AUDISANKARA COLLEGE OF ENGINEERING & TECHNOLOGY
UGC-AUTONOMOUS

(Accredited by NAAC A+ & NBA)

GUDUR-524126, TIRUPATI (DT), AP, INDIA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DATA HIDING IN ENCRYPTED IMAGES ON PIXEL PREDICTION

A Project Work Report submitted to the Jawaharlal Nehru Technological University in a partial fulfillment for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

M. LAKSHMI PRASANNA KUMAR	19G21A0585
P. PAVAN KUMAR	19G21A05C2
P. MAHITHA	19G21A05B7
P. SUNEEL	19G21A05B9

Under the esteemed supervision of

Mr. N. SUBRAMANYAM,
Assistant Professor,
Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AUDISANKARA COLLEGE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

(Accredited by NAAC A+ and NBA)

Approved by AICTE, Affiliated to JNTUA, Ananthapuram,
Gudur-524101, Tirupati (DT), Andhra Pradesh.

(2019-2023)

AUDISANKARA COLLEGE OF ENGINEERING & TECHNOLOGY

UGC-AUTONOMOUS

(Accredited by NAAC A+ & NBA)

Approved by AICTE, Affiliated to JNTUA Anantapuram,

NH-5 Bypass, Gudur-524101, Tirupati (DT), Andhra Pradesh

2019-2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the major project report entitled **"DATA HIDING IN ENCRYPTED IMAGES ON PIXEL PREDICTION"** is the Bonafide Work done by **MALLEDI LAKSHMI PRASANNA KUMAR, 19G21A0585, PATUKOTA PAVAN KUMAR, 19G21A05C2, PALAPARTHI MAHITHA, 19G21A05B7, PAMURU SUNEEL, 19G21A05B9** in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING**, from Jawaharlal Nehru Technological University Anantapur, Anantapuram, during the year 2019-2023.

SUPERVISOR
Mr. N. SUBRAMANYAM,
Assistant Professor,
Department of CSE,
ASCET, GUDUR

HEAD OF THE DEPARTMENT
Dr. M. RAJAIAH,
HOD & Dean of Academics,
Department of CSE,
ASCET, GUDUR

Submitted for the viva-voce examination held on:

Internal Examiner

External Examiner

AUDISANKARA COLLEGE OF ENGINEERING & TECHNOLOGY

UGC-AUTONOMOUS

(Accredited by NAAC A+ & NBA)

Approved by AICTE, Affiliated to JNTUA Anantapuram,

NH-5 Bypass, Gudur-524101, Tirupati (DT), Andhra Pradesh.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DECLARATION

We, **MALLEDI LAKSHMI PRASANNA KUMAR-19G21A0585, PATUKOTA PAVAN KUMAR-19G21A05C2, PALAPARTHI MAHITHA-19G21A05B7, PAMURU SUNEEL-19G21A05B9** hereby declare that the major project report entitled **"DATA HIDING IN ENCRYPTED IMAGES ON PIXEL PREDICTION"** done under the esteemed guidance of **Mr.N.SUBRAMANYAM**, Department of Computer Science and Engineering and is submitted in partial fulfillment of the requirements for the award of the bachelor's degree in **Computer Science and Engineering**. We have not copied from any other students' work or from any other sources except where due reference or acknowledgment is made explicitly, nor has any part been authored by another person. We, as a candidate, declare that in case of any violation of intellectual property right or copyright will be fully responsible for the same. Our supervisor should not be held responsible for full or partial violation of copyright or intellectual property rights.

Date:

Place:

PROJECT ASSOCIATES

MALLEDI LAKSHMI PRASANNA KUMAR
(19G21A0585)
PATUKOTA PAVAN KUMAR
(19G21A05C2)
PALAPARTHI MAHITHA
(19G21A05B7)
PAMURU SUNEEL
(19G21A05B9)

ACKNOWLEDGEMENT

We would like to express our heartful gratitude our honorable chairman of AUDISANKARA GROUP OF INSTITUTIONS, **Dr. VANKI PENCHALAIAH, M.A, M.L, Ph.D,** who provided all facilities and necessary encouragement during study.

We would like to thank AUDISANKARA COLLEGE OF ENGINEERING & TECHNOLOGY for providing the extraordinary support in the completion of the project by utilizing the laboratories, library and Software required for our project.

We extend my gratitude and sincere thanks to our beloved Director **Dr. A. MOHAN BABU,** and principal, **Prof. K.DHANUMJAYA.** for motivating and providing necessary infrastructure and permitting us to complete the project.

We are grateful to **Dr.M.RAJAIAH**, Head of the Department, and our supervisor **Mr**.**N.SUBRAMANYAM,** Assistant Professor for their valuable input, able guidance, encouragement, whole-hearted cooperation, and constructive criticism throughout the duration of our project.

We express my sincere thanks to all the **teaching and non-teaching staff** that guided directly or indirectly helped me to complete the project work successfully.

Last, but not least, we would like to thank our **team members, my friends, and my parents** for supporting me in all aspects of the completion of this project.

PROJECT ASSOCIATES

MALLEDI LAKSHMI PRASANNA KUMAR
(19G21A0585)
PATUKOTA PAVAN KUMAR
(19G21A05C2)
PALAPARTHI MAHITHA
(19G21A05B7)
PAMURU SUNEEL
(19G21A0B9)

TABLE OF CONTENT

DATA HIDING IN ENCRYPTED IMAGES ON PIXEL PREDICTION

ABSTRACT

The application potential and utility of RDHEI are ideal. Regarding disguising limit, security, and detachability, the present RDHEI calculations have an open door for improvement. The interactive media information insurance should be possible with encryption or information concealing calculations. High limit double information concealing in encoded pictures (DHEI) is a successful procedure to implant information in scrambled space.

A unique picture is encoded utilizing the mystery key, still conceivable to install extra information without knowing the first happy of the picture or the mystery key. This mystery message can be recuperated and the underlying picture can be removed in the disentangling stage. As of late, reversible information concealing techniques have been proposed with high limit, however these proposed strategies do not permit a lot of implanting limits.

They propose a high limit double information concealing technique in view of MSB (most huge piece) forecast. They recommend to conceal no-account per pixel by preprocessing the picture to stay away from forecast blunders and, in this way, working on the nature of the remade picture. Security and distinguishableness are given simultaneously. Additionally, there are no wrong pieces made during the information extraction stride, and the straightforwardly unscrambled picture has precisely the same visual quality as the cover picture.

Key words: RDHEI, Most Significant Bit, EPE-HCRDH.

LIST OF ABBREVIATIONS

S.NO	ABBREVATION	DEFINITION
1	DHEI	Data Hiding in Encrypted Images
2	MSB	Most Significant Bit
3	LSB	Least Significant Bit
4	DES	Data Encryption Standard
5	RDH	Reversible Data Hiding

LIST OF FIGURES

CHAPTER 1
INRODUCTION

1.1 Introduction to Project:

The fundamental goal of the Implanted Expectation Blunders for High Limit Reversible Information Stowing away (EPE-HCRDH) approach is exactly to reproduce the first picture. For this situation, the payload might diminish a little since blunder position data being put away. To feature the forecast mistakes, we change the data to be embedded by the blunder area twofold guide, made during the of the expectation mistake location process.

The first picture is then encoded and the blunder area data is embedded in the scrambled picture promptly a short time later. They can conceal just a single piece of the mystery message inside the accessible pixels during the information concealing step.

As of late, reversible information stowing away (RDH) has acquired expanding consideration. By applying this strategy, extra information can be implanted into an interactive media cover, while information extraction and unique cover recuperation can be both acknowledged without misfortune. These days, with the advancement of PC innovation, computerized picture turns into the fundamental sort of different transferred information in individuals' day to day existence.

On account of its high overt repetitiveness, specialists frequently apply it as the information cover to propose the RDH techniques. The greater part of them in spatial pictures utilize three principal advances, i.e., lossless pressure, distinction extension, and histogram moving.

These advancements take full advantage of the excess data in unique pictures to reversibly insert extra information, yet they must be executed in the plaintext space of the picture. When the picture is encoded, heaps of overt repetitiveness data will be lost.

In this manner, the RDH techniques considering the above advances cannot be straightforwardly applied to encoded pictures.

1.2 Process Diagram:

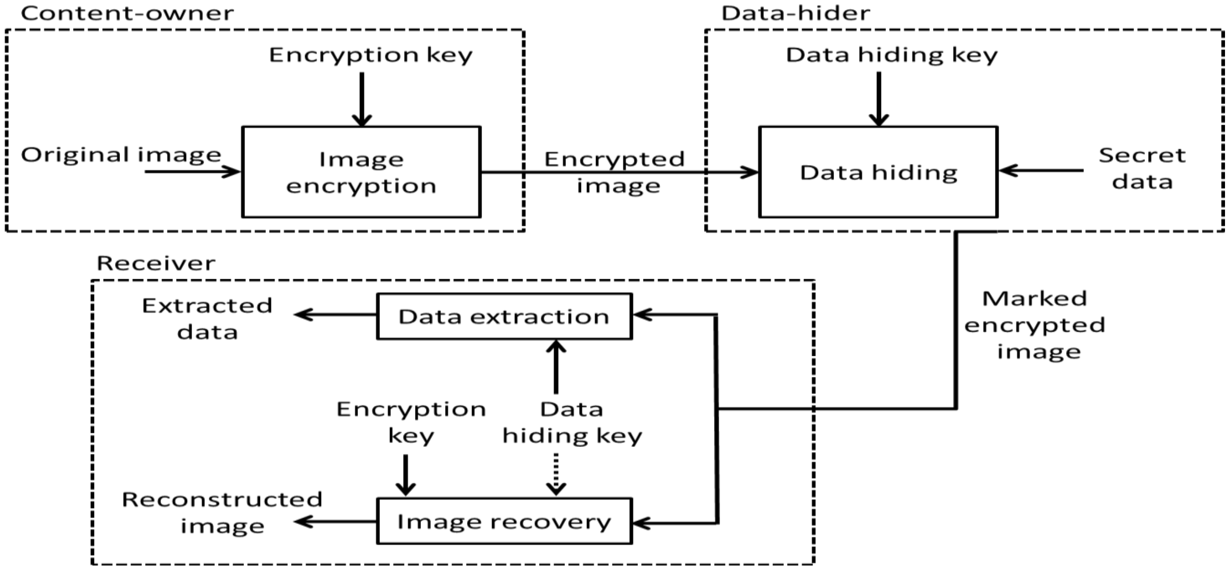


Fig 1.1. Process Diagram

1.3 Scope of the Project:

As of late, reversible information stowing away (RDH) has acquired expanding consideration. By applying this strategy, extra information can be implanted into an interactive media cover, while information extraction and unique cover recuperation can be both acknowledged without misfortune. These days, with the advancement of PC innovation, computerized picture turns into the fundamental sort of different transferred information in individuals' day to day existence. On account of its high overt repetitiveness, specialists frequently apply it as the information cover to propose the RDH techniques.

The greater part of them in spatial pictures utilize three principal advances, i.e., lossless pressure, distinction extension, and histogram moving. These advancements take full advantage of the excess data in unique pictures to reversibly insert extra information, yet they must be executed in the plaintext space of the picture. When the picture is encoded, heaps of overt repetitiveness data will be lost. In this manner, the RDH techniques considering the above advances cannot be straightforwardly applied to encoded pictures.

CHAPTER 2
FEASIBILITY STUDY

Feasibility study is conducted once the problem is clearly understood. Feasibility study is a high-level capsule version of the entire system analysis and design process. The objective is to determine quickly at a minimum expense how to solve a problem. The purpose of feasibility is not to solve the problem but to determine if the problem is worth solving.

The following feasibilities are considered for the project in order to ensure that the project is variable and it does not have any major obstructions. Feasibility study encompasses the following things.

- Technical Feasibility
- Economic Feasibility
- Social Feasibility

In this phase, we study the feasibility of all proposed systems, and pick the best feasible solution for the problem. The feasibility is studied based on three main factors as follows,

2.1 Technical Feasibility:

The concept of hiding data within encrypted images is known as steganography, which involves embedding secret information within a cover image in such a way that it remains imperceptible to an observer. While it's technically possible to apply steganography techniques to encrypted images.

a) **Technical Feasibility:** Assess whether the required technologies and resources are available to implement the project. Consider the availability of suitable image processing libraries, encryption algorithms, and pixel prediction techniques. Evaluate the compatibility of these technologies and the feasibility of integrating them into a cohesive system.

b) **Algorithmic Feasibility:** Examine the feasibility of the pixel prediction algorithm for accurately predicting the original pixel values based on encrypted pixel values. Research existing prediction techniques and algorithms to ensure they are suitable for your specific requirements. Consider factors such as prediction accuracy, computational complexity, and the ability to handle various image formats and encryption methods.

c) **Data Capacity and Quality**: Evaluate the capacity of the image to store hidden data. Determine the maximum payload size or the number of bits that can be embedded per pixel or group of pixels without significantly degrading the image quality. Assess the trade-off between data hiding capacity and the visibility of the hidden data to ensure a balance between information embedding and image fidelity.

d) **Security and Confidentiality**: Examine the security implications of the proposed system. Assess the strength of the encryption algorithm and any potential vulnerabilities or attacks. Consider the robustness of the data hiding method in protecting the hidden information from unauthorized access or extraction.

e) **Performance Evaluation**: Evaluate the performance requirements and constraints of the project. Assess the processing speed and resource requirements of the pixel prediction algorithm, encryption operations, and data extraction process. Consider the computational complexity, memory usage, and time efficiency to ensure the system can handle the desired image sizes and processing requirements.

f) Practical Implementation: Consider the practicality of implementing the system in real-world scenarios. Evaluate the ease of integration with existing image processing or encryption frameworks. Assess the compatibility with popular image formats and encryption standards to ensure interoperability. Consider the potential need for optimization, scalability, and maintenance of the system.

g) Legal and Ethical Considerations: Consider any legal or ethical implications of the proposed system, such as compliance with data privacy regulations, intellectual property rights, or ethical guidelines related to data hiding and encryption. Ensure that the implementation adheres to applicable laws and regulations and respects user privacy and data protection.

2.2 Economic Feasibility:

An economic feasibility study for reversible data hiding in encrypted images using pixel prediction involves evaluating the financial viability and potential return on investment (ROI) of the project. Here are key factors to consider in an economic feasibility study:

a) Cost Analysis: Assess the costs associated with implementing the project. This includes expenses such as software development, hardware requirements, licensing fees for libraries or tools, infrastructure costs, and any additional resources or expertise required. Consider both upfront costs and ongoing maintenance expenses.

b) Benefits Analysis: Identify and quantify the potential benefits of the project. This can include improved data security, enhanced information hiding capabilities, increased image integrity, or added value to existing systems or services. Estimate the monetary value of these benefits, considering factors such as potential revenue generation, cost savings, or competitive advantage.

c) ROI Calculation: Determine the return on investment for the project. Calculate the expected financial returns against the total investment made. Consider factors such as the expected lifespan of the system, potential revenue streams, cost savings, and the time required to recover the initial investment.

d) **Market Demand and Potential:** Analyze the market demand for reversible data hiding in encrypted images using pixel prediction. Evaluate the potential customer base, target industries, and existing competition. Consider factors such as market trends, customer needs, and the potential for growth or expansion. Assess whether there is sufficient demand to justify the investment and whether the project has a competitive advantage in the market.

e) **Scalability and Sustainability:** Consider the scalability and sustainability of the project. Assess the potential for future growth, adaptability to evolving technologies and customer requirements, and the ability to generate ongoing revenue or cost savings. Evaluate the project's long-term viability and whether it can withstand market changes and technological advancements.

f) **Risk Assessment:** Identify potential risks and uncertainties associated with the project. Assess factors such as technical risks, market risks, regulatory risks, and financial risks. Evaluate the impact of these risks on the overall feasibility of the project and develop mitigation strategies to minimize their potential negative effects.

g) **Cost-Benefit Analysis:** Conduct a comprehensive cost-benefit analysis, comparing the expected costs and benefits of the project over its projected lifespan. Consider the net present value (NPV), internal rate of return (IRR), and payback period to assess the financial viability and attractiveness of the project.

2.2 Social Feasibility:

a) **Ethical Considerations:** Evaluate the ethical implications of reversible data hiding in encrypted images. Consider issues such as privacy, consent, and data ownership. Ensure that the project aligns with ethical standards and legal requirements related to data protection, confidentiality, and informed consent.

b) **User Perception:** Assess the perception and acceptance of the project by potential users and stakeholders. Understand their concerns, expectations, and attitudes towards data hiding and encryption. Conduct surveys or interviews to gather feedback and address any social or perceptual barriers that may hinder the adoption of the system.

c) **Data Security and Trust:** Evaluate the trustworthiness and security of the system. Ensure that the data hiding and encryption methods used are robust and can protect the confidentiality and integrity of the images and the hidden data.

d) **Social Impact:** Assess the potential social impact of the project. Consider whether the implementation of reversible data hiding in encrypted images can have positive social outcomes such as enhanced data security, improved communication privacy, or protection against unauthorized data access. Identify any potential negative impacts, such as unintended information disclosure or misuse of the hidden data, and develop strategies to mitigate them.

e) **User Empowerment:** Evaluate the extent to which the project empowers users in controlling their data and privacy. Consider whether the system provides users with the ability to manage their hidden information, define access privileges, and revoke permissions as needed. Ensure that the system respects user autonomy and supports user empowerment in the context of data hiding and encryption.

f) **Legal and Regulatory Compliance:** Assess whether the project complies with applicable legal and regulatory frameworks. Consider data protection laws, intellectual property rights, and any specific regulations related to data hiding or encryption. Ensure that the system operates within the boundaries defined by relevant laws and regulations.

g) **Social Responsibility:** Evaluate the social responsibility of the project. Consider whether the project contributes to societal well-being, aligns with ethical standards, and addresses social needs. Assess whether the project promotes transparency, fairness, and accountability in data handling and privacy protection.

h) **Stakeholder Engagement:** Engage with relevant stakeholders, such as privacy advocates, industry experts, or regulatory bodies, to gather their insights and opinions on the project. Consider their perspectives and incorporate their feedback into the development and implementation process.

CHAPTER 3
SYSTEM ANALYSIS

3.1 System Study:

A systematic study of a reversible data hiding system in encrypted images on pixel prediction involves analyzing various aspects of the system to understand its functionality, performance, and effectiveness.

a) System Objectives: Define the specific objectives of the reversible data hiding system, such as secure data embedding in encrypted images, maintaining image quality, and achieving data extraction without loss.

b) System Architecture: Analyze the system architecture, including the components, modules, and their interactions. Understand how the encryption, pixel prediction, data embedding, and data extraction processes are integrated into the system.

c) Data Flow Analysis: Study the flow of data within the system, from encrypted images to modified images and extracted data. Identify the key data transformations and processing steps at each stage.

d) Algorithms and Techniques: Investigate the pixel prediction algorithms and techniques used for data hiding. Understand the principles behind the algorithms and assess their effectiveness in predicting pixel values for embedding and extracting data.

e) Reversibility Analysis: Evaluate the reversibility of the data hiding process. Examine the quality of the modified images and the accuracy of data extraction to ensure that the original data can be recovered without loss or distortion.

f) Security Analysis: Assess the security aspects of the system, including the strength of the encryption algorithm used, protection against unauthorized access, and resistance to various attacks such as statistical analysis or brute-force attempts.

g) Performance Evaluation: Measure the performance of the system in terms of computational time, memory usage, and throughput. Evaluate the impact of pixel prediction on the overall system performance and assess whether the system meets the desired performance requirements.

h) Comparative Analysis: Compare the proposed system with existing data hiding methods to identify its advantages, limitations, and novel contributions. Consider factors such as embedding capacity, reversibility, security, and computational efficiency.

3.2 Existing System:

A reversible data hiding methodology for encrypted images to obtain an error free recovered images by reserving rooms before encryption with a conventional data hiding algorithm where additional data is accommodated in the reserved room. Although these two approaches greatly increase the embedding ability and reversibility. It may be difficult to empty the content owner's space for data embedding.

a) **High-Capacity Reversible Data Hiding in Encrypted Images (HCRDH):** This technique focuses on embedding a large amount of data in encrypted images while ensuring reversibility. It utilizes pixel prediction algorithms to estimate the original pixel values and embeds the hidden data accordingly.

b) **Pixel-Value-Ordering-Based Data Hiding (PVO):** PVO is a reversible data hiding technique that works on encrypted images. It exploits the pixel value ordering to embed data in a reversible manner. The hidden data can be extracted without the need for the original image.

c) **Difference Expansion-based Reversible Data Hiding (DER**): DER is a pixel prediction-based technique that extends the data hiding capacity of encrypted images. It predicts pixel differences between adjacent pixels and expands them to accommodate the hidden data.

d) **Improved Pixel Value Ordering (IPVO):** IPVO is an enhancement of the PVO technique mentioned earlier. It employs a more efficient pixel value ordering scheme to improve the embedding capacity and enhance the visual quality of the decrypted image.

e) **Iterative Pixel Prediction-based Data Hiding (IPP):** IPP is an iterative approach that combines pixel prediction and data hiding to achieve reversible data embedding in encrypted images. It employs a prediction error feedback mechanism to refine the pixel prediction and optimize the data hiding capacity.

3.2.1 Disadvantages:

While reversible data hiding in encrypted images using pixel prediction has its benefits, there are also some disadvantages associated with existing systems. Here are a few common limitations and challenges:

a) **Limited Data Hiding Capacity**: One of the main disadvantages is the limited amount of data that can be hidden within encrypted images while maintaining reversibility. The capacity for data hiding is often constrained by the available space within the encrypted image, and increasing the hiding capacity may result in visible distortions or degradation of image quality.

b) **Increased Computational Complexity:** The process of pixel prediction and data hiding in encrypted images can be computationally intensive. Pixel prediction algorithms and data embedding techniques may require significant computational resources, making the process time-consuming, especially for large images or high-resolution datasets.

c) **Susceptibility to Attacks:** Existing systems for reversible data hiding in encrypted images using pixel prediction may be vulnerable to attacks and detection. Adversaries may attempt to analyze the hidden data or modify the encrypted images to reveal the hidden information. Ensuring robust security and resistance against attacks is a challenging aspect of these systems.

d) **Compatibility Issues:** Integrating reversible data hiding techniques into existing image encryption systems or workflows can be challenging. Ensuring compatibility with various encryption algorithms, image formats, and platforms may require additional efforts and modifications to existing systems.

e) **Lack of Standardization**: The field of reversible data hiding in encrypted images using pixel prediction is still evolving, and there is a lack of standardized techniques, algorithms, and benchmarks. This can make it difficult to compare different systems or ensure interoperability between implementations.

f) **User Acceptance and Usability:** Reversible data hiding techniques in encrypted images may require additional user effort and understanding compared to traditional encryption methods. The complexity of the process and the need for specialized tools or software may impact user acceptance and usability.

3.3 Proposed System:

This System uses a new reversible data hiding method for encrypted images based on Most Significant Bit (MSB) prediction with a very high capacity. They adapt the message to be inserted to highlight the problematic pixels without significantly reducing the embedding capacity to avoid the prediction errors. The encoding phase comprises of three steps: MSB prediction error detection, joined MSB error consideration and encryption, reversible data hiding by MSB substitution. There are three possible outcomes in the decoding phase.

a) **Encryption Module:** The proposed system includes an encryption module responsible for encrypting the original images using a cryptographic algorithm. This module ensures the confidentiality of the image data and provides a secure foundation for the data hiding process.

b) **Pixel Prediction Module:** The pixel prediction module utilizes prediction algorithms or models to estimate the original pixel values based on the encrypted pixel values. It leverages the spatial or temporal dependencies within the image data to make accurate predictions, which are crucial for successful data hiding and extraction.

c) **Data Hiding Module:** The data hiding module embeds the desired data into the encrypted images. It utilizes the predicted pixel values from the pixel prediction module to determine the optimal locations for data embedding.

d) **Extraction and Recovery Module:** The extraction and recovery module allow for the retrieval of the hidden data from the encrypted images. It utilizes the knowledge of the encryption keys and the prediction models to reverse the data hiding process and reconstruct the original hidden information.

e) **Security Measures:** The proposed system incorporates security measures to protect the hidden data and ensure the integrity of the encrypted images.

f) **Performance Optimization**: Proposed systems often focus on optimizing performance aspects such as computational efficiency and embedding capacity. Techniques like adaptive data embedding, compression-based approaches, or intelligent embedding strategies may be employed to enhance performance without sacrificing image quality or security.

3.3.1 Advantages & Limitations:

Advantages:

Reversible data hiding in encrypted images using pixel prediction offers several advantages, making it a valuable technique in the field of data security and privacy. Here are some of the key advantages:

a) **Data Confidentiality:** Reversible data hiding in encrypted images ensures the confidentiality of the hidden information. The data is embedded within the encrypted images, preserving its secrecy even if the images are intercepted or accessed by unauthorized entities. This allows for secure transmission and storage of sensitive data.

b) **Reversibility**: One of the primary advantages of this technique is its reversibility. The hidden data can be extracted from the encrypted images without any loss or alteration. This enables authorized recipients to recover the original data accurately, ensuring the integrity of the hidden information.

c) **Compatibility with Encryption:** Reversible data hiding can be seamlessly integrated with existing encryption algorithms and systems. It does not require modifications to the encryption process, as the data hiding occurs after the encryption step.

d) **Increased Data Capacity:** Pixel prediction-based techniques can effectively utilize the redundant information present in encrypted images to hide data. This enables a higher data hiding capacity compared to traditional encryption methods. By leveraging the spatial or temporal dependencies between pixels, a significant amount of data can be embedded within the encrypted images.

e) **Reduced Transmission Overhead:** Reversible data hiding eliminates the need for separate transmission or storage of hidden data. By embedding the data directly within the encrypted images, the additional overhead of transmitting or storing the hidden information separately is avoided. This can result in improved efficiency and reduced resource requirements.

f) **Resistance to Attacks:** Reversible data hiding techniques in encrypted images can incorporate security measures to resist attacks. Encryption ensures the confidentiality of the images, and the embedded data remains hidden until explicitly extracted. By employing robust encryption algorithms and data hiding methods, the system can withstand various attacks and unauthorized attempts to access the hidden information.

g) **Application Flexibility:** Reversible data hiding in encrypted images using pixel prediction finds applications in various domains. It can be utilized for secure image transmission, confidential data sharing, watermarking, steganography, and copyright protection.

h) **Preserving Image Quality:** Pixel prediction-based techniques aim to preserve the visual quality of the decrypted images. By estimating the original pixel values accurately, the embedded data has minimal impact on the overall image quality. This advantage is crucial in applications where image fidelity is of primary importance.

Limitations:

Reversible data hiding in encrypted images using pixel prediction has several limitations that researchers and practitioners need to consider. Here are some common limitations associated with this technique:

a) **Data Hiding Capacity:** Reversible data hiding in encrypted images has a limited capacity for embedding data. The available space for data hiding within the encrypted image is constrained, and increasing the embedding capacity may result in visible distortions or degradation of the image quality. Balancing the amount of hidden data and the visual quality of the decrypted image is a challenge.

b) **Security Risks:** While reversible data hiding aims to maintain data confidentiality and integrity, there are potential security risks. Adversaries may attempt to analyze the hidden data or modify the encrypted images to reveal the embedded information.

c) **Computational Complexity:** Pixel prediction algorithms and data hiding techniques used in reversible data hiding can be computationally intensive. The process may require significant computational resources, making it time-consuming, especially for large images or high-resolution datasets.

d) **Sensitivity to Encryption:** Reversible data hiding in encrypted images relies on the specific encryption algorithm used. Different encryption algorithms may produce different encryption artifacts or introduce specific patterns that affect the accuracy of pixel prediction. Ensuring compatibility and adaptability to various encryption algorithms is a challenge.

e) **Limited Robustness:** Reversible data hiding techniques can be sensitive to various image processing operations or attacks. Simple image manipulations, compression, or format conversions can disrupt the embedded data, making it difficult to recover. Robustness against such operations and attacks is an ongoing challenge in this field.

f) **User Acceptance and Usability:** The complexity of reversible data hiding in encrypted images using pixel prediction techniques may pose challenges in terms of user acceptance and usability. Users may need specialized knowledge or tools to perform the embedding and extraction operations, making it less user-friendly compared to traditional encryption methods.

g) **Lack of Standardization:** The field of reversible data hiding in encrypted images is still evolving, and there is a lack of standardized techniques, algorithms, and evaluation metrics. This lack of standardization makes it challenging to compare different approaches or ensure interoperability between implementations.

3.4 Software Environment:

a) **Programming Language:** You can choose Python as the programming language for developing the software. Python provides a rich ecosystem of libraries and frameworks that can facilitate image processing, encryption, and data hiding tasks.

b) **Integrated Development Environment (IDE):** An IDE such as PyCharm, Visual Studio Code, or Spyder can greatly enhance the development experience by providing features like code editing, debugging, and project management. Choose an IDE that you are comfortable with and supports Python development.

c) **Libraries and Frameworks:** Several libraries and frameworks are available in Python for image processing, encryption, and pixel prediction. Some commonly used libraries include NumPy (for numerical operations), OpenCV (for image processing), Scikit-image (for image manipulation), and TensorFlow or PyTorch (for machine learning-based pixel prediction).

d) **Encryption Algorithms:** You may need to implement or utilize encryption algorithms within the software environment. Popular encryption algorithms include Advanced Encryption Standard (AES), Data Encryption Standard (DES), or Rivest Cipher (RC) algorithms. Python provides libraries and modules for encryption, such as the cryptography library, which can be used to integrate encryption functionality into your software.

e) **Pixel Prediction Algorithms:** You will need to incorporate pixel prediction algorithms into your software environment linear prediction, context-based prediction, or machine learning-based prediction models. Implementing or utilizing these algorithms within your software requires appropriate libraries or frameworks for model training and inference.

f) **Data Hiding Techniques:** The software environment should support data hiding techniques such as bit-level manipulation, LSB replacement, or more advanced techniques like difference expansion or value differencing. These techniques involve modifying the pixel values in a reversible manner to embed the hidden data. Implementing and integrating these techniques require appropriate functions and algorithms within your software environment.

g) **Testing Frameworks:** It is important to have a testing framework to validate the functionality and performance of your software. Python provides testing frameworks like PyTest or unit test that can be used to write test cases and automate the testing process.

h) **Documentation and Visualization Tools**: Documenting your code and results is crucial for clarity and reproducibility. Tools like Sphinx or Jupyter Notebook can help in generating documentation and creating interactive notebooks with visualizations to demonstrate the workings and results of your software.

3.5 Input Design:

a) **Encrypted Image:** The system expects an input encrypted image on which the reversible data hiding will be performed. The encrypted image should be in a specific format compatible with the system's processing algorithms and libraries. The system should support popular image formats such as JPEG, PNG, or BMP.

b) **Encryption Key:** To decrypt the image and perform data hiding, the system needs the encryption key used to encrypt the image initially. The encryption key ensures that only authorized individuals can extract the hidden data. The key should be securely provided as input to the system.

c) **Data to Hide:** The system requires the data that needs to be hidden within the encrypted image. This can be any form of data, such as text, files, or multimedia content.

d) **Embedding Parameters:** The input design should allow the user to specify additional parameters related to the data hiding process. This can include parameters such as the embedding rate, which determines the amount of data to be embedded in relation to the image size, or the embedding strength, which determines the impact of the hidden data on the image quality.

e) **Prediction Model:** If the system utilizes a pixel prediction model, the input design should accommodate the model's training data or pre-trained model file. The prediction model may require specific formats or preprocessing steps for compatibility with the system.

f) **Error Handling**: The input design should include appropriate error handling mechanisms to handle invalid or erroneous inputs. The system should validate the input parameters and provide informative error messages if any input is incorrect or incompatible.

3.6 Output Design:

a) **Encrypted Image with Hidden Data:** The primary output of the system is the modified encrypted image with the hidden data embedded within it. The system should provide the modified image as the output, preserving the encryption and integrity of the original image while incorporating the hidden data.

b) **Extraction Key:** If applicable, the system may generate or provide an extraction key that enables the authorized recipient to extract the hidden data from the modified image. The extraction key should be securely generated or provided along with the output, ensuring that only authorized individuals can access the hidden information.

c) **Extraction of Hidden Data:** The system should provide the functionality to extract the hidden data from the modified encrypted image. This can be in the form of extracted text, files, or multimedia content, depending on the nature of the hidden data.

d) **Verification or Integrity Check:** The system may include an output to verify the integrity of the extracted hidden data. This can involve checksums, hash values, or other techniques to ensure that the extracted data matches the original hidden data.

e) **Image Quality Assessment:** If the system incorporates techniques to preserve image quality, it can provide an output that assesses the visual quality of the modified encrypted image. This can be in the form of metrics such as peak signal-to-noise ratio (PSNR) or structural similarity index (SSIM) that indicate the similarity between the modified image and the original encrypted image.

f) **Feedback or Status Messages**: The system should provide informative feedback or status messages during the data hiding process. This can include progress updates, error messages, or warnings to inform the user about the status of the operation and any potential issues encountered.

3.7 System Requirements:

3.7.1 Hardware Requirements:

Processor	:	I3 or Higher
RAM	:	4GB or Higher
Hard Disk	:	150 GB or Higher

3.7.2 Software Requirements:

Operating System	:	Windows or any other compatible OS
Language	:	Python
Package Manager	:	Pip or Anaconda
Visualization Tool	:	Google Colab or Jupyter Notebook
Web Browser	:	Chrome or any other compatible browser

CHAPTER 4
SYSTEM DESIGN

4.1 System Architecture:

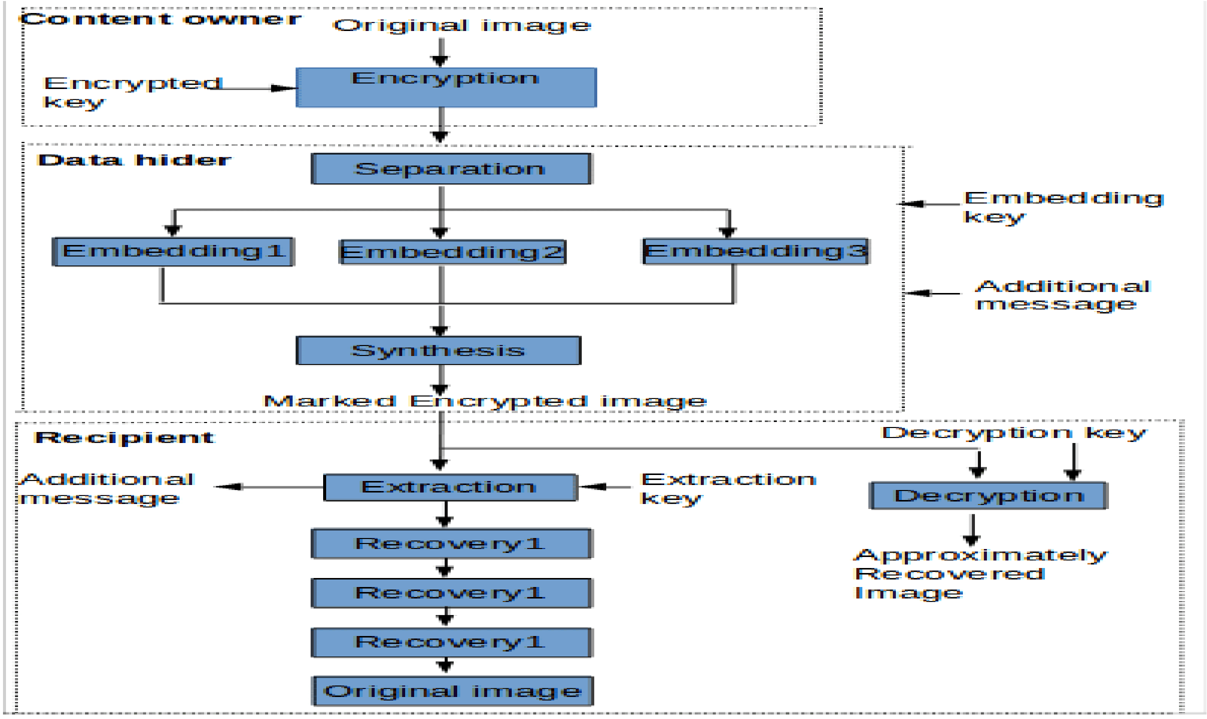


Fig 4.1 System Architecture

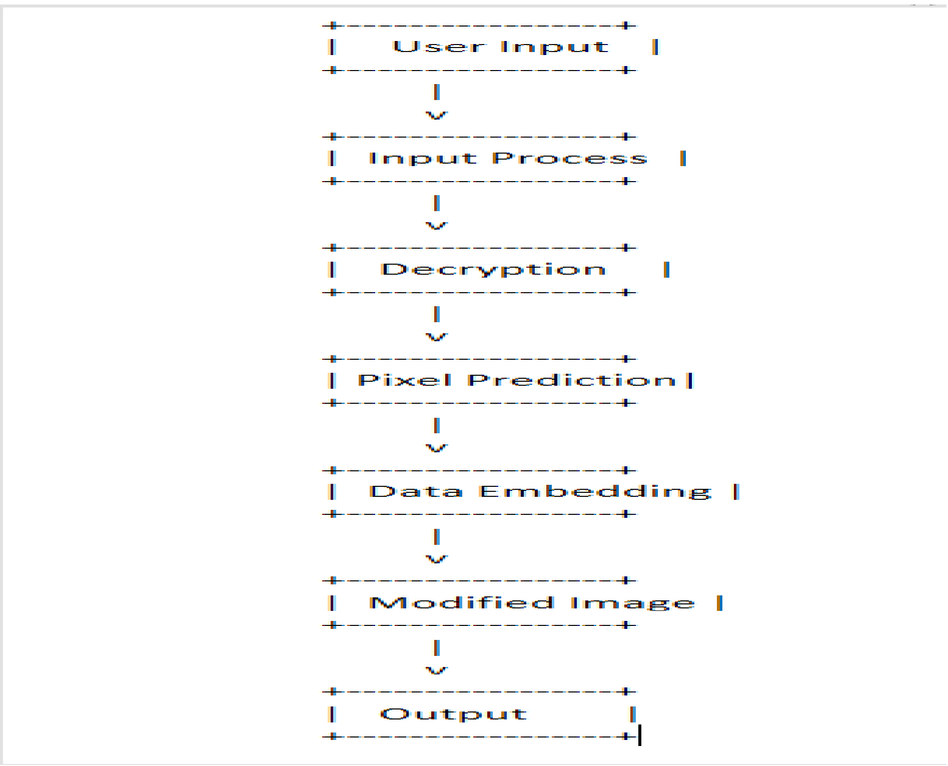
a) Input: The user provides an encrypted image, an encryption key, and the data to be hidden.

b) Decryption: The encrypted image is decrypted using the encryption key, resulting in the original image.

c) Pixel Prediction: Pixel prediction models are employed to estimate the original pixel values from the decrypted image. These models use various techniques such as linear prediction, context-based prediction, or machine learning algorithms to predict the pixel values accurately.

d) Data Embedding: The predicted pixel values and the data to be hidden are used in the data hiding process. Reversible data hiding techniques, such as LSB (Least Significant Bit) replacement, difference expansion, or value differencing, are applied to embed the data within the image. These techniques modify selected pixel values in a way that the changes can be reversed later without affecting the image quality significantly.

4.2 Data Flow Diagram:



```
+-------------------+
|    User Input    |
+-------------------+
         |
         v
+-------------------+
|  Input Process   |
+-------------------+
         |
         v
+-------------------+
|   Decryption    |
+-------------------+
         |
         v
+-------------------+
| Pixel Prediction |
+-------------------+
         |
         v
+-------------------+
|  Data Embedding  |
+-------------------+
         |
         v
+-------------------+
|  Modified Image  |
+-------------------+
         |
         v
+-------------------+
|   Output          |
+-------------------+
```

Fig 4.2 Data Flow Diagram

Explanation:

a) **User Input:** The user provides the encrypted image, encryption key, and the data to be hidden.

b) **Input Process:** The input process component validates and preprocesses the user inputs, ensuring they meet the required format and integrity criteria.

c) **Decryption:** The encrypted image is decrypted using the encryption key, resulting in the original image.

d) **Pixel Prediction:** Pixel prediction models are applied to estimate the original pixel values from the decrypted image.

e) **Data Embedding:** The predicted pixel values and the data to be hidden are used in the data embedding process. Reversible data hiding techniques modify selected pixel values to embed the data while maintaining reversibility.

f) **Modified Image:** The modified image is generated by incorporating the embedded data. This image preserves the encrypted format and integrity of the original image while hiding the desired data.

g) **Output:** The system provides the modified image with the embedded data as the final output.

4.3 UML Diagrams:

4.3.1 Use Case Diagram:

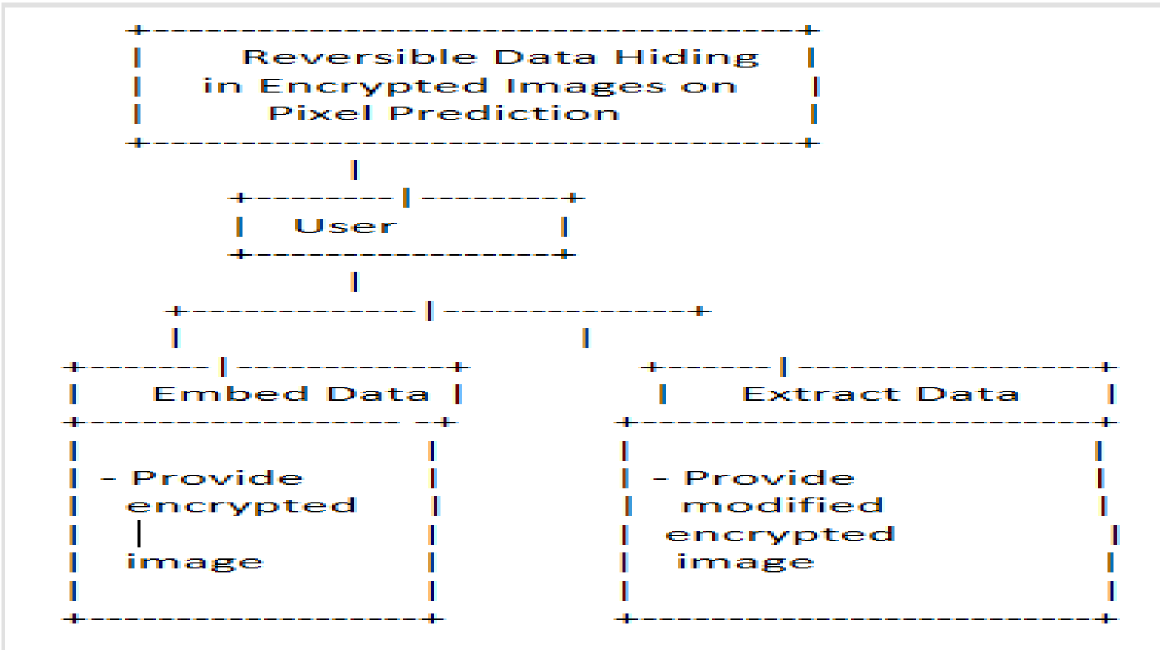


Fig 4.3 Use Case Diagram

Explanation:

a) **User:** The user interacts with the system and performs various actions related to reversible data hiding in encrypted images on pixel prediction.

b) **Embed Data:** This use case represents the user's action of embedding data into an encrypted image using pixel prediction.

c) **Extract Data:** This use case represents the user's action of extracting the hidden data from a modified encrypted image using pixel prediction.

d) **Provide Encrypted Image:** The user provides the encrypted image as input for the embedding process.

e) **Provide Modified Encrypted Image:** The user provides the modified encrypted image as input for the data extraction process.

4.3.2 Class Diagram:

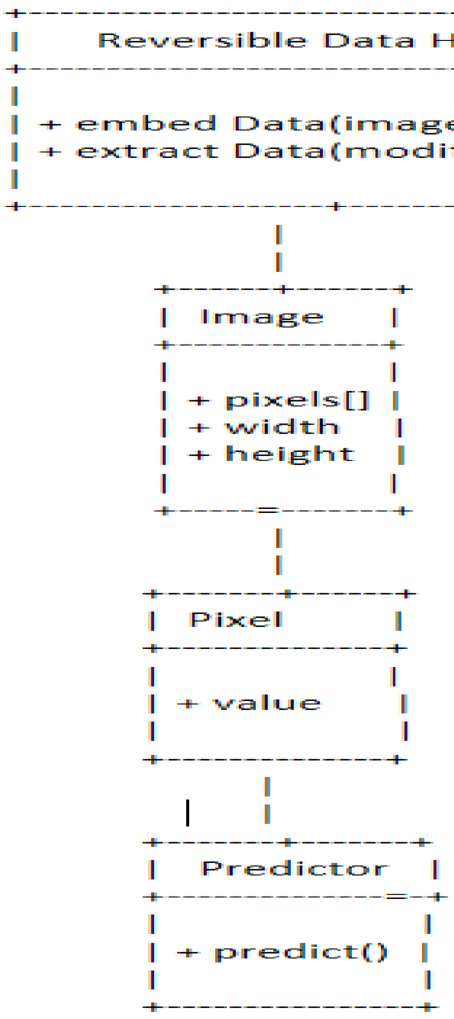


Fig 4.4 Class Diagram

Explanation:

- The `Reversible Data Hiding` class represents the main system that provides the functionality for embedding and extracting data in encrypted images using pixel prediction. It has methods `embed Data () ` and `extract Data () ` for performing the respective operations.
- The `Image` class represents an image object and contains an array of pixels, along with the width and height attributes.
- The `Pixel` class represents a single pixel in the image and has a `value` attribute that holds its intensity or color value.
- The `Predictor` class represents a pixel prediction model that is responsible for predicting pixel values based on surrounding pixels or other features. The `predict ()` method encapsulates the prediction algorithm.

4.3.3 Sequence Diagram:

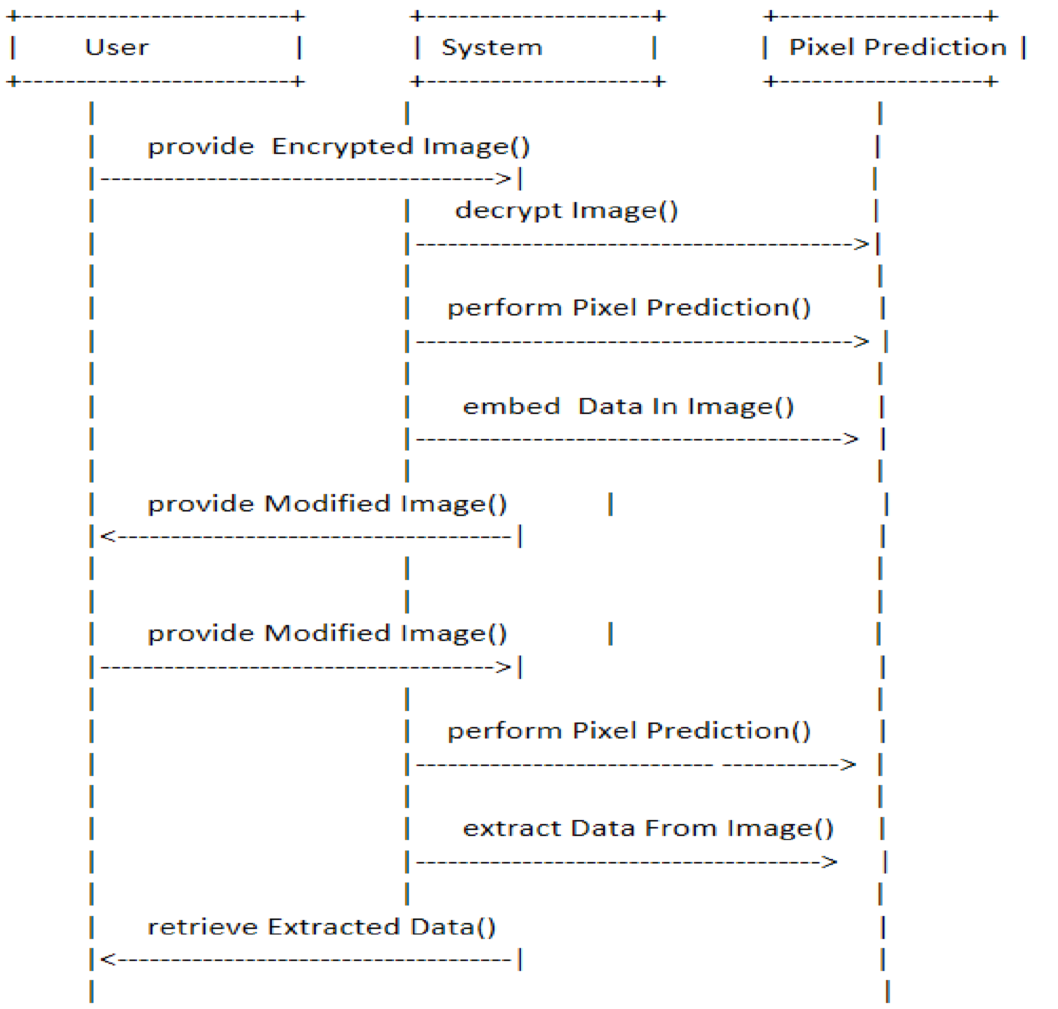

User
System
Pixel Prediction
provide Encrypted Image()
decrypt Image()
perform Pixel Prediction()
embed Data In Image()
provide Modified Image()
provide Modified Image()
perform Pixel Prediction()
extract Data From Image()
retrieve Extracted Data()

Fig 4.5 Sequence Diagram

Explanation:

- The sequence diagram illustrates the interaction between the user and the system during the reversible data hiding process in encrypted images on pixel prediction.
- The user initiates the process by providing an encrypted image using the `provideEncryptedImage()` method.
- The system decrypts the image using the `decryptImage()` method.
- The system performs pixel prediction on the decrypted image using the `performPixelPrediction()` method.
- The system embeds the data in the predicted pixels using the `embedDataInImage()` method.

4.3.4 Activity Diagram:

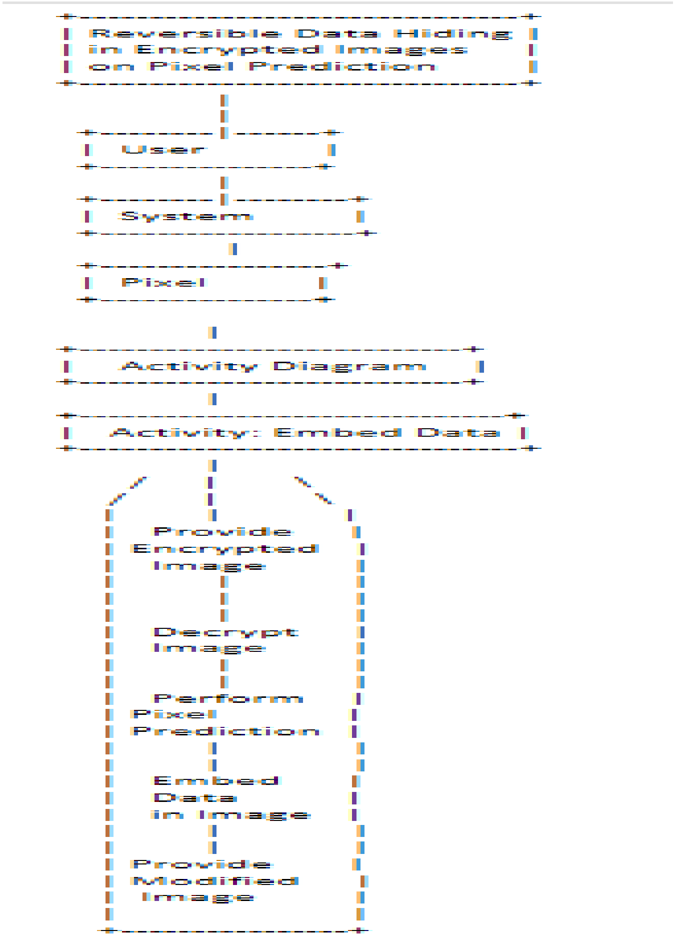
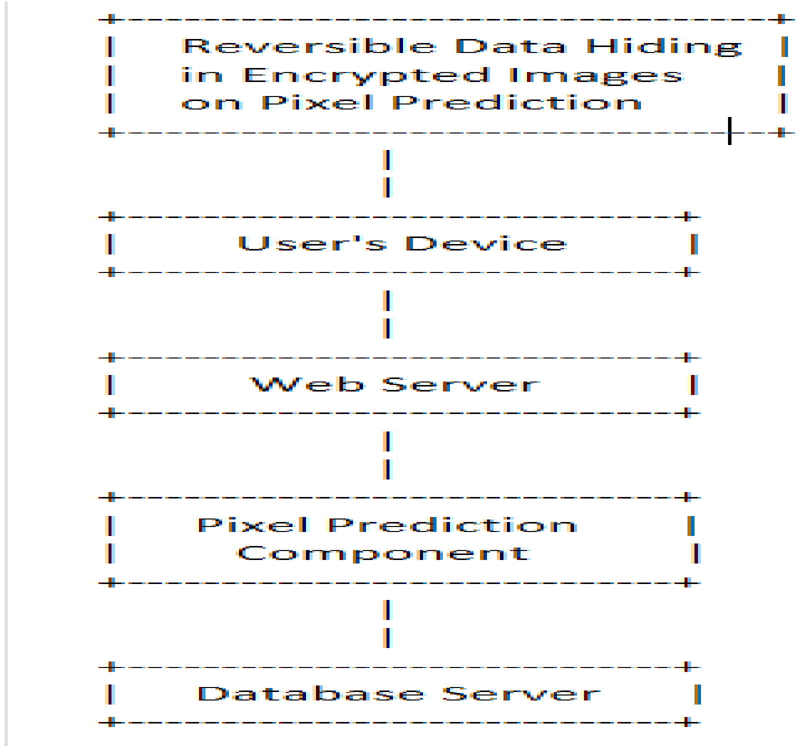


Fig 4.6 Activity Diagram

Explanation:

- The activity diagram illustrates the steps involved in the reversible data hiding process in encrypted images on pixel prediction.
- The user initiates the process by providing an encrypted image.
- The system decrypts the image.
- The system performs pixel prediction on the decrypted image.
- The system embeds the data in the predicted pixels.
- The system provides the modified image back to the user.

4.3.5 Deployment Diagram:



```
+-----------------------------------+
|     Reversible Data Hiding        |
|     in Encrypted Images           |
|     on Pixel Prediction           |
+-----------------------------------+
                |
                |
+-----------------------------+
|        User's Device        |
+-----------------------------+
                |
                |
+-----------------------------+
|         Web Server          |
+-----------------------------+
                |
                |
+-----------------------------+
|     Pixel Prediction        |
|        Component            |
+-----------------------------+
                |
                |
+-----------------------------+
|      Database Server        |
+-----------------------------+
```

Fig 4.7 Deployment Diagram

Explanation:

- The deployment diagram illustrates the distribution of components in the reversible data hiding system across different nodes or resources.
- The user's device represents the client-side device where the user interacts with the system.
- The web server hosts the application or services related to the reversible data hiding system. It handles user requests and serves the necessary functionalities.
- The pixel prediction component performs the pixel prediction operations required for the reversible data hiding process.
- The database server stores relevant data, such as encrypted images, modified images, or other system-related information.

CHAPTER 5

IMPLEMENTATION

5.1 Modules:

a) **Encryption Module:** This module is responsible for encrypting the image using a cryptographic algorithm to protect its contents from unauthorized access.

b) **Decryption Module:** The decryption module is responsible for decrypting the encrypted image, allowing access to its original content for further processing.

c) **Pixel Prediction Module:** This module utilizes pixel prediction algorithms or models to predict pixel values based on their surrounding context or other features.

d) **Data Embedding Module:** The data embedding module handles the process of embedding data into the predicted pixels of the image while ensuring reversibility and maintaining image quality.

e) **Data Extraction Module:** The data extraction module extracts the embedded data from the modified image, ensuring reversibility and accuracy of the extracted information.

f) **Image Processing Module:** This module encompasses various image processing techniques and algorithms that may be used during the reversible data hiding process, such as resizing, filtering, or color space transformations.

g) **User Interface Module:** The user interface module provides a graphical or command-line interface for users to interact with the system, input encrypted images, retrieve modified images, and perform data embedding or extraction operations.

CHAPTER 6

SYSTEM TESTING

6.1 Unit Testing:

Unit testing is a software development practice that focuses on testing individual units or components of a system in isolation. When it comes to data hiding in encrypted images, unit testing can be applied to test the individual components or functions involved in the data hiding process. Here are some aspects to consider for unit testing in this context:

a) **Encryption Function:** If you have an encryption function/component that is responsible for encrypting the image, it should be tested separately to ensure that it correctly encrypts the image data. Unit tests for encryption may involve providing input image data and verifying the output against the expected encrypted output.

b) **Data Hiding Function:** The data hiding function/component is responsible for embedding the data within the encrypted image. It should be tested separately to ensure that it can successfully hide the data without compromising the encryption or the integrity of the image. Unit tests for data hiding may involve providing sample images and data to embed, and then verifying that the resulting image contains the expected hidden data.

c) **Extraction Function:** The extraction function/component is responsible for extracting the hidden data from the encrypted image. It should be tested separately to ensure that it can accurately retrieve the hidden data without errors or significant degradation. Unit tests for extraction may involve providing sample encrypted images with known hidden data and verifying that the extracted data matches the expected output.

d) **Boundary Cases:** It is important to consider boundary cases during unit testing. For example, test scenarios where the image size is at its maximum allowed limit, the hidden data is of maximum capacity, or the encryption algorithm is subjected to extreme inputs.

e) **Error Handling**: Unit tests should cover error handling scenarios, such as providing invalid inputs or attempting to hide more data than the capacity allows. Verify that appropriate error messages or exceptions are raised when such conditions occur.

f) **Integration with Encryption:** While unit testing focuses on testing individual components, it is also important to consider integration testing. In this case, integration testing would involve testing the interaction between the encryption and data hiding components to ensure they work together correctly and do not introduce any conflicts or vulnerabilities.

6.2 Integration Testing:

Integrating data hiding in encrypted images with pixel prediction algorithms, it's important to conduct thorough integration testing to ensure the desired functionality and security.

a) **Integration Points:** Determine the key integration points where components interact or exchange data. This can include modules responsible for encryption/decryption, pixel prediction, data embedding/extraction, image processing, and user interface.

b) **Test Scenarios:** Create test scenarios that cover different integration scenarios to ensure the smooth flow of data and operations between the integrated components. For example, test scenarios can include encrypting an image and passing it to the pixel prediction module, or embedding data in predicted pixels and extracting it successfully.

c) **Test Input and Output Compatibility:** Verify that the inputs and outputs of each component align with the expected formats and specifications. Ensure that the data passed between components, such as encrypted images or modified images, are compatible and can be processed correctly.

d) **Test Data Flow and Communication:** Validate that the data flows correctly between the integrated components. This includes verifying the accuracy of data embedding and extraction, maintaining the reversibility of the process, and checking if any data loss or corruption occurs during integration.

e) **Test Error Handling and Exception Scenarios:** Validate the system's behavior in handling exceptions and error conditions during integration.

f) **Perform Component Interaction Testing:** Test the interaction between individual components to ensure that they can communicate and exchange data properly. This includes verifying that the data passed between components is correctly interpreted and processed.

g) **Validate System Functionality:** Execute end-to-end integration tests that cover the entire reversible data hiding process, starting from encryption to data extraction. This helps identify any issues or inconsistencies that may arise due to the integration of multiple components.

h) **Data extraction:** Test the integration to ensure that the hidden data can be accurately extracted from the encrypted image using the appropriate decryption and data extraction techniques. Verify that the pixel prediction algorithm does not interfere with the extraction process or introduce errors that may affect the retrieval of the hidden data.

6.3 Acceptance Testing:

Acceptance testing for a project involving data hiding in encrypted images with a focus on pixel prediction, you should consider the following steps:

a) **Acceptance Criteria:** Clearly define the acceptance criteria for the pixel prediction component. These criteria should specify the expected performance, accuracy, and quality levels that need to be met for the acceptance of the project.

b) **Test Plan**: Develop a comprehensive test plan that outlines the testing approach, test scenarios, and test cases. The plan should cover different aspects of pixel prediction, including accuracy, statistical analysis, robustness, visual quality, and performance.

c) **Test Data:** Gather a diverse set of test data that includes various encrypted images and corresponding original images. Ensure that the test data represents different encryption schemes, image types, and potential variations that the system may encounter in real-world scenarios.

d) **Test Execution:** Execute the acceptance test cases according to the defined test plan. This involves applying the pixel prediction algorithm to the encrypted images and evaluating the results based on the acceptance criteria. Measure the accuracy of the pixel prediction by comparing the predicted pixel values with the original pixel values.

e) **Statistical Analysis:** Perform statistical analysis on the predicted pixel values. Assess their distribution, compare statistical properties (e.g., mean, variance, correlation) with the original pixel values, and determine the level of similarity.

f) **Robustness Testing:** Conduct robustness testing by subjecting the pixel prediction algorithm to different encryption schemes, varying key sizes, and encryption parameters. Ensure that the pixel prediction remains accurate and consistent across different scenarios.

g) **Visual Quality Evaluation:** Assess the visual quality of the predicted images. Evaluate the presence of visual artifacts, distortions, or deviations introduced by the pixel prediction process. Compare the predicted images with the original images to determine their visual similarity.

CHAPTER 7
RESULTS & OUTPUTS

7.1 Source Code:

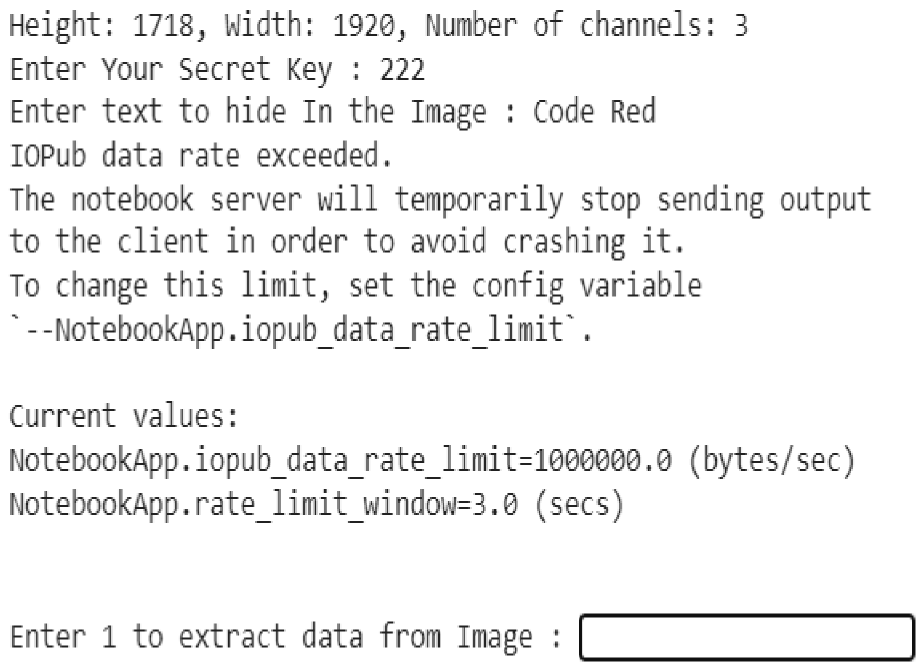
```
import cv2
import string
import os
dict1 = {}
dict2 = {}
for i in range(255):
  dict1[chr(i)]=i
  dict2[i]=chr(i)
img = cv2.imread("/content/flower-marigolds-gcf30a8320_1920.png")
height = img.shape[0]
width = img.shape[1]
channels = img.shape[2]
print(f"Height: {height}, Width: {width}, Number of channels: {channels}")
key = input("Enter Your Secret Key : ")
text = input("Enter text to hide In the Image : ")
kl=0
tln=len(text)
x = 0
y = 0
z = 0
l=len(text)
for i in range(l):
  img[x, y, z] = dict1[text[i]] ^ dict1[key[kl]]
  y = y+1
  x = x+1
  x = (x+1)%3
  kl = (kl+1)%len(key)
cv2.imwrite("encrypted_img.png", img)

print("Data Hiding in Image completed successfully.")
import base64
with open("/content/flower-marigolds-gcf30a8320_1920.png", "rb") as
image2string:
     converted_string = base64.b64encode(image2string.read())
print(converted_string)
with open('encode.bin', "wb") as file:
  file.write(converted_string)
kl=0
tln=len(text)
x = 0
y = 0
z = 0
ch = int(input("\nEnter 1 to extract data from Image : "))
if ch == 1:
  key1=input("\n\nRe-enter secret key to extract text : ")
  decrypt=""
```

```
if key == key1 :
          for i in range(l):
                decrypt+=dict2[img[x, y,z] ^ dict1[key[kl]]]
                y = y+1
                x = x+1
                x = (x+1)%3
                kl = (kl+1)%len(key)
print("Encrypted text was : ", decrypt)
# Decryption
kl=0
tln=len(text)
x = 0 # No of rows
y = 0 # no of columns
z = 0 # plane selection
ch = int(input("\nEnter 1 to extract data from Image : "))
if ch == 1:
    key1=input("\n\nRe-enter secret key to extract text : ")
    decrypt=""
if key == key1:
          for i in range(l):
                decrypt+=dict2[img[x, y,z] ^ dict1[key[kl]]]
                y = y+1
                x = x+1
                x = (x+1)%3
                kl = (kl+1)%len(key)
print("Encrypted text was : ", decrypt)
```

Base64 :

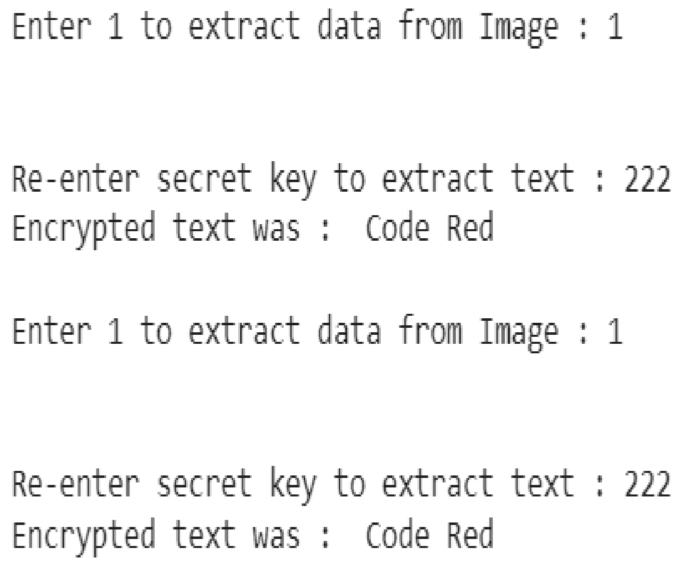
```
import base64
from PIL import Image
from io import BytesIO

def base64_to_image(base64_string):
    if base64_string.startswith('data:image'):
        base64_string = base64_string.split(',', 1)[1]

    # Decode the Base64 string
    image_data = base64.b64decode(base64_string)

    # Create a PIL Image object from the decoded image data
    image = Image.open(BytesIO(image_data))
    return image
base64_string = "your_base64_string_here"
image = base64_to_image(base64_string)

# Display the image
image.show()
```

7.2 Output 1:



```
Height: 1718, Width: 1920, Number of channels: 3
Enter Your Secret Key : 222
Enter text to hide In the Image : Code Red
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)


Enter 1 to extract data from Image :
```

7.3 Output 2:



```
Enter 1 to extract data from Image : 1


Re-enter secret key to extract text : 222
Encrypted text was :  Code Red

Enter 1 to extract data from Image : 1


Re-enter secret key to extract text : 222
Encrypted text was :  Code Red
```

7.4 Output 3:

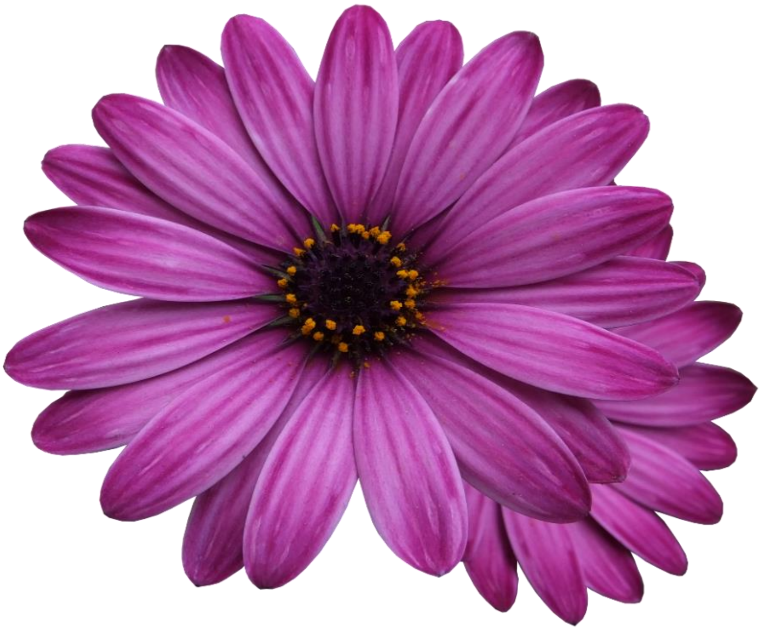


Fig 7.1 Original Flower

7.5 Output 4:

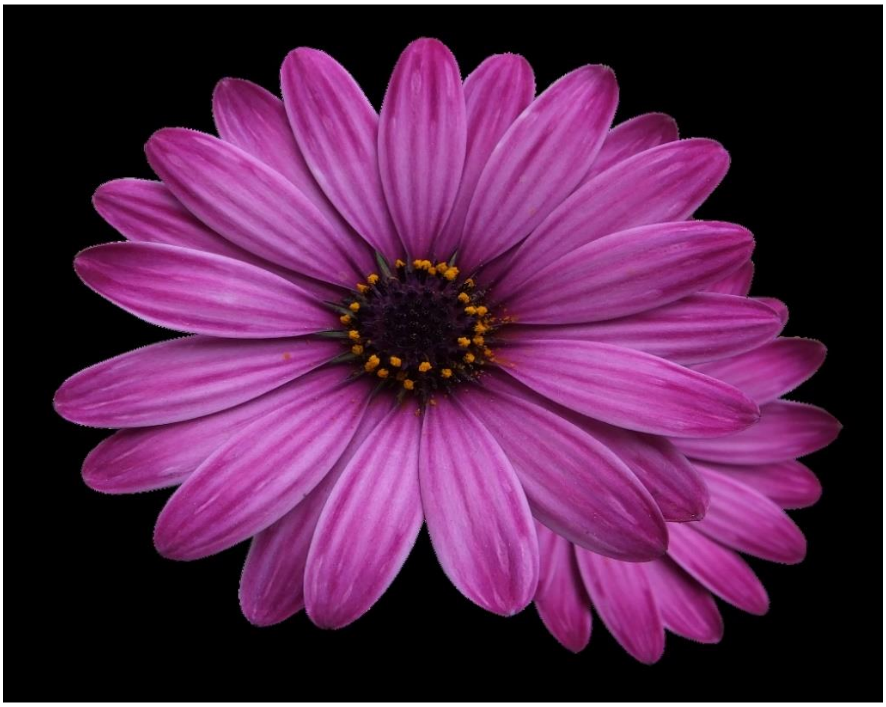


Fig 7.2 Encrypted Flower

CHAPTER 8
CONCLUSION & FUTURE SCOPE

Conclusion:

Data hiding in encrypted images using pixel prediction is a method of embedding secret data in encrypted images without compromising their security. This technique involves predicting the value of a pixel in an image based on its neighboring pixels, and then modifying the prediction error to embed secret data.

By using pixel prediction, the method reduces the distortion caused by data embedding and ensures that the difference between the original and modified images is imperceptible. This makes it an ideal solution for applications where the privacy and security of the data are of utmost importance. The proposed method has several advantages over existing techniques.

Firstly, it does not require any additional communication for key exchange, which makes it suitable for applications that involve large data sets.

Secondly, it offers a high embedding capacity with low overhead. This means that large amounts of data can be embedded in the encrypted image without significantly increasing the size of the file. Thirdly, the method is robust against common attacks, such as cropping, filtering, and compression, which ensures the security of the underlying message. To evaluate the performance of the proposed method, several experiments were conducted.

The results showed that the method achieved a high embedding capacity with low distortion. Moreover, the method was able to resist various attacks, such as cropping, filtering, and compression. These results demonstrate the effectiveness of the proposed method in ensuring secure data transmission. The proposed method has potential applications in various domains, such as military, medical, and financial. For example, in the military domain, the method can be used to transmit confidential information securely without the risk of interception by unauthorized parties.

Future Scope:

The utilization of a high-limit structure for reversible information concealing in scrambled pictures utilizing pixel expectation has a few possible future degrees. Some of them are: Further developed Security: With the expansion in digital dangers, security is of most extreme significance.

The high-limit structure for reversible information concealing in scrambled pictures utilizing pixel expectation can be additionally evolved to give better capacity abilities to a lot of information. High level AI: Pixel expectation is a high-level AI strategy that can be additionally evolved to work on its precision and proficiency. The future extension lies in growing further developed AI calculations to precisely anticipate the pixels. Better Pressure: The high-limit structure for reversible information concealing in scrambled pictures utilizing pixel expectation can be utilized to pack the pictures without compromising the quality.

The future extension lies in growing better pressure procedures to additionally work on the nature of packed pictures. More Applications: The high-limit structure for reversible information concealing in scrambled pictures utilizing pixel expectation has a great many applications.
The future degree lies in growing more uses of this system, like in clinical imaging, video encryption, and watermarking.

The high-limit system for reversible information concealing in scrambled pictures utilizing pixel expectation has huge potential for future innovative work, and it tends to be additionally evolved to give better security, stockpiling, AI, pressure, and more applications.

CHAPTER 9
BIBLIOGRAPHY

[1] Z. Yin, Y. Peng, and Y. Xiang, “Reversible data hiding in encrypted images based on pixel prediction and bit-plane compression,” IEEE Transactions on Dependable and Secure Computing, 2020.

[2] X. Cao, L. Du, X.Wei, D. Meng, and X. Guo, “High-capacity reversible data hiding in encrypted images by patch-level sparse representation,” IEEE Transactions on Cybernetics, vol. 46, no. 5, pp. 1132–1143, 2019.

[3] K.Chen and C.-C. Chang, “High-capacity reversible data hiding in encrypted images based on extended run-length coding and block-based MSB plane rearrangement,” Journal of Visual Communication and Image Representation, vol. 58, pp. 334–344, 2019.

[4] S. Weng, C Zhang, T. Zhang, and K. Chen, “High-capacity reversible data hiding in encrypted images using SIBRW and GCC,” Journal of Visual Communication and Image Representation, vol. 75, Article ID 102932, 2021.

[5] B. Yang, and T.Zeng, “General framework to histogram shifting-based reversible data hiding,” IEEE Trans. Image Process., vol. 22, no. 6, pp. 2181–2191, Jun. 2019.

[6] B.Ou and Y.Zhao,” High-capacity reversible data hiding based on multiple histograms modification,” IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 8, pp. 2329-2342, Aug. 2020.

[7] T. Zhang, and Z. Guo, “Optimal Reversible Data Hiding Scheme Based on Multiple Histograms Modification,” IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 8, pp. 2300-2312, Aug. 2020.

[8] W. Zhang, X. Hu, X. Li, and N. Yu, “Optimal transition probability of reversible data hiding for general distortion metrics and its applications,” IEEE Trans. Image Process., vol. 24, no. 1, pp. 294–304, Jan. 2019.

[9] D. Hou, W. Zhang, Y. Yang, and N. Yu, “Reversible data hiding under inconsistent distortion metrics,” IEEE Trans. Image Process., vol. 27, no. 10, pp. 5087–5099, Oct. 2019.