August 2023

PhD Dissertation

A Cryptographic Approach towards Privacy-Preserving Image Data Transmission, Storage and Computation

Graduate School of Chosun University

Department of Computer Engineering

IJAZ AHMAD

A Cryptographic Approach towards Privacy-Preserving Image Data Transmission, Storage and Computation

개인정보 보호를 위한 이미지 데이터 전송, 저장, 연산에 대한 암호화 접근법

August 25, 2023

Graduate School of Chosun University

Department of Computer Engineering

IJAZ AHMAD

A Cryptographic Approach towards Privacy-Preserving Image Data Transmission, Storage and Computation

Advisor: Prof. Seokjoo Shin

A dissertation submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

April 2023

Graduate School of Chosun University

Department of Computer Engineering

IJAZ AHMAD

AHMAD IJAZ 의 박사학위논문을 인준함

위원장	조선대학교	교수	모상만 (인)
위 원	조선대학교	교수	강문수 (인)
위 원	조선대학교	부교수	최우열 (인)
위 원	ETRI	부장	이성원 (인)
위 원	조선대학교	교수	신석주 (인)

2023 년 6 월

조선대학교 대학원

To my family – all 0b10110 of them.

TABLE OF CONTENTS

ABBREVIATIONS

AES	Advanced Encryption Standard
AI	Artificial Intelligence
BD	Bjøntegaard delta measure
CNN	Convolutional Neural Network
COVID-19	Coronavirus Disease 2019
CPE	Compressible Perceptual Encryption
CRT	Chinese Remainder Theorem
CtE	Compression-then-Encryption
CXR	Chest X-ray
DL	Deep Learning
DCT	Discrete Cosine Transform
DES	Data Encryption Standard
DP	Differential Privacy
EtC	Encryption-then-Compression
FL	Federated Learning
FN	False Negative
FP	False Positive
JPEG	Joint Photographic Experts Group
LE	Learnable Encryption
ML	Machine Learning
MS–SSIM	Multiscale Structural Similarity Index Measure
PE	Perceptual Encryption
PPDL	Privacy-Preserving Deep Learning
PPML	Privacy-Preserving Machine Learning

PSNR	Peak-Signal-to-Noise Ratio
RD	Rate Distortion
ROI	Region-of-Interest
SE	Selective Encryption
SGD	Stochastic Gradient Descent
SSIM	Structural Similarity Index Measure
SVM	Support Vector Machine
TB	Tuberculosis
TN	True Negative
TP	True Positive

LIST OF ALGORITHMS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

A Cryptographic Approach towards Privacy-Preserving Image Data Transmission, Storage and Computation

Ijaz Ahmad
Advisor: Prof. Seokjoo Shin, Ph.D.
Department of Computer Engineering
Graduate School of Chosun University

In today′s world, most of the automated applications ranging from the health sector to the entertainment industry are driven by Artificial Intelligence (AI), owing to the success of deep learning (DL) algorithms. There are two main challenges in the development and implementation of DL-based solutions. First, DL algorithms are characterized as compute-intensive tasks, and their training requires innovative technology and high computational resources. Second, training DL models for a particular task requires a large volume of sample data, which in some domains such as in the field of medical image analysis, is expensive and difficult to acquire. To overcome these limitations, cloud services such as computing, and storage resources are emerging as one of the cost-effective solutions. For example, in the first case, organizations can avail cloud-computing services to access the latest technology to speed-up the training process and allow DL models to scale efficiently with a lower capital cost. Similarly, to mitigate the data deficiency challenge, an organization can benefit from a community cloud, where services are shared by organizations with common interests to achieve their goals. In this case, cloud storage services can be utilized as a shared data repository for joint projects and collaboration among the organizations. Nonetheless, like all

communication systems, when data is outsourced to the avail of cloud services, there is a risk of information leakage, which can lead to privacy concerns. A straightforward solution to this is the encryption of data before transmission and for which the full encryption algorithms based on the number theory and chaos theory are proven to be the most secure techniques. Though this guarantees security during transmission, it is necessary to decrypt the data prior to performing any computations on them. This data reveal may be tolerable in certain scenarios; however, when dealing with privacy-sensitive data such as medical images, surveillance data, financial data, etc., such encryption techniques are not adequate to cater to the requirements of privacy preserving computation. In addition, when transmitting large volumes of data (especially image data), compression is necessary to efficiently utilize the available limited bandwidth. On the other hand, techniques specifically proposed to enable computation in the encryption domain have their associated computational cost, communication overhead and specialized design requirement that may reduce data utility and degrade the DL model performance. Therefore, privacy-preserving techniques that can jointly satisfy the dual requirements of data transmission, data storage and computation in the encryption domain are of immense importance.

In this work, we first investigate the order of performing compression and encryption processes that gives a better trade-off between compression savings and encryption efficiency and measure its impact on the downstream application performance. Next, we present a detailed taxonomy and comprehensive analysis of the JPEG compatible perceptual encryption methods in terms of their encryption and compression efficiencies. We adapt the assorted practices that have been proposed to effectively manage the encryption and compression trade-off, into a uniform framework, which may serve as a guideline for selecting appropriate techniques according to the privacy-preserving system requirements. To find proper trade-offs between achieving necessary privacy-preservation (during transmission and computation), preserving compression savings and downstream application accuracy, we present a novel transformation function to overcome the limitations of the perceptual encryption methods. In our proposed end-to-end system pipeline for

privacy-preserving computation, the compression block introduces certain information loss that may degrade the model accuracy, we propose a novel noise-based data augmentation technique to mitigate the impact of the compression artifacts on the trained DL model performance. To validate the usefulness of the proposed method, we consider a wide range of privacy-preserving applications such as privacy-preserving face recognition, privacy-preserving natural image classification and privacy-preserving COVID-19 detection in Chest X-ray images.

Our simulation results show that the proposed simultaneous image encryption and compression scheme for secure and efficient data transmission and/or storage, preserves the lossless compression saving, and with our data-to-symbol mapping function the compression saving is improved on average from 6% to 15%. On the other hand, in the privacy-preserving computation domain, the proposed PE-based scheme at best introduces a decrease of ≈5% in the prediction accuracy of a DL model for natural image classification task while ≈3% drop in the model′s accuracy and sensitivity scores for medical image analysis. In the face recognition application, the proposed privacy preservation scheme delivers the same recognition accuracy as that of the plain images. Moreover, the proposed noise-based augmentation method has reduced the difference in model accuracy from 11% to 2% for classification of natural images.

요약

개인정보 보호를 위한 이미지 데이터 전송, 저장, 연산에 대한 암호화 접근법

아흐마드 이자즈
지도교수: 신석주
컴퓨터공학과
조선대학교

딥러닝(DL) 알고리즘의 발전이 인공지능(AI) 기반 자동화 애플리케이션의 확산을 이끌고 있다는 사실은 보건부터 엔터테인먼트까지 넓은 범위에서 목격된다. 그러나, DL 기반 솔루션을 개발하고 구현하는 데는 크게 두 가지 주요 장애물이 있다. 첫째로, DL 알고리즘이 매우 연산 집약적이며, 그 학습 과정은 혁신적인 기술과 상당한 계산 자원을 필요로 한다. 둘째로, DL 모델의 학습을 위해선 풍부한 샘플 데이터가 요구되는데, 특히 의료 이미지 분석 등 일부 분야에서는 비용 문제와 확보의 어려움이 동시에 발생한다.

이러한 한계를 극복하기 위한 방법 중 하나로 클라우드 서비스, 특히 컴퓨팅과 스토리지 자원이 주목 받고 있다. 예를 들어, 딥러닝 기술을 사용하는 조직들은 클라우드 컴퓨팅 서비스를 이용하여 최신 기술에 접근하고, 학습 과정을 가속화하며, DL 모델을 보다

저렴한 비용으로 효율적으로 확장할 수 있다. 또한, 데이터 부족 문제 해결이라는 공동의 목표를 위해 서비스를 공유하는 커뮤니티 클라우드의 이점을 활용할 수 있다. 이 경우, 클라우드 스토리지 서비스는 조직 간의 공동 프로젝트 및 협업을 위한 공유 데이터 저장소 역할을 할 수 있다.

그러나, 클라우드 서비스를 이용하여 데이터를 아웃소싱할 때는 데이터 유출 위험이 있으며, 이는 개인정보 보호 문제로 이어질 수 있다. 이 문제를 간단히 해결하는 방법은 데이터를 전송하기 전에 암호화하는 것이다. 정수론과 혼돈 이론에 기반한 완전 암호화 알고리즘이 가장 안전하다고 알려져 있다. 이 방법은 데이터의 보안성을 보장하지만, 데이터를 처리하기 전에는 암호를 해독해야 한다. 이 방법은 일부 응용 시나리오에서는 적용 가능하지만, 의료 이미지나 감시 데이터, 재무 데이터 등 개인 정보에 민감한 데이터 처리의 경우 개인정보 보호 요구 사항을 만족시키기는 어렵다. 또한, 큰 데이터(특히 이미지 데이터)를 전송하면서 제한된 대역폭을 효과적으로 활용하기 위해 데이터 압축이 필요하다. 한편으로, 암호화 영역에서 계산을 가능하게 하는 PPDL 기술은 연산 비용, 통신 오버헤드 및 특수 설계 요구 사항 등으로 인해 데이터 유틸리티를 줄이고 DL 모델의 성능을 저하시킬 수 있다. 따라서, 데이터 전송과 저장, 그리고 연산에 대한 보안 요구를 모두 충족시킬 수 있는 개인정보 보호 기술은 매우 중요한 연구 주제라 할 수 있다.

본 연구에서는 먼저 압축 절약과 암호화 효율성 사이의 더 나은 균형을 찾기 위해 압축과 암호화 과정의 수행 순서를 조사하고, 이것이 다운스트림 애플리케이션의 성능에

어떠한 영향을 미치는지 측정하였다. 다음으로, JPEG 호환 지각 암호화 방법에 대한 자세한 분류 체계와 포괄적인 분석을 제시하였으며, 이는 암호화와 압축 효율성 측면에서 고려되었다. 본 연구에서는 암호화와 압축의 균형을 효과적으로 관리하기 위해 제안된 다양한 방법을 표준 프레임워크로 적용하여, 개인정보 보호 시스템의 요구 사항에 따라 적절한 기술을 선택하는 지침이 될 수 있도록 지표들을 제시하였다. 필요한 개인정보 보호(전송 및 계산 중), 압축 절약의 보존, 그리고 다운스트림 애플리케이션의 정확도 사이의 적절한 균형을 찾기 위해, 연구에서는 지각 암호화 방법의 한계를 극복하는 새로운 변환 함수를 제시하였다. 연구에서 제안하는 개인정보 보호 연산을 위한 종단간 시스템 파이프라인에서는, 압축 블록이 모델 정확도를 저하시킬 수 있는 특정 정보의 손실을 초래하였다. 이에 대응하기 위해, 연구에서는 훈련된 DL 모델의 성능에 압축 아티팩트의 영향을 완화하는 새로운 노이즈 기반 데이터 확대 기술을 제안하였으며, 제안된 방법의 유용성을 확인하기 위해 흉부 X-선 이미지에서의 개인정보 보호 얼굴 인식, 개인정보 보호 자연 이미지 분류, 그리고 개인정보 보호 COVID-19 감지 등 다양한 개인정보 보호 애플리케이션을 고려하였다.

시뮬레이션 결과로부터, 제안된 동시 이미지 암호화 및 압축 기법이 안전하고 효율적인 데이터 전송 및 저장을 가능하게 하며, 무손실 압축 절약을 유지하고, 데이터 대 심볼 매핑 함수를 통해 압축 절약을 평균 6%에서 15%까지 향상시킨다는 것을 확인하였다. 또한, 제안된 PE 기반 방식을 사용하면, 개인정보 보호 연산 영역에서 자연 이미지 분류

작업에 대한 DL 모델의 예측 정확도는 약 5% 감소하는 반면, 의료 이미지 분석에 대한 모델의 정확도와 민감도 점수는 약 3% 감소함을 확인하였다.

I. INTRODUCTION

1.1. Preface

For the improved quality of life, deep learning-based solutions are widely adopted in a multitude of application domains such as computer vision, pattern recognition, natural language processing, and medical image analysis etc. For a well-performing DL model, large volume of sample data and high computation resources are needed. These requirements can be fulfilled by taking advantage of powerful infrastructures such as cloud-computing services, to avail high-powered computational resources, and cloud-storage services, for adopting collaborative learning. However, this comes with security and privacy concerns as there are potential risks of leakage of privacy-sensitive information associated with outsourcing the data. In addition, given the large volume of data, bandwidth and storage efficiencies should also be considered. Traditional privacy preservation approaches treat the requirements of data transmission and computation separately even though both are necessary to be fulfilled to fully reap the benefits of DL for data-driven applications. Hence, this necessitates efficient privacy-preserving techniques and solutions for many emerging applications. In this thesis, we aim to preserve user privacy by implementing privacy-preserving solutions that satisfy the dual requirements of data transmission, data storage and computation in the encryption domain, altogether.

1.2. Thesis Statement

The exchange of privacy-sensitive data to outsource the computation and/or storage requirements results in privacy concerns and thereby necessitates privacy preservations. Finding proper trade-offs between maintaining a desired level of compression savings, preserving necessary privacy while achieving acceptable downstream application performance by means of implementing suitable encryption and compression techniques, and avoiding (or at best, reducing) the potential negative impact of the visual degradation (because of applying either privacy preservation or compression technique) on the DL model performance.

1.3. Research Objectives and Questions

This thesis has the following objectives and their respective research questions. Our first objective (O1) is to design and analyze simultaneous image encryption and compression scheme for photo sharing and archiving purposes. Our second objective (O2) is to taxonomize and evaluate perceptual encryption techniques for encryption-then-compression systems. The third objective (O3) is to design and evaluate noise-based data augmentation method to mitigate the impact of the compression artifacts on a trained DL model performance, and finally, the fourth objective (O4) is to find better tradeoffs between compression savings, user privacy and dataset utility, and measure its impact on DL model performance in domain specific applications.

O1 **Design and evaluation of simultaneous image encryption and compression scheme for image data sharing and archiving.** Compression and encryption are often performed together for image sharing and/or storage and the order in which these two processes are coupled together affects the overall efficiency of digital image services. For example, encrypted data has less or no compressibility while it is challenging to ensure reasonable security without downgrading the compression performance. The problem lies in treating compression and encryption as two separate processes. There is a need to develop simultaneous image encryption and compression schemes to meet the dual requirements of image data transmission. We describe more about incorporating one requirement into another in Chapter 3 and Papers [1]–[4].

Research Question. How to incorporate compression requirement in an encryption algorithm?

O2 **Analysis of the state-of-the-art and taxonomy of perceptual encryption techniques for encryption-then-compression systems.** In general, when encryption is completed prior to compression then less or no redundancy is left for the compression algorithm to exploit in order to reduce the image size. Therefore, almost in all cases it is preferred to perform compression before encryption – compression-then-encryption (CtE). However, when a multimedia application such as photo-storage service has the requirement of preserving the image format after the compression and encryption processes, then it is beneficial to reverse the conventional order of CtE. In this direction, a new class of image encryption techniques called perceptual encryption (PE) algorithms is emerging, which provides a necessary level of security while preserving the compression savings of the JPEG algorithm. A tradeoff in these methods, however, is between security efficiency and compression savings due to the chosen block size. Several solutions such as the processing of each color component independently, and image representation have been proposed to effectively manage this tradeoff. There is a necessity to develop a detailed taxonomy of the PE methods and adapt these assorted practices into a uniform framework to provide a fair comparison of their results. To mitigate the weaknesses of existing PE methods and make them suitable for privacy-preserving applications, novel encryption transformation function is necessary. We present comprehensive analysis of the JPEG compatible PE methods in Chapter 4 and paper [5] and describe our proposed method in detail in Chapters 4 and 5 and papers [6]–[12].

Research Question. How to achieve a better trade-off between compression savings and encryption efficiency –specifically, during transmission and format-compatible storage?

O3 **Design and evaluation of noise-based augmentation method to mitigate the impact of the JPEG compression artifacts on a trained deep learning model performance.** Lossy image compression provides an efficient solution to the exchange and storage of large volumes of image data for various applications. The main design principle of a lossy compression algorithm is to discard visually insignificant information as much as possible while keeping the resulting visible artifacts at a minimum. These unperceivable defects significantly degrade the performance of a trained deep learning (DL) model. Because for an efficient model, it is necessary that the training and testing are performed against the data that come from the same target application distribution. The model generalization can be improved with the aid of data augmentation for which the existing techniques are not adequate as they do not take noise into consideration. There is a need to evaluate and design a data augmentation strategy that can enhance the quality of training datasets such that the model generalizes well on any future noisy images. Explanation of our proposed augmentation method is given in Chapter 5 and papers [7], [13], [14].

Research Questions. How to improve a DL model generalization in the presence of compression distortions? Importantly, how to mitigate the compression artifacts impact on a trained DL model performance without any pre-processing for efficient inference?

O4 **Finding better tradeoffs between compression savings, user privacy, and dataset utility, and measuring its impact on DL model performance in domain specific applications.** After satisfying the dual requirements of image data transmission by finding a proper tradeoff between compression and encryption efficiencies, we can measure the impact of perceptual encryption for preserving user privacy during computations mainly in terms of DL model performance on a downstream task. We describe this in different application contexts in Chapter 6 and papers [6], [9], [15]–[17].

Research Question. How to find a proper trade-off between preserving necessary user privacy and maintaining acceptable DL model performance?

1.4. Contributions

To address the aforementioned objectives and answer their associated research questions, the main contributions of this thesis are summarized below:

C1 *Compression.* **A novel partitioning method for data-to-symbol mapping.** We proposed a hybrid simultaneous image encryption and compression algorithm that incorporates the compression requirement of a data communication system into the encryption algorithm as part of our objective O1. Encryption is based on Chaos theory and is carried out in two steps, i.e., permutation and substitution. The lossless compression is performed on the shuffled image and then the compressed bitstream is grouped into 8-bit blocks for substitution stage. The lossless nature of the proposed scheme makes it suitable for medical image compression and encryption applications. To improve the performance of the entropy encoder of the compression algorithm, we propose a novel data-to-symbol mapping method based on Chinese remainder theorem to represent adjacent pixel values as a block. With such representation, the compression saving is improved on average from 5.76% to 15.45%.

C2 *Encryption.* **An analysis and taxonomy of compressible perceptual encryption methods.** Perceptual encryption hides identifiable information of an image in such a way that its intrinsic characteristics remain intact. This recognizable perceptual quality can be used to enable computer vision applications in the encryption domain. A class of PE algorithms based on block level processing has recently gained popularity for their ability to generate JPEG compressible cipher images. A tradeoff in these methods, however, is between security efficiency and compression savings due to the chosen block size. Several methods (such as the processing of each color component independently, image representation, and sub–block level processing) have been proposed to effectively manage this tradeoff. We present a taxonomy of these assorted practices and adapt them into a uniform framework to provide a fair comparison of their results (objective O2). Specifically, their compression quality is investigated under various design parameters such as the choice of colorspace, image representations, chroma subsampling, quantization tables, and block size. Also, their encryption quality is quantified in terms of several statistical analyses (Objective O4).

C3 *Encryption.* **A new geometric transformation function for perceptual encryption methods.** In the conventional PE methods, for better security, a larger number of blocks is achieved by decreasing the block size; therefore, the key size expansion is limited by the smallest allowable block size used in the compression algorithm. In addition, the PE extended methods that process each color component independently or represent an input color image as a pseudo grayscale image to achieve a larger number of blocks degrade image quality and compression savings, and remove color information, which limits their applications. To overcome these limitations, we propose inter and intra block processing that performs the encryption steps that only change correlation's direction on a sub-block level, thereby improving the encryption efficiency without compromising the compression performance. The intra block processing results in a new transformation –inside-out transformation function (Objective O2). Besides security, the main advantage of the proposed method is that it retained color information, which makes it suitable for privacy-preserving computations without the need of decryption (Objective O4).

C4 *Deep learning.* **A novel data augmentation technique.** Large volume of data is necessary for training an efficient DL model to avoid overfitting and to generalize well on the unseen data, which is difficult and expensive to acquire. Data augmentation is one of the effective solutions to this problem. We propose a novel noise-based data augmentation technique to enhance the quality of training datasets such that the model generalizes well on the noisy images in future (Objective O3). Specifically, we consider the JPEG distortions to generate new images. The main advantages of the proposed method are that it does not require artifact correction as a preprocessing step and can preserve the model performance on the uncompressed images. Simulation results show that the proposed technique mitigates the impact of the compression artifacts on the trained DL model performance and on heavily compressed images, the accuracy difference is reduced from 11% to 2% for classification of natural images and 6% to 1% for COVID-19 detection in CXR images. In addition, for larger resolution images the model is immune against noise but with the proposed method there is a 2% gain in the model's performance.

C5 *Applications.* **Privacy-preserving image data transmission, format-compatible storage, and computation applications of perceptual encryption techniques.** We first extended the applications of PE schemes to the privacy preserving computation domain. Given their limitations, we then proposed a method to improve their security efficiency without compromising compression savings and DL model accuracy (Objective O4). For natural images classification task, our proposed PE-based privacy preserving scheme at best introduces a decrease of ≈5% in the prediction accuracy of the trained models (Objective O4). For face recognition application, the proposed privacy preserving scheme delivers the same recognition accuracy as that of the plain images (Objective O4). For COVID-19 screening in CXR images, the proposed PE-based privacy preserving scheme at best introduces a ≈3% drop in the model's accuracy and sensitivity scores (Objective O4). On the other hand, the JPEG compression performance on the cipher images is reduced ≈3% at best (Objectives O2 and O4).

1.5. Publications and Thesis Outline

The core chapters of this thesis are derived from research papers that are already published in journals and conferences during my Ph.D. candidature in collaboration with my Ph.D. supervisor and other collaborators. This thesis uses a standard pronoun of "We" for reporting collaborative research; however, I have carried out the main research activities such as conceptualization, methodology, software development, formal analysis, investigation, data curation, writing--original draft preparation and visualization. The thesis chapters along with their corresponding publications are given below:

Chapter 1 – This chapter describes the thesis statement, research objective and their associated research questions, the contributions of the thesis, and the thesis outline along with the publications.

Chapter 2 – This chapter provides the preliminaries necessary for the foundation of this thesis, and the evaluation metrics that have been used throughout this thesis.

Chapter 3 – This chapter addresses the research question associated with O1 by designing a simultaneous image encryption-compression technique and proposed a novel data-to-symbol mapping technique based on Chinese Remainder Theorem. The optimal parameters selection and the analysis presented in this chapter have been published in:

P1 **I. Ahmad** and S. Shin, *A novel hybrid image encryption–compression scheme by combining chaos theory and number theory*, Signal Processing: Image Communication, vol. 98, p. 116418, Oct. 2021, Elsevier.

P2 **I. Ahmad**, B. Lee and S. Shin, *Analysis of Chinese Remainder Theorem for Data Compression,* The 34th International Conference on Information Networking (ICOIN), IEEE, Barcelona, Spain, Jan 7-10, 2020. (Poster Presentation).

P3 **I. Ahmad** and S. Shin, *Analysis of Chinese Remainder Theorem Moduli for Image Compression,* KICS Fall Conference 2019, KICS, Seoul, Korea, Nov 16, 2019. (Oral Presentation).

P4 **I. Ahmad** and S. Shin, *Performance analysis of Chinese Remainder Theorem for Data Compression*, Korea Computing Conference (KCC), KIISE, Virtual, Jul 2-4, 2020. (Oral Presentation).

Chapter 4 – This chapter addresses the research question associated with O2. The chapter first develop a detailed taxonomy of the conventional perceptual encryption methods and adapts the assorted practices, into a uniform framework, that have been proposed to find a proper tradeoff between compression savings and encryption efficiency of encryption-then-compression schemes. The chapter then identifies the weaknesses in the existing perceptual encryption methods, and towards which proposes a novel transformation function. The analysis presented in this chapter have been published in:

P5 **I. Ahmad**, W. Choi and S. Shin, *Comprehensive Analysis of Compressible Perceptual Encryption Methods – Compression and Encryption Perspectives*, Sensors, vol. 23, no. 8, p. 4057, Apr. 2023, MDPI.

P6 **I. Ahmad** and S. Shin, *IIB–CPE: Inter and Intra Block Processing-Based Compressible Perceptual Encryption Method for Privacy-Preserving Deep Learning*, Sensors, vol. 22, no.

20, p. 8074, Oct. 2022, MDPI.

P7 **I. Ahmad** and S. Shin, *Effect of Inter and Intra Block-level shuffling on the JPEG Compression Performance,* Summer Workshop on Computer Communication (SWCC), KIISE, Virtual, Aug 25, 2021. (Oral Presentation).

P8 **I. Ahmad** and S. Shin, *Perceptual Encryption-based Privacy-Preserving Deep Learning in Internet-of-Things Applications*, The 13th International Conference on Information and Communication Technology Convergence (ICTC), IEEE, Jeju, Korea, Oct 19-21, 2022. (Oral Presentation).

Chapter 5 – This chapter addresses the research questions associated with objectives O2 and O3. The chapter first extends the applications of proposed Perceptual Encryption method to cloud-based medical image analysis and presents a novel noise-based data augmentation technique to make the DL model robust against the compression artifacts. The applications and analysis are published in:

P9 **I. Ahmad** and S. Shin, *A Perceptual Encryption-Based Image Communication System for Deep Learning-Based Tuberculosis Diagnosis Using Healthcare Cloud Services*, Electronics, vol. 11, no. 16, p. 2514, Aug. 2022, MDPI.

P10 **I. Ahmad** and S. Shin, *Encryption-then-Compression System for Cloud-based Medical Image Services*, The 36th International Conference on Information Networking (ICOIN), IEEE, Jeju, Korea, Jan 12-15, 2022. (Oral Presentation) **[Best Paper Award]**.

P11 **I. Ahmad** and S. Shin, *Noise-cuts-Noise Approach for Mitigating the JPEG Distortions in Deep Learning,* The 5th International Conference on Artificial Intelligence in Information and Communication (ICAIIC), IEEE, Bali, Indonesia, Feb 20-23, 2023. (Oral Presentation).

P12 **I. Ahmad** and S. Shin, *Quantitative Assessment of the Impact of Lossy JPEG Compression on Deep Learning Models,* The 8th International Conference on Next Generation Computing (ICNGC), KINGPC, Jeju, Korea, Oct 6-8, 2022. (Poster Presentation).

P13 **I. Ahmad** and S. Shin, *A Comparison of EfficientNets for Tuberculosis Detection in Chest Radiographs,* The 3rd Korea Artificial Intelligence Conference, KICS, Jeju, Korea, Sep 28-30, 2022. (Oral Presentation).

P14 **I. Ahmad** and S. Shin. "Leveraging Transfer Learning in EfficientNetv2-based Tuberculosis Detection." Fall Conference, KICS, Geyeongju, Korea, Nov 16-18, 2022. (Oral Presentation).

Chapter 6 – This chapter addresses research question associated with O4 by considering Perceptual Encryption-based Privacy Preserving Deep Learning for various applications such natural image classification, face recognition and medical image analysis in the encryption domain. Thereby preserves user privacy during computation. The applications presented in this chapter appeared in P6 and P8, and in the following papers:

P15 **I. Ahmad**, S. Hwang, E. Kim, and S. Shin, *Privacy-Preserving Surveillance for Smart Cities,* 13th International Conference on Ubiquitous and Future Networks (ICUFN), IEEE, Barcelona, Spain, Jul 5-8, 2022. (Oral Presentation).

P16 **I. Ahmad** and S. Shin, *Perceptual Encryption-based Privacy-Preserving Deep Learning for Medical Image Analysis,* The 37th International Conference on Information Networking (ICOIN), IEEE, Bangkok, Thailand, Jan 11-14, 2023. (Oral Presentation) **[Best Paper Award]**.

P17 **I. Ahmad** and S. Shin, *Block-based Perceptual Encryption Algorithm with Improved Color Components Scrambling*, Spring Conference, KINGPC, Jeju, Korea, May 19-21, 2022. (Oral Presentation).

Chapter 7 – This chapter summarizes the thesis by stating its main key findings and outlines possible future research directions.

1.6. Other Publications

Besides the above publications, I have contributed to the following publications as first author or as a co-author, and inventor of one patent:

P18 Korean Patent Application No. 10-2021-0174309, *Apparatus and Method for Encrypting and Compressing Image*, (filing date 08/12/2021).

P19 **I. Ahmad**, N. Islam and S. Shin, *Performance Analysis of Cloud-based Deep Learning Models on Images Recovered without Channel Correction in OFDM System*, The 27th Asia-Pacific Conference on Communications (APCC), IEEE, Jeju, Korea, Oct 19-21, 2022. (Oral Presentation).

P20 **I. Ahmad**, S. Hwang and S. Shin, *Determining Jigsaw Puzzle State from an Image Based on Deep Learning*, The 4th International Conference on Artificial Intelligence in Information and Communication (ICAIIC), IEEE, Jeju, Korea, Apr 21-24, 2022. (Oral Presentation).

P21 **I. Ahmad** and S. Shin, *Fine-Tuning Pre-Trained Deep Learning Models for Multiclass Grayscale Images Classification*, The 7th International Conference on Next Generation Computing (ICNGC), KINGPC, Jeju, Korea, Nov 4-6, 2021. (Poster Presentation).

P22 **I. Ahmad** and S. Shin, *An Approach to Run Pre-Trained Deep Learning Models on Grayscale Images*, The 3rd International Conference on Artificial Intelligence in Information and Communication (ICAIIC), IEEE, Jeju, Korea, Apr 13-16, 2021. (Oral Presentation).

P23 **I. Ahmad** and S. Shin, *Region-based Selective Compression and Selective Encryption of Medical Images*, The 9th International Conference on Smart Media and Applications (SMA), ACM, Jeju, Korea, Sep 17-19, 2020. (Oral Presentation) **[Best Paper Award]**.

P24 **I. Ahmad** and S. Shin, *Just-Noticeable-Difference Based Edge Map Quality Measure*, The

4th International Conference on Next Generation Computing (ICNGC), KINGPC, Vung Tau, Vietnam, Dec 20-22, 2018. (Poster Presentation) **[Best Poster Award]**.

P25 **I. Ahmad** and S. Shin, *Convolutional Autoencoder for Image Quality Assessment*, Summer Conference, KICS, Jeju, Korea, Jun 21-24, 2023. (Oral Presentation).

P26 **I. Ahmad**, N. Islam and S. Shin, *Performance Analysis of Deep Learning Models with Communication Channel Distortion*, Summer Conference, KICS, Jeju, Korea, Jun 22-24, 2022. (Oral Presentation).

P27 **I. Ahmad** and S. Shin, *A pixel-based Encryption Method for Privacy-Preserving Deep Learning Models*, Winter Conference, KICS, Pyeongchang, Korea, Feb 9-11, 2022. (Oral Presentation).

P28 **I. Ahmad** and S. Shin, *Optimal Batch Size for Fine-Tuning Pre-Trained Deep Learning Models*, The 2nd Korea Artificial Intelligence Conference, KICS, Jeju, Korea, Sep 29- Oct 1, 2021. (Oral Presentation).

P29 **I. Ahmad** and S. Shin, *Optimal Resolution Selection to Run Pre-Trained Deep Learning Models on Tiny Images*, Spring Conference, KINGPC, Gwangju, Korea, May 13-14, 2021. (Poster Presentation) **[Best Poster Award]**.

P30 **I. Ahmad** and S. Shin, *Data Rate of End-to-End Learning of Communication Systems: A Survey,* 2020 Summer Conference, KINGPC, Jeju, Korea, Aug 22, 2020. (Oral Presentation)

P31 **I. Ahmad** and S. Shin, *Channel Model for End-to-End Learning of Communications Systems: A Survey,* 2020 Spring Conference, KISM, Gwangju, Korea, May 22-23, 2020. (Oral Presentation).

P32 **I. Ahmad**, N. Sahar, and S. Shin, *Algorithmic Design of Korean Dancheong Patterns*, The 9th Workshop on Convergent and Smart Media Systems (CSMS), KISM, Muju, Korea, Jan 27-29, 2019. (Oral Presentation).

P33 N. Islam, **I. Ahmad** and S. Shin, *Robustness of Deep Learning Enabled IoT Applications Utilizing Higher Order QAM in OFDM Image Communication System*, The 5th International Conference on Artificial Intelligence in Information and Communication (ICAIIC), IEEE, Bali, Indonesia, Feb 20-23, 2023. (Oral Presentation).

II. BACKGROUND

2.1. Chinese Remainder Theorem

Chinese remainder theorem (CRT) states that if moduli m_k are pairwise relative prime, then for any given integers n_k, the system of congruence

$$x \equiv n_i \ \ mod \ \ m_i, i = 1, 2, \cdots, k, \tag{1}$$

has a unique solution x in the product of their moduli as,

$$x \equiv \sum_{i=1}^{k} n_i M_i N_i \ \text{mod} \ M\,, \tag{2}$$

Given that $min(m_k) > max(n_k)$, k is the number of congruence equations, product of moduli m_k is $M = \ \prod_{i=1}^{k} m_i$, $M_i = \frac{M}{m_i}$ and $N_i \equiv M_i^{-1} \text{mod} \ m_i$.

The integers n_k can be recovered from (2) by taking modulus of x with the corresponding m_k as,

$$n_i \equiv x \ \ mod \ \ m_i, \tag{3}$$

2.2. 2D Hyper-Chaotic System

Due to excellent properties of chaotic maps such as ergodicity, sensitivity to initial conditions, system parameters space and pseudo-randomness make them suitable for image encryption systems. Hyper-chaos is a special phenomenon of chaos where instability occurs in more than one direction. In our proposed scheme, we have used 2D hyper-chaos discrete nonlinear dynamic system to shuffle the input image. The system is given by [18]

$$\begin{cases} x_{i+1} = & f(x_i, y_i) \\ y_{i+1} = & g(x_i, y_i)\text{’} \end{cases} \tag{4}$$

where,

$$\begin{cases} f(x_i, y_i) = & a_1 + a_2 x_i + a_3 x_i^2 + a_4 y_i + a_5 y_i^2 + a_6 x_i y_i \\ g(x_i, y_i) = & b_1 + b_2 x_i + b_3 x_i^2 + b_4 y_i + b_5 y_i^2 + b_6 x_i y_i \end{cases}, \tag{5}$$

Based on (5), we have derived the following 2D hyper-chaos with discrete nonlinear dynamic system for the control parameters a and b,

$$\begin{array}{rl} x_{i+1} = & a_1 + a_2 x_i + a_4 y_i \\ y_{i+1} = & b_1 + b_3 x_i^2 \end{array}, \tag{6}$$

where x, y are state variables and a_1, a_2, a_4, b_1, b_3 are control parameters. We have set the initial values $x_0 = 0.0394$ and $y_0 = 0.0001$. When $a_1 = 0.2; a_2 = 0.3; a_4 = 0.5; b_1 = -1.7; b_3 = 3.7$, the system is hyper-chaotic. The Lyapunov exponents of the system are 0.161 and 0.095 for x_{i+1} and y_{i+1}, respectively.

2.3. Logistic Map

We have used the 1-D chaotic logistic map in the substitution step. It has been shown by [19] that logistic maps have lower complexity than Chebyshev maps and Lorenz system. It is given by

$$d_{i+1} = \mu d_i (1 - d_i), \tag{7}$$

where, μ is the control parameter. The system is chaotic for $\mu \in [3.57,4]$. Both parameter μ and the initial value d_0 serve as the key for substitution stage. The logistic map has Lyapunov exponent 0.0012 and 0.0693 for $\mu = 3.57$ and $\mu = 4$, respectively.

2.4. Lossless Compression and Information Theory

An information source consists of a set S with finite number of unique symbols called the source alphabet. It is capable of generating a sequence of symbols $\{x_1, x_2, \cdots x_M\}$ drawn from the source alphabet. The probability that output symbol x_j is drawn from S is $p_j = P(x_j), 0 \leq j \leq M - 1$. The associated information (self-information) with a symbol is

$$I_{x_j} = -\log_2(p_j). \tag{8}$$

The source entropy $\mathrm{H}(S)$ measures the information content of the source in terms of the average amount of information per symbol. The $\mathrm{H}(S)$ of a source is given by:

$$H(S) = -\sum_{j=1}^{M} p_j \log_2(p_j). \tag{9}$$

For a lossless code C assigned to a code symbol s_j, the average bit rate is $B_C = p_j l_j$ (bits per symbol), where l_j is the length of code word assigned to a symbol s_j. For C to be uniquely decodable, the lower bound on the average bit rate B_C is the source entropy $\mathrm{H}(S)$ i.e. $\mathrm{H(S)} \leq B_C$. When using Huffman coding as the entropy encoder in the final stage of lossless compression then a uniquely decodable prefix code C can be constructed as: $\mathrm{H(S)} \leq B_C \leq H(S) + 1$. The upper limit on the average bit rate can be improved upon, when the most probable symbol has the probability much less than 1 [20]. Then, the limit on Huffman coding can be defined as:

$$B_C < \begin{cases} H(S) + P_{max}, & P_{max} < 0.5 \\ H(S) + P_{max} + 0.086, & P_{max} \geq 0.5 \end{cases}, \tag{10}$$

where P_{max} is the probability of the most probable symbol. One way to achieve the lower limit on B_C in (10) is by extending the alphabet size. This can be done by grouping adjacent data symbols into blocks, and each block is treated as a symbol. In addition, the block representation enables Huffman coding to assign non-integer-length codeword to each symbol separately. Also, for larger block size (Q) the average bitrate monotonically reaches the source entropy as

$$H(S) \leq B_C \leq H(S) + \frac{1}{Q}. \tag{11}$$

In our proposed scheme, we have used Chinese remainder theorem (CRT) to represent adjacent data elements as a single CRT solution.

2.5. The JPEG Standard

The JPEG compression standard [21] is one of the most widely used image formats. A block diagram of the JPEG algorithm is illustrated in Figure 1. The JPEG compression and decompression procedures can be described in the following steps.

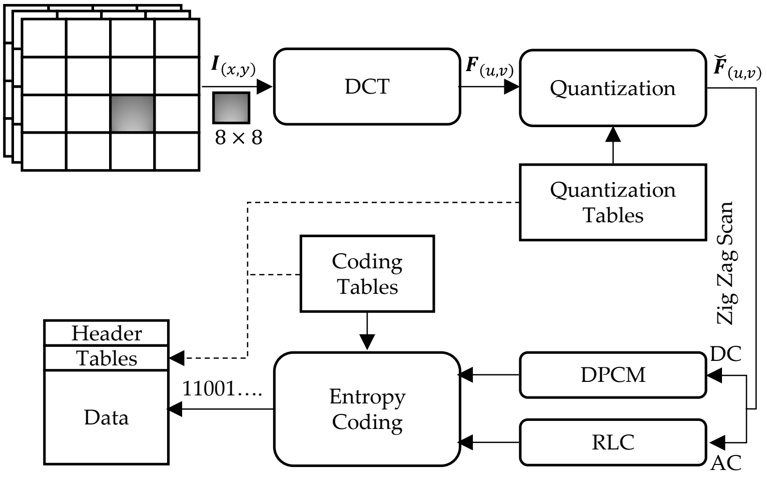


Figure 1. Illustration of the JPEG compression algorithm.

Step 1. Colorspace Conversion

In the first step, the luminance component of an input image is separated from its color component, which is necessary to achieve more compression savings. The human visual system (HVS) is less sensitive to color than the image luminosity; therefore, the JPEG algorithm represents color component in a smaller resolution thus, achieves more savings [21]. This process is called color– or chroma–subsampling. The ratio for chroma–subsampling depends on the application requirements; however, the most commonly used ratios are 4:2:2 (half of the color) and 4:2:0 (quarter of the color). The image luminance component (**Y**) can be separated from the image color components (**Cb** and **Cr**) by a colorspace conversion function defined as:

$$\begin{aligned} \boldsymbol{Y} &= \quad 0.3 \times \boldsymbol{R} + (0.59 \times \boldsymbol{G}) + (0.11 \times \boldsymbol{B}) \\ \boldsymbol{C_b} &= 128 - (0.17 \times \boldsymbol{R}) - (0.33 \times \boldsymbol{G}) + (0.5 \times \boldsymbol{B}), \\ \boldsymbol{C_r} &= 128 + (0.5 \times \boldsymbol{R}) - (0.42 \times \boldsymbol{G}) - (0.08 \times \boldsymbol{B}) \end{aligned} \tag{12}$$

where, $\boldsymbol{R}$ is Red, $\boldsymbol{G}$ is Green and $\boldsymbol{B}$ is Blue color channels of the image. The (12) converts an image from the RGB colorspace to YCbCr colorspace. During decoding, an inverse operation is performed that converts back the YCbCr image to RGB image, and this operation is defined as:

$$\begin{aligned} \boldsymbol{R} &= \boldsymbol{Y} + 1.40 \times (\boldsymbol{C_r} - 128) \\ \boldsymbol{G} &= \boldsymbol{Y} - 0.34 \times (\boldsymbol{C_b} - 128) - 0.71 \times (\boldsymbol{C_r} - 128). \\ \boldsymbol{B} &= \boldsymbol{Y} + 1.77 \times (\boldsymbol{C_b} - 128) \end{aligned} \tag{13}$$

Note that when chroma–subsampling is performed during compression, then it is necessary to up sample the color components before the YCbCr to RGB conversion function during decompression to recover the full resolution image.

Step 2. Discrete Cosine Transformation (DCT)

The YCbCr image is divided into non–overlapping blocks and each block is then transformed using the DCT function [22]. The goal here is to represent a large amount of information from a few data samples by exploiting the correlations among the adjacent pixels. In natural images, the pixels are usually high correlated up to 8 pixels neighbors in either direction [1]. Therefore, in the JPEG standard, a block size of 8×8 is used. The forward DCT function on an image block $\boldsymbol{B}$ is defined as

$$\boldsymbol{F}_{u,v} = \frac{1}{4}\alpha(u)\alpha(v)\left[\sum_{i=0}^{7}\sum_{j=0}^{7}\boldsymbol{B}_{i,j} \times \cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\right], \tag{14}$$

where,

$$\alpha(u), \alpha(v) = \begin{cases} \frac{1}{\sqrt{2}} & u, v = 0, \\ 1 & otherwise. \end{cases}$$

The result of the DCT function for an 8 × 8 image block is a 64 coefficients matrix that contains the 2D spatial frequencies. The element (0,0) in the matrix is called “DC coefficient” and has zero frequency in both directions. The remaining 63 elements are called the “AC coefficients”, for which the frequencies increase from left–top corner to right–bottom corner in the matrix [21]. The inverse function of Equation (3) during decompression can be defined as

$$\breve{\boldsymbol{B}}_{i,j} = \frac{1}{4}\left[\sum_{u=0}^{7}\sum_{v=0}^{7}\alpha(u)\alpha(v)\boldsymbol{F}_{u,v} \times cos\frac{(2i+1)u\pi}{16}cos\frac{(2j+1)v\pi}{16}\right]. \tag{15}$$

Step 3. Quantization

As a result of the DCT function most of the image contents are preserved in a few coefficients (low frequency) mostly in the top–left corner of each block. The rest of the DCT coefficients corresponding to the higher frequencies are visually insignificant – psycho–visual redundancies – and can be discarded. Therefore, the next step in the JPEG compression is quantization, which divides each DCT coefficient by its corresponding element given in a 64–element quantization table (***QT***). The quantization step is controlled by a scalar value known as the JPEG quality factor (*qf*). The range is [0,100], where 0 represents lowest and 100 represents highest quality image. The quantization function of the JPEG compression can be defined as

$$\widehat{\boldsymbol{F}}_{u,v} = round\left(\frac{\boldsymbol{F}_{u,v}}{\boldsymbol{QT}_{u,v}}\right). \tag{16}$$

The JPEG standard provides two quantization tables $\boldsymbol{QT}_{50}$ for $\mathrm{QF} = 50\%$, one for each of the luminance and chrominance components. The standard tables can be used to construct different quantization tables corresponding to different quality factors. The element $qt_{\mathrm{QF}}(u,v)$ of a quantization table QT for a quality factor $qf\%$ is defined as

$$qt_{qf}(u,v) = \begin{cases} \mathcal{G}\left(\left\lfloor \dfrac{qt_{50}(u,v) \times \left(\dfrac{5000}{qf}\right) + 50}{100} \right\rfloor\right), & qf < 50 \\ \mathcal{G}\left(\left\lfloor \dfrac{qt_{50}(u,v) \times (200 - 2qf) + 50}{100} \right\rfloor\right), & qf > 50 \end{cases}, \tag{17}$$

where the function $\mathcal{G}(x)$ ensures that the elements in Equation (5) remain integers and are between 1 to 255 as required by the standard recommendation. The function is defined as

$$\mathcal{G}\left(qt_{qf}(u,v)\right) = \begin{cases} 1, & qt_{qf}(u,v) < 1, \\ 255, & qt_{qf}(u,v) > 255, \\ qt_{qf}(u,v), & \text{otherwise.} \end{cases} \tag{18}$$

In addition, these tables can also be user defined input to the encoder. During decoding, the inverse function of Equation (5) simply performs a multiplication operation to estimate the closest representation of the original DCT values as

$$\breve{\boldsymbol{F}}_{u,v} = \widehat{\boldsymbol{F}}_{u,v} \times \boldsymbol{QT}_{u,v}. \tag{19}$$

Step 4. Intermediate Encoding

In this step, the quantized DCT coefficients are represented in such a way that more compression savings can be achieved in the final step. First, the coefficients $\breve{\boldsymbol{F}}_{u,v}$ of each block are scanned in a zigzag order onto a vector, called the Minimum Code Unit (MCU). As a result, zeros corresponding to the higher frequencies end up together and can be encoded in an efficient way, that is, an End of Block (EOB) symbol is added to the MCU after the last non–zero coefficient. The DC and AC coefficients have different properties thus the DC coefficient is treated differently from the rest of 63 AC coefficients. The DC coefficients of adjacent blocks have higher correlation; therefore, the coefficients are differentially pulse code modulated (DPCM) with each other. A prediction error between the adjacent DC coefficients is encoded as the amplitude value $A_{\breve{\boldsymbol{F}}_{u,v}}, (u, v = 0)$ of the coefficient in ones' complement form. The size category of the prediction error is included in the head $H_{\breve{\boldsymbol{F}}_{u,v}}, (u, v = 0)$ of the coefficient. The quantized AC coefficients are run–length encoded (RLC) such that the consecutive zero coefficients are compressed. The non–zero coefficients are encoded as [(run–length, size), amplitude], where run–length is the number of zeros between two consecutives non–zero AC coefficients and size is the number of bits required to represent the amplitude. The run–length together with size are encoded as head $H_{\breve{\boldsymbol{F}}_{u,v}}, (u \neq 0, v \neq 0)$ of the coefficient. The value of the coefficient is encoded as an amplitude $A_{\breve{\boldsymbol{F}}_{u,v}}, (u \neq 0, v \neq 0)$ in ones' complement form. The head parameter of each coefficient is entropy encoded as discussed below.

Step 5. Entropy Encoding

In the previous step, the quantized DCT coefficients are represented in such a way that they can be efficiently compressed with an entropy encoder such as the Huffman encoder [23]. The

Huffman encoding scheme assigns a variable length code (VLC) to each symbol based on its probability. The main idea of VLC is to assign shorter codes to the most probable symbols and longer codes to the less probable symbols. During decompression, Huffman decoder along with the coding tables are used to recover the symbols from the compressed bitstream.

2.6. Evaluation Metrics

2.6.1. Encryption Analysis Metrics

2.6.1.1. Correlation Analysis

An encryption algorithm should eliminate correlation among adjacent pixels in an image for better security. In general, the correlation coefficient $\rho(\boldsymbol{x}, \boldsymbol{y})$ between two distributions $\boldsymbol{x}$ and $\boldsymbol{y}$ each with N elements is given by,

$$\rho(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\boldsymbol{x}_i - \mu_{\boldsymbol{x}}}{\sigma_{\boldsymbol{x}}}\right)\left(\frac{\boldsymbol{y}_i - \mu_{\boldsymbol{y}}}{\sigma_{\boldsymbol{y}}}\right), \tag{20}$$

where μ_a is the mean and σ_a is the standard deviation defined as

$$\mu_{\boldsymbol{a}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{a}_i,$$

$$\sigma_{\boldsymbol{a}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|\boldsymbol{a}_i - \mu|^2}.$$

The coefficient $\rho \in \{-1.0,1.0\}$, where $\rho = 0$ shows that there is no correlation, $\rho < 0$ shows negative correlation and $\rho > 0$ shows positive correlation. The negative correlation means that when one value is increasing the other is decreasing and the positive correlation means that both values are either increasing or decreasing.

2.6.1.2. Histogram Analysis

The histogram of an image gives the intensity distribution as the number of pixels at each intensity level. For a plain image, the histogram is a skewed distribution concentrated at one location and a cipher image has a uniform distribution. To quantify the characteristics of a histogram $\boldsymbol{R}$, histogram variance $V(\boldsymbol{R})$ is calculated as,

$$V(\boldsymbol{R}) = \frac{\sum_{i=1}^{N}(\boldsymbol{R}_i - \mu_{\boldsymbol{R}})^2}{N-1}, \tag{21}$$

where,

$$\mu_{\boldsymbol{R}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{R}_i,$$

and N is the level of intensities in the image and μ is the mean of image histogram. Small value of $V(\boldsymbol{R})$ means a uniform distribution.

2.6.1.3. Information Entropy Analysis

The information entropy shows the degree of randomness in an image. The entropy of an image $H(\boldsymbol{I})$ is given by

$$H(\boldsymbol{I}) = -\sum_{i=1}^{M} p_i log_2(p_i), \tag{22}$$

where, p_i is the probability of a pixel value in the image. For a truly random image with $N = 256$ intensity levels, the ideal value of the entropy should be closer to $H(\boldsymbol{I}) = log_2(N) = 8$.

2.6.1.4. Differential Attack Analysis

In order to be resistant against differential attack, an encryption algorithm should have the ability to generate two different cipher images for plain images with a minor difference. The degree of change can be quantified by two metrics, namely, the number of pixels change rate (NPCR) and the unified average changing intensity (UACI). The NPCR gives the percentage difference between two cipher images and UACI gives the average intensity of differences between the two images. For this purpose, a plain-image $\boldsymbol{I}_1$ of size M is slightly modified by randomly changing one of its pixel values to generate another image $\boldsymbol{I}_2$. The two plain-images $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ are encrypted using the same encryption key to obtain the cipher images $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, respectively. The NPCR and UACI parameters are calculated for the cipher images $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ as,

$$NPCR = \frac{\sum_{i,j} \boldsymbol{D}_{i,j}}{M} \times 100\%, \tag{23}$$

where,

$$\boldsymbol{D}_{i,j} = \begin{cases} 1, & \boldsymbol{C}_1(i,j) \neq \boldsymbol{C}_2(i,j) \\ 0, & \boldsymbol{C}_1(i,j) = \boldsymbol{C}_2(i,j) \end{cases}.$$

$$UACI = \frac{1}{M}\left[\sum\nolimits_{i,j} \frac{|\boldsymbol{C}_1(i,j) - \boldsymbol{C}_2(i,j)|}{255}\right] \times 100\%.$$

For $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ to have the ideal value of NPCR and UACI, the minor change in the plain images should be reflected across the whole cipher images. Usually, the diffusion process, which makes the current ciphertext dependent on the previous ones, achieves this property.

2.6.1.5. Jigsaw Puzzle Solver Attack Analysis

The robustness of a block-based perceptual encryption can be studied using the following three measures proposed in [24], [25]:

Direct Comparison ($\mathbf{D_c}$): measures the ratio of blocks in the correct position in the recovered image. Let I be the original image, I_r the recovered image, p_i the ith piece, and N the total number of pieces, then, $D_c(I_r)$ is given by

$$\mathbf{D_c}(I_r) = \frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}} d_c(p_i)\,, \tag{24}$$

where,

$$d_c(p_i) = \begin{cases} 1, & I_r(p_i) = I(p_i), \\ 0, & I_r(p_i) \neq I(p_i). \end{cases}$$

Neighbor Comparison ($\mathbf{N_c}$): measures the ratio of correctly joined pairwise blocks. For the recovered image I_r with B boundaries among the pieces and b_i being the i th boundary, $N_c(I_r)$ is given by

$$\mathbf{N_c}(I_r) = \frac{1}{\mathrm{B}}\sum_{i=1}^{\mathrm{B}} n_c(b_i)\,, \tag{25}$$

where,

$$n_c(b_i) = \begin{cases} 1, & \text{if } b_i \text{ is joined correctly,} \\ 0, & \text{otherwise.} \end{cases}$$

Largest Comparison ($\mathbf{L_c}$): measures the ratio of the largest joined blocks that have correct pairwise adjacencies with other blocks in the component. For a recovered image I_r, the $L_c(I_r)$ is given by

$$\mathbf{L_c}(I_r) = \frac{1}{\mathrm{N}}\max_i\{l_c(I_r, i)\}\,, \tag{26}$$

where N is the number of partially correct assembled regions, $i = 1, 2, \cdots, \mathrm{N}$, and $l_c(I_r, i)$ is number of blocks in the ith assembled area. The scores, $\mathbf{D_c}, \mathbf{N_c}, \mathbf{L_c} \in [0,1]$, where a larger value indicates a better reconstruction of the cipher image.

2.6.2. DL Performance Analysis Metrics

A DL model performance can be measured using different evaluation metrics such as accuracy, sensitivity, specificity, and receiver operating characteristic curve (ROC) measures. For the definition of these metrics, it is important to define different terminologies. The number of predictions that belong to the positive class and are correctly classified as such are called true positives (TP) and misclassified as negative class are called false negatives (FN). Similarly, the number of observations belonging to the negative class and classified as such are known as true negatives (TN) and misclassified as positive class are known as false positives (FP).

Accuracy calculates the total number of correct predictions (TP + TN) made by the model and is given as:

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}. \tag{27}$$

It is an important metric when FP and FN have equal significance. For the diagnosis, however, it is crucial to identify the positives and the FN occurrence is unacceptable. For instance, a healthy individual identified as a patient (FP) is a smaller problem than identifying a patient as healthy (FN). For this purpose, sensitivity measures a model performance by how few FN are predicted. It is defined as

$$\text{Sensitivity} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{28}$$

As opposed to sensitivity, specificity measures a ratio of TN to total negative in the observations as

$$\text{Specificity} = \frac{\text{TN}}{(\text{TN} + \text{FP})}. \tag{29}$$

Finally, the ROC curve measures a model classification performance at all classification thresholds. The curve plots false positive rate (FPR) and true positive rate (TPR) at different thresholds. The TPR and FPR are calculated as

$$\text{TPR} = \text{Sensitivtiy} = \frac{\text{TP}}{(\text{TP} + \text{TN})}. \tag{30}$$

$$\text{FPR} = 1 - \text{Specificity} = \frac{\text{FP}}{(\text{FP} + \text{TN})}. \tag{31}$$

The area under the ROC curve (AUC) can be used to calculate the area underneath the entire curve.

III. SIMULTANEOUS IMAGE ENCRYPTION AND COMPRESSION

3.1. Motivation

Like any other type of data, image data is compressed and encrypted for efficient and secure storage and/or transmission. Due to the intrinsic features of image, e.g., bulk of data and high redundancy, compression is performed before encryption. Performing encryption on compressed data reduces the computational cost of the encryption operation; however, the main challenge is how to ensure reasonable security without downgrading the compression performance.

3.2. Related Work

Image compression and encryption are performed together for efficient and secure image transmission and/or storage. A straightforward way is to perform compression and encryption as two separate processes. However, the problem with such an approach is that if the encryption is performed before, then the compression efficiency (especially in lossless mode) is severely reduced. Conversely, it is challenging to ensure reasonable security in the compression domain while preserving the compression savings. Therefore, in the last few years, there has been a growing research interest in combining compression and encryption into a single algorithm. In the schemes, image compression and encryption are performed jointly and can be categorized as joint JPEG compression and encryption, and simultaneous image compression and encryption schemes.

In the joint JPEG compression and encryption schemes, the DCT coefficients are encrypted, and such schemes have the advantage of being format compliant and they preserve the compression efficiency. One class of such schemes is selective encryption (SE) algorithms. Droogenbroeck et al. [26] proposed a SE method for JPEG compressed images, in which only the non-zero AC coefficients are encrypted. The encryption is based on DES, Triple-DES or IDEA algorithms. The DC coefficients are left unmodified because they carry visible information and are predictable. The encrypted image can be decompressed without knowing the secret key to access an image of degraded quality. Alternatively, Puech et al. [27] proposed a SE technique for JPEG compressed medical images based on encrypting both DC and AC coefficients of DCT function. The DC coefficients are combined with some AC coefficients of the lowest frequencies to form a 128 bits stream to be encrypted by AES algorithm. The SE techniques can be extended to methods based on region of interest (ROI). Ou et al. [28] proposed a ROI based SE method that deals with the security of medical images during the compression stage. Their method exploits the ROI coefficients identified by JPEG2000 standard. The idea is to invert only the sign bits and some of the most significant bits of wavelet coefficients belonging to ROI in the high frequency sub-bands. However, Zhou et al. [29] shows that non-compression methods are desirable for protecting medical images. Puech et al. [30] proposed an ROI based image encryption technique in which the SE is embedded in a standard JPEG coding algorithm. The encryption is performed by using the AES algorithm on quantized AC coefficients corresponding to the human skin, during the JPEG entropy coding phase. The resultant image is termed as a crypto-compressed image. Itier et al. [31] proposed a recompression method for crypto-compressed images in the encrypted domain which benefits cloud storage owners. Further compression of the compressed image is achieved by removing the last bit of each non-zero quantized DCT coefficient. However, this process degrades the image quality quantitatively, which may not be desirable by the image owner. Another class of joint JPEG compression and encryption schemes is scrambling-based image encryption methods. In such schemes, either entire blocks or inter-block coefficients are shuffled. Li et al. [32] proposed to shuffle DC coefficients and same frequency coefficients to produce an unintelligible image. However, this method reduces the compression efficiency of RLC and predictive coding of DC coefficients [31]. Minemura et al. [33] proposed to scramble only the AC coefficients of a JPEG compressed image in order to encrypt the image. However, leaving DC coefficients in plain reveals the outline of the image [31]. In natural and computer-generated images, on average 8 to 16 adjacent pixels are correlated in horizontal, vertical and diagonal directions [34]. Therefore, block based scrambling methods with an appropriate block size do not disrupt the correlation of the original image and can achieve a higher compression ratio. Kurihara et al. [35] proposed to scramble blocks in the spatial domain for lossless coding of color images. However, security of the scheme is vulnerable to jigsaw puzzle solvers [36]. Chuman et al. [37] proposed a block scrambling-based encryption-then-compression (ETC) system which is robust against jigsaw puzzle solver and brute-forces attacks. The scheme first converts an RGB image into YCbCr color space, and then combines all of the three channels to generate a single grayscale image. The image is encrypted by applying block scrambling-based methods and compressed using JPEG in lossy mode. The enhanced security is attributed to the fact that the system generates a grayscale cipher image. However, the compression performance of the scheme is degraded, as the color channels correlation is not used [38].

Chaos-based encryption methods are popular for image security. In recent years, simultaneous image compression and encryption schemes have been proposed that combine chaos-based permutation and number theory-based diffusion processes. Zhu et al. [18] designed an image encryption algorithm integrated with compression using 2D hyper-chaos and Chinese remainder theorem (CRT). The 2D hyper-chaos is used to shuffle the position of pixels in the original image. Then, the adjacent pixels values in the shuffled image are represented as a single CRT solution. The chaotic system parameters along with the moduli values of CRT are used as encryption keys. The achievable compression ratio is claimed to be the block size (i.e., the number of congruence equations in CRT). The same approach has been adopted in several other works. For example, Guo et al. [39] used a quantum chaotic map and DNA complementary rule for permutation followed by CRT-based diffusion; Brindha et al. [40] proposed to use a hyper chaos system for both permutation and diffusion, then compression is carried out using CRT; and Duseja et al. [41] designed image sharing scheme using CRT. However, the cryptanalysis of Li et al. [42] and Bai et al. [43] have shown that since the moduli are used as sub-keys; therefore, the schemes are insecure and are vulnerable to chosen plaintext attack. In addition, they have shown that the resultant compression savings are marginal or even negative. Huang et al. [44] proposed a nonlinear multi-image encryption scheme based on logistic chaotic map and 2D linear canonical transform. The chaotic map is used for image permutation and diffusion. The scheme enables multi-image encryption and compression. However, [45] has shown that the secret key of the scheme is dependent on the plaintext and cannot be obtained before encryption, which makes the secret key storage and transmission difficult. In addition, the [44] scheme cannot be implemented in the digital domain without complex optical equipment [45]. Alternatively, [45] has proposed a plaintext-related pixel adaptive diffusion process where the secret keys are independent of the plaintext image while the cipher image is sensitive to the secret keys and the plaintext image. Their system is based on the quaternion discrete fractional Hartley transform and can perform multi-image encryption and compression. A multi-image compression-encryption scheme based on hyper-chaotic system, discrete cosine transform and discrete fractional random transform is presented in [46]. In recent years the applications of compressive sensing (CS) have been extended to image encryption domain to achieve simultaneous compression and encryption of an image [47]–[53]. A detailed survey on the topic is presented in [54].

3.3. Proposed Method

We have proposed a scheme that incorporates the compression requirements in an image encryption system as shown in Figure 2. The first step is to shuffle the positions of pixels in the original image by using a permutation key. A lossless compression process of the shuffled image follows the permutation stage. Then, the compressed bitstream goes through a diffusion process. The

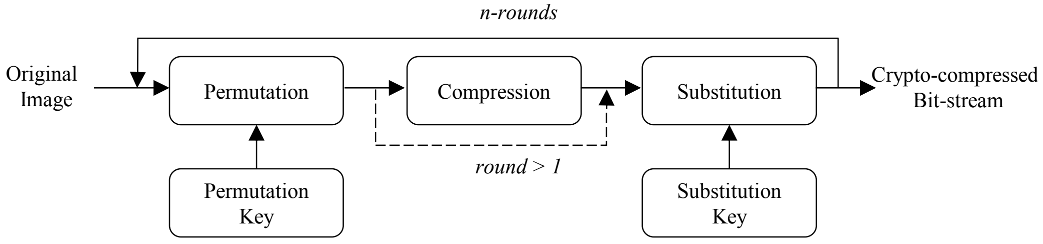


Figure 2. Architecture of the proposed novel hybrid image encryption-compression scheme. The lossless compression module is placed between the permutation and diffusion blocks. The compression is only performed in the first round. The input to the system is a plain image and output is a crypto-compressed bit-stream.

compressed bitstream is grouped into a group of 8 bits and padded as necessary, which are then XOR with the substitution key. As suggested in the literature, that when using chaos-based image encryption it is necessary to perform the encryption process in more than one round [34]. In the proposed scheme, the compression is performed only in the first round, as in the subsequent rounds compressing compressed data is of less advantage and may not be compressible at all. In our experiments, we have used three rounds of encryption process. The security of the proposed system is based on chaotic maps and consists of two steps, i.e., permutation and substitution. The compression is performed in lossless mode, and we have used Huffman encoding as the entropy encoder. Arithmetic entropy encoder tends to achieve better compression ratio than Huffman encoder; however, Huffman encoding has the advantage of high speed and simplicity in both hardware and software implementation. In addition, Huffman coding is the entropy encoder of widely used JPEG image standard, which is supported across all web browsers. Our proposed scheme is explained in the following sections.

3.3.1. Permutation Block

Unlike text data, image pixels have strong correlations with neighboring pixels. In the permutation stage, image pixels are shuffled in order to de-correlate their strong relationship with the adjacent pixels. The process is carried out in such a way that the pixels' original values do not change. The proposed permutation process is given in Algorithm 1. The algorithm takes the control parameters, input image vector and parameter R as input. The algorithm is explained in the following steps:

Step 1: Convert the input image into a vector of $H \times W$ elements as an input sequence G_n, where H is the height and W is the width of the image.

Step 2: Generate two hyper-chaotic sequences using the 2D hyper-chaos discrete nonlinear dynamic system discussed in Chapter 2 as $X = \{X_1, X_2, \cdots X_n\}$ and $Y = \{Y_1, Y_2, \cdots Y_n\}$ shown in Line 1 to 7 of Algorithm 1. In order to avoid the harmful effect of 2D hyper-chaos system iterate the system for

$n+R$ times and discard the first R elements of the sequences. In our experiments, we have set $R=3000$. The X and Y are then used as the keys to shuffle the input image sequence.

Step 3: The shuffled image G' is obtained in Line 8 to 13 of Algorithm 1 by rearranging the input sequence G by X (i.e. $T_j = G_{X_j}$) and then by Y (i.e. $G'_k = T_{Y_k}$).

Algorithm 1: Permutation process

```
Input: The control parameters $x_0, y_0,\ a_1,\ a_2,\ a_4,\ b_1,\ b_3$; input image vector $G[1 \cdots n]$; and $R$
Output: Shuffled image vector $G'$
PERMUTATION($G[1 \cdots n]$)
        //Hyper-chaotic sequence generation
    1.  Initialize $X[1] = x_0$
    2.  Initialize $Y[1] = y_0$
    3.  for $i = 2:(n+R)$
    4.      $X[i] = a_1 + a_2 * X[i-1] + a_4 * Y[i-1]$
    5.      $Y[i] = b_1 + b_3 * X[i-1]^2$
    6.  Endfor

    7.  Discard first $R$ elements of $X$ and $Y$

        // Shuffle the image
    8.  for $j = 1:n$
    9.      $T[j] = G[X[j]]$
    10. Endfor

    11. for $k = 1:n$
    12.     $G'[k] = T[Y[k]]$
    13. Endfor

    14. return $G'$
```

When recovering the original image, the permutation process can be reversed as shown in Algorithm 2. The algorithm takes a shuffled image vector G' and the keys X, and Y used in Algorithm 1 as input. The original image vector can be recovered by rearranging the input sequence G' first by Y as (i.e. $T_{Y_j} = G'_j$) and then by X as (i.e. $G_{X_k} = T_k$). To obtain the original image reshape the resultant vector to dimensions $H \times W$.

Algorithm 2: Inverse permutation process

```
Input: The permutation keys $X$, and $Y$; shuffled image vector $G'[1 \cdots n]$
Output: Final recovered image vector $G$
INVERSEPERMUTATION($G'[1 \cdots n]$)
        // Rearrange the image vector
    for $j = 1{:}n$
        $T[Y[j]] = G'[j]$
    Endfor

    for $k = 1{:}n$
        $G[X[k]] = T[k]$
    Endfor

    return $G$
```

3.3.2. Compression Block

The compression block performs lossless image compression in two steps; the first step is data-to-symbol mapping for efficient compression and the second step is the entropy encoding where the actual compression occurs. In data-to-symbol mapping, usually, the input data is first transformed using DCT, DWT and/or color space conversion. Such mapping exploits the correlation exists in the image data. However, in our scheme the input data is the shuffled image (i.e., the correlation is removed in the permutation step for better security); therefore, the transformation is set to the identity mapping thus avoids the additional cost of performing a transformation. The Huffman entropy encoder in our proposed scheme operates directly on the shuffled data as shown in Figure 3. Indeed, lossless compression of the shuffled image is possible because the entropy encoder heavily relies on the statistics of the input source. In the permutation stage only pixels position is changed while the values are left unmodified. Therefore, the shuffled image has the same intensity distribution as that of the original image. In order to compensate for any degradation in compression efficiency resulted by omitting the transformation stage, we have proposed a partitioning method as discussed in Chapter 2 to represent adjacent data elements as a block. Our partitioning algorithm is based on Chinese

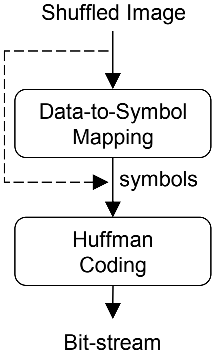


Figure 3. The lossless compression module of the proposed hybrid image encryption-compression scheme. The data-to-symbol map function is based on Chinese remainder theorem. The dashed line shows omitting of data-to-symbol map.

remainder theorem (CRT) where its solution is treated as a symbol. The block representation improves the compression efficiency of Huffman encoder, e.g., the source alphabet is extended from M to M^Q, where Q is the block size, thus achieves lower bound on B_C of (10), also for larger values of Q the average bit rate B_C monotonically reaches the source entropy. The compression procedure of the proposed scheme is given in Algorithm 3 and is explained in the following subsections:

3.3.2.1. Data-to-Symbol mapping using CRT

For better runtime of the proposed data-to-symbol map based on CRT, we have pre-calculated the CRT solutions by using (2) and stored them in a lookup table. Therefore, replaces the runtime computations with a simpler array indexing operation. In our experiments, we have used block size of $Q = 2$ and moduli values are $m_1 = 256$ and $m_2 = 257$ as suggested in [2], [3]. The shuffled image adjacent pixels can be represented as blocks by using SymbolMap function of Algorithm 3 as,

Step 1: Group the adjacent Q elements of sequence G' into blocks as $B_1, B_2, \cdots, B_{(n)/Q}$. Each block has Q elements as $B_{i+1} = \{G'_{i*Q+1}, G'_{i*Q+2}, \cdots, G'_{i*Q+Q}\}$, and $i = 0, 1, 2, \cdots, n/Q - 1$.

Algorithm 3: Compression process

```
Input: Chosen block size $Q$, and shuffled image vector $G'[1 \cdots n]$
Output: Compressed bitstream
//Data-to-symbol mapping
SymbolMap ($G'[1 \cdots n]$)
    for $i = 1:(n/Q)$
        Set $B$ with $Q$ array entries of $G'$
        $S'[i] = $ LookUpCrt$(B)$
    Endfor

    Set $f[1 \cdots m]$ with the probabilities of each symbol in $S'[1 \cdots m]$

// Huffman encoding
HuffmanEncoding($f[1 \cdots m]$)
    $T$ = empty binary tree
    $Q$ = priority queue of pairs $(i, f[i]), i = 1:m,$ with $f$ as comparison key
    for $k = 1:m-1$
        $i = Q.$ExtractMin$(Q)$
        $j = Q.$ExtractMin$(Q)$
        $f[m+k] = f[i] + f[j]$
        InsertNode$(T, m+k)$ with children $i, j$
        InsertRear$(Q, (m+k, f[m+k]))$
    Endfor

    return $T$
```

Step 2: For each block B_i get CRT solution value from the lookup table to represent it with a unique symbol. The result is a block representation of the data.

It is noteworthy that block representation only reduces the dimension of G' from n to n/Q not the information content. The reason is that the resultant solution of CRT lies in the product of moduli; therefore, elements of S' can no longer be represented as 8-bit integers. The block representation aids the entropy encoder, where the actual compression occurs.

3.3.2.2. Entropy Coding

The block symbols obtained in Section 3.3.2.1 can now be assigned with unique decodable codes by using Huffman coding algorithm. The algorithm uses a binary coding tree to assign codewords to symbols. A leaf in the coding tree represents each symbol of the alphabet and the path

Algorithm 4: Decompression process

```
Input: Bitstream $B$; Huffman tree structure $T$
Output: Decompressed bitstream
// Huffman decoding
HuffmanDecoding($B[1 \cdots t]$)
    1.  $current = T.root$
    2.  $ind = 1$
    3.  for $i = 1{:}t$
    4.      if ($B[i] == 0$)
    5.          $current = current.left$
    6.      else
    7.          $current = current.right$
    8.      if ($current.left == null$ && $current.right == null$)
    9.          $S'[ind] =$ LeafNodeValue
    10.         $ind++$
    11.         $current = T.root$
    12. Endfor

//Blocks to symbols
InverseSymbolMap($S'[1 \cdots n]$)
    13. $ind = 1$
    14. for $i = 1{:}n$
    15.     $G'[ind] = S'[i]\%256$
    16.     $G'[ind + 1] = S'[i]\%257$
    17.     $ind += 2$
    18. Endfor

    19. return $G'$
```

from that leaf to the root corresponds to the code of a symbol. The Huffman coding procedure is as follow:

Step 1: Get the probability distribution of the source S' as shown in Line 5.

Step 2: Combine the two symbols with smallest probability and repeat to get the binary coding tree as shown in Line 6 to 14 of Algorithm 3. We have used a priority queue as the main data structure to store the nodes.

The binary coding tree T obtained in Step 2 can be used to represent the block symbols of the sequence S' and the resultant output is a compressed binary bitstream B. During decompression,

```
Algorithm 5: Substitution process
Input: $miu, d,$ compressed bitstream $B$
Output: Final Encrypted-Compressed image
SUBSTITUTION($B[1 \cdots t]$)
        //Key Generation
    Set $p[1 \cdots t/8]$ with $8$ array entries of $B$
    for $i = 1:(t/8)$
        $d = miu * d * (1 - d)$
        $K[i] = \left(floor(d/2 * 10^{14})\right)\%256$
        $c[i] = XOR\left(XOR\left((p[i] + K[i]\ \ mod\ \ 256), K[i]\right), c[i-1]\right)$
    Endfor

    return $c$
```

the prefix codes are first translated to individual block symbols (i.e., the CRT solutions) by HuffmanDecoding function given in Algorithm 4. The function takes the input bitstream and starts at the root node. Based on the input bit in the sequence the function traverses the left or the right path in the graph until a leaf node is reached. The value of the leaf is the decoded symbol. The process is repeated for each bit in the input bitstream. The output of the function is block representation of the pixel values.

In order to recover the original pixel values from S' the InverseSymbolMap function of Algorithm 4 can be used. The function maps symbols back to data by using (3) with the set of moduli values used in Section 3.3.2.1. In the traditional data-to-symbol map, it is necessary to hold the same codebook on each side, which may incur additional cost of transmission energy. However, during decompression stage the proposed method only requires computing the modulus operation to recover the original data as in (3).

3.3.3. Substitution Block

In the permutation stage only pixels position is changed to remove correlation between adjacent pixels. However, the intensity distribution of the plain image is left unchanged, which makes statistical attacks feasible. In order to alter the intensity distribution of the image the final step of the proposed method is substitution. In the substitution process, the pixel values are modified sequentially by performing XOR operation on them with key stream elements to confuse the relationship between plain image and cipher image. The substitution process can be performed as given in Algorithm 5 and is described below:

Step 1: The first step is to generate a key stream for the substitution stage. An element k_i of the key can be obtained as

$$k_i = \left(\left\lfloor\left(\frac{d_{i+1}}{2}\right) \times 10^{14}\right\rfloor\right) \; mod \;\; N, \tag{32}$$

where $\lfloor\cdot\rfloor$ is the floor operation and returns the nearest integer value, d is the state of the map given by (3) and N is the intensity level of the input image.

Step 2: The encrypted pixel value is obtained in Line 5 of Algorithm 5 as,

$$c_i = k_i \oplus \big((p_i + k_i) \;\; mod \;\; N\big) \oplus c_{i-1}, \tag{33}$$

where $\oplus$ is the XOR operation, p_i is the currently diffused pixel, k_i is the key stream element, c_i is the output cipher pixel, and c_{i-1} is the previous cipher-pixel and N is the intensity level. The bitstream from compression stage can be used as p_n by grouping 8 bits (since each key element is 8 bits long) to form a single element. The resultant cipher image not only depends on the corresponding key stream element but also on all the previous pixel values. During decoding to recover the value of p_i, the inverse of (33) is by

$$p_i = (k_i \oplus c_i \oplus c_{i-1} + n - k_i) \;\; mod \;\; N. \tag{34}$$

Algorithm 6: Inverse substitution process

```
Input: Final Encrypted-Compressed image $c$; Substitution key $K$
Output: Compressed bitstream
InverseSubstitution ($c[1 \cdots t]$)
    Set $D[1 \cdots t/8]$ with $8$ array entries of $c$
    for $i = 1:(t/8)$
        $p[i] = (XOR(XOR(K[i], D[i]), D[i-1]) + 256 - K[i]) \;\; mod \;\; 256$
    Endfor

    return $p$
```

The inverse of the substitution process is given in Algorithm 6. The algorithm takes the encrypted-compressed bitstream c and the substitution key K used in Algorithm 5 as input. The bits in c are first grouped into 8 and then XORed with K as in (34) to obtain the plaintext.

3.4. Experimental Results and Analysis

In this section, we present the simulation results obtained by applying our proposed method to compress and encrypt images. Section 3.4.1 gives detailed security analysis of the proposed method. We have performed experiments on the KODAK dataset [55] and the UCID dataset [56] to evaluate the compression performance of our scheme in Section 3.4.2.

3.4.1. Security Analysis and Discussion

The proposed scheme provides a high level of security and can resist all attacks. In this section, we have included security analysis results of the scheme including keyspace analysis, statistical analysis and resistance against differential analysis.

3.4.1.1. Keyspace Analysis

3.4.1.1.1. Number of Control Parameters

An efficient encryption system should have a keyspace large enough to make the system resistant to brute force attack. Our proposed system is based on permutation and substitution process; therefore, the secret key consists of two parts: the permutation key K_P and the substitution key K_S. The K_P consists of the control parameters $(a_1, a_2, a_4, b_1, b_3)$ and the initial values (X_0, Y_0) of the dynamic system. In substitution block, x_0 and μ are the key parameters used for logistic maps. The computational precision of 64-bit double-precision number is about 10^{15}. Therefore, the K_P parameters and x_0 can be any value among 10^{15} numbers. μ can have any number from 0.43×10^{15} values. The total key space of the proposed scheme is:

$$\begin{aligned} K(K_P, K_D) &= 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \times 0.43 \times 10^{15} \\ &= 0.43 \times 10^{135} \approx 2^{448} \end{aligned} \tag{35}$$

The keyspace of the proposed scheme is sufficiently large enough to resist brute force attack.

3.4.1.1.2. Key Sensitivity Test

For key sensitivity, we have performed two tests: the first test is to see the difference between two cipher images obtained by encrypting the same plain image with two slightly different keys. Second, test the robustness of the system to information leak when partial key is obtained by the intruder.

For the first case, the sensitivity test has been performed according to the following procedure:

- The plain image is first encrypted with the chosen parameters for permutation key K_P and substitution key K_S to obtain cipher image C_1. The initial values of K_P in the permutation block are $x_0 = 0.0394$ and $y_0 = 0.0001$.

- Then, the initial value x_0 is changed to $0.0394 + 10^{-15}$ to obtain cipher image C_2.

- Finally, C_1 and C_2 are compared in terms of their intensity values. The result shows that C_1 and C_2 has about 99% difference when there is a slight change in the secret key. Similar results can be obtained when other control parameters of the keyspace are changed.

For the second case, the following sensitivity test has been performed:

- The cipher image is obtained by encrypting the plain image using the key $K(K_P, K_S)$.

- The cipher image is decrypted with a slightly different key. For example, the intruder has access to the same key except $x_0 = 0.0394 + 10^{-15}$ instead of $x_0 = 0.0394$.

The results for the key sensitivity are shown in Figure 4. Figure 4. (a) is the plain pepper image of size 384×512 and Figure 4. (b) is the corresponding compressed cipher image. The dimensions of the cipher image are reduced because of the 8-bits grouping of the compressed bitstream before the substitution stage. Each color component has a different number of bits which results in a different number of pixels. For example, the red component has 177,232 pixels, the green component has 173,225 pixels and the blue component has 167,579 pixels. For illustration, we have padded each color component to form rectangular images of the same size. The resultant cipher image

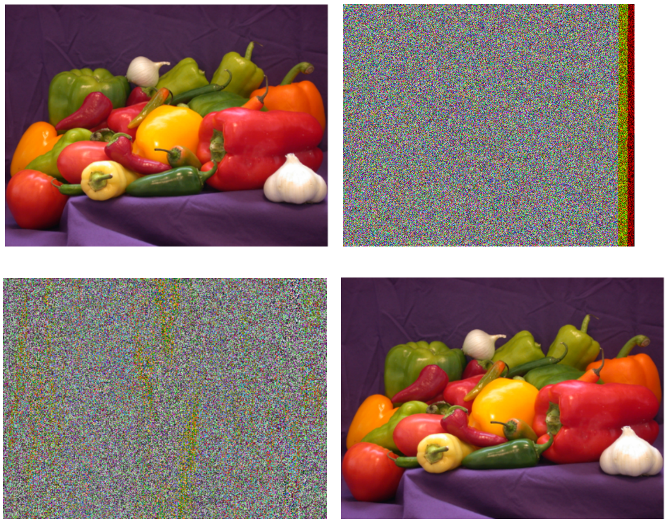


Figure 4. The key sensitivity test of the proposed method. (a) is the plain Pepper image, (b) is the crypto-compressed Pepper image, (c) is the recovered pepper image when the intruder has access to partial key information and (d) is the recovered image when using correct key information.

shown in Figure 4. (b) has dimension of 384×462 where the yellow and red strips appear due to padding for matching the size of the color components. The recovered image is shown in Figure 4. (c) and Figure 4. (d), when a slightly different key is used and when the correct decryption key is used, respectively. The figure shows that no information about the original image has been leaked.

3.4.1.2. Statistical Analysis

3.4.1.2.1. Histogram Analysis

An image histogram gives the intensity distribution of an image by plotting the number of pixels at each intensity level [57]. The histogram of a plain image is a skewed probability distribution as shown in Figure 5. (First column-blue bars) An encryption algorithm should alter this distribution in such a way that the resultant cipher image histogram does not reveal any information about the original image or its relationship with the original image. The histogram of the cipher image is shown in Figure 5. (First column-orange bars), which is fairly a flat distribution and significantly different from the original image histogram. For visual analysis, we have shown the plain image of each RGB channel and their corresponding cipher image in Figure 5.

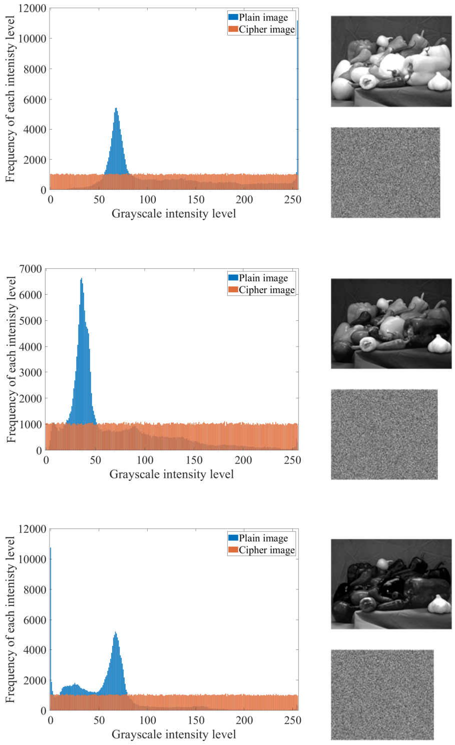


Figure 5. Histogram analysis of the crypto-compressed image obtained from the proposed architecture. The histogram of cipher image and plain image (left), and the plain image (right top) and cipher image (right-bottom). a. b. and c. are the red, green and blue channels, respectively.

3.4.1.2.2. Correlation of adjacent pixels

To quantify the correlation between two adjacent pixels in vertical, horizontal and diagonal direction, randomly choose 3000 pairs of adjacent pixels of an image. Then, calculate the correlation coefficient based on (20). The correlation coefficient for original image and cipher image in vertical, horizontal and diagonal direction is given in Table 1. The quantitative analysis of correlation shows that the cipher image obtained by the proposed method has sufficiently low correlation among adjacent pixels, indicating favorable diffusion performance.

Table 1. Security analysis of the proposed method in-terms of correlation coefficients of adjacent pixels in two images and information entropy.

Image	Channel	Correlation			Entropy
		Horizontal	Vertical	Diagonal	
Cipher	Red	-0.0343	-0.0092	-0.0186	7.9993
	Green	-0.0217	-0.0085	0.00007	7.9994
	Blue	0.0318	0.008	-0.0447	7.9993
Plain	Red	0.9637	0.9674	0.9572	7.3388
	Green	0.9817	0.9774	0.9724	7.4963
	Blue	0.9706	0.9617	0.9333	7.0583

3.4.1.2.3. Information entropy

Information entropy $H(S)$ is the measure of randomness of an information source S, i.e., higher the entropy better the encryption. The entropy of a source can be calculated as in (5). For an information source with N symbols, its outcome is truly random when $H(S) = \log_2(N)$. Therefore, for an image with 256 intensity levels a good cipher image will be the one with $H(S) = 8$. The entropy calculated in our experiments for both plain image and cipher image is given in the last column of Table 1. One can see that the entropy of the cipher image reaches the ideal value of $H(S) = 8$, when approximated. On the other hand, the plain image has entropy less than 8. Since the entropy of the cipher image is close to the ideal entropy value, the proposed scheme provides the highest possible security against potential key attacks.

3.4.1.3. Differential Attacks

In such an attack, the intruder slightly modifies the plain image and observes the changes in the result. The goal is to find out some meaningful relationship between the plain image and encrypted image. An encryption scheme can be protected against such an attack, if a minor change in the plain image can be reflected across the whole encrypted image. Our proposed scheme shows this behavior, as each pixel in the cipher text not only depends on the corresponding key stream element but also on the previous pixel value as in (10). In addition, to quantify the degree of change, we have used the two most common metrics: the number of pixels change rate (NPCR) and the

Table 2. Security analysis of the proposed method in-terms of NPCR and UACI.

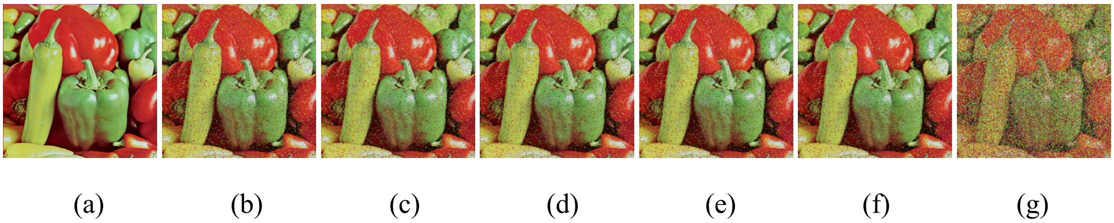
Channel	NPCR (%)	UACI (%)
Red	99.5461	35.9229
Green	99.7066	36.6366
Blue	99.6272	36.1533

unified average changing intensity (UACI). Let P_1 and P_2 be the plain images with only one pixel difference and their corresponding cipher images are C_1 and C_2. UACI gives the average intensity of differences between the two images and has an ideal value of 33.4635% as suggested in [58]. In our experiments, the proposed method achieves 99.54% and 35.92% for NPCR and UACI for red channel image, respectively. The rest of the analysis results are given in Table 2. The analysis results show that the proposed scheme is robust to differential attacks.

3.4.1.4. Robustness Analysis

3.4.1.4.1. Noise Analysis

A communication channel is susceptible to noise. Transmission over noisy channels may corrupt the cipher image and makes it difficult to recover the plain image. In this subsection, we have shown the robustness of the proposed scheme against Salt and Pepper noise (SPN). We have carried out our experiments on Pepper512 color image as shown in Figure 6. (a). In the experiments, different noise intensity was added to the cipher image, and the recovered images from the noisy images are shown in Figure 6. (b)-(g). We have used the peak signal-to-noise ratio (PSNR) and Structural Similarity Index Measure (SSIM) to quantify image quality degradation caused by the noisy channel. The PSNR and SSIM values between the recovered image and the plain image are given in Table 3.



(a) (b) (c) (d) (e) (f) (g)

Figure 6. Recovered images under different noise intensities (a) Original image, (b) 0.00001 SPN, (c) 0.00003 SPN, (d) 0.00005 SPN, (e) 0.00007 SPN, (f) 0.0001 SPN, (g) 0.001 SPN.

Table 3. PSNR and SSIM Values between recovered images from noisy cipher images and original Image.

Noise Intensity	Ours	[43]
0.00001	19.28/0.82	32.22/-
0.00003	19.28/0.82	31.88/-
0.00005	19.28/0.82	29.63/-
0.00007	19.28/0.82	28.07/-
0.0001	19.28/0.82	-
0.001	12.27/0.48	-

The PSNR between the decrypted images and plain image of the proposed method is smaller than [47]. However, as opposed to [47] the PSNR values remain the same when the noise intensity slightly changes. We have further increased the noise intensity by a factor of 10 and 100. In the first case, no quality degradation occurs while in the latter case the quality of the image degrades drastically. Our scheme has achieved a SSIM value greater than 80 for SPN. Figure 6 shows that the recovered images are still visually recognizable. Our scheme is robust against SPN; however, in the presence of Gaussian noise it is difficult to recover the plain image. The reason is that the cipher image is an encrypted-compressed bitstream obtained with variable length coding (VLC). With VLC, a single bit error can cause the decoder to decode a longer or shorter codeword and the subsequent symbols are likely to be decoded incorrectly due to the loss of synchronization at the decoder [59].

3.4.1.4.2. Data loss Analysis

In this section, we have shown robustness of the proposed method against data loss due to transmission fault. The data loss in cipher image affects quality of the recovered image. In our experiments, the test image is Pepper512 color image as shown in Figure 6. (a). The cipher image with the cropped pixels and their corresponding decrypted images are shown in Figure 7. In the experiments, we have cropped 8×8, 16×16 and 32×32 pixels in the middle (as shown in Figure 7. (a) – (c)) and bottom right (as shown in Figure 7. (h) – (j)) of the encrypted image. To quantify the quality of the recovered images we have calculated PSNR and SSIM given in Table 4. We have compared the average PSNR of the proposed method with [47]. The recovered image quality degrades with the size of crop while the position has a smaller effect on the quality of the image in [47]. However, for our proposed method we have observed the opposite trend. The image

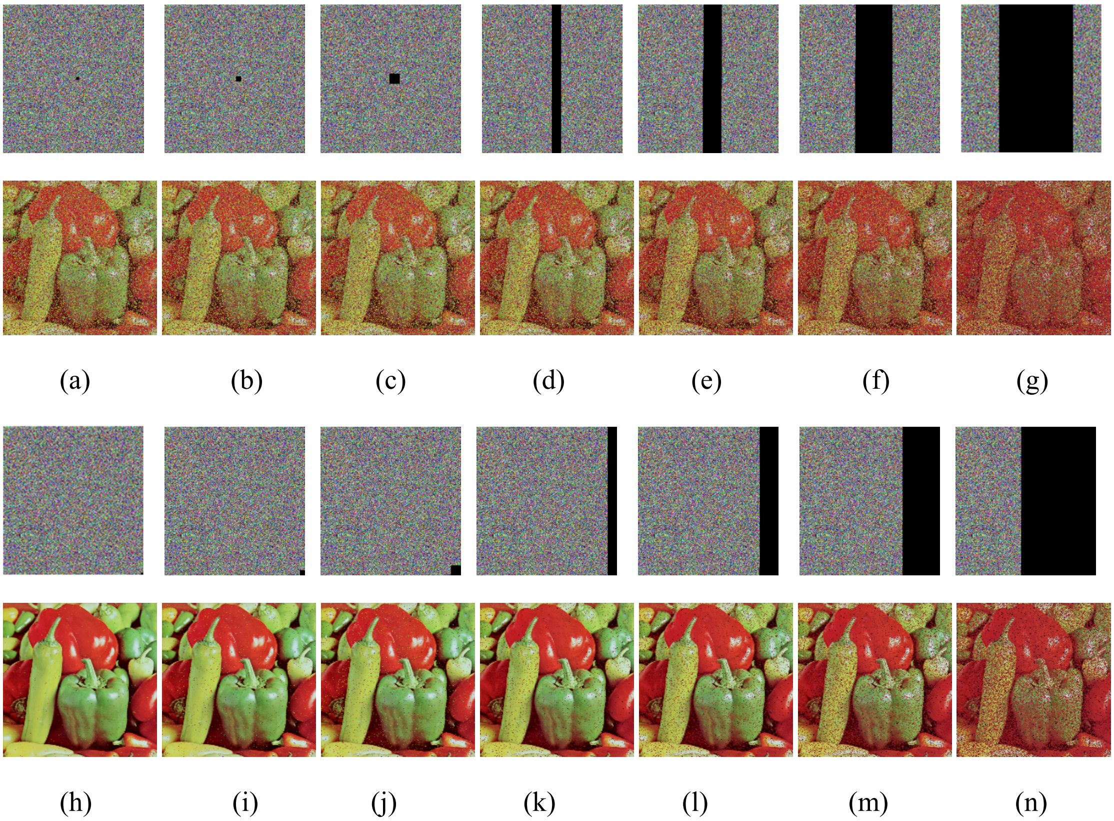


Figure 7. Decrypted images when the cipher image suffers from different sizes of data loss at different positions. (a)-(g) the data loss occurred in the middle of the cipher image and (h)-(n) the data loss occurred at the right side of the cipher image.

quality varies more with the position of the cropped pixels than the size of the crop. For example, when the position of the cropped pixels moves in the direction of top left from the bottom right the quality of the recovered image degrades. The reason is that in VLC the correct decoding of the subsequent symbols depends on the previously decoded symbols [59]. To further analyze this behavior, we have cropped $1/16, 1/8,$ $1/4$, and $1/2$ pixels of the cipher image from the middle (as shown in Figure 7. (d) – (g)) and right side (as shown in Figure 7. (k) – (n)). The same behavior can be seen with the increased crop size. For example, when half of the cipher image pixels are lost from the right side, the recovered image quality is better than when only 8×8 pixels are lost from the middle of the cipher image. From Figure 7, it can be seen that the recovered distorted image is

Table 4. PSNR and SSIM values between recovered images from cropped cipher images and original image.

Position and crop size	PSNR/SSIM		Average PSNR/SSIM	
	Middle	Right	Ours	[43]
8×8	13.10/0.53	32.70/0.99	22.90/0.76	27.90/-
16×16	13.00/0.52	26.77/0.96	19.88/0.74	25.98/-
32×32	12.85/0.51	23.20/0.92	18.02/0.72	8.58/-
$1/16$	12.91/0.52	22.29/0.89	17.60/0.70	-
$1/8$	12.68/0.50	19.28/0.81	15.98/0.66	-
$1/4$	12.25/0.48	16.28/0.70	14.26/0.59	-
$1/2$	11.51/0.45	13.27/0.56	12.39/0.51	-

still recognizable and Table 4 shows that SSIM score is around 0.45 even when half of the cipher image pixels are lost. Therefore, the proposed scheme can resist a higher degree of data loss with a restriction on the position of the data loss.

3.4.1.5. Time Complexity

In this section, we have analyzed the run-time efficiency of the proposed scheme. The proposed algorithm consists of three main steps: permutation, compression and substitution. In the experiment, we encrypted and compressed Pepper color image of size 512×512. In order to recover the plain image, the steps should be performed in reverse order as inverse substitution, decompression and inverse permutation. The time taken by the proposed algorithm is shown in Table 5. Since the same keys can be used to encrypt multiple images, we have not considered the keys generation time. During encryption, the substitution step requires more time as it has decimal to binary conversion function to carry out the XOR operation. During compression when data-to-symbol map function is not used then Huffman entropy encoder takes about 0.25 seconds. For the proposed data-to-symbol map based on CRT, we have pre-calculated the CRT solutions and stored them in a lookup table. Therefore, replaces the runtime computations with a simpler array indexing operation. The proposed data-to-symbol map adds 0.085 seconds to the compression runtime. The overall process takes 1.4636 seconds to encrypt and compress the image that is only 36% of the time required by [47]. During decryption, the decompression process dominates the computation time and requires 6.36 seconds to complete. To recover the original symbols from the CRT solutions requires only computing the modulus operation. The overall process requires 7.8065 seconds to recover the

Table 5. Time analysis of the proposed method.

Process	Encryption	Decryption
Permutation	0.0091	0.0092
Compression	0.2482+0.0845	6.3602+0.0532
Substitution	1.1515	1.3839
Total	1.4088+0.0845	7.7533+0.0532

plain image that is only 57% of [47] computation time.

3.4.1.6. Comparison with other studies

In this section, we have carried out comparisons with [47] and [60] to show the effectiveness of the proposed method. We have used color Lena256 image as a test image. The comparison analysis is given in Table 6. (a) The proposed method has achieved comparable correlation values for the cipher image and are all near to zero. (b) The information entropy values for each color component are larger than 7.99 and closer to the ideal value 8, and larger than [60] but smaller than [47]. (c) The average pixel changed ratio for the three studies is close to the recommended value suggested in [58]. (d) The key space of the proposed algorithm is smaller than [60] but larger than [47] and 2^{100}; therefore, can resist brute force attacks on the key space. (e) Key in the proposed method and [60] is independent of the original image that makes secret key storage and transmission simple. In [47], the key varies with the original image and has an advantage to use different key for different test images. However, when the key is dependent on the plaintext it cannot be obtained before encryption, thus making secret key storage and transmission difficult [45]. (f) Our proposed algorithm has better run time than [47] and [60].

Table 6. Comparison results with other studies for color image Lena (256x256).

Measures		Ours	[43]	[56]
Correlation Values	Horizontal	-0.0085	-0.00074	-0.0037
	Vertical	0.016	0.0012	0.0001
	Diagonal	0.0019	-0.0032	-0.0230
Entropy	Red	7.9958	7.9983	7.9893
	Green	7.9954	7.9985	7.9898
	Blue	7.9967	7.9982	7.9894
NPCR		99.43%	99.62%	99.79%
Key space		$\approx 2^{448}$	$> 2^{168}$	2^{280}
Plain Image dependence		No	Yes	No
Encryption time (s)		0.796	1.1168	1.25

3.4.2. Compression Analysis and Discussion

The performance of a compression algorithm can be evaluated in various ways depending on the optimization requirements. If the mode of compression is lossy, then the criteria is to evaluate the quality of the reconstructed image. For actual implementation of the algorithm then coder complexity e.g., memory requirement, power requirement and operations per second is taken into account. The utmost important evaluation criteria are coding efficiency. It can be measured in terms of the amount of information being reduced. We have calculated percentage compression savings (CS) as,

$$\mathrm{CS} = \frac{(Compressed\ \ image\ \ size * 100)}{Original\ \ image\ \ size}. \tag{36}$$

The image size is the number of bits required to represent the image. We have performed the same experiment of Pepper image on KODAK [55] and UCID [56] image datasets. The compression savings of the baseline and the proposed methods in comparison to lossless coding of encrypted image is given in Table 7. The given values have been obtained by averaging the CS of all the images in the corresponding dataset. First, we have compressed the unencrypted images in the datasets using the lossless compression algorithm discussed in Section 3.3.2. Since the round function on the transformation output introduces loss; therefore, we have used the entropy encoder directly on the plain image. As a result, the compression savings achieved for the KODAK dataset is about 8.33% and for the UCID dataset is about 5.76%. We have carried out a comparison of the proposed method with two encryption-then-compression (ETC) systems proposed in [35], [37] for compression performance. The ETC systems are block based scrambling methods. The ETC system of [35] performs lossless compression of color images while ETC system of [37] is grayscale-based image encryption and can perform both lossless and lossy compression. For fair comparison, we have implemented [37] in lossless mode. Both [35] and [37] have the same compression performance in lossless mode and the savings have been degraded by more than 50% when compared to lossless compression of unencrypted images. The reason is that in ETC systems there is a tradeoff between the key space and compression performance. For example, for larger block size the compression performance improves; however, the system becomes vulnerable to jigsaw puzzle solver as the number of blocks decreases. On the other hand, to improve the security of the ETC systems by taking smaller block size degrades the compression performance. In our implementation of [35] and [37], we have used block size of 8×8 as proposed in [37]. For the completion purpose, we have implemented the Advanced Encryption Standard (AES) algorithm for encrypting a plain image [61] and then applied lossless compression on the cipher image. The simulation results agree with the analysis, i.e., for both datasets the resultant crypto-compressed image size expanded rather than being compressed. For the proposed method, two compression algorithms are being implemented in lossless mode as discussed in Section 3.3.2. In Method 1, the entropy encoder is directly applied to the shuffled image. The compression performance of the proposed hybrid image encryption-compression scheme on the shuffled image is similar to that of the lossless compression of unencrypted image. Therefore, performing compression after the shuffling stage preserves the compression efficiency in the encrypted domain. In addition, to analyze the effect of CRT based block representation of the input symbols on the compression efficiency; in Method 2, the entropy encoder is applied on the blocks. The block representation aids the entropy encoder stage of the lossless compression algorithm. The average compression savings have been improved from 8.33% to 18.06% on the KODAK dataset and from 5.76% to 15.45% on the UCID dataset. In addition, we

Table 7. Comparison of the proposed method in terms of compression savings (%).

Methods		KODAK Dataset	UCID Dataset
Encryption-Then-Compression	Lossless Compression	8.33	5.76
	[31]	4.54	2.94
	[33]	4.54	2.94
	AES-Lossless Compression	-2.84	-3.95
Proposed Methods	Method 1	8.33	5.76
	Method 2	18.06	15.45

have shown the compression savings across all images of the UCID dataset for the proposed Method 1 and Method 2 in Figure 8.

3.5. Chapter Summary

We proposed a novel hybrid image encryption-compression scheme that incorporates compression in the encryption process. The encryption is based on Chaos theory and consists of two steps: shuffling the pixel positions (permutation) and modifying the pixel values (substitution). The compression is applied on the shuffled image before the substitution process. Indeed, compression saving is possible, since the shuffled image still has the statistical characteristics of the original image. The output-compressed bitstream is grouped into 8-bit elements, which then goes through the substitution process to alter the intensity distribution of the original image. In the substitution stage,

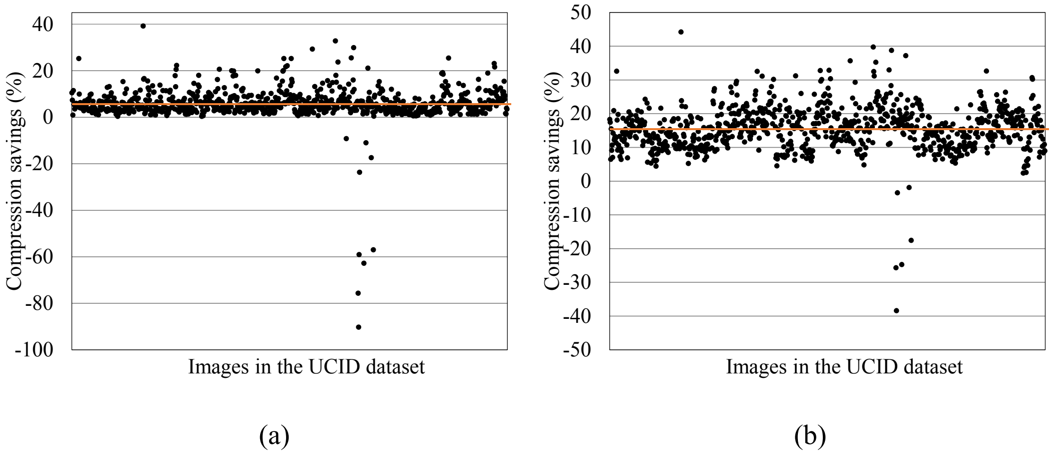


Figure 8. Compression savings for each image in the UCID dataset. (a) without and (b) with the proposed data-to-symbol map function in the compression module. The orange line shows the average compression savings across the dataset.

the 8-bit elements are XOR with substitution key. The XOR operation guarantees no expansion in the input bitstream, except in the case when padding is required to form the last 8-bit element. In such case, maximum of 7 bits increment may happen in the size of compressed bitstream, which is negligible. As shown in the results section, the proposed method achieved the necessary security level while preserving the lossless compression efficiency. In addition, the Huffman entropy encoder of the compression algorithm benefits from block representation of input data. We have proposed a data-to-symbol technique based on Chinese remainder theorem to represent adjacent pixel values as a block. The experimental results show the compression efficiency of block representation.

IV. PERCEPTUAL ENCRYPTION-BASED ENCRYPTION-THEN-COMPRESSION SCHEME

4.1. Introduction

The recent surge in cloud-based computing services, the popularity of social networking services, and cloud-based storage have motivated the exchange of a large amount of information, especially multimedia data, over the Internet. However, data transmission over public networks is always at risk of information leakage, and a large volume of data requires large bandwidth. One solution is to encrypt and compress the data. Traditional number theory and chaos theory-based full encryption algorithms [1], [61]–[63] have proven to be efficient in protecting multimedia data confidentiality during transmission. However, these algorithms are applicable for encrypting uncompressed raw images as they always perform stream encryption and/or scrambling of pixels values. However, they are not suitable for encrypting the JPEG compressed images as they may disturb the markers and render them uninterpretable. In addition, re-encoding the encrypted image as a JPEG image may increase the file size. Therefore, encryption of JPEG compressed images has some additional requirements compared to uncompressed images: (1) the encrypted image should be JPEG format-compatible, (2) there should be no or small increment in the encrypted image size, and (3) the encryption should provide a necessary level of security. Overall, an encryption algorithm should provide a balance between security and usability.

To meet these requirements, perceptual encryption (PE) of images has emerged as an alternative that provides the necessary level of security while allowing computation over encrypted data. PE-based algorithms reverse the conventional order of performing compression prior to encryption and are called encryption-then-compression (EtC) methods. It may seem inefficient in the sense that the encryption process may have destroyed the correlation present in a plain image that is exploited by a compression algorithm to provide reduction in image size. However, the EtC encryption algorithm protects only perceptual information of an image and preserves its intrinsic properties necessary for compression. The encryption algorithm of the EtC system is block-based and performs four steps: block permutation, block rotation, block inversion, and negative–positive transformation. The steps are computationally inexpensive and encrypt images in such a way that resulting cipher images retain their intrinsic properties necessary for compression. The EtC schemes are robust against various types of attacks, including brute-force and cipher-text-only attacks [36]. These schemes have the additional advantage of being JPEG-compatible, which makes them suitable for several applications, such as cloud-based photo storage and social networking services [37], [38], image retrieval systems in the privacy domain [64], and medical image services [7], [10].

Nonetheless, there is a tradeoff between compression and encryption efficiencies because of the block size choice. Several studies have improved the encryption efficiency of the CPE schemes. For example, [65] proposed a CPE scheme with an additional step to permute the blocks in the color channels for improved encryption efficiency. However, this scheme has a limitation on the keyspace size resulted from the choice of block size that is, a smallest block size that can be used is 16×16 to avoid distortion in the recovered image, when the JPEG chroma subsampling is being used. In [66] the authors proposed to process each color component independently for larger keyspace size. However, the methods are only compatible with the JPEG lossless compression standard. To deal with these issues, [37], [38] proposed to represent the input image as a pseudo grayscale image by combining the color channels along the horizontal or vertical direction; therefore, allowing the use of a smaller block size such as 8×8, thus improves the encryption efficiency. However, such representation degrades image quality and compression savings, and removes color information, which limits their applications.

To solve these limitations, we proposed inter and intra block processing for compressible PE methods (IIB–CPE). The method represents an input as a color image and performs block-level inter processing and sub-block-level intra processing on it. The intra block processing results in an inside–out geometric transformation that disrupts the symmetry of an entire block thus achieves visual encryption of local details while preserving the global contents of an image. IIB–CPE performs the encryption steps that only change correlation’s direction on a sub-block level, thereby improving the encryption efficiency. In other words, if the orientation of a sub-block in a block is changed, it is difficult to recover the correct orientation of the entire block without reconstructing the entire image.

Contributions. Our main contributions can be summarized as follows:

- We propose an efficient block-based compressible perceptual encryption algorithm, which eliminates the security vulnerabilities of existing PE methods. The proposed scheme uses inter and intra block processing, which allows a smaller block size, thus expanding the keyspace of the encryption algorithms and providing resistance against different attacks.
- The proposed method can provide security and bandwidth efficiency during image data transmission and can provide compression savings while enabling privacy-preserving photo storage.
- We explain why PE cipher images are compressible for the first time, to the best of our knowledge.
- We propose an extended jigsaw puzzle solver to accommodate the sub-block-level processing.
- We demonstrate the encryption efficiency and compression savings of the proposed method in comparison with the existing methods on different real image datasets.

4.2. Taxonomy of Compressible Perceptual Encryption Methods

In general, the encryption algorithm of a CPE scheme is block–based and consists of four steps: blocks permutation, blocks rotation, blocks inversion, and negative and positive transformation. There is an optional color channel shuffling step that is used when the input is a color image. The existing CPE methods can be classified based on their input image representation such as Color–CPE, Extended–CPE, inter and intra block processing–based CPE (IIB–CPE) and pseudo grayscale–based CPE (PGS–CPE) methods. In Color–CPE, Extended–CPE and IIB–CPE methods, an input color image is represented by its three-color components. Whereas in PGS–CPE methods, the color components of an input color image are concatenated along the horizontal or vertical direction to form a pseudo–grayscale image. An alternative classification of CPE methods can be based on their mode of processing for example, methods that transform an entire block include Color–CPE, Extended–CPE and PGS–CPE methods, and methods that incorporate sub–block processing include IIB–CPE methods. This CPE classification is beneficial when the input is a grayscale image. The following subsections present the related work of each category.

4.2.1. Color-CPE Methods

Watanabe et al., proposed a Color–CPE method that performs color channel shuffling step for better security and their method is compatible with the JPEG 2000 standard [67] and the motion JPEG 2000 standard [68]. The applications of their method have been further extended by Kurihara et al. to the JPEG standard [65], the motion JPEG standard [69], the JPEG XR standard [70] and lossless image compression standards [35]. The Color–CPE methods process image blocks with the same key in each color channel. The methods use block size of (16 × 16) in the encryption algorithm to take advantage of the JPEG chroma subsampling step for better compression savings without any adverse effect. The methods preserve the JPEG file format and almost the same compression savings. However, the use of the common key to encrypt each channel leaves the color distribution unaltered and the larger block size results in a smaller keyspace. This information makes the Color–CPE schemes vulnerable to JPS attack [66].

4.2.2. Extended-CPE Methods

To alter the color distribution in the Color–CPE methods efficiently, Imaizumi et al. [66], [71] proposed to process each color component individually in the permutation, rotation, inversion, and negative–positive transformation steps. This independent processing expands the keyspace size and modifies the color distribution significantly; however, this results in the JPEG format compatibility issues. The main reason is that the JPEG standard requires colorspace conversion prior to compression and the Extended–CPE methods are not suitable for this conversion function.

4.2.3. PGS-CPE Methods

In order to deal with the issue of Extended–CPE methods, Chuman et al., proposed in [37] to perform the JPEG colorspace conversion prior to the encryption process. In addition, they proposed to concatenate the color components along horizontal or vertical direction to form a pseudo grayscale image. This grayscale representation can benefit from the smallest allowable block size that is the JPEG performs a grayscale image compression on (8 × 8) block size. This use of small block size results in a larger keyspace size than the Color–CPE and Extended–CPE schemes. However, the PGS–CPE method proposed in [37] is not suitable with the JPEG chroma subsampling function. To deal with this issue, Sirichotedumrong et al. proposed in [38], [72] to perform both the JPEG colorspace conversion and chroma subsampling functions prior to the encryption. The idea is to down sample the color components after the colorspace conversion and concatenate them with the luminance component. In addition, they proposed custom quantization tables in [38] that can be used in the JPEG standard for better compression performance.

4.3. Block-Based Compressible Perceptual Encryption Methods

The main idea of the CPE methods is to divide an image into blocks and perform some geometric and color transformations on them in order to protect the image global contents. Such block level processing preserves the image local contents such as spatial correlation of the neighboring pixels within a block. This correlation can be exploited by an image compression algorithm for example, the JPEG standard, to compress the cipher images. A careful consideration of the block size is required to achieve a best tradeoff between the compression and encryption efficiencies. For example, in the JPEG standard the smallest allowable block size is (16 × 16) for color images compression and (8 × 8) for grayscale images compression. In general, the CPE methods consist of the following three steps:

Step 1. Input image representation:

An input color image $\boldsymbol{I}$ whose dimensions are specified by H rows, W columns and C components, can either be represented as a true color image $\boldsymbol{I}_{W,H,C}$ or a pseudo–grayscale image by concatenating the color components in either vertical direction as $\boldsymbol{I}_{(H\times C),W}$ or horizontal direction as $\boldsymbol{I}_{H,(W\times C)}$. On the other hand, when the input is a grayscale image $\boldsymbol{I}_{W,H}$ then this step is omitted.

Step 2. Block–based encryption:

The CPE methods perform geometric transformations to change blocks positions (blocks

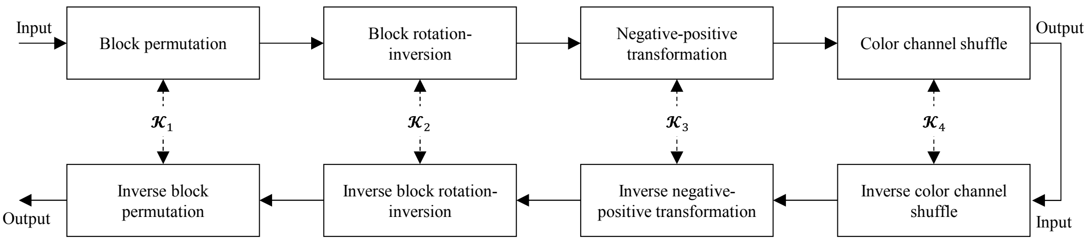

Input
Block permutation
Block rotation-inversion
Negative-positive transformation
Color channel shuffle
Output
$\mathcal{K}_1$
$\mathcal{K}_2$
$\mathcal{K}_3$
$\mathcal{K}_4$
Output
Inverse block permutation
Inverse block rotation-inversion
Inverse negative-positive transformation
Inverse color channel shuffle
Input

Figure 9. An illustration of block–based CPE encryption and decryption processes. Each $\mathcal{K}_i, i = 1, \ldots, 4$ is a set of keys used in each step to process the color channels.

permutation) and blocks orientations (blocks rotation and inversion), and color transformations (color channel shuffle and negative–positive transformation) to alter pixels values in the blocks. The encryption and decryption processes are shown in Figure 9.

Step 3. Compression:

The final step is to compress the cipher image using the JPEG image standard. The JPEG color or grayscale image compression mode is chosen based on the input image representation in Step 1, where $\mathcal{K}_i$ is the secret key used in the ith step.

Based on the input image representation, the PE methods can be classified as methods that represent the input as a color image and methods that represent the input as a pseudo grayscale image. The basic form of the first category is to process each color component with the same key, we named them as Color–CPE methods. These methods can be extended to process each color component independently (Extended–CPE) and to introduce sub–block level processing (IIB–CPE). The second category where the input is represented as a grayscale, is named as PGS–CPE methods.

4.3.1. Color-CPE Methods

In the Color–CPE algorithms, an image $\boldsymbol{I}_{H,W,C}$ with $H \times W$ pixels in $C = 3$ color channels, is divided into $L \times M$ blocks where $L = H/N$ and $M = W/N$. A cipher image can be generated as shown in Figure 10 and the procedure described is below:

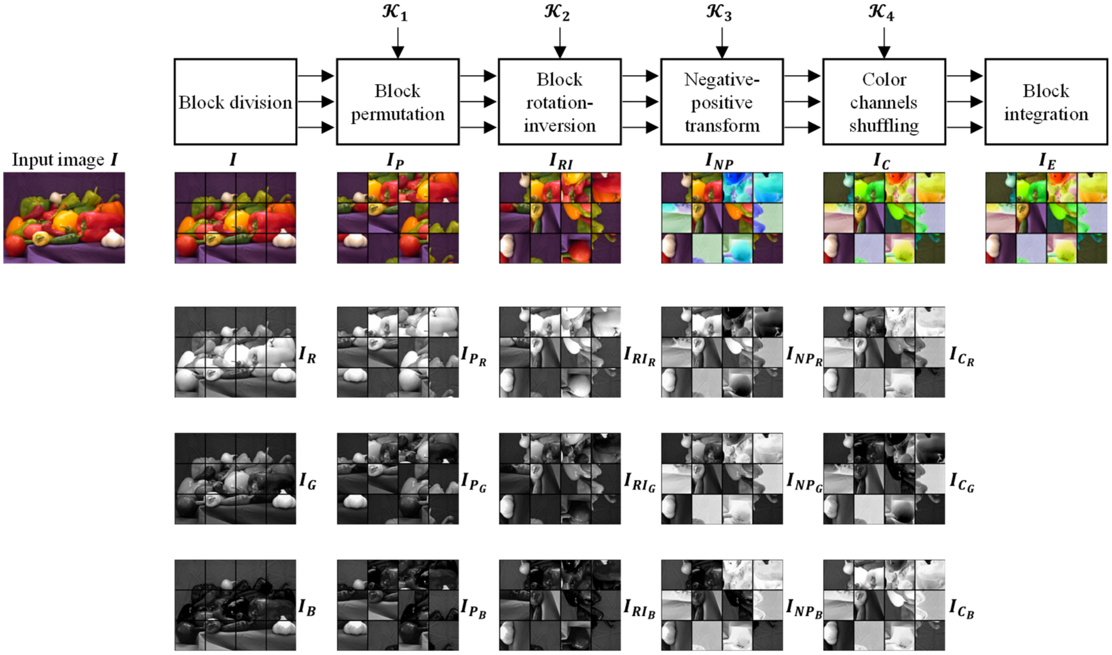


Figure 10. The encryption algorithm steps of a Color–CPE scheme. For visual analysis, the effect of each transformation function on the image is shown across each color channel. The keys $\mathcal{K}_i, i = 1, \dots, 4$ is a set of keys used in each step to process the color channels. Since, a common key is used to process each color channel, the blocks have the same appearances in each channel.

Step 1. Input image representation:

An input color image $\boldsymbol{I}$ whose dimensions are specified by H rows, W columns and C components, is represented as a true color image $\boldsymbol{I}_{H,W,C}$ in the RGB colorspace.

Step 2. Block–based encryption:

- The image $\boldsymbol{I}_{H,W,C}$ is divided into $L \times M$ blocks where $L = H/N$ and $M = W/N$, and each block has C color channels with N^2 pixels.
- Shuffle the blocks positions in the image using a secret key $\mathcal{K}_1$ generated randomly. The key size is equal to the number of blocks where each of its entries represent a block's new position

in the scrambled image.

- Change the blocks orientations in the shuffled image by a composite function of rotation and inversion transformations. This transformation is controlled by a randomly generated key $\boldsymbol{\mathcal{K}}_2$ where its entries represent rotation and inversion axis.

- Change the pixels values by applying a negative–positive transformation function to each pixel in a block randomly chosen by a key $\boldsymbol{\mathcal{K}}_3$. The $\boldsymbol{\mathcal{K}}_3$ is a binary key where the elements are uniformly distributed. The negative–positive transformation function for a block $\boldsymbol{B}$ is defined as:

$$\acute{p}_{s,t} = \begin{cases} p_{s,t}, & \boldsymbol{\mathcal{K}}_{3_i} = 0 \\ 255 - p_{s,t}, & \boldsymbol{\mathcal{K}}_{3_i} = 1 \end{cases}, \tag{37}$$

where $p_{s,t}$ $(s, t = 1, \cdots, N)$ is a pixel value in the block and $\acute{p}_{s,t}$ is its modified value.

- Shuffle the color components of each block using key $\boldsymbol{K}_4$. Each element of the $\boldsymbol{K}_4$ represents a unique permutation of the color channels.

Step 3. Compression

The final step is to JPEG compress the cipher image obtained in the previous step. Since the input was represented as a color image in the RGB colorspace (Step 1), the JPEG compression can be carried out in the color mode either using RGB or YCbCr colorspace. When a suitable block size is used during encryption, such as $N = 16$, then a user can benefit from the JPEG chroma subsampling for additional compression savings.

4.3.2. Extended-CPE Methods

The Extended-CPE methods extend the Color–CPE method to alter the color distribution better. The principal idea is to process each color component independently. A cipher image can be generated as shown in Figure 11 and the procedure is described below:

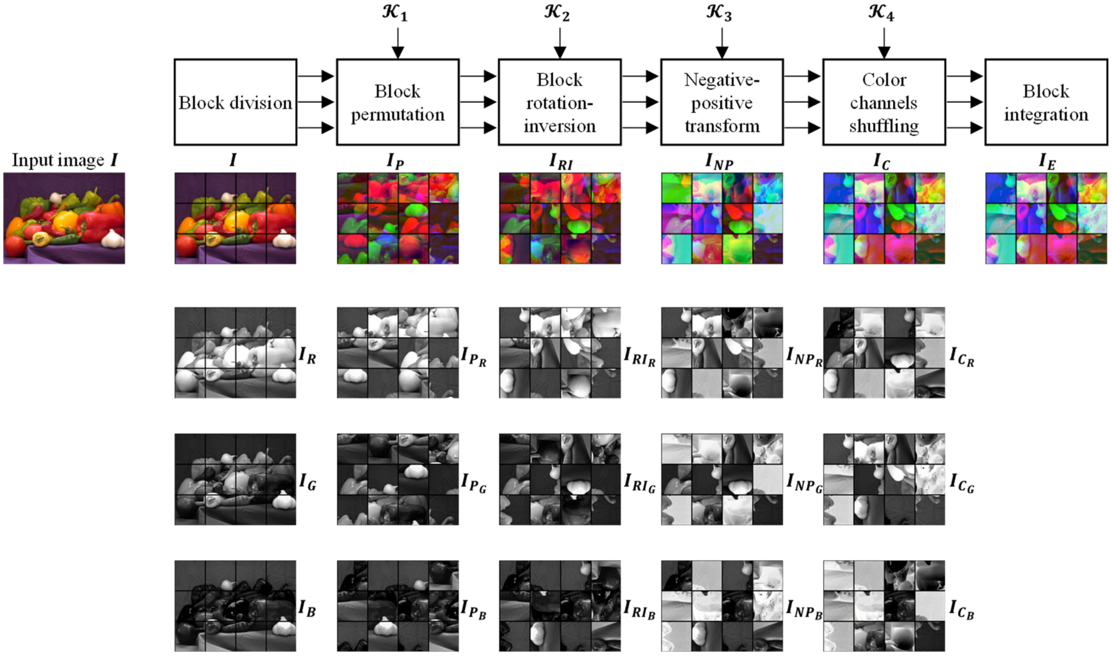


Figure 11. The encryption algorithm steps of an Extended–CPE scheme. For visual analysis, the effect of each transformation function on the image is shown across each color channel. The keys $\boldsymbol{\mathcal{K}_i, i = 1, \dots, 4}$ is a set of keys used in each step to process the color channels. Each color component is process independently; therefore, the blocks have different appearances in them.

Step 1. Input image representation:

An input color image $\boldsymbol{I}$ whose dimensions are specified by H rows, W columns and C components, is represented as a true color image $\boldsymbol{I}_{H,W,C}$ in the RGB colorspace.

Step 2. Block–based encryption:

- The image $\boldsymbol{I}_{H,W,C}$ is split into three individual color components such as $\boldsymbol{I}_{H,W,R}$, $\boldsymbol{I}_{H,W,G}$, and $\boldsymbol{I}_{H,W,B}$. Here, each component is independently divided into $L \times M$ blocks where $L = H/N$ and $M = W/N$, and each block has N^2 pixels. The image has a total of $L \times M \times 3$ blocks.

- Shuffle the blocks positions in each color component by using a secret key $\boldsymbol{\mathcal{K}}_1$ generated randomly. The blocks are processed independently; therefore, $\boldsymbol{\mathcal{K}}_1 = \{\boldsymbol{K}_1^R, \boldsymbol{K}_1^G, \boldsymbol{K}_1^B\}$ and $\boldsymbol{K}_1^R \neq \boldsymbol{K}_1^G \neq \boldsymbol{K}_1^B$.
- Rotate and invert each block to change their orientations. Each block is processed independently by using a randomly generated key $\boldsymbol{\mathcal{K}}_2$ and $\boldsymbol{\mathcal{K}}_2 = \{\boldsymbol{K}_2^R, \boldsymbol{K}_2^G, \boldsymbol{K}_2^B\}$ where $\boldsymbol{K}_2^R \neq \boldsymbol{K}_2^G \neq \boldsymbol{K}_2^B$.
- Reverse the pixels values in a block by applying a negative–positive transformation function as in (37). The random key $\boldsymbol{\mathcal{K}}_3$ process each color component independently and $\boldsymbol{\mathcal{K}}_3 = \{\boldsymbol{K}_3^R, \boldsymbol{K}_3^G, \boldsymbol{K}_3^B\}$ where $\boldsymbol{K}_3^R \neq \boldsymbol{K}_3^G \neq \boldsymbol{K}_3^B$.
- Shuffle the color components in each block using key $\boldsymbol{K}_4$.

Step 3. Compression:

The final step is to JPEG compress the cipher image obtained in the previous step. Since the input was represented as a color image in the RGB colorspace (Step 1), the JPEG compression can be carried out in the color mode. However, because of the independent color component processing the compression of the cipher image should be carried out in lossless mode such as RGB colorspace and without chroma subsampling.

4.3.3. PGS-CPE Methods

The PGS–CPE schemes are proposed to deal with format compatibility and chroma sub–sampling issues in the color–based CPE methods. The principal idea is to represent the input color image in a pseudo grayscale form in order to benefit from the allowable smallest block size in the JPEG standard for better encryption efficiency. A cipher image can be generated as illustrated in Figure 12 and the procedure is described below:

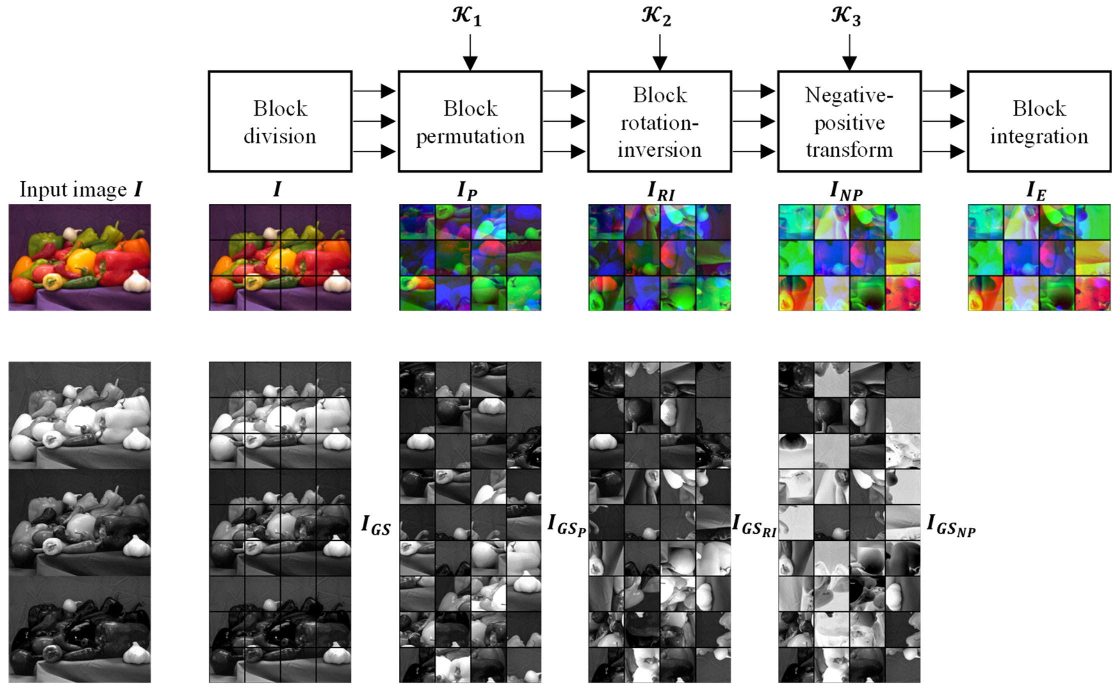


Figure 12. The encryption algorithm steps of a PGS–CPE scheme. The pseudo grayscale representation was obtained by concatenating the RGB components along the vertical axis. For visual analysis, the effect of each transformation function on the image is shown. The keys $\boldsymbol{K_i, i = 1, \dots, 3}$ is used in each step to transform the blocks. The appearances of the blocks is similar to the methods that process the color component independently.

Step 1. Input image representation:

An input color image $\boldsymbol{I}$ in the RGB colorspace, whose dimensions are specified by H rows, W columns and C components $\boldsymbol{I}_{H,W,C}$, is converted into YCbCr colorspace. The three components $\boldsymbol{Y}_{H,W}$, $\boldsymbol{Cb}_{H,W}$, and $\boldsymbol{Cr}_{H,W}$ are concatenated either in horizontal direction to form an image $\boldsymbol{I}_{H,(C\times W)}$ or vertical direction to form an image $\boldsymbol{I}_{(C\times H),W}$ as shown in Figure 13. However, for the color subsampling function (for example, a ratio of 4:2:0), the chroma components are down sampled as $\acute{\boldsymbol{Cb}} = \boldsymbol{Cb}_{H/2,W/2}$ and $\acute{\boldsymbol{Cr}} = \boldsymbol{Cr}_{H/2,W/2}$. The three components $\boldsymbol{Y}_{H,W}$, $\acute{\boldsymbol{Cb}}_{H/2,W/2}$, and $\acute{\boldsymbol{Cr}}_{H/2,W/2}$ are concatenated either in horizontal direction to form an image $\boldsymbol{I}_{H,(C\times(W/2))}$ or vertical

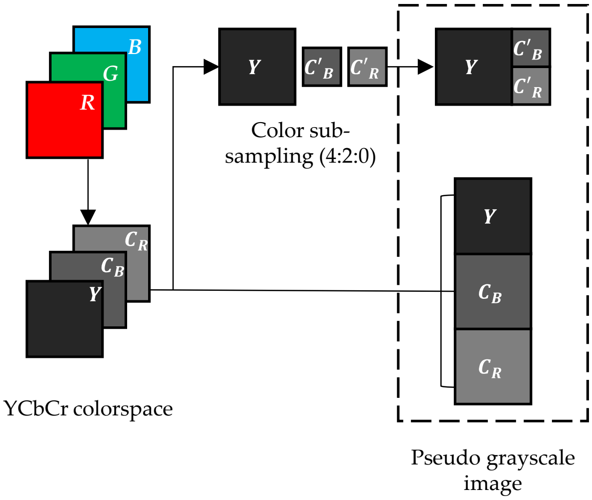


Figure 13. The pseudo–grayscale image representation generation for both chroma subsampling and without chroma subsampling.

direction to form an image $\boldsymbol{I}_{(C\times(H/2)),W}$. Here, we assumed that the input image $\boldsymbol{I}_{H,W,C}$ is represented in pseudo grayscale form without the chroma subsampling as $\boldsymbol{I}_{H,(C\times W)}$.

Step 2. Block–based encryption:

- The image $\boldsymbol{I}_{H,(C\times W)}$ is divided into $L \times M$ blocks where $L = H/N$ and $M = (C \times W)/N$, and each block has N^2 pixels.
- Shuffle the blocks positions in the image using a secret key $\boldsymbol{K}_1$ generated randomly.
- Change the blocks orientations in the shuffled image by a composite function of rotation and inversion transformations. This transformation is controlled by a randomly generated key $\boldsymbol{K}_2$.
- Change the pixels values by applying a negative–positive transformation function to each pixel in a block chosen using a random key $\boldsymbol{K}_3$ as in (37).

Step 3. Compression:

The final step is to JPEG compress the cipher image obtained in the previous step. Since the input was represented as a grayscale image, the JPEG compression can be carried out in the grayscale mode by using either the luminance or chrominance standard table in the quantization step.

4.4. Proposed Method

4.4.1. Motivation

The Extended–CPE and PGS–CPE methods have improved the security efficiency of the Color–CPE methods as the color distribution are scrambled significantly and keyspace is expanded (especially, in PGS–CPE method). However, these schemes have color image as an input a pre–requisite for example, to achieve a large number of blocks, the individual color component processing (Extended–CPE methods) and the pseudo grayscale image representation (PGS–CPE methods) are only possible when the input is a color image. This advantage of these methods diminishes when the input image is a grayscale image with only one channel [6]. Overall, in the CPE schemes – block–based perceptual encryption methods –there is an efficiency tradeoff between encryption and compression efficiencies because of the choice of block size. Specifically, a block size of no smaller than (16 × 16) and (8 × 8) should be used when considering the compression efficiency of the JPEG standard for color and grayscale images, respectively.

4.4.2. Proposed Extended BPE Methods (EBPE)

To overcome the limitations of the existing CPE methods, we propose an inside–out transformation function that performs the rotation and inversion step on a sub–block level. The main idea of the proposed method is to benefit from smaller block size on a sub-block level as suggested in [7], [10]. Based on this principle, there are several possible extensions of conventional BPE methods as described below.

4.4.2.1. EBPE-1

An example EBPE–1 encrypted image is shown in Figure 14. (c). Compared to BPE cipher image, the EBPE–1 has better scrambled image information. For an image I with $H \times W \times C$ pixels, divided into n blocks each with $B_h \times B_w$ elements, the cipher image is generated as:

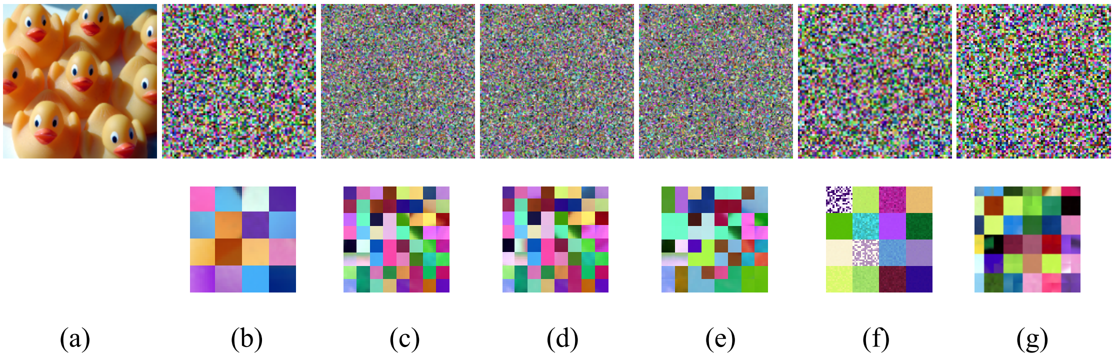


Figure 14. Example image from the dataset. (a) is original image and (b) – (f) are its cipher images obtained from BPE and extended EBPE 1 – 5, respectively. The bottom right corner in each cipher image is magnified and shown below.

Step 1. Shuffle the blocks positions by using a randomly generated secret key K_1^1.

Step 2. Divide each block into sub-blocks of size $B_{h'} \times B_{w'}$ such that each block has l blocks.

Step 3. Shuffle the sub-blocks positions within a block by using a randomly generated key K_2^1.

Step 4. Rotate and invert each sub-block by using a random key K_3^1 in order to change their orientations.

Step 5. Randomly apply negative-positive transformation to each sub-block using a uniformly distributed key K_4^1 as in (1).

Step 6. Randomly shuffle the sub-blocks in the three-color channels by using key K_5^1.

Note that performing Steps 4 – 6 on an entire block does not result in new transformations; however, it only reverses the transformations of the sub-blocks. For example, changing orientation of an entire block will scramble the positions of sub-blocks in that block, which has already been performed in Step 2. Similarly, performing color transformations on an entire block will reverse Step 5 and 6 transformations on the sub-blocks.

4.4.2.2. EBPE-2

The EBPE–2 extension omits the sub-block scrambling as performed by EBPE–1 while performing the rest of the steps on a sub-block level. An example EBPE–2 cipher image is shown in Figure 14. (d). For an image I with $H \times W \times C$ pixels, divided into n blocks each with $B_h \times B_w$ elements, perform the following steps:

Step 1. Shuffle the blocks positions by using a randomly generated secret key K_1^2.

Step 2. Divide each block into sub-blocks of size $B_{h'} \times B_{w'}$ such that each block has l blocks.

Step 3. Rotate and invert each sub-block by using a random key K_2^2 in order to change their orientations.

Step 4. Randomly apply negative-positive transformation to each sub-block using a uniformly distributed key K_3^2 as in (1).

Step 5. Randomly shuffle the sub-blocks in the three-color channels by using key K_4^2.

4.4.2.3. EBPE-3

This extension preserves the same information in each color channel; therefore, the color channel shuffling step is performed on a block level instead of sub-block. An example of EBPE–3 cipher image is shown in Figure 14. (e). It can be seen in that color information in each channel is preserved on a block level. For an image I with $H \times W \times C$ pixels, divided into n blocks each with $B_h \times B_w$ elements, perform the following steps:

Step 1. Shuffle the blocks positions by using a randomly generated secret key K_1^3.

Step 2. Randomly shuffle the blocks in the three-color channels by using key K_2^3.

Step 3. Divide each block into sub-blocks of size $B_{h'} \times B_{w'}$ such that each block has l blocks.

Step 4. Shuffle the sub-blocks positions within a block by using a randomly generated key K_3^3.

Step 5. Rotate and invert each sub-block by using a random key K_4^3 in order to change their orientations.

Step 6. Randomly apply negative-positive transformation to each sub-block using a uniformly distributed key K_5^3 as in (1).

4.4.2.4. EBPE-4

In the previous extensions, the local contents of the image are preserved on a sub-block level. This extension disrupts the contents of a block on the smallest possible unit that is pixels in the block, to achieve better security. An example of EBPE–4 cipher image is shown in Figure 14. (f). For an image I with $H \times W \times C$ pixels, divided into n blocks each with $B_h \times B_w$ elements, perform the following steps:

Step 1. Shuffle the blocks positions by using a randomly generated secret key K_1^4.

Step 2. Scramble each pixel position in the block by using a randomly generated key K_2^4.

Step 3. Randomly apply negative-positive transformation to each block by using a uniformly distributed key K_3^4 as in (1).

Step 4. Randomly shuffle the blocks in the three-color channels by using key K_4^4.

Note that changing the orientation of an entire block does not result in new transformations. For example, rotating a block will only change the positions of pixels in that block, which has already been performed in Step 2. Therefore, block rotation-inversion step is omitted from this extension.

4.4.2.5. EBPE-5

An example EBPE–5 cipher image is shown in Figure 14. For an image I with $H \times W \times C$ pixels, divided into n blocks each with $B_h \times B_w$ elements, perform the following steps:

Step 1. Shuffle the blocks positions by using a randomly generated secret key K_1^5.

Step 2. Randomly apply negative-positive transformation to each sub-block using a uniformly distributed key K_2^5 as in (1).

Step 3. Randomly shuffle the blocks in the three-color channels by using key K_3^5.

Step 4. Divide each block into sub-blocks of size $B_{h'} \times B_{w'}$ such that each block has l blocks.

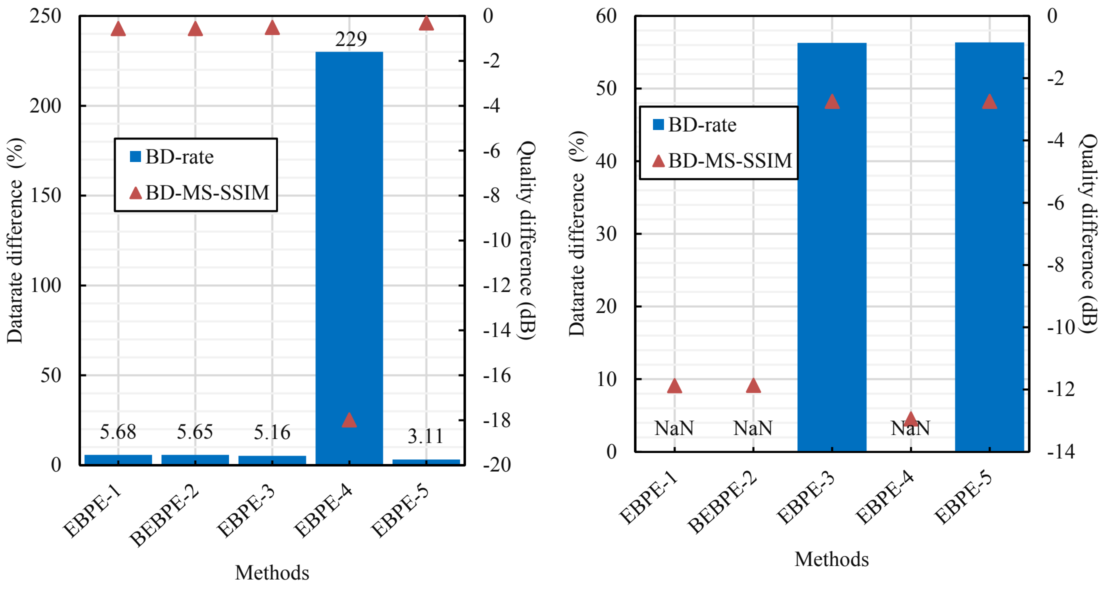
Step 5. Rotate and invert each sub-block by using a random key K_4^5 in order to change their orientations.

4.4.2.6. Intermediate Analysis

This section first presents compression and encryption performance of conventional BPE [65] and proposed EBPE methods. The analysis was carried out on Tecnick sampling dataset [73], which consists of 120 color images of 1200×1200 dimensions. For the compression of cipher images, the JPEG standard algorithm [21] was used with and without chroma sub-sampling. In the quantization step, the standard luminance and chrominance quantization tables were used. In the second set of experiments, we have analyzed classification performance of a DL model on the plain and cipher images obtained from different PE methods.

4.4.2.6.1. Compression Analysis

For the compression efficiency analysis, Figure 15. shows the rate distortion curves (RD) for bitrate savings (bpp) against image quality in terms of peak-signal-to-noise ratio (PSNR) (dB). The RD curves are for the JPEG quality factors of $Q_f = \{80, 85, 90, 95, 100\}$. The block size is 16×16 for BPE and entire block processing of proposed EBPE, while 8×8 is used for sub-block processing of proposed EBPE. The average compression savings and quality differences of the methods are shown in Figure 15. (a) and (b), with and without chroma sub-sampling. When the images are compressed without sub-sampling, the difference in bit rate and image quality is almost negligible for BPE, EBPE–1, EBPE–2, and EBPE-3 compared to plain images compression. However, EBPE–4 drastically decreased the compression savings and image quality. When the compression is carried out with sub-sampling, the image quality of proposed EBPE methods is almost reduced by 11 dB. However, the compression savings remain almost the same except for EBPE–4. Overall, the results show that sub-block processing of the extensions EBPE–1, EBPE–2, and EBPE–3 has negligible effect on compression savings; however, when high quality images are required then the methods are not adequate.



(a) (b)

Figure 15. Compression savings and recovered image quality analysis of plain and cipher images compression in terms of Bjøntegaard delta measures. The JPEG compression was performed without and with chroma sub-sampling in (a) and (b), respectively.

For visual analysis the recovered images from different methods are shown in Figure 16. When the images are compressed without chroma sub-sampling then no visible distortion can be seen in the recovered images across all methods. However, when the JPEG algorithm is implemented with chroma sub-sampling, then the sub-block processing results in block artifacts visible in the recovered images. For EBPE 1 – 3, the distortion appeared to be line on the borders of the blocks, whereas for EBPE–4 the whole block appearance is changed.

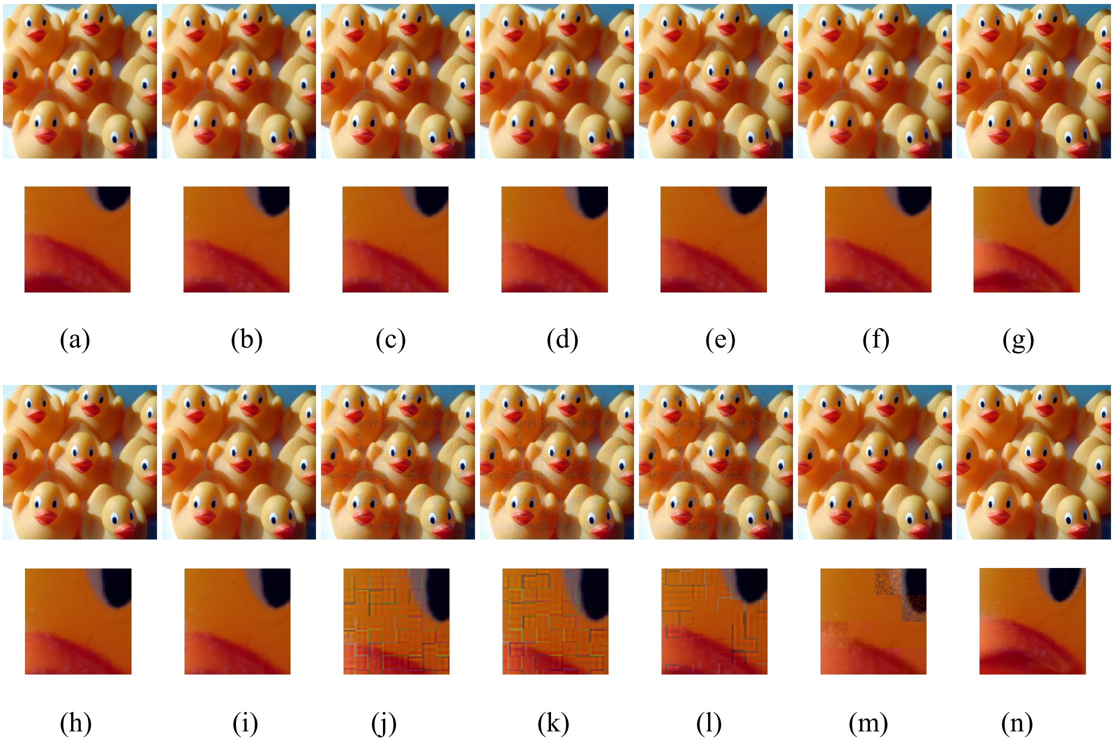


Figure 16. Visual analysis of decoded images of Figure 14. (a). The JPEG compression is performed in lossless mode (a) – (g) and lossy mode (h) – (n). (a) and (h) are recovered by decompressing the plain compressed image while (b) – (g) and (i) – (n) are recovered from the compressed cipher images for BPE and EBPE 1 – 5, respectively. The center duck left eye in each recovered image is magnified and shown below it.

4.4.2.6.2. Encryption Analysis

This section presents robustness of PE methods against brute-force attack in terms of key space. The security efficiency of symmetric encryption algorithms has direct relation with the algorithms key size that is, larger the key size, the more secure the scheme is [74]. Each step in the PE algorithm has a random key and its size depends on the number of blocks the image has been divided. For an image I with $H \times W \times C$ pixels, divided into blocks of size $B_h \times B_w$ pixels, the number of blocks n are given as

$$n = \frac{H}{B_h} \times \frac{W}{B_w}, \tag{38}$$

and when the blocks are divided into smaller blocks of size $B_{h'} \times B_{w'}$ pixels, then the number of sub-blocks m in the image is

$$m = \frac{H}{B_{h'}} \times \frac{W}{B_{w'}}, \tag{39}$$

and the number of sub-blocks l in a block is given as

$$l = \frac{B_h}{B_{h'}} \times \frac{B_w}{B_{w'}}. \tag{40}$$

For PE methods, the keyspace can be derived as a product of key sizes selected in each encryption step. For example, the keyspace $K_{[61]}$ of conventional BPE is given as

$$\begin{gathered} K_{[61]} = K_1 \cdot K_2 \cdot K_3 \cdot K_4 \\ = n! \cdot 8^n \cdot 2^n \cdot 3!^n. \end{gathered} \tag{41}$$

The proposed EBPE methods incorporate sub-block processing for better security as opposed to BPE that only performs block level processing. The security efficiency is because of the fact that the recovery of sub-block transformations (for example, orientation and color transformations) does not necessarily result in the correct appearance of their entire block [7]. For EBPE–1, the keyspace $K_{[EBPE-1]}$ is given as

$$\begin{gathered} K_{[EBPE-1]} = K_1^1 \cdot K_2^1 \cdot K_3^1 \cdot K_4^1 \cdot K_5^1 \\ = n! \cdot (n \cdot l!) \cdot (8^n \cdot 8^m) \cdot (2^n \cdot 2^m) \cdot (3!^n \cdot 3!^m). \end{gathered} \tag{42}$$

Note that sub-block permutation key $K_2^1 \neq m!$ as the sub-blocks positions in each block (that is, $l!$) can be recovered independent of other blocks. The product term $(n \cdot l!)$ shows that the process is repeated for each block in the image. Similarly, the product terms in keys $K_3^1, K_4^1,$ and K_5^1 indicate that once the sub-block transformations are recovered, then the entire block transformations should be recovered as well. When the sub-blocks permutation is omitted then the keyspace $K_{[EBPE-2]}$ for EBPE–2 is given as

$$\begin{aligned} K_{[EBPE-2]} &= K_1^2 \cdot K_2^2 \cdot K_3^2 \cdot K_4^2 \\ &= n! \cdot (8^n \cdot 8^m) \cdot (2^n \cdot 2^m) \cdot (3!^n \cdot 3!^m). \end{aligned} \tag{43}$$

For EBPE–3, the keyspace $K_{[EBPE-3]}$ is given as

$$\begin{aligned} K_{[EBPE-3]} &= K_1^3 \cdot K_2^3 \cdot K_3^3 \cdot K_4^3 \cdot K_5^3 \\ &= n! \cdot 3!^n \cdot (n \cdot l!) \cdot (8^n \cdot 8^m) \cdot (2^n \cdot 2^m). \end{aligned} \tag{44}$$

For EBPE–4, the keyspace $K_{[EBPE-4]}$ is given as

$$\begin{aligned} K_{[EBPE-4]} &= K_1^4 \cdot K_2^4 \cdot K_3^4 \cdot K_4^4 \\ &= n! \cdot (n \cdot (h \times w)!) \cdot (2^n \cdot 2^m) \cdot (3!^n \cdot 3!^m). \end{aligned} \tag{45}$$

For EBPE–5, the keyspace $K_{[EBPE-5]}$ is given as

$$\begin{aligned} K_{[EBPE-5]} &= K_1^5 \cdot K_2^5 \cdot K_3^5 \cdot K_4^5 \\ &= n! \cdot 2^n \cdot 3!^n \cdot (8^n \cdot 8^m). \end{aligned} \tag{46}$$

Based on (41) – (46) it can be seen that the proposed EBPE method has larger keyspace than BPE; therefore, can resist brute-force attacks. Among the proposed extensions, EBPE–4 has the largest keyspace.

4.4.3. Principal Design: Inside-Out Transformation

In natural images, on average, eight pixels are spatially correlated in either direction; therefore, in the JPEG compression algorithm a block size of (8 × 8) is used for better compression, and for manageable computational complexity and memory requirements. For compressibility of the perceptually encrypted images, neighboring pixels must have the same correlation as the original images. Therefore, in conventional methods, encryption steps are inter block processes that transform an entire block to preserve the intrinsic properties of an image as shown in Figure 17. However, different steps of the scheme affect the correlation differently. For example, block permutation and negative–positive transformation steps change the correlation coefficient, whereas rotation and inversion only change the correlation direction. Therefore, performing the later steps at a sub-block level (intra block process) may preserve compression savings while improving encryption efficiency.

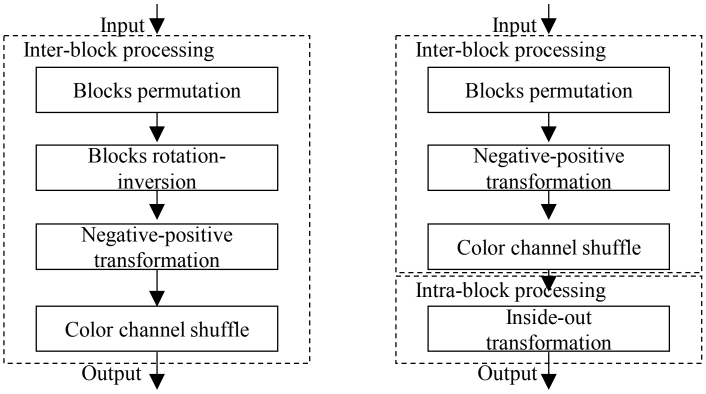


Figure 17. Block-based perceptual encryption algorithm. The conventional perceptual encryption scheme (left). The proposed Inter and Intra Block processing based perceptual encryption scheme (right).

In addition, the JPEG color sampling artifacts can be avoided since the correlation is preserved within a block. The basic principle is to divide blocks into sub-blocks and implement scrambling inside each block to achieve visual encryption of local details. The goal is to preserve global content such as spatial information and correlation among adjacent pixels, which can be used to enable several applications in encryption domain. The intra block processing results in an inside–out transformation that disrupts symmetry of an entire block as opposed to inter block processing of conventional methods. The geometric transformations performed only on an entire block are rigid motions that preserve symmetry of a block, that is, pixels on edges remain the same. As discussed earlier, existing PE methods have a prerequisite of color image as an input for better security. As opposed to them, the encryption efficiency of a proposed method is independent of input image representation because of the intra block processing. Therefore, sub-block operations make the proposed method suitable for both grayscale and color images.

4.4.4. Proposed Compressible PE Method

IIB–CPE consists of the following block-based steps:

Step 1. Divide the input image I_{RGB} with $\mathrm{W} \times \mathrm{H} \times \mathrm{C}$ pixels into non-overlapping blocks, each with $B_{\mathrm{M}} \times B_{\mathrm{N}}$ pixels, and permute them using a randomly generated secret key K_1^{inter}.

Step 2. Apply a negative–positive transformation to each pixel in a block randomly chosen by a uniformly distributed binary key K_2^{inter} as in Equation (37).

Step 3. Shuffle the blocks among the color channels using key K_3^{inter} where each entry represents a unique permutation of the color components.

Step 4. Perform inside–out transformation of an entire block as:

- Divide each block into sub–blocks of size $B_{\mathrm{M}_{intra}} \times B_{\mathrm{N}_{intra}}$ pixels;
- Apply rotation–inversion randomly to each sub-block using a key K_4^{intra} where each entry represents rotation degrees and flipping axis.

Step 5. Finally, apply JPEG compression to the cipher image.

The proposed PE method is a symmetric-key algorithm that requires the same set of keys used for both encryption of plain images and decryption of cipher images. Figure 18.a shows an example image from Tecnick dataset and its cipher images (b) and (c–e) obtained from Color–EtC and proposed methods, respectively. For visual analysis, the bottom left corner in each cipher image is enlarged. It can be seen that the proposed method achieves high visual encryption of local details. According to the application requirements, the encryption level can be controlled by performing the above encryption steps only on selected blocks. The shuffling key K_1 consists of blocks permutations that map each block to a random location in the output image. Scrambling only selected blocks can retain the global information of an image. Similarly, the key K_i where $i \in \{2,3,4\}$ of the ith step consists of an element (for example, 0) that represents an identity map. Therefore, increasing the number of such elements can decrease the level of encryption. In addition, the use of larger sub-blocks can preserve local information inside a block.

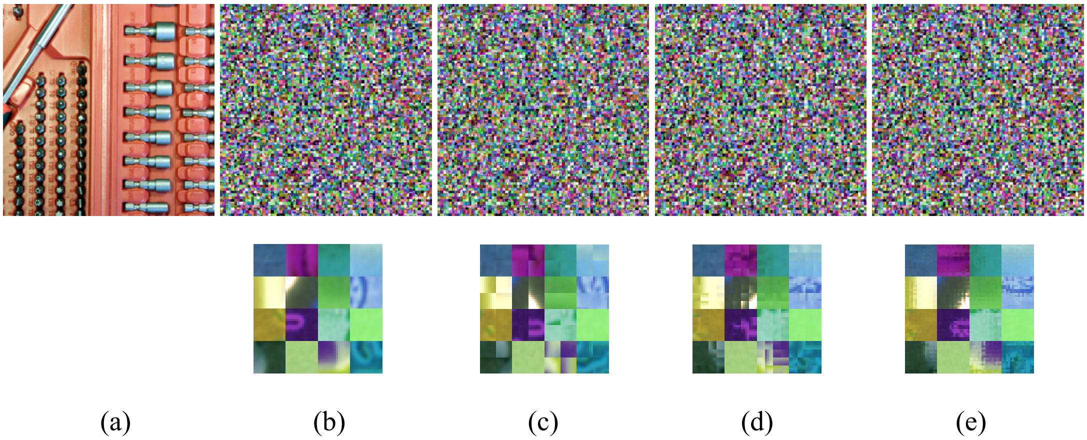


Figure 18. Visual analysis of inter and intra block processing on example image from Tecnick dataset. (a) Plain image (b) Conventional EtC method (16x16), (c)(d)and (e) are encrypted image of proposed method with sub-block size 8x8, 4x4 and 2x2, respectively. The last row shows enlarged image of left-bottom corner in each encrypted image. Compared to conventional methods proposed method achieves visual encryption of local details.

4.5. Simulation Results and Analysis

In this section, we first evaluate compression savings and encryption efficiency of the IIB–CPE. For the evaluation, we conducted our experiments on two datasets. The Tecnick sampling dataset [73] comprises 120 true color images with dimensions of 1200 × 1200. For the baseline methods, we implemented compressible PE algorithms proposed in [65], [69] (Color–EtC), and [37], [38] (GS–EtC). For the JPEG software, we have used the implementation provided in [75].

4.5.1. Compression Analysis

4.5.1.1. Compressibility–Energy Compaction Analysis

At the core of JPEG is the discrete cosine transform (DCT), which reduces the data correlation and provides a compact representation of a large amount of information as few data samples. The JPEG standard divides an image into $\mathrm{N}^2 = 8^2$ blocks of pixels, with each block producing 64 coefficients. The DCT–II for a block, where $I(i,j)$ is the pixel value at position (i,j), is defined as:

$$F(u,v)=\alpha(u)\alpha(v)\sum_{i=0}^{\mathrm{N}-1}\sum_{j=0}^{\mathrm{N}-1}I(i,j)\times\cos\left[\frac{(2i+1)\pi u}{2\mathrm{N}}\right]\cos\left[\frac{(2j+1)\pi v}{2\mathrm{N}}\right], \tag{47}$$

where

$$\alpha(x)=\begin{cases}\sqrt{\dfrac{1}{N}} & x=0\\ \sqrt{\dfrac{2}{N}} & x>0\end{cases}.$$

and $F(u,v)$ is the corresponding computed DCT value. For the DC coefficient, $u=v=0$ and (47) can be simplified as

$$F(0,0)=1/\mathrm{N}\sum\sum I(i,j),$$

which gives the average value of pixels in the image block. Therefore, the DC coefficient is independent of the pixel position. The remaining 63 values are AC coefficients that correspond to a progressive increase in frequency in both the horizontal and vertical directions. When the input image is divided into multiple blocks, the DCT can be independently computed for each block. In each calculation, the basis function points remain the same, whereas the pixel values are changed. Therefore, an efficient method is to pre-compute the function points and multiply them by each block to obtain DCT (D) as

$$D=TMT', \tag{48}$$

where M is the image data and T is the DCT matrix obtained as

$$T_{i,j}=\begin{cases}\dfrac{1}{\sqrt{\mathrm{N}}}, & if\ \ i=0,\\ \sqrt{\dfrac{2}{\mathrm{N}}}\cos\left[\dfrac{(2j+1)i\pi}{2\mathrm{N}}\right], & otherwise.\end{cases}$$

The multiplication of T on the left transforms the rows, and T' on the right transforms the columns of M. The product PM represented by P is a linear combination of matrix T columns with weights given by matrix M columns. The matrix M with c columns, each with r elements, can be represented in a column–vector form as $M = [\boldsymbol{M}_0, \boldsymbol{M}_1, \ldots, \boldsymbol{M}_{\mathrm{c}-1}]$, where for $i = 0, 1, \cdots, \mathrm{c} - 1$, $\boldsymbol{M}_i$ denotes the ith column of M, and for $k = 0, 1, \cdots, \mathrm{r} - 1$, m_{ki} denotes kth element of the ith column. The product P is obtained as $P = [T\boldsymbol{M}_0, T\boldsymbol{M}_1, \ldots, T\boldsymbol{M}_{\mathrm{c}-1}]$, where the i^{th} column of P is $\boldsymbol{P}_i = T\boldsymbol{M}_i$ and is defined as a product of matrix T with the ith column of M. It is calculated by

$$\boldsymbol{P}_i = \sum_{k=0}^{r-1} \boldsymbol{T}_{*k} m_{ki}. \tag{49}$$

This representation simplifies the change in the product matrix elements with respect to weight. For example, in (49), the weights belong to an image distribution, which has the intrinsic property of being highly correlated; thus, the adjacent pixels have smaller differences. Therefore, swapping the pixel positions in a block by intra block processing has a smaller effect on the product of the two matrices.

For a better understanding of the energy compaction analysis, we extracted two 8 × 8 blocks from the standard Lena image, and both blocks have different correlation coefficients. In the first image block, the horizontal correlation factor is $\sigma_h = 0.95$ and the vertical correlation factor is $\sigma_v = 0.96$, whereas in the second image block, the horizontal correlation factor is $\sigma_h = 0.49$ and the vertical correlation factor is $\sigma_v = 0.52$. The DCT transformation of the original and scrambled image blocks are shown in Figure 19 (a–d) and Figure 19 (e–h), respectively. The scrambled images in Figure 19 (b,f) were obtained by changing the entire block orientation (that is rotation by 90°). The scrambled images in Figure 19 (c,d,g,h) were obtained by dividing the blocks into sub-blocks and then changing the orientations of the sub-blocks randomly. In this example, one sub-block was rotated at 90° and one sub-block was flipped over the vertical axis. It can be seen in Figure 19 (b,f) that because of the entire block transformation, the DCT coefficient values remain the same, and only their positions change. Here, the DCT matrix obtained is equivalent to the diagonal flip of the original matrix. On the other hand, the sub-block processing changed the DCT coefficient values, as shown in Figure 19 (c,d,g,h). However, the DCT transformation in the JPEG algorithm is followed by a quantization step, to reduce each coefficient value by dividing each value $F(u,v)$ in (47) by the corresponding quantization table element $qt_{\mathrm{QF}}(u,v)$ as

$$F'^{(u,v)} = \left\lfloor \frac{F(u,v)}{qt_{\mathrm{QF}}(u,v)} \right\rfloor, \tag{50}$$

where function $\lfloor\cdot\rfloor$ rounds a value to the closest smallest integer and $F'(u,v)$ is the quantized value. Since the quantization reduces the values to small integers, the JPEG quantization step significantly reduced the difference in the DCT coefficients of the original and transformed image blocks, as shown in Figure 19. In the quantization step, the standard luminance quantization table with $qf = 80$ was used. In fact, during intermediate encoding, the zigzag scan of the DCT matrix resulted in almost the same number of zero AC coefficients, which can be encoded as the JPEG EOB identifier in the same manner in all of the cases. Therefore, the sub-block processing has a negligible impact on the energy compaction of the DCT and it allows the quantizer to discard high-frequency coefficients without introducing any visual distortion in the recovered image.

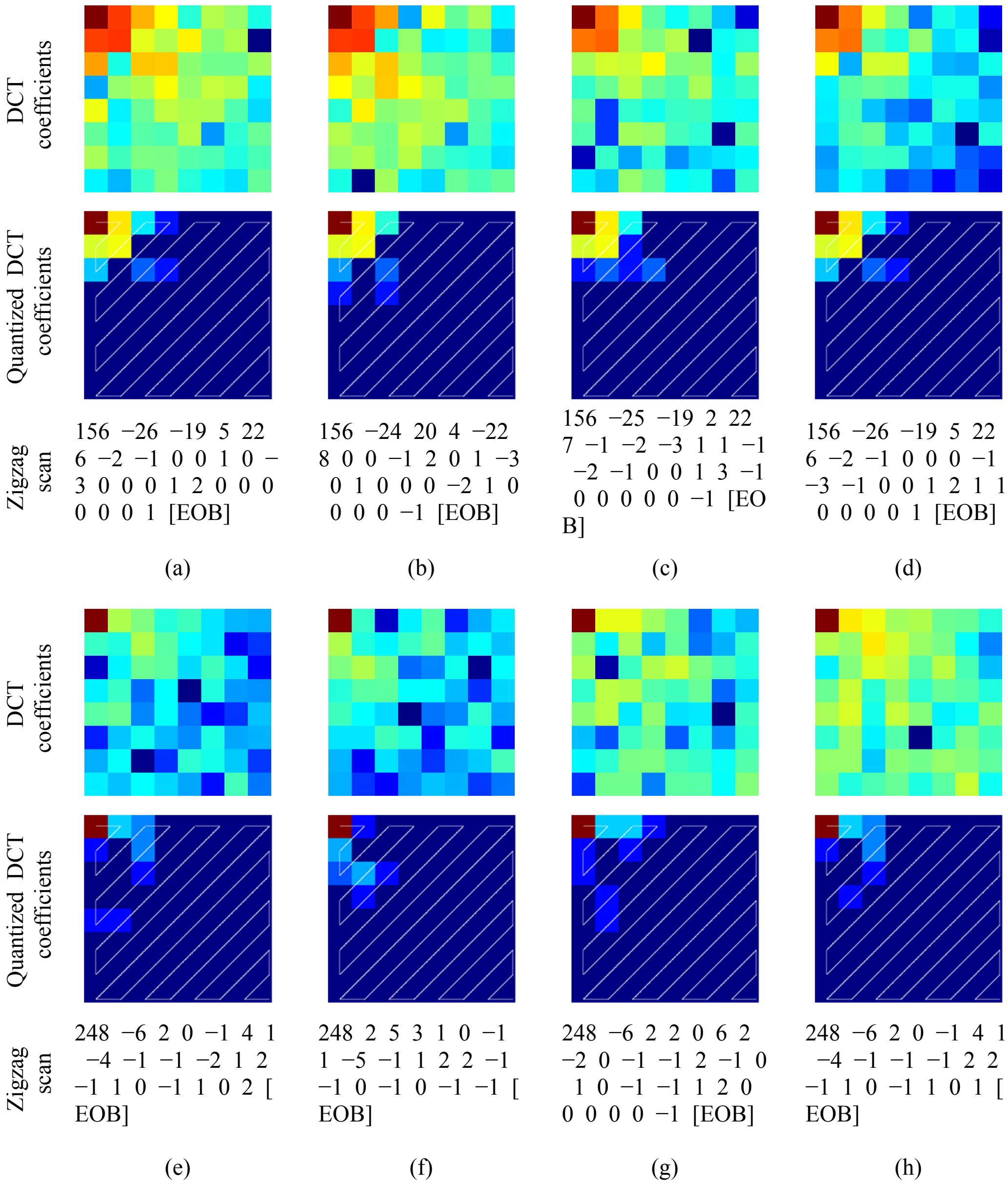


Figure 19. The proposed method compressibility analysis based on the DCT energy compaction. (a–d) The DCT of the block where correlation coefficients were $\sigma_h = 0.95$ and $\sigma_v = 0.96$. (e–h) The DCT of the block where correlation coefficients were$\sigma_h = 0.49$ and $\sigma_v = 0.52$. (a,e) The original block transformations. (b,f) The scrambled block transformations obtained by processing the entire blocks. (c,g) The scrambled block transformations obtained by the sub-block (4 × 4) processing. (d,h) The scrambled block transformations obtained by the sub-block (2 × 2) processing.

4.5.1.2. Compression–Efficiency Analysis

For compression analysis, we plotted the rate distortion (RD) curve as shown in Figure 20 and 21 for color images compression without chroma subsampling, and with 4:2:0 subsampling ratio, respectively. The y–axis is the recovered image quality represented as a multiscale structural similarity index measure (MS–SSIM) in dB against the compression savings given as the bitrate on the x–axis. The RD curves are for the JPEG quality factors of 70–100. To compare the RD curves, we use Bjøntegaard delta (BD) measures [76], where the BD rate difference is the percent difference between two bitrates of the equivalent quality, and the BD quality difference is the average dB difference for the equivalent bandwidth. Instead of peak–signal–to–noise ratio (PSNR), used in the conventional literature, the BD rate difference is computed for a better image quality measure such as MS–SSIM [36]. Note that the value of MS–SSIM (M) is $-10\log_{10}(1-\mathrm{M})$.

Table 8 summarizes the rate savings and image quality differences of PE methods and JPEG–compressed images using BD measures for the RD curves presented in Figure 20 and 21. First, we considered the JPEG algorithm for color images compression without chroma subsampling. The results show that the proposed IIB–CPE (8 × 8) preserves the compression savings of color–EtC (16 × 16) while outperforming GS–EtC by almost 15%. However, there is a 3% datarate difference as compared to compression of the original images. The main reason is that the DC coefficients in adjacent blocks have higher correlation, and the JPEG algorithm treats them differently than the AC coefficients. The DC coefficients in adjacent blocks are differentially pulse code modulated (DPCM) with each other and their prediction errors are entropy encoded. In the proposed scheme, the permutation step disrupts this correlation, and compression savings that could have been achieved by the DPCM are lost. On the other hand, for smaller sub–block sizes such as, (4 × 4) and (2 × 2), there is an increment in the image size across all methods. However, compared to conventional PE methods, where the file size drastically increases, the proposed method is still able to deliver compression savings. For example, to achieve the highest quality image (that is, $\mathrm{QF} = 100$) when using the smallest block size of (2 × 2), the bitrate of IIB–CPE is 4.6 bpp whereas for conventional

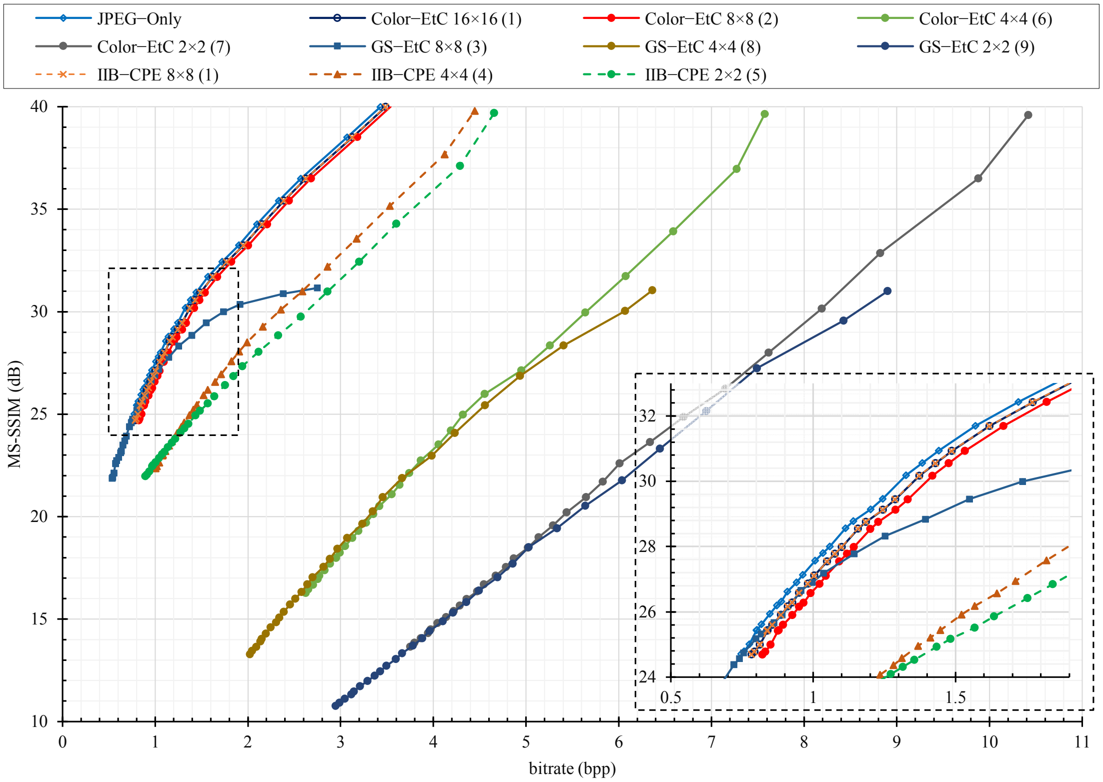


Figure 20. The JPEG compression performance without chroma subsampling on different perceptual encryption methods in terms of rate distortion (RD) curves with respect to MS–SSIM (dB) on Tecnick color dataset. The number enclosed in parentheses at the end of each series name shows its performance rank. The overlapping regions in the graph are zoomed in and shown in the bottom right corner.

Color–EtC and GS–EtC methods the bitrate is 10.4 and 8.9 bpp, respectively. The bitrate of conventional PE methods is in fact an increment from the original image size.

Second, we considered the JPEG algorithm for color images compression with subsampling ratio of 4:2:0. The JPEG encoder performs chroma sub-sampling (such as 4:2:0) on a block size of (16 × 16). Therefore, when the images are encrypted with a block size smaller than the standard specified size, then the resulting (8 × 8) blocks in color components will consist of pixels from different blocks. Such pixels have low correlation and performing interpolation on them to recover the original image resolution results in block artifacts (Type I).

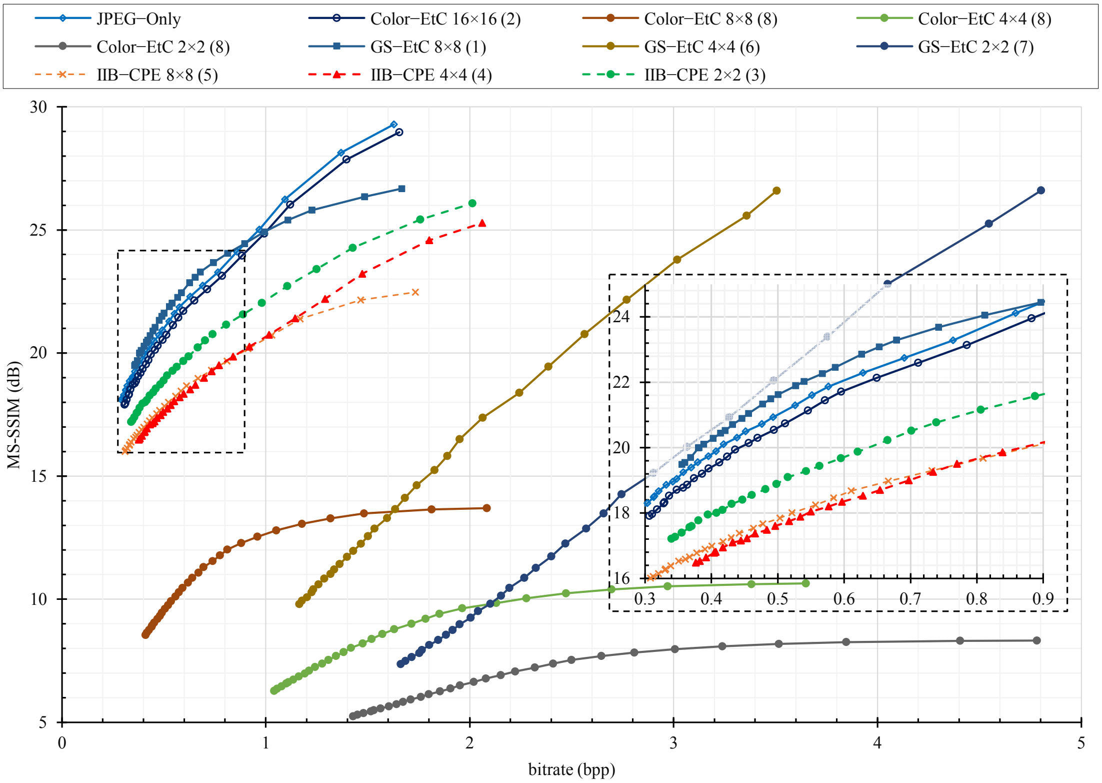


Figure 21. The JPEG compression performance with chroma subsampling ratio (4:2:0) on different perceptual encryption methods in terms of rate distortion (RD) curves with respect to MS–SSIM (dB) on Tecnick color dataset. The number enclosed in parentheses at the end of each series name shows its performance rank. The overlapping regions in the graph are zoomed in and shown in the bottom right corner.

In addition, when the luminance component (Y) is shuffled with either of the chrominance components (Cb or Cr) as a result of the channel shuffling step, then during encoding this block is treated as a chrominance component. Based on human color perception capabilities, the JPEG algorithm processes the chrominance components differently than the luminance. For example, the chrominance component resolution is reduced during the subsampling step and the values are represented with fewer bits during the quantization step. Therefore, when a block that belongs to the luminance component goes under these processing then regardless of its size, the block is recovered with blur distortion (Type II). For visual inspection, Figure 22 shows images recovered from PE encryption methods with different block sizes. For the conventional Color-EtC method, when (16 × 16) block size is used then the Type II distortion appears in the regions where there is an abrupt change in the pixel's values. Images encrypted with block sizes smaller than 16×16 have both Type I and Type II distortions. Since, in the GS-EtC method, the chroma subsampling is carried out before encryption; therefore, no block artifacts are visible. However, for smaller block sizes there is visible distortion as a result of the quantization step. On the other hand, for the proposed IIB-CPE method, only Type II distortion appears in the similar regions as that of the Color-EtC (16 × 16) method. The proposed method preserved the correlation among the adjacent pixels within a block as discussed in Section 4.4.3; therefore, avoids Type I distortion. The Type II distortion is reduced when using a larger value for the JPEG quality factor as shown in Figure 23.

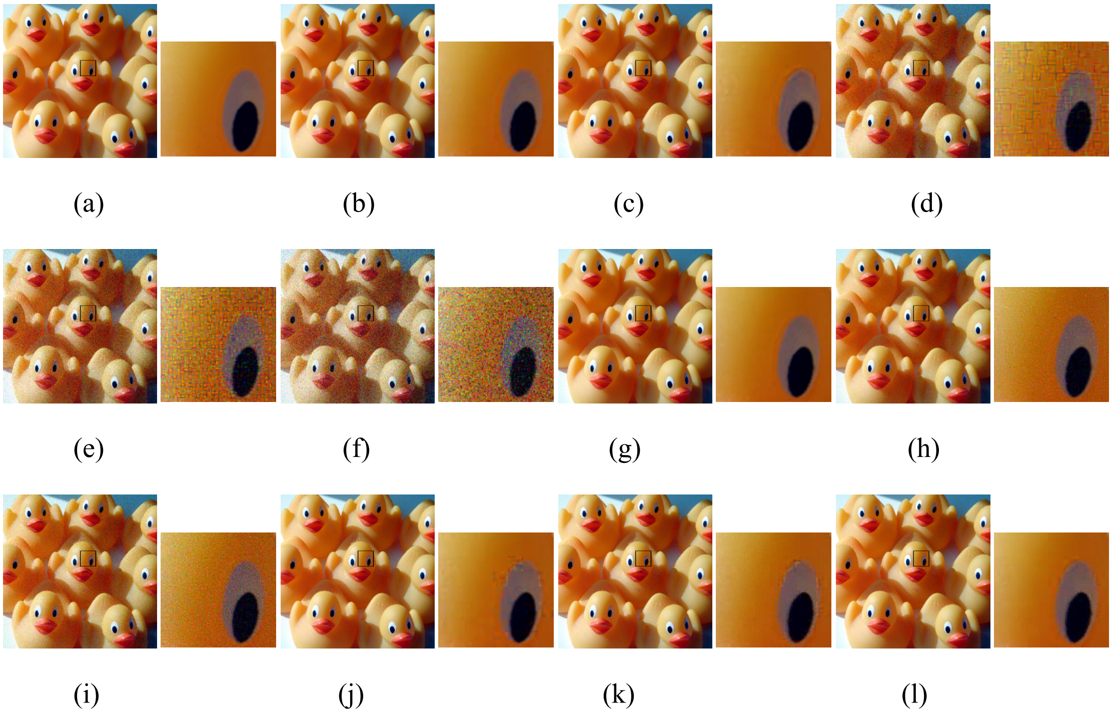


Figure 22. Images recovered from PE methods using different block sizes. The JPEG quality factor is 71. (a) is the original image. (b) is recovered from the plain image. (c)–(f) are images recovered from the Color-EtC method with block sizes (16 × 16, 8 × 8, 4 × 4, and 2 × 2), respectively. (g)–(i) are images recovered from the GS-EtC method with block sizes (8 × 8, 4 × 4, and 2 × 2), respectively. (j)–(l) are images recovered from the proposed with block sizes (8 × 8, 4 × 4, and 2 × 2), respectively. The boxed region in each image is zoomed and shown on its right side.

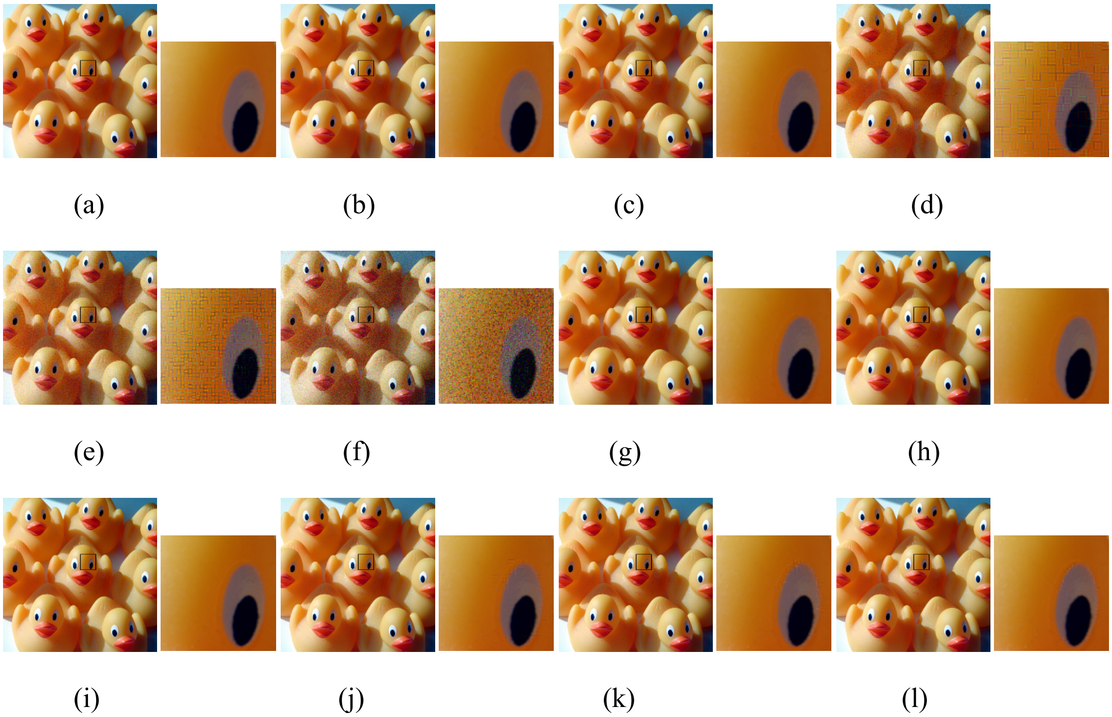


Figure 23. Images recovered from PE methods using different block sizes. The JPEG quality factor is 100. (**a**) is the original image. (**b**) is recovered from the plain image. (**c**)–(**f**) are images recovered from the Color-EtC method with block sizes (16 × 16, 8 × 8, 4 × 4, and 2 × 2), respectively. (**g**)–(**i**) are images recovered from the GS-EtC method with block sizes (8 × 8, 4 × 4, and 2 × 2), respectively. (**j**)–(**l**) are images recovered from the proposed with block sizes (8 × 8, 4 × 4, and 2 × 2), respectively. The boxed region in each image is zoomed and shown on its left side.

For the compression savings when using chroma subsampling, it can be seen in Table 8 that GS-EtC with block size (8 × 8) has achieved better performance among the compared methods. For block size (8 × 8), the method has an improved bitrate than the plain images. The reason for this is that GS-EtC uses a single quantization table (for example, the standard luminance table in our simulations); therefore, in the low bitrate region, it has achieved better image quality as shown in Figure 21. The Color-EtC with block size (16 × 16) has a 6% difference in the bitrate with negligible difference in the image quality. However, for both Color-EtC and GS-EtC the performance gain is reduced significantly with the smaller block size. On the other hand, the proposed method has higher bitrate requirement than the conventional methods. However, it is still able to deliver compression savings across different blocks sizes with a negligible difference in the image quality. For the proposed method, there is an opposite trend between the measures and the block sizes. Because when using sub-blocks of smaller sizes for intra block processing, then the correlation is better preserved within the block.

Table 8. The JPEG compression performance on different perceptual encryption methods in terms of Bjøndegaard Delta measures. The rate differences are for the equivalent quality relative to the JPEG under MS–SSIM for RD curves plotted in Figures 6 and 7.

		Block-Size	BD-Rate	BD-MS-SSIM
4:4:4	Color–EtC	16 × 16	3.18	−0.32
		8 × 8	6.18	−0.61
		4 × 4	267.72	−19.88
		2 × 2	424.69	−27.8
	GS–EtC	8 × 8	17.76	−2.16
		4 × 4	405.32	−20.04
		2 × 2	646.57	−27.39
	IIB–CPE	8 × 8	3.11	−0.31
		4 × 4	64.04	−5.4
		2 × 2	78.61	−6.01
4:2:0	Color–EtC	16 × 16	5.98	−0.39
		8 × 8	nan	−12.55
		4 × 4	nan	−19.94
		2 × 2	nan	−23.39
	GS–EtC	8 × 8	−1.41	−0.21
		4 × 4	352.1	−16.33
		2 × 2	549.1	−22.26
	IIB–CPE	8 × 8	112.44	−4.1
		4 × 4	105.81	−4.19
		2 × 2	60.52	−2.76

4.5.2. Encryption Analysis

4.5.2.1. Jigsaw Puzzle Solver Attack Analysis

Several statistical tests, for example, histogram analysis, correlation analysis, and entropy analysis usually assess the security of encryption algorithms. These tests are commonly used for analysis of full encryption techniques, where an attacker tries to figure out details of the algorithm. However, different from full encryption algorithms, the goal of perceptual encryption cryptanalysis is to recover a better-quality image out of the unencrypted data and its semantics [77]. Compressible PE preserves correlation of the original images on a bock level, which may vulnerate to the jigsaw puzzle solver attack proposed in [36]. It is a ciphertext–only attack (COA) where the main goal is to reconstruct the original image fully or partially from the cipher image by exploiting the correlation that exists in each block. To demonstrate robustness against the attack, we extended the jigsaw puzzle attack to accommodate sub–block processing. The proposed jigsaw puzzle solver reconstructs the original image in two steps. First, correct orientations of sub–blocks in an entire block are recovered. For this purpose, the cipher image can be treated as a type of jigsaw puzzle where only orientation of the pieces is unknown. Here, the change in orientation is a result of the combination of rotation and inversion transformations, which is given by a composite function as

$$f_{R,I}(p_i) = f_R \circ f_I(p_i), \tag{51}$$

The function f_R rotates a piece where $R \in \{0°, 90°, 180°, 270°\}$, and the function f_I flips a piece, where $I \in \{\mathrm{H}, \mathrm{V}, \mathrm{HV}, \mathrm{No\ \ Flip}\}$ and H: horizontal flip, V: vertical flip and HV: horizontal flip followed by vertical flip. For the composite function, the rotation function $R \in \{0°, 90°\}$ or $R \in \{180°, 270°\}$ in order to avoid collision. The orientation can be recovered by minimizing the total sum of the cost across the boundaries of any two given pieces. The cost is a pairwise compatibility between two pieces calculated as Mahalanobis Gradient Compatibility (MGC) measure proposed in [25]. For example, for two pieces p_1 and p_2, the compatibility between top of p_1 and bottom of p_2 is represented as $C_{TB}(p_1, p_2)$. The minimum compatibility of the two pieces for the proposed solver is given as

$$C_{TB}(p_1, p_2) = \min_{f_{R,I}} \left\{ C_{TB}\left(p_1, f_{R,I}(p_2)\right) \right\}. \tag{52}$$

Since the position of the sub–blocks are not changed; therefore, each sub–block is compared only with its neighbor in the block. Once the sub–blocks orientation is recovered, the next step is to solve the puzzle for the transformation performed on the entire blocks. Note that the recovery of sub–blocks orientation with respect to their neighbors in a block does not necessarily guarantee the correct orientation of the entire block. Therefore, the entire block has the transformation of f_{RI} and has an additional function f_{NP} that applies negative–positive transformation on a block, where $NP \in \{0, 1\}$ and 1 being transformation is applied. The overall transformations on a block P_i can be defined by a composite function as

$$f_{R,I,NP}(P_i) = f_R \circ f_I \circ f_{NP}(P_i), \tag{53}$$

and the minimum compatibility in (52) for two blocks P_1 and P_2 becomes:

$$C_{TB}(P_1, P_2) = \min_{f_{R,I,NP}} \left\{ C_{TB}\left(P_1, f_{R,I,NP}(P_2)\right) \right\}. \tag{54}$$

The position of each block is changed because of the permutation step in the encryption; therefore, compatibility of each block should be computed with every other block in the puzzle. Finally, the calculated compatibility scores are then used to solve the puzzle by using a constrained minimal spanning tree algorithm proposed in [25]. The $\mathbf{D_c}$ in (24), $\mathbf{N_c}$ in (25), and $\mathbf{L_c}$ in (26) measures proposed in [24], [25] were investigated in this study to show robustness against the jigsaw puzzle solver attacks. The scores, $\mathbf{D_c}, \mathbf{N_c}, \mathbf{L_c} \in [0,1]$, where a larger value indicates a better reconstruction of the cipher image. For the robustness analysis, 20 images were randomly chosen from the Tecnick dataset. First, we encrypted the images and then assembled them using the jigsaw puzzle solver. Table 9 summarizes the average $\mathbf{D_c}, \mathbf{N_c}$, and $\mathbf{L_c}$ scores of those 20 images. The smaller score value of the proposed method is attributed to intra block processing, which reduces the efficiency of the compatibility measure used by the jigsaw puzzle solver.

4.5.2.2. Correlation Analysis

An encryption algorithm should eliminate correlation among adjacent pixels in an image for better security. In general, the correlation coefficient $\rho(\boldsymbol{x}, \boldsymbol{y})$ between two distributions $\boldsymbol{x}$ and $\boldsymbol{y}$ each with N elements can be calculated as in (20). The coefficient $\rho \in \{-1.0, 1.0\}$, where $\rho = 0$ shows that there is no correlation, $\rho < 0$ shows negative correlation and $\rho >$

Table 9. Robustness of the perceptual encryption methods against jigsaw puzzle solver attacks. (Nc: neighbor comparison, Lc: largest component comparison, Dc: direct comparison).

Methods	Nc	Lc	Dc
Color–EtC 16 × 16	0.11	0.12	0.01
GS–EtC 8 × 8	0.001	0.002	0.001
IIB–CPE 8 × 8	0.08	0.02	0.01
IIB–CPE 4 × 4	0.05	0.02	0.01
IIB–CPE 2 × 2	0.06	0.02	0.01

Table 10. The encryption analysis of the CPE schemes under different statistical tests.

	Correlation Coefficient							
Methods	Image level			Block level			Entropy	Histogram variance
	diagonal	horizontal	vertical	diagonal	horizontal	vertical		
Plain	0.87	0.91	0.9	0.42	0.55	0.51	6.51	237.92
Color–CPE	0.84	0.91	0.91	0.01	0	0	7.42	40.58
PGS–CPE	0.73	0.85	0.85	–0.01	0	0	6.83	112.01
Extended–CPE [47]	0.84	0.91	0.91	0	0	0	7.42	40.59
Extended CPE [63]	0.84	0.91	0.91	0	0	0	7.42	40.59
IIB–CPE (8 × 8)	0.83	0.9	0.9	0	0.01	0	7.42	40.6
IIB–CPE (4 × 4)	0.82	0.89	0.89	0	0	0	7.42	40.6
IIB–CPE (2 × 2)	0.83	0.9	0.89	0.01	0.01	0	7.42	40.57

0 shows positive correlation. The negative correlation means that when one value is increasing the other is decreasing and the positive correlation means that both values are either increasing or decreasing. For the correlation analysis, we have performed two experiments. First, we have shown the correlation between adjacent pixels randomly chosen from the whole image. Since the encryption algorithms are block based, the correlation among the neighboring pixels was still high in the cipher images as shown in Table 10. In order to preserve the JPEG compression performance efficiency on the cipher images, the correlation in the block of at least (8 × 8) size should not be altered. At first, it may seems like the CPE algorithms are vulnerable, also mentioned in [5]; therefore, in the second experiment, we have analyzed correlation among adjacent blocks that is by taking the pixels on the borders only. It can be seen that on a block level the cipher image had low correlation and exhibits favorable encryption properties. Table 10 presents the correlation analysis for the entire dataset in diagonal, horizontal and vertical directions for plain images and CPE cipher images.

4.5.2.3. Histogram Analysis

The histogram of an image gives the intensity distribution as the number of pixels at each intensity level. For a plain image, the histogram is a skewed distribution concentrated at one location and a cipher image has a uniform distribution. To quantify the characteristics of a histogram $\boldsymbol{R}$, histogram variance $V(\boldsymbol{R})$ is calculated as in (21). Small value of $V(\boldsymbol{R})$ means a uniform distribution. Table 10 shows the mean $V(\boldsymbol{R})$ values across the whole dataset for plain and cipher images. In all cases, $V(\boldsymbol{R})$ values of cipher images are smaller than that of the plain images; therefore, reduces the information characteristics of the image. The PGS–CPE has the greater $V(\boldsymbol{R})$ value among the evaluated methods.

4.5.2.4. Information Entropy Analysis

The information entropy shows the degree of randomness in an image. The entropy of an image $H(\boldsymbol{I})$ is calculated as in (22). For a truly random image with $N = 256$ intensity levels, the ideal value of the entropy should be closer to $H(\boldsymbol{I}) = log_2(N) = 8$. Table 10 shows the mean of entropy values across the whole dataset for plain and cipher images. The entropy values are smaller than the ideal value of $H(\boldsymbol{I}) = 8$, because the PE methods preserve the image contents on a block level. Nonetheless, $H(\boldsymbol{I})$ values of cipher images were greater than that of the plain images; therefore, resulted in better randomness. In addition, PGS–CPE methods have the smaller $H(\boldsymbol{I})$ value among the evaluated CPE methods.

4.5.2.5. Differential Attack Analysis

In order to be resistant against differential attack, an encryption algorithm should have the ability to generate two different cipher images for plain images with a minor difference. The degree of change can be quantified by two metrics, namely, the number of pixels change rate (NPCR) and the unified average changing intensity (UACI). The NPCR gives the percentage difference between two cipher images and UACI gives the average intensity of differences between the two images. For this purpose, a plain image $\boldsymbol{I}_1$ of size M is slightly modified by randomly changing one of its pixel values to generate another image $\boldsymbol{I}_2$. The two plain-images $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ are encrypted using the same encryption key to obtain the cipher images $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, respectively. The NPCR and UACI parameters are calculated for the cipher images $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ as in (23). For $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ to have the ideal value of NPCR and UACI, the minor change in the plain images should be reflected across the whole cipher images. Usually, the diffusion process, which makes the current ciphertext dependent on the previous ones, achieves this property. However, in the CPE schemes there is no such operation. In fact, the only step that changes pixel values is the negative–positive transformation function where 50% of the blocks or pixels are randomly XORed with 255. As a result, the CPE schemes may be vulnerable to differential attacks. Nonetheless, the use of different keys for each image as in the literature provides a certain level of resistance against the attack.

4.5.2.6. Robustness Analysis

In this section, we analyze the robustness of CPE schemes against the data loss attack and

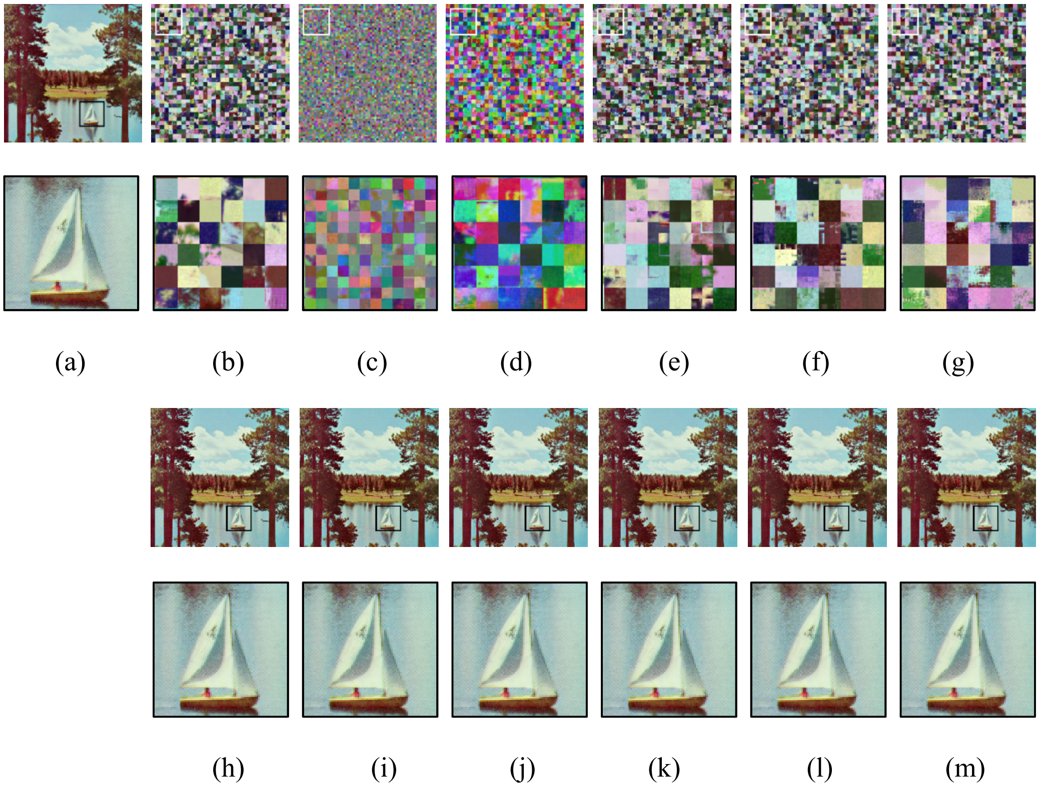


Figure 24. Visual analysis of the images recovered from the CPE processing.(a) is the original image. (b – g) are cipher images obtained from the Color–CPE, PGS–CPE, Extended–CPE, IIB–CPE (8 × 8), IIB–CPE (4 × 4) and IIB–CPE (2 × 2) methods, respectively. Their corresponding recovered images are shown in (h – m). In each image, the boxed region is zoomed and shown below them. Note that the JPEG compression was not performed on the cipher images.

noise attack. Figure 24 (a) shows the original image and its cipher images in Figure 24 (b) – (g) were obtained from the Color–CPE, Extended–CPE, PGS–CPE and IIB–CPE schemes. For the data loss attack analysis, we have cropped different regions (i.e., setting the pixels values equal to zero) from the cipher image as shown in Figure 25 (a) – (f) for the cipher images in Figure 24 (b) – (g). Their corresponding recovered images are shown in Figure 25 (g) – (l). It can be seen that the images have recovered successfully without the corrupted blocks. In case of the Color–CPE (Figure 25 (h)) and IIB–CPE (Figure 25 (j–l)) images, the lost blocks do not have any color because in each channel blocks from the same locations have been lost and the white blocks are the result of the negative–positive transformation step. On the other hand, for the Extended–CPE (Figure 25 (i)) and PGS–CPE (Fig 24 (h)) images, the lost blocks are not from the same locations in the color–channels; therefore, the missing blocks have color and certain spatial information appears in them.

Similarly, for the noise attack analysis, the cipher images (Figure 24 (b–g)) were added with Gaussian noise (Figure 26 (a–f)) and salt–pepper noise (Figure 27 (a–f)). Their corresponding recovered images are shown in Figure 26 (g–l) and Figure 27 (g–l), respectively. In the case of Gaussian noise, the recovered images are blurred as compared to the original image across all CPE methods. For the salt–pepper noise, the noisy pixels of the cipher images were inherited in the recovered image without affecting the rest of the image. For quantitative analysis, Table 11 summarizes the average MS–SSIM of the recovered images across the whole dataset. Overall, the methods that represent input as a color image have better resilience against data loss and noise. The CPE methods are robust against the noise and data loss attacks owing to the lack of diffusion process.

Table 11. Quality of the recovered images under different types of loss attacks (Figure 25 – 27).

Methods	Data Loss	Gaussian Noise	Salt & Pepper Noise
Color–CPE	0.54	0.95	0.91
PGS–CPE	0.24	0.88	0.86
Extended–CPE	0.54	0.95	0.91
IIB–CPE (8 × 8)	0.55	0.95	0.91
IIB–CPE (4 × 4)	0.54	0.95	0.91
IIB–CPE (2 × 2)	0.54	0.95	0.91

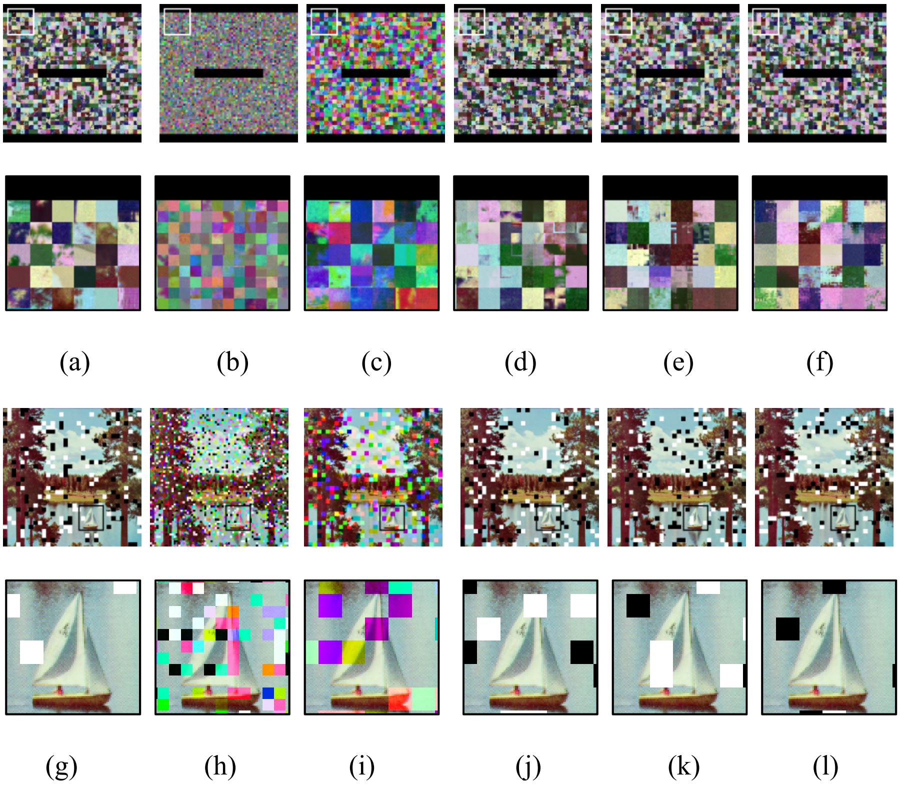


Figure 25. The CPE methods robustness against the data loss attack. (a – f) are the cipher images given in Figure 20 (b – g) with the data loss attack. Their corresponding recovered images are shown in (g – l). For better visual inspection, the boxed region in each image is enlarged and shown below them.

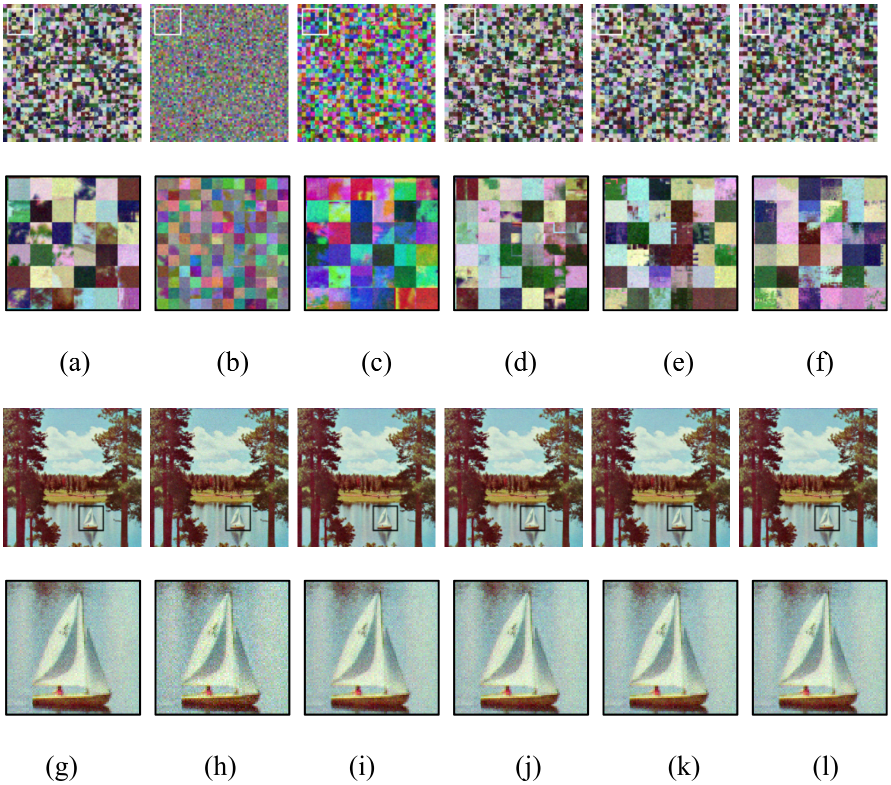


Figure 26. The CPE methods robustness against the noise attack. (a – f) are the cipher images given in Figure 24 (b – g) with the noise attack by adding Gaussian noise. The recovered images for (a – f) are shown in (g – l). For better visual inspection, the boxed region in each image is enlarged and shown below them.

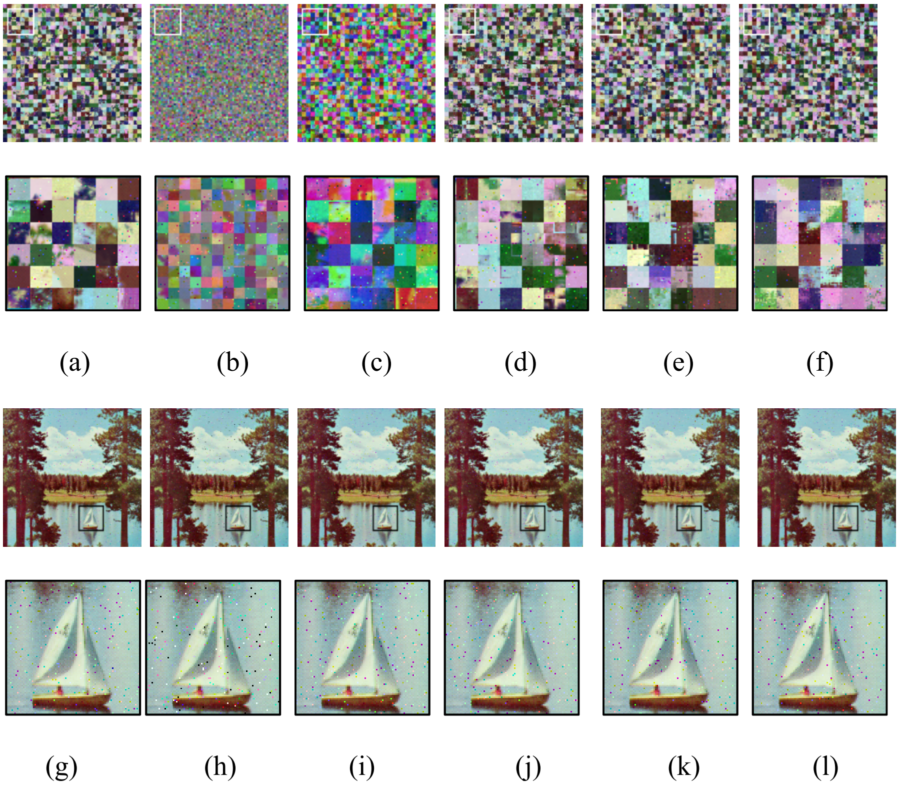


Figure 27. The CPE methods robustness against the noise attack. (a – f) are the cipher images given in Figure 24 (b – g) with noise attack by adding Salt and Pepper noise. The recovered images for (a – f) are shown in (g – l). For better visual inspection, the boxed region in each image is enlarged and shown below them.

4.5.27. Keyspace Analysis

In general, the encryption algorithm of the CPE consisted of four secret symmetric keys: $\boldsymbol{\mathcal{K}}_1$ permutation key, $\boldsymbol{\mathcal{K}}_2$ rotation and inversion key, $\boldsymbol{\mathcal{K}}_3$ negative–positive transformation key, and $\boldsymbol{K}_4$ color–channel shuffling key. Each key $\boldsymbol{\mathcal{K}}_i, i = \{1,2,3\}$ is a set of three keys, one for each component of the image and is denoted as $\boldsymbol{\mathcal{K}}_i = \{\boldsymbol{K}_i^R, \boldsymbol{K}_i^G, \boldsymbol{K}_i^B\}$. The keyspace $\boldsymbol{\mathcal{K}}$ of a CPE algorithm is the set of all keys used in the encryption steps as $\boldsymbol{\mathcal{K}} = \{\boldsymbol{\mathcal{K}}_1, \boldsymbol{\mathcal{K}}_2, \boldsymbol{\mathcal{K}}_3, \boldsymbol{K}_4\}$ and the key size is given by the set cardinality as $|\boldsymbol{\mathcal{K}}|$.

As discussed earlier, in the CPE methods an input color image $\boldsymbol{I}_{W\times H\times C}$, with $W \times H$ pixels in C color channels, is grouped into non–overlapping square blocks with N^2 pixels. The number of blocks B_c in a color channel c is given by

$$B_c = L \times M, \tag{55}$$

and the number of blocks B in the image is given by

$$B = 3 \times B_c. \tag{56}$$

When a block $\boldsymbol{B}$ of size $N \times N$ pixels is divided into $SL \times SL$ smaller blocks of size SN^2 for the sub block processing, the number of sub blocks SB_c in a color channel c is

$$SB_c = (SL \times SL) \times B_c, \tag{57}$$

and the number of sub blocks SB in the image is given by

$$SB = 3 \times SB_c. \tag{58}$$

The keyspace $\boldsymbol{\mathcal{K}}_{CC}$ for the Color–CPE scheme based on (55) can be derived as:

$$\begin{gathered} \boldsymbol{\mathcal{K}}_{CC} = \{\boldsymbol{\mathcal{K}}_{1,CC}, \boldsymbol{\mathcal{K}}_{2,CC}, \boldsymbol{\mathcal{K}}_{3,CC}, \boldsymbol{K}_{4,CC}\}, \\ |\boldsymbol{\mathcal{K}}_{CC}| = 3(B_c!) \cdot 3(8^{B_c}) \cdot 3(2^{B_c}) \cdot 6^{B_c}. \end{gathered} \tag{59}$$

Since, the Color–CPE scheme used the same key for each color component, its keyspace size becomes

$$|\boldsymbol{\mathcal{K}}_{CC}| = B_c! \cdot 8^{B_c} \cdot 2^{B_c} \cdot 6^{B_c}. \tag{60}$$

The keyspace $\boldsymbol{\mathcal{K}}_{EC}$ for Extended–CPE schemes based on (55) can be derived as:

$$\begin{gathered}\boldsymbol{\mathcal{K}}_{EC} = \{\boldsymbol{\mathcal{K}}_{1,EC}, \boldsymbol{\mathcal{K}}_{2,EC}, \boldsymbol{\mathcal{K}}_{3,EC}, \boldsymbol{K}_{4,EC}\}, \\ |\boldsymbol{\mathcal{K}}_{EC}| = 3(B_c!) \cdot 3(8^{B_c}) \cdot 3(2^{B_c}) \cdot 6^{B_c}.\end{gathered} \tag{61}$$

Here, the keyspace for the first three steps increased by a factor of three as compared to $|\boldsymbol{\mathcal{K}}_{CC}|$ in (59). The reason is that the Extended–CPE schemes perform the encryption steps independently in each color component. In addition, the color channel shuffling step scrambles the blocks in the three-color components; therefore, (61) can be simplified as

$$\begin{aligned}|\boldsymbol{\mathcal{K}}_{EC}| &= (3B_c)! \cdot 8^{3B_c} \cdot 2^{3B_c} \\ &= B! \cdot 8^{B} \cdot 2^{B}.\end{aligned} \tag{62}$$

The keyspace $\boldsymbol{\mathcal{K}}_{IC}$ for the IIB–CPE can be derived as:

$$\begin{gathered}\boldsymbol{\mathcal{K}}_{IC} = \{\boldsymbol{\mathcal{K}}_{1,IC}, \boldsymbol{\mathcal{K}}_{2,IC}, \boldsymbol{\mathcal{K}}_{3,IC}, \boldsymbol{K}_{4,IC}\}, \\ |\boldsymbol{\mathcal{K}}_{IC}| = 3(B_c!) \cdot \big(3(8^{SB_c}) \cdot 3(8^{B_c})\big) \cdot 3(2^{B_c}) \cdot 6^{B_c}.\end{gathered} \tag{63}$$

Similar to the Color–CPE scheme, IIB–CPE used the same key for each color component, its keyspace becomes

$$|\boldsymbol{\mathcal{K}}_{IC}| = B_c! \cdot (8^{SB_c} \cdot 8^{B_c}) \cdot 2^{B_c} \cdot 6^{B_c}. \tag{64}$$

Compared to $|\boldsymbol{\mathcal{K}}_{CC}|$ in (55), $|\boldsymbol{\mathcal{K}}_{IC}|$ is increased by a factor of 8^{SB_c} because of the sub–block processing. This increment depends on the sub–block size – specifically, when the number of pixels in a sub–block is $SN^2 \in \{8^2, 4^2, 2^2\}$ then the keyspace size is increased by a factor of $8^{SB_c} \in \{8^{4B_c}, 8^{16B_c}, 8^{64B_c}\}$, respectively. The keyspace $\boldsymbol{\mathcal{K}}_{PC}$ for the PGS–CPE scheme can be derived without the last term $\boldsymbol{\mathcal{K}}_4$ as the methods lack color–channel shuffling step:

$$\begin{gathered}\boldsymbol{\mathcal{K}}_{PC} = \{\boldsymbol{K}_{1,PC}, \boldsymbol{K}_{2,PC}, \boldsymbol{K}_{3,PC}\}, \\ |\boldsymbol{\mathcal{K}}_{PC}| = B! \cdot 8^{B} \cdot 2^{B}.\end{gathered} \tag{65}$$

The number of blocks has increased by a factor of three compared to the Color–CPE schemes. Similar to Extended–CPE methods, the PGS–CPE schemes process each image block independently as the color channels are concatenated in a single component. In addition, on the contrast of the color–based CPE methods where the smallest block size used is (16 × 16), the PGS–CPE schemes can benefit from the smallest allowable block size in the JPEG standard that is (8 × 8), the number of blocks is increased four times and (61) can be modified as

$$|\boldsymbol{\mathcal{K}}_{PC}| = (4B)! \cdot 8^{(4B)} \cdot 2^{(4B)}. \tag{66}$$

Overall, based on (60)(62)(64) and (66) the relation between the keyspace sizes of the CPE methods for color image encryption can be established as

$$|\boldsymbol{\mathcal{K}}_{PC}| \gg |\boldsymbol{\mathcal{K}}_{EC}| \gg |\boldsymbol{\mathcal{K}}_{IC}| > |\boldsymbol{\mathcal{K}}_{CC}|. \tag{67}$$

For the completeness of security analysis, it is necessary to show robustness of the proposed method against known–plaintext (KPA) and chosen–plaintext attack (CPA). Like the conventional EtC methods, the proposed scheme is robust against KPA because of the use of different keys for encryption of each image. Since the proposed method is symmetric encryption scheme and does not need to disclose any information about the key; therefore, it can resist CPA.

4.6. Chapter Summary

We have proposed a compressible perceptual encryption algorithm based on inter and intra block processing, which improved the encryption efficiency of existing methods without compromising the compression savings. The proposed scheme allows smaller block size in rotation and inversion steps, thereby improving robustness against different attacks, which was confirmed by various statistical tests. The proposed method provided security and bandwidth efficiency during transmission and storage. The JPEG format–compatibility of the encoded images makes them suitable for a wide range of applications, such as privacy–preserving cloud–based photo storage, privacy–preserving content–based image retrieval and social networking services.

V. THE CASE OF PRIVACY IN CLOUD-BASED MEDICAL IMAGE ANALYSIS

5.1. Introduction

Cloud services provide a cost-effective solution to meet the Information and Communication Technology (ICT) needs of an organization. The organization can use ICT resources, services, and software of a Cloud Services Provider (CSP) via the internet without a necessity of internal infrastructure or hardware on-site installations. With the recent success of Machine Learning (ML) in computer vision, automatic computer aided diagnosis (CAD) systems have emerged in healthcare organizations to assist doctors and practitioners. Particularly, Deep Learning (DL), a subfield of ML, has achieved state-of-the-art performance for image classification [78]. However, DL models are compute-intensive tasks, and their training requires cutting-edge technology and high computational resources. In this regard, healthcare organizations can avail cloud-computing services to access the latest technology to speed up the training process and allow DL models to scale efficiently with a lower capital cost [10], [79]. In addition, training DL models requires a large volume of sample data, which in some cases such as the medical domain, is expensive and difficult to acquire. To overcome this issue, healthcare organizations can benefit from a community cloud, where services are shared by organizations with common interests. In this case, cloud storage services can be used as a shared central data repository for joint projects and collaboration among the organizations. However, like all communication systems, when data is outsourced for cloud services, there is a risk of information leakage and a large volume of data requires high bandwidth [1], [80]–[82].

Compression and encryption are two processes that satisfy the dual requirements of data transmission over bandwidth constraint and public channels. Image compression gives a compact representation to an image such that it requires a smaller number of bits. It can be achieved either in lossless or lossy mode. In lossless compression, an image can be recovered with almost the same quality as that of the original image, whereas in lossy mode the image quality degrades. Compared to lossless mode, lossy compression offers better savings; however, resulting quality degradation in lossy mode may not be acceptable in certain domains. For example, medical images contain information crucial for correct diagnosis of diseases; therefore, their compression should be carried out in such a way that the diagnostic information remains intact in them while their sizes are reduced [27], [28], [83]. One of the popular approaches to achieve this goal is to compress the region-of-interest (ROI) necessary for diagnosis in lossless mode and non-ROI in lossy mode [28], [28], [83]. Such methods can achieve a significant reduction in the image size while preserving its important details. However, they require segmentation of an image beforehand, which is computationally expensive and is a target task to be performed using cloud-computing resources. Therefore, ROI-based methods are not suitable for efficient image data transmission [10].

Encryption makes image data unintelligible, which can only be recovered by its inverse decryption process. The number theory and chaos theory-based encryption algorithms are proven efficient for securing image data [10]. These conventional encryption algorithms perform stream encryption and/or scrambling of pixel values; however, they are only suitable for encrypting raw images. For example, the JPEG compressed image consists of format markers and any changes in them by an external operation will leave the image uninterpretable. Similarly, re-encoding a cipher image as a JPEG image results in file size increment. Different from other forms of data, encryption of image data can be carried out only by disrupting their intrinsic properties. For example, changing pixel correlation and/or redundancy in an image can result in an unintelligible image with a necessary level of security. Based on this observation, a new class of encryption algorithms has emerged called Perceptual Encryption (PE) algorithms to meet the aforementioned requirements of encrypting compressed images. The main idea is to reverse the conventional order of performing compression prior to encryption. PE performs block-based operations that hides only perceptual information of an image, thereby preserves image intrinsic properties necessary to make the cipher images JPEG compressible, which makes them suitable for numerous applications, such as cloud photo storage and social networking services [37], [38] and image retrieval in the encryption domain [64].

Nonetheless, PE methods are resilient against various attacks, including brute-force and cipher-text-only attacks [36].

Based on an input image representation, PE methods can be grouped as Color-PE and Grayscale-PE methods. The Color-PE represents an input color image as a three-component image and uses same encryption keys for each component [65], whereas their extended versions encrypt each color component independently [15], [16]. The latter methods have larger keyspace as they have increased the number of blocks. However, this increment is limited by the smallest allowable block size in the JPEG algorithm, for instance, block size no smaller than 16 × 16 should be used for color image compression. This recommended size is necessary to avoid block artifacts resulting from the JPEG chroma-subsampling step [10]. Smaller block size such as 8 × 8, can be utilized in the JPEG algorithm without any adverse effect, for compression of grayscale images. Therefore, to exploit the smaller block size for an expanded keyspace, Grayscale-PE represents color input as a pseudo-grayscale image by combining the color components along the horizontal or vertical direction [37], [38]. Overall, in conventional methods, color image as an input is a prerequisite for better security. However, in domains such as medical image processing, the unavailability of color images makes the conventional PE methods inadequate for their secure transmission and storage. Therefore, the current study proposes a PE method that is applicable for both color and grayscale images. In the proposed method, efficiency is achieved by considering smaller block size in encryption steps that have smaller effect on compressibility of an image, and, importantly, the processing does not compromise quality of the recovered images. As an application of the proposed method, we have considered a smart hospital that avails healthcare cloud services to outsource their DL computations and data storage needs.

Contributions. Our main contributions are summarized as:

- Proposed a PE algorithm for secure and efficient transmission and/or storage of medical images.

- A DL-based solution is implemented for automatic Tuberculosis (TB) screening in chest X-ray (CXR) images.
- Analysis of the proposed DL model against distortions resulted from compression process.
- Proposed noise-based augmentation method to improve generalization of DL model on smaller dataset.
- The analysis comprised of encryption, compression and DL-based classification were carried out on three datasets.

5.2. Related Work – Deep Learning-based Tuberculosis Screening

Grivkov et al. [84] implemented InceptionNetV3 [85] for diagnosis of TB in Shenzhen (SH) and Montgomery (MG) datasets [86] and achieved 86.8% accuracy. Das et al. [87] exploited transfer learning to improve InceptionV3 accuracy to 91.7% on the same datasets. Priya et al. [88] implemented transfer learning on VGG19 [89], ResNet50 [90], DenseNet121 [91] and InceptionV3 models. In their analysis, pre-trained VGG19 has achieved 89% and 95% best accuracies on MG and SH datasets, respectively. Cao et al. [92] implemented DenseNet121, VGG and ResNet152 [90] models and achieved best accuracy of 90.38% classification accuracy with DenseNet121. Rahman et al. [93] adopted a somewhat different approach than the aforementioned methods. They have used three pre-trained models (ResNet101 [90], VGG19, and DenseNet201 [91]) to extract features from CXR images and use eXtreme Gradient Boosting (XG-Boost) (1.6.1, Tianqi Chen and Carlos Ernesto Guestrin, Seattle, USA) [94] model to classify TB and non-TB in them. In their experiments, DenseNet201 with XG-Boost architecture achieved the highest accuracy of 99.92% as compared to its counterparts. Munadi et al. [95] proposed to enhance CXR quality before feeding them to pre-trained ResNet and EfficientNet [96] models. They have used three different image-enhancing techniques (unsharped masking, high-frequency emphasis filtering, and contrast limited adaptive histogram equalization). In their analysis, EfficientNet with unsharped masking image enhancement achieved 89.92% accuracy on SH dataset. Msnoda et al. [97] implemented ResNet, GoogLeNet [98], and AlexNet [99] with an extra Spatial Pyramid Pooling (SPP) [100] layer. Among the implemented architectures, GoogLeNet achieved the highest classification accuracy of 97%, which was then improved to 98% by using the SPP layer.

The methods discussed so far rely on the architecture of an individual model for classification efficiency. There are methods that combine two or even more models to form an ensemble network to achieve better performance. For example, Rajaraman et al. [101] implemented VGG16, InceptionResNetV2 [102], InceptionV3, XceptionNet [103] and DenseNet121, and then ranked them based on their accuracy. In their experiments, the top-3 models were InceptionV3 (accuracy = 94%), DenseNet121 (accuracy = 92.8%) and InceptionResNetV2 (accuracy = 92.5%). They have evaluated multiple ensemble methods to combine the top-3 models such as majority voting, simple averaging, weighted averaging stacking and blending to make an ensemble network. Their analysis showed that stacking ensemble demonstrated better performance and achieved 94.1% accuracy. Dasanayaka et al. [104] have implemented an ensemble of only two models (VGG16 and InceptionV3), and achieved 97.10% accuracy, which is higher than the ensemble of the three models proposed in [101]. Oloko-Oba et al. [105] have implemented an ensemble of VGG16, ResNet50 and InceptionV3 and achieved best accuracy of 96.14%. In their other study [106], they have explored ensemble of EfficientNets [96] for the diagnosis of TB. In their analysis of individual models, EfficientNet-B4 achieved best accuracy of 94.35% on SH dataset, which was then improved to 97.44% through ensemble learning. The ensemble was built by averaging the performance of the three best individual EfficientNets (B2, B3, and B4). Saif et al. [107] proposed to combine the traditional hand-engineered feature with an ensemble of DenseNet169, ResNet50 and InceptionV3 models. Their ensemble model has achieved best accuracy of 99.7% on SH dataset. Overall, ensemble methods have shown superior performance for TB screening in CXR images than the individual models.

5.3. Extension of CPE Methods for Grayscale Image Processing

Besides, color images encryption and compression, the CPE methods presented in Chater 4 can be used with the grayscale images as well. A grayscale image consists of only one component as opposed to a color image which has three components. The CPE methods consist of the following two steps for grayscale images encryption and compression:

Step 1. Block–based encryption:

The CPE methods perform geometric transformations to change blocks positions (blocks permutation) and orientations (blocks rotation and inversion), and intensity transformation (negative–positive transformation) to alter pixels values.

Step 2: Compression:

The final step is to compress the cipher image using the JPEG image standard in the grayscale mode either using the standard luminance or chrominance quantization tables.

For the grayscale input, the image representation step is omitted (Step 1 in Section 4.3) and the PE methods can be classified as methods that transform an entire block (GS–CPE) and methods that incorporate sub–block processing (GS–IIB–CPE). The methods Color–CPE, Extended–CPE and PGS–CPE are of class GS–CPE and IIB–CPE is of class GS–IIB–CPE. The following subsections provide an overview of these methods.

5.3.1. GS-CPE

A cipher image can be generated by following the procedure described below:

Step 1. Block–based encryption:

- The grayscale image $\boldsymbol{I}_{H,W}$ is divided into $L \times M$ blocks where $L = H/N$ and $M = W/N$, and each block has N^2 pixels.
- Shuffle the blocks positions in the image using a secret key $\boldsymbol{K}_1$ generated randomly.
- Change the blocks orientations in the shuffled image by a composite function of rotation and inversion transformations. This transformation is controlled by a randomly generated key $\boldsymbol{K}_2$.
- Change the pixels values by applying a negative–positive transformation function to each

pixel in a block randomly chosen using a random key $\boldsymbol{K}_3$ as in (37).

Step 2. Compression:

The final step is to JPEG compress the cipher image obtained in the previous step. Since the input image is a grayscale image, the JPEG compression is carried out in the grayscale mode with either of the standard quantization tables.

5.3.2. GS-IIB-CPE

A cipher image can be generated by following the procedure described below:

Step 1. Block–based encryption:

- The grayscale image $\boldsymbol{I}_{H,W}$ is divided into $L \times M$ blocks where $L = H/N$ and $M = W/N$, and each block has N^2 pixels.
- Perform inside–out transformation on each block. Divide each block into sub–blocks and then change the orientation of each sub–block. For example, a block $\boldsymbol{B}_{N,N}$ can be divided into $SL \times SL$ sub–blocks where $SL = N/SN$ and each sub–block has SN^2 pixels. Change the sub–blocks orientations in a given block by a composite function of rotation and inversion transformations with a random key $\boldsymbol{K}_1$.
- Shuffle the whole blocks positions using a secret key $\boldsymbol{K}_2$ generated randomly.
- Change the pixels values by applying a negative–positive transformation function to each pixel in a block randomly chosen by using a random key $\boldsymbol{K}_3$ as in (37).

Step 2. Compression:

The final step is to JPEG compress the cipher image obtained in the previous step. Since the input image is a grayscale image, the JPEG compression is carried out in the grayscale mode with either of the standard quantization tables.

5.4. Simulation Results and Analysis

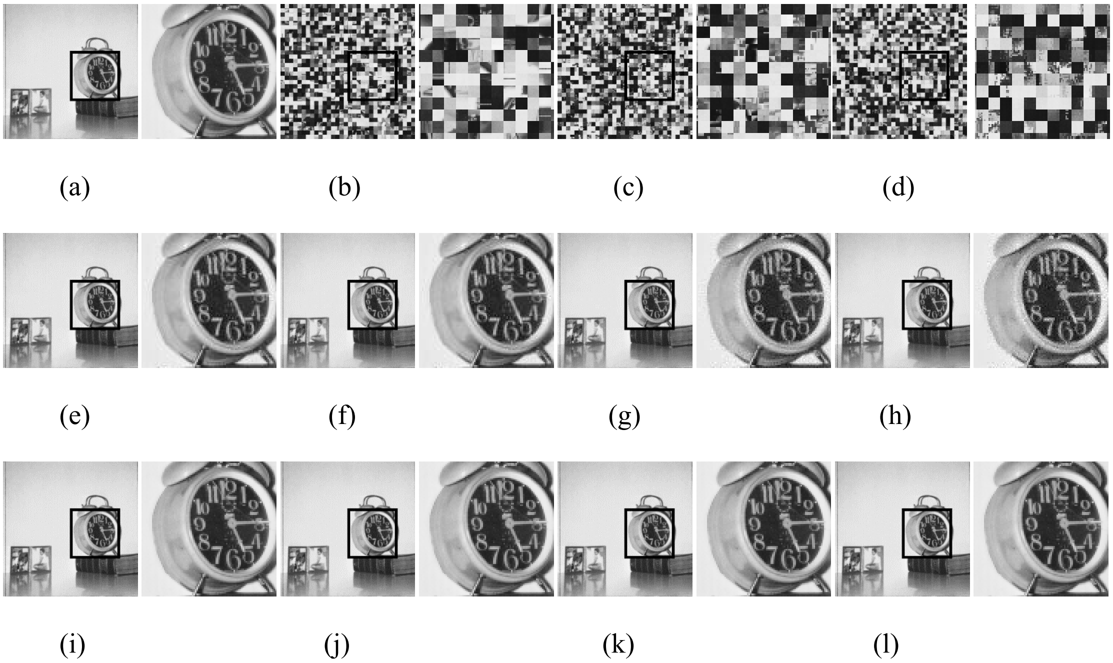


Figure 28. Visual analysis of the grayscale images recovered from the CPE processing. (a) is the original image. (b – d) are cipher images obtained from the GS–CPE, GS–IIB–CPE (4 × 4) and GS–IIB–CPE (2 × 2) methods, respectively. The images recovered were compressed with the JPEG $\boldsymbol{qf} = \mathbf{71}$ in (e – h), and $\boldsymbol{qf} = \mathbf{100}$ in (i – l). In each image, the boxed region is zoomed and shown its right side.

In this section, we present the security and compression efficiency of the proposed PE method against conventional PE methods. When input image is grayscale then conventional color-PE [15], [16], [65], [71] and grayscale-PE [37], [38] methods can be implemented in the same way as described in Section 5.3.1. For the analysis, Shenzhen dataset [86] was used, which consists of 662 images and all images were resized to same dimension of 2048 × 2048. The JPEG algorithm was implemented in grayscale mode with luminance quantization standard table and quality factors were chosen as $Q_f \in \{71, 72, \cdots, 100\}$. In addition, a deep learning model based on EfficientNetV2 [108] was implemented as an image classifier.

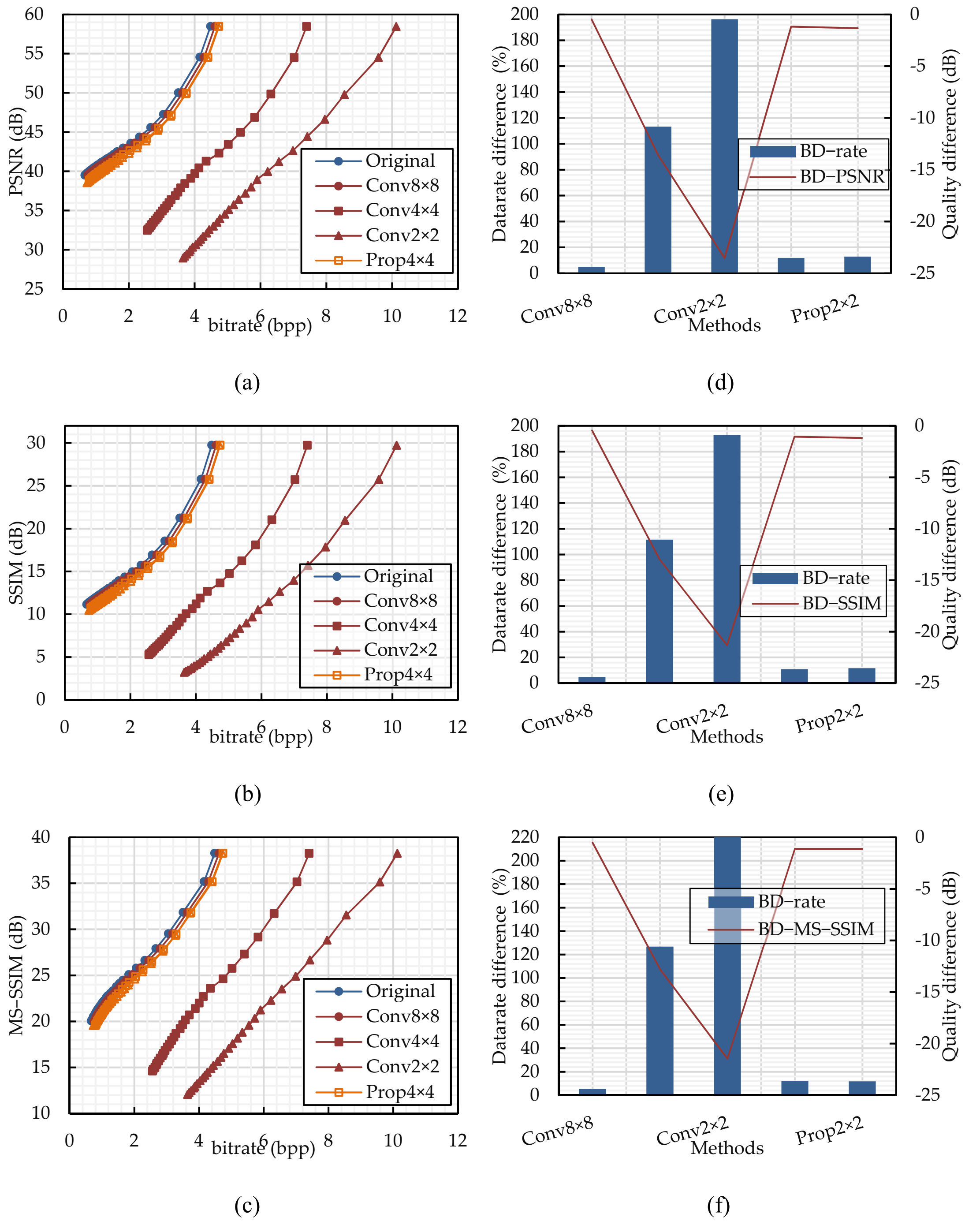
5.4.1. Visual Analysis

For the grayscale image visual analysis, Figure 28 (a) shows an example image from the USC–

SIPI dataset and its cipher images (b – d) obtained from GS–CPE and GS–IIB–CPE methods, respectively. For visual analysis, the square bounded area in each image is zoomed and shown beside their corresponding images. It can be seen that the global contents of the image are being scrambled. Owing to the sub–block processing, the GS–IIB–CPE method has achieved better visual encryption of the local details. The cipher images were compressed the JPEG algorithm under different quality factors and their corresponding recovered images are shown in Fig 27 (e – l). During compression, the quality factor was set to $qf = 71$ in Figure 11 (e – h), and $qf = 100$ in Figure 11 (i – l). The images recovered from the cipher images have the same visual appearance as the recovered plain images.

5.4.2. Compression Analysis

Rate distortion curves (RD) of compression algorithms give a relationship between encoded image qualities with respect to bit rates. For this purpose, different image quality metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [109], and MS-SSIM can be used. In the literature, only PSNR between original and PE encoded images has been computed as an image quality metric. However, SSIM and MS-SSIM are regarded as better measures to quantify visual degradation of an image. Therefore, this subsection presents analysis of PE methods with respect to SSIM (dB) and MS-SSIM (dB) along with PSNR (dB). SSIM and MS-SSIM values are $-10\log_{10}(1 - \mathrm{M})$, where M is either SSIM or MS-SSIM score. This logarithmic transform is necessary when methods can achieve high quality results. For compression analysis, images in the Shenzhen dataset were first encrypted by conventional and proposed PE methods and then the resulting cipher images were compressed using the JPEG algorithm. Figure 29 shows RD curves for different PE methods on the dataset using PSNR, SSIM, and MS-SSIM measures. For the RD curves, the JPEG quality factors were set to 70–100. The original images were obtained by the JPEG image compression without any encryption. The graphs show that compression efficiency of conventional methods decreases with smaller block sizes. On the other hand, for proposed method, block-size has smaller effect on the JPEG compression efficiency.



(a)

(d)

(b)

(e)

(c)

(f)

Figure 29. Compression performance analysis of perceptual encryption methods. (a–c) shows rate distortion curves for different image quality measures and (d–f) show rate savings (Bjøntegaard Delta) relative to JPEG under different image quality measures.

Although, the RD curves give a reliable subjective analysis; however, it is difficult to quantify performance difference between two encoding schemes, as they do not have points at the exact same bit rates. Therefore, Bjøntegaard delta (BD) [76] rate measures use a polynomial fit of the curves, then sample 100 points over the fitted curve and compute areas by using the trapezoidal integration method. As a result, BD measures summarize data rate savings (BD-rate) or quality improvements (BD-QM) between two codecs. The BD-rate metric gives percent difference in area between two RD curves for equivalent quality after the logarithmic transform of bitrate. Similarly, BD-QM gives an average difference in quality for equivalent bitrate. This study reports the BD measures for PSNR (BD-PSNR), SSIM (BD-SSIM) and MS-SSIM (BD-MS-SSIM) quality measures. Figure 29 shows the rate savings and quality improvements for different PE methods with respect to the JPEG compressed images using PSNR, SSIM and MS-SSIM, respectively. The compression of cipher images obtained from conventional PE methods requires 5% more bitrate for equivalent quality (in terms of MS-SSIM) of plain images compression. However, the bitrate drastically increases when smaller block size is used. For example, with block size 4 × 4, conventional method requires almost 113% more bitrate, and the requirement increases to almost 220% for block size 2 × 2. The reason is that for conventional PE methods, there is a tradeoff relation between encryption and compression efficiency because of the block size choice. In contrast, the proposed PE method requires about 12% more bitrate when using block size of (4 × 4) and (2 × 2). The analysis is discussed in terms of MS-SSIM; however, the same trend can be seen when using PSNR and SSIM.

5.4.3. Encryption analysis.

For the encryption of grayscale images, the CPE consisted of three secret symmetric keys: $\boldsymbol{K}_1$ permutation key, $\boldsymbol{K}_2$ rotation and inversion key, and $\boldsymbol{K}_3$ negative–positive transformation key. The keyspace $\boldsymbol{\mathcal{K}}$ of a CPE algorithm for the grayscale image encryption is the set of all keys used in the encryption steps as $\boldsymbol{\mathcal{K}} = \{\boldsymbol{K}_1, \boldsymbol{K}_2, \boldsymbol{K}_3\}$.

Similar to the encryption of color images, in the CPE methods an input grayscale image $\boldsymbol{I}_{W\times H}$, with $W \times H$ pixels, is divided into non–overlapping square blocks with N^2 pixels. The number of blocks B in the image is given by

$$B = L \times M, \tag{68}$$

and when a block $\boldsymbol{B}$ of size $N \times N$ pixels is divided into $SL \times SL$ smaller blocks of size SN^2 for the sub block processing, the number of sub blocks SB in the image is given by

$$SB = (SL \times SL) \times B. \tag{69}$$

The keyspace $\boldsymbol{\mathcal{K}}_{GC}$ for the GS–CPE schemes based on (68) can be derived as:

$$\begin{gathered}\boldsymbol{\mathcal{K}}_{GC} = \{\boldsymbol{K}_{1,GC}, \boldsymbol{K}_{2,GC}, \boldsymbol{K}_{3,GC}\},\\ |\boldsymbol{\mathcal{K}}_{GC}| = B! \cdot 8^{B} \cdot 2^{B}.\end{gathered} \tag{70}$$

The keyspace $\boldsymbol{\mathcal{K}}_{GIC}$ for the GS–IIB–CPE can be derived as:

$$\begin{gathered}\boldsymbol{\mathcal{K}}_{GIC} = \{\boldsymbol{K}_{1,GIC}, \boldsymbol{K}_{2,GIC}, \boldsymbol{K}_{3,GIC}\},\\ |\boldsymbol{\mathcal{K}}_{GIC}| = B! \cdot (8^{SB} \cdot 8^{B}) \cdot 2^{B}.\end{gathered} \tag{71}$$

Compared to GS–CPE where an entire block is transformed, GS–IIB–CPE has larger keyspace because of the sub–block processing in the rotation and inversion step.

5.4.4. DL-based TB Screening in CXR Images Analysis

Dataset. In this study, a publicly available dataset of postero-anterior chest radiograph called Shenzhen (SH) China dataset [86] was used. This dataset consists of 326 chest X-ray (CXR) images of normal cases and 336 CXR images of TB cases along with radiologist reading. To balance the dataset, we have used only 326 images from each class. The dataset was split into training, validation and testing sets which account for 80%, 10 and 10%, respectively. In addition, the input images were resized with the following steps: (1) all black borders and regions on the edges of images were cropped and (2) the images were resized to 224 × 224 dimensions from the center. In analysis, we considered samples with TB as positive and healthy samples as negative classes.

Proposed Augmentation Method. Data augmentation significantly increases the number of samples, which in turn can improve the accuracy of a deep learning model [110]. It is popular in domains, for example, medical image processing, where the datasets available are small and it is difficult to acquire more data [111]. The conventional data augmentation techniques are based on basic image manipulations such as geometric transformations, color transformations, mixing images and deep learning approaches such as adversarial training, neural style transfer, and GAN data augmentation [112], [113]. However, when the image data is corrupted due to noise, then noise-based data augmentation is performed to improve robustness and generalization of DL models and to overcome data deficiency [112]. The data augmentation can be carried out in two ways: on the fly during training—online augmentation and transforming and storing data on the disk before training—offline augmentation [110]. The main consideration when choosing either of the augmentation techniques is their associated additional memory and computational requirements. The online augmentation can save memory at the expense of slower training time whereas offline augmentation can provide efficiency during training at the cost of higher storage [110].

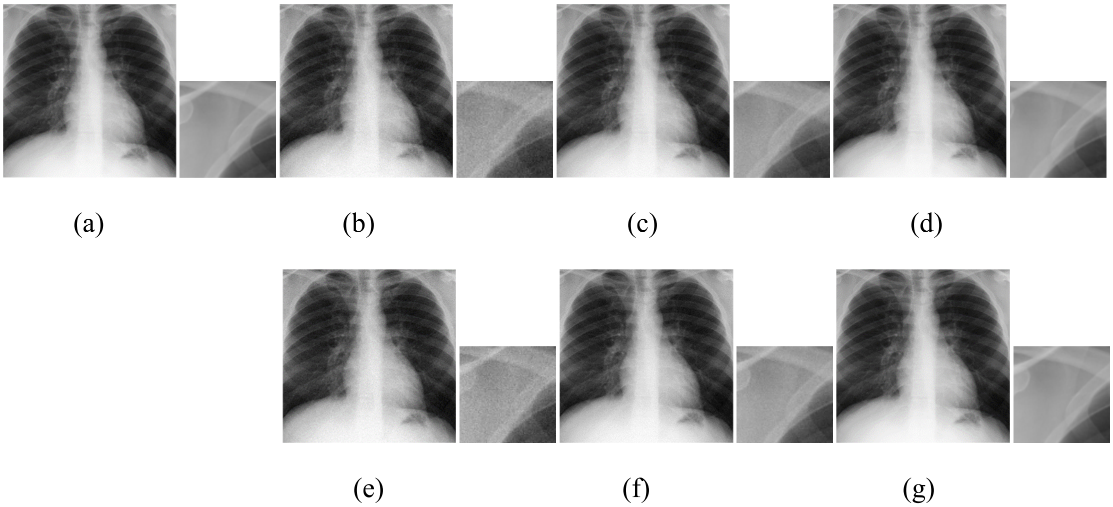


Figure 30. Example images of the proposed noise-based augmentation method. (a) is original image. (b–d) are conventional method decoded noisy images. (e–g) are proposed method decoded noisy images. The JPEG quality factor is set to (80, 90, 100) from left to right. The top left corners are zoomed in every image and are shown beside them.

In the current study, we have proposed a new noise-based data augmentation method that takes advantage of distortion resulting from the JPEG compression algorithm. The data augmentation is carried out offline to benefit from faster training time and as the dataset size is small, the storage requirement incurred is manageable. For this purpose, 528 images from the dataset were encrypted and compressed with the JPEG quality factors $Q_f \in \{71, 75, 80, 85, 90, 95, 100\}$, and then the original images were recovered with distortions. The resulting images were combined to form a dataset that consists of 4130 samples uniformly distributed between the two labels. Figure 30 shows sample images of the proposed data augmentation.

Proposed Model. The model used in this study is based on EfficientNet family [96], [108], which was selected based on their superior performance in natural images while having a low computational cost compared to popular CNNs such as VGGs and ResNets. Given their efficiency, they have been widely adopted in medical image processing domain as well [95], [105], [114]–[121]. The architecture of the proposed model is illustrated in Figure 31. It is based on EfficientNetV2-B0 [108], which has better parameter efficiency and faster training speed than the EfficientNetV1 [96]. These are achieved by combining training-aware neural architecture search with scaling during development of the model [108]. The proposed model consists of three Fused-MBConv[122] in early layers and three MBConv in later layers. The Fused-MBConv replaces the combination of depthwise Conv3×3 and expansion Conv1×1 in MBConv [96], [123] with a single regular Conv3×3 in order to utilize modern accelerators fully. Squeeze-and-Excite (SE) blocks [124] were utilized in MBConv layers to perform channel-wise feature recalibration for improved representational power of the model. The classifier consists of a fully connected layer followed by dropout and fully connected layers. The proposed model can either classify an observation to be positive or negative.

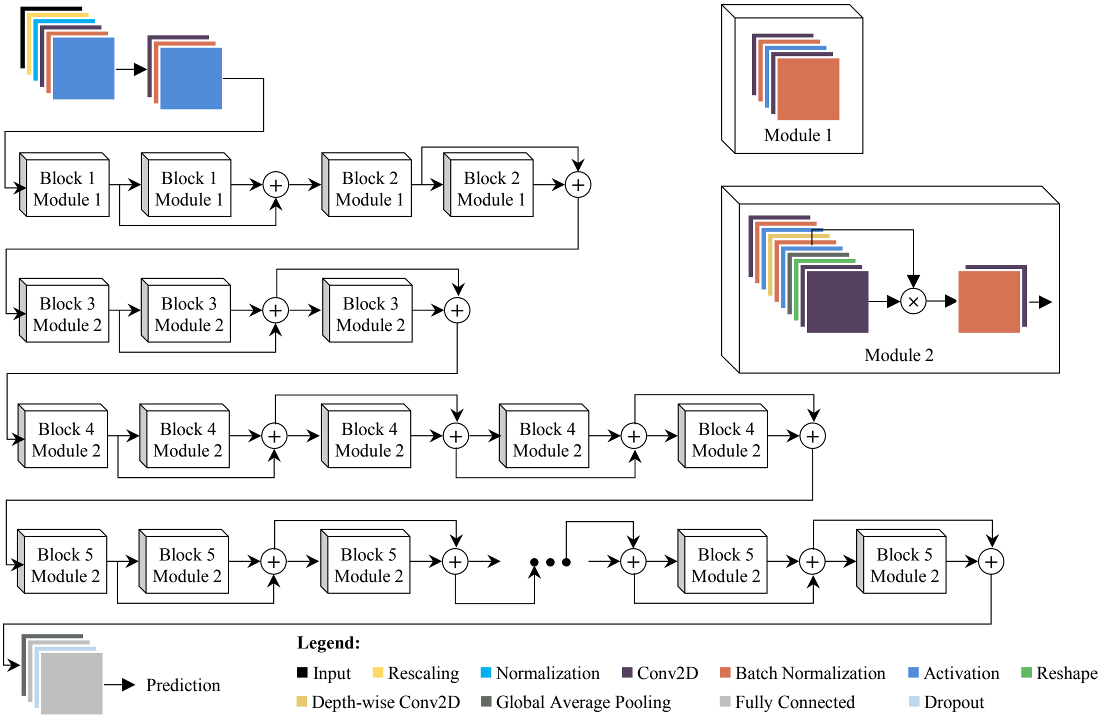


Figure 31. Illustration of the proposed deep learning model for automatic tuberculosis screening in chest X-ray images. The architecture is based on EfficientNetV2 model.

Analysis. In this study, we considered accuracy (27), sensitivity (28), and specificity (29) measures to evaluate performance of the proposed classifier. First, the model was trained on cleaned images for comparison with existing works. Next, to show robustness of the proposed model against different levels of distortions resulting from compression, the model was trained on images compressed with various JPEG quality factors. Finally, to improve accuracy of the model on limited amount of available data, the model is trained using proposed augmentation method.

Table 12 presents the accuracy of the proposed model for TB detection compared to existing works. Hwang et al. [125] proposed a deep CNN for TB classification, which achieved 83.7% accuracy without transfer learning. Pasa et al. [126] proposed a simple and fast CNN that achieved 84.4%. The main advantage of their model is being lightweight and the use of fewer parameters than the state-of-the-art models; however, this simplicity comes at a cost of being not robust against

Table 12. Performance analysis of the proposed deep learning model in terms of classification accuracy (%) on Shenzhen dataset with comparison of existing methods.

[121]	[122]	[123]	[74]	Proposed
83.7	84.4	85.0	87.0	89.52

different levels of compression noise as shown by [10]. An et al. [127] proposed E-TBNet, a deep neural network model based on efficient channel attention mechanism for automatic detection of TB. The E-TBNet achieved 85.0% accuracy on SH dataset. Showkatian et al. [78] proposed CNN model has achieved 87.0% accuracy, which is the highest accuracy on the SH dataset among the compared works. In comparison with existing works, our model based on EfficientNetV2 has achieved 89.52% accuracy for automatic classification of CXR images as normal or TB. Note that all of the aforementioned methods used basic transformations such as rotation, scaling etc. only on a training dataset to avoid overfitting.

In Table 12, the analysis is limited to the works that have presented their model performance on SH dataset without transfer learning. Otherwise, there are methods like ensemble and pre-training on larger dataset methods that can achieve better accuracies as summarized in Section 5.2. For example, Hwang et al. [125] model accuracy was improved from 83.7% to 90.3% when using transfer learning. Similarly, Showkatian et al. [78] compared pre-trained Exception, InceptionV3, ResNet50, and VGG models and their analysis showed that when using transfer learning, pre-trained Xception, ResNet50 and VGG16 achieved highest accuracy of 90.0%. For detailed review of ensemble and transfer learning methods for TB classification, please refer to [128].

Table 13 shows robustness of the proposed model against various levels of noise introduced by compression of cipher images. The accuracy of the model is shown alongside the images quality in terms of MS-SSIM and Perceptual Image Quality Evaluator (PIQUE) [129]. The MS-SSIM is a full-reference image matric that compares an input image against a reference image with no distortion. A full-reference matric measures the quality of a distorted image, as a human would perceive that. Therefore, it can be seen in Table 13 that the image quality degrades as Qf becomes smaller. On the other hand, PIQUE (also abbreviated as PIQE) is a no-reference image quality matric that exploits local block features for measuring an image quality. PIQUE value is in the range of 0–100 and the score are interpreted in steps of twenties, for example, 0–20 means excellent and 81–100 means bad quality of an image. Despite visible noise in the images, local features are still intact as suggested by the PIQUE measure. It can be seen that accuracy of the proposed model on different levels of distortion remains closer, that is, the variance is less than 2 for original, Conventional (8 × 8) and proposed (2 × 2) methods and less than 3 for Conventional (4 × 4), (2 × 2) and proposed (4 × 4) methods. Therefore, the proposed model is robust against compression distortions. The accuracy of a classifier gives its total number of correct predictions; however, in medical image analysis it is important to judge a model by how fewer number of FN are predicted. Table 14 presents sensitivity and specificity analysis of the proposed model. The model was able to achieve a lower error rate across all distorted images.

Table 13. Robustness analysis of the proposed model against different types of distortions resulted by the JPEG compression of plain and cipher images. Image quality is PIQUE | MS-SSIM, Acc is classification accuracy (%), σ^2 is variance and σ is standard deviation.

<table>
<tr><th rowspan="2">Q_f</th><th colspan="2">Original</th><th colspan="2">Conventional
(8 × 8)</th><th colspan="2">Conventional
(4 × 4)</th><th colspan="2">Conventional
(2 × 2)</th><th colspan="2">Proposed
(4 × 4)</th><th colspan="2">Proposed
(2 × 2)</th></tr>
<tr><th>Image Quality</th><th>Acc</th><th>Image Quality</th><th>Acc</th><th>Image Quality</th><th>Acc</th><th>Image Quality</th><th>Acc</th><th>Image Quality</th><th>Acc</th><th>Image Quality</th><th>Acc</th></tr>
<tr><td>−</td><td>43.16 | −</td><td>89.52</td><td>43.16 | −</td><td>89.52</td><td>43.16 | −</td><td>89.52</td><td>43.16 | −</td><td>89.52</td><td>43.16 | −</td><td>89.52</td><td>43.16 | −</td><td>89.52</td></tr>
<tr><td>71</td><td>42.65 | 38.49</td><td>87.9</td><td>42.60 | 38.49</td><td>84.68</td><td>42.19 | 32.59</td><td>84.68</td><td>40.43 | 31.66</td><td>84.68</td><td>43.37 | 38.28</td><td>85.48</td><td>43.51 | 38.34</td><td>84.68</td></tr>
<tr><td>75</td><td>42.77 | 39.21</td><td>87.1</td><td>42.77 | 39.21</td><td>86.29</td><td>42.37 | 32.81</td><td>84.68</td><td>41.18 | 32.00</td><td>84.68</td><td>43.38 | 39.05</td><td>86.29</td><td>43.41 | 39.08</td><td>84.68</td></tr>
<tr><td>80</td><td>42.87 | 40.28</td><td>84.68</td><td>42.92 | 40.33</td><td>85.48</td><td>42.75 | 33.17</td><td>86.29</td><td>41.81 | 32.47</td><td>86.29</td><td>43.45 | 40.11</td><td>88.71</td><td>43.54 | 40.13</td><td>84.68</td></tr>
<tr><td>85</td><td>43.12 | 41.09</td><td>86.29</td><td>42.92 | 41.11</td><td>87.9</td><td>42.83 | 33.47</td><td>85.48</td><td>42.43 | 32.92</td><td>84.68</td><td>43.44 | 40.90</td><td>87.9</td><td>43.54 | 40.90</td><td>87.1</td></tr>
<tr><td>90</td><td>43.03 | 42.90</td><td>84.68</td><td>43.13 | 42.9</td><td>86.29</td><td>43.06 | 33.78</td><td>88.71</td><td>42.73 | 33.39</td><td>88.71</td><td>43.31 | 42.62</td><td>84.68</td><td>43.41 | 42.61</td><td>85.48</td></tr>
<tr><td>95</td><td>43.02 | 44.04</td><td>87.9</td><td>43.07 | 44.04</td><td>84.68</td><td>43.11 | 34.03</td><td>84.68</td><td>43.06 | 33.83</td><td>84.68</td><td>43.15 | 43.86</td><td>84.68</td><td>43.14 | 43.83</td><td>87.1</td></tr>
<tr><td>100</td><td>43.13 | 45.99</td><td>86.29</td><td>43.13 | 45.88</td><td>87.1</td><td>43.12 | 34.21</td><td>87.1</td><td>43.20 | 34.15</td><td>84.68</td><td>43.11 | 45.88</td><td>84.68</td><td>43.10 | 45.63</td><td>86.29</td></tr>
<tr><td>σ²</td><td>0.03 | 7.37</td><td>1.82</td><td>00.04 | 7.19</td><td>1.45</td><td>0.14 | 0.37</td><td>2.35</td><td>1.06 | 0.87</td><td>2.38</td><td>0.02 | 7.44</td><td>2.75</td><td>0.03 | 6.97</td><td>1.24</td></tr>
<tr><td>σ</td><td>0.18 | 2.71</td><td>1.35</td><td>00.20 | 2.68</td><td>1.21</td><td>0.37 | 0.61</td><td>1.53</td><td>1.03 | 0.93</td><td>1.54</td><td>0.14 | 2.73</td><td>1.66</td><td>0.19 | 2.64</td><td>1.11</td></tr>
</table>

Table 15. Performance analysis of the proposed model in-terms of specificity (SPE) and sensitivity (SEN).

<table>
<tr><th rowspan="2">Q_{f.}</th><th colspan="2">Original</th><th colspan="2">Conventional
(8 × 8)</th><th colspan="2">Conventional
(4 × 4)</th><th colspan="2">Conventional
(2 × 2)</th><th colspan="2">Proposed
(4 × 4)</th><th colspan="2">Proposed
(2 × 2)</th></tr>
<tr><th>SPE|SEN</th><th>Acc</th><th>SPE|SEN</th><th>Acc</th><th>SPE|SEN</th><th>Acc</th><th>SPE|SEN</th><th>Acc</th><th>SPE|SEN</th><th>Acc</th><th>SPE|SEN</th><th>Acc</th></tr>
<tr><td>−</td><td>0.87 | 0.92</td><td>89.52</td><td>0.87 | 0.92</td><td>89.52</td><td>0.87 | 0.92</td><td>89.52</td><td>0.87 | 0.92</td><td>89.52</td><td>0.87 | 0.92</td><td>89.52</td><td>0.87 | 0.92</td><td>89.52</td></tr>
<tr><td>71</td><td>0.87 | 0.89</td><td>87.9</td><td>0.81 | 0.89</td><td>84.68</td><td>0.87 | 0.82</td><td>84.68</td><td>0.79 | 0.90</td><td>84.68</td><td>0.87 | 0.84</td><td>85.48</td><td>0.76 | 0.94</td><td>84.68</td></tr>
<tr><td>75</td><td>0.84 | 0.90</td><td>87.1</td><td>0.81 | 0.92</td><td>86.29</td><td>0.81 | 0.89</td><td>84.68</td><td>0.77 | 0.92</td><td>84.68</td><td>0.81 | 0.92</td><td>86.29</td><td>0.76 | 0.94</td><td>84.68</td></tr>
<tr><td>80</td><td>0.79 | 0.90</td><td>84.68</td><td>0.79 | 0.92</td><td>85.48</td><td>0.82 | 0.90</td><td>86.29</td><td>0.81 | 0.92</td><td>86.29</td><td>0.85 | 0.92</td><td>88.71</td><td>0.82 | 0.87</td><td>84.68</td></tr>
<tr><td>85</td><td>0.79 | 0.94</td><td>86.29</td><td>0.87 | 0.89</td><td>87.9</td><td>0.85 | 0.85</td><td>85.48</td><td>0.84 | 0.85</td><td>84.68</td><td>0.85 | 0.90</td><td>87.9</td><td>0.81 | 0.94</td><td>87.1</td></tr>
<tr><td>90</td><td>0.82 | 0.87</td><td>84.68</td><td>0.77 | 0.95</td><td>86.29</td><td>0.87 | 0.90</td><td>88.71</td><td>0.90 | 0.87</td><td>88.71</td><td>0.79 | 0.90</td><td>84.68</td><td>0.84 | 0.87</td><td>85.48</td></tr>
<tr><td>95</td><td>0.85 | 0.90</td><td>87.9</td><td>0.79 | 0.90</td><td>84.68</td><td>0.85 | 0.84</td><td>84.68</td><td>0.79 | 0.90</td><td>84.68</td><td>0.84 | 0.85</td><td>84.68</td><td>0.82 | 0.92</td><td>87.1</td></tr>
<tr><td>100</td><td>0.76 | 0.90</td><td>86.29</td><td>0.90 | 0.84</td><td>87.1</td><td>0.76 | 0.85</td><td>87.1</td><td>0.81 | 0.89</td><td>84.68</td><td>0.85 | 0.84</td><td>84.68</td><td>0.79 | 0.94</td><td>86.29</td></tr>
<tr><td>σ²</td><td>0.17 | 0.04</td><td>1.82</td><td>0.22 | 0.11</td><td>1.45</td><td>0.15 | 0.13</td><td>2.35</td><td>0.2 | 0.07</td><td>2.38</td><td>0.08 | 0.13</td><td>2.75</td><td>0.14 | 0.09</td><td>1.24</td></tr>
<tr><td>σ</td><td>0.04 | 0.02</td><td>1.35</td><td>0.05 | 0.03</td><td>1.21</td><td>0.04 | 0.04</td><td>1.53</td><td>0.04 | 0.03</td><td>1.54</td><td>0.03 | 0.04</td><td>1.66</td><td>0.04 | 0.03</td><td>1.11</td></tr>
</table>

Table 14. Performance analysis of the proposed noise-based augmentation method.

[126]	Proposed Data Augmentation Method					
	Original	Conventional			Proposed	
		8 × 8	4 × 4	2 × 2	4 × 4	2 × 2
99.86	99.77	99.77	99.54	99.31	99.71	99.77

Finally, Table 15 shows the efficiency of the proposed noise-based augmentation method for TB screening. W0ng et al. [130] proposed TB-Net, a self-attention deep CNN for automatic TB detection. Their dataset consists of 6939 CXR images collected from three different datasets. The TB-Net achieved 99.85% accuracy. In addition, they have also used EfficientNetB0 on the same dataset, and achieved 98.99% accuracy. On the other hand, our proposed model has achieved 99.77% classification accuracy on original images and preserved the same accuracy across different PE decoded images. Note that a user can compress images with different quality factors depending on their bandwidth requirements. Therefore, to accommodate this in our analysis, the test set was compressed with different quality factors and the model performance was evaluated on distorted images. The test accuracy presented in Table 15 is an average value across different quality factors.

5.5. Compression Artifacts Impact on DL Model Performance and its Remedy

5.5.1. Motivation

Images are characterized by their large volume of data. Therefore, compression is often carried out to give them a compact representation in order to efficiently utilize the available limited bandwidth and storage resources. Unlike textual data, images can be compressed either in lossless or lossy mode. In the lossless mode, the compression is carried out in such a way that the original quality is ensured while the lossy mode introduces certain degradation in the image quality. The lossless mode offers better image quality whereas lossy mode offers better compression savings. The JPEG standard [21] is one of the widely used image compression standards and is available on most consumer devices and on the internet [131]. The JPEG algorithm was designed in the early 90s with human subjects in mind. Therefore, the perceptual quality in the lossy compressed images was based on the characteristics of human visual system (HVS). For example, HVS is more sensitive to brightness and low frequencies than color and high frequencies. Therefore, two images with same brightness but one with lower color resolution than the other will appear similar to the human. Though, lossy image compression provides an efficient solution to the exchange and storage of large volumes of image data for various applications. It is necessary to analyze its impact on a trained DL model.

5.5.2. Related Work

DL models have achieved state-of-the-art performance across various domains, especially for image classification tasks [7]. For an efficient model, it is necessary that the training and testing are performed against the data that come from the same target application distribution [110]. However, when the distribution is altered by distortions such as blur and noise in the images then the classification accuracy degrades [132]. For example, [132], [133] have shown that the DL models' performance is influenced by the noise, blur, image correction and compression artifacts. In [134], the authors have shown that the JPEG compression is friendly to DL-based object detectors. The authors in [135] have studied the effects of image compression on the performance of DL models in the context of pathology image analysis and showed the models can preserve accuracy error under 5% on images compressed by 85%. Similarly, the authors in [7] have shown that the JPEG distortions reduce DL based tuberculosis classification accuracy by 2%. The authors in [136] have shown that the quantization of the color information in an image reduces the DL model accuracy by 2%. The most comprehensive analysis of the JPEG impact on DL based different computer vision tasks is carried out in [131].

The prior works stated above mainly analyzed the impact of JPEG noise on DL performance while they do not provide any solution for mitigating this penalty. The only exception is [131] that have proposed artifact correction method to improve the model performance in applications where labeled data is not available. Artifact correction as a preprocessing step is able to preserve the model performance on uncompressed images as well. However, from the inference speed point of view any preprocessing is not desirable. Therefore, we propose a data augmentation strategy to enhance the quality of training datasets such that the model generalizes well on any future noisy images. Specifically, we considered the JPEG distortions to generate new images. The proposed method does not require artifact correction as a preprocessing step and can preserve the model performance on the uncompressed images.

5.5.3. Proposed Noise-Based Augmentation Method

Figure 32. shows the system model that we have assumed in this study. The production cycle of a DL model consists of two phases: development phase and deployment phase. In the first phase, the model is trained and validated on available images in order to learn how to solve a problem. In the second phase, the trained model is deployed to solve the real-world problems related to the domain they were previously trained on. For the DL model to perform well after deployment, it is necessary that the model is trained and tested against data that come from the same target application distribution. However, in practice this is usually not the case and there are several potential sources of bias such as positional, translational, lightning, color and style that can separate the training and testing data distributions [110]. Here, we assumed that in the development phase the available data is clean, and the images produced by the end devices have the JPEG artifacts. Thus, the source of bias for the data distribution is noise.

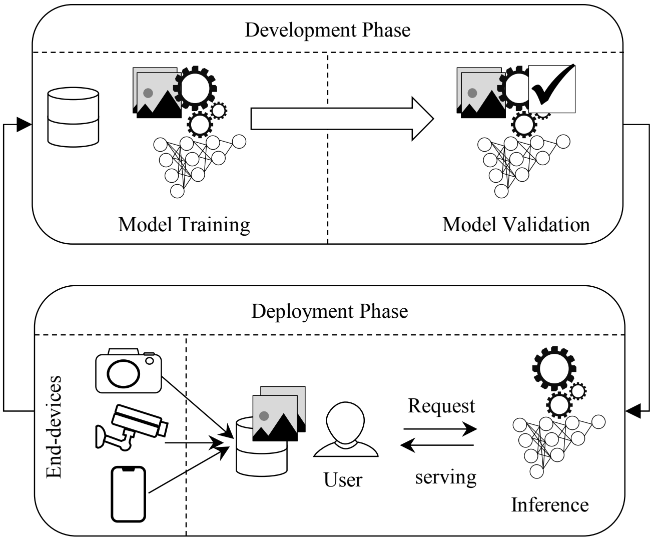


Figure 32. An illustration of the system model describing the production cycle of DL models.

Data augmentation is one of the techniques that has been developed to avoid overfitting in DL models. The data augmentation methods include geometric and colorspace transformations, adversarial training, and neural style transfer methods, to name a few. Each of these methods is designed to deal with the bias that exists in the data. However, when noise is the potential bias that separates the distributions of the training and testing data, then the addition of noise can be one approach to data augmentation [112]. In this study, we proposed a data augmentation technique that uses the JPEG distortions to generate new images. The proposed method is called noise-cuts-noise approach as the noise is used for image augmentation to make the DL models robust against the noise.

In the JPEG algorithm, the information loss is resulted from the quantization (Q) forward and inverse functions. The two transformation functions, that is, colorspace conversion function (C) and DCT function (D) are lossless in nature; however, when their values are rounded or truncated then certain information is lost. For an image I, its JPEG compressed image I_c without the chroma-subsampling can be defined as:

$$I_c = \left[Q\left(D\left(C_{YCbCr}(I)\right), Qf\right)\right], \tag{72}$$

and I_c can be decompressed to obtain I_d as

$$I_d = \left[\left[C_{RGB}\left(\left[D^{-1}\left(Q^{-1}(I_c, Qf)\right)\right]\right)\right]\right|_{0}^{255}, \tag{73}$$

where, [.] and [.| rounds and truncates the values to a valid range in the spatial domain, respectively. For a given dataset, the JPEG noisy images using (3) and (4) can be generated as shown in Figure 33.

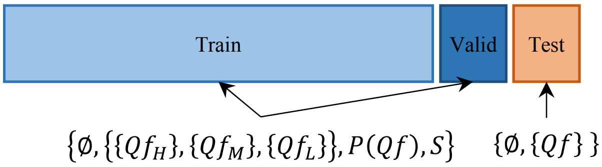


Figure 33. The parameters specification for the proposed noise-based image augmentation method. The element $\emptyset$ means that no distortion is applied to the images, *Qf* is the JPEG quality factor, *Qf*$_x$ is the compression level and *S* determines the size of a dataset.

For the proposed augmentation method, a set is specified whose elements consists of Qf range, along with their probabilities $P(Qf)$ and a factor S that determines the augmented dataset size. In order to avoid a mismatch between training and the validation datasets, the JPEG noise can be generated in the validation images. In this case, the factor S should always be equal to 1. As mentioned earlier, that the end devices produce distorted images; therefore, the holdout test dataset is compressed with each value in the specified Qf range. For the augmentation, instead of using the whole range of Qf values, a specific compression level can be chosen for example, high compression (Qf_H), moderate compression (Qf_M) and light compression (Qf_L).

5.5.4. Simulation Results and Analysis

In our experiments, we chose Qf in the range of $[25,100]$ and assumed Qf value in range $[25,45]$ as Qf_H, range $[50,70]$ as Qf_M and range $[75,100]$ as Qf_L compression levels. Throughout our analysis, the images were compressed before any pre-processing required by the model. For the evaluation, the images were always compressed at each Qf value in steps of 5 in the

Table 16. Different configurations used during the analysis.

Training	Pseudonym	Qf range	Step size	S	$P(Qf)$	Validation images
Naïve methods	Setup 1	[25,100]	5	-	-	Noisy
	Setup 2 (a)	[25,100]	5	-	Uniform	Clean
	Setup 2 (b)	[25,100]	5	-	Uniform	Clean
Proposed augmentation method	Setup 3 (a)	[25,95]	5	2	Uniform	Clean
	Setup 3 (b)	[25,60	5	2	Uniform	Clean
	Setup 3 (c)	{25}	-	2	-	Clean
	Setup 4 (a)	[25,60]	5	2	Uniform	Clean
	Setup 4 (b)	[25,60]	5	3	$P(25) = 0.5$	Clean
	Setup 4 (c)	[25,60]	5	3	$P(25) = 0.5$	Noisy
	Setup 4 (d)	[25,60]	5	3	$P(25) = 0.5$	Mixed
	Setup 5 (a)	[25,55]	10	2	Uniform	Clean
	Setup 5 (b)	[25,55]	10	2	Uniform	Noisy
	Setup 5 (c)	[25,55]	10	2	Uniform	Mixed

specified range. The rest of the parameters are summarized in Table 16.

For baseline methods, we evaluated the models with two naïve approaches namely, Setup 1 and Setup 2. In Setup 1, the model was trained and tested on noisy images. Here, we assumed that distorted images were available during training. In Setup 2, the model was trained on mixed images, that is, the images in the training set were compressed with different JPEG quality factors. Specifically, mixed of compressed and never compressed images (Setup 2 (a)) and mixed of only compressed images (Setup 2 (b)) were considered. All the images in the dataset were unique and the Qf values were chosen uniformly for compression.

For the proposed data augmentation method, we considered three different configurations. 1) Setup 3: the dataset size is increased from N to $2N$ and Qf for compression were chosen uniformly from a specified range. 2) Setup 4: the dataset size is increased from N to $3N$ by using the best augmentation method obtained in Setup 3. Here, $P(Qf_{25})$, the probability of choosing Qf_{25} was 0.5. In both setups, the images were compressed with Qf values in steps of 5 in the given range. 3) Setup 5: The Qf values were chosen in steps of 10 for the best augmentation method obtained in Setup 3.

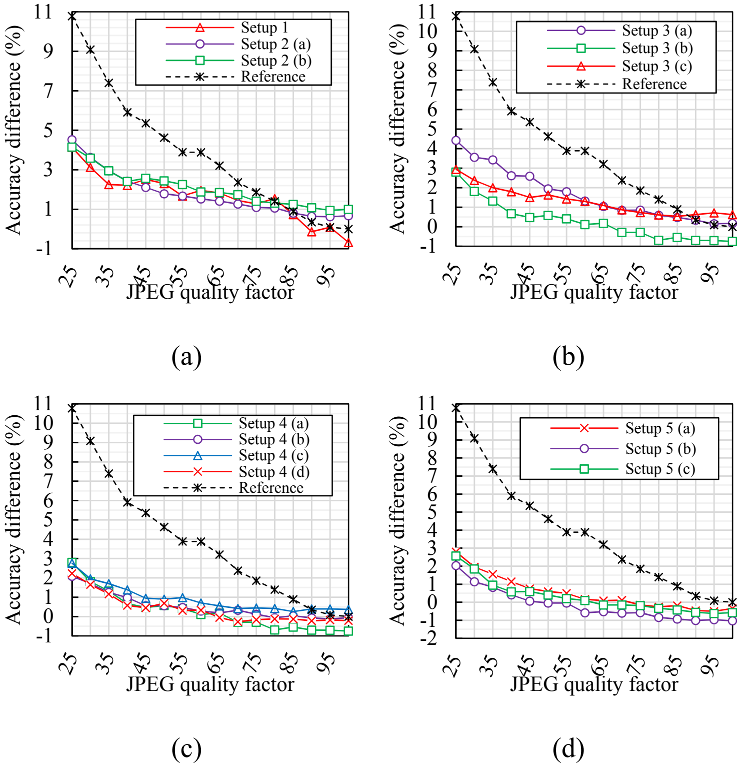
5.5.4.1. Deep Learning Robustness Analysis

In this section, we first present our analysis on natural images classification and establish optimal configurations for the proposed noise-based data augmentation method. To show the benefits of the proposed method, the feature space analysis is conducted using t-Distributed Stochastic Neighbor Embedding (t-SNE) [137] visualization. Finally, as an application of the proposed method, we design a case study of medical image analysis for multi-label classification task. In both cases, we have used EfficientNetV2-B0 [108] with the parameters suggested in [7] as our classification model.

5.5.4.1.1. Natural Images Classification Analysis:

For this purpose, we have used the CIFAR10 dataset [138], which consists of 50K training and 10K testing images. We have used the test set as a holdout subset to calculate a final estimate of the DL model's performance. From the training set, 10K images were used for the model validation.

When the model is trained and validated on original images without any JPEG distortions, it achieved the classification accuracy of 88% on the test set. The trained model's performance was then evaluated on distorted images as shown in Figure 34 (the Reference graph). When the model is trained only on clean images then its performance significantly degraded on distorted images especially on highly and moderately compressed images. In other words, the model did not generalize well and is not robust against the JPEG defects. To improve the model generalization, we first analyzed the naïve approaches. When the model is trained on noisy images (Setup 1) and evaluated on the corresponding noisy test images, then a performance improvement can be observed. The improvements are mainly in highly and moderately compressed images while no significant improvements have been achieved on the lightly compressed images. Next, when the model is trained on mixed images (Setup 2), its performance has improved on the highly and moderately compressed images; however, the accuracy on lightly images suffered. Including the original images in the mix (Setup 2 (a)) has no significant gain in the model performance compared to Setup 2 (b). For the proposed method, we performed the following three analysis.



(a)

(b)

(c)

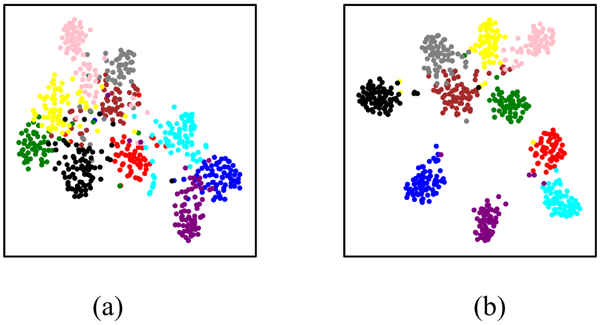
(d)

Figure 34. The DL model performance with naïve training and proposed augmentation method on the JPEG distorted CIFAR-10 dataset. The reference curve was obtained by training the model on clean images and testing on distorted images. The naïve methods performance is compared in (a) and the proposed method is compared in (b)–(d).

Optimal Qf range analysis. For the first configuration of the proposed augmentation method (Setup 3), the training set size was increased from N to $2N$, and the augmented images were obtained by compressing the original images. The Qf values for the compression were selected uniformly from different ranges. For example, in Setup 3 (a) the images were compressed with Qf values belonging to all compression levels, that is $[25,95]$. With the proposed augmentation method, the model was able to achieve better accuracy on highly and moderately compressed images while preserving the same accuracy on lightly compressed images as Figure 34 (b). Since our goal is to improve the model performance in the low bitrate regions, next we considered compression levels correspond to high and moderate levels only. For example, in Setup 3 (b), the Qf range is $[25,60]$ while follows the same configuration as Setup 3 (a). The model performance further improved across all Qf values. Inspired by this improvement, in Setup 3 (c) a single value $Qf = 25$ is used for augmentation. On the heavily compressed images, the model performance further improved compared to Setup 3 (a), but no improvement has been achieved compared to Setup 3 (b). In addition, the model performance suffered on the lightly compressed images. Among the different augmentation settings analyzed in Setup 3, Setup 3 (b) performed better. Therefore, next we considered increasing the dataset size under Setup 3 (b) configuration to improve the model performance further.

Optimal dataset size analysis. For the second configuration of the proposed augmentation method (Setup 4), the augmented images were obtained using Setup 3 (b). Specifically, the training set size was increased from N to $3N$ in such a way that N were the original images, N images were augmented with $Qf = 25$ and N images were augmented with Qf selected from the range $[30,60]$. As a result, Setup 4 uses non-uniform probability to choose a Qf value from the specified range. The probability $P(Qf_{25}) = 0.5$ is inspired from the Setup 3 (c) analysis. In addition, we also considered using a mix of clean and noisy images for the validation dataset to avoid the mismatch between the training and validation data distributions. This is achieved by replacing the original images with uniform probability in the validation set by compressed images. The images were compressed with each value of Qf in the range specified for the training set. For test images compressed at $Qf = 25$, the accuracy difference further reduced by 1% in Setup 4 (b) and (d) compared to Setup 3 (b) as shown in Figure 34. (c). However, for different compression levels, Setup 3 (b) has better performance than Setup 4 (b) – (d). Therefore, we select Setup 3 (b) in our next analysis. In addition, when using only clean images (Setup 4 (b)) or mix of clean and noisy images (Setup 4 (d)) are better than using only noisy images for the validation.

Optimal step size analysis. The optimal Qf range and dataset size are obtained in Setup 3



(a) (b)

Figure 35. Feature space analysis using 2D t-SNE visualization of the embeddings obtained from (a) based model and (b) model trained with the proposed augmentation method. Each color of the dots represents the CIFAR-10 classes. The images were compressed with the JPEG algorithm for *Qf=25*.

and Setup 4, respectively. Finally, we analyzed the optimal step size to choose Qf values in a given range for augmentation. For the third configuration analysis of the proposed augmentation method (Setup 5), the images were augmented using Setup 3 (b) and the dataset size was increased from N to $2N$. Instead of step size equal to 5 as in Setup 3 (b), this time it is set to 10. The reason is that an image compressed with values that are nearby in the Qf range have similar quality; therefore, increasing the step size will generate images of varying qualities. Setup 5 (a)–(c) have differences in their validation sets. The analysis showed that the model performance is significantly improved across all levels of compression as shown in Figure 34 (d). In addition, it can be observed that changing the distribution of the validation set has a negligible impact on the performance gain.

Feature space analysis. To show the advantage of using the proposed augmentation method during training, we calculate 2-dimensional t-SNE feature embeddings from the layer prior to SoftMax non-linearity in the model. Figure 35. plots the embeddings to analyze the effect of the JPEG noise on the trained model. The embeddings separations produced by the model trained without and with the proposed noise-based augmentation method are shown in Figure 35 (a) and (b), respectively. It can be seen that the proposed method provides noticeably better separation between the CIFAR-10 classes, resulting in better classification performance.

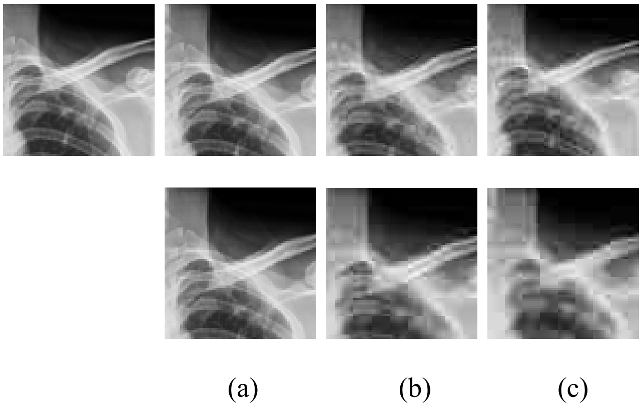


Figure 36. Example images of the JPEG compressed dataset. (a) is original image. (b) – (c) are compressed with *Qf*={100, 50, 25}, respectively. The luminance and chrominance standard tables are used in first and second rows, respectively.

5.5.4.1.2. Case Study – Chest X-ray images classification:

As an application of the proposed method, we have considered multi-label CXR images classification task. For this purpose, we have used the TBX11K dataset [139] that consists of CXR images collected from 4 publicly available CXR images datasets. The dataset has samples belonging to three labels: healthy, sick and tuberculosis. To balance the dataset, we have chosen a total of 2,400 images, that is 800 images per class. The dataset was initially split into 90% for training and 10% for testing. From the training set, 20% of the images were used for the model validation. The original images are of 512×512 resolution, which we have resized to 224×224 to meet the model input size requirement. For visual analysis of the distortions resulting from the JPEG compression, Figure 36 shows compressed images examples. The compression was carried out with different JPEG quality factors and quantization tables. From the visual appearance of the images, it can be seen that the JPEG defects become more visible with smaller Qf values. In addition, when the standard luminance table is used instead of the standard chrominance table during quantization, then the images are recovered with better quality. To quantify the degradation of image quality, Figure 37. plots MS-SSIM value for each quality factor.

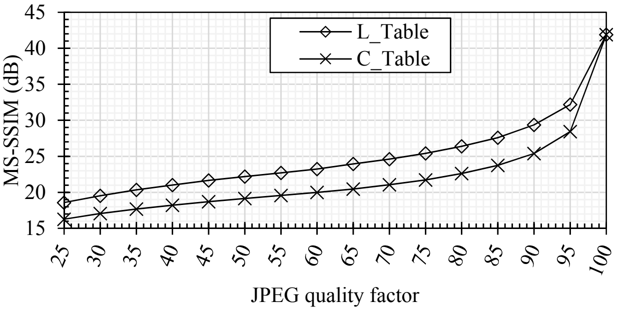
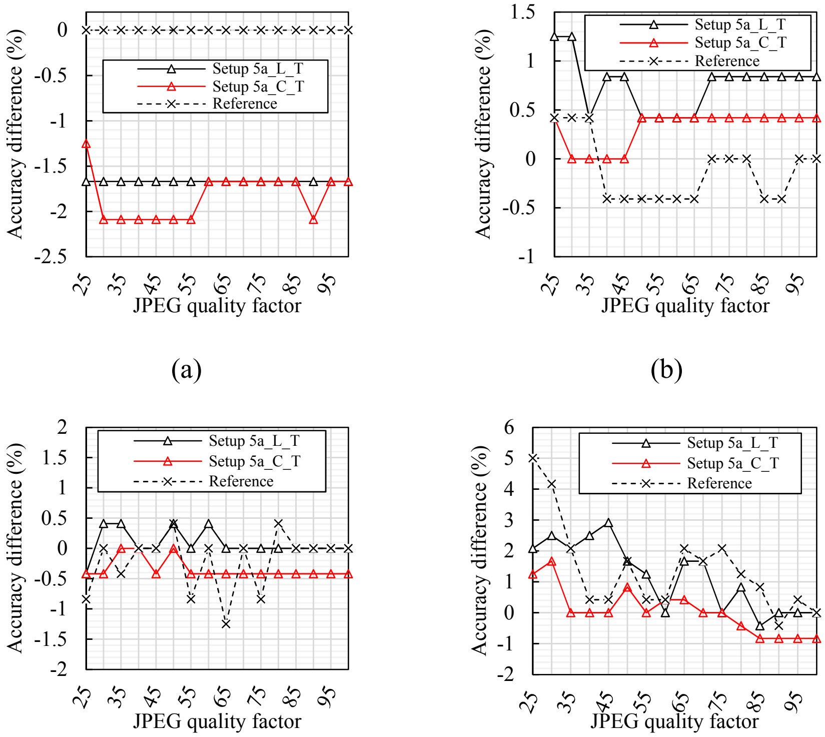


Figure 37. The image quality comparison in terms of MS-SSIM score between the JPEG compressed images using luminance (L_Table) and chrominance (C_Table) quantization tables.

The model was first trained and validated on clean images, which was then evaluated on the clean holdout set. To analyze the model robustness against the JPEG distortion for the CXR image classification task, the model performance was evaluated on the compressed test set. For grayscale images compression, either of the JPEG standard quantization tables can be used. Therefore, the test set was compressed with both tables and the DL performance was compared on them.

Image resolution analysis. The analysis presented in Section 5.5.4.1.1. was performed on images with smaller resolution that is, 32×32. However, with higher resolution images a DL model performance improves and generalizes well. The dataset we have used for our case study consists of images with higher resolution 512×512, as described earlier. Therefore, we have experimented with different image resolutions as in Figure 38 and 39. For example, the images were resized to 32×32, 64×64, 128×128 and 224×224 sizes. When the full resolution images were used, then the trained model is robust against the JPEG distortions for both quantization tables across all the Qf values. However, when the resolution of the images is reduced to 128×128 and 64×64, then the model performance varies by ±1.5%. For the smallest images resolution (32×32), the difference in accuracy increased by 5% and 26% at maximum for luminance and chrominance tables, respectively. The analysis presented so far is for the model performance without our proposed augmentation method. Next, we trained the model with the proposed augmentation configuration that performed well for the natural images classification that is, Setup 5 (a). For the full resolution images though, there was no performance gap and the model generalized well on the distorted images, with the proposed augmentation method the model performance further improved by 2%. For images resolution 128×128 and 64×64, there is no significant difference in the model performance. However, for smaller images, where the difference in accuracy was drastically degraded (specifically for the highly compressed images), with the proposed augmentation method, the error is reduced to only 2% and 3% for images compressed with the luminance and chrominance quantization tables, respectively.



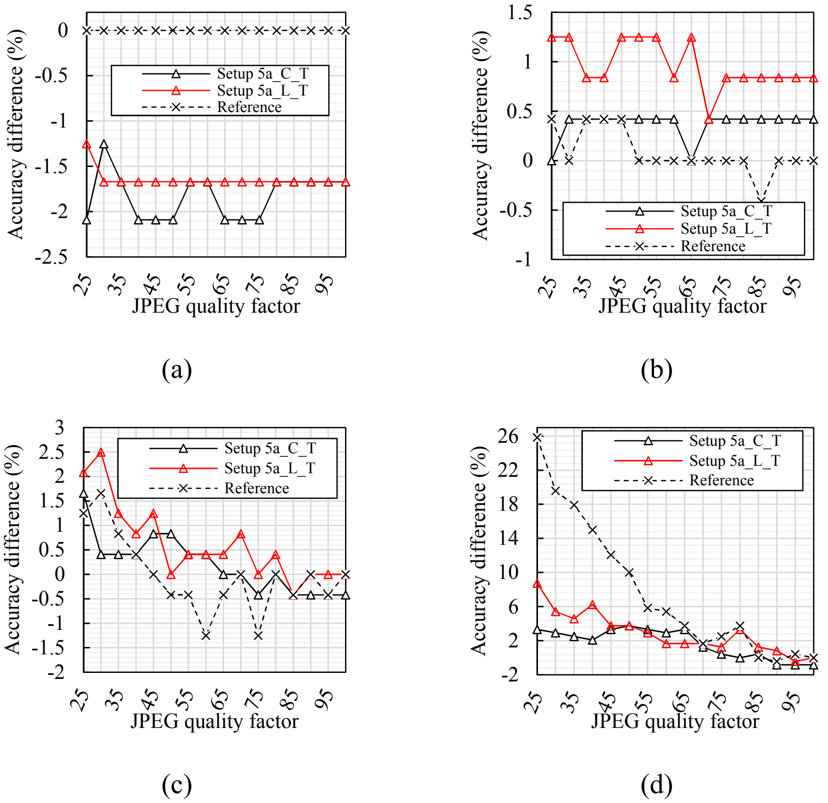
(a)

(b)

(c)

(d)

Figure 38. The DL model performance with the proposed augmentation method (setup 5a) on the JPEG distorted TBX11K dataset for different images resolutions. The reference curve was obtained by training the model on clean images and testing on distorted images. Images resolutions used are {224×224,128×128, 64×64, 32×32} in (a) – (b), respectively. Images were compressed using the standard luminance quantization table.



(a)

(b)

(c)

(d)

Figure 39. The DL model performance with the proposed augmentation method (setup 5a) on the JPEG distorted TBX11K dataset for different images resolutions. The reference curve was obtained by training the model on clean images and testing on distorted images. Images resolutions used are {224×224,128×128, 64×64, 32×32} in (a) – (b), respectively. Images were compressed using the standard chrominance quantization table.

Noise level analysis. For the grayscale image compression, either the luminance or chrominance tables specified in the JPEG standard can be used, as mentioned earlier. The luminance quantization table is designed to preserve more details and better quality of an image than the chrominance quantization table. In other words, compressing the images with different quantization tables results in different types of distortions. Since the proposed method takes advantage of the JPEG defects to make the model robust against the noise. Therefore, it is necessary to find the optimal noise level. For example, in the analysis presented in Figure 38, it can be seen that when the training set was augmented using the luminance table then there is not a significant gain in the accuracy difference for the smaller resolution images (Figure 38. (b)–(d)). To gain performance efficiency, we alternate the quantization tables used for data augmentation and compression of the test set. In the first case (Setup 5b-C_T in Figure 38. (a) – (d), the model was trained on images augmented using the chrominance table which was then evaluated on test set that was compressed with the luminance table. A significant improvement can be observed in the model performance particularly for the smaller resolution images. For the completeness purpose, we have also analyzed when the model was trained on augmented images generated from the luminance table and tested on the images compressed with the chrominance table (Setup 5b-L_T in Figure 39. (a)–(d)). However, no performance improvement was achieved. Based on these analyses, it can be proposed that to make DL robust against the JPEG distortions, it is necessary to use chrominance table during augmentation.

Recommendations. Given the importance of medical image analysis for disease diagnosis, our results presented in Figure 38 and 39. suggest that: 1) the model can generalize well (that is robust against the JPEG distortions), when trained with higher resolution images. 2) The proposed noise-based augmentation method can be exploited to further improve the model’s performance. 3) The chrominance quantization table can be used to achieve higher bitrate savings. 4) When the available images are of smaller resolution, then the DL model can highly benefit from the proposed method, mainly when the augmentation is done with very noisy images (that is, using the chrominance table for compression).

5.6. Chapter Summary

We extended the applications of block-based perceptual encryption (PE) methods to grayscale image processing in medical image analysis domain. Experimental results showed that the proposed scheme (IIB-CPE) is suitable to avail healthcare cloud services for medical image analysis. The compression was performed in lossy mode and distortion in recovered images has no effect on performance of the proposed deep learning (DL) model for tuberculosis (TB) screening in chest X-ray images. In addition, we proposed a new noise-based data augmentation method that takes advantage of distortion that resulted from the JPEG compression algorithm to improve robustness and generalization of the proposed DL model on smaller dataset.

VI. PERCEPTUAL ENCRYPTION-BASED PRIVACY-PRESERVING DEEP LEARNING APPLICATIONS

6.1. Privacy-Preserving Face Recognition Scheme

6.1.1. Motivation

This work presents privacy-preserving surveillance system for smart cities based on perceptual encryption algorithm (PE). The encryption is block-based and provides a necessary level of security while preserving intrinsic properties of an image necessary for compression. Unlike existing PE methods, the proposed method retains color information, thus can enable processing in encryption domain. The analysis shows that our method achieves the same compression performance as existing methods while providing better security. In addition, we have performed face recognition on the encrypted images and demonstrated that the proposed method delivers the same recognition accuracy as that of the plain images.

6.1.2. Introduction

Video surveillance is one of the main building blocks of smart cities, which provides safer communities and efficient city operations. However, the convenience comes at the cost of relinquishing personal data and privacy. Given the large volume of data, cloud-based storage is emerging as a cost effective and efficient solution. In addition, the data is often outsourced to third-party computational resource providers for performing several computer vision tasks such as action and activity recognition, people counting, age and gender estimation, fire and smoke detection and vehicle detection [140]. The data collected by the surveillance system often consists of privacy sensitive data that can be exploited to recognize an individual. Therefore, it is important to keep the citizens data secure while providing them with the facilities. When transmitting data over unprotected public channels, the traditional encryption algorithms can be used for the protection of multimedia data. However, when the goal is to enable other requirements like low computational complexity, format compliancy and processing in the encryption domain then the number theory-based encryption algorithms are not adequate. On the other hand, for privacy-preserving techniques like federating learning, differential privacy, and homomorphic encryption, there is a privacy and model accuracy tradeoff [141], [142]. Therefore, to solve these problems a new class of encryption techniques is emerging for protecting image data called perceptual encryption algorithms.

6.1.3. Related Work

The perceptual encryption (PE) algorithms have simple computational steps that protect the human perceivable information while retaining the intrinsic properties of images to enable several applications. The algorithms are block based and perform four steps: block permutation, block rotation and inversion, and pixel level negative and positive transformation. The main advantage of the methods is that the encrypted images are JPEG compressible and are referred to as encryption-then-compression (EtC) methods. The applications of EtC schemes have been extended to social networking services and cloud-based photo storage [37], [38], image retrieval systems [64] and for protecting medical images [10]. Several studies have improved the encryption as well as compression performance of the EtC schemes. For example, [65] proposed color image based EtC system (Color-EtC) with an additional step to permute the blocks in the color channels for improved encryption efficiency. However, the scheme has a limitation on the block size. The smallest block size that can be used is 16×16 in order to avoid block distortion in the recovered image. Therefore, [37] proposed to represent the input image as grayscale image by combining the color channels along the horizontal or vertical direction. Such representation allows using a smaller block size of 8×8 and can improve the encryption efficiency. However, the grayscale image based EtC does not consider JPEG color subsampling. An alternative grayscale image based EtC is proposed in [38], which can enable color subsampling by using YCbCr color space.

The grayscale EtC methods (GS-EtC) have improved encryption efficiency; however, the lack of color information limits their applications. In this paper, we proposed an efficient PE method that uses different keys for each color channel in rotation-inversion, and negative-positive transformation

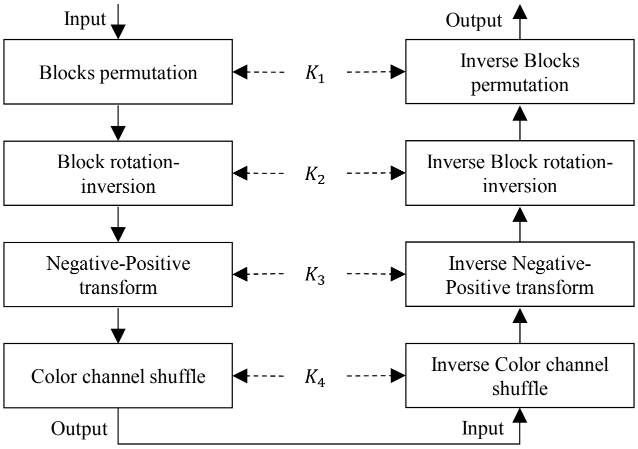

Input
Blocks permutation
Block rotation-inversion
Negative-Positive transform
Color channel shuffle
Output
K_1
K_2
K_3
K_4
Output
Inverse Blocks permutation
Inverse Block rotation-inversion
Inverse Negative-Positive transform
Inverse Color channel shuffle
Input

Figure 40. Proposed block-based perceptual encryption method.

steps. Thus, it improves encryption efficiency of the existing methods without compromising their compression savings. In addition, the presence of color information in the encrypted images makes them suitable for privacy-preserving machine learning (ML) tasks. Color is crucial for face recognition tasks and improves performance of an algorithm significantly [143]. As an application of the proposed method, we have implemented ML based face recognition for color images in the encryption domain as opposed to [144] which is only applicable to grayscale images. The main advantage of the proposed method is that privacy sensitive images do not require to be exposed to the third-party cloud owners for storage and/or computation. The proposed PE algorithm can be integrated as a component of the surveillance system used in smart cities.

6.1.4. Proposed Method

6.1.4.1. PE Method

The proposed method is a block-based perceptual encryption (PE) algorithm that makes an image difficult to recognize visually. Figure 40. shows a high-level illustration of the proposed method encryption and decryption processes. The proposed method consists of the following steps:

Step 1. Input image representation.

The proposed method represents an input image as a true color image in order to preserve color and spatial information of the image.

Step 2. Block-based transformations

- Divide an image with $\mathrm{W} \times \mathrm{H}$ pixels into blocks, each with $B_{\mathrm{w}} \times B_{\mathrm{h}}$ pixels, and permute the divided blocks by using a randomly generated secret key K_1. The same permutation key is used in each color channel, which is important to preserve the same spatial information in each color channel.
- Randomly rotate and invert each block by using a key K_2 where each entry represents a different combination of rotation and inversion. The key $K_2 \in \{K_2^R,\ K_2^G, K_2^B\}$ for the color channels red (R), green (G), and blue (B), where $K_2^R \neq K_2^G \neq K_2^B$.
- Randomly apply negative-positive transformation to each block by using a uniformly distributed key K_3 as in (1). The key $K_3 \in \{K_3^R,\ K_3^G, K_3^B\}$ for the color channels red (R), green (G), and blue (B), where $K_3^R \neq K_3^G \neq K_3^B$.
- Shuffle the blocks in the three channels randomly by key K_4 where the key elements represent a permutation of the channels.

Step 3. The encrypted image obtained in the last step can be compressed by the JPEG standard in the RGB or YCbCr mode. The original image can be recovered by performing the above steps in reverse order with the same keys.

The conventional color PE method [65] uses the same key for each color channel in the second and third steps while the proposed method uses different keys for each color channel in the same steps, and thus improves the encryption efficiency. On the other hand, the conventional grayscale PE method improves the security efficiency of the algorithm. However, lack of color information and disoriented spatial information limit their applications. Since the proposed method represents an input image in color and preserves almost the same spatial information in each color component; therefore, overcomes limitations of exiting PE methods.

6.1.4.2. ML-based Face Recognition

Support vector machines (SVMs) [145] are a type of supervised machine learning (ML) algorithms for discriminative classification, which finds a line in two dimension or manifold in multiple dimension data to separate classes from each other. In general, there exist several separators to differentiate between the classes, which makes it difficult to choose the best fit. Instead of making a zero-width line as a decision boundary, we can draw a margin of some width on both sides of each line up to the nearest data point. SVMs choose a line that maximizes the margin, as an optimal model. The points that touch the margin are called support vectors. When fitting the model, loss function is computed based on the support vectors and any points beyond the margins do not modify the fit. In addition, for faster computation, instead of fitting the model on the original images, they can be treated as a vector in a high-dimensional space to derive a lower dimensional representation. One example of the method is called principal component analysis (PCA). The PCA is an unsupervised algorithm that describes a dataset by finding a list of principal components, which are strictly eigenvectors, and are often called eigenfaces [146] when used for face recognition.

6.1.5. Privacy-Preserving Face Recognition Analysis

For privacy-preserving face recognition task, we have implemented support vector machines (SVMs) as discussed in Section 6.1.4.2. In the experiment, we chose people with at least 70 images from the LFW dataset. As a result, we have 7 different classes and 1,288 images in total. The images were resized to 120×88 pixels in order to avoid padding required to fit the block size (i.e., 8) of the encryption algorithm. For training, 75% of the images were used. In addition, for the dimensionality reduction, we have used randomized principal component analysis (PCA) instead of standard PCA for its faster computation. When using PCA for dimensionality reduction, only the largest principal components that represent the maximal data variance are preserved and the rest are zeroed out. The number of components needed for describing the data can be determined by the cumulative explained variance ratio as a function of the principal components number as shown in Figure 41. shows some example principal components (also called eigenfaces) for the plain dataset, EtC encrypted dataset and PE encrypted dataset. The first few eigenfaces show the angle of lighting on the face and the latter corresponds to more details of the face. In experiments, principal components $\boldsymbol{N} = (500{,}250{,}150{,}100)$ have been used for face recognition, which accounts for 99%, 96%, 92% and 89% of the variance, respectively. Figure 42. gives classification accuracy on the test dataset and the training time required for each value of $\boldsymbol{N}$. The training time increases as the number of components increases. The accuracy of SVM for face recognition on plain and encrypted images remains the same with a negligible difference. The best accuracy is achieved with only using 92% variance with acceptable time. To get a better understanding of the trained estimator for $\boldsymbol{N} = 150$ components, Figure 43. shows confusion matrices, which gives the labels that are likely to be missed by the estimator.

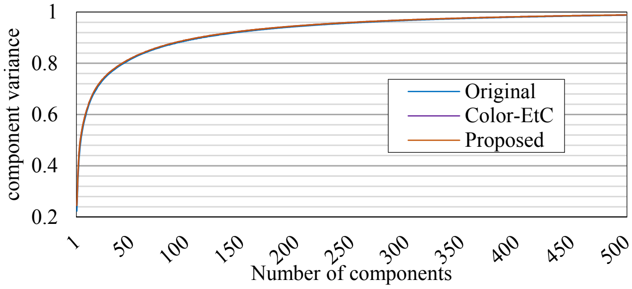


Figure 41. Number of components required to represent the variance of LFW dataset.

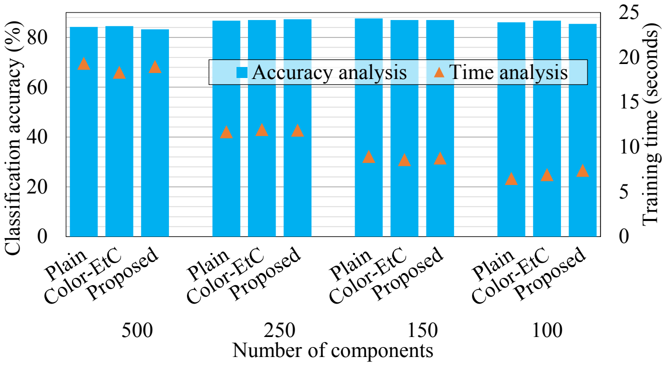
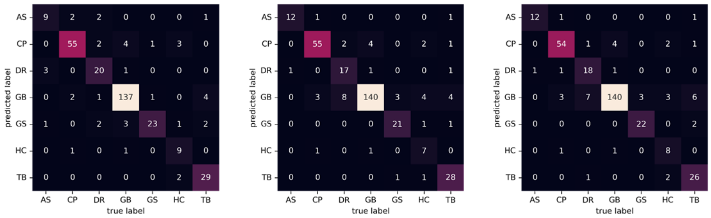


Figure 42. Face recognition accuracy for different methods by varying the number of eigenfaces.



(a) (b) (c)

Figure 43. Confusion matrices of SVM predictions on plain images (a), EtC encrypted images (b) and proposed encrypted images (c). The labels are the name initials where AS: Ariel Sharon; CP: Colin Powel; DR: Donald Rumsfeld; GB: George W. Bush; GS: Gerhard Schroeder; HC: Hugo Chavez; TB: Tony Blair.

6.1.6. Summary

In this work, we proposed block-based perceptual encryption algorithm for secure image data transmission and storage. The encryption is carried out in such a way that the cipher image retains intrinsic properties of the original image. Thereby, can enable computation in the encryption domain. The main advantage of the proposed method is that it retains color information, which makes it suitable for privacy-preserving machine learning (ML). As an application, we have implemented face recognition for privacy-preserving surveillance that can be used in smart cities. The analysis shows that the encryption has no effect on the algorithm accuracy.

6.2. Privacy-Preserving Image Classification

In contrast to the centralized paradigm, a decentralized system (e.g., *Federated learning (FL)* [147] *where algorithm is distributed instead of data gathering*) relying on the principle of remote executions and distributed data storage provides an infrastructural approach to security and confidentiality [148]. However, a decentralized system does not fully guarantee privacy. For example, a lack of encryption puts the data and algorithm parameters at risk of being stolen or tampered with, and reconstruction of data from the model weights is also possible [141]. In addition, a limited computational capacity or a small amount of data at a node may affect the quality of the results [141].

In this regard, data manipulation techniques, such as differential privacy (DP) [149] and secure aggregation techniques such as homomorphic encryption (HE) [150], can provide security in FL. However, DP can degrade the data and may reduce the accuracy of the model, especially in domains with limited data, for example, medical imaging [141]. The main challenge in implementing HE is the computational cost and requirement of a specifically designed algorithm to enable computation in the encryption domain [141]. Similarly, secure multiparty computation (SMPC) [151] has the disadvantage of communication overhead.

As an application of the IIB–CPE scheme, we propose an end–to–end image communication system for privacy–preserving deep learning–based image classification. The proposed system considers a secure and efficient transmission of images to a remote location, thereby *end–to–end,* and enables classification in the encryption domain without the need for decryption, thereby *privacy–preserving deep learning.* For this purpose we performed two sets of experiments. First, we analyzed the suitability of the proposed extensions of block-based PE methods for PPDL applications. Next, we chose the best performing method from these extensions and compare it with the existing methods.

For privacy-preserving classification analysis, we have implemented the PyramidNet model proposed in [152]. The model was 110 layers in depth with a widening factor of α=270. The ShakeDrop regularization [153] was utilized for better performance. The model was trained for 200 epochs using Stochastic Gradient Descent (SGD) with the Nesterov accelerated gradient and momentum method. The momentum of 0.9, weight decay of 0.0001 and batch size of 512 were used during training. The initial learning rate was set to 0.1, which was then decayed by a factor of 0.1 at 75, 110 and 150 epochs. The dataset used was Cifar10 dataset [138], which consists of 50K and 10K training and test images. The images were uniformly distributed among 10 classes. During training, random crop and flip were used as augmentation methods. Table 17 summarizes the test accuracy of the model on the PE cipher images. The model has achieved better accuracy on BPE and EBPE–5 cipher images as they preserves image local contents on a larger block size. The accuracy drop is only 3.5% as compared to the model tested on plain images. Among the other extended BPE methods,

Table 17. Privacy-preserving image classification analysis of the DL model in terms of accuracy (%).

Methods		Test
Baseline	Original	96.30
	BPE [61]	92.74
Proposed	EPBE–1	89.04
	EPBE–2	89.33
	EPBE–3	90.16
	EPBE–4	77.85
	EPBE–5	92.49

EBPE 1–3 have a negligible difference in their performance. However, they resulted in additional 2.5% drop in the test accuracy. The most secure EBPE–4 has drastically reduced accuracy of the model. Based on this analysis, we chose EBPE–5 (or IIB–CPE) as our PE scheme for PPDL-based classification and compare it with the existing PE methods.

We used the same PyramidNet [152] model of 110 layers in depth and a widening factor of α = 270 with ShakeDrop regularization [153] as in the previous experiments, however, with different training parameters. The model was trained for 300 epochs using Stochastic Gradient Descent (GSD) with the Nesterov accelerated gradient and momentum method. The momentum of 0.9, weight decay of 0.0001 and batch size of 128 were used during training. The initial learning rates were set to 0.1 for CIFAR10 and 0.5 for CIFAR100 datasets, which were then decayed by a factor of 0.1 at 150 and 225 epochs. Figure 44 summarizes the classification accuracy of PPDL models. The learnable PE algorithms LE and PBE have the closest accuracy to plain images; however, they are vulnerable to chosen–plaintext attacks, as demonstrated by [154]. ELE, the most secure PE algorithm, degrades classification accuracy. Among the learnable PE methods, chaos–based SPBE has preserved the classification accuracy of plain images while providing a necessary level of security. However, like any other learnable PE method, they do not provide compression savings. In contrast, for compressible PE algorithms, Color–EtC (16 × 16) has the highest accuracy, but its smaller keyspace and larger block size may compromise system security. However, its variant Color–EtC (4 × 4) and conventional GS–EtC (8 × 8) have improved security, but their classification accuracy is reduced, along with compression savings. For the proposed method, when the sub–block size is 8×8, a comparable classification accuracy can be achieved while preserving the compression savings and improving the security of the conventional EtC schemes. Furthermore, for smaller block sizes, there was a slight reduction in the accuracy of the DL model, but improved security strength.

In Figure 44, the same model and datasets were used for all of the PE methods for fair comparison. In the literature, Color–EtC cipher images were used with two isotropic networks to enable privacy–preserving classification. In their analysis, best accuracy of 87.89% and 92.76% were achieved on the CIFAR 10 dataset for vision transformer (ViT–16) [155] and ConvMixer [156], respectively. For the same encryption method (Color–EtC (16 × 16)), our PPDL model has achieved almost 6% and 1% better accuracy than ViT and ConvMixer, respectively. However, the main advantage of using ConvMixer is the smaller number of parameters than the proposed method. On the other hand, when the images were encrypted with the proposed IIB–CPE method, then the achieved accuracy is up to 5% better than ViT model and almost 2% less than ConvMixer across different smaller block sizes. As mentioned earlier that the use of larger block size makes the Color–EtC method vulnerable to the COA attack. Therefore, the IIB–CPE based PPDL has a main advantage of better security.

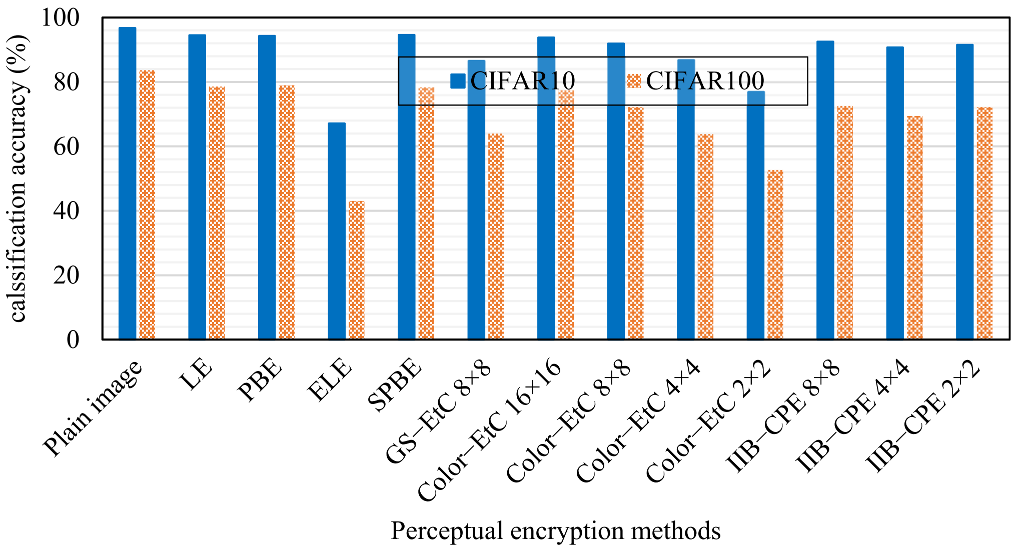


Figure 44. Classification accuracy (%) of the deep learning model in the encryption domain on CIFAR10 and CIFAR100 datasets.

6.3. Privacy-Preserving COVID-19 Detection in Chest X-Ray Images

6.3.1. Motivation

The widespread adoption of deep learning (DL) solutions in the healthcare organizations is obstructed by their compute intensive nature and dependability on massive datasets. In this regard, cloud–services such as cloud storage and computational resources are emerging as an effective solution. However, when the image data are outsourced to avail such services, there is a privacy concern that the data should be kept protected not only during transmission but during computations as well. To meet these requirements, this study proposed a privacy-preserving DL (PPDL) scheme based on IIB-CPE that can enable computations without the need of decryption. The encryption is based on perceptual encryption (PE) that only hides the perceivable information in an image while preserves other characteristics that are necessary for DL computations. Precisely, we have implemented a binary classifier based on EfficientNetV2 for the COVID-19 screening in the chest X-ray (CXR) images. For the PE algorithm, the suitability of two pixel-based and two block-based PE methods was analyzed.

6.3.2. Introduction

The automatic computer aided medical diagnosis (CAD) systems have emerged in the healthcare sector to assist clinicians and doctors by speeding up a disease diagnosis with improved accuracy. This progress is primarily due to the success of Machine Learning (ML) in different fields of the computer vision [7]. Particularly, a ML subfield called Deep Learning, has achieved the state-of-the-art accuracy for the classification of images [78]. However, there are two main challenges when implementing DL based solutions. First, DL algorithms are characterized as compute-intensive tasks, and training them requires high computational resources. Second, training models for a particular task requires a large amount of sample data, which in some domains, for example, in the field of medical image analysis, is expensive and difficult to acquire. In the first case, a healthcare organization can access the latest technology by availing third-party provided cloud-computing resources to speed-up models training and allow them to scale in a cost effective way [7], [79]. To overcome the data deficiency issue, an organization can take advantage of a community cloud, where services are shared to achieve a common goal. For example, for joint projects and collaborations among the organizations, cloud-storage services can be utilized as a centralized data repository shared among them [7]. Nonetheless, when the data are exchanged for cloud services, there is a security concern [1]. One solution is to encrypt the data before transmission. The full encryption algorithms based on the number theory and chaos theory are the most secure choices for protecting image data. However, decryption of the cipher images is required in order to realize computer vision and image processing applications. This data reveal may be allowed in certain cases; however, for privacy-sensitive data such as medical images, surveillance data, etc., such encryption methods become insufficient [80].

6.3.3. Related Work

In recent years, the perceptual encryption (PE) schemes have been proposed that encrypt images in such a manner that they are visually unidentifiable, but their characteristics remain intact. The PE algorithms can be categorized as pixel-based [157]–[160] and block-based encryption algorithms [7], [10], [35]. The pixel-based methods obfuscate an image on a pixel level; therefore, protects local information while preserve global information of the image. A pixel-based PE algorithm [158], known as Learnable Encryption (LE), successfully preserved the classification accuracy of a DL-based classifier. However, LE is susceptible to chosen–plaintext attack (CPA) [154]. Their security has been enhanced by the extended learnable encryption (ELE) scheme presented in [159]; however, they significantly reduced the classification accuracy. An alternative algorithm presented in [160] has better security (that is larger key size) than LE while preserving the DL model accuracy. However, it cannot resist CPA as shown in [154]. To overcome their security vulnerabilities, [157] proposed to use a random values' sequence ranging from 0 – 255 for the xor operation. On the other hand, block-based algorithms perform encryption on a block level in such a way that the global contents of the image are protected while local information remains intact. The authors in [35] proposed a block-based PE method for color image encryption. Their algorithm consists of geometric and color transformations steps, which are computationally inexpensive and provide a necessary level of security. However, there is an efficiency tradeoff between compression savings and encryption strength because of the choice of block size. For example, larger block size is used for better compression savings, which may raise some security vulnerabilities [10]. Therefore, the authors in [6], [7] have proposed sub-block processing to overcome these security issues without compromising the compression savings.

Different from full image encryption, PE only hides an image perceptual information while preserving its intrinsic properties, which can be interpreted by machines. These methods were originally designed to provide bandwidth/storage efficiency and security during image transmission and/or storage. In this study, we have extended their applications to medical image analysis based on PPDL. Consequently, the proposed method provides security during image data transmission, storage and computations. For this purpose, we have proposed a DL-based binary classifier for the COVID-19 detection in CXR images.

6.3.4. Setup and System Model

In the simulations, PBE [160], SPBE [157], KBPE [35], and EBPE [7] methods were implemented to encrypt the medical images. Since, CXR images are grayscale with one channel; therefore, in the first three methods the color channel shuffling step was omitted. For the block-based methods, block level processing was carried out on a fixed block size of 8×8 while for the sub-block processing (in the case of EBPE) block sizes 4×4 and 2×2 were used. For the PPDL classification analysis, publicly available CXR dataset [161] and for the encryption analysis UCID dataset available on [162] were used.

Figure 45. Illustrates a schematic of cloud-based medical images analysis and storage. The PE scheme is used to encrypt the images, which are then transferred over a public channel to avail third party own cloud-services. For later usage, these cipher images can be archived, which can be

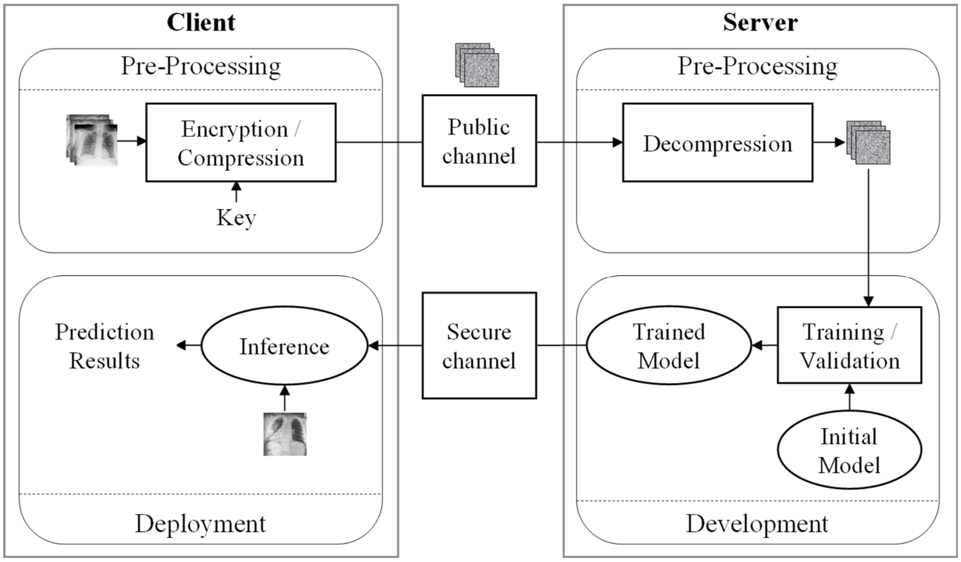


Figure 45. A basic architecture to realize PPDL for medical image analysis by availing third-party owned cloud services.

accessed for collaboration among authorized experts. Here, we consider a model that the cloud-services providers are either curious or untrustworthy; therefore, the secret key information is hidden from them. A DL algorithm is implemented in the cloud for diagnosis of diseases. The final diagnosis report is sent back to the hospital for treatment. The subsections analyzed different PE methods for the protection of medical images and PPDL performance on them.

6.3.5. PPDL-based Classification Analysis

Dataset. The dataset used in this study consists of CXR images and is publicly available on [161]. In the experiments, we have used 4,626 images uniformly distributed between two classes: healthy and COVID-19. The dataset was split into 80% for training and 20% for validation and testing, equally divided. The images were preprocessed and resized in two steps: (1) all black borders on the images edges were removed and (2) they were resized from the center to meet the input requirements of the classification model that is, 224×224. Note that the preprocessing steps were carried out on the client side for better utilization of the available bandwidth. In addition, all the images were encrypted with the encryption methods discussed in Chapter 5. Figure 46. shows

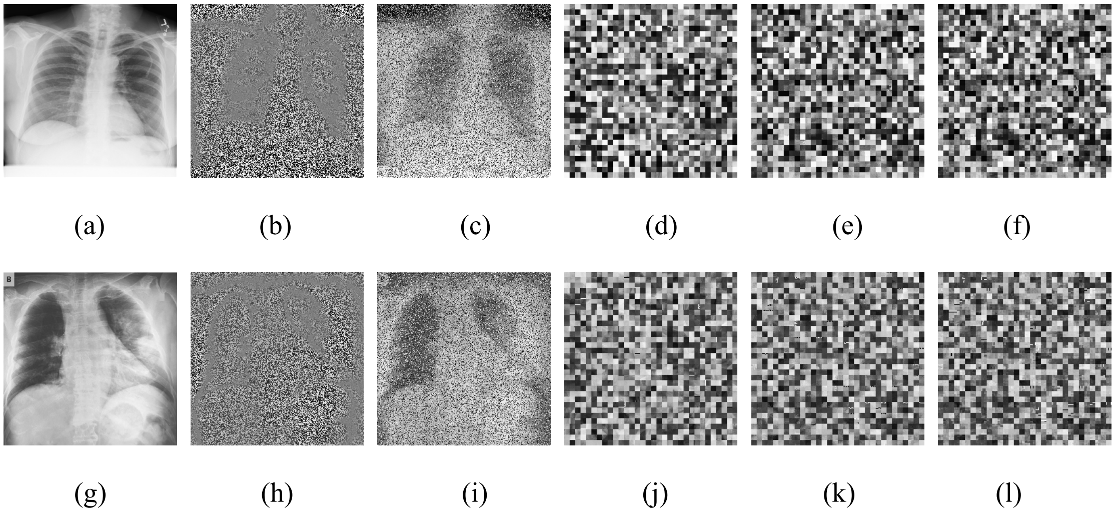


Figure 46. Example CXR images shown here are from the COVID-19 dataset as input to the DL model. Healthy sample (top row) and Covid-19 sample (bottom row). (a) and (g) are plain images. Their corresponding cipher images are in (b) – (f) and (h) – (l) obtained from PBE, SPBE, KBPE, EBPE2×2, and EBPE4×4 encryption methods, respectively.

example images from the dataset for both labels along with their cipher images.

Proposed Model. In the present study, a classification model belongs to EfficientNet DL models family was used. The model choice was based on their better accuracy for natural images while maintaining low computational cost as compared to other conventional CNN architectures. Based on this efficiency, these models have been used in medical analysis [7]. EfficientNet (EfficientNetV1) [96] is a family of lightweight CNN models, which are optimized for parameter and FLOPs performance. It takes advantage of neural architecture search (NAS) to design a baseline EfficientNet-B0 that has better trade-off on FLOPs and accuracy. To obtain a family of models B1-B7, this network is then uniformly scaled-up (width, depth and resolution) with a simplified yet an effective compound scaling strategy. The models have superiority over existing CNN models in terms of number parameters and FLOPs as they use depthwise convolutions. However, such convolutions often cannot utilize modern accelerators fully; therefore, EfficientNetV1 have main a limitation in terms of training or inference speed (for example, compare to ResNet-RS-420) [108].

Table 18. Architecture of the proposed model for COVID-19 detection in CXR images. The EfficientNetV2-B0 is used as a feature extractor. (Param #: Number of parameters)

Layer name (type)	Output shape	Param #
EfficeintNetV2-B0 (Model)	7×7×1280	5,919,312
global_average_pooling_2d	1280	0
dropout_1 (Dropout)	1280	0
dense_1 (Dense)	1	1281
Total parameters: 5,920,593		
Trainable parameters: 5,859,985		
Non-trainable parameters: 60,608		

Therefore, EfficientNetV2 [108] improves training speed of EfficientNetV1 models while maintaining the parameter efficiency. Specifically, EfficientNetV2 provides three solutions to EfficientNetV1 training bottleneck. 1) For better training speed, EfficientNetV2 proposed to progressively adjust size of an image and regularization during training. 2) EfficientNetV2 proposed a scaling strategy that is non-uniform and gradually add more layers to later stages as opposed to EfficientNetV1 where a simple compound scaling rule scales up all stages equally. 3) To fully utilize modern accelerators, EfficientNetV2 proposed that FusedMBConv in early stage can improve training speed with a small overhead on FLOPs and parameters efficiency. The combination of depthwise Conv3×3 and expansion Conv1×1 in MBConv is replaced with a single regular Conv3×3 layer in Fused-MBConv. Table 18 summarizes architecture of the proposed binary classifier.

Analysis. For the PPDL based COVID-19 detection analysis, the proposed model (summarized in Table 18) was trained using Stochastic Gradient Descent (SGD) with batch size 16 for 120 epochs. The initial learning rate was set to 0.1, which was then reduced by a factor of 10 when validation accuracy stopped improving. In addition, we used early stopping criteria once the model validation accuracy stopped improving for 60 epochs. During training, random flip, rotation, zoom, translation and contrast were used as data augmentation methods. The proposed model was implemented for binary classification and can either identify an observation to be in positive or negative class. In the experiments, we considered positive classes to be the COVID-19 samples and negative classes to be the healthy samples. Figure 47. illustrates the model's training and validation

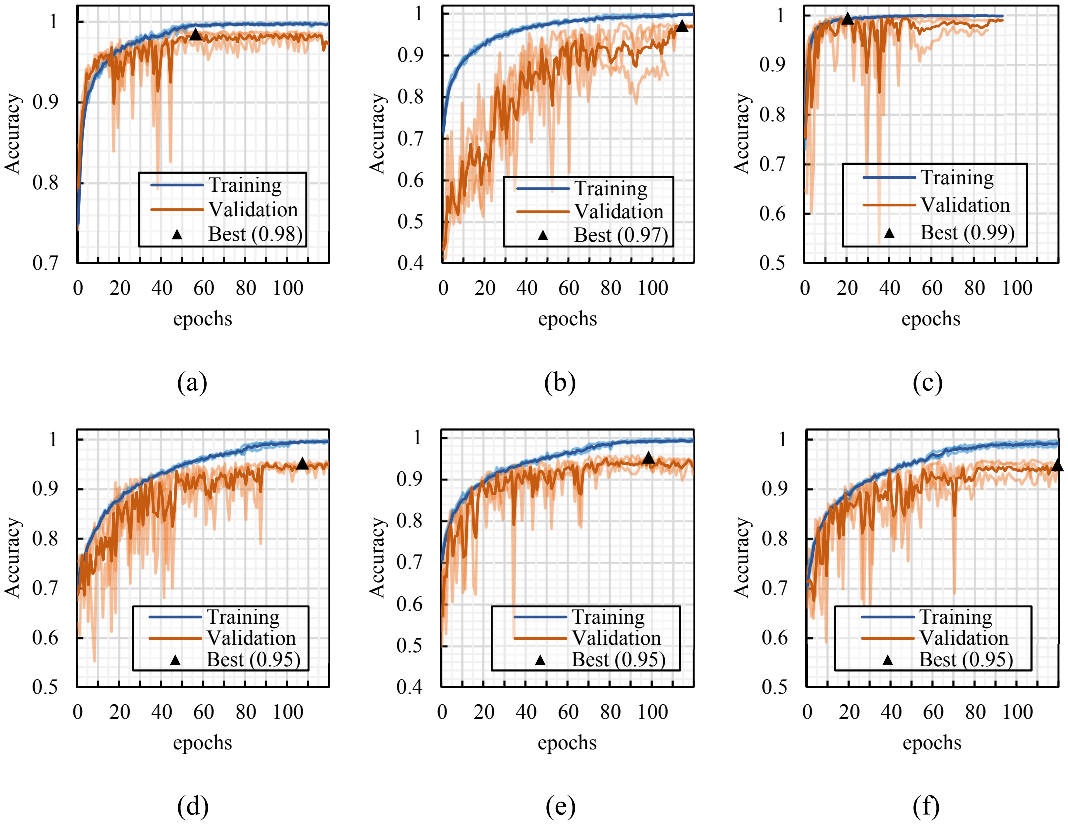


Figure 47. Training and validation accuracy of the PPDL-based COVID detection in CXR images. (a) is performance of the model on plain images while (b) – (f) are privacy-preserving classification performance on the encrypted images obtained from PBE, SPBE, KBPE, EBPE2×2, and EBPE4×4, respectively. The accuracies were obtained as a mean of 3 runs shown in dark color while each run accuracy is shown in a light color.

accuracy on plain and cipher images obtained from different encryption methods. Table 19 summarizes the detailed performance of the model in terms of accuracy, sensitivity specificity and AUC of receiver operating characteristic curve (ROC) measures. The values were obtained on the test set as the mean and standard deviation of 3 runs. The proposed model has achieved state-of-the-art performance for COVID-19 detection in the plain CXR images. For the PPDL classification, the model has achieved better results on the pixel-based encryption methods (PBE and SPBE) than the block-based encryption methods (KBPE and EBPE). However, when compression of the data is required then block-based PE methods are suitable for PPDL. Such cipher images are JPEG compressible as demonstrated in [6], [7], [35].

Table 19. Performance analysis of the proposed model using different evaluation metrics. The measures values are reported as the mean (black color) and standard deviation (gray color) for 3 ru ns. (AUC: area under the ROC curve).

Methods	Accuracy	Specificity	Sensitivity	AUC
Plain	98.78 ± 0.31	0.98 ± 0.005	0.99 ± 0.005	1.00 ± 0.000
PBE [156]	96.43 ± 1.68	0.98 ± 0.005	0.95 ± 0.031	1.00 ± 0.005
SPBE [153]	99.82 ± 0.05	1.00 ± 0.000	1.00 ± 0.000	1.00 ± 0.000
KBPE [31]	95.56 ± 0.14	0.96 ± 0.005	0.95 ± 0.005	0.99 ± 0.005
IIB-CPE 2×2 [6]	95.46 ± 0.32	0.95 ± 0.008	0.96 ± 0.005	0.99 ± 0.005
IIB-CPE 4×4	95.85 ± 0.05	0.96 ± 0.008	0.96 ± 0.009	0.99 ± 0.000

6.3.6. Summary

Cloud-services provide a cost effective solution to meet the information and communications technology (ICT) needs of an organization. To avail such services, the privacy sensitive data should be protected during transmission and computations. We proposed a PE-based PPDL scheme that satisfied these requirements. The analysis have shown that it is necessary to preserve global contents instead of local contents in the images for better classification performance.

6.4. Chapter Summary

In this chapter we implemented several applications of the proposed method that require privacy preservation. The PE methods that can enable PPDL applications can be classified as learnable PE methods and Compressible PE (CPE) methods. A learnable encryption (LE) method is proposed [158] that forms a 6-channel image from an RGB image by splitting predefined blocks into upper and lower 4-bit images. Then, the encryption is achieved by randomly changing pixels intensities and shuffle their positions. The method has been successfully applied to PPDL; however, it is vulnerable to chosen plaintext attack demonstrated in [154]. To deal with the security vulnerability of [158], an extended version of LE method (ELE) is proposed in [159], which uses a different key in each channel. However, it has severely degraded the DL model's performance. A pixel-based PE (PBE) method is proposed [160] that randomly changes pixel intensities and shuffle the color components in an image. The method is applicable to PPDL; however, it is vulnerable to chosen plaintext attack, as demonstrated in [154]. The main reason for the scheme vulnerability is the use of a single value subtraction from the pixel intensities. Therefore, the authors of [157] proposed to xor a random sequence generated by chaotic map with the pixel values for better security without compromising the DL performance (SPBE). Alternatively, [163] proposed to divide the encrypted image of [160] into blocks and apply three different types of filters on randomly selected blocks (FPBE).

On the other hand, the CPE schemes have an important property as pointed out in [144] that under its different transformation functions, both the Euclidean distance and inner product of two vectors are preserved. Therefore, allows the computation of machine learning algorithms in the encryption domain. For privacy-preserving face recognition task, the CPE is combined with Support Vector Algorithm (SVM) in [144] and the Extended-CPE is combined with SVM in [16]. Besides face recognition tasks, CPE-based PPDL has been considered in [6], [9], [164] for natural image classification. Specifically, IIB-CPE was combined with a convolutional neural network (such as the EfficientNetV2-B0) in [6], and CPE was combined with an isotropic network (such as vision transformers) in [164]. In [9], the authors implemented four different extensions of IIB–CPE and analyzed their effect on a CNN model's accuracy. Similarly, the applications of PE schemes are extended to medical image analysis domain. For example, [17] implements a CNN-based model with IIB–CPE scheme for COVID-19 diagnosis in chest X-ray images. A comparison between different CPE methods for PPDL applications is given in Table 20.

Table 20. Comparison of different PE-based PPDL schemes for various applications. The accuracy difference is between a DL model performance on plain and PE cipher images.

Class of PE	Method Name	PPDL Task	Model	Dataset	Difference in accuracy (%)
Learnable PE	LE [158]	Natural image classification	PyramidNet	CIFAR10	-2.21
				CIFAR100	-5.07
	ELE [159]		PyramidNet	CIFAR10	-29.6
				CIFAR100	-40.54
	PBE [160]		PyramidNet	CIFAR10	-2.43
				CIFAR100	-4.63
	SPBE [157]		PyramidNet	CIFAR10	-2.12
				CIFAR100	-5.2
	PBE [160] for PPDL in [17]	Medical Image Analysis	EfficientNetV2-B0	COVID19 CXR	-2.35
	SPBE [23] for PPDL in [17]		EfficientNetV2-B0	COVID19 CXR	1.04
	PBE [160] for PPDL in [163]		DenseNet	MRI Brain tumor 3 datasets	-5.0
	PBE [160] for PPDL in [163]		XceptionNet	MRI Brain tumor 3 datasets	-6.94
	FPBE [163]		DenseNet	MRI Brain tumor 3 datasets	-9.06
	FPBE [163]		XceptionNet	MRI Brain tumor 3 datasets	-9.34
	PBE [160] for PPDL in [163]		DenseNet	COVID19 CXR	-3.35
	PBE [160] for PPDL in [163]		XceptionNet	COVID19 CXR	-3.73
	FPBE [163]		DenseNet	COVID19 CXR	-1.54
	FPBE [163]		XceptionNet	COVID19 CXR	-2.58
Compressible PE	CPE [65] for PPDL in [144]	Face Recognition	SVM with PCA	Labeled Faces in the Wild	-0.62
	Extended CPE [16] (This thesis)		SVM with PCA	Labeled Faces in the Wild	-0.62
	CPE [65] for PPDL in [164]	Natural Image Classification	ViT-B_16	CIFAR10	-11.17
	CPE [65] for PPDL in [164]		Conv-Mixer256/8	CIFAR10	-3.35
	CPE [65] for PPDL in [6]		PyramidNet	CIFAR10	-6.07
				CIFAR100	-11.39
	IIB-CPE [6] (This thesis)		PyramidNet	CIFAR10	-4.21
				CIFAR100	-11.05
	IIB-CPE [6] for PPDL in [17] (This thesis)	Medical Image Analysis	EfficientNetV2-B0	COVID19 CXR	-3.32
	CPE [65] for PPDL in [17] (This thesis)		EfficientNetV2-B0	COVID19 CXR	-3.22

VII. CONCLUSION

6.1. Conclusion

For an efficient DL model, large volume of sample data and high computation resources are needed. These requirements can be fulfilled by taking advantage of powerful infrastructures such as cloud-computing services, to avail high-powered computational resources, and cloud-storage services, for adopting collaborative learning. However, this comes with security and privacy concerns as there are potential risks of leakage of privacy-sensitive information associated with outsourcing the data. The existing privacy-preserving schemes have their associated computational cost, communication overhead and specialized design requirement that may reduce data utility and degrade the DL model performance. In addition, given the large volume of data, bandwidth and storage efficiencies should also be considered. Traditional privacy preservation approaches treat the requirements of data transmission and computation separately even though both are necessary to be fulfilled to fully reap the benefits of DL for data-driven applications. Therefore, in this thesis we propose an end-to-end framework to satisfy the dual requirements (compression and encryption) of a communication system while preserving user's privacy in downstream applications. Our proposed privacy scheme is based on perceptual encryption algorithm and is compatible with the widely used image compression standard such as the JPEG format. Importantly, the proposed system supports lossless DL model construction which does not modify any of the computation of the original model training algorithm. Therefore, it can be used with the existing state-of-the-art DL models without any modification. We have presented applications of the proposed privacy scheme in three different domains. For natural images classification task, our proposed PE-based privacy preserving scheme at best introduces a decrease of ≈5% in the prediction accuracy of the trained models. For face recognition application, the proposed privacy preserving scheme delivers the same recognition accuracy as that of the plain images. For COVID-19 screening in CXR images, the proposed PE-based privacy preserving scheme at best introduces a ≈3% drop in the model's accuracy and sensitivity scores.

6.2. Future Work

Training an effective DL model requires high computational resources and massive datasets. Leveraging powerful infrastructures such as cloud computational resources and adopting collaborative learning for training DL models can mitigate these issues. This thesis focused on meeting the computational requirements of DL in individual training setup. The proposed PE-based PPDL technique can be adopted in a collaborative learning environment. Based on the location of data collaborative learning has two types: direct collaborative learning and indirect collaborative learning. In direct collaborative learning, local data from multiple participants is directly uploaded to the central server and the model is trained on the server – centralized training. Transmitting large volumes of data demands high bandwidth and sharing privacy sensitive data results in users' privacy concerns. The proposed PE-based PPDL can jointly mitigate both challenges of direct collaborative learning. On the other hand, in indirect collaborative learning, instead of sharing data, copies of a model are trained locally on the data available with each participant and local model updates are shared with the centralized serve for aggregation – distributed or decentralized training. Indirect collaborative learning (e.g., Federated Learning) provides an infrastructural approach to security and confidentiality but does not fully guarantee privacy because the lack of encryption puts the data and algorithm parameters at risk of being stolen or tampered with and reconstruction of data from the model weights is also possible. The proposed PE-based PPDL can be used to protect the training data to ensure security in indirect collaborative learning. Integrating the proposed privacy scheme to ensure security in the collaborative learning paradigm (direct and indirect collaborative learning) could be an interesting future research direction.

REFERENCES

[1] I. Ahmad and S. Shin, “A novel hybrid image encryption–compression scheme by combining chaos theory and number theory,” *Signal Processing: Image Communication*, vol. 98, p. 116418, Oct. 2021, doi: 10.1016/j.image.2021.116418.

[2] I. Ahmad and S. Shin, “Analysis of Chinese Remainder Theorem Moduli for Image Compression,” presented at the KICS Fall Conference, Kookmin University, Seoul, Korea, Nov. 2019, pp. 108–109.

[3] I. Ahmad, B. Lee, and S. Shin, “Analysis of Chinese Remainder Theorem for Data Compression,” in *2020 International Conference on Information Networking (ICOIN)*, Barcelona, Spain: IEEE, Jan. 2020, pp. 634–636. doi: 10.1109/ICOIN48656.2020.9016442.

[4] I. Ahmad and S. Shin, “Performance analysis of Chinese Remainder Theorem for Data Compression,” in *Korea Computing Conference (KCC), KIISE*, Virtual, Jul. 2020.

[5] I. Ahmad, W. Choi, and S. Shin, “Comprehensive Analysis of Compressible Perceptual Encryption Methods—Compression and Encryption Perspectives,” *Sensors*, vol. 23, no. 8, p. 4057, Apr. 2023, doi: 10.3390/s23084057.

[6] I. Ahmad and S. Shin, “IIB–CPE: Inter and Intra Block Processing-Based Compressible Perceptual Encryption Method for Privacy-Preserving Deep Learning,” *Sensors*, vol. 22, no. 20, p. 8074, Oct. 2022, doi: 10.3390/s22208074.

[7] I. Ahmad and S. Shin, “A Perceptual Encryption-Based Image Communication System for Deep Learning-Based Tuberculosis Diagnosis Using Healthcare Cloud Services,” *Electronics*, vol. 11, no. 16, p. 2514, Aug. 2022, doi: 10.3390/electronics11162514.

[8] I. Ahmad and S. Shin, “Effect of Inter and Intra Block-level Shuffling on the JPEG Compression Performance,” in *2021 Summer workshop on computer communications (SWCC)*, Online, Aug. 2021.

[9] I. Ahmad and S. Shin, “Perceptual Encryption-based Privacy-Preserving Deep Learning in Internet of Things Applications,” in *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea, Republic of: IEEE, Oct. 2022, pp. 1817–1822. doi: 10.1109/ICTC55196.2022.9952589.

[10] I. Ahmad and S. Shin, “Encryption-then-Compression System for Cloud-based Medical Image Services,” in *2022 International Conference on Information Networking (ICOIN)*, Jeju-si, Korea, Republic of: IEEE, Jan. 2022, pp. 30–33. doi: 10.1109/ICOIN53446.2022.9687214.

[11] I. Ahmad and S. Shin, “Leveraging Transfer Learning in EfficientNetv2-based Tuberculosis Detection,” in *Fall Conference, KICS*, Geyeongju, Korea, Nov. 2022.

[12] I. Ahmad and S. Shin, “A Comparison of EfficientNets for Tuberculosis Detection in Chest Radiographs,” The 3rd Korea Artificial Intelligence Conference, KICS, Sep. 2022.

[13] I. Ahmad and S. Shin, “Noise-cuts-Noise Approach for Mitigating the JPEG Distortions in Deep Learning,” in *2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Bali, Indonesia: IEEE, Feb. 2023, pp. 221–226. doi: 10.1109/ICAIIC57133.2023.10067012.

[14] I. Ahmad and S. Shin, “Quantitative Assessment of the Impact of Lossy JPEG Compression on Deep Learning Models,” in *The 8th International Conference on Next Generation Computing (ICNGC), KINGPC*, Jeju, Korea, Oct. 2022.

[15] I. Ahmad and S. Shin, “Block-based Perceptual Encryption Algorithm with Improved Color Components Scrambling,” in *Korean institute of next generation computing*, Jeju Island, Korea (South), May 2022, pp. 155–158. [Online]. Available:

https://www.earticle.net/Article/A412335

[16] I. Ahmad, E. Kim, S.-S. Hwang, and S. Shin, "Privacy-Preserving Surveillance for Smart Cities," in *2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN)*, Barcelona, Spain: IEEE, Jul. 2022, pp. 301–306. doi: 10.1109/ICUFN55119.2022.9829680.

[17] I. Ahmad and S. Shin, "Perceptual Encryption-based Privacy-Preserving Deep Learning for Medical Image Analysis," in *2023 International Conference on Information Networking (ICOIN)*, Bangkok, Kingdom of Thailand: IEEE, Jan. 2023, pp. 224–229.

[18] H. Zhu, C. Zhao, and X. Zhang, "A novel image encryption–compression scheme using hyper-chaos and Chinese remainder theorem," *Signal Processing: Image Communication*, vol. 28, no. 6, pp. 670–680, Jul. 2013, doi: 10.1016/j.image.2013.02.004.

[19] C. Fu *et al.*, "An efficient and secure medical image protection scheme based on chaotic maps," *Computers in Biology and Medicine*, vol. 43, no. 8, pp. 1000–1010, Sep. 2013, doi: 10.1016/j.compbiomed.2013.05.005.

[20] R. Gallager, "Variations on a theme by Huffman," *IEEE Transactions on Information Theory*, vol. 24, no. 6, pp. 668–674, Nov. 1978, doi: 10.1109/TIT.1978.1055959.

[21] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consumer Electron.*, vol. 38, no. 1, pp. xviii–xxxiv, Feb. 1992, doi: 10.1109/30.125072.

[22] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transform," *IEEE Trans. Comput.*, vol. C–23, no. 1, pp. 90–93, Jan. 1974, doi: 10.1109/T-C.1974.223784.

[23] D. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952, doi: 10.1109/JRPROC.1952.273898.

[24] T. S. Cho, S. Avidan, and W. T. Freeman, "A probabilistic image jigsaw puzzle solver," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA: IEEE, Jun. 2010, pp. 183–190. doi: 10.1109/CVPR.2010.5540212.

[25] A. C. Gallagher, "Jigsaw puzzles with pieces of unknown orientation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI: IEEE, Jun. 2012, pp. 382–389. doi: 10.1109/CVPR.2012.6247699.

[26] M. Droogenbroeck and R. Benedett, "Techniques For A Selective Encryption Of Uncompressed And Compressed images," Oct. 2002.

[27] W. Puech and J. M. Rodrigues, "Crypto-compression of medical images by selective encryption of DCT," in *13th Eurpoean Signal Processing Conference*, Antalya, Sep. 2005, pp. 1–4.

[28] Y. Ou, C. Sur, and K. H. Rhee, "Region-Based Selective Encryption for Medical Imaging," in *Frontiers in Algorithmics*, F. P. Preparata and Q. Fang, Eds., in Lecture Notes in Computer Science, vol. 4613. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 62–73. doi: 10.1007/978-3-540-73814-5_6.

[29] Y. Zhou, K. Panetta, and S. Agaian, "A lossless encryption method for medical images using edge maps," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Minneapolis, MN: IEEE, Sep. 2009, pp. 3707–3710. doi: 10.1109/IEMBS.2009.5334799.

[30] W. Puech, A. G. Bors, and J. M. Rodrigues, "Protection of Colour Images by Selective Encryption," in *Advanced Color Image Processing and Analysis*, C. Fernandez-Maloigne, Ed., New York, NY: Springer New York, 2013, pp. 397–421. doi: 10.1007/978-1-4419-6190-7_12.

[31] V. Itier, P. Puteaux, and W. Puech, "Recompression of JPEG Crypto-Compressed Images Without a Key," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30,

no. 3, pp. 646–660, Mar. 2020, doi: 10.1109/TCSVT.2019.2894520.

[32] W. Li and Y. Yuan, “A leak and its remedy in JPEG image encryption,” *International Journal of Computer Mathematics*, vol. 84, no. 9, pp. 1367–1378, Sep. 2007, doi: 10.1080/00207160701294376.

[33] K. Minemura, Z. Moayed, K. Wong, X. Qi, and K. Tanaka, “JPEG image scrambling without expansion in bitstream size,” in *2012 19th IEEE International Conference on Image Processing*, Orlando, FL, USA: IEEE, Sep. 2012, pp. 261–264. doi: 10.1109/ICIP.2012.6466845.

[34] Y. Mao and G. Chen, “Chaos-Based Image Encryption,” in *Handbook of Geometric Computing*, Berlin/Heidelberg: Springer-Verlag, 2005, pp. 231–265. doi: 10.1007/3-540-28247-5_8.

[35] K. Kurihara, S. Imaizumi, S. Shiota, and H. Kiya, “An Encryption-then-Compression System for Lossless Image Compression Standards,” *IEICE Transactions on Information and Systems*, vol. E100.D, no. 1, pp. 52–56, 2017, doi: 10.1587/transinf.2016MUL0002.

[36] T. Chuman, K. Kurihara, and H. Kiya, “On the security of block scrambling-based ETC systems against jigsaw puzzle solver attacks,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA: IEEE, Mar. 2017, pp. 2157–2161. doi: 10.1109/ICASSP.2017.7952538.

[37] T. Chuman, W. Sirichotedumrong, and H. Kiya, “Encryption-Then-Compression Systems Using Grayscale-Based Image Encryption for JPEG Images,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1515–1525, Jun. 2019, doi: 10.1109/TIFS.2018.2881677.

[38] W. Sirichotedumrong and H. Kiya, “Grayscale-based block scrambling image encryption using YCbCr color space for encryption-then-compression systems,” *APSIPA Transactions on Signal and Information Processing*, vol. 8, 2019, doi: 10.1017/ATSIP.2018.33.

[39] L. Guo, J. Chen, and J. Li, “Chaos-Based color image encryption and compression scheme using DNA complementary rule and Chinese remainder theorem,” in *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, China: IEEE, Dec. 2016, pp. 208–212. doi: 10.1109/ICCWAMTIP.2016.8079839.

[40] M. Brindha and N. Ammasai Gounden, “A chaos based image encryption and lossless compression algorithm using hash table and Chinese Remainder Theorem,” *Applied Soft Computing*, vol. 40, pp. 379–390, Mar. 2016, doi: 10.1016/j.asoc.2015.09.055.

[41] T. Duseja and M. Deshmukh, “Image compression and encryption using chinese remainder theorem,” *Multimedia Tools and Applications*, vol. 78, no. 12, pp. 16727–16753, Jun. 2019, doi: 10.1007/s11042-018-7023-0.

[42] C. Li, Y. Liu, L. Y. Zhang, and K. Wong, “Cryptanalyzing a class of image encryption schemes based on Chinese Remainder Theorem,” *Signal Processing: Image Communication*, vol. 29, no. 8, pp. 914–920, Sep. 2014, doi: 10.1016/j.image.2014.06.011.

[43] S. Bai, G. B. Zhu, and X. Y. Ji, “Comments on ‘A Novel Image Encryption-Compression Scheme Using Hyper-Chaos and Chinese Remainder Theorem,’” *Applied Mechanics and Materials*, vol. 743, pp. 333–337, Mar. 2015, doi: 10.4028/www.scientific.net/AMM.743.333.

[44] Z.-J. Huang, S. Cheng, L.-H. Gong, and N.-R. Zhou, “Nonlinear optical multi-image encryption scheme with two-dimensional linear canonical transform,” *Optics and Lasers in Engineering*, vol. 124, p. 105821, Jan. 2020, doi: 10.1016/j.optlaseng.2019.105821.

[45] H.-S. Ye, N.-R. Zhou, and L.-H. Gong, “Multi-image compression-encryption scheme based on quaternion discrete fractional Hartley transform and improved pixel adaptive diffusion,”

Signal Processing, vol. 175, p. 107652, Oct. 2020, doi: 10.1016/j.sigpro.2020.107652.

[46] L. Gong, C. Deng, S. Pan, and N. Zhou, "Image compression-encryption algorithms by combining hyper-chaotic system with discrete fractional random transform," *Optics & Laser Technology*, vol. 103, pp. 48–58, Jul. 2018, doi: 10.1016/j.optlastec.2018.01.007.

[47] X. Chai, J. Bi, Z. Gan, X. Liu, Y. Zhang, and Y. Chen, "Color image compression and encryption scheme based on compressive sensing and double random encryption strategy," *Signal Processing*, vol. 176, p. 107684, Nov. 2020, doi: 10.1016/j.sigpro.2020.107684.

[48] M. Zhang *et al.*, "Image Compression and Encryption Scheme Based on Compressive Sensing and Fourier Transform," *IEEE Access*, vol. 8, pp. 40838–40849, 2020, doi: 10.1109/ACCESS.2020.2976798.

[49] X. Chai, H. Wu, Z. Gan, Y. Zhang, and Y. Chen, "Hiding cipher-images generated by 2-D compressive sensing with a multi-embedding strategy," *Signal Processing*, vol. 171, p. 107525, Jun. 2020, doi: 10.1016/j.sigpro.2020.107525.

[50] N. Zhou, A. Zhang, F. Zheng, and L. Gong, "Novel image compression–encryption hybrid algorithm based on key-controlled measurement matrix in compressive sensing," *Optics & Laser Technology*, vol. 62, pp. 152–160, Oct. 2014, doi: 10.1016/j.optlastec.2014.02.015.

[51] L. Gong, K. Qiu, C. Deng, and N. Zhou, "An optical image compression and encryption scheme based on compressive sensing and RSA algorithm," *Optics and Lasers in Engineering*, vol. 121, pp. 169–180, Oct. 2019, doi: 10.1016/j.optlaseng.2019.03.006.

[52] N. Zhou, H. Jiang, L. Gong, and X. Xie, "Double-image compression and encryption algorithm based on co-sparse representation and random pixel exchanging," *Optics and Lasers in Engineering*, vol. 110, pp. 72–79, Nov. 2018, doi: 10.1016/j.optlaseng.2018.05.014.

[53] Y. Zhang, B. Xu, and N. Zhou, "A novel image compression–encryption hybrid algorithm based on the analysis sparse representation," *Optics Communications*, vol. 392, pp. 223–233, Jun. 2017, doi: 10.1016/j.optcom.2017.01.061.

[54] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen, and X. He, "A Review of Compressive Sensing in Information Security Field," *IEEE Access*, vol. 4, pp. 2507–2519, 2016, doi: 10.1109/ACCESS.2016.2569421.

[55] "True Color Kodak Images." http://r0k.us/graphics/kodak/ (accessed Aug. 20, 2019).

[56] G. Schaefer, "UCID-RAW – A Colour Image Database in Raw Format," in *VipIMAGE 2017*, J. M. R. S. Tavares and R. M. Natal Jorge, Eds., in Lecture Notes in Computational Vision and Biomechanics, vol. 27. Cham: Springer International Publishing, 2018, pp. 179–184. doi: 10.1007/978-3-319-68195-5_19.

[57] M. Petrou and C. Petrou, *Image processing: the fundamentals*, 2nd ed. Chichester, U.K: Wiley, 2010.

[58] Y. Wu, "NPCR and UACI Randomness Tests for Image Encryption," *Cyber Journals: Journal of Selected Areas in Telecommunications*, Apr. 2011.

[59] D. R. BULL, *COMMUNICATING PICTURES.* Place of publication not identified: ELSEVIER ACADEMIC Press, 2017.

[60] X. Wu, Y. Li, and J. Kurths, "A New Color Image Encryption Scheme Using CML and a Fractional-Order Chaotic System," *PLoS ONE*, vol. 10, no. 3, p. e0119660, Mar. 2015, doi: 10.1371/journal.pone.0119660.

[61] Y. Zhang, "Test and Verification of AES Used for Image Encryption," *3D Research*, vol. 9, no. 1, Mar. 2018, doi: 10.1007/s13319-017-0154-7.

[62] A. Arab, M. J. Rostami, and B. Ghavami, "An image encryption method based on chaos system and AES algorithm," *J Supercomput*, vol. 75, no. 10, pp. 6663–6682, Oct. 2019, doi: 10.1007/s11227-019-02878-7.

[63] L. Chen, C. Li, and C. Li, “Security measurement of a medical communication scheme based on chaos and DNA coding,” *Journal of Visual Communication and Image Representation*, vol. 83, p. 103424, Feb. 2022, doi: 10.1016/j.jvcir.2021.103424.

[64] K. Iida and H. Kiya, “Privacy-Preserving Content-Based Image Retrieval Using Compressible Encrypted Images,” *IEEE Access*, vol. 8, pp. 200038–200050, 2020, doi: 10.1109/ACCESS.2020.3035563.

[65] K. Kurihara, S. Shiota, and H. Kiya, “An encryption-then-compression system for JPEG standard,” in *2015 Picture Coding Symposium (PCS)*, Cairns, Australia: IEEE, May 2015, pp. 119–123. doi: 10.1109/PCS.2015.7170059.

[66] S. Imaizumi, T. Ogasawara, and H. Kiya, “Block-Permutation-Based Encryption Scheme with Enhanced Color Scrambling,” in *Image Analysis*, P. Sharma and F. M. Bianchi, Eds., in Lecture Notes in Computer Science, vol. 10269. Cham: Springer International Publishing, 2017, pp. 562–573. doi: 10.1007/978-3-319-59126-1_47.

[67] O. Watanabe, A. Uchida, T. Fukuhara, and H. Kiya, “An Encryption-then-Compression system for JPEG 2000 standard,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, Queensland, Australia: IEEE, Apr. 2015, pp. 1226–1230. doi: 10.1109/ICASSP.2015.7178165.

[68] O. Watanabe, T. Fukuhara, and H. Kiya, “A perceptual encryption scheme for Motion JPEG 2000 standard,” in *2015 15th International Symposium on Communications and Information Technologies (ISCIT)*, Nara, Japan: IEEE, Oct. 2015, pp. 125–128. doi: 10.1109/ISCIT.2015.7458323.

[69] K. Kurihara, M. Kikuchi, S. Imaizumi, S. Shiota, and H. Kiya, “An Encryption-then-Compression System for JPEG/Motion JPEG Standard,” *IEICE Trans. Fundamentals*, vol. E98.A, no. 11, pp. 2238–2245, 2015, doi: 10.1587/transfun.E98.A.2238.

[70] K. Kurihara, O. Watanabe, and H. Kiya, “An encryption-then-compression system for JPEG XR standard,” in *2016 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Nara, Japan: IEEE, Jun. 2016, pp. 1–5. doi: 10.1109/BMSB.2016.7521997.

[71] S. Imaizumi and H. Kiya, “A Block-Permutation-Based Encryption Scheme with Independent Processing of RGB Components,” *IEICE Trans. Inf. & Syst.*, vol. E101.D, no. 12, pp. 3150–3157, Dec. 2018, doi: 10.1587/transinf.2018EDT0002.

[72] W. Sirichotedumrong, T. Chuman, S. Imaizumi, and H. Kiya, “Grayscale-Based Block Scrambling Image Encryption for Social Networking Services,” in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, San Diego, CA: IEEE, Jul. 2018, pp. 1–6. doi: 10.1109/ICME.2018.8486525.

[73] N. Asuni and A. Giachetti, “TESTIMAGES: a Large-scale Archive for Testing Visual Devices and Basic Image Processing Algorithms,” *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference*, p. 8 pages, 2014, doi: 10.2312/STAG.20141242.

[74] M. Suárez-Albela, P. Fraga-Lamas, and T. Fernández-Caramés, “A Practical Evaluation on RSA and ECC-Based Cipher Suites for IoT High-Security Energy-Efficient Fog and Mist Computing Devices,” *Sensors*, vol. 18, no. 11, p. 3868, Nov. 2018, doi: 10.3390/s18113868.

[75] “Independent JPEG Group.” http://www.ijg.org/ (accessed Jul. 28, 2021).

[76] Gisle Bjontegaard, “Calculation of average PSNR differences between RD-curves.” VCEG-M33 ITU-T Q6/16, Apr. 2001.

[77] R. Lukac, Ed., *Perceptual digital imaging: methods and applications*. in Digital imaging and computer vision series. Boca Raton: CRC Press, Taylor & Francis Group, 2013.

[78] E. Showkatian, M. Salehi, H. Ghaffari, R. Reiazi, and N. Sadighi, “Deep learning-based

automatic detection of tuberculosis disease in chest X-ray images," *pjr*, vol. 87, no. 1, pp. 118–124, 2022, doi: 10.5114/pjr.2022.113435.

[79] A. I. Awad, M. M. Fouda, M. M. Khashaba, E. R. Mohamed, and K. M. Hosny, "Utilization of mobile edge computing on the Internet of Medical Things: A survey," *ICT Express*, p. S2405959522000753, May 2022, doi: 10.1016/j.icte.2022.05.006.

[80] J. Jain and A. Jain, "Securing E-Healthcare Images Using an Efficient Image Encryption Model," *Scientific Programming*, vol. 2022, pp. 1–11, Mar. 2022, doi: 10.1155/2022/6438331.

[81] Y. Siriwardhana, G. Gür, M. Ylianttila, and M. Liyanage, "The role of 5G for digital healthcare against COVID-19 pandemic: Opportunities and challenges," *ICT Express*, vol. 7, no. 2, pp. 244–252, Jun. 2021, doi: 10.1016/j.icte.2020.10.002.

[82] E. L. C. Macedo *et al.*, "On the security aspects of Internet of Things: A systematic literature review," *J. Commun. Netw.*, vol. 21, no. 5, pp. 444–457, Oct. 2019, doi: 10.1109/JCN.2019.000048.

[83] I. Ahmad and S. Shin, "Region-based Selective Compression and Selective Encryption of Medical Images," in *The 9th International Conference on Smart Media and Applications*, Jeju Republic of Korea: ACM, Sep. 2020, pp. 34–38. doi: 10.1145/3426020.3426027.

[84] A. V. Grivkov and A. A. Smirnov, "Application of convolutional neural networks for diagnostics of tuberculosis," presented at the THE VII INTERNATIONAL YOUNG RESEARCHERS' CONFERENCE – PHYSICS, TECHNOLOGY, INNOVATIONS (PTI-2020), Ekaterinburg, Russia, 2020, p. 080011. doi: 10.1063/5.0032964.

[85] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *arXiv:1512.00567 [cs]*, Dec. 2015, Accessed: Mar. 08, 2021. [Online]. Available: http://arxiv.org/abs/1512.00567

[86] S. Jaeger, S. Candemir, S. Antani, Y.-X. J. Wáng, P.-X. Lu, and G. Thoma, "Two public chest X-ray datasets for computer-aided screening of pulmonary diseases," *Quantitative Imaging in Medicine and Surgery*, vol. 4, no. 6, 2014.

[87] D. Das, K. C. Santosh, and U. Pal, "Inception-based Deep Learning Architecture for Tuberculosis Screening using Chest X-rays," in *2020 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy: IEEE, Jan. 2021, pp. 3612–3619. doi: 10.1109/ICPR48806.2021.9412748.

[88] P. Anu Priya and E. R. Vimina, "Tuberculosis Detection from CXR: An Approach Using Transfer Learning with Various CNN Architectures," in *International Conference on Communication, Computing and Electronics Systems*, V. Bindhu, J. M. R. S. Tavares, A.-A. A. Boulogeorgos, and C. Vuppalapati, Eds., in Lecture Notes in Electrical Engineering, vol. 733. Singapore: Springer Singapore, 2021, pp. 407–418. doi: 10.1007/978-981-33-4909-4_31.

[89] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Apr. 2015, Accessed: Dec. 18, 2020. [Online]. Available: http://arxiv.org/abs/1409.1556

[90] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385 [cs]*, Dec. 2015, Accessed: Mar. 08, 2021. [Online]. Available: http://arxiv.org/abs/1512.03385

[91] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *arXiv:1608.06993 [cs]*, Jan. 2018, Accessed: Mar. 08, 2021. [Online]. Available: http://arxiv.org/abs/1608.06993

[92] K. Cao, J. Zhang, M. Huang, and T. Deng, "X-ray Classification of Tuberculosis Based on Convolutional Networks," in *2021 IEEE International Conference on Artificial Intelligence*

and Industrial Design (AIID), Guangzhou, China: IEEE, May 2021, pp. 125–129. doi: 10.1109/AIID51893.2021.9456476.

[93] M. Rahman, Y. Cao, X. Sun, B. Li, and Y. Hao, "Deep pre-trained networks as a feature extractor with XGBoost to detect tuberculosis from chest X-ray," *Computers & Electrical Engineering*, vol. 93, p. 107252, Jul. 2021, doi: 10.1016/j.compeleceng.2021.107252.

[94] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[95] K. Munadi, K. Muchtar, N. Maulina, and B. Pradhan, "Image Enhancement for Tuberculosis Detection Using Deep Learning," *IEEE Access*, vol. 8, pp. 217897–217907, 2020, doi: 10.1109/ACCESS.2020.3041867.

[96] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *arXiv:1905.11946 [cs, stat]*, Sep. 2020, Accessed: Mar. 08, 2021. [Online]. Available: http://arxiv.org/abs/1905.11946

[97] P. Msonda, S. A. Uymaz, and S. S. Karaağaç, "Spatial Pyramid Pooling in Deep Convolutional Networks for Automatic Tuberculosis Diagnosis," *TS*, vol. 37, no. 6, pp. 1075–1084, Dec. 2020, doi: 10.18280/ts.370620.

[98] C. Szegedy *et al.*, "Going Deeper with Convolutions." arXiv, Sep. 16, 2014. Accessed: Aug. 04, 2022. [Online]. Available: http://arxiv.org/abs/1409.4842

[99] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[100] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," 2014, pp. 346–361. doi: 10.1007/978-3-319-10578-9_23.

[101] S. Rajaraman and S. K. Antani, "Modality-Specific Deep Learning Model Ensembles Toward Improving TB Detection in Chest Radiographs," *IEEE Access*, vol. 8, pp. 27318–27326, 2020, doi: 10.1109/ACCESS.2020.2971257.

[102] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *arXiv:1602.07261 [cs]*, Aug. 2016, Accessed: Mar. 08, 2021. [Online]. Available: http://arxiv.org/abs/1602.07261

[103] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *arXiv:1610.02357 [cs]*, Apr. 2017, Accessed: Mar. 08, 2021. [Online]. Available: http://arxiv.org/abs/1610.02357

[104] C. Dasanayaka and M. B. Dissanayake, "Deep Learning Methods for Screening Pulmonary Tuberculosis Using Chest X-rays," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 9, no. 1, pp. 39–49, Jan. 2021, doi: 10.1080/21681163.2020.1808532.

[105] M. Oloko-Oba and S. Viriri, "Ensemble of Convolution Neural Networks for Automatic Tuberculosis Classification," in *Computational Collective Intelligence*, N. T. Nguyen, L. Iliadis, I. Maglogiannis, and B. Trawiński, Eds., in Lecture Notes in Computer Science, vol. 12876. Cham: Springer International Publishing, 2021, pp. 549–559. doi: 10.1007/978-3-030-88081-1_41.

[106] M. Oloko-Oba and S. Viriri, "Ensemble of EfficientNets for the Diagnosis of Tuberculosis," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–12, Dec. 2021, doi: 10.1155/2021/9790894.

[107] A. F. M. Saif, T. Imtiaz, C. Shahnaz, W.-P. Zhu, and M. O. Ahmad, "Exploiting Cascaded

Ensemble of Features for the Detection of Tuberculosis Using Chest Radiographs," *IEEE Access*, vol. 9, pp. 112388–112399, 2021, doi: 10.1109/ACCESS.2021.3102077.

[108] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," *arXiv:2104.00298 [cs]*, Jun. 2021, Accessed: May 10, 2022. [Online]. Available: http://arxiv.org/abs/2104.00298

[109] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. on Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[110] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[111] D. Kim, J. Joo, and S. C. Kim, "Fake Data Generation for Medical Image Augmentation using GANs," in *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Jeju Island, Korea, Republic of: IEEE, Feb. 2022, pp. 197–199. doi: 10.1109/ICAIIC54071.2022.9722700.

[112] M. Momeny *et al.*, "Learning-to-augment strategy using noisy and denoised data: Improving generalizability of deep CNN for the detection of COVID-19 in X-ray images," *Computers in Biology and Medicine*, vol. 136, p. 104704, Sep. 2021, doi: 10.1016/j.compbiomed.2021.104704.

[113] E. Kim, J. Kim, H. Lee, and S. Kim, "Adaptive Data Augmentation to Achieve Noise Robustness and Overcome Data Deficiency for Deep Learning," *Applied Sciences*, vol. 11, no. 12, p. 5586, Jun. 2021, doi: 10.3390/app11125586.

[114] E. Luz, P. L. Silva, R. Silva, L. Silva, G. Moreira, and D. Menotti, "Towards an Effective and Efficient Deep Learning Model for COVID-19 Patterns Detection in X-ray Images," *Res. Biomed. Eng.*, vol. 38, no. 1, pp. 149–162, Mar. 2022, doi: 10.1007/s42600-021-00151-6.

[115] M. Chetoui and M. A. Akhloufi, "Deep Efficient Neural Networks for Explainable COVID-19 Detection on CXR Images," in *Advances and Trends in Artificial Intelligence. Artificial Intelligence Practices*, H. Fujita, A. Selamat, J. C.-W. Lin, and M. Ali, Eds., in Lecture Notes in Computer Science, vol. 12798. Cham: Springer International Publishing, 2021, pp. 329–340. doi: 10.1007/978-3-030-79457-6_29.

[116] I. U. Khan *et al.*, "Remote Diagnosis and Triaging Model for Skin Cancer Using EfficientNet and Extreme Gradient Boosting," *Complexity*, vol. 2021, pp. 1–13, Sep. 2021, doi: 10.1155/2021/5591614.

[117] G. Marques, D. Agarwal, and I. de la Torre Díez, "Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network," *Applied Soft Computing*, vol. 96, p. 106691, Nov. 2020, doi: 10.1016/j.asoc.2020.106691.

[118] M. Chetoui, M. A. Akhloufi, B. Yousefi, and E. M. Bouattane, "Explainable COVID-19 Detection on Chest X-rays Using an End-to-End Deep Convolutional Neural Network Architecture," *BDCC*, vol. 5, no. 4, p. 73, Dec. 2021, doi: 10.3390/bdcc5040073.

[119] S. Kim, B. Rim, S. Choi, A. Lee, S. Min, and M. Hong, "Deep Learning in Multi-Class Lung Diseases' Classification on Chest X-ray Images," *Diagnostics*, vol. 12, no. 4, p. 915, Apr. 2022, doi: 10.3390/diagnostics12040915.

[120] H. Zhu, S. Salcudean, and R. Rohling, "Gaze-Guided Class Activation Mapping: Leveraging Human Attention for Network Attention in Chest X-rays Classification." arXiv, Feb. 14, 2022. Accessed: Aug. 04, 2022. [Online]. Available: http://arxiv.org/abs/2202.07107

[121] J. Liu, W. Sun, X. Zhao, J. Zhao, and Z. Jiang, "Deep feature fusion classification network (DFFCNet): Towards accurate diagnosis of COVID-19 using chest X-rays images," *Biomedical Signal Processing and Control*, vol. 76, p. 103677, Jul. 2022, doi:

10.1016/j.bspc.2022.103677.

[122] "EfficientNet-EdgeTPU: Creating Accelerator-Optimized Neural Networks with AutoML," *Google AI Blog*. http://ai.googleblog.com/2019/08/efficientnet-edgetpu-creating.html (accessed Aug. 03, 2022).

[123] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *arXiv:1801.04381 [cs]*, Mar. 2019, Accessed: Mar. 08, 2021. [Online]. Available: http://arxiv.org/abs/1801.04381

[124] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks." arXiv, May 16, 2019. Accessed: Aug. 03, 2022. [Online]. Available: http://arxiv.org/abs/1709.01507

[125] S. Hwang, H.-E. Kim, J. Jeong, and H.-J. Kim, "A novel approach for tuberculosis screening based on deep convolutional neural networks," presented at the SPIE Medical Imaging, G. D. Tourassi and S. G. Armato, Eds., San Diego, California, United States, Mar. 2016, p. 97852W. doi: 10.1117/12.2216198.

[126] F. Pasa, V. Golkov, F. Pfeiffer, D. Cremers, and D. Pfeiffer, "Efficient Deep Network Architectures for Fast Chest X-Ray Tuberculosis Screening and Visualization," *Sci Rep*, vol. 9, no. 1, p. 6268, Dec. 2019, doi: 10.1038/s41598-019-42557-4.

[127] L. An *et al.*, "E-TBNet: Light Deep Neural Network for Automatic Detection of Tuberculosis with X-ray DR Imaging," *Sensors*, vol. 22, no. 3, p. 821, Jan. 2022, doi: 10.3390/s22030821.

[128] M. Oloko-Oba and S. Viriri, "A Systematic Review of Deep Learning Techniques for Tuberculosis Detection From Chest Radiograph," *Front. Med.*, vol. 9, p. 830515, Mar. 2022, doi: 10.3389/fmed.2022.830515.

[129] Venkatanath N, Praneeth D, Maruthi Chandrasekhar Bh, S. S. Channappayya, and S. S. Medasani, "Blind image quality evaluation using perception based features," in *2015 Twenty First National Conference on Communications (NCC)*, Mumbai, India: IEEE, Feb. 2015, pp. 1–6. doi: 10.1109/NCC.2015.7084843.

[130] A. Wong, J. R. H. Lee, H. Rahmat-Khah, A. Sabri, and A. Alaref, "TB-Net: A Tailored, Self-Attention Deep Convolutional Neural Network Design for Detection of Tuberculosis Cases from Chest X-ray Images," 2021, doi: 10.48550/ARXIV.2104.03165.

[131] M. Ehrlich, L. Davis, S.-N. Lim, and A. Shrivastava, "Analyzing and Mitigating JPEG Compression Defects in Deep Learning," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, BC, Canada: IEEE, Oct. 2021, pp. 2357–2367. doi: 10.1109/ICCVW54120.2021.00267.

[132] S. Dodge and L. Karam, "Understanding How Image Quality Affects Deep Neural Networks." arXiv, Apr. 21, 2016. Accessed: Aug. 26, 2022. [Online]. Available: http://arxiv.org/abs/1604.04004

[133] S. Ghosh, R. Shet, P. Amon, A. Hutter, and A. Kaup, "Robustness of Deep Convolutional Neural Networks for Image Degradations," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB: IEEE, Apr. 2018, pp. 2916–2920. doi: 10.1109/ICASSP.2018.8461907.

[134] T. Gandor and J. Nalepa, "First Gradually, Then Suddenly: Understanding the Impact of Image Compression on Object Detection Using Deep Learning," *Sensors*, vol. 22, no. 3, p. 1104, Feb. 2022, doi: 10.3390/s22031104.

[135] Y. Chen, A. Janowczyk, and A. Madabhushi, "Quantitative Assessment of the Effects of Compression on Deep Learning in Digital Pathology Image Analysis," *JCO Clinical Cancer Informatics*, no. 4, pp. 221–233, Nov. 2020, doi: 10.1200/CCI.19.00068.

[136] K. De and M. Pedersen, "Impact of Colour on Robustness of Deep Neural Networks," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, BC,

Canada: IEEE, Oct. 2021, pp. 21–30. doi: 10.1109/ICCVW54120.2021.00009.

[137] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE.,” *Journal of machine learning research*, vol. 9, no. 11, 2008.

[138] A. Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” p. 60.

[139] Y. Liu, Y.-H. Wu, Y. Ban, H. Wang, and M.-M. Cheng, “Rethinking Computer-Aided Tuberculosis Diagnosis,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 2643–2652. doi: 10.1109/CVPR42600.2020.00272.

[140] M. A. Ezzat, M. A. Abd El Ghany, S. Almotairi, and M. A.-M. Salem, “Horizontal Review on Video Surveillance for Smart Cities: Edge Devices, Applications, Datasets, and Future Trends,” *Sensors*, vol. 21, no. 9, p. 3222, May 2021, doi: 10.3390/s21093222.

[141] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, “Secure, privacy-preserving and federated machine learning in medical imaging,” *Nat Mach Intell*, vol. 2, no. 6, pp. 305–311, Jun. 2020, doi: 10.1038/s42256-020-0186-1.

[142] W. Kim and J. Seok, “Privacy-preserving collaborative machine learning in biomedical applications,” in *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Jeju Island, Korea, Republic of: IEEE, Feb. 2022, pp. 179–183. doi: 10.1109/ICAIIC54071.2022.9722703.

[143] B. Karimi and A. Krzyzak, “A Study on Significance of Color in Face Recognition using Several Eigenface Algorithms,” in *2007 Canadian Conference on Electrical and Computer Engineering*, Vancouver, BC, Canada: IEEE, 2007, pp. 1309–1312. doi: 10.1109/CCECE.2007.333.

[144] A. Kawamura, Y. Kinoshita, T. Nakachi, S. Shiota, and H. Kiya, “A Privacy-Preserving Machine Learning Scheme Using EtC Images,” *IEICE Trans. Fundamentals*, vol. E103.A, no. 12, pp. 1571–1578, Dec. 2020, doi: 10.1587/transfun.2020SMP0022.

[145] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach Learn*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[146] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Maui, HI, USA: IEEE Comput. Sco. Press, 1991, pp. 586–591. doi: 10.1109/CVPR.1991.139758.

[147] J. Konečný, B. McMahan, and D. Ramage, “Federated Optimization:Distributed Optimization Beyond the Datacenter,” *arXiv:1511.03575 [cs, math]*, Nov. 2015, Accessed: Mar. 07, 2022. [Online]. Available: http://arxiv.org/abs/1511.03575

[148] M. Kim and J. Lee, “Information-theoretic privacy in federated submodel learning,” *ICT Express*, p. S2405959522000297, Feb. 2022, doi: 10.1016/j.icte.2022.02.008.

[149] C. Dwork and A. Roth, “The Algorithmic Foundations of Differential Privacy,” *FNT in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2013, doi: 10.1561/0400000042.

[150] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, “A Survey on Homomorphic Encryption Schemes: Theory and Implementation,” *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–35, Jul. 2019, doi: 10.1145/3214303.

[151] C. Zhao *et al.*, “Secure Multi-Party Computation: Theory, practice and applications,” *Information Sciences*, vol. 476, pp. 357–372, Feb. 2019, doi: 10.1016/j.ins.2018.10.024.

[152] D. Han, J. Kim, and J. Kim, “Deep Pyramidal Residual Networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 6307–6315. doi: 10.1109/CVPR.2017.668.

[153] Y. Yamada, M. Iwamura, T. Akiba, and K. Kise, “Shakedrop Regularization for Deep Residual Learning,” *IEEE Access*, vol. 7, pp. 186126–186136, 2019, doi:

10.1109/ACCESS.2019.2960566.

[154] A. H. Chang and B. M. Case, “Attacks on Image Encryption Schemes for Privacy-Preserving Deep Neural Networks,” *arXiv:2004.13263 [cs]*, Apr. 2020, Accessed: Dec. 20, 2021. [Online]. Available: http://arxiv.org/abs/2004.13263

[155] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” 2020, doi: 10.48550/ARXIV.2010.11929.

[156] A. Trockman and J. Z. Kolter, “Patches Are All You Need?,” 2022, doi: 10.48550/ARXIV.2201.09792.

[157] I. Ahmad and S. Shin, “A Pixel-based Encryption Method for Privacy-Preserving Deep Learning Models,” *arXiv:2203.16780 [cs]*, Mar. 2022, Accessed: Apr. 05, 2022. [Online]. Available: http://arxiv.org/abs/2203.16780

[158] M. Tanaka, “Learnable Image Encryption,” *arXiv:1804.00490 [cs]*, Mar. 2018, Accessed: Dec. 20, 2021. [Online]. Available: http://arxiv.org/abs/1804.00490

[159] K. Madono, M. Tanaka, M. Onishi, and T. Ogawa, “Block-wise Scrambled Image Recognition Using Adaptation Network,” *arXiv:2001.07761 [cs, eess]*, Jan. 2020, Accessed: Jul. 28, 2021. [Online]. Available: http://arxiv.org/abs/2001.07761

[160] W. Sirichotedumrong, Y. Kinoshita, and H. Kiya, “Pixel-Based Image Encryption Without Key Management for Privacy-Preserving Deep Neural Networks,” *IEEE Access*, vol. 7, pp. 177844–177855, 2019, doi: 10.1109/ACCESS.2019.2959017.

[161] “COVID19_Pneumonia_Normal_Chest_Xray_PA_Dataset.” https://bit.ly/3SpBxTy (accessed Sep. 22, 2022).

[162] G. Schaefer and M. Stich, “UCID: an uncompressed color image database,” presented at the Electronic Imaging 2004, M. M. Yeung, R. W. Lienhart, and C.-S. Li, Eds., San Jose, CA, Dec. 2003, pp. 472–480. doi: 10.1117/12.525375.

[163] Q.-X. Huang, W. L. Yap, M.-Y. Chiu, and H.-M. Sun, “Privacy-Preserving Deep Learning With Learnable Image Encryption on Medical Images,” *IEEE Access*, vol. 10, pp. 66345–66355, 2022, doi: 10.1109/ACCESS.2022.3185206.

[164] M. AprilPyone and H. Kiya, “Privacy-Preserving Image Classification Using an Isotropic Network,” *IEEE MultiMedia*, vol. 29, no. 2, pp. 23–33, Apr. 2022, doi: 10.1109/MMUL.2022.3168441.

ACKNOWLEDGMENTS

My sincerest gratitude to my advisor and supervisor, Prof. Seokjoo Shin for his continuous support and encouragement, which have always kept me motivated throughout my Ph.D. He has supported me in both technical and administrative capacities. His comments and suggestions have improved my work and helped me to make my findings understandable and presentable. Prof. Shin provides a very healthy atmosphere in our research group in which he guides our intuitions and gives us enough space to grow. I thank him for the group and one-on-one meetings, which were always useful problem-solving and brainstorming sessions.

I am thankful to my thesis committee members Prof. Sangman Moh, Prof. Moonsoo Kang, Prof. Wooyeol Choi, and Dr. Sungwon Yi for their valuable feedback and comments. I am also thankful to the prolific professors of the department of computer engineering with whom I had a chance to take their courses. Especially Prof. Jeong-A Lee and Prof. Ilyong Chung for their courses that gave me a foundation in the residue number system and number theory, which inspired my earlier work in Ph.D.

I have the great benefit of being born into a family where higher education is encouraged. I am thankful to them for their unconditional love and for giving me the ability to think *otherwise*, which has made me a good researcher. Each of them has made my life special and I thanked them for always being there for me.

I would like to take this opportunity to thank the talented alumni of WHYNET-LAB with whom I got to work with: Ms. Madiha Razzaq, Dr. Keyvan Jaferzadeh, Dr. Samaneh Gholami, Dr. Devarani D. Ningombam, Dr. Ezat Ahmadzadeh, Dr. Lilian C. Mutalemwa, and Mr. Iftekharul I. Shovon. I am very thankful to Mr. Nazmul Islam for joining our research group in 2022. Taking him through the research journey made me realize to reinvent certain ways of doing things that I had already settled in. I am thankful to my friend Dr. Muhammad Usman for his fruitful discussions over coffee and at the balcony of IT building. I am grateful to Dr. Hoon Ko for our morning elevator talks on career and being a researcher. I am very thankful to my friends Hwan Kim and Kyungho Yu for making my life easy in Korea. I am also thankful to my friend Dr. Geoffrey Solano for his kind words and always keeping his door open for me. I am thankful to my friends in the department of Computer Engineering and other departments of Chosun University: Dr. Ramesh K. Lama, Mr. S M Asiful Huda, Mr. Uttam Khatri, Mr. Rukesh Prajapati, and Dr. M. Morshed Alam.

I am thankful to my sath-samandar-paar-bhai, Dr. Inayat Ullah for his continuous guidance, support, and encouragement. I am also thankful to one of my favorite couples and my good friends Rezoan A. Nazib and Rehenuma T. Rodoshi, we had a very good time when they were in Korea, for their continuous support to this day. I really enjoyed my daily lunches (11:40/7) with my dear friend Mr. Suhardi A. Junoh, I really appreciated his company and the wide range of discussion topics he would bring to our table. My best friend Mr. Shehzad Ali, a poli-sci major, whose earlier research work on privacy and surveillance, which we discussed intensively in our routine powwow, has made me understand the social aspect of technological advancements, for which I am thankful to him.

My girlfriend Dr. Giang Thuy Lam—*the girl who cures cancer*—thank you for inspiring me to consider medical image analysis as an application of my work. Despite her background in biology, she always showed a great deal of interest in my ideas, listened to their anecdotes, and bear my overexcitement about them, and I always found her suggestions and comments on my writings helpful. I am glad that we shared the journey of Ph.D. together.

Lastly, I would like to acknowledge the Chosun university tuition fee scholarship, the research fund, and the research incentives. I also thank the National Research Foundation of Korea from the Ministry of Education for supporting my research work through the Basic Science Research Program fund. I also thank WHYNET-LAB for providing monthly stipend, research incentives, travel grants and various fundings to support my research. I would also like to acknowledge the Korea NIPA (National IT Industry Promotion Agency) for providing me with high computational resources to carry out the experiments presented in this thesis. I would also like to thank the International Office of Chosun University for always helping us with immigration and health insurance, and the Graduate School of Chosun University for their help in all the administrative tasks.

Ijaz Ahmad