論文目録

2022 年 1 月 13 日

提出者: 孟 昭雄 （工学研究科工学専攻電気電子情報工学領域 201970175）

主論文

1. **論文題目**: Design proposal of perceptual hashing based on convolutional neural network for digital watermarking（電子透かしのための畳み込みニューラルネットワークに基づく知覚ハッシュの設計提案）
2. **印刷公表の方法及び時期**:
   (1) Zhaoxiong Meng, Tetsuya Morizumi, Sumiko Miyata and Hirotsugu Kinoshita, "Design scheme of copyright management system based on digital watermarking and blockchain," 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), vol. 02, pp.359–364, 23–27, July 2018.
   (2) Zhaoxiong Meng, Tetsuya Morizumi, Sumiko Miyata and Hirotsugu Kinoshita, "Perceptual hashing based on machine learning for blockchain and digital watermarking," 2019 Third World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), pp. 193–198, 30–31, July 2019.
   (3) Zhaoxiong Meng, Tetsuya Morizumi, Sumiko Miyata and Hirotsugu Kinoshita, "A Scheme of Digital Copyright Management System Based on Blockchain and Digital Watermarking--Research on Improvement Method of Perceptual Hashing based on Machine Learning," IEICE Technical Report; IEICE Tech. Rep., 119(329), 21–27.
   (4) Zhaoxiong Meng, Tetsuya Morizumi, Sumiko Miyata and Hirotsugu Kinoshita, "An Improved Design Scheme for Perceptual Hashing based on CNN for Digital Watermarking," 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), pp.1789–1794, 13–17, July 2020.
   (5) Zhaoxiong Meng, Tetsuya Morizumi, Sumiko Miyata and Hirotsugu Kinoshita, "Design Scheme of Perceptual Hashing based on Output of CNN for Digital Watermarking," 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), pp.1545–1550, 12–16, July 2021.
   (6) Zhaoxiong Meng, Tetsuya Morizumi, Sumiko Miyata, Hirotsugu Kinoshita, "Design scheme of perceptual hashing for image groups based on CNN for security systems," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences.（投稿中）
   (7) Zhaoxiong Meng, Tetsuya Morizumi, Sumiko Miyata, Hirotsugu Kinoshita, "A design scheme for digital rights management system using CNN-based perceptual hashing," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences.（投稿準備中）
3. **部数**: 7 部

参考論文

なし

和文要旨

2022 年 1 月 13 日

提出者: 孟　昭雄（工学研究科工学専攻電気電子情報工学領域 201970175）

論文題目: Design proposal of perceptual hashing based on convolutional neural network for digital watermarking

要旨:

電子透かし技術は、ディジタル著作権保護に優れた応用を持っている。しかし、電子透かし技術に関しては、まだ解決されていない問題点がいくつかある。1番目の問題点は、作者の画像の透かし情報が二次利用者の画像に流用されるのを防ぐために、画像の特徴情報を使用して透かし情報を生成する必要があるということである。 2番目の問題点は、作成者が加工・編集した画像と二次利用者が加工・編集した画像の著作権がすべて原画像の作成者であることを証明するために、透かし情報が加工・編集された画像と元の画像の間同一であることを証明できる必要があることである。3番目の問題点は、作成者と二次利用者が画像を加工・編集した順序を証明するために、信頼できる第三者に依存せずに多重電子透かしの埋め込み順序を証明する必要があることである。

電子透かし、ブロックチェーン、知覚ハッシュに基づくディジタル著作権管理システムを提案した。上記の1番目および2番目の問題点に対処するために、このシステムは知覚ハッシュを使用して透かし情報を生成した。知覚ハッシュは、視覚的に同じ画像に対して同じ知覚ハッシュ値を生成できるため、画像を加工・編集してもハッシュ値は変化しない。画像が加工・編集されるたびに、電子透かしが埋め込まれ、多重電子透かしが形成される。したがって、この透かし情報は、加工・編集された画像と原画像の同一性を証明できる。3番目の問題点については、透かし情報をブロックチェーンと組み合わせることによって解決した。ブロックチェーンのセキュリティを利用して、知覚ハッシュに基づいて生成された透かし情報を保存および管理し、ブロックチェーンのタイミングを使用して透かし情報の保存順序を証明し、それによって多重電子透かしの埋め込み順序と画像の加工・編集の順序を証明した。

ディジタル著作権管理システムでは、従来の知覚ハッシュを使用した。従来の知覚ハッシュは主に画像検索に使用され、ディジタル著作権管理システムでは、知覚ハッシュが電子透かしに使用される。従来の知覚ハッシュをテストしたところ、回転や反転などの画像処理方法では、従来の知覚ハッシュは原画像と加工・編集された画像に異なるハッシュ値を計算した。このように、原画像と加工・編集された画像の同一性を検証することはできない。

畳み込みニューラルネットワーク（CNN）に基づく知覚ハッシュスキームを提案した。CNN は、マシンビジョン、画像分類などの分野で広く使用されている。CNN の主な特徴は、画像の特徴を保持および抽出するために、中間層で畳み込み層とプーリング層を使用することである。画像を回転または反転すると、加工・編集された画像と原画像が同一であることを認識できる。具体的な実装方法は、既存の CNN モデルで自分のトレーニングデータを使用して fine-tuning を実行し、学習済み CNN を取得することである。CNN モデルのトレーニングデータは、対象画像と他の無関係な画像の 2 つのクラスの画像で構成される。出力は、対象画像のクラスと他の無関係な画像のクラスの 2 つのクラスで構成される。CNN モデルのトレーニングには異なる画像セットが使用されるため、トレーニング後の重みとバイアスも異なるため、学習済み CNN の重みとバイアスは対象画像の知覚ハッシュ値として使用する。

実際のアプリケーションでは、複数の異なる画像が、紙、本、画像コレクションなどのメディアの形式で公開される。これらの画像ごとに同一のハッシュ値を一度に生成すると、コンテンツの管

理が容易になり、fine-tuningの計算時間が短縮される。グループ内の画像ごとに共通のハッシュ値を生成するCNNに基づく知覚ハッシュの設計スキームを提案した。画像グループを処理するために、CNNに基づく知覚ハッシュスキームを発展する。このスキームでは、グループの画像がトレーニングされて、学習済みCNNが生成される。重みやバイアスなどの学習済みCNNのパラメーターは、グループの画像の共通のハッシュ値を計算するために使用される。このようにして、スキームは同じハッシュ値を持つグループのすべての画像を管理できる。

Fine-tuning後のCNNの重みとバイアスを使用した知覚ハッシュスキームは、電子透かし、画像グループ、およびさまざまなアプリケーションに適用できる。ただし、対象画像ごとにCNNのfine-tuningを行う問題がある。したがって、CNNの出力に基づく知覚ハッシュのスキームを提案した。提案で使用されているVGG16は、トレーニングデータとしてImageNetの画像セットを使用して学習済みCNNである。ImageNetには多数の画像特徴が含まれている。ImageNetの画像セットをトレーニングデータとして使用してCNNをトレーニングすると、学習済みCNNは入力画像を高精度で分類できる。また、同じ画像を入力すると、学習済みCNNの出力層の同じ応答が得られる。逆に、異なる画像を入力すると、学習済みCNNの出力層の異なる応答が得られる。したがって、fine-tuningを実行しなくても、CNNの出力層の応答を使用して、対象画像の知覚ハッシュ値を生成できる。

英文要旨

2022 年 1 月 13 日

提出者: 孟　昭雄（工学研究科工学専攻電気電子情報工学領域 201970175）

論文題目: Design proposal of perceptual hashing based on convolutional neural network for digital watermarking

要旨:

Digital watermarking technology has good applications in digital copyright protection, but there are some problems with digital watermarking. For the secondary use of copyrighted works, in the system that embeds digital watermarks for each modification/editing, we need to prevent the watermark information from being diverted to other contents. And, we need to verify the embedding order of multiple watermarks without depending on trusted third parties. To solve these problems, we propose a digital rights management system using digital watermarking, perceptual hashing, and blockchain.

However, because we used conventional perceptual hashing in this system, we could not draw sufficient conclusions about the first and second problems. In order to obtain a stable digest message of an image for digital watermarking, we propose a new construction method for perceptual hashing using Convolutional Neural Network (CNN). We construct a machine-learned CNN for accepting the target image. The perceptual hash value is the message digest of the parameters such as weights and biases that make up the CNN.

In practical applications, multiple different images are published in a form of media, such as in a paper or in a book. If an identical hash value for each of these images is generated all at once, the management of the content will become easy. Therefore, we propose a perceptual hashing scheme that generates an identical hash value for images of a group by improving the perceptual hashing scheme using weights and biases of the trained CNN. This scheme will reduce the calculation time for fine-tuning a CNN compared with generating a perceptual hash value for each image in a group.

For the perceptual hashing scheme using weights and biases of the trained CNN, we needed to fine-tune the CNN for each target image, which led to inefficiency. To reduce the calculation time of perceptual hash values, we propose a construction method for perceptual hashing based on CNN that does not require fine-tuning. An image is input to the CNN and the perceptual hash value is calculated based on the response of the output layer of the CNN.

Design Proposal of Perceptual Hashing Based on Convolutional Neural Network for Digital Watermarking

MENG Zhaoxiong

Student Number: 201970175

Kinoshita Laboratory

Course of Electrical, Electronics, and Information Engineering

Graduate School of Engineering, Kanagawa University

Supervisor: KINOSHITA Hirotsugu

January 13, 2022

Contents

List of Figures

Abstract

Digital watermarking technology has good applications in digital copyright protection, but there are some problems with digital watermarking. For the secondary use of copyrighted works, in the system that embeds digital watermarks for each modification/editing, we need to prevent the watermark information from being diverted to other contents. And, we need to verify the embedding order of multiple watermarks without depending on trusted third parties. To solve these problems, we propose a digital rights management system using digital watermarking, perceptual hashing, and blockchain.

However, because we used conventional perceptual hashing in this system, we could not draw sufficient conclusions about the first and second problems. In order to obtain a stable digest message of an image for digital watermarking, we propose a new construction method for perceptual hashing using Convolutional Neural Network (CNN). In the proposed method, we construct a machine-learned CNN for accepting the target image. The perceptual hash value is the message digest of the parameters such as weights and biases that make up the CNN.

And, in practical applications, multiple different images are published in a form of media, such as in a paper or in a book. If an identical hash value for each of these images is generated all at once, the management of the content will become easy. Therefore, we propose a perceptual hashing scheme that generates an identical hash value for images of a group by improving the perceptual hashing scheme using weights and biases of the trained CNN. This scheme will reduce the calculation time for fine-tuning a CNN compared with generating a perceptual hash value for each image in a group.

Moreover, for the perceptual hashing scheme using weights and biases of the trained CNN, we needed to fine-tune the CNN for each target image, which led to inefficiency. In order to reduce the calculation time of perceptual hash values, we propose a construction method for perceptual hashing based on CNN that does not require fine-tuning. In this scheme, an image is input to the CNN and the perceptual hash value is calculated based on the response of the output layer of the CNN.

Chapter 1

Introduction

Along with the rapid development of Internet, operations such as copying, reprinting and downloading of digital works have become easier. Almost all persons who use Internet can quickly obtain a complete copy of the digital work and do not need to pay a lot of cost. The problem of copyright protection became serious. In the field of copyright protection [1], encryption technology [2], digital signature technology [3][5], and digital watermarking technology [4][5] are general methods. Encryption technology based on private or public keys can be used by users to control access to data to achieve the effect of copyright protection. Only the person with an accurate key can decrypt, but this also causes the criminal's caution and try to break the key. When the key is broken, copyright cannot be protected, and digital works can arbitrarily be reprinted, downloaded and used. The best example is the various cracked computer software. Digital signature technology is the technology in which the author of digital works signs each digital work using his own key and the detector uses a public detection algorithm to check whether the digital signature of digital works is correct or not. This method is not convenient and practical for digital works that require large amounts of copies. For example, digital images, it is necessary to add a large amount of digital signature at once. In an actual application, the effect of digital watermarking technology is better than encryption technology and digital signature technology. First, digital watermarks are embedded in digital works. Digital works do not produce any change in the senses, and they do not alert the criminals. Secondly, digital watermarks can be copied together by a copy of digital works. In other words, all copies of one digital work have the same watermark and can protect their copyright. Finally, for the existing technology, the act of attacker destroying the digital watermark may destroy the quality of digital work at the same time. As a result, the value of this digital work is lost, and the attacker's tort is made meaningless. In this way, digital watermarking technology is a strong weapon for protecting digital copyright.

However, several problems remain to be solved when it comes to digital watermarking technology [8][9]. First, for digital watermarking used for copyright protection, the watermark information should be able to prove the copyright ownership of the image, in the form of the personal information of the image author, the serial number of the image, etc. However, the watermark information is only the metadata which is irrelevant to image features and structure. Therefore, the image is only used as the carrier of the embedded watermark information, and the watermark information may be diverted to other images. If the digital watermark has a signature function, it is not possible to prevent fraud and tampering. To solve this problem, this information should be generated based on the original image. Second, digital images often need to

be modified/edited. As such, it is necessary to use multiple digital watermarks to prove the order in which the image was modified/edited, and the watermark information should also prove that the modified/edited image and the original image are same in copyright. Third, current digital rights management systems mainly depend on trusted third parties, which means the watermark information also depends on trusted third parties for verification and management. There is a certain risk that the information will be tampered with or deleted, and may cause issues which personal information protection and service continuity. Therefore, we need a digital copyright management system that does not depend on trusted third parties to provide reliable information for digital watermarking, and that can prove the embedding order of multiple digital watermarks.

We propose a digital rights management system based on digital watermarking, blockchain, and perceptual hashing [10][11]. To deal with the first and second problems discussed above, this system used perceptual hashing to generate watermark information. The functions of perceptual hashing [30] differ from those of cryptographic hash function. For example, in the cryptographic hash function [27][28][29], if the image is modified/edited, even if the embedded digital watermark, the information will be different for each bit, and we need the hash value to not change after the image is modified/edited. Perceptual hashing [30][31][32] is thus indispensable for image modifying/editing because it can generate the same perceptual hash value for visually the same image. That is, the hash value does not change even if the image is modified/edited. Each time an image is modified/edited, a digital watermark is embedded to form multiple digital watermarks. This watermark information can consequently prove that the images came from the same original image. As for the third digital watermarking problem, our system solved it by combining watermark information with a blockchain. We took advantage of the security of the blockchain [24][25][26] to store and manage the watermark information generated based on perceptual hashing and then used the timing of the blockchain [24][25][26] to prove the storage order of the watermark information, thereby proving the embedding order of multiple digital watermarks, that is, the order of image modification/editing.

In addition, in research [6], Zhao et al. proposed a digital watermark management scheme based on smart contracts and blockchain. The main purpose of this scheme is to suppress copyright infringement by increasing the cost of infringement. The main purpose of our proposed digital rights management system [10][11] is to prevent the secondary use of digital images, and to be able to verify the equivalence between the modified/edited image and the original image. And, in research [7], Ante evaluated the non-fungible token (NFT) market. NFT is transferable and unique digital assets on public blockchain, such as artwork. A NFT is to upload a file to the NFT auction market. This will create a copy of the file recorded on the digital ledger as a NFT, which can be purchased and resold in cryptocurrency. In our scheme [10][11], we use blockchain technology to store watermark information and prove the order of image modification/editing. In other words, the blockchain is used to record the creation process of the image without involving the transaction of the image.

In the proposed digital rights management system [10][11], it applied conventional perceptual hashing. Conventional perceptual hashing [33][34][35] is mainly applied for image retrieval, and in our proposed digital rights management system, perceptual hashing is applied for digital watermarking. In researches [8][9], we tested the conventional perceptual hashing and found that for image processing methods such as rotation and flipping, the conventional perceptual hashing will not achieve suitable results, that is, the perceptual hash value of the original image and the perceptual hash value of the modified/edited image by rotation or flipping is different, which the Hamming distance is not 0. In this way, the perceptual hash value cannot be applied to verify the equivalence between the original image and the modified/edited image. Conventional perceptual hashing is difficult to apply for digital rights management based on digital watermarking, so it needs propose a perceptual hash value algorithm suitable for security systems such digital watermarking and digital rights management.

In recent years, in the research on perceptual hashing, in addition to the research on conventional image feature extraction methods, more and more researchers try to use machine learning to extract image features to generate perceptual hash values [57][58][61]. We focus on the research of perceptual hashing based on machine learning and propose a perceptual hashing scheme suitable for digital rights management.

We propose a perceptual hashing scheme based on Convolutional Neural Network (CNN) [12] suitable for digital rights management system. CNN [51][52][53] have been widely used in fields related to machine vision, image classification, and so on [47][48][49][50]. The major characteristic of CNN is its use of convolutional layers and pooling layers in its intermediate layer to effectively retain and extract the features of an image. In the conventional perceptual hashing algorithms, a series of image processing steps are performed before calculating the hash values, such as reducing the size, simplifying color, removing details, and retaining only the structure information of an image. The conventional perceptual hashing algorithms do not take into account changes in the pixel position such as rotation, flipping, etc. Since the positions of the pixels are different, the computer will produce completely different data expressions, but for human, the latent structure of the image itself has not changed, while the position of the pixels has changed. Therefore, when we move the pixel position in the image, the data for the computer will be very different, resulting in a corresponding big difference between the perceptual hash value of the original image and the modified/edited image. When the image is rotated or flipped, or when other processing is performed to transform the pixel position, it can also effectively recognize that it is a similar image. This technique can make up for the shortcomings of conventional perceptual hashing algorithms and improve the calculation accuracy of perceptual hash values.

As shown in Figure 1.1, the key point of our proposal [12] is to use CNN to extract the features of the image and then hash the extracted image features to obtain the perceptual hash value of the original image. If this hash value is embedded in the original

image as watermark information, the image is both the provider and the carrier of the watermark information, which solves the problems in digital watermarking.

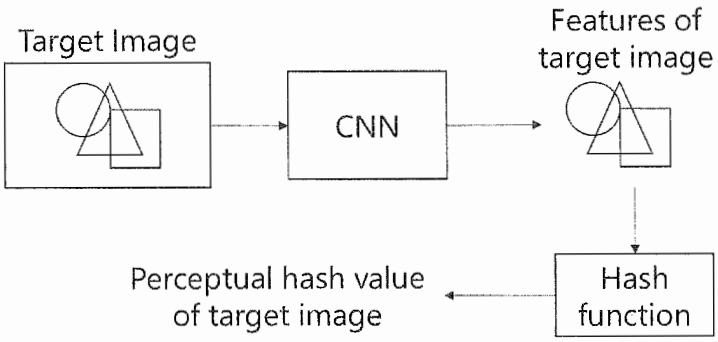

Figure 1.1: Concept of perceptual hashing based on machine learning.

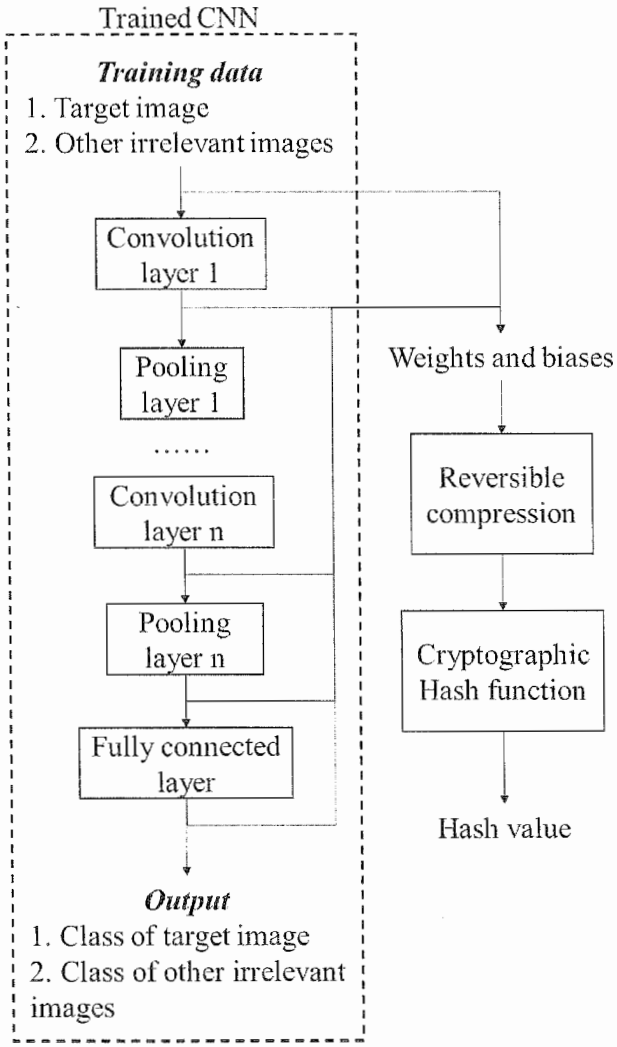

Figure 1.2: Perceptual hashing using weights and biases of trained CNN.

As shown in Figure 1.2, the specific implementation method of our proposed proposal is that, we perform fine-tuning using own training data on the existing CNN model to obtain a trained CNN. The training data of the CNN model consists of two classes of images, the target image and other irrelevant images. The output also consists of two classes, the class of target image and the class of other irrelevant images. Since different image sets are used to train the CNN models, the weights and biases after training are also different, we generate the perceptual hash value of target image using weights and biases of the trained CNN.

A difficulty is that these weights are distributed among each layer of the CNN model, so the total amount is huge. It is thus inconvenient to directly use them to calculate the hash value. Therefore, we perform reversible compression on the weights and biases, and input these compressed weights and biases into a cryptographic hash function to calculate the message digest. This message digest is used as the perceptual hash value of the target image.

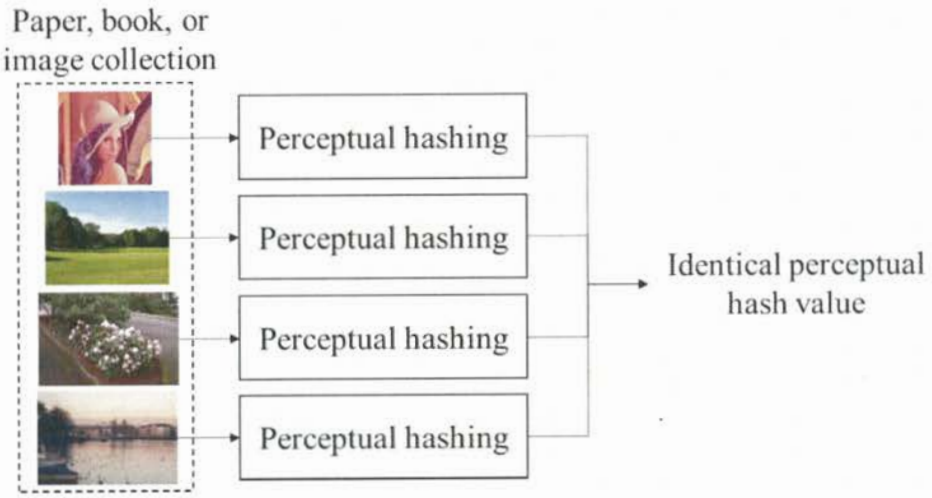

Figure 1.3: Identical perceptual hash value for each image of a form of media.

In practical applications, as shown in Figure 1.3, multiple different images are published in a form of media, such as in a paper, in a book, or in an image collection. If an identical hash value for each of these image is generated all at once, the management of the content will become easy, and the calculation time for fine-tuning will be reduced.

We will propose a design scheme for perceptual hashing based on CNN that generates an identical hash value for each image in a group. We will extend the previous perceptual hashing scheme based on CNN [12] to deal with image groups. In this scheme, the images of a group are trained to generate a trained CNN. The parameters of the trained CNN, such as the weights and biases, are used to calculate an identical hash value for the images of the group. In this way, the scheme can manage all images of a group with an identical hash value.

Perceptual hashing scheme using weights and biases of CNN after fine-tuning [12] can be applied to digital watermarking, image groups, and various applications. However, there is the problem that this proposal is necessary to perform fine-tuning on CNN for each target image. If there are too many images, it will take a lot of time. Therefore, we propose a scheme of perceptual hashing based on the output of CNN does not need fine-tuning [13]. The VGG16 [54][55] used in the proposal is CNN trained with image set of ImageNet as training data. ImageNet [71] contains a large number of image features, if the image set of ImageNet is used as training data to train CNN, the trained CNN can classify the input image with high precision. And for the trained CNN, if input the same image, it will obtain the same response of the output layer of the trained CNN. On the contrary, if input different images, it will obtain different responses of the output layer of the trained CNN. In other words, there is a one-to-one correspondence between the input image and the response of the output layer of the CNN. Therefore, even if do not perform on CNN for the image, it can use the response of the output layer of the CNN to generate the perceptual hash value of the target image.

In this paper, we analyze the problems of digital watermarking for secondary use including decretive work, and propose a digital rights management system based on digital watermarking, blockchain, and perceptual hashing. We test conventional perceptual hashing and found that conventional perceptual hashing does not provide sufficient performance for the proposed digital rights system. We analyze the requirements of perceptual hashing for digital rights management system and propose the concept of perceptual hashing based on machine learning. On the basis of the concept of perceptual hashing based on machine learning, we propose a perceptual hashing scheme using the weights and biases of CNN after fine-tuning, and the application of this scheme to image groups. And, we propose a perceptual hashing scheme using probability variable of output of general CNN applied for image classification.

The rest of this paper is organized as follows. Chapter 2 explains related works. Chapter 3 explains digital rights management system based on digital watermarking, blockchain, and perceptual hashing. Chapter 4 explains requirements for perceptual hashing. Chapter 5 explains perceptual hashing based on machine learning. Chapter 6 explains perceptual hashing using weights and biases of CNN after fine-tuning. Chapter 7 explains perceptual hashing using probability variable of output of the general CNN applied for image classification. Chapter 8 explains application for image groups based on perceptual hashing using weights and biases. At the end, Chapter 9 concludes this paper.

Chapter 2

Related Works

2.1 Digital Watermarking

Digital watermarking technology is developed from information hiding technology and is a cross-disciplinary field of digital signal processing, image processing, cryptography applications, and algorithm design. Digital watermarking was first proposed by Tirkel et al. in 1993, and the first paper about digital watermarking was published [14]. It proposed the concept and possible application of digital watermarking, and also proposed two kinds of algorithm for embedding watermarks into the least significant bit of images for grayscale images.

As shown in Figure 2.1, digital watermarking [15] is that embeds some identification information directly into digital carriers, including multimedia, documents, software, and so on, or indirectly shows, that is, modifies the structure of specific area. Digital watermarking will not affect the use value of the original carrier and not easy to be detected and modified again. But the watermark can be recognized and recognized by the producer. Through the information hidden in the carrier, it is possible to confirm the content creator, the purchaser, transmit the secret information, or determine whether the carrier has been tampered with. Digital watermarking is an effective way to protect information security, realize anti-counterfeiting traceability, and protect the copyright.

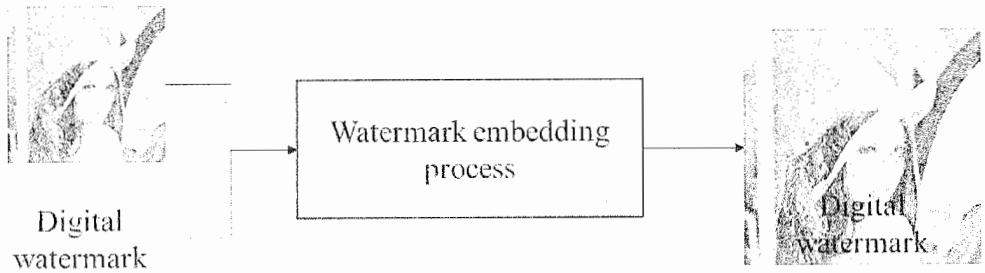

Figure 2.1: Schematic diagram of digital watermarking.

Digital watermark can be applied to many fields, the most popular of which is copyright protection. Before this, there have been many copyright protection schemes based on digital watermarks [16][17][18][19]. As can be seen from these schemes, a trusted digital watermark for copyright protection should have the following characteristics [20].

(1) Invisibility

Invisibility means that digital watermark is not easily perceived by human

perception system.

(2) Robustness

Robustness means that digital watermark can still be detected by watermark detection algorithm after various operations or attacks, and is clearly readable.

(3) Tamper resistance

Tamper resistance means that once digital watermark is embedded, it is difficult to be altered or forged by someone other than the legal holder of watermark.

(4) Watermark payload

Watermark payload means that digital watermark must have enough information to prove the copyright to play the role of copyright protection.

(5) Security and low error rate

Security means that digital watermark should have confidentiality and low false detection rate. Low error rate means that probability must be very small, which watermark is not detected when digital watermark exists (missing detection), and watermark is detected when digital watermark does not exist (false detection).

The technical realization of digital watermarking is as follows [20].

(1) Space domain

The LSB method is the easiest way to embed a watermark. In fact, any image has a certain degree of noise tolerance, which is manifested in the fact that the least significant bit (LSB) of the pixel data has little visual impact on the human vision, and the secret information is hidden in each pixel of the image. The least significant bit or the next least significant bit to achieve its invisibility.

(2) Frequency domain

The grayscale intensity of the image is regarded as the frequency domain of the picture. Transform the image to the frequency domain (wavelet domain) by some transformation means (Fourier transform, discrete cosine transform, wavelet transform, etc.), add a watermark to the image in the frequency domain, and then convert the image to the spatial domain through inverse transformation. Compared with space domain methods, frequency domain methods are more stealthy and more resistant to attack.

1) Discrete Fourier transform (DCT)

The DCT is performed in units of 8x8 pixels, and an 8x8 block of DCT coefficient data is generated. The biggest feature of DCT transform is that for general images, the energy of the block can be concentrated on a few low-frequency DCT coefficients, that is, in the generated 8x8 DCT coefficient block, only a small number of low-frequency coefficients in the upper left corner have large values, and the values of the remaining coefficients are small. This makes it possible to encode and transmit only a few coefficients without seriously affecting the image quality.

2) Wavelet transform

The wavelet is a waveform with a small area, a finite length, and a mean value

of 0. The wavelet transform is to select the appropriate basic wavelet or mother wavelet, and form a series of wavelets through the translation and expansion of the basic wavelet. Project the signal, such as an image, to be analyzed into various signal subspaces of different sizes to observe the corresponding characteristics. In this way, it is equivalent to observing an object with different focal lengths, which can be observed in great detail from macro to micro, from overview to details. So wavelet transform is also called mathematical microscope.

In this paper, we will use digital watermarking for copyright protection. It can add some important hidden information, such as copyright owner's personal information, to digital works without causing the alertness of digital work users. Moreover, the embedded watermark can be copied together with the copy of digital work. It is more convenient and effective in practical applications. We use the above-mentioned DCT method of frequency domain to implement embed and extract the digital watermark. However, digital watermarking technology has some problems, which misappropriation of watermark information, ensure equivalence between original image and modified/edited image, and prove the embedding order of multiple digital watermarks without depending on trusted third party. Therefore, we will propose a digital watermarking scheme with blockchain and perceptual hashing to solve these problems.

2.2 Blockchain

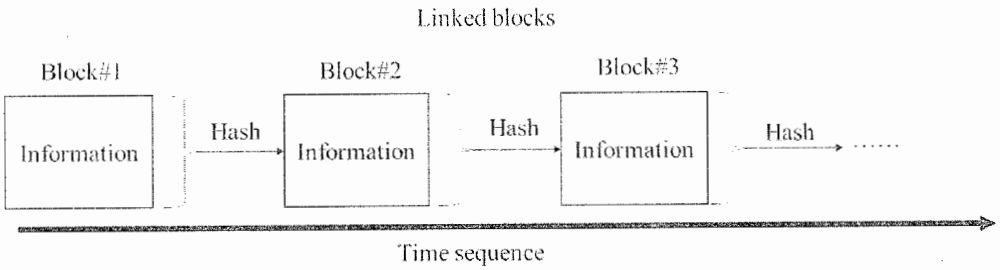

Figure 2.2: Schematic diagram of blockchain.

Blockchain technology originated in 2008, a groundbreaking paper published by the scholar named Satoshi Nakamoto in the cryptography mailing group [22], which blockchain definition has not yet been recognized in the industry. In a narrow sense, blockchain is a cannot be tampered with and unforgeable decentralized shared ledger that combines data blocks in a chronological order into a specific data structure and cryptographically guaranteed. It can securely store simple, sequential data that can be verified in the system. Broadly speaking, blockchain is a new decentralized infrastructure and distributed computing paradigm that use cryptographic chained block structures to validate and store data, use distributed node consensus algorithm

to generate and update data, and use automate script code (smart contracts) to program and manipulate data. As shown in Figure 2.2, it is the basic structure of blockchain.

Blockchain mainly has the following special features [23].

(1) Decentralization

Decentralization means that there is no centralized hardware or management mechanism in the system. The rights and obligations of any node are equal. The data blocks in system are jointly maintained by nodes with maintenance functions in the whole system.

(2) Openness

Openness means that the system is open. In addition to the private information of parties to transaction being encrypted, blockchain data is open to everyone. Anyone can query blockchain data and develop related applications through a public interface, so the entire system information is highly transparent.

(3) Autonomy

Autonomy means that blockchain uses consensus-based specification and protocol, such as a set of transparent and transparent algorithms, to enable all nodes in the entire system to exchange data freely and securely in a trusted environment, so that trust in human is changed to trust in machine, any human intervention does not work.

(4) Tamper resistance

Tampered resistance means that once the information is verified and added to blockchain, it will be stored permanently. Unless more than 51% of nodes in the system can be controlled at the same time, the modification of database on a single node is invalid, so the stability and reliability of data of blockchain are extremely high.

(5) Anonymity

Anonymity means that since the exchange between nodes follows a fixed algorithm, the data interaction does not need to be trusted, that is, the program rules in the blockchain will judge whether the activity is valid, so the counterparty does not need to open the identity to let the other party trust himself. It is helpful for the accumulation of credit.

In blockchain technology, Ethereum [80] is a decentralized open source public blockchain platform with smart contract functions. Compared to most other cryptocurrencies or blockchain technologies, characteristics of Ethereum include the following.

(1) Smart contract

The program stored on the blockchain is executed by each node, and the person who needs to execute the program pays the fee to the miner or stakeholder of the node.

(2) Decentralized Applications

Decentralized applications on Ethereum do not go down and cannot be shut down.

(3) Tokens
Smart contracts can create tokens for use by decentralized applications. The tokenization of decentralized applications aligns the interests of users, investors, and managers. Tokens can also be used for initial coin offerings.

(4) Uncle block
Incorporate shorter blockchain that have not been included in the parent chain in time due to their slow speed to increase transaction volume. The related technique of directed acyclic graph is used.

(5) Proof-of-stake
Compared with proof of work, it is more efficient, can save a lot of computer resources wasted during mining, and avoid network centralization caused by special application integrated circuits.

(6) Gas
Expanded from the concept of transaction fees, when performing various computations, it is necessary to calculate the gas consumption and pay the gas fee, including the transfer of ether or other tokens is also regarded as a computing action .

(7) Sharding
Reduce the amount of data each node needs to record, and improve efficiency through parallel computing.

Since blockchain was originally proposed as the underlying program for Bitcoin, in the first few years, the study of blockchain mainly focused on cryptocurrencies in the financial sector [23]. In recent years, the various features of blockchain have been considered by researchers to play a huge role in the field of copyright protection [24][25][26]. For conventional copyright protection, copyright owners need to provide digital works and some personal information as watermark information to the copyright registration agency. The centralized agency will manually review the submitted information and store it in the centralized server. This not only results in inefficiencies and cost increases, but also has the risk of information being tampered with and leaked. At the same time, it also brings a lot of trouble to copyright verifiers for doing digital forensics, because it is necessary to prove that this information is indeed the original information, not to be altered. For the our proposed digital rights management system in this paper, blockchain is used to store watermark information, and once this information is written into the blockchain, it will be hard to be changed. This will greatly facilitate digital forensics of copyright verifiers. In practical applications, blockchain can also help confirm multiple watermarks, because each block contains an unchangeable timestamp. If all watermark information is obtained, retrieve for the corresponding blocks in the blockchain and check the timestamps. The embedding order of the multiple watermarks can be known, in other words, the order of creation of digital images can be known. And, assuming that the blockchain applied in the

proposed digital rights management system is Ethereum, we can use the above-mentioned characteristics such as smart contracts and tokens of Ethereum to achieve copyright management and content distribution.

2.3 Cryptographic Hash Function and Perceptual Hashing

2.3.1 Concept of Hash Function

Hash function [27][28][29] is the way that create small digital fingerprints from any kind of data. The hash function compresses the message or data into a digest, making the amount of data smaller and fixing the format of data. This function scrambles the data and recreates a fingerprint called hash value. The hash value is usually represented by a string of short random letters and numbers. A good hash function rarely have hash collisions in the input domain. In hash table and data processing, not suppressing collisions to distinguish data makes database records more difficult to find.

All hash functions have a basic feature [27]: If two hash values are not the same according to the same function, then the original input of these two hash values is also different. This property is a deterministic result of the hash function, and a hash function with this property is called a one-way hash function. On the other hand, the input and output of the hash function are not unique. If two hash values are the same, the two input values are likely to be the same, but they may be different. This phenomenon is called hash collision, which is usually two input values of different lengths, deliberately calculating the same output value. Enter some data to calculate the hash value, then partially change the input value. A hash function with strong aliasing will produce a completely different hash value.

Typical hash functions have very large domains, such as SHA-2 accepts a byte string of up to $(2^{64}-1)/8$ length. At the same time, the hash function must have a finite range, such as a fixed length bit string. In some situations, the hash function can be designed to have a single size between the domain and the range. The hash function must be irreversible.

Due to the variety of applications of hash functions, they are often designed for the only one application. For example, cryptographic hash function assumes that there is an enemy that wants to find the original input with the same hash value. A well-designed cryptographic hash function is a one-way operation: there is no practical way to calculate an original input for a given hash value, which means that it is difficult to forge. Functions designed for cryptographic hash, such as SHA-2, are widely used as the checking hash functions. When the software is downloaded, the correct file portion will be downloaded after the verification code. This code may change due to changes in environmental factors such as machine configuration or IP address changes, to ensure the security of source file. Error monitoring and repair functions are primarily

used to identify instances where data is disturbed by random processes. When a hash function is used for the checksum, the hash value of a relatively short (but not shorter than a security parameter, usually no shorter than 160 bits) can be used to verify that any length of data has been altered. The function that is most widely used in hash functions is cryptographic hash function.

2.3.2 Cryptographic Hash Function

As shown in Figure 2.3, cryptographic hash function [27][28][29] is considered to be a one-way function, which means that it is extremely difficult to output the result of hash function, and what is the input data. Such a one-way function is called "the hummer of modern cryptography." The input data of such a hash function is often referred to as a message, and its output is often referred to as a message digest or a digest. In information security, there are many important applications that use cryptographic hash function to implement, such as digital signature and message authentication code.

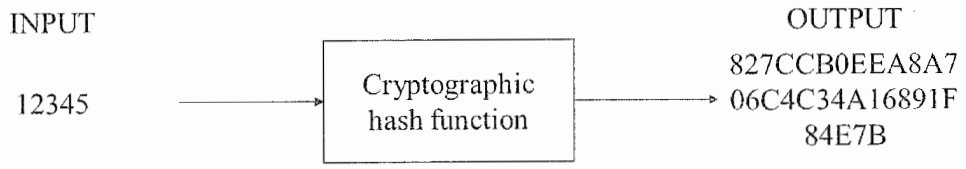

Figure 2.3: Schematic diagram of cryptographic hash function.

An ideal cryptographic hash function should have four main features.

(1) It is easy to calculate hash values for any given message.
(2) It is difficult to derive the original message from a known hash value.
(3) Modifying the message content is not feasible without changing the hash value.
(4) For two different messages, it cannot give the same hash value.

Among cryptographic hash functions, SHA256 [28] is a commonly used algorithm. SHA256 is defined as part of the SHA-2 standard devised by the National Security Agency (NSA) and standardized by the National Institute of Standards and Technology (NIST) as one of the Federal Information Processing Standards (FIPS 180-4) in 2001. SHA-2 also defines SHA224 with a hash value of 224 bits, SHA384 with 384 bits, SHA512 with 512 bits, etc. Among them, SHA256 is easy to implement. It has an excellent balance of calculation speed and cryptographic security, and is the most widely used. SHA-256 defines the calculation algorithm of the hash function, and generates a hash value of 256 bits from data of any length up to 2 to the 64th power-1 bit. While the same source text always yields the same value, even slightly different source text yields completely different values. Designed as a so-called cryptographic hash function, it is difficult to efficiently search for another source text that has the same value based on one source text.

It is because of these features that make cryptographic hash function widely used in the field of information security. However, if image is modified/edited even a little, cryptographic hash function will output a different hash value. Although image can be guaranteed that it has not been modified/edited, cryptographic hash function is not suitable for copyright management because it will be identified as different data if image is modified/edited. Therefore, we need a hash function that even if image is modified/edited, the identical hash value is output for image.

2.3.3 Perceptual Hashing

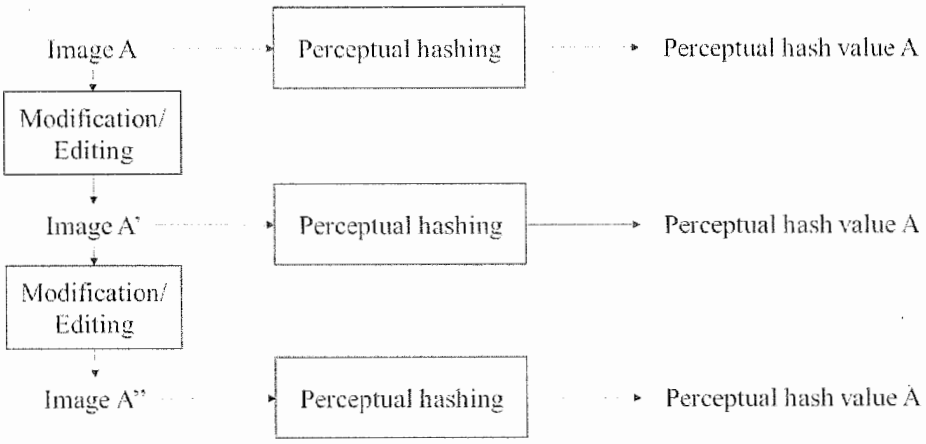

Figure 2.4: Schematic diagram of perceptual hashing.

Perceptual hashing [30][31][32] is a kind of information processing theory based on cognitive psychology. From multimedia data set to one-dimensional mapping of multimedia perceptual abstract set, the multimedia digital representation with the same perceptual content is uniquely mapped into a digital digest and satisfies perceived security requirements. Perceptual threshold theory states that objective things can only be perceived by humans when the stimulus brought by objective things exceeds the perceptual threshold. Before that, they are all the same data. Therefore, the mapping between sets of cognitive processes is a many-to-one mapping. A class of elements whose difference is less than the perceptual threshold is mapped to an element in the next set. As shown in Figure 2.4, it is the schematic diagram of perceptual hashing.

Since perceptual hashing is based on human cognitive psychology, in addition to the nature of conventional cryptographic hash function, there are some special properties [35].

(1) Collision resistance

Collision resistance means that multimedia digital representations that are different in perceived content will be not mapped to the same perceptual hash value.

(2) Robustness

Robustness means that after content retention operation, different multimedia

digital representations that are aware of the same content will be still mapped to the same hash value.

(3) Onewayness

Onewayness means that given a multimedia message and its perceptual hash value, it is easy to calculate perceptual hash value of this multimedia message by using perceptual hashing, but it is difficult to reverse the content of this multimedia information by using its perceptual hash value.

(4) Randomness

Randomness means that the ideal perceptual hash should be completely random.

(5) Transitivity

Transitivity means that perceptual hashing is transitive under the constraints of perceptual threshold, and vice versa.

(6) Compactness

Compactness means that under the premise of satisfying the above basic features, the data capacity occupied by perceptual hash value should be as small as possible.

Table 2.1: Calculation steps of the four perceptual hash algorithms.

Algorithm	Step
Average hash algorithm	Calculate the average of all pixels in the image. Compare each pixel with the average. If the pixel is greater than or equal to the average, it outputs as 1; if the is pixel less than the average, it outputs as 0. The resulting 64-bit sequence is used as the perceptual hash value.
Difference hash algorithm	In each row unit of the image matrix, for the two adjacent pixels, the left pixel minus the right pixel, get 8 differences in a row, so a total of 64 differences in all 8 rows. If the difference is positive or 0, it outputs as 1; if the difference is negative, it outputs as 0. The resulting 64-bit sequence is used as the perceptual hash value.
Perceptual hash algorithm	Perform discrete cosine transform (DCT) on the image and make the image into the 32x32 DCT matrix. Retain the 8x8 matrix in the upper left corner of the DCT matrix. Calculate the average of all coefficients in the matrix. Compare each coefficient with the average. If the coefficient is greater than or equal to the average, it outputs as 1; if the coefficient is less than the average, it outputs as 0. The resulting 64-bit sequence is used as the perceptual hash value.
Wavelet hash algorithm	Perform discrete wavelet transform (DWT) on the image and make the image into the 8x8 DWT matrix. Calculate the average of all coefficients in the matrix. Compare each coefficient with the average. If the coefficient is greater than or equal to the average, it outputs as 1; if the coefficient is less than the average, it outputs as 0. The resulting 64-bit sequence is used as the perceptual hash value.

There are four candidate calculation methods for perceptual hashing, which average hash algorithm (AHA), difference hash algorithm (DHA), perceptual hash algorithm (PHA), and wavelet hash algorithm (WHA) [30]. Note that the term "perceptual hash algorithm" refers to a general calculation method for perceptual hashing, while PHA above is a specific algorithm. Although the names are similar, they are related but not the same concept. Because the core idea of the proposed system is to use blockchain to verify and store watermark information, it is necessary to ensure that the hash value of an image as the digital fingerprint is consistent with human visual perception. That is, if it is not perceived that an image has changed, then the corresponding hash value should not change. This raises the issue of which calculation method is the most robust and suitable for this system.

The specific calculation steps and program implementation of the four conventional perceptual hash algorithms are described in detail in research [30], we will only make a brief introduction. In the calculation steps of the four algorithms, there are two identical image pre-processing steps. The first is that reduce the size of the image. This is to remove the high frequency and detail information of the image, and only retain the structure and color information of the image. Specifically, regardless of the size and proportion of the original image, reduce the image to the specific size. This is to remove differences caused by different sizes and ratios. For AHA and WHA, the image is reduced to 8x8 size, a total of 64 pixels. For DHA, the image is reduced to a size of 9x8, a total of 72 pixels. And for PHA, the image is reduced to a size of 32x32, a total of 1024 pixels. Since the main purpose of conventional perceptual hashing is to retrieve images, when implement the program, it uses the resize function to reduce the image size, such as the nearest neighbor algorithm. The second is that convert color image to grayscale image. This is to remove differences caused by different color intensities. And, the remaining calculation steps of the four algorithms are as shown in Table 2.1.

(1) For AHA, there are two steps. The first is to calculate the average of the pixels in the processed image after the pre-processing steps mentioned above. Then compare each pixel of this processed image with the calculated average. If the pixel is greater than or equal to the average, it outputs as 1; if the pixel is less than the average, it outputs as 0. The resulting 64-bit sequence is used as the perceptual hash value.

(2) For DHA,, there are two steps. The first is that in each row unit of the image matrix, for the two adjacent pixels, the left pixel minus the right pixel, get 8 differences in a row, so a total of 64 differences in all 8 rows. And then, if the difference is positive or 0, it outputs as 1; if the difference is negative, it outputs as 0. The resulting 64-bit sequence is used as the perceptual hash value.

(3) For PHA, there are four steps. The first is to perform discrete cosine transform (DCT) on the image and make the image into the 32x32 DCT matrix. This is to decompose the frequency information of the image. And then, reduce the size of DCT matrix. Specifically, retain the 8x8 matrix in the upper left corner of the original DCT matrix, because this part presents the lowest frequency in the image.

That is, only retain the general outline of the image, while removing the details and noise of the image. Next, calculate the average of all coefficients in the reduced DCT matrix. Then compare each coefficient in the reduced matrix with the average. If the coefficient is greater than or equal to the average, it outputs as 1; if the coefficient is less than the average, it outputs as 0. The resulting 64-bit sequence is used as the perceptual hash value.

(4) For WHA, there are three steps. The first is to perform discrete wavelet transform (DWT) on the image and make the image into the 8x8 DWT matrix. And then, calculate the average of all coefficients in the DWT matrix. Next, compare each coefficient in the DWT matrix with the average. If the coefficient is greater than or equal to the average, it outputs as 1; if the coefficient is less than the average, it outputs as 0. The resulting 64-bit sequence is used as the perceptual hash value.

Since perceptual hashing has a feature different from the conventional hash function, which the robustness to the modification of multimedia data, that is, the hash value does not change drastically, so perceptual hashing is usually used for similar image retrieval and image content based authentication [33][34][36][37][38].

As mentioned in Section 2.2, it is not practical to store digital images directly in the blockchain. A more practical and convenient method is to hash the images, record the hash values of these images in the blockchain, and the image files are stored elsewhere for calling. However, for multimedia file such as image file, conventional cryptographic hash algorithms such as MD5 and SHA256 are not very suitable [30][31][32]. Because in addition to tampering attacks on the content structure, digital images will undergo normal operations such as embedding digital watermarks, filtering, rotation, compression and others. These operations will not cause structural changes in the image content, therefore will not cause human sensory system produces sensory differences, certainly these images are still considered to be the same image by human. However, the data structure of this digital image file has changed for computer, so the calculated results by conventional hash functions will become completely different. Obviously this is not the result that we want to see, so it needs a new hash algorithm that is robust to content manipulation and sensitive to content tampering. In this paper, perceptual hashing performs a series of processing on images before calculating hash values, such as reducing size and simplifying color, removing details of the images, and retaining only the structure information of these images. As long as the structure of a certain image has not changed, the hash value will not change [35]. In other words, the structure information will not change after adding digital watermark to the original image, calculating the watermarked image by the same perceptual hashing, and the calculated hash value being compared with the extracted digital watermark information. In this way, a certain watermarked digital image can be self-certified without the original image [39][40].

For the proposed digital rights management system [10], an image may be modified/edited using various methods. These images may have differences, such as

rotation, cropping, symmetry, color, and sharpness, but they are essentially the same image because the content in the image has not changed. Therefore, each image will ideally result in the same perceptual hash value. For another image with different appearance, the images' perceptual hash values will differ.

However, in studies [8][9], we found that the perceptual hash values of modified/edited images such as rotation and cropping significantly differ from that of the original image, even though these images have the identical appearance for humans. That is to say, an image is modified/edited using various methods, and some modified/edited images result in different perceptual hash values. These images have the identical appearance, and the perceptual hash values recorded in the blockchain differ, which is not conducive to the management of watermark information because it is difficult to modify the information that has been recorded in the blockchain. Therefore, conventional perceptual hashing cannot satisfy the performance requirement for the proposed digital rights management system. We need to propose a perceptual hashing scheme suitable for security systems such as digital watermarking and copyright management.

2.4 Machine Learning

2.4.1 Concept of Machine Learning

Machine learning is a branch of artificial intelligence [41]. The research of artificial intelligence first focuses on reasoning, then on knowledge, and then on learning [41]. In this research process, machine learning is a way to realize artificial intelligence, that is, using machine learning as a means to solve artificial intelligence problems. In the past few decades, machine learning [42] has developed into a multi-field interdisciplinary subject, involving probability theory, statistics, approximation theory, convex analysis, computational complexity theory and other subjects. Machine learning theory is mainly to design and analyze some algorithms that allow computers to automatically learn. Machine learning algorithm [43] is a kind of algorithm that automatically analyzes and obtains rules from data, and uses the rules to predict unknown data. Because a large number of statistical theories are involved in learning algorithms, machine learning is particularly closely related to inferential statistics, and is also known as statistical learning theory. In terms of algorithm design, machine learning theory focuses on achievable and effective learning algorithms. Many inference problems are difficult to follow, so part of the machine learning research is to develop approximate algorithms that are easy to handle.

Machine learning [43] has been widely used in data mining, computer vision, natural language processing, biometric recognition, search engines, medical diagnosis, detection of credit card fraud, stock market analysis, DNA sequence sequencing, speech and handwriting recognition, strategic games, and robotics.

2.4.2 Convolutional Neural Network (CNN)

Deep learning [45][46] is part of a broader series of machine learning methods based on learning data representation, rather than task-specific algorithms. Deep learning architectures [44], such as deep neural networks, deep belief networks and recurrent neural networks, have been applied to computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis And other fields. Material inspection and board game programs, they produce results comparable to human experts, and in some cases better than human experts. In deep learning, convolutional neural network (CNN) [51] is a type of deep neural network, most commonly used to analyze visual images.

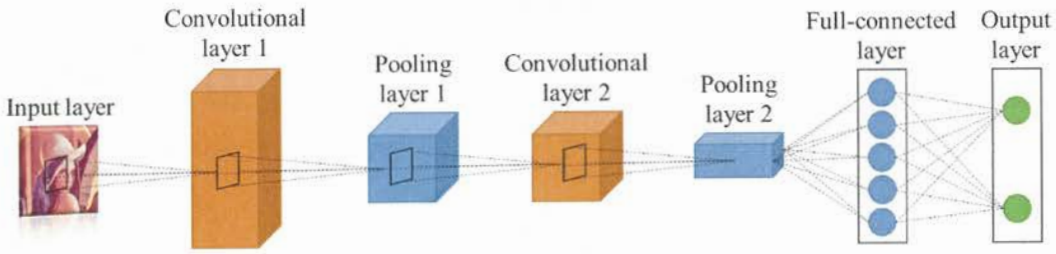

Figure 2.5: Schematic diagram of CNN.

As shown in Figure 2.5, CNN [51][52][53] uses a variant design of multi-layer perceptron and requires minimal preprocessing. They are also called shift invariant or space invariant artificial neural networks (SIANN), based on their shared weight architecture and translation invariance characteristics. Convolutional networks are inspired by biotechnology in which the connection patterns between neurons resemble the visual cortex of animal tissues. Individual cortical neurons only respond to stimuli in a restricted area of the visual field called the receptive field. The receptive fields of different neurons overlap so that they cover the entire field of view.

Compared with other image classification algorithms, CNN [51][53] uses relatively few preprocessing. This means that the network learns the filters manually designed in conventional algorithms. This independence from prior knowledge and manpower in feature design is a major advantage.

CNN [51] can be used for image and video recognition, recommendation systems, image classification, medical image analysis and natural language processing. In our research, we will propose a perceptual hashing scheme based on CNN suitable for the proposed digital rights management system.

For the CNN model for perceptual hashing, we use the VGG16 model. VGG16 [55] is a CNN architecture which was used to win ILSVR (ImageNet) competition in 2014. It is considered to be one of the excellent vision model architecture till date. Most unique thing about VGG16 is that instead of having a large number of hyper-parameter they focused on having convolution layers of 3x3 filter with a stride 1 and always used

same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 fully connected layers followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights and biases. This network has about 138 million parameters.

For the training data of VGG16, ImageNet [70] is a standard research dataset for image recognition research, which targets natural images. It shows what is reflected in more than 14 million images with more than 20,000 labels. From this ImageNet dataset, 1.2 million training sheets and 1000 label types were taken out and used in a competition called ILSVRC. In image recognition by deep learning, a deep convolutional neural network is used, but these first few layers mainly represent local information of the image, and this part is whether the target is a natural image. There is a possibility that it can be widely reused, not limited to ImageNet datasets. For this reason, it is widely practiced to reduce the overall training time by using a part of the deep convolutional neural network trained by ImageNet and retraining it with the image of the target domain.

2.5 Related Works of Perceptual Hashing Based on Machine Learning

In recent years, due to the extensive needs of multimedia security, many researchers have devoted themselves to studying perceptual hashing. Research on perceptual hashing can be roughly divided into two categories. The first category is to use conventional feature extraction methods to extract the features of the image, and then perform feature compression on this basis to obtain the perceptual hash value of the image. The second category is the perceptual hash based on machine learning we proposed in this paper, which uses machine learning to obtain the perceptual hash value of the image from the feature data set of the image. We will briefly explain the related research of perceptual hash based on machine learning below.

Faith et al. proposed perceptual hashing based on Discrete Wavelet Transform (DWT) and Support Vector Machine (SVM) for CNN to image classification [56]. After that, they applied the DWT-SVM-based perceptual hashing and CNN fusion to the classification of liver images [60]. Jiang et al. proposed perceptual hashing based on Deep Convolutional Neural Network (DCNN) to extract the features of the image and generate a hash sequence [57]. In order to detect whether the video has been tampered with, CNN was used to learn the characteristics of each frame of the video to generate the perceptual hash value of the video [58]. To automatically classify glaucoma images, Discrete Wavelet Transform (DWT) and Principal Component Analysis (PCA) were used to construct perceptual hashing, and then DWT-PCA-based perceptual hashing and CNN are applied to image classification [59]. Qin et al. proposed a new CNN training method that dynamically adjusts the structure of the training set according to

the change of the constraint value, and then generates the perceptual hash sequence of the image [61]. In [62], feature extraction was performed by applying the CNN model to reduce high-dimensional data into low-dimensional discriminative features. Then, by applying The Iterative Quantization (ITQ), the continuous real-valued features were quantized into discrete binary codes as the perceptual hash value of the image.

In our research, we propose a perceptual hashing scheme [12] using weights and biases of CNN after fine-tuning and the application of this scheme for image groups, and a perceptual hashing scheme [13] using probability variables of output of general CNN applied for image classification. Compared with related research, our scheme obtains less calculation time and higher verification accuracy of perceptual hash value.

2.6 Distributed File System

2.6.1 Concept of Distributed file System

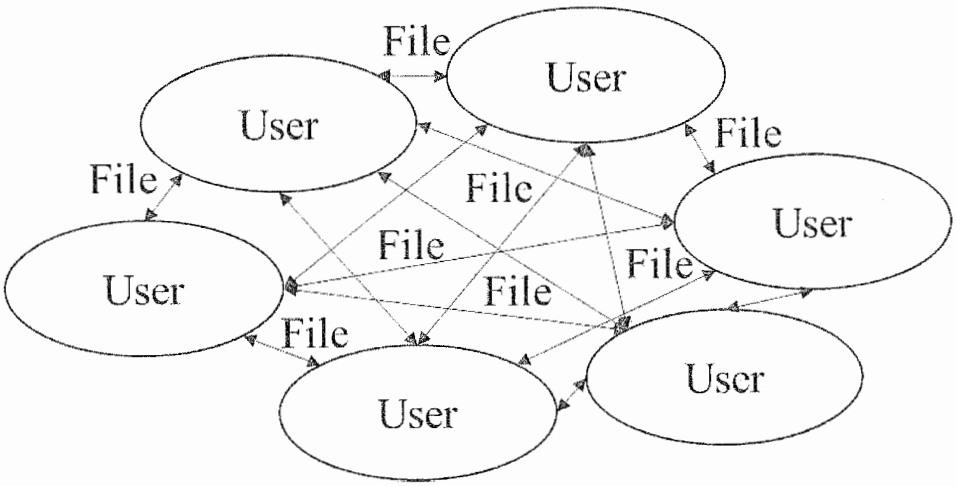

Figure 2.6: Schematic diagram of distributed file system.

As shown in Figure 2.6, compared with the file system on the local side, distributed file system [83] is a file system that allows files to be shared on multiple hosts through network, allowing multiple users on multiple machines to share files and storage space. In such a file system, clients do not directly access the underlying data storage block, but communicate with servers through network with specific communication protocol. By designing the communication protocol, both clients and servers can restrict access to the file system according to the access control list or authorization. Distributed file system mainly contains two features.

(1) Transparency

Transparency means that let the action of actually accessing the file through network, as seen by programs and users, is like accessing a local disk. The

architectural diversity and separation between servers and storage devices are hidden. The location of local files and the mechanism of data transfer in the system depends on the design of network file system.

(2) Fault tolerance

Fault tolerance means that even if a small number of nodes in the system are offline, this system as a whole can continue to operate without data loss.

The first file server was developed in the 1970s. In 1976, Digital Equipment Corporation (DEC) designed File Access Listener (FAL) as part of the second generation of DECnet. This system implements Data Access Protocol and is the first widely used distributed file system. In 1985, Sun Microsystems created Network File System (NFS), the first distributed file system widely used based on Internet Protocol. In the evolution of related technologies, other distributed file systems worth mentioning are Andrew File System (AFS), Apple Filing Protocol (AFP), NetWare Core Protocol (NCP), and Server Message Block (SMB) generally known as Common Internet File System (CIFS), etc [83].

One common way to measure the performance of a distributed file system is that how much time does it take to complete a service request [84]. In the conventional system, the time required to complete a request includes the actual hard disk access time, and a small portion of the CPU processing time. However, in a distributed file system, due to the distributed architecture, the remote access action creates additional recurring burdens, including the time to send the request from the client to the server, the time which the response is sent back from servers to clients, and the central processing time used to run the network transport protocol during these two transmissions.

In recent years, with the continuous innovation of Internet technology, the performance of distributed file systems is constantly being optimized. Through previous learning, we discovered a new high-performance distributed file system - InterPlanetary File System (IPFS).

2.6.2 IPFS (InterPlanetary File System)

IPFS (InterPlanetary File System) is a network transport protocol designed to create persistent and distributed storage and shared files. It is a content-addressable peer-to-peer hypermedia distribution protocol. The nodes in the IPFS network will form a distributed file system. It is an open source project that has been developed by Protocol Labs in the open source community since 2014 and was originally designed by Juan Benet [79]. Initially, the IPFS protocol used the advantages of Bitcoin blockchain protocol and network infrastructure to store unalterable data, remove duplicate files on the network, and obtain address information for storage nodes used to search for files on the network. Therefore, IPFS has a good correlation with blockchain.

IPFS [79] is a peer-to-peer distributed file system that attempts to connect to the

same file system for all computing devices. In some ways, IPFS is similar to World Wide Web (WWW), but it can also be thought of as a separate BitTorrent group that exchanges objects in the same Git repository. In other words, IPFS provides a high-throughput, content-addressable block storage model and content-related hyperlinks. This forms a generalized Merkle directed acyclic graph (DAG). IPFS combines decentralized hash tables, encouraging block swapping, and a self-certifying namespace. IPFS does not have a single point of failure, and nodes do not need to trust each other. Distributed content delivery can save bandwidth and prevent DDoS attacks that HTTP scenarios can encounter. This file system can be accessed in a variety of ways, including FUSE and HTTP. Adding local files to IPFS makes it available to the whole world. The file representation is based on its hash and is therefore good for caching. The distribution of files uses a BitTorrent-based protocol. Other users viewing the content also help to deliver content to others on the web. IPFS has a name service called IPNS, which is a PKI-based global namespace for constructing a chain of trust that is compatible with other NSs and can map DNS, .onion, .bit, etc. to IPNS.

As with other conventional network platforms, it needs to have a place to store images for users to browse and download. The conventional centralized storage scheme has many drawbacks, for example, it requires large-scale server storage devices. This increases the operating costs, and once the server has lost power, physical damage and other serious issues, it will affect image users' use and bring a lot of inconvenience. What is even more serious is that once the server is attacked by hackers, it will leak or destroy a large amount of important information, resulting in serious and incalculable losses. InterPlanetary File System is a peer-to-peer distributed file storage system, communications protocol and content delivery network [79]. For ease of description, this will be followed by an acronym, IPFS, to indicate this system. Different from HTTP, IPFS no longer cares about the location of a central server, and does not consider the file name and path. It only pays attention to what may appear in the file. After any file is placed on an IPFS node, a cryptographic hash is calculated based on contents of this file. When IPFS is asked for a file hash, it uses a distributed hash table to find the node where this file is located, then retrieves this file and verifies it. Therefore, using IPFS can significantly reduce the operating costs of network platform and improve the security factor of image file storage. In addition, IPFS implements an HTTP gateway, and image users can use a common browser to browse any content and download them.

Chapter 3

Digital Rights Management System Based on Digital Watermarking, Blockchain, and Perceptual Hashing

3.1 Image Modification/Editing, Secondary Use, and Derivative Work

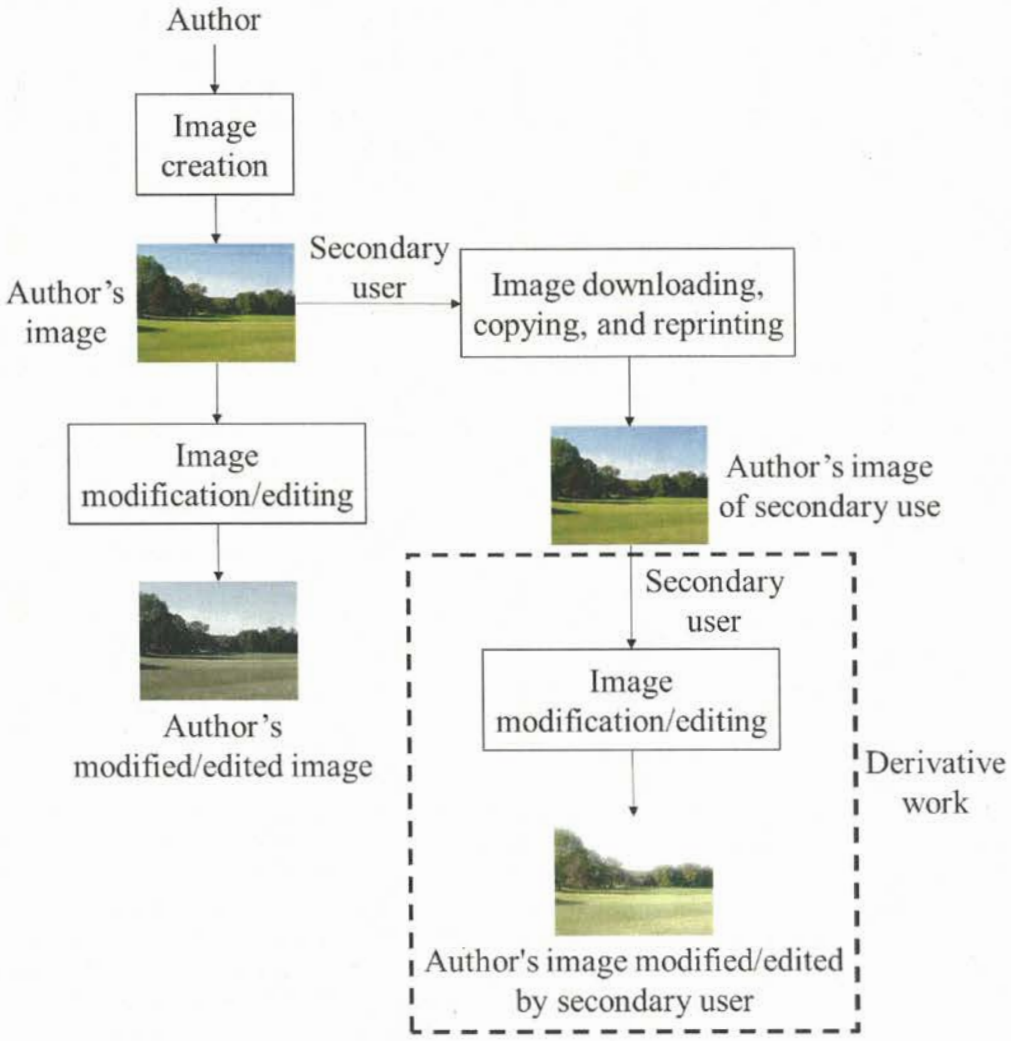

Figure 3.1: Image modification/editing, secondary use, and derivative work.

In practical application, digital images may need to be modified or edited by the author, but derivative work is created by people other than the author also occur. As shown in Figure 3.1, the author created an image and then modify/edit the image. The secondary user can get the author's image by downloading, copying, reprinting, and directly use the image. However, the secondary user may modify/edit the author's image to create the derivative work. Assuming the secondary use of copyrighted works including derivative works, in order to protect the copyright of the author, we need to propose a digital copyright management system that can prove the equivalence between the image modified/edited by the author and the image modified/edited by the secondary user.

3.2 Overview of Digital Rights Management System Scheme using Digital Watermarking, Blockchain, and Perceptual Hashing

In order to prove the equivalence between the image modified/edited by the author and the image modified/edited by the secondary user and protect the copyright of the author, we propose a digital rights management system using digital watermarking, blockchain, and perceptual hashing.

Perceptual hashing is applied to generate the message digest of the image as the watermark information. Each time an image is modified or edited, a digital watermark is embedded to form multiple digital watermarks. This watermark information can consequently prove that the equivalence between the modified/edited image and the original image.

And, blockchain is applied to register and store watermark information, and prove the embedding order of the digital watermarks and the modification or editing order of images. The requirement for blockchain is that, we use the characteristics of blockchain which without depending on the trusted third party to run, and the information recorded in the block is difficult to be tampered with or deleted. In this way, the blockchain can securely manage the watermark information and provide the verifier with the watermark information to verify the copyright of the image. And, we use the characteristic of blockchain which generate a time stamp attached to transaction information, combined with the characteristic that the information recorded in the block is difficult to be tampered with or deleted, it can prove that the embedding order of the digital watermark embedded into the images after every modification/editing. Therefore, the multiple digital watermarks are formed that can prove the image modification/editing order.

In the digital rights management system, in addition to the watermark information such as the perceptual hash value, the image file also need to be stored. Whether author creates image or modifies/edits it for others' use, the image file must be stored and distributed. The size of the image file is usually above 1MB. The image file is too large

to be stored on the blockchain. We need a storage system to store image files. We have some requirements for the distributed file system.

(1) Same as blockchain, the storage system can run without relying on a trusted third party.
(2) Files cannot be deleted or tampered with after being registered into the storage system.
(3) Files can be stored permanently without being removed.

According to the above requirements, a distributed file system is applied to store files. As shown in Figure 3.2, in the storage and sharing scheme based on blockchain and distributed file system, the image file and metadata, and the CNN model file are stored in the distributed file system such as InterPlanetary File System (IPFS) [79]. And, the watermark information of the image is recorded on the blockchain such as Ethereum [80]. This is because there is a trade-off between the amount of watermark information that can be embedded in an image and the robustness of the watermark, and we need to reduce watermark information as much as possible.

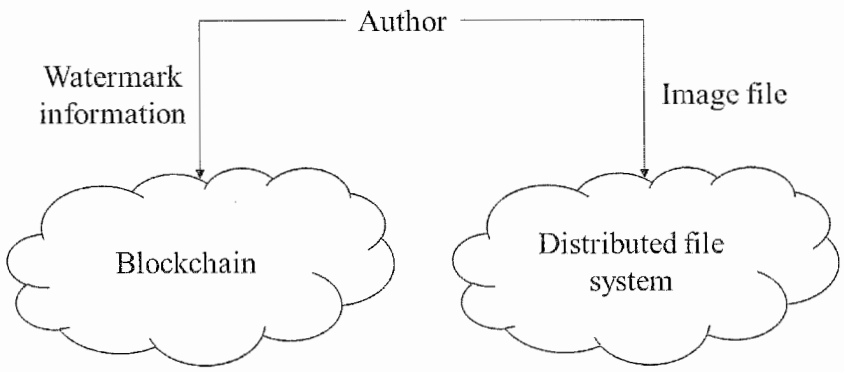

Figure 3.2: Storage of watermark information and image file.

As shown in Figure 3.3, it illustrates the overview of digital rights management system scheme.

(1) Creation of image: Author creates an image.
(2) Creation of watermark information: Author generates some information to as components of watermark information, such as the perceptual hash value, the author information, and the image meta data. The specific composition of the watermark information will be described in the following section. And, the watermark payload that the maximum length of the digital watermark information for embedding [15] needs to be considered. Because there is a trade-off between the amount of watermark information, the image quality, and the ratio of watermarks that can be detected, if the amount of watermark information exceeds the payload, the image quality and the ratio of watermarks that can be detected will be decreased. In the digital rights management system, as shown in Figure 3.4, the watermark

information is recorded in the blockchain, the watermark information is recorded in the blockchain, and the cryptographic hash value is embedded into the image. In this way, it can ensure that the watermark information does not exceed the payload, and can guarantee the security storage of the cryptographic hash value and the watermark information.

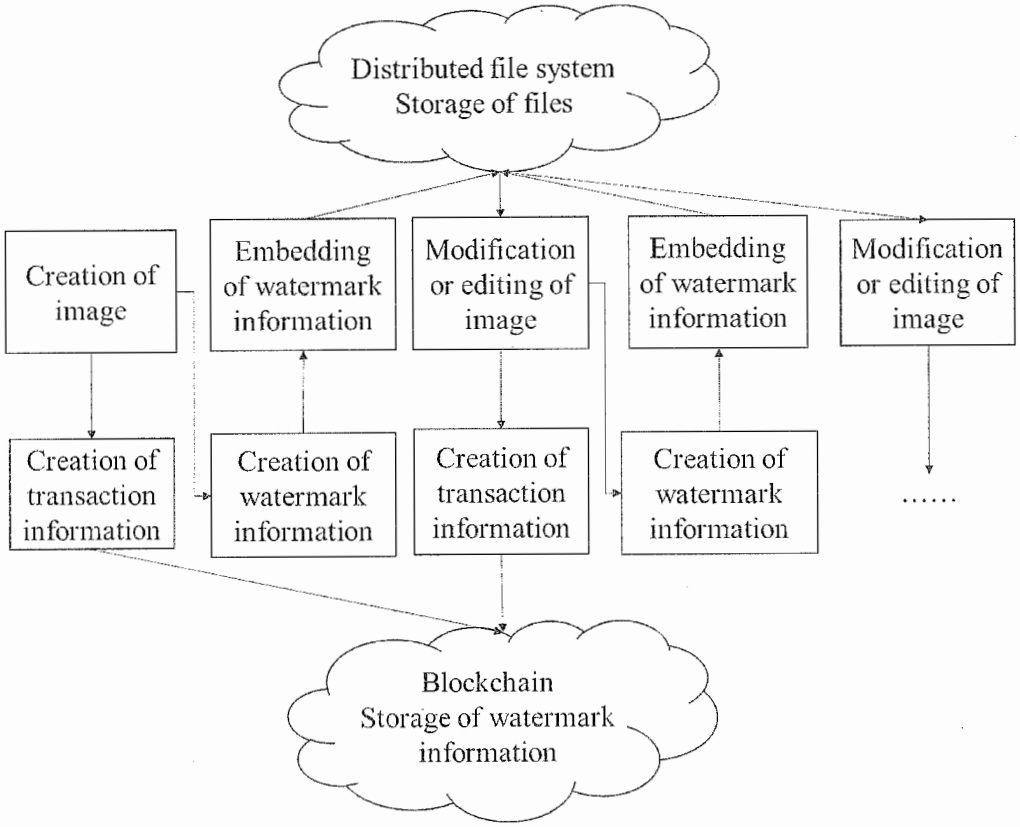

Figure 3.3: Overview of proposed digital rights management system.

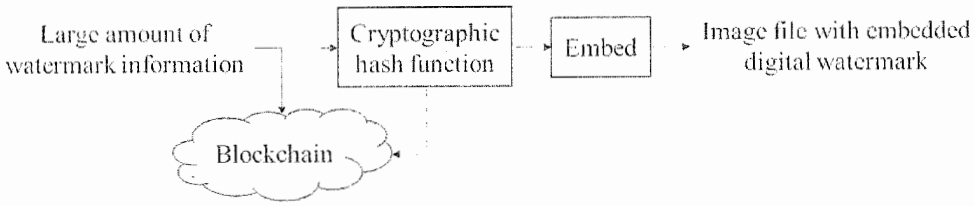

Figure 3.4: Digital watermark using blockchain.

(3) Embedding of watermark information: Author embeds the generated hash value in (2) into the image.

(4) Storage of files: Author stores the watermarked image of (3) in the distributed file system. And, author gets the information that can retrieval files from distributed file system.

(5) Creation of transaction information: Transaction information consists of the

components of the watermark information in (2), such as the perceptual hash value, author information, image metadata, and the information that can retrieval files from distributed storage system. And, blockchain generates a transaction hash to identify and search for blockchain transactions. The transaction hash consists of a character segment required to mark a transfer in a blockchain asset transaction and generally contains dozens of digits and letters.

(6) Storage of watermark information: Author stores the watermark information of (2) and transaction hash of (5) in the blockchain.

(7) Modification or editing of image: Author and secondary user get the information recorded in blockchain that can retrieval image file from distributed storage system, and modify/edit the image.

3.3 Composition of Watermark Information

Table 3.1: Composition of watermark information.

Information	Explanation
Perceptual hash value	Generated by conventional perceptual hashing [30].
Cryptographic hash value	Used to record the modification/editing history of the image, and distinguish the image before and after modification/editing.
Author information	Name, email address, and other information that can contact with author.
Image information	Title, size, and other information that can describe image.
CID of image before modification/editing	A label used to point to image file before modification/editing in the distributed file system [79].
CID of image after modification/editing	A label used to point to image file after modification/editing in the distributed file system [79].
Author's public key	Used to decrypt the author's digital signature [82].
Author's digital signature	Generated by RSA signature algorithm with SHA256 [81].

As listed in Table 3.1, some pieces of information are selected as the watermark information.

(1) The perceptual hash value of image is generated by the perceptual hashing.

(2) The cryptographic hash value of image is generated by SHA256 [28]. The cryptographic hash value is used to record the modification/editing history of the image, and distinguish the image before and after modification/editing.

(3) Author's name, email address, and other information that can contact with author are used as author information.

(4) The title, size, and other information that can describe image are used as image meta data.

(5) Content identifier (CID) is the label of the file in the distributed file system [79].

Distributed file system uses the SHA256 [28] to generate a message digest as a CID of the file. Users can use the CID retrieve the file from the distributed storage system. The author shares the image before modification/editing, and the image after modification/editing in a distributed file system. And author gets the CID of the image file before modification/editing, and the CID of the image file after modification/editing.

(6) The public key [82] is distributed to the verifier to decrypt the author's digital signature and confirm the watermark information.

(7) The digital signature is generated by RSA signature algorithm with SHA256 [81]. Author uses SHA256 to generate the message digest $h(M)$ of the message M. Author uses the secret key S_{author} to encrypt the message digest $h(M)$, and gets the digital signature $D(h(M), S_{author})$. And, verifier uses the public key P_{author} to decrypt the digital signature $D(h(M), S_{author})$, and gets the message digest $H(M)$. If $H(M) = h(M)$, the digital signature $D(h(M), S_{author})$ is confirmed that it has not been tampered with. In the scheme, the message M is concatenation of author information, meta data of image, cryptographic hash value of image, perceptual hash value, CID of image before modification/editing, CID of image after modification/editing, and the public key P_{author}.

It should be noted that we have used three kinds of hash values, which are perceptual hash value, cryptographic hash value, and CID. These three hash values are generated based on different information and different methods, and have their own uses in the system, which should be careful not to confuse.

3.4 Process of the Proposed Digital Rights Management System Scheme

Figure 3.5 shows the process of the digital rights management system. The specific composition is as follows.

(1) Digital watermarking: In the digital rights management scheme, the existing digital watermark embedding schemes are used. Specially, in order to improve the robustness of digital watermarking, the watermark embedding process adopts a method based on the frequency domain by using the discrete cosine transformation (DCT) [63][64][65]. Furthermore, the digital watermark embedding algorithm combines a scale factor [66] controlling the strength of digital watermark embedding, the frequency composition of an image [67], and the RGB-YUV mode of the image [68]. The message M and the various types of information composing message M are as the composition of watermark information. Cryptographic hash function is used to generate the message digest of the composition of watermark information, and the generated hash value is embedded into the image to match the payload of digital watermarking.

(2) Perceptual hashing: The author uses perceptual hashing to generate the perceptual hash value of the watermarked image. After comparing with the robustness of the digital watermarking scheme mentioned in (1) for image processing, it turns out that the perceptual hashing is sufficiently resistant to image processing. In this way, the equivalence between the modified/edited image and the original image can be ensured, and the copyright of the author can be protected.

(3) Message M: As mentioned in Section 3.4, the message M consists of author information, meta data of image, cryptographic hash value of image, perceptual hash value, CID of image before modification/editing, CID of image after modification/editing, and the public key P_{author}. The message M and the various types of information composing message M may constitute threats, such as the risk of being tampered with. Therefore, author uses the secret key S_{author} to encrypt the message M to get the digital signature $D(h(M), S_{author})$. And, the public key P_{author} [82] is used by the verifier to decrypt the digital signature $D(h(M), S_{author})$.

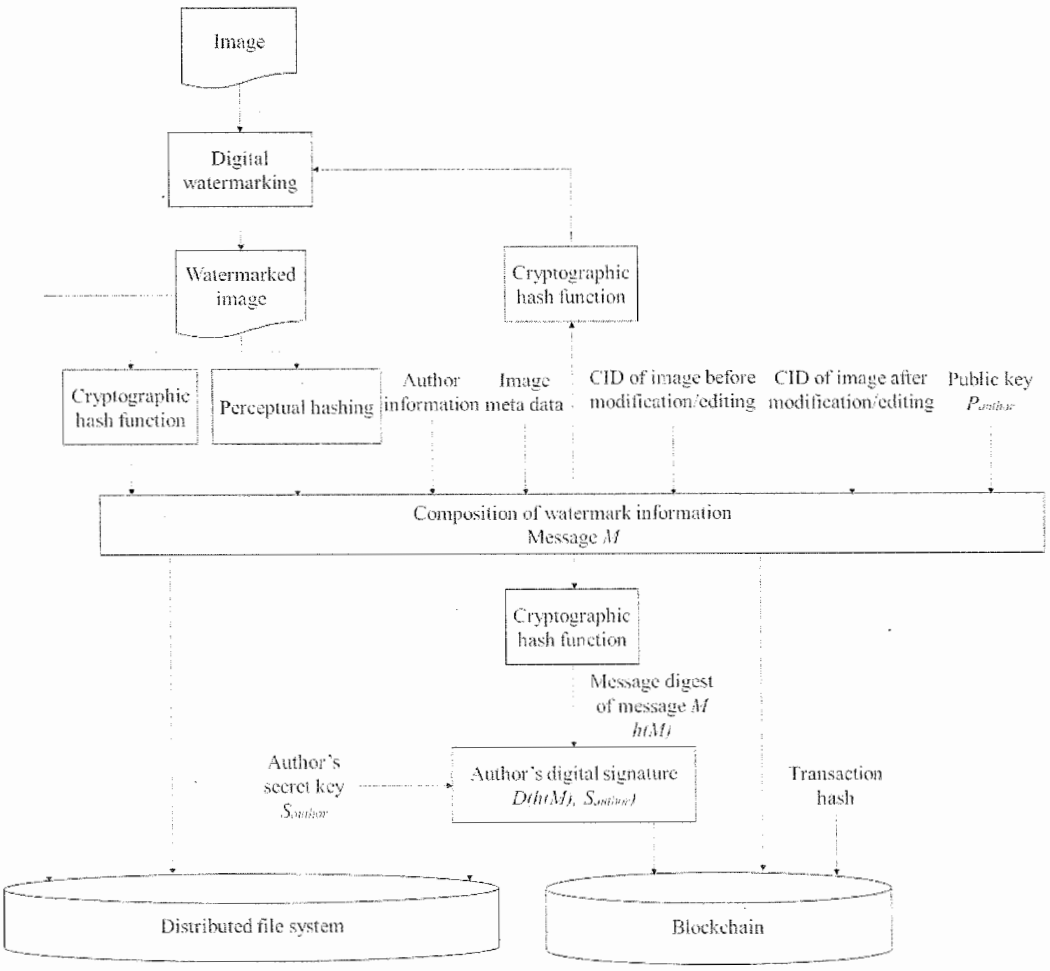

Figure 3.5: Process of digital rights management system.

(4) Blockchain: The message M and the various types of information composing M and the digital signature $D(h(M), S_{author})$ are stored on the blockchain. For each transaction, the blockchain generates a transaction hash to identify and search for transaction information, which providing protection for the information stored in the blockchain.

(5) Distributed file system: The watermarked image, the message M and the various types of information composing message M, and the author's digital signature $D(h(M), S_{author})$ are stored in distributed file system. The verifier uses the public key P_{author} to decrypt the author's digital signature $D(H(M), S_{author})$ to verify that it is the same as the information stored in the blockchain, and retrieves the hash value to obtain the file.

For image modification/editing, author or secondary user get CID of image before modification/editing from blockchain to download the image file. After the modification/editing is complete, the above steps are repeated to form a multiple digital watermark to prove the image's copyright and the modification/editing order.

3.5 Evaluation of Watermark Embedding Method

At any time, one method for copyright protection is not enough. Neither blockchain nor digital watermarking is perfect, so there is still a risk of problems appearing. The advantages of both, however, can be combined. For example, the security of blockchain can be used to store watermark information, the timing of blockchain can be used to prove the embedding order of the watermark, and the digital watermark is invisible and can be copied along with the digital format, provide protection for digital rights in the process of Internet circulation. Thus, digital watermarking technology still has a certain deterrent effect.

We used the existing digital watermark embedding scheme. Specially, to improve the robustness of digital watermarking, the watermark embedding process adopts a method based on the frequency domain by using the discrete cosine transformation (DCT) [63][64][65]. Furthermore, the digital watermark embedding algorithm combines a scale factor [66] controlling the strength of digital watermark embedding, the frequency composition of an image [67], and the RGB-YUV mode of the image [68]. As shown in Figure 3.6, first convert the image from RGB mode to YUV mode, then extract the U layer of the YUV image and embed the watermark image into the frequency domain of the U layer. After reorganizing the Y layer, U layer, and V layer, the image is converted back to RGB mode.

The experimental steps were as follows.

(1) Original images were selected from the open source datasets [76][77].

(2) The watermark is embedded into original images by the watermark embedding scheme mentioned above.

(3) 7 image-processing methods were used to modify/edit watermarked images as

shown in Table 3.2. These 7 types of image processing methods were considered typical in secondary use of copyrighted work. For each watermarked image, in each image processing method, it will randomly generate 1000 modified/edited images within the set processing intensity range.

(4) Digital watermarks were extracted from modified/edited images. And, extracted digital watermarks were verified whether the watermark information can be read clearly.

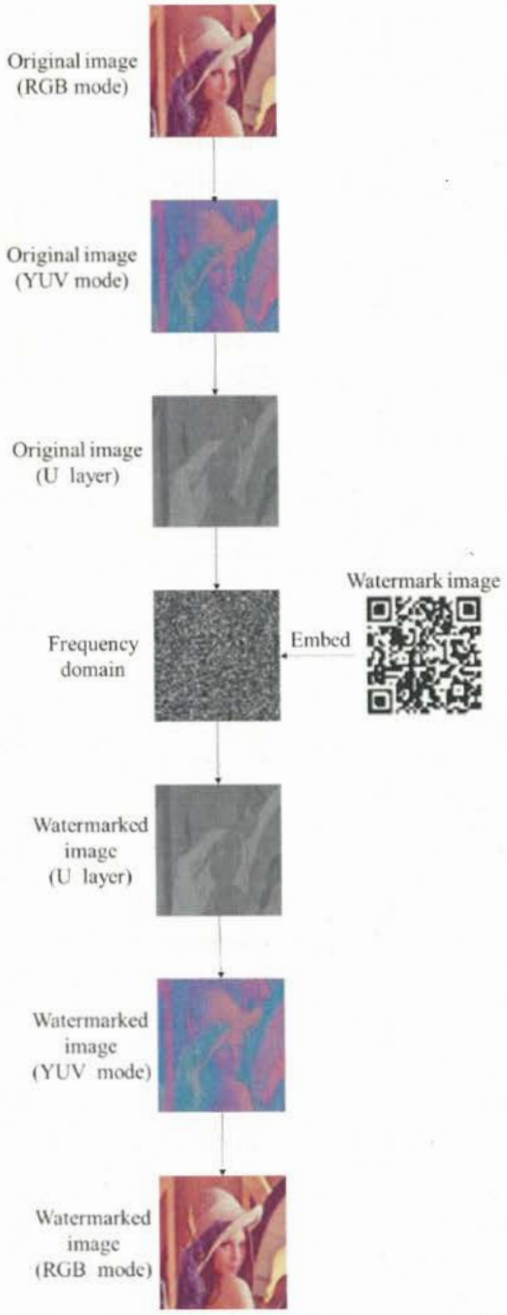

Figure 3.6: Process of embedding watermark.

It should be noted here that the image processing intensity we set is based on the basis that it will not cause image quality degradation, that is, it will not cause image users to be alert to image quality changes. Because the attacker's purpose of destroying the digital watermark is to steal the copyright interests of the original author of the image, but if the digital watermark is destroyed and the image quality is reduced at the same time, the attacker's behavior will be meaningless.

After completing the process of embedding digital watermarks, extract digital watermarks from 1000 watermarked images that have not yet been processed, and the readable rate all of them is 1.00. The readable rate mentioned here refers to the proportion of watermarked image in which the extracted digital watermark can clearly read the watermark information in all watermarked images. In this way, the impact of embedding failure on test results can be minimized.

Processing methods are performed on these 1000 watermarked images, and watermarks of the processed watermarked images are extracted. It should be noted that for rotation and flipping, we added the angle offset when extracting the digital watermark to ensure that the test results will not be affected as much as possible. After the image is rotated or flipped, the position of the digital watermark embedded element in the frequency domain (DCT) will also change. If the angle offset is not added when extracting the digital watermark, it cannot extract the clear digital watermark.

Table 3.2: Readable rate of extracted watermarks for each image processing method.

Processing	Readable rate
Noise (white noise)	0.9963
Filtering (median filtering)	0.9571
Rotation (bilinear interpolation)	0.9079
Compression (Haar wavelet transform)	0.9883
Horizontal flipping	0.9982
Vertical flipping	0.9974
Cropping (part of image)	0.9894

The readable rate shown in Table 3.2 refers to the ratio of the processed watermarked images whose watermark information can be clearly read out by the extracted digital watermark among all the watermarked images in the corresponding image processing method. It can be seen from Table 3.2 that the readable rate of the digital watermark of the seven image processing methods used in this test is above 0.90.

The existing digital watermark embedding scheme used in this paper is robust to common image processing methods. In most cases, the digital watermark can be extracted from the processed image, and the watermark information can also be read clearly.

3.6 Evaluation of Conventional Perceptual Hashing

For evaluation method, we will use a judgment benchmark of conventional perceptual hashing used for image retrieval [30]. When the Hamming distance is 0 bit, it can prove that the appearance of the two images is identical. When the Hamming distance is less than 10 bits, it can prove that the appearance of the two images is similar. When the Hamming distance is greater than 10 bits, it can prove that the appearance of the two images is different. We mentioned that the perceptual hash value is 64 bits, which is the result of the calculation steps based on the perceptual hash algorithm in Table 3.3. And according to the proportion of the length of the Hamming distance in 64 bits, it can get the appearance similarity of the two images.

In our proposed digital rights management system, perceptual hashing is applied for digital watermarking to prove the equivalence between the modified/edited image and the original image, rather than for image retrieval. For the proposed scheme, it should use a more obvious judgment benchmark than image retrieval. Therefore, for our proposed system, the judgment benchmark is that, when the Hamming distance is 0 bit, it can prove that the appearance of the two images is identical. When the Hamming distance is greater than 0 bits, it can prove that the appearance of the two images is different.

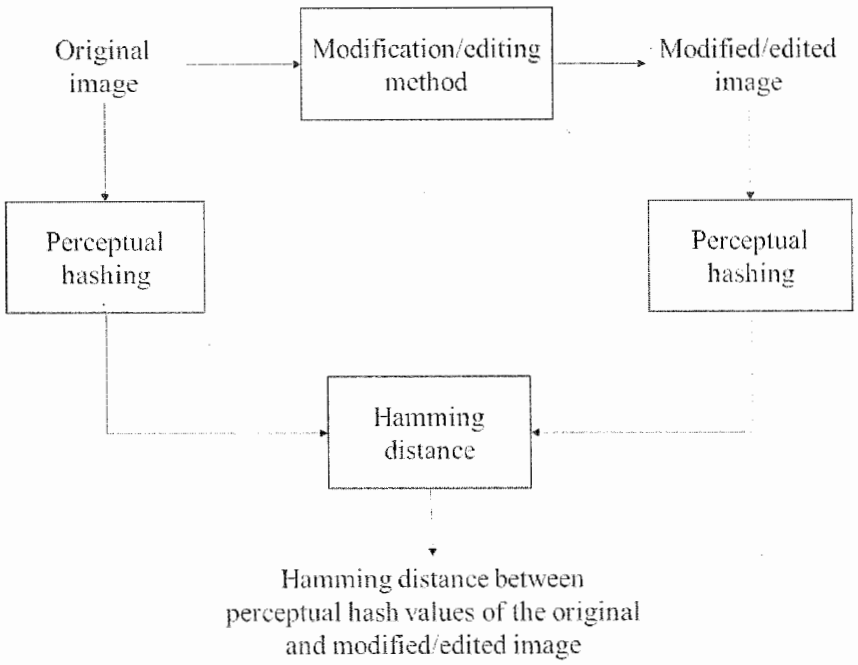

Figure 3.7: Comparison of the perceptual hash values of the original image and the modified/edited image.

We need to test them to determine which perceptual hash algorithm will be used to calculate the perceptual hash value of the image. Specially, as shown in Figure 3.7, we used the four perceptual hash algorithms to test how the perceptual hash values of images processed by different image processing methods, differed from the perceptual

hash values of the original images. The perceptual hash algorithm in the figure is the same every time calculate the perceptual hash value of the image. To compare the perceptual hash values of the processed images with those of the original images, we used the Hamming distance between the perceptual hashes of two images.

The experimental steps were as follows.

(1) Original images were selected from the open source dataset [76][77].

(2) According to the evaluation method of perceptual hashing described in the research [30], 8 image-processing methods were used to modify/edit original images as shown in Table 3.3. These 8 types of image processing methods were considered typical in secondary use of copyrighted work. For each original image, in each image processing method, it will randomly generate 1000 modified/edited images within the set processing intensity range.

(3) Perceptual hash values of original images and perceptual hash values of modified/edited images were generated by four conventional perceptual hash algorithms. And, the Hamming distances between the perceptual hash values of original images and perceptual hash values of modified/edited image were calculated.

Table 3.3: Mean Hamming distance of four perceptual hash algorithms (all values in bits).

Mean / Algorithm / Processing	Average hash algorithm	Difference hash algorithm	Perceptual hash algorithm	Wavelet hash algorithm
Watermark (discrete cosine transform)	0	0	0	0
Noise (white noise)	0.15	0.66	0.22	0.14
Filtering (median filtering)	0.2	0.9	0.42	0.34
Compression (Haar wavelet transform)	0.32	0.97	0.92	0.44
Cropping (part of image)	8.63	3.22	9.24	5.12
Rotation (bilinear interpolation)	9.34	17.03	20.08	10.04
Horizontal flipping	18.17	29.93	31.36	19.74
Vertical flipping	33.74	25.28	31.58	34.46

As shown in Table 3.3, these are experimental results of four perceptual hash algorithms. For watermark, noise, filtering, and compression, four perceptual hash algorithms got results that meet the above-mentioned benchmark for our proposed digital rights management system. However, for cropping, rotation, horizontal flipping, and vertical flipping, the Hamming distance between perceptual hash values of the modified/edited image and the original image is greater than 0 bits. Conventional perceptual hashing do not meet the benchmark for the proposed digital rights management system, which generates different hash values for the modified/edited

image and the original image, and cannot verify the equivalence between the modified/edited image. Therefore, we need to propose a perceptual hashing scheme that can be applied to the proposed digital rights management system.

3.7 Analysis of the Proposed Digital Rights Management System Scheme

3.7.1 Composition of the Proposed Digital Rights Management System Scheme

Table 3.4: Composition of the proposed scheme.

Composition	Function
Author	Create and modify/edit the image.
Secondary user	Modify/edit author's image or secondary used images.
Author's image	Be created and modified/edited by author.
Secondary used image	Be modified/edited author's image or secondary used images.
Perceptual hashing	Generate the digital digest that if the image is considered to be visually the same, the digital digest does not change even if the image is modified/edited. This digest serves as evidence for digital forensics that can prove the relevance of modified/edited images.
Blockchain	Manage copyright and watermark information of author and multiple digital watermarks. Responsible for the robustness of system, and as a security system to prevent tampering. This information serves as evidence that can prove the sequence of image modification/editing and the sequence of multiple digital watermarks.
Digital watermarking	Embed watermark information into author's image to protect copyright.

As shown in Table 3.4, the proposed scheme consists of seven components. They are respectively author, secondary user, author's image, secondary used image, perceptual hashing, blockchain, and digital watermarking.

(1) Author is the people who create and modify/edit image.

(2) Secondary user is the people who modify/edit author's image or secondary used images. In this part, the author is the main user of the system, and the secondary user is the main attacker of the system.

(3) Author's image is the image created and modified/edited by author.

(4) Secondary used image is modified/edited author's image or secondary used images. For secondary used image, this refers to the general term that the secondary user makes multiple modifications/edits to the image, not just the second time. Secondary user use author's image and secondary used image by modifying/editing,

such as uploading to the Internet for browsing and distributing.

(5) Perceptual hashing is used to generate the digital digest that if the image is considered to be visually the same, the digital digest does not change even if the image is modified/edited. This digest serves as evidence for digital forensics that can prove the relevance of modified/edited images.

(6) Blockchain is used to manage copyright and watermark information of author and multiple digital watermarks. Responsible for the robustness of system, and as a security system to prevent tampering. This information serves as evidence that can prove the sequence of image modification/editing and the sequence of multiple digital watermarks.

(7) Digital watermarking is used to embed watermark information into author's image to protect copyright.

In this research, the main purpose is to propose a design scheme for digital rights management system combining digital watermarking, blockchain, and perceptual hashing. And, in order to improve the performance of digital watermarking in copyright protection, used the characteristics of perceptual hashing and blockchain to make up for the problems of digital watermarking mentioned in Chapter 1, which first, for conventional digital watermarking, a digital image is used only as a carrier for embedded watermarking information, and as this information may be diverted to other images, the watermark information needs to be generated based on the original image. Second, after the original image is modified/edited, the watermark information needs to prove that it is from the original image. Third, multiple digital watermarks need to be stored and managed without depending on trusted third parties.

The above perceptual hash value, watermark information, multiple digital watermarks, and other watermark information stored and managed by the blockchain, provide evidence for digital forensics.

Therefore, in view of the main purpose of this paper mentioned above, specific schemes of digital watermarking, blockchain, and perceptual hashing are beyond the scope of this paper. We will use the characteristics of digital watermarking, blockchain, and perceptual hashing to analyze the threats that the system of providing digital forensics may encounter, and propose corresponding countermeasures.

3.7.2 Threat Analysis and Corresponding Countermeasures

3.7.2.1 Treats and Countermeasures of the Proposed Digital Rights Management System Scheme

It can be seen from Table 3.4, and the content mentioned above that the subject that poses a threat to our proposed system is the secondary user of the image. Secondary user may attack perceptual hashing, blockchain, and digital watermarking used in the system, thereby undermine evidence left by the system for digital forensics.

As shown in Table 3.5, we summarize the threats that may encounter based on the

above, and propose corresponding countermeasures.

Table 3.5: Threats and countermeasures of the proposed scheme.

Threat	Countermeasure
Misappropriation of watermark information	Use perceptual hashing to generate watermark information based on the information of the image itself.
Proof of image modification or editing	Use perceptual hashing to generate watermark information, use blockchain to manage watermark information, and finally form multiple digital watermarks that can prove the watermark embedding order and image relevance.
Reversal attack against perceptual hashing	For reversal attack, it is difficult for images generated in reverse by perceptual hash value to have practical value, so it is clear that reversal attack correspond to perceptual hashing used for copyright is meaningless.
Collision attack against perceptual hashing	For collision attack, although it is possible to generate two images with the same perceptual hash value but different appearances, the generated images have no practical value, so it is clear that collision attack correspond to perceptual hashing used for copyright is meaningless.

3.7.2.2 Misappropriation of Watermark Information

Threat: After completing the creation, the image author will use the system to record the watermark information on the blockchain, and then embed the digest of this information as watermark information into the image. This watermark information not only proves the copyright of the image, but also plays the role of the image author's signature. The secondary user of the image may misappropriate this watermark information and pretend to be the identity of the original author of the image to deceive the system. Finally, the secondary user achieves the purpose of capturing the copyright interests of the image author.

Countermeasure: Use perceptual hashing used to generate a hash value based on the content of the image itself. The image author uses this hash value as the watermark information. The secondary user needs to make the perceptual hash value of the image in which the watermark information is illegally embedded, equal to the hash value of the original image. For perceptual hashing, each bit of the perceptual hash value corresponds to each 8x8 pixel, the metric is arbitrarily determined within the range corresponding to the value of each bit of the hash value. Therefore, by performing appropriate enlargement processing corresponding to each pixel value, it can generate an image in which hash values collide. However, for this attack method, it is impossible to apply to different images due to deterioration of image quality. That is, it is difficult for the secondary user to pretend to be the original author.

3.7.2.3 Proof of Image Modification/Editing

Threat: The image author often needs to modify or edit the created image. Therefore, it is also necessary to use the system to record the entire modification or editing process, such as, the order of image modification or editing, and the connection between different versions of modified/edited images. The secondary user of the image may use the opportunity to modify or edit the image to secondary use the image and become the copyright owner of the modified or edited image.

Countermeasure: Combine perceptual hashing, blockchain and digital watermarking used in the system. Image author can use perceptual hashing to prove the relationship between different versions of modified/edited images on the hash value, that is, images that look the same will have the same perceptual hash value, and images that look similar will have similar perceptual hash values. The image author calculates the perceptual hash value of the image after each modification/editing, records it on the blockchain, and embeds it into the modified/edited image as watermark information. Image author can use the timing of blockchain and multiple digital watermarks to prove the order of image modification/editing. In this way, the secondary user of image will be difficult to destroy the order of image modification/editing and the connection between different versions of modified/edited images. Therefore, it is difficult for the secondary user to take the opportunity to become the copyright owner of the modified/edited image.

3.7.2.4 Reversal Attack and Collision attack Against Perceptual Hashing

Threat: In related research [69], it is found that conventional perceptual hash algorithms are at risk of being subjected to reverse attacks, that is, the perceptual hash value can be used to reversely generate the image that can calculate this perceptual hash value. Another risk that may be encountered is the risk of hash collisions, that is, two different images are calculated with the same perceptual hash value by the same perceptual hash algorithm.

Countermeasure: For reversal attack, although it was found in [69] that the perceptual hash value can be used to reversely generate an image, the application scenario of perceptual hashing in our research is copyright, and the generated image needs to have no problems in actual application to pose a threat. It can be seen from the results that it is difficult for images generated in reverse by perceptual hash value to have practical value, so it is clear that reversal attack correspond to perceptual hashing used for copyright is meaningless. For collision attack, in [69], although it is possible to create images with the same hash value but different appearances, that is, generate collision images, any image can be used for copyright purposes to the extent that there is no practical problem. It is clear that it is meaningless to generate a meaningful image that is, an image that is fraudulent in copyright management, and not to generate a similar hash value.

Chapter 4

Requirements of Perceptual Hashing for Digital Rights Management

Table 4.1: Requirements and applications for different hash functions.

Type of hash function	Requirement	Application
Cryptographic hash function [28]	A hash value $Hash\ X$ is generated for the digital data $Data\ X$. If the digital data $Data\ X$ is modified/edited to generate the digital data $Data\ X'$, $Hash\ X' \neq Hash\ X$. For the digital data $Data\ Y$ that is different from the digital data $Data\ X$, $Hash\ Y \neq Hash\ X$.	Ensures that digital data has not been tampered with for digital forensics.
Conventional perceptual hashing [34]	According to difference in message digests, similarity in appearance of images is evaluated. Certain allowable range of message digests between images is required.	Image retrieval that matches human perception.
Perceptual hashing applied for digital watermarking and copyright management	An identical perceptual hash value $Hash\ A$ is generated for an image $Image\ A$ and its modified/edited images. Another identical perceptual hash value $Hash\ B$ is generated for an image $Image\ B$ and its modified/edited images. If $Image\ A \neq Image\ B$, then $Hash\ A \neq Hash\ B$.	Ensures equivalence between modified or edited image and original image.

The message digest is an important fundamental technology along with cryptography in security systems. As shown in Table 4.1, in fields such as digital forensics, a cryptographic hash function [28] is applied to generate a message digest to ensure that digital data has not been tampered with. However, this function depends on each bit of digital data to generate a message digest, and it is sensitive to changes of bits in the

data. This function cannot be used to inspect the identity of original and secondary images that are premised on modification/editing.

Perceptual hashing [34] generates the same message digest for images considered to be the same to human perception. Perceptual hashing is suitable for guaranteeing identity on the premise of modification/editing. As shown in Table 1, conventional perceptual hashing [34] is mainly used for image retrieval that matches human perception. It can evaluate the similarity in the appearance of images on the basis of the difference in message digests. For example, the Hamming distance between message digests is used to evaluate similarity. However, conventional perceptual hashing requires a certain allowable range of message digests for image retrieval. In researches [8][9], we found a problem with conventional perceptual hashing. For image processing methods such as rotation and flipping, the modified/edited and original images are generated with different message digests. As a result, for rotation and flipping, conventional perceptual hashing will not be able to identify the equivalence between modified/edited and original images.

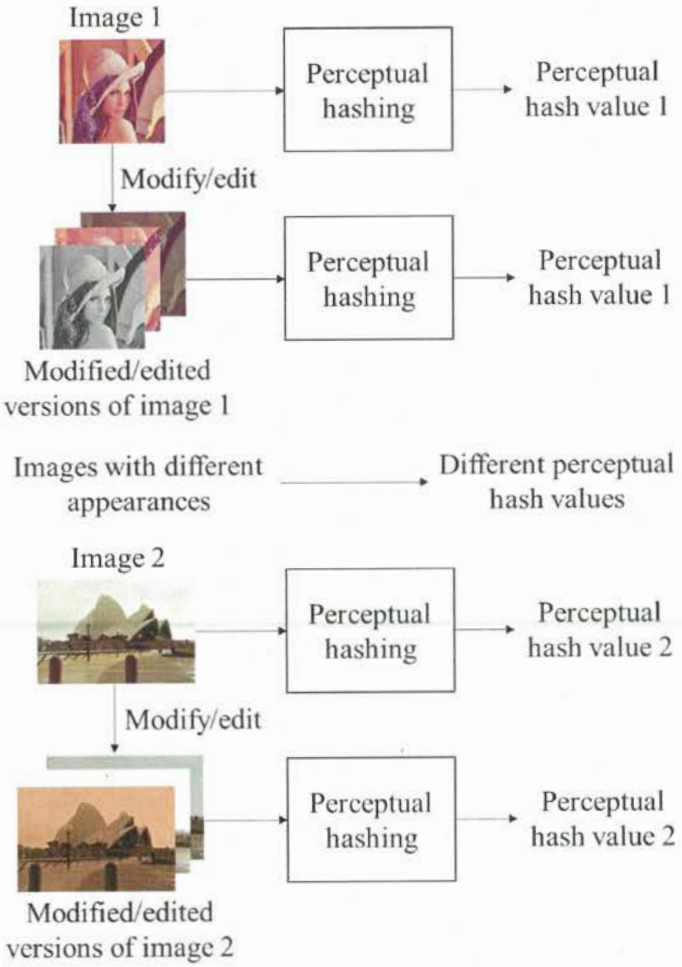

Figure 4.1: Requirements of perceptual hashing for digital rights management.

For perceptual hashing used for digital watermarking and copyright management, there are some requirements for perceptual hashing as shown in Table 4.1.
An identical perceptual hash value $Hash\ A$ is generated for an image $Image\ A$ and its modified/edited images. Another identical perceptual hash value $Hash\ B$ is generated for an image $Image\ B$ and its modified/edited images. If $Image\ A \neq Image\ B$, then $Hash\ A \neq Hash\ B$.

As shown in Figure 4.1, it is the concrete example of requirements for perceptual hashing. Image 1 and each version of the modified/edited image 1 are generated an identical perceptual hash value 1. And, image 2 and each version of the modified/edited image 2 are generated an identical perceptual hash value 2, which is different perceptual hash value 1.

Since conventional perceptual hashing does not meet the proposed requirements, we need to propose a perceptual hashing scheme for digital rights management system.

Chapter 5

Perceptual Hashing Based on Machine Learning

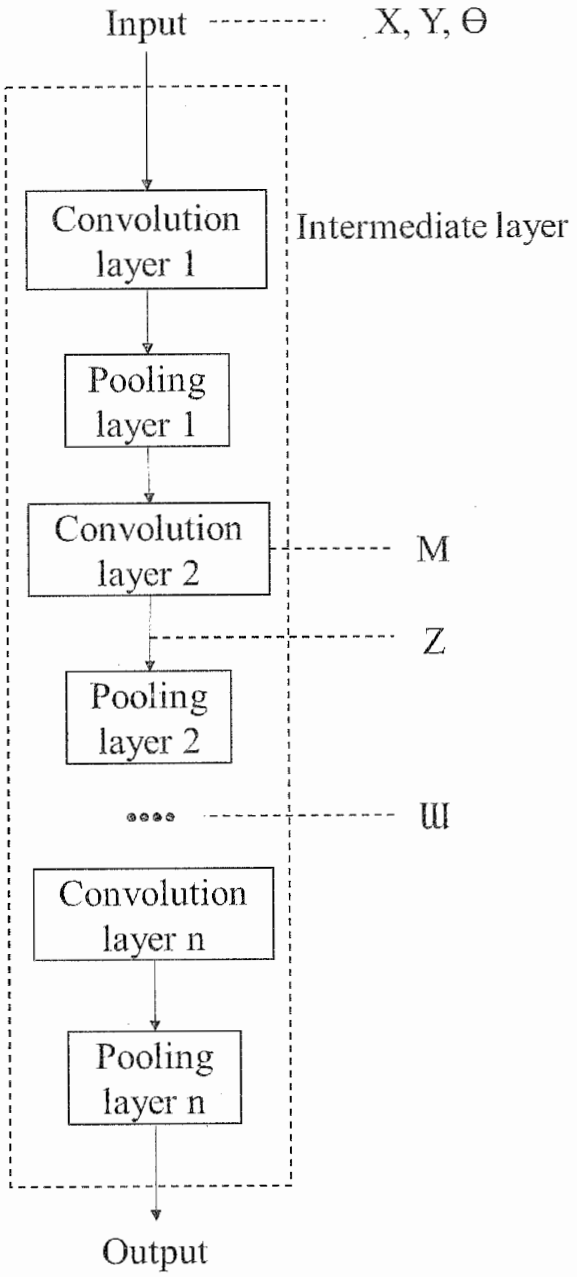

Figure 5.1: Schematic of machine learning with CNN.

The key to calculating perceptual hash values is to extract suitable image features. If these image features do not change after the image has been modified/edited, the calculated perceptual hash value will not change. However, it is not easy to find these constant image features with current perceptual-hash algorithms. To solve this problem, we use machine learning [42][43], which is widely used in computer vision to simulate the image learning and analysis process of humans, to continuously learn and analyze images and finally extract better image features to calculate the perceptual hash value. Machine learning is largely based on probability theory and statistics because it is used to learn and analyze known data to obtain the law then use this law to predict unknown data. In the case of probability theory, it is the probability that a certain situation occurs in unknown data on the premise that a certain situation occurs in the known data. This is the Bayes' theorem in probability theory. Therefore, machine learning can also be explained as a model based on this theorem, which uses known data to generate unknown data.

However, the amount of features of each image is large. Before carrying out machine learning, we need to further process the image set, i.e., reducing the amount of features of the image set before learning, finding image features suitable for perceptual hashing from a wide variety of image features, and finally enabling the output of the machine-learning algorithm to represent common features of each image in the image set. For this purpose, we use a CNN to extract the features. The artificial neurons of a CNN simulate the characteristics of neurons in the human vision [51]. The convolutional layers of a CNN extract different features of the input image. The first convolutional layer may only extract some low-level features of the image such as edge and line. The higher convolutional layers can extract more advanced and complex features such as information representing the overall structure of the image by continuous convolution. A CNN has a certain degree of translation, scaling, and rotation invariance, making it more accurate in computer vision. For the CNN shown in Figure 5.1, we focus on the intermediate layers such as convolutional layers and pooling layers [51]. From the different image features extracted from each convolutional layer, the features suitable for calculating perceptual hash values are selected as the learning data of the machine-learning algorithm, the amount of features of the image set is reduced, and the learning efficiency and effectiveness of machine learning is improved. Parameters X, Y, Θ, M, $\mathbb{W}$, and Z in the figure are defined in later in this section.

As mentioned above, machine learning can also be explained as a model based on Bayes' theorem, which uses known data to generate unknown data that the Bayesian generation model. We use the probabilistic graphical model to represent this improvement. Figure 5.2 illustrates the probabilistic graphical model of machine learning with CNN. The parameters in the figure are all stochastic variables. The black nodes represent the random variables given to the observed values, i.e., stochastic variables calculated from given known data. The white nodes represent the stochastic variables that are estimated, i.e., we need to estimate the latent stochastic variables.

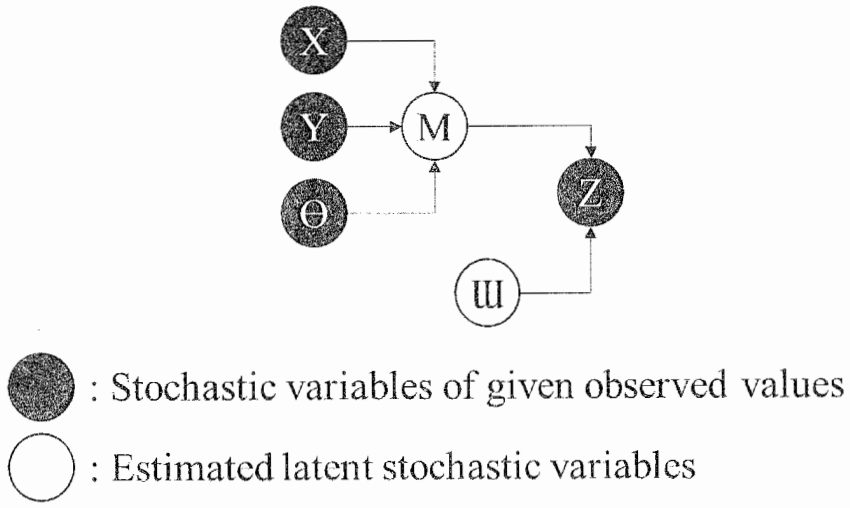

Figure 5.2: Probabilistic graphical model of machine learning with CNN.

We now define each random variable. Variable X represents the original image, Y represents the image set formed from X after various modifications/edits, Θ represents the parameters of the image modification/editing, which is the random variable of each modification/editing style. As shown in Figure 5.1, these variables are used as inputs to a CNN. Variable M represents the intermediate-layer output after X, Y and Θ are input into the CNN, which some image features of Y, Z represents the feature data of the image and is used to calculate the perceptual hash value that was not changed after modification/editing, and Ш represents the latent stochastic variable, which the rule find out Z from M. At this point, our method of improving perceptual hashing based on machine learning finishes.

Chapter 6

Perceptual Hashing Using Weights and Biases of CNN after Fine-tuning

6.1 Concept of Perceptual Hashing Based on CNN

In work [12], we propose a design scheme for perceptual hashing based on CNN. VGG16 [54][55] is used for this scheme, which is trained with an image set of ImageNet [70]. ImageNet [71] contains a large number of image features. Therefore, VGG16 can be used to classify a variety of image structures.

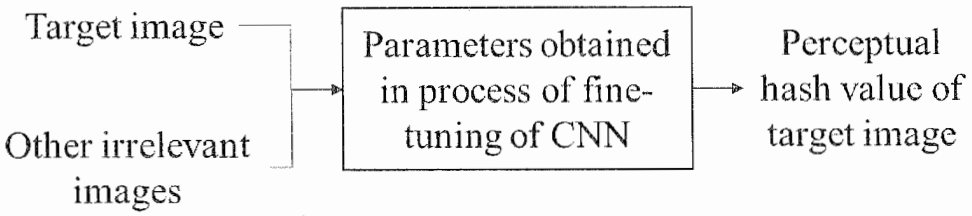

Figure 6.1: Overview of perceptual hashing based on CNN.

Figure 6.1 shows an overview of the proposed perceptual hashing scheme [12]. In this scheme, a target image and other irrelevant images are used to fine-tune the CNN. Parameters of the process of fine-tuning are obtained. A perceptual hash value of a target image is generated on the basis of the obtained parameters. The specific scheme is as follows.

(1) A CNN is fine-tuned with a target image and other irrelevant images. Thus, both types of images can be distinguished.

(2) The output of the trained CNN is categorized into an acceptance class and a rejection class. The CNN is trained to accept the target image and reject other irrelevant images. Thus, the structure information of the trained CNN, including weights and biases, is utilized as a fingerprint of the target image.

(3) A message digest of the weights and biases of the trained CNN is generated by using a cryptographic hash function as the perceptual hash value of the target image.

For the CNN-based perceptual hashing scheme, the requirements for perceptual hashing mentioned in Chapter 4 are redefined as follows.

- The trained CNN model, $Model(A)$ is composed of the weights and biases of the trained CNN, which categorizes images belonging to $Acceptance$ and $Rejection$ using image set A. For $a_x \in A$, which images to be calculated the identical

perceptual hash value are elements of set A, $Acceptance$ is output, and an identical trained CNN model, $Model(A)$, is generated. The cryptographic hash value of model $h(Model(A))$ is used as the perceptual hash value. For a set B that is different from set A, it generates another identical trained CNN model, $Model(B)$, that classifies all images belonging to set B into acceptance. If $A \neq B$, then $Model(A) \neq Model(B)$, and $h(Model(A)) \neq h(Model(B))$. The above definition is formulated as follows.

$$\begin{aligned} Classify(Model(A), a) &= Acceptance, if\ a \in A \\ Classify(Model(A), a) &= Rejection, if\ a \notin A \end{aligned}$$

6.2 Explanation of Data Augmentation

Before training the CNN, we perform data augmentation on the target image and other irrelevant images. We have two main purposes for data augmentation.

(1) We need to perform data augmentation on images to increase the amount of training data.

(2) We need to use various image processing methods to modify/edit images in the process of augmentation. In addition, we need to ensure that the modified/edited images maintain the same appearance as the original image by changing parameters for augmentation. Thus, the trained CNN can be used to ensure equivalence between the modified/edited and original images.

For the proposed perceptual hashing scheme [12], the definition is as follows.

- K types of data augmentation are performed on image a, such as white noise, median filtering, and rotation.. In data augmentation, the kth modification/editing operation is performed with parameter p in P types to generate images $Aug_{k,p}(a)$. Consisting of $Aug_{k,p}(a)$ and the image to be accepted, image set A is as follows.

$$A = \{a\} \cup \bigcup_{k=1,p=1}^{K,P} \{Aug_{k,p}(a)\}$$

6.3 Process of Perceptual Hashing Based on CNN

The processes of the author side and the verifier side are shown in Figure 6.2.

On the author side:

(1) Data augmentation is performed on the target image and other irrelevant images to generate modified/edited images.

(2) The CNN is fine-tuned with the modified/edited images to accept the target image

and reject other irrelevant images.

(3) Data for reconstructing the trained CNN is generated for the processes of the verifier side, such as the weights and biases. Since weights and biases are the information that characterizes the target image, the hash value is generated on the basis of this information. In addition, the size of the weights and biases is about 80 MB. To meet the size of the message digest, we use a cryptographic hash function to reduce the size of the weights and biases.

(4) Data for reconstructing the trained CNN is stored and distributed by a separate database, which the proposed digital rights management system based on digital watermarking, blockchain and perceptual hashing mentioned in Chapter 3. The trained CNN model for perceptual hash value verification is shared in the distributed file system, and the CID used to retrieve the trained CNN model from the distributed file system is recorded in the blockchain. The verifier can obtain the CID to retrieve trained model for identifying the perceptual hash value.

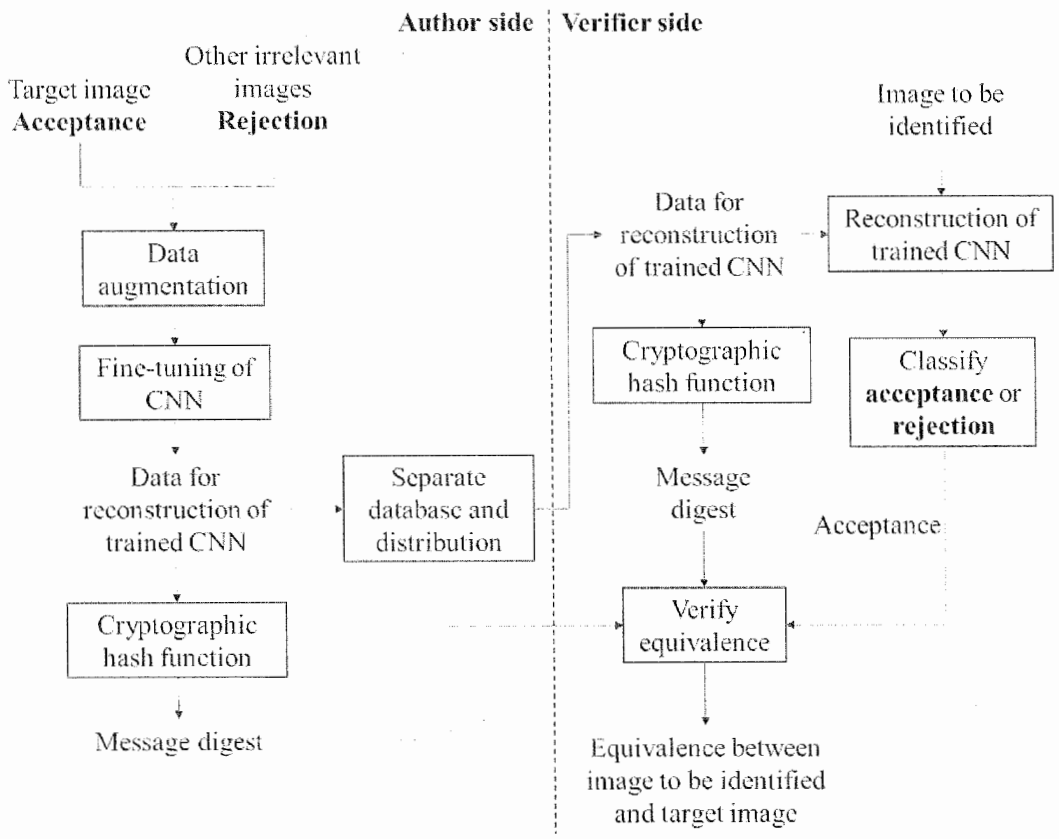

Figure 6.2: Process of perceptual hashing based on CNN.

On the verifier side:

(1) Distributed data for reconstructing the trained CNN is obtained from separate databases. In addition, the cryptographic hash function of the author side is applied

to calculate the message digest of the data for reconstructing the trained CNN. This is done to ensure that the distributed data for reconstructing the trained CNN is equivalent to the original data.

(2) The trained CNN is reconstructed.

(3) The image to be identified is classified as acceptance or rejection.

(4) To ensure equivalence between the image to be identified on the verifier side and the target image on the author side, we propose two conditions.

1) The message digest of the data for reconstructing the trained CNN on the verifier side and the message digest on the author side are equivalent.

2) The result of classifying the image to be identified is acceptance.

6.4 Simulation and Results Analysis

6.4.1 Evaluation Method of Perceptual Hashing Scheme Based on CNN

For the perceptual hashing scheme based on CNN, the performance of perceptual hashing can be evaluated on the basis of the image classification accuracy of the trained CNN. If the classification result is the target image, the result is recorded as acceptance. If the classification result is other irrelevant images, the result is recorded as rejection. The availability of the perceptual hash value depends on whether the trained CNN can distinguish the target image in terms of acceptance and rejection, and we will ensure this in experiments.

6.4.2 Fine-tuning of CNN

In our experiments, we used Keras and TensorFlow in Python [75] to fine-tune the VGG16 model.

Table 6.1: Types, methods, and parameters of image processing.

Types	Methods	Parameters
Noise	(1) White noise	Sigma between 0 to 0.5
	(2) Salt-and-pepper noise	Sigma between 0 to 0.05
Filtering	(3)Average filtering	Kernel size between 2x2 to 7x7
	(4) Median filtering	Kernel size between 3x3 to 11x11
Compression	(5) Haar wavelet transform	Sigma between 0 to 0.5
Rotation	(6) Bilinear interpolation	Direction clockwise or counterclockwise, degrees between 0 to 180
Flipping	(7) Horizontal flipping	

	(8) Vertical flipping	
Cropping	(9) Crop out part of the image	Random location, cropping size between 8x8 to 64x64
Brightness	(10) Change brightness of the image	Between 50% to 150% of original pixel value

Table 6.1 lists the types and methods of image processing used to augment the target image. Figure 6.3 shows this image in its original form and parts of its modified/edited images. We use *imgaug* [78] to modify/edit images in the experiment.

For using image processing methods different from perceptual hashing evaluation, compared with image processing methods for perceptual hashing evaluation, data augmentation needs to provide more image features for fine-tuning of CNN to improve the robustness of the trained CNN to image modification/editing. Therefore, we add image processing methods such as salt-pepper noise and average filtering to the image processing method of the perceptual hashing evaluation, so that CNN can learn the features of modified/edited images and ensure that the perceptual hashing based on CNN can be applied to digital copyright management system.

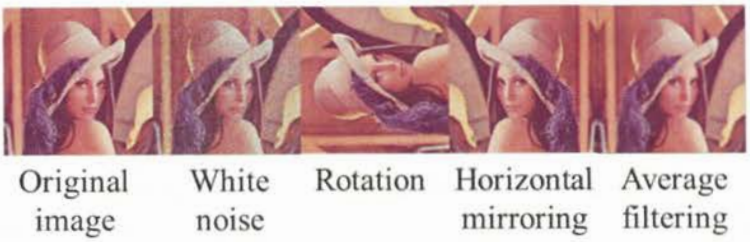

Figure 6.3: Original image and parts of modified/edited images.

Figure 6.4: Parts of other irrelevant images.

Figure 6.4 shows parts of irrelevant images not related to the target image. These images are selected from an open source project [76][77] and include large classes consisting of animals, plants, human faces, natural landscapes, buildings, and others, along with small classes consisting of flowers, trees, chickens, dogs, Europeans, Asians, campuses, monuments, rivers, mountains, and others. These images have rich diversity

so as to ensure image classification that is as accurate as possible. Since the proposed scheme mainly focuses on the image whose hash value we want to calculate, these irrelevant images do not need to be augmented, and they are included only to ensure rich diversity.

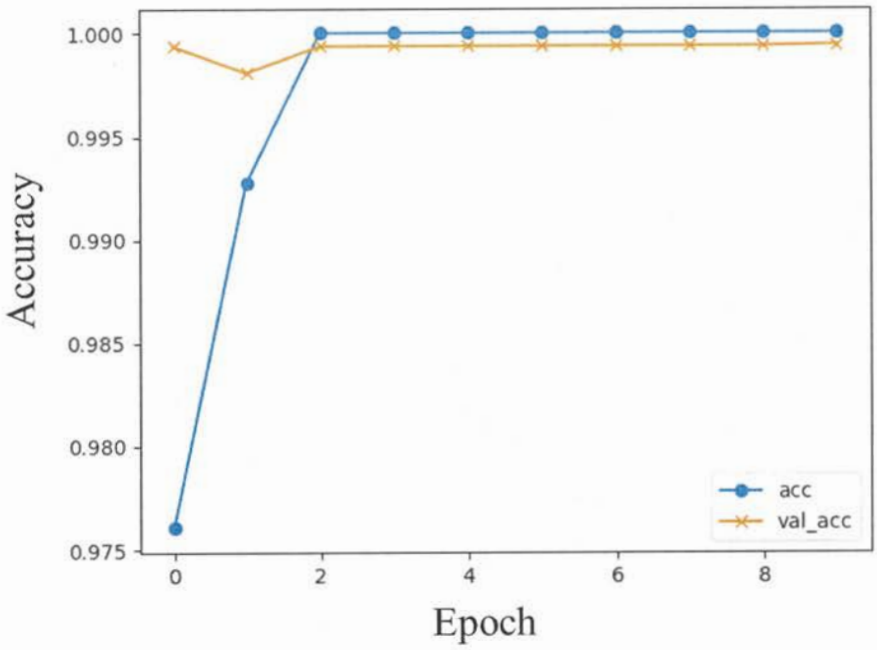

Figure 6.5: Classification accuracy of fine-tuning.

The classification accuracy of the fine-tuning based on our image sets is shown in Figure 6.5, where the horizontal axis represents the number of epochs and the vertical axis represents the classification accuracy. acc represents the correct answer rate of the training set (1,100,000 sheets). Starting from epoch 2, the correct answer rate of the training set was maintained at 1.00. val_acc represents the correct answer rate of the test set (110,000 sheets). Starting from epoch 2, the correct answer rate of the test set was maintained at 0.9994.

In CNN model training, an epoch represents the process of sending all data into the network to complete a forward calculation and back propagation [75]. When training a model, multiple epochs are generally set because when seeking the optimal solution, such as gradient descent, it is not sufficient to send data into the network just once; it usually takes multiple iterations to converge. It is difficult to determine which number of epochs is appropriate because the answer will be different for different data sets. In our simulation, the amount of data was not large and the diversity was rich, so there was no need to set too many epochs to avoid overfitting. As shown in Figure 6.5, we set the number of epochs to 10, and the network converged well before that, with a very high classification accuracy.

In addition to the target image, we performed the above experiment on more than a dozen images of plants, animals, buildings, and other types. We omit the details here, but the results were almost identical.

6.4.3 Generation of Perceptual Hash Value

After the CNN model was trained, following the manual of Keras [75], it generated two files to save the trained model, which structure of trained CNN, and weights and biases of trained CNN. The format and size of the files are listed in Table 6.2.

Table 6.2: Information for generating perceptual hash value.

Information	Format	Size
Structure of trained CNN	Json	12077 bytes
Weights and biases of trained CNN	HDF5	84605904 bytes

Although the json file can be directly clicked to view the model structure information in it, this is not possible with the HDF5 file. The full name of this format is Hierarchical Data Format, and it is a set of file formats designed to store and organize large amounts of data. HDF5 is the latest version of this format, generally written as h5. Therefore, we use the h5py library in Python to read the information in the h5 file.

According to the structural information in the json file, the trained CNN model has one input layer, 13 convolutional layers, four maximum pooling layers, one flatten layer, and three fully connected layers. The weights are distributed among the convolutional layers and the fully connected layers, and h5py is used to read the weight information of these layers. We can see from the weight information in the h5 file that the weights of each layer form a matrix composed of many float32 type numbers. A weight number occupies 32bit, which is 4bytes.

Because the weights are distributed in each layer and there are so many of them, the h5 file containing all the weights of the model should be compressed as a whole. We use the DEFLATE [73] to compress this h5 file. The generated string after file compression is recorded in a txt file for later calculation of the hash value and decoding. The size of this file is 78,424,430 bytes. Compared with the original h5 file, the file size has been reduced by more than 6 million bytes, and the weights distributed in the various layers have been compressed to facilitate calculation of the hash value later.

Since the weight information of the model has been fully compressed into this txt file, as long as we calculate the hash value of information recorded in it, it can be used as the perceptual hash value of the acceptance class images in our CNN model. We use the SHA256 [27] to calculate the hash value of information recorded in this txt file. This hash value is used as the perceptual hash value of the target image and its subsequently modified/edited images.

6.4.4 Verification of Perceptual Hash Value

We use the cryptographic hash function to calculate the hash value of the

compressed weights retrieved from the digital rights management system. We use the SHA256 to calculate the hash value. This hash value is the same as the hash value of the original compressed weights and biases, that is, the retrieved compressed weights and biases are the same as the original compressed weights and biases.

And, we decode the compressed weights mentioned above. We use the DEFLATE to decode the txt file containing the compressed weight information. The decoded file is still in h5 format and the size is 84,605,904 bytes, which is the same size as the original weight file. This shows that in the compression and decoding processes, the weight information is not lost, which is a reversible compression process.

Then, we reconstruct the trained CNN model mentioned in Section 6.4.2. We use the json file of the model structure and the h5 file of the decoded model weights and biases to reconstruct the trained CNN model to classify images.

In Section 6.4.1, we explained the performance of the evaluation method for the proposed perceptual hashing scheme. Specifically, we needed to ensure that the trained CNN can correctly identify images.

The images used for identification were divided into two parts.

(1) Data augmentation was performed on the target image by changing the parameters in Table 6.1 to generate the modified/edited images. These images did not overlap with the images used for fine-tuning. If the images were the same as the images used for fine-tuning, the verification experiment would be meaningless. Therefore, we changed the parameters to generate images used for identification.

The same as (1), data augmentation was performed on the other irrelevant images.

Table 6.3: Classification results and number of images.

Classification result	Number of images
Correctly identified target images	1,000,000
Incorrectly identified target images	0
Correctly identified other irrelevant images	999,998
Incorrectly identified other irrelevant images	2

With the above method, we generated 1 million target images used for identification and 1 million other irrelevant images used for identification. There were four types of classification results. In the first, target images were correctly identified. In the second, they were incorrectly identified. In the third, the other irrelevant images were correctly identified. For the fourth, they were incorrectly identified. The numbers of correctly identified images and incorrectly identified images were used to illustrate the performance of the proposed perceptual hashing scheme.

As shown in Table 6.3, the number of correctly identified target images was 1,000,000. The number of incorrectly identified ones was 0. The number of correctly identified other irrelevant images was 999,998. The number of incorrectly identified ones was 2. According to the above results, for the 1 million target images used for the

experiment, all images were correctly identified. For the 1 million other irrelevant images used for the experiment, most of the images were correctly identified, and only 2 images were incorrectly identified.

In Section 6.1, for the perceptual hashing scheme based on CNN, we redefined the requirements for perceptual hashing mentioned in Chapter 4. Specifically, the trained CNN was used to accept target images and reject the other irrelevant images. According to the classification results in Table 6.3, the proposed perceptual hashing scheme can meet the requirements.

For other irrelevant images that have been incorrectly identified, correct identification of target images is an important requirement for the proposed perceptual hashing scheme. In addition, it is difficult to generate a practical image that has the same perceptual hash value as target images. Therefore, the proposed perceptual hashing scheme has no problem in terms of practical application.

Chapter 7

Application for Image Groups Based on Perceptual Hashing Using Weights and Biases

7.1 Methods of Generating Identical Hash Value for Each Image in Group

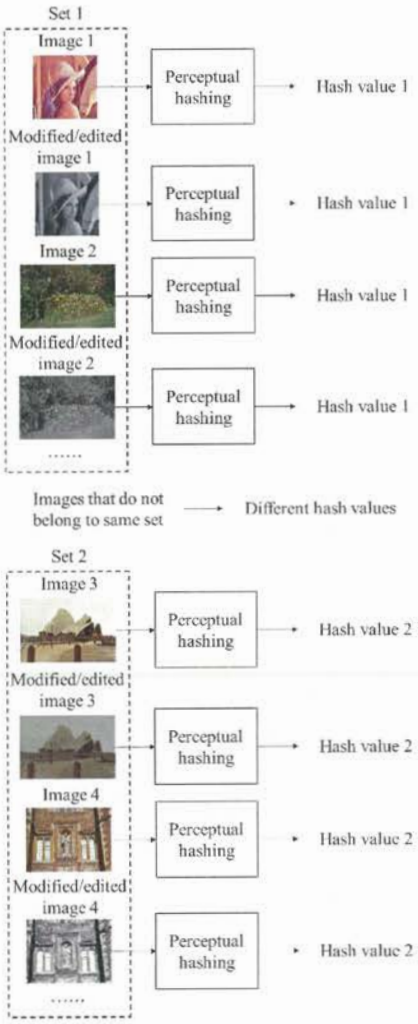

Figure 7.1: Generation of identical hash value for each image in the group.

In practical applications, multiple different images are published in different forms of media, such as in a paper, in a book, or in an image collection. As shown in Figure 7.1, image 1, modified/edited image 1, image 2, modified/edited image 2, and other images belong to set 1. Hash value 1 is generated for each image in set 1. Image 3, modified/edited image 3, image 4, modified/edited image 4, and other images belong to set 2, not set 1. Hash value 2 is generated for each image in set 2, which is different from hash value 1 of each image in set 1. If an identical hash value for each of these image is generated all at once, the management of the content will become easy, and the amount of calculation for fine-tuning will be reduced. In addition, each image in the group can be identified by using the identical hash value. Therefore, we need a perceptual hashing scheme that generates an identical hash value for each image in a group.

For the perceptual hashing scheme applied to image groups, the definition of a set is as follows.

- K types of data augmentation are performed on N images included in image group G, such as white noise, median filtering, and rotation. In data augmentation, the kth modification/editing operation is performed with parameter p in P types to generate images $Aug_{k,p}(a)$. Consisting of $Aug_{k,p}(a)$ and the image to be accepted, image set A is as follows.

$$A = \bigcup_{n=1}^{N}(\{a_n\} \cup \bigcup_{k=1,p=1}^{K,P} \{Aug_{k,p}(a_n)\})$$

To generate an identical perceptual hash value for all images in a group, we need to apply fine-tuning to the images in the group to generate the trained CNN. The identical hash value is generated on the basis of parameters such as weights and biases in the process of fine-tuning. We propose two schemes for fine-tuning as shown in Figure 7.2.

(1) $1+1$ *classes* scheme: There are n images in a group, and there are other irrelevant images. The solid line represents the image used for fine-tuning and the corresponding output class. The images in the group have the identical output class of the image group, and the output class of the other irrelevant images is the class of the other irrelevant images. The image group is regarded as acceptance, and the other irrelevant images are regarded as rejection. With this method, there will be two output classes, that is, the class of the image group and the class of the other irrelevant images. This scheme will be called $1+1$ *classes* later. The $1+1$ *classes* scheme can simply fine-tune a CNN.

(2) n+1 classes scheme: There are n images in a group, and there are other irrelevant images. The solid line represents the image used for fine-tuning and the corresponding output class. The output class of image i(i=1,2,⋯,n) in the group is the class of image i. The output class of the other irrelevant images is the class of the other irrelevant images. Image 1, image 2, and until image n are regarded as

acceptance, and the other irrelevant images are regarded as rejection. With this method, there will be multiple output classes, that is, the class of image n and the class of the other irrelevant images. This scheme will be called n+1 classes later. The n+1 classes scheme can identify which image in a group. We will compare the 1+1 classes scheme and n+1 classes scheme in terms of the amount of calculation for fine-tuning the CNN and the classification accuracy of fine-tuning.

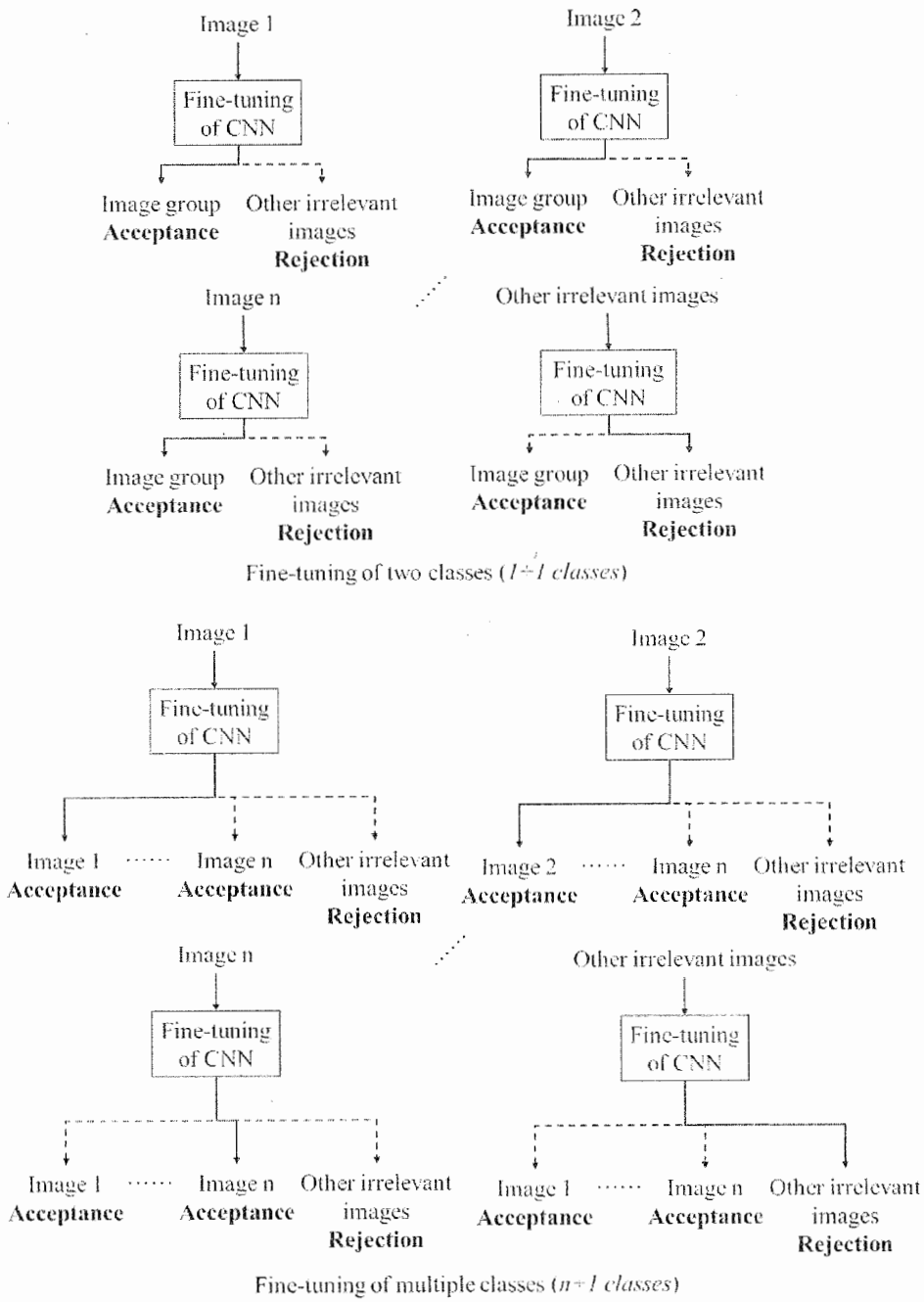

Figure 7.2: Fine-tuning of two classes and fine-tuning of multiple classes.

7.2 Process of Generating Perceptual Hash Value

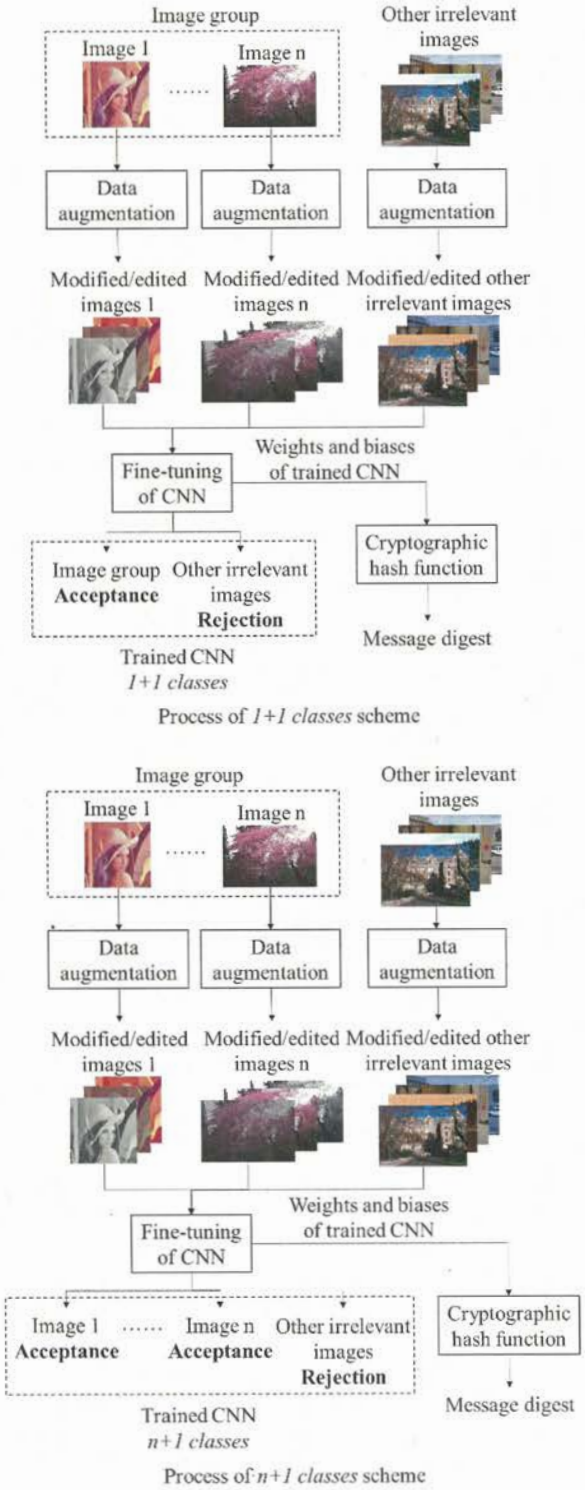

Figure 7.3: Process of 1+1 classes scheme and n+1 classes scheme.

The process of the $1+1$ *classes* scheme and the process of the $n+1$ *classes* scheme are shown in Figure 7.3.

(1) Data augmentation is applied to images in a group and other irrelevant images. Specifically, image processing methods such as white noise, filtering, and compression are applied to modify/edit the images in the group and other irrelevant images.

(2) Modified/edited images are used for fine-tuning to generate a trained CNN. For the $1+1$ *classes* scheme, there are two classes in the trained CNN. The image group is regarded as acceptance. For the $n+1$ *classes* scheme, there are multiple classes in the trained CNN. Image 1, image 2, and until image n are regarded as acceptance. Other irrelevant images are regarded as rejection for the two schemes.

A message digest of the weights and biases of the trained CNN is generated by using a cryptographic hash function that is an identical perceptual hash value for all images in the group. The size of the weights and biases of the trained CNN is about 80 MB. To meet the size of the message digest, we use the cryptographic hash function to reduce the size of the weights and biases.

7.3 Process of Verifying Perceptual Hash Value

The method of the verifier side in Section 6.3 is used for the proposed perceptual hashing scheme.

(1) Distributed weights and biases of the trained CNN are obtained from a database. A cryptographic hash function is applied to calculate the message digest of the weights and biases of the trained CNN. This is done to ensure that the distributed weights and biases of the trained CNN are equivalent to the original weights and biases.

(2) The trained CNN is reconstructed.

(3) The image to be identified is classified as acceptance or rejection.

(4) To ensure equivalence between the image to be identified and images in the group, we propose two conditions.

   1) The message digest of the distributed weights and biases of the trained CNN is equivalent to the message digest of the original weights and biases of the trained CNN.

   2) The result of classifying the image to be identified is acceptance.

7.4 Simulation and Results Analysis

7.4.1 Evaluation Method of Perceptual Hashing Scheme for Image Groups

For the perceptual hashing scheme for image groups, the performance of perceptual hashing can be evaluated on the basis of the image classification accuracy of the trained

CNN. If the classification result is an image in the group, the result is recorded as acceptance. If the classification result is other irrelevant images, the result is recorded as rejection. The availability of the perceptual hash value depends on whether the trained CNN can distinguish the target image in terms of acceptance and rejection, and we will ensure this in experiments. To ensure that the trained CNN has sufficient classification accuracy to perform distinguishing experiments, we will perform experiments on 1+1 classes and n+1 classes for fine-tuning. In addition, to verify the impact on the classification accuracy of fine-tuning, we will perform experiments on the ratio between other irrelevant images and the images in a group. We will explain the ratio later.

7.4.2 Data Augmentation of Images

In our experiments, we use Keras and TensorFlow in Python [75] to generate a perceptual hash value based on CNN. In the experiment, images are selected from open source datasets [76] and [77]. The datasets include classes consisting of animals, plants, human faces, natural landscapes, buildings, and others. These images have rich diversity so as to ensure the accuracy of our experiments. Table 7.1 lists the types, methods, and parameters of the image processing used for data augmentation. There are 10 image-processing methods for augmentation. When augmentation is performed, each image-processing method generates the same number of modified/edited images. We use *imgaug* [78] to modify/edit images in the experiment. *imgaug* [78] is a library for data augmentation in machine learning experiments. The generated images will be used for fine-tuning experiments and perceptual hash value verification experiments.

For using image processing methods different from perceptual hashing evaluation, compared with image processing methods for perceptual hashing evaluation, data augmentation needs to provide more image features for fine-tuning of CNN to improve the robustness of the trained CNN to image modification/editing. Therefore, we add image processing methods such as salt-pepper noise and average filtering to the image processing method of the perceptual hashing evaluation, so that CNN can learn the features of modified/edited images and ensure that the perceptual hashing based on CNN can be applied to digital copyright management system.

Table 7.1: Types, methods, and parameters of image processing.

Types	Methods	Parameters
Noise	(1) White noise	Sigma between 0 to 0.5
	(2) Salt-and-pepper noise	Sigma between 0 to 0.05
Filtering	(3)Average filtering	Kernel size between 2x2 to 7x7
	(4) Median filtering	Kernel size between 3x3 to 11x11
Compression	(5) Haar wavelet transform	Sigma between 0 to 0.5

Rotation	(6) Bilinear interpolation	Direction clockwise or counterclockwise, degrees between 0 to 180
Flipping	(7) Horizontal flipping (8) Vertical flipping	
Cropping	(9) Crop out part of the image	Random location, cropping size between 8x8 to 64x64
Brightness	(10) Change brightness of the image	Between 50% to 150% of original pixel value

7.4.3 Comparison of $1+1$ *classes* and $n+1$ *classes*

We have two purposes for performing experiments on fine tuning.

(1) Fine-tuning with $1+1$ *classes* and $n+1$ *classes*: In Section 7.1, we proposed two schemes for fine-tuning, $1+1$ *classes* and $n+1$ *classes*. We will compare $1+1$ *classes* and $n+1$ *classes* in terms of the calculation time for CNN fine-tuning and the classification accuracy for fine-tuning.

(2) Ratio between other irrelevant images and images of group: Ratio refers to the value obtained by dividing the other irrelevant images by images in the group. For example, if there are 100 other irrelevant images and 5 images in a group, the ratio will be 20. In research [12], we tested different ratios between other irrelevant images and the target image. After examining the required time and classification accuracy for fine-tuning, we set the ratio between other irrelevant images and the target image to 100. However, in the current work, a group generally contains multiple images. To ensure the classification accuracy of fine-tuning, an appropriate ratio between other irrelevant images and images in a group needs to be selected.

The experimental steps were as follows.

(1) Ten types of image groups were experimented with. Each group contained different numbers of images, and the number of images was incremented from 10 to 100 in steps of 10. The ratio was incremented from 1 to 20 in steps of 1.
(2) Original images were selected from the open source datasets [76] and [77]. These images were divided into images of the group and other irrelevant images.
(3) Data augmentation was performed on the images in the group and other irrelevant images. There were 10 image-processing methods for data augmentation as shown in Table 7.1. For each original image, each method generated 1,000 modified/edited images.
(4) Fine-tuning was performed on the images in the group and other irrelevant images by using the $1+1$ *classes* scheme and $n+1$ *classes* scheme.

For the 10 types of image groups, we got almost the same experimental results. Since the experimental results of these 10 types of groups cannot express uniqueness, we will explain the experimental results for a group containing 50 images as a representative

value.

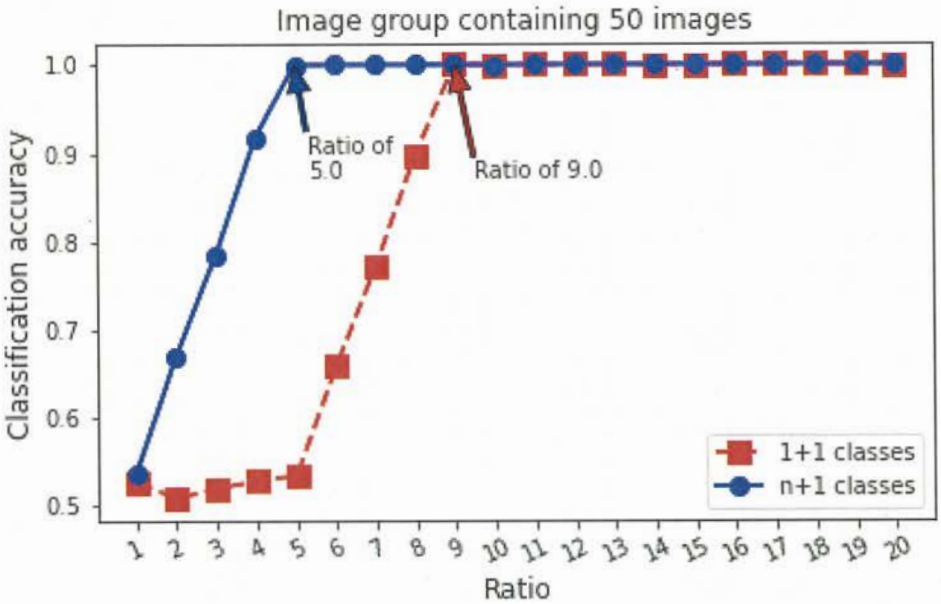

Figure 7.4: Classification accuracy of 1+1 classes and n+1 classes with different ratios.

The experimental results for the group containing 50 images are shown in Figure 7.3. The horizontal axis is the ratio, and the vertical axis is the classification accuracy of fine-tuning. The blue solid lines and dots are the classification accuracy of $n+1$ *classes* for different ratios, and the red dashed lines and squares are the classification accuracy of $1+1$ *classes* for different ratios. As the ratio increased, the classification accuracy of $n+1$ *classes* and that of $1+1$ *classes* increased.

According to the experimental results, the ratio will affect the classification accuracy of fine-tuning. For $1+1$ *classes* and $n+1$ *classes*, the appropriate ratio was different. For $1+1$ *classes*, a ratio of 9 or higher should be used when performing fine-tuning. For $n+1$ *classes*, a ratio of 5 or higher should be used when performing fine-tuning. As a result, $n+1$ *classes* had a lower calculation amount for fine-tuning than did $1+1$ *classes*. In addition, when verifying the perceptual hash value, $n+1$ *classes* can identify which image in a group. Therefore, the $n+1$ *classes* scheme should be used for fine-tuning.

7.4.4 Verification of Perceptual Hash Value

In Section 7.4.1, we explained the performance of the evaluation method for the proposed perceptual hashing scheme. Specifically, we needed to ensure that the trained CNN can correctly identify images. The experimental steps were roughly the same as those in Section 7.4.3. The difference is that the ratio was fixed at 10, and only the $n+1$ *classes* scheme was used for fine-tuning.

The images used for identification were divided into two parts.

(1) Data augmentation was performed on images in the group by changing the parameters in Table 7.1 to generate 1,000 modified/edited images. These images did not overlap with the images used for fine-tuning. If the images were the same as the images used for fine-tuning, the verification experiment would be meaningless. Therefore, we changed the parameters to generate images used for identification.

(2) The same as (1), data augmentation was performed on the other irrelevant images.

With the above method, we generated 55 million images in the group used for identification and 55 million other irrelevant images used for identification. There were four types of classification results. In the first, images in the group were correctly identified. In the second, they were incorrectly identified. In the third, the other irrelevant images were correctly identified. For the fourth, they were incorrectly identified. The numbers of correctly identified images and incorrectly identified images were used to illustrate the performance of the proposed perceptual hashing scheme.

As shown in Table 7.2, the number of correctly identified images in the group was 55,000,000. The number of incorrectly identified ones was 0. The number of correctly identified other irrelevant images was 54,999,931. The number of incorrectly identified ones was 69. According to the above results, for the 55 million images in the group used for the experiment, all images were correctly identified. For the 55 million other irrelevant images used for the experiment, most of the images were correctly identified, and only 69 images were incorrectly identified.

Table 7.2: Classification results and number of images.

Classification result	Number of images
Correctly identified images in group	55,000,000
Incorrectly identified images in group	0
Correctly identified other irrelevant images	54,999,931
Incorrectly identified other irrelevant images	69

In Section 7.1, for the perceptual hashing scheme based on CNN used for image groups, we redefined the requirements for perceptual hashing mentioned in Chapter 4. Specifically, the trained CNN was used to accept images in the group and reject the other irrelevant images. According to the classification results in Table 7.2, the proposed perceptual hashing scheme can meet the requirements.

For other irrelevant images that have been incorrectly identified, correct identification of images in a group is an important requirement for the proposed perceptual hashing scheme. In addition, it is difficult to generate a practical image that has the same perceptual hash value as images in a group. Therefore, the proposed perceptual hashing scheme has no problem in terms of practical application.

Chapter 8

Perceptual Hashing Using Probability Variable of Output of General CNN Applied for Image Classification

8.1 Concept of Perceptual Hashing Based on CNN Output

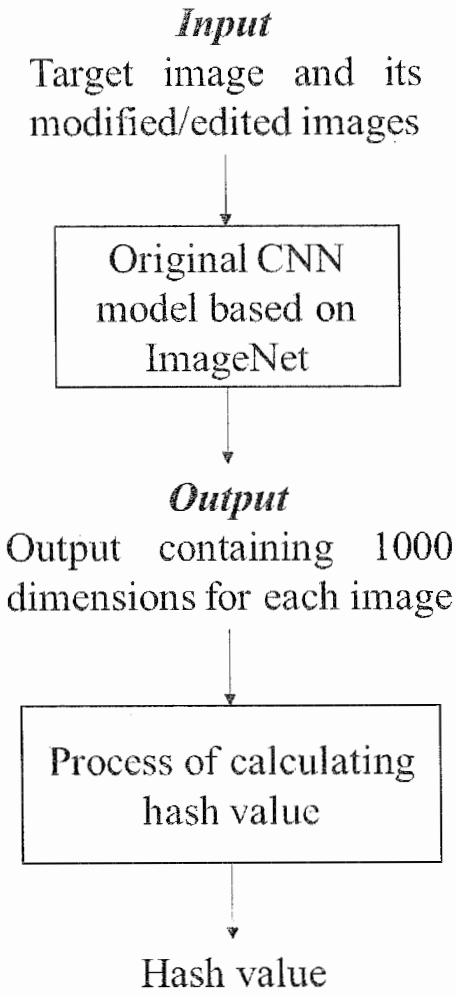

Figure 8.1: Scheme of perceptual hashing based on output of CNN.

Figure 8.1 shows the proposed scheme of perceptual hashing based on CNN output that does not require fine-tuning. The input of the original CNN based on ImageNet is the image to be hashed and its consequent images that have been modified/edited with image augmentation. After inputting these images to the CNN, we obtain the output of the CNN for each image. This output refers to the probability of 1000 categories as dimensions of VGG16 trained on ImageNet. We generate the perceptual hash value of the image on the basis of this output.

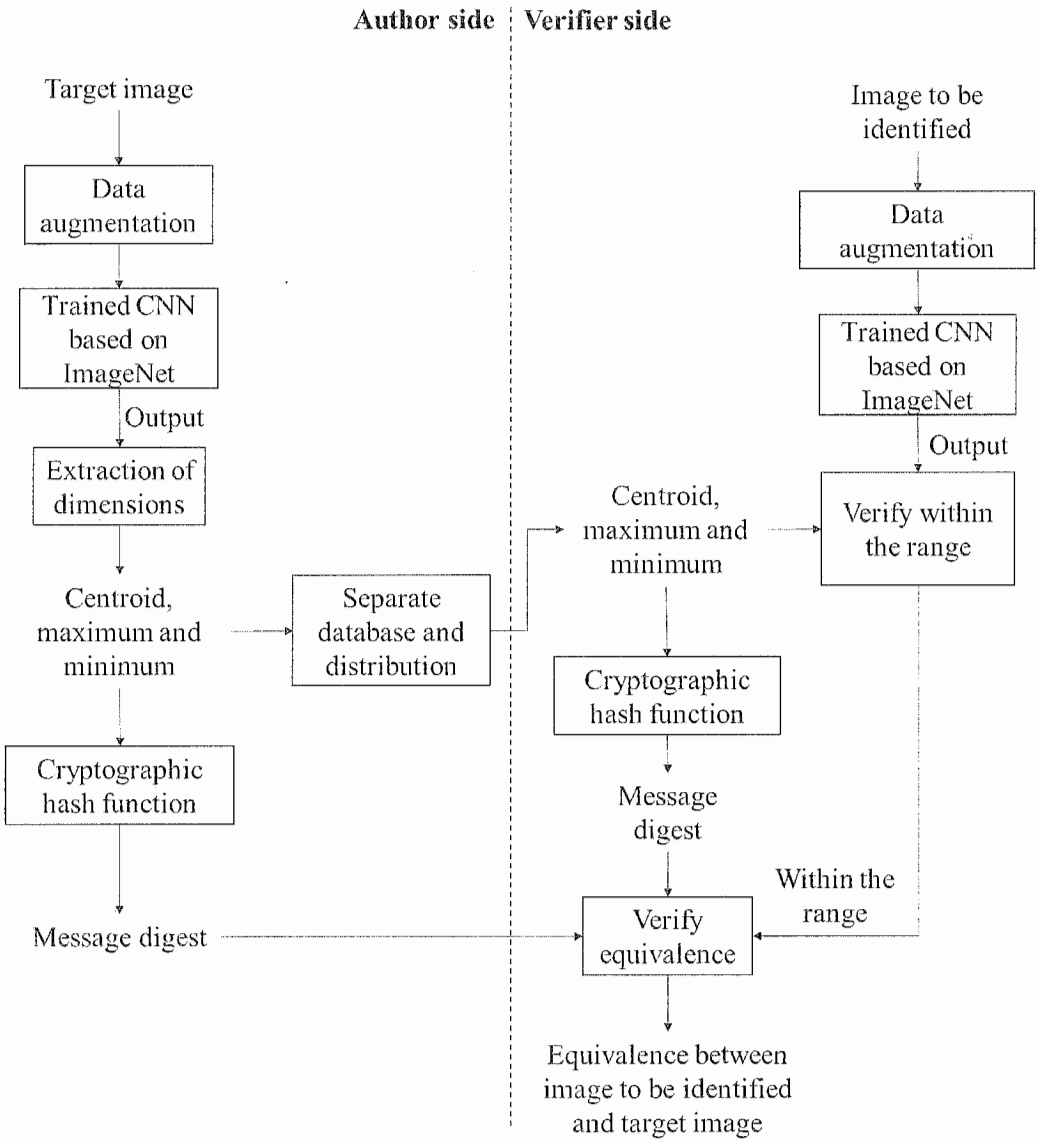

Figure 8.2: Schematic diagram of perceptual hashing scheme based on output of CNN.

The process of the proposed scheme are shown in Figure 8.2. We first give a brief overview and then explain the specific content in the following subsections.

On the author side:

(1) Data augmentation is performed on the target image.

(2) The modified/edited images are input into the CNN trained with ImageNet to obtain the response of the output layer of the trained CNN, hereafter referred to as the output.

(3) The centroid, the maximum, and the minimum are extracted from the output containing 1000 dimensions for the target image and its modified/edited images. The centroid is the output with the shortest distance from other outputs containing 1000 dimensions. The maximum and the minimum are the maximum and minimum values of each dimension of the output containing 1000 dimensions. Since the centroid, the maximum, and the minimum are the information that characterizes the target image, the hash value is generated on the basis of this information. In addition, to meet the size of the message digest, we use a cryptographic hash function to reduce the size of the centroid, the maximum, and the minimum.

(4) The centroid, the maximum, and the minimum are stored and distributed by a separate database, which the proposed digital rights management system based on digital watermarking, blockchain and perceptual hashing mentioned in Chapter 3. The centroid, the maximum, and the minimum for perceptual hash value verification is shared in the distributed file system, and the CID used to retrieve The centroid, the maximum, and the minimum from the distributed file system is recorded in the blockchain. The verifier can obtain the CID to retrieve the centroid, the maximum, and the minimum for identifying the perceptual hash value.

On the verifier side:

(1) Data augmentation is performed on the image to be identified using the same processing methods as the author side.

(2) The modified/edited images are input to the same CNN using the same method as the author side to obtain the output.

(3) The distributed centroid, maximum, and minimum of the author from separate databases. In addition, the cryptographic hash function of the author side is applied to calculate the message digest of the centroid, maximum, and minimum. This is done to ensure that the distributed data is equivalent to the original data.

(4) To ensure equivalence between the image to be identified on the verifier side and the target image on the author side, we propose two conditions.

   1) The message digest of the centroid, maximum, and minimum on the verifier side and the message digest on the author side are equivalent.

   2) The verifier's output is within the range of the maximum and minimum of the author's output.

8.2 Generation of Modified/Edited Images

In the proposed scheme, "data augmentation" refers to the use of image processing

methods to generate modified/edited images similar to the original image by ourselves. It does not refer to the generation of a data set for fine-tuning.

We take the modified/edited image and the original image as a set, and calculate the perceptual hash value of this set. This is to improve the robustness of the perceptual hash algorithm for image processing, such as rotation and flipping. Even if the image is modified/edited, it can calculate the same perceptual hash value as the original image, thereby proving that the equivalence between the modified/edited image and the original image.

On the verifier side, we select the same augmentation using the same processing methods as the author side to control any variables that may affect the verification accuracy in the experiment.

8.3 Process of Generation Perceptual Hash Value

In the proposed scheme, we use the CNN model named VGG16 [55][56] trained with ImageNet [70] to obtain the output. ImageNet [71] contains a large number of image features. Therefore, VGG16 can be used to classify a variety of image structures. Perceptual hashing calculates the perceptual hash value based on the appearance of image that matches the human vision. If the appearance of images is the same, the perceptual hash values should also be the same. In contrast, if the appearances of images are different, the perceptual hash values will be different. For the CNN, it will obtain the same output for the same input image, and will obtain different outputs for different input images. In other words, there is a one-to-one correspondence between the input image and the output. Therefore, we use the CNN output to calculate the perceptual hash value of the target image.

For the VGG16, as shown in Figure 8.3, the input of any image will result in an output containing the probability of 1000 categories as dimensions.

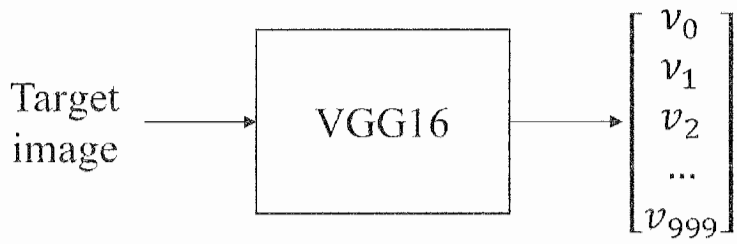

Figure 8.3: Output of VGG16 for any target image.

We obtained a lot of modified/edited images by augmentation and input these images to VGG16, so the output of VGG16 also contains a lot of dimensions. In practical applications, it is very inconvenient to directly use outputs that contain a large number of dimensions. Therefore, we extract some dimensions from the output of VGG16 for generating and verifying the perceptual hash value.

(1) Centroid of output: We extract the centroid from the output as representative

dimensions of many augmented images. Calculation steps are as follows.

1) Calculate the sum of the squares of the distance between each dimension of the output and other dimensions of the output.
2) Compare the size of results of step 1).
3) Select a dimension from results of step b) with the smallest sum of the squares of the distance with other dimensions as the centroid of the output.

(2) Maximum and minimum of output: We extract the maximum and the minimum from the output to verify the equivalency of two images, and the specific usage method is explained in the next section. Calculation steps are as follows.

1) Compare the size of each dimension of the output.
2) Select the maximum value of each dimension from results of step 1) as the maximum of the output.
3) Select the minimum value of each dimension from results of step 1) as the minimum of the output.

We save the outputs of VGG16 containing the probability of 1000 categories as dimensions for each image, the centroid, the maximum, and the minimum to a file in npz format by NumPy in Python [74]. This is done to put all the outputs for the image we need to calculate the perceptual hash value for and the modified/edited images into the same file for calculation of the centroid, the maximum, and the minimum.

In the proposed scheme, we use SHA256 [27] to calculate the cryptographic hash value of the centroid, the maximum, and the minimum of the output as the perceptual hash value of the image. This cryptographic hash function is very sensitive to the change of each bit of the input data, and the output hash value has 256 bits, which meets the requirements for our experiment.

8.4 Process of Verifying Perceptual Hash Value

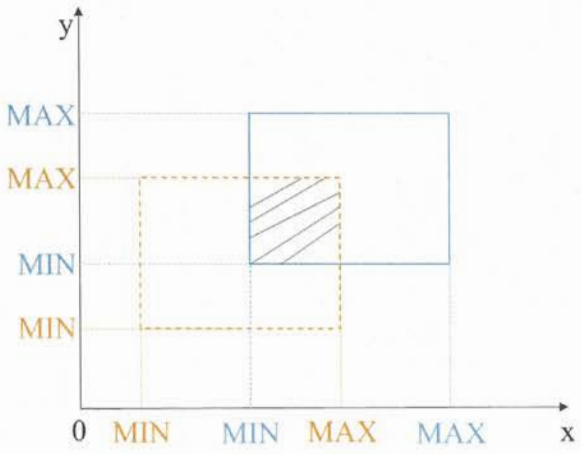

Figure 8.4: Range of output of the image to be identified and the target image.

In our scheme, we propose a new verification method for the equivalence of two images that use the maximum and the minimum of the output.

As shown in Figure 8.4, although it is shown in two dimensions, the actual situation is 1000 dimensions. In the figure, MIN and MAX of the solid blue line are the maximum and minimum of the output of the image we need to calculate the perceptual hash value for, and MAX and MIN of the orange dotted line are the maximum and minimum of the output of the image to be verified. Therefore, the rectangle composed of blue solid lines and the rectangle composed of orange dotted lines represent the range of the output of the two images.

And, the shaded part in the figure represents where the number of dimensions that the output of the image to be verified falls in the output of the image we need to calculate the perceptual hash value for. If the number of dimensions where the output of the image to be verified falls in the output of the image we need to calculate the perceptual hash value for is less than the set threshold, it will prove that the image to be verified is inconsistent with the image we need to calculate the perceptual hash value for. We describe the settings of our method in Section 8.5.

8.5 Simulation and Results Analysis

8.5.1 Data Augmentation of Images

In our experiments, we use Keras and TensorFlow in Python [75] to calculate the perceptual hash value based on CNN.

Figure 8.5 shows some examples of the test images we used. These images are selected from an open source project [76][77] and include large classes consisting of animals, plants, human faces, natural landscapes, buildings, and others, along with small classes consisting of flowers, trees, chickens, dogs, Europeans, Asians, campuses, monuments, rivers, mountains, and others. These images have rich diversity so as to ensure the accuracy of our experiments.

Figure 8.5: Parts of test images.

Table 8.1 lists the types, methods, and parameters of the image processing used to augment images. We use *imgaug* [78] to modify/edit images in the experiment.

Table 8.1: Types, methods and parameters of image processing.

Types	Methods	Parameters
Noise	(1) White noise (2) Salt-and-pepper noise	Sigma between 0 to 0.5 Sigma between 0 to 0.05
Filtering	(3)Average filtering (4) Median filtering	Kernel size between 2x2 to 7x7 Kernel size between 3x3 to 11x11
Compression	(5) Haar wavelet transform	Sigma between 0 to 0.5
Rotation	(6) Bilinear interpolation	Direction clockwise or counterclockwise, degrees between 0 to 180
Flipping	(7) Horizontal flipping (8) Vertical flipping	
Cropping	(9) Crop out part of the image	Random location, cropping size between 8x8 to 64x64
Brightness	(10) Change brightness of the image	Between 50% to 150% of original pixel value

Figure 8.6 shows this image in its original form and parts of its modified/edited images. In the experiment, for each image, we randomly generated 1000 modified/edited images using the image processing methods in the table to ensure the accuracy of the experiment.

Original image | Salt-and-pepper noise | Horizontal flipping | Compression | Median filtering

Figure 8.6: Examples of modified/edited images.

For using image processing methods different from perceptual hashing evaluation, compared with image processing methods for perceptual hashing evaluation, data augmentation needs to provide more image features for fine-tuning of CNN to improve the robustness of the trained CNN to image modification/editing. Therefore, we add image processing methods such as salt-pepper noise and average filtering to the image processing method of the perceptual hashing evaluation, so that CNN can learn the features of modified/edited images and ensure that the perceptual hashing based on

CNN can be applied to digital copyright management system.

8.5.2 Setting of Threshold

In order to verify the equivalence between the image to be identified and the target image, we count the number of matching dimensions in the output of the image to be identified and the target image, which the probability of 1000 categories as dimensions of VGG16 trained on ImageNet. We also set a threshold for the number of matching dimensions to verify the equivalence between the image to be identified and the target image. When the number of matching dimensions is greater than or equal to the threshold, the appearance of the image to be identified is the same as the target image. When the number of matching dimensions is less than the threshold, the appearances of the image to be identified and the target image are different.

We use 1000 images in the verification experiment, which are the same 1000 we used for generating the perceptual hash value.

In the initial experiment, we found that the counted number of dimensions falling in the range of maximum and minimum was too much, so instead we counted the number of dimensions falling outside the range of maximum and minimum, that is, the number of dimensions of matching errors.

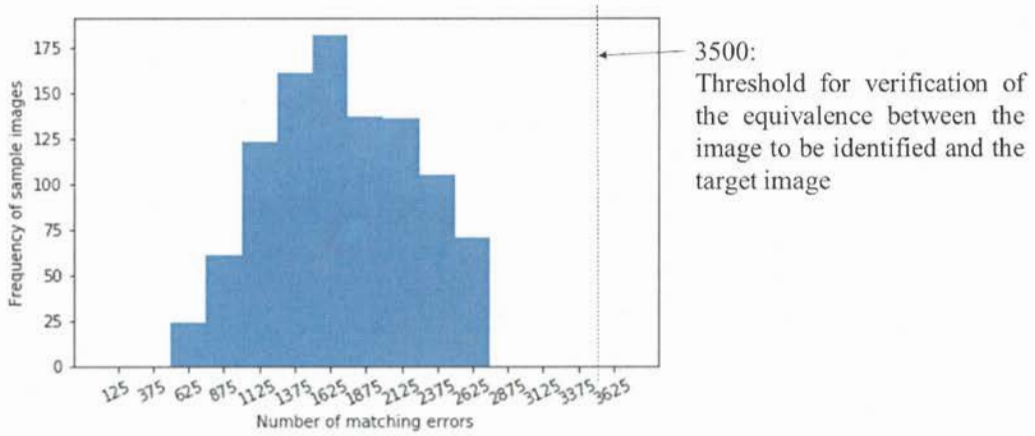

Figure 8.7: Distribution of matching errors between image to be identified and image of same appearance in target images.

Figure 8.7 shows the number of matching errors between the output of the image to be verified and the output of the image that has the same appearance as the image to be verified in the images we need to calculate the perceptual hash value for, which is the probability of 1000 categories as dimensions of VGG16 trained on ImageNet, and the frequency of sample images for each number of matching errors. Horizontal axis represents the number of dimensions of the image to be verified that fall outside the maximum and minimum range of the image we need to calculate the perceptual hash

value for. The range of the horizontal axis is 0-3649, divided into 13 ranges, and the scale of each range is 250. The scale value in the figure is the middle value of the range. Vertical axis represents the number of images we need to calculate the perceptual hash value for in each range on the horizontal axis. The values of the vertical axis of the range to the right of 2500-2749 are all 0, so the statistics of the number of matching errors are up to the range of 2500-2749.

Figure 8.8 shows the number of matching errors between the output of the image to be verified and the output of the image that has a different appearance to the image to be verified in the images we need to calculate the perceptual hash value for, which is the probability of 1000 categories as dimensions of VGG16 trained on ImageNet, and the frequency of sample images for each number of matching errors. Horizontal axis represents the number of dimensions of the image to be verified that fall outside the maximum and minimum range of the image we need to calculate the perceptual hash value for. The range of the horizontal axis is 0-99999, divided into 100 ranges, and the scale of each range is 2000. The scale value in the figure is the middle value of the range. Although there are still experimental results after 99999, the sample images on the vertical axis from the peak in the figure gradually decrease, and the number of experimental results after 99999 is very small, so the part after 99999 is omitted. Also, because the scale of the horizontal axis is too dense, we display it every five scales. Vertical axis represents the number of images we need to calculate the perceptual hash value for in each range on the horizontal axis. The values on the vertical axis of the range to the left of 4000-5999 are all 0, so the statistics of the number of matching errors start from the range of 4000-5999.

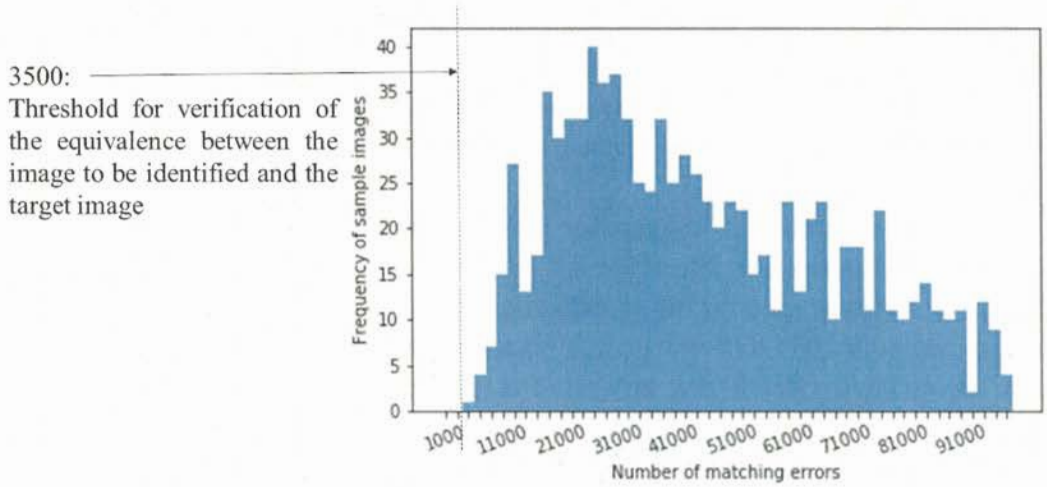

Figure 8.8: Distribution of matching errors between image to be identified and image of difference appearance in target images.

According to the ending range of 2500-2749 of the number of matching errors in

Figure 8.7, and the starting range of 4000-5999 of the number of matching errors in Figure 8.8, the distributions of Figures 8.7 and 8.8 do not cover each other. We select the middle value 3500 from the two ranges, which 2500-2749 and 4000-5999, as the threshold for the number of matching errors to verify the equivalency between the image to be verified and the image to be calculated. As shown in Figure 8.7 and Figure 8.8, the value of the horizontal axis indicated by the black dashed line is 3500. If the number of dimensions of the output of the image to be verified that fall outside the range composed of the maximum and the minimum of the output of the image we need to calculate the hash value for is less than or equal to 3500, it proves that the image to be verified and the image to be calculated is the same. In contrast, if the number of dimensions is larger than 3500, it proves that the images are different.

8.5.3 Experimental Results of Verifying Perceptual Hash Value

The images used for identification is that, data augmentation was performed on the images by changing the parameters in Table 8.1 to generate the modified/edited images. These images did not overlap with the images used for generating perceptual hash value. If the images were the same as the images used for generating the perceptual hash value, the verification experiment would be meaningless. Therefore, we changed the parameters to generate images used for identification.

With the above method, we generated 1 million target images used for identification. There were two types of classification results. In the first, images were correctly identified. In the second, they were incorrectly identified. The numbers of correctly identified images and incorrectly identified images were used to illustrate the performance of the proposed perceptual hashing scheme.

Table 8.2: Classification results and number of images.

Classification result	Number of images
Correctly identified images	1,000,000
Incorrectly identified images	0

As shown in Table 8.2, the number of correctly identified images was 1,000,000. The number of incorrectly identified ones was 0. According to the above results, for the 1 million images used for the experiment, all images were correctly identified. And, we experimented 1000 different original images and their modified/edited images, but there are no hash collision that generate the identical hash value for different images.

8.6 Comparison of CNN-based Perceptual Hashing Schemes

As shown in Table 8.3, there are some CNN-based perceptual hashing schemes. Among these schemes, the (3), (4), and (5) schemes are our proposed research schemes.

The (3) scheme is proposed in Chapter 5, the (4) scheme is proposed in Chapter 6, and the (5) scheme is proposed in Chapter 8. The (1), (2), and (6) schemes are related works.

Table 8.3: Comparison of CNN-based Perceptual Hashing Schemes.

Authors	Object	Purpose	Training for target image	Information used for generating hash value	Average verification time (s)	Average verification accuracy	Publish date
(1) C. Jiang et al. [57]	Image	Content authentication	Need	Response on fully connected layer of CNN	3.5630	0.9682	Aug. 2018
(2) H. Wu et al. [58]	Video	Tamper detection	Need	Parameters of CNN training process	2.7940	0.9839	June 2019
(3) Z. Meng et al. [8]	Image	Content authentication	Need	Proposal of concept and model of CNN-based perceptual hashing			July 2019
(4) Z. Meng et al. [12]	Image	Content authentication	Need	Model of trained CNN (e.g. weights)	0.0156	0.9999	July 2020
(5) Z. Meng et al. [13]	Image	Content authentication	Not need	Response on Output layer of CNN	0.0077	1.0000	July 2021
(6) C. Qin et al. [61]	Image	Content authentication	Not need	Response on fully connected layer of CNN	0.0198	0.9998	Nov. 2021

The schemes are compared based on the research object, the research purpose, whether the research object is necessary to be trained, the information used to generate the hash value, the average verification time, and the average verification accuracy. The average verification time refers to the average time required for the scheme to verify the perceptual hash value of an image. And, for the (2) scheme, since the scheme divides a video into 150 frames to calculate and verify the perceptual hash value of the video [58], the average verification time refers to the average time required for the scheme to verify the perceptual hash value of a 150-frame video.

From the results of the average verification time and average verification accuracy in the table, compared with the related works, our proposed schemes (4) and (5) take less verification time and get higher verification accuracy.

After comparing with related works, our proposed two CNN-based perceptual hashing schemes are compared. For the (4) scheme in the table, we performed augmentation on images and then fine-tuned the CNN based on these images to generate a CNN of two classes for classifying the image we need to calculate the perceptual hash value for and other irrelevant images. This scheme was based on the whole weight coefficients and the network structure information of the CNN after fine-tuning to calculate the perceptual hash value of the image.

For the (5) scheme in the table, we performed augmentation on images to generate modified/edited images and then input these images to the CNN to obtain the corresponding output. This scheme is based on the output of the CNN to calculate the perceptual hash value of the image.

Compared with the (4) scheme, the (5) scheme does not require fine-tuning on the CNN for each target image and only needs the output of the CNN to calculate and verify the perceptual hash value of the target image. As for performance, the experimental results of the (4) scheme showed a verification accuracy of 1.00 for 1 million target images, while in the (5) scheme, with the threshold set to 3500, the verification accuracy for 1 million images which the same ones was also 1.00.

Therefore, perceptual hashing scheme using CNN output without fine-tuning reduce the calculation time while maintaining the classification accuracy. However, perceptual hashing scheme using weights and biases of the trained CNN can be applied to applications such as digital watermarking and image groups, so it cannot be evaluated which scheme is better only in terms of calculation time.

Chapter 9

Conclusion

In this paper, in order to solve problems for digital watermarking that misappropriation of watermark information, ensure equivalence between original image and modified/edited image, and multiple digital watermarks without depending on trusted third party, we proposed a design scheme of digital rights management system using digital watermarking, blockchain and perceptual hashing. The system uses blockchain to record and store watermark information, provide reliable watermark information for digital watermarking, and reduce the amount of necessary watermark information for copyright protection, by using the security of blockchain. In addition, this system uses the robustness of perceptual hashing for image modification and editing, and the similarity between the perceptual hash values of the original image and a modified or edited image, to provide information that is difficult to destroy. This is done through multiple digital watermarks to prove the author's copyright and the order of image modification or editing.

And, in order to improve conventional perceptual hashing for digital rights management, we proposed a perceptual hashing scheme based on CNN. The proposed scheme generates an identical perceptual hash value for the target image and its modified/edited images. A CNN is applied to solve the problem of conventional perceptual hash algorithms, that is, rotation and flipping. In addition, a trained CNN is used to classify images to be identified. When an image to be identified is classified with the target image, the message digest of the weights and biases of the trained CNN is used as a perceptual hash value of the image to be identified. In order to use an identical perceptual hash value to manage all images in copyrighted works, we developed the perceptual hashing scheme using weights and biases of the CNN after fine-tuning to deal with image groups, which generates an identical perceptual hash value for all images in group.

Moreover, in order to reduce the calculation time of fine-tuning, we proposed a design scheme for perceptual hashing based on CNN output for digital watermarking. The proposed scheme uses CNN to effectively retain the image features, solves the problems caused by image processing methods such as rotation and flipping that lead to pixel displacement, and makes up for the lack of conventional perceptual hash algorithms. For calculation of the perceptual hash value, after inputting different images to the CNN, it will obtain a different output. In addition, the cryptographic hash function calculates the hash value based on each bit of the input data. We use this cryptographic hash function to calculate the CNN output of the image as the perceptual hash value of the image.

After analyzing the proposals in this paper, we proposed two future topics. The first

is that embedding method of digital watermarking using CNN. This scheme fine-tunes images to output the bit series of watermark information, and expose these coefficients. And, inputs coefficients into the reconstructed CNN to extract the watermark, the watermark information can be obtained.

The second is perceptual hashing scheme for images containing multi-objects. This scheme segments target image according to objects, and performs fine-tuning with target image and objects. If the image to be identified contains each object of target image, the perceptual hash value can be verified.

Acknowledgements

First of all, thanks to my supervisor, Professor Kinoshita. In the course of doctoral research, Professor Kinoshita has brought me a lot of help. His rigorous academic attitude and scientific research methods have brought me great influence and will definitely have positive effects on my future career. I would like to express my sincere gratitude to Professor Kinoshita.

Secondly, I would like to thank the special assistant who guided me, Dr. Morizumi. In the course of doctoral research, Dr. Morizumi gave me a lot of help, so that my doctoral research work is carried out in an orderly manner. There are a lot of problems that I do not understand, and under his guidance, I will open my mind. I would like to express my gratitude to Dr. Morizumi.

The doctoral course is coming to an end, and next year I will start working in Japan. Looking back on the doctoral three years, I have left countless good memories. Every day of life and study is happy and fulfilling, and the teachers of Electrical, Electronics, and Information Engineering have also brought me great help. I would like to express my heartfelt thanks to all teachers, my classmates and friends.

Finally, I want to thank my parents. Without their support behind them, I can't complete my doctoral degree. And they are also the driving force for my study and life. I would like to express my heartfelt thanks.

Bibliography

[1] Herrigel Alexander, Joseph ó Ruanaidh, Holger Petersen, Shelby Pereira, and Thierry Pun, "Secure copyright protection techniques for digital images," International Workshop on Information Hiding, pp. 169-190, Springer, Berlin, Heidelberg, 1998.

[2] Li Shujun, "Multimedia encryption," Encyclopedia of Multimedia Technology and Networking, Second Edition (2009): 972-977.

[3] Nikolaidis Nikos, and Ioannis Pitas, "Copyright protection of images using robust digital signatures," 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, vol. 4, pp. 2168-2171, 1996.

[4] O'Ruanaidh J. J. K., W. J. Dowling, and F. M. Boland, "Watermarking digital images for copyright protection," IEE Proceedings-Vision, Image and Signal Processing 143, no. 4 (1996): 250-256.

[5] Lin Ching-Yung, "Watermarking and digital signature techniques for multimedia authentication and copyright protection," COLUMBIA UNIVERSITY Graduate School of Arts and Sciences, Thesis, 2001.

[6] Zhao Bo, Liming Fang, Hanyi Zhang, Chunpeng Ge, Weizhi Meng, Liang Liu, and Chunhua Su, "Y-DWMS: A digital watermark management system based on smart contracts," Sensors 19, no. 14 (2019): 3091.

[7] Ante Lennart, "Non-fungible token (NFT) markets on the Ethereum blockchain: Temporal development, cointegration and interrelations," Available at SSRN 3904683 (2021).

[8] MENG Zhaoxiong, MORIZUMI Tetsuya, MIYATA Sumiko, KINOSHITA Hirotsugu, "Perceptual hashing based on machine learning for blockchain and digital watermarking," 2019 Third World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), pages:193-198, 30-31 July 2019.

[9] MENG Zhaoxiong, MORIZUMI Tetsuya, MIYATA Sumiko, KINOSHITA Hirotsugu, "A Scheme of Digital Copyright Management System Based on Blockchain and Digital Watermarking -- Research on Improvement Method of Perceptual Hashing based on Machine Learning," Social Implications of Technology and Information Ethics (SITE), IEICE Technical Report 119, no. 329 (2019): 21-27.

[10]MENG Zhaoxiong, MORIZUMI Tetsuya, MIYATA Sumiko, KINOSHITA Hirotsugu, "Design scheme of copyright management system based on digital watermarking and blockchain," 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), volume:02, pages:359-364, 23-27 July 2018.

[11]MENG Zhaoxiong, "Design proposal of digital image copyright management system based on digital watermarking and blockchain," KANAGAWA UNIVERSITY Department of Electrical, Electronics and Information Engineering, Master thesis, 2019.

[12]MENG Zhaoxiong, MORIZUMI Tetsuya, MIYATA Sumiko, KINOSHITA Hirotsugu, "An Improved Design Scheme for Perceptual Hashing based on CNN for Digital Watermarking," 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), pages:1789-1794, 13-17 July 2020.

[13]MENG Zhaoxiong, MORIZUMI Tetsuya, MIYATA Sumiko, KINOSHITA Hirotsugu, "Design Scheme of Perceptual Hashing based on Output of CNN for Digital Watermarking," 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), pages:1345-1350, 12-16 July 2021.

[14]Tirkel A Z, Rankin G A, Van Schyndel R M, et al., "Electronic watermark," Digital Image Computing, Technology and Applications (DICTA'93), 1993: 666-673.

[15]R. Chandramouli, N. D. Memon, M. Rabbani, "Digital watermarking," Encyclopedia of Imaging Science and Technology, New York:Wiley, 2002.

[16]Nicholas Paul Sheppard, Reihaneh Safavi-Naini and Philip Ogunbona, "Digital watermarks for copyright protection," Journal of Law and Information Science, 12 (1), pages 110-130, 2002.

[17]Deepa Merin Jose, R.Karuppathal and A.Vincent Antony Kumar, "Copyright Protection using Digital Watermarking," National Conference on Advances in Computer Science and Applications with International Journal of Computer Applications (NCACSA), International Journal of Computer Applications (IJCA), 2012.

[18]Chaw-Seng WOO, "Digital Image Watermarking Methods for Copyright Protection and Authentication," PhD thesis, Queensland University of Technology, March 2007.

[19]C. Obimbo, B. Salami, "Using digital watermarking for copyright protection," Mithun Das Gupta (Ed.), Watermarking, vol. 2., University of Guelph, Canada (2012).

[20]Liu, Lin, "A survey of digital watermarking technologies," Department of Electrical and Computer Engineering, State University of New York at Stony Brook, NY (2005): 11794-2350.

[21]Li Xiao-Wei, and In-Kwon Lee, "Robust copyright protection using multiple ownership watermarks," Optics Express 23, No.3, 2015, pp.3035-3046.

[22]Nakamoto Satoshi, "Bitcoin: A peer-to-peer electronic cash system," Decentralized Business Review (2008): 21260.

[23]Crosby MA, Pattanayak P, Verma S, Kalyanaraman V, "BlockChain Technology: Beyond Bitcoin," Applied Innovation, No. 2, pp. 6-10, 2016.

[24]Alexander Savelyev, "COPYRIGHT IN THE BLOCKCHAIN ERA: PROMISES AND CHALLENGES," National Research University Higher School of Economics (HSE), Basic Research Program Working Paper, 2017.

[25]Shigeru Fujimura, Hiroki Watanabe, Atsushi Nakadaira, Tomokazu Yamada, Akihito Akutsu and Jay (Junichi) Kishigami, "BRIGHY: A Concept for a Decentralized Rights Management System Based on Blockchain," IEEE 5th International Conference on Consumer Electronics Berlin (ICCE-Berlin), 2015.

[26]Ruzhi Xu, Lu Zhang, Huawei Zhao and Yun Peng, "Design of Network Media's Digital Rights Management Scheme Based on Blockchain Technology," IEEE 13th International Symposium on Autonomous Decentralized Systems, 2017.

[27]Bart PRENEEL, "Analysis and Design of Cryptographic Hash Functions," Doctoral Dissertation, Katholieke Universiteit Leuven, 1993.

[28]Rajeev Sobti, G.Geetha, "Cryptographic Hash Functions: A Review," IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 2, March 2012.

[29]Saif Al-Kuwari, James H. Davenport, Russell J. Bradford, "Cryptographic Hash Functions: Recent Design Trends and Security Notions," IACR Cryptology ePrint Archive, 2011.

[30]Christoph Zauner, "Implementation and Benchmarking of Perceptual Image Hash Functions," Fachhochschule-Master program Secure Information Systems, Thesis, July 2010.

[31]A. Hadmi, W. Puech, B.A.E. Said, A.A. Quahman, "Perceptual image hashing," M.D. Gupta (Ed.), Computer and Information Science: Watermarking, vol. 2, Rijeka, Croatia: InTech, 2012.

[32]Li WENG, "Perceptual Multimedia Hashing", PhD thesis, Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven, Belgium, 2012.

[33]Vladimir Viies, "POSSIBLE APPLICATION OF PERCEPTUAL IMAGE HASHING," TALLINN UNIVERSITY OF TECHNOLOGY Faculty of Information Technology Department of Computer Engineering, Master thesis, 2015.

[34]Ruchita Kesarkar and Mrs R W Deshpande, "A Survey on Perceptual image hash for authentication of content," International Research Journal of Engineering and Technology (IRJET), Volume: 03, Issue: 01, January 2016.

[35]Andrea Drmic, Marin Silic, Goran Delac, Klemo Vladimir, Adrian S. Kurdija, "Evaluating Robustness of Perceptual Image Hashing Algorithms," 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 22-26 May 2017.

[36]Rhayma Hanen, Achraf Makhloufi, and Ahmed Ben Hmida, "Self-authentication scheme based on semi-fragile watermarking and perceptual hash function," In International Image Processing, Applications and Systems Conference, pp. 1-6. IEEE, 2014.

[37]Gauri Barse, S.D.Satav, "Content Authentication and Forge Detection using Perceptual Hash for Image Database," International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8169, Volume: 4, Issue: 11, 125- 129, 2015.

[38]L. Weng and B. Preneel, "A secure perceptual hash algorithm for image content authentication," In Proc. of International Conference on Communications and Multimedia Security, volume 7025 of LNCS, pages 108-121, 2011.

[39]KINOSHITA Hirotsugu, SATOH Masafumi, KOBAYASHI Terunobu, "A WATERMARK SYSTEM BASED ON THE STRUCTURED INFORMATION," European Association for Signal Processing 98, Vol.4, pages 2273-2276, Greece, September 1998.

[40]KINOSHITA Hirotsugu, "AN IMAGE DIGITAL SIGNATURE SYSTEM WITH ZKIP FOR THE GRAPH ISOMORPHISM," IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, ICIP96, Vol. III, pages 247-250, September 1996.

[41]Langley Pat, Elements of machine learning, Morgan Kaufmann, 1996.

[42]Sammut Claude, and Geoffrey I. Webb, eds, Encyclopedia of machine learning, Springer Science & Business Media, 2011.

[43]Jordan Michael I., and Tom M. Mitchell, "Machine learning: Trends, perspectives, and prospects," Science 349, no. 6245 (2015): 255-260.

[44]Deng Li, and Dong Yu, "Deep learning: methods and applications," Foundations and trends in signal processing 7, no. 3-4 (2014): 197-387.

[45]Goodfellow Ian, Yoshua Bengio, and Aaron Courville, Deep learning, MIT press, 2016.

[46]Guo Yanming, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S. Lew," Deep learning for visual understanding: A review," Neurocomputing 187 (2016): 27-48.

[47]Jain, Ramesh, Rangachar Kasturi, and Brian G. Schunck, Machine vision, Vol. 5. New York: McGraw-hill, 1995.

[48]Davies E. Roy, Machine vision: theory, algorithms, practicalities, Elsevier, 2004.

[49]Milan Sonka, Vaclav Hlavac and Roger Boyle, "Image processing, analysis, and machine vision," 2014.

[50]Steger Carsten, Markus Ulrich, and Christian Wiedemann, Machine vision algorithms and applications, John Wiley & Sons, 2018.

[51]Keiron O'Shea and Ryan Nash, "An introduction to convolutional neural networks, " arXiv preprint arXiv:1511.08458, 2015.

[52]Albawi Saad, Tareq Abed Mohammed, and Saad Al-Zawi, "Understanding of a convolutional neural network," In 2017 International Conference on Engineering and Technology (ICET), pp. 1-6, IEEE, 2017.

[53]Li Yandong, Z. B. Hao, and Hang Lei, "Survey of convolutional neural network," Journal of Computer Applications 36, no. 9 (2016): 2508-2515.

[54]Matthew D. Zeiler, and Rob Fergus, "Visualizing and understanding convolutional networks," European conference on computer vision. springer, Cham, 2014.

[55]Simonyan Karen, and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[56]Özyurt Fatih, Hüseyin Kutlu, Engin Avci, and Derya Avci, "A new method for classification of images using convolutional neural network based on Dwt-Svd perceptual hash function," In 2018 3rd International Conference on Computer Science and Engineering (UBMK), pp. 410-413. IEEE, 2018.

[57]Jiang Cuiling, and Yilin Pang, "Perceptual image hashing based on a deep convolution neural network for content authentication," Journal of Electronic Imaging 27, no. 4 (2018): 043055.

[58]Wu Huisi, Yawen Zhou, and Zhenkun Wen, "Video tamper detection based on convolutional neural network and perceptual hashing learning," Computer Graphics International Conference, pp. 107-118. Springer, Cham, 2019.

[59]Venugopal Narmatha, and Kamarasan Mari, "An Automated Glaucoma Image Classification model using Perceptual Hash-Based Convolutional Neural Network," In 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 185-190. IEEE, 2019.

[60]Özyurt Fatih, Türker Tuncer, Engin Avci, Mustafa Koç, and İhsan Serhatlioğlu, "A novel liver image classification method using perceptual hash-based convolutional neural network," Arabian Journal for Science and Engineering 44, no. 4 (2019): 3173-3182.

[61]Qin Chuan, Enli Liu, Guorui Feng and Xinpeng Zhang, "Perceptual Image Hashing for Content Authentication Based on Convolutional Neural Network with Multiple Constraints," IEEE Transactions on Circuits and Systems for Video Technology, pp. 4523 - 4537, 2021.

[62]Al-Ghadi Musab, Théo Azzouza, Petra Gomez-Krämer, Jean-Christophe Burie, and Mickaël Coustaty, "Robust Hashing for Character Authentication and Retrieval Using Deep Features and Iterative Quantization," In International Conference on Document Analysis and Recognition, pp. 466-481, Springer, Cham, 2021.

[63]Adrian G. Bors and Ioannis Pitas, "IMAGE WATERMARKING USING DCT DOMAIN CONSTRAINTS," Proceedings of 3rd IEEE International Conference on Image Processing, Lausanne Switzerland, 19-19 September 1996.

[64]Faisal Alurki and Russell Mersereau, "A ROBUST DIGITAL WATERMARK PROCEDURE FOR STILL IMAGES USING DCT PHASE MODULATION," 10th European Signal Processing Conference, Tampere Finland, 4-8 September 2000.

[65]Chetna, "Digital Image Watermarking using DCT," A Monthly Journal of Computer Science and Information Technology, Vol.3, Issue.9, pages 586-591, September 2014.

[66]Heena Shaikh,Mohd.Imran Khan, Yashovardhan Kelkar, "A Robust DWT Digital Image Watermarking Technique Basis On Scaling Factor," International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.2, No.4, August 2012.

[67]Hong-an Li, Zhanli Li, Zhuoming Du, Qi Wang, "Digital Image Watermarking Algorithm Using the Intermediate Frequency," TELKOMNIKA, Vol.14, No.4, December 2016, pp. 1424-1431, 2016.

[68]V. Santhi, and Arunkumar Thangavelu, "DWT-SVD combined full band robust watermarking technique for color images in YUV color space," International Journal of Computer Theory and Engineering 1, No.4, 2009, pp.424.

[69]Nick Locascio, https://towardsdatascience.com/black-box-attacks-on-perceptual-image-hashes-with-gans-cc1be11f277.

[70]Krizhevsky, Alex, Ilya Sutskever and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems 25 (2012): 1097-1105.

[71]ImageNet, http://www.image-net.org/.

[72]Python, Image Hashing library, https://pypi.org/project/ImageHash/.

[73]Deutsch Peter, "DEFLATE compressed data format specification version 1.3," 1996.

[74]Van Der Walt, S., Colbert, S. C. and Varoquaux, G., "The NumPy array: a structure for efficient numerical computation," Computing in science & engineering, 13(2), 22-30, 2011.

[75]Gulli Antonio and Sujit Pal, Deep learning with Keras, Packt Publishing Ltd, 2017.

[76]Department of Computer Science and Engineering, University of Washington, "Object and Concept Recognition for Content-Based Image Retrieval," available: http://imagedatabase.cs.washington.edu/.

[77]University of Southern California, The USC-SIPI Image Database, http://sipi.usc.edu/database/database.php/.

[78]"imgaug," available: https://imgaug.readthedocs.io/en/latest/index.html.

[79]Juan Benet, "Ipfs-content addressed, versioned, p2p file system," arXiv preprint arXiv: 1407.3561, 2014.

[80]Wood Gavin, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum project yellow paper 151, no. 2014 (2014): 1-32.

[81]Aufa Farah Jihan, and Achmad Affandi, "Security system analysis in combination method: RSA encryption and digital signature algorithm," In 2018 4th International Conference on Science and Technology (ICST), pp. 1-5. IEEE, 2018.

[82]Myers Michael, Rich Ankney, Ambarish Malpani, Slava Galperin, and Carlisle Adams, "X. 509 Internet public key infrastructure online certificate status protocol-OCSP," 1999.

[83]E. Levy and A. Silberschatz, "Distributed file systems: Concepts and examples," ACM Computing Surveys, 22(4):321- 375, Dec. 1990.

[84]John Howard, Michael Kazar, Sherri Menees, David Nichols, Mahadev Satyanarayanan, Robert Sidebotham, and Michael West, "Scale and performance in a distributed file system," ACM Transactions on Computer Systems, 6(1):51-81, February 1988.