End-to-end Learning for Encrypted Image Retrieval

Qihua Feng, Peiya Li *, ZhiXun Lu, Guan Liu, Feiran Huang
Jinan University, Guangzhou, China
* E-mail: lpy0303@jnu.edu.cn

***Abstract*—Traditional image encryption methods impede the process of extracting features from cipher-images, which makes next step's retrieval become difficult. There are some encrypted image retrieval works that extract features from encrypted images by human initially, then build model to enforce retrieval by these features. In the paper, we propose end-to-end encrypted image retrieval, using deep learning model to extract features from cipher-images and conducting retrieval. We do not need to extract hand-craft features from cipher-images, because our retrieval model can extract features by end-to-end learning. Images are encrypted by block permutation and color value substitution operation for partial blocks. Our retrieval model uses Vision Transformer (ViT) as BackBone, combines triplet loss and cross entropy loss when training. In experiments, we compare different BackBones — ViT and ResNet50, and the results show that the retrieval performance is better when using ViT as BackBone. Our scheme not only achieves end-to-end encrypted image retrieval, but also obtains significant improvement on retrieval performance when compared with current methods.**

I. Introduction

Image retrieval is common in our daily life, but nowadays people gradually pay more attention to data security. People upload the images to cloud servers because of the fast growth of the image data, in which they can conveniently retrieve the images from the servers. But image privacy becomes huge concern when the images are outsourced to the servers. So encrypted image retrieval has attracted many researchers, and many related works have been proposed in recent years [1–6]. These works can be mainly divided into two categories: one-stage [4–6] and two-stage [1–3]. Two-stage means to extract features from plain-images, then encrypt features and images. Differently, one-stage only needs to encrypt images, then extract features from cipher-images. It is apparently that two-stage is more inconvenience and causes extra computational workload for user than one-stage. Therefore, in this paper, we focus on one-stage which extracts features directly from cipher-images.

The crucial problem is how to extract features effectively from encrypted images for one-stage method, and many researchers have proposed methods to deal with this problem. Zhang et al. [7] proposed to encrypt images by permuting discrete cosine transform (DCT) coefficients, then extract these coefficients' histogram to perform retrieval. Li et al. [8] proposed a new block transform encryption method using new orthogonal transforms rather than 8×8 DCT. Liang et al. [9] proposed to encrypt images by stream cipher and permutation cipher, then extract Huffman-code histogram features to conduct retrieval. Xia et al. [10] encrypted DC coefficients by stream cipher on the Y component and encrypted U and V components by value permutation and position scrambling, the AC-coefficients histogram of Y component and color histograms of U and V components were extracted when retrieval. In [11], the image was divided into two different components by Gaussian orthogonal matrix, one component was encrypted by Advanced Encryption Standard (AES), and the other component was unencrypted which was used to extract features. The above works calculated distances of features which did not use learning algorithm to conduct retrieval, therefore some works proposed to build learnable model to conduct retrieval. Cheng et al. [12] proposed to encrypt images by cipher and permutation encryption, then extract features from cipher-images by Markov process. In addition, they used support vector machine (SVM) to conduct retrieval with extracted features. Xia et al. [13] extracted local color histogram features from cipherimages which were encrypted by color value substitution and permutation encryption, then they built bag-of-encrypted-words (BOEW) model to achieve retrieval. In [14], images were encrypted by AES and local random features were extracted for image retrieval, all local features were clustered by K-means algorithm to form the visual word. Xia et al. [15] proposed to extract secure Local Binary Pattern (LBP) features, then build Bag-of-Words (BOW) model to conduct retrieval. These works [12–15] used learning algorithm to build model to conduct retrieval, but they extracted features by human initially, which are obviously not end-to-end learning model.

Compared with traditional machine learning algorithms, deep learning does not need to extract hand-craft features, which can achieve end-to-end learning. In this paper, we use deep learning related techniques to extract features from cipher-images rather than extracting features by ourselves. We encrypt images by block permutation and color value substitution for partial blocks. In our retrieval model, we use vision Transformer (ViT) [16] as BackBone to extract features, and our model combines triplet loss [17] and cross entropy (CE) loss. Using encrypted images as inputs, we can train a deep neural network model to perform retrieval. In summary, the contributions of this work are concluded as follows:

1) To the best of our knowledge, we are the first to propose end-to-end learning for encrypted image retrieval, which uses deep learning to extract features directly from cipher-images and conduct retrieval. Our retrieval model uses ViT as BackBone and combines triplet loss and CE loss, the experiments demonstrate that the retrieval performance of our scheme improves greatly than current

encrypted image retrieval works which are also one-stage methods.

2) We explore the retrieval performance with different encryption parameters. We select partial blocks to apply color value replacement after block permutation, and the experiments show that when the number of selected blocks are not more than three fourths, our model can obtain well retrieval performance.
3) We compare the retrieval performance with different BackBones — ViT and ResNet50, and the experiments show that ViT is more fit to our encryption method than ResNet50 [18] which is a typical model of convolutional neural network (CNN).

II. Proposed Scheme

Our scheme contains three parts: content owner, server and authorized user. In order to protect privacy, the content owner encrypts images and trains a retrieval model by these cipher-images, then stores the encrypted images and trained model in the server. When an authorized user needs to retrieve images, he/she only needs to provide the encrypted query image to the server. Then the server takes the query as the input of the model, and returns similar cipher-images to the authorized user. Finally, the authorized user decrypts and obtains the corresponding plain-images. The process mainly includes image encryption and image retrieval, which will be presented in detail as follows.

A. Image encryption

Our encryption method includes two steps: block permutation and color value substitution for partial blocks. The pseudo-random key generator we select is BLAKE2. The sketch of our encryption process is shown in Fig. 1(a). Firstly, we divide the plain-image (I) into non-overlapping blocks, the block size is $P \times P$, the width of image is W and the height of image is H, so the number of blocks (denoted as $blknum$) is $\frac{W \times H}{P^2}$. We generate randomly block position sequence (denoted as $perm$) by the encryption key key_{perm} [19]. The encrypted image ($I^{'}$) through block permutation can be defined as follows:

$$I^{'}[i] = I[perm[i]], \qquad 1 \leq i \leq blknum \tag{1}$$

where i is block position. The second step is color value substitution for partial blocks, we generate random replacement sequences (denoted as $subs$) by the encryption key $\{key_{subs,*}\}_{*\in\{R,G,B\}}$ [19]. Suppose the original value sequence is $oris = [0, 1, 2, \ldots, 254, 255]$, color value substitution can be calculated as follows:

$$pv^{'} = subs[pv], \qquad pv \in oris \tag{2}$$

where pv is the original pixel value, $pv^{'}$ is the corresponding pixel in the encrypted image, we take an example in Fig. 2. M ($0<M<blknum$) blocks are selected to apply value replacement encryption. We use encryption key key_{select} to generate randomly block positions [19], then select the top M blocks from the block positions. It is noted that the $\{key_{subs,*}\}_{*\in\{R,G,B\}}$ are different for different components (R/G/B), while the key_{select} and key_{perm} are same for different components.

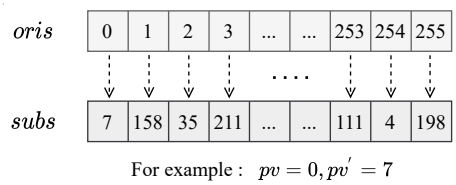
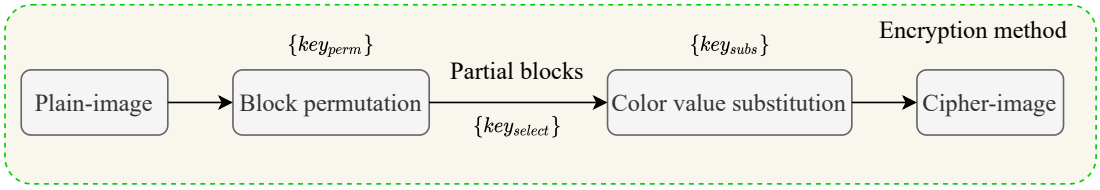
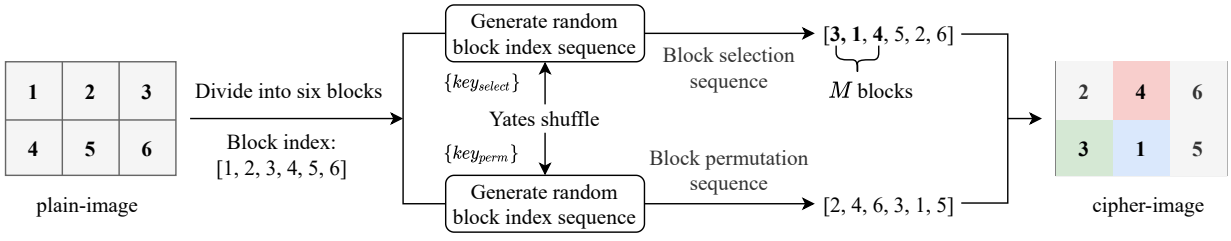


Fig. 2. An example of value replacement.

An example of our encryption method is illustrated in Fig. 1(b). Suppose we divide the plain-image into six blocks, and



(a) A sketch of our encryption method.



(b) An example of our encryption method.

Fig. 1. A sketch of our encryption method (a) and an exampple of our encryption method (b).

the block indexs are $[1, 2, 3, 4, 5, 6]$, then we generate block permutation sequence and block selection sequence. Suppose $M = 3$ and the top M blocks in the randomly block selection sequence is $3, 1, 4$, then these three blocks are applied to value substitution encryption.

B. Image retrieval

Since Vision Transformer (ViT) [16] divides image into non-overlapping blocks, and each block is equivalent to a word. Our encryption method also splits image into blocks, and we use ViT as BackBone to extract features from cipher-image rather than CNN, and the experiments show that ViT is more fit to our encryption method than CNN.

1) Loss: The retrieval model aims at learning representations of encrypted images, then calculates the distances among these representations when retrieval. Image retrieval is a typical type of deep metric learning [20], and the most common loss function in deep metric learning is triplet loss [17]. The triplet loss is defined as follows:

$$L_{Tri} = max(0, d_{positive} - d_{negative} + m) \tag{3}$$

where $d_{positive}$ and $d_{negative}$ are deep feature distances of positive pair and negative pair, m is the margin of triplet loss. The goal of triplet loss is to make the intra-class compactness and the inter-class separability in the embedding space. But triplet loss cannot provide globally optimal constraint, our model loss combines triplet loss with cross entropy (CE) loss, which is defined as follows:

$$\begin{aligned} L_{CE} &= -\sum_{i=1}^{N} p_i log(q_i) \\ p_i &= \begin{cases} 1, i = y \\ 0, i \neq y \end{cases} \end{aligned} \tag{4}$$

where q_i is the predicted probability of class i, y is the real label, and N is the number of classes. The combination of triplet loss and CE loss can learn more discriminative features, and the total loss of our model can be defined as follows:

$$L_{total} = L_{Tri} + \alpha * L_{CE} \tag{5}$$

where α is the weight factor of CE loss.

2) Network Architecture: As shown in Fig. 3, our model includes two parts: BackBone and Head. The aim of BackBone is to extract features from cipher-images, and our BackBone refers to ViT [16]. The ViT is inspired by the standard Transformer [21], which often deals with natural language processing (NLP) tasks. We split cipher-image into non-overlapping blocks which are treated as words like NLP application, and the block size is $P \times P$, which is the same as the block size in block permutation encryption. Then we flatten the blocks and do a learnable linear projection [16], the outputs of this linear project are block embeddings. Class embedding [16, 22] is prepended to the sequence of block embeddings, which can learn the representations of cipher-images. In order to keep positional information, position embeddings [16, 21] are added to the block embeddings, then these resulting embeddings (z_0) are served as inputs of the Transformer encoder [16]. The Transformer encoder contains stacked encoders, and the number of encoders is L, the lth encoder can be calculated as:

$$z_l^{'} = MSA(LN(z_{l-1})) + z_{l-1}, \qquad l = 1, 2 \ldots L \tag{6}$$

$$z_l = MLP(LN(z_l^{'})) + z_l^{'}, \qquad l = 1, 2 \ldots L \tag{7}$$

where MSA is multi-head self-attention [21], and MLP is multi-layer perceptron blocks [16], LN is layer normalization [23]. The output of Transformer encoder is z_L, and the corresponding class embedding is z_L^*.

The module Head contains two fully connected (FC) layers, and acts as classification. Through the first fully connected

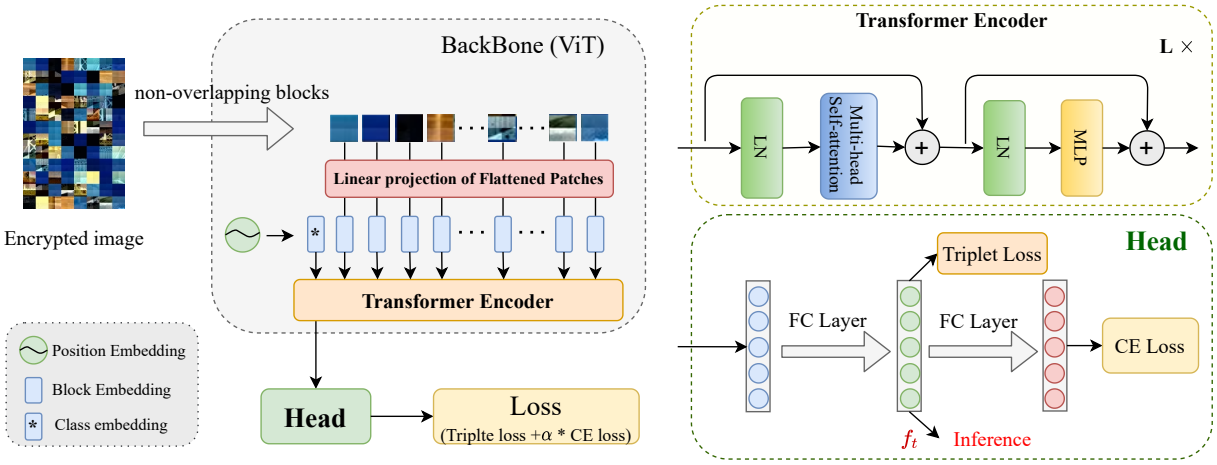


Fig. 3. The overview of our proposed retrieval model (We use ViT [16] as BackBone to extract features from cipher-image; The Transformer Encoder and Head module are detailed on the right; Our loss combine triplet loss and CE loss).

layer, we can learn deep feature f_t. The module Head can be defined as follows:

$$f_t = \sigma(\boldsymbol{W}_1(LN(z_L^*))) \tag{8}$$

$$pred = \boldsymbol{W}_2 f_t \tag{9}$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2$ are matrices about fully connected layer, σ is activation function which we apply Tanh [24]. In the training stage, f_t is used to compute triplet loss, and the output of the model returns cross entropy (CE) loss. In the stage of inference, we calculate the cosine distances of deep features f_t, then rank the distances and return top K similar cipher-images. Our retrieval model is end-to-end in the training stage which can extract features from cipher-images by itself, unlike [7–15] which extract features from cipher-images by human, then calculate distances of features or build machine learning model to do retrieval.

III. EXPERIMENTS

In this section, we demonstrate the performance of our proposed scheme by using dataset Corel10K [25]. The Corel10K dataset has 10000 images and 100 classes, each class has 100 images. The size of images are mostly 126×187 or 187×126. Our programming language is Python, and the retrieval model is implemented on the PyTorch [26] platform.

1) Encryption performance: As mentioned in Section II-A, the block size of block permutation is $P \times P$ and we apply color value replacement on M blocks. In experiments, we compare two different block sizes, and set $P \in \{8, 16\}$. In order to make the image dimensions be multiples of 16, we resize images to 128×192 or 192×128. So when $P = 8$, the number of total blocks is 384, when $P = 16$, the number of total blocks is 96. For the parameter M, we select three different M values which are one quarter, one half and three quarters of the total number of blocks ($blknum$) respectively, namely $\frac{M}{blknum} \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}\}$. Specifically, we select $M \in \{24, 48, 72\}$ when $P = 16$, and $M \in \{96, 192, 288\}$ when $P = 8$. As shown in Fig. 4, we present the corresponding encrypted images for different encryption parameters.

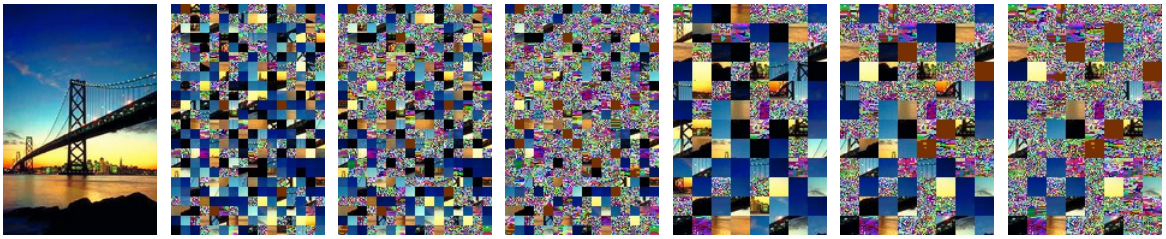
In order to evaluate the visual safety, we compare the Peak Signal-to-Noise Ratio (PSNR) values under different encryption parameters. The mean PSNR of 10000 images under various encryption parameters are calculated, and smaller PSNR indicates better visual safety. As shown in Tab. I, we can see that when $P = 16$, the visual safety gradually increases as M gets bigger, the same applies to $P = 8$. In addition, when $P = 8, M = 288$ and $P = 16, M = 72$, namely M accounts for $\frac{3}{4}$ of the total number of blocks ($blknum$), $P = 16, M = 72$ has better visual safety.

TABLE I
COMPARISON OF PSNR FOR DIFFERENT ENCRYPTION PARAMETERS

Encryption parameters		PSNR
$P = 16$	$M = 24$ ($\frac{1}{4}$)	9.754
	$M = 48$ ($\frac{1}{2}$)	9.354
	$M = 72$ ($\frac{3}{4}$)	9.032
$P = 8$	$M = 96$ ($\frac{1}{4}$)	9.814
	$M = 192$ ($\frac{1}{2}$)	9.379
	$M = 288$ ($\frac{3}{4}$)	9.061

Our encryption scheme has five keys ($\{key_{subs,*}\}_{*\in\{R,G,B\}}, key_{perm}, key_{select}$), these keys are 256-bit each, so the key space of our encryption scheme is $(2^{256})^5$, which is enough to resist the brute-force attack. In addition, the encryption space of value substitution, block permutation and block selection are $(256!)^3$, $blknum!$ and $blknum!$, respectively, thus the encryption space of our encryption scheme is $(256!)^3 \times blknum! \times blknum!$, which also is enough for resisting the brute-force attack. The decryption process of our scheme is just the reverse of image encryption operations with secret keys.

2) Retrieval performance: When training our model, we select 7000 images as training set, and the rest 3000 images as testing set. Training set has 100 classes and each class has 70 images. We train our network with the Adam [27] optimizer, and set learning rate to be 5×10^{-5},epochs to be 60, momentum to be 0.9 and weight decay to be 1×10^{-5}. Batch-size is 50, each batch-size samples 10 classes and each class randomly samples 5 images when training. Here we compare the retrieval performance of our scheme with current encrypted image retrieval works which also extract features from cipher-images, then analysis the experimental results of our proposed scheme. All schemes are tested on the same testing set, and we use the standard evaluation metric mean Average Precision (mAP) [28] for comparison, the higher mAP implies better



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

Fig. 4. Encryption examples of different encryption parameters ((a):plain-image; (b):$P = 8, M = 96$, (c):$P = 8, M = 192$; (d):$P = 8, M = 288$; (e):$P = 16, M = 24$; (f):$P = 16, M = 48$; (g):$P = 16, M = 72$).

TABLE II
RETRIEVAL PERFORMANCE COMPARISON (MAP) FOR DIFFERENT SCHEMES

Schemes		mAP
Zhang [7]		0.1478
Li [8]		0.1493
Liang [9]		0.1008
Xia [10]		0.1298
Xia [13]		0.1269
Xu [11]		0.4252
Wang [14]		0.0511
Xia [15]		0.0928
Proposed	$P = 8$, $M = 96$ ($\frac{1}{4}$)	0.5373
	$P = 8$, $M = 192$ ($\frac{1}{2}$)	0.5109
	$P = 8$, $M = 288$ ($\frac{3}{4}$)	0.4929
	$P = 16$, $M = 24$ ($\frac{1}{4}$)	0.5089
	$P = 16$, $M = 48$ ($\frac{1}{2}$)	0.4818
	$P = 16$, $M = 72$ ($\frac{3}{4}$)	0.4342

retrieval performance.

As shown in Tab. II, we can see that when $P = 8, M = 96$, our scheme can obtain the best retrieval performance with 0.5373 mAP, which is higher than other schemes. When increasing M from 96 to 288 , we can find that the mAP only decrease to 0.4929 which is also higher than other schemes. When $\frac{M}{blknum} = \frac{3}{4}$, namely $P = 8, M = 288$ and $P = 16, M = 72$, we can see that the retrieval performance is better when $P = 8, M = 288$ with 0.4929 mAP, higher than $P = 16, M = 72$ with 0.4342 mAP. There are similar results when $\frac{M}{blknum} = \frac{1}{4}$ and $\frac{M}{blknum} = \frac{1}{2}$, which means that our scheme is more fit to $P = 8$ than $P = 16$ under the same $\frac{M}{blknum}$. In addition, Xu [11] can get 0.4252 mAP, which is higher than schemes in [7–10, 13–15]. This is because that Xu [11] divided image into two parts by orthogonal decomposition, and they only encrypted one part, the other part was without encryption and was used to extract features. But this may cause serious information leakage. On the contrary, Wang [14] encrypted images by AES which provided good security, but the retrieval performance is only 0.0511 mAP on Corel10K dataset. So the encryption performance and retrieval performance may restrict each other. We also test our scheme's retrieval performance when all blocks are applied to color value replacement, namely $\frac{M}{blknum} = 1$, but only obtains 0.1 mAP. This means that in order to improve the retrieval performance for end-to-end learning scheme, some color information should be retained in encrypted images, which is achieved by sacrificing the encryption performance a little.

In Section II-B, we mention that our loss function has two hyper-parameters, m in Eq. 3 and α in Eq. 5. Here we discuss the retrieval performance of our model under different m and α. As shown in Tab. III, we test the retrieval performance on different encryption parameters with different α and m. From Tab. III, we can see that for $P = 8, M = 96$, the retrieval performance is better when $\alpha = 1, m = 0.5$; for $P = 8, M \in \{192, 288\}$, the retrieval performance is better when $\alpha = 0.5, m = 0.5$. So different parameters have different impact on the final retrieval performance.

TABLE III
THE RETRIEVAL PERFORMANCE (MAP) FOR DIFFERENT ENCRYPTION PARAMETERS WITH HYPER-PARAMETERS m, α.

	$\alpha = 1$ $m = 0.3$	$\alpha = 1$ $m = 0.5$	$\alpha = 0.5$ $m = 0.3$	$\alpha = 0.5$ $m = 0.5$
$P = 8$ $M = 96$	0.4910	**0.5373**	0.5075	0.5211
$P = 8$ $M = 192$	0.4898	0.5031	0.4994	**0.5109**
$P = 8$ $M = 288$	0.4839	0.4897	0.4768	**0.4929**
$P = 16$ $M = 24$	**0.5089**	0.4927	0.4932	0.4813
$P = 16$ $M = 48$	0.4681	0.4756	**0.4818**	0.4721
$P = 16$ $M = 72$	0.4060	**0.4342**	0.4232	0.4319

TABLE IV
THE RETRIEVAL PERFORMANCE (MAP) FOR DIFFERENT BACKBONE (VIT AND RESNET50) WHEN HYPER-PARAMETERS $m = 0.5, \alpha = 0.5$.

	BackBone	
	ViT	ResNet50
$P = 8$, $M = 96$	**0.5211**	0.2197
$P = 8$, $M = 192$	**0.5109**	0.1738
$P = 8$, $M = 288$	**0.4929**	0.1395
$P = 16$, $M = 24$	**0.4813**	0.3192
$P = 16$, $M = 48$	**0.4721**	0.2627
$P = 16$, $M = 72$	**0.4319**	0.2232

As mentioned in Section II-B, our retrieval model uses ViT as BackBone rather than CNN, now we analysis the reason. Because CNN needs to keep spatial structure of input images, which is very senstive to block permutation and color value replacement. CNN is local but ViT is more global with self-attention [16]. In Tab. IV, we compare the retrieval performance with different BackBones. Considering that ResNet50 [18] is a typical CNN model, we use ResNet50 and ViT as different BackBones. We set hyper-parameters $m = 0.5, \alpha = 0.5$, and it can be seen from Tab. IV that the retrieval performance of BackBone ViT is better than BackBone ResNet50 on our encryption method. To be specific, when $P = 8, M = 288$, the BackBone ViT can achieve 0.4929 mAP, which is higher about 36% mAP than BackBone ResNet50; When $P = 8, M = 96$, the BackBone ViT can achieve 0.5211 mAP, which is higher about 30% mAP than BackBone ResNet50. What's more, when M increases from 96 to 288, the BackBone ViT only decreases by less than 3% mAP, but BackBone ResNet50 decreases by about 8% mAP. In addition, when $P = 8, M = 96$ and $P = 16, M = 24$, namely $\frac{M}{blknum} = \frac{1}{4}$, ResNet50 obtains 0.3192 mAP when $P =$

$16, M = 24$, which is higher about 10% than $P = 8, M = 96$ with 0.2197 mAP. It means that ResNet50 is more fit to large block size $P = 16$, which can keep more local information. On the contrary, the performance of BackBone ViT changes little when $P = 8, M = 96$ and $P = 16, M = 24$, 0.5211 mAP and 0.4813 mAP respectively. The advantages of ViT are on full display in Tab. IV, so our retrieval model uses ViT as BackBone rather than CNN.

IV. Conclusions

In this paper, we propose an end-to-end learning for encrypted image retrieval. Unlike other encrypted image retrieval works which extract features from cipher-images by human, our retrieval model can learn features from cipher-images by itself. We use deep learning to conduct encrypted image retrieval, use ViT as BackBone and combine triplet loss and CE loss when training. Images are encrypted by block permutation and color value replacement for partial blocks. We not only implement the end-to-end learning scheme for encrypted image retrieval, but also the experiments show that the retrieval performance of our scheme is better than that of current schemes. In addition, we compare different BackBones: ResNet50 and ViT, and the experiments show that ViT is more fit to our encrypted image retrieval scheme. In the future, we will promote the encryption performance of cipher-images and seek for unsupervised encrypted image retrieval for end-to-end.

Acknowledgment

This work is supported by the fundamental Research Funds for the Central Universities (No. 21619314), Guang-Dong Basic and Applied Basic Research Foundation (No. 2020A1515110513)

References

[1] Wenjun Lu, Avinash L Varna, Ashwin Swaminathan, and Min Wu. Secure image retrieval through feature protection. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1533–1536. IEEE, 2009.

[2] Zhihua Xia, Xinhui Wang, Liangao Zhang, Zhan Qin, Xingming Sun, and Kui Ren. A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing. *IEEE transactions on information forensics and security*, 11(11):2594–2608, 2016.

[3] Zhihua Xia, Neal N Xiong, Athanasios V Vasilakos, and Xingming Sun. Epcbir: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing. *Information Sciences*, 387:195–204, 2017.

[4] Hang Cheng, Xinpeng Zhang, and Jiang Yu. Ac-coefficient histogram-based retrieval for encrypted jpeg images. *Multimedia Tools and Applications*, 75(21):13791–13803, 2016.

[5] Dandan Liu, Jian Shen, Zhihua Xia, and Xingming Sun. A content-based image retrieval scheme using an encrypted difference histogram in cloud computing. *Information*, 8(3):96, 2017.

[6] Hang Cheng, Xinpeng Zhang, Jiang Yu, and Yuan Zhang. Encrypted jpeg image retrieval using block-wise feature comparison. *Journal of Visual Communication and Image Representation*, 40:111–117, 2016.

[7] Xinpeng Zhang and Hang Cheng. Histogram-based retrieval for encrypted jpeg images. In *2014 IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP)*, pages 446–449. IEEE, 2014.

[8] Peiya Li and Zhenhui Situ. Encrypted jpeg image retrieval using histograms of transformed coefficients. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1140–1144. IEEE, 2019.

[9] Haihua Liang, Xinpeng Zhang, and Hang Cheng. Huffman-code based retrieval for encrypted jpeg images. *Journal of Visual Communication and Image Representation*, 61:149–156, 2019.

[10] Zhihua Xia, Lihua Lu, Tong Qiu, HJ Shim, Xianyi Chen, and Byeungwoo Jeon. A privacy-preserving image retrieval based on ac-coefficients and color histograms in cloud environment. *Computers, Materials & Continua*, 58(1):27–44, 2019.

[11] Yanyan Xu, Jiaying Gong, Lizhi Xiong, Zhengquan Xu, Jinwei Wang, and Yun-qing Shi. A privacy-preserving content-based image retrieval method in cloud environment. *Journal of Visual Communication and Image Representation*, 43:164–172, 2017.

[12] Hang Cheng, Xinpeng Zhang, Jiang Yu, and Fengyong Li. Markov process-based retrieval for encrypted jpeg images. *EURASIP Journal on Information Security*, 2016(1):1, 2016.

[13] Zhihua Xia, Leqi Jiang, Dandan Liu, Lihua Lu, and Byeungwoo Jeon. Boew: A content-based image retrieval scheme using bag-of-encrypted-words in cloud computing. *IEEE Computer Architecture Letters*, (01):1–1, 2019.

[14] Hua Wang, Zhihua Xia, Jianwei Fei, and Fengjun Xiao. An aes-based secure image retrieval scheme using random mapping and bow in cloud computing. *IEEE Access*, 8:61138–61147, 2020.

[15] Zhihua Xia, Lan Wang, Jian Tang, Neal N Xiong, and Jian Weng. A privacy-preserving image retrieval scheme using secure local binary pattern in cloud computing. *IEEE Transactions on Network Science and Engineering*, 8(1):318–330, 2020.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[17] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Sir Ronald Aylmer Fisher and Frank Yates. *Statistical Tables for Biological, Agricultural and Medical Research... Revised and Enlarged*. London, Edinburgh, 1943.

[20] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[23] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[24] Barry L Kalman and Stan C Kwasny. Why tanh: choosing a sigmoidal function. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 4, pages 578–581. IEEE, 1992.

[25] James Ze Wang, Jia Li, and Gio Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on pattern analysis and machine intelligence*, 23(9):947–963, 2001.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[28] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.