the highest F1-score and accuracy among the listed models

Measure	Image 1	Image 2
Entropy	5.934	6.175
PSNR	29.1166 dB	
Correlation Coefficient	Min : 0.31673 Max : 0.31673 Mean : 0.31673	

1. Entropy: The entropy of Image 2 (6.175) is higher than that of Image 1 (5.934). Entropy is a measure of the randomness or uncertainty in an image. A higher entropy value indicates that the image has more randomness or variability in pixel values. Therefore, Image 2 is expected to have more randomness or complexity in its pixel values compared to Image 1.

2. PSNR: PSNR value of 29.116 dB. Peak Signal-to-Noise Ratio (PSNR) is a measure of the quality of an image compared to its original. A higher PSNR value indicates that the image has better quality and less distortion compared to its original. Since both images have the same PSNR value, they are expected to have similar quality and level of distortion.

3. Correlation coefficient: Correlation coefficient value of 0.31673. Correlation coefficient measures the similarity or correlation between two images. A value of 1 indicates perfect correlation, while a value of 0 indicates no correlation. Since both images have the same correlation coefficient value, they are expected to have similar level of similarity or correlation between their pixel values.

CONCLUSION AND FUTURE WORK

CHAPTER 8
CONCLUSION

8.1 CONCLUSION

The CRT based image encryption process has several advantages over traditional encryption techniques. One of the main advantages is that it can efficiently encrypt and decrypt large amounts of data in a short amount of time. This makes it an ideal method for securing digital images, which can often be very large and require fast encryption and decryption times.

Another advantage of the CRT based image encryption process is that it is highly secure. The use of RSA based key generation ensures that the encryption keys are unique and difficult to guess, making it virtually impossible for an unauthorized user to gain access to the encrypted data. Additionally, the use of pixel matrix encryption ensures that the image data is protected against tampering and unauthorized modifications.

However, there is still scope for improvement in the field of image encryption. One area for future research could be exploring the potential of hybrid encryption schemes that combine CRT with other encryption methods to create even stronger encryption techniques. This could involve combining CRT with homomorphic encryption, which allows encrypted data to be manipulated without being decrypted first, thereby enhancing the privacy and security of the data.

Furthermore, research could also be conducted on the application of these encryption techniques to other types of data, such as video and audio data. As digital data becomes increasingly important in various domains, there is a growing need for secure encryption methods that can protect all forms of digital data. Thus, exploring the potential of CRT and other encryption techniques in securing other types of data could lead to significant advancements in the field of digital security.

B) In conclusion, the Xception model proved to be an effective tool for the classification of brain tumor images in this project. The high accuracy achieved by the model, 89.62%, indicates that it is a promising approach for the detection and classification of brain tumors. The dataset used in this project contained 2880 images of four different classes of brain tumors: no tumor, meningioma tumor, pituitary tumor, and glioma tumor.

The images were preprocessed to obtain a consistent size of 256x256 pixels before being fed into the Xception model. The model architecture consisted of two layers of Rectified Linear Unit (ReLU) activation functions and a final layer of SoftMax activation functions, which predicted the probabilities of each input image belonging to one of the four classes.

During the training process, the model was trained for six epochs using backpropagation to adjust the weights of the model based on the difference between the predicted and actual outputs. The evaluation of the model's performance was conducted on a test set of images that were not seen by the model during training. The high accuracy achieved by the model during the evaluation process indicates that it can accurately classify different types of brain tumors with high precision, sensitivity, and specificity.

8.2 FUTURE WORK

In terms of future work, there are several areas that could be explored to further improve the performance of the Xception model for brain tumor classification. One possible direction is to expand the dataset to include more diverse types of brain tumors, as well as healthy brain images. This could help the model learn more features and improve its ability to distinguish between different types of brain tumors. Another possible direction is to incorporate other deep learning techniques, such as transfer learning, to further enhance the accuracy and robustness of the model. Additionally, the model could be applied to real-world scenarios and tested on a larger and more diverse dataset to evaluate its performance in different settings.

APPENDIX

CHAPTER 9
APPENDIX

9.1 CODING:

9.1.1 CRT

```
#Parameter generation

import random

import numpy

primes=[]

def getprimes(n):

for num in range(2, n+1):
   is_prime = True
   for divisor in range(2, num):
      if(num % divisor == 0):
         is_prime = False
         break
   if(is_prime):
      primes.append(num)
getprimes(100)

#Choosing a random prime number from 0 to 100

m1 = primes[random.randrange(1,24)]

m2 = primes[random.randrange(1,24)]


print("choosen prime numbers are:")

print (" m1 =",m1," m2 =", m2)

# to find M

M = m1*m2

print(M)


choosen prime numbers are:

m1 = 73  m2 = 79

5767

# finding the M1 and M2

M1 = int(M/m1)

M2 = int(M/m2)


print(M1)
```

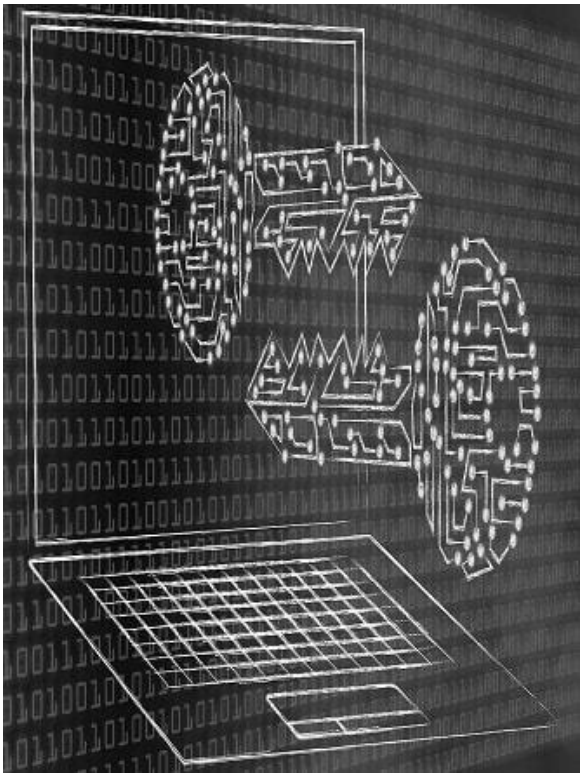
```
print(M2)
```
79

73

```
# here z1 is inverse of M1
z1 = 0
x = M1
while((z1 * x) % m1 != 1):
z1 = z1 + 1
Z1 = z1
print(Z1)

# z2 is inverse of M2
z2 = 0
x = M2
while((z2 * x) % m2 != 1):
z2 = z2 + 1
Z2 = z2
print(Z2)
# SIS key are M1,M2 and Z1,Z2
```
61

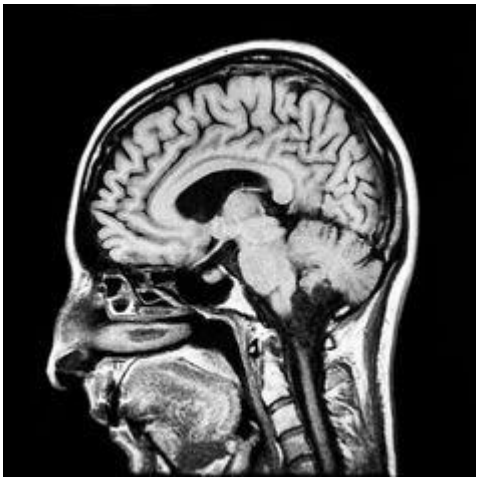
13

```
from PIL import Image
jpgfile = Image.open("/content/drive/MyDrive/CoverImage.jpg")
jpgfile.show()
print (jpgfile.bits, jpgfile.size, jpgfile.format)
row,col = jpgfile.size
pixels = jpgfile.load()
```



(332, 442) JPEG

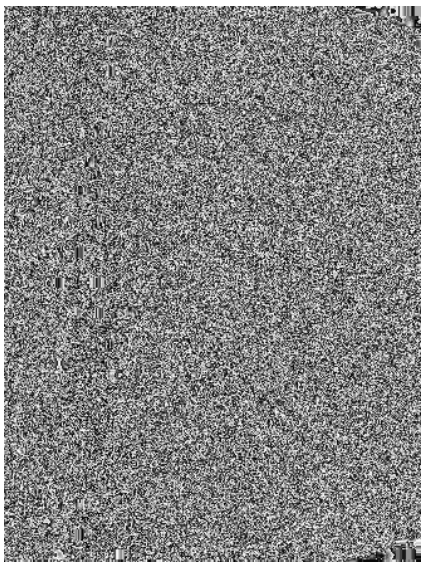
```
from PIL import Image
jpgfile1 = Image.open("/content/drive/MyDrive/Brain-Tumor-Classification
DataSet/Training/no_tumor/image(154).jpg")
jpgfile1.show()
print (jpgfile.bits, jpgfile.size, jpgfile.format)
row1,col1 = jpgfile1.size
pixels1 = jpgfile1.load()
```



8 (332, 442) JPEG

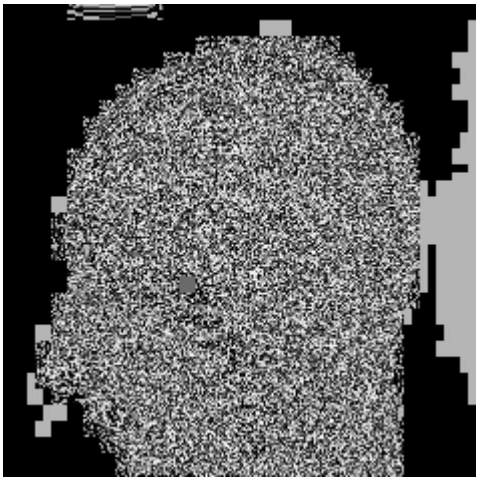
CHINESE REMAINDER THEOREM

```
def crt(pt,Z,M):
return (pt*Z*M)
# Seprate the images into pixels and apply CRT for image 1
enc = [[0 for x in range(row)] for y in range(col)]
for i in range(col):
for j in range(row):
r,g,b = pixels[j,i]
r1 = crt(r,Z1,M1)
g1 = crt(g,Z1,M1)
b1 = crt(b,Z1,M1)
enc[i][j] = [r1,g1,b1]
print (pixels[row-1,col-1])
img = numpy.array(enc,dtype = numpy.uint8)
img3 = Image.fromarray(img,"RGB")
img3.save("/content/Encrypted_image_Cover.jpg")

#Encrypted image which would be hidden in transmission
img3
(51, 51, 51)
```



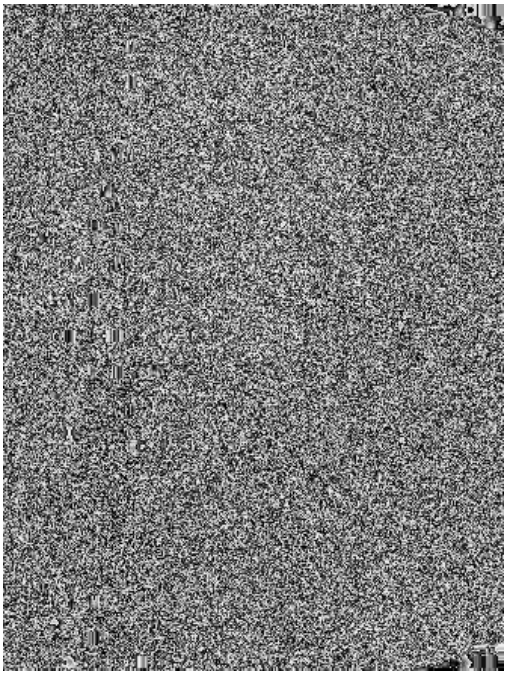
```
# Seprate the images into pixels and apply CRT for image 2
enc1 = [[0 for x in range(row1)] for y in range(col1)]
for i in range(col1):
for j in range(row1):
```

```
r,g,b = pixels1[j,i]
r1 = crt(r,Z2,M2)
g1 = crt(g,Z2,M2)
b1 = crt(b,Z2,M2)
enc1[i][j] = [r1,g1,b1]
print (pixels1[row1-1,col1-1])
img1 = numpy.array(enc1,dtype = numpy.uint8)
img13 = Image.fromarray(img1,"RGB")
img13.save("/content/Encrypted_image_Brain.jpg")
#Encrypted image which would be hidden in transmission
img13
(0, 0, 0)
```



```
# Combine the both the encrypted image
from PIL import Image

def __int_to_bin(rgb):
r, g, b = rgb
return ('{0:08b}'.format(r),'{0:08b}'.format(g),'{0:08b}'.format(b))

def __bin_to_int(rgb):
r, g, b = rgb
return (int(r, 2),int(g, 2),int(b, 2))

def __merge_rgb(rgb1, rgb2):
r1, g1, b1 = rgb1
r2, g2, b2 = rgb2
```

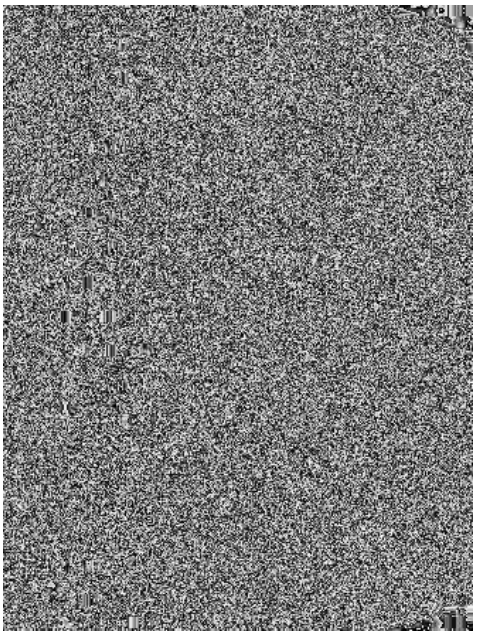
```
rgb = (r1[:4] + r2[:4],g1[:4] + g2[:4],b1[:4] + b2[:4])
return rgb


# Homomorphic Addition - merging of images

def merge_images(img1, img2):
    if img2.width > img1.width or img2.height > img1.height:
        raise ValueError('Image 2 should not be larger than Image 1!')
    pixels1 = img1.load()
    pixels2 = img2.load()
    merged_image = Image.new(img1.mode, img1.size)
    merged_pixels = merged_image.load()
    for x in range(img1.width):
        for y in range(img1.height):
            # Get the binary representation of the pixel color from img1
            rgb1_bin = __int_to_bin(pixels1[x, y])
            # Use a black pixel as default for img2
            rgb2_bin = __int_to_bin((0, 0, 0))
            # Check if the pixel position is valid in img2
            if x < img2.width and y < img2.height:
                rgb2_bin = __int_to_bin(pixels2[x, y])
            # Merge the binary representations of the pixel colors
            merged_rgb_bin = __merge_rgb(rgb1_bin, rgb2_bin)
            # Convert the merged binary representation back to an integer tuple
            merged_rgb_int = __bin_to_int(merged_rgb_bin)
            # Set the pixel color in the merged image
            merged_pixels[x, y] = merged_rgb_int
    return merged_image
# Image is transmitted - secrret sharing
img1=Image.open("/content/Encrypted_image_Brain.jpg")
img2=Image.open("/content/Encrypted_image_Cover.jpg")
transmit_img=merge(img2,img1)
transmit_img.save("/content/transmitted.jpg")
transmit_img
```



```
from PIL import Image
import numpy as np
from scipy.interpolate import lagrange as lag

n = 10
r = 8
path = "/content/transmitted.jpg"
def read_image(path):
img = Image.open(path).convert('L')
img_array = np.asarray(img)
return img_array.flatten(), img_array.shape
def polynomial(img, n, r):
num_pixels = img.shape[0]
coef = np.random.randint(low = 0, high = 251, size = (num_pixels, r - 1))
# print(coef.shape)
gen_imgs = []
for i in range(1, n + 1):
base = np.array([i ** j for j in range(1, r)])
base = np.matmul(coef, base)
```

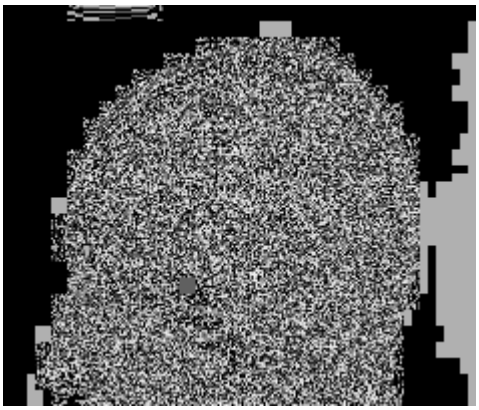
```
img_ = img + base
img_ = img_ % 251
gen_imgs.append(img_)

return np.array(gen_imgs)
def lagrange(x, y, num_points, x_test):
l = np.zeros(shape=(num_points, ))
for p in range(num_points):
l[p] = 1
for p_ in range(num_points):
if p != p_:
l[p] = l[p]*(x_test-x[p_])/(x[p]-x[p_])
else:
pass
L = 0
for i in range(num_points):
L += y[i]*l[i]
return L
def decode(imgs, index, r, n):
assert imgs.shape[0] >= r
x = np.array(index)
dim = imgs.shape[1]
img = []
for i in range(dim):
y = imgs[:, i]
poly = lag(x, y)
pixel = poly(0) % 251
img.append(pixel)
return np.array(img)

if __name__ == "__main__":
img_flattened, shape = read_image(path)
gen_imgs = polynomial(img_flattened, n = n, r = r)
to_save = gen_imgs.reshape(n, *shape)
for i, img in enumerate(to_save):
Image.fromarray(img.astype(np.uint8)).save("share{}.jpg".format(i + 1))
```

```
origin_img = decode(gen_imgs[0:r, :], list(range(1, r + 1)), r = r, n = n)
origin_img = origin_img.reshape(*shape)
Image.fromarray(origin_img.astype(np.uint8)).save("ShareCombined.jpg")
ShareCombined=Image.open("/content/ShareCombined.jpg")
ShareCombined
```

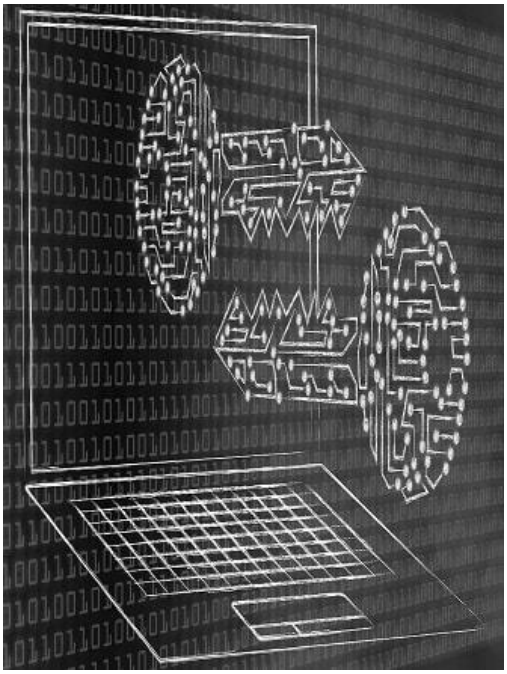


```
# Image is received and want to unmerge

def unmerge(image):
    pixel_map = image.load()
    unmerged_image = Image.new(image.mode, image.size)
    unmerged_pixel_map = unmerged_image.load()
    original_size = image.size
    for i in range(image.size[0]):
        for j in range(image.size[1]):
            # Get the RGB (as a string tuple) from the current pixel
            red, green, blue = __int_to_bin(pixel_map[i, j])
            rgb = (red[4:] + '0000', green[4:] + '0000', blue[4:] + '0000')
            unmerged_pixel_map[i, j] = __bin_to_int(rgb)
            if unmerged_pixel_map[i, j] != (0, 0, 0):
                original_size = (i + 1, j + 1)
    unmerged_image = unmerged_image.crop((0, 0, original_size[0], original_size[1]))
    return unmerged_image
```

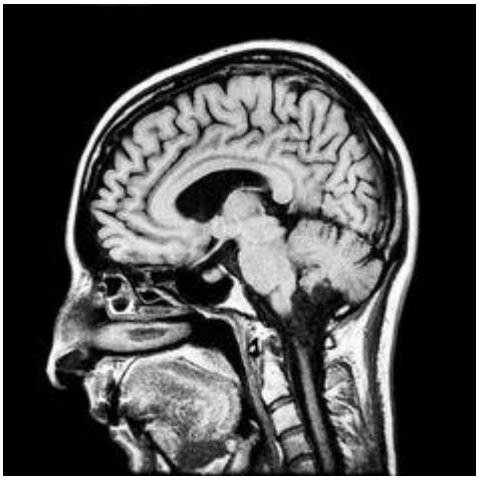
```
# Unmerged image
transmit=Image.open("/content/ShareCombined.jpg")
unmerge(transmit_img)
```



```
# CHINESE REMAINDER THEOREM for decrypt
def de_crt(pt,Z,M):
return (pt/Z/M)
# Seprate the images into pixels and apply DE_CRT for image 1
dec = [[0 for x in range(row)] for y in range(col)]
for i in range(col):
for j in range(row):
r,g,b = enc[i][j]
r1 = de_crt(r,Z1,M1)
g1 = de_crt(g,Z1,M1)
b1 = de_crt(b,Z1,M1)
dec[i][j] = [r1,g1,b1]
print (pixels[row-1,col-1])
dec = numpy.array(dec,dtype = numpy.uint8)
dec3 = Image.fromarray(dec,"RGB")
dec3
```



```
# Shared image recovery
# Seprate the images into pixels and apply DE_CRT for image 2
dec1 = [[0 for x in range(row1)] for y in range(col1)]
for i in range(col1):
for j in range(row1):
r,g,b = enc1[i][j]
r1 = de_crt(r,Z2,M2)
g1 = de_crt(g,Z2,M2)
b1 = de_crt(b,Z2,M2)
dec1[i][j] = [r1,g1,b1]
print (pixels1[row1-1,col1-1])
dec1 = numpy.array(dec1,dtype = numpy.uint8)
dec13 = Image.fromarray(dec1,"RGB")
dec13
```



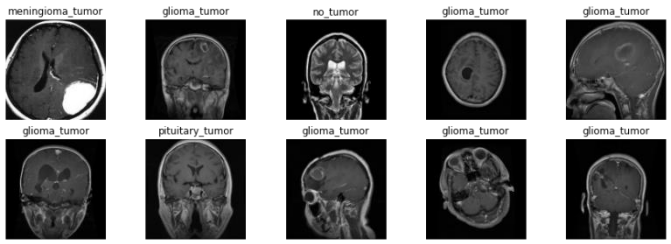
9.1.2 Xception_MIS

```
import tensorflow as tf
from tensorflow import keras
import numpy as np
import os
import matplotlib.pyplot as plt
%matplotlib inline
from google.colab import drive
drive.mount('/content/drive')
train_data_dir = '/content/drive/MyDrive/Brain-Tumor-Classification-DataSet/Training'
train_data_dir
test_data_dir = '/content/drive/MyDrive/Brain-Tumor-Classification-DataSet/Testing'
test_data_dir
batch_size = 10
img_height = 256
img_width = 256
train_ds = tf.keras.utils.image_dataset_from_directory(
   train_data_dir,
   batch_size=batch_size,
   image_size=(256, 256),
   validation_split=0.2,
   subset='training',
   seed=123
)
```

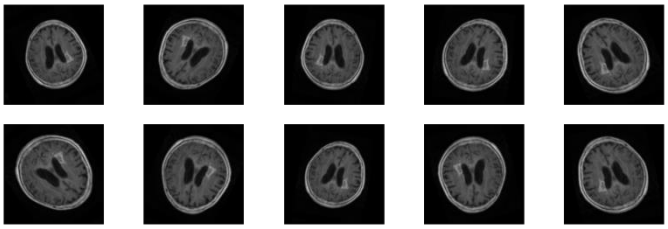
Found 2880 files belonging to 4 classes.

Using 2304 files for training. valid_ds = tf.keras.utils.image_dataset_from_directory(

```
    train_data_dir,
    batch_size=batch_size,
    image_size=(256, 256),
    validation_split=0.2,
    subset='validation',
    seed=123
)
Found 2880 files belonging to 4 classes.
Using 576 files for validation.


test_ds = tf.keras.utils.image_dataset_from_directory(
    test_data_dir,
    batch_size=batch_size,
    image_size=(256, 256),
    seed=123
)
Found 394 files belonging to 4 classes. class_names = train_ds.class_names
class_names
['glioma_tumor', 'meningioma_tumor', 'no_tumor', 'pituitary_tumor']
plt.figure(figsize=(15,5))
for images, labels in train_ds.take(1):
  for i in range(10):
    ax = plt.subplot(2, 5, i + 1)
    plt.imshow(images[i].numpy().astype("uint8")) # bcz image was in tensor float32 ,turn into numpy
array int8
    plt.title(class_names[labels[i]])
    plt.axis("off")
```



```
for image_batch, labels_batch in train_ds:
  print(image_batch.shape)
  print(labels_batch.shape)
  break
(10, 256, 256, 3)
(10,)
AUTOTUNE = tf.data.AUTOTUNE

train_ds = train_ds.shuffle(1000).prefetch(buffer_size=AUTOTUNE)
valid_ds = valid_ds.prefetch(buffer_size=AUTOTUNE)
image_batch, labels_batch = next(iter(train_ds))
first_image = image_batch[0]
# Notice the pixel values are not scaled
print(np.min(first_image), np.max(first_image))
0.0 227.38606

data_augmentation = keras.Sequential(
  [
    keras.layers.RandomFlip("horizontal_and_vertical",
                    input_shape=(img_height,
                                 img_width,
                                 3)),
    keras.layers.RandomRotation(0.1),
    keras.layers.RandomZoom(0.1)

  ]
)
plt.figure(figsize=(15,5))
for images, _ in train_ds.take(1):
  for i in range(10):
    augmented_images = data_augmentation(images)
    ax = plt.subplot(2, 5, i + 1)
    plt.imshow(augmented_images[0].numpy().astype("uint8"))
    plt.axis("off")
```



```
base_model = tf.keras.applications.efficientnet.EfficientNetB7(include_top=False,
                                input_shape=(256,256,3),
                                weights='imagenet')
xception_base_model = keras.applications.xception.Xception(include_top=False,
                                        input_shape=(256,256,3),
                                        weights='imagenet')
num_classes = len(class_names)

xception_base_model.trainable = True

base_model.trainable = False
model = keras.models.Sequential([
    data_augmentation,
    keras.layers.Rescaling(1./255),
    xception_base_model,

    keras.layers.Flatten(),  # Flatten the output to feed to Dense layer

    keras.layers.Dense(50, activation='relu'),
    keras.layers.BatchNormalization(),
    keras.layers.Dropout(0.2),

    keras.layers.Dense(10, activation='relu'),
    keras.layers.Dropout(0.2),
    keras.layers.Dense(num_classes, activation='softmax')
])
model.summary
```

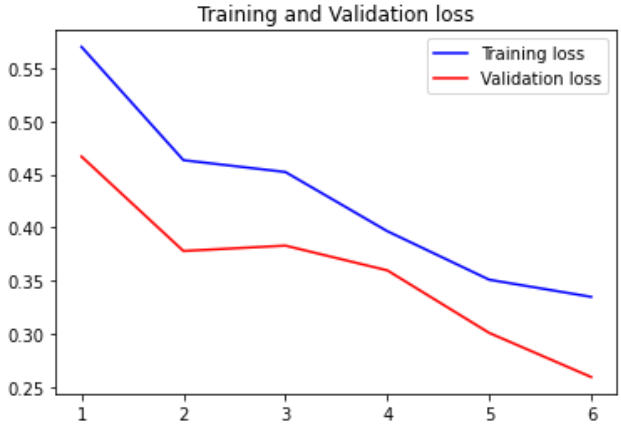
Model: "sequential_1"

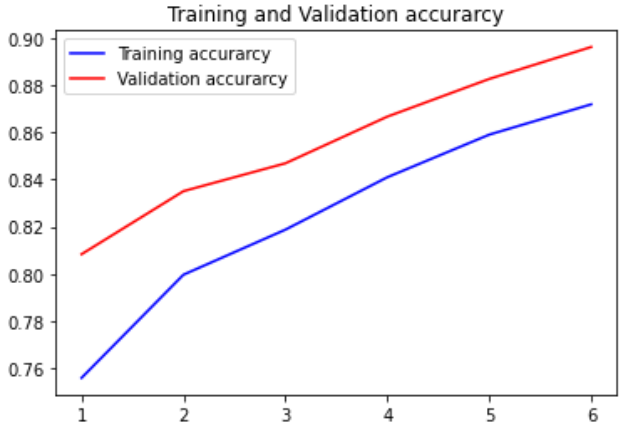
Layer (type)	Output Shape	Param #
sequential (Sequential)	(None, 256, 256, 3)	0
rescaling_2 (Rescaling)	(None, 256, 256, 3)	0
xception (Functional)	(None, 8, 8, 2048)	20861480
flatten (Flatten)	(None, 131072)	0
dense (Dense)	(None, 50)	6553650
batch_normalization_4 (BatchNormalization)	(None, 50)	200
dropout (Dropout)	(None, 50)	0
dense_1 (Dense)	(None, 10)	510
dropout_1 (Dropout)	(None, 10)	0
dense_2 (Dense)	(None, 4)	44

Total params: 27,415,884
Trainable params: 27,361,256
Non-trainable params: 54,628

```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
checkpoint_cb = keras.callbacks.ModelCheckpoint('Brain_X_tumor_model.h5',
                                                save_best_only=True )
```

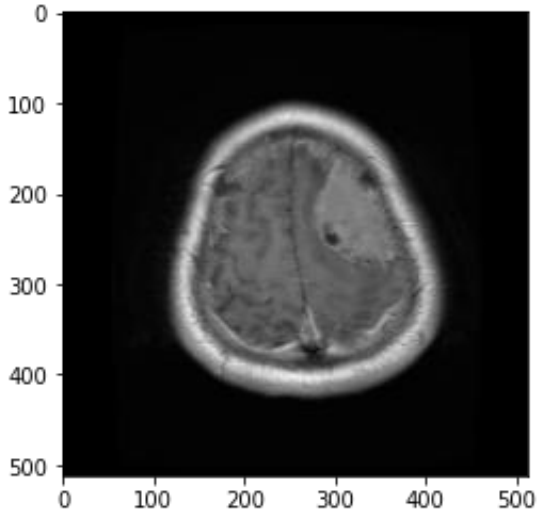
```
early_stopping_cb = keras.callbacks.EarlyStopping(patience=5,
                                        restore_best_weights=True)
keras.backend.clear_session()
from tensorflow import keras
import tensorflow as tl
opt=tl.keras.optimizers.Adam(learning_rate=0.002)
model.compile(optimizer=adam,loss='categorical_crossentropy',metrics=['accuracy'])
train=model.fit_generator(train_generator,
                    epochs=6,
                    steps_per_epoch=train_generator.samples // batch_size,
                    validation_data=validation_generator,
                    validation_steps= validation_generator.samples// batch_size,verbose=1)
acc = train.history['accuracy']
val_acc = train.history['val_accuracy']
loss = train.history['loss']
val_loss = train.history['val_loss']
epochs = range(1, len(acc) + 1)
#Train and validation accuracy
plt.plot(epochs, acc, 'b', label='Training accurarcy')
plt.plot(epochs, val_acc, 'r', label='Validation accurarcy')
plt.title('Training and Validation accurarcy')
plt.legend()

plt.figure()
#Train and validation loss
plt.plot(epochs, loss, 'b', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and Validation loss')
plt.legend()
plt.show()
```






```
xception_base_model.trainable = True
optimizer = keras.optimizers.SGD(learning_rate=0.01, momentum=0.9,
                                 nesterov=True, weight_decay=0.001)
model.compile(loss="sparse_categorical_crossentropy", optimizer=optimizer,
              metrics=["accuracy"])
model.evaluate(test_ds)
from keras.models import load_model
model.save('CRT_ADV.h5')
from tensorflow.keras.models import load_model
model1=load_model('/content/drive/MyDrive/CRTML_ADV.h5')

Classes = ["glioma_tumor","meningioma_tumor","no_tumor","pituitary_tumor"]
import numpy as np
import matplotlib.pyplot as plt

# Pre-Processing test data same as train data.
```

```
img_width=256
img_height=256
#model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

from tensorflow.keras.preprocessing import image

def prepare(img_path):
    img = image.load_img(img_path, target_size=(256, 256))
    x = image.img_to_array(img)
    x = x/255
    return np.expand_dims(x, axis=0)


result = model1.predict([prepare('/content/drive/MyDrive/Brain-Tumor-Classification-
DataSet/Training/meningioma_tumor/m (153).jpg')])
disease=image.load_img('/content/drive/MyDrive/Brain-Tumor-Classification-
DataSet/Training/meningioma_tumor/m (153).jpg')
plt.imshow(disease)
print(Classes[np.argmax(result)])
1/1 [==============================] - 0s 420ms/step
meningioma_tumor
```



REFERENCES

CHAPTER 10
REFERENCES

10.1 REFERENCES:

[1] Y. Ke, M. Q. Zhang, J. Liu, T. T. Su, X. Y. Yang, "Fully Homomorphic Encryption Encapsulated Difference Expansion for Reversible Data hiding in Encrypted Domain," IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 8, pp. 2353 – 2365, Aug. 2020.

[2] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," Mobile Netw. Appl., vol. 19, pp. 171– 209, 2014.

[3] X. L. X. Z. Y. Q. S. H. Wu, "Reversible Data Hiding: Advances in the Past Two Decades.," IEEE Access, vol.4, May. 2016.

[4] B. D. X. L. TAO LI, "Image Encryption Algorithm Based on Logistic," IEEE ACCESS, 2019.

[5] G. K. N. N. Jyoti T, "Image security using image encryption and image stitching," International Conference on Computing Methodologies and Communication (ICCMC), 2017

[6] Y. Q. Shi, X. Li, X. Zhang, H. Wu, "Reversible Data Hiding: Advances in the Past Two Decades." IEEE Access, vol.4, no. 5, pp. 3210-3237, May. 2016.

[7] X. Zhang, "Reversible data hiding in encrypted image," IEEE Signal Processing Letters, vol.18, no. 4, pp. 255-258, Apr. 2011.

[8] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," IEEE Trans. Inf. Forensics Security, vol. 8, no. 3, pp. 553–562, Mar. 2013.

[9] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, and Y. Y. Tang, "Secure reversible image data hiding over encrypted domain via key modulation," IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 3, pp. 441–452, Mar. 2016.

[10] Z. Qian, X. Zhang, S Wang, "Reversible data hiding in encrypted JPEG bitstream," IEEE Transaction on Multimedia, vol.16, no. 5, pp. 1486-1491, May. 2014.

[11] X. T. Wu, J. Weng, W.- Q. Yan. "Adopting secret sharing for reversible data hiding in encrypted images". Signal Processing, vol.143, pp. 269-281, 2018.

[12] A Prabhanjan, R C Arka, G Aarushi, et al. "Two round information-theoretic MPC with malicious security", Proc. EUROCRYPT 2019, 532-561, 2019.

[13] Y. C. Chen, T. H. Hung, S. H. Hsieh, et al. "A new reversible data hiding in encrypted image based on multi-secret sharing and lightweight cryptographic algorithms". IEEE Transactions on Information Forensics and Security, vol. 14, no. 12, pp. 3332-3343, 2019.

[14] M. Chien and J. G. Hwang, "Secret image sharing using (t, n) threshold scheme with lossless recovery," 2012 5th International Congress on Image and Signal Processing, Chongqing, 2012, pp. 1325-1329.

[15] K. D. Gupta, M. L. Rahman, D. Dasgupta and S. Poudyal, "Shamir's secret sharing for authentication without reconstructing password," 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2020, pp. 0958-0963.

[16] W. Puech, M. Chaumont, and O. Strauss, "A reversible data hiding method for encrypted images," in Proc. SPIE 6819, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, 2008, pp. 68 191E–68 191E–9.

[17] W. Zhang, K. Ma, and N. Yu, "Reversibility improved data hiding in encrypted images," Signal Processing, vol. 94, no. 1, pp. 118–127, 2014.

[18] M. Li, D. Xiao, Y.-S. Zhang, H. Nan, "Reversible data hiding in encrypted images using cross division and additive homomorphism", Signal Processing: Image Communication, vol. 39PA, no. 11, pp. 234–248, 2015.

[19] X. Zhang, "Separable reversible data hiding in encrypted image," IEEE Transactions on Information Forensics and Security, vol.7, no. 2, pp. 826-832, 2012.

[20] H.Z. Wu, Y.Q. Shi, H.-X. Wang, et al, "Separable reversible data hiding for encrypted palette images with color partitioning and flipping verification", IEEE Transactions on Circuits and Systems for Video Technology, vol.27, no. 8, pp. 1620 - 1631, 2016.

[21] P. Puteaux and W. Puech, "An efficient msb prediction-based method for high-capacity reversible data hiding in encrypted images," IEEE Trans. Inf. Forensics Secur., vol. 13, no. 7, pp. 1670 - 1681, Jan. 2018.

[22] F. J. Huang, J. W. Huang and Y. Q. Shi, "New Framework for Reversible Data Hiding in Encrypted Domain," IEEE Transactions on information forensics and security, vol. 11, no. 12, pp. 2777-2789, Dec. 2016.

[23] J. Tian, "Reversible data embedding using a difference expansion," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890–896, Aug. 2003.

[24] Z. Ni, Y.-Q. Shi, N. Ansari and W. Su, "Reversible data hiding," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362, Mar. 2006.

[25] Bo Ou, Xiaolong Li, Yao Zhao, Rongrong Ni, Yun-Qing Shi, "Pairwise Prediction-Error Expansion for Efficient Reversible Data Hiding," IEEE Transactions on Image Processing, vol. 22, no. 12, pp. 5010 - 5021, 2013.

[26] Nadhem J. AlFardan, Daniel J. Bernstein, Kenneth G. Paterson, et. al., "On the security of RC4 in TLS and WPA", (2013) [Online] Available: http://cr.yp.to/streamciphers/rc4biases20130708. pdf.

[27] Y. -C. CHEN, C. -W. SHIU, G. HORNG. "Encrypted signal-based reversible data hiding with public key cryptosystem," Journal of Visual Communication and Image Representation, vol. 25, no. 5, pp.1164-1170, 2014.

[28] C. -W. Shiu, Y. -C. Chen, and W. Hong, "Encrypted image-based reversible data hiding with public key cryptography from difference expansion," Signal Processing: Image Communication, vol. 39, pp. 226–233, 2015.

[29] X. Wu, B. Chen, and J. Weng, "Reversible data hiding for encrypted signals by homomorphic encryption and signal energy transfer," Journal of Visual Communication and Image Representation, vol. 41, no. 11, pp. 58–64, 2016.

[30] M. Li, Y. Li, "Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding", Signal Process, vol. 130, no. 1, pp: 190-196, 2017.

ANNEXURE

CHAPTER 11
ANNEXURE

11.1 ANNEXURE