applied sciences

MDPI

Article

ZWNet: A Deep-Learning-Powered Zero-Watermarking Scheme with High Robustness and Discriminability for Images

Can Li [1], Hua Sun [2,3], Changhong Wang [2,3], Sheng Chen [2,3], Xi Liu [2,3], Yi Zhang [2,3], Na Ren [4,5] and Deyu Tong [1,*]

[1] College of Information Engineering, Nanjing University of Finance and Economics, Nanjing 210023, China; 2120201832@stu.nufe.edu.cn
[2] Hunan Engineering Research Center of Geographic Information Security and Application, Changsha 410007, China
[3] The Third Surveying and Mapping Institute of Hunan Province, Changsha 410018, China
[4] Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing 210023, China
[5] Nanjing Geomarking Information Technology Co., Ltd., Nanjing 210023, China
* Correspondence: tongdeyu@nufe.edu.cn

Abstract: In order to safeguard image copyrights, zero-watermarking technology extracts robust features and generates watermarks without altering the original image. Traditional zero-watermarking methods rely on handcrafted feature descriptors to enhance their performance. With the advancement of deep learning, this paper introduces "ZWNet", an end-to-end zero-watermarking scheme that obviates the necessity for specialized knowledge in image features and is exclusively composed of artificial neural networks. The architecture of ZWNet synergistically incorporates ConvNeXt and LK-PAN to augment the extraction of local features while accounting for the global context. A key aspect of ZWNet is its watermark block, as the network head part, which fulfills functions such as feature optimization, identifier output, encryption, and copyright fusion. The training strategy addresses the challenge of simultaneously enhancing robustness and discriminability by producing the same identifier for attacked images and distinct identifiers for different images. Experimental validation of ZWNet's performance has been conducted, demonstrating its robustness with the normalized coefficient of the zero-watermark consistently exceeding 0.97 against rotation, noise, crop, and blur attacks. Regarding discriminability, the Hamming distance of the generated watermarks exceeds 88 for images with the same copyright but different content. Furthermore, the efficiency of watermark generation is affirmed, with an average processing time of 96 ms. These experimental results substantiate the superiority of the proposed scheme over existing zero-watermarking methods.

Keywords: zero-watermarking; deep learning; robustness; discriminability; ConvNeXt; LK-PAN

Citation: Li, C.; Sun, H.; Wang, C.; Chen, S.; Liu, X.; Zhang, Y.; Ren, N.; Tong, D. ZWNet: A Deep-Learning-Powered Zero-Watermarking Scheme with High Robustness and Discriminability for Images. *Appl. Sci.* **2024**, *14*, 435. https://doi.org/10.3390/app14010435

Academic Editors: Frank Y. Shih and Xin Zhong

Received: 17 October 2023
Revised: 19 December 2023
Accepted: 21 December 2023
Published: 3 January 2024

1. Introduction

In contrast to cryptography, which primarily focuses on ensuring message confidentiality, digital watermarking places greater emphasis on copyright protection and tracing [1,2]. Classical watermarking involves the covert embedding of a watermark (a sequence of data) within media files, allowing for the extraction of this watermark even after data distribution or manipulation, enabling the identification of data sources or copyright ownership [3]. However, this embedding process necessarily involves modifications to the host data, which can result in some degree of degradation to data quality and integrity. In response to the demand for high fidelity and zero tolerance for data loss, classical watermarking has been supplanted by zero-watermarking. Zero-watermarking focuses on extracting robust features and their fusion with copyright information [4,5]. Notably, a key characteristic of zero-watermarking lies in the generation or construction of the zero-watermark itself, as opposed to its embedding.

Commonly, zero-watermarking algorithms are traditionally reliant on handcrafted features and typically involve a three-stage process. The first stage entails computing robust features, followed by converting these features into a numerical sequence in the second stage. The third stage involves fusing the numerical sequence with copyright identifiers, resulting in the generation of a zero-watermark without any modifications to the original data. Notably, the specific steps and features in these three stages are intricately designed by experts or scholars, thereby rendering the performance of the algorithm contingent upon expert knowledge. Moreover, once a zero-watermarking algorithm is established, continuous optimization becomes challenging, representing a limitation inherent in handcrafted approaches.

Introducing deep learning technology is a natural progression to overcome the reliance on expert knowledge and achieve greater optimization in zero-watermarking algorithms. Deep learning has recently ushered in significant transformations in computer vision and various other research domains [6–9]. Numerous tasks, including image matching, scene classification, and semantic segmentation, have exhibited remarkable improvements when contrasted with classical methods [10–14]. The defining feature of deep learning is its capacity to replace handcrafted methods reliant on expert knowledge with Artificial Neural Networks (ANNs). Through training ANNs with ample samples, these networks can effectively capture the intrinsic relationships among the samples and model the associations between inputs and outputs. Inspired by this paradigm shift, the zero-watermarking method can also transition towards an end-to-end mode with the support of ANNs, eliminating the need for handcrafted features.

In addressing watermark performance, conventional handcrafted methods encounter challenges in simultaneously enhancing robustness and discriminability. To address this limitation and achieve concurrent improvements, this paper proposes a novel zero-watermarking network named ZWNet, which employs distinct strategies. Firstly, ZWNet combines ConvNeX and LK-PAN to enhance the extraction of local features and consider the global context more comprehensively. Secondly, the watermark block is strategically designed as the leading component, integrating with copyright information, encrypting the watermark, and generating a distinctive image identifier. Thirdly, the training strategy for ZWNet focuses on the image identifier it produces. To enhance robustness, ZWNet is trained to yield the same image identifier for both the original and attacked images. Simultaneously, to improve discriminability, ZWNet is trained to produce different image identifiers for distinct images. Experimental results further validate the effectiveness of ZWNet, as evidenced by the normalized coefficient of the zero-watermark consistently exceeding 0.97 for robustness and the Hamming distance between watermarks with the same copyright and different images surpassing 88 for discriminability.

The contributions of this paper can be summarized as follows:

(1) Introduction of a novel approach that combines ConvNeXt and LK-PAN to enhance feature extraction, effectively addressing both global context and local features.

(2) Transformation of the problem of improving watermark performance into a classification task, leveraging the common framework provided by deep learning.

(3) ZWNet exhibits notable discriminability, ensuring that the generated zero-watermark is distinct enough to differentiate between different images sharing the same copyright. This capability addresses the challenge of zero-watermark confusion.

The structure of this paper is organized as follows: Section 2 provides a concise background on zero-watermarking, ConvNeXt, and LK-PAN. Section 3 introduces the proposed zero-watermarking scheme, "ZWNet". Section 4 delves into the presentation of experimental results and subsequent discussion. Finally, Section 5 presents the concluding remarks.

2. Related Work

This section introduces three key components of related work. Section 2.1 provides a concise introduction to zero-watermarking and analyzes similar methods based on deep learning. Section 2.2 presents an overview of ConvNeXt as a feature extraction

network. Furthermore, Section 2.3 introduces LK-PAN, delineating its role in providing an optimization structure to enhance ConvNeXt.

2.1. Zero-Watermarking

The concept of zero-watermarking in image processing was originally introduced by Wen et al. [15]. This technology has garnered significant attention and research interest due to its unique characteristic of preserving the integrity of media data without any modifications. Taking images as an example, the zero-watermarking process can be broadly divided into three stages. The first stage involves the computation of robust features. In this phase, various handcrafted features such as Discrete Cosine Transform (DCT) [16,17], Discrete Wavelet Transform (DWT) [18], Lifting Wavelet Transform [19], Harmonic Transform [5], and Fast Quaternion Generic Polar Complex Exponential Transform (FQGPCET) [20] are calculated and utilized to represent the stable features of the host image. The second stage focuses on the numerical conversion of these features into a numerical sequence. Mathematical transformations such as Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) are employed to filter out minor components and extract major features [16,21]. The resulting feature sequence from this stage serves as a condensed identifier of the original image. However, this sequence alone cannot serve as the final watermark since it lacks any copyright-related information. Hence, the third stage involves the fusion of the feature sequence with copyright identifiers. Copyright identifiers can encompass the owner's signature image, organization logos, text, fingerprints, or any digitized media. To ensure the zero-watermark cannot be forged or unlawfully generated, cryptographic methods such as Advanced Encryption Standard (AES) or Arnold Transformation [22] are often utilized to encrypt the copyright identifier and feature sequence. The final combination can be as straightforward as XOR operations [16]. Consequently, the zero-watermark is generated and can be registered with the Intellectual Property Rights (IPR) agency. Additionally, copyright verification is a straightforward process involving the regeneration of the feature sequence and its comparison with the registered zero-watermark. The process of zero-watermarking technology is illustrated in Figure 1.

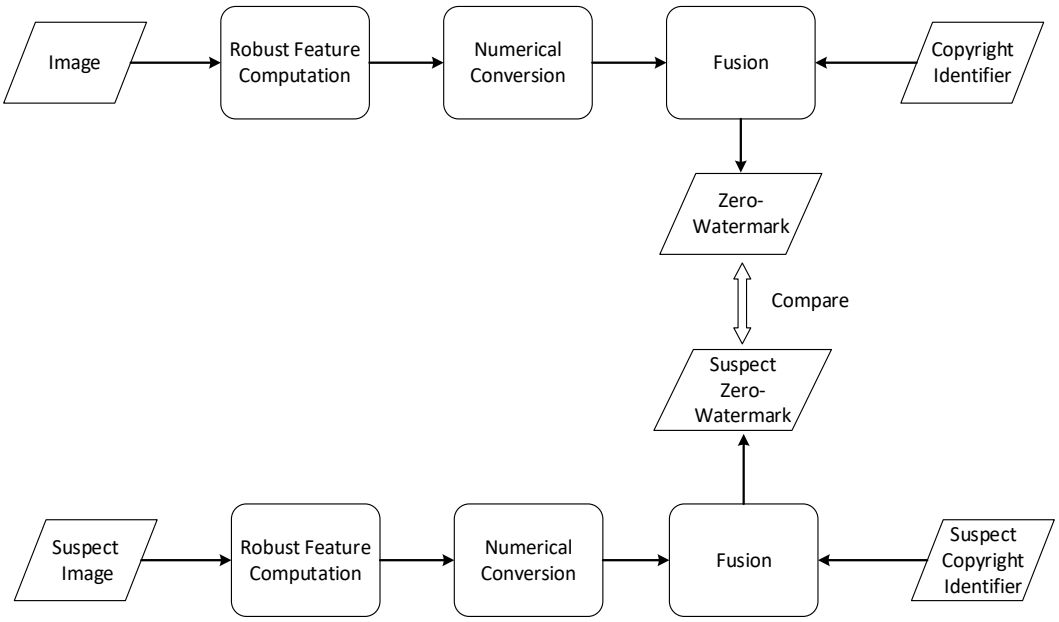

Figure 1. The process of zero-watermark generation and verification.

Presently, several research efforts focus on deep-learning-based watermarking methods, encompassing both classical watermarking of embedding style and zero-watermarking of generative style. In the domain of embedding-style watermarking, a method inspired

by the architecture of Autoencoder has been proposed. In this approach, Autoencoders encode the watermark and embed it using convolutional networks. For watermark extraction, Autoencoders are also employed to extract and decode the watermark [23]. Other Autoencoder-based methods aim to enhance robustness or improve efficiency [24,25]. Despite their superior performance in robustness and elimination of reliance on prior knowledge, embedding-style watermarking significantly differs from zero-watermarking methods, as the latter maintains the original data unchanged. Additionally, it is noteworthy that zero-watermarking places greater emphasis on discriminability, a focus less pronounced in embedding-style watermarking.

In the realm of deep-learning-based zero-watermarking methods, a hybrid scheme that combines traditional Discrete Wavelet Transform (DWT) and the deep neural network ResNet-101 has been proposed. This approach involves applying DWT to the host image and subsequently sending the wavelet coefficients to ResNet-101 [26]. While exhibiting strong robustness against translation and clipping, this scheme falls short of being an end-to-end solution. Regarding end-to-end zero-watermarking, some studies employ Convolutional Neural Networks (CNN), VGG-19 (developed by the Oxford Visual Geometry Group), or DenseNet to generate robust watermark sequences [27–29]. Another line of research predominantly revolves around the concept of style transfer [30]. In the watermark generation phase, it utilizes VGG to merge the content of the copyright logo with the style of the host image. In the verification stage, another CNN is employed to eliminate the style component and extract the copyright content. Although these approaches have demonstrated promising levels of robustness compared to handcrafted methods, we believe they fall short in adequately considering multi-level features within the image. This limitation arises because when using CNN or VGG to upsample the image, the higher-level features have a less effective receptive field than the theoretical receptive field [31]. Furthermore, one drawback of these zero-watermark networks is the insufficient emphasis on discriminability. This means the generated zero-watermarks for different images should be distinct enough to prevent copyright ambiguity.

2.2. ConvNeXt

As discussed in Section 2.1, Convolutional Neural Networks (CNN) have been employed as the feature extraction component in existing watermarking methods. However, it is noteworthy that the performance of CNN has become outdated in various tasks. Hence, Liu et al. introduced ConvNeXt, a nomenclature devised to distinguish it from traditional Convolutional Networks (ConvNets) while signifying the next evolution in ConvNets [32]. Rather than presenting an entirely new architectural paradigm, ConvNeXt draws inspiration from the ideas and optimizations put forth in the Swin Transformer [33] and applies similar strategies to enhance a standard ResNet [8]. These optimization strategies can be summarized as follows:

(1) Modification of stage compute ratio: ConvNeXt adjusts the number of blocks within each stage from (3, 4, 6, 3) to (3, 3, 9, 3).

(2) Replacement of the stem cell: The introduction of a patchify layer achieved through non-overlapping 4×4 convolutions.

(3) Utilization of grouped and depthwise convolutions.

(4) Inverted Bottleneck design: This approach involves having the hidden layer dimension significantly larger than that of the input.

(5) Incorporation of large convolutional kernels (7×7) and depthwise convolution layers within each block.

(6) Micro-level optimizations: These include the replacement of ReLU with GELU, fewer activation functions, reduced use of normalization layers, the substitution of Batch Normalization with Layer Normalization, and the implementation of separate downsampling layers.

Remarkably, the amalgamation of these strategies results in ConvNeXt achieving a state-of-the-art level of performance in image classification, all without requiring substantial changes to the network's underlying structure. Furthermore, a key feature of this paper lies in its detailed presentation of how each optimization incrementally enhances performance, effectively encapsulated in Figure 2.

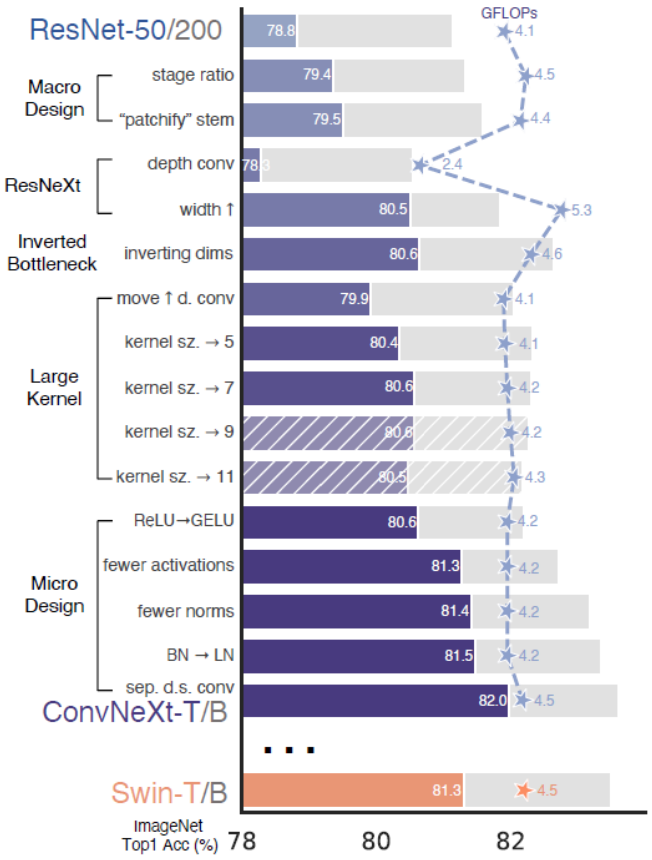

Figure 2. Incremental improvement through optimization steps in ConvNeXt [32]. The foreground bars represent results from ResNet-50/Swin-T, while the gray bars represent results from ResNet-200/Swin-B.

From Figure 2, it is evident that employing the strategy of stage ratio modification and patchify stem leads to an improvement in accuracy, increasing from 78.8% to 79.5%. Further enhancements are observed with the introduction of depth convolution and larger width, resulting in an accuracy improvement of 80.5%. The utilization of an inverted bottleneck and larger kernel size contributes to a higher accuracy of 80.6%. Finally, with micro-optimizations, the accuracy of ConvNeXt reaches 82.0%, surpassing that of Swin.

2.3. LK-PAN

While ConvNeXt offers a straight-line structure that effectively captures local features, it may fall short in dedicating sufficient attention to the global context. To address this limitation and enhance the capabilities of ConvNeXt, a path aggregation mechanism, LK-PAN, is introduced. LK-PAN originates from the Path Aggregation Network (PANet), which was initially introduced in the context of instance segmentation to bolster the hierarchy of feature extraction networks. The primary structure of PANet is depicted in Figure 3.

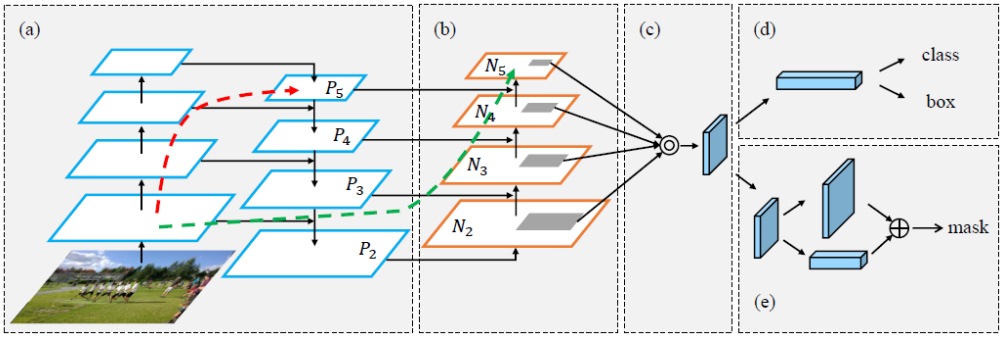

Figure 3. The primary structure of PANet [34]. (**a**) The backbone part of PANet; (**b**) Bottom-up path augmentation; (**c**) Adaptive feature pooling; (**d**) Box branch; (**e**) Fully-connected fusion.

In Figure 3, we observe that part (a) represents the classical network structure of the Feature Pyramid Network (FPN), which is named for its pyramid-like arrangement [35]. However, it's important to note that the influence of low-level features on high-level features is limited due to the long paths, as indicated by the red dashed lines in Figure 3. These paths can comprise over 100 layers. As mentioned in Section 1, while the theoretical receptive field of P5 may be quite large, it does not manifest as such in practice due to the numerous convolution, pooling, and activation operations. Therefore, PANet introduced a bottom-up path augmentation, as depicted in Figure 3b. This approach aggregates the top-most features from both the low-level features and features at the same level. Consequently, this mechanism substantially shortens the connection between low-level features and the top-most features to around 10 layers. Thus, it effectively enhances feature expression for local areas and minor details. PANet's contributions also encompass adaptive feature pooling (Figure 3c) and fully-connected fusion (Figure 3d). However, these two mechanisms are more closely related to the task of instance segmentation and will not be elaborated upon here.

Building upon the foundation of PANet, the Large Kernel-PANet, abbreviated as LK-PAN [36], introduces some improvements. The primary feature of LK-PAN is the enlargement of the convolution kernel size. In contrast to PANet, LK-PAN utilizes 9×9 convolution kernels instead of the original 3×3 size. This augmentation is aimed at expanding the receptive field of the feature map, thereby enhancing the ability to discern minor features with greater precision. Another key change in LK-PAN is the adoption of a concatenation operation, replacing Figure 3c, for fusing features from different levels.

3. Proposed Scheme

3.1. Main Idea

At the heart of zero-watermarking technology lies the extraction of robust image features. While ConvNeXt offers deep insights for extracting dense features, it alone may not provide sufficient attention to fine-grained image semantics and local details. This can result in situations where the watermark differences between substantially different images are not distinct enough, affecting the discriminability of the zero-watermark. To fully leverage both local features and global context, ZWNet integrates ConvNeXt and LK-PAN as the backbone and neck components, enhancing the robustness and distinctiveness of multi-level features.

After the image feature extraction, a crucial challenge remains in training ZWNet to achieve robustness and discriminability. Additionally, there are requirements such as combining the watermark with copyright logos and encrypting the watermark. To address these issues, the watermark block is introduced as the head component of ZWNet. This block includes a linear layer for generating an image identifier, encryption layers, and copyright-mixture layers. In summary, the primary architecture of ZWNet is illustrated in Figure 4, with further network details elucidated below.

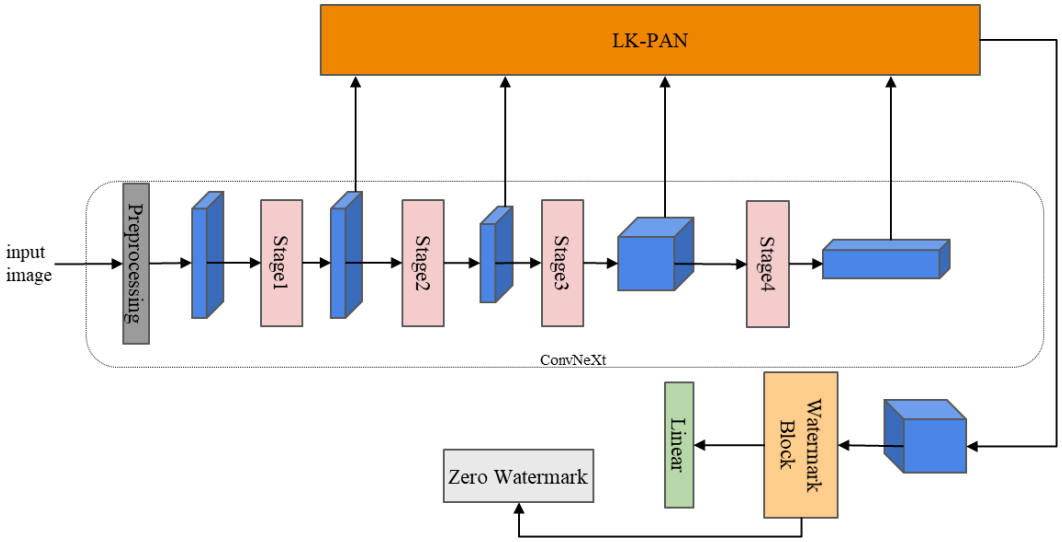

Figure 4. ZWNet structure.

3.2. Backbone Component

Prior to entering ZWNet's backbone component, training images undergo various attacks, which are managed by the preprocessing module depicted in Figure 4. Subsequently, they are fed into the ConvNeXt network, the details of which are illustrated in Figure 5.

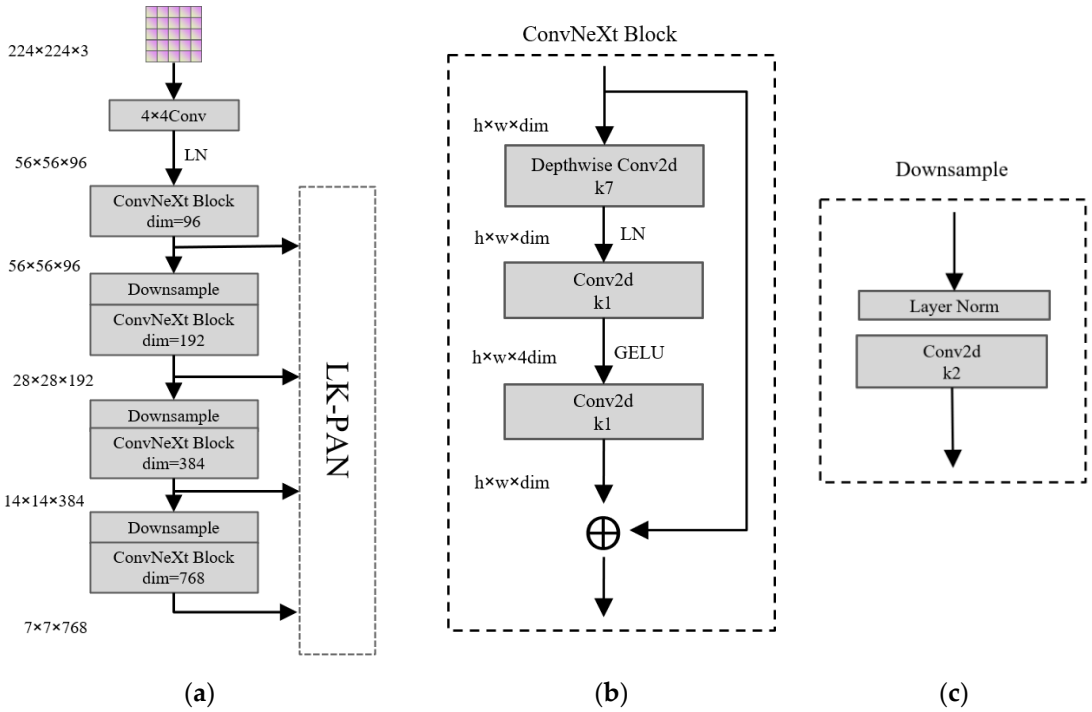

Figure 5. ZWNet's backbone details. (**a**) Main structure. (**b**) ConvNeXt block details. (**c**) Downsample layers details.

The input image has dimensions of 224 × 224 with three color channels (Red, Green, and Blue). Both the training and test datasets are formatted as JPG images with a resolution of 72 dpi. The image initially undergoes processing through a convolutional layer and layer normalization. Subsequently, it is directed through four ConvNeXt blocks and three downsample blocks. Each ConvNeXt block includes a residual connection, a depthwise convolution layer, and standard convolution layers. The downsample layer comprises

a normalization layer and a convolution layer. Importantly, it should be noted that the feature maps generated after each ConvNeXt block are then passed to LK-PAN, which serves as the neck component of ZWNet.

3.3. Neck Component

The neck component of ZWNet draws inspiration from LK-PAN, and its specifics are outlined in Figure 6.

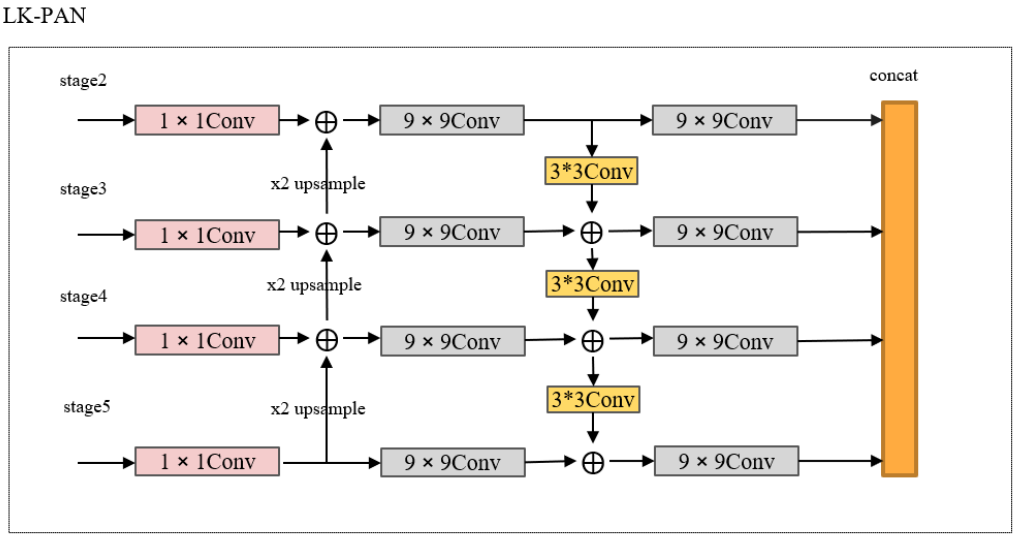

Figure 6. ZWNet's neck details.

The input to ZWNet's neck component comprises four branches, each corresponding to a feature map generated by one of the four ConvNeXt blocks from the backbone. Each branch begins with a 1×1 convolution operation, followed by the addition of upsampled features from higher levels and subsequent upsampling to match the low-level branch. These features then pass through a larger-kernel convolutional layer (9×9). Following the convolution operation, the features are combined with downsampled features and split into two branches. The first branch is downsampled using a 3×3 convolutional layer and sent to the higher level. The other branch undergoes an additional 9×9 convolutional layer and ultimately contributes to the final concatenate layer.

3.4. Head Component

The head component is the watermark block, encompassing four key functions: optimizing the feature maps, generating image identifiers, encryption, and merging with copyright information. The intricate structure is illustrated in Figure 7.

Within the watermark block, the input comprises feature maps generated by the neck component. These feature maps undergo processing through an exceptionally large depthwise convolutional layer, utilizing a 21-unit kernel. Subsequently, they are subjected to adaptive max pooling to maintain the size of the output feature map, fixed at $16 \times 16 \times 1$. This $16 \times 16 \times 1$ feature map can be viewed as the robust features of the input host image.

The feature map is then divided into two branches. The first branch is directed through a linear layer to produce an image identifier. This image identifier serves as a unique code differentiating the input image from others, which will be further elucidated in Section 3.5. The second branch is funneled through an encrypt-conv layer within a loop function. This loop function emulates the encryption process of the Arnold transformation, where the encrypt-conv layer, abbreviated as the encryption-convolution layer, executes a single permutation of the Arnold transformation. Key1 represents the secret key of the Arnold transformation, which is fed into the loop function. Post-encryption, the feature map undergoes quantization based on a threshold, T, resulting in the conversion of the

feature map into a binary sequence. This binary sequence is then merged with copyright information through an XOR operation.

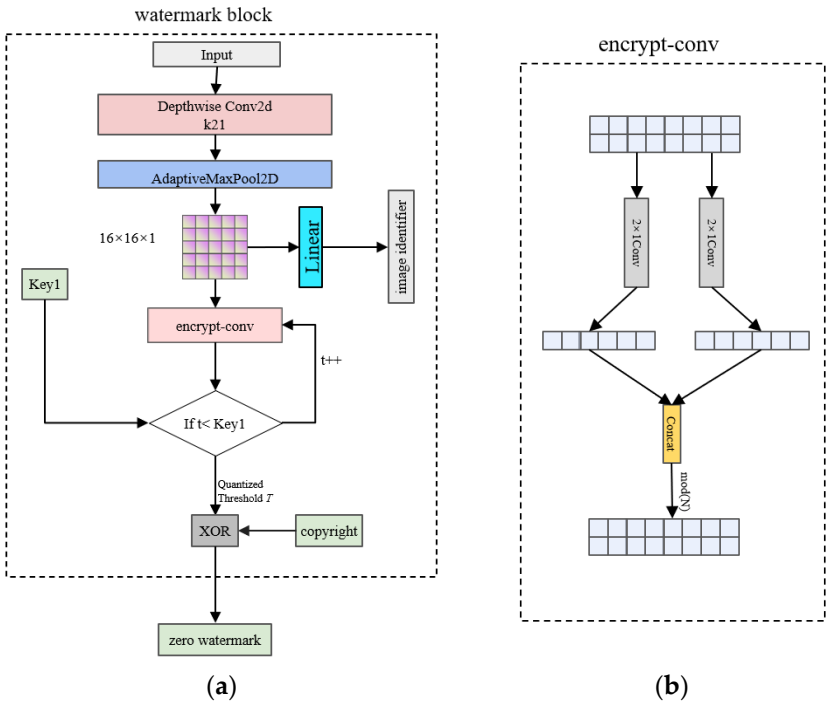

Figure 7. Structure of the watermark block. (**a**) Main structure. (**b**) Illustration of encrypt-conv layer.

3.5. Training

The image identifier plays a crucial role in distinguishing the input image from others and serves as the training target for ZWNet. There are various methods for generating an image identifier, such as assigning a unique value or utilizing a hash function. The key requirement is that different images should map to distinct identifiers. In the case of ZWNet, the training process can be conceptualized as optimizing the identifier output by the entire network to match the target identifier.

To ensure the robustness and discriminability of ZWNet simultaneously, we employ two strategies. The first strategy involves training ZWNet to generate the same identifier for both the original input image and the attacked versions. This approach encourages the network to extract consistent features even for images subjected to different attacks. The second strategy entails training ZWNet to produce different identifiers for different host images, promoting the network's ability to create distinct feature maps for varying images. Through these strategies, we reframe the problem of improving zero-watermark performance as a common task in deep learning, akin to multi-label classification.

In terms of implementation details, the image identifier in ZWNet is represented as a 256-bit binary sequence, and the network is treated as a multi-label task. Consequently, BCEWithLogitsLoss is employed as the loss function. This loss function sigmoidalizes the output first and then computes the difference between the target identifier T_i and the actual output value S_i as follows:

$$Loss = -\sum(T_i \times \log(S_i) + (1 - T_i) \times \log(1 - S_i))$$

Here, i represents the sequence index.

It is important to note that the image identifier is solely used during the training stage. In the testing phase or deployment, it remains unused, although it still generates the identifier. This is because the image identifier is utilized to train the network, and the network should not adapt beyond the training phase.

3.6. Application Usage of ZWNet

Once ZWNet has been trained, the generation of a zero-watermark involves the following steps:

(1) Input the host image with a size of 224 × 224 and jpg format into ZWNet to compute the dense features. Note that other image sizes are also compatible, as the preprocessing step will resample the input image to 224 × 224.

(2) Provide the key and copyright information to ZWNet to generate the final zero-watermark.

(3) Register the zero-watermark and record the key and copyright information.

If an image with an uncertain copyright is encountered, the copyright can be identified through the following steps:

(1) Input the suspected copyright image into ZWNet to generate the dense feature.

(2) Select the corresponding recorded key and copyright to generate the verification zero-watermark.

(3) Calculate the normalized correlation coefficient (NC) to assess the similarity between the original zero-watermark and the verification zero-watermark. NC is calculated as follows:

$$\mathrm{NC} = \frac{\sum_{i=1}^{N} W_i W\prime_i}{\sqrt{\sum_{i=1}^{N} W_i^2}\sqrt{\sum_{i=1}^{N} W\prime_i^2}}$$

Here, W, $W\prime$, and N represent the original zero-watermark sequence, the verification zero-watermark sequence, and the sequence length.

(4) If the NC value exceeds a predefined threshold, it indicates that the copyright of the image matches the registered copyright information. Otherwise, the image does not belong to the recorded copyright.

4. Experimental Results and Analysis

4.1. ZWNet Training

For this experiment, ZWNet is implemented using PaddlePaddle (https://github.com/PaddlePaddle/Paddle (accessed on 30 July 2023)) and executed on the PaddlePaddle AI Studio (https://aistudio.baidu.com (accessed on 30 July 2023)) with cloud computation powered by Nvidia A100. The optimizer employed is Adamax, and a Step Decay strategy is applied to the learning rate. The initial learning rate is set to 0.0001, with a gamma factor of 0.8. The learning rate is adjusted for each epoch.

The training dataset comprises a selection of images from mini-ImageNet (https://www.kaggle.com/datasets/arjunashok33/miniimagenet (accessed on 30 July 2023)) and additional images collected from the internet. The images are selected to train ZWNet with varied image features, aiming to improve its generalization ability. The training dataset consists of 2000 images; a subset of these images is displayed in Figure 8 for visualization.

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

Figure 8. A few examples of training datasets. (**a**) Car. (**b**) Lake. (**c**) Panda. (**d**) Koala.

We employed data augmentation as a preprocessing step to expand the training dataset and enhance ZWNet's robustness. The image processing methods used in data augmentation are detailed in Table 1. These methods encompass a mix of techniques, and following data augmentation, the number of training images increased to 100,000.

Table 1. Methods of data augmentation.

Methods	Method Description
Noise	Includes white noise and salt-and-pepper noise
Filter	Includes Average filter, median filter and Gaussian filter
Rotation	Rotates the image around the center with different angles
Crop	Crop out part of the image with different sizes
Mirror	Horizontal mirror and vertical mirror

When training ZWNet, two principles were adhered to, as mentioned in Section 3.5. First, an image and its augmented versions were assigned the same image identifier. Second, different images (including their augmented versions) were allocated different identifiers. We used an auto-incrementing number as the image identifier for simplicity.

To assess ZWNet's performance, we employed a test dataset consisting of one hundred images. Four of these test images are presented in Figure 9 for visualization. It is noted that none of these images are in the training dataset.

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

Figure 9. Four test images. (**a**) Lena. (**b**) Mandrill. (**c**) Tree. (**d**) Girl.

The training process involved updating ZWNet using the training data for each epoch and then evaluating the loss with the test images. If the loss on the test data no longer decreased or even began to increase, the training was terminated. Consequently, the test data was solely used to verify whether the training process was sufficient and did not impact the updating of ZWNet. Furthermore, once successfully trained, ZWNet remained stable and deployable. It could process arbitrary images for zero-watermark service without the need for retraining or adjustments. The changes in loss during the training stage are illustrated in Figure 10.

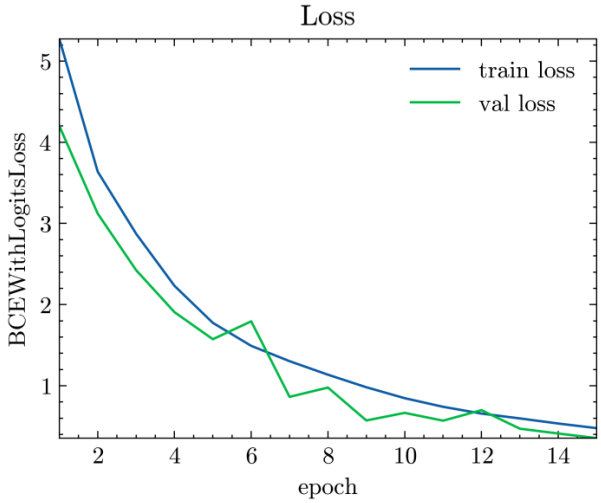

Figure 10. Loss changes in the training stage.

4.2. Robustness

Robustness is a critical feature of digital watermarking technology. We assess ZWNet's robustness by comparing the zero-watermark of the original image with the zero-watermark

of the attacked image. The evaluation index employed is NC, as explained in Section 3.6, with a threshold set to 0.8. The attack methods used are consistent with those applied in data augmentation. Using the image Lena (Figure 9a) as an example, the visuals of the original image and the attacked image are displayed in Figure 11.

Figure 11. Examples of images under different attacks. (**a**) Original image. (**b**) Salt and pepper noise. (**c**) Horizontal mirror. (**d**) Rotation. (**e**) Gaussian filter. (**f**) Median filter. (**g**) Average filter. (**h**) Crop.

The NC results of the images in Figure 9 under different attacks are listed in Table 2.

Table 2. NC results of four test images under different attacks.

Attack Description	Lena	Mandril	Tree	Girl
Rotation (15°)	0.9688	0.9609	0.9765	0.9688
Rotation (30°)	0.9258	0.9297	0.9258	0.9375
Rotation (45°)	0.8555	0.9063	0.8320	0.8203
Pepper and salt noise (intensity = 0.01)	0.9922	0.9961	0.9727	0.9805
Pepper and salt noise (intensity = 0.05)	0.9531	0.9609	0.9531	0.9375
Pepper and salt noise (intensity = 0.1)	0.9219	0.9336	0.9609	0.8750
Gaussian noise (mean = 0, variance = 0.005)	1	0.9843	0.8828	0.9570
Random crop (1/8)	0.9063	1	1	0.9336
Random crop (1/6)	1	0.8945	0.8164	0.8984
Random crop (1/4)	0.8633	0.8984	0.8203	0.8710
Crop upper-left corner (1/4)	0.9414	0.8750	0.9570	0.8086
Crop lower-left corner (1/4)	0.9922	0.9258	0.9531	0.8867
Crop upper-right corner (1/4)	0.8945	0.9531	0.9375	0.8789
Crop lower-right corner (1/4)	0.9883	0.9609	0.9414	0.8086
Crop upper-left corner (1/8)	0.9961	0.9219	0.9805	0.9297
Crop lower-left corner (1/8)	1	0.9766	1	0.9922
Crop upper-right corner (1/8)	0.9882	0.9922	0.9883	0.9453
Crop lower-right corner (1/8)	1	0.9727	1	0.9453
Blur (3 × 3)	1	0.9883	0.9922	0.9882
Blur (5 × 5)	1	0.9883	0.9922	0.9922
Blur (9 × 9)	1	0.9766	0.9922	0.9883
Blur (11 × 11)	0.9961	0.9805	0.9609	0.9883
Gaussian Blur (3 × 3)	1	0.9883	0.9922	0.9883
Gaussian Blur (5 × 5)	0.9961	0.9727	0.9609	0.9844
Gaussian Blur (9 × 9)	0.9844	0.9766	0.9492	0.9766
Gaussian Blur (11 × 11)	0.9922	0.9844	0.9609	0.9766
Median Blur (3 × 3)	0.9922	0.9688	0.9063	0.9688
Median Blur (5 × 5)	0.9688	0.9648	0.9258	0.9688
Median Blur (9 × 9)	0.9883	0.9805	0.9570	0.9805
Median Blur (11 × 11)	0.9766	0.9648	0.8984	0.9688

The results in Table 2 clearly indicate that the test images, even when subjected to different types and intensities of attacks, all exhibit NC values above the 0.8 threshold. These results demonstrate the robustness of the proposed scheme. The robustness can be attributed to two key factors: the utilization of ConvNeXt combined with LK-PAN and the effective training strategy for image identifiers.

Beyond the numerical results, it is essential to address overfitting when evaluating the effectiveness of a neural network. Overfitting occurs when a network memorizes the training data instead of learning the target function. In the context of zero-watermarking, overfitting could lead to a network that memorizes images under various attacks instead of extracting robust features. However, ZWNet effectively avoids overfitting. This is primarily due to the strict separation of training and testing images. During the training stage, ZWNet has not been exposed to the four test images shown in Figure 9, preventing it from memorizing these images based on prior knowledge. Therefore, it is evident that ZWNet has successfully learned to extract robust image features rather than merely memorizing them.

4.3. Discriminability

Discriminability is a crucial aspect of a zero-watermarking algorithm, ensuring that different images generate distinct zero-watermarks, especially when copyright identifiers are the same. In ZWNet's training, we assessed the similarity between images of Koala and Panda in the evaluation mode and observed the changes in the NC values, as depicted in Figure 12.

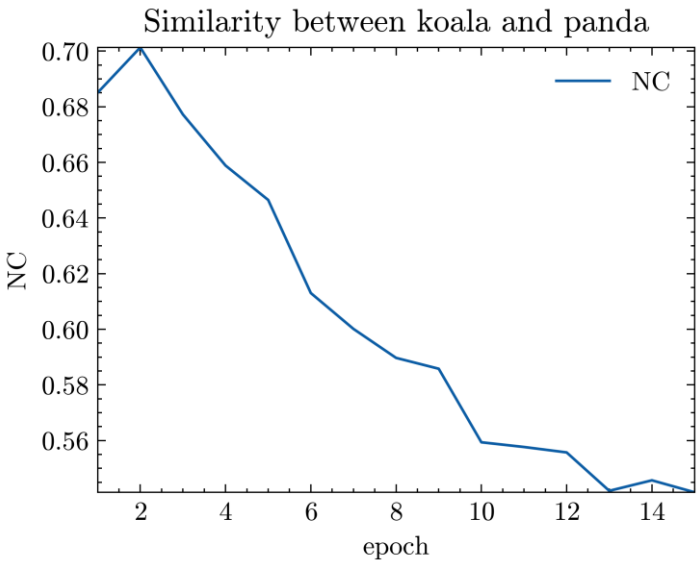

Figure 12. Similarity changes of test images in the training stage.

From Figure 12, it is evident that the NC value decreases from around 0.70 as the training epoch increases. After the 14th epoch, the NC value drops to approximately 0.54. Considering the NC threshold is set at 0.8, a value of 0.54 is relatively low, indicating that the zero-watermarks are dissimilar and the copyright will not be confused.

To conduct a more precise assessment of ZWNet's discriminability, we used the Hamming distance to compare zero-watermarks generated by different images with the same copyright. The Hamming distance is calculated as follows:

$$dist(A, B) = \sum_{i=0}^{n} A[i] \oplus B[i]$$

Here, A and B represent two zero-watermark sequences, n is the sequence length (which is 256 for ZWNet's watermark block) and $\oplus$ represents the exclusive OR operation. Hamming distances among the zero-watermarks of the four test images are detailed in Table 3.

Table 3. Hamming distances between the zero-watermarks of four test images.

Test Image	Lena	Mandril	Tree	Girl
Lena	0	90	88	113
Mandril	90	0	92	97
Tree	88	92	0	91
Girl	113	97	91	0

Table 3 displays the Hamming distances among the zero-watermarks of the four test images. The high Hamming distances indicate that the zero-watermarks differ significantly from one another, with more than 80 different bits. Given that the total length of the zero-watermark is 256, this observation suggests that the zero-watermarks of each image are substantially distinct from the others. Hence, the discriminability of ZWNet is substantiated.

Two factors contribute to this discriminability. Firstly, using LK-PAN within ZWNet helps extract local features and fuse them with the global context during feature map generation, as described in Section 2.2. Secondly, incorporating unique identifiers in the watermark block plays a crucial role. During the training phase, ZWNet is trained to extract robust features, assigning the same identifier to identical images and different identifiers to distinct images. As a result, ZWNet strives to produce dissimilar feature maps for different inputs, resulting in its discriminative capability. Importantly, this discriminative feature is not a result of overfitting since, as previously mentioned, ZWNet has not seen the test images during training.

4.4. Comparisons with Existing Methods

To provide a more objective evaluation of ZWNet's performance, we have chosen to compare it with three other zero-watermarking methods. The first method, named Yang's method, is a classical zero-watermarking technique that utilizes FQGPCET and has been recognized for its robustness and discriminability [20]. The second method, Liu's method, is an end-to-end neural network-based approach centered on style transfer and removal, renowned for its robustness and is named after its creator, Liu [30]. The third method, referred to as Nawaz's method, is a hybrid scheme that combines DWT and ResNet-101 together [26]. These three methods will be assessed alongside ZWNet in terms of their robustness, discriminability, and efficiency.

4.4.1. Robustness

To compare the robustness of ZWNet with Yang's method, Liu's method, and Nawaz's method, we conducted tests using the same test image, Lena, and subjected them to identical attacks. The results are summarized in Table 4.

From Table 4, it is evident that under the same attack conditions, ZWNet exhibits higher NC values compared to the other methods in most cases. It excels in robustness, with only a slight decrease in performance under rotation attacks. This suggests that the features extracted by the convolutional layer may not be highly robust when it comes to rotation attacks. However, ZWNet still achieves a substantial NC value greater than 0.9 in this scenario, which is more than adequate for copyright identification.

Table 4. NC results of the comparison methods.

Attack Description	ZWNet (Proposed)	Yang's Method	Liu's Method	Nawaz's Method
Rotation (15°)	0.9688	0.9943	0.9466	0.9247
Rotation (30°)	0.9258	1	0.9466	0.9058
Pepper and salt noise (intensity = 0.01)	0.9922	0.9375	0.9766	0.8935
Gaussian noise (mean = 0, variance = 0.005)	1	0.9531	1	0.9340
Random crop (1/6)	1	0.9023	0.9522	0.8706
Blur (3 × 3)	1	0.9414	0.9766	0.9172
Blur (11 × 11)	0.9961	0.9414	0.9302	0.8388
Gaussian blur (3 × 3)	1	0.9063	1	0.8902
Gaussian blur (11 × 11)	0.9922	0.9375	0.9766	0.8253
Median blur (3 × 3)	0.9922	0.9297	0.9961	0.9049
Median blur (11 × 11)	0.9766	0.9648	0.9102	0.8138

4.4.2. Discriminability

We utilized the four test images from Figure 9, generating zero-watermarks with identical copyrights using the comparison methods. To assess the differences in these zero-watermarks, we employed Hamming distance, and the results are summarized in Table 5.

Table 5. Hamming distance of comparison methods.

Hamming Distance of Zero-Watermarks	ZWNet (Proposed)	Yang's Method	Liu's Method	Nawaz's Method
Lena and Mandril	90	28	59	43
Lena and Tree	88	29	26	35
Lena and Girl	113	40	90	93
Mandril and Tree	92	21	61	66
Mandril and Girl	97	20	57	21
Tree and Girl	91	37	62	55

From Table 5, when calculating the Hamming distance among the test images, ZWNet exhibits higher values compared to the other comparison methods. This result demonstrates the excellent discriminability of the proposed ZWNet.

4.4.3. Efficiency

While enhancing efficiency was not the primary focus of our study, we conducted an efficiency assessment as it is a crucial consideration for practical usage. For the same set of test images, we ran each of the three zero-watermarking methods 10 times and calculated the average processing time. The efficiency results for these methods are presented in Table 6.

Table 6. Efficiency comparison.

Methods	ZWNet (Proposed)	Yang's Method	Liu's Method	Nawaz's Method
Average cost time	96 ms	2100 ms	2440 ms	1384 ms

The efficiency comparison presented in Table 6 clearly shows that the time required to generate a single zero-watermark with ZWNet is significantly lower than that of the other two methods. Yang's method appears to be less efficient than ZWNet due to the utilization of the CPU for computation without GPU acceleration. In fact, classical zero-

watermarking methods are generally impractical to run on GPUs, as many of their steps cannot be efficiently implemented with tensor operations.

In the case of Liu's method, although both it and ZWNet are based on neural networks and can be accelerated by GPUs, Liu's method involves learning different style features for various host images. This learning process includes training and multiple epochs, making it more time-consuming compared to ZWNet. In contrast, ZWNet can produce a zero-watermark directly for different images without the need for retraining by passing the host image through the ZWNet's layers only once.

5. Conclusions

This paper introduced an end-to-end zero-watermarking approach built on neural networks, which has practical applicability in scenarios such as image copyright registration, copyright authentication, and piracy detection. In contrast to traditional approaches that rely on handcrafted features, our methodology employs pure neural networks to learn robust features automatically. The structure of ZWNet consists of ConvNeXt and LK-PAN as the backbone and neck, respectively. Furthermore, we introduced the watermark block as the head component, transforming the challenge of enhancing robustness and discriminability into a multi-label classification task based on image identifiers. The experimental results clearly demonstrate that ZWNet effectively extracts resilient image features and generates zero-watermarks without the need for retraining. Moreover, ZWNet exhibits superior robustness, discriminability, and efficiency compared with existing methods. The results suggest that through the implementation of the proposed training strategy on image identifiers, the zero-watermark performance has been notably enhanced in terms of both robustness and discriminability simultaneously.

Author Contributions: Conceptualization, C.L. and D.T.; Data curation, S.C. and X.L.; Formal analysis, C.L. and D.T.; Funding acquisition, N.R. and D.T.; Investigation, X.L. and Y.Z.; Methodology, C.L. and H.S.; Project administration, D.T.; Resources, H.S. and N.R.; Software, C.L. and H.S.; Supervision, D.T.; Validation, C.W. and S.C.; Visualization, H.S. and C.W.; Writing—original draft, C.L., H.S. and C.W.; Writing—review and editing, N.R. and D.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Jiangsu Province, grant number BK20200839, the National Natural Science Foundation of China, grant numbers 42301484 and 42071362, and the Open Topic of Hunan Engineering Research Center of Geographic Information Security and Application, grant number HNGISA2023004.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset of mini-ImageNet is available at https://www.kaggle.com/datasets/arjunashok33/miniimagenet (accessed on 30 July 2023).

Conflicts of Interest: Author Na Ren was employed by the company Nanjing Geomarking Information Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Costa, G.; Degano, P.; Galletta, L.; Soderi, S. Formally verifying security protocols built on watermarking and jamming. *Comput. Secur.* **2023**, *128*, 103133. [CrossRef]
2. Razaq, A.; Alhamzi, G.; Abbas, S.; Ahmsad, M.; Razzaque, A. Secure communication through reliable S-box design: A proposed approach using coset graphs and matrix operations. *Heliyon* **2023**, *9*, e15902. [CrossRef] [PubMed]
3. Tao, H.; Chongmin, L.; Zain, J.M.; Abdalla, A.N. Robust Image Watermarking Theories and Techniques: A Review. *J. Appl. Res. Technol.* **2014**, *12*, 122–138. [CrossRef]

4. Liu, X.; Wang, Y.; Sun, Z.; Wang, L.; Zhao, R.; Zhu, Y.; Zou, B.; Zhao, Y.; Fang, H. Robust and discriminative zero-watermark scheme based on invariant features and similarity-based retrieval to protect large-scale DIBR 3D videos. *Inf. Sci.* **2021**, *542*, 263–285. [CrossRef]
5. Xia, Z.; Wang, X.; Han, B.; Li, Q.; Wang, X.; Wang, C.; Zhao, T. Color image triple zero-watermarking using decimal-order polar harmonic transforms and chaotic system. *Signal Process.* **2021**, *180*, 107864. [CrossRef]
6. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
7. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
9. Dong, S.; Wang, P.; Abbas, K. A survey on deep learning and its applications. *Comput. Sci. Rev.* **2021**, *40*, 100379. [CrossRef]
10. Gao, S.H.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P. Res2Net: A New Multi-Scale Backbone Architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 652–662. [CrossRef]
11. Ma, J.; Jiang, X.; Fan, A.; Jiang, J.; Yan, J. Image Matching from Handcrafted to Deep Features: A Survey. *Int. J. Comput. Vis.* **2021**, *129*, 23–79. [CrossRef]
12. Garcia-Garcia, B.; Bouwmans, T.; Silva, A.J.R. Background subtraction in real applications: Challenges, current models and future directions. *Comput. Sci. Rev.* **2020**, *35*, 100204. [CrossRef]
13. Taghanaki, S.A.; Abhishek, K.; Cohen, J.P.; Cohen-Adad, J.; Hamarneh, G. Deep semantic segmentation of natural and medical images: A review. *Artif. Intell. Rev.* **2021**, *54*, 137–178. [CrossRef]
14. Cheng, G.; Xie, X.; Han, J.; Guo, L.; Xia, G.S. Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 3735–3756. [CrossRef]
15. Wen, Q.; Sun, T.; Wang, A. Concept and Application of Zero-Watermark. *Acta Electron. Sin.* **2003**, *31*, 214–216.
16. Jiang, F.; Gao, T.; Li, D. A robust zero-watermarking algorithm for color image based on tensor mode expansion. *Multimedia Tools Appl.* **2020**, *79*, 7599–7614. [CrossRef]
17. Dong, F.; Li, J.; Bhatti, U.A.; Liu, J.; Chen, Y.W.; Li, D. Robust Zero Watermarking Algorithm for Medical Images Based on Improved NasNet-Mobile and DCT. *Electronics* **2023**, *12*, 3444. [CrossRef]
18. Kang, X.-B.; Lin, G.-F.; Chen, Y.-J.; Zhao, F.; Zhang, E.-H.; Jing, C.-N. Robust and secure zero-watermarking algorithm for color images based on majority voting pattern and hyper-chaotic encryption. *Multimedia Tools Appl.* **2020**, *79*, 1169–1202. [CrossRef]
19. Chu, R.; Zhang, S.; Mou, J.; Gao, X. A zero-watermarking for color image based on LWT-SVD and chaotic system. *Multimedia Tools Appl.* **2023**, *82*, 34565–34588. [CrossRef]
20. Yang, H.-Y.; Qi, S.-R.; Niu, P.-P.; Wang, X.-Y. Color image zero-watermarking based on fast quaternion generic polar complex exponential transform. *Signal Process. Image Commun.* **2020**, *82*, 115747. [CrossRef]
21. Leng, X.; Xiao, J.; Wang, Y. *A Robust Image Zero-Watermarking Algorithm Based on DWT and PCA*; Springer Berlin Heidelberg: Berlin, Heidelberg, 2012.
22. Singh, A.; Dutta, M.K. A robust zero-watermarking scheme for tele-ophthalmological applications. *J. King Saud Univ.—Comput. Inf. Sci.* **2020**, *32*, 895–908. [CrossRef]
23. Zhong, X.; Huang, P.-C.; Mastorakis, S.; Shih, F.Y. An Automated and Robust Image Watermarking Scheme Based on Deep Neural Networks. *IEEE Trans. Multimedia* **2021**, *23*, 1951–1961. [CrossRef]
24. Mahapatra, D.; Amrit, P.; Singh, O.P.; Singh, A.K.; Agrawal, A.K. Autoencoder-convolutional neural network-based embedding and extraction model for image watermarking. *J. Electron. Imaging* **2022**, *32*, 021604. [CrossRef]
25. Dhaya, D. Light Weight CNN based robust image watermarking scheme for security. *J. Inf. Technol. Digit. World* **2021**, *3*, 118–132. [CrossRef]
26. Nawaz, S.A.; Li, J.; Shoukat, M.U.; Bhatti, U.A.; Raza, M.A. Hybrid medical image zero watermarking via discrete wavelet transform-ResNet101 and discrete cosine transform. *Comput. Electr. Eng.* **2023**, *112*, 108985. [CrossRef]
27. Fierro-Radilla, A.; Nakano-Miyatake, M.; Cedillo-Hernandez, M.; Cleofas-Sanchez, L.; Perez-Meana, H. A Robust Image Zero-watermarking using Convolutional Neural Networks. In Proceedings of the 2019 7th International Workshop on Biometrics and Forensics (IWBF), Cancun, Mexico, 2–3 May 2019; pp. 1–5.
28. Han, B.; Du, J.; Jia, Y.; Zhu, H. Zero-Watermarking Algorithm for Medical Image Based on VGG19 Deep Convolution Neural Network. *J. Health Eng.* **2021**, *2021*, 5551520. [CrossRef] [PubMed]
29. Gong, C.; Liu, J.; Gong, M.; Li, J.; Bhatti, U.A.; Ma, J. Robust medical zero-watermarking algorithm based on Residual-DenseNet. *IET Biom.* **2022**, *11*, 547–556. [CrossRef]
30. Liu, G.; Xiang, R.; Liu, J.; Pan, R.; Zhang, Z. An invisible and robust watermarking scheme using convolutional neural networks. *Expert Syst. Appl.* **2022**, *210*, 118529. [CrossRef]
31. Li, H.; Xiong, P.; An, J.; Wang, L. Pyramid Attention Network for Semantic Segmentation. *arXiv* **2018**, arXiv:1805.10180.
32. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
33. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, Canada, 11–17 October 2021; pp. 10012–10022.

34. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
35. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
36. Li, C.; Liu, W.; Guo, R.; Yin, X.; Jiang, K.; Du, Y.; Du, Y.; Zhu, L.; Lai, B.; Hu, X.; et al. PP-OCRv3: More Attempts for the Improvement of Ultra Lightweight OCR System. *arXiv* **2022**, arXiv:2206.03001.