Computers, Materials & Continua
DOI: 10.32604/cmc.2023.034748
Article

Tech Science Press

LKAW: A Robust Watermarking Method Based on Large Kernel Convolution and Adaptive Weight Assignment

Xiaorui Zhang[1,2,3,*], Rui Jiang[1], Wei Sun[3,4], Aiguo Song[5], Xindong Wei[6] and Ruohan Meng[7]

[1]Engineering Research Center of Digital Forensics, Ministry of Education, Jiangsu Engineering Center of Network Monitoring, School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[2]Wuxi Research Institute, Nanjing University of Information Science & Technology, Wuxi, 214100, China
[3]Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science & Technology, Nanjing, 210044, China
[4]School of Automation, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[5]School of Instrument Science and Engineering, Southeast University, Nanjing, 211189, China
[6]School of Teacher Education, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[7]School of Computer Science Engineering, Nanyang Technological University, Singapore
*Corresponding Author: Xiaorui Zhang. Email: zxr365@126.com
Received: 26 July 2022; Accepted: 20 October 2022

Abstract: Robust watermarking requires finding invariant features under multiple attacks to ensure correct extraction. Deep learning has extremely powerful in extracting features, and watermarking algorithms based on deep learning have attracted widespread attention. Most existing methods use 3×3 small kernel convolution to extract image features and embed the watermarking. However, the effective perception fields for small kernel convolution are extremely confined, so the pixels that each watermarking can affect are restricted, thus limiting the performance of the watermarking. To address these problems, we propose a watermarking network based on large kernel convolution and adaptive weight assignment for loss functions. It uses large-kernel depth-wise convolution to extract features for learning large-scale image information and subsequently projects the watermarking into a high-dimensional space by 1×1 convolution to achieve adaptability in the channel dimension. Subsequently, the modification of the embedded watermarking on the cover image is extended to more pixels. Because the magnitude and convergence rates of each loss function are different, an adaptive loss weight assignment strategy is proposed to make the weights participate in the network training together and adjust the weight dynamically. Further, a high-frequency wavelet loss is proposed, by which the watermarking is restricted to only the low-frequency wavelet sub-bands, thereby enhancing the robustness of watermarking against image compression. The experimental results show that the peak signal-to-noise ratio (PSNR) of the encoded image reaches 40.12, the structural similarity (SSIM) reaches 0.9721, and the watermarking has good robustness against various types of noise.

Keywords: Robust watermarking; large kernel convolution; adaptive loss weights; high-frequency wavelet loss; deep learning

1 Introduction

The task of robust watermarking for digital images is to secretly embed watermarking into the cover image and to extract it correctly after malicious or unintentional attacks. In recent years, digital image watermarking technology has been widely studied in copyright protection, traceability, anti-counterfeiting, and authentication of 2D images. Many researchers have proposed various methodological strategies that balance the imperceptibility and robustness of watermarking well.

Deep learning [1] uses neural networks' powerful feature extraction ability to extract image features while adaptively embedding the watermarking to obtain better results. Zhu et al. [2] proposed an auto-encoder-based neural network for watermarking, which embeds the watermarking into the cover image through an encoder, and then a decoder to parse out the watermarking. Using a noise layer to simulate the attacks on the encoded image and adding a Generative Adversarial Network (GAN) to reinforce the imperceptibility, thus affecting the process of watermarking embedding and extraction. Luo et al. [3] further proposed using the idea of adversarial training instead of a specific noise layer to extend the distortion to an unknown range. Although they have achieved good results [4], these methods all use small kernel convolutional networks, and the continuous accumulation of a mass of small convolutional kernels will deepen the neural network and lead to the degradation problem of deep neural networks, which makes the difficulty of optimization significantly increase. Although ResNet [5] eases the optimization problem by skip connections to some extent, the number of layers in the network is too large, resulting in substantial training parameters. Further studies [6–8] found that deep neural networks stacked with multiple small convolutional kernels do not obtain larger effective perception fields, and the last convolutional layer only can see an intensely limited range of the original image, as a result of which deep convolutional neural networks (CNNs) [9,10] only learn features in a small area. Global information on the image is extremely scarce. Applying this to watermarking methods will result in fewer pixels associated with the extracted features, making watermarking more modifiable for the cover image and fewer regions available for watermarking embedding. In contrast, large kernel convolution [7] inherently has a large range of effective perception fields, even in the last layer of the deep network, which allows the network to learn features to approximate the whole image. This is especially crucial for watermarking. Because embedding the watermarking that is according to the large kernel convolution changes the cover image, which will be reflected in more pixels. When the encoded image is attacked, there will still be enough pixels that retain the watermarking information for extraction.

In addition, considering imperceptibility and robustness, our method needs to accomplish two tasks: (1) The encoded image should be as similar as possible to the cover image and indistinguishable from the human eye. (2) The decoded watermarking should be as identical as possible to the input watermarking. For this purpose, the watermarking network uses multiple losses to guide the network training. However, the previous multitask approaches [2,3] usually use a naive weighted sum of losses, with fixed loss weights set by manual mode. The performance of the model is heavily dependent on the choice of weights. Determining an optimal weight by manual tuning is also extremely expensive. Such a mode using fixed weights does not yield the best training results due to the varying speed of convergence of each loss [11].

Image compression is a prevalent attack in the field of watermarking. How to improve the robustness of watermarking against image compression is extremely important. Image compression

usually removes the texture and edge parts of an image that are not sensitive to the human visual system, to reduce the image volume. The texture and edge parts of the image are usually reflected as high-frequency components in the frequency domain, such as the HH sub-band in the wavelet domain. If the watermarking is embedded in the high-frequency component of the image, the watermarking will be removed and difficult to extract once the image is compressed. Traditional discrete wavelet transform (DWT) and discrete cosine transform (DCT) watermarking are only embedded in the low-frequency components to avoid damage to the watermarking from image compression. In frequency domain watermarking based on deep learning, there is a similar need to embed the watermarking in the low-frequency components.

To solve the above challenge, this study proposes a wavelet domain watermarking network, named LKAW, based on large kernel convolution and adaptive weight assignment for loss functions. The large-kernel convolution is used to enhance the learning ability of our LKAW for a wide range of information and adjust the scale of features learned by the large kernel. In addition, an adaptive weight assignment strategy for loss functions is proposed in this paper. The weights are trained as network parameters, and their size is dynamically adjusted in the training process. Further, a high-frequency wavelet loss is proposed to keep the cover image and the encoded image to be as similar as possible in the high-frequency wavelet sub-bands, by which the watermarking can be embedded into the wavelet low-frequency sub-bands, thus improving the robustness of the watermarking against image compression.

In summary, the main contributions of this paper are summarized as follows:

- We introduce a novel watermarking network based on large kernel convolution. By integrating large-scale features with one 21×21 large kernel and two 1×1 small kernels, the features involving more pixels can be obtained, which makes the extent of the watermarking modification smaller, and recover the correct watermarking information from more pixels when attacked. Meanwhile, GAN is introduced to reduce the difference between the cover image and the encoded image, thus strengthening the imperceptibility of the watermarking.
- An adaptive weight assignment strategy for loss functions is designed to dynamically adjust each loss function's weight to obtain a better training effect and solve the problem of gradient disappearance. Furthermore, a high-frequency wavelet loss is proposed to ensure that our LKAW can adaptively embed the watermarking into the low-frequency sub-bands of the cover image after the wavelet transform, which prevents the watermarking from being corrupted by image compression.

The rest of this paper is structured as follows. Section 2 presents the work related to the model design. Section 3 shows the detailed architecture and loss functions of our proposed model. Section 4 gives the experimental results. Section 5 finally concludes the paper.

2 Related Work

This section dwells on the related work involved in our approach in terms of digital image watermarking, large kernel convolution, multitask learning, and wavelet-based watermarking.

2.1 Digital Image Watermarking

Digital image watermarking can effectively solve the problems of copyright protection, content authentication, and traceability. The basic idea is to use the embedding algorithm to embed the watermarking into the cover image and extract the watermarking by the extraction algorithm to prove

the copyright ownership. The traditional robust watermarking algorithm mainly adopts the manual method to embed the watermarking into the spatial domain or transform domain of the cover image. Still, it is often difficult to find the most suitable embedding areas, and the embedding capacity is shallow. With the development of deep learning, more neural networks based on an auto-encoder framework for watermarking have been proposed [12–15]. Zhong et al. [16] proposed a method to encode the watermarking and then embed it into the feature map of the cover image. Finally, using a neural network to preserve the watermarking and reject noise after converting the encoded image into a high-dimensional transform space, they achieve the robustness of watermarking without a priori knowledge of image distortion. Although a multi-branch small kernel convolution is used to extract multiple features of the image in parallel, it does not consider extracting features with larger effective perception fields to embed the watermarking. Liu et al. [17] proposed a two-stage watermarking training method: noise-free encoder-decoder training and noise-decoder training. It circumvents the requirement that the noise must be differentiable and enhances the robustness of watermarking against undifferentiable noise. But the two-stage training is essentially a greedy algorithm, and the optimal solution in each stage is not necessarily the optimal global solution, which limits the watermarking performance. Previous studies have shown that neural networks have great potential for digital image watermarking.

2.2 Large Kernel Convolution

Recently, CNNs is challenged by the vision transformers [18,19] in computer vision. In many visual downstream tasks, such as semantic segmentation and object detection, the vision transformers performs better [20,21]. Previously, it was often thought that the multi-head self-attention of the vision transformers played a decisive role. It is less inductive bias, is more robust to distortions, and can model long-range dependencies [7]. Still, the use of simple spatial pooling or Multilayer Perceptron (MLP) instead of the self-attention also shows better performance, such as MetaFormer [22] and MLP-Mixer [23], which demonstrates that it may not be irreplaceable. Recent studies have shown that expanding of the effective perception fields due to the global relational modeling of the vision transformers may be the winning formula. Liu et al. [8] proposed a novel network ConvNext that modeled the design of Swin Transformer [17] architecture and introduced 7×7 large kernel convolutions. Only using convolutions, it exceeds the Swin Transformer of the same scale. Watermarking also requires extracting a wide range of image features to affect more image pixels, thus mitigating the magnitude of changes to each pixel to enhance imperceptibility. Ding et al. [7] proposed a network named RepLKNet, which uses 31×31 super large kernel convolution and performs similarly to Swin Transformer of the same scale in image classification. However, due to the more substantial shape bias learned, it outperforms CNNs in other tasks, such as object detection and semantic segmentation. Therefore, using features extracted from large kernel convolution and the shape bias learned to embed the watermarking, the watermarking can be extracted from more features between pixels and high-level semantic information of the image. Guo et al. [6] proposed a network named VAN that uses large kernel attention that both absorbs the advantages of convolution and self-attention to decompose the large kernel convolution into depth-wise convolution, depth-wise dilation convolution, and point-wise convolution. It achieves a large kernel convolution with lower computational cost while taking into account the importance of the channel dimension. It beats vision transformers and CNNs on several vision tasks [24,25]. These studies show that the perception fields improved by large kernel convolution can effectively enhance the performance of CNNs, and it is suitable for the watermarking domain. Therefore, introducing large kernel convolution into the watermarking domain can achieve better results.

2.3 Multitask Learning

The purpose of multitask learning is to learn multiple targets in a shared representation to improve learning efficiency and prediction accuracy [26]. Multitask learning is extremely active in computer vision because of its convenience and efficiency in solving multiple problems at the same time, as well as it benefits stronger generalization performance and better learning due to the information sharing across models that cannot be possessed by a single task [27,28]. Watermarking requires both imperceptibility and robustness. It is a typical multitasking system, but previous approaches [2,3,12–17,29,30] use a weighted sum of losses to balance the importance of individual tasks, and the loss weights are manual tuning. Liu et al. [31] proposed an end-to-end training model named MTAN that learns task-specific features from the globally shared representation while allowing features to be shared across tasks. It has a simple implementation and achieves better results. However, an additional network structure needs to be added, and loss weights are directly related to the learned features, which are not stable enough. Cipolla et al. [11] proposed that model performance is highly dependent on an appropriate choice of weighting between each task's loss and offers a principled way of combining multiple loss functions to learn multiple objectives using homoscedastic uncertainty. It obtains better performance compared to learning each task individually. The adaptive weight assignment for loss functions can learn to balance these weightings optimally, resulting in better imperceptibility and robustness.

2.4 Wavelet-Based Watermarking

The essence of the traditional frequency domain watermarking method to achieve better imperceptibility and robustness is that the watermarking can be well diffused into the cover image. Because the frequency domain coefficients are associated with almost every pixel of the cover image. Modifying the frequency domain coefficients will also influence almost virtually any pixel, reducing the amount of modification for each pixel to obtain a higher image quality. At the same time, the features of more pixels can be integrated into the extraction process, which is more robust to noise attacks. Traditional frequency domain watermarking algorithms mainly include DCT, DWT, and discrete Fourier transform (DFT). The discrete wavelet transform can decompose the image into four sub-bands: LL, LH, HL, and HH. The LL sub-band can continue to decompose to form a multi-scale wavelet domain so that different scale wavelet domains can be selected for watermarking embedding according to different needs, which can better adapt to the visual characteristics of the human eyes and obtain higher imperceptibility and robustness. In addition, it can avoid the block effect of DCT-based watermarking algorithms. Therefore, traditional frequency domain watermarking algorithms often choose to embed watermarks in the DWT domain. Most deep-learning-based watermarking methods [2,3,12–17,29] extract deep semantic features directly on the cover image by convolution and embed the watermarking adaptively based on the features to achieve the effect like frequency domain embedding, but the embedding effect depends on whether the learned features can be associated with almost all pixels, which is almost unfeasible for the small kernel convolution they use. Some deep-learning-based watermarking methods realize the advantage of embedding watermarks in the frequency domain. For example, Ahmadi et al. [30] used a 1×1 convolution to transform the cover image into an unknown frequency domain to learn one or more watermarking embedding methods and use circular convolution to spread the watermarking across multiple image blocks, thus having good superiority in imperceptibility, robustness, and speed. However, the experimental results show that the frequency domain learned by 1×1 convolution is not better than traditional methods' DCT or DWT domain. In the DWT domain, where traditional frequency domain watermarking algorithms have proven to

have excellent performance, it is more effective to embed watermarking using the powerful learning ability of deep learning.

3 Proposed Method

In this section, we propose a watermarking neural network based on large kernel convolution and adaptive weight assignment for loss functions that can extract image features in a larger range and find better watermarking embedding regions to achieve adaptive embedding of the watermarking. Meanwhile, the weights are dynamically adjusted and the watermarking is embedded into the low-frequency sub-bands in the DWT domain, which can further enhance the robustness of the watermarking against the image compression.

3.1 Network Architecture

Fig. 1 shows the architecture of the LKAW. Large kernel convolution usually refers to the convolution kernel with more than 3×3, and increasing the convolution kernel size can enlarge the effective perception fields to favorably learn the semantic shape information of objects instead of detailed texture information. Therefore, using the semantic segmentation network U-Net [32] based on the large kernel convolution enables the combination of deep features and spatial information and enhances the visual quality of the encoded images. The embedding loss, LPIPS perceptual loss [33] calculated by AlexNet [34], and adversarial loss based on WGAN-GP [35] encourage the encoder to embed watermarking imperceptibly. The decoder also uses the large kernel convolution to obtain more vital watermarking extraction ability and updates the decoder parameters by extracting loss. Meanwhile, to jointly guide the network to train for accomplishing both high imperceptibility and strong robustness of the watermarking, we propose an adaptive weight assignment strategy for loss functions to help converge better. Furthermore, traditional wavelet watermarking methods embed the watermarking in the wavelet low-frequency sub-bands to avoid the destruction of the watermarking by image compression. Therefore, high-frequency wavelet loss is proposed as a soft constraint to achieve this purpose for the wavelet watermarking method based on deep learning.

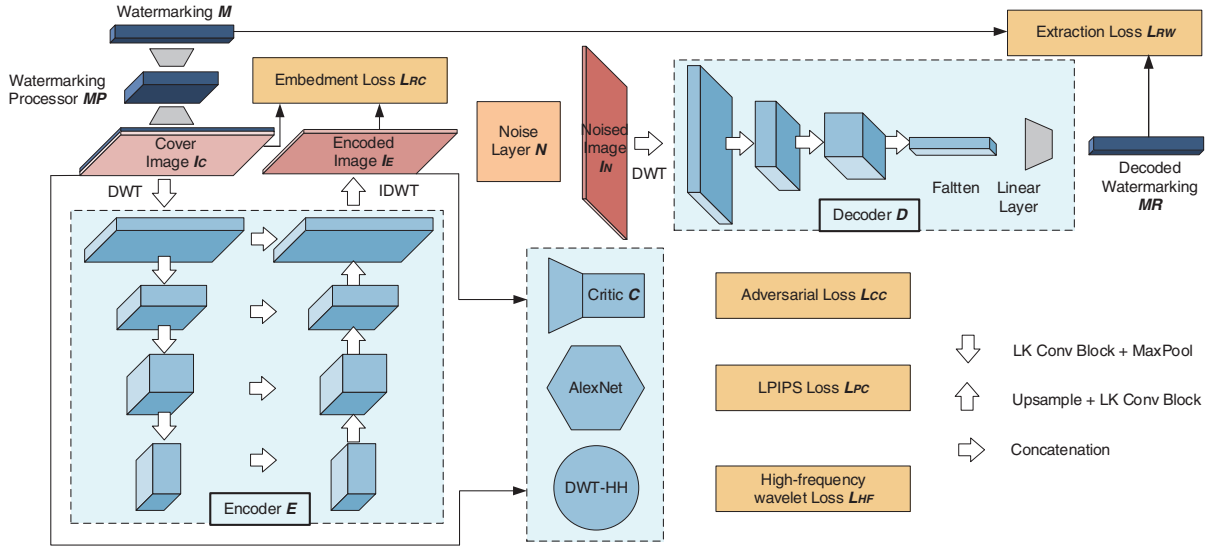

Figure 1: LKAW architecture

The watermarking processor uses transposed convolution and SENet [36] blocks to extract the watermarking features, and increase the watermarking redundancy, which can support the image encoding process. The noise layer adds a random type of noise to the cover image to form the noised image in each mini-batch, simulating possible noise attacks to the watermarking in reality and improving the extraction ability of the decoder.

3.2 Large Kernel Convolution Block

Fig. 2 shows the large kernel convolution block used, where DW-Conv represents depth-wise convolution, C_{mid} is designed to be four times larger than C_{in}, and Skip Connection contains a convolution of $1 \times 1 \times C_{out}$. The existing studies [16,25] show that the large effective perception fields brought by 21×21 large kernel convolution can find more reasonable embedding areas, but the large kernel convolution is slower and more expensive than the small kernel convolution. The depth-wise convolution optimized by implicit gemm algorithm [15] is used to overcome this problem. It accelerates the convolution computation by transforming the convolution process into implicit matrix multiplication and chunking the whole process to achieve parallelization. Moreover, depth-wise convolution adopts a convolution kernel for each feature channel. It can be regarded as multi-head self-attention in vision transformers, but the number of heads is the same as the number of channels, thus extracting image features from multiple dimensions. In the deep neural network, different channels often represent different features and objects, so it is overwhelmingly crucial to use two 1×1 convolutions to strengthen the adaptability of the channel dimension, which is conducive to learning the features with attack invariance. Among them, the first 1×1 convolution increases the number of channels, which can effectively represent the semantic representations. At the same time, it can also capture the relationship between the channel dimensions of the feature map. The second 1×1 convolution reduces the channel of features, which meets the requirements of the output dimension.

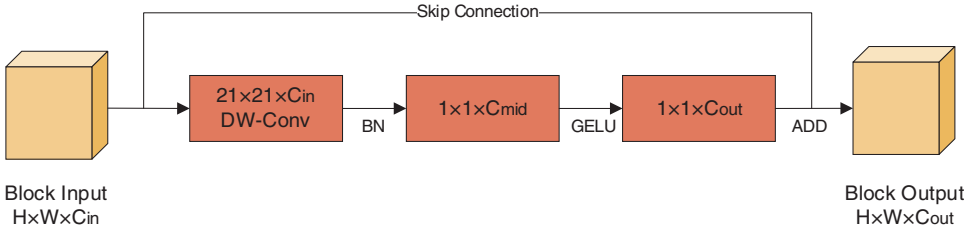

Figure 2: Large kernel convolution block

The combination of large kernel convolution and 1×1 convolution decomposes the dense convolution into a spatial representation extraction process and a channel representation extraction process, which reduces the floating-point operations (FLOPs) of the network and improves the efficiency of the model. It also reduces the parameters of the network, which plays a paramount role in preventing over-fitting. In addition, the skip connection is adopted to ensure the reusability of features and avoid the degeneration of the network. Because the number of input channels is inconsistent with that of the output channel, a 1×1 convolution is added to ensure consistency in the number of channels in the skip connection.

3.3 Adaptive Weight Loss Function

The adaptive weight loss function consists of a combination of five losses, but the size scale and convergence speed of each loss varies, and it is arduous to balance by using fixed weights. Inspired by

[17], an adaptive weight assignment strategy for loss functions is used, wherein five weights are input to the network and trained together. The last term acts as a correction term to prevent the trained weight parameters from being too large.

$$L = \frac{1}{2\log \lambda_{RC}{}^2} L_{RC} + \frac{1}{\log \lambda_{RW}{}^2} L_{RW} + \frac{1}{2\log \lambda_{PC}{}^2} L_{PC} + \frac{1}{2\log \lambda_{CC}{}^2} L_{CC} + \frac{1}{2\log \lambda_{HF}{}^2} L_{HF} + \log\left(1+\lambda_{RC}\right)\left(1+\lambda_{RW}\right)\left(1+\lambda_{PC}\right)\left(1+\lambda_{CC}\right)\left(1+\lambda_{HF}\right) \quad (1)$$

where λ_{RC}, λ_{RW}, λ_{PC}, λ_{CC}, and λ_{HF} are the weights of different losses, which are adaptively adjusted according to the image data and participate in network training, thus dynamically balancing the imperceptibility and robustness of the watermarking. Here L_{RC}, L_{RW}, L_{PC}, L_{CC}, and L_{HF} are 5 different losses.

High-frequency wavelet loss: To avoid the watermarking being destroyed by image compression, we propose a high-frequency wavelet loss L_{HF}, which embeds the watermarking in the low-frequency sub-bands of the cover image to enhance the robustness to image compression.

$$L_{HF}\left(\theta_E\right) = \frac{1}{N}\sum_{n=1}^{N} \left\| DWT\left(I_E\right)_{HH} - DWT\left(I_C\right)_{HH} \right\|_2^2 \quad (2)$$

where θ_E is the parameters of the encoder, N represents the total number of training samples, $DWT\left(\cdot\right)_{HH}$ is the operation of extracting high-frequency sub-bands after DWT, and 2-norm is adopted to represent the difference between I_E and I_C in the high-frequency sub-bands.

Embedding loss: The purpose of invisible watermarking embedding is to make the encoded image I_E indistinguishable from the cover image I_C, and it is necessary to minimize the difference between the corresponding pixels of them, so the embedding loss L_{RC} is adopted as follows:

$$L_{RC}\left(\theta_E\right) = \frac{1}{N}\sum_{n=1}^{N} \left\| I_E - I_C \right\|_2^2 \quad (3)$$

where θ_E is the parameters of the encoder, 2-norm is adopted to represent the difference between I_E and I_C.

Extracting loss: In the process of watermarking extraction, the network should be able to extract the watermarking from the noise image I_N, and the difference between the decoded watermarking MR and the input watermarking M needs to be reduced. Therefore, the extracting loss L_{RW} is adopted as follows:

$$L_{RW}\left(\theta_D\right) = \frac{1}{N}\sum_{n=1}^{N} -\left[M\log\left(MR\right) + \left(1-M\right)\log\left(1-MR\right)\right] \quad (4)$$

where θ_D is the parameters of the decoder. The decoder usually is exploited to predict the binary watermarking, regarded as a binary classification problem, therefore, the binary cross entropy loss is adopted to measure the difference between MR and M.

LPIPS perceptual loss and adversarial loss: To minimize the difference between the encoded image and the cover image, LPIPS perceptual loss L_{PC} [33] and adversarial loss L_{CC} based on WGAN-GP [35] are used.

3.4 Algorithm Implementation

Algorithm 1 shows the pseudo-code of our LKAW training. For each mini-batch: (1) the cover image I_C of shape $H \times W \times 3$ is decomposed into wavelet sub-bands I_W of shape $H/4 \times W/4 \times 12$

by DWT. (2) The watermarking processor MP receives the binary watermark $M \in \{0,1\}^L$ with length L, and outputs the watermarking feature map $I_{LM} \in R^{\mathrm{H/4\times W/4\times 12}}$. (3) I_{LM} and I_C are concatenated and input into the encoder E with the parameters θ_E, and E products the encoded image I_E after the inverse discrete wavelet transform (IDWT). (4) Noise layer N receives I_E of shape $H \times W \times 3$, and randomly adds a type of noise to I_E, thus generating the noised image I_N of same shape. (5) The noised image I_N is also decomposed into wavelet sub-bands by DWT and then input to the decoder D with the parameters θ_D, to extract the watermarking $MR \in \{0,1\}^L$. (6) Critic C accepts image I_C and I_E, calculates the distance of data distribution between them with gradient penalty, and trains its parameters θ_C through loss function L_C; (7) AlexNet accepts I_C and I_E for calculating the LPIPS loss; (8) Update network parameters θ_E and θ_D by adaptive weight loss function L.

Algorithm 1: Pseudocode of LKAW

Input: I_C,M
Output: Trained networks E, D, C
Training Variables: θ_E, θ_D, θ_C

while epoch $<$ *max_epoch* **do**
  Compute $I_E = E(I_C, M)$
  for $i = 1$ to *num_iter* **do**
    Compute *Critic* (I_E, I_C)
    Update $\theta_C = \theta_C + lr \times Optim(L_C)$
  Compute $I_N = N(I_E)$, $MR = D(I_N)$
  Update $\theta_E = \theta_E + lr \times Optim(L)$
  Update $\theta_D = \theta_D + lr \times Optim(L)$

4 Experiments

To verify the performance of our proposed watermarking method, we compare it with state-of-the-art watermarking models. Also, we do relevant ablation experiments to manifest the effectiveness of the proposed method. In addition, some noise attack experiments are added to prove the robustness of the watermarking method.

4.1 Experimental Setup

Dataset and setting: We selected 10,000 images from the COCO data set [37] as the cover images for training, 1000 images from the COCO data set as the validation set, and 1000 high-resolution images from the DIV2K data set [38] as the test set. The initial values of the weights λ_{RC}, λ_{RW}, λ_{PC}, λ_{CC} and λ_{HF} of the loss function are all set to 1.0. The size of the mini-batch in training is 16, and the size of the mini-batch in testing is 1. The initial learning rate is 1×10^{-4}, and the total epoch of training is 400. The warmup strategy is used for 40 epochs, and then the cosine annealing strategy is used to reduce the learning rate to 0 gradually. The noise layer is set to randomly choose one kind of noise for each mini-batch of the encoded image as a combined noise layer. The length of the watermarking is 30 bits. Training is based on Pytorch 1.11.0 framework, and one NVIDIA RTX3080 is used for calculation. In gradient calculation, Ranger21 [39] is adopted, which integrates AdamW and LookAhead [40] to achieve better learning ability. Except for the last layer of the decoder, all activation functions adopt GELU, which can be regarded as a combination of dropout and RELU. Introducing randomness into the activation functions makes the model training more robust. The activation function of the last

layer of the decoder adopts the Sigmoid function so that the range of the output is limited to (0,1) and rounded off as the decoded watermarking.

Evaluation metrics: Four metrics are used to measure the quality of the encoded image: PSNR, SSIM [41], root mean square error (RMSE), and mean absolute error (MAE). They can reflect the similarity between the encoded image and the cover image, which are used to verify the imperceptibility of watermarking. Meanwhile, two metrics are used to test the extraction ability of watermarking, including bit error rate (BER) and normalized correlation (NC). BER calculates the percentage of error bits in the decoded watermarking, and NC calculates the similarity between the decoded watermarking and the input watermarking, all of which are used to verify the robustness of the watermarking. There are all calculated as the average on the test set.

Noise attacks: We utilize combined noise to train all watermarking models, which can enhance the robustness of the watermarking. Specifically, the combined noise includes Identity, Gaussian noise, Salt-and-pepper noise, Gaussian blur, Quantization, Median filtering, Crop, Cropout, Dropout, and JPEG compression. Dropout, Cropout, and Quantization are from HiDDeN [2]. JPEG compression includes non-differentiable real JPEG and JPEG-MASK [2] which is a differentiable noise layer and simulates the high-frequency information loss caused by real JPEG. Although in some batches the non-differential real JPEG is applied for end-to-end training, we can use the momentum-based updating optimization method to ensure the correctness of the whole updating direction [42]. During the testing, a total of six kinds of noises are used to attack the encoded image, to detect the robustness of the watermarking, including Gaussian noise ($\sigma = 0.5$), Salt-and-pepper noise ($p = 0.5$), Midden blur ($k = 3$), real JPEG compression ($q = 50\%$), Crop ($p = 0.3$) and Gaussian blur ($\sigma = 0.7$).

4.2 Comparison with the Previous Methods

To verify the effectiveness of the proposed watermarking methods, which have two versions: using the 7×7 or 21×21 kernel in the large kernel convolution block, we compare it with HiDDeN and MBRS [42].

Imperceptibility: Table 1 records the similarity between the encoded image and the cover image of HiDDeN, MBRS, and our LKAW. As can be seen from Table 1, our LKAW significantly outperforms HiDDeN in terms of all four metrics: PSNR increases by 8.13, SSIM increases by 0.0426, RMSE decreases by 4.04, MAE decreases by 3.42. The visual quality of our LKAW is also better than MBRS. Using the 21×21 kernel has higher visual quality than the 7×7 kernel, but they have similar metrics. As shown in Fig. 3, it is difficult to distinguish the encoded image from the cover image visually, and the difference between them (the modification of the cover image by embedding watermarking) is more in the region with abundant edges and textures, but there is almost no change in the smooth region, which proves that our LKAW can according to the content of the cover image to embed the watermarking adaptively. Further, the changes to the cover image using the 21×21 kernel to embed the watermarking are more widely distributed than the 7×7 kernel, which means more slight changes and greater robustness. The larger kernel has better performance in the watermarking domain. In addition, as can be seen from Fig. 3d, because the watermarking is embedded by using the 21×21 large kernel in the wavelet domain, the watermarking can diffuse the whole image, which makes the watermarking smaller on the cover image than 7×7 kernel.

Robustness: Tables 2 and 3 record BER and NC of the decoded watermarking under various noise attacks, respectively. GN, S&P, MF, and GB refer to Gaussian Noise, Salt-and-pepper noise, Median filtering, and Gaussian blur, respectively. As shown in Tables 2 and 3, our LKAW has better watermarking extraction ability than HiDDeN and MBRS in almost all cases. Moreover, our model

is superior to HiDDeN in resisting JPEG compression. Although, JPEG compression is realized by quantizing and deleting the high-frequency components in the DCT domain, which may not be completely consistent with the high-frequency sub-bands of DWT. The essence that high-frequency components reflect the texture and edge information for the image is the same. Therefore, high-frequency wavelet loss can indeed enhance the robustness of JPEG compression.

Table 1: Comparison with other method imperceptibility results

Methods	Kernel	PSNR↑	SSIM↑	RMSE↓	MAE↓
HiDDeN [2]	3×3	31.99	0.9295	6.59	5.32
MBRS [42]	3×3	31.68	0.8099	6.72	5.68
Ours	7×7	37.86	0.9632	3.36	2.33
Ours	21×21	**40.12**	**0.9721**	**2.55**	**1.90**

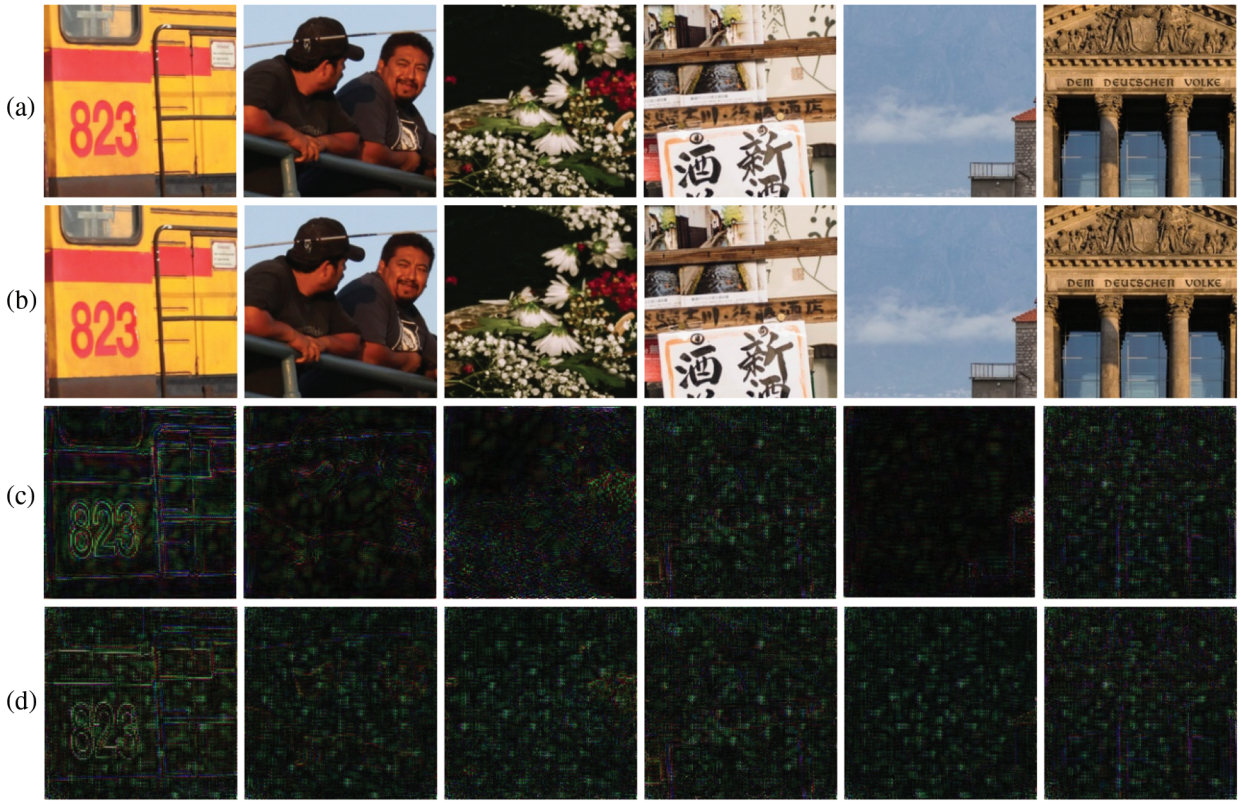

Figure 3: Examples of embedding watermarking to the cover image. (a) The cover image (b) The encoded image by the 21×21 kernel of our LKAW (c) The difference that is multiplied by 20 between the cover image and the encoded image of 7×7 kernel (d) The difference that is multiplied by 20 between the cover image and the encoded image of 21×21 kernel

Table 2: Comparison of BER (%) results with other methods against noise attacks

Methods	Kernel	Identity	GN $\sigma = 0.5$	S&P $p = 0.5$	MF $k = 3$	JPEG 50%	Crop $p = 0.3$	GB $\sigma = 0.7$
HiDDeN [2]	3×3	19.02	44.68	47.37	22.48	47.61	20.76	21.32
MBRS [42]	3×3	0.00	10.61	18.76	0.00	2.03	**0.00**	0.00
Ours	7×7	0.00	0.00	0.00	5.35	35.63	22.83	0.00
Ours	21×21	**0.00**	**0.00**	**0.00**	**0.00**	**2.01**	3.08	**0.00**

Table 3: Comparison of NC results with other methods against noise attacks

Methods	Kernel	Identity	GN $\sigma = 0.5$	S&P $p = 0.5$	MF $k = 3$	JPEG 50%	Crop $p = 0.3$	GB $\sigma = 0.7$
HiDDeN [2]	3×3	0.80	0.56	0.62	0.76	0.54	0.78	0.77
MBRS [42]	3×3	0.99	0.90	0.82	0.99	0.95	**0.99**	0.99
Ours	7×7	0.99	0.99	0.99	0.94	0.64	0.76	0.99
Ours	21×21	**0.99**	**0.99**	**0.99**	**0.99**	**0.95**	0.96	**0.99**

4.3 Ablation Study

An ablation experiment is designed to demonstrate our method's the effectiveness in terms of large kernel convolution, adaptive loss weights, and high-frequency wavelet loss, an ablation experiment is designed. Among them, the small kernel convolution block is a $Conv3 \times 3 - BN - RELU$ block, and the fixed loss weights of λ_{RC}, λ_{RW}, λ_{PC}, λ_{CC}, and λ_{HF} are set to 1.5, 2.0, 1.5, 0.5, and 1.0.

Imperceptibility: Table 4 records the similarity between the encoded image and the cover image, where LK denotes large kernel convolution, WH denotes high-frequency wavelet loss, and AW denotes adaptive weight assignment for loss functions. Under only using the large kernel convolution, for PSNR and SSIM, our LKAW increases by 1.10 dB and 0.0375, respectively; for RMSE and MAE, our LKAW decreases by 0.69 and 0.05, respectively, compared to using small kernel convolution. Furthermore, adding high-frequency wavelet loss results in a reduced visual quality, PSNR decreases by 0.43 dB, and SSIM decreases by 0.0375. RMSE and MAE are also higher. Because the large kernel can see an extensive range of feature maps, so it can use the shape basis to make the best embedding region decision, and the features extracted by the large kernel convolution correspond to a larger range of pixels. Note that, the changes to the cover image are diffused to more pixels, making the changes to each pixel smaller, thus obtaining a higher similarity to the cover image. In addition, the integrated larger range of neighboring features can further reduce the probability of artifacts affecting the image's visual quality. The adaptive weight assignment strategy dynamically adjusts the relationship between each loss, which makes the network converge better and the visual quality stronger than that of fixed weight training. The high-frequency wavelet loss modulates the watermarking embedding region, thus obtaining worse imperceptibility.

Robustness: From Tables 5 and 6, the architecture using small kernel convolution shows a higher BER overall. After using high-frequency wavelet loss to select the embedding region of the watermarking in the frequency domain, both large kernel convolution and small kernel convolution

have lower BER. Using all of them, BER is the lowest and NC is the highest in most cases. This indicates that the large kernel convolution can sense a larger range of image changes, as a result, it is more sensitive to watermarking modifications and can have a stronger watermarking extraction ability. In addition, introducing the adaptive weight assignment strategy improves the training and achieves stronger robustness. Adding high-frequency wavelet loss to avoid watermarking embedding in image texture and contour regions can indeed effectively enhance the watermarking resistance for JPEG compression.

Table 4: Results of imperceptibility in ablation study

LK	WH	AW	PSNR↑	SSIM↑	RMSE↓	MAE↓
			32.98	0.9105	5.76	3.95
√			34.08	0.9480	5.07	3.90
√	√		33.65	0.9105	5.33	3.55
√	√	√	**40.12**	**0.9721**	**2.55**	**1.90**

Table 5: BER (%) results against noise attacks in an ablation study

LK	WL	AW	Identity	GN $\sigma = 0.5$	S&P $p = 0.5$	MF $k = 3$	JPEG 50%	Crop $p = 0.3$	GB $\sigma = 0.7$
			0.00	0.00	0.16	0.00	50.30	35.24	4.16
√			0.00	0.00	0.00	0.41	37.61	14.51	2.63
√	√		0.00	0.00	0.00	0.00	21.47	26.35	0.00
√	√	√	**0.00**	**0.00**	**0.00**	**0.00**	**2.01**	**3.08**	**0.00**

Table 6: NC results against noise attacks in an ablation study

LK	WL	AW	Identity	GN $\sigma = 0.5$	S&P $p = 0.5$	MF $k = 3$	JPEG 50%	Crop $p = 0.3$	GB $\sigma = 0.7$
			0.99	0.99	0.76	0.99	0.49	0.67	0.99
√			0.99	0.99	0.99	0.99	0.54	0.84	0.97
√	√		0.99	0.99	0.99	0.99	0.71	0.71	0.99
√	√	√	**0.99**	**0.99**	**0.99**	**0.99**	**0.97**	**0.96**	**0.99**

4.4 Noise Attack Experiments

To examine the performance of our LKAW against many different attacks, we design noise attack experiments: 6 common types of noise are added to the encoded image. The BER of the decoded watermarking with the attack strength gradually increasing is recorded.

Figs. 4 and 5 show the BER between the decoded watermarking and the input watermarking under 6 attacks, respectively. Our LKAW shows a high tolerance to these noise attacks, especially

for Gaussian noise, Salt-and-pepper noise, and Gaussian blur. The decoded watermarking still has a low error even under the stronger distortion. For example, as can be seen in Fig. 4, the decoded watermarking has a low average BER of 0.6%, 7.1%, and 3.0% under severe distortions including Gaussian noise ($\sigma = 3.0$), Salt-and-pepper noise ($p = 0.9$), and Crop ($p = 0.3$). This indicates that the training with combination noise can have better robustness to prevalent image distortion, and the large kernel convolution can indeed learn robust features of the cover image that help embed and extract the watermarking.

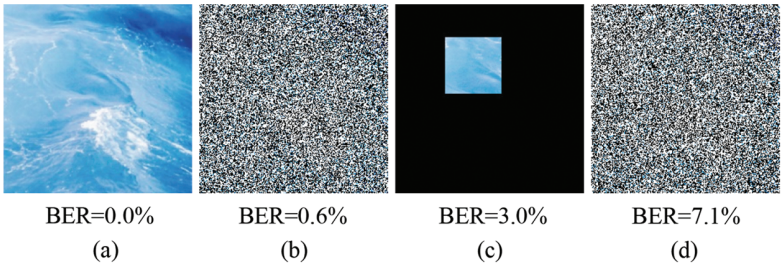

Figure 4: Examples of the noise image added a strong noise. (a) The noised image added no noise (b) Adding Gaussian noise ($\sigma = 3.0$) (c) Adding Crop ($p = 0.3$) (d) Adding Salt-and-pepper noise ($p = 0.9$)

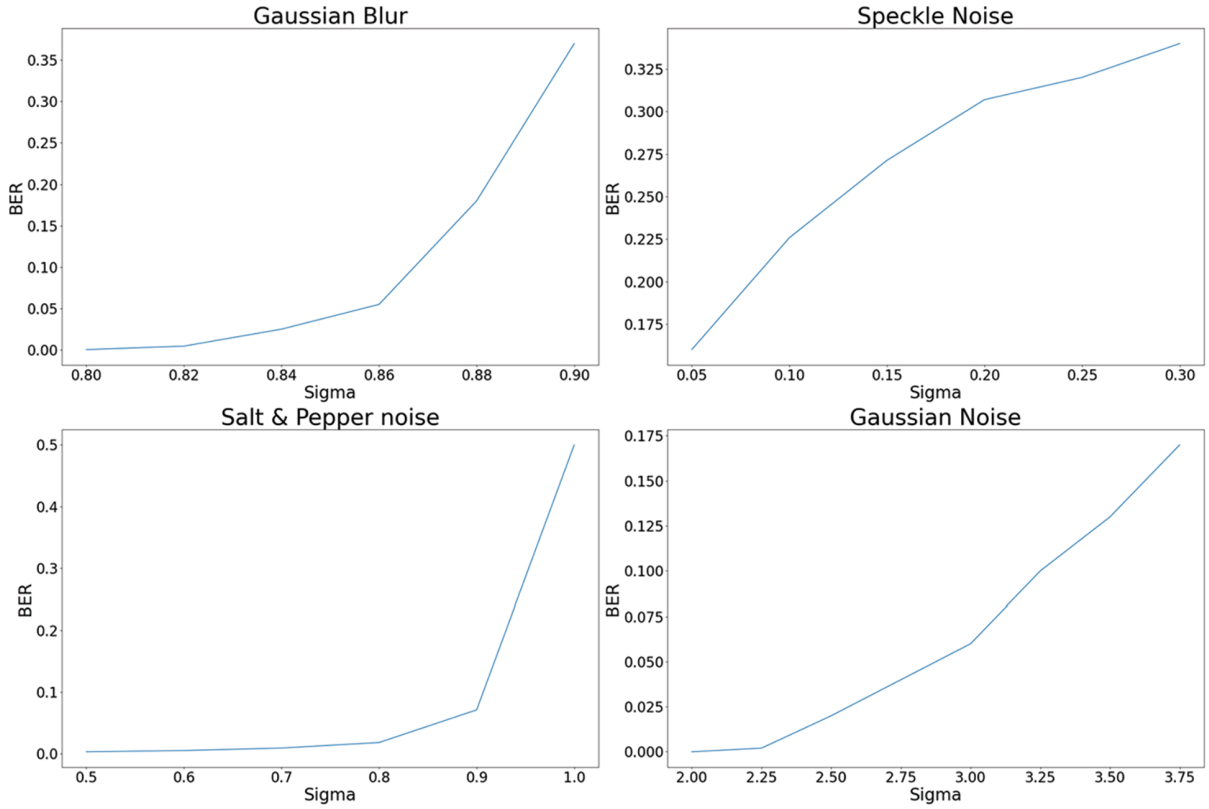

Figure 5: (Continued)

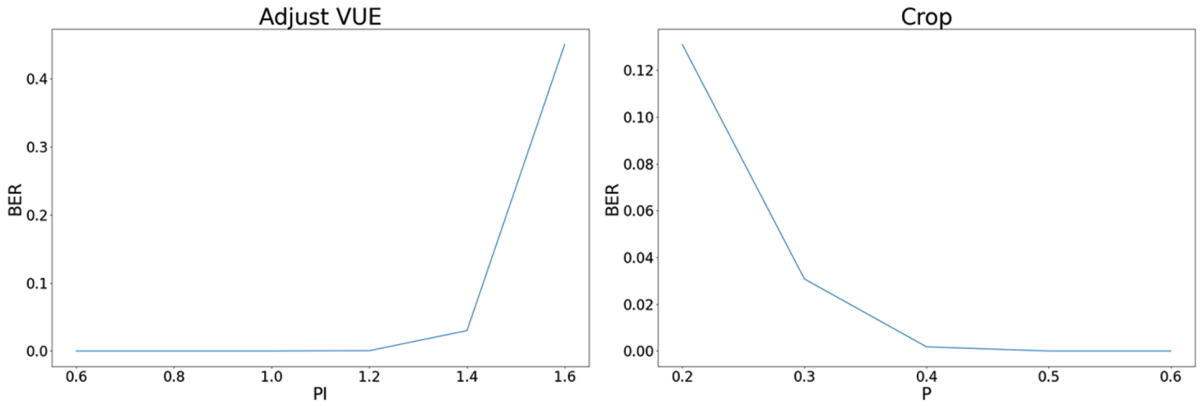

Figure 5: The average BER for watermarking extraction under four different intensities of noise attacks

5 Conclusion

In this paper, we propose a robust watermarking method based on large kernel convolution and adaptive weight assignment for loss functions, which captures the robust features invariant to image attacks, so that the optimal embedding method for watermarking can be learned. Meanwhile, due to the different convergence speeds and magnitude of the five losses, we use an adaptive weight assignment for loss functions to dynamically adjust the weight and add a regular term to prevent the weight from being too large. Moreover, to avoid the watermarking being removed from the image compression, we propose a high-frequency wavelet loss to force the watermarking to be embedded in the low-frequency sub-bands in the wavelet domain. The experimental results demonstrate that the method better maintains the balance of robustness and imperceptibility of watermarking. We will further improve our work in the following aspects in the future: (1) Exploring better methods to expand the effective perception fields, such as the global modeling capability brought by the self-attention mechanism of the vision transformer [18,19]. (2) Exploring the robustness of watermarking for unknown distortion, thus improving the generalizability of watermarking, such as using adversarial training to simulate unknown distortion [3,16].

Acknowledgement: The resources and computing environment was provided by the Nanjing University of Information Science and Technology, Nanjing, China. We are thankful for their support.

Funding Statement: This work was supported, in part, by the National Nature Science Foundation of China under grant numbers 62272236; in part, by the Natural Science Foundation of Jiangsu Province under grant numbers BK20201136, BK20191401; in part, by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] X. R. Zhang, X. Sun, W. Sun, T. Xu and P. P. Wang, "Deformation expression of soft tissue based on BP neural network," *Intelligent Automation & Soft Computing*, vol. 32, no. 2, pp. 1041–1053, 2022.

[2] J. Zhu, R. Kaplan, J. Johnson and L. Fei-Fei, "HiDDeN: Hiding data with deep networks," in *Proc. ECCV*, Munich, Germany, pp. 657–672, 2018.

[3] X. Luo, R. Zhan, H. Chang, F. Yang and P. Milanfar, "Distortion agnostic deep watermarking," in *Proc. CVPR*, Seattle, WA, USA, pp. 13548–13557, 2020.
[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, San Diego, CA, USA, pp. 1–14, 2015.
[5] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Las Vegas, NV, USA, pp. 770–778, 2016.
[6] M. H. Guo, C. Z. Lu, Z. N. Liu, M. M. Cheng and S. M. Hu, "Visual attention network," arXiv: 2202.09741, 2022.
[7] X. Ding, X. Zhang, Y. Zhou, J. Han, G. Ding *et al.,* "Scaling up your kernels to 31 × 31: Revisiting large kernel design in CNNs," in *Proc. CVPR*, New Orleans, LA, USA, pp. 11963–11975, 2022.
[8] Z. Liu, H. Mao, C. Y. Wu, C. Feichtenhofer, T. Darrell *et al.,* "A ConvNet for the 2020s," in *Proc. CVPR*, New Orleans, LA, USA, pp. 11976–11986, 2022.
[9] W. Sun, L. Dai, X. R. Zhang, P. S. Chang and X. Z. He, "RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring," *Applied Intelligence*, vol. 50, no. 8, pp. 8448–8463, 2022.
[10] Y. Xue, Y. K. Wang, J. Y. Liang and A. Slowik, "A self-adaptive mutation neural architecture search algorithm based on blocks," *IEEE Computational Intelligence Magazine*, vol. 16, no. 3, pp. 67–78, 2021.
[11] R. Cipolla, Y. Gal and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. CVPR*, Salt Lake City, UT, USA, pp. 7482–7491, 2018.
[12] M. Tancik, B. Mildenhall and R. Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *Proc. CVPR*, Seattle, WA, USA, pp. 2114–2123, 2020.
[13] S. M. Mun, S. H. Nam, H. U. Jang, D. Kim and H. K. Lee, "A robust blind watermarking using convolutional neural network," arXiv: 1704.03248, 2017.
[14] H. Kandi, D. Mishra and S. R. K. S. Gorthi, "Exploring the learning capabilities of convolutional neural networks for robust image watermarking," *Computers & Security*, vol. 65, no. C, pp. 247–268, 2017.
[15] C. Zhang, P. Benz, A. Karjauv, G. Sun and I. S. Kweon, "Universal deep hiding for steganography, watermarking, and light field messaging," in *Proc. NIPS*, Virtual, pp. 10223–10234, 2020.
[16] X. Zhong, P. C. Huang, S. Mastorakis and F. Y. Shih, "An automated and robust image watermarking scheme based on deep neural networks," *IEEE Transactions on Multimedia*, vol. 23, pp. 1951–1961, 2021.
[17] Y. Liu, M. Guo, J. Zhang, Y. Zhu and X. Xie, "A novel two-stage separable deep learning framework for practical blind watermarking," in *Proc. ACM MM*, Nice, France, pp. 1509–1517, 2019.
[18] A. Dosovitskiy, L. Beyer, K. Alexander, D. Weissenborn, X. H. Zhai *et al.,* "An image is worth 16 × 16 words: Transformers for image recognition at scale," arXiv:2010.11929, 2020.
[19] Z. Liu, Y. T. Lin, Y. Cao, Y. X. Wei, Z. Zhang *et al.,* "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. ICCV*, Virtual, pp. 10012–10022, 2021.
[20] Y. Cao, Z. L. Zhou, C. Chakraborty, M. M. Wang, X. M. Sun *et al.,* "Generative steganography based on long readable text generation," *IEEE Transactions on Computational Social Systems*, pp. 1–11, 2022. DOI https://doi.org/10.1109/TCSS.2022.3174013.
[21] X. R. Zhang, J. Zhou, W. Sun and S. K. Jha, "A lightweight CNN based on transfer learning for COVID-19 diagnosis," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 1123–1137, 2022.
[22] W. Yu, M. Luo, P. Zhou, C. Y. Si, Y. C. Zhou *et al.,* "MetaFormer is actually what you need for vision," in *Proc. CVPR*, New Orleans, LA, USA, pp. 10819–10829, 2022.
[23] I. Tolstikhin, H. Houlsby, A. Kolesnikov, L. Beyer, X. H. Zhai *et al.,* "MLP-Mixer: An all-MLP architecture for vision," in *Proc. NIPS*, Virtual, pp. 24261–24272, 2021.
[24] W. Sun, G. C. Zhang, X. R. Zhang, X. Zhang and N. N. Ge, "Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning strategy," *Multimedia Tools and Applications*, vol. 80, no. 20, pp. 30803–30816, 2021.
[25] Y. Xue, Y. H. Tang, X. Xu, J. Y. Liang and F. Neri, "Multi-objective feature selection with missing data in classification," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 355–364, 2022.
[26] R. Caruana, "Multi-task learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[27] W. Sun, G. Z. Dai, X. R. Zhang, X. Z. He and X. Chen, "TBE-Net: A three-branch embedding network with part-aware ability and feature complementary learning for vehicle re-identification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14557–14569, 2022.
[28] Y. J. Ren, F. J. Zhu, P. K. Sharma, T. Wang, J. Wang *et al.,* "Data query mechanism based on hash computing power of blockchain in Internet of things," *Sensors*, vol. 20, no. 1, pp. 1–22, 2020.
[29] B. Wen and S. Aydore, "ROMark: A robust watermarking system using adversarial training," arXiV: 1910.01221, 2019.
[30] M. Ahmadi, A. Norouzi, N. Karimi, S. Samavi and A. Emami, "ReDMark: Framework for residual diffusion watermarking based on deep networks," *Expert Systems with Applications*, vol. 146, pp. 113157, 2020.
[31] S. Liu, E. Johns and A. J. Davison, "End-to-end multi-task learning with attention," in *Proc. CVPR*, Long Beach, CA, USA, pp. 1871–1880, 2019.
[32] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, Munich, Germany, pp. 234–241, 2015.
[33] R. Zhang, P. Isola, A. A. Efros, E. Shechtman and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. CVPR*, Salt Lake City, UT, USA, pp. 586–595, 2018.
[34] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[35] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. Courville, "Improved training of Wasserstein GANs," in *Proc. NIPS*, Long Beach, CA, USA, pp. 5769–5779, 2017.
[36] J. Hu, L. Shen and G. Sun, "Squeeze-and-excitation networks," in *Proc. CVPR*, Salt Lake City, UT, USA, pp. 7132–7141, 2018.
[37] T. Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick *et al.,* "Microsoft COCO: Common objects in context," in *Proc. ECCV*, Zurich, Switzerland, pp. 740–755, 2014.
[38] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. CVPR*, Honolulu, HI, USA, pp. 126–135, 2017.
[39] L. Wright and N. Demeure, "Ranger21: A synergistic deep learning optimizer," arXiv:2106.13731, 2021.
[40] M. R. Zhang, J. Lucas, G. Hinton and J. Ba, "Lookahead optimizer: k steps forward, 1 step back," in *Proc. NIPS*, Vancouver, Canada, pp. 9597–9608, 2019.
[41] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
[42] Z. Jia, H. Fang and W. Zhang, "MBRS: Enhancing robustness of DNN-based watermarking by mini-batch of real and simulated JPEG compression," in *Proc. ACM MM*, Chengdu, China, pp. 41–49, 2021.