IMAGE ENCRYPTION TECHNIQUE FOR PRIVACY PRESERVING IMAGE RETRIEVAL

BENXUAN WANG

PhD

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University

Department of Electrical and Electronic Engineering

Image Encryption Technique for Privacy Preserving Image Retrieval

Benxuan Wang

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

August 2024

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best ofmy knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: _____

Name of Student: Benxuan Wang

_____

Abstract

Recently, along with the rapid development of multimedia techniques, many images have been generated daily over various network platforms and those images can be stored to cloud servers for convenience. However, the rich sensitive information embedded in those images often results in security and privacy issues when personal images are outsourced. So the need for secure storage and transmission of images has become increasingly important. However, image retrieval methods require the image information before it can be retrieved. Therefore, it would be necessary to search for effective image encryption and retrieval techniques to protect privacy and also maintain the availability of encrypted data. To handle the conflict, thesis presents our contributions in realizing the encrypted image retrieval system and three different encrypted image retrieval schemes are proposed. The proposed three models can broadly categorize into pixel based method, DCT coefficient based method and deep network based method.

In the first encrypted image retrieval scheme, the encryption operations are adopted on image pixels. For a given image, we first divide it into 8×8 non-overlapped blocks due to the JPEG standard. And the 8-bit binary sequences of pixels in each block are confused by two-level permutation. Specifically, more significant 4-bit binary sequence of the pixel is confused by block permutation, while intra-block permutation is conducted on the less significant 4-bit binary sequence. After encryption on binary sequence, the image confusion is used by block permutation to increase image security and the index is generated from a logistic map. The histogram features can be generated from the confused blocks directly for retrieval processing.

In the second encrypted image retrieval scheme, we extract the features for retrieval from the frequency domain which would consume less computation and communication resources. DCT coefficients are utilized to obtain feature vectors. The encryption operations, including coefficient value substitution and intra-block coefficient shuffling, are performed on JPEG images. With the proposed encryption and compression scheme, the feature would be directly extracted from the frequency domain by using the learning network. And the Siamese architecture for metric learning is used for capturing the similarity well. Finally, the user can receive several encrypted images with similar content according to the query.

As for the third encrypted image retrieval scheme, a new image compression and encryption framework is proposed which integrates encryption algorithms with a learning-based compression network. Our model employs Auto-Encoder (AE) based compression network as the backbone and encryption layers are added. And for higher security, the parameters of the synthesis network are replaced by a new parameter matrix based on a logistic map controlled by a secret key. The encryption key of the system is derived from the image content, which will be embedded in the deep feature vectors. And to learn the entropy model from the scrambled feature maps, an attention scheme is exploited in estimating parameters to achieve more effective compression. In this scheme, the encrypted feature maps are the inputs of another deep network for retrieval. And the training loss function for this retrieval model consists of ranked list loss and cross-entropy loss.

List of Publications

Journal Papers:

[1] Benxuan Wang, Kwok-Tung Lo, “Autoencoder-based joint image compression and encryption,” Journal of Information Security and Applications, vol. 80, 2024, doi: 10.1016/j.jisa.2023.103680.

Submitted Paper:

[2] Benxuan Wang, Kwok-Tung Lo, “Encrypted JPEG Image retrieval based on DCT coefficients,” submitted to Signal Processing: Image Communication.

[3] Benxuan Wang, Kwok-Tung Lo, “Joint image encryption and compression schemes for content-based image retrieval,” submitted to Journal of Visual Communication and Image Representation.

Acknowledgments

During my PhD study period, I would like to express my sincere appreciate to all those who have supported and encouraged me.

First, I want to thank my supervisor Dr. Kwok-Tung Lo, who always supported me with invaluable guidance and helpful assistance during my research life at the Hong Kong Polytechnic University. He provides me the chance to purse the doctoral degree in the department of Electrical and Electronic Engineering. And His constructive feedback on my research works always benefits me and inspires me with new ideas. I appreciate all his contributions to our research work. His expertise and encouragement are crucial factors in the successful completion of my thesis, which will always influence me in future research.

Also, I want to express my appreciate to all my lab-mates for providing much help and advice during my research. Mingjie Zheng and Tianshan Liu have given me a lot of advice and helped me better accomplish my research. They are patient and warm-hearted. Thank to Ziyin Huang, Yanfang Luo, Manhui Wu, Yang Qiang, Xinni Xiong and all other friends for bringing color and vitality into my PhD life.

Last, I would like to thank my parents for their unconditional love and encouragement. With their love, I have gained the motivation to tackle challenges.

Table of Contents

List of Abbreviations

DCT Discrete Cosine Transform

VLI Variable-length Integer

AE Auto-Encoder

RNN Recurrent Neural Network

GAN Generative Adversarial Network

ETC Encryption-then-Compression

SCE Simultaneous Compression Encryption

CTE Compression-then-Encryption

CS Compressive Sensing

SVD Singular Value Decomposition

EOB End-of-Block

RSV Run/Size and Value

EMD Earth Mover's Distance

kNN k-Nearest Neighbor

CBIR Content-based Image Retrieval

NPCR Net Pixel Change Ratio

UACI Unified Average Change in Intensity

SGD Stochastic Gradient Descent

mAP mean Average Precision

PSNR Peak Signal-to-Noise Ratio

QF Quality Factor

AUC Area Under Curve

GDN Generalized Divisive Normalization

MS-SSIM Multiscale Structural Similarity

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Background

Recently, along with the rapid development of digital devices, large amounts of images have been generated in our daily life. As an essential carrier of human communication, images are delivered by users through personal laptops or computers, smart mobile phones and various network platforms. And thanks to the rapid growth of computing technology, many images can be storing to cloud servers such as Google Drive, One Drive and Dropbox by users for lower cost and more convenience. However, the rich personal information is contained in those images which often results in security and privacy issues when outsourcing images to third parties. So the in order to protect data privacy, images are usually encrypted by users before transmitting them to the server for outsourcing. After encryption, the image will be converted to an obviously different one, and nobody can get to know the original content of the image when the encryption key is unknown. And the content of images would be protected with varying methods of encryption. Then the protected images may be outsourced to cloud servers. But this privacy operation may cause an impediment for servers to provide further processing services, like image retrieval, since many image retrieval techniques need to obtain image content and require the image to be decrypted before it can be retrieved. Thus the problem is whether to preserve the retrieval function or to sacrifice it to ensure privacy. Therefore, it would be desirable to develop effective retrieval techniques in encrypted domain under the premise of protecting privacy.

To deal with the conflict between security and retrieval performance, many searchable image encryption methods have been proposed and other researchers tend to find a retrieval scheme which can search for specific images in a database of encrypted images without decrypting them. For those searchable image encryption algorithms [1-4], the statistical information, such as histograms and local descriptors, is preserved which is contained in the original image, to ensure the retrieval processing. However, those methods ignore the compression performance when encrypting images and sacrifice security to a certain extent. As an essential carrier of human communication, the compression performance needs to be considered for better delivery. And most of the images we use are compressed except for some special and professional occasions. Considering the requirement of compression, some researchers propose

cascaded compression encryption methods [5-9], which can achieve high efficiency in compression. In [5, 6], encryption operations are adopted into the compressing stage, while [7-9] encrypt the data with learning-based methods. But these methods only focus on the security and compression efficiency of image and ignore the data availability which is important for retrieval function. Besides, to decrease the computation cost of image retrieval, some researchers tend to transform images to the frequency domain first and then retrieval schemes are exploited on the frequency domain with compressed and encrypted representations [10-13].

In summary, image encryption techniques for privacy preserving image retrieval are required to achieve compression-friendliness, privacy security, and data availability. But the requirements are restricted to each other. For protect security, the content of image can be confused to forbid the attacker from reconstructing the image from the encrypted one. In this case, the relationship between images would be removed, which restricts the similarity measurement. And the confusion operations used for encryption may increase the entropy of the image and decrease the compression performance on encrypted images. To address this challenging issue, this thesis will deeply investigate three specific tasks, pixel based searchable encryption, DCT based image encryption and retrieval, and deep image encryption and retrieval.

1.2 Research Motivations and Contributions

For encrypted image retrieval, the image encryption module used needs to ensure the privacy and security of images by conducting the encryption operation, preventing the reconstruction of the plain image from the ciphertext. But in this case, cipher images may be unavailable for obtaining effective features from the content of images, which limits encrypted image retrieval. In summary, the encrypted image retrieval system needs to ensure the privacy and availability of image. And since images are almost compressed when transmitted, the image encryption and retrieval system may meet the requirements of privacy security, friendly compression and better retrieval accuracy. And different proposed schemes may focus on different points.

In this thesis, three encrypted image retrieval schemes are proposed and the main contributions can be summarized as follows:

- A new searchable encryption scheme is proposed to achieve privacy protection and it can support retrieval on encrypted domains by extracting features from the content of cipher images. The proposed encryption scheme is realized by using pixel-based

operations on images. First, the plain images are segmented into 8×8 non-overlapped blocks due to the JPEG standard. And the proposed pixel-based encryption method is realized by two-level permutation on binary sequences. Using block based permutation method on image confusion can ensure local feature extraction conducted on confused blocks, which can benefit privacy protection and retrieval processing. With the proposed encryption method, the feature for similarity measurement can be directly obtained from confused blocks. And the problem of image retrieval on cipher images can be defined based on the local histogram. With experiments, the security and retrieval performance of the proposed scheme are verified on the dataset.

- A new encrypted JPEG image retrieval scheme is proposed based on DCT coefficients. In this work, the features for retrieval can be extracted from the frequency domain which would consume less computation and communication resources. Here, DCT coefficients are utilized for obtaining feature vectors and also for encryption. The encryption operations are exploited on JPEG images with coefficient value substitution and intra-block pixel permutation. In this case, encrypted images can maintain format compliance. We use the deep architecture in network to get more effective features for encrypted image retrieval from frequency domain. For a given encrypted query image, the server obtain the image descriptors for retrieval from DC and AC coefficients as the which are the inputs of the network. And without decrypting, similarity measurement is conducted between the encrypted query image and database image in the cloud sever. The channel attention module is integrated to select the effective frequency components and reduce the impact of encryption. And we adopt the Siamese architecture for metric learning since the learned image embedding can help the Euclidean distance captures the similarity well. Finally, several encrypted images containing similar content are sent to the user. Experiment results show the encryption and retrieval performance of our scheme.
- A deep encryption and retrieval scheme is proposed. We develop a novel joint compression-encryption model which could be an early attempt to introduce end-to-end learning to the security system. The shuffling operations are conducted on deep feature maps. And for a higher level of visual security, part parameters of the network are replaced by a new parameter matrix based on a logistic map controlled by a secret key. The encryption key of the system is derived from the image content, which will be embedded in the deep feature vectors with a fixed key to save the cost of sending the

key for different images. An attention scheme is exploited in estimating parameters to achieve more effective compression to learn the deep model from the scrambled feature maps. After encryption and compression, the confused deep representations are sent as inputs to another deep similarity network for retrieval. And the training loss function for this retrieval model consists of ranked list loss and cross-entropy loss.

1.3 The Outline of Thesis

The remaining chapters of this thesis are outlined as follows:

Chapter 2 introduces some basic related work in the literature, which includes image compression techniques, image encryption techniques and encrypted image retrieval techniques. For image compression techniques, traditional schemes and deep models are all presented in this chapter. And three types of image encryption techniques are introduced.

Chapter 3 presents a privacy-preserving content based image retrieval scheme, which is realized by using pixel-based encryption operations on images and extracts features use for retrieval from the content of encrypted images.

Chapter 4 presents a encrypted JPEG image retrieval scheme based on DCT coefficients, which can encrypt image by coefficient value substitution and intra-block pixel permutation on coefficients. And the features for retrieval are learned from the frequency domain.

Chapter 5 presents a deep encryption and retrieval scheme that introduces end-to-end learning to the security system. The encryption operations are conducted on deep feature maps. An attention scheme is exploited for compression. After encryption and compression, the confused deep representations are sent to a similarity network for retrieval.

Chapter 6 summarizes the research work of this thesis and provides several potential directions for future research.

Chapter 2 Literature Review

2.1 Image Compression

2.1.1 Traditional Compression

Image compression is one of the foundational research tasks in computer vision and many different approaches have been proposed in the previous decades, such as JPEG and JPEG2000. Most of them apply linear transformation to convert correlated pixels into non-correlated transform coefficients, quantize and then encode the resulting discrete representation by entropy coding. In general, the baseline compression procedure comprises three components – transform, quantizer, and entropy coder. Figure 2.1 shows the block diagram of JPEG compression standard as an example. And as one of the popular image compression standards, JPEG uses a discrete cosine transform on blocks of pixels in an image, while JPEG 2000 uses wavelet transform. But all those algorithms are based on handcrafted encoding/decoding diagrams and use a fixed operation.

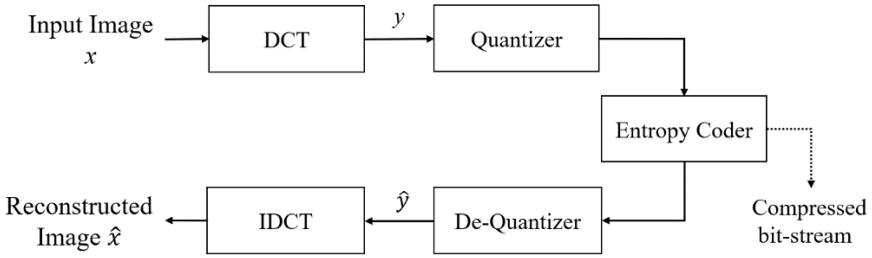


Figure 2.1: Block diagram of the JPEG standard.

JPEG is one of the most common image compression standards and JPEG images are commonly used in many areas. JPEG standard is a kind of lossy compression method which will remove the high frequency information which is not visually obvious. According to JPEG standard [14], a color JPEG image has three components Y, U and V. For each component, it will be divided into 8×8 non-overlapped blocks and each of them is quantized after DCT. Here, the quantization step can discard the visually non-noticeable information by crudely quantizing higher frequencies. In each block, there are one DC coefficient and 63 AC coefficients. And the DC coefficients are converted into binary bits. For the AC coefficients, a set of pairs (r, v) are generated with zigzag scanning. Here, r represents the number of consecutive zero-valued

AC coefficients, and v defines the nonzero AC coefficient. Then, these pairs are entropy encoded into binary bits. All binary bits from DC and AC coefficients are entropy encoded into binary sequence by Huffman table. Finally, the JPEG file bit-stream is generated.

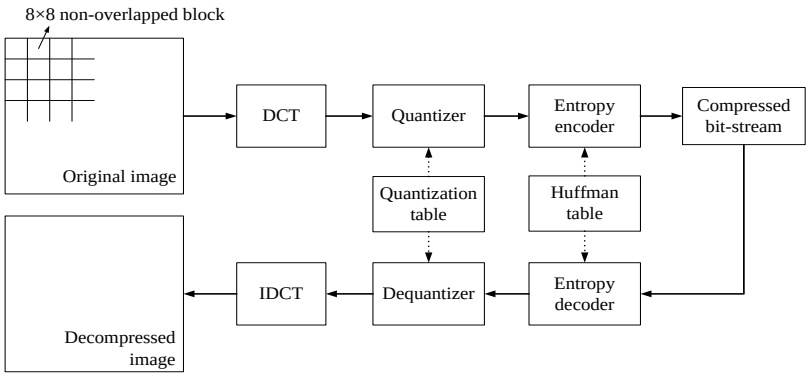


Figure 2.2: JPEG coding and decoding.

2.1.2 Deep Compression

The deep neural network has recently shown remarkable accuracy in multiple visual areas, which implies the potential of learning-based methods to improve the performance of lossy image compression. And several different neural networks have been introduced to image compression. According to the network architectures, those works can broadly categorize into auto-encoder (AE) based methods [15-18], recurrent neural network based methods [19,20] and generative adversarial network based methods [21,22]. AE based methods extract the compressed presentation of an image by replacing the fixed operation, like DCT, with some convolutional transforms. And because of this inherent property of autoencoder, AE based methods can be comparable with traditional transforms and outperform JPEG and JPEG2000. But this kind of method needs separate training for obtaining images at different resolutions. In RNN-based methods, the network weights are shared in various iterations to provide variable output rates without retraining the network. But the residual between the original and predicted images is taken as the input for the next iteration in the loop iteration process that will cause an impact on introducing the encryption operation. GAN-based methods perform well at a high compression ratio, but they are difficult to generate high-quality images.

Excellent joint image compression and encryption scheme should achieve sufficiently high security with good performance of the underlying compression algorithm in terms of compression efficiency. The loop iteration process of RNN-based compression methods may impede the restoration when decryption. And GAN-based compression methods often generate

some extra details after decoding. Therefore, we choose the AE-based compression method as the backbone of our model to achieve a good balance between security and compression ability. The network architecture for AE-based image compression is shown in Figure 2.3.

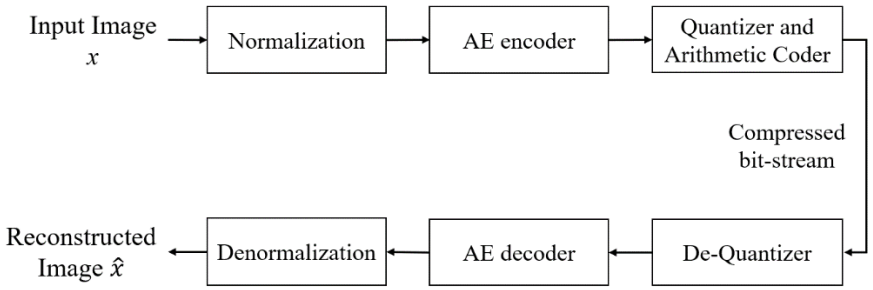


Figure 2.3: Block diagram for autoencoder-based image compression.

In AE-based compression, auto-encoders are used to obtain a hopefully lower-dimensional representation y of data x via hidden layers. The network can be done by the operation $x \rightarrow y \rightarrow \hat{x}$, where y $= f(x)$, $\hat{x} = g(f(x))$, from original data x to a simpler representation and back. The reconstruction error between x and $g(f(x))$ needs to be minimized. Convolutional auto-encoder (CAE) is a type of autoencoder replacing the fixed operation, like DCT, with some convolutional transforms. And the convolution and deconvolution filters are utilized for up-sampling or down-sampling. For autoencoder-based image compression, the goal is to find a better feature function f and generative function g that would give a short code (log-likelihood) of the image data x. In an autoencoder, given an encoder f, a decoder g and a probabilistic model p, the loss function is as

$$\mathbb{E}_x\left[-log\ p\left(f(x)\right)\right] + \beta \cdot \mathbb{E}_x\left[\left\|x - g\left(f(x)\right)\right\|^2\right] \tag{2.1}$$

Here, the discrete probability distribution p is used to assign bits to lower-dimensional representation for entropy coding. And the quantized output of the encoder is the compressed bitstream.

Recently, various AE-based deep network methods have shown promising results in image compression. [23] aimed at directly optimizing the rate-distortion trade-off produced by an autoencoder, while [16] used different strategies to deal with the quantization and entropy rate estimation. To improve the performance, soft relaxation of quantization and entropy was proposed in [24]. And the authors introduced a hyperprior on scale parameters of the latent representation to get a more powerful entropy model in [15]. In [25], an enhanced entropy model is derived to learn the conditional probability model of latent representation. As an

improvement of [15], [17] generalized the hierarchical Gaussian scale mixture model in [15] to a Gaussian mixture model by adding an autoregressive component. Then, [18] proposed using discretized Gaussian mixture likelihoods and smaller residual blocks similar to [16]. Also, an attention module was introduced in [18].

2.2 Image Encryption

Joint image compression and encryption methods have attained much attention among different encryption schemes since compression is a must-do step for most images on the Internet. The framework for image compression and encryption system is to first encrypt and compress the original images by the content owner and then transmit the encrypted and compressed data to the cloud by the network provider. And the decryption and decompression operations are performed at the receiver for image recovery. Usually, encryption operations can be performed before compression, during compression or after compression to safeguard images. So, there are three types of schemes: encryption-then-compression, simultaneous compression encryption and compression-then-encryption.

2.2.1 Encryption then Compression Scheme

For the ETC system, the image data will be first encrypted and then compressed, while the decryption operation will occur after decompression. The feasibility of the ETC system has been demonstrated in [26]. However, applying encryption before compression affects the compression performance since the contents of encrypted images are masked by the secret key and the statistical correlations among neighbouring pixels are destroyed. Thus, many works on ETC system mainly focus on finding a suitable compression algorithm for the proposed encryption scheme. Different sampling and compression approaches are introduced in ETC systems, such as compressive sensing [27,28], uniform downsampling [29,30] and scalar quantizer [31,32]. In [27], a linear transformation of the pixels is used for encrypting data and lossy compression is achieved by CS. To avoid the negative effect of the linear operation on compression, two nonlinear operations are introduced in [28]. Also, gray mapping and 2D projected gradient are utilized for lower compressing complexity and better image recovery. For the uniform downsampling-based scheme, encrypted images are uniformly down-sampled. In [29], adaptive sampling is proposed for compressing the encrypted images and the corresponding multiscale interpolation is conducted for image recovery. Here, bitwise XOR

operation is exploited for masking image content. And in [30], images are encrypted by modulo-256 addition method while deep residual network and U-Net-based attention mechanism are utilized for reconstructing images from encrypted and down-sampled data sequences. In [31], modulo-256 addition method is used for image encryption. Cipher images are decomposed and quantized, and then the quantized sub-images are utilized for generating the bitstream. And in [32], non-overlapping blocks of images are masked by modulo-256 addition method and block permutation is conducted to improve security. Then all encrypted sub-blocks are classified and encoded by difference quantization to achieve flexible compression.

2.2.2 Compression then Encryption Scheme

In contrast to the ETC system, the image will be compressed before conducting the encryption operation in CTE system. In CTE framework, original images are used to compresses for less redundancy, and then encrypted to ensure the security of content information. Then the CTE data are transmitted to the receiver side through public channel. After receiving the transmitted CTE data, the decryption and decompression is performed to reconstruct the original images. Hence, the CTE system is more compression-friendly and encryption operation will be sped up due to the compressed data [33, 34]. In [35], a neural network with small number of hidden layers is used for compression, then zigzag confusion and XOR operations between scrambled data and chaotic sequence are conducted for encryption. In [36], a deep learning-based CS strategy is introduced, and the encryption operations are conducted on multiple CS measurements. However, the CTE-based approaches usually cannot meet format compliance and may increase the data size since encryption will destroy the format and other image information. And in this thesis, all works meet the requirement to maintain format compliance.

2.2.3 Simultaneous Compression Encryption Scheme

Different from the above two systems, the content is compressed and encrypted by incorporating encryption operations into one or more stages of compression in the SCE system. And SCE approaches aim to enhance compression and security efficiency and are usually used to overcome the limitations of the above two strategies. However, achieving SCE system is a complex task since both compression and encryption schemes are considered when evaluating the performance. Many scholars have proposed a number of related works. And for CS-based SCE methods, the measurement matrix generation can be controlled by random sequences for compressing and encrypting the data simultaneously. In [37], a key-controlled measurement

matrix is used, and the obtained measurements are scrambled for encryption with a key generated by a logistic map. To shorten the key, in [37], the plain image is divided into 4 blocks before compressing and encrypting, and random pixel scrambling is introduced to CS-based SCE method. Subsequently, a 2D CS-based encryption-compression approach is proposed for better compression performance and cycle shift operation controlled by a hyper-chaotic system is exploited for re-encryption in [38]. In [39], a double image encryption algorithm is proposed and a co-sparse analysis model is used before compressing and encrypting by CS. Moreover, in [40], integer wavelet transform is combined with CS and the image information is embedded into IWT coefficients. The key is extracted from plain images by SHA-3 algorithm to achieve high key sensitivity and the final results show the high security of the method. Besides, some algorithms combining discrete wavelet transform and CS have been proposed [41,42] to ensure a higher security level, where DWT first transforms the images to get sparse matrices. In [41], the coefficients of the obtained sparse matrixes are confused by a zigzag path, then encrypted into a compressed bit-stream using CS. The simulation results demonstrate that this method has a high-security level, but the time complexity is high especially for large images. In [42], the row and column permutations are conducted on the coefficient matrix after DWT and the measurement matrix used for CS is generated by the 2D-SLIM map. The GF(257) multiplication algorithm is used in diffusion to enhance security. In [43], the sparse representations from the DWT operation are permuted by the Lorenz system and then compressed by SVD. And through the chaotic scrambling and XOR operation, the final compressed cipher image is obtained. The simulation results indicate that this approach is highly secure and robust, but the computational complexity of reconstruction is high. Besides, since JPEG is a commonly used compression method of digital images, simultaneous image compression and encryption scheme for JPEG images is proposed in [44]. In their work, new order-8 transforms are developed, and the new transforms are applied alternatively controlled by a key sequence. Block permutation is then applied after the transformation process. As the improvement of [44], AC- and DC-coefficient encryption algorithms are applied in [45]. The key for block permutation and DC confusion is generated through the BLAKE2 hash function and embedded into AC coefficients. In [46], order-16 DCT transform is used for the new block-based encryption and compression scheme instead of order-8 orthogonal transforms and the corresponding algorithm for coefficients distribution is proposed. Block permutation and shuffling RSV pairs of AC coefficients are then applied to achieve a higher security level. And end-of-block identifiers are used to ensure format-compliant for JPEG decompression which are embedded into AC coefficients after permuting RSV pairs. A good compromise between

the security and compression efficiency is obtained. But the original correlation cannot be completely removed for these block-based methods. In [47], the images are represented in a 2D discrete wavelet domain and measured by 2D CS in which the measurement matrices are constructed with a chaotic system, and then the measurement data are re-encrypted by taking a double random scrambling and a multiple random diffusion. However, its resistance to both Gaussian noise attack and occlusion attack is not good enough. Moreover, singular value decomposition can be used to replace CS.

2.3 Retrieval on Encrypted Image

Recently, the encrypted image retrieval system has attracted much research interest and many works for performing encrypted image research have been proposed. The aim of the encrypted image retrieval system is to achieve effective retrieval of encrypted images while ensuring the privacy requirement. And to search images better, the works for retrieval on encrypted images can be divided into several types. The first type is extracting features from plain images for retrieval and then the images and features are both encrypted before being transmitted to the server. And the retrieval is conducted by comparing the encrypted feature of the query image and that of each image in the database to search which images are similar to the query. For the methods in this category, images are encrypted by image encryption algorithms and the focus is on the problem of image feature protection which enable the similarity measures among encrypted features.

The first scheme for searching on encrypted images is proposed in [48], which introduces the content-based image retrieval to the encrypted domain. The retrieval is achieved by using secure search indexes to match visual strings. The search indexes are extracted from plain images and encrypted by word IDs scrambling, order-preserving encryption, and min-Hash algorithm with randomized hash functions. To remain approximate similarity between protected features, three visual feature protection schemes are proposed in [49], including bit-plane randomization, random projection, and randomized unary encoding. And these protection methods can maintain the correlation among features after encryption, so the encrypted image retrieval can be ensured. In [50], MPEG-7 visual descriptors can be represented as the feature vectors which are extracted from plain images, and the secure k-nearest neighbour (kNN) algorithm is employed to protect these feature vectors. And pre-filter tables constructed by local sensitive hash (LSH) are performed for better retrieval results. Since homomorphic encryption is one of the essential ways for secure computation, [51] focuses on comparing homomorphic

encryption-based schemes and randomization-based schemes for privacy-preserving image search. As an extension of [48,49], [51] comprehensively compares two randomization-based encryption methods in [48,49] with homomorphic encryption. The homomorphic-based schemes have proved to be more secure and perform better in large image databases but require extensive computation which make them hard to use in practice, while randomization-based schemes can offer efficient retrieval but reveal some information about randomized features. In [52], the proposed content-based system supports local feature-based encrypted image retrieval and evaluates the similarity by using Earth Mover's Distance (EMD) with improved locality-sensitive hashing. The scheme in [52] focuses on performing efficient similarity measurements to improve retrieval performance while the bag-of-words model and improved EMD with a linear transformation are conducted. In [53], encrypted image retrieval is developed by extracting visual descriptors from plain images and encrypting them by using a secure kNN algorithm. The pre-filter tables are constructed by LSH to increase retrieval efficiency. And to deter illegally copy and distribution to unauthorized users, a watermark-based protocol is also used. The method in [54] utilizes compact binary sequences to replace the high dimensional representations and adapts the asymmetric scalar-product-preserving encryption for privacy protection. And the secure kNN scheme is combined with the binary-based vector quantization and the similarity measure with asymmetric distance. In summary, these systems can solve the privacy issue while performing efficient retrieval. But the independent image feature extraction operation and feature encryption operation will incur extra computation costs and inconvenience for users.

For the above works, too much computation and communication resources are consumed. Therefore, to overcome the limitation, other retrieval schemes have been developed focusing on feature extraction from the encrypted domain. In [55], images are encrypted by shuffling DCT coefficients before transmitting to the server. For different blocks, the coefficients are permuted pseudo-randomly which means that each coefficient is moved to another block and the frequency position remains unchanged. After that, the retrieval is conducted based on the histograms of these coefficients. This encryption method can maintain the format of JPEG file but the information about the coefficient histogram can be leaked. In [56], the color and texture information are encrypted by two different methods respectively. The pixel color values encryption is employed for color information while pixel positions permutation can protect texture information. Here, image compression is the optional step since the scheme in [56] is pixel-based encryption method. And the image retrieval is performed on the global color

features. Then, Chou [57] proposes the encryption scheme based on block transformation to protect plaintext images and introduces the white noise images into block transformation for better protection. The image retrieval can be performed by comparing the distance based on the color histogram between the query and images from the cloud server. Besides, image convolution on the encryption domain is also discussed in [57], and it proved that the effect of encryption and decryption operations on image convolution can almost be neglected. In [58], different encryption operations are conducted on AC and DC coefficients respectively. Exclusive-or operation is conducted on DC Huffman codes, while AC coefficients are encrypted by using the scrambling operation. Then the AC coefficients are used to extract histograms as the features for retrieval. Compared with [55], the scheme in [58] can achieve better security, but it may face the information leakage of AC coefficients. In another work by Cheng [59], the encrypted image retrieval system is realized by encrypting the coded data using the stream cipher and permutation operation. Huffman codes are modified by using exclusive-or operation with a standard stream cipher, and permutation is conducted on encoded binary sequences of DC coefficients. And for retrieval, the Markov features can be obtained from the encrypted domain in this approach directly. Multi-class supporting vector machine is used for obtaining low-dimensional feature vectors. Cheng et al. [60] developed the encrypted image retrieval system by extracting intra-block-based features from DCT coefficients which is better than the global features used in [58]. Inter-block permutation and exclusive-or operation are performed on DC coefficients, while AC coefficients are encrypted by intra-block permutation. However, this method is weak in terms of security against differential attacks. Xu et al. [61] utilized orthogonal decomposition to divide the image into two parts: the encryption field and the feature extraction field. The encryption operations are conducted on the encryption field. Then the encryption field and the feature extraction field would merge by the inverse orthogonal transform for the final encryption data. However, the feature extraction field is not encrypted which may cause information leakage. In [62], stream cipher and permutation cipher are used for encryption and then the Huffman-code histogram is changed after encryption with JPEG format maintenance. Here, Huffman-code histograms are used for retrieval and QT encryption method is exploited to improve accuracy when meeting different QFs. But redundant space is created during encryption.

In addition to finding suitable encryption schemes for retrieval, some researchers are focusing on improving retrieval results. In these schemes, the encryption operations are conducted on original images or feature vectors from original images. Among them, the privacy is protected

by these encryption techniques. In [50, 53], LSH is used for getting features from plain images and the secure kNN algorithm is introduced for privacy security. In [63], HSV histograms and DCT histograms are integrated to obtain features which would be protected by the secure inner. And the copy of the database is merged into encrypted features to resist the statistical attack. In [64], a secure multiparty computing technology is developed to achieve multiple-owner communication and privacy. By using this scheme, the system can retrieve images gathered from multiple sources, and personal information will not be leaked during multiple-owner communication. In [65], the improved Harris algorithm is proposed to extract features and the speeded up feature algorithm is adopted for generated feature vectors. The chaotic encryption scheme is used for indexes security. In [66], the hyperchaotic system is used for image encryption while an improved pairwise-supervised hashing scheme is adopted for encrypting index. The scheme in [66] is focused on improving search efficiency and also considers the security issue. In [67], an encrypted hierarchical index tree is employed to obtain the secure index and speed up the retrieval processing. Similar to [50, 53], the secure kNN algorithm is used to obtain the secure index, and then a secure hierarchical index graph is developed to speed up retrieval processing. And with the development of the deep network, some deep architectures are employed to obtain efficient retrieval. In [68], transformed convolutional neural network is used to extract features and the encrypted hierarchical index tree can be employed for efficient search process. In [69], the fine-tuned convolutional neural network is performed to extract image features and the features are encrypted by the secure kNN algorithm. In [70], end-to-end encrypted image retrieval is proposed. Vision transformer model with triplet loss and cross-entropy loss is used as the backbone to extract features for search.

2.4 Summary

This chapter presents a brief introduction on traditional image compression and deep image compression. And some existing works are introduced which include image compression techniques, image encryption techniques and encrypted image retrieval techniques. For image compression techniques, traditional schemes and deep models are briefly introduced in this chapter. Various image compression and encryption schemes are introduced, and these algorithms can be categorized into three classes: encryption-then-compression scheme, simultaneous compression encryption scheme and compression-then-encryption scheme. The advantages and limitations are also discussed.

Chapter 3 Privacy-preserving Content based Image Retrieval

Content-based Image Retrieval (CBIR) techniques are commonly used for similarity measurement on large amounts of images and can return the images quickly and effectively. But outsourcing CBIR service to cloud servers may cause privacy concerns. In this chapter, a new searchable encryption method is proposed to achieve privacy protection and it can support the similarity search scheme conducted on encrypted domains by extracting features from the content of encrypted images. The work in this chapter is to handle the conflict between security and retrieval performance and the proposed scheme tend to preserve the content of images for better retrieval without sacrificing privacy security. Here, the proposed encryption scheme is realized by using pixel-based operations on images. First, plain images are segmented to 8×8 non-overlapped blocks due to the JPEG standard. For each pixel in the block, it can be represented with 8-bit binary sequence. Then, more significant 4-bit binary sequence of the pixel is confused by block permutation, while intra-block permutation is conducted on the less significant 4-bit binary sequence. After encryption on binary sequence, the image confusion is used by block permutation to increase image security and the index is generated from a logistic map. The histogram features for retrieval can be directly extracted from encrypted blocks. The major contributions of our method are listed as follows:

1) The proposed pixel-based encryption method is realized by block permutation and intra-block permutation on binary sequences. The value of the pixel can be replaced by the new one to protect the content of images.

2)Block-based permutation method on image confusion is adopted, which can further improve security without restricting the local feature extraction. And a logistic map generated from image content provides a new index for image confusion. Different index is different images since the initial value of the map is calculated from the content of images. With the proposed encryption method, the feature can be obtained from confused blocks directly.

3) The problem of similarity measurement on the encryption domain can be defined based on the local histogram. Experimental results show the security and retrieval performance of the proposed method and factors affecting the performance are discussed.

The rest of this chapter is organized as follows: Section 3.1 gives an overview of the proposed system module. Section 3.2 introduces pixel-based encryption and compression scheme. Section 3.3 explains the details of similarity measurement for retrieval. Performance evaluations on compression efficiency and security will be given in Section 3.4 and 3.5, respectively, with a comparison with other schemes. Section 3.6 provides a concluding remark.

3.1 Overview of the Proposed Module

The proposed scheme in this section includes two parts: the image encryption method and the image retrieval method. For a given image, the user needs to generate a set of secret keys for binary sequence permutation and obtain the initial value of the logistic map from image first. After getting keys, the encryption operations are employed on the image including binary sequence permutation on two different levels and image confusion before compression. After that the encrypted image is uploaded to the cloud server to generate the image database. For query, the user submits encrypted query data to the server to retrieve similar images. After receiving the query request, the cloud server extracts all features from the encrypted database through similarity measurement. Similar images are returned to the user as retrieval results, and the user can decrypt the results using the secret key.

3.2 Pixel based Encryption and Compression Model

The framework of the proposed encryption and compression model is shown in Figure 3.1. The proposed encryption scheme includes binary sequence permutation on two different levels and image confusion by block permutation. First, the plain images are segmented into 8×8 non-overlapped blocks due to the JPEG standard. For each pixel in the block, it can be represented with 8-bit binary sequence. Then, more significant 4-bit binary sequence of the pixel is confused by block permutation, while intra block permutation is conducted on the less significant 4-bit binary sequence. These two-level permutation operations are controlled by predefined secret $Key\ 1$. The new index is generated by using BLAKE256 hash function. After encryption on binary sequence, the image confusion is used by block permutation to increase image security. The new index is generated from a logistic map which is controlled by $Key\ 2$.

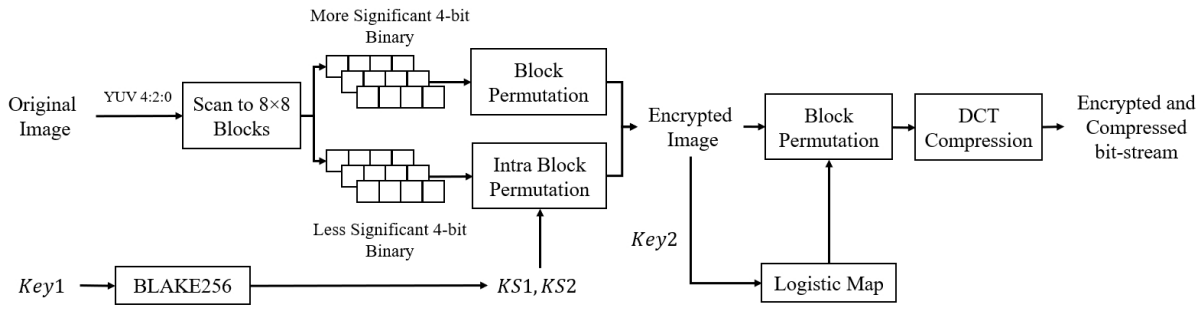


Figure 3.1: Framework of the proposed image encryption algorithm.

3.1.1 Key Generation

In our encryption scheme, the BLAKE256 hashing algorithm is used to generate the encryption keystream by taking the predefined key $Key1$ as input. Here, the predefined key $Key1$ is a random sequence with fixed-length which is used for value substitution and generating $KS1$ and $KS2$. The predefined key $Key1$ is the initial seed to generate. For a given index array R with r elements, Fisher-Yates Shuffle [71] do

for $p \leftarrow r$ to 2 **do**
    $q \leftarrow$ random integer $(1 \leq q \leq p)$
    exchange $R[p]$ and $R[q]$
end for

Here, the random integer is obtained from the key stream $KS1$ and $KS2$.

For image confusion by intra-block permutation, a logistic map is used to generate a random sequence. And this random sequence can be used to obtain the new index when scrambling the pixel position within each block. The logistic map we used is defined as:

$$U_{t+1} = \theta U_t(U_t + 1),\ U_t \in (0,1) \tag{3.1}$$

where U_t is the value for t iterations, and θ is the system parameter. The system is chaotic when θ is in the range of [3.57, 4]. Here, the initial value U_0 is calculated with the average pixel value of the encrypted image after two-level binary permutation. And the secure key $Key\ 2$ consists of the initial value U_0 and parameter θ. Since the logistic map is sensitive to its initial value, different logistic maps can be generated with different values of U_0 for different images, which can improve security.

3.1.2 Two-level Binary Permutation

For most pixel-based encryption methods, the encrypted image may preserve most of content information of the original image and leak information easily when the attacker knows the chipper image. This kind of encryption method usually cannot handle statistical attacks.

In order to ensure image security, we propose a two-level binary permutation scheme including intra-block permutation and block permutation. For each pixel in the block, it can be represented with 8-bit binary sequence. Then, more significant 4-bit binary sequence of the pixel is confused by block permutation, while intra-block permutation is conducted on the less significant 4-bit binary sequence. Block permutation is conducted on more significant 4-bit binary sequence, including first four bytes. During permutation, we generate random permutation index to shuffle the binary position of each pixel in image. The details are described in Algorithm1.

Algorithm 1: BlockPermut

Input: 8×8 blocks, the random sequence $KS1$

Output: Encrypted blocks

1: Denote q_i as more significant 4-bit binary sequences of pixels in *i*th block

2: Denote $q_i^{'}$ as more significant 4-bit binary sequences of pixels in *i*th encrypted block

3: Perform Yates Shuffle algorithm where the random integer in each loop is from $KS1$ to generate a new index H for permutation

4: **for** each q_i **do**

5: $q_i^{'} \leftarrow q_i[H]$

6: **end for**

Intra-block permutation is conducted on less significant 4-bit binary sequence, including last four bytes. During block permutation, we generate random permutation index to shuffle the binary position of each pixel in block. The details are described in Algorithm2.

Algorithm 2: IntrablockPermut

Input: Pixels in 8×8 block, the random sequence $KS2$

Output: Encrypted block

1: Denote q_i as less significant 4-bit binary sequence of *i*th pixel in block

2: Denote $q_i^{'}$ as less significant 4-bit binary sequence of *i*th pixel in encrypted block

3: Perform Yates Shuffle algorithm where the random integer in each loop is from $KS1$ to generate a new index *G* for permutation

4: **for** each q_i **do**

5: $q_i^{'} \leftarrow q_i[G]$

6: **end for**

3.1.3 Image Confusion

After two-level binary permutation, image confusion is conducted by block permutation, which can preserve the local information of chipper images. Here, the logistic map is used to generate a random sequence. And this random sequence can be used to obtain the new index when scrambling the pixel position in image. The details of the permutation stage are described in Algorithm 3.

Algorithm 3: ImagePermut

Input: 8×8 blocks, the initial value U_0 and parameter θ of logistic map from $Key2$

Output: Encrypted block

1: Denote b_{ij} as *i*th pixel in *j*th block B_j

2: Denote $b_{ij}^{'}$ as *i*th pixel in *j*th encrypted block $B_j^{'}$

3: Iterate the logistic map with the initial value U_0 and parameter θ and obtain chaotic sequence *X*

4: Sort *X* in ascending order and use its index values as the new index *F1* and *F2* for permutation

for block B_j **do**

5: **for** each b_{ij} in block B_j **do**

6: $b_{ij}^{'} \leftarrow b_{ij}[F1]$

7: **end for**

8: **end for**

9: **for** each block B_j **do**

10: $B_j^{'} \leftarrow B_j[F2]$

11: **end for**

3.3 Content-based Image Retrieval

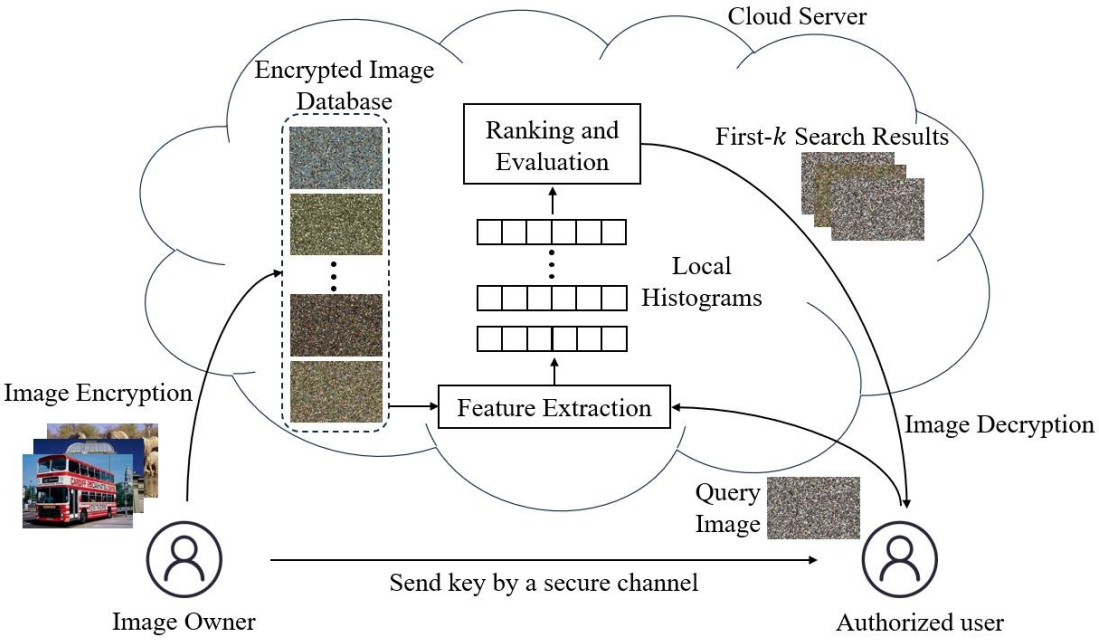


Figure 3.2: Framework of the content-based encrypted image retrieval.

The framework of the proposed image retrieval system is shown in Figure 3.2. The image owner uploads encrypted images to the cloud server, and those images are stored in the encrypted image database. When the authorized user sends the query image to the server, the features of the query image and all images in the dataset are extracted. And after ranking and evaluating, serval images that are similar to the query image are sent to the user as the search results. Finally, the authorized user can use the corresponding key to decrypt the search results.

Local histogram is used for content based image retrieval since the global histogram cannot capture the spatial information of the image. Some researchers use other classic local features, such as SIFT which is robust to scaling, rotation, affine distortion, and illumination changes. But these methods need additional communication between the server and users, which may cause undesirable burdens. So we use local histogram to extract feature. First, the original images are divided into non-overlapping 8×8 blocks before compression during the image encryption process and then image encryption is conducted. Next, local histogram is calculated from each block, and denote the local histogram of jth block in ith image as h_{ij}. Here the local histograms we obtained are the encrypted ones since all pixel values have been encrypted by two-level binary permutation.

We introduce the similarity measurement to process image spatial information for retrieval. For two images with c channels, EI_1 and EI_2 respectively represent the encrypted image of them.

EB_{j1} and EB_{j2} respectively represent jth block of two images. The similarity between block can be calculated by:

$$\Delta\left(EB_{j1}, EB_{j2}\right) = \sum_{c=1}^{3} \sum_{v=0}^{255} \left(1 - \frac{\left|p_{i1}^{c}(v) - p_{i2}^{c}(v)\right|}{1 + p_{i1}^{c}(v) + p_{i2}^{c}(v)}\right) \tag{3.2}$$

Here, $p_{i1}^{c}(v)$ and $p_{i2}^{c}(v)$ respectively denote the frequency of pixel value v in the c channel of block EB_{j1} and block EB_{j2}. And the similarity between the image EI_1 and the image EI_2 is as follows:

$$\Delta(EI_1, EI_2) = \sum_{EB_j \in EI} \Delta\left(EB_{j1}, EB_{j2}\right) \tag{3.3}$$

3.4 Performance Evaluation

In this section, we evaluate the compression performance and the perceptual security of our proposed scheme. We implement the proposed method with MATLAB 2019a on Win-10 operating system. The performance evaluation is conducted on the publicly available Kodak dataset [72]. There are 24 high-quality images in this dataset, and some of them are shown in Figure 3.3. In Figure 3.4, the encryption images of test set are shown. It is obvious that the visual information of the plain images has been well masked with the proposed encryption method.

Figure 3.3: Test images from Kodak dataset.

Figure 3.4: Encryption images of test images.

In the perceptual security evaluation of encryption methods, the peak signal-to-noise ratio (PSNR) is used for evaluating the compression performance. In Figure 3.5, when the encryption key is provided for decryption, the PSNR values of our proposed model can illustrate the compression efficiency of our model. The closer the curve is to JPEG, the higher the compression efficiency. Similar to [81], pixel based encryption operations are used in our scheme, which affects the compression performance. And in [62], a joint encryption and compression scheme is employed for better encryption.

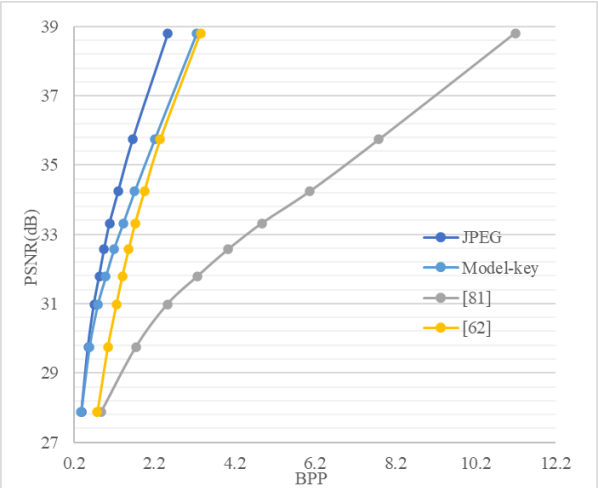


Figure 3.5: Rate distortion performance for different methods.

3.5 Security Analysis

This section is going to discuss various cryptographic attacking methods, such as ciphertext-only attack, differential attack, and statistical attack. And the robustness against those attacks will be evaluated to analyze the security of our scheme.

3.5.1 Ciphertext-only Attack

The ciphertext-only attack is one of the most basic cryptanalysis techniques. And to handle it, the key space of the cryptosystem need to be large. In our model, the 256-bit encryption key $Key1$ is defined which controls two-level binary permutation in the proposed model. And for logistic map, the initial value U_0 and parameter θ from $Key2$ are both $10^{(16)}$ as mentioned in [73]. Therefore, we obtain a $2^{(256)} \times 10^{(16)}$ keyspace which is lager than the theoretical requirement with $2^{(100)}$. Because of the large keyspace of our model, it is tough for the attacker to break down and our encryption system can resist this attack easily.

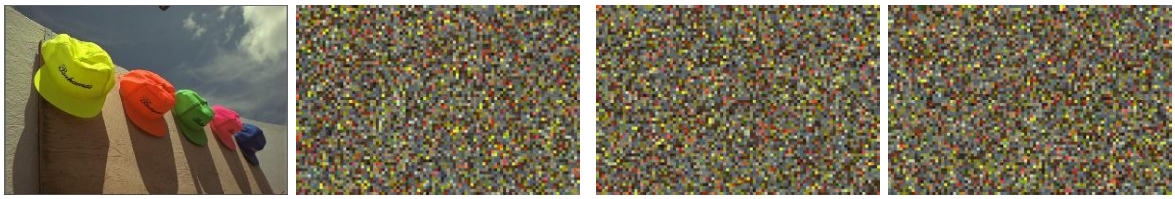
3.5.2 Key sensitivity Analysis

In general, the security of an encryption system should only rely on the secrecy of keys but not the underlying techniques. In this regard, a cryptosystem needs to be highly sensitive to the encryption and decryption keys used. Here, we use the plain image 'kodim03'as examples. The high key sensitivity level can be demonstrated in two parts:

1) A completely different ciphertext image would be generated for the same plain image if the encryption keys used change slightly.

2) The encrypted image should not be decrypted when a key having minor change to the encryption key is used.

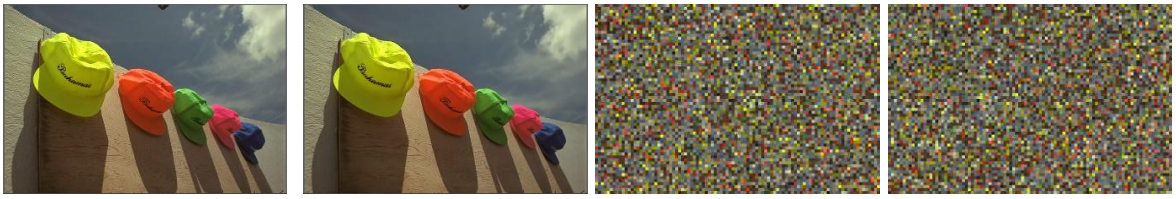
In the first case, we first make a minor change in encryption key $Key1$ to generate the new key stream $KS1$. We then generate two encrypted images using the two different keys for the same input image. And the encrypted images are shown in Figure 3.5(b) and 3.5(c). Then we make a minor change on both encryption key $Key1$ and $Key2$, and the encrypted images are shown in Figure 3.5(d). It is clearly seen that the encrypted images obtained by two slightly different keys are very different, and our proposed model can fulfil the first case of key sensitivity.



(a) (b) (c) (d)

Figure 3.6: Comparison of encrypted images with different keys. (a) plain images, (b) cipher image through the original key, (c) cipher image through the new $Key1$, (d) cipher image through the new $Key1$ and $Key2$.

For the second case, the original key and the new key generated are utilized to decrypt the same cipher-image encrypted with the original key. The and the decrypted images using the two different keys are shown in Figure 3.6(c) and 3.6 (d). It is seen that only the original key can recover the original image.

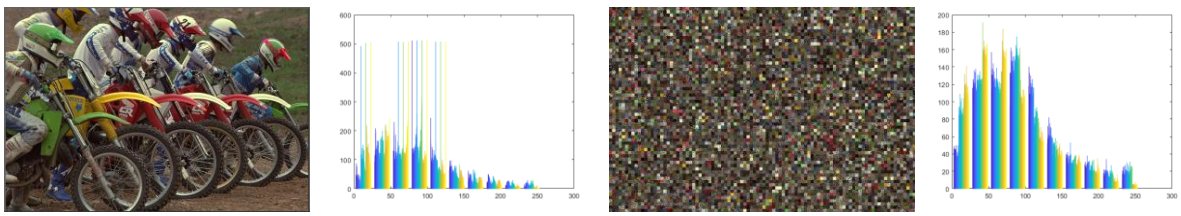


(a) (b) (c) (d)

Figure 3.7: Decryption with different keys. (a) original image, (b) decrypted image using the original key $Key1$, (c) decrypted image using the new $Key1$, (d) decrypted image using the new $Key1$and $Key2$.

3.5.3 Statistical Attack

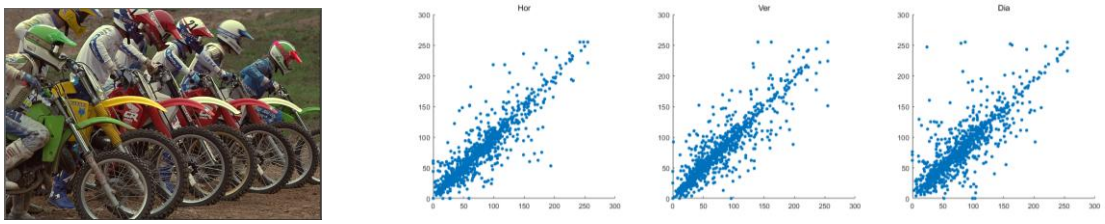
A good encryption scheme needs to reduce the statistical relationship between plain images and encrypted images to defend against this attack. And the histogram and correlation chart are two standard methods to illustrate the correlation. To evaluate the robustness against the statistical attack, the histogram of image ‘kodim05’ and corresponding encrypted images are given in Figure 3.7. It is seen that there are large differences between the histograms of images before and after encryption, which means that the encryption operation can decrease the pixel’s correlation.



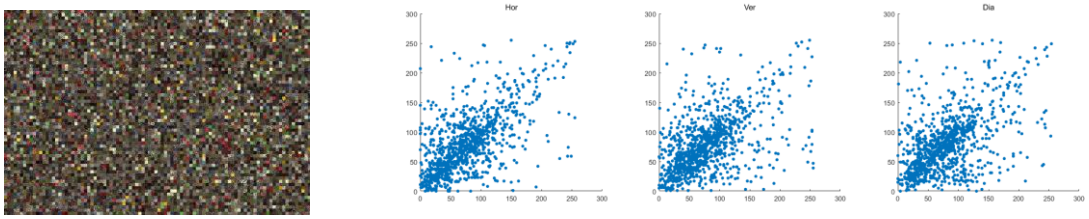
(a) (b)

Figure 3.8: Histogram charts of plain-image and cipher-image. (a) plain-image (b) cipher-image.

And Figure 3.8 shows the correlation charts of image 'kodim05' and ciphertext images under our encryption model. From the result, the encrypted image still contains a lot of spatial information.



(a)



(b)

Figure 3.9: Correlation charts of images before and after encryption. (a) original image (b) encrypted image from the proposed model.

3.5.4 Differential Attack

In differential attacks, attackers try to obtain the encryption keys by studying the influence of input differences on output changes. To resist differential attack, the encryption system needs to be sensitive to changes in the image, and minor changes in plain images should cause large changes in encryption images. The common criteria used in measuring the degree of image change are net pixel change ratio (NPCR) and unified average change in intensity (UACI). The value of NPCR measures the rate of change between pixels in images before and after encryption. Generally, the higher value of NPCR, the better performance of encryption. The value of UACI refers to the average intensity difference between two images. When UACI is close to 33%, the encryption system has higher security. For an image, NPCR and UACI are given as follows:

$$T(m,n) = \begin{cases} 0, & if\ I_{e1}(m,n) = I_{e2}(m,n) \\ 1, & if\ I_{e1}(m,n) \neq I_{e2}(m,n) \end{cases} \quad (1 \leq m \leq M, 1 \leq n \leq N) \quad (3.4)$$

$$\text{NPCR} = \frac{\sum_{m,n} T(m,n)}{m \times n} \times 100\% \quad (3.5)$$

$$\text{UACI} = \frac{\sum_{m,n} \frac{|I_{e1}(m,n) - I_{e2}(m,n)|}{255}}{M \times N} \times 100\% \quad (3.6)$$

where M and N are the width and height of images. I_{e1} and I_{e2} are the encrypted images from two different original images. These two parameters can show slight changes in images. Therefore, to evaluate the robustness against differential attack, we conduct a slight modification on pixel values to generate the slightly changed image. In the experiment, only about 1% of the pixels in the image would be changed by adding 1 to the value. Then both images will be encrypted by the secure key. Table 3.1 gives the mean NPCR and UACI values for our encryption system. The mean NPCR of our encryption system is higher than that of [75], which is 57.35%. Besides, the average UACI mentioned in [75] is less than 10% which is lower than our model. That main because the logistic map is sensitive to its initial value and different images are confused by different logistic maps.

Table 3.1: Mean NPCR and UACI of cipher-images.

	NPCR%	UACI%		NPCR%	UACI%
Kodim01	99.13	15.35	Kodim13	99.44	22.29
Kodim02	98.02	11.03	Kodim14	99.44	24.11
Kodim03	99.15	21.18	Kodim15	99.53	32.22
Kodim04	99.17	15.26	Kodim16	99.25	18.96
Kodim05	99.38	21.04	Kodim17	99.35	21.18
Kodim06	99.37	28.85	Kodim18	99.26	19..96
Kodim07	99.18	18.05	Kodim19	99.33	22.12
Kodim08	99.49	28.16	Kodim20	91.57	34.58
Kodim09	99.19	17.75	Kodim21	99.13	17.68
Kodim10	99.11	17.32	Kodim22	99.23	18.95
Kodim11	99.11	17.15	Kodim23	99.36	24.51
Kodim12	99.05	17.01	Kodim24	99.22	21.88

3.5.5 Time Efficiency Analysis

In this part, the encryption efficiency of our proposed scheme is analyzed. The tested 24 images are from the publicly Kodak dataset. The size of images in this dataset is 512×768 or 768×512. The mean encryption speed of different encryption schemes is shown in Table 3.2.

Table 3.2: Encryption efficiency with different schemes.

	Proposed Model
Speed(s)	14.78

3.6 Retrieval Accuracy on Encrypted Images

The retrieval performance evaluation is conducted on the image database Corel-1k [74] which contain 1k color JPEG images in 10 categories, and some of them are shown in Figure 3.9.

Figure 3.10: Eight images from Corel dataset with different classes.

Here, we analyze the retrieval performance of proposed algorithm compared with other schemes by using Top-k precision and precision–recall curve. The precision and recall rate are defined as follows:

$$\text{precision} = \frac{\text{Num(relevant images in returned images)}}{\text{Num(returned images)}} \times 100\% \tag{3.7}$$

$$\text{recall} = \frac{\text{Num(relevant images in returned images)}}{\text{Num(returned images in database)}} \times 100\% \tag{3.8}$$

Figure 3.10 shows the precision of retrieval accuracy of different encrypted image retrieval schemes when implementing Top-k search (k=5, 10, 15, 20). Unencrypted images and encrypted images using the same similarity measurement. Here all schemes are performed on unencrypted Corel-1k image database. In [75], the images are encrypted by pixel value confusion and pixel position shifting, while value replacement and position scrambling are conducted on images in [76]. For the average performance, our proposed method is close to the results with unencrypted images due to the spatial information in encrypted images.

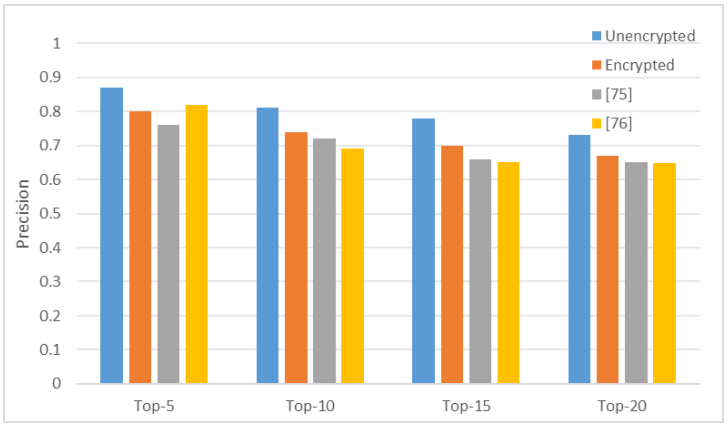


Figure 3.11: Top-k precision (k=5, 10, 15, 20).

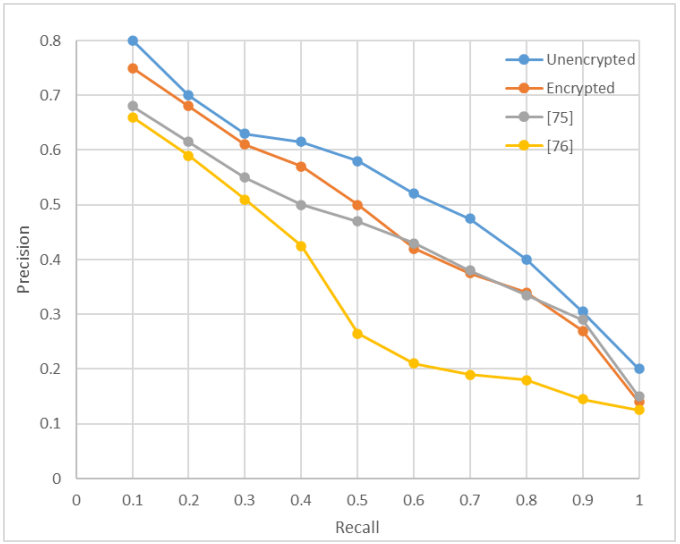


Figure 3.12: Precision-recall curve.

Figure 3.11 shows the precision–recall curve. And when recall = 0.1, the retrieval accuracy of the proposed encryption and retrieval model is closest to the performance with unencrypted images. When the recall rate is between 0.6 and 1, the retrieval accuracy of [75] is slightly higher than the proposed scheme. But the retrieval accuracy of proposed scheme is always better than that of [76]. Moreover, the area under curve (AUC) is calculated for futher measuring the overall retrieval accuracy. The AUC of curve Unencrypted, Proposed, [75] and [76] are 0.4813, 0.4257, 0.3845 and 0.2962 respectively.

3.7 Summary

In this chapter, a new searchable encryption method based on pixel is proposed to achieve privacy protection and it can support similarity measurement on encrypted domains. The

features are extracted from the content of images. And to protect the privacy security, two-level sequence permutation is conducted on pixels in each 8×8 block. Using block based permutation method on image confusion can guarantee local feature extraction, further improving the image security and retrieval accuracy. With the proposed encryption method, the feature can be directly extracted from encrypted blocks. The problem of similarity search on encrypted images is defined based on the local histogram. And the retrieval accuracy and image security of proposed method are tested. Moreover, we discuss the factors affecting the performance of algorithm. From the security analysis, we can find that the security level is not so strong since block based encryption operation cannot remove spatial relationships well. In the next chapter, we will explore the possibility of using value substitution for further security and make full use of DCT coefficient to obtain better encryption and retrieval perfromence.

Chapter 4 Encrypted JPEG Image retrieval based on DCT coefficients

In this chapter, we tend to extract the features for retrieval from the frequency domain which would consume less computation and communication resources. And for the encrypted image retrieval, the Discrete Cosine Transform (DCT) domain can be a potential domain in extracting visual information directly from cipher-images. Here, DCT coefficients are utilized for obtaining feature vectors. The encryption operation used is performed on JPEG images since JPEG is one of the most popular image standards in daily life. The images are entirely encrypted along with format compliance and file size preservation. And Due to the successful application of deep learning in multiple visual areas, we use the deep architecture in network to get more effective features for encrypted image retrieval. The novel scheme for encrypted image retrieval is proposed based on learning network. For a given encrypted query image, the server exploit the DC and AC coefficients as the inputs of the network to obtain the image descriptors for retrieval to measure the similarity between the encrypted query image and database image, without first decrypting images. And we adopt the Siamese architecture for metric learning since the learned image embedding can help the Euclidean distance captures the similarity well. Finally, the encrypted images with plaintext content similar to the query image are returned to the user. The major contributions of our method are summarized as follows:

1) We encrypt images by coefficient value substitution and intra-block pixel permutation, which provide high security and compression efficiency. With the proposed encryption method, the feature can be directly extracted from the frequency domain.

2) We proposed a method of learning in the frequency domain which would consume less computation and communication resources. And we modify the existing ResNet model to ensure DCT coefficients as input.

3) We use a Siamese network that combines three streams with a triplet loss to optimizes the weights of our feature extraction model to produce representations well suited for a retrieval task. During training, hard non-matching (negative) examples and hard matching (positive) examples are learned to enhance the representation.

The rest of this chapter is organized as follows: Section 4.1 gives an overview of the proposed encryption and retrieval model. Section 4.2 explains the implementation details of DCT based image encryption and compression model. Section 4.3 introduces the learned retrieval network, including deep feature generation and similarity loss training. Performance evaluations on the compression efficiency and security will be given in Section 4.4 and 4.5, respectively. Section 4.6 provides the discussion on retrieval performance. And Section 4.7 gives the summary of this chapter.

4.1 Overall of the Proposed Scheme

The proposed scheme in this paper includes two parts: the image encryption method and the image retrieval method. For a given image database, user needs to generate a set of secret keys first. Then, images in the database are encrypted by coefficients substitution and intra-block permutation during compression. After that the encrypted image database is uploaded to the cloud server. To retrieve similar images, user submit an encrypted query image data to the cloud server. Once receiving the query request, all features are extracted from the encrypted database by cloud server through a learned network. Similar images are returned to the user as search results, and the user can decrypt the results using the secret key.

4.2 DCT based Image Encryption and Compression Model

JPEG is one of the most common image compression standards. And it is a kind of lossy compression method which will remove the high frequency information which is not visually obvious. In this paper, our scheme is conduct on JPEG image. According to JPEG standard [14], a color JPEG image will be divided into 8×8 non-overlapped blocks and 64 DCT coefficients (one DC coefficient and 63 AC coefficients) are generated. When transmitting, DCT coefficients are converted into binary bits. Here, we conduct encryption operations on DCT coefficients to enhance security while maintaining JPEG format compatibility. Also, for further image retrieval, a searchable encryption scheme is necessary. The proposed encryption algorithm is shown below.

Algorithm 1: Encryption Algorithm

1: Load original image I and the predefined key K

2: $KS1 \longleftarrow \mathrm{BLAKE256}(K)$

3: Get YCbCr components from chroma sampling for each image

4: **for** each 8×8 non-overlapped block of component **do**

5: 8×8 DCT transformation and get quantized DCT coefficient matrix
6: end for
7: Value substitution on all DC coefficients
8: Intra-block permutation of DC and AC coefficients with $KS1$
9: Generate the encrypted bit-steam and transmit it to the cloud

4.2.1 Keystream Generation Process

In our encryption scheme, the BLAKE256 hashing algorithm is used to generate the encryption keystream by taking the predefined key K as input. Here, the predefined key K is a random sequence with fixed-length which is used for value substitution and generating $KS1$. The predefined key K is the initial seed to generate. For shuffling all DC and AC coefficients, the new index can be generated by the Fisher-Yates Shuffle algorithm using the random key stream $KS1$.

4.2.2 Value Substitution

As presented above, there are two steps in the image encryption scheme including coefficients value substitution and intra-block permutation. Here, we present a sub-algorithm to specify the process of value substitution. After substituting values, the same value at different positions can be substituted with the same value, which helps to improve retrieval performance. The new value is sensitive to the change in the original pixel value, which can help resist differential attacks.

Algorithm 2: ValueSubstitution

Input: DC coefficients
Output: Encrypted DC coefficients
1: Generate a random sequence S from rang [-1024,..,1024]
2: Denote p_i as *i*th value in DC coefficient matrix
3: Denote $p_i^{'}$ as *i*th value in the encrypted DC coefficient matrix
4: **for** each p_i **do**
5: $p_i^{'} \leftarrow S_{p_i}$
6: end for

4.2.3 Intra-block Permutation

For intra-block permutation, we generate a random permutation index to shuffle the intra-block coefficient position to further improve security.

Algorithm 3: IntrablockPermut

Input: Encrypted DC coefficients and AC coefficients, the random sequence $KS1$

Output: Encrypted DC and AC coefficients

1: Denote p_i as ith value in coefficient matrix

2: Denote $p_i^{'}$ as ith value in encrypted coefficient matrix

3: Perform Yates Shuffle algorithm where the random integer in each loop is from $KS1$ to generate a new index G for permutation

4: **for** each p_i **do**

5: $p_i^{'} \leftarrow p_i[G]$

6: end for

4.3 Image Retrieval Model

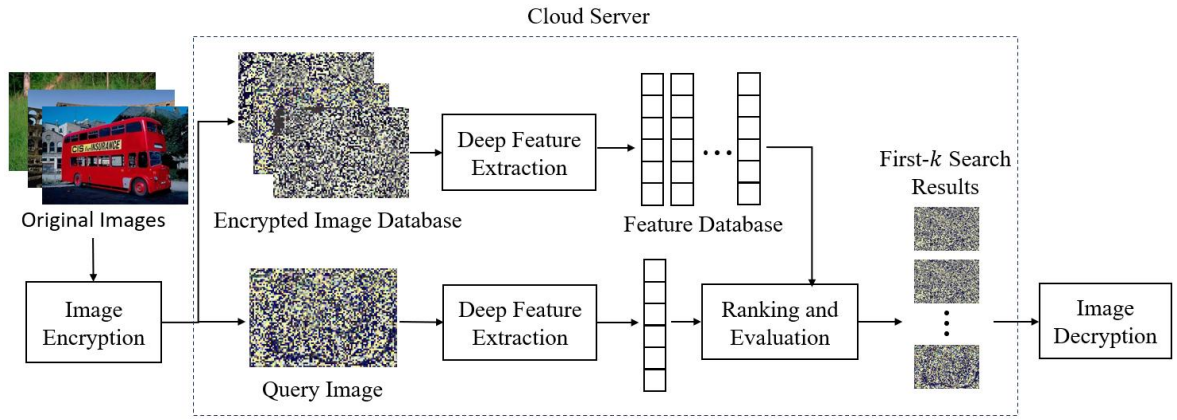


Figure 4.1: Framework of the content-based encrypted image retrieval.

In our encrypted image retrieval model, users encrypt images before uploading them to the cloud and generating an encrypted image database to preserve privacy. Besides the image encryption, computation and storage costs are all outsourced to cloud servers. After storing the images, users may want to obtain the images with similar content to a query image and then send the encrypted query image to the cloud. The feature extraction and search operation are finished by the cloud server. Here, the feature extraction operation is conducted on the frequency domain and the deep features are generated from the DCT coefficients directly. After

searching, several images are sent to the users, and users can use the corresponding key to decrypt the search results.

4.3.1 Feature Extraction

In this chapter, we propose a feature extraction model that is learning in the frequency domain. And to meet the size requirement of general network, a data pre-processing is introduced as the input data size pruning scheme. Due to the successful application of deep learning in multiple visual areas, we tend to realize more efficient retrieval with the help of deep architecture. For using the learning method, original images are usually pre-processed on a CPU and then transmitted to graphics processing units for further processing. In our system, the compressed and encrypted bitstreams are transmitted from users to the cloud server. Thus, bandwidth requirements for communication between the CPU and graphics processing units will be addressed since high-resolution RGB images are compressed. Also, data security can be protected well with the encryption scheme mentioned in Section 4.2.

In our method, images are pre-processed on a CPU for encryption. And after encryption operation, the encrypted DCT coefficients are grouped into multiple frequency channels as the inputs of the feature extraction model. Here, we demonstrate that minimal modification on existing deep models developed in the spatial domain can suit the inputs from the frequency domain. Specifically, we remove the original deep input layer and reserve the remaining deep architecture. In our experiment, we chose ResNet as the backbone of the feature extraction model. Since our encryption scheme is based JPEG images. The components with the same frequency in all the 8×8 blocks are grouped into one channel and then each color component provides 64 channels, with a total of 192 channels in the frequency domain. For a given H×W×C color image, the input frequency feature shape becomes H/8×W/8×192 after converting to the frequency domain. Thus, we skip the normal input layer and max-pooling operator, and set the first residual layer as the input layer. In Figure 4.2, we take 64 channel as the example to show the modification. After that, the number of input channels in deep architecture is modified to fit the dimensions of the DCT coefficient inputs. In this way, the modified deep model is similar to the original deep network in terms of parameter count and computational complexity.

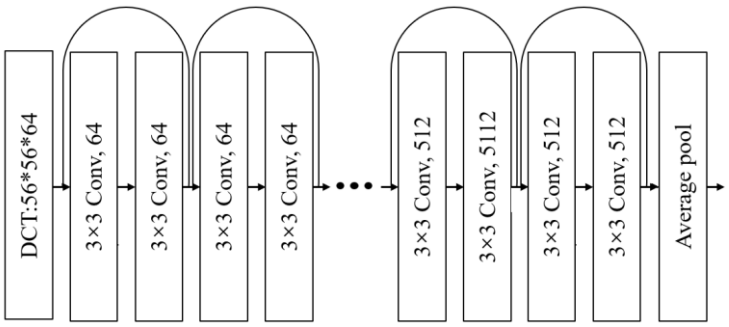


Figure 4.2: Feature extraction network.

4.3.2 Learning with Ranking Loss

Retrieval is a kind of metric learning problem and two-branch Siamese or triplet network is one of the typical solutions. So for better decreasing the distance between similar image samples, we employ the Siamese architecture with triplet loss to enhance the representation. In Siamese architecture, each branch is a clone of the other and the parameters of each branch are shared. And the matching and non-matching pairs (i, j) are employed for training. Thus, the contrastive loss function can be given as follows:

$$L = \begin{cases} \mathbb{E}[\|f(i) - f(j)\|^2] & if\ Y(i,j) = 1 \\ \mathbb{E}[max\{0, \tau - \|f(i) - f(j)\|\}^2] & if\ Y(i,j) = 0 \end{cases} \tag{4.1}$$

For image pair selection, positive images and negative images are selected for training loss.

Positive images: positive examples are selected from clusters where query image is also there. The image that has the lowest descriptor distance to the query is chosen as positive.

Negative images: negative examples are selected from clusters different than the cluster of the query image, as the clusters are non-overlapping. We choose hard negatives which is the non-matching images with the most similar descriptor.

4.4 Performance Evaluation

In this section, experiments will be conducted to evaluate the compression performance and the perceptual security of our proposed encryption and compression scheme. The performance evaluation is conducted on the publicly available Kodak dataset [72] with 24 high-quality images, and some of them are shown in Figure 4.3. In Figure 4.4, the encryption images of test set are shown.

Figure 4.3: Test images from Kodak dataset.

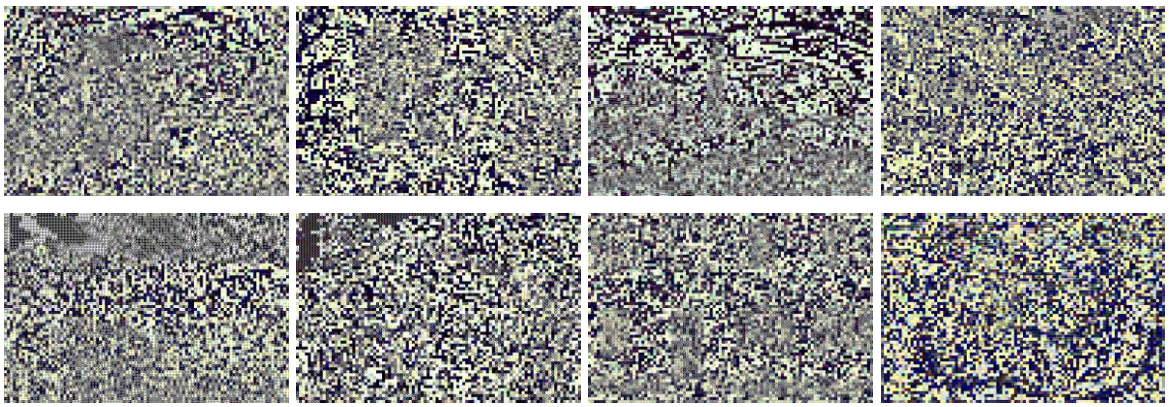


Figure 4.4: Encryption images of test images.

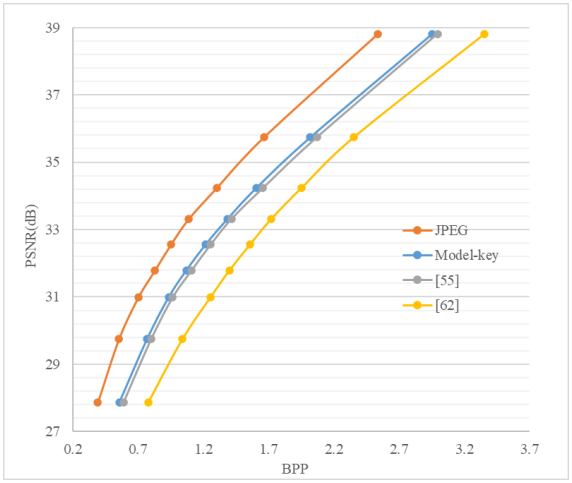


Figure 4.5: Rate distortion performance for different methods.

In the perceptual security evaluation of encryption methods, the peak signal-to-noise ratio (PSNR) is used for evaluating the compression performance. In Figure 4.5, when the encryption key is provided for decryption, the PSNR values of our proposed model can illustrate the compression efficiency of our model. The closer the curve is to JPEG, the higher

the compression efficiency. Similar to [55], coefficient permutation is used in our scheme, which affects the compression performance. And in [62], more bits are needed since the DC Huffman table is shuffled.

4.5 Security Analysis

This section is going to discuss various cryptographic attacking methods, such as ciphertext-only attack, differential attack, and statistical attack. And the robustness against those attacks will be evaluated to analyze the security of our scheme.

4.5.1 Ciphertext-only Attack

To defend against the ciphertext-only attack, the key space of the cryptosystem should be large. In our model, the parameters of the learning network are given. The 256-bit encryption key K is given and then we obtain a $2^{(256)}$ keyspace which is lager than the theoretical requirement with $2^{(100)}$. So it is tough for the attacker to break down and our encryption system can handle this attack.

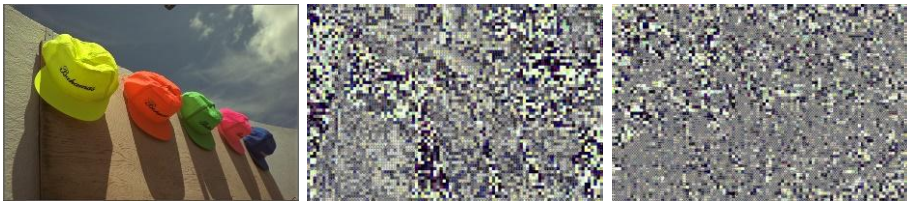
4.5.2 Key Sensitivity Analysis

Considering to privacy, a cryptosystem needs to be highly sensitive to the encryption and decryption keys used. Here, we use the plain images ‘kodim03’as examples. The high key sensitivity level can be demonstrated in two parts:

1) A completely different ciphertext image would be generated for the same plain image if the encryption keys used change slightly.

2) The encrypted image should not be decrypted when a key having minor change to the encryption key is used.

In the first case, we make a minor change in encryption key K to generate the new key stream $KS1$. We then generate two encrypted images using the two different keys for the same input image. And the encrypted images are shown in Figure 4.6(b) and 4.6(c). It is clearly seen, from both the MSE figure and the difference image, that the encrypted images obtained by two slightly different keys are very different, and our proposed model fulfils the first case of key sensitivity.



(a) (b) (c)

Figure 4.6: Comparison of encrypted images with different keys. (a) plain images, (b) cipher image through the original key, (c) cipher image through the new K.

For the second case, the original key and the new key generated are utilized to decrypt the same cipher-image encrypted with the original key. The decrypted images using the two different keys are shown in Figure 4.7(b) and 4.7(c). It is seen that only the original key can recover the original image.

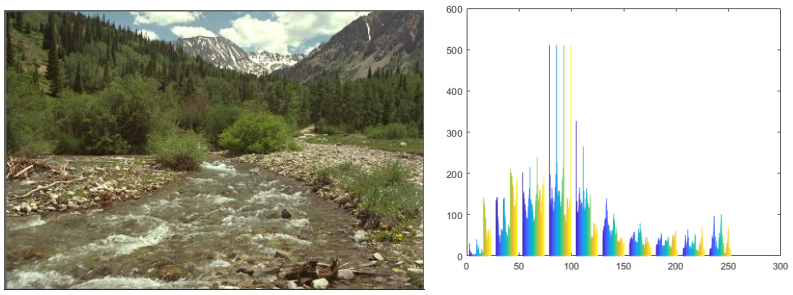
(a) (b) (c)

Figure 4.7: Decryption with different keys. (a) original image, (b) decrypted image using the original key K, (c) decrypted image using the new K.

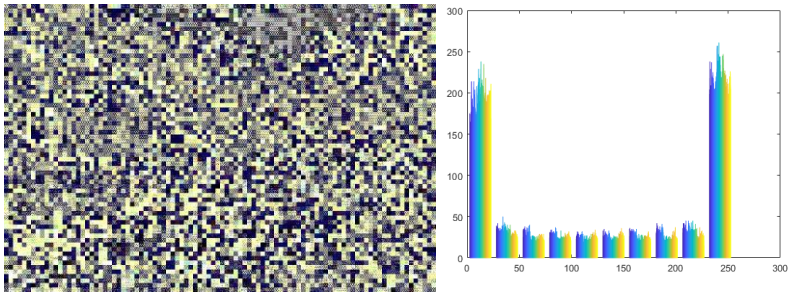
4.5.3 Statistical Attack

In statistical attacks, attackers utilize the high correlation between pixels and obtain original images through the predictable relationship between plain images and encrypted images. Here, we use the histogram and correlation chart to illustrate the correlation. And the histogram of image 'kodim13' and corresponding encrypted images are given in Figure 4.8. It is seen that there are large differences between the histograms of images before and after encryption.

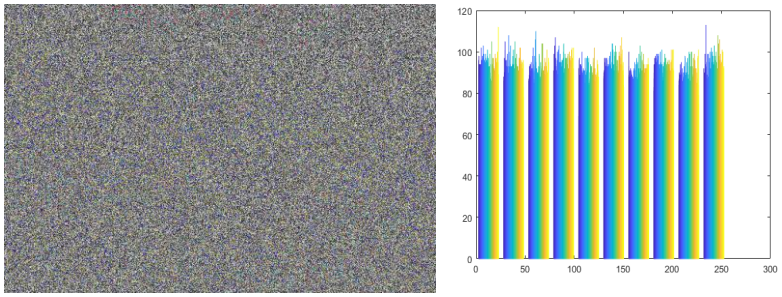
We can find that the shuffling operation can extremely decrease the pixel's correlation. And the substitution can further reduce the correlation. From Figure 4.8, our method and [45] does not show a uniform distribution since there are still some correlations among pixels, while [30] can achieve the uniform distributed histogram which reveals the excellent property to resist the statistical attack.



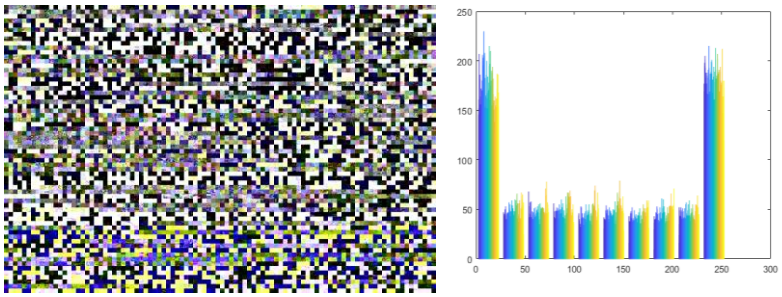
(a)



(b)



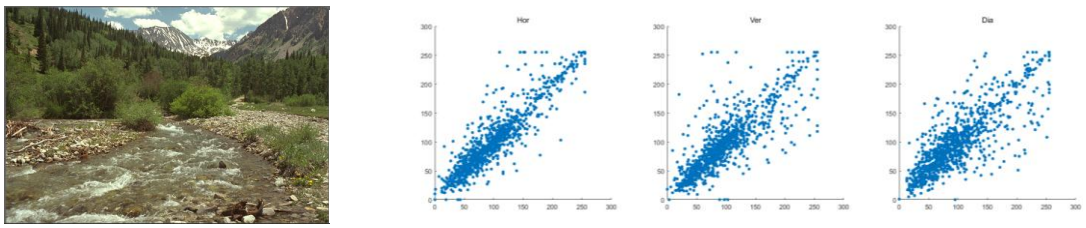
(c)



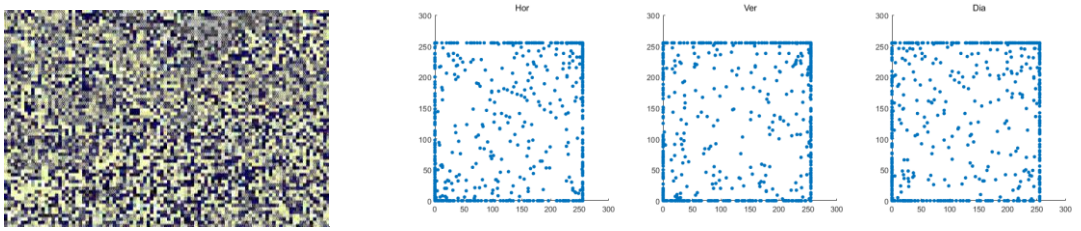
(d)

Figure 4.8: Histogram charts of plain-image and cipher-image. (a) plain-image (b) cipher-image of the proposed method, (c) Ref [30], (d) Ref [45].

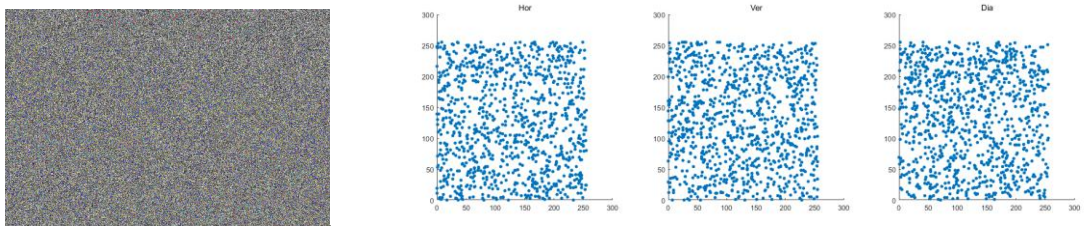
And Figure 4.9 shows the correlation charts of image ‘kodim13’ and ciphertext images under our encryption model, [30] and [45]. A similar shuffling operation is also utilized in [45]. But the results show that the shuffling operation in our model achieves better performance on decreasing the pixel’s correlation compared with [45].



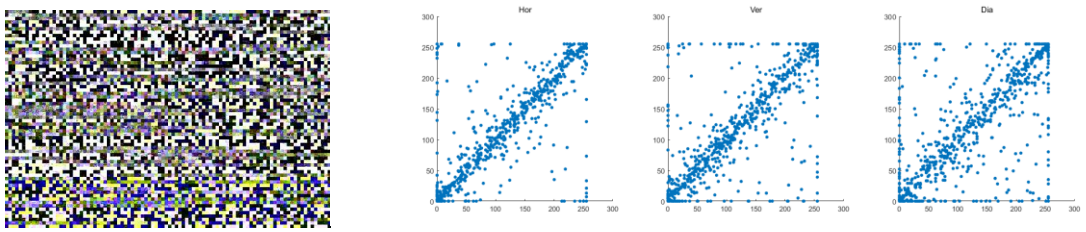
(a)



(b)



(c)



(d)

Figure 4.9: Correlation charts of images before and after encryption. (a) original image (b) encrypted image from the proposed model, (c) Ref [30], (d) Ref [45].

4.5.4 Differential Attack

For differential attack evaluation, the encryption system needs to be sensitive to changes in the image, and changes in plain images should cause different encryption images. The common criteria NPCR and UACI are used to measure the degree of image change. A robust encryption system needs to get a high value of NPCR and the value of UACI needs to close to 33%.

To start the evaluation, we slightly modify some random pixel values to generate the slightly changed image. In the experiment, only about 1% of the pixels in the image would be changed by adding 1 to the value. Table 4.1 gives the mean NPCR and UACI values for our encryption system. From the table, the proposed model has the defence capability against the differential attack to a certain extent, while the NPCR and UACI values are all almost zero for the encryption method in [30], indicating the low property.

Table 4.1: NPCR and UACI of cipher-images.

	Proposed Model		Ref [30]		Ref [45]	
	NPCR%	UACI%	UACI%	UACI%	NPCR%	UACI%
Kodim01	97.57	35.93	0.01	8.22e-05	96.62	39.32
Kodim02	98.22	38.31	0.01	8.98e-05	97.97	36.48
Kodim03	98.43	34.38	0.01	9.74e-04	98.91	28.36
Kodim04	96.68	32.35	0.01	7.12e-05	98.18	38.23
Kodim05	97.25	34.96	0.01	7.58e-05	97.42	36.31
Kodim06	94.87	36.78	0.01	6.45e-05	96.86	36.48
Kodim07	97.17	29.66	0.01	7.31e-04	97.91	29.37
Kodim08	98.48	36.78	0.01	9.74e-04	98.62	38.23
Kodim09	98.26	34.86	0.01	7.11e-05	98.02	37.82
Kodim10	98.45	34.92	0.01	7.58e-05	97.23	36.48
Kodim11	98.54	37.22	0.01	6.45e-04	97.93	29.46
Kodim12	98.43	37.05	0.01	5.91e-05	99.21	38.23
Kodim13	98.55	37.40	0.01	9.74e-04	98.02	36.62
Kodim14	98.82	37.62	0.01	4.12e-04	97.24	36.48
Kodim15	98.37	34.14	0.01	7.52e-05	97.67	29.33
Kodim16	97.25	29.43	0.01	5.45e-04	98.62	36.23
Kodim17	98.68	35.17	0.01	8.78e-05	98.02	35.31
Kodim18	98.73	37.57	0.01	9.74e-04	97.29	36.49
Kodim19	97.85	29.42	0.01	9.14e-04	97.95	28.34

Kodim20	97.25	36.49	0.01	7.68e-05	97.62	28.23
Kodim21	98.83	35.65	0.01	7.45e-04	98.02	37.62
Kodim22	98.04	36.88	0.01	8.12e-04	97.25	36.58
Kodim23	97.21	29.32	0.01	7.88e-05	97.02	28.96
Kodim24	98.78	36.31	0.01	6.45e-04	99.25	38.23

4.5.5 Time Efficiency Analysis

In this part, the encryption efficiency of our proposed scheme is analyzed. The size of 24 images from the Kodak dataset is 512×768 or 768×512. The mean encryption speed of different encryption schemes is shown in Table 4.2. Ref [30] needs the least computational time since the modulo-256 encryption method is low complexity. And the time of proposed method is mostly spent on permutation operations which is important for security, while Ref [45] has the same problem.

Table 4.2: Encryption efficiency with different schemes.

	Proposed Model	Ref [30]	Ref [45]
Speed(s)	4.78	0.29	4.33

4.6 Retrieval Accuracy on Encrypted Images

The proposed methods are trained on the Nvidia GTX 2080Ti. A sub-dataset from ImageNet 2012 Large-Scale Visual Recognition Challenge dataset [77] is used for training which contains more than 80k images with 100 classes. We choose ResNet-34 [78] as the backbone since the residue blocks and depthwise separable convolutions are widely used in deep models. The stochastic gradient descent (SGD) optimizer is used with an initial learning rate of 0.1, a momentum of 0.9, and a weight decay of 4e-5. And the learning rate decays by 0.1 every 50 epochs. The mean and variance of the DCT coefficients for each of the 192 frequency channels separately on all the training images are calculated for normalization.

In Table 4.3, we compare the proposed feature extraction model with the backbone. With the comparison, the proposed frequency-domain learning can extract features from encrypted data efficiently. The experiments in Table 4.3 is performed on a valid set of training dataset.

Table 4.3: Accuracy of feature extraction model with different inputs.

Inputs	Channels	Size Per Channel	Top-1	Top-5
RGB	3	224×224	74.75	91.45
YCbCr	3	224×224	74.34	91.18
DCT	192	56×56	74.19	91.02
Encrypted DCT	192	56×56	66.78	84.26

The retrieval performance evaluation is conducted on the image database Corel-1k [74] which contains 1k color JPEG images in 10 categories. Here, we analyze the retrieval performance of the proposed algorithm compared with other schemes by using Top-k precision and mean average precision (mAP) [79]. Figure 4.10 shows the precision of retrieval accuracy of different encrypted image retrieval schemes when implementing Top-k search (k=5, 10, 15, 20). The result of unencrypted images with the proposed retrieval mode is also shown. From Figure 4.10, it is obvious that ranking loss can improve retrieval performance. In [75], the images are encrypted by pixel value confusion and pixel position shifting, while value replacement and block permutation are conducted on images in [80]. For the average performance, our proposed method is close to the results with unencrypted images, which makes it seem that the effective feature is captured in the frequency domain. As for mAP, our scheme also has better accuracy than other unsupervised schemes, including histrogm[56] and bag-of-words [81].

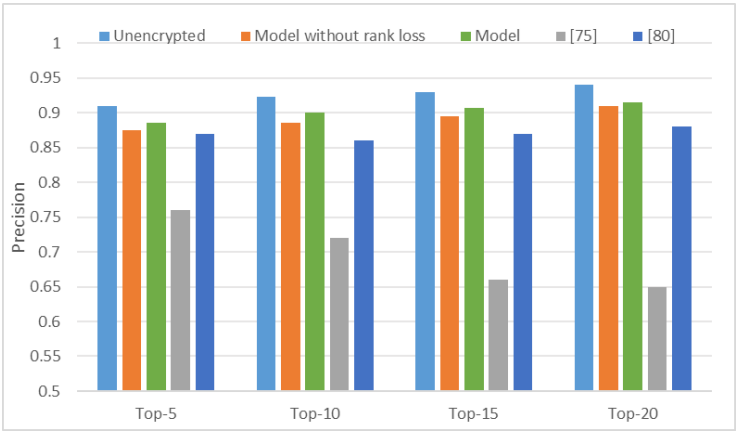


Figure 4.10: Top-k precision (k=5, 10, 15, 20).

Table 4.4: Retrieval accuracy of different methods.

	Model	[56]	[81]
mAP	54.41	46.37	49.09

The quality of images is controlled by QF for JPEG images. So to test the influence of image quality, we performed experiments on the Corel10k dataset with different compression rates. As shown in Table 4.5, the retrieval model performs better with higher QF. And for experiments in Table 4.4, QF is set to 90.

Table 4.5: Retrieval accuracy with different compression degrees of cipher images.

QF	20	50	70	90
mAP	35.71	46.39	50.87	54.41

4.7 Summary

In this chapter, a new encrypted image retrieval method based on deep learning is proposed. The encryption operations are conducted on JPEG images during the JPEG compression process by using coefficients substitution and intra-block permutation. And the encrypted images are entirely encrypted along with format compliance. For the retrieval part, our proposed learning approach utilizes ResNet-34 as the backbone to extract features while accepting encrypted DCT coefficients as input. Extracting the features for retrieval from the frequency domain would consume less computation and communication resources. And a Siamese architecture with triplet loss is used to produce a global representation that is well-suited to image retrieval. Finally, the encrypted images with plaintext content similar to the query image are returned to the user. Experiment results show that our method can achieve higher accuracy than other related schemes and meanwhile further improve security.

Chapter 5 Deep Compression and Encryption for Image Retrieval

Learning-based methods have obtained promising results in many areas and deep learning models are expected to be the next-generation optimal image compression solutions. In this chapter, a new image compression and encryption framework is proposed which integrates encryption algorithms with a deep-learning compression network. This new model is developed by applying a deep learning network which could be an early attempt to introduce end-to-end learning to the SCE system. After encryption, the ciphertext would be sent to the proposed similarity network.

We formulate the task of compression and encryption as the constrained optimization problem which will minimize the expected length of the bitstream and maximize the visual quality of reconstructed images. Here, we utilize the deep network to solve this optimization problem. Differing from the CS-based compression method using single-layer transform, the deep model gets the compressed representation from the layer-by-layer learning and can learn pixel representation well through data-driven supervised learning, which can be a more general compression scheme. We adopt deep AE as the learning network in our proposed joint image compression and encryption framework. That is mainly because the AE-based network achieves a good performance in areas of domain transformation and image reconstruction through rate-distortion loss and flexible probability density estimation [15]. And the AE-based network in our work can convert an image into a compact representation, and its process is similar to traditional compression with linear transformation. Referring to the usual SCE methods, the encryption operations are embedded into the compression process in our approach. We offer to conduct the shuffling operation on deep feature maps generated by analysis transform during encoding. The scrambled feature maps are obtained by scrambling the orders of all three dimensions with the secure key. Then, the side representations are extracted from scrambled feature maps through hyper-network which will be used to generate the learned entropy model for compression. To achieve a higher level of visual security after decoding, the parameters of synthesis transform are replaced by a new parameter matrix which is the result of dividing the original parameter matrix with the logistic map. In our compression and

encryption model, the encryption key for shuffling operation and controlling the logistic map is generated from plain images by BLAKE256 hash algorithm. Then the key is embedded in the side representations during the quantization stage in the hyper-network controlled by another secret key. And it will be extracted from the side representations only when decrypting.

In our learning model, when deep features are encrypted and decrypted, the plain image can still be reconstructed through the decoding sub-network that meets the basic requirements of a cryptographic system. And experiments in [82] have proved that a larger kernel size can be conducive to coding efficiency. But a larger kernel size will cause a smaller size of the feature map, and it can decrease the complexity of encryption algorithm since permutation operation is conducted on feature maps. Thus, we chose the medium size kernel according to the experimental results. And for the learned entropy model, the parameter estimation part is used to extract more efficient parameters for recovering scrambled representations from the bitstream. The attention scheme is introduced here to help learn entropy model since the correlations among neighboring pixels can be hard to be exploited after permutation operation on feature maps.

Unlike other deep encryption and compression methods using networks just for compressing the encrypted image or the original one, the network in our work is used for compressing and encrypting the data simultaneously, and an end-to-end compression and encryption framework is established. And different from deep encryption works, like [83] using Cycle-GAN to transfer the original image to the encrypted one, our method focuses on joint compression-encryption methods working together with the image compression scheme. Meanwhile, different from other SCE schemes, our encryption operations are conducted on the semantic features learning from the network, and the encryption/decryption layer added will not impact the network's training. Extensive experiments conducted on the Kodak dataset show that the proposed encryption scheme can resist various attacks with high compression efficiency. And the encrypted representation is sent to the proposed similarity network to improve retrieval accuracy.

The major contribution of our work can be summarized as follows:

1) A novel joint compression-encryption model is developed by applying a deep learning network which could be an early attempt to introduce end-to-end learning to the security system. The shuffling operations are conducted on deep feature maps.

2) For a higher level of visual security, part parameters of the network are replaced by a new parameter matrix based on a logistic map controlled by a secret key.

3) The encryption key of the system is derived from the image content, which will be embedded in the deep feature vectors with a fixed key to save the cost of sending the key for different images.

4) An attention scheme is exploited in estimating parameters to achieve more effective compression to learn the deep model from the scrambled feature maps.

The rest of this chapter is organized as follows: Section 5.1 gives an overview of the proposed deep compression and encryption model. Section 5.2 explains the implementation details for realizing compression and encryption jointly. Section 5.3 introduces the retrieval network. Performance evaluations on the compression efficiency and security will be given in Section 5.4 and Section 5.5, respectively, with a comparison with other schemes. And the discussion on retrieval performance is introduced in Section 5.6. Section 5.7 provides the summary.

5.1 Deep Network Model for Joint image Compression and Encryption

As shown in Figure 5.1, our joint model introduces encryption techniques into deep compression, which contains two autoencoders. The core autoencoder consists of analysis transform, quantizer, synthesis transform, arithmetic encoder and decoder. It is designed for learning the quantized latent representation of images to produce a compact bitstream for compression. The analysis transform here is composed of three parts: convolution, down-sampling and generalized divisive normalization (GDN), while the synthesis transform consists of convolution, up-sampling and inverse GDN [15]. After the analysis transform, permutation operations are conducted on the deep feature maps for encryption and then the scrambled feature maps will be input to the hyper-network H. And the network H is the sub-autoencoder and can learn a probability model over latent representation. The parameter estimation module is responsible for efficiently transforming the hyper-latent representations into the parameters of the Gaussians which make sure the parameters from H can be appropriate for the core autoencoder. To obtain the compressed and encrypted images, the deep representations are processed by the synthesis transform while the parameters of synthesis transform have been substituted. And in Figure 3, the solid and dashed lines denote the compression and encryption process and the decryption and decompression process of our proposed model, respectively.

In the encryption process, BLAKE256 hash function is utilized to generate the encryption key $Key1$ based on the input image content, while the pseudo-random stream $KS1$ is derived from $Key1$ using BLAKE256 hash function. Then the key stream KS1 is used to control all the encryption/scrambling operations in the proposed system. To save the cost of transmitting, $Key1$ will be embedded into deep feature maps by the embedding key $Key2$. $Key2$ is predefined and produces the pseudo-random stream $KS2$ for embedding through BLAKE256 hash function. And this embedding key will be shared for different plain images and transmitted by the general key transmitting methods. The permutation layer for shuffling the orders of the deep feature maps is added after the analysis transform layer, controlled by the content sensitive encryption key $Key1$ for a different input image. We conduct permutation operations on the third dimension k of feature maps y firstly, then on the first and second dimension i,j of y with $KS1$, and finally, get the encrypted feature maps y_e. After the quantization stage, the encrypted deep representations are compressed into a bitstream using an arithmetic encoder. Meanwhile, the side string could be extracted by hyper coding and quantization through hyper-network. And the encryption key $Key1$ will be embedded into the side string by $KS2$ at the quantization stage. When transmitting, the string from the encrypted representation y_e would be combined with the side string to produce the final compressed and encrypted bitstream. For higher visual security, the parameter matrix of the synthesis transformation is replaced by a new parameter matrix which is the result of dividing the original parameter matrix with the logistic map controlled by the encryption key $Key1$. Then through the synthesis transformation with modified parameters, the plain image will finally be encrypted and compressed. If we add the decryption operation, only the compressed image will be produced, regarded as the decrypted image. And the key for decryption will be extracted from the side string controlled by $KS2$ only when decrypting. Also, the logistic map is generated by $Key1$ for recovery of the parameters of synthesis transformation.

In this learning-based method, quantization is approximated by a uniform noise to generate $\hat{y}_e$. The Gaussian mixture model is used for entropy coding. After parameter estimation, the mean and scale parameters $\hat{\mu}, \hat{\sigma}$ of the Gaussian Likelihoods can be generated. The quantized representation will be compressed by the lossless arithmetic coder with the probability model $p_{\hat{y}}(\hat{y})$.

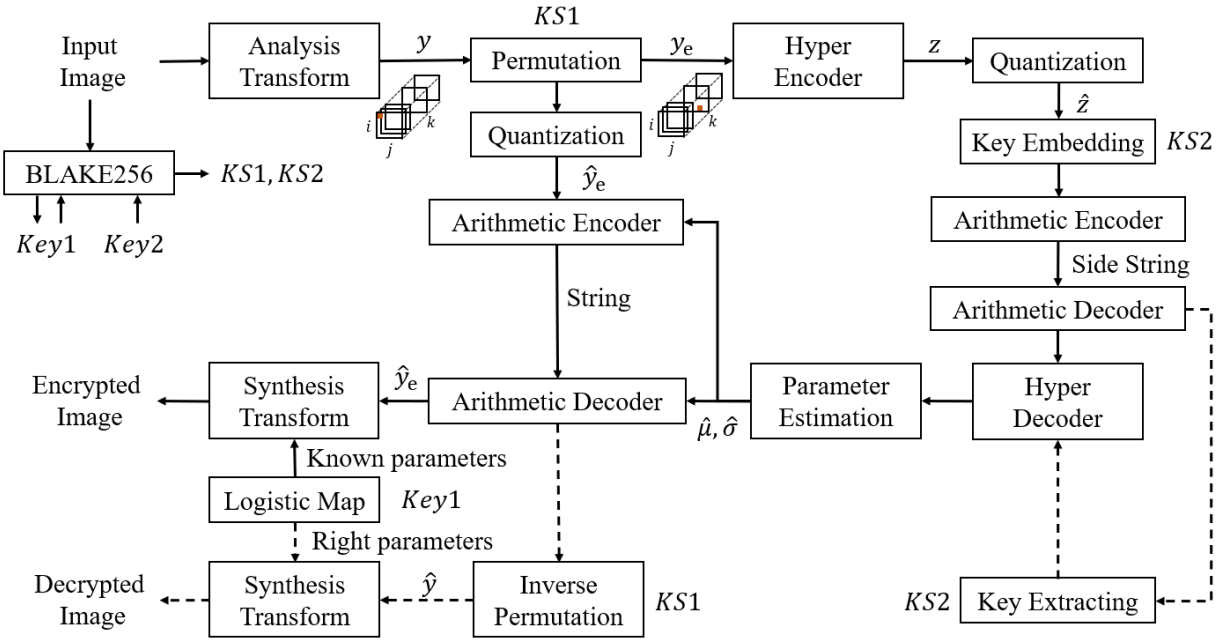
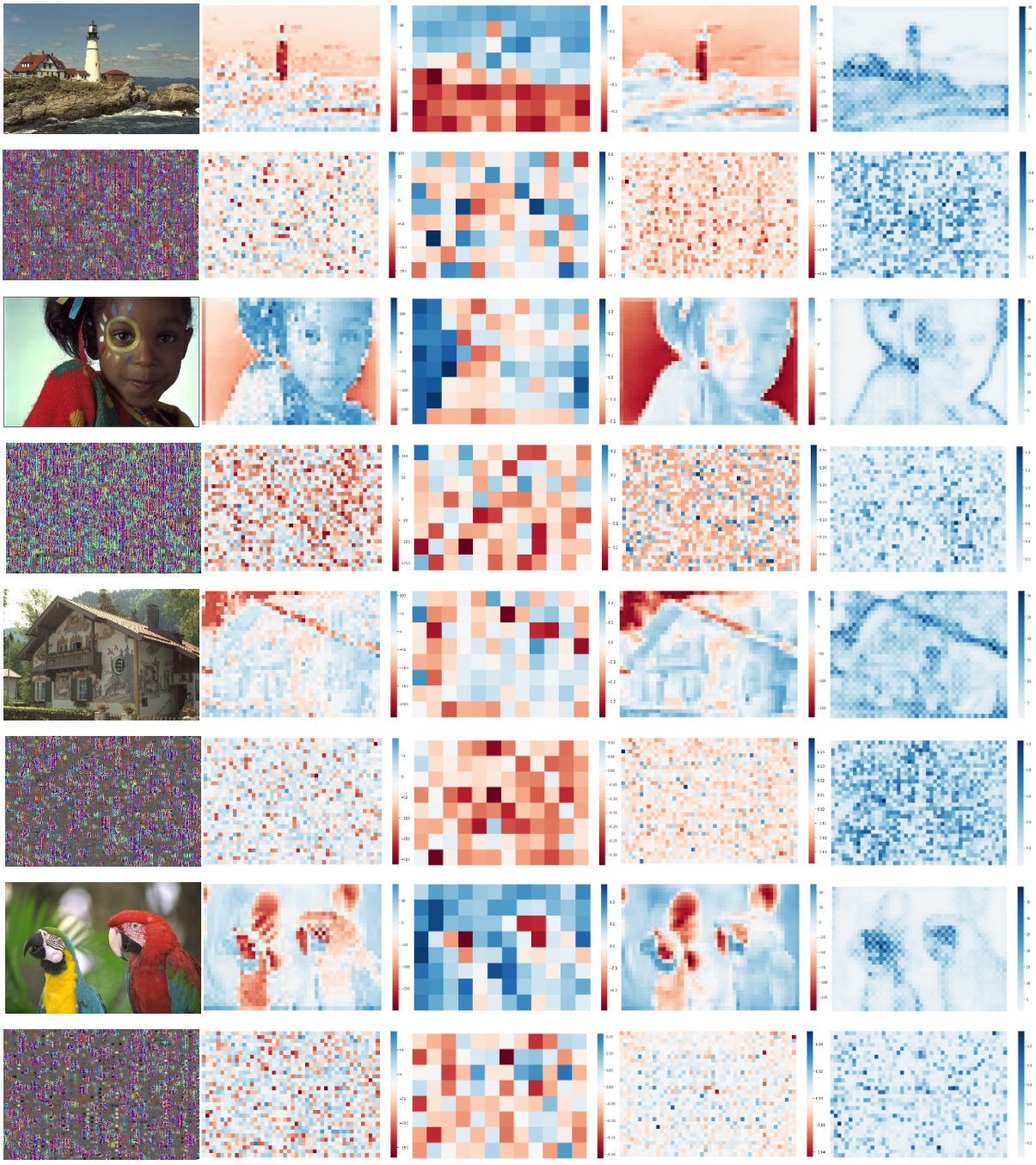


Figure 5.1: The architecture of the proposed joint model.

To illustrate how the permutation operations work here, we visualize the compression and encryption model in Figure 5.2. The odd rows show the images at different stages of the original compression model, while the even rows visualize the changes after the permutation operation. The plain images used here for illustration are all from the Kodak dataset. From the figure, we can see the clearly visible structure around edges and textures of latent representation y, which contains much information of the original images. And for the powerful compression model, the predicted mean $\hat{\mu}$ need to be close to $\hat{y}$ and predicted scales $\hat{\sigma}$ will be large in the complex regions while being small in the smooth areas. So, the parameters $\hat{\mu}, \hat{\sigma}$ also contain the visible information of images. Meanwhile, it is evident that the side information $\hat{z}$ does not contain any content of original images visually since it is generated from the encrypted feature maps y_e. And when we conduct permutation operation on z, the side information cannot be wholly recovered after decryption in the experiments. Therefore, we propose not to perform scrambling operation but to embed the encryption key $Key1$ into $\hat{z}$, such that it will not affect too much the visual quality of the decrypted image. To protect image data from eavesdropping when transmitting, we add the permutation layer and encrypt the latent representation y. The experiment results can prove that encryption on the latent representation y alone can affect the security of parameters $\hat{\mu}, \hat{\sigma}$. So we produce the permutation operation on latent representation y and have no permutation process on side information $\hat{z}$.



(a) (b) (c) (d) (e)

Figure 5.2: Visualization of the proposed model only with permutation operation. (a) input image, (b) latent representation y, (c) side information $\hat{z}$, (d) mean $\hat{\mu}$, (e) scale $\hat{\sigma}$.

5.2 Compression and Encryption Network

5.2.1 Learning Network Model and Layer Details

As shown in Figure 5.1, the deep compression and encryption model can mainly be composed of three units: encryption unit, hyper-network H and decryption unit. The encryption unit is designed for obtaining compressed representation and encrypting the plain image, which contains analysis transform, permutation, quantizer, arithmetic encoder and decoder, and the synthesis transform with known parameters from a logistic map. For the decryption unit, the plain image is recovered from the encrypted string through key extracting, arithmetic decoder, inverse permutation and synthesis transform with the right parameters. The hyperprior network H is responsible for summarizing the hyper-latent representation $\hat{z}$ to learn the entropy model, containing hyper encoder and decoder, arithmetic encoder and decoder, quantizer and key embedding. And $\hat{z}$ will be used to provide the appropriate probability estimates fitting the marginal distribution of $\hat{y}$. Here we use the Gaussian mixture likelihoods which have been proven to have a better rate-distortion performance and recover images efficiently in [15]. In learning methods, networks will be trained based on loss function to find the optimal solution. And for our learning model, the loss function is given as follows:

$$L = \beta \cdot \mathbb{E}_x[\|x - \hat{x}\|^2] + \mathbb{E}_x\left[-log\ p_{\hat{y}}(\hat{y})\right] + \mathbb{E}_x[-log\ p_{\hat{z}}(\hat{z})] \qquad (5.1)$$

The first term is the squared error between the input plain image x and the decrypted one $\hat{x}$, the output of the synthesis transform, weighted by β. The second and third terms represent the entropies of coefficient distribution for coding $\hat{y}$ and $\hat{z}$.

When the learning network model is training, images will be loaded into the encryption unit and get encrypted feature maps y_e which will be input into H to generate $\hat{z}$. Then the scrambled representation $\hat{y}_e$ will be processed through the decryption unit with all keys known and obtain $\hat{y}$ and $\hat{x}$. The whole network will be trained based on the loss function in Equation (5.1). And the value of β in Equation (5.1) is changed in every time of training to control the compression rate of the proposed model, since the AE-based method needs separate training for obtaining images at different resolutions.

The traditional compression process, like orthogonal linear transforms, chosen to reduce data correlations, usually has higher-order dependencies. In our end-to-end encryption and decryption model, we utilize a generalized divisive normalization (GDN) transform with optimized parameters, which have been shown to be highly efficient in the Gaussification of

local statistics in images previously. It is unlike most of the training stages for deep convolutional networks using the operation called batch normalization [84]. As evident in [15], there are still significant spatial dependencies in feature representation y, and we can use network H to capture this spatial structure for estimating the distribution of y. But for the encrypted latent representation y_e, the spatial structure has been scrambled and will be hard to learn. Thus we introduce the attention scheme into parameter estimating since many advanced works use attention schemes in the image processing field [30, 85, 86]. After the hyper-decoding, the mean and scale parameters for the Gaussian mixture model are predicted from the restored hyper-latent representations through the parameter estimation module. Table 5.1 details the network structures of our proposed framework and illustrates the parameters of corresponding components. And each row corresponds to a layer, while Conv denotes a convolution layer with the kernel size and number of output channels shown in Table 5.1. S is the downsampling/upsampling stride and IGDN is the approximate inverse operation of GDN. AM represents the attention module we used which is shown in Figure 5.3.

Table 5.1: The details of the layers in our proposed model.

Analysis Transform	Synthesis Transform	Hyper Encoder	Hyper Decoder	Parameter Estimation
Conv: 5×5×192 s2	Deconv: 5×5×192 s2	Conv: 3×3×192 s1	AM	AM
GDN	IGDN	Leaky ReLU	Deconv: 5×5×320 s2	Conv: 1×1×640 s1
Conv: 5×5×192 s2	Deconv: 5×5×192 s2	Conv: 5×5×192 s2	Leaky ReLU	Leaky ReLU
GDN	IGDN	Leaky ReLU	Deconv: 5×5×480 s2	Conv: 1×1×640 s1
Conv: 5×5×192 s2	Deconv: 5×5×192 s2	Conv: 5×5×192 s2	Leaky ReLU	
GDN	IGDN		Deconv: 5×5×640 s2	
Conv: 5×5×320 s2	Deconv: 5×5×3 s2			

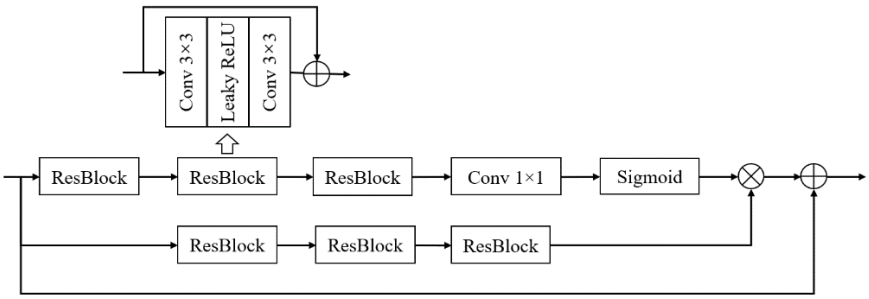


Figure 5.3: The architecture of the attention module.

5.2.2 Keystream Generation Process

In our encryption scheme, the BLAKE256 hashing algorithm is used to generate the encryption key, by taking the plain image as input, such that the encryption key is sensitive to the plain image. Here, different input plain images lead to different fixed-length random hash values. And the following encryption operations are controlled by these values. The outputs will also be available in the decoding stage for restoring images.

We take the plain image as the input of BLAKE256 hash algorithm and obtain a 256-bit random key $Key1$ which is the initial seed to generate a random key stream $KS1$. $KS1$ is then used for controlling the encryption/decryption operations in the proposed system. The formula for generating the sequence is given as follows

$$KS1_{n+1} = \text{BLAKE256}(KS1_n)(n = 0,1,2, \dots) \tag{5.2}$$

The key stream $KS1$ is composed of $KS1_n$, and $Key1$ is the initial seed $KS1_0$. And the random key stream $KS2$ is also generated through Equation (5.2) when the predefined embedding key $Key2$ is the initial value. $KS2$ is then used for controlling the data embedding process for $Key1$.

For shuffling all elements in the feature map y, the new index of the feature map can be generated by the Fisher-Yates Shuffle algorithm using the random key stream $KS1$. For a given index array R with r elements, Fisher-Yates Shuffle [71] do

for $p \leftarrow r$ to 2 **do**
    $q \leftarrow$ random integer $(1 \leq q \leq p)$
    exchange $R[p]$ and $R[q]$
end for

The random integer is obtained from the key stream $KS1$. Similarly, the index of embedding positions is controlled by the key stream $KS2$ using the Fisher-Yates Shuffle algorithm at the key embedding stage.

5.2.3 Key Embedding

To against various cryptanalysis techniques, we use different encryption keys for the different input images, which increases the cost of transmission. So, we propose to embed the encryption key $Key1$ into side information $\hat{z}$ and then compress them by the arithmetic coder to save the cost. The embedding process will only be conducted on the non-zero values in the side information to ensure the compression efficiency of learning-based compression will not be affected too much. Here, the values in $\hat{z}$ are all integers. And when generating an encrypted image, the key embedding into $\hat{z}$ will not be extracted.

The encryption key $Key1$ is 256-bit in length. But the number of non-zero values in the side information is less than 256 at a low bit rate from the experiment results. So, to reduce the impact of the value changes in $\hat{z}$, we only embed 1-bit data at each position of $\hat{z}$ when there are more than 255 non-zero values in $\hat{z}$. For example, a value in $\hat{z}$ is 4 and it can be converted to '00000100' while the 1-bit data for embedding is '1'. The bitstream can be '00001001' after embedding data at the end, and it will be converted to the number 9 which is the new value. When the number of non-zero values is less than 255 and more than 127, we embed 2-bit data at each position. And when the number is less than 127, we embed 4-bit data at each position. The selection of positions for data embedding is controlled by $KS2$ from embedding key $Key2$, which will be shared for different plain images. The detailed key embedding algorithm is shown below.

Key Embedding

for each plain image **do**:
    Get L positions of non-zero values in the side information $\hat{z}$
    Generate the random index of positions for embedding bitstream by the Fisher-Yates Shuffle algorithm with $KS2$
    if $L < 128$
        Select the first 64 positions from the index for data embedding
        Change the values of all selected positions with the appended 4-bit data from the encryption key $Key1$

7: else if $127 < L < 256$

8: Select the first 128 positions from the index for data embedding

9: Change the values of all selected positions with the appended 2-bit data from the encryption key $Key1$

10: else

11: Select the first 256 positions from the index for data embedding

12: Change the values of all selected positions with the appended 1-bit data from the encryption key $Key1$

13: end if

14: end for

5.2.4 Parameter Substitution

To obtain a higher level of visual security, we conduct the parameter substitution on then synthesis transform when decoding the images. The parameter matrix of the last convolution layer in synthesis transform will be replaced by a new matrix, which is generated after the division operation between the original parameter matrix and the logistic map. And the new matrix can be converted to the original one by multiplying the logistic map only during the decryption process. The logistic map we used is defined as:

$$U_{t+1} = \theta U_t(U_t + 1),\ U_t \in (0,1) \tag{5.3}$$

where U_t is the value for t iterations, and θ is the system parameter. The system is chaotic when θ is in the range of [3.57, 4]. Here, the first two decimal values of encryption key $Key1$ are normalized and then generate the initial value U_0 and parameter θ. So different logistic maps will be generated with different values of U_0 and θ for different images.

5.2.5 Encryption and Decryption Algorithm

After obtaining the encryption keys, the input images are encrypted through proposed encryption operations and obtain the compressed and encrypted images, then restored from the encrypted bit-string to compressed images by the decryption process with keys. And hyper-network H is utilized to obtain side information $\hat{z}$ and estimate the parameters of the probabilistic model which is used for compression. The permutation operations are conducted on all three dimensions of deep features y and the new index arrays are generated by the Yates

Shuffle algorithm mentioned in Section 5.2.2. The details of the encryption and decryption algorithms are shown below.

Algorithm 1: Encryption Algorithm

Load original image I and the embedding key $Key2$

$Key1 \leftarrow \text{BLAKE256}(I)$

$KS1 \leftarrow \text{BLAKE256}(Key1)$

$KS2 \leftarrow \text{BLAKE256}(Key2)$

Enter image into analysis transform part and get feature matrix y with three dimensions

for all elements in each dimension of y **do**

    Perform Yates Shuffle algorithm where the random integer in each loop is from $KS1$ to generate a new index *G* for permutation

    Change the order of elements in this dimension according to the new index *G*

end for

Collect all permutated elements and combine them into a new feature matrix y_{e}

Get side information $\hat{z}$ and embed $Key1$ into $\hat{z}$ controlled by $KS2$, then get the side string

Decompress the side string and use parameter estimation to predict the parameters $\hat{\sigma}$ and $\hat{\mu}$ of entropy model through H

Use parameters $\hat{\sigma}$ and $\hat{\mu}$ to compress the quantized representation $\hat{y}_{\text{e}}$ and produce an encrypted bitstream (combining the side string)

Conduct the parameter substitution on the synthesis transform and the parameter matrix is replaced by the new one

The final encrypted image can be obtained through the decoder and the synthesis transform with wrong parameters

Algorithm 2: Decryption Algorithm

$KS2 \leftarrow \text{BLAKE256}(Key2)$

Extract $Key1$ using $KS2$ from the side string and recover $\hat{z}$

$KS1 \leftarrow \text{BLAKE256}(Key1)$

Estimate parameters $\hat{\sigma}$ and $\hat{\mu}$ from hyper-decoder and parameter estimation through H for recovering y_{e} from the encrypted bitstream by decoder

for all elements in each dimension of y_{e} **do**

6: Perform Yates Shuffle algorithm controlled by $KS1$ to get the index G

7: Restore the permuted elements to their original positions according to the index G

8: end for

9: Conduct the parameter substitution on the synthesis transform and recover the original parameter matrix

10: The decrypted image can be obtained from reconstructed y through synthesis transform with the right parameters

5.3 Deep image Retrie

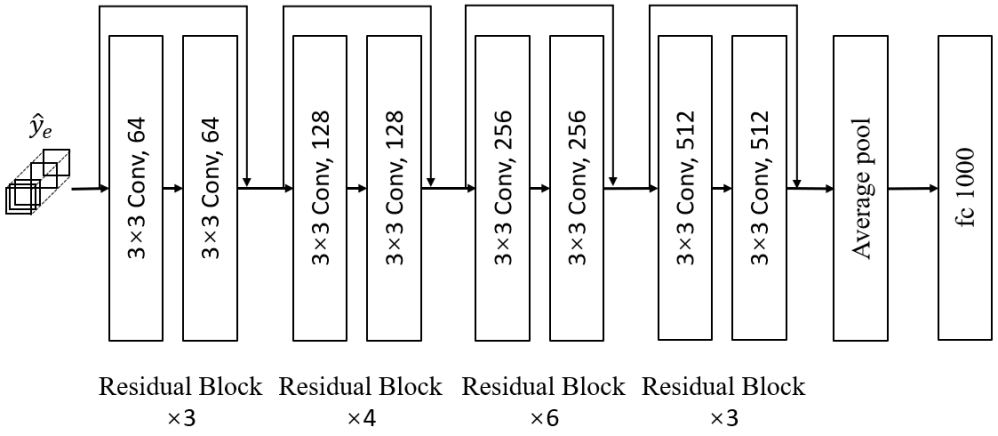


Figure 5.4: The structure details of the deep model.

In the proposed image encryption and retrieval system, encrypted data is uploaded to the cloud to outsource data. When an encrypted query image is submitted, the features of this query data and all other data in the database of the cloud are extracted, and the compressed and encrypted representations are used as the inputs of the deep retrieval network. Then, other cipher images with similar content of the query image are returned to the user. Finally, the authorized user can decrypt the images with the corresponding key. Here the compressed and encrypted representation $\hat{y}_e$ is used as the input of the proposed image retrieval model, which can avoid the cost caused by decompression and ensure privacy and availability requirements. And the structure details of the deep model are shown in Figure 5.4. To achieve better retrieval performance, ranked list loss (RLL) [87] and cross-entropy loss are introduced for training the deep architecture to get results. For sample s_i and corresponding label ls_i in a batch set, f is the corresponding deep feature and the RLL can be given as follows:

$$L_{RLL} = \sum((1-\lambda)L_P(s_i^c; f) + \lambda L_N(s_i^c; f)) \tag{5.4}$$

$$L_P(s_i^c; f) = \frac{1}{|P_{c,i}^*|} \sum_{s_j^c \in P_{c,i}^*} L_m(s_i^c, s_j^c; f) \tag{5.5}$$

$$L_N(s_i^c; f) = \sum_{s_j^k \in N_{c,i}^*} \frac{w_{ij}}{\sum_{s_j^k \in N_{c,i}^*} w_{ij}} L_m(s_i^k, s_j^c; f) \tag{5.6}$$

$$L_m(s_i, s_j; f) = (1 - ls_{ij})[\alpha - d_{ij}]_+ + ls_{ij}[d_{ij} - (\alpha - m)]_+ \tag{5.7}$$

Here L_P is the loss function used on positive samples and L_N is for the negative sample in Equation (5.4). Non-trivial positive set is represented as $P_{c,i}^* = \{s_j^c | j \neq i, d_{ij} > (\alpha - m)\}$ while the negative set is denote as $N_{c,i}^* = \{s_j^k | k \neq c, d_{ij} < \alpha)\}$, where c represents the class. Considering a large number of negative samples, negative samples are weighted in L_P in Equation (5.6). In Equation(5.7), $ls_{ij} = 1$ if $ls_i = ls_j$, and $ls_{ij} = 0$ otherwise. d_{ij} is the Euclidean distance and $[.]_+$ is the hinge function. In the experiment, we set parameters as those in [87].

Cross-entropy loss is commonly used in the classification area. And the final loss function can be defined as follows:

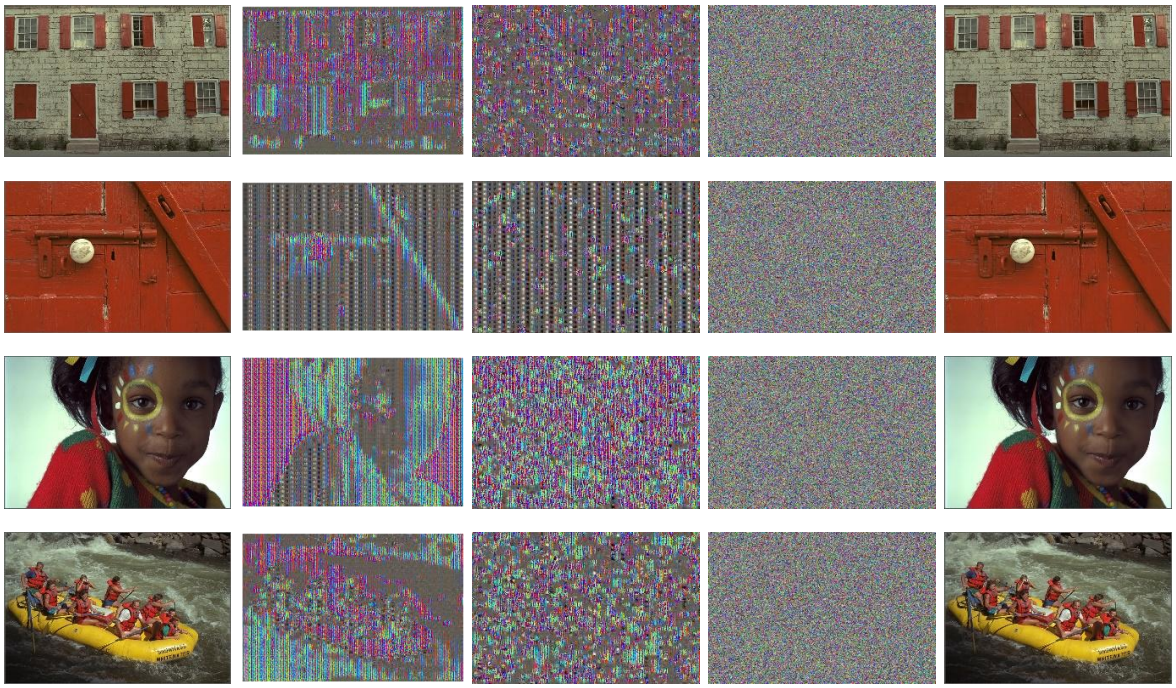
$$L_{RE} = L_{RLL} + L_{CE} \tag{5.8}$$

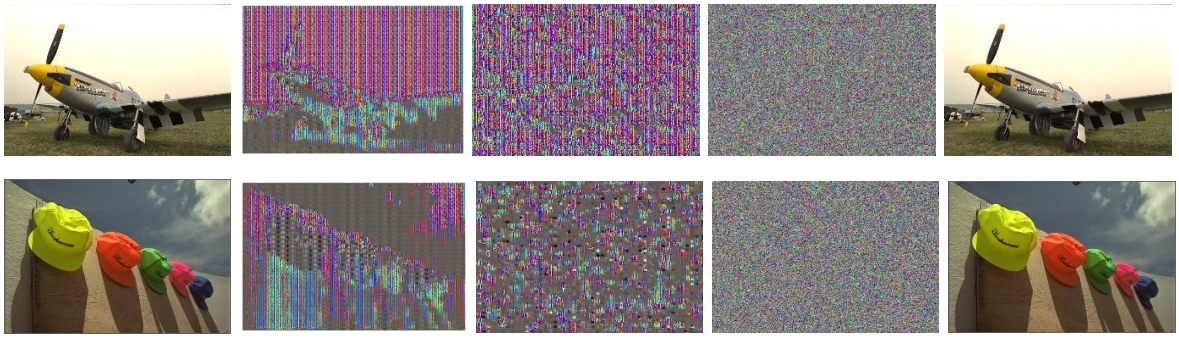
5.4 Performance Evaluation

In this section, various experiments will be conducted to evaluate the compression performance and the perceptual security of the proposed compression and encryption model. The performance evaluation is conducted on Kodak dataset [72]. There are 24 high-quality images in this dataset, and some of them are shown in Figure 5.5. The proposed methods are trained on the Nvidia GTX 2080Ti and the training dataset contains more than 13k images extracted from the Vimeo-90k dataset [88]. When training, the embedding key $Key2$ will be known and the encryption key $Key1$ will be generated from input images. All training images will be loaded and processed through the encryption algorithm to get the string, and then the decryption images will be recovered from the string through the decryption algorithm. Our learning model is optimized by mean square error (MSE) between the input image and the decryption one. The learning rate is initially set to 1×10^{-4} and decreases during the training. The parameter β of the loss function is within the set {0.0018, 0.0035, 0.007, 0.015, 0.03, 0.045, 0.09, 0.18}. And Adam optimizer is used with a batch size of 32.

Figure 5.5: Test images from Kodak dataset.

In the proposed model, we first permute the elements of feature matrix y in the third dimension for encryption, and the performance of this encryption step can be shown in Figure 5.6(b). We can find that the encrypted image only under this permutation still reveals some edge information contained in the original image. So, to obtain more chaotic cipher-images, the Fisher-Yates Shuffle operation on the first and second dimensions of y would be conducted after the permutation procedure on the third dimension. And the performance of the permutation operation is shown in Figure 5.6(c). And for higher visual security, the parameter substitution is conducted when generating the final encrypted images and some encrypted images are shown in Figure 5.6(d), while the corresponding decrypted images are illustrated in Figure 5.6(e). And no content information about original images can be seen in those encrypted images which illustrates the visual security of our encryption scheme.





(a) (b) (c) (d) (e)

Figure 5.6: Encryption and decryption results. (a) original image, (b) encrypting image only on third dimension of y, (c) encrypting image on all dimensions of y, (d) final cipher image, (e) decrypted image.

In the perceptual security evaluation of encryption methods, the PSNR and multiscale structural similarity (MS-SSIM) are the typical metrics for measuring the perceptual performance of images after encryption. And to illustrate the difference clearly in Figure 5.7, we use the value of $-10\log_{10}(1-\text{MS-SSIM})$ to replace MS-SSIM. The pictures in the Kodak dataset are encrypted and compressed using our proposed model. The average PSNR values and MS-SSIM values of these ciphertext images under our encryption and compression model are shown in Figure 5.7, compared with the values when encryption keys are known. From Figure 5.7, we can observe significant drops in PSNR and MS-SSIM without the decryption process for the proposed model, which means our encryption scheme has good content protection. For example, at a bit rate of 0.54 bits/pixel, the PSNR value of our compression and encryption model is 8.56dB while the value of the model with key is 33.15dB. It is obvious that shuffling all three dimensions of the feature map y can get better security performance than only conducting permutation operation on the 3rd dimension. And when only conducting the permutation operation on y, the values of PSNR and MS-SSIM are larger than the values of our proposed encryption model. Here lower PSNR and MS-SSIM values prove that the security of the encryption model has been improved after parameter substitution.

When the encryption key is provided for decryption, the PSNR and MS-SSIM values of our proposed model can illustrate the compression efficiency of our model. In general, our model performs better than the model without AM. The difference is more obvious at low bit rates, which benefit from the attention-based parameter estimation. The average PSNR values of our model are 0.4-0.5 dB higher than the model without AM under the same BPP value. Table 5.2 and Table 5.3 present the comparison of PSNR and MS-SSIM values with different bit rates

between our model and the model without AM. In Figure 8, for comparison purpose, we also show the performance of JPEG, the original AE-based compression model, and the joint compression and encryption schemes in [30] and [45]. A learning network and an attention mechanism are also exploited in [30]. And [45] is the SCE-based method which also uses Fisher-Yates Shuffle to generate a new index for permutation operation. Here we used the JPEG model with the default configuration (4:2:0). It is observed from Figure 5.7 that our proposed model achieves significantly better performance in terms of PSNR and MS-SSIM than other methods. Moreover, our proposed encryption and compression model is very close to the original compression model which removes all encryption operations. The experiment results demonstrate that our encryption and compression model can obtain a high protection ability with a slight sacrifice on compression efficiency.

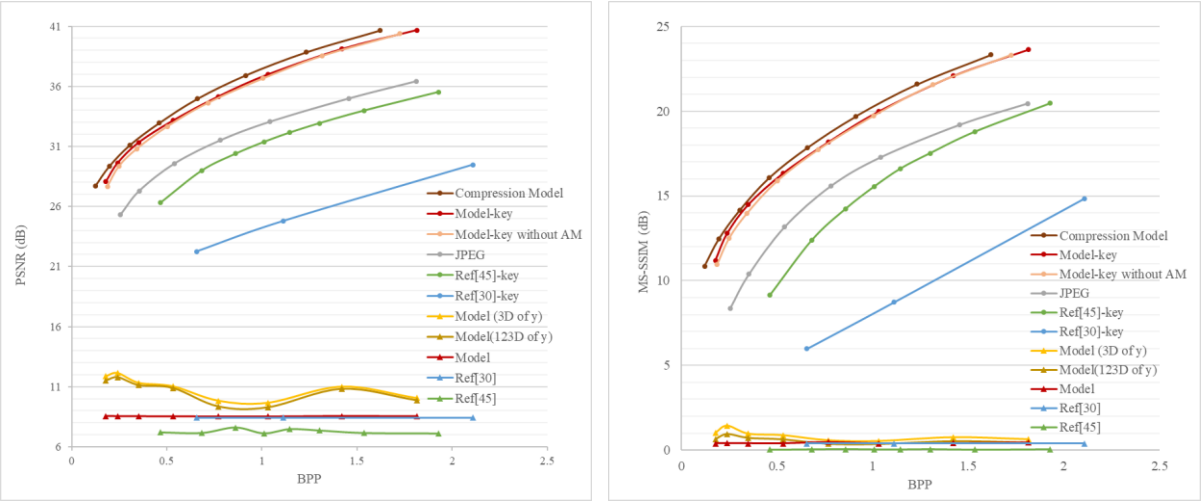


Figure 5.7: Rate distortion performance for different methods.

Table 5.2: Comparison of compression and encryption performance at low bit rate (low value of β).

	Model-key			Model-key without AM			Model		
	BPP	PSNR	MS-SSIM	BPP	PSNR	MS-SSIM	BPP	PSNR	MS-SSIM
Kodim01	0.561	29.07	0.9694	0.533	28.52	0.9652	0.561	9.26	0.0802
Kodim02	0.237	32.04	0.9384	0.231	31.43	0.9289	0.237	7.77	0.1126
Kodim03	0.212	33.82	0.9696	0.207	33.26	0.9667	0.212	8.81	0.0958
Kodim04	0.252	32.23	0.9543	0.246	31.74	0.9477	0.252	8.78	0.0949

Kodim05	0.597	29.27	0.9744	0.585	28.80	0.9712	0.597	8.06	0.0536
Kodim06	0.417	30.25	0.9595	0.403	29.66	0.9537	0.417	8.56	0.1011
Kodim07	0.282	33.28	0.9807	0.272	32.73	0.9796	0.282	9.25	0.0781
Kodim08	0.642	28.55	0.9759	0.627	28.08	0.9730	0.642	8.41	0.0507
Kodim09	0.236	33.66	0.9762	0.230	33.16	0.9745	0.236	9.72	0.0987
Kodim10	0.246	33.34	0.9698	0.243	32.86	0.9668	0.246	9.66	0.0988
Kodim11	0.345	30.81	0.9565	0.334	30.25	0.9489	0.345	8.73	0.0849
Kodim12	0.214	33.36	0.9551	0.207	32.64	0.9492	0.214	8.84	0.1124
Kodim13	0.706	26.78	0.9609	0.688	26.38	0.9553	0.706	8.50	0.0740
Kodim14	0.431	29.78	0.9607	0.421	29.23	0.9548	0.431	8.33	0.0748
Kodim15	0.247	32.33	0.9635	0.245	31.82	0.9595	0.247	7.08	0.0884
Kodim16	0.251	32.00	0.9579	0.248	31.29	0.9512	0.251	9.11	0.1083
Kodim17	0.246	32.59	0.9699	0.241	32.14	0.9664	0.246	7.93	0.0822
Kodim18	0.444	29.33	0.96	0.435	28.91	0.9555	0.444	7.8	0.09
Kodim19	0.314	31.23	0.9596	0.307	30.74	0.9531	0.314	9.05	0.0908
Kodim20	0.227	33.00	0.9743	0.222	32.47	0.9716	0.227	6.30	0.0993
Kodim21	0.374	30.84	0.9709	0.359	30.33	0.9689	0.374	9.30	0.0925
Kodim22	0.319	30.73	0.9490	0.312	30.21	0.9413	0.319	9.11	0.0891
Kodim23	0.189	34.52	0.9718	0.185	34.05	0.9690	0.189	8.4	0.0832
Kodim24	0.454	29.16	0.9678	0.444	28.69	0.9633	0.454	8.65	0.0788

Table 5.3: Comparison of compression and encryption performance at high bit rate (high value of β).

	Model-key			Model-key without AM			Model		
	BPP	PSNR	MS-SSIM	BPP	PSNR	MS-SSIM	BPP	PSNR	MS-SSIM
Kodim01	2.675	40.27	0.9976	2.568	39.98	0.9975	2.675	9.26	0.0759
Kodim02	1.642	40.77	0.9929	1.582	40.30	0.9925	1.642	7.76	0.1121
Kodim03	1.096	42.52	0.9953	1.020	42.36	0.9952	1.096	8.81	0.0967
Kodim04	1.574	41.08	0.9947	1.484	40.80	0.9944	1.574	8.78	0.0938
Kodim05	2.519	39.67	0.9980	2.432	39.41	0.9979	2.519	8.06	0.0522

Kodim06	2.065	40.74	0.9961	1.960	40.47	0.9959	2.065	8.56	0.1028
Kodim07	1.227	42.33	0.9969	1.153	42.10	0.9968	1.227	9.25	0.0779
Kodim08	2.823	39.06	0.9978	2.731	38.79	0.9977	2.823	8.40	0.0475
Kodim09	1.148	41.37	0.9938	1.083	41.01	0.9933	1.148	9.73	0.0954
Kodim10	1.253	41.34	0.9946	1.194	40.96	0.9943	1.253	9.67	0.0989
Kodim11	1.944	40.93	0.9961	1.835	40.63	0.9959	1.944	8.73	0.0841
Kodim12	1.277	41.88	0.9942	1.207	41.55	0.9939	1.277	8.84	0.1131
Kodim13	3.227	38.26	0.9975	3.141	38.02	0.9974	3.227	8.51	0.0745
Kodim14	2.327	39.73	0.9968	2.219	39.47	0.9967	2.327	8.33	0.0741
Kodim15	1.478	41.07	0.9946	1.390	40.77	0.9943	1.478	7.08	0.0919
Kodim16	1.524	41.98	0.9956	1.418	41.60	0.9954	1.524	9.12	0.1047
Kodim17	1.367	41.25	0.9958	1.284	40.90	0.9955	1.367	7.94	0.0823
Kodim18	2.339	38.76	1.00	2.243	38.45	1.00	2.339	7.79	0.87
Kodim19	1.783	40.75	0.9949	1.684	40.41	0.9945	1.783	9.06	0.0949
Kodim20	1.305	41.55	0.9944	1.191	41.23	0.9940	1.305	6.28	0.0999
Kodim21	1.816	40.42	0.9944	1.699	40.09	0.9940	1.816	9.29	0.0913
Kodim22	1.986	39.99	0.9944	1.884	39.66	0.9940	1.986	9.11	0.0946
Kodim23	0.911	41.85	0.9948	0.866	41.60	0.9946	0.911	8.41	0.0868
Kodim24	2.240	38.43	0.9969	2.122	38.22	0.9968	2.240	8.64	0.0761

5.5 Security Analysis

This section is going to discuss various cryptographic attacking methods, such as ciphertext-only attack, differential attack, and statistical attack. And the robustness against those attacks will be evaluated to analyze the security of our scheme.

5.5.1 Ciphertext-only Attack

The ciphertext-only attack is one of the most basic and realistic methods for various cryptanalysis techniques. To defend against this kind of brute-force attack, the key space of the cryptosystem should be large. In our model, the parameters of the learning network are given. The 256-bit encryption key $Key1$ is produced by the BLAKE256 hashing from images and controls all encryption operations in the proposed model. Therefore, we obtain a $2^{(256)}$ keyspace which is lager than the theoretical requirement with $2^{(100)}$. Because of the large

keyspace of our model, it is tough for the attacker to break down even though the attacker knows the learning network and our encryption system can handle this attack easily.

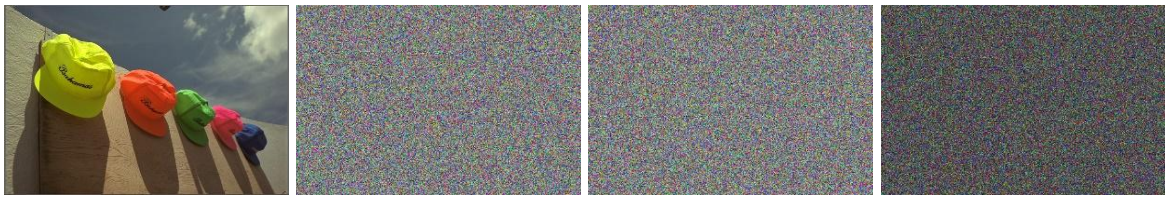
5.5.2 Key sensitivity Analysis

In general, the security of an encryption system should only rely on the secrecy of keys but not the underlying techniques. In this regard, a cryptosystem needs to be highly sensitive to the encryption and decryption keys used. Here, we use the plain images 'kodim03'as examples. The high key sensitivity level can be demonstrated in two parts:

1) A completely different ciphertext image would be generated for the same plain image if the encryption keys used change slightly.

2) The encrypted image should not be decrypted when a key having minor change to the encryption key is used.

In the first case, we make a minor change in encryption key $Key1$ to generate the new key stream $KS1$. We then generate two encrypted images using the two different keys for the same input image. And the encrypted images are shown in Figure 5.8(b) and 5.8(c), while their difference is shown in Figure 5.8(d). The mean square error (MSE) between these two cipher images is measured as 3.65×10^3, which is very large. It is clearly seen, from both the MSE figure and the difference image, that the encrypted images obtained by two slightly different keys are very different, and our proposed model fulfils the first case of key sensitivity.



(a) (b) (c) (d)

Figure 5.8: Comparison of encrypted images with different keys. (a) plain images, (b) cipher image through the original key, (c) cipher image through the new $Key1$, (d) difference image between cipher images.

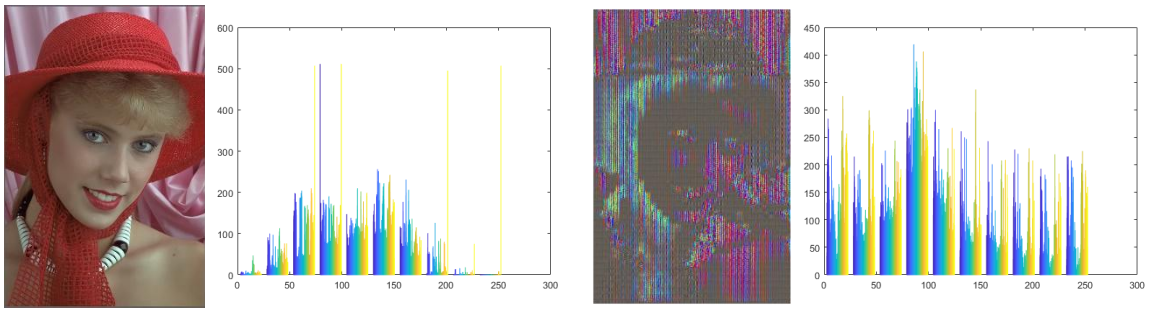
For the second case, the original key and the new key generated are utilized to decrypt the same cipher-image encrypted with the original key. The cipher image is shown in Figure 5.9(b), and the decrypted images using the two different keys are shown in Figure 5.9(c) and 5.9(d). It is seen that only the original key can recover the original image.

(a) (b) (c) (d)

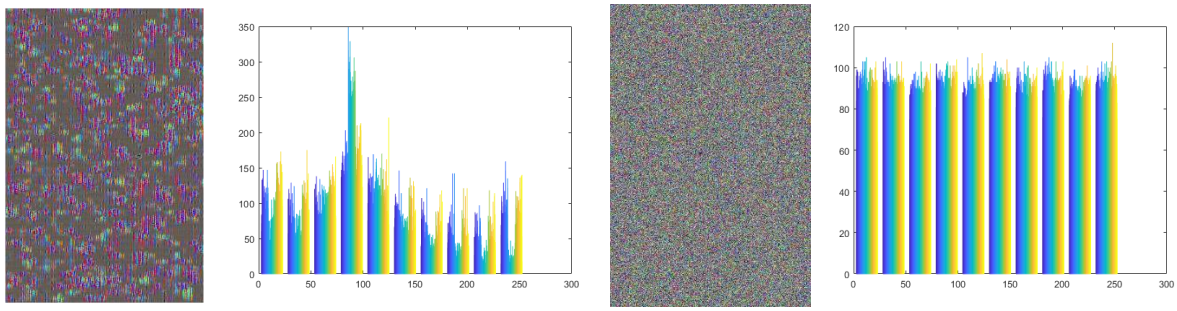
Figure 5.9: Decryption with different keys. (a) original image, (b) cipher image, (c) decrypted image using the original key $Key1$, (d) decrypted image using the new $Key1$.

5.5.3 Statistical Attack

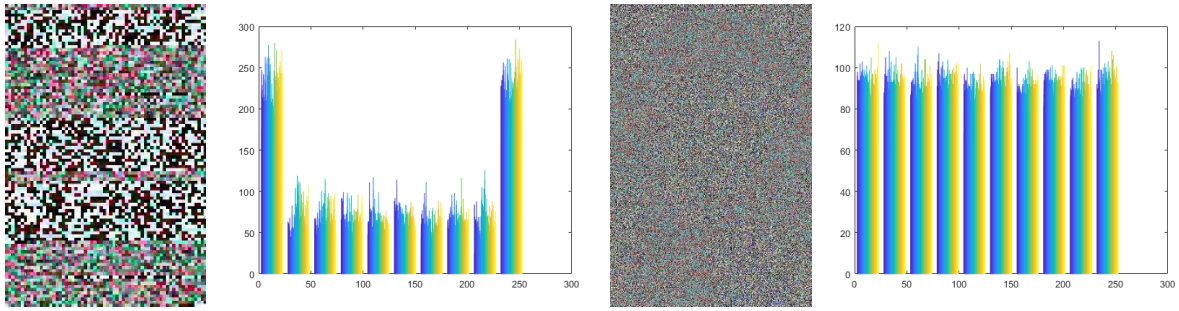
In statistical attacks, attackers utilize the high correlation between pixels and obtain original images through the predictable relationship between plain images and encrypted images. So, a good encryption scheme needs to reduce the statistical relationship between plain images and encrypted images to defend against this attack. And the histogram and correlation chart are two standard methods to illustrate the correlation. To evaluate the robustness against the statistical attack, the histogram of image 'kodim04' and corresponding encrypted images are given in Figure 5.10. It is seen that there are large differences between the histograms of images before and after encryption. We can find that the shuffling operation on feature maps can extremely decrease the pixel's correlation. And the parameter substitution can further reduce the correlation. According to the results, our model and [45] can achieve the uniform distributed histogram, revealing the excellent property to resist the statistical attack. In comparison, the cipher image from [30] does not show a uniform distribution since there are still some correlations among pixels.



(a) (b)



(c) (d)



(e) (f)

Figure 5.10: Histogram charts of plain-image and cipher-image. (a) plain-image (b) cipher-image only with permutation on the third dimension of y, (c) cipher-image with permutation on all dimensions of y, (d) final cipher-image, (e) Ref [30], (f) Ref [45].

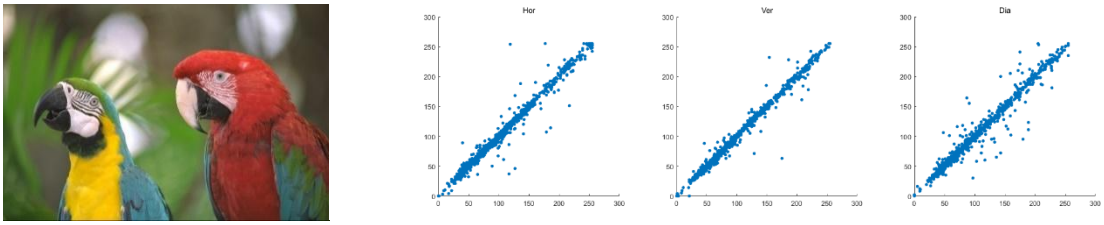
For the correlation chart, the correlation of adjacent pixels in images will be measured in the horizontal direction (Hor), the vertical direction (Ver), and the diagonal direction (Dia). Table 5.4 shows the correlation coefficients for the original images and encrypted images. It is shown that the encrypted image has less correlation of pixels compared to the original image, illustrating the good decorrelation performance of our encryption model.

Table 5.4: Correlation coefficients of adjacent pixels.

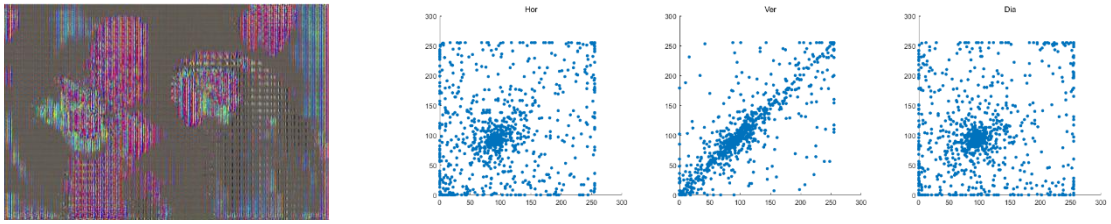
	Original Image			Encrypted Image		
	Hor	Ver	Dia	Hor	Ver	Dia
Kodim01	0.9050	0.8315	0.7536	0.0392	-0.0026	0.0002
Kodim02	0.8823	0.8499	0.8126	-0.0086	0.0151	-0.0666
Kodim03	0.9801	0.9731	0.9459	-0.0157	-0.0211	-0.0175
Kodim04	0.9632	0.9713	0.9498	0.0220	0.0161	0.0204

Kodim05	0.8990	0.8840	0.8156	0.0130	-0.0186	0.0457
Kodim06	0.9726	0.9321	0.9303	-0.0433	0.0070	0.0429
Kodim07	0.9720	0.9328	0.9207	0.0300	0.0165	-0.0011
Kodim08	0.8962	0.9207	0.8272	-0.0052	0.0288	0.0391
Kodim09	0.9445	0.9584	0.9322	0.0271	0.0122	0.0027
Kodim10	0.9617	0.9646	0.9035	-0.0668	-0.0247	-0.0434
Kodim11	0.9387	0.9269	0.8945	0.0849	-0.0134	0.0654
Kodim12	0.9732	0.9577	0.9513	-0.0344	-0.0303	-0.0464
Kodim13	0.8865	0.8509	0.7978	0.0414	-0.0312	-0.0161
Kodim14	0.9569	0.9153	0.9007	-0.0288	-0.0054	0.0402
Kodim15	0.9889	0.9893	0.9788	-0.0254	0.0251	-0.0476
Kodim16	0.9790	0.9359	0.9387	-0.0071	-0.0129	0.0219
Kodim17	0.9682	0.9715	0.9622	-0.0117	-0.0205	-0.0299
Kodim18	0.9031	0.8919	0.8534	0.0207	-0.0418	0.0187
Kodim19	0.9275	0.9505	0.8853	-0.0171	-0.0158	-0.0133
Kodim20	0.9906	0.9856	0.9768	0.0213	0.0383	0.0185
Kodim21	0.9384	0.8905	0.8558	0.0515	-0.0309	0.0115
Kodim22	0.9506	0.9631	0.9258	0.0207	-0.0373	0.0168
Kodim23	0.9833	0.9763	0.9565	0.0068	-0.0577	-0.0490
Kodim24	0.9363	0.9420	0.9110	-0.0108	-0.0199	0.0046

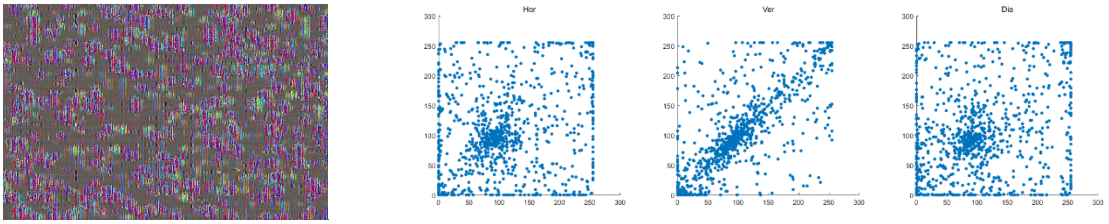
And for example, Figure 5.11 shows the correlation charts of image ‘kodim23’ and ciphertext images under our encryption model, [30] and [45]. A similar shuffling operation is also utilized in [45]. But the results show that the shuffling operation in our model achieves better performance on decreasing the pixel’s correlation compared with [45]. Then the parameter substitution can further reduce the correlation.



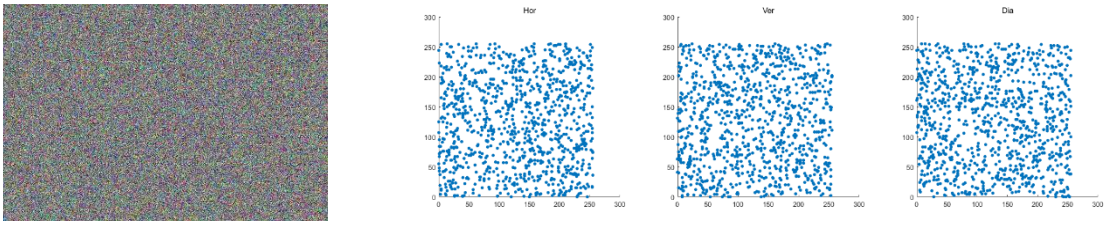
(a)



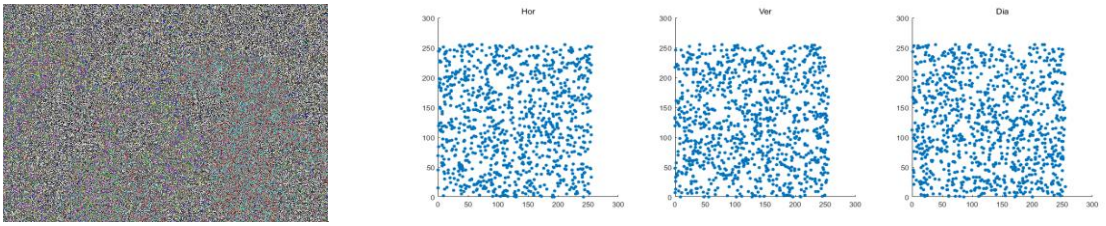
(b)



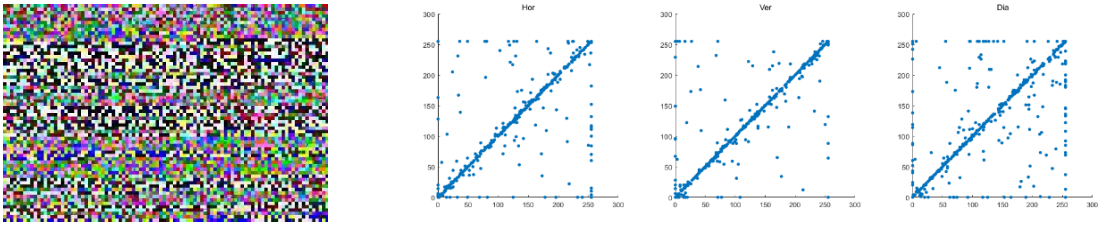
(c)



(d)



(e)



(f)

Figure 5.11: Correlation charts of images before and after encryption. (a) original image (b) encrypted image only with permutation on the third dimension of y, (c) encrypted image from the proposed model, (e) Ref [30], (f) Ref [45].

5.5.4 Differential Attack

In differential attacks, attackers try to obtain the encryption keys by studying the influence of input differences on output changes. To resist differential attack, the encryption system needs to be sensitive to changes in the image, and minor changes in plain images should cause large changes in encryption images. The common criteria used in measuring the degree of image change are net pixel change ratio (NPCR) and unified average change in intensity (UACI). Generally, the higher value of NPCR, the better performance of the encryption system. When UACI is close to 33%, the encryption system has higher security.

Therefore, to evaluate the robustness against differential attack, we conduct a slight modification on some random pixel values to generate the slightly changed image. In the experiment, only about 1% of the pixels in the image would be changed by adding 1 to the value. Then both images will be encrypted by the same key. Table 5.5 gives the mean NPCR and UACI values for our encryption system. From the table, the proposed model has the defense capability against the differential attack and the encryption operation on dimensions i,j of y can enhance the robustness, while the NPCR and UACI values are all almost zero for the encryption method in [30], indicating the low diffusion property. Also, the mean NPCR of our encryption system is higher than that of [45], which is less than 98%. Besides, [83] uses the deep network to transfer images to another domain for encryption, but the average NPCR mentioned in the paper is less than 95% which is lower than our model.

Table 5.5: Mean NPCR and UACI of cipher-images.

	Proposed Model		Ref [30]		Ref [45]	
	NPCR%	UACI%	UACI%	UACI%	NPCR%	UACI%
Kodim01	99.61	33.47	0.01	8.22e-05	96.62	39.32
Kodim02	99.62	33.47	0.01	8.98e-05	97.97	36.48
Kodim03	99.59	33.44	0.01	9.74e-04	98.91	28.36
Kodim04	99.63	33.44	0.01	7.12e-05	98.18	38.23
Kodim05	99.59	33.54	0.01	7.58e-05	97.42	36.31
Kodim06	99.60	33.50	0.01	6.45e-05	96.86	36.48
Kodim07	99.59	33.40	0.01	7.31e-04	97.91	29.37
Kodim08	99.62	33.49	0.01	9.74e-04	98.62	38.23
Kodim09	99.61	33.42	0.01	7.11e-05	98.02	37.82
Kodim10	99.61	33.50	0.01	7.58e-05	97.23	36.48

Kodim11	99.62	33.48	0.01	6.45e-04	97.93	29.46
Kodim12	99.61	33.50	0.01	5.91e-05	99.21	38.23
Kodim13	99.60	33.52	0.01	9.74e-04	98.02	36.62
Kodim14	99.60	33.46	0.01	4.12e-04	97.24	36.48
Kodim15	99.60	33.43	0.01	7.52e-05	97.67	29.33
Kodim16	99.63	33.43	0.01	5.45e-04	98.62	36.23
Kodim17	99.62	33.46	0.01	8.78e-05	98.02	35.31
Kodim18	99.61	33.38	0.01	9.74e-04	97.29	36.49
Kodim19	99.60	33.53	0.01	9.14e-04	97.95	28.34
Kodim20	99.61	33.46	0.01	7.68e-05	97.62	28.23
Kodim21	99.60	33.50	0.01	7.45e-04	98.02	37.62
Kodim22	99.62	33.49	0.01	8.12e-04	97.25	36.58
Kodim23	99.61	33.51	0.01	7.88e-05	97.02	28.96
Kodim24	99.61	33.52	0.01	6.45e-04	99.25	38.23

5.5.5 Robustness Analysis

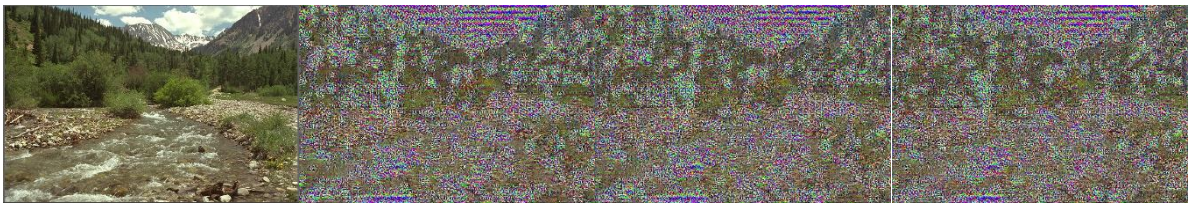
When images are transmitted over the Internet, information blocking and loss can sometimes occur which may affect the recovery. So the encryption system needs to effectively resist clipping and noise attacks to show good robustness.

(1) Clipping attacks: When attacking by pixel clipping, the quality of the decrypted image will decrease significantly. Here, we set 1/64, 1/16 and 1/4 area pixels of the ciphertext image of "Kodim13" to 0 and then decrypted it with the correct key, and the results are shown in Figure 5.12. As the image size of "Kodim13" is 512×768, the sizes of the blocking area are 64×96, 128×192 and 256×384 respectively. As can be seen, we can observe that more and more information is lost when the blocking area increases. That is mainly because the backbone we used is the AE-based model and it is sensitive to the changes in input. This can be perfected in later work by using the deep architecture GAN.

(a) (b) (c) (d)

Figure 5.12: Clipping attack result. (a) Original image and the decryption images when meeting (b) 1/64, (c)1/16, (d) 1/4 blocking.

(2) Noise attack: To test the performance when meeting the noise attack, we add different levels of salt and pepper noise for experimentation. Figure 5.13 shows decryption images after adding salt and pepper noise with intensities 0, 0.01, 0.05 and 0.1. Compared with the plaintext image, the decrypted image can only display some color and edge information about the image. Since the encryption key is embedded in the encrypted image, changes in the encryption image may impede image recovery. Also, the deep architecture is sensitive to the changes in the image.



(a) (b) (c) (d)

Figure 5.13: Noise attack result. (a) Original image, and the salt and pepper noise decryption results with the intensity of (b) 0.01, (c) 0.05, (d) 0.1.

5.5.6 Time Efficiency Analysis

In this part, the encryption efficiency of our proposed scheme is analyzed. The tested 24 images are from the publicly Kodak dataset. The size of images in this dataset is 512×768 or 768×512. The mean encryption speed of different encryption schemes is shown in Table 5.6. Ref[30] needs the least computational time since [30] proposed an encryption-then-lossy-compression scheme. In [30], images are encrypted by the modulo-256 addition method which is low complexity. Since deep architecture only spends a lot of time on training, the running speed of our proposed method is not very slow. And the time is mostly spent on permutation operations which is important for security, while Ref [45] has the same problem.

Table 5.6: Encryption efficiency with different schemes.

	Proposed Model	Ref [30]	Ref [45]
Speed(s)	2.78	0.29	4.33

5.6 Retrieval Accuracy on Encrypted Images

The retrieval performance evaluation is conducted on the image database Corel-10k [72] which contain 1k/10k color JPEG images in 10/100 categories. And a sub-dataset from ImageNet 2012 Large-Scale Visual Recognition Challenge dataset [77] is used for training retrieval model which contains more than 80k images with 100 classes. We choose ResNet-34 [78] as the backbone. The stochastic gradient descent (SGD) optimizer is set with an initial learning rate of 0.1, a momentum of 0.9, and a weight decay of 4e-5. And the learning rate decays by 0.1 every 50 epochs.

To compare the proposed retrieval model with other methods, we adopt the Top-k precision for evaluation with k=5, 10, 15, 20. The retrieval performance evaluation is conducted on the image database Corel-10k. Since the compression degree can affect the retrieval accuracy which is mentioned in Section 4.6, We change the value of β to get different compression ratios and the performance with different compression degrees is shown in Figure 5.14. The parameter β of the model is within the set {0.0018, 0.0035, 0.007, 0.015, 0.03, 0.045, 0.09, 0.18}. The value of β is higher, the image quality is better and thus the retrieval accuracy is better.

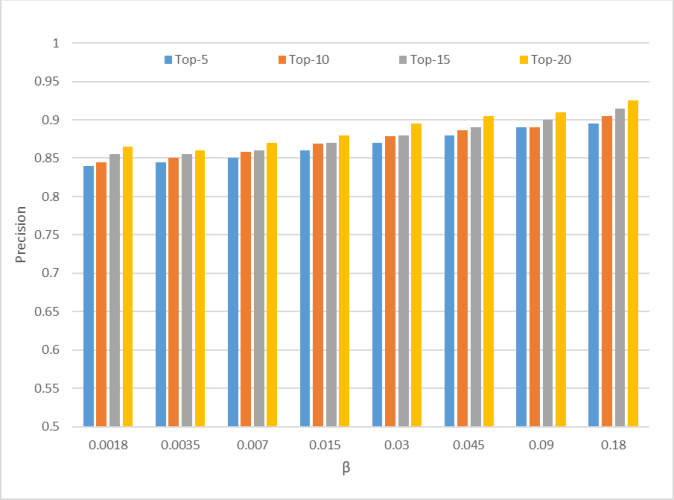


Figure 5.14: Top-k precision (k=5, 10, 15, 20) with different compression degrees.

Figure 5.15 shows the precision of retrieval accuracy of different encrypted image retrieval schemes. The result of the unencrypted compressed representation with the proposed retrieval mode is also shown. From Figure 5.15, it is obvious that the proposed deep retrieval obtains the best performance in the case of encrypted inputs. And our proposed method is very close

to the results with unencrypted images, which makes it seem that the compressed and encrypted representations meet the requirement of availability.

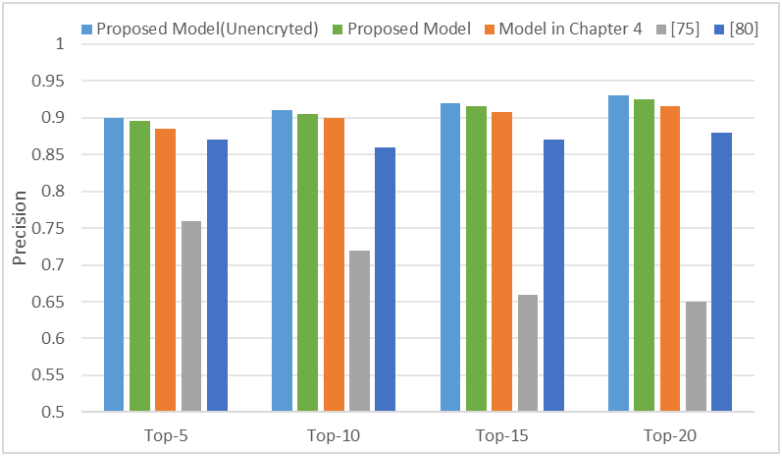


Figure 5.15: Top-k precision (k=5, 10, 15, 20) of different encryption and retrieval schemes.

5.7 Summary

This chapter proposes a novel deep joint compression-encryption model to achieve a good balance between encryption performance and compression efficiency. And with the encrypted and compressed representation, an exellent retrieval result is obtained. Here, the AE-based compression network as the base network architecture in our compression and encrytion model. When deep features are encrypted and decrypted, the plain image can still be reconstructed through a decoder that meets a cryptographic system's basic requirements. So, we encrypt images during compressing when using deep architecture. The encryption keys are from the plain images, and the encryption operations are performed on latent representations during compression, which can protect images with a high-security level. And part parameters of the deep model are replaced for higher visual security. The embedding key controls the key embedding process to save the transmitting cost when the plain images change. Moreover, attention scheme is introduced to estimate the parameters of the learned entropy model to achieve more effective compression. The proposed encryption scheme obtains a high protection ability from the experiment results with high compression efficiency. And from the evaluation of retrieval accuracy, our model can enable encrypted image retrieval well. The compressed and encrypted representation from the proposed model can ensure compression-friendliness, privacy security and availability for retrieval.

Chapter 6 Conclusion and Future Work

6.1 Conclusion

In this thesis, we first introduce the background of image compression, image encryption and encrypted image retrieval. And the necessity and challenge of encrypted image retrieval have been discussed. Then our contributions in realizing encrypted image retrieval are introduced to handle the conflict between encryption and retrieval. Three different encrypted image retrieval schemes are proposed which can broadly categorize into pixel based method, DCT based method and learning based method. For those three methods, learning based method performs the best both on security and compression, but it has some implementation issues like we need to train the network for different compression ratio. The DCT based method performs next to the learning based method but it is compression friendly that can be implemented in the existing systems without much extra work, since we introduce encryption operations into the compression stage. And for the pixel based method, we can find that the security level is not so strong due to the block-based encryption operations but it is also compression-friendly.

In Chapter 2, some basic relate methods are presented which include image compression techniques, image encryption techniques and encrypted image retrieval techniques. For image compression techniques, traditional schemes and deep models are briefly introduced in this chapter. Various image compression and encryption schemes are introduced, and these algorithms can be categorized into three classes: encryption-then-compression scheme, simultaneous compression encryption scheme and compression-then-encryption scheme. The advantages and limitations are also discussed.

Chapter 3 presents a privacy-preserving content-based image retrieval scheme, which extracts features from the content of encrypted images. To achieve privacy protection, two-level sequence permutation is conducted on pixels in each 8×8 block. Pixels in the block are represented with 8-bit binary sequence first. Then, more significant 4-bit binary sequence of the pixel is confused by block permutation, while intra-block permutation is conducted on the less significant 4-bit binary sequence. After encryption on binary sequence, the image confusion is used by permutation to increase image security and the index is generated from logistic map. This block based permutation operation can guarantee local feature extraction,

further improving the image security and retrieval accuracy. The histogram features for retrieval can be directly extracted from encrypted blocks. And the retrieval accuracy and image security of the proposed method are discussed. The factors affecting the performance of the algorithm are mentioned. The security is limited by the block permutation operation since it cannot remove spatial relationships well. And adding the value substitution operation may improve the security performance of the method.

Chapter 4 presents a encrypted JPEG image retrieval scheme based on DCT coefficients, which encrypts images during the JPEG compression process by coefficient value substitution and intra-block pixel permutation on coefficients. DC coefficients are all substituted by new values and the replacement is determined by the original coefficient values. Intra-block pixel permutation on coefficients is conducted for further security. After encryption, our proposed learning approach utilizes ResNet-34 as the backbone to extract features while accepting encrypted DCT coefficients as input to consume less computation and communication resources. And a Siamese architecture with triplet loss is used to produce a global representation that is well-suited to image retrieval. Experiment results show that our method can achieve higher accuracy than other related schemes and security requirements.

Chapter 5 presents a deep encryption and retrieval scheme that introduces end-to-end learning to the security system, which can achieve a good balance between encryption performance and compression efficiency. And with the encrypted and compressed representation obtained from proposed model, an excellent retrieval result is obtained. Here, the AE-based compression network is the backbone of the our compression and encryption model. When deep features are encrypted, the plain image can still be reconstructed through a decoder that meets a cryptographic system's basic requirements. So, the encryption operations are conducted on latent representations during deep compression processing. The encryption keys are from the plain images, which can protect images with a high-security level. And part parameters of the deep model are replaced for higher visual security. The embedding key controls the key embedding process to save the transmitting cost when the plain images change. Moreover, an attention scheme is introduced to improve compression performance. The proposed encryption scheme obtains a high protection ability from the experiment results with high compression efficiency. After generating the encrypted and compressed representation, a deep retrieval model is introduced which is training with ranked list loss and cross-entropy loss. And from the evaluation of retrieval accuracy, the model can enable encrypted image retrieval well. The

compressed and encrypted representation from the proposed model meets the requirements in compression-friendliness, privacy security and availability.

6.2 Future Work

Encrypted image retrieval is a complex research topic since privacy security, compression efficiency and retrieval performance are all needed to be considered. Based on the methods proposed in this thesis, there are still several parts can be further studied. First, various block sizes can be considered since 8×8 block is commonly used in JPEG based compression. And improve the capability of block operation to against attacks can also be considered. Second, more effective features can be captured from DCT coefficients with different encryption operations and the new model can be regarded as a retrieval-oriented encryption scheme. Third, different deep architecture can be considered for compression and encryption since the proposed model needs to be retrained to get the results from different compression rates. Forth, some special image categories can be considered, such as very low-resolution images. Finally, other deep architecture can be considered since the backbone we used in this thesis are all based on the spatial domain.

References

[1] W. Lu, A. L. Varna, A. Swaminathan, and M. Wu, "Secure image retrieval through feature protection," in *International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2009, pp. 1533-1536.

[2] K. Huang, M. Xu, S. Fu, and D. Wang, "Efficient privacy-preserving content-based image retrieval in the cloud," in *Web-Age Information Management: 17th International Conference*, 2016, pp. 28-39.

[3] Y. Wang, M. Miao, J. Shen, and J. Wang. "Towards efficient privacy-preserving encrypted image search in cloud computing, " *Soft Computing*, vol 23, pp. 2101–2112, 2019.

[4] H. Cheng, C. Weng, and Y. Yang, "TPEIP: Thumbnail preserving encryption based on sum preserving for image privacy," *Journal of Information Security and Applications*, vol 70, 103352, 2022.

[5] P. Li and K. -T. Lo, "Joint image compression and encryption based on alternating transforms with quality control," in *Visual Communications and Image Processing (VCIP)*, 2015, pp. 1-4.

[6] T. Chuman, W. Sirichotedumrong and H. Kiya, "Encryption-Then-Compression Systems Using Grayscale-Based Image Encryption for JPEG Images," in *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1515-1525, 2019.

[7] H. Hu, Y. Cao, J. Xu, C. Ma, and H. Yan, "An Image Compression and Encryption Algorithm Based on the Fractional-Order Simplest Chaotic Circuit," *IEEE Access*, vol. 9, pp. 22141-22155, 2021.

[8] R. Ni, F. Wang, J. Wang and Y. Hu, "Multi-Image Encryption Based on Compressed Sensing and Deep Learning in Optical Gyrator Domain," *IEEE Photonics Journal*, vol. 13, no. 3, pp. 1-16, June 2021, Art no. 7800116.

[9] K. A. Suhail and S.Sankar, "Image compression and encryption combining autoencoder and chaotic logistic map," *Iranian Journal of Science and Technology, Transactions A: Science*, vol. 44, pp. 1091-1100, 2020.

[10] N. T. Bani and S. Fekri-Ershad, "Content-based image retrieval based on combination of texture and colour information extracted in spatial and frequency domains", *The Electronic Library*, Vol. 37 No. 4, pp. 650-666.

[11] A. Preethy Byju, B. Demir and L. Bruzzone, "A Progressive Content-Based Image Retrieval in JPEG 2000 Compressed Remote Sensing Archives," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 8, pp. 5739-5751, 2020.

[12] Q. Zhang, D. Liu and H. Li, "Deep network-based image coding for simultaneous compression and retrieval," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 405-409.

[13] S. Temburwar, B. Rajesh, and M. Javed, M, "Deep Learning-Based Image Retrieval in the JPEG Compressed Domain," in *Advanced Machine Intelligence and Signal Processing*, 2022, pp. 351-363.

[14] "ISO/IEC JTC 1/SC 29/WG 1, JPEG Privacy and Security Call for Proposals," The JPEG Committee, 2017, available at https://jpeg.org/ downloads/privacy and security/wg1n74015 Draft CfP.pdf.

[15] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations*, pp. 1-10, 2018.

[16] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimization of nonlinear transform codes for perceptual quality," in *2016 Picture Coding Symposium (PCS)*, 2016, pp. 1-5.

[17] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Advances in Neural Information Processing Systems*, 2018, pp. 10771-10780.

[18] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned Image Compression With Discretized Gaussian Mixture Likelihoods and Attention Modules," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7936-7945.

[19] G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. Covell, "Full Resolution Image Compression with Recurrent Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5306-5314.

[20] K. Islam, L. M. Dang, S. Lee, and H. Moon, "Image Compression with Recurrent Neural Network and Generalized Divisive Normalization, " in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1875-1879.

[21] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool, "Generative Adversarial Networks for Extreme Learned Image Compression," in *IEEE International Conference on Computer Vision*, 2019, pp. 221-231.

[22] T. Park, M. Y. Liu, T. C. Wang, and J. Y. Zhu, "Semantic Image Synthesis With Spatially-Adaptive Normalization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2332-2341.

[23] F. Huszár, L. Theis, W. Shi, and A. Cunningham, "Lossy image compression with compressive autoencoders," in *International Conference on Learning Representations*, 2017, pp. 1-19.

[24] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, " Soft-to-hard vector quantization for end-to-end learning compressible representations, " *Advances in Neural Information Processing Systems*, 2017, pp. 1141-1151.

[25] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. V. Gool, "Conditional Probability Models for Deep Image Compression," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4394-4402.

[26] M. Johnson, P. Ishwar, V. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data," *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 2992–3006, 2004.

[27] A. A. Kumar and A. Makur, "Lossy compression of encrypted image by compressive sensing technique," in *TENCON 2009 - 2009 IEEE Region 10 Conference*, Singapore, 2009, pp. 1-5.

[28] B. Zhang, D. Xiao, and Y. Xiang, "Robust Coding of Encrypted Images via 2D Compressed Sensing," *IEEE Transactions on Multimedia*, vol. 23, pp. 2656-2671, 2021.

[29] J. Zhou, O. C. Au, G. Zhai, Y. Tang, and X. Liu, "Scalable compression of stream cipher encrypted images through context-adaptive sampling," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 11, pp. 1857-1868, 2014.

[30] C. Wang, T. Zhang, H. Chen, Q. Huang, J. Ni, and X. Zhang, "A Novel Encryption-Then-Lossy-Compression Scheme of Color Images Using Customized Residual Dense Spatial Network," *IEEE Transactions on Multimedia*, vol. 25, pp. 4026-4040, 2023.

[31] X. Zhang, G. Feng, Y. Ren, and Z. Qian, "Scalable Coding of Encrypted Images," *IEEE Transactions on Image Processing*, vol. 21, no. 6, pp. 3108-3114, 2012.

[32] C. Qin, Q. Zhou, F. Cao, J. Dong, and X. Zhang, "Flexible lossy compression for selective encrypted image with image inpainting," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 11, pp. 3341–3355, 2019.

[33] J. Uthayakumar, T. Vengattaraman and P. Dhavachelvan, "A survey on data compression techniques: From the perspective of data quality coding schemes data type and

applications," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, pp.119-140, 2021.

[34] L. Zhou, C. Cai, Y. Gao, S. Su, and J. Wu, "Variational autoencoder for low bit-rate image compression," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2617-2620, Jun. 2018.

[35] F. Yang, J. Mou, K. Sun, and R. Chu, "Lossless image compression-encryption algorithm based on BP neural network and chaotic system, " in *Multimedia Tools and Applications*, vol. 79, pp. 19963-19992, 2020.

[36] R. Ni, F. Wang, J. Wang, and Y. Hu, "Multi-Image Encryption Based on Compressed Sensing and Deep Learning in Optical Gyrator Domain," *IEEE Photonics Journal*, vol. 13, no. 3, pp. 1-16, 2021, Art no. 7800116.

[37] N. Zhou, A. Zhang, J. Wu, D. Pei, and Y. Yang, "Novel hybrid image compression–encryption algorithm based on compressive sensing," *Optics & Laser Technology*, vol. 125, no. 18, pp. 5075-5080, 2014.

[38] N. Zhou, S. Pan, S. Cheng, and Z. Zhou, "Image compression–encryption scheme based on hyper-chaotic system and 2D compressive sensing," *Optics & Laser Technology*, vol. 82, pp. 121-133, 2016.

[39] 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. 11, pp. 72-79, 2018.

[40] M. Zhang and X. Tong, "Joint image encryption and compression scheme based on IWT and SPIHT," *Optics and Lasers in Engineering*, vol. 90, pp. 254-274, 2017.

[41] X. Chai, Z. Gan, Y. Chen, and Y. Zhang, "A visually secure image encryption scheme based on compressive sensing, " *Signal Processing*, vol. 134, pp. 35-51, 2017.

[42] Z. Gan, J. Bi, W. Ding, and X. Chai. "Exploiting 2D compressed sensing and information entropy for secure color image compression and encryption," *Neural Computing & Applications*, vol. 33, pp. 12845-12867, 2021.

[43] A. Ghaffari, "Image compression-encryption method based on two-dimensional sparse recovery and chaotic system," *Scientific Reports*, no. 369, 2021.

[44] P. Li and K.-T. Lo, "Joint image compression and encryption based on order-8 alternating transforms, " *Journal of Visual Communication and Image Representation*, vol. 44, pp. 61-71, 2017.

[45] P. Li and K. -T. Lo, "A Content-Adaptive Joint Image Compression and Encryption Scheme," *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 1960-1972, 2018.

[46] P. Li and K.-T. Lo, "Joint image encryption and compression schemes based on 16× 16 DCT," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 12-24, 2019,

[47] D. Huo, Z. Zhu, L. Wei, C. Han, and X. Zhou, "A visually secure image encryption scheme based on 2D compressive sensing and integer wavelet transform embedding, " *Optics Communications*, vol. 492, 2021.

[48] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, "Enabling search over encrypted multimedia databases," in *Proceedings IS&T/SPIE Electronic Imaging*, 2009, pp. 725418.

[49] W. Lu, A. L. Varna, A. Swaminathan, and M. Wu, "Secure image retrieval through feature protection," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1533-1536.

[50] Z. Xia, N. N. Xiong, A. V. Vasilakos, and X. Sun, "EPCBIR: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing," *Information Sciences*, vol. 387, pp. 195-204, 2017.

[51] W. Lu, A. L. Varna, and M. Wu, "Confidentiality-preserving image search: a comparative study between homomorphic encryption and distance-preserving randomization," *IEEE Access*, 2014, vol. 2, pp. 125-141.

[52] Z. Xia, Y. Zhu, X. Sun, Z. Qin and K. Ren, "Towards Privacy-Preserving Content-Based Image Retrieval in Cloud Computing," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 276-286, 2018.

[53] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun and K. Ren, "A Privacy-Preserving and Copy-Deterrence Content-Based Image Retrieval Scheme in Cloud Computing," in *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594-2608, 2016.

[54] K. Huang, M. Xu, S. Fu, and D. Wang, "Efficient privacy preserving content-based image retrieval in the cloud, " in *Web-Age Information Management*, 2016, pp. 28-39.

[55] X. Zhang and H. Cheng, "Histogram-based retrieval for encrypted JPEG images," in *2014 IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP)*, 2014, pp. 446-449.

[56] B. Ferreira, J. Rodrigues, J. Leitão and H. Domingos, "Privacy-Preserving Content-Based Image Retrieval in the Cloud," in *2015 IEEE 34th Symposium on Reliable Distributed Systems (SRDS)*, 2015.

[57] J.-K. Chou, C.-K. Yang, and H.-C. Chang, "Encryption domain content-based image retrieval and convolution through a block-based transformation algorithm," *Multimedia Tools and Applications*, vol. 75, pp. 13805-13832, 2016

[58] H. Cheng, X. Zhang, and J. Yu, "AC-coefficient histogram-based retrieval for encrypted JPEG images," *Multimedia Tools and Applications*, vol. 75, pp. 13791-13803, 2015.

[59] H. Cheng, X. Zhang, J. Yu, and F. Li, "Markov Process Based Retrieval for Encrypted JPEG Images," in *2015 10th International Conference on Availability, Reliability and Security*, 2015, pp. 417-421.

[60] H. Cheng, X. Zhang, J. Yu, and Y. Zhang, "Encrypted JPEG image retrieval using block-wise feature comparison," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 111-117, 2016.

[61] Y. Xu, J. Gong, L. Xiong, L. Xiong, Z. Xu, J. Wang, and Y.-Q. Shi, "A privacy-preserving content-based image retrieval method in cloud environment, " *Journal of Visual Communication and Image Representation*, vol. 43, pp. 164 -172, 2017.

[62] H. Liang, X. Zhang, and H. Cheng, "Huffman-code based retrieval for encrypted JPEG images," *Journal of Visual Communication and Image Representation*, vol. 61, pp. 149-156, 2019.

[63] H. Liang, X. Zhang, H. Cheng and Q. Wei, "Secure and efficient image retrieval over encrypted cloud data," *Security and Communication Networks*, vol. 2018, pp. 1-14, 2018.

[64] M. Shen, G. Cheng, L. Zhu, X. Du, and J. Hu, "Content-based multi-source encrypted image retrieval in clouds with privacy preservation," *Future Generation Computer Systems*, vol. 109, pp. 621-632, 2020.

[65] J. Qin, H. Li, X. Xiang, Y. Tan, W. Pan, W. Ma, and N. N. Xiong, "An encrypted image retrieval method based on harris corner optimization and LSH in cloud computing," *IEEE Access*, vol. 7, pp. 24626-24633, 2019.

[66] A. Du, L. Wang, S. Cheng, and N. Ao, "A privacy-protected image retrieval scheme for fast and secure image search," *Symmetry*, vol. 12, no. 2, 2020, Art. no. 282.

[67] J. Yuan, S. Yu, and L. Guo, "SEISA: Secure and efficient encrypted image search with access control," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2083-2091.

[68] X. Li, Q. Xue and M. C. Chuah, "CASHEIRS: Cloud assisted scalable hierarchical encrypted based image retrieval system," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1-9,

[69] Z. Huang, M. Zhang, and Y. Zhang, "Toward efficient encrypted image retrieval in cloud environment," *IEEE Access*, vol. 7, pp. 174541–174550, 2019.

[70] Q. Feng, P. Li, Z. Lu, G. Liu and F. Huang, "End-to-end Learning for Encrypted Image Retrieval," in *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2021, pp. 1839-1845.

[71] R. A. Fisher and F. Yates, Statistical tables for biological agricultural and medical research, 6th ed., rev. & enl.. Harlow: Longman, 1963.

[72] Eastman Kodak, Kodak Lossless True Color Image Suite (PhotoCD PCD0992). URL: http: //r0k.us/graphics/kodak/.

[73] C. Pak and L. Huang, "A new color image encryption using combination of the 1D chaotic map," *Signal Processing*, vol. 138, pp. 129-137, 2017.

[74] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947-963, 2001.

[75] B. Ferreira, J. Rodrigues, J. Leitão, H. Domingos, "Towards an image encryption scheme with content-based image retrieval properties, " in *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance*, 2015, pp. 311–318.

[76] D. Liu, J. Shen, Z. Xia, and X.M. Sun, "A content-based image retrieval scheme using an encrypted difference histogram in cloud computing," *Information*, vol. 8, no. 3, pp. 96, 2017.

[77] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248-255.

[78] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition, " in *Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

[79] J. Philbin, O. Chum, M. Isard, J. Sivic and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1-8.

[80] Y. Su, Y. Wo, and G. Han, "Reversible cellular automata image encryption for similarity search, " *Signal Processing: Image Communication*, vol. 72, pp. 134147, 2019.

[81] Z. Xia, L. Jiang, D. Liu, L. Lu, and B. Jeon, "BOEW: A Content-Based Image Retrieval Scheme Using Bag-of-Encrypted-Words in Cloud Computing" *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 202-214, 2022.

[82] Z. Cheng, H. Sun, M. Takeuchi and J. Katto, "Deep Residual Learning for Image Compression, " in *Computer Vision and Pattern Recognition*, 2019, pp. 1-4.

[83] Y. Ding, G. Wu, D. Chen, N. Zhang, L. Gong, M. Cao, and Z. Qin, "DeepEDN: A Deep-Learning-Based Image Encryption and Decryption Network for Internet of Medical Things," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1504-1518, 2021.

[84] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, " in *32nd International Conference on Machine Learning*, 2015, pp. 448-456.

[85] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module, " in *European Conference on Computer Vision*, 2018, pp. 3-19.

[86] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang and X. Tang, "Residual Attention Network for Image Classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6450-6458.

[87] X. Wang, Y. Hua, E. Kodirov, G. Hu, R. Garnier, and N. M. Robertson, "Ranked list loss for deep metric learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5207-5216.

[88] T. Xue, B. Chen, J. Wu, D. Wei, and W. T Freeman, " Video enhancement with task-oriented flow, " *International Journal of Computer Vision*, vol. 127, pp. 1106-1125, 2019.