Robust Image Hashing Based on Contrastive Masked Autoencoder with Weak-Strong Augmentation Alignment

Cundian Yang[1], Guibo Luo[1], Yuesheng Zhu[1 *], Jiaqi Li[2], Xiyao Liu[2 *]

[1] Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology, Shenzhen Graduate School, Peking University

[2] School of Computer Science and Engineering, Central South University

yangcundian@stu.pku.edu.cn, {luogb, zhuys}@pku.edu.cn, {jiaqi-li, lxyzoewx}@csu.edu.cn

Abstract

Recently, numerous robust image hashing schemes have been developed for content identification. However, many of these schemes face the challenges of maintaining discrimination while simultaneously resisting large-scale attacks. In this paper, we propose a robust image hashing scheme based on **C**ontrastive **M**asked **A**utoencoder with weak-strong augmentation **A**lignment (CMAA). Leveraging contrastive learning, CMAA is designed to learn features that are robust to large-scale and hybrid attacks while maintaining the discrimination of those features. Specifically, it utilizes distribution divergence to align weak attack augmented features with strong attack augmented features, namely weak-strong augmentation alignment, to enhance the robustness to strong attacks. In addition, a masked vision transformer is incorporated to further enhance content identification performance. CMAA also includes a parameter-free quantization layer to mitigate the loss induced by binarization. Experimental results demonstrate that our method exhibits remarkable robustness against various attacks, including challenging ones such as rotation and hybrid attacks, and delivers excellent identification performance with a F_1 score close to 1.0. Our code and supplementary materials are available on Github.

Github — https://github.com/pikeyang/cmaa

Introduction

With the development of the Internet, users can share their image creations online, including photographs and artwork, etc. These contents are the ingenuity of creators, and some of them even have commercial value. However, it is inevitable that individuals engage in illegal copying and transmission of images by tampering or attacks on the images. Therefore, there is a need for content identification systems that facilitate automatic media identification, which are capable of detecting different digital images and associating the attacked images with their respective original counterparts.

Currently, prevalent solutions for content identification include watermarking (Wan et al. 2022) and robust image hashing (Liang et al. 2023). However, embedded watermarking will compromise the integrity and quality of images, making it unsuitable for distortion-free scenarios, such as medical images, etc. Consequently, there is a growing interest among researchers in robust image hashing schemes that leverage either traditional hand-crafted or deep learning feature extraction methods. These schemes extract robustness features from images and map them to fixed-length hash sequences without altering the image content, thereby attracting an increasing number of researchers.

The robust image hashing algorithm typically comprises pre-processing, feature extractor, and hash generator, the most crucial component of which is feature extractor. In traditional robust image hashing methods, researchers use spatial domain-based methods (Shen and Zhao 2020) and transformation domain-based (Liang et al. 2023) methods to design feature extractors to address different types of attacks. However, feature extractors are often designed for specific attacks. Consequently, the emergence of new attacks necessitates the development of novel feature extraction methods.

To alleviate this issue, robust image hashing schemes based on deep learning (Qin et al. 2020; Fang et al. 2023) have been proposed. These methods resist multiple types of attacks simultaneously by utilizing data augmentation techniques. However, they require a well-designed loss function to trade off robustness and discrimination, which often makes it difficult to balance the performance of both. To overcome this challenge, (Fonseca-Bustos, Ramírez-Gutiérrez, and Feregrino-Uribe 2022) introduces contrastive learning to better balance discrimination and robustness. They regard an image and the new views formed after the attack as positive samples and different images as negative samples. With the contrastive learning loss function, they minimize the distance between positive samples to ensure robustness against various types of attacks, and maximize the distance between negative samples to ensure discrimination, achieving effective content identification performance.

However, existing robust image hashing methods based on contrastive learning still have following challenges: (1) They use single attacks and did not consider that more complex attacks such as hybrid attacks may be encountered in real world. In addition, (Wang and Qi 2022) has shown that directly incorporating complex attacks during training could lead to degradation of model performance, as the distortion induced by complex attacks may severely damage the image structure, resulting in misclassification of the augmented

*Corresponding Authors.

image as negative samples. (2) Most of them still use convolutional neural network (CNN) as the backbone, while Vision Transformer (ViT) and masked ViT have greatly advanced the progress of many tasks in the field of computer vision. (3) Existing learning-based methods typically train their models in a continuous feature space and directly perform binary quantization on the continuous features when generating hash sequences, which lacks consideration for the information loss caused by the quantization process.

To tackle the above issues, we propose a robust image hashing method based on an end-to-end contrastive masked autoencoder with weak-strong augmentation alignment (WSA). Specifically, we utilize a contrastive loss to pull the views of same image augmented by different attacks to guarantee robustness, while simultaneously pushing different images to ensure discrimination. To achieve better robustness to strong attacks, we further design WSA to enhance the robustness against strong attacks. Here, we define a single attack as weak attack (WA) data augmentation, while the accumulation of multiple single attacks is defined as strong attack (SA) data augmentation. We align the features of the two views by minimizing the distribution divergence of WA and SA augmented views relative to the negative samples. Therefore, SA augmented view could inherit the discrimination of WA augmented view relative to negative samples, while WA augmented view inherit the robustness from SA view against large-scale and hybrid attacks. In addition, inspired by (He et al. 2022; Mishra et al. 2022; Huang et al. 2023), we incorporate a masked ViT to further improve the performance of the CMAA. Furthermore, we utilize a parameter-free quantization layer to achieve end-to-end learning of binary hash sequence, thereby mitigating the loss of information during the quantization process.

Our contribution in this work are as follows:

- We propose a robust image hashing framework for content identification based on an end-to-end contrastive masked autoencoder.
- We design a weak-strong augmentation alignment to enhance the ability of the framework against strong attack by minimizing the distribution divergence of weak attack and strong attack augmented views relative to negative samples.
- We explore the compensatory effects of ViT backbone and the mask mechanism to further improve the performance of contrastive learning for robust hash extraction.

Related Work

Traditional robust image hashing

Traditional image robust hashing schemes are based on manually designed features to extract robust characteristics from images. They can be divided into two categories: spatial domain based schemes and transform domain based schemes. The former computes statistical characteristics of images to accomplish content identification (Tang et al. 2015; Shen and Zhao 2020; Roy, Thounaojam, and Pal 2022). Transform domain based schemes generate image hash sequence through orthogonal transformations (Abdullahi, Wang, and Li 2020; Liang et al. 2023; Tang et al. 2018), and then extract image features in the transform domain. As the attack manipulations become increasingly complex, the limitations of traditional approaches in terms of poor robustness are becoming more evident (Fang et al. 2023).

Learning-based robust image hashing

The key objective of learning-based robust image hashing schemes lies in how to train neural network so that it has the ability to trade off the discrimination and robustness of robust image hashing. To achieve this goal, (Fang et al. 2023) incorporate multi-layer constraints strategy in training feature extractors. Meanwhile, (Liu et al. 2023) present a Swin-Transfomer-based robust image hashing scheme along with a distortion simulator in the training process. To better trade off the discrimination and robustness of robust image hashing, (Fonseca-Bustos, Ramírez-Gutiérrez, and Feregrino-Uribe 2022) incorporates contrastive learning and carefully designed data augmentation mechanism to minimize the distance between an image and its corresponding attacked image, while simultaneously maximizing the distance between different images.

Contrastive learning

The contrastive learning loss function can simultaneously minimize the distance between positive samples while maximizing the distance among negative samples. This aligns perfectly with the requirements of robust hashing schemes that need to strike a balance between robustness and discrimination. Extensive research (Chen et al. 2020a,b; Tian et al. 2020) demonstrates that well-designed data augmentation mechanisms improve the performance of contrastive learning. Subsequent studies (Zhang and Ma 2022; Deng et al. 2023) confirm that employing strong augmentations further improves the performance of the model. This holds promise for enhancing the robustness of the contrastive learning-based robust image hashing method against strong attacks. In addition, recent studies (Chen, Xie, and He 2021; Mishra et al. 2022; Huang et al. 2023) have revealed that the combination of ViT and Masked ViT with contrastive learning to further improve the performance of contrastive learning.

Proposed Method

Overview

As shown in Figure 1, we utilize contrastive learning to trade off the robustness and discrimination and propose weak-strong augmentation alignment to align the features of the two views by minimizing the distribution divergence of weak attack and strong attack augmented views relative to the negative samples. Furthermore, the mask mechanism and the end-to-end training that takes into account the information loss caused by quantization further improve the performance of CMAA.

Weak and strong data augmentation

We define attacks in Table 1 as weak attack. And the accumulation of multiple weak attacks is referred to as strong attack. For example, given a weak attack augmented image,

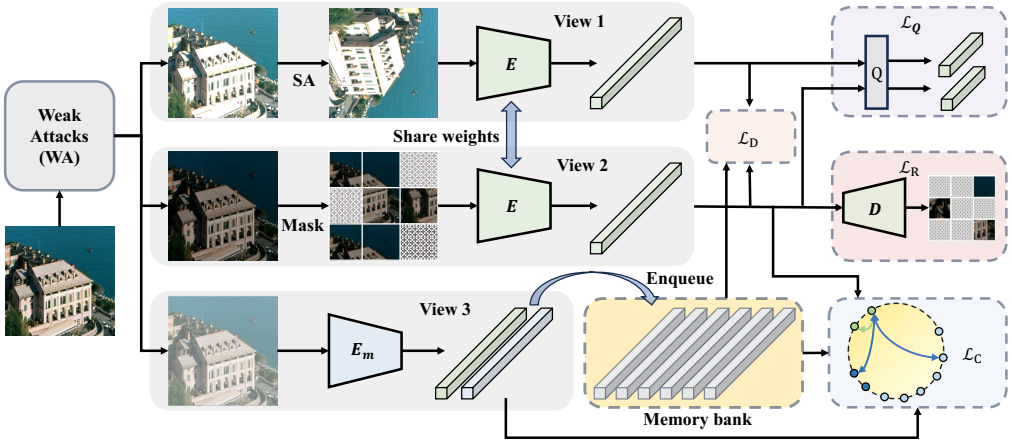

Figure 1: The CMAA framework: Three views of an image are generated using weak attack (WA) data augmentation. Subsequently, we apply strong attack (SA) for view 1 and mask 30% of the patches in view 2, while view 3 remains unchanged. The encoders of view 1 and view 2 share weights, denoted as E, while the encoder weights in view 3 are updated using exponential moving average, denoted as E_m. Here the memory bank consists of K stored features of previous batches. Additionally, Q represents for the quantization layer and D represents for the decoder for reconstructing the masked patches.

Attack	Parameters
Brightness adjustment (BA)	0.7, 0.8, 0.9, 1.1, 1.2, 1.3
Contrast adjustment (CA)	0.7, 0.8, 0.9, 1.1, 1.2, 1.3
Saturation adjustment (ST)	0.7, 0.8, 0.9, 1.1, 1.2, 1.3
Sharpness adjustment (SH)	0.7, 0.8, 0.9, 1.1, 1.2, 1.3
Average filter (AF)	3×3, 5×5, 7×7, 9×9
Median filter (MF)	3×3, 5×5, 7×7, 9×9
Gaussian filter (GF)	3×3, 5×5, 7×7, 9×9
Gaussian noise (GN)	0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2
Speckle noise (SN)	0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2
Salt & Pepper noise (SPN)	0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2
JPEG compression (JC)	10, 20, 30, 40, 50, 60, 70, 80, 90
Posterization (PT)	4, 5, 6, 7
Resizing (RS)	0.5, 0.75, 0.9, 1.1, 1.5, 2.0
Rotation (RT)	-90:-10:10, 10:90:10
Flip (FL)	Horizontal, Vertical
ShearX/Y (SHR)	-0.3, -0.2, -0.1, 0.1, 0.2, 0.3
TranslateX/Y (TR)	-0.3, -0.2, -0.1, 0.1, 0.2, 0.3

Table 1: Weak attack types and parameters. The accumulation of multiple weak attacks is referred to as strong attack.

we randomly select an attack from Table 1 and apply it to the image with 50% probability. The process is repeated five times to obtain the strong attack augmented view of images.

Contrastive masked autoencoder

For each image $\mathbf{x}_i$ in a batch of N images $\{\mathbf{x}\}_{i=1}^{n}$, we generate three views $\mathbf{x}_i^1, \mathbf{x}_i^2, \mathbf{x}_i^3 \in \mathbb{R}^{h\times w\times 3}$ of each image using the WA data augmentation mentioned. Then, we further augment $\mathbf{x}_i^1$ using the SA data augmentation mentioned, resulting in $\mathbf{x}_i^{1,strong}$. Each view of the image is split into $T = (h/p) \times (w/p)$ patches of size $p \times p$: $\mathbf{x}_{i,patch}^{1,strong}, \mathbf{x}_{i,patch}^2, \mathbf{x}_{i,patch}^3$. To further improve the performance of CMAA, we randomly mask some patches of $x_{i,patch}^2$ with a default masking rate of $m = 30\%$, resulting in $T^{'} = (1-m) \times T$ patches, denoted as $\mathbf{x}_{i,patch,um}^2$. The masks $M_i \in \{0,1\}^T$ are generated independently, with a 1 at coordinate $t \in \{1, \dots T\}$ indicating that the t th patch is masked. Each patch is left unmasked independently with probability m, conditioned on always having exactly $T' = m \times T$ patches unmasked. Following (He et al. 2022), only the T' *unmasked* patches are passed to the ViT encoder. It should be noted that $\mathbf{x}_{i,patch}^{1,strong}$ and $\mathbf{x}_{i,patch,um}^2$ are fed into the encoder, while $\mathbf{x}_{i,patch}^3$ is passed into the momentum encoder. Finally, we collect the feature embeddings of the three views $\mathbf{z}_i^1, \mathbf{z}_i^3 \in \mathbb{R}^{T\times d}$ and $\mathbf{z}_i^2 \in \mathbb{R}^{T'\times d}$. Among them, $\mathbf{z}_i^2$ is passed through a comparatively lightweight ViT decoder to produce outputs $\hat{\mathbf{x}}_i^2$ in image space $\mathbb{R}^{h\times w\times 3}$.

Training objective

To trade off discrimination and robustness, we use the features of view 2 ($\mathbf{z}_i^2$) and view 3 ($\mathbf{z}_i^3$) to calculate the contrastive loss ($\mathcal{L}_C$). Specifically, contrastive loss minimizes distances between attack augmented views of the same image and maximizes distances between different images. Furthermore, a distribution divergence loss ($\mathcal{L}_D$) is used to align the features of weak attack augmented view (view 1) and strong attack augmented (view 2), namely weak-strong augmentation alignment (WSA), thereby enhancing the robustness to strong attack. A lightweight decoder is trained to reconstruct the masked patches of view 2 with patch reconstruction loss ($\mathcal{L}_R$), which further improves the performance of the contrastive learning framework with ViT as the backbone. Moreover, CMAA also incorporates a parameter-free quantization layer that reduces the information loss from binarization by calculating the quantization loss ($\mathcal{L}_Q$).

Contrastive loss. The feature embedding $\mathbf{z}_i^1, \mathbf{z}_i^3 \in \mathbb{R}^{T\times d}$ and $\mathbf{z}_i^2 \in \mathbb{R}^{T'\times d}$ returned by the encoder are pooled

via a simple mean along the $T/T^{'}$ dimension to form d-dimensional feature embedding, which are passed through a lightweight MLP projection head that maps into a lower dimension space $\mathbb{R}^l$, $l < d$, and normalized to unit length to produce feature embedding $\mathbf{u}_i^1, \mathbf{u}_i^2, \mathbf{u}_i^3 \in \mathbb{R}^l$ for $i = 1, \ldots, n$. It is worth mentioning that we only calculate contrastive loss between $\mathbf{u}_i^2$ and $\mathbf{u}_i^3$ in Eq. (1).

$$\mathcal{L}_C = \mathbb{E}_{i \in N}\left[-\log \frac{e^{(sim(\mathbf{u}_i^2, \mathbf{u}_i^3)/t)}}{e^{(sim(\mathbf{u}_i^2, \mathbf{u}_i^3)/t)} + \sum_{k=1}^{K} e^{(sim(\mathbf{u}_i^2, \mathbf{u}_k^3)/t)}}\right] \tag{1}$$

with

$$\begin{cases} sim(\mathbf{u}_i^2, \mathbf{u}_i^3) = \frac{\mathbf{u}_i^{2^T}\mathbf{u}_i^3}{||\mathbf{u}_i^2|| \cdot ||\mathbf{u}_i^3||} \\ sim(\mathbf{u}_i^2, \mathbf{u}_k^3) = \frac{\mathbf{u}_i^{2^T}\mathbf{u}_k^3}{||\mathbf{u}_i^2|| \cdot ||\mathbf{u}_k^3||} \end{cases} \tag{2}$$

where $e^{(sim(\mathbf{u}_i^2, \mathbf{u}_i^3)/t)}$ is the exponent of the temperature-smoothed cosine similarity of positive pairs between $\mathbf{u}_i^2$ and $\mathbf{u}_i^3$, while $e^{(sim(\mathbf{u}_i^2, \mathbf{u}_k^3)/t)}$ is the exponent of the temperature-smoothed cosine similarity of negative pairs between $\mathbf{u}_i^2$ and $\mathbf{u}_k^3$, $t > 0$ is a temperature parameter set to $t = 0.1$ by default. Here K is the size of the Memory Bank (first in first out queue) that stores the feature embedding of other images from previous batches, where K is set to 2048. Minimizing $\mathcal{L}_C$ means minimizing $e^{(sim(\mathbf{u}_i^2, \mathbf{u}_i^3)/t)}$ and maximizing $e^{(sim(\mathbf{u}_i^2, \mathbf{u}_k^3)/t)}$. Minimizing $e^{(sim(\mathbf{u}_i^2, \mathbf{u}_i^3)/t)}$ will ensure the robustness of the features, and maximizing $e^{(sim(\mathbf{u}_i^2, \mathbf{u}_k^3)/t)}$ will ensure the discrimination of the features. We will discuss the effect of t and K on CMAA in supplementary materials.

Distribution divergence loss. As shown in Eq. (3), we employ KL-divergence to optimize the distributional divergence of weak attack and strong attack augmented views relative to negative samples, such that the $\mathbf{u}_i^1$ of SA augmented view will inherit the $\mathbf{u}_i^2$ of the WA augmented view regarding not only its similarity of the corresponding positive target u_i^3, but also its discrimination with the negative samples $\mathbf{u}_k^3$ in the representation bank.

$$\begin{aligned} \mathcal{L}_D = \mathbb{E}_{i \in N}[&-p(\mathbf{u}_i^3|\mathbf{u}_i^2) \log p(\mathbf{u}_i^3|\mathbf{u}_i^1) \\ &- \sum_{k=1}^{K} p(\mathbf{u}_k^3|\mathbf{u}_i^2) \log p(\mathbf{u}_k^3|\mathbf{u}_i^1)] \end{aligned} \tag{3}$$

where $p(\mathbf{u}_k^3|\mathbf{u}_i^2)$ is an better mimic of $p(\mathbf{u}_k^3|\mathbf{u}_i^1)$, which aligns the features of WA augmented view and SA augmented view.

Patch reconstruction loss. The outputs $\hat{\mathbf{x}}_i^2$, $i = 1, \ldots, n$ of the ViT decoder are trained to reconstruct the missing patches of each image. Following (He et al. 2022), we only compute the reconstruction loss on masked patches:

$$\mathcal{L}_{\mathrm{R}} = \mathbb{E}_{i \in N}\left[\sum_{i=1}^{n} \|M_i \circ (\mathbf{x}_i - \hat{\mathbf{x}}_i)\|_2^2\right] \tag{4}$$

where $\circ$ multiplies all pixels in the t-th patch of the residual image $(\mathbf{x}_i - \hat{\mathbf{x}}_i)$ by the t-th element of $M_i \in \{0, 1\}^T$.

Attacks	Parameters
RT + JC (H1)	(30, 70), (50, 70), (50, 50)
RS + BA (H2)	(0.5, 0.8), (2.0, 0.8), (0.5, 1.2), (2.0, 1.2)
TR + GN (H3)	(0.1(X/Y), 0.05), (0.2(X/Y), 0.05), (0.1(X/Y), 0.1)
FL + SH (H4)	(Horizontal, 0.8),(Horizontal, 1.2), (Vertical, 0.8), (Vertical, 1.2)
GF + PT + CA (H5)	(7×7, 4, 1.2), (5×5, 8, 1.2), (9×9, 8, 1.1),
ST + RT + SHR (H6)	(0.7, 50, 0.2(X/Y)), (0.8, 50, 0.3(X/Y)), (0.9, 70, 0.1(X/Y))
SH + TR + SN (H7)	(1.2, 0.1(X/Y), 0.05), (1.1, 0.2(X/Y), 0.05), (1.2, 0.1(X/Y), 0.1)
RS + SPN + AF (H8)	(0.5, 0.05, 7×7), (0.5, 0.05, 9×9) (2.0, 0.05, 7×7), (2.0, 0.05, 9×9)

Table 2: Hybrid attacks used to evaluate the performance of each scheme.

Quantization loss. We adopt the binary layer proposed by (Li and van Gemert 2021) to train an end-to-end autoencoder, aiming to minimize the information loss during the quantization process. During training, we maximize the similarity between $\mathbf{u}_i^1/\mathbf{u}_i^2$ and the binarized features processed by the binary layer, denoted as $\mathbf{f}_i^1/\mathbf{f}_i^2$, ensuring minimal information loss in the quantization process. It is important to note that, since $\mathbf{u}_i^3$ is the output of the momentum encoder, which does not directly undergo gradient back-propagation, we only calculate the information loss of $\mathbf{u}_i^1$ and $\mathbf{u}_i^2$ after passing through the binary layer as shown in Eq. (6).

$$\mathcal{L}_{\mathrm{Q}} = \mathbb{E}_{i \in N}\left[\frac{\mathbf{u}_i^{1^T}\mathbf{f}_i^1}{||\mathbf{u}_i^1|| \cdot ||\mathbf{f}_i^1||} + \frac{\mathbf{u}_i^{2^T}\mathbf{f}_i^2}{||\mathbf{u}_i^2|| \cdot ||\mathbf{f}_i^2||}\right] \tag{5}$$

The process of binarization can be represented by Eq. (6)

$$\mathbf{f}_{i,j}^v = \begin{cases} 1 & \text{if } \mathbf{u}_{i,j}^v > median(\mathbf{u}_{i,1}^v \ldots, \mathbf{u}_{i,l}^v) \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

where $median$ calculates the middle value of $\mathbf{u}_{i,j}$, $i = 1, \ldots N$ represents for the each sample of N images, and $j = 1, \ldots l$ represents for the each dimension of $\mathbf{u}_i^v/\mathbf{f}_i^v$, and $v \in \{1, 2\}$.

The discrete binary codes have no continuous derivatives and cannot be directly optimized by gradient descent, but we can follow (Bengio, Léonard, and Courville 2013; Li and van Gemert 2021) to use a proxy derivative approximated by straight through estimator(STE) to avoid the vanishing gradients.

The joint objective function. The overall CMAA objective trains the autoencoder to optimize four losses combined:

$$\mathcal{L} = \mathcal{L}_{\mathrm{C}} + \mathcal{L}_{\mathrm{D}} + \mathcal{L}_{\mathrm{R}} + \mathcal{L}_{\mathrm{Q}} \tag{7}$$

where $\mathcal{L}_{\mathrm{C}}$ is the contrastive learning loss from Eq. (1), $\mathcal{L}_{\mathrm{D}}$ is the KL-divergence loss from Eq. (3), $\mathcal{L}_{\mathrm{R}}$ is the patch reconstruction loss from Eq. (4) and $\mathcal{L}_{\mathrm{Q}}$ is the quantization loss from Eq. (5). Although fine-tuning the weights of each loss can lead to better results, we observe that assigning equal weights to each loss has achieved remarkable content identification performance. Meanwhile, the fine-tuning process also requires more computing resources.

	ProtoHash	TMICH	Swin-T	GLIF	ISM	Ours
$\tau \uparrow$	0.1172	0.2813	**0.3520**	0.1875	0.2188	0.3203
$\mu \uparrow$	0.4744	0.4632	0.4577	0.4106	**0.4999**	0.4947
$\sigma \downarrow$	0.1751	0.0771	**0.0350**	0.1070	0.1296	0.0566
$\tau \uparrow$	0.1367	0.2813	**0.3620**	0.1625	0.1563	0.3203
$\mu \uparrow$	0.4445	0.4761	0.4691	0.3680	**0.4938**	0.4916
$\sigma \downarrow$	0.1601	0.0746	**0.0351**	0.0920	0.1447	0.0570

Table 3: Discrimination metrics in terms of NHD_d, where τ represents the threshold of NHD_d given false positive rate to 1%, μ represents the average values of NHD_d, and σ represents the standard deviation values of NHD_d.The upper half of the table presents results on the CASIA dataset, while the lower half presents results on the Copydays dataset.

Hashing-based content identification system

Once the encoder is trained, we utilize the encoder to extract features of the N_t test images and binarize the features, resulting in a hash sequence, denoted as $\mathbf{f} \in \mathbb{R}^{N_t \times l}$, so as to create a database in the hash registration stage. Note that we did not perform any fine-tuning or re-training on any test datasets. In the hash identification stage, the hashing-based content identification system uses this database to associate the query images with known identities, employing a similarity metric with a given threshold (τ).

The Hamming distance (HD) is a well-known similarity metric for comparing two hash sequence, which measures the number of positions at which the corresponding values are different between two hash sequence as Eq. (8).

$$\text{HD}(\mathbf{f}_i, \mathbf{f}_k) = \sum_{j=1}^{l} \mathbf{f}_{i,j} \oplus \mathbf{f}_{k,j} \tag{8}$$

where, $j = 1, \dots l$ represents for the each dimension of $\mathbf{f}$, $i, k = 1, \dots N_t$ represent for the each sample from N_t test images.

Due to the varying hash lengths of different robust image hashing schemes, Normalized Hamming Distance (NHD) provides a more intuitive and convenient measure of similarity for hash sequences. NHD represents the probability of a bit difference between the two hash values, denoted as $\text{NHD}(\mathbf{f}_i, \mathbf{f}_k) = \text{HD}(\mathbf{f}_i, \mathbf{f}_k)/l$.

Experiments

Experimental setup

We train our autoencoder using the COCO mini-train dataset proposed by (Samet, Hicsonmez, and Akbas 2020), which is a subset of the COCO train2017 dataset. This subset comprises 25,000 images, accounting for approximately 20% of the train2017 set. Similarly to previous work (Chen, Xie, and He 2021), we use ViT-Small (Dosovitskiy et al. 2020) as our encoder backbone E and our momentum encoder backbone E_m. Following (He et al. 2020), the momentum encoder (E_m) is updated from encoder (E) using exponential moving average (EMA) with a momentum smoothing factor $\alpha = 0.999$, denoted as $E_m = \alpha * E_m + (1 - \alpha) * E$.

Attack	ProtoHash	TMICH	Swin-T	GLIF	ISM	Ours
BA	0.1635	0.0567	0.0975	0.2323	0.0681	0.1281
CA	0.0886	0.0568	0.0990	0.2363	0.1373	0.1158
ST	0.0604	0.0477	0.0811	0.2313	0.0877	0.0662
SH	0.0061	0.0432	0.0760	0.2281	0.0642	0.0087
AF	0.0397	0.2016	0.2456	0.2361	0.1305	0.0717
MF	0.0334	0.1943	0.2335	0.2396	0.1084	0.0519
GF	0.0582	0.2433	0.2823	0.2386	0.1534	0.1015
GN	0.0747	0.3125	0.3074	0.2740	0.2892	0.1450
SN	0.0444	0.2039	0.2230	0.2480	0.1334	0.0746
SPN	0.0521	0.2516	0.2637	0.2595	0.2163	0.0920
JC	0.0122	0.0904	0.1280	0.2308	0.0936	0.0204
PT	0.0224	0.0507	0.0818	0.2287	0.0773	0.0248
RS	0.0063	0.0785	0.1178	0.2293	0.0685	0.0094
RT	0.1820	0.3091	0.3021	0.3770	0.4221	0.1248
FL	0.2022	0.1649	0.1765	0.3020	0.3741	0.0270
SHR	0.1228	0.1563	0.1670	0.3376	0.3413	0.0996
TR	0.1844	0.1603	0.1916	0.3295	0.4060	0.1316
H1	0.1823	0.3289	0.3159	0.3875	0.4357	0.1181
H2	0.1551	0.1130	0.1538	0.2322	0.0869	0.1190
H3	0.1591	0.3457	0.3525	0.3320	0.4134	0.2092
H4	0.2020	0.1626	0.1760	0.3021	0.3755	0.0279
H5	0.0852	0.3034	0.3300	0.2437	0.1948	0.1333
H6	0.2277	0.3647	0.3554	0.3933	0.4616	0.1617
H7	0.1451	0.2484	0.2646	0.3260	0.3749	0.1338
H8	0.0758	0.2954	0.3143	0.2475	0.2010	0.1378
Avg.	0.1104	0.1926	0.2124	0.2893	0.2528	**0.0987**

Table 4: Robustness metric in terms of NHD_r ($\downarrow$) for CASIA dataset.

In the experiments, we use three datasets, i.e., CASIA dataset (Dong, Wang, and Tan 2013), Copydays dataset (Jégou et al. 2011), and UCID dataset (Schaefer and Stich 2003) to evaluate the performance of different methods. These datasets are widely used in robust image hashing. Due to page limitation, we include the results on the UCID dataset in the supplementary materials.

We compare CMAA with five state-of-the-art robust approaches, namely the ProtoHash (Fonseca-Bustos, Ramírez-Gutiérrez, and Feregrino-Uribe 2022), the TMICH (Fang et al. 2023), Swin Transformer-based scheme (Swin-T) from (Liu et al. 2023), GLIF-based scheme from (Huang and Liu 2021), ISM-based scheme from (Liang et al. 2021), and evaluate all schemes in terms of discrimination, robustness and content identification performance.

Performance metrics

We evaluate CMAA from four perspectives: discrimination, robustness, content identification performance and time complexity (due to page limitation, we include time complexity in the supplementary material). Following ProtoHash, we calculate the NHD to evaluate both discrimination and robustness. For different images, a larger NHD indicates better discrimination of the scheme. Conversely, a smaller NHD between an image and its attacked view represents better robustness of the scheme. For convenience of writing and reading, we denote the NHD that measures discrimination as NHD_d, and the NHD that measures robust-

Attack	ProtoHash	TMICH	Swin-T	GLIF	ISM	Ours
BA	0.1838	0.0523	0.0955	0.2226	0.0416	0.1425
CA	0.0932	0.0494	0.0990	0.2252	0.1005	0.1285
ST	0.0577	0.0364	0.0638	0.2196	0.0877	0.0618
SH	0.0038	0.0214	0.0488	0.2168	0.0642	0.0040
AF	0.0022	0.0420	0.0832	0.2222	0.0297	0.0061
MF	0.0079	0.0455	0.0842	0.2224	0.0262	0.0101
GF	0.0046	0.0604	0.1019	0.2233	0.0379	0.0098
GN	0.0831	0.1454	0.1912	0.2226	0.1429	0.1244
SN	0.0523	0.1000	0.1369	0.2244	0.0646	0.0946
SPN	0.0571	0.1147	0.1600	0.2214	0.1032	0.0913
JC	0.0051	0.0596	0.1008	0.2187	0.0449	0.0150
PT	0.0238	0.0359	0.0742	0.2179	0.0773	0.0301
RS	0.0013	0.0252	0.0555	0.2182	0.0246	0.0033
RT	0.2003	0.2877	0.2807	0.3540	0.3595	0.1223
FL	0.2386	0.1738	0.1756	0.2770	0.3364	0.0264
SHR	0.1312	0.1518	0.1631	0.3212	0.3413	0.1006
TR	0.1976	0.1631	0.1812	0.3095	0.4060	0.1351
H1	0.1964	0.2925	0.2739	0.3634	0.4357	0.1198
H2	0.1676	0.0528	0.1060	0.2191	0.0869	0.1404
H3	0.1713	0.2158	0.2594	0.2866	0.4134	0.1829
H4	0.2386	0.1725	0.1763	0.2769	0.3755	0.0261
H5	0.0701	0.0981	0.1545	0.2296	0.1948	0.1159
H6	0.2346	0.3320	0.3224	0.3732	0.4616	0.1566
H7	0.1618	0.1767	0.1962	0.3048	0.3749	0.1445
H8	0.0291	0.1249	0.1669	0.2288	0.2010	0.0568
Avg.	0.1141	0.1359	0.1614	0.2689	0.2174	**0.0905**

Table 5: Robustness metric in terms of NHD_r (↓) for Copydays dataset.

Attack	ProtoHash	TMICH	Swin-T	GLIF	ISM	Ours
BA	0.4987	1.0000	1.0000	0.5733	0.9883	0.9947
CA	0.8469	1.0000	1.0000	0.5605	0.9125	0.9990
ST	0.9221	1.0000	1.0000	0.5778	0.6580	1.0000
SH	1.0000	1.0000	1.0000	0.5841	0.9905	1.0000
AF	0.9773	0.9233	0.9625	0.5487	0.9087	1.0000
MF	0.9889	0.9362	0.9853	0.5421	0.9378	1.0000
GF	0.9375	0.8117	0.8312	0.5353	0.8714	0.9968
GN	0.8715	0.5853	0.7604	0.3851	0.4814	0.9696
SN	0.9648	0.8838	0.9754	0.4899	0.9888	1.0000
SPN	0.9435	0.7452	0.9007	0.4444	0.4494	0.9952
JC	0.9979	0.9935	0.9950	0.5813	0.9579	1.0000
PT	0.9923	1.0000	1.0000	0.5841	0.6786	1.0000
RS	1.0000	0.9968	1.0000	0.5756	0.8271	1.0000
RT	0.4415	0.6298	0.8099	0.0369	0.0817	1.0000
FL	0.3990	0.9074	0.9702	0.3837	0.2509	1.0000
SHR	0.7050	0.9942	1.0000	0.1415	0.4672	1.0000
TR	0.4176	0.9926	0.9883	0.2293	0.1202	0.9935
H1	0.4614	0.5524	0.8006	0.0000	0.6766	1.0000
H2	0.5126	0.9872	1.0000	0.5680	0.9707	0.9969
H3	0.5428	0.4984	0.6650	0.1879	0.1392	0.9524
H4	0.3990	0.9203	0.9666	0.3837	0.5272	1.0000
H5	0.8958	0.6351	0.6777	0.5276	0.3329	0.9958
H6	0.2045	0.4407	0.5948	0.0123	0.3179	1.0000
H7	0.5918	0.8629	0.9638	0.2147	0.0902	0.9979
H8	0.8678	0.6093	0.6896	0.4833	0.7186	0.9718
Avg.	0.7053	0.8371	0.9042	0.3579	0.5457	**0.9946**

Table 6: Content identification performance in terms of F_1 score (↑) for CASIA dataset.

ness as NHD_r. Additionally, NHD_d is the average of NHD among different images, NHD_r is the mean value of NHD over all test images under a specific attack. With the aim of evaluating content identification of our method, we consider a binary classification scenario, where we aim to decide if two hash values represent the same identity or not. Therefore, we define a True Positive (TP) as a correct identification and a True Negative (TN) as a correct rejection, given an identification threshold τ. False Positives (FP) and False Negatives (FN) follow the same direction. Then, we use the $F_1 = TP/(TP + \frac{1}{2}(FP + FN))$ to evaluate the content identification performance of the robust image hashing schemes. In this paper, the value of τ is chosen as the False Positive rate set to 1%.

Discrimination evaluation

In terms of discrimination, NHD_d represents for the difference between distinct images, and a larger value of NHD_d indicates better hash discrimination. As presented in Table 3, the discrimination of CMAA is remarkable. The average NHD_d values (μ) are close to 0.5 for both CASIA and Copydays dataset. These results are better than those of ProtoHash, TMICH, Swin-T and GLIF, and are comparable with those of ISM. Our remarkable discrimination is due to our exclusively designed contrastive loss function. Additionally, the threshold (τ) and standard deviation (σ) values of NHD_d also demonstrate the comparative discrimination of our method.

Robustness evaluation

We conducted the attacks in Table 1 and Table 2 on test datasets and compare the robustness of the different schemes in terms of NHD_r, to evaluate the hash robustness. The results for the CASIA dataset are presented in Table 4, while the results for the Copydays dataset can be found in Table 5. Here, NHD_r is calculated as the difference between the hash sequence of the original image and the hash sequence of the original image under various attacks, and a smaller NHD_r indicates better hash robustness.

As we can see from Tables 4 and 5, the average NHD_r results of CMAA are smaller compared to all other schemes for the CASIA and Copydays datasets. This improvement is attributed to our proposed weak-strong augmentation alignment, the masked ViT backbone, the parameter-free quantization layer and the joint loss function together guaranteeing the robustness.

Content identification evaluation

We evaluate the content identification performance in terms of F_1 score given a threshold (τ) that corresponds to a false positive rate of 1%. As can be observed in Table 6 and 7, most of the F_1 score values for CMAA are greater than 0.99, while the average F_1 score is very close to 1 for both the CASIA and Copydays datasets. Specifically, for most type of attacks, we achieve vastly superior results, with other methods failing for attacks such as rotation and hybrid attacks.

Attack	ProtoHash	TMICH	Swin-T	GLIF	ISM	Ours
BA	0.5141	1.0000	1.0000	0.4578	0.9823	0.9928
CA	0.8732	1.0000	1.0000	0.4401	0.8842	0.9968
ST	0.9570	1.0000	1.0000	0.4669	0.9738	1.0000
SH	1.0000	1.0000	1.0000	0.4819	0.9937	1.0000
AF	1.0000	1.0000	1.0000	0.4738	1.0000	1.0000
MF	1.0000	1.0000	1.0000	0.4683	1.0000	1.0000
GF	1.0000	1.0000	1.0000	0.4625	0.9935	1.0000
GN	0.8581	0.9801	0.9890	0.4541	0.7777	0.9952
SN	0.9679	0.9968	1.0000	0.4492	0.9582	0.9952
SPN	0.9315	0.9875	0.9960	0.4588	0.8882	0.9984
JC	1.0000	0.9975	0.9975	0.4723	0.9761	1.0000
PT	0.9961	1.0000	0.9994	0.4742	0.9820	1.0000
RS	1.0000	1.0000	1.0000	0.4843	0.9963	1.0000
RT	0.4366	0.6435	0.8838	0.0438	0.1273	0.9998
FL	0.4141	0.8592	0.9767	0.3118	0.2902	1.0000
SHR	0.7480	0.9960	0.9989	0.0861	0.3211	1.0000
TR	0.4645	0.9873	0.9981	0.1578	0.1411	0.9951
H1	0.4556	0.6480	0.9548	0.0252	0.0408	1.0000
H2	0.5642	1.0000	1.0000	0.4794	0.9706	1.0000
H3	0.5688	0.9244	0.9780	0.2122	0.0861	0.9880
H4	0.4125	0.8631	0.9740	0.3118	0.2417	1.0000
H5	0.9603	0.9957	0.9946	0.4194	0.8024	1.0000
H6	0.2225	0.4538	0.8479	0.0084	0.0443	0.9995
H7	0.6135	0.9810	0.9979	0.1629	0.1333	0.9984
H8	1.0000	0.9827	0.9944	0.4129	0.7711	1.0000
Avg.	0.7259	0.9166	0.9761	0.3045	0.5837	**0.9982**

Table 7: Content identification performance in terms of F_1 score (↑) for Copydays dataset.

To further demonstrate the superiority of CMAA, we generate detection error tradeoff (DET) graphs by calculating false negative rate (P_{fn}) for different values of false positive rate (P_{fp}). As Figure 2 shown, the curves of CMAA are continuously and significantly below those of the other methods for both CASIA and Copydays dataset, confirming the superior overall performance.

Ablation study

To understand the effects of the key components that we adopt in CMAA, we conduct numerous ablation experiments. Unless otherwise defined, all models of ablation experiments are trained for 100 epochs on COCO mini-train dataset with ViT-small as the backbone. In addition, ablation experiments on the hyper-parameters mask ratio(m), temperature(t), size of memory bank(K), and hash length(l) and backbone network are in the supplementary material.

As shown in Table 8, we observe a significant improvement in content identification performance with the utilization of KL-divergence loss ($\mathcal{L}_D$) and patch reconstruction loss ($\mathcal{L}_R$), from the four sets of experiments, namely (a), (b), (c) and CMAA. Furthermore, comparing (d) and CMAA, we can find that the quantization loss ($\mathcal{L}_Q$) contributes to mitigating the distortion to the features during the quantization process. It is worth noting that SA is used as part of data augmentation in the training process even when $\mathcal{L}_D$ is not employed. To further prove the effectiveness of CMAA, we demonstrate the advantages of CMAA in resisting strong attacks in the form of tables in the supplementary material.

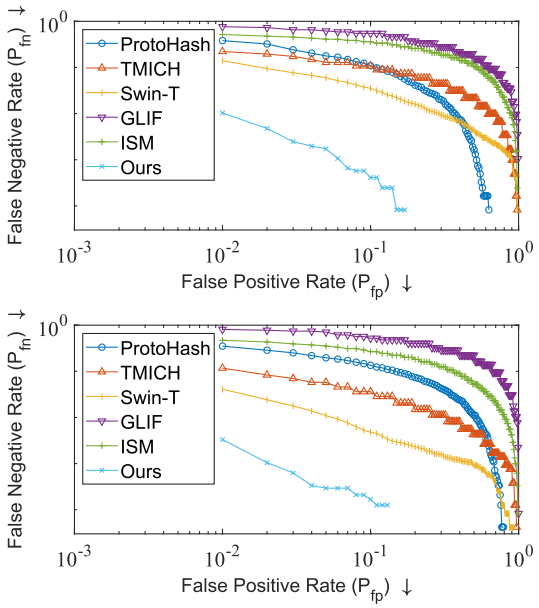

Figure 2: DET graphs on CASIA (top) and Copydays (bottom) dataset for the different schemes.

	(a)	(b)	(c)	(d)	Ours
$\mathcal{L}_C$	✓	✓	✓	✓	✓
$\mathcal{L}_D$	✗	✓	✗	✓	✓
$\mathcal{L}_R$	✗	✗	✓	✓	✓
$\mathcal{L}_Q$	✓	✓	✓	✗	✓
NHD_d ↑	**0.4962**	0.4905	0.4930	0.4917	<u>0.4947</u>
NHD_r ↓	0.1343	0.1060	0.1122	**0.0946**	<u>0.0987</u>
F_1 score ↑	0.9641	0.9906	0.9856	<u>0.9918</u>	**0.9946**

Table 8: Effect of key components on CMAA for CASIA dataset. The underlined <u>results</u> are suboptimal.

Conclusion

In this paper, we propose a robust image hashing scheme based on an end-to-end contrastive masked autoencoder with weak-strong augmentation alignment (WSA). Specifically, the proposed WSA enhances robustness against large-scale and hybrid attacks by minimizing the distribution divergence of weak attack and strong attack augmented views relative to negative samples. In addition, we introduce a masked ViT and a parameter-free quantization layer to further improve the content identification performance of CMAA. Our method improves the robustness of robust image hashing against strong attacks while maintaining a trade-off between robustness and discrimination, achieving superior content identification performance compared to existing robust image hashing methods.

Acknowledgments

This work is supported by Shenzhen Science and Technology Program (JCYJ20230807120800001), 2023 Shenzhen sustainable supporting funds for colleges, universities (20231121165240001), Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology (2024B1212010006), Natural Science Foundation of Hunan Province, China (2022GK5002,2024JK2015, 2024JJ5440), the Special Foundation for Distinguished Young Scientists of Changsha (kq2209003), the National Natural Science Foundation of China (62472446) and the Foundation of State Key Laboratory of High Performance Computing, National University of Defense Technology (202401-13).

References

Abdullahi, S. M.; Wang, H.; and Li, T. 2020. Fractal coding-based robust and alignment-free fingerprint image hashing. *IEEE Transactions on Information Forensics and Security*, 15: 2587–2601.

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020b. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Chen, X.; Xie, S.; and He, K. 2021. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9640–9649.

Deng, X.; Huang, D.; Chen, D.-H.; Wang, C.-D.; and Lai, J.-H. 2023. Strongly augmented contrastive clustering. *Pattern Recognition*, 139: 109470.

Dong, J.; Wang, W.; and Tan, T. 2013. Casia image tampering detection evaluation database. In *2013 IEEE China summit and international conference on signal and information processing*, 422–426. IEEE.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fang, Y.; Zhou, Y.; Li, X.; Kong, P.; and Qin, C. 2023. TM-CIH: Perceptual Robust Image Hashing with Transformer-based Multi-layer Constraints. In *Proceedings of the 2023 ACM Workshop on Information Hiding and Multimedia Security*, 7–12.

Fonseca-Bustos, J.; Ramírez-Gutiérrez, K. A.; and Feregrino-Uribe, C. 2022. Robust image hashing for content identification through contrastive self-supervised learning. *Neural Networks*, 156: 81–94.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

Huang, Z.; Jin, X.; Lu, C.; Hou, Q.; Cheng, M.-M.; Fu, D.; Shen, X.; and Feng, J. 2023. Contrastive masked autoencoders are stronger vision learners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Huang, Z.; and Liu, S. 2021. Perceptual Image Hashing With Texture and Invariant Vector Distance for Copy Detection. *IEEE Transactions on Multimedia*, 23: 1516–1529.

Jégou, H.; Perronnin, F.; Douze, M.; Sánchez, J.; Pérez, P.; and Schmid, C. 2011. Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, 34(9): 1704–1716.

Li, Y.; and van Gemert, J. 2021. Deep unsupervised image hashing by maximizing bit entropy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2002–2010.

Liang, X.; Tang, Z.; Wu, J.; Li, Z.; and Zhang, X. 2021. Robust image hashing with isomap and saliency map for copy detection. *IEEE Transactions on Multimedia*.

Liang, X.; Tang, Z.; Zhang, X.; Yu, M.; and Zhang, X. 2023. Robust hashing with local tangent space alignment for image copy detection. *IEEE Transactions on Dependable and Secure Computing*.

Liu, T.; Yao, H.; Li, X.; and Qin, C. 2023. Robust Perceptual Image Hashing for Screen-Shooting Attack. *IEEE Transactions on Consumer Electronics*.

Mishra, S.; Robinson, J.; Chang, H.; Jacobs, D.; Sarna, A.; Maschinot, A.; and Krishnan, D. 2022. A simple, efficient and scalable contrastive masked autoencoder for learning visual representations. *arXiv preprint arXiv:2210.16870*.

Qin, C.; Liu, E.; Feng, G.; and Zhang, X. 2020. Perceptual image hashing for content authentication based on convolutional neural network with multiple constraints. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(11): 4523–4537.

Roy, M.; Thounaojam, D. M.; and Pal, S. 2022. Perceptual hashing scheme using KAZE feature descriptors for combinatorial manipulations. *Multimedia Tools and Applications*, 81(20): 29045–29073.

Samet, N.; Hicsonmez, S.; and Akbas, E. 2020. Houghnet: Integrating near and long-range evidence for bottom-up object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, 406–423. Springer.

Schaefer, G.; and Stich, M. 2003. UCID: An uncompressed color image database. In *Storage and retrieval methods and applications for multimedia 2004*, volume 5307, 472–480. SPIE.

Shen, Q.; and Zhao, Y. 2020. Perceptual hashing for color image based on color opponent component and quadtree structure. *Signal Processing*, 166: 107244.

Tang, Z.; Huang, Z.; Yao, H.; Zhang, X.; Chen, L.; and Yu, C. 2018. Perceptual Image Hashing with Weighted DWT Features for Reduced-Reference Image Quality Assessment. *The Computer Journal*, 61(11): 1695–1709.

Tang, Z.; Zhang, X.; Li, X.; and Zhang, S. 2015. Robust image hashing with ring partition and invariant vector distance. *IEEE transactions on information forensics and security*, 11(1): 200–214.

Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; and Isola, P. 2020. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33: 6827–6839.

Wan, W.; Wang, J.; Zhang, Y.; Li, J.; Yu, H.; and Sun, J. 2022. A comprehensive survey on robust image watermarking. *Neurocomputing*, 488: 226–247.

Wang, X.; and Qi, G.-J. 2022. Contrastive learning with stronger augmentations. *IEEE transactions on pattern analysis and machine intelligence*, 45(5): 5549–5560.

Zhang, J.; and Ma, K. 2022. Rethinking the augmentation module in contrastive learning: Learning hierarchical augmentation invariance with expanded views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16650–16659.