KARYA ILHAM PUBLISHING
CONNECTING MINDS, BRIDGING IDEAS, INSPIRING READERS

International Journal of Advanced Research in Computational Thinking and Data Science

Journal homepage:
https://karyailham.com.my/index.php/ctds/index
ISSN: 3030-5225

Enhancing Secured QR Code Large Storage Capacity through Steganography and Compression Techniques

Azizi Abas[1,*], Roshidi Din[1], Fazli Azzali[1]

[1] Department of Computer Science, School of Computing, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia

ARTICLE INFO

Article history:
Received 18 January 2025
Received in revised form 8 February 2025
Accepted 1 March 2025
Available online 5 March 2025

Keywords:
QR Code; steganography; compression

ABSTRACT

QR codes are widely used for quick and efficient data transmission in various applications. However, their storage capacity is limited and not secured, which can be difficult to save large dataset and expose to the data leakage. To address this issue, researchers have explored various methods to extend the storage capacity without altering the QR code structure and have secured information inside. This research is to enhance the storage capacity of QR codes by combining steganography and compression techniques. Steganography allows for the concealment of additional data inside the QR code's visual elements, while compression reduces the size of the data being encoded. In this study, we implement a two-step approach: first, data is embedded using steganographic techniques then it is compressed using advanced algorithms to minimize its size. By applying these methods, the overall storage efficiency is significantly increased without compromising the code's scan ability. The principal results of the research demonstrate that the combined use of steganography and compression can increase QR code storage capacity by up to 20% and it is secured. Major conclusions indicate that this method can effectively extend the QR code applications in industries requiring the encoding of large datasets, such as banking and authorization function. This study presents a novel approach to overcoming QR code storage limitations, offering a practical solution for maximizing data storage and totally secured.

1. Introduction

QR codes, short for "Quick Response" codes, are two-dimensional barcodes designed to store a variety of data types, including URLs, text, and contact information, enabling rapid access via scanning devices like smartphones. Initially developed in 1994 by Denso Wave, a subsidiary of Toyota, QR codes were intended to streamline the tracking of automotive parts in manufacturing. Unlike traditional one-dimensional barcodes that encode data in a single line, QR codes utilize both vertical and horizontal data encoding, allowing them to store significantly more information. Today, QR codes are ubiquitous, finding applications across industries such as retail, marketing, logistics, and

* *Corresponding author.*
E-mail address: azizia@uum.edu.my

https://doi.org/10.37934/ctds.5.1.2849a

healthcare, where they enable efficient product identification, information retrieval, and contactless transactions [1,2].

The structure of a QR code consists of several key components: the finder patterns, alignment patterns, timing patterns, and data and error correction regions. The finder patterns are located at three corners of the QR code, helping scanners quickly locate and identify the code's orientation. Alignment patterns assist in correcting distortion, especially for larger QR codes, while the timing patterns ensure the scanner can properly read the data modules. The data region encodes the actual information, which is protected by error correction codes based on Reed-Solomon algorithms, allowing the code to be read accurately even if partially damaged.

Despite its widespread adoption, QR codes have certain limitations. They can be susceptible to damage or distortion, reducing their readability if the error correction capacity is exceeded. Additionally, standard QR codes have limited data capacity, typically storing up to 3 KB of data, which can be insufficient for larger data applications. However, their advantages include quick scanning, versatility in data storage, and high fault tolerance, making them ideal for various real-world applications. Innovations like dynamic QR codes and integration with technologies like steganography and data compression have emerged to address these limitations, enhancing both the capacity and security of QR codes [2,3].

1.1 Research Gap

QR codes are widely used for their convenience and versatility, yet their storage capacity remains a significant limitation, especially for applications requiring large data embedding. Despite advancements in compression techniques, most existing solutions struggle to balance efficient data reduction with the need to preserve the readability and scanability of the QR code. Additionally, enhancing storage often comes at the cost of reduced security or increased decoding complexity, making the trade-off between capacity and security a pressing issue. There is an opportunity to investigate how customized compression algorithms and innovative steganography techniques can work together to overcome these limitations without compromising the core functionality of QR codes. This presents a compelling research gap, as current studies rarely address these needs holistically. Existing studies enhancing QR code storage capacity have largely overlooked ethical implications, such as data privacy, and misuse, creating a significant gap in current literature.

Furthermore, the real-world application of enhanced QR codes presents unresolved practical challenges. Few studies offer standardized benchmarks for evaluating hybrid techniques that combine steganography and compression, leaving a gap in understanding their performance regarding storage capacity, security strength, and usability. Additionally, embedding large volumes of data often leads to issues with data integrity, potentially causing errors during scanning or decoding. There is also a critical need for scalable and adaptable solutions that cater to diverse use cases, such as dynamic QR codes or applications in industries like healthcare and logistics. Finally, considering the computational intensity of these techniques, developing lightweight algorithms optimized for resource-constrained environments like smartphones or IoT systems is essential for broader adoption. These gaps highlight the need for focused research to address the intersection of storage, security, and usability in QR code enhancements.

1.2 Challenges in QR Code Storage and Security

QR codes, while versatile and widely used, face significant challenges related to their storage capacity and security. Their two-dimensional matrix structure allows for the encoding of data both

horizontally and vertically, which offers a higher storage capacity compared to traditional one-dimensional barcodes. However, this storage capacity is still limited when handling complex or large datasets. A standard QR code typically holds up to 3 KB of data, which restricts its use in applications requiring the transmission of extensive or multimedia content. As industries like retail, logistics, and digital marketing increasingly adopt QR codes for seamless data exchange, the demand for enhanced storage capacity continues to grow, presenting a key challenge for developers and researchers [1], [4].

In addition to storage constraints, QR codes also face several security vulnerabilities. By design, QR codes are intended for open and rapid scanning, which makes them susceptible to malicious attacks like phishing, malware injection, and URL spoofing. Attackers can easily create malicious QR codes that redirect users to fraudulent websites or trigger harmful downloads without the user's awareness. This lack of inherent security poses significant risks, especially in scenarios involving financial transactions or sensitive data exchanges. For instance, the use of QR codes in mobile payments and contactless transactions has surged, making them a target for cybercriminals. Techniques like steganography and encryption have been explored to address these security concerns, yet their implementation adds complexity and can affect the user experience [5,6].

The structural limitations of QR codes also impact their reliability and efficiency. QR codes consist of finder patterns, alignment patterns, timing patterns, and data modules, with a built-in error correction mechanism based on Reed-Solomon codes. Although this error correction feature allows for data recovery even if a portion of the code is damaged or obscured, it comes at the cost of reduced storage capacity. Higher levels of error correction require more redundancy, which decreases the amount of usable data that can be stored. This trade-off between error correction and storage capacity presents a design challenge, particularly in environments with physical wear and tear or potential interference, such as outdoor advertising or industrial settings. Innovations such as dynamic QR codes, which allow for real-time updates, and the integration of data compression techniques, are being explored to mitigate these issues, but widespread implementation remains complex and costly [3,7].

1.3 Purpose of Using Steganography and Compression in QR Codes

The primary purpose of integrating steganography and compression techniques in QR codes is to address two major challenges: enhancing data capacity and ensuring the security of embedded information. QR codes, in their standard form, have limited storage capacity, typically handling up to 3 KB of data, which is insufficient for many modern applications requiring the transmission of larger datasets, multimedia files, or secure information. By applying compression algorithms, the data can be reduced in size before encoding into the QR code, effectively increasing the amount of information that can be stored without altering the code's physical size. For instance, lossless compression methods such as Huffman coding or Lempel-Ziv-Welch (LZW) can optimize the data while preserving its integrity, making them suitable for QR code applications where data fidelity is critical [3,8].

Steganography, the practice of hiding information within other data, serves a dual purpose when used in QR codes: it enhances security and preserves the visual appearance of the code. Unlike traditional methods of data encryption that visibly alter the encoded data, steganography embeds hidden information within the redundant or unused portions of the QR code, such as the error correction regions. This concealed data is not easily detectable by standard QR code readers, adding an extra layer of security against unauthorized access. This technique is especially valuable in applications requiring covert data transmission, such as digital watermarks for copyright protection or secure authentication in financial transactions. Moreover, integrating steganography with existing

error correction mechanisms allows for a balance between data concealment and error resilience, making the approach robust even in environments with potential code [9,10].

Another significant advantage of using compression and steganography together is the ability to maintain the QR code's aesthetic and usability while increasing its functionality. High-density QR codes with extensive data storage can become visually cluttered and more challenging to scan, reducing their practicality in real-world applications. Compression reduces the data volume, thus minimizing the visual complexity of the QR code. When combined with steganographic techniques, additional layers of data can be hidden without affecting the visible structure of the QR code, preserving both its scannability and appearance. This dual approach is increasingly used in industries like advertising and product labeling, where the QR code must remain unobtrusive yet capable of storing significant amounts of interactive or promotional content [10,11].

Despite these benefits, the integration of steganography and compression in QR codes also introduces certain challenges. The use of complex compression algorithms may require more computational power, potentially affecting the performance of QR code generation and scanning, especially on devices with limited processing capabilities. Additionally, embedding hidden data via steganography can interfere with the error correction features if not carefully implemented, leading to potential data loss or reduced reliability. Balancing the trade-offs between enhanced storage, security, and usability is an ongoing research focus. Future developments aim to optimize these techniques, leveraging advanced algorithms and machine learning methods to create more efficient and secure QR code solution [4,12]. Table 1 provides a concise overview of the distinct purposes and combined benefits of using steganography and compression techniques in QR code applications.

Table 1
The distinct purposes and combined benefits of using steganography and compression techniques in QR code applications

Aspect	Steganography	Compression	Combined Use
Primary Purpose	Hides data within QR code for enhanced security	Reduces data size to increase storage capacity	Enhances both storage capacity and security simultaneously
Security Enhancement	Conceals sensitive information, preventing unauthorized access	Not directly focused on security, but reduces the data footprint for safer handling	Provides covert data transmission while preserving data integrity
Impact on QR Code Structure	Utilizes redundant or unused regions for data hiding (e.g., error correction areas)	Lowers the data density, making the QR code less complex	Balances scannability, aesthetics, and functionality without altering the visible structure
Real-World Applications	Digital watermarking, secure authentication, copyright protection	Optimizing storage for multimedia content, large datasets	Advertising, product labeling, secure financial transactions
Challenges	May interfere with error correction, risk of reduced reliability if not implemented carefully	Requires computational resources for data compression and decompression	Balancing trade-offs between enhanced storage, security, and QR code usability

2. Literature Review

2.1 Overview of Data Encoding Techniques in QR Codes

QR codes utilize different data encoding techniques to efficiently store and retrieve information. The most common encoding techniques include numeric, alphanumeric, and byte/binary encoding. These methods allow QR codes to handle various types of data, ensuring their versatility across multiple applications. Additionally, the error correction capability embedded within QR codes, based

on Reed-Solomon codes, plays a critical role in maintaining data integrity. This sub topic review discusses three primary sub-topics: Basic Data Encoding Techniques, Error Correction and Data Recovery, and Advanced Encoding Methods for Enhanced Storage and Security.

2.1.1 Basic data encoding techniques

QR codes support several fundamental data encoding techniques: numeric, alphanumeric, and byte encoding, each tailored to different data requirements. Numeric encoding is the most efficient, capable of storing up to 7,089 digits, making it suitable for numerical data such as phone numbers and identification codes. Alphanumeric encoding, while slightly less efficient, can store letters, digits, and symbols, allowing up to 4,296 characters. This versatility makes alphanumeric encoding ideal for text-based data like URLs and product information. Byte encoding is the most flexible, supporting the encoding of binary data including special characters and non-Latin scripts, up to 2,953 bytes [1,4,13].

These encoding techniques ensure that QR codes can be used in a wide range of applications, from marketing to logistics. For instance, the flexibility of byte encoding has been instrumental in enabling QR codes to represent complex text and multimedia data. However, the choice of encoding directly impacts the capacity and size of the QR code. Numeric encoding, despite its efficiency, is limited in scope and cannot handle alphabetic characters, while byte encoding, though versatile, occupies more space. As a result, selecting the appropriate encoding method is crucial for optimizing the QR code's performance [14,15].

2.1.2 Error correction and data recover

QR codes incorporate error correction mechanisms to ensure data integrity even if the code is partially damaged or obscured. This feature is based on Reed-Solomon error correction, which divides the QR code data into multiple blocks with additional redundancy. There are four levels of error correction—Low (L), Medium (M), Quartile (Q), and High (H)—offering varying degrees of data recovery, ranging from 7% to 30% of the total data area (ISO/IEC 18004:2015). For instance, in high-error environments such as outdoor advertising, a higher error correction level can be used to improve reliability [14,16].

The trade-off between error correction and data capacity is a critical design consideration. Higher levels of error correction require more redundancy, which reduces the space available for actual data encoding. This trade-off can affect the usability of QR codes, especially in scenarios where a balance between data capacity and reliability is necessary. Recent studies have explored adaptive error correction methods to dynamically adjust based on the application's specific requirements, enhancing the efficiency of QR code scanning [9,17].

2.1.3 Advanced encoding methods for enhanced storage and security

To address the limitations of basic encoding techniques, advanced encoding methods have been developed, incorporating steganography and data compression. Steganographic encoding hides additional data within the error correction and unused regions of the QR code, providing a layer of security while maintaining the code’s visual appearance. This technique is especially useful for applications requiring covert data transmission, such as secure authentication and digital watermarking [9,16]. By embedding hidden data, steganographic methods enhance the overall capacity of QR codes without compromising their scan ability.

In addition to steganography, data compression techniques such as Huffman coding and Lempel-Ziv-Welch (LZW) are employed to reduce the size of the encoded data, allowing for more efficient use of the QR code's capacity. Compression algorithms minimize the data footprint, enabling the encoding of larger or more complex datasets without increasing the physical size of the QR code. This approach is particularly effective in applications involving multimedia data or extensive text content. However, the integration of these advanced methods requires careful consideration to avoid interference with the QR code's error correction features, highlighting the need for balanced design strategies [10,15].

2.2 Previous Research

Below is Table 2 listing 5 previous research studies focused on steganography and compression techniques in QR codes, highlighting their main contributions and key findings.

Table 2
An overview of notable research contributions focusing on the integration of steganography and compression techniques in QR codes

No.	Research Study	Authors	Main Contribution	Key Findings
1	"An effective steganographic technique for hiding the image data using the LSB technique "	Panigrahi, Rasmita, and Neelamadhab Padhy [16]	Developed an LSB-based steganographic technique with a user-friendly GUI for secure data hiding, encryption, and watermarking in images.	The proposed method achieves high image quality retention, with PSNR values reaching 77.67 dB, ensuring effective and secure data concealment.
2	"Compression Algorithms for Enhanced Storage in QR Codes"	Salomon [14]	Analyzed the application of data compression techniques like Huffman coding and LZW in QR code encoding.	Showed increased storage capacity while preserving the readability of the QR code across various compression methods.
3	"Steganographic Encoding for Secure Data in QR Codes"	Johnson and Jajodia [9]	Examined the potential of steganographic encoding within QR code error correction areas for covert data transmission.	Found that hidden data could be effectively embedded without altering the visual appearance or functionality of the QR code.
4	"Steganography in QR Codes—Information Hiding with Suboptimal Segmentation"	Koptyra, Katarzyna, and Marek R. Ogiela [17]	Using the segmentation feature of QR codes for steganography, where secret messages are embedded by selecting alternative segment types, ensuring that the QR codes remain valid and decodable by standard readers without compromising error correction quality.	The new steganographic method for QR codes utilizes the segmentation feature rather than relying on the error correction property, allowing for the embedding of secret messages by selecting alternative segment types without compromising the QR code's functionality.
5	"Research and Development of QR Code Steganography Based on JSteg Algorithm in DCT Domain"	Sun *et al.,*[18]	Use of the *JSteg* algorithm in the DCT domain for embedding QR codes into cover images has also been shown to maintain high security and imperceptibility, with PSNR values exceeding 47.6 dB, indicating minimal distortion	The research demonstrates a steganography algorithm that successfully embeds QR codes containing secret information into cover images using the *JSteg* algorithm, ensuring both information security and picture quality

2.3 Existing Steganographic Methods for QR Code Security

Existing steganographic methodologies that are employed to ensure the security of QR codes capitalize on a multitude of sophisticated techniques that are designed to effectively embed concealed information while simultaneously preserving the overall integrity, functionality, and readability of the QR codes in question. These methods systematically incorporate various strategies, including but not limited to the segmentation of data, the application of cryptographic principles, and the utilization of advanced computational algorithms, all aimed at significantly enhancing the security features and robustness of the QR codes in order to withstand potential malicious attacks and vulnerabilities. The subsequent sections of this discourse will meticulously delineate the principal approaches that have been developed and are currently being explored within this specialized domain of study.

2.3.1 Segmentation Techniques

Suboptimal segmentation embeds secret messages by selecting alternative segment types without relying on error correction properties, ensuring compatibility with standard QR code readers [19]. Similarly, segment manipulation divides input text into parts, utilizing multiple encoding modes to inject additional data into unused segments, thereby improving robustness and error correction [20].

2.3.2 Cryptographic enhancements

The combination of cryptography and steganography enhances security by integrating cryptographic techniques with steganographic methods, effectively addressing the vulnerabilities present in conventional cryptographic algorithms [21].

2.3.3 Visual cryptography

Layered security can be achieved by employing visual cryptography, which splits a secret image into multiple shares to enhance QR code protection. By combining cryptographic techniques with steganographic methods, this approach provides an added layer of security, ensuring dual protection for sensitive information [22].

2.4 Review of Compression Techniques for Data Optimization

Compression techniques for data optimization play a crucial role across various fields, including machine learning, text processing, and image compression. These methodologies are designed to reduce the size of data while preserving its integrity, making them invaluable for improving efficiency in data management and transmission. By minimizing data volume, these techniques facilitate more effective handling and storage, especially in systems with limited resources.

Such methods are particularly vital in resource-constrained environments where efficient data utilization is critical. They not only reduce storage requirements but also enhance transmission speed and lower bandwidth usage, enabling seamless data processing. Below is a detailed overview of the key principles and applications of these compression techniques.

i. Model compression in machine learning focuses on improving the efficiency of models for deployment in resource-constrained environments, such as mobile devices and IoT systems. By utilizing hybrid techniques that integrate multiple compression strategies, this approach optimizes performance while reducing computational and memory demands. It plays a critical role in promoting sustainable AI development, enabling high-performance solutions without imposing excessive resource requirements [23].
ii. Text compression algorithms are essential for efficient data storage and communication, with key techniques like Huffman Coding and LZ77 being widely adopted in various applications. Hybrid methods, such as the Deflate Algorithm, combine the strengths of Huffman Coding and LZ77 to achieve greater compression efficiency. These algorithms are particularly vital in cloud computing, where they enable effective management of large data volumes [24].
iii. Image compression techniques encompass both lossy and lossless methods, including Wavelet Coding, Run-Length Coding, and Arithmetic Coding, which are widely used to reduce file sizes while retaining visual fidelity [25]. A key challenge in image compression lies in achieving high compression rates without compromising image quality, as compression artifacts can degrade visual appeal. To address this, tailored approaches are recommended based on the specific characteristics of the image and its intended application, ensuring optimal performance [26].

3. Methodology

An essential aspect of research methodology is the careful selection of appropriate methods that align with the objectives and nature of the study. For instance, qualitative methods may be employed to gain in-depth insights into participants' experiences, while quantitative approaches can facilitate statistical analysis and generalization of findings to larger populations [27]. The interplay between these methodologies often leads to a more comprehensive understanding of complex issues, allowing researchers to triangulate data for enhanced validity [28]. Furthermore, it is critical for researchers to remain aware of ethical considerations throughout their work, ensuring that participant rights are safeguarded and biases minimized, thereby fostering trust and integrity within the research process [29].

3.1 Experimental Research Design

The selection of an appropriate research methodology is a fundamental component of any scholarly investigation, as it directly influences the validity and reliability of the findings. This process involves a meticulous consideration of various methods that align with the study's objectives and its inherent nature. For example, quantitative approaches enable researchers to conduct statistical analyses that can generalize findings across larger populations, thus enhancing the breadth of the research implications. The integration of qualitative and quantitative methodologies—often referred to as mixed-methods research—facilitates a more nuanced understanding of complex issues by allowing researchers to triangulate data, thereby strengthening the overall validity of the study. Additionally, ethical considerations are paramount throughout the research process; researchers must safeguard participant rights and strive to minimize biases, which is essential for fostering trust and integrity. In this context, the exploration of experimental research design emerges as a critical area of focus, further illuminating the intricate relationship between methodology and ethical research practices.

In addition to the considerations of methodology and ethics, researchers must also pay close attention to the implications of their chosen research design on data collection and analysis. The

choice between qualitative and quantitative methods significantly influences both data collection and interpretation. Quantitative data is numerical and measurable, providing insights into "how many," "how much," or "how often." In contrast, qualitative data is descriptive and observational, helping to understand the "why," "how," or "what" behind certain behaviors. Selecting the appropriate method affects the type of data gathered and the subsequent analysis approach [30]. On the other hand, quantitative designs facilitate statistical rigor but may inadvertently overlook the subtleties inherent in participants' lived experiences. This dichotomy underscores the importance of employing a mixed-methods approach where feasible, as it allows for a more holistic exploration of research questions by integrating diverse perspectives and methodologies [27].

Ultimately, this comprehensive strategy enhances the robustness of findings and ensures that both the breadth and depth of inquiry are adequately addressed, paving the way for impactful and ethically sound contributions to the field. The flow of experimental research methodology encompasses a structured approach that guides researchers from hypothesis generation to data analysis. This process is crucial for ensuring scientific validity and reliability in findings. The methodology typically includes several key stages, which will be elaborated upon sub-sections.

3.1.1 Stages of Experimental Research Methodology

The research process follows a structured methodology, beginning with the formulation of methods grounded in theoretical frameworks and existing literature. This is followed by a robust experimental design, where variables are identified, and appropriate samples, such as images and text embedded in QR codes, are selected. Data collection involves employing simulations and observational methods to measure processing and storage times across different scenarios, providing a comprehensive dataset for analysis. Statistical techniques are then applied to uncover causal relationships, enabling researchers to identify optimal conditions and highlight potential areas for improvement in experimental methodologies. The stages (see Figure 1) of the experiment research are:

i. Identify Previous Research: Researchers begin by formulating a clear method based on existing literature and theoretical frameworks [31].
ii. Experimental Design: This involves selecting appropriate samples and determining variables, ensuring that the design can effectively test the hypothesis [31,32]. This stage takes a sample image and text to hide inside the QR code.
iii. Data Collection: Various methods, including simulations, field trials, and observational studies, are employed to gather data. For instance, computer simulations can facilitate experiments in complex urban environments [32,33]. This research collects the processing time and total storage time required for different scenarios, allowing researchers to analyze the efficiency and effectiveness of various experimental setups. This analysis not only helps in identifying optimal conditions for the experiments but also provides insights into potential areas for improvement in future research methodologies.
iv. Data Analysis: Statistical methods are applied to analyze the collected data, allowing researchers to draw conclusions about causal relationships [32].
v. Conclusion: The statement of the conclusion is generated based on the data analysis result.

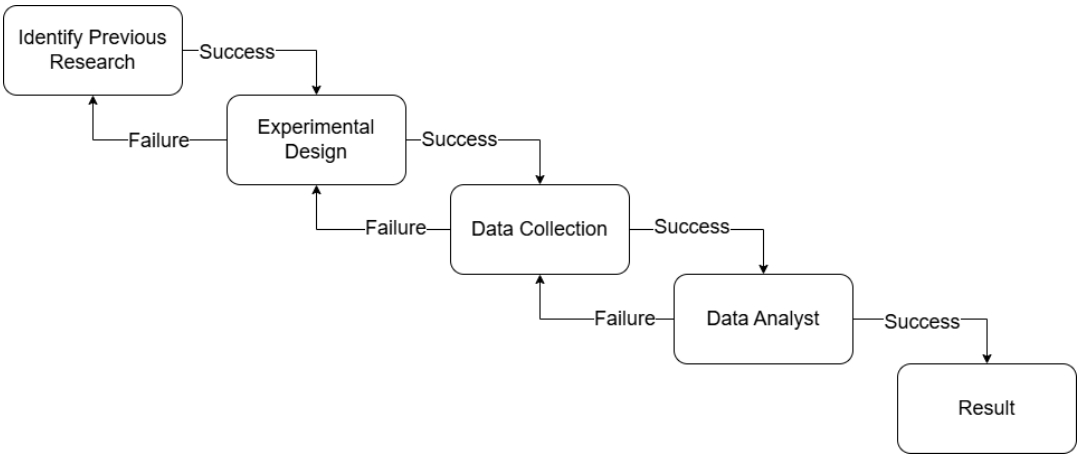

Fig. 1. Research methodology stages of the experiment research

In conclusion, the meticulous selection of research methodologies is paramount in shaping the validity and reliability of scholarly investigations. This research underscores the significance of aligning chosen methods—whether qualitative, quantitative, or a mixed-methods approach—with the specific objectives and nature of the study. By integrating diverse methodologies, researchers can achieve a more comprehensive understanding of complex issues, enhancing the overall robustness of their findings. Furthermore, ethical considerations remain a cornerstone of the research process, ensuring participant rights are protected and biases minimized, thus fostering trust and integrity. As the exploration of experimental research design unfolds, it becomes evident that the structured flow from hypothesis generation to data analysis is essential for scientific rigor. Ultimately, the thoughtful application of these methodologies not only enriches the research landscape but also paves the way for impactful contributions to the field, highlighting the intricate interplay between methodological choices and ethical research practices

3.2 Tools and Technologies Used

The effective utilization of tools and technologies in research methodology further enhances the integrity and efficiency of data collection and analysis. For instance, advancements in programming language software such as Java allow researchers to systematically code and collect data. Moreover, the integration of cloud-based platforms facilitates real-time collaboration among researchers, enabling them to share insights and refine methodologies dynamically throughout the study process [29].

This technological synergy not only streamlines workflow but also promotes transparency and reproducibility—key tenets of ethical research practices—as it allows for thorough documentation and validation of each methodological step taken. Ultimately, embracing these innovative tools can significantly bolster the overall rigor of experimental designs, ensuring that findings are both reliable and applicable across varied contexts.

3.3 Procedures for Implementing Steganography and Compression

In the realm of research, the methodology employed is a pivotal determinant of the validity and reliability of findings. A well-structured research methodology not only informs the design and execution of a study but also shapes the interpretation and implications of the results. Researchers face the critical task of selecting appropriate methods that align with the study's objectives and the

nature of the inquiry. Quantitative methods, for instance, are approach to facilitate robust statistical analyses that allow for generalization across experiment. The integration of these methodologies, often termed mixed-methods research, offers a comprehensive lens through which complex issues can be examined, enabling researchers to triangulate data for enhanced validity and depth of understanding. Furthermore, ethical considerations are an integral aspect of research methodology, necessitating that researchers prioritize the safeguarding of participant rights and the minimization of biases throughout the research process. This ethical framework fosters trust and integrity, which are essential for the credibility of any scholarly investigation. As researchers navigate the intricate relationship between methodological choices and ethical practices, the exploration of experimental research design emerges as a critical area of focus. This design encompasses a systematic flow from hypothesis generation to data analysis, ensuring scientific rigor and reliability in findings. Moreover, the advent of digital tools has revolutionized data collection methods, enabling researchers to harness vast amounts of information more efficiently. This technological evolution not only facilitates a broader scope of inquiry but also raises questions regarding data privacy and participant consent in an increasingly interconnected world. As researchers adopt these innovative methodologies, they must remain vigilant about ethical implications—ensuring that participant confidentiality is upheld while maximizing the potential for rich and varied data acquisition. Ultimately, this intersection of technology and ethics underscores the necessity for ongoing dialogue within the research community about best practices in methodological design and implementation.

4. Implementation

The selection of research methodology is a foundational element in the conduct of scholarly investigations, significantly influencing the validity and reliability of findings. A well-defined methodology not only guides the design and execution of a study but also shapes the interpretation of results and their implications. Researchers face the critical task of choosing appropriate methods that align with their study's objectives and the nature of the inquiry. For instance, while quantitative methods facilitate robust statistical analyses that allow for generalization across larger populations, qualitative approaches provide in-depth insights into participants' experiences, capturing the complexities of human behavior. The integration of these methodologies, often referred to as mixed-methods research, offers a comprehensive lens through which intricate issues can be examined, enabling researchers to triangulate data for enhanced validity and depth of understanding.

4.1 Application of Steganography in QR Codes

Steganography in QR codes serves multiple applications, primarily focusing on secure data transmission and authentication. By embedding sensitive information within the QR code, it ensures that only authorized users can access the data, thereby enhancing confidentiality during mobile communications. Additionally, steganography can authenticate QR codes by embedding hidden signatures, which helps verify the code's integrity and prevent tampering, making it crucial for secure transactions like banking. Furthermore, data hiding techniques conceal secret information within QR codes, ensuring both confidentiality and integrity. The integration of encryption with steganography adds an extra layer of security, safeguarding the data against unauthorized access. Lastly, digital watermarking techniques can be employed to protect intellectual property by embedding ownership information within the QR code, thus preventing unauthorized use. Text characters are used to hide a message. Image that fixed with the storage of QR code is encode inside code. The running program

is batch execution. Together, these applications illustrate the versatility and importance of steganography in enhancing QR code security.

4.2 Data Compression for Enhancing Storage Capacity

To enhance storage capacity through data compression, implementing lossless compression techniques is essential. Lossless compression reduces data size without sacrificing information integrity, making it suitable for applications requiring high data fidelity, such as digital audio and medical imaging. Techniques like Huffman coding, which assigns shorter codes to frequently occurring symbols, can significantly reduce data size while maintaining quality. Additionally, dictionary-based algorithms like LZ77 and LZ78 identify repeated patterns in data, replacing them with references to previous occurrences, further optimizing storage. The Deflate algorithm, which combines LZ77 and Huffman coding, is particularly effective and widely used in formats like ZIP files and PNG images. Lastly, LZW compression, known for its application in GIF images, builds a dictionary of substrings to enhance compression efficiency. By leveraging these techniques, organizations can maximize their storage capabilities effectively.

4.3 Combined Approach for Improved Security and Storage

Combining steganography and compression enhances both security and storage efficiency. One effective method is through Least Significant Bit (LSB) algorithms, which protect compressed data from unauthorized access, making hidden information more secure. Additionally, transform domain steganography allows for embedding secret data within the transform coefficients of compressed files, leveraging the compression process to further conceal hidden information. Reversible data hiding techniques (decode) enable the original data to be restored after secret information extraction, ensuring data integrity while embedding more secret bits. Furthermore, lossless compression is crucial for applications where data accuracy is paramount, as it preserves the original data while enhancing security. The best compression for this research is GZIP compression because it takes a chunk of data and makes it smaller compared with other technique. Lastly, fractal compression can also be utilized for steganography, hiding information within fractal codes, which improves both security and storage efficiency. By integrating these approaches, one can achieve a robust framework for secure data handling.

4.4 Comparative Analysis with Existing QR Code Solutions

Comparative analysis of existing QR code solutions involves evaluating various components such as QR code technology, generators, readers, and data storage capacity. QR Code Technology serves as the foundation for understanding how these codes are created and utilized, highlighting areas for potential innovation. QR Code Generators play a crucial role in ensuring user-friendly and efficient code creation, which is essential for widespread adoption. Additionally, QR Code Readers impact the usability and accessibility of these solutions, as consumers appreciate the convenience and speed they offer. The efficiency of Barcode Scanning is also significant, as it determines how effectively QR codes can be interpreted. Lastly, Data Storage Capacity is a critical factor, as it limits the amount of information that can be embedded in a QR code, with advancements potentially increasing this capacity significantly. Also, the processing time allows to get how efficiently encode and decode the data. These elements provide a comprehensive framework for assessing and improving QR code solutions.

5. Experimental Analysis and Results

Experimental analysis and results are executed through a structured process that begins with a well-defined experimental design, which ensures that the data collected is relevant, reliable, and valid [34-36]. To facilitate this analysis, a small Java programming language was developed and employed, enabling efficient processing and interpretation of large datasets [37-39]. Finally, ensuring experimental validity is essential, as it confirms that the experiment accurately measures what it intends to measure, thereby enhancing the credibility of the results [37]. Together, these components form a comprehensive framework for executing experimental analysis effectively.

5.1 Experiment elements requirements

The experiment metric employed in this research is as follows:

i. Data density: To validate the total amount of ASCII codes that can be stored in the colored QR code. The larger the value, the better the algorithm is. It is formulated based on how many characters are successfully encoded. The unit of measurement used in this testing is the total amount of characters saved in the colored QR code.
ii. Accuracy order by error correction level: To validate the data restoration rate. The higher the value, the better the algorithm is. The unit of measurement used in this accuracy test is the percentage of characters lost, which is percentage of loss data, Eq. (1):

    (Total unit character decode / Total unit character encode) * 100 (1)

iii. Processing time: To verify the total amount of time taken to complete the process. The smaller the value, the better the algorithm is. This processing time in milliseconds is formulated based on Eq. (2):

    End task time – Start task time (2)

5.2 Experiment Flow

The experiment begins with designing and testing a decoding algorithm for an RGB mono-color QR code, utilizing steganography and compression techniques to maximize data storage. It concludes with evaluating the algorithm's performance in terms of accuracy, efficiency, and practical applicability.

5.2.1 Input and output testing

Several numbers of characters are used as input. The characters include alphanumeric, numeric, binary, and several common symbols in the ASCII. The total characters need to be identified during decode processes and the total processing time will be calculated during encode and decode. The encode and decode include steganography using Least Significant Bits (LB) and compression and follow by compress and decompress the images. After these processes completed the image will be inserted to the QR Code. The text to hide is *"Hello Malaysia"* and the image used as shown in Figure 2. The output file is similar with the Figure 2 but it contains hidden messages.

Fig. 2. Input and output image with hidden message

5.2.2 Logical errors

This testing is to check the unintended or undesired output or other behavior in the algorithms that will cause them to operate incorrectly.

5.2.3 Comparison testing.

The comparison testing involves examining the result of the proposed algorithms against QR version 40 and purposes secured QR Code. The QR version 40 is used as a benchmark for all comparison testing including to see the contents inside QR code.

5.3 Result of Proposed Steganography and Compression QR Code

The flowchart depicts the process of encoding a hidden text into a QR code using steganography and compression techniques. It begins with the "Start" node, followed by an "input text to hide" stage where the user provides the text that needs to be concealed. This text is then encoded using the LSB) method, a common technique in steganography, where information is hidden in the least significant bits of a data set. After the text has been encoded, the process moves to a "compression" step, suggesting that the hidden message is compressed to further reduce its size and optimize storage.

The compressed data is then converted from The American National Standards Institute (ANSI) encoding to Unicode Transformation Format-8 (UTF-8), a more universal encoding standard. Following this, the data undergoes QR code encoding in the next step. The flowchart shows the final stage, where the encoded, compressed, and steganographically hidden data is embedded within a QR code, completing the process. The flow terminates with an "End" node, indicating the completion of the process. This flow aims to enhance QR code storage by combining data hiding, compression, and encoding techniques (see Figure 3).

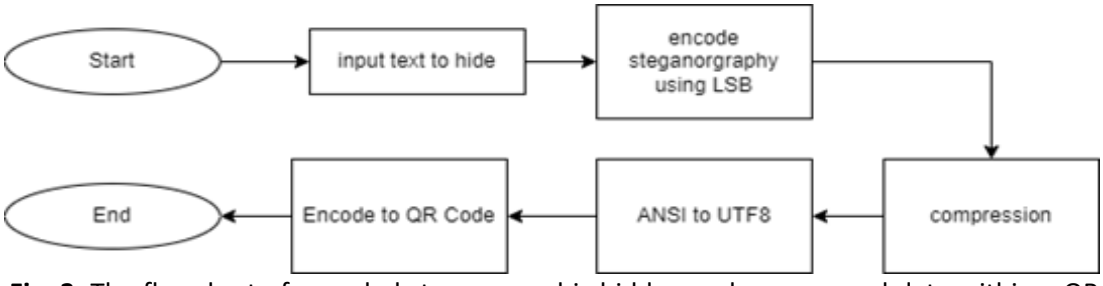

Fig. 3. The flowchart of encoded steganographic hidden and compressed data within a QR code

The flowchart illustrates in Figure 4 is the process of decoding hidden text from a QR code using steganography and decompression techniques. The process begins at the "Start" node, followed by a "Decode to QR Code" stage, where the QR code is decoded to extract the embedded data. Once decoded, the data is converted from UTF-8 encoding to ANSI, ensuring that the text is in the correct format for further processing. The next step involves decompression, where the compressed hidden text is restored to its original form.

After decompression, the flow proceeds to the "decode steganography using LSB" step, where the LSB steganographic technique is applied in reverse to extract the concealed text. Once the text has been successfully decoded, it is displayed as the "output text." The process concludes at the "End" node, signaling the completion of the decoding and extraction operation. This reverse flow mirrors the encoding process, ensuring the hidden text can be recovered from the QR code.

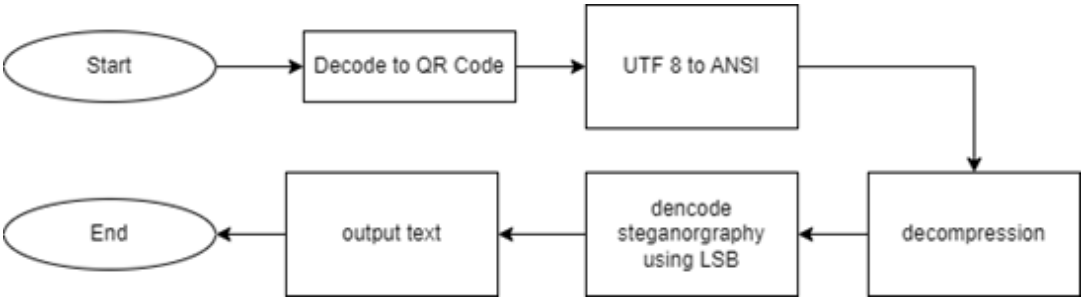

Fig. 4. The flowchart of decoded steganographic hidden and compressed data within a QR code

5.4 Steganography result

The Table 3 provides details on the meantime processing time related to the Least Significant Bit (LSB) steganography technique. It shows the average time it takes to hide and reveal text within an image, based on 100 repetitions. The process of hiding text, where information is embedded into the least significant bits of the image's pixels, takes an average of 15.46 milliseconds. In contrast, revealing the hidden text is significantly faster, with an average time of 3.24 milliseconds. When combined, the total time for both operations are 18.7 milliseconds. This highlights that hiding text is more time-consuming compared to extracting it.

Table 3
The mean time processing time related to the Least Significant Bit (LSB) steganography technique

Events (LSB)	Time Average Duration (milliseconds) (100 times repeat)
Hide Text	15.46
Reveal Text	3.24
TOTAL	18.7

5.5 Compression

This module compresses the text file by using compression tools developed by using Java programming language. The compression utility was used to reduce the capacity of text file. The text file was converted to a binary file by a compression utility. The GZip compression tool was chosen based on an experiment to obtain the best compression tool [40]. The experiment was divided into two phases. In the first phase, several compression utilities were used as an experiment to compress the text file. The compress utilities are Lempel–Ziv–Welch (LZW) [41], Zip [42]), GZip [43], Huffman

Coding [44,45], Huffman Coding merged with GZip [46], and Huffman Coding merged with Zip [40], [46]. The input data was taken from ASCII printed mode characters and it was repeated 20 times to attain the minimal value compressed. After the compression was completed, all the compressed items were kept in the QR code with error correction level 3. The result is shown in Table 4.

Table 4
The minimum character's total amount value from 20 times repeated experiment with error correction level H

Normal	Zip	GZip	LZW	Huffmann Coding	Huffman and GZip
1,271	469	632	433	109	466

The Table 4 presents the minimum total character amount values derived from 20 repeated experiments using QR codes with error correction level H. This dataset seems to be comparing the efficiency of various compression techniques in terms of minimizing the total number of characters required to encode information in a QR code. Error correction level H, which offers high-level data recovery (about 30%), would generally increase the QR code's data requirements, making compression even more critical in such scenarios

Below is an explanation of each column in the Table 4:

i. Normal (No Compression): The value (1,271) represents the baseline, where no compression algorithm is applied. This is the highest character count across all tests, which is expected since no attempt to reduce the data size is made.
ii. Zip Compression: The algorithm reduces the total character amount to 469. This significant reduction (compared to 1,271) demonstrates the efficiency of the Zip algorithm in shrinking the QR code's data requirements.
iii. GZip Compression: It combines elements of the DEFLATE compression algorithm, and reduces the character count to 632. This is less efficient compared to Zip (469), possibly due to the specific data structure being encoded.
iv. LZW Compression: The Lempel-Ziv-Welch is highly efficient, producing a result of 433 characters, the second-lowest total after Huffman Coding (109). The lower character count indicates that this dictionary-based algorithm performs well for the given dataset.
v. Huffman Coding: The Huffman Coding exhibits the most extreme compression, reducing the character count to just 109. Huffman Coding's ability to map frequently occurring symbols to shorter bit-lengths explains why it is the most effective technique in this scenario. This suggests the data structure contains repeating elements or patterns that Huffman exploits effectively.
vi. Huffman + GZip: When combining Huffman Coding and GZip, the total character count is 466, very close to the Zip-only method (469). While Huffman alone (109) outperforms this combination, adding GZip appears to diminish its standalone efficiency. This could be due to redundancy between the two methods, as both involve compression of repeating patterns in data.

5.5.1 Researcher perspective

From a researcher's point of view, this table highlights the importance of selecting appropriate compression algorithms, especially when aiming to minimize data size in QR codes with high error correction levels.

The Huffman Coding emerges as the most efficient standalone method, suggesting that the data tested may consist of highly compressible, repetitive patterns. The combination of Huffman and GZip, despite being efficient, does not outperform Huffman alone, which may indicate that GZip's compression introduces some overlap with Huffman's technique. The LZW performs remarkably well, indicating its potential for applications where fast, lossless compression is needed for structured data.

Further investigation could delve into the nature of the encoded data, as certain algorithms may outperform others depending on the structure and redundancy within the dataset. Additionally, compression methods could be tested on other types of data (e.g., binary or multimedia) to see if the trends observed here hold across different domains. This experiment contributes to optimizing QR code storage capacity, especially in scenarios demanding high error correction, where space is a premium.

Based on the data provided, the Huffman Coding is the best compression technique, as it reduces the total character count to 109, the lowest among all methods tested. This indicates that Huffman Coding is most effective for the dataset used in this experiment, likely because it exploits the frequency of repeated patterns in the data, assigning shorter codes to more frequent symbols and achieving maximum compression efficiency. The Huffman Coding stands out because it reduces the character count significantly compared to other methods, even when combined with other algorithms like GZip. Moreover, it is designed to work optimally with data that has frequent, repetitive elements, making it ideal for datasets where certain patterns occur often. However, while Huffman Coding gives the best results for this specific experiment, the choice of the best compression method depends on the characteristics of the data. For highly structured or repetitive data, Huffman Coding may be optimal. But for other types of data, such as those with less repetition or more complex structures, algorithms like LZW, Zip, or GZip might perform better. As a conclusion, by using this dataset and experiment, the Huffman Coding is the best compression technique

5.5.2 Overall result

The Table 5 presents the maximum total characters stored in QR codes across different error correction levels (H, Q, M, L) and shows the impact of various compression techniques. The Normal (uncompressed) data shows a progressive increase in character count as the error correction level decreases, ranging from 1,270 characters for error level H to 2,952 characters for error level L. When compression techniques are applied, the character count reduces significantly.

Zip compression reduces the maximum total characters for error level H to 1,560, and this trend continues with higher error levels, reaching 4,226 characters for error level L. GZip, another compression method, stores more characters than Zip, ranging from 1,784 characters at error level H to 4,480 at error level L.

LZW compression shows better performance than both Zip and GZip, storing only 1,167 characters for error level H and 3,253 characters for error level L. The most significant reduction is observed with Huffman Coding, which compresses the data down to 212 characters at error level H and 503 characters at error level L, making it the most efficient compression technique in this dataset.

However, combining Huffman with GZip or Huffman with Zip introduces some inefficiency. Huffman and GZip result in 1,364 characters for error level H, while Huffman and Zip show a similar value of 1,166 characters. These combinations, although effective, do not outperform Huffman Coding alone, which achieves the lowest character count across all error levels. The overall trend indicates that Huffman Coding is the most efficient compression method in reducing data size, particularly at higher error correction levels.

Table 5
The maximum total characters stored in the QR code by error level

Error Level	Normal	Zip	Gzip	LZW	Huffman Coding	Huffman And GZip	Huffman And Zip
H	1270	1560	1784	1167	212	1364	1166
Q	1662	2114	2405	1627	282	1827	1639
M	2330	3188	3470	2441	392	2607	2425
L	2952	4226	4480	3253	503	3323	3095

The Table 6 shows the elapsed time for compression and decompression processes for QR codes across four different error correction levels (L, M, Q, H). The time is measured in milliseconds (ms). At the L (Low) error correction level, the compression process takes the longest at 64 ms, while decompression takes 12 ms. As the error correction level increases, the compression time decreases. At M (Medium) error correction, compression takes 29 ms, and decompression takes 11 ms. For Q (Quartile) error correction, the compression time reduces further to 24 ms, with decompression taking 9 ms. Finally, at the H (High) error correction level, which offers the most robust error correction, the compression time is the fastest, taking only 21 ms, and decompression is also the quickest at 8 ms. Overall, the results indicate that as the error correction level increases, both compression and decompression times decrease, with the H level being the fastest in both processes.

Table 6
The elapsed time of compression and decompression process

Error Correction Level	Compression (milliseconds)	Decompression (milliseconds)
L	0s 64ms	0s 12ms
M	0s 29ms	0s 11ms
Q	0s 24ms	0s 9ms
H	0s 21ms	0s 8ms

5.6 Steganography and Compression

The Table 7 compares the results of data extraction and total characters stored in a QR Version 40 code using error correction level L, with and without the application of steganography and compression techniques. In the standard QR Version 40 without steganography or compression, data can be easily extracted using a phone camera, and the total data capacity is 4,296 characters. This demonstrates that under normal circumstances, the QR code is readable, and data is accessible with standard error correction.

However, when steganography and compression techniques are applied, the QR code becomes more secure, to the point where the data cannot be extracted with a phone camera, enhancing security by hiding the content. Additionally, the total data capacity increases to 5,073 characters, indicating that the combination of steganography and compression not only secures the data but also allows the QR code to store more information compared to the standard QR Version 40. This result highlights that while the application of steganography and compression techniques adds security, it also increases the total capacity of the QR code, though it sacrifices ease of access for data extraction.

Table 7
The comparison between QR code version 40 and the purpose extended QR code

Items	Secured Extract Data (Phone camera)	Total Data (Characters) with error correction level L
QR Version 40	Visible data	4296 characters
Steganography and Compression Technique	Invisible data	5073 characters

6. Conclusion

The advent of QR codes as a versatile tool for information storage and retrieval has significantly transformed digital communication. Despite their widespread adoption across various industries, conventional QR codes face notable challenges in storage capacity and security, limiting their application in scenarios demanding extensive or secure data handling. This research was undertaken to address these limitations by proposing an innovative framework that integrates steganography and compression techniques to enhance QR code storage capacity while preserving its scalability and functionality.

This study successfully demonstrated the potential of combining advanced data compression with secure data hiding methodologies to overcome the constraints of standard QR codes. By leveraging these techniques, the research provides a pathway for developing QR codes that not only accommodate larger datasets but also safeguard sensitive information against unauthorized access. The findings mark a significant step forward in QR code technology, showcasing its adaptability to modern demands in secure data transmission and expanded storage applications.

6.1 Summary of Key Contributions

This research provides significant advancements in enhancing the storage capacity of QR codes through the integration of steganography and compression techniques. The two-step approach effectively demonstrated that combining these methods could extend the storage capacity of QR codes by up to 20% without altering their scalability. The use of steganography allowed for securely embedding additional data within the QR code's visual elements, while compression minimized the size of the encoded data, optimizing storage and maintaining high data fidelity. Experimental results confirmed that this dual approach enables QR codes to handle larger datasets and improves their usability in high-demand applications such as banking, secure authentication, and digital marketing.

Further analysis of the compression methods showed Huffman Coding as the most efficient technique, achieving the highest compression ratio. Additionally, the study illustrated the applicability of steganography in securely concealing sensitive data, rendering it invisible to unauthorized readers. This pioneering methodology not only addressed the limitations of standard QR codes but also set a foundation for secure, high-capacity QR code systems suitable for diverse industrial applications.

6.2 Limitations and Future Research Directions

While the results of this study are promising, certain limitations remain. The computational overhead introduced by advanced compression and steganography techniques can affect the performance of QR code generation and scanning, especially on devices with limited processing power. Additionally, the embedding of hidden data via steganography may interfere with the error

correction features of QR codes, potentially reducing reliability in environments where codes are subject to physical wear and tear.

Future research should focus on optimizing the computational efficiency of the algorithms to ensure real-time processing capability across various platforms. Exploring adaptive error correction methods integrated with steganography and compression could further balance storage capacity and error resilience. Additionally, employing machine learning and artificial intelligence to dynamically adjust encoding parameters based on data characteristics could enhance overall efficiency. Expanding the study to multimedia content and non-standard QR codes would also offer broader insights into the versatility of the proposed methods.

6.3 Final Remarks on Enhancing QR Code Storage Capacity

The findings of this study highlight the transformative potential of steganography and compression techniques in addressing the inherent limitations of QR code storage and security. By blending these methodologies, QR codes can transcend their conventional role, evolving into robust data carriers capable of supporting modern applications that demand higher storage and heightened security. This advancement not only expands the applicability of QR codes in industrial and commercial domains but also fosters innovation in secure data transmission technologies.

The contributions made through this research serve as a stepping stone for future developments in QR code technology. With continuous refinement, the methods proposed here could redefine the standard for secure, high-capacity QR codes, paving the way for their integration into emerging fields such as IoT, blockchain, and AI-driven systems. Ultimately, the evolution of QR codes aligns with the broader vision of enhancing data accessibility and security in a rapidly digitizing world.

Acknowledgment

This research was funded by the Ministry of Higher Education (MOHE) of Malaysia under the Fundamental Research Grant Scheme for Research Acculturation of Early Career Researchers (FRGS-RACER) (RACER/1/2019/ICT03/UUM//2).

References

[1] Denso ADC. *Qr code essentials*. Denso Adc 1–12, 2011.
[2] Zhang, Linfan, and Shuang Zheng. "Enhancing QR Code Security." (2015).
[3] Rani, M. Mary Shanthi, and K. Rosemary Euphrasia. "Data security through qr code encryption and steganography." *Advanced Computing: An International Journal (ACIJ)* 7, no. 1/2 (2016): 1-7. https://doi.org/10.5121/acij.2016.7201
[4] *ISO*. "ISO/IEC 18004:2015," 2015.
[5] Ford, Richard, and Marco Carvalho. "Protecting Me." *IEEE Security & Privacy* 12, no. 1 (2014): 80-82. https://doi.org/10.1109/MSP.2014.10
[6] Hajduk, Vladimír, Martin Broda, Ondrej Kováč, and Dušan Levický. "Image steganography with using QR code and cryptography." In *2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 350-353. IEEE, 2016. https://doi.org/10.1109/RADIOELEK.2016.7477370
[7] Ali, Ammar Mohammed, and Alaa Kadhim Farhan. "Enhancement of QR code capacity by encrypted lossless compression technology for verification of secure E-Document." *IEEE Access* 8 (2020): 27448-27458. https://doi.org/10.1109/ACCESS.2020.2971779
[8] *Data Compression*. Springer London, 2007.
[9] Johnson, Neil F., and Sushil Jajodia. "Exploring steganography: Seeing the unseen." *Computer* 31, no. 2 (1998): 26-34. https://doi.org/10.1109/MC.1998.4655281

[10] Chen, Rongjun, Yue Huang, Kailin Lan, Jiawen Li, Yongqi Ren, Xianglei Hu, Leijun Wang, Huimin Zhao, and Xu Lu. "A fast adaptive binarization method for QR code images based on dynamic illumination equalization." *Electronics* 12, no. 19 (2023): 4134. https://doi.org/10.3390/electronics12194134
[11] Jagtap, Shilpa, and J. L. Mudegaonkar. "Steganographic Data Hiding in QR Codes." *AIJR Proceedings* (2021): 233-237. https://doi.org/10.21467/proceedings.118.38
[12] Ford, Richard, and Marco Carvalho. "Protecting Me." *IEEE Security & Privacy* 12, no. 1 (2014): 80-82. https://doi.org/10.1109/MSP.2014.10
[13] Arora, Mukesh, and Atul Kumar Verma. "Increase capacity of QR code using compression technique." In *2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, pp. 1-5. IEEE, 2018. https://doi.org/10.1109/ICRAIE.2018.8710429
[14] Reed IS, Solomon G. *Error-Correcting Codes*. Springer, 1998.
[15] Salomon, David. *A concise introduction to data compression*. Springer Science & Business Media, 2007. https://doi.org/10.1007/978-1-84800-072-8
[16] Panigrahi, Rasmita, and Neelamadhab Padhy. "An effective steganographic technique for hiding the image data using the LSB technique." *Cyber Security and Applications* 3 (2025): 100069. https://doi.org/10.1016/j.csa.2024.100069
[17] Koptyra, Katarzyna, and Marek R. Ogiela. "Steganography in QR Codes—Information Hiding with Suboptimal Segmentation." *Electronics* 13, no. 13 (2024): 2658. https://doi.org/10.3390/electronics13132658
[18] Sun, Yanfei, Mengyuan Yu, and Junyu Wang. "Research and development of QR code steganography based on JSteg algorithm in DCT domain." In *2020 IEEE 15th International Conference on Solid-State & Integrated Circuit Technology (ICSICT)*, pp. 1-4. IEEE, 2020. https://doi.org/10.1109/ICSICT49897.2020.9278285
[19] Koptyra, Katarzyna, and Marek R. Ogiela. "Steganography in QR Codes—Information Hiding with Suboptimal Segmentation." *Electronics* 13, no. 13 (2024): 2658. https://doi.org/10.3390/electronics13132658
[20] Koptyra, Katarzyna, and Marek R. Ogiela. "Information Hiding in QR Codes using Segment Manipulation." In *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 397-400. IEEE, 2024. https://doi.org/10.1109/PerComWorkshops59983.2024.10502885
[21] Mendhe, Abhijeet, Deepak Kumar Gupta, and Krishna Pal Sharma. "Secure QR-code based message sharing system using cryptography and steganography." In *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, pp. 188-191. IEEE, 2018. https://doi.org/10.1109/ICSCCC.2018.8703311
[22] Bhardwaj, Prince, Shubham Agrawal, Aniket Srivatsava, and Rajeswari Mukesh. "Enhancing QR Code Security: Authentication and Tamper Detection Using Visual Cryptography." In *2024 2nd World Conference on Communication & Computing (WCONF)*, pp. 1-7. IEEE, 2024. https://doi.org/10.1109/WCONF61366.2024.10692048
[23] Dantas, Pierre Vilar, Waldir Sabino da Silva Jr, Lucas Carvalho Cordeiro, and Celso Barbosa Carvalho. "A comprehensive review of model compression techniques in machine learning." *Applied Intelligence* 54, no. 22 (2024): 11804-11844. https://doi.org/10.1007/s10489-024-05747-w
[24] Aruna, P., L. Yamuna Devi, E. Danusri, V. Y. Pragathy, and Sri Jaya Vaishnavi KS. "Analysis on Text Compression Algorithms." In *2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pp. 430-437. IEEE, 2023. https://doi.org/10.1109/I-SMAC58438.2023.10290181
[25] Guntuboina, Venkata Sankirtana, and Nishtha Srivastava. "Efficient Image Data Compression Techniques: A Comprehensive Review and Comparative Study." (2024). https://doi.org/10.21203/rs.3.rs-4219237/v1
[26] Al-jawaherry, Marwa Adeeb, and Saja Younis Hamid. "Image Compression techniques: literature review." *Journal of Al-Qadisiyah for computer science and mathematics* 13, no. 4 (2021): Page-10. https://doi.org/10.29304/jqcm.2021.13.4.860
[27] Abdulyakeen, Abdulrasheed, and Yusuf Abdu Yusuf. "Social Media and Political Participation among Youth in South-Eastern Nigeria: A Case Study of 2015 and 2019 General Elections." *Acta Politica Polonica* 54 (2022): 147-173. https://doi.org/10.18276/ap.2022.54-10
[28] Kushwaha, Ram Singh, Sandip Kumar Mishra, Anjita Srivastava, And Priya Tiwari. *Research Methodology*. 1st ed. Noble Science Press International Publishing, 2024.
[29] Hassan, Arooj, Sabeen Hussain Bhatti, Sobia Shujaat, and Yujong Hwang. "To adopt or not to adopt? The determinants of cloud computing adoption in information technology sector." *Decision Analytics Journal* 5 (2022): 100138. https://doi.org/10.1016/j.dajour.2022.100138
[30] FullStory, "Qualitative vs. quantitative data: What's the difference?," FullStory.
[31] Denson, Thomas F., Craig A. Anderson, Thomas F. Denson, and Craig A. Anderson. "Experimental Methods." *The Cambridge Handbook of Research Methods and Statistics for the Social and Behavioral Sciences (1st ed., pp. 333–356). Cambridge University Press. https://doi.org/10.1017/9781009010054.017*

[32] Papadimitriou, Eleonora, George Yannis, Frits Bijleveld, and João L. Cardoso. "Exposure data and risk indicators for safety performance assessment in Europe." *Accident Analysis & Prevention* 60 (2013): 371-383. https://doi.org/10.1016/j.aap.2013.04.040

[33] Apritasari, Yaseri Dahlia, Iwan Sudrajat, and Surjamanto Wonorahardjo. "Experimental Research with Computer Simulation (Case Study Of Urban Cool Island)." *International Journal of Built Environment and Scientific Research* 7, no. 1 (2023): 41-50. https://doi.org/10.24853/ijbesr.7.1.41-50

[34] Skidmore, Susan. "Experimental Design and Some Threats to Experimental Validity: A Primer." *Online Submission* (2008).

[35] Ramachandran, Kandethody M, and Chris P Tsokos. "Design of Experiments." *Elsevier EBooks*, 2015, 459–94. https://doi.org/10.1016/B978-0-12-417113-8.00009-6

[36] Antony, Jiju. "Fundamentals of Design of Experiments." *Elsevier EBooks*, 2003, 6–16. https://doi.org/10.1016/B978-075064709-0/50003-X

[37] Cleveland, Gareth. "A Monte Carlo Simulation Study of the Performance of Hypothesis Tests Under Assumption Violations." (2013).

[38] Casella, George, and Roger L. Berger. "Hypothesis Testing." *An Article for the International Encyclopedia of the Social and Behavioral Sciences.* (1999).

[39] Groner, Rudolf, and Beat Keller. "Hypothesis testing strategies and instruction." *Cognitive psychology and instruction* (1978): 309-319. https://doi.org/10.1007/978-1-4684-2535-2_28

[40] Abas, Azizi, Yuhanis Yusof, and Farzana Kabir Ahmad. "Expanding the data capacity of QR codes using multiple compression algorithms and base64 encode/decode." *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 9, no. 2-2 (2017): 41-47.

[41] Victor, Nancy. "Enhancing the data capacity of QR codes by compressing the data before generation." *International Journal of Computer Applications* 60, no. 2 (2012): 17-21. https://doi.org/10.5120/9663-1104

[42] Bhardwaj, Cheshtaa, and Hitendra Garg. "An Approach for Enhancing Data Storage Capacity in Quick Response Code using Zip Compression Technique." In *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, pp. 520-524. IEEE, 2023. https://doi.org/10.1109/AISC56616.2023.10085559

[43] Yusuf, Bakhtiar, Takashi Kato, Mime Hashimoto, Takahiro Takeda, Kenji Iwasaki, Hiroko Sugatani, and Naoyuki Kubota. "Multicolor and multiple QR Code based information support system during disaster for elderly people." In *2015 International Conference on Informatics, Electronics & Vision (ICIEV)*, pp. 1-6. IEEE, 2015. https://doi.org/10.1109/ICIEV.2015.7333991

[44] Blasinski, Henryk, Orhan Bulan, and Gaurav Sharma. "Per-colorant-channel color barcodes for mobile applications: An interference cancellation framework." *IEEE Transactions on Image Processing* 22, no. 4 (2012): 1498-1511. https://doi.org/10.1109/TIP.2012.2233483

[45] Kumar, Kapil, Prateek Sharma, and Ajay Kumar Singh. "Configuring the system to share internet from single user to multi-user with single internet dongle." *Int. J. Soft Comput. Eng* 4 (2012): 32-35.

[46] Abas, Azizi, Yuhanis Yusof, and Farzana Kabir. "Improving data capacity of qr code version 40 using multiplexing and multilayered techniques: Embedding text based short story in qr code." *Advanced Science Letters* 22, no. 10 (2016): 2841-2846. https://doi.org/10.1166/asl.2016.7098