iMark: Invisible Watermarking in Live Captured Photo Using DCT and RSA

An Undergraduate Thesis
Presented to
the Faculty of Computer Science
and Information Technology Department
BICOL UNIVERSITY COLLEGE OF SCIENCE
Legazpi City

In Partial Fulfillment of the
Requirements for the
Degree of
BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Karlo Jeric L. Balucio

Christian James E. Concepcion

Ma. Darlene J. Malasa

May 2024

Republic of the Philippines
Bicol University
College of Science
Legazpi City

RECOMMENDATION FOR ORAL DEFENSE

The undergraduate thesis entitled, "**iMark: Invisible Watermarking in Live Captured Photo Using DCT and RSA**", prepared and submitted by **KARLO JERIC L. BALUCIO**, **CHRISTIAN JAMES E. CONCEPCION** and **MA. DARLENE J. MALASA** in Partial fulfillment of the requirements for the degree BACHELOR OF SCIENCE IN COMPUTER SCIENCE, is hereby submitted to the thesis committee for oral examination.

LEA D. AUSTERO, DIT
Content Adviser

RYAN A. RODRIGUEZ, MIT, MSCS
Programming Adviser

In partial fulfillment of the requirements for the degree Bachelor of Science in Information Technology, this undergraduate thesis entitled, "**iMark: Invisible Watermarking in Live Captured Photo Using DCT and RSA**", prepared and submitted by **KARLO JERIC L. BALUCIO**, **CHRISTIAN JAMES E. CONCEPCION,** and **MA. DARLENE J. MALASA** is hereby recommended for Oral Examination.

THESIS COMMITTEE

JENNIFER L. LLOVIDO, DIT
Member

MICHAEL ANGELO BROGADA, DIT
Member

ARLENE A. SATUITO
Chair, Defense Panel

Republic of the Philippines
Bicol University
College of Science
Legazpi City

RESULTS OF FINAL DEFENSE

Researchers: **KARLO JERIC L. BALUCIO**

CHRISTIAN JAMES E. CONCEPCION

MA. DARLENE J. MALASA

Title: **iMark: Invisible Watermarking in Live Captured Photo Using DCT and RSA**

Place: Bicol University Building 2 room 104/105

Date:

Time: 10-11 AM

PANEL OF EXAMINERS	**ACTION**
ARLENE A. SATUITO Chair, Defense Panel	_____
JENNIFER L. LLOVIDO, DIT Panel Member	_____
MICHAEL ANGELO D. BROGADA, DIT Panel Member	_____

`

Republic of the Philippines
Bicol University
College of Science
Legazpi City

APPROVAL SHEET

Upon recommendation of the Oral Examination Committee, this undergraduate thesis entitled, "**iMark: Invisible Watermarking in Live Captured Photo Using DCT and RSA**", prepared and submitted by **KARLO JERIC L. BALUCIO, MA. DARLENE J. MALASA** and **CHRISTIAN JAMES E. CONCEPCION**, are hereby approved in partial fulfillment of the requirements for the degree **Bachelor of Science in Computer Science.**

LEA D. AUSTERO, DIT
Content Adviser

RYAN A. RODRIGUEZ, MIT, MSCS
Programming Adviser

THESIS COMMITTEE

JENNIFER L. LLOVIDO, DIT
Member

MICHAEL ANGELO BROGADA, DIT
Member

ARLENE A. SATUITO
Chair, Defense Panel

Accepted and approved for the conferral of the degree **Bachelor of Science in Computer Science**

RYAN A. RODRIGUEZ, MIT, MSCS
Department Chair

JOCELYN E. SERRANO, M.Sc
Dean, BUCS

Bicol University
College of Science
COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY DEPARTMENT
Legazpi City

CONTENT ADVISER'S CERTIFICATION

This is to certify that this undergraduate thesis entitled, "**iMark: Invisible Watermarking in Live Captured Photo Using DCT and RSA**", prepared and submitted by **KARLO JERIC L. BALUCIO**, **MA. DARLENE J. MALASA**, and **CHRISTIAN JAMES E. CONCEPCION**., in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science, has been read and edited by the undersigned.

Issued this _ day of __ at Bicol University College of Science, Legazpi City.

LEA D. AUSTERO, DIT
Content Adviser

Bicol University
College of Science
COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY DEPARTMENT
Legazpi City

PROGRAMMING ADVISER'S CERTIFICATION

This is to certify that this undergraduate thesis entitled, "**iMark: Invisible Watermarking in Live Captured Photo Using DCT and RSA**", prepared and submitted by **KARLO JERIC L. BALUCIO**, **MA. DARLENE J. MALASA**, and **CHRISTIAN JAMES E. CONCEPCION**., in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science, has been evaluated by the undersigned.

Issued this _ day of __ at Bicol University College of Science, Legazpi City.

RYAN A. RODRIGUEZ, MIT, MSCS
Programming Adviser

Bicol University
College of Science
COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY DEPARTMENT
Legazpi City

EDITOR'S CERTIFICATION

This is to certify that this undergraduate thesis entitled, "**iMark: Invisible Watermarking in Live Captured Photo Using DCT and RSA**", prepared and submitted by **KARLO JERIC L. BALUCIO**, **MA. DARLENE J. MALASA**, and **CHRISTIAN JAMES E. CONCEPCION**., in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science, has been evaluated by the undersigned.

Issued this _ day of __ at Bicol University College of Science, Legazpi City.

ALYSSA NICOLE C. LODANA, LPT
Editor

ACKNOWLEDGEMENT

This thesis would not have been possible without the guidance and support of several individuals who contributed and extended their valuable assistance in the preparation and completion of this research study.

This study would like to thank their adviser, Ma'am Lea D, Austero, DIT, for her encouragement and guidance of this study

To their programming adviser, Sir Ryan A. Rodriguez MIT, MSCS**,** for his support and guidance in conducting the research and giving advice in developing the application;

To Their Thesis 1 Professor Lea D. Austero, DIT, whose unwavering guidance and encouragement were instrumental in propelling us through the challenges of this research project. Your dedication to fostering problem-solving skills within our thesis class has left an indelible mark on this work. We are deeply grateful for your contributions.

Also for their Thesis 2 professor, Sir Benedicto B. Balilo Jr., DIT, for his dedicated support and guidance in time of writing the manuscript.

The researchers would also like to express their utmost gratitude to Ma'am Arlene A. Satuito, Ma'am Jennifer L. Llovido, DIT, and Sir Michael Angelo D. Brogada, DIT for encouraging them to pursue this study.

To their friends, classmates, and families, an earnest appreciation for their never-ending support and encouragement which inspired the research.

The researchers would like to extend their deepest gratitude to Sir Ryan Rodriguez for his unwavering and tireless support during their moments of greatest

need. His dedication and selflessness have been a beacon of hope and strength for all the researchers. Without his invaluable assistance and encouragement, they would not have been able to overcome the numerous challenges they faced. Thank you, Sir Ryan, for your boundless generosity and for being a true pillar of support in their journey. His kindness will always be remembered and cherished.

To the previous researchers of this research topic about invisible watermarking Justine J. Baldovino, Jefferson F. Clemente, and Exequiel M. Lustan for authorizing the researchers to use their research as the foundational basis of the researchers' endeavors.

Above all, to our Lord GOD, for the strength, pearl of knowledge, and patience. Your guidance made the researchers believe that everything is possible with hard work and teamwork.

K.J.B

C.J.C

M.D.M

ABSTRACT

KARLO JERIC L. BALUCIO, CHRISTIAN JAMES E. CONCEPCION, and **MA.DARLENE J. MALASA**, “iMark: Invisible Watermarking in Live Captured Photo Using DCT and RSA” (Unpublished Undergraduate Thesis, Bicol University College of Science, Legazpi City, May 2024)

The non-blind semi-blind watermarking technique using DCT and RSA was utilized as invisible watermarking for digital images. This technique features instantaneous embedding, extracting, and detecting invisible watermarks in live captured images using a smartphone’s camera. The watermarked images are evaluated to determine their imperceptibility and robustness against filter and geometric transformation attacks applied to the live captured watermarked images. A mobile application was developed to read and validate the required input (images and keys) from the user to process embedding, detecting, extracting, and displaying the corresponding result of the process performed by the algorithm. This paper presents the proposed Non-Blind-Semi blind Watermarking technique based on DCT and RSA algorithms. The watermarking scheme used the RSA method to process three distinctive keys: encryption, decryption, and seed keys. The watermarked images are robust against blur and noise attacks and weak against geometric transformation attacks in output evaluation. Future researchers are recommended to add more algorithms for creating invisible watermarking to withstand attacks and increase its robustness. Additional features could include enabling users to use images from storage to embed more watermarks in the already-watermarked images, using other metrics to evaluate the robustness of the image, and adding the capability to detect tampered watermarked images.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER I
INTRODUCTION

Background of the study

In today's world, where smartphones are everywhere and mobile applications are widely used for capturing and sharing photos, protecting digital content has become a top priority. The ease of sharing images across different platforms has also increased the risk of unauthorized use and distribution of photographs. To address this issue, invisible watermarking has emerged as a sophisticated solution that allows information to be embedded within images, enabling the identification and tracking of the content's origin.

Wei et al, (2020), expressed that while traditional watermarking techniques have been used for years to protect the intellectual property associated with digital images, visible watermarks can be intrusive and detract from the visual appeal of the image. This has led to a growing demand for invisible watermarking methods. This research focuses on exploring the feasibility and effectiveness of invisible watermarking specifically designed for live-captured photos within mobile applications.

Najafri, E.(2019) stated mobile applications have become the primary means of capturing and sharing photos in real time. However, the dynamic nature of mobile photography poses unique challenges for implementing invisible watermarking.

This research aims to bridge this gap by investigating innovative techniques that allow invisible watermarks to be embedded in real-time, ensuring that the integrity of the image is preserved without any noticeable alterations.

Techniques used for encryption can be said to be the protection tools for secret data. Encryption is the mechanism where the plain data can be converted to cipher or protected data, and it can be read only by decrypting it. The reverse process of encryption is known as decryption, which uses an encryption key to decrypt the original data. Data encryption has become the most important for all the secret data, especially when used over the internet, extranets, or intranets. The encryption is done by applying a mathematical function that generates a key; later, the key is used to get the encrypted/ciphered data. Modern encryption algorithms are designed to be highly secure, making unauthorized access extremely difficult. Additionally, the use of encryption is vital in protecting sensitive information such as financial transactions, personal communications, and intellectual property.

Skaf (2019) states that Watermark is a message, usually a logo, stamp, or signature superimposed onto an image, with a great deal of transparency. This protects any digital photos or works of people institutions in the means of copyright and interests, preventing any illegal use without permission from the authors. Watermarks also serve as a signature to make sure that other people know who the author is, especially if they want to know if the digital image came from a certain institution

Alshoura et al. (2021) state that DCT is a very popular transform function used in signal processing tasks and manipulations. It transforms a signal from a spatial domain to a frequency domain. Due to its good performance, it has been used in JPEG

standards for image compression. DCT has been applied in many fields such as data compression, pattern recognition, image processing, and so on. This versatility has made it an essential tool in modern digital communication systems.

The study's aim is to create a mobile-based application that would instantaneously take a picture and then add an invisible watermark to the images using the Digital Image Processing technique with the help of an invisible watermarking scheme. Evaluating the embedded watermarks after applying certain common attacks to the watermark images to showcase its robustness and imperceptibility.

Objectives of the study

The study's main objective is to implement and evaluate a comprehensive invisible watermarking scheme on live captured digital images. The scheme will focus on imperceptibility and robustness, ensuring the watermark is imperceptible by human observers while being robust to certain common attacks. Specifically, the present study aims:

1. To develop an Android-based application that features embedding, extracting, and detecting invisible watermarks on digital images;
2. To utilize a Non-blind - Semi Blind watermarking technique using the RSA and DCT algorithm for embedding and extracting images; and
3. To evaluate the robustness and weaknesses of the proposed watermark scheme against image geometric transformations and signal processing attacks using SSIM and PSNR.

Significance of the study

This study aims to contribute to the development of a novel watermarking scheme that offers these functionalities. The focus on imperceptibility and robustness ensures seamless integration into digital images without compromising visual quality. The study also helps advanced researchers find solutions to protect owners in the digital world. The study would benefit the following:

Software Developers and Technology Companies. The study's findings can inform the development of new watermarking tools and applications for various purposes, such as content management systems, digital rights management, and image authentication platforms.

Digital Signal Processing Researchers (DSP). Researchers can apply their expertise to design robust watermarking embedding and extraction algorithms. They can enhance methods for embedding watermarks that are resistant to common signal processing manipulations, such as filtering and geometric transformations.

The General Public. By improving content security and facilitating tamper detection, this research can contribute to a more trustworthy digital environment for everyone

Computer Science. This study contributes to the field of Computer Science by expanding knowledge and experience in writing thesis papers. This, in turn, benefits future exploration and investigation endeavors.

Scope and Delimitation

Development of an Android mobile application that allows users to embed invisible watermarks in digital images and extract them to evaluate the extracted watermarks robustness.

The application is accessible on various Android devices, including smartphones and tablets. Watermarks to be used are only limited to JPEG/JPG and PNG. The embedded watermarks are robust against various attacks, such as noise addition, compression, and filtering. The study also includes using measuring tools to evaluate the quality of the watermarked images and the effectiveness of the watermarking algorithm against attacks. The image captured used for watermarking, cannot be used again for embedding watermark to it.

CHAPTER II

RELATED LITERATURE AND STUDIES

This section presents existing literature and studies that were relevant and significant to the current study. It also includes a synthesis of the study, a conceptual framework, and the definition of terms for further comprehension of the study.

Digital Watermarking in Digital Images

According to the study conducted by Wang et al (2018), Multimedia such as images and video is becoming more and more indispensable to people's lives. People share images on Facebook, Twitter, and other social media platforms. We watch a variety of videos every day - TV series, movies, and so on. Along with the rapidly developing technologies, we have a growing demand for definition of video and images. However, piracy caused great harm to the interest of multimedia data copyright owners. Then, copyright protection gets more and more attention in academia and industry. Images, video audio, and text files are losing their credibility day by day as they can be distorted or manipulated by using several tools. Digital exchange of contents such as photos, text, audio, video,and with others, has been significantly enhanced due to the internet. Because of this, copying and manipulation of these digital contents became easier according to Kulkarni et al (2019).

According to Anuja and Rahul Dixit (2018) highlight the critical issue of ensuring authenticity and integrity in digital media. The ease with which forgery tools manipulate content makes it increasingly difficult to discern genuine from fabricated. This vulnerability exposes multimedia data to illegal distribution, duplication, and manipulation, raising concerns about the information it conveys.

Wang (2019), Along with the rapidly developed technology, we have a growing demand for definition of video and images. However, piracy caused great harm to the interests of multimedia data copyright owners. With multimedia sources and content, legitimate information security and other difficulties have become a significant concern. The so-called digital data type could be converted, altered, and copied to be widely distributed while keeping the data in a high-quality status. It has become relatively easy to manipulate digital images and create forgeries that are difficult to distinguish from authentic photographs.

According to Chourdhary et al. (2018), watermarking started as “distinctive marks on the paper.” The concept was first applied to bank notes to hide the originality and novelty of bank financial notes, and watermarking techniques are imprinted on them. Over time, watermarking evolved to include digital media, becoming a crucial method for protecting intellectual property online. Today, digital watermarking is widely used to safeguard images, videos, and other multimedia content against unauthorized use and distribution.

According to CAMEO, (2022), It is common practice up to the 20th century. After that watermarking was also used in postage stamps and currency notes of any country. One of the most common strategies used to protect and secure ownership and copyright is watermarking. Watermarking is a technique used to embed information into digital or physical media in a way that is difficult to remove or alter, thus protecting ownership and ensuring authenticity.In recent years, digital watermarking has gained prominence due to the widespread dissemination of digital content. This method is crucial not only for copyright protection but also for tracking and identifying the distribution path of multimedia files.

Bhargava, (2019), proposed that watermarking can be used in different aspects of ownership authenticity of any work done by an individual. It is one of the challenges to ensure the originality and authenticity of the works of the owner and the copyright of the specific digital data because hackers may violate it. By using watermarking techniques, they can detect or expose any modified ideas and ensure that the image is authentic, resulting in the images being used by the hospital professional every time.

As stated by Bernito et al, (2018), digital watermarking is a leading candidate that solves the problems regarding legal ownership including copyright protection, authentication, and data hiding. Digital watermarking helps detect and verify ownership and not allowing any other person to remove the digital watermark while protecting images from image manipulations and image processing**.** The roots of digital watermarking can be traced back to the pioneering work of Hembrooke, E. (1954), When he patented a method for embedding identification codes into music. Hembrooke's invention laid the foundation for a field that would blend technology with intellectual property management. At this core, digital watermarking was initially developed as a tool to address concerns related to the unauthorized use and distribution of digital content. The 1990s marked a pivotal turning point for digital watermarking. This surge in interest was largely fueled by the rapid prefiltration of digital media and the internet, which raised significant concerns about copyright infringement. Industry groups like the Copy Protection Technical Working Group (CPTWG) and Strategic Digital Music Initiative (SDMI) were established during this period, underscoring the need for robust digital rights management systems During this era, digital watermarking emerged as a promising solution to safeguard intellectual property in the digital realm.

The technology offered the potential to embed hidden information within multimedia content, allowing content creators and distributors to assert ownership and protect their assets. Digital watermarking found applications in various domains, including transaction tracking, proof of ownership, and broadcast monitoring. Its adaptability to different media types, including audio, video, and images, made it an invaluable tool in commercial and security applications.

According to Imatag, (2023), Academic research played a vital role in advancing digital watermarking techniques. Research tackled challenges related to fidelity, robustness, and security, aiming to refine and enhance watermarking methods. Their contributions were instrumental in making digital watermarking a more reliable and effective technology. According to Desoubeaux (2023), the potential of digital watermarking remains vast. Advancements in technologies like machine learning and blockchain are poised to open new avenues for watermarking applications. These developments may lead to novel use cases in content authentication, blockchain-based rights management, and enhanced interactive media experiences. The integration of watermarking with emerging technologies could offer unprecedented levels of security and interactivity. Transforming the way we engage with digital content. As we enter the era where AI-generated content becomes increasingly prevalent, digital watermarking is set to play a crucial role in distinguishing authentic content from that generated by Artificial intelligence. Furthermore, the ongoing standardization efforts in watermarking technology hold promise for its widespread adoption. While challenges related to security and standardization persist, the value of watermarking in content protection and management is undeniable. From its modest beginnings in the mid-20th century to its status as a cornerstone of digital content

management, digital watermarking has significantly shaped the digital landscape. Its evolution mirrors the technological advancements and changing needs of content protection management, and the verification of content authenticity. Digital watermarking's journey underscores its importance in safeguarding intellectual property and ensuring the integrity of digital content in an era marked by rapid technological progress and the growing prevalence of digital media.

Rahim et al, (2019) stated that there are three types of invisible watermarks: robust, fragile, and the semi-fragile. Most of the many watermarking schemes today are robust, to apply watermarking techniques in an image invisibly. Digital watermarking is one of the most effective techniques for hiding information. Several watermarking techniques are offered to conceal the secret information in the form of digital data such as text, audio, and video. Digital watermarking provides the facility to attach secret data in the cover image which could be afterward used for extraction or detection for several uses including identification of the owner.

According to Anju and Vandana (2019), visible watermarking is related to the human eye's perception. So, watermarks are embedded in the data and can be seen without extraction. Visible watermarks can be in the form of logos used in papers and videos. This type of watermarking ensures that the ownership or branding of the content is immediately apparent, thereby deterring unauthorized use. Alternately, invisible watermarking is inconspicuous to the human eye. It is embedded in the data without affecting the image quality. The owner or the person authorized can be the only one who can extract it. Invisible watermarks provide a layer of security and authenticity, allowing the rightful owner to prove ownership without altering the user experience. For example, distribute photos via the internet. As per Savakar et al, (2020), a visible watermark is easily exposed by the human eye. An invisible watermark is imperceptible or cannot be seen by the human eye. A visible

watermark is a fragile watermark because of how easily it can be altered and destroyed when an image is modified by a sequence of linear and non-linear transformations. The embedding algorithm used for embedding the invisible image watermark image into the target image is an imperceptible way that is undetectable to the user and the attacker and has a better solution for retaining the original quality and the message content of the cover image. Invisible image watermarking is regarded as the last line of defense of information security and perhaps the last significant barrier to protecting intellectual property rights in the digital world.

According to Brook (2023), there are different types of digital watermarking. Invisible watermarks are embedded in the media using steganography techniques and are invisible to the naked eye. They are used to prove the authenticity of digital content and are often used for copyright protection and as a means of hidden communication. On the other hand, visible watermarks are perceptible watermarks that can be readily seen by the human eye. They are typically in the form of brand logos, images, copyrighted text, personal signatures, and more. To establish the security arrangements and watermark validity, high computational complexity is required. For real-time applications, efficiency and speed are of great importance. Additionally, advancements in digital watermarking are continually being developed to counteract increasingly sophisticated piracy techniques. As technology evolves, the balance between robust watermarking methods and maintaining media quality remains a critical challenge.

According to Bytescote (2022), An invisible watermark is an embedded image that cannot be perceived with the human’s eyes. Only electronic devices (or specialized software) can extract the hidden information to identify the copyright owner.

Invisible watermarks mark specialized digital content (text, images, or even audio content) to prove its authenticity. Furthermore, A visible watermark is seen by viewers. It's the semi-transparent identifiers overlaid on the original images. These identifiers can be text, address, URL, logo, or codes, or anything to show the ownership of the images. While they are semi-transparent, they still allow the original images to be viewed.

Watermarking Pre-requisites

Anunja and Rahul Dixit, (2019) stated that watermarks should be capable of resisting malicious attacks, to prevail over common distortions and not easily emphasized. Watermarks should be able to adjust with other coexisting watermarks and should not be too complex for insertion and deletion. Watermark should satisfy the following properties: 1. Transparency: when a watermark is embedded in an image it should not distort the quality of the image or the message conveyed in the image. Embedded images should be perceptually invisible. The end user should not be able to observe any visual discrepancy. 2. Robustness: Watermarks should be able to resist various attacks. This could be geometrical or non-geometrical methods. Elimination or manipulation of watermarks should be impossible if they try to be tampered with without sufficient knowledge of the embedding process related to specific fields. 3. Security: Watermarks should be secured using secret keys so that they cannot be altered without specific knowledge of the secret key. Watermarks should be highly secured to protest all attempts of addition, deletion, and updating used by unauthorized persons. And lastly 4. Computational Complexity: The time taken by the watermarking algorithm in encoding and decoding is referred to as computational complexity. To establish the security arrangements and watermark validity, high computational complexity is

required. For real-time applications efficiency and speed are of great importance.

According to Chang, Tsai, and Lin, (2018), there are several essential requirements for a watermarking method to be effective. It must be undeletable or cannot be removed by any unauthorized persons. It should be perceptually invisible, and the watermark should be indistinguishable from the human eye, preventing an unauthorized person's manipulation. the fundamental requirements in digital watermarking schemes are invisibility and Robustness. The watermark should be invisible to ensure image quality and the process of watermark embedding. The embedded watermark must not be easily removed or altered by certain attacks.

Also as cited by (Yongjian et al, 2018), robust watermarks should have the following requirements: 1. They should be invisible and have no interference with the visible watermark. 2. Watermarking schemes can extract the invisible watermark without resorting to the visible watermark image. 3. Watermarks must be challenging to remove and resist malicious changes such as image compression and malicious attacks such as inpainting or replacement. Digital watermarking comprises various techniques to protect digital content from unauthorized access and modifications performed over them. Watermarking techniques can be divided into two main categories: spatial and transform domain. In the spatial domain, direct functions are performed over pixels. Watermarks are embedded in the spatial domain by altering pixel values.

Digital Image Watermarking Techniques and Algorithms

According to Singh et al, (2019), Digital watermarking comprises various techniques to protect digital content from unauthorized access and modifications performed over them. Watermarking techniques can be divided into two main

categories: spatial and transform domain. In the spatial domain, direct functions are performed over pixels. Watermarks are embedded in the spatial domain by altering pixel values. The discrete wavelet transform (DWT) is one of the popular methods watermarking methods for embedding a watermark in an image alongside Discrete Cosine Transform (DCT), Singular Value Decomposition, and lastly, the Discrete Fourier transform (DFT), watermark embedding can be done by spatial and transform domains. Also, according to Baghdad et al, (2018), using the spatial domain method as a watermarking technique puts the values directly in the pixel of the images which affects the robustness since it affects the quality of the image. Although it is easy to implement, also it can be easily attacked as a downside. For example, adding noise to the image can quickly downgrade the overall quality of the image. On the other hand, unlike the transform domain where it enhances the performance of the watermarking by selecting pixels that are more resistant and robust against attacks, it uses a modulator coefficient to be used in transforms like DFT, DCT, DWT, and SVD; these transforms increase the efficiency of the watermarking scheme. In the case of the transform domain DCT, DWT, and DFT are the most widely used techniques for watermarking techniques. To obtain robustness and security, transform domain techniques are more effective than spatial domain watermarking techniques.

Elbasi et al, (2022), stated there are three categories in watermarking detection which are Non-blind, Blind, and Semi-blind watermarking. The use of a blind technique to detect watermarks does not require the use of the original image or keys. The key and the watermarked documents are necessary and required to detect in the semi-blind watermarking process. Furthermore, according to Bors and Pitas, (2018), the watermarking algorithm enables the insertion of a hidden code within the images. The

watermarked attached would be invulnerable to image-altering algorithms, which also include the standard compression algorithms. Many different methods were proposed for watermarking, and these calculations install the sign alterations in the picture force area or the recurrence space. Utilizing the eight × eight square DCT coefficients for installing the watermark in the image was proposed. The DCT is applied to the whole picture, and the watermarking strategy is like the spread range procedures. Transform domains such as DFT, DCT, DWT, and SVD are significantly more robust than spatial domains due to energy compaction principles. Watermarks are embedded in less perceptually significant coefficients, making them less vulnerable to common image processing operations. Furthermore, spatial domains are more susceptible to noise filtering, and compression attacks. Modification directly affects pixel values, making watermarks vulnerable to manipulation.

Watermarking Schemes

The study of Priya et al. (2018) proposed an improved DWT-DCT-SVD algorithm in watermarking and fulfilled the criteria to prove the hypothesis. The proposed watermarking scheme guarantees invisibility and is robust against removal and geometric attacks. After attempting most attacks, both primary and secondary watermarks are extracted from the distorted watermarked image with high normalized correlation values. The algorithm has more significant image quality and robustness performance than the other watermarking algorithms. The standard deviation of the cover image’s second and first-level mid-frequency coefficients is used in both blind and non-blind methods, respectively.

In the research conducted Mansrah and Mohammed (2019), proposed a blind watermark, an imperceptible and robust watermarking technique using DCT and DWT

to apply the hidden watermark. Combining these two mathematical transforms has resulted in more significant features and results, ensuring the essential watermarking requirements, including robustness and imperceptibility.

In the study of Run et al. (2018), the proposed two watermarking schemes are proposed algorithms using SVD and pure SVD. In the study results, both watermarking techniques have invisibility and robustness of watermarked images. However, in the PSNR evaluation, the proposed algorithm using SVD is better than pure SVD. In addition, the proposed algorithm has a better correlation coefficient tested in several attacks. The SVD (Singular Value Decomposition) was used in digital watermarking schemes such as rotation in multiple integers of 90 degrees and flipping of images.

According to the study of Makbol et al. (2020), a Novel reliable SVD-based image watermarking scheme in the IWT domain was used. Using the values of UWA of a grayscale image, it is embedded into the singular values SLL of the LL sub band of the host image. The proposed scheme uses SWA and VWA as secret keys to avoid FPP. The proposed blind watermarking improves the security issues ensuring the reliability and security requirements of watermarking techniques.

In the study of Hisham et al., (2019), a proposed watermarking scheme uses a grayscale image designed to be a good numbering pattern which results in accurate detection and image recovery. Its purpose is to verify the integrity and authenticity of medical images stored in HIS (Hospital Information Systems). Using the watermarking scheme, any modified images can be detected or exposed and ensured that the image is authentic, making the image usable by the hospital professionals every time.

In the study of Liu et al., (2019), telemedicine images were used from telemedicine applications to improve authenticity and integrity, ensuring the medical

images are not forged by attackers and belong to the correct medical institutes or patients using the proposed novel robust reversible watermarking scheme based on SLT-SVD transform. The results of the proposed watermarking scheme have remarkable performance in terms of robustness, imperceptibility, authentication, tamper detection, tamper localization, and tamper recovery.

In the study of the study conducted by Zhang et al. (2018), with the help of Arnold Transform the proposed scheme has good robustness against attacks like salt and pepper noise, crop attacks, and other attacks, this can improve the security of watermark embedding that also ensures the confidentiality of copyright information.

Hu & Chen (2019) used SVD-based self-embedding watermarking for image authentication. The watermark generation is fragile to content modification and robust to standard image processing. The received image authentication needs no information about the original image or watermark, increasing watermark security and preventing forged watermarks. Therefore, the proposed watermarking method is practicable for image authentication and adapts well to multimedia communication in lossy channels.

According to the study, He et al. (2018), claimed that a wavelet-based fragile watermarking technique was developed for safe picture authentication. The approach generates an embedded watermark using the Discrete Wavelet Transform (DWT) and then embeds the enhanced security watermark jumbled by a chaotic system into the picture data's LSB. As a result, the watermarking system's security is greatly improved. The suggested method's tamper localization and discrimination capabilities have been shown through simulations. The next step is to create a secure fragile watermarking method that includes tamper detection and recovery.

Furthermore, the research conducted by Agarwel et al. (2018) shows that when watermarking is used with DWT transform, the fraction value of the wavelet coefficient is lost during the process of embedding, resulting in a degraded visual quality of the produced watermarked image. This study made a combined watermarking approach scheme based on IWT to boost the imperceptibility of invisible watermarking and reduce the fraction loss effect of wavelet transform compared to DWT+DCT-based watermarking. The results indicated that the IWT-based approach significantly improved the visual quality and robustness of the watermarked images. Moreover, this technique demonstrated a higher resilience against common image processing attacks, making it a superior choice for practical applications.

According to Robert (2019), watermarking is a branch of information hiding that is used to hide proprietary information in digital media like photographs, digital music, or digital video. This technique ensures that the embedded information remains concealed yet recoverable, providing a means for copyright protection and content authentication. Additionally, watermarking can be used to trace the source of unauthorized copies and distribution channels, acting as a deterrent to potential infringers. As digital media continues to evolve, the methods for embedding and detecting watermarks are also becoming more sophisticated, ensuring robust protection for digital assets.

Cited by Laksman (2021), watermarking is about putting a piece of data inside digital data to show and prove its authenticity. By concealing signals protecting a digital medium from copyright infringement, the original image serves as the image on which the authentication message is encoded. The cover image becomes a watermarked image after the watermark is implanted. The ease with which digital content can be exchanged over the Internet has created copyright infringement issues. Copyrighted

material can be easily exchanged over peer-to-peer networks, and this has caused major concerns to those content providers who produce this digital content. Thus, the need for security of media such as images, text, audio, video, and other media types has grown significantly. In our internet community, digital watermarking has become quite important. Digital watermarking techniques have been used to prevent the unauthorized distribution of digital media in bitmaps, audio, video, or other media. With the proliferation of high-speed internet and digital technologies, unauthorized copying and distribution have become even easier and more widespread. Advanced watermarking algorithms are continually being developed to counteract more sophisticated methods of digital piracy. Consequently, digital watermarking is now an essential tool in the protection of intellectual property rights in the digital age.

Synthesis

The related literature and studies discussed and mentioned presented the conceptualization and theories of watermarking. These discussions spanned from the history and early years of watermarking to its possible future directions, illustrating the evolution and significance of this technology. It was highlighted that watermarking has prerequisites to satisfy several requirements to ensure the robustness of the watermarked images, especially in today's digital era where image manipulation tools like Adobe Photoshop, GIMP, Corel PaintShop Pro, and others are readily available to the market. Moreover, researchers emphasized the importance of utilizing transform domains such as Discrete Wavelet Transform (DWT), Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Singular Value Decomposition (SVD) over the spatial domain due to their ability to enhance security against attacks on watermarked images. These mathematical techniques play a crucial role in making

invisible watermarks undecipherable or clear to verify their authenticity, thus strengthening copyright protection and establishing ownership of digital images.. Most research was conducted using PSNR (peek-signal-noise-ratio), SSIM (Structural Similarity Index Measure), and NCC (Normalized Cross-Correlation) to measure the quality and the distinguishable and indistinguishable of the image. These metrics not only help assess the effectiveness of proposed watermarking techniques but also contribute to ensuring the authenticity and protection of images against copyright infringement. The proposed watermarking techniques in the study assist in ensuring the image’s authenticity and protection of watermarked images against copyright attacks.

Conceptual Framework

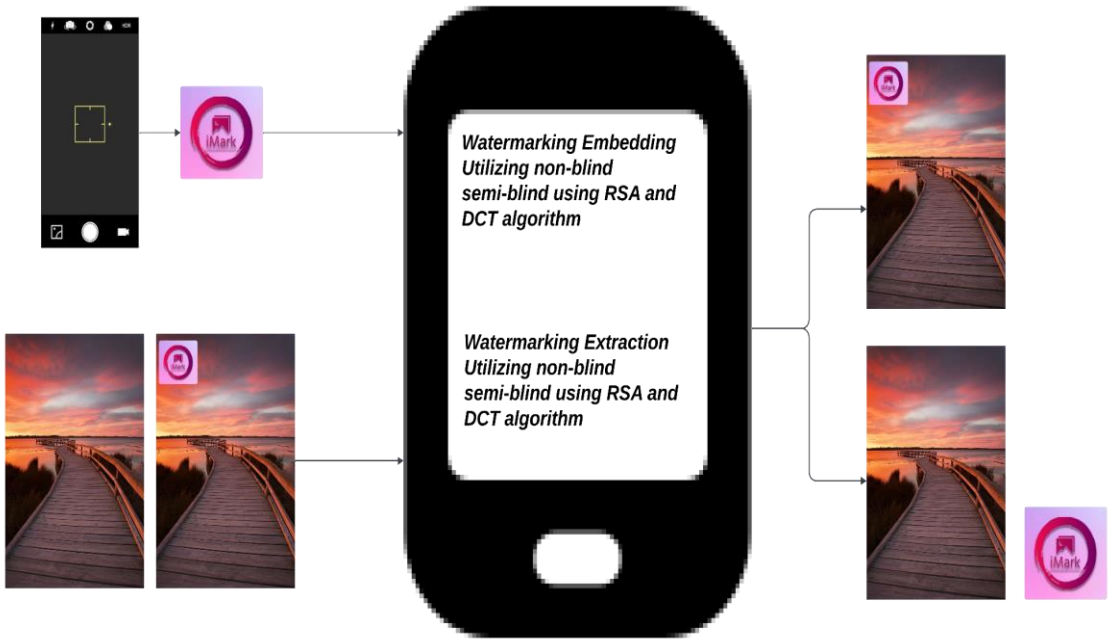


Figure 1. Conceptual Diagram

Figure 1 shows the visual representation of the study's conceptual diagram. Any smaller digital image will be used as the invisible watermark to the captured cover image and utilizes the smartphone's camera to capture the image used as the cover image. Using RSA and DCT in Non-Blind-Semi-Blind watermarking algorithms are being utilized in a mobile application for embedding. The utilization of the camera of the Android phone to live capture the cover image for watermarking limits the number of images the user can embed per cover image to only 1, since when the captured image already has an invisible watermark embedded in it, users will not be able to embed to the same image or use the already watermarked image as input. Using the above approach would require the original and the watermarked images and keys embedded for the process of extracting the watermarked image to the cover image.

Definition of terms

Copyright. Copyright is a legal term used to describe creators' rights over their literary and artistic works. This means that the original creators of products and anyone they authorize are the only ones with the exclusive right to reproduce the work.

Cover Image. The cover image is the original image that is used to embed a watermark without significantly altering the visual appearance of the image.

Digital Invisible watermarking. Refers to the process of embedding information, often imperceptible to human senses, into a piece of digital media to prove ownership.

Discrete Fourier Transform (DFT). A mathematical technique used in signal processing and other fields to analyze the frequency content of a discrete signal. It helps break down a signal into its frequency components.

Fragile watermarking. Fragile watermarking is intentionally made to be sensitive to any alterations in the content, and its main purpose is to provide a means of verifying the integrity of the data.

Filter Attack. Refers to attacks involving the intentional manipulation of pixels in a digital image. Filters or image processing techniques are applied to change the visual appearance of the image by altering pixel values.

Geometrical Attack. Refers to various manipulations or alterations to the digital content directly. These attacks aim to disrupt or remove watermarks embedded in the content.

Intellectual property rights. Refers to the legal protections granted to the authors, creators, and owners of intellectual property, which includes inventions, and literary and artistic works.

Live Captured Images. Refers to the used and taken by the camera of the android application in watermarking processes.

Peak-signal-to-Noise-Ratio (PSNR). Is a metric used to measure the quality of a reconstructed or compressed image or video compared to its original, uncompressed version.

Photographer. Refers to a person who takes photographs as a job or professionally.

Robust watermarking. Refers to a type of digital watermarking technique designed to withstand various intentional or unintentional attacks, distortions, or transformations applied to the watermarked content.

Visible Watermarking. Involves the addition of visible marks or logos directly onto digital media to identify the ownership, copyright information, or other details related to the content.

Watermarking. is a technique used to embed information, known as a "watermark," atop digital media (such as images, audio, video, or documents) in a way that is imperceptible or difficult to remove and protect copyright.

Watermark Attack. Deliberate or intentional action taken to undermine, remove, or alter a digital watermark that has been embedded in digital media.

Watermarking Scheme. A proposed watermarking method was presented and evaluated.

CHAPTER III

METHODOLOGY

This chapter outlines the comprehensive methodology employed for the development, implementation, and evaluation of the mobile application. The application seamlessly integrates watermarking and encryption algorithms to enhance image security and privacy. The to be used in this study as shown in the Figure 2, including user interface design, the implementation of RSA and DCT algorithms, encryption and decryption processes, watermark embedding, watermark extraction, software requirements, evaluation, and ethical considerations.

Research Methodology

Figure 2 illustrates the waterfall model methodology followed for developing the invisible watermarking system for live captured photos. This structured approach emphasizes a non-overlapping progression, where each phase is completed before moving to the next. The development process starts with data gathering, where relevant information and resources are collected to inform the system's design and implementation. This is followed by the creation phase, where the actual watermarking system is designed and implemented based on the gathered data. Finally, the system undergoes rigorous evaluation to assess its effectiveness in embedding and extracting watermarks, as well as its robustness against potential attacks. This linear progression ensures a systematic and thorough development process, minimizing the risk of errors and ensuring each component is fully developed and tested before proceeding.

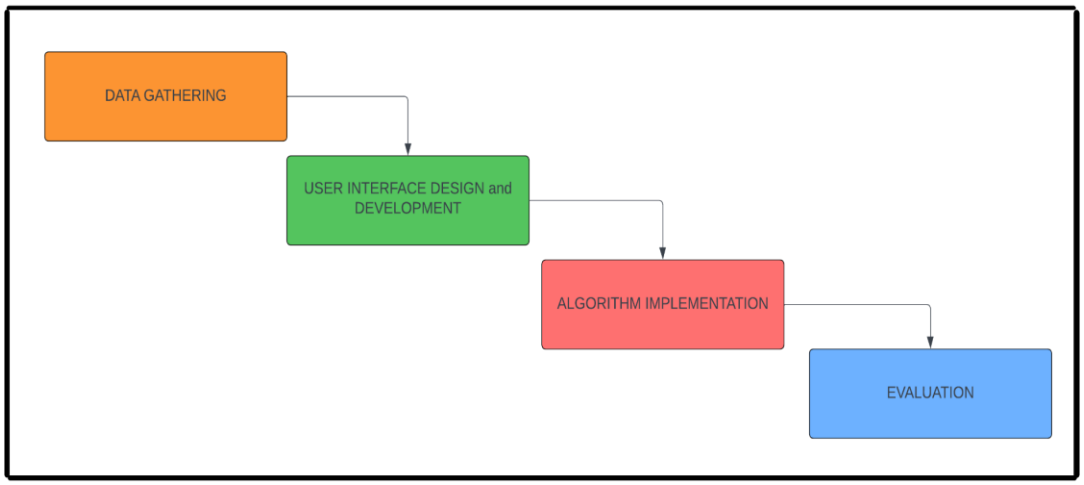


Figure 2. Waterfall Model Methodology of Invisible Watermarking

Phase 1 of this methodology is Data Gathering. This phase focused on creating a unique watermark specifically designed for the application. Table 1 details the customized images chosen for this purpose. The dimensions of these watermark images were intentionally kept smaller than the cover images (also shown in Table 1) to ensure they would be seamlessly embedded without disrupting the main content. Furthermore, the watermark design considered factors like imperceptibility and robustness. The chosen images were carefully selected to minimize their visual impact on the original photos while maintaining their ability to withstand various image processing operations and potential attacks. This balance between invisibility and resilience is crucial for the effectiveness of the watermarking system. Additionally, extensive testing was conducted to ensure that the watermarks remained undetectable to the naked eye while still being reliably extractable under various conditions. Feedback from these tests was used to make necessary adjustments, further optimizing the watermark design for practical application and usage

Table 1

Customized watermark images

IMAGE	FILE FORMAT	DIMENSION
	**JPG**	202 X 186

Following the development of the watermarking system, Phase 2 focused on designing and developing the user interface (UI) for the mobile application. This UI was meticulously crafted to provide a seamless and intuitive experience for users interacting with the watermarking functionalities. The application was designed to facilitate embedding, detecting, and extracting invisible watermark images within captured photos. The primary goal of the UI was to enhance user interaction with the application during the evaluation process. By providing clear and user-friendly interfaces for input and output, the UI aimed to streamline data collection and analysis. The development of the application utilized Android Studio IDE, Kotlin for the user interface, and XML for handling user input and displaying system output. Additionally, Gradle was employed to manage smooth fragment-to-fragment interactions within the application. Figure 3 illustrates the system architecture of the developed invisible watermarking mobile application. Rigorous user testing was conducted to ensure that the UI met usability standards and provided a positive user experience. Continuous feedback loops were established to iteratively refine the UI based on user experiences and testing outcomes.

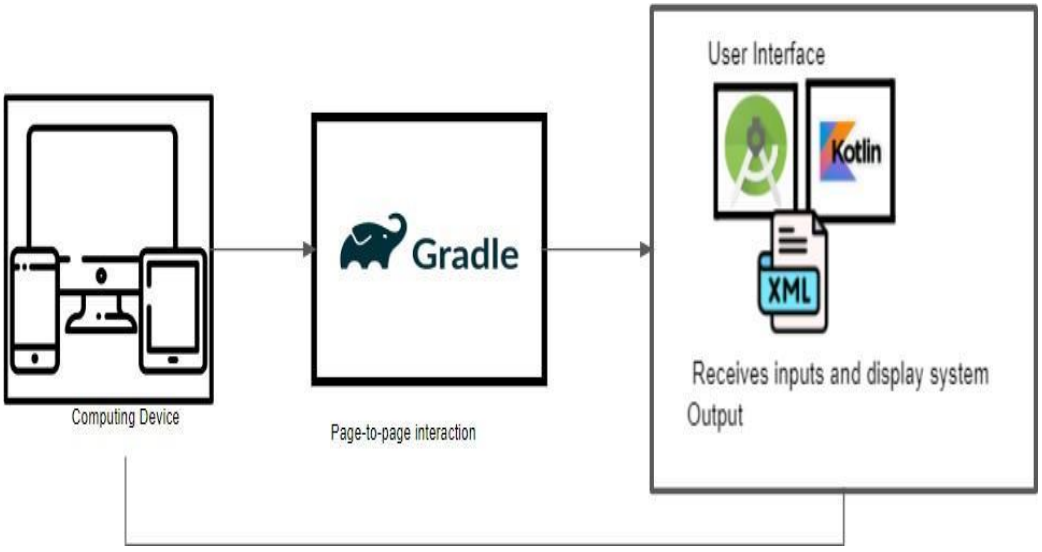


Figure 3. Invisible Watermarking Mobile Application Architecture

The Xiaomi Redmi 10C Android smartphone was chosen for installing and implementing the application. This choice offered several advantages. Firstly, it eliminated the need for additional hardware, such as a computer or laptop, saving time and resources and making research more efficient and cost-effective. Additionally, smartphones are highly portable, allowing researchers to carry them around easily and use them in various settings. This flexibility and convenience further enhanced the research process. Furthermore, the Redmi 10C boasts a capable processor and sufficient RAM, ensuring smooth operation of the application. The device also features a user-friendly interface and a long-lasting battery, contributing to a positive user experience during testing and data collection. Table 2 details the features and specifications of the Android smartphone used for implementation.

Table 2

Android smartphone specification for implementation

Xiaomi redmi 10C	
Dimension	169.6 x 76.6 x 8.3 mm (6.68 x 3.02 x 0.33 in)
Display	6.71 inches, 106.5 cm^2 (~82.0% screen-to-body ratio)
Resolution	720 x 1650 pixels (~268 ppi density)
OS	Android 11, MIUI 13
RAM	64GB 3GB RAM
GPU	Adreno 610
CPU	Octa-core (4x2.4 GHz Kryo 265 Gold & 4x1.9 GHz Kryo 265 Silver)
Chipset	Qualcomm SM6225 Snapdragon 680 4G (6 nm)
Storage	64 GB
Camera	5 mp, f/2.2
Battery	Li-Po 5000 mAh, non-removable

Phase 3 is the utilization of the Non-blind-semi-blind watermarking technique, using RSA and DCT algorithms during the algorithm implementation phase for embedding and extracting invisible watermarks after the designing of the user interface. This phase ensures robust security and high fidelity in the watermarking process, maintaining the integrity and quality of the original content.

The Discrete Cosine Transform algorithm or (DCT), serves as a pivotal component in the watermarking process, laying the foundation for discreetly embedding an invisible watermark into captured images. This detailed explanation outlines the intricacies of the DCT algorithm, emphasizing its role in transforming images into the frequency domain, selectively modifying coefficients for discrete watermark embedding, and seamlessly applying the inverse DCT to produce the final watermarked image. The DCT helps separate the image into parts (or spectral sub-bands) of differing importance (concerning the image's visual quality). It transforms a signal from the spatial domain to the frequency domain and enables the embedding of an invisible watermark in the cover image. It gives good robustness against attacks like filtering and image compressions.

Image Block Partitioning

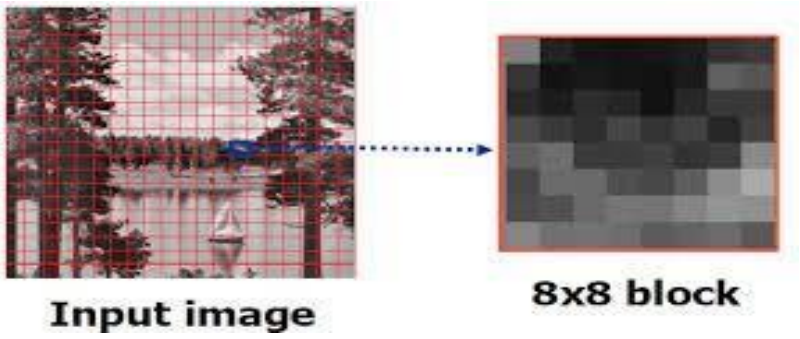


Figure 4: Sample of Image partition (left) and DCT blocks(right)

Divide the image into non-overlapping blocks of fixed size. The DCT algorithm operates on small image blocks to efficiently capture frequency information. The choice of block size is a trade-off between frequency resolution Figure 4 shows the illustration of image partitioning in an image (left) and the DCT fixed non- overlapping blocks(right).

The use of DCT was to embed the watermark invisibly without affecting the original image, as stated as one of the recommended algorithms for implementing invisible watermarks in Chapter 2. The formula of 2D-DCT was represented by Equation 1.

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N} \quad (1)$$

Apply 2D DCT to Each Block

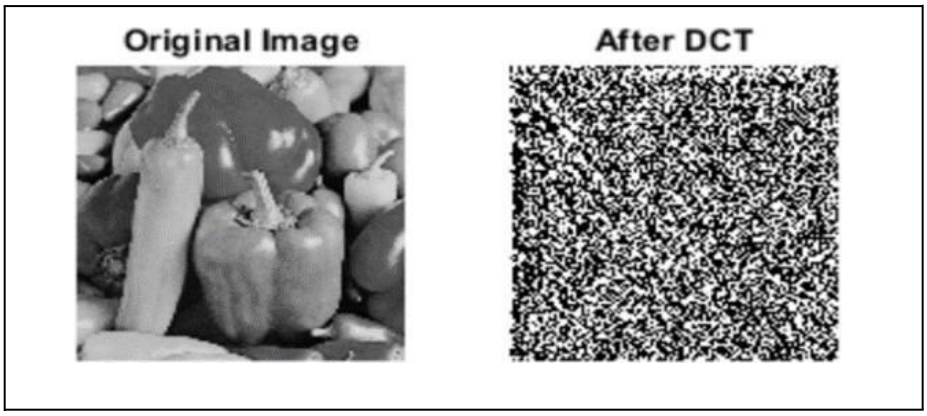


Figure 5. Results of original image (left) and after DCT (right)

After partitioning the image into blocks, apply a 2-dimensional DCT to each block independently. The 2D DCT transforms each block from the spatial domain to the frequency domain, representing the image in terms of its frequency components. The resulting coefficients reflect the contributions of different frequency components in each block. These coefficients are crucial for various image processing tasks, including compression and denoising. Figure 5 shows the before and after the application of DCT to an image.

Figure 6. Results of DCT Transformed image and after IDCT

DCT had an inverse function to restore transformed 2D-DCT images into their original form. The inverse Discrete Cosine Transform (IDCT) is crucial in signal processing and image compression, as it allows for the reconstruction of images from their transformed representations, aiding in data recovery and fidelity restoration. Figure 6 shows the before and after the application of IDCT to a transformed image. The formula for 2D - Inverse Discrete Cosine Transform is represented in Equation 2. This process ensures that the compressed image maintains a high level of quality and detail upon decompression.

$$f(x,y) = \frac{2}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} C(m)C(n)F(m,n)\cos\frac{(2x+1)m\pi}{2M}\cos\frac{(2y+1)n\pi}{2N} \quad (2)$$

RSA Algorithm Implementation

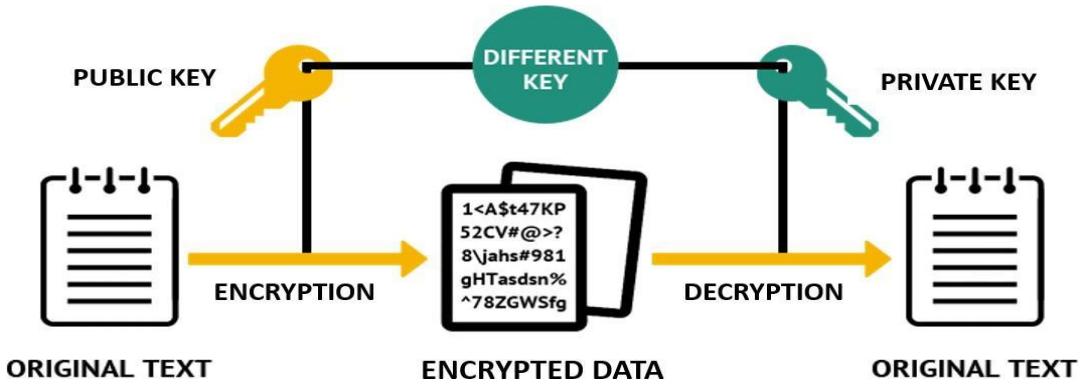


Figure 7. RSA Algorithm Encryption and Decryption Process

RSA is named after Rivest, Shamir, and Adleman, the three creators of the RSA calculation. It is an asymmetric cryptographic algorithm that contains two keys: public key and private key. A public key is used to encrypt the date at the transmitter, and for decryption, both keys are used at the receiver. The Primary rationale behind the RSA algorithm is that it generates these two keys by factoring a given large integer. Since a public key has two numbers, one of them is a product of two prime numbers. RSA keys are commonly 1024 or 2048 bits length. If the key's size increases, the encryption's strength increases exponentially. Using the RSA Algorithm to create a ciphertext based on the date and time processing. In addition, it is used as a seed to shuffle image pixels of the watermark image resulting in an unrecognizable image adding minimal security before it is embedded in the captured image. This would prevent attackers from stealing the watermark image or hidden information inside the images. Figure 7 shows RSA's Encryption and Decryption Process. The more comprehensive step by step process of generating encryption and decryption key are explained below:

To generate an encryption and decryption key pair, one begins by selecting two large 10-digit prime numbers, p and q. These primes are fundamental to the RSA algorithm's security. Next, n is calculated as the product of p and q, forming part of both the public and private keys. This value, n, is essential for the encryption and decryption processes in the RSA algorithm.

To generate an encryption and decryption key pair, the process begins with the selection of two large 10-digit prime numbers, denoted as p and q. These prime numbers are fundamental to the security of the RSA algorithm. Subsequently, the product n is calculated by multiplying p and q. This value, n, becomes a component of both the public and private keys. The next step involves computing the totient function $\phi(n)$, defined as $\phi(n) = (p - 1) \times (q - 1)$. This function is crucial for the key generation process. A public exponent e is then chosen, ensuring that $1 < e < \phi(n)$ and e is coprime with $\phi(n)$. Typically, e is set to 65537 due to its efficiency and security benefits. Following the determination of e, the modular multiplicative inverse of e modulo $\phi(n)$ is calculated, yielding d. This ensures that $d \times e \equiv 1 \pmod{\phi(n)}$. The value d forms the private key, enabling the decryption of messages encrypted with the public key. The public key consists of the pair (e, n), while the private key is represented by (d, n). These keys are then compiled into a format suitable for secure storage and retrieval, ensuring the integrity of the encryption and decryption process. Proper management and protection of these keys are vital to maintaining the overall security of the RSA encryption system. Regular updates and secure backups of the keys are recommended to prevent unauthorized access and potential data loss. Additionally, implementing multi-factor authentication (MFA) can further enhance the security of key access and management processes.

RSA Algorithm’s advantages are as follows:

Users can use the same encryption key for encrypting multiple images for watermarking and decrypting the watermarked images with the same decryption key.

The user has two different keys (encryption and decryption key), The encryption key known as the Public Key can be shared with anyone for watermark embedding, and these watermarked images can still be decrypted to retrieve the watermark image without disclosing the decryption key to the public.

RSA uses a pair of keys – a public key for encryption and a private key for decryption. This eliminates the need for securely sharing a secret key between the communicating parties, which is a major limitation of symmetric key algorithms. The public key can be freely distributed without compromising security, while the private key is kept confidential by the owner. This asymmetric nature of RSA not only enhances security but also simplifies key management in secure communications.

In the embedding process, the required inputs are the cover image, watermark image, and RSA-generated encryption key. Accepted image formats are JPEG/JPG or PNG. If an image embedding has been done before, the application reuses the same encryption key unless the watermark changes, in which case a new key is generated. RSA encrypts the current date and time, producing a ciphertext seed. This seed ensures the uniqueness and security of the embedding process, preventing unauthorized duplication. The encrypted seed is then used to embed the watermark into the cover image, maintaining the integrity and confidentiality of the embedded information this seed shuffles the watermark image pixels before embedding them into the cover image.

Figure 8 illustrates the complete watermark embedding process with all necessary inputs and steps.

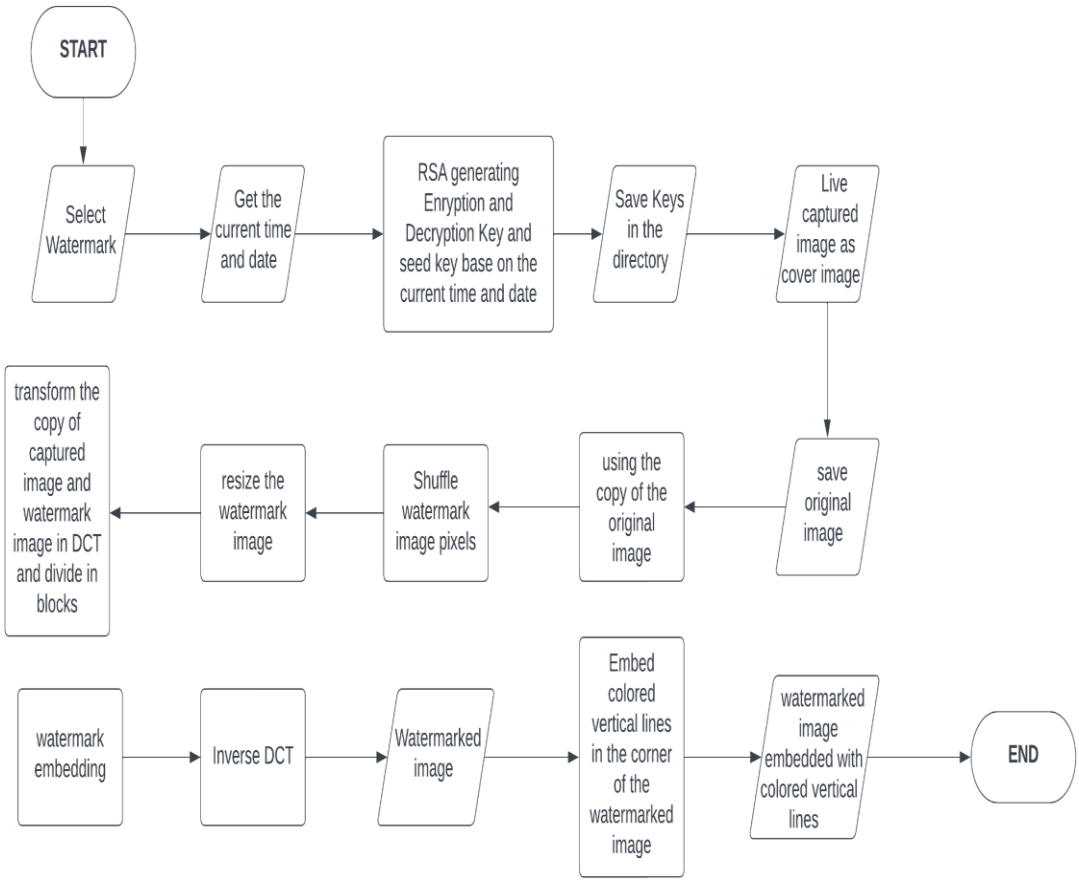


Figure 8. Flowchart of Watermark Embedding

The output after watermarking is the watermarked image with vertical lines, original image, decryption key, and seed key. The seed key is the ciphertext message of the seed used to shuffle the watermark before embedding it to the cover image. This ensures that the watermark is securely and uniquely embedded into the cover image. Additionally, the decryption key is essential for extracting the watermark from the watermarked image, ensuring that only authorized users can retrieve the hidden information. The process of watermark embedding is explained in detail below:

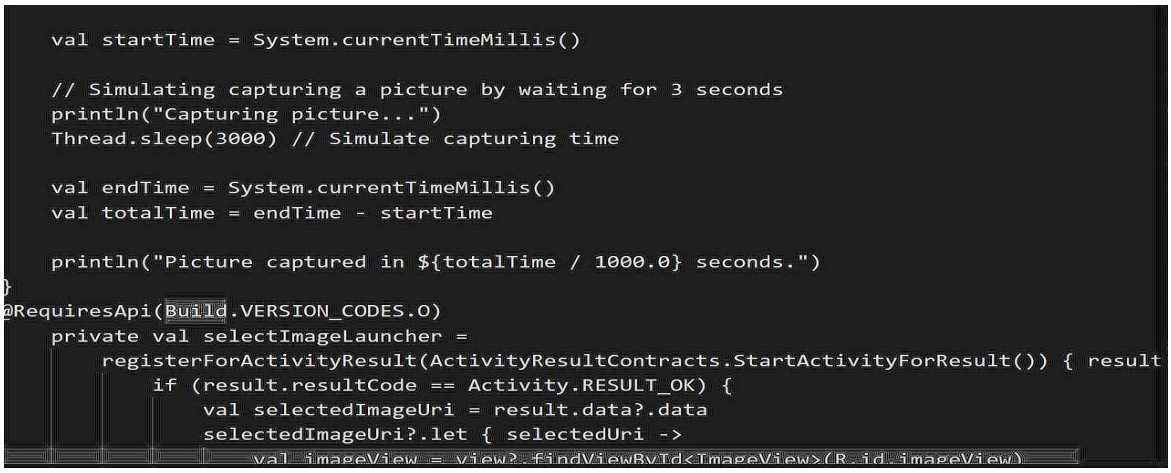
The watermark embedding process begins with selecting a watermark image if not already acquired. The current time and date are used as a seed for shuffling watermark pixels, enhancing security and integrity. The system generates encryption keys and encrypts the time and date, saving the ciphertext in storage. The application captures the cover image with the camera and saves a duplicate. The shuffled watermark pixels are resized to 512x512 for compatibility with the cover image. Both images are split into RGB channels and undergo Discrete Cosine Transform (DCT) into 32x32 pixel blocks, preserving visual quality while embedding the watermark. The R, G, and B channels of the DCT-transformed watermark are embedded into the corresponding channels of the cover image. After merging the RGB channels back into one image, the Inverse DCT is applied. Finally, color vertical lines are added to each corner of the image.

These colored vertical lines are embedded in the four sides of the image containing the secret message, namely; “KARLO BALUCIO” - upper left corner, “CHRISTIAN JAMES” - lower left corner , “DARLENE MALASA” - upper right corner, “THESIS 2”- lower left corner and 908941321 as a key. Below explain the process of embedding the messages resulting in a colored vertical line:

The process starts by creating an ASCII list named "dictionary" that serves as a container for the ASCII values of each letter in the secret message. This dictionary allows easy conversion of each character in the secret message into its corresponding ASCII value. For embedding the secret message into the image, each character's ASCII value is pulled from the dictionary.

An XOR operation is then performed between this ASCII value and the ASCII value of a secret key. This operation produces an encoded value that ensures added security by making the message harder to decode without the secret key. The resulting encoded values are then embedded into the green channel of the image, which is chosen for its balance between luminance sensitivity and minimal perceptual impact, maintaining the visual quality of the image while securely embedding the secret message. To ensure the robustness of the embedded data, redundancy techniques may be applied, embedding the encoded values multiple times across different parts of the green channel. This approach further enhances the security and reliability of the watermark, making it resistant to common image processing attacks.

The live capture embedding process allows for the successive watermarking of images without any delay and waiting. Furthermore, the runtime allows for the optimization of user experience, proving that the system is fast and can perform the necessary processing functions. This also can be used as a comparison and reference for future researchers to compare the performance of the embedding process in other studies in terms of running time. The efficient runtime ensures that the system is suitable for real-time applications, where immediate processing and feedback are critical. Figure 9 shows a code snippet for the function of the timer for runtime measuring, highlighting the system's capability to track and report processing times accurately. Additionally, this functionality provides valuable data for continuous performance improvements and benchmarking against other methods. Overall, the live capture feature enhances the system's usability and relevance in practical, time-sensitive scenarios.



```
    val startTime = System.currentTimeMillis()

    // Simulating capturing a picture by waiting for 3 seconds
    println("Capturing picture...")
    Thread.sleep(3000) // Simulate capturing time

    val endTime = System.currentTimeMillis()
    val totalTime = endTime - startTime

    println("Picture captured in ${totalTime / 1000.0} seconds.")
}
@RequiresApi(Build.VERSION_CODES.O)
    private val selectImageLauncher =
        registerForActivityResult(ActivityResultContracts.StartActivityForResult()) { result
            if (result.resultCode == Activity.RESULT_OK) {
                val selectedImageUri = result.data?.data
                selectedImageUri?.let { selectedUri ->
                    val imageView = view?.findViewById<ImageView>(R.id.imageView)
```

Figure 9. Code Snippet of Timer for Measuring the Running Time of Embedding

The watermark detection feature of the mobile application is its ability to detect the presence of an invisible watermark embedded in a watermarked image. This detection can occur even without using the watermarking app. Upon zooming the image to 550%, colored vertical lines become visible in each corner of the watermarked image. The length of these colored vertical lines corresponds to the number of characters in the secret code embedded within them, providing a visual indication of the embedded watermark's presence and the extent of the encoded information. Figure 10 illustrates an image with vertical lines in its corners when zoomed in to 550%. This visual marker enables the detection of the watermark without the need for the watermarking application, providing a straightforward method to verify the existence of the watermark. This feature enhances the utility of the watermarking system by making verification accessible and user-friendly. Furthermore, it ensures that the embedded information can be quickly and easily identified, reinforcing the system's practicality for various real-world applications.

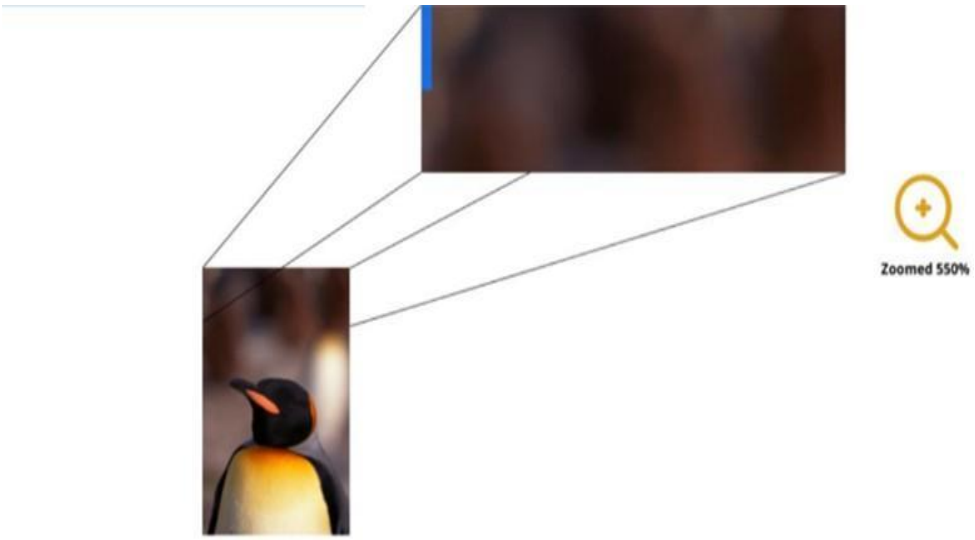


Figure 10. Watermarked Image with a Color Vertical Line Zoomed to 550%

The extracted secret message in the image is used to compare if it matches the original secret messages to verify that the watermark is already embedded by using the same key in the embedding process. The process of retrieving the message inside the colored vertical line is explained below:

To extract the embedded secret message from the watermarked image, an array named "dictionary" is first created to store the ASCII values of characters. The process begins by locating the pixel coordinates that contain the colored vertical lines, which were embedded during the watermarking process. Each pixel value at these coordinates is then retrieved, and its corresponding ASCII value is obtained using the "dictionary" array.

Next, an XOR operation is performed between the ASCII value from the pixel and the ASCII value of the secret key to decode each character of the message. This operation is repeated for the length of the embedded message, effectively reconstructing the original secret message.

Finally, the extracted message is compared with the original embedded message, and a prompt is displayed to indicate whether they match, ensuring the accuracy and integrity of the watermark extraction process. Figure 11 shows the code snippet for detecting a watermark in an image.

```
class DetectFragment : Fragment() {

    @RequiresApi(Build.VERSION_CODES.O)
    private val selectImageLauncher =
        registerForActivityResult(ActivityResultContracts.StartActivityForResult()
            if (result.resultCode == Activity.RESULT_OK) {
                val selectedImageUri = result.data?.data
                selectedImageUri?.let { selectedUri ->
                    val imageView = view?.findViewById<ImageView>(R.id.imageView)
                    imageView?.setImageURI(selectedUri)

                    val decodedBitmap = loadScaledBitmap(selectedUri, MAX_WIDTH, MA

                    decodedBitmap?.let { bitmap ->
                        val watermarkedImage = convertBitmapToWatermarkedImage(bitm
                        val detected = detectColoredVerticalLine(watermarkedImage)
                        if (detected) {
                            showToast("Watermark Detected")
                        } else {
                            showToast("Watermark Not Detected")
                        }
                    }
                }
            }
        }
```

Figure 11. Code Snippet for Watermarking Detection

In the extraction process, only the individual in possession of the decryption key, seed key, and the original image can successfully extract and view the invisible watermark. This stringent requirement ensures that unauthorized users cannot retrieve the hidden watermark, thereby securing the integrity and confidentiality of the embedded information. The extraction method is meticulously designed, involving a detailed and intricate procedure that guarantees the accurate and seamless retrieval of the watermark image from its watermarked counterpart. Figure 12 illustrates the comprehensive steps involved in the watermark embedding process, highlighting each critical phase to emphasize the robustness and precision of the entire operation in the flowchart, each process is presented, in the flowchart, each process is presented,

with inputs that are needed for processing.

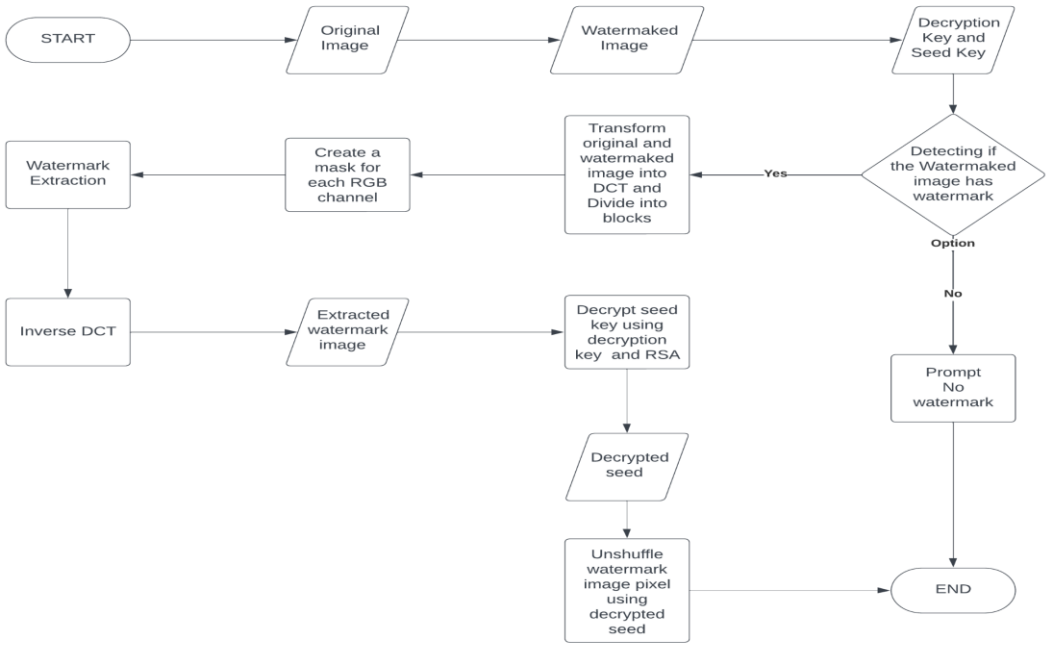


Figure 12. Flowchart of Extraction Process

The process of extracting the embedded watermark from within the cover image is explained below. This process involves several critical steps, each designed to ensure the accurate and secure retrieval of the watermark. Initially, the cover image is subjected to a series of operations using the decryption key, seed key, and the original image.

The process of extracting the embedded watermark from the cover image begins with inputting both the original and watermarked images, along with the decryption and seed keys. The system then aligns and compares the two images to locate discrepancies introduced by the watermarking process. Once these discrepancies are identified, the decryption key and seed key are used to reconstruct the original watermark accurately.

The system checks if the watermark is already present; if so, the process

terminates. If not, both images are decomposed into their RGB channels, and the Discrete Cosine Transform (DCT) is applied to convert each channel into 32x32 pixel blocks. This step isolates the embedded watermark in the frequency domain, enhancing extraction precision.

Next, a mask is generated for each RGB channel to store the extracted watermark data. By comparing the DCT-transformed blocks of the watermarked image with those of the original, the differences corresponding to the embedded watermark are identified and stored in the RGB masks. These masks are then merged into a single composite mask. An inverse DCT is applied to revert the DCT-transformed watermark back to its spatial domain in a shuffled pixel form. Finally, the seed key is decrypted using the decryption key to unshuffled the watermark pixels based on the current time and date, restoring the watermark to its original form. This thorough process ensures the accurate and secure retrieval of the embedded watermark.

Table 3 lists the software requirements for developing mobile applications. The development environment uses Kotlin, the preferred language for Android development, ensuring compatibility and optimized performance. Key tools include Android Studio, a robust IDE, and Visual Studio Code for flexible code editing. Security is ensured with cryptographic libraries for RSA and image processing libraries for DCT. For local data management, the URI Library is used for storing and accessing images. These tools collectively form the system requirements for building a secure and efficient Android application.

Table 3

System Requirements for Development

Tools/ Programming Language	Description
Kotlin	The application was developed using Kotlin for Android development, ensuring compatibility and performance optimization.
Android Studio Visual Studio Code	Robust and widely used IDE for Android application development, providing a comprehensive set of tools and features.
RSA and DCT Libraries	Integration of cryptographic libraries for RSA and image processing libraries for DCT, ensuring the efficient implementation of algorithms
URI Library	A library in Kotlin and Android Studio has been developed specifically for image storage, facilitating convenient access to images whenever they are needed.
Extensible Markup Language (XML)	XML played a pivotal role in defining the layout and structure of the user interface components, showcasing its extensive utilization and significance within the development process.

In addition to assessing imperceptibility and robustness, Phase 4 also included an evaluation of the watermarking technique's capacity to withstand common signal processing operations such as filtering, color manipulation, and contrast adjustments, further ensuring its viability in diverse digital environments. This evaluation ensures that the watermarking technique remains effective and reliable under various image manipulation scenarios, crucial for maintaining the integrity and persistence of embedded watermarks in real-world applications.

The invisible watermarking algorithm would determine the robustness and imperceptibility by applying several attacks on the watermarked image. By applying these diverse attacks, the algorithm's resilience to various threats and its ability to maintain the integrity and invisibility of the watermark are thoroughly assessed, ensuring its effectiveness in real-world scenarios. The SSIM and PSNR (Peak Signal-to-Noise Ratio) measuring tools are used to determine the results from the original and extracted watermark images.

The following tests are conducted for evaluation as listed in Table 4. Test case 1 consists of test cases with no attacks involved in the watermarked image. Test Cases 2-5 (TC 001-005) are test cases for attacking the watermarked image with geometric attacks. Test case 6 (TC006) is a test case for attacking the watermark image with a filter attack and getting the scores received in PSNR and SSIM. Furthermore, the results obtained from these tests are analyzed to gauge the watermarking scheme’s performance under different attack scenarios providing an understanding of its effectiveness. These comprehensive evaluations ensure that the watermarking technique is robust and reliable, capable of protecting digital content in a variety of conditions.

Table 4
Test Cases for Watermarked Image

TEST CASE	TEST SCENARIO	TEST STEP	TEST DATA	EXPECTED RESULTS
TC001	Test, if the watermark is retrieved with no attack, applied to the watermarked image	1.Feed the watermarked image for extraction. 2. Feed the original and watermark image to Evaluation tools.	Watermarked Image Original Image Secret Key 1 Secret Key 2	1. Display the actual hidden watermark PNSR and SSIM evaluation score
TC002	Check if the watermark is retrieved after by applying rotation attack	1. Applying rotation to the watermarked image 2. Feed the attacked image for extraction 3. Feed the original and watermark image to Evaluation tools.	Watermarked Image Original Image Secret Key 1 Secret Key 2	1. Display the actual hidden watermark 2.PSNR and SSIM evaluation score.
TC003	Check if the watermark is retrieved after applying ripple distortion	1. Applying ripple distortion to the watermarked image 2. Feed the attacked image for extraction 3. Feed the original and watermark to evaluation tools.	Watermarked Image Original Image Secret Key 1 Secret Key 2	1. Display the actual hidden watermark 2. PNSR and SSIM evaluation score
TC004	Check if the watermark is retrieved after	1. Applying noise to watermarked images.	Watermarked Image Original Image	1. Display the actual hidden watermark

	applying flip attack to the watermarked image.	2. Feed the attacked image for extraction. 3. Feed the original and watermark to evaluation tools.	Secret Key 1 Secret Key 2	2.PNSR and SSIM evaluation score
TC005	Check if the watermark is retrieved after applying cropping attack	1. Applying image cropping to the watermarked image 2. Feed the image for extraction Watermarked Image 3. Feed the original and watermark to evaluation tools.	Watermarked Image Original Image Secret Key 1 Secret Key 2	1. Display the actual hidden watermark 2.PNSR and SSIM evaluation score
TC006	Applying Different kinds of filter attack to check the robustness of the algorithm	1.apply filter attacks (gaussian blur, noise) to the image. 2.feed in the extraction page	Watermarked Image Original Image Secret Key 1 Secret Key 2	1. Display the actual hidden watermark

Geometric attacks include rotation, flip, ripple distortion, and cropping. Filter attacks encompass noise addition, Gaussian noise, and Gaussian blur for affecting blurring, contrast, and sharpness. These methods are commonly used in image processing to test the robustness of various algorithms.

Image Quality Measuring Metrics

```
# and is represented as a floating point data type in the range [0,1]
# so we must convert the array to 8-bit unsigned integers in the range
# [0,255] before we can use it with OpenCV
diff = (diff * 255).astype("uint8")

# Threshold the difference image, followed by finding contours to
# obtain the regions of the two input images that differ
thresh = cv2.threshold(diff, 0, 255, cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]
contours = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
contours = contours[0] if len(contours) == 2 else contours[1]

mask = np.zeros(before.shape, dtype='uint8')
filled_after = after.copy()

for c in contours:
```

Figure 13. Python Code Snippet For SSIM Measuring Metrics

The Structural Similarity Index (SSIM) compares two images by evaluating their luminance, contrast, and structure to determine similarity. It first examines luminance by calculating the average brightness, then assesses contrast by checking intensity variations, and finally analyzes the structural patterns of pixels.

These components are combined into a single score ranging from -1 to 1, where 1 means identical, 0 means no similarity, and -1 indicates complete difference. SSIM calculates the score over multiple small regions of the images, providing a perceptually accurate measure of similarity that aligns with human visual perception. The formula to compute SI is represented in Equation 3 (E3). Figure 13 shows the screenshot of SSIM code snippet.

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (3)$$

```python
from math import log10, sqrt
import cv2
import numpy as np

def PSNR(original, compressed):
    mse = np.mean((original - compressed) ** 2)
    if(mse == 0): # MSE is zero means no noise is present in the signal .
                  # Therefore PSNR have no importance.
        return 100
    max_pixel = 255.0
    psnr = 20 * log10(max_pixel / sqrt(mse))
    return psnr

def main():
    original = cv2.imread("iMark.jpg")
    compressed = cv2.imread("iMark4.jpg", 1)
    value = PSNR(original, compressed)
    print(f"PSNR value is {value} dB")

if __name__ == "__main__":
    main()
```

Figure 14. Python Code Snippet for PSNR Measuring Metrics

The Peak Signal-to-Noise Ratio (PSNR) is a metric used to compare the quality of two images by measuring the difference between them. It starts by calculating the Mean Squared Error (MSE), which quantifies the average squared differences between corresponding pixels in the original and distorted images. A lower MSE indicates higher similarity. PSNR then uses the MSE to compute the ratio of the maximum possible pixel value to the distortion, expressed in decibels (dB). The higher the PSNR value, the closer the images are in quality, with higher values typically indicating better fidelity to the original image. The formula to compute the PSNR is represented in Equation 4 (E4). Figure 14 shows the screenshot of the python code snippet of the PSNR.

$$PSNR = 10 \log_{10}\left(\frac{R^2}{MSE}\right) \quad (4)$$

Image Attack Tool

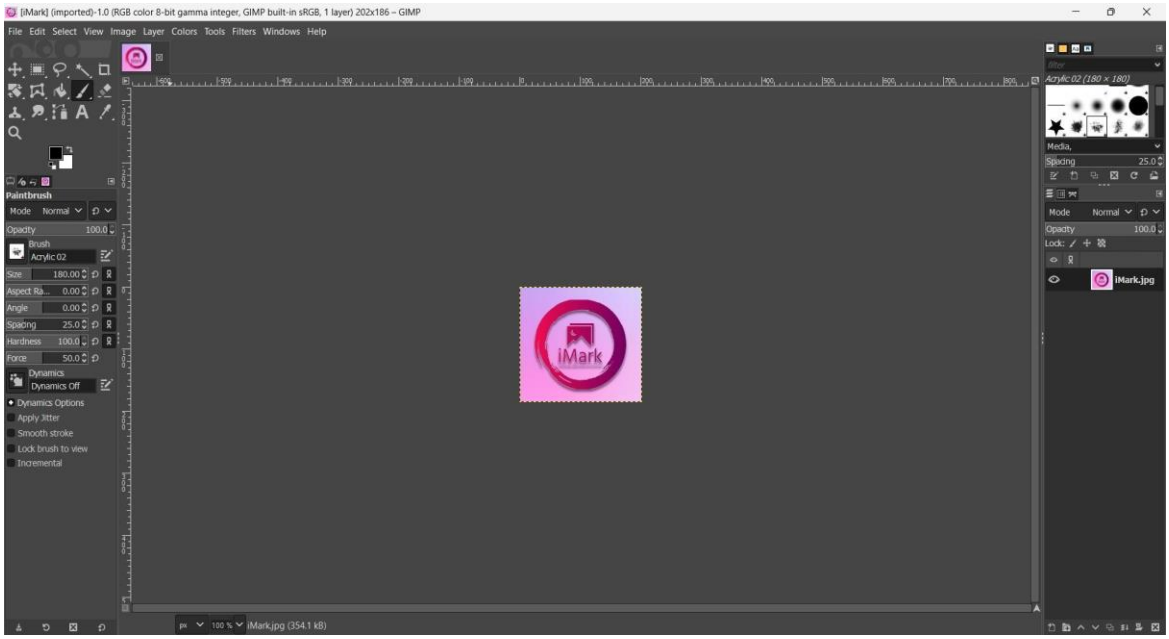


Figure 15. Image attack program for watermarked image

Figure 15 is the tool used for attacking watermarked images. GNU Image Manipulation Program, commonly known by its acronym GIMP, is a free and open-source raster graphics editor used for image manipulation and image editing, free-form drawing, transcoding between different image file formats, and more specialized tasks. It is extensible by means of plugins, and scriptable. It is not designed to be used for drawing, though some artists and creators have used it in this way. The watermarked image will be attacked using this tool, then the extracted watermarked image will be compared to the original and the attacked watermark using SSIM and PSNR image quality metrics evaluations. These evaluations help quantify the effectiveness of the watermark attack and the extent of image quality degradation.

Ethical Considerations

A pivotal role in the development and deployment of the mobile application, ensuring responsible and secure usage. Ensures that the app is not only functional and user-friendly but also respects the rights and well-being of its users.:

User Privacy. Robust measures are implemented to guarantee the secure handling of user data and images, safeguarding user privacy.

Data Collection. Collect only the data that is necessary for the app's functionality. Avoid collecting excessive or sensitive personal information unless necessary.

I**nformed Consent**. Ensure users provide informed consent before their data is collected or used. This includes clear and concise explanations of what they are consenting to.

Data Security. Implement robust security measures to protect user data from breaches and unauthorized access. Encrypt sensitive information both in transit and at rest.

Data Storage. Adherence to secure storage practices for encryption keys and user images, preventing unauthorized access and potential data breaches.

Chapter IV

RESULTS AND DISCUSSION

This chapter indicates the results and analysis of findings made in the conduct of the study. The created Android application interface is presented that features embedding, detecting, and extracting invisible watermarks using DCT and RSA algorithms to create an invisible watermarking feature in live captured images. The researchers also performed different attacks on the watermarked image to determine the watermarking scheme's robustness against various attacks and tampering on the watermarked image to be evaluated using Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) metrics of the retrieved watermarks and original watermarks.

Features of Developed Mobile Application

The developed mobile application can capture and embed images in real-time (cover image/original image) and acquire and upload desired smaller images to be used as the watermark for instantaneous watermark embedding. This data is necessary for the watermark embedding feature. The supported images are in JPEG/JPG and PNG formats. After the watermarking process is complete, users cannot use the same watermarked image as another input for embedding.

Figure 16 shows the interface images for the watermark embedding page and the setting for changing the desired watermark image. The right figure is the interface for the main camera used for the watermark embedding, the left figure shows the interface for the options of the app for changing watermark, detecting, and extracting.

These interfaces provide a user-friendly experience, allowing users to easily navigate and utilize the various features of the watermarking application.

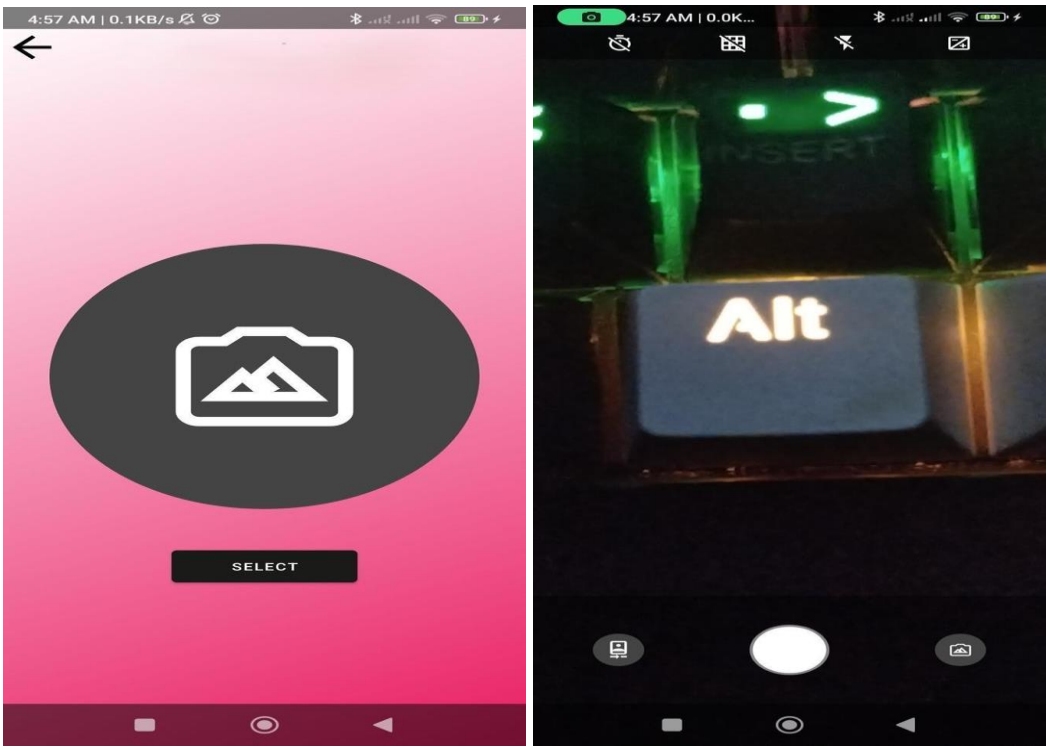


Figure 16. Watermark change setting (left) and Interface of the Camera (right)

Figure 17 displays a screenshot of the storage interface, revealing the generated keys essential for the watermarking process, including the seed key, encryption key, and decryption key stored within the mobile phone's storage system. This centralized storage of cryptographic keys ensures secure and efficient management during the watermarking and extracting procedure, safeguarding sensitive information from unauthorized access or loss. It also provides convenient access to these keys for subsequent watermark embedding or extraction tasks, facilitating seamless operation of the application.

Figure 17: Storage Preview

The detection phase of the mobile application is the ability to determine the presence of a watermark in images. In this interface, users can upload images to detect whether a watermarking process has already been applied to a specific image. This process effectively identifies if an image has been embedded with a watermark, ensuring the reliability and validity of digital media. Figure 18 illustrates the interface images for watermark detection on this page. This user-friendly feature allows for quick and efficient verification of watermarks, aiding users in safeguarding their content. Future updates will aim to enhance detection capabilities, further improving the app's effectiveness in various use cases. These improvements may include advanced algorithms for detecting hidden watermarks, support for additional image formats, and integration with cloud-based storage solutions for enhanced security and accessibility.

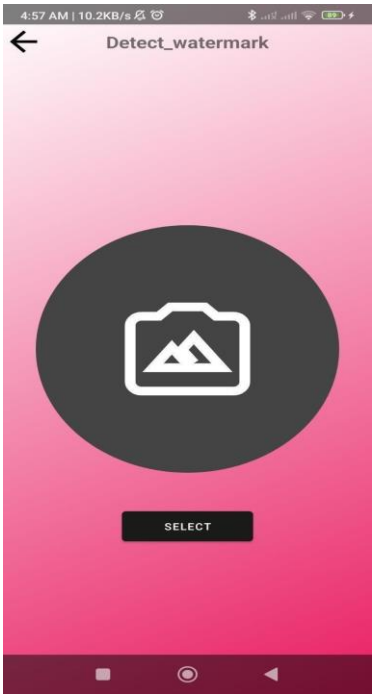


Figure 18: Watermark Detection Interface

Figure 19 visually represents the outcome of watermark detection. On the left, the absence of a watermark is indicated, while the right side confirms the presence of a watermark in the image. This system plays a crucial role in verifying the presence of watermarks in images, ensuring the reliability and accuracy of watermark detection processes. The implementation of this watermark detection system is instrumental in verifying the presence of watermarks in images, contributing significantly to the reliability and accuracy of the detection processes. Its effectiveness in discerning the presence or absence of watermarks ensures that digital assets can be securely managed and authenticated, adding an extra layer of assurance to digital content management strategies. As technology evolves, ongoing improvements and updates will continue to enhance the system's capabilities, adapting to emerging challenges and user needs in digital watermarking technology.

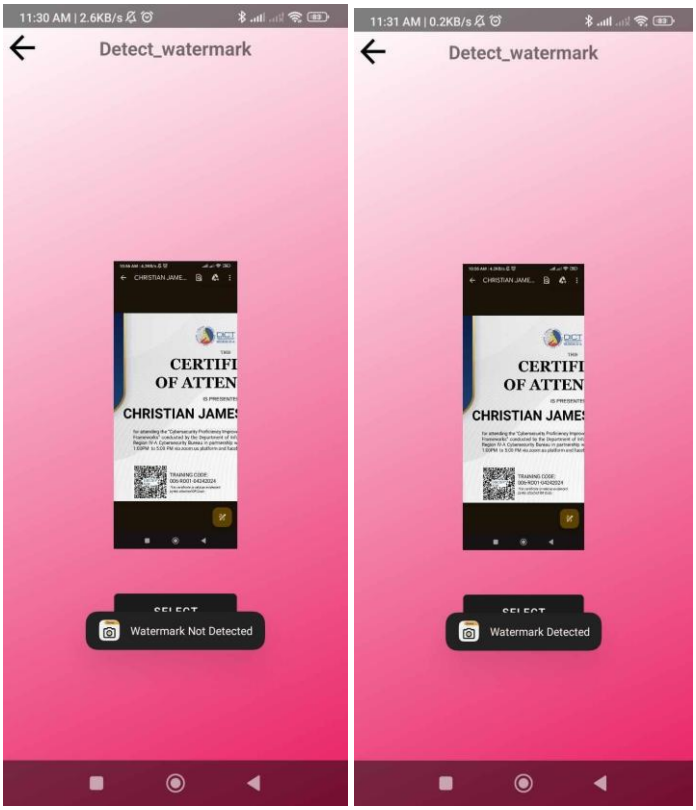


Figure 19: Watermark Detection Results

Extraction phase involves the extraction of watermarks from images. This process entails extracting the embedded watermark from a cover image, requiring user inputs such as the original image, watermarked image, decryption key, and seed key. After completing the extraction process, the extracted watermark(s) are saved to the device's storage. Figure 20 showcases the interface designed for the watermark extraction page. This feature aims to facilitate efficient retrieval and verification of embedded watermarks, supported by various encryption methods that enhance the security and reliability of the extraction process. Users can securely manage and verify their digital content through this intuitive interface, ensuring the integrity and authenticity of embedded information.

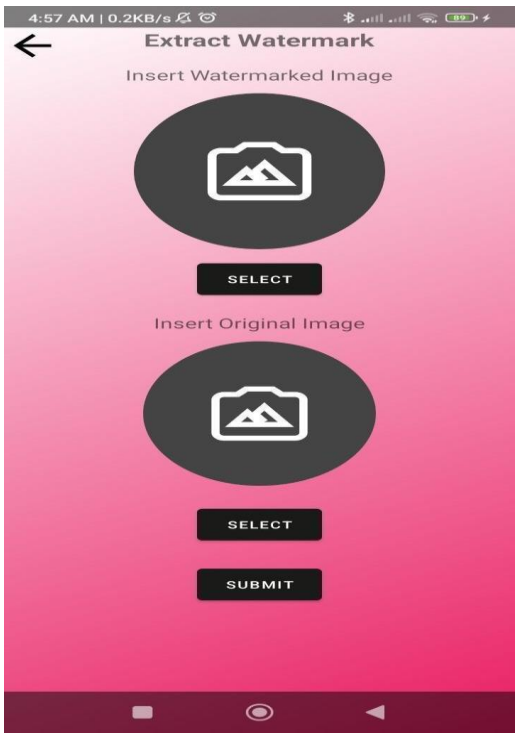


Figure 20: Watermark Extraction Interface

Utilization of Non-Blind-Semi-Blind Watermarking technique

After shuffling the watermark image pixels, the next step involves applying the Discrete Cosine Transform (DCT) to both the cover image and the shuffled watermark image. This transforms the spatial domain data into the frequency domain, facilitating the embedding of the watermark. The DCT coefficients are then modified based on the RSA-generated keys, ensuring secure and imperceptible embedding of the watermark into the cover image. This method enhances the robustness and security of the watermarking process against various attacks, making it suitable for applications requiring copyright protection and authentication in digital media. By embedding the watermark in the frequency domain, the approach effectively leverages the human visual system's sensitivity to different frequency components, further ensuring that the watermark remains imperceptible to the naked eye.

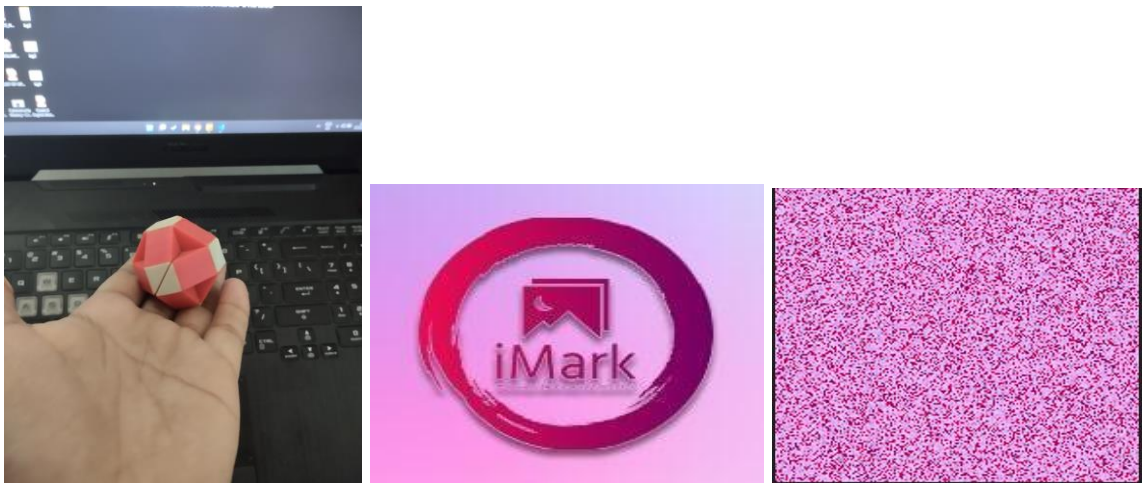


Figure 21. Preview of Cover Image (Left), Original Watermark Image (Middle), and Shuffled Watermark Image (Right)

In the watermarking system, RSA keys are generated using a process that emphasizes security and unpredictability. The key generation process begins by selecting two distinct random prime numbers, p and q, based on the current time and date. These prime numbers are crucial as they form the foundation of the RSA algorithm, ensuring the strength and uniqueness of the generated keys. Once selected, these primes are used to compute the modulus and totient, which are integral components in the creation of the public and private keys, thereby guaranteeing robust encryption.

The system will generate encryption, decryption, and seed key. It will serve as the input for the decryption process of the watermark, if the user does not have access to the keys, they will not be able to perform any extraction of watermarks. The utilization of RSA-generated keys, such as those depicted in Table 5, not only guarantees the confidentiality and integrity of the embedded watermark but also reinforces the system's ability to resist potential tampering or unauthorized extraction attempts, thereby ensuring robust digital content protection in various applications.

Table 5

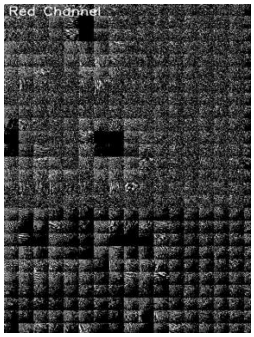
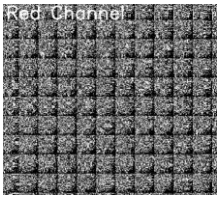
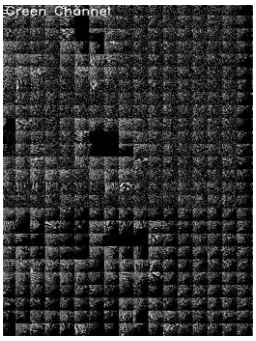
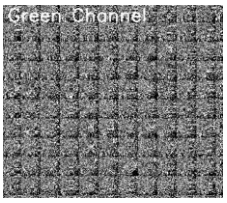
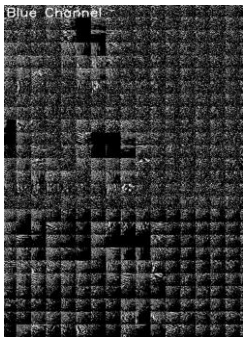
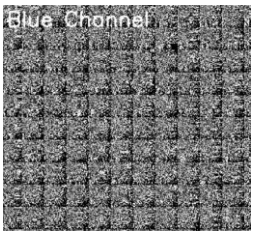
Generated Keys of RSA Algorithm and Seed

Key	**Value**
Encryption	7 1001703000028795247
Decryption	715502141447923543 1001703000028795247
Seed	587068342272 1522435234375 373669453125 587068342272 1174711139837 373669453125 781250000000 587068342272 781250000000 1028071702528 69833729609375 587068342272 781250000000 373669453125 897410677851 897410677851 373669453125 897410677851 678223072849
Seed (Time and date	13-05-2024_19-20-30

The utilization of DCT in the watermarking process involves separating the watermark and the cover image into their respective RGB channels. After shuffling the pixels of the watermark image, the DCT block image transformation is performed. Separating the image into its RGB channels before DCT allows for more precise control over the watermark embedding process. This is important for watermarking because certain frequencies are less perceptible to the human eye, allowing the watermark to be embedded more imperceptibly. The results of the transformed watermark and cover images for each color channel are shown in Table 6. Moreover, by analyzing and embedding the watermark separately in each RGB channel after DCT transformation, the watermarking process achieves enhanced resilience and fidelity across different color components, ensuring robust protection of digital content against potential unauthorized use or manipulation.

Table 6

Transformed shuffled watermark and cover image using DCT

Color Channel	Cover	Shuffled Watermark
Red		
Green		
Blue		

After transforming both images, the embedding process begins. Specifically, the Red channel of the watermark is embedded into the Red channel of the cover image, the Green channel of the watermark is embedded into the Green channel of the cover image, and the Blue channel of the watermark is embedded into the Blue channel of the cover image.

This channel-wise embedding preserves the color information of both the watermark and the cover image, leading to a more natural-looking watermarked image. Figure 22 shows the combined result of the color channels after embedding the transformed watermark image into the transformed cover image.

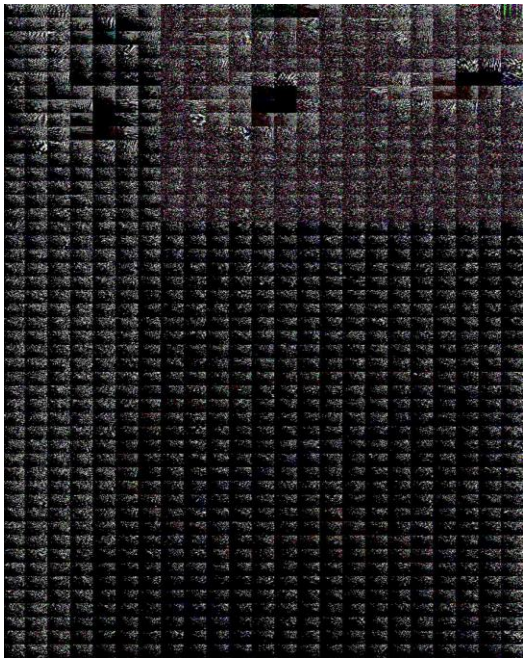


Figure 22: Result after embedding the transformed watermark to the transformed watermarked image

After applying the inverse DCT function to the watermarked image, the resulting image in Figure 23 restores the embedded watermark to its original spatial domain. This transformation is crucial for ensuring that the watermarked image appears perceptually like the original cover image while retaining the embedded information. The effectiveness of this process lies in maintaining the integrity of both the watermark and the cover image, essential for reliable identification and protection of digital content.

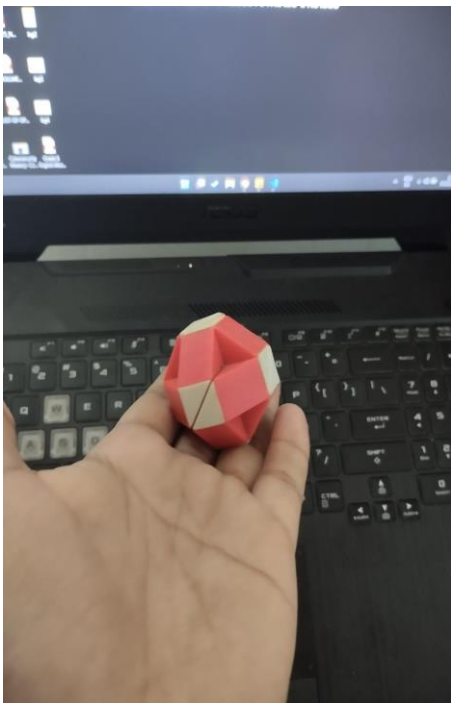


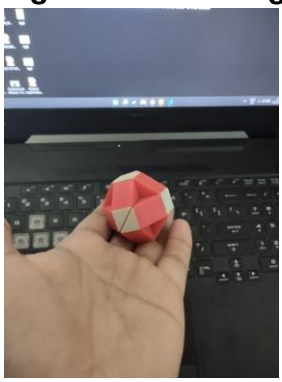
Figure 23: Result after applying inverse DCT to the embedded transform image

After completing the inverse DCT process to restore the watermarked image, the next step involves embedding the secret message strategically into every corner of the processed image. This approach ensures robustness and redundancy in message embedding, enhancing the message's resilience against potential alterations or attacks.

Table 7 shows the comparison between the original image and the actual watermarked image. The figure shows that no such difference in quality and texture has been made after embedding the watermark, aside from the embedded image which is 1.18 mb MB compared to the original image which is sitting at only 506kb. Table 7 shows and proves that the watermark had been embedded invisibly in the human eye.

Table 7

Comparison between Original Image and the Embedded Watermarked Image

Images	Size
Original Cover Image 	**506 kb**
 Embedded Watermarked Image	**1.18mb**

Several tests were conducted to determine the execution time required for each iteration of the watermark embedding process. Table 8 presents the measured time (in seconds or milliseconds) for each iteration, starting from live photo capture and continuing until the end of embedding. Additionally, the table shows the overall average time for the entire embedding process, highlighting the efficiency and performance metrics crucial for evaluating the system's operational feasibility and reliability in real-world applications.

Table 8
Runtime of Each Iteration of the Embedding Process

Number of Test Iterations	Running Time
1	**1.22 seconds**
2	**1.14 seconds**
3	**1.3 seconds**
4	**1.26 seconds**
5	**1.09 seconds**
6	**1.12 seconds**
7	**1.15 seconds**
Average	**1.18 seconds**

The image shows the four corners of the watermarked image with colored vertical lines indicating that a hidden watermark is already embedded. In Figure 22, the corners of the image are zoomed to almost 550% and the brightness is increased to make the watermark more visible. This enhancement allows for a clearer inspection of the watermark's integration and ensures the subtle details are perceptible for accurate evaluation. By closely examining these magnified sections, one can better understand the effectiveness of the embedding process and detect any potential distortions. This detailed view is crucial for validating the watermark's resilience against various image processing attacks. Moreover, scrutinizing the magnified corners of the watermarked image at increased brightness levels enables precise verification of the watermark's robust integration, ensuring its visibility and legibility under different viewing conditions and enhancing overall security and authenticity verification capabilities.

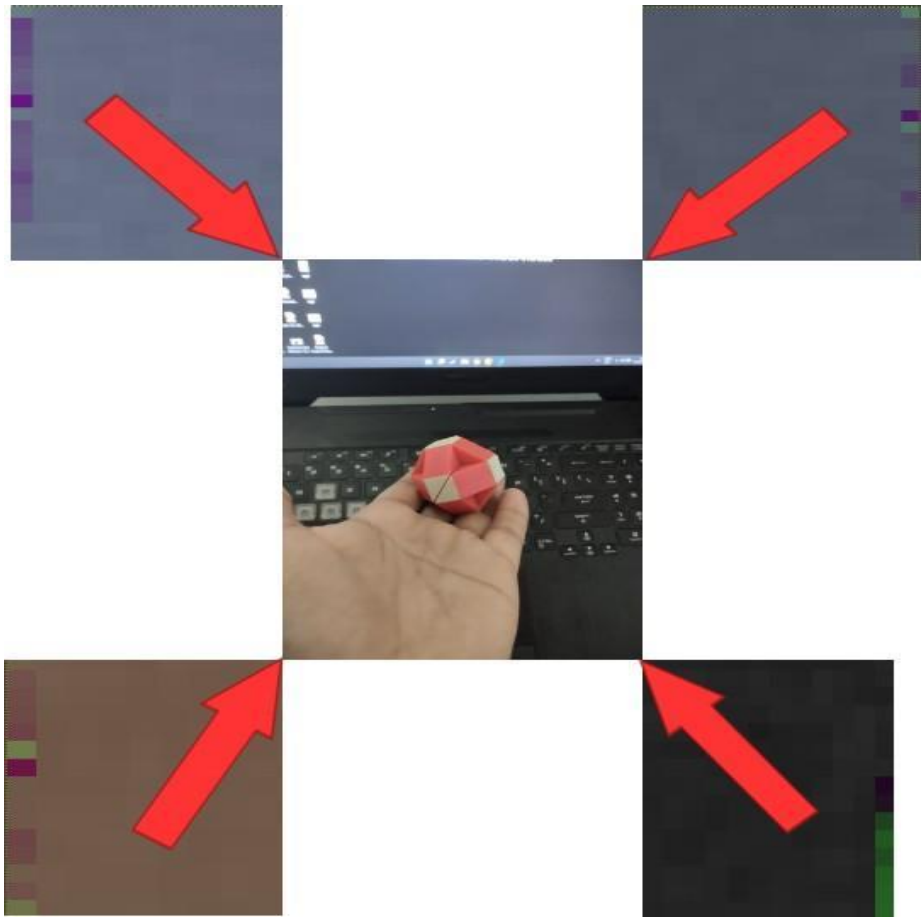
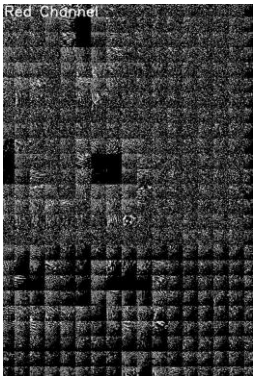
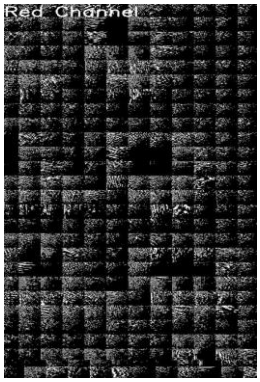
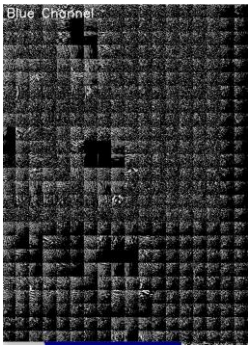
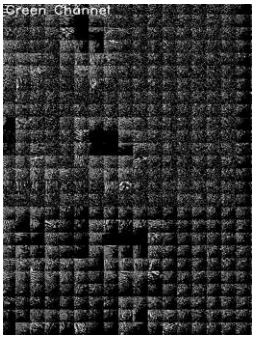
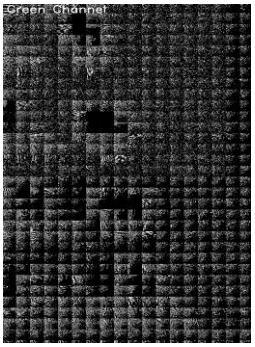
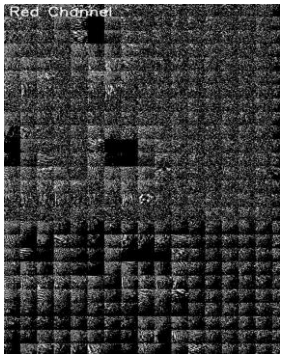


Figure 24. Colored Vertical Lines in 4 Corners of Watermarked Image

Extracting the watermark from the watermarked image requires the original image, decryption key, and encrypted seed key to retrieve the hidden watermark. DCT blocks image transformation in each colored channel of the watermarked image and original image, this process will serve as a way to identify the modification happened in the watermarked to locate the watermark. Table 9 is the result of image transformation for watermarked and original images. The transformed images are performed for every color channel of the images (RED, GREEN, BLUE).

Table 9

DCT Result of Watermarked Image and Original Image

Color Channel	Watermarked	Original
Red		
Blue		
Green		

After the transformation of the two images and locating the modification during the comparison of the original and watermarked image, the watermark extraction process is performed.

Then all channels are merged to get the transformed extracted watermark image. Figure 25 shows the results of the extracted watermark in a DCT-transformed image.

Figure 25. Extracted Watermark after performing extraction process

The extracted watermark is processed through the inverse DCT process to restore its original form. The encrypted seed and decryption key are processed through RSA to retrieve the seed. It was also used to unshuffled the pixels of resulting image of the Inverse DCT process. Figure 26 shows the result of IDCT in 32*32 blocks and unshuffled extracted images.

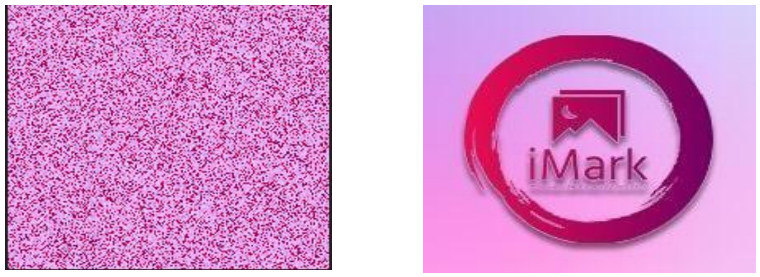


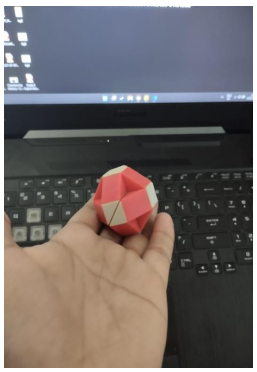
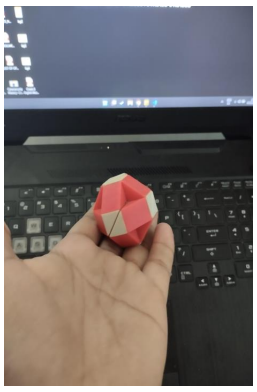
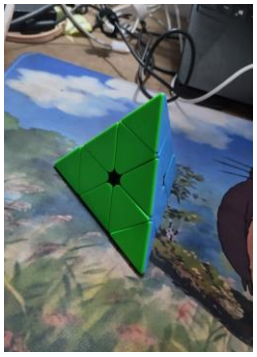
Figure 26. Result of Inverse DCT process on extracted watermark image (Left) and Result of unshuffling the result of inverse DCT process (Right)

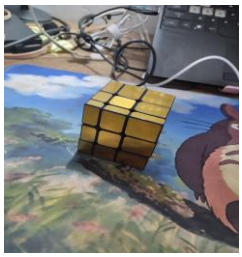
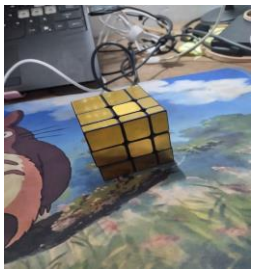
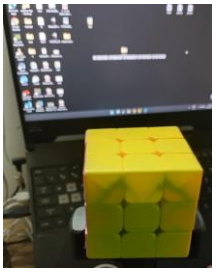
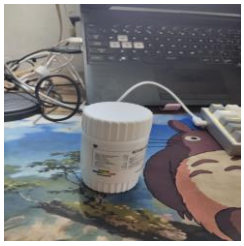
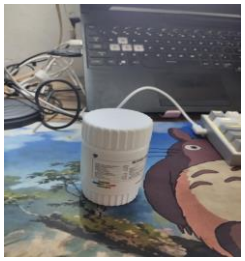
Evaluation of Non-Blind-Semi-Blind Watermarking Scheme

Table 10 shows the previews of the watermarked images and the attacked watermarked image created by the Attack tool. The attacks made by the researchers are stealth attacks to steal or destroy the ownership of the image by distorting the hidden watermark. The Geometric transformation attacks used are Flip, Rotate, and Crop. On the other hand, the filter attacks used are Blur, Noise, and Ripple

Table 10

Preview of the Watermarked and Attacked Watermarked Images

Watermarked Images	Attacked Images
	 **No Attack**
	 **Rotate**

	 **Flip**
	 **Crop**
	 **Blur**
	 **Noise**
	 **Ripple**

After the process of watermark embedding, adding attacks to every test case, extracting each attacked image, and then evaluating them using the PSNR and SSIM image quality metrics, a comprehensive analysis was conducted. Table 11 shows every SSIM and PSNR score gathered for every attack used in every watermarked image. Following the evaluation's outcome, an analysis comparing the test case results from this study with the evaluation of a related study that employed the same metrics revealed how well the developed watermarking scheme performed in comparison to previously published research. The results of Li, H. and Guo, X. (2018), studies are displayed in the adjacent row of the study's SSIM and PSNR results shown in Table 11, in their research on the DCT algorithm-based embedding and extraction of digital watermarking. This comparative assessment highlights the strengths and weaknesses of the current watermarking scheme in the context of established methodologies. Additionally, it provides insights into potential areas for improvement and future research directions to enhance the robustness and effectiveness of digital watermarking techniques. The comparative data underscores the need for continuous refinement of the algorithm to address specific vulnerabilities identified during the evaluation process. By systematically addressing these weaknesses, future iterations of the watermarking scheme can achieve higher levels of security and imperceptibility. Ultimately, the goal is to develop a watermarking solution that offers a robust defense against a wide range of digital attacks, ensuring the protection and authenticity of digital media. The continuous evolution and adaptation of watermarking techniques are essential to stay ahead of emerging threats and ensure the integrity of digital content in an ever-evolving technological landscape.

Table 11

Study's Test Cases vs. Similar Study Evaluation Comparison

ATTACKS	PSNR		SSIM	
	Study's Results	Li and Guo's Study	Study's Results	Li and Guo's Study
Rotate	**20**	0.04	**0.12**	10.04
Flip	**20**	6.06	0.2	0.2
Noise	30.88	**35.79**	0.68	**0.81**
Blur	30.09	**35.78**	0.6	**0.733**
No Attack	36	36	1	1
Ripple	15			0.16
Crop	15.92			0.01

Figure 27 shows the graph results comparing the PSNR (Peak Signal-to-Noise Ratio) of watermarked images extracted in different test cases. The graph indicates that the watermark extraction process for rotations and flips in the current study is more robust (yields higher PSNR) compared to Li and Guo's results. However, Li and Guo's study achieved higher overall PSNR scores, suggesting their watermarking method is more resilient against noise and blurring compared to the method used in this study. This indicates that while the current method excels in maintaining watermark integrity under certain geometric transformations, it may be more susceptible to degradation from noise and blurring. To improve overall robustness, future work could focus on enhancing

resistance to both geometric distortions and image quality degradation simultaneously.

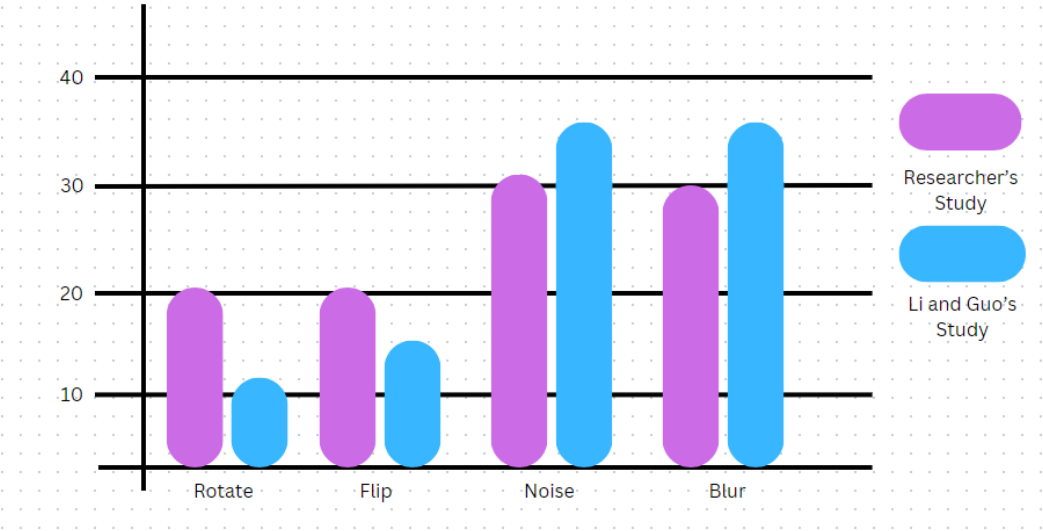


Figure 27. PSNR chart of Researcher's and Similar Study Evaluation Scores

Figure 28 shows the graph of SSIM evaluation scores of watermarked images extracted in different test cases. The graph indicates that the watermark extraction process for rotations and flips in the current study is more robust (yields higher SSIM) compared to Li and Guo's results. However, Li and Guo's study achieved higher overall SSIM scores, suggesting their watermarking method is more resilient against noise and blurring compared to the method used in this study. This discrepancy highlights the trade-offs between robustness to geometric transformations and resilience to image degradation caused by noise and blurring. Further research could explore hybrid approaches that balance these aspects more effectively.

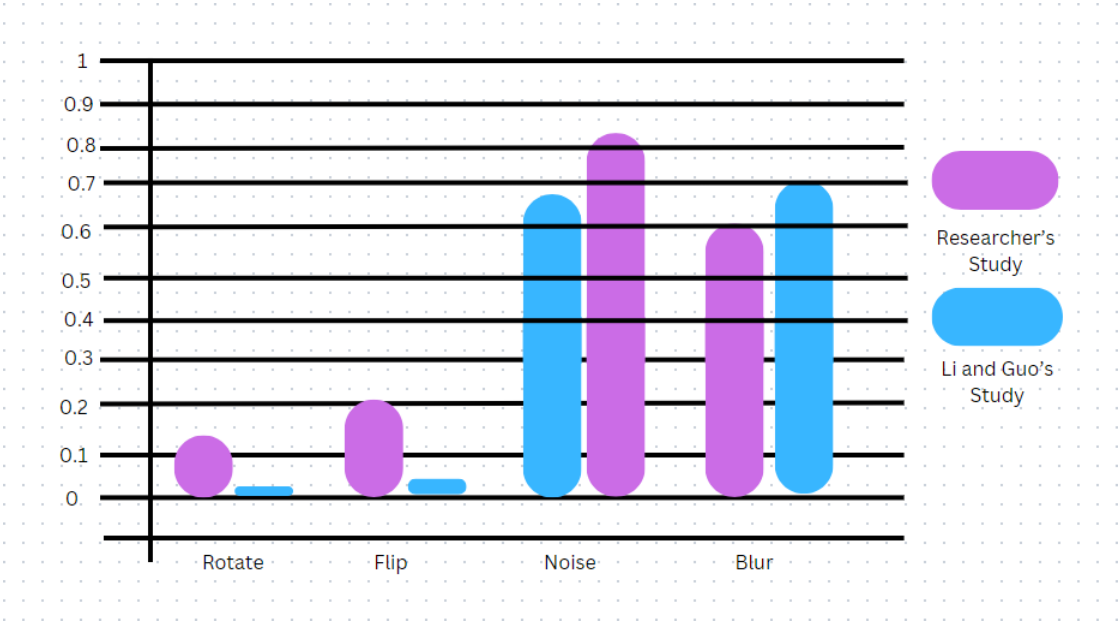


Figure 28. PSNR chart of Researcher's and Similar Study Evaluation

Chapter V

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

This chapter presents the summary, conclusions, and Recommendations of the study Invisible Watermarking in Live Captured Photo Using DCT and RSA

Summary

The created mobile application provided an interface to embed, detect, and extract watermarks by reading the required user's input (Live captured original Image, Watermark Image, Watermarked Image, Encryption, Decryption, Seed Keys) for processing and displaying the results of embed and extract after the process of embedding is finished. The utilization of the pre-set watermark for instantaneous watermarking makes this possible and efficient. The utilization of the DCT and RSA algorithm to perform the Non-Blind-Semi-Blind Watermarking Technique was discussed by displaying each process output from the embedding, extraction, significant codes, and results in detection and sample keys generated by the RSA Algorithm. Runtime results of each iteration of the embedding process.

The Non-Blind-Semi-Blind Watermarking Technique using algorithm DCT and RSA received an average score in Structural Similarity Index Measure (SSIM) by 0.3 and Peak-signal-to-Noise-Ratio (PSNR) by 33.1 based on the extracted watermark images used by applying attacks in watermarked images and non-attacked watermark image.

Conclusions

The researchers came to the following conclusions based on the summary:

1. The Created application successfully featured instantaneous embedding, and extraction of invisible watermarks. The Mobile application interfaces developed, used software tools to accomplish the features of the mobile application.
2. The Non-Blind-Semi-Blind Watermarking Technique effectively integrates and extracts hidden watermarks using DCT and RSA algorithms, ensuring secure extraction while being robust and imperceptible, with an average embedding time of 1.18 seconds per iteration demonstrating efficient performance.
3. The evaluation of the Non-Blind-Semi-blind Watermarking Technique using DCT and RSA algorithms revealed strengths in robustness against noise and blur distortions, as measured by PSNR and SSIM, while also indicating vulnerabilities to flip and rotate attacks, suggesting areas for algorithmic enhancement in digital media applications.

Recommendations

Several recommendations are made based on the conclusions drawn from this study.

1. Add a feature to take images from the gallery to add more watermarks to the already watermarked images.
2. Add more algorithms in creating invisible watermarking algorithms to help increase the robustness of the watermarked images against geometric transformation attacks.

3. Use other image quality metrics for more legibility and credibility that the watermarking scheme is robust against filter attacks.
4. Add an authentication feature in the detection phase of the app, where users will be notified if the watermarked images have been tampered with or edited before feeding them to the detection page of the app.
5. Create a dedicated server for the mobile application for the handling all the processing of all the image in all of the features of the application.

References

Alabdulatif, A. et al, (2021). Hybrid SVD-Based Image Watermarking Schemes: A Review. IEEE Access, 9, 32931-32968.

Alabdulatif, A., Alawida, M., Alshoura, W.H., Zainol, Z., Teh, J., & (2021). Hybrid SVD-Based Image Watermarking Schemes: A Review. IEEE Access, 9, 32931-32968.

Angelopoulou, E. et al, (2020), An Evaluation of Popular Copy–move Forgery Detection Approaches, IEEE Transactions on Information Forensics and Security, 7(6), pp. 1841–1854.

Anil, J., et al, (2022), EVALUATION OF DIGITAL IMAGE WATERMARKING TECHNIQUES, ISSN : 0044-0477

Araghi, T.K., & Manaf, A.A. (2019). An enhanced hybrid image watermarking

Assini, I., Badri, Baghdad, A. A., Safi, K., & Sahel, A., (2018). A Robust Hybrid Watermarking Technique for Securing Medical Image. International Journal of Intelligent Engineering and Systems, 11, 169-176.

Awaghate, A., Kakde, S., & Thakare, R. (2019). A Brief Review on: Implementation of Digital Watermarking for Color Image using DWT Method. International Conference on Communication and Signal Processing (ICCSP), 0161-0164.

Badshah, G., Hisham, S., Johari, N.H., Muhammad, A.N., & Zain, J. (2019). Numbering with spiral pattern to prove authenticity and integri

Badshah, G., Hisham, S., Johari, N.H., Muhammad, A.N., & Zain, J. (2019). Numbering with spiral pattern to prove authenticity and integrity in medical images. Pattern Analysis and Applications, 20, 1129-1144.

Begum, M. and Uddin, M. (2021) Multiple Image Watermarking with Discrete Cosine Transform. *Journal of Computer and Communications*, **9**, 88-94. doi: 10.4236/jcc.2021.93006.

Bernito, A., Neeraj, K., Ramesh, N. & Tripathi, S. (2010). A DWT based Dual Image Watermarking Technique for Authenticity and Watermark Protection. Signal & Image Processing: An International Journal, 1, 33-45.

Bernito, A., Neeraj, K., Ramesh, N. & Tripathi, S. (2018). A DWT based Dual Image Watermarking Technique for Authenticity and Watermark Protection. Signal & Image Processing: An International Journal, 1, 33-45.

Bhargava, P. (2019). Digital Image Watermarking using DCT and SVD. International Journal for Research in Applied Science and Engineering. Technology, 7, 188-191.books/dgital-images.

C. S. Lu. et al (2019), “Multimedia Security: Steganography and Digital Watermarking for Protection of Intellectual Property”, Idea Group Publishing. DOI: https://doi.org/10.4018/978-1-59140-192-6

Chang, C., Lin, C., & Tsai, P., (2019). SVD-based digital image watermarking scheme. Pattern Recognition. Lett., 26, 1577-1586.

Chen, Z., & Hu, Y., (2007). An SVD-Based Watermarking Method for Image Authentication. 2007 International Conference on Machine Learning and Cybernetics, 3, 1723-1728.

Cox I.J. and Miller M.L. (2018). A review of watermarking and the importance of perceptual modeling. Proc. SPIE Electronic Imaging ‘97, Storage and Retrieval for Image and Video Databases.

Dai, Congying, (2022), Analysis on Digital Watermarking Technology and it’s applications, 2022 International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI), **DOI:** 10.1109/ICDACAI57211.2022.00048

Delp, E., Podilchuk, C., & Wolfgang, R.B. (2019). Perceptual watermarks for digital images and video. Electronic Imaging.

Derrick, Michele, (2022), The Conservation and Art Materials EncyclopediaOnline (CAMEO), WATERMARK, https://cameo.mfa.org/wiki/Watermark

Desoubeaux, Mathieu.,(2023), Digital Watermarking: Is your content safe online?, IMATAG, https://www.imatag.com/blog/the-invisible-digital-watermarking-sag a-a-journey-through-timeDOI: https://doi.org/10.1109/icip.1996.560423

Duang, Y.F., Wang,Y,. Cao, X., Zhang, X. (2022), Research on Digital Watermarking Algorithm Based on Discreet Cosine Transform, 2022 15th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), **DOI:** 10.1109/CISP-BMEI56279.2022.9980072

Ebrahimi, H., & Poovendran, R. (2018). A comprehensive study of similarity measures for image quality assessment. Signal Processing Letters, 16(12), 901-904. (This study compares several metrics, including PSNR and SSIM, and discusses their strengths and weaknesses.)

Elbasi, E., Mostafa, N., Cina, E. (2022), Robust, Secure and Semi-Blind Watermarking Technique Using Flexible Scaling Factor in Block-Based Wavelet Algorithm, *Electronics* **2022**, *11*(22), 3680; https://doi.org/10.3390/electronics11223680

Evsultin, O., Melman, A. (2020). Digital Steganography and Watermarking for Digital Images: A review of Current Research Directions, e Russian Science Foundation under Grant 19-71-00106, Digital Object Identifier 10.1109/ACCESS.2020.3022779

H. L. Wei and J. X. Wang, (2020), “Simulation research on edge sharpening enhancement of motion blurred digital image,” *Computer Simulation*, vol. 37, no. 7, pp. 459–462, Article ID 497.

Hartung, F. et al, (2019), “Multimedia Watermarking Techniques”, Proc. IEEE, Special Issue on Identification and Protection of Multimedia Information, vol. 87, pp. 1079–1107. DOI: https://doi.org/10.1109/5.771066https://postcron.com/en/blog/watermark-on-facebook/https://www.encyclopedia.com/computing/news-wires-white-papers-and

Jane, O. et al. (2018), Hybrid non-blind watermarking based on Discrete Cosine Transform

Ketipearachchi, T., & Wickramasinghe, M. (2020). Invisible Colour Image Watermarking Technique for Colour Images Using DWT and SVD. International Journal on Advances in Ict for Emerging Regions (icter), 13.

Kwong and Jiwu Huang, Yongjian Hu, S. (2004). " Using invisible watermarks to protect visibly watermarked images," IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512), 2004, pp. 49

Lakshman Ji, D.S. (2021). Robust Digital Watermarking Techniques For Protecting Copyright.

Lakshmi, T. (2017). Fuzzy Based Invisible Watermarking. International Journal of Advanced Research in Computer Science, 8, 1063-1068.

Li, H. and Guo, X. (2018) Embedding and Extracting Digital Watermark Based on DCT Algorithm. *Journal of Computer and Communications*, **6**, 287-298. doi: 10.4236/jcc.2018.611026.

Li, Kang., Xiao-ping, Cheng, (2020). 2020 3rd International Congress on Image and Signal Processing (CISP2010)

M. Kumar et al., (2019), A Robust and Secure Image Watermarking Scheme using RSA Cryptosystem and Discrete Wavelet Transform"

Mankar, S.K., and Gurjar, A.A. (2018).: ‘Image Forgery Types and Their Detection: A Review’, International Journal of Advanced Research in Computer Science and Software Engineering, 5, (4), pp. 174-178

Mitrevski, M. (2017). Experimental comparison of PSNR and SSIM metrics for video quality estimation. Multimedia Tools and Applications, 76(4), 3387-3402. (This research examines the applicability of PSNR and SSIM for video quality assessment, offering insights into their performance in different contexts.)

Najafi, E. (2019). A robust embedding and blind extraction of image watermarking based on discrete wavelet transform. Mathematical Sciences, 11, 307-318.

Nasir, Ibrahim. (2018). Digital Watermarking of Images towards Content Protection.

P. Tao and A. M. Eskicioglu, (2018), “A Robust Multiple Watermarking Scheme in the DWT Domain,” Optics East 2004 Symposium, Internet Multimedia Management Systems V Conference, Philadelphia, PA, pp. 133-144,

Q. Wang et al., (2023), Hybrid Encrypted Watermarking Algorithm for Medical Images Based on DCT and Improved DarkNet53"

R. B. Wolfgang and E. J. Delp (2019), “A watermark for Digital Images,” in Proc. IEEE Int. Conf. Images Processing,” Lausanne, Switzerland, pp. 219–222

Rahim, M.A. (2019). Invisible Watermarking on Grayscale Image.

Raju, U., Sethi, K., Choudhary, S., & Jain, P. (2019). A new hybrid watermarking technique using DCT and DWT based on scaling factor. 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 232-235.

Reena, A., et al (2019). Modified Algorithm for Digital Image Watermarking. Using Combined DCT and DWT. International Journal of Information and Computation Technology. ISSN 0974-2239 Volume 3, Number 7, pp.691-700, from http://www. irphouse.com /ijict.h.S. Rajkumar and S. V. Subashini., (2019), "Encryption-Based Image Watermarking Algorithm in 2DWT-DCT Domains"

Savakar, D., & Pujar, S.N. (2020). Digital Image Watermarking at Different Levels of DWT using RGB Channels.

Shaimaa H. et al, (2018). Forgery Detection Based Image Processing Techniques. International Journal of Scientific & Engineering Research Volume 9, Issue 11, November-2018, ISSN 2229-5518.

Skaf, E. (2019). What is a watermark? and, how to watermark photos. Hapa,

M. and Sood, S. (2011) On Secure Digital Image Watermarking

Techniques. *Journal of Information Security*, **2**, 169-184. doi: 10.4236/jis.2011.24017.

Wang, C., & Zhou, X., Zhang, Z. (2018). Image watermarking scheme based on Arnold transform and DWT-DCT-SVD. 2016 IEEE 13th International Conference on Signal Processing (ICSP), 805-810.

Wang, Z., Li, Q., & Li, Q. (2018). Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study. Journal of Computer Science and Applications, 2(3), 51-59. (This paper compares PSNR, SSIM, MSE, and FSIM on denoised images, highlighting the importance of considering human perception.)

Weiss, I. (2021). "Digital Images" Computer Sciences.

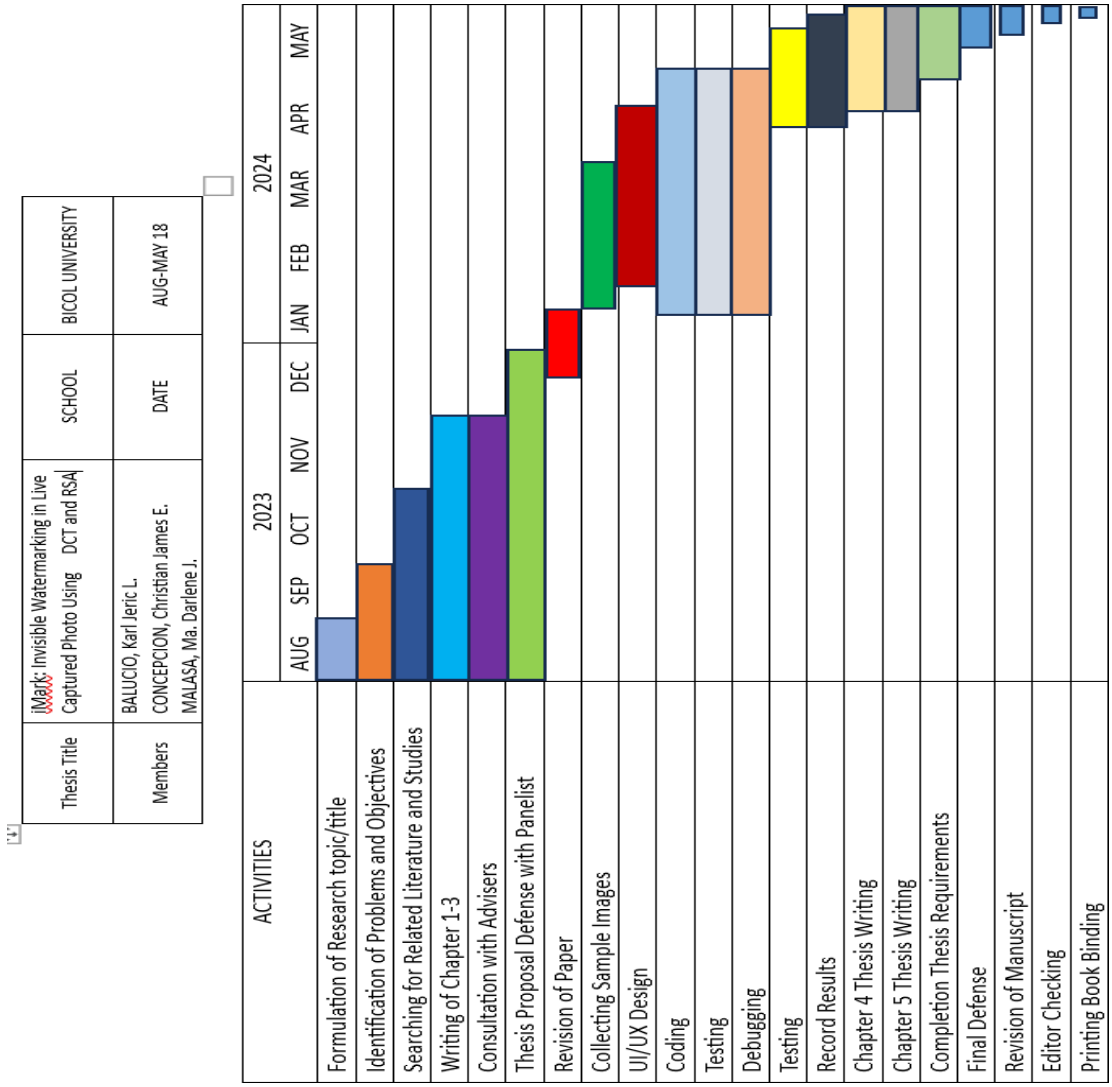
Zhang, Ruiyi., Yuan, Song. (2023), 2023 IEEE 9th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), **DOI:** 10.1109/BigDataSecurity-HPSC-IDS58521.2023.00021

Zhang, X., Hu, Guan., Ying, H., Shuwu, Z. (2020), Research on Digital Image Watermarking Technology, 2020 International Conference on Culture-oriented Science & Technology (ICCST).

APPENDICES

APPENDIX A

Gantt Chart

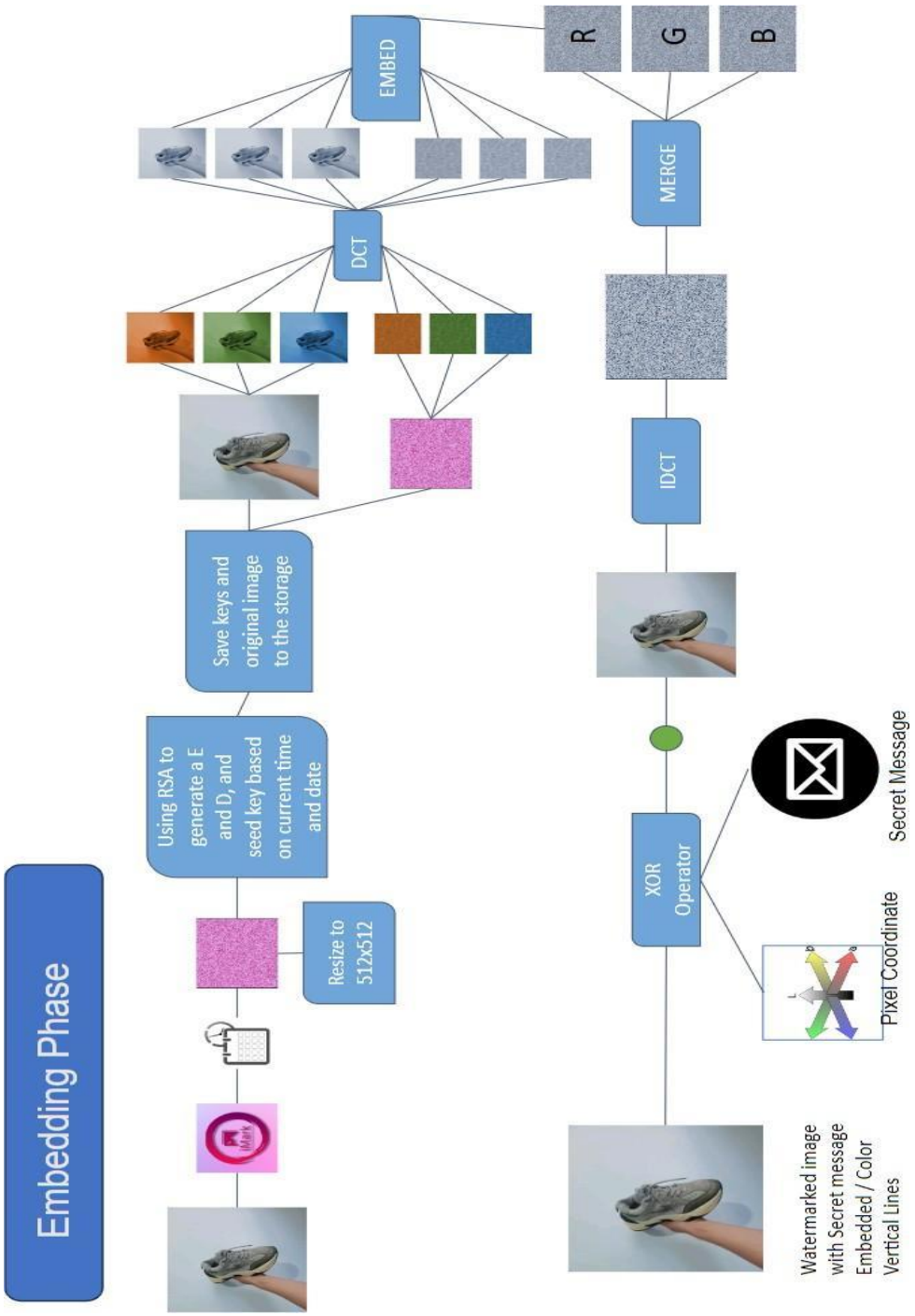


Thesis Title	iMark: Invisible Watermarking in Live Captured Photo Using DCT and RSA	SCHOOL	BICOL UNIVERSITY
Members	BALUCIO, Karl Jeric L. CONCEPCION, Christian James E. MALASA, Ma. Darlene J.	DATE	AUG-MAY 18

ACTIVITIES	2023					2024					
	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR	APR	MAY
Formulation of Research topic/title										
Identification of Problems and Objectives										
Searching for Related Literature and Studies										
Writing of Chapter 1-3										
Consultation with Advisers										
Thesis Proposal Defense with Panelist										
Revision of Paper										
Collecting Sample Images										
UI/UX Design										
Coding										
Testing										
Debugging										
Testing										
Record Results										
Chapter 4 Thesis Writing										
Chapter 5 Thesis Writing										
Completion Thesis Requirements										
Final Defense										
Revision of Manuscript										
Editor Checking										
Printing Book Binding										

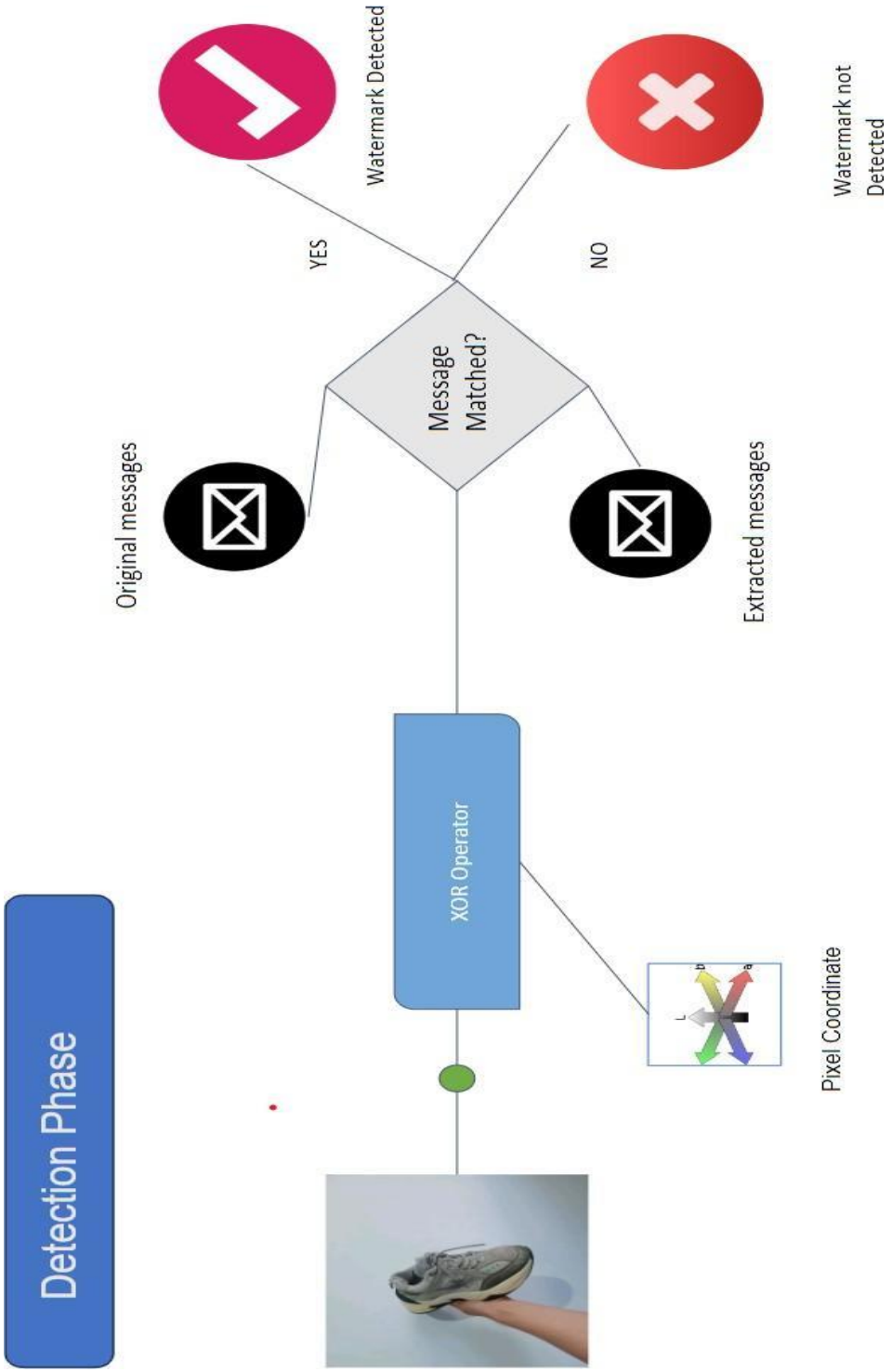
APPENDIX B

Watermark Embedding Process



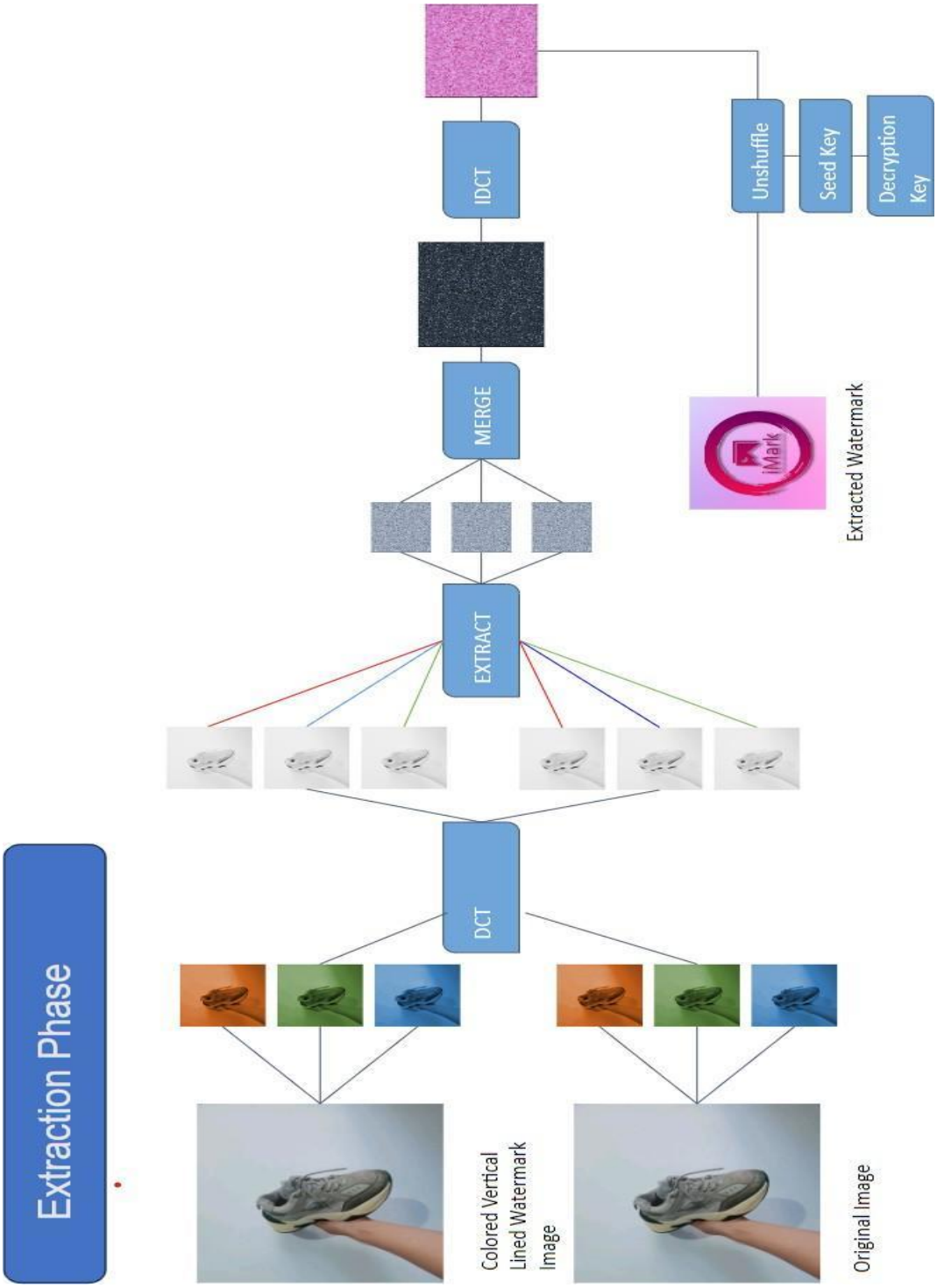
APPENDIX C

Watermark Detection Process



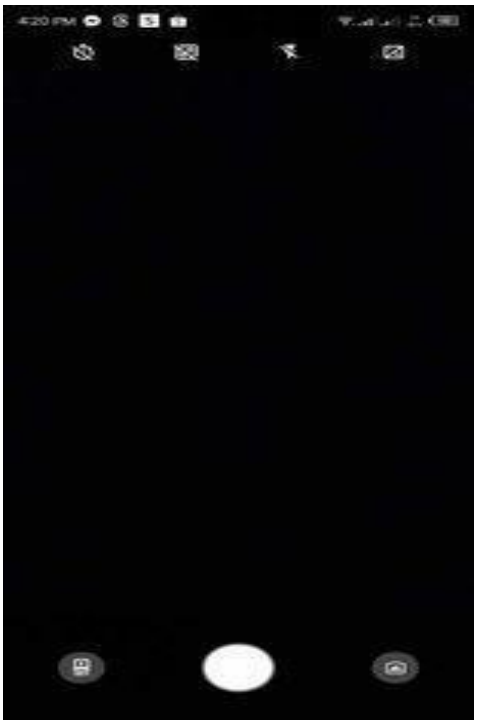
APPENDIX D

Watermark Detection Process

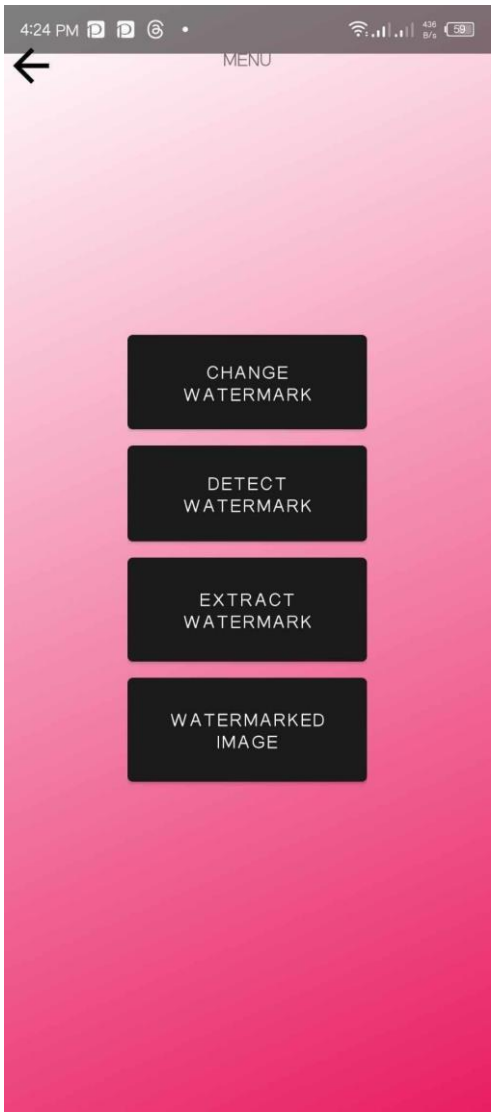


APPENDIX E

User's Manual



The middle button serves as the shutter button of the main camera for capturing and embedding watermark to the image. The left button is the button for switching the camera to front camera in case the user wants to capture a selfie or wants to use the front camera of the app. Right button serves as the gallery for all of the captured images using the application.



In order to use the other features of the app, users will need to swipe the screen to left to bring out these features of the app. The top button serves as the button for changing the watermark for embedding process. Second button is the feature to detect any watermark in input images. Third button is the button for the extraction feature of the app, this helps the users to extract any images they embedded in an image. The last button serves as the button for watermarked images. If users wants to see all images with watermark, this is the button to show them all.

APPENDIX F

Invisible Watermarking Code Snippet

DCT and INVERSE DCT ALGORITHM

```
fun transformDCT(channel: Mat): Mat {
    val blockSize = 32
    val channelFloat = Mat(channel.size(), CV_32F)
    channel.convertTo(channelFloat, CV_32F)
    val dctCoeffs = Mat(channel.size(), CV_32F)

    for (i in 0 until channel.rows() step blockSize) {
        for (j in 0 until channel.cols() step blockSize) {
            val block = channelFloat.submat(i, i + blockSize, j, j + blockSize)
            val dctBlock = Mat(block.size(), block.type())
            Core.dct(block, dctBlock)
            dctBlock.copyTo(dctCoeffs.submat(i, i + blockSize, j, j + blockSize))
        }
    }

    return dctCoeffs
}

fun inverseDCT(trans: Mat): Mat {
    val blockSize = 32
    val idctCoeffs = Mat(trans.size(), CV_32F)

    for (i in 0 until trans.rows() step blockSize) {
        for (j in 0 until trans.cols() step blockSize) {
            val block = trans.submat(i, i + blockSize, j, j + blockSize)
            val idctBlock = Mat(block.size(), block.type())
            Core.idct(block, idctBlock)
            idctBlock.copyTo(idctCoeffs.submat(i, i + blockSize, j, j + blockSize))
        }
    }

    return idctCoeffs
}
```

DETECT COLORED VERTICAL LINES

```
private fun detectColoredVerticalLine(watermarkedImage: Array<Array<Array<Int>>>): Boolean {
    // Initialize ASCII Dictionary
    val dictionary = mutableMapOf<Char, Int>()
    val character = mutableMapOf<Int, Char>()
    for (i in 0 until 256) {
        val char = i.toChar()
        dictionary[char] = i
        character[i] = char
    }

    // Initialize Messages
    val universalKey = "778911245"
    val upperLeftMsg = "Karlo Balucio"
    val lowerLeftMsg = "Christian James"
    val upperRightMsg = "Darlyn Malasa"
    val lowerRightMsg = "Thesis 2"
    var decryptedMsgLeft = ""
    var decryptMsgLowLeft = ""
    var decryptedMsgRight = ""
    var decryptMsgLowRight = ""

    // Upper Left Corner Color Channel and Coordinate
    val upLeftChannel: Int = 1
    var upLeftRow = 0
    val upLeftColumn = 0

    // Lower Left Corner Color Channel and Coordinate
    val lowLeftChannel: Int = 1
    var lowLeftRow = watermarkedImage.size - 1
    val lowLeftColumn = 0

    // Upper Right Corner Color Channel and Coordinate
    val upRightChannel: Int = 1
    var upRightRow = 0
    val upRightColumn = watermarkedImage[0].size - 1
```

```
        // Decode Message at Lower Right Corner of Image
        for ((index, char) in lowerRightMsg.withIndex()) {
            val pixelValue = watermarkedImage.getOrNull(lowRightRow)?.getOrNull(lowRightColumn)?.getOrNull(lowRightChannel) ?: 0
            pixelValue.let { value ->
                val dictionaryValue = dictionary[universalKey[index % universalKey.length]] ?: 0
                decryptMsgLowRight += character[value xor dictionaryValue]
            }
            lowRightRow--
        }

        // Check if Decoded Messages Match Original Encoding Messages
        return decryptedMsgLeft == upperLeftMsg &&
                decryptMsgLowLeft == lowerLeftMsg &&
                decryptedMsgRight == upperRightMsg &&
                decryptMsgLowRight == lowerRightMsg
    }

    private fun showToast(message: String) {
        Toast.makeText(requireContext(), message, Toast.LENGTH_SHORT).show()
    }

    companion object {
        private const val MAX_WIDTH = 5000
        private const val MAX_HEIGHT = 5000
    }
}
```

```
// Lower Right Corner Color Channel and Coordinate
val lowRightChannel: Int = 1
var lowRightRow = watermarkedImage.size - 1
val lowRightColumn = watermarkedImage[0].size - 1

// Decode Message at Upper Left Corner of Image
for ((index, char) in upperLeftMsg.withIndex()) {
    val pixelValue = watermarkedImage.getOrNull(upLeftRow)?.getOrNull(upLeftColumn)?.getOrNull(upLeftChannel) ?: 0
    pixelValue.let { value ->
        val dictionaryValue = dictionary[universalKey[index % universalKey.length]] ?: 0
        decryptedMsgLeft += character[value xor dictionaryValue]
    }
    upLeftRow++
}

// Decode Message at Lower Left Corner of Image
for ((index, char) in lowerLeftMsg.withIndex()) {
    val pixelValue = watermarkedImage.getOrNull(lowLeftRow)?.getOrNull(lowLeftColumn)?.getOrNull(lowLeftChannel) ?: 0
    pixelValue.let { value ->
        val dictionaryValue = dictionary[universalKey[index % universalKey.length]] ?: 0
        decryptMsgLowLeft += character[value xor dictionaryValue]
    }
    lowLeftRow--
}

// Decode Message at Upper Right Corner of Image
for ((index, char) in upperRightMsg.withIndex()) {
    val pixelValue = watermarkedImage.getOrNull(upRightRow)?.getOrNull(upRightColumn)?.getOrNull(upRightChannel) ?: 0
    pixelValue.let { value ->
        val dictionaryValue = dictionary[universalKey[index % universalKey.length]] ?: 0
        decryptedMsgRight += character[value xor dictionaryValue]
    }
    upRightRow++
}
```

INSERT VERTICAL COLORED LINES

```
fun insertColoredVerticalLine(imagePath: String, universalKey: String = "778911245", upperLeftMsg: String = "Karlo Balucio", lowerLeftMsg
    // Initialize ASCII dictionary
    val dictionary = mutableMapOf<Char, Int>()
    for (i in 0..255) {
        dictionary[charArrayOf(i.toChar())[0]] = i
    }

    // Load image
    val img = imread(imagePath)

    // Key length
    var kl = 0
    var kl2 = 0
    var kl3 = 0
    var kl4 = 0

    // Upper left corner coordinate
    val upLeftChannel = 1
    var upLeftRow = 0
    var upLeftColumn = 0

    // Lower left corner coordinate
    val lowLeftChannel = 1
    var lowLeftRow = img.rows() - 1
    var lowLeftColumn = 0

    // Upper right corner coordinate
    val upRightChannel = 1
    var upRightRow = 0
    var upRightColumn = img.cols() - 1

    // Lower right corner coordinate
    val lowRightChannel = 1
    var lowRightRow = img.rows() - 1
    var lowRightColumn = img.cols() - 1
```

```
    // Encode message at upper left corner of image
    for (i in upperLeftMsg.indices) {
        img.put(upLeftRow, upLeftColumn, upLeftChannel, dictionary[upperLeftMsg[i]]!! xor dictionary[universalKey[kl]]!!)
        upLeftRow += 1
        kl = (kl + 1) % universalKey.length
    }

    // Encode message at lower left corner of image
    for (i in lowerLeftMsg.indices) {
        img.put(lowLeftRow, lowLeftColumn, lowLeftChannel, dictionary[lowerLeftMsg[i]]!! xor dictionary[universalKey[kl2]]!!)
        lowLeftRow -= 1
        kl2 = (kl2 + 1) % universalKey.length
    }

    // Encode message at upper right corner of image
    for (i in upperRightMsg.indices) {
        img.put(upRightRow, upRightColumn, upRightChannel, dictionary[upperRightMsg[i]]!! xor dictionary[universalKey[kl3]]!!)
        upRightRow += 1
        kl3 = (kl3 + 1) % universalKey.length
    }

    // Encode message at lower right corner of image
    for (i in lowerRightMsg.indices) {
        img.put(lowRightRow, lowRightColumn, lowRightChannel, dictionary[lowerRightMsg[i]]!! xor dictionary[universalKey[kl4]]!!)
        lowRightRow -= 1
        kl4 = (kl4 + 1) % universalKey.length
    }

    return img
```

LOCATING PIXEL COORDINATE

```
private fun calculateInSampleSize(options: BitmapFactory.Options): Int {
    val height = options.outHeight
    val width = options.outWidth
    var inSampleSize = 1

    if (height > MAX_HEIGHT || width > MAX_WIDTH) {
        val halfHeight: Int = height / 2
        val halfWidth: Int = width / 2

        while ((halfHeight / inSampleSize) >= MAX_HEIGHT && (halfWidth / inSampleSize) >= MAX_WIDTH) {
            inSampleSize *= 2
        }
    }

    return inSampleSize
}

private fun convertBitmapToWatermarkedImage(bitmap: Bitmap): Array<Array<Array<Int>>> {
    val width = bitmap.width
    val height = bitmap.height
    val tileSize = 100 // Adjust based on memory constraints
    val watermarkedImage = Array(height) { Array(width) { Array(3) { 0 } } }

    for (y in 0 until height step tileSize) {
        for (x in 0 until width step tileSize) {
            val tileWidth = minOf(tileSize, width - x)
            val tileHeight = minOf(tileSize, height - y)
            val tileBitmap = Bitmap.createBitmap(bitmap, x, y, tileWidth, tileHeight)
            processTile(tileBitmap, x, y, watermarkedImage)
            tileBitmap.recycle() // Release the tile bitmap after processing
        }
    }
    return watermarkedImage
}
```

```
private fun processTile(tileBitmap: Bitmap, offsetX: Int, offsetY: Int, watermarkedImage: Array<Array<Array<Int>>>) {
    val tileWidth = tileBitmap.width
    val tileHeight = tileBitmap.height
    for (y in 0 until tileHeight) {
        for (x in 0 until tileWidth) {
            val pixel = tileBitmap.getPixel(x, y)
            watermarkedImage[offsetY + y][offsetX + x][0] = (pixel shr 16) and 0xFF
            watermarkedImage[offsetY + y][offsetX + x][1] = (pixel shr 8) and 0xFF
            watermarkedImage[offsetY + y][offsetX + x][2] = pixel and 0xFF
        }
    }
}
```

RSA ALGORITHM

```
fun isPrime(n: BigInteger): Boolean {
    return n.isProbablePrime(20)
}

// Calculate gcd using Euclidean algorithm
fun gcd(a: BigInteger, b: BigInteger): BigInteger {
    return if (b == BigInteger.ZERO) a else gcd(b, a % b)
}

// Extended Euclidean Algorithm
fun eea(a: BigInteger, b: BigInteger): Triple<BigInteger, BigInteger, BigInteger> {
    return if (b == BigInteger.ZERO) {
        Triple(a, BigInteger.ONE, BigInteger.ZERO)
    } else {
        val (gcd, x, y) = eea(b, a % b)
        Triple(gcd, y, x - (a / b) * y)
    }
}

// Multiplicative inverse
fun multInv(e: BigInteger, r: BigInteger): BigInteger? {
    val (gcd, x, _) = eea(e, r)
    return if (gcd != BigInteger.ONE) null else (x % r + r) % r
}
```

```
// Generate seed using RSA Encryption
fun generateSeed(e: BigInteger, n: BigInteger, method: Int): Triple<String, String, BigInteger?> {
    val dateTime = LocalDateTime.now().format(DateTimeFormatter.ofPattern("dd-MM-yyyy_HH-mm-ss"))
    var cipherText = ""
    var d: BigInteger? = null
    val msg = dateTime

    if (method == 1) {
        val random = Random()
        val p = BigInteger.probablePrime(32, random)
        val q = BigInteger.probablePrime(32, random)
        if (isPrime(p) && isPrime(q)) {
            val n = p * q
            val tntN = (p - BigInteger.ONE) * (q - BigInteger.ONE)
            var e = BigInteger.TWO

            while (gcd(e, tntN) != BigInteger.ONE) {
                e += BigInteger.ONE
            }

            d = multInv(e, tntN)

            for (char in msg) {
                val encryptedChar = BigInteger.valueOf(char.toLong()).modPow(e, n)
                cipherText += "$encryptedChar "
            }
        }
        return Triple(dateTime, cipherText.trim(), d)
    } else if (method == 2) {
        for (char in msg) {
            val encryptedChar = BigInteger.valueOf(char.toLong()).modPow(e, n)
            cipherText += "$encryptedChar "
        }
        return Triple(dateTime, cipherText.trim(), d)
    }
```

```
    return Triple(dateTime, cipherText.trim(), d)
}

// Decrypt the seed
fun decryptSeed(toDecrypt: String, d: BigInteger, n: BigInteger): String {
    val parts = toDecrypt.split(" ")
    var decrypted = ""

    for (part in parts) {
        val decryptedChar = BigInteger(part).modPow(d, n).toInt().toChar()
        decrypted += decryptedChar
    }

    return decrypted
}
```

APPENDIX G

Letters and Appointments

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science and Information Technology Department
Legazpi City

APPOINTMENT OF THESIS 2 EVALUATORS

May 14, 2024

Chairman: **PROF. ARLENE A. SATUITO**
Member:**JENNIFER L. LLOVIDO, DIT**
Member: **MICHAEL ANGELO BROGADA, DIT**

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Balucio, Karlo Jeric L. , Concepcion, Christian James E. , Malasa, Ma. Darlene J.** who will work on the topic, "**iMark: Invisible Watermarking in Live Captured Digital Images Using DCT and RSA",** which is scheduled for its Proposal Defense on May 14, 2024 at 10:00 am in CSB2 Room 104.

As member of the panel you are asked to:

1) Appraise the validity and acceptability of the thesis work in terms of its scholarly quality, correctness of the facts and claims contained therein; and completeness as to its basic components.
2) Make sure that all the suggestions are judiciously incorporated.
3) Evaluate the research report based on adopted.
4) Provide ample time to his advisee in relation to the thesis work.
5) Orient the advisee on what might/will transpire in the defense session and
6) Be physically present during the oral defense.

You shall be entitled to an honorarium as chairman and as member of the panel, as per Board Resolution No.93, s 2006.

Very truly yours,

JOCELYN E. SERRANO, M.Sc
Dean, BUCS

Conforme:

PROF. ARLENE A. SATUITO
Chairman

JENNIFER L. LLOVIDO, DIT
Member

MICHAEL ANGELO BROGADA, DIT
Member

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science and Information Technology Department
Legazpi City

APPOINTMENT OF THESIS 2 EVALUATORS

May 14, 2024

JENNIFER L. LLOVIDO, DIT
Professor
College of Science
Legazpi City

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Balucio, Karlo Jeric L. , Concepcion, Christian James E. , Malasa, Ma. Darlene J.** who will work on the topic, "**iMark: Invisible Watermarking in Live Captured Digital Images Using DCT and RSA",** which is scheduled for its Proposal Defense on May 14, 2024 at 10:00 am in CSB2 Room 104.

As member of the panel you are asked to:

1) Appraise the validity and acceptability of the thesis work in terms of its scholarly quality, correctness of the facts and claims contained therein; and completeness as to its basic components.
2) Make sure that all the suggestions are judiciously incorporated.
3) Evaluate the research report based on adopted.
4) Provide ample time to her advisee in relation to the thesis work.
5) Orient the advisee on what might/will transpire in the defense session and
6) Be physically present during the oral defense.

You shall be entitled to an honorarium as chairman of the panel, as per Board Resolution No.93, s 2006.

Very truly yours,

JOCELYN E. SERRANO, M.Sc
Dean, BUCS

Conforme:

JENNIFER L. LLOVIDO, DIT
Member of the Panel

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science and Information Technology Department
Legazpi City

APPOINTMENT OF THESIS 2 EVALUATORS

May 14, 2024

MICHAEL ANGELO BROGADA, DIT
Professor
College of Science
Legazpi City

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Balucio, Karlo Jeric L. , Concepcion, Christian James E. , Malasa, Ma. Darlene J.** who will work on the topic, **"iMark: Invisible Watermarking in Live Captured Digital Images Using DCT and RSA",** which is scheduled for its Proposal Defense on May 14, 2024 at 10:00 am in CSB2 Room 104.

As member of the panel you are asked to:

1) Appraise the validity and acceptability of the thesis work in terms of its scholarly quality, correctness of the facts and claims contained therein; and completeness as to its basic components.
2) Make sure that all the suggestions are judiciously incorporated.
3) Evaluate the research report based on adopted.
4) Provide ample time to her advisee in relation to the thesis work.
5) Orient the advisee on what might/will transpire in the defense session and
6) Be physically present during the oral defense.

You shall be entitled to an honorarium as member of the panel, as per Board Resolution No.93, s 2006.

Very truly yours,

JOCELYN E. SERRANO, M.Sc
Dean, BUCS

Conforme:

MICHAEL ANGELO BROGADA, DIT
Member of the Panel

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science and Information Technology Department
Legazpi City

APPOINTMENT OF THESIS 2 EVALUATORS

May 14, 2024

LEA D. AUSTERO, DIT
Professor
College of Science
Legazpi City

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Balucio, Karlo Jeric L. , Concepcion, Christian James E. , Malasa, Ma. Darlene J.** who will work on the topic, "**iMark: Invisible Watermarking in Live Captured Digital Images Using DCT and RSA",** which is scheduled for its Proposal Defense on May 14, 2024 at 10:00 am in CSB2 Room 104.

As an adviser, you shall perform the following task:

1) Check the format of the manuscript.
2) Provide general editing of thesis work.
3) Attend the defense session of the advisees and record suggestions and recommendations at the panel.
4) Orient the advisee on what might/will transpire in the defense session
5) Be physically present during the oral defense.

This designation shall be entitled to a professional fee as authorized under Board Resolution No.065, s 2004.

Very truly yours,

JOCELYN E. SERRANO, M.Sc
Dean, BUCS

Conforme:

LEA D. AUSTERO, DIT
Content Adviser

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science and Information Technology Department
Legazpi City

APPOINTMENT OF THESIS 2 EVALUATORS

May 14, 2024

RYAN A. RODRIGUEZ, MIT,MCS
Professor
College of Science
Legazpi City

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Balucio, Karlo Jeric L. , Concepcion, Christian James E. , Malasa, Ma. Darlene J.** who will work on the topic, "**iMark: Invisible Watermarking in Live Captured Digital Images Using Mobile App",** which is scheduled for its Proposal Defense on May 14, 2024 at 10:00 am in CSB2 Room 102.

As an adviser, you shall perform the following task:

1) Provide technical guidance and expertise to the student in their programming tasks, algorithms, and code development.
2) Ensure that the coding and technical aspects of the thesis meet the highest standards of quality, efficiency, and security.
3) Review and provide feedback on the code, debugging, and optimization to help the student achieve their technical objectives.
4) Assist the student in selecting appropriate tools, technologies, and methodologies for the project.
5) Attend the defense session of the advisees and record suggestions and recommendations at the panel.
6) Assist in troubleshooting technical issues and roadblocks that the student may encounter during the research and implementation phases.
7) Be physically present during the oral defense.

This designation shall be entitled to a professional fee as authorized under Board Resolution No.093, s 2006.

Very truly yours,

JOCELYN E. SERRANO, MSc
DEAN, BUCS

:

Conforme:
RYAN A. RODRIGUEZ,
MIT, MSCS
Programming Advisor

Curriculum Vitae

Curriculum Vitae

PERSONAL INFORMATION

Name: Karlo Jeric L. Balucio

Address: Purok 4, San Juan St.

Sto.Domingo, Albay

Date of Birth: October 9,2000

Civil Status: Single

Religion: Protestant

Citizenship: Filipino

Motto: *"Winners are just losers that tried one more time"*

Father: Rico Balucio

Mother: Remegia Balucio

EDUCATION BACKGROUND

College: Bicol University College of Science
Legazpi City
Bachelor of Science in Computer Science
2019-2024

Secondary: STI COLLEGE LEGAZPI
LEGAZPI City
2013 - 2019

Primary: Sto. Domingo Central School
Sto. Domingo
2007-2013

Curriculum Vitae

PERSONAL INFORMATION

Name: Ma. Darlene J. Malasa

Address: Purok 3, Namantao

Daraga, Albay

Date of Birth: December 8, 2000

Civil Status: Single

Religion: Roman Catholic

Citizenship: Filipino

Motto: *"Just because you fail once, doesn't mean you're gonna fail at everything"*

Father: Edwin P. Malasa

Mother: Marivic J. Malasa

EDUCATION BACKGROUND

College: Bicol University College of Science
Legazpi City
Bachelor of Science in Computer Science
2019-2024

Secondary: Anislag National High School
Anislag, Daraga, Albay
2013 - 2019

Primary: Acacia Elementary School
Acacia, Malabon City
2007-2013

Curriculum Vitae

PERSONAL INFORMATION

Name: Christian James E. Concepcion

Address: Purok 4, Brgy 32. San Roque,

Legazpi City, Albay

Date of Birth: August 13, 2000

Civil Status: Single

Religion: Roman Catholic

Citizenship: Filipino

Motto: *“if it works don't touch it”*

Father: Friday Rechie I. Concepcion

Mother: Darlyn E. Concepcion

EDUCATION BACKGROUND

College: Bicol University College of Science
Legazpi City
Bachelor of Science in Computer Science
2019-2024

Secondary: Bicol University College of Education Integrated Laboratory School - High School
Rizal St. Legazpi City, Albay
2013 - 2019

Primary: Nazarene Elementary and Kindergarten School
Bañadero, Legazpi City, Albay
2007- 2013