

MNet: A Multi-Scale Network for Visible Watermark Removal

WenHong Huang^a, YunShu Dai^a, JianWei Fei^a and FangJun Huang^a

^aSchool of Cyber Science and Technology, Shenzhen Campus of Sun Yat-sen University, Shenzhen, China

^aGuangdong Provincial Key Laboratory of Information Security Technology, Guangzhou, China

ARTICLE INFO

Keywords:

Deep Neural Networks
Multi-Scale Network
Visible Watermark Removal
Multi-Task Learning
Watermark

ABSTRACT

Superimposing visible watermarks on images is an efficient way to indicate ownership and prevent potential unauthorized use. Visible watermark removal technology is receiving increasing attention from researchers due to its ability to enhance the robustness of visible watermarks. In this paper, we propose MNet, a novel multi-scale network for visible watermark removal. In MNet, a variable number of simple U-Nets are stacked in each scale. There are two branches in MNet, i.e., the background restoration branch and the mask prediction branch. In the background restoration branch, we propose a different approach from current methods. Instead of directly reconstructing the background image, we pay great attention to predicting the anti-watermark image. In the watermark mask prediction branch, we adopt dice loss. This further supervises the predicted mask for better prediction accuracy. To make information flow more effective, we employ cross-layer feature fusion and intra-layer feature fusion among U-Nets. Moreover, a scale reduction module is employed to capture multi-scale information effectively. Our approach is evaluated on three different datasets, and the experimental results show that our approach achieves better performance than other state-of-the-art methods. Code will be available at <https://github.com/itchson-Hwang/MNet>.

1. Introduction

Visible watermarks are visible marks or motifs embedded in digital images or videos, which are used to protect images and represent image ownership [1, 2, 3, 4, 5]. The adversarial technology, known as visible watermark removal, is designed to obtain the watermark-free image from a watermarked image, which can help to develop more advanced anti-removal watermark embedding strategies [6]. However, watermarks designed by different producers exhibit considerable variation in texture, structure, and color, which makes the visible watermark removal task quite challenging.

Initially, traditional removal strategies [7, 8, 9, 10] are employed, which usually require user guidance in locating the watermark region or some strict settings to get the watermark-free image. Nowadays, with the widespread application of deep neural networks (DNNs), visible watermark removal methods based on deep learning (DL) have also received extensive attention. Nevertheless, they still have some limitations. For example, some of them [11, 12, 13, 14] rely on prior knowledge such as the transparency of watermark embedding, or are only applicable to grayscale watermarks. Hertz *et al.*, [15] present a single-stage network for the localization and removal of visual motifs embedded in images, which is called BVMR (Blind Visual Motif Removal). BVMR is the first to remove visible watermarks from images blindly (i.e., without explicit prior information). It does not require any user guidance or making intricate assumptions about the visible watermarks. Currently, the latest visible watermark removal methods [6, 16, 17, 18] suggest that traditional single-stage networks may encounter issues such as incomplete watermark localization and poor

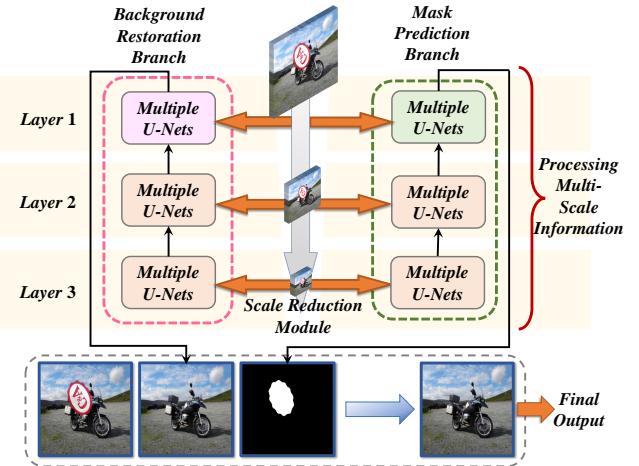


Figure 1: The main framework of MNet. MNet consists of two branches, which can perform background restoration and mask prediction simultaneously. Each branch has a three-layer structure, and each layer has a series of cascaded U-Nets. From Layer 1 to Layer 3, the input scale decreases continuously.

background restoration, resulting in rough visual quality of watermark-free image. Therefore, they construct an additional network stage to refine it. Specifically, they extend the single-stage network of BVMR, to a two-stage network with a series of specific modules, e.g. attention and transformer modules.

Nevertheless, these two-stage networks can introduce new challenges, such as optimization difficulties and error accumulation. When designing the loss functions, it is essential to consider all information from multiple stages, potentially complicating the training process and necessitating meticulous adjustment of hyper-parameters. Moreover, the

E-mail addresses: huangwh79@mail2.sysu.edu.cn, daiyunshu0102@163.com, fjh826244895@163.com, huangfj@mail.sysu.edu.cn.
ORCID(s):

output of the first stage will become the input of the second stage. If the original output of the first stage contains error information such as noise or distortion, these errors may be further amplified in the second stage, which will lead to a lower visual quality of the generated watermark-free image.

To this end, we propose MNet, a novel single-stage network for visible watermark removal. Since MNet is a single-stage network, the optimization process of the entire network is relatively simple, and it avoids the error accumulation issues that may exist in two-stage networks. To address the problems of incomplete watermark localization and low quality of generated watermark-free images that may exist in traditional single-stage networks, we design a new multi-scale structure in the proposed MNet network, with each layer consisting of multiple simple U-Nets, as shown in Figure 1. A scale reduction module is utilized to convert the original input into various scale versions and input them into different layers separately. In each layer, the input feature map is continuously refined by a series of cascaded simple U-Nets. After refinement, the feature map is up-sampled. The up-sampled feature map is then used as the input for the next higher layer. This process actually takes advantage of the benefits of the two-stage networks mentioned above. In general, MNet is a multi-task learning framework with a three-layer architecture that can perform watermark localization (also known as mask prediction) and watermark removal (also known as background restoration) simultaneously, where the two task branches share all parameters in the second and third layers. Furthermore, MNet is a highly flexible network since the number and cascading paradigm of U-Nets as well as the location of each U-Net in MNet, can be adjusted at will. Considering that U-Nets at different layers capture different-scale structural and texture information, we propose cross-layer and intra-layer feature fusion schemes. These schemes facilitate more efficient information flow across these U-Nets, which will be introduced in detail later.

Totally, our contributions are summarized as follows:

- We propose a novel single-stage and multi-scale (or multi-layer) network for visible watermark removal, MNet, which consists of two branches, i.e., the background restoration branch and mask prediction branch.
- We develop a new multi-scale feature extraction strategy based on a multi-layer structure and a scale reduction module. To facilitate efficient information flow in different layers, we introduce cross-layer and intra-layer feature fusion schemes.
- We introduce a novel approach to predict the anti-watermarked images during background restoration, alongside a loss function optimization strategy for mask prediction. This strategy incorporates the use of dice loss, which will be detailed later.
- Extensive experiments conducted on several diverse datasets demonstrate that our proposed method significantly outperforms other state-of-the-art (SOTA)

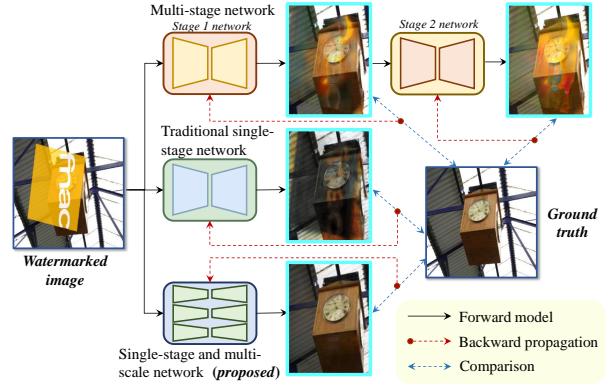


Figure 2: Partial implementation and optimization process of a multi-stage network, a traditional single-stage network, and the proposed single-stage multi-scale network (presented from top to bottom). The output images from these networks are the restored background images (for better visualization of the watermark removal effect, only the watermark region is shown in the figure), generated by DENet (multi-stage network) [18], BVMR (traditional single-stage network) [15], and the proposed MNet, respectively.

methods in both background restoration and mask prediction, showcasing exceptional practical value for visible watermark removal.

2. Related Work

In this section, we first introduce some works related to visible watermark removal, namely *Image Content Removal*. Then, we introduce the works on *Visible Watermark Removal*, including early works and recent advanced works.

2.1. Image Content Removal

Image content removal tasks, such as deraining [19, 20, 21], shadow removal [22, 23], dehazing [24, 25, 26, 27] and deblurring [28, 29, 30, 31], are similar tasks to visible watermark removal. These tasks usually deal with repeated or similar elements (such as raindrops, shadows, etc.), allowing the trained DNN model to effectively learn to recognize and process these patterns, thus simplifying the detection and removal of these elements.

In contrast, the task of visible watermark removal is more challenging than removing elements like raindrops or haze, since the diverse nature of watermarks, which often vary in shape, color, and texture. This diversity makes it challenging for models to consistently identify and remove watermarks without affecting the image's visual quality.

2.2. Visible Watermark Removal

Initial researches [7, 8, 9] on visible watermark removal usually require user guidance to locate the position of the watermark for subsequent tasks such as background restoration, where this reliance on the user interaction may limit the practicality of these methods. The method proposed in [10]

Table 1
Symbol glossary.

Categorie	Symbol	Description	Categorie	Symbol	Description
Image and feature map	I	Original image / Ground truth image.	Network	k_1	The number of U-Nets in the layer 1
	I_w	Watermarked image.		k_2	The number of U-Nets in the layer 2
	M	Ground truth mask.		k_3	The number of U-Nets in the layer 3
	x_1	The input of layer 1 in MNet.		U_M^3	The 3-th U-Net in the mask prediction branch of the layer 1.
	x_2	The input of layer 2 in MNet.		U_B^3	The 3-th U-Net in the background restoration branch of the layer 1.
	x_3	The input of layer 3 in MNet.		$U_2^{k_2}$	The last U-Net in layer 2.
	\hat{W}_a	Predicted anti-watermark image.		U_2^1	The first U-Net in layer 2.
	\hat{I}_r	Restored background.		$U_3^{k_3}$	The last U-Net in layer 3.
Loss function	\hat{M}	Predicted mask.	Hyper-parameter	λ_{L_1}	The hyper-parameter multiplied by \mathcal{L}_{L_1} .
	\hat{I}	Final watermark-free image.		λ_{perc}	The hyper-parameter multiplied by \mathcal{L}_{perc} .
	\mathcal{L}_{L_1}	Loss for background restoration.		λ_{bce}	The hyper-parameter multiplied by \mathcal{L}_{bce} .
	\mathcal{L}_{perc}	Loss for background restoration.		λ_{iou}	The hyper-parameter multiplied by \mathcal{L}_{iou} .
	\mathcal{L}_{bce}	Loss for mask prediction.		λ_{dice}	The hyper-parameter multiplied by \mathcal{L}_{dice} .
	\mathcal{L}_{iou}	Loss for mask prediction.			
	\mathcal{L}_{dice}	Loss for mask prediction.			

does not require any user guidance. However, it requires all images in the dataset to have the same watermark.

The pioneer single-stage watermark removal method based on DL is BVMR [15]. It considers visible watermark removal as a multi-task learning problem, which has multiple branches and can perform mask prediction and background restoration simultaneously. The input of BVMR is the watermarked image I_w , which is obtained via superimposing the watermark W onto the original image I , and the outputs of the two branches of BVMR are the restored background (\hat{I}_r) and predicted mask (\hat{M}), respectively. The final output of the BVMR network, i.e., the watermark-free image \hat{I} , can be written as

$$\hat{I} = \hat{I}_r \times \hat{M} + I_w \times (1 - \hat{M}). \quad (1)$$

Modern watermark removal methods such as SplitNet [16], SLBR [17] and DENet [18] believe that the single-stage method BVMR may suffer from poor visual quality of the generated watermark-free images. Therefore, they regard the output of the single-stage network as a coarse output \hat{I}_c that needs further refinement, and input this \hat{I}_c into the second-stage network, using Eq. (1) again to obtain the final output.

Nevertheless, in these methods, the first-stage and the second-stage networks have two separate sets of modules that are only linked by some simple connections. This separation of two stages limits the network's performance and may lead to some new issues, such as optimization difficulties and error accumulation issues mentioned earlier. Therefore, we believe that designing a superior single-stage network is more efficient than optimizing a two-stage network for visible watermark removal.

2.3. Differences Between the Proposed Method and Current Works

In the above, Section 2.2 provides a comprehensive review of existing methods for visible watermark removal,

including these single-stage and multi-stage networks. In Figure 2, we illustrate the partial implementation and optimization process of multi-stage networks, traditional single-stage networks, and our proposed network. The figure highlights several challenges faced by multi-stage networks. As shown in the first row of Figure 2, the stage 1 network of a multi-stage network generates an output with incorrect colors in the watermark area. This color distortion becomes even worse after being input into the stage 2 network, which is an example of the error accumulation problem. As seen in these red and blue dotted lines of Figure 2, designing the loss functions for a multi-stage network and optimizing it requires careful consideration of the outputs and information from multiple stages, which demands more precise adjustments of the hyper-parameters in the loss functions and increases the complexity of the optimization process.

The single-stage network naturally does not have the above-mentioned problems caused by building the second-stage network. However, as shown in the second row of Figure 2, it can be observed that the visual quality of the background image restored by a traditional single-stage network is the poorest. To address these issues, we propose MNet, a novel single-stage and multi-scale network, as shown in the third row of Figure 2

The unique contribution of MNet lies in the multi-layered structural design and efficient information flow schemes. Unlike traditional single-stage networks, MNet utilizes a multi-scale feature extraction strategy that combines both cross-layer and intra-layer feature fusion, significantly enhancing the visual quality of the output image. MNet's single-stage design also avoids the issues associated with multi-stage networks. In addition, when performing background restoration tasks, MNet predicts the anti-watermark image instead of directly predicting the entire background

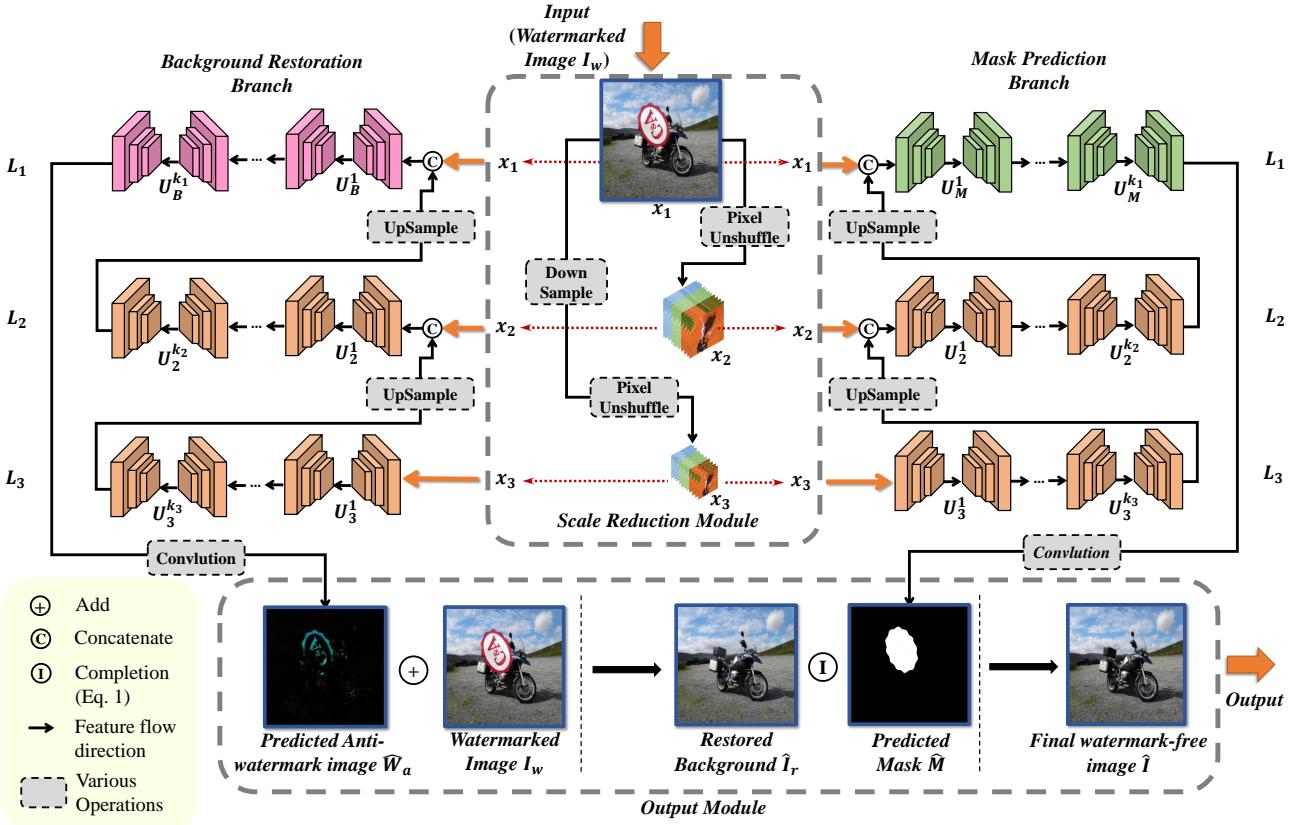


Figure 3: Illustration of our proposed MNet. The proposed network mainly consists of the following parts: a scale reduction module, two task branches, and an output module. Note that we share the parameters in L_2 and L_3 of the two branches for lower model complexity.

image like current works [15, 16, 17, 18], which can reduce the difficulty of the model's prediction.

3. Methodology

In this paper, we address visible watermark removal from a new perspective. Our objective is to design a single-stage network that can generate watermark-free images with high visual quality. To this end, we proposed MNet, a single-stage and multi-scale network for visible watermark removal, specifically comprising the following parts: a scale reduction module, two task branches (i.e., background restoration branch and mask prediction branch), and an output module. Note that the two task branches have the same three-layer structure, where each layer is stacked with a series of **simple** U-Nets. In each layer, the feature maps are processed among these U-Nets. We improve the efficiency of information flow and feature integration by introducing the scale reduction module and the innovative feature fusion schemes. The scale reduction module enables the network to capture image features at different scales, while cross-layer and intra-layer feature fusion ensures that these features can be effectively shared and optimized among various U-Nets,

thereby improving the performance of watermark localization and removal without increasing excessive additional computational costs.

In the following parts of this section, we will introduce the specific details of the various modules and schemes applied in MNet, i.e., the scale reduction module (Section 3.1), two task branches (Section 3.2), the output module (Section 3.3) and the feature fusion scheme (Section 3.4), followed by the loss functions (Section 3.5) used in MNet. In addition, a glossary of the symbols used in our paper is provided, as shown in Table 1.

As shown in Figure 3, we use L_i to represent the i -th ($i = 1, 2, 3$) layer of two branches, and the number of U-Nets stacked in L_i is represented with k_i ($k_i \in \mathbb{N}^+$). \mathbf{U}_B and \mathbf{U}_M represent the collection of all U-Nets in L_1 of the background restoration branch and mask prediction branch, and the j -th ($1 \leq j \leq k_1$) U-Nets of \mathbf{U}_B and \mathbf{U}_M are denoted as U_B^j and U_M^j , respectively. In L_2 and L_3 , the U-Nets in the two branches share the same parameters, and we use \mathbf{U}_s ($s = 2, 3$) to represent the collection of all U-Nets in L_s .

3.1. Scale Reduction Module

This module transforms the original input, i.e., the watermarked image I_w with the size of (W, H, C) , into different scales. The transformed images with different sizes

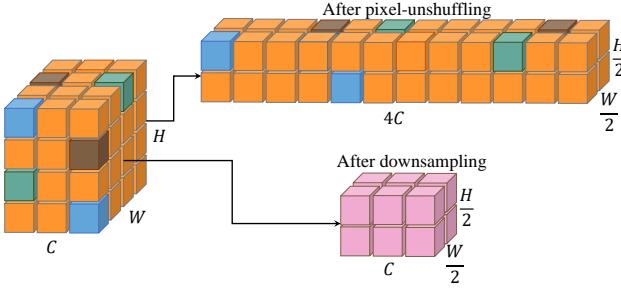


Figure 4: Details of pixel-unshuffling and downsampling operations. The input feature map is with the size of (W, H, C) . After applying the pixel-unshuffling operation to the input, the spatial dimensions of the feature map change to $(W/2, H/2)$, and the number of channels increases to $4C$, while the pixel values remain unchanged but are rearranged. After applying the downsampling operation, the spatial dimensions are reduced to $(W/2, H/2)$, and the number of channels remains unchanged, with the pixel values being modified due to operations like convolution.

are fed into different layers of the network. We denote the input of L_i as x_i . In the first layer L_1 , the input x_1 is the original watermarked image I_w , where the original I_w is directly used as the input to preserve the most detailed information; in the second layer L_2 , the input x_2 is the pixel-unshuffled [32] version of the original watermarked image I_w ; in the third layer L_3 , the input is x_3 , which is obtained via down-sampling x_1 first followed by pixel-unshuffling. The pixel-unshuffling operation reduces the feature map's spatial size to 1/4 of the original and increases the number of channels by 4 times, while the downsampling operation also reduces the spatial size to 1/4 but the number of channels remains unchanged, as shown in Figure 4. Thus, x_2 is with the size of $(W/2, H/2, 4C)$ and x_3 is with the size of $(W/4, H/4, 4C)$. Note that the pixel-unshuffling operation only rearranges the values within the feature map without altering them, while the downsampling operation changes the values of the feature map through operations like convolution. Totally, this scale reduction module comprehensively processes the watermarked image I_w to obtain structural and texture information at different coarse levels. By capturing these features, this module enables a more detailed reconstruction of the original image.

3.2. Two Task Branches

As seen in Figure 3, the background restoration branch and mask prediction branch have the same three-layer structure. They share all the parameters in the lowest two layers L_2 and L_3 , since sharing partial parameters may lead to better model performance, which is based on previous research [15], sharing partial parameters may lead to better model performance. In these two branches, the feature flow is as follows: 1) input x_3 into the first U-Net U_3^1 in L_3 ; 2) features are propagated between the U-Nets of U_3 ; 3) the outputs of L_3 are up-sampled and concatenated with x_2 , and then input into the first U-Net U_2^1 in L_2 ; 4) features are

propagated between the U-Nets of U_2 ; 5) the outputs of L_2 are up-sampled and concatenated with x_1 , which are then input into the first U-Net $U_B^1 (U_M^1)$ in L_1 in two branches; 6) features are propagated between the U-Nets of $U_B (U_M)$; 7) the output features of L_1 are convoluted to get the outputs of the two branches, i.e. the predicted anti-watermark image \hat{W}_a and the predicted mask \hat{M} . These outputs are then used in the output module, detailed in the next section.

3.3. Output Module

Note that the output of our background restoration branch is different from that in previous works [15, 16, 17, 18]. In this method, we propose a new background restoration strategy. That is, we do not predict the background image \hat{I}_r directly, whereas predict the anti-watermark image \hat{W}_a instead. We then add this \hat{W}_a to the watermarked image I_w , to restore the background image \hat{I}_r , which can be written as:

$$\hat{I}_r = I_w + \hat{W}_a. \quad (2)$$

Finally, with the restored background image \hat{I}_r , the predicted mask \hat{M} and the original input I_w , the final generated watermark-free image \hat{I} of our MNet can be obtained by using Eq. (1), where \hat{I} is the final output.

In this new background restoration strategy, as illustrated in Figure 3, most pixels in the predicted anti-watermark image \hat{W}_a are with zero values (the black areas). This indicates that the network only needs to learn to adjust non-zero value pixels, without requiring the generation of the entire background image. Through experiments, we have found that this strategy significantly reduces the training difficulty of the network, makes it easier to converge, and improves the visual quality of the final generated watermark-free image.

3.4. Feature Fusion Schemes

Moreover, we enhance the feature map flow by integrating two feature fusion schemes, i.e., intra-layer feature fusion (In-F) and cross-layer feature fusion (Cr-F).

Intra-Layer Feature Fusion (In-F), illustrated by the orange dotted lines in Figure 5, occurs between two adjacent U-Nets within the same layer. This process can be described in the following steps: 1) generate feature maps from each encoder and decoder of the previous U-Net, following the direction of the feature flow (the feature flow direction is illustrated in Figures 3 and 5); 2) add together the feature maps from both the encoder and decoder at the same level in the previous U-Net; 3) feed the combined feature map from the previous U-Net into the encoder of the next U-Net at the same level. As a result, the input to each encoder in the next U-Net consists of two concatenated feature maps: one is the merged feature map produced by summing the encoder and decoder outputs from the previous U-Net at the same level, and the other is the feature map passed along the feature flow direction. Additionally, Figure 5 provides a schematic representation of the encoders, decoders, and levels within one U-Net.

Cross-Layer Feature Fusion (Cr-F), represented by the gray dotted line in Figure 5, connects two U-Nets, i.e., the

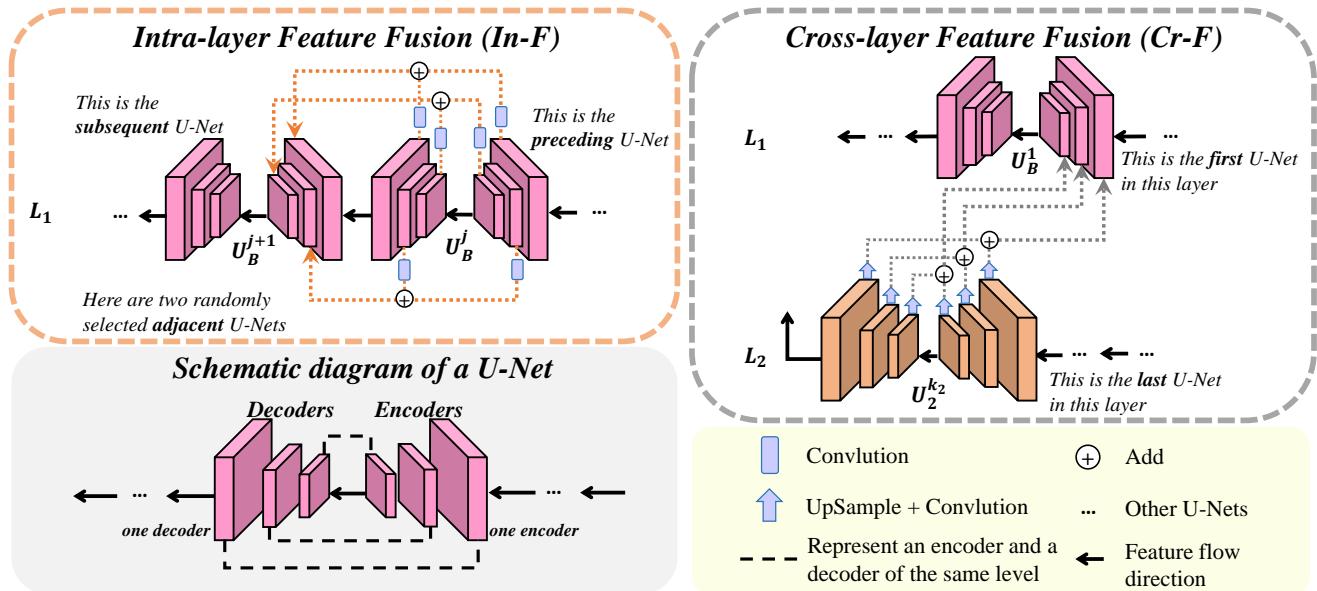


Figure 5: Illustration of our feature fusion scheme in MNet. Note that the intra-layer feature fusion (In-F) exists in every pair of the adjacent U-Nets in one layer and the cross-layer feature fusion (Cr-F) exists in every pair of the adjacent layers in MNet. The schematic diagram of the U-Net structure used in MNet is also shown in the lower-left corner of the figure. In this diagram, the encoder and decoder at the same level are connected by a black dashed line.

last U-Net in a lower layer and the first U-Net in the next higher layer. Cr-F works similarly to In-F, except that the feature maps from the lower U-Net are upsampled to match the spatial dimensions of the higher U-Net’s feature maps. Cr-F allows the model to combine detailed local information from the lower layers with global contextual information from the higher layers.

In summary, both intra-layer and cross-layer feature fusion contribute to cascading the U-Nets, promoting more effective information extraction and feature flow. The integration of these two fusion strategies enhances the model’s ability to capture intricate patterns and relationships within the input data, ultimately improving performance in removing visible watermarks and restoring backgrounds.

3.5. Loss Functions

In this section, we will detail the loss functions utilized in our approach to achieve effective background restoration and accurate mask prediction.

3.5.1. Loss for Background Restoration

Background restoration task aims to obtain the restored background image \hat{I}_r , which should be as close as possible to the ground truth background image I . Here, we adopt the L_1 loss and perception loss [33, 34] to evaluate the difference between the restored background image \hat{I}_r and the ground truth image I . The L_1 loss and perception loss are as follows:

$$\mathcal{L}_{L_1} = ||I - \hat{I}_r||_1, \quad (3)$$

$$\mathcal{L}_{perc} = \sum_{k \in 1,2,3} ||\Phi^k(\hat{I}_r) - \Phi^k(I)||_1, \quad (4)$$

where perception loss is based on VGG16 [35] pretrained on ImageNet [36], and $\Phi^k(*)$ means the activation map of k -th layer in VGG16 [35].

3.5.2. Loss for Mask Prediction

The mask prediction task aims to obtain the predicted mask \hat{M} , which is a two-dimensional matrix only containing elements of 1 and 0. Here, 1 and 0 correspond to the watermarked area and non-watermarked area, respectively. Predicting the mask can be seen as a binary pixel-level segmentation task [16]. The main terms of the loss functions of mask prediction in our network are based on SLBR [17], including the BCE loss and the iou loss:

$$\mathcal{L}_{bce} = M \log(\hat{M}) + (1 - M) \log(1 - \hat{M}), \quad (5)$$

$$\mathcal{L}_{iou} = 1 - \frac{\sum(M \cap \hat{M})}{\sum(M \cup \hat{M})}. \quad (6)$$

Nevertheless, in practical situations, watermarks usually occupy a smaller area to maintain visual quality. This results in an imbalance of pixel counts between watermarked and non-watermarked areas, causing the network to be biased to detect background-biased areas rather than watermarked areas. To address this issue, we additionally employ dice loss [37], a loss term that promotes a balance between watermarked and non-watermarked labels in the mask prediction,

Table 2

Quantitative comparisons on LOGO-H, LOGO-L, and LOGO-Gray datasets for visible watermark removal. We calculate the average PSNR, SSIM (displayed as percentiles), and LPIPS (displayed as percentiles) values between the watermark-free images generated by different methods and the ground truth images. Higher PSNR and SSIM values, along with lower LPIPS values, indicate that the generated watermark-free images are more similar to the ground truth images, demonstrating better visible watermark removal performance. The best results are highlighted in **bold**.

Method		LOGO-H			LOGO-L			LOGO-Gray		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Image Segmentation	(MICCAI'15) UNet [38]	30.51	96.12	5.44	34.87	98.14	2.97	32.15	97.28	3.53
Image Content Removal	(CVPR'18) SIRF [22]	32.35	96.73	8.01	36.25	98.25	6.55	34.33	97.82	6.72
	(TIP'20) BS ² AM [28]	31.93	96.77	4.45	36.11	98.39	2.23	32.91	97.54	3.05
	(AAAI'20) DHAN [23]	35.68	98.09	6.61	38.54	98.87	5.91	36.39	98.36	5.94
Visible Watermark Removal	(CVPR'19) BVMR [15]	36.51	97.99	2.37	40.24	98.95	1.26	38.90	98.73	1.15
	(AAAI'21) SplitNet [16]	40.05	98.97	1.15	42.53	99.24	0.87	42.01	99.28	0.73
	(ACMMM'21) SLBR [17]	40.56	99.13	1.06	44.10	99.47	0.70	42.21	99.36	0.69
	(AAAI'23) DENet [18]	40.83	99.19	0.89	44.24	99.54	0.54	42.60	99.44	0.53
	MNet (Ours)	44.34	99.40	0.65	46.25	99.60	0.44	46.67	99.68	0.25

which can be written as:

$$\mathcal{L}_{dice} = 1 - \frac{2 \times \sum(M \cap \hat{M})}{\sum(M) + \sum(\hat{M})}, \quad (7)$$

where $\sum(*)$ represents the sum of all elements in matrix $(*)$.

Overall, the total loss in our algorithm is a combination of the watermark removal loss and the mask prediction loss, i.e.,

$$\mathcal{L}_{all} = \lambda_{L_1} \mathcal{L}_{L_1} + \lambda_{perc} \mathcal{L}_{perc} + \lambda_{bce} \mathcal{L}_{bce} + \lambda_{iou} \mathcal{L}_{iou} + \lambda_{dice} \mathcal{L}_{dice}, \quad (8)$$

where λ_{L_1} , λ_{perc} , λ_{bce} , λ_{iou} and λ_{dice} are hyper-parameters.

4. Experiments

In this section, we first introduce the experimental datasets and implementation details. Then, a series of comparisons are made between our proposed method and other state-of-the-art methods. Finally, the ablation studies regarding our proposed MNet and some discussions on our proposed method are given.

4.1. Experimental Settings

In this part, we detail the experimental settings of our research, including the datasets used for training and testing, the implementation specifics of our proposed method, and the evaluation metrics used to assess performance.

4.1.1. Datasets

Our experiments are conducted on three diverse datasets (i.e., LOGO-L, LOGO-H, LOGO-Gray), the same datasets used by the latest advanced method, DENet [18]. In these datasets, the background images are collected from the real-world dataset MSCOCO [39], and the watermarks are collected from the Internet, including more than 1,000 famous

and different logos (watermarks). To simulate real-world application scenarios, these watermarks are adjusted to various sizes and embedded with varying degrees of transparency at random positions in different background images. The watermark embedding method involves overlaying the watermark onto the background image pixel by pixel, according to the specified transparency. The specific details of these datasets are as follows.

- **LOGO-L:** This dataset contains 12,151 watermarked images for training and 2,025 watermarked images for testing. Each watermarked image is embedded with a colored watermark. The watermark size accounts for 35% to 60% of the width (or height) of the background image. The transparency of the watermarks varies between 35% and 60%.
- **LOGO-H:** This dataset contains the same number of images as LOGO-L. But the watermarks in LOGO-H are larger (account for 60% to 85%), and the watermark transparency is set from 60% to 85%.
- **LOGO-Gray:** This dataset also contains 12,151 watermarked images for training and 2,025 watermarked images for testing. But LOGO-Gray only contains gray-scale watermarks. The watermark size accounts for 35% to 85%, and the watermark transparency is randomly set from 35% to 85%.

Note that the watermarked images have variable sizes, typically larger than 256×256. To enhance model training, both SplitNet [16] and DENet [18] resize the images to 256×256 during their usage, and we adopt the same approach.

4.1.2. Implementation Details

We use Pytorch [40] to implement our method and train the model with Adam optimizer [41], where the learning rate

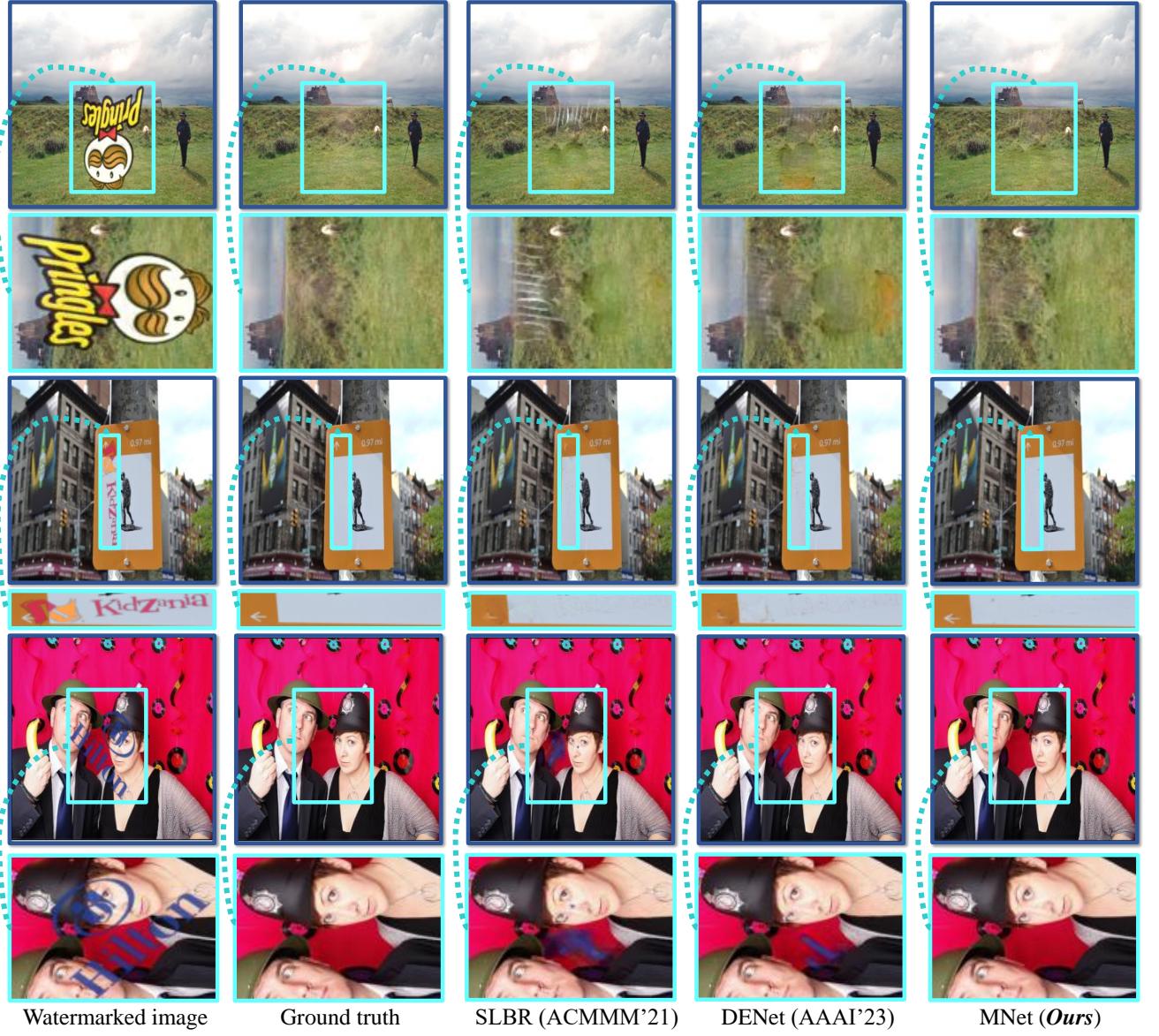


Figure 6: Qualitative comparisons of watermark-free images generated by our method and two latest and advanced methods (i.e. DENet and SLBR). The first column displays the input watermarked images, the second column contains corresponding ground truth background images, and the subsequent columns display watermark-free images generated by different methods. It can be observed that MNet can generate the watermark-free images with the best visual quality.

is set to 2^{-4} by default, and decreased to 1^{-6} with cosine annealing strategy [42]. We train our model for 100 epochs, while the input image resolution is set to 256×256 and the training batch is 16. In our proposed MNet, we set k_1 to 5, k_2 to 2, and k_3 to 1, where k_1 , k_2 and k_3 represent the number of U-Nets in each layer. The hyper-parameters in Eq. (8) are set as $\lambda_{L_1} = 1$, $\lambda_{perc} = 0.5$, $\lambda_{bce} = 0.5$, $\lambda_{iou} = 1$, and $\lambda_{dice} = 1$, respectively. Study on these hyper-parameters is given in the Section 4.4.2 *Studies on hyper-parameters*.

4.1.3. Evaluation Metrics

The goal of the visible watermark removal task is to generate a watermark-free image as similar to the ground truth

background image as possible. To evaluate this, we compare the average similarity of generated watermark-free images with the ground truth images by using similarity evaluation metrics, including peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [43], and learned perceptual image patch similarity (LPIPS) [34], which are all widely used for image similarity evaluation. Note that higher PSNR and SSIM values, as well as lower LPIPS values, indicate a greater similarity between the generated watermark-free image and the ground truth image.

Table 3

Model generalization comparison. Training Set: LOGO-H in the first row, indicates that the models are trained on the LOGO-H dataset. LOGO-H in the second row, indicates that the models are tested on the LOGO-H dataset. The best results are highlighted in bold.

Method	Training Set: LOGO-H			Training Set: LOGO-L			Training Set: LOGO-Gray		
	LOGO-H	LOGO-L	LOGO-Gray	LOGO-H	LOGO-L	LOGO-Gray	LOGO-H	LOGO-L	LOGO-Gray
	PSNR↑	PSNR↑	PSNR↑	PSNR↑	PSNR↑	PSNR↑	PSNR↑	PSNR↑	PSNR↑
(CVPR'19) BVMR [15]	36.51	39.18	35.06	31.07	40.24	33.37	31.75	37.34	38.90
(AAAI'21) SplitNet [16]	40.05	41.39	36.50	33.59	42.53	35.33	33.35	38.54	42.01
(ACMMM'21) SLBR [17]	40.56	41.59	36.60	33.80	44.10	35.80	33.52	39.05	42.21
(AAAI'23) DENet [18]	40.83	41.03	36.27	33.54	44.24	35.71	33.33	38.75	42.60
MNet (Ours)	44.34	45.56	39.58	36.32	46.25	37.46	34.46	40.16	46.67

Table 4

Comparisons for mask prediction. The F_1 and IoU scores are in percentage. The best results are highlighted in **bold**.

Method	LOGO-H		LOGO-L		LOGO-Gray	
	$F_1 \uparrow$	IoU↑	$F_1 \uparrow$	IoU↑	$F_1 \uparrow$	IoU↑
BVMR [15]	85.43	73.29	72.76	59.72	80.66	68.75
SplitNet [16]	85.83	74.06	72.80	59.98	80.43	68.63
SLBR [17]	87.51	75.65	78.01	65.42	84.18	72.39
DENet [18]	85.27	73.41	75.28	62.36	80.82	68.90
MNet (Ours)	91.75	80.88	83.35	70.35	89.05	78.37

4.2. Comparisons with State-of-the-art Methods

Quantitative comparisons between our method and other advanced approaches are summarized in Table 2. Among these methods, UNet [38], SIRF [22], BS²AM [28], DHAN [28] are migrated from related tasks such as blind image harmonization, shadow removal, etc. BVMR [15], SplitNet [16], SLBR [17] and DENet [18] are the latest approaches dedicated to visible watermark removal. As seen, the experimental results of MNet outperform those state-of-the-art approaches in all evaluation metrics, including those two-stage networks [16, 17, 18], which demonstrates the effectiveness of our proposed method. Notably, in comparison with the latest advanced method, DENet [18], MNet achieves significant PSNR improvement of **3.51 dB** on LOGO-H dataset, **2.01 dB** on LOGO-L dataset and **4.07 dB** on LOGO-Gray dataset, demonstrating that the watermark-free images generated by MNet are closer to the ground truth images. In Figure 6, we also present the qualitative comparisons, where the watermark-free images generated by MNet exhibit finer details and have the best visual quality. For instance, in the second row and last row, images generated by other methods often have afterimages or blurry details, whereas images generated by MNet are with very clear textures.

To evaluate the generalization capabilities of various visible watermark removal methods, we subject the models trained by different methods to be applied to multiple datasets, as shown in Table 3. It can be observed that the

models trained by MNet consistently outperform the models trained by other methods when applied to datasets not used during training. This suggests that compared to other methods, MNet can more effectively handle watermarks of different sizes, transparency, and colors, and that MNet does not overfit the training data. Additionally, we randomly collect a series of exquisite images. On these images, we embed various watermarks (including the emblem of Sun Yat-sen University) with differing levels of transparency at random positions. These watermarked images are then processed through multiple visible watermark removal models (all trained on the LOGO-H dataset). The resulting watermark-free images are displayed in Figure 7. It can be observed that our MNet achieves the most satisfactory results in both watermark removal and background restoration. This superior performance highlights MNet’s practical value and potential for real-world applications.

Note that the methods mentioned above, i.e., BVMR [15], SplitNet [16], SLBR [17], DENet [18] and our proposed MNet all require using Eq. (1) to generate the final output. Therefore, the accuracy of the predicted mask will affect the quality of the generated watermark-free image, which can be obtained by computing the F_1 score and the IoU score between the predicted mask and the ground truth mask. The F_1 scores and IoU scores corresponding to different methods are shown in Table 4. As seen, compared with other methods, our proposed MNet can predict watermarked areas the most accurately. Some visualization results are presented in Figure 8. It can be observed that the masks predicted by MNet are the closest to the ground truth masks, while the masks predicted by DENet [18] and SLBR [17] are incomplete and contain many incorrectly predicted pixels. This demonstrates the accuracy of MNet in locating watermarks.

In Table 5, we give the number of model parameters (Params) for each method and the average inference time for each method to process a watermarked image to evaluate the model complexity of each method. It can be seen that compared with other methods, our method increases some complexity (about 7%), but it is worth noting that our method

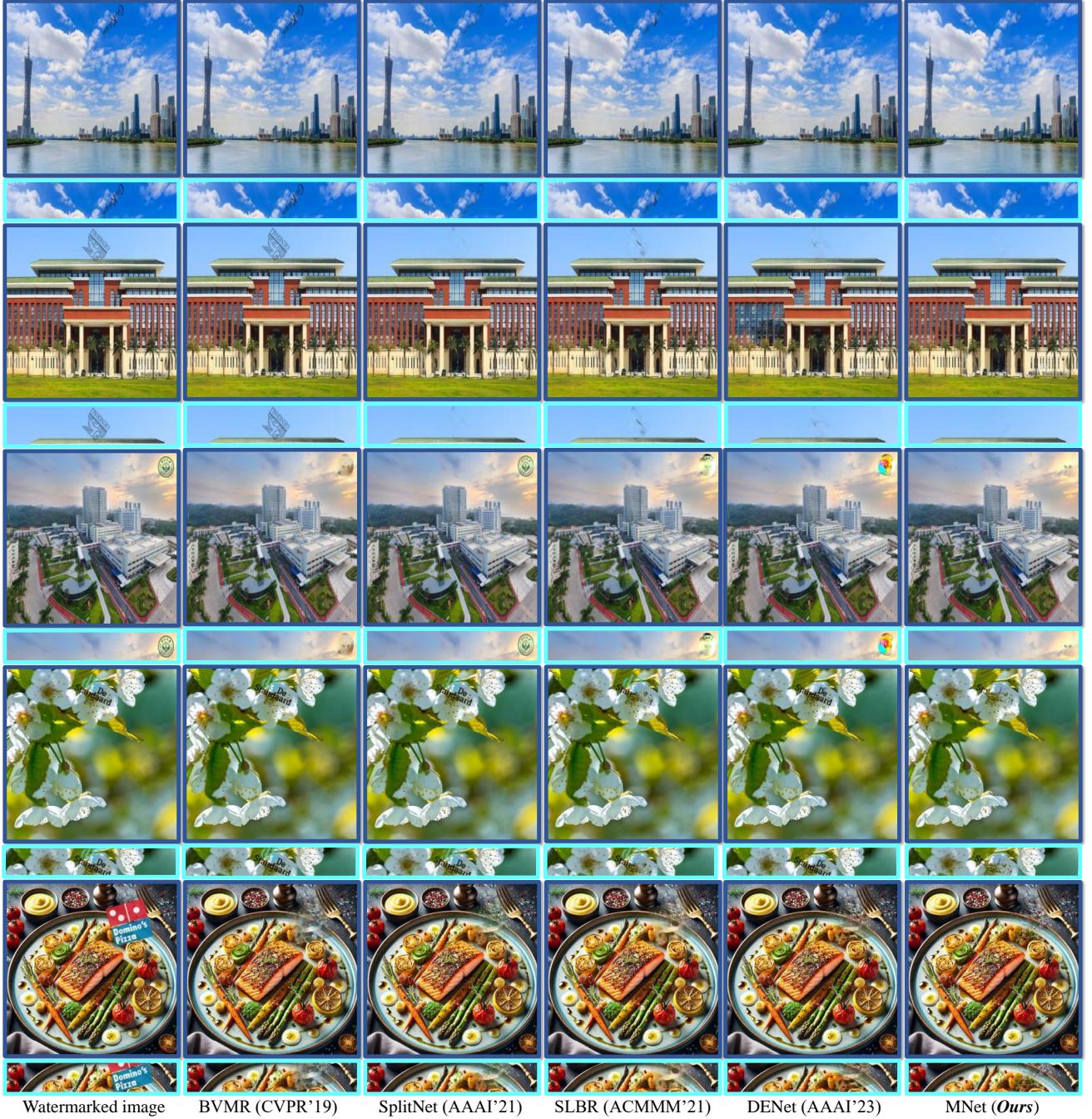


Figure 7: Comparison of watermark removal results on randomly selected watermarked images using various models that have been trained on the LOGO-H dataset. The first column shows the watermarked images. The subsequent columns display the results of different watermark removal models: BVMR [15], SplitNet [16], SLBR [17], DENet [18], and our proposed MNet. It can be observed that the proposed MNet demonstrates superior performance in both watermark removal and background restoration, highlighting its practical application value.

improves the PSNR value on the LOGO-H dataset by 3.51–7.83dB (about 22%), which indicates that MNet can significantly improve the quality of the generated watermark-free images. Even with a slight increase in complexity, the negative impact is negligible in today’s computing environment.

4.3. Ablation Studies

In our proposed MNet, several new strategies have been introduced, such as multi-scale feature extraction strategy (MFE), cross-layer feature fusion (Cr-F), and intra-layer feature fusion (In-F). Next, We will evaluate the effectiveness of each strategy through ablation experiments. In addition, as introduced in previous work [15], sharing partial parameters

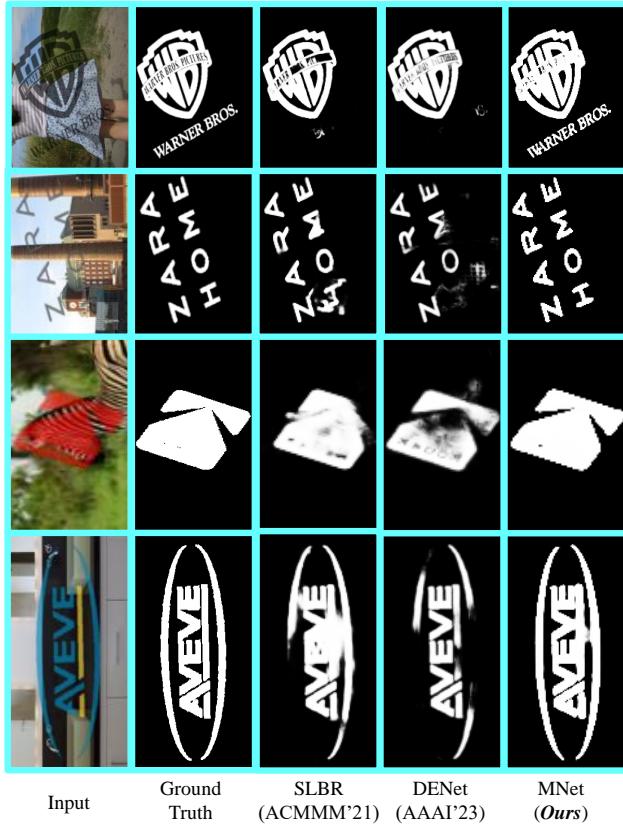


Figure 8: Visualization comparisons of predicted masks in details. In order to facilitate comparisons, we do not display the complete images. The displayed image is the watermark area and its surrounding area of the complete image.

Table 5

Model complexity comparisons. H, L, and Gray in the first row refer to LOGO-H, LOGO-L, and LOGO-Gray datasets, respectively. The best results are highlighted in **bold**.

Method	Parms ($\times 10^6$)	Inference Time (ms)	H	L	Gray
			PSNR↑	PSNR↑	PSNR↑
BVMR [15]	20.51	128.58	36.51	40.24	38.90
SplitNet [16]	32.62	140.82	40.05	42.53	42.01
SLBR [17]	21.35	140.34	40.56	44.10	42.21
DENet [18]	22.42	137.71	40.83	44.24	42.60
MNet (<i>Ours</i>)	33.43	151.86	44.34	46.25	46.67

between the background restoration branch and mask prediction branch can usually improve the generation performance of the visible watermark removal models, and we will also evaluate the impact of it on MNet below.

In the beginning, we construct the simplest model called BaseNet, which is a single-layer structure consisting of two identical branches. Each branch has eight cascaded U-Nets, without feature fusion and parameter sharing. The ablation experimental results are shown in Table 6, where “✓” and “✗” indicate whether the aforementioned strategies,

Table 6

Ablation studies on various strategies and modules. Experiments are conducted on the LOGO-H dataset. The best results are highlighted in **bold**.

BaseNet	Shared	MFE	Cr-F	In-F	Params ($\times 10^6$)	PSNR↑
✓	✗	✗	✗	✗	40.06	42.33
✓	✓	✗	✗	✗	32.55	42.78
✓	✓	✓	✗	✗	32.78	44.07
✓	✓	✓	✓	✗	32.97	44.10
✓	✓	✓	✓	✓	33.43	44.34

Table 7

Analysis of employing dice loss. Experiments are conducted on the LOGO-H dataset. The best results are highlighted in **bold**.

BaseNet	Shared	MFE	Cr-F	In-F	\mathcal{L}_{dice}	PSNR↑	$F_1(\%) \uparrow$
✓	✓	✓	✓	✓	✗	44.18	91.71
✓	✓	✓	✓	✓	✓	44.34	91.75

such as MFE, Cr-F, and In-F, are included in our testing. Our ablation experiments are conducted as follows. Firstly, use the BaseNet to generate the watermark-free image directly. Then, generate the watermark-free image with sharing the parameters of the first three U-Nets between the two branches. Thirdly, generate the image with the MFE strategy further, where the eight U-Nets in each branch are placed into three layers (5, 2, and 1 U-Nets in the first, second, and third layers, respectively) and the scale reduction module is used to provide input for each layer. Lastly, generate the watermark-free image with Cr-F and In-F. As seen from Table 6, each strategy has its contribution to the final MNet and does not significantly increase the number of model parameters. Totally, our proposed MNet can improve the PSNR value by **2.01 dB** and can reduce the number of parameters by 6.53M compared to the BaseNet.

Furthermore, considering that we additionally employ the dice loss to supervise the predicted mask, to access its impact, we record the performance of MNet with and without its supervision, as shown in Table 7. It can be observed that MNet with dice loss for supervision can generate the watermark-free images with higher visual quality and the masks with higher accuracy.

4.4. Supplementary Analyses

As introduced before, the proposed MNet allows stacking a variable number of U-Nets in each layer. In this section, we will first analyze the impact of the number of U-Nets on the performance of MNet. Then, we will give a study on hyper-parameters in Eq. (8). In addition, considering that the proposed MNet is a single-stage network, however, the recent mainstream methods all adopt two-stage networks [16, 17, 18], at the end of this section, we will give a discussion on the superiority of the single-stage network adopted by MNet.

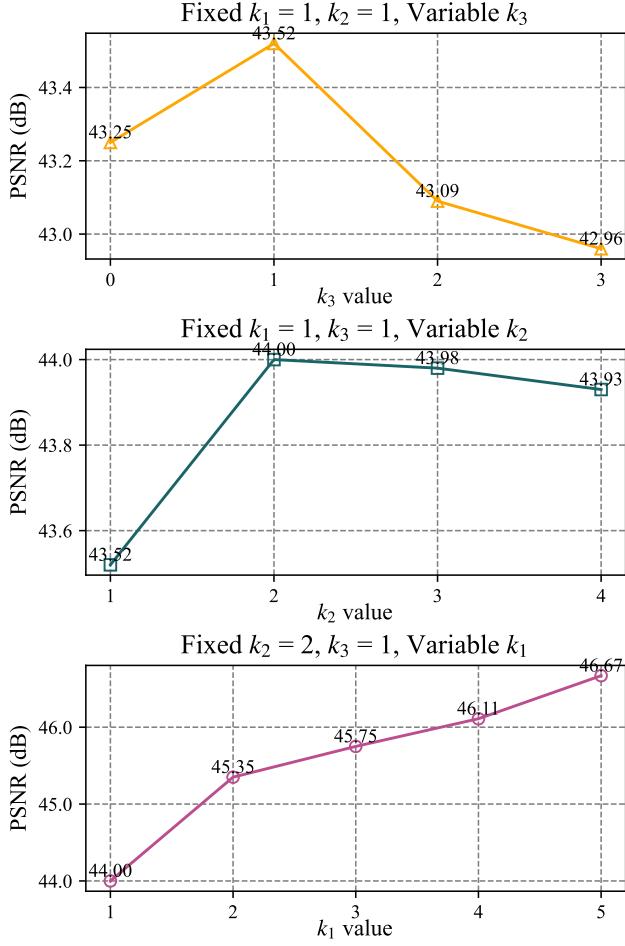


Figure 9: The PSNR values between the watermark-free images generated by MNet and the ground truth images under different k -value settings. k_1 , k_2 and k_3 represent the number of U-Nets in layer 1, layer 2, and layer 3, respectively. Experiments are conducted on the LOGO-Gray dataset.

4.4.1. Number of U-Nets Analysis

Note that the number of U-Nets in different layers can vary infinitely, thus, we can only conduct partial experiments. Here, we vary the number of U-Nets in one layer while keeping the number of U-Nets in the other two layers constant to analyze the impact of the number of U-Nets on MNet performance. The experimental results corresponding to various configurations of parameters k_1 , k_2 , and k_3 are shown in Figure 9. It can be observed from Figure 9 that increasing the number of U-Nets in the first layer will result in a continuous increase in PSNR values. When the number of U-Nets in the second or third layer increases, the PSNR value usually increases first and then decreases. This is because the images processed in the second and third layers have been downsampled or pixel-unshuffled. Compared to the original image processed in the first layer, these downsampled or pixel-unshuffled images have less texture information. Therefore, increasing the number of U-Nets in these layers may not increase the network's learning capacity, but it increases the computational cost of the network and may

Table 8

Study on hyper-parameters in Eq. (8). Experiments are conducted on the LOGO-H dataset. The best results are highlighted in **bold**.

λ_{L_1}	λ_{vgg}	λ_{bce}	λ_{iou}	λ_{dice}	PSNR↑
1	0.5	0.25	0.25	0.5	44.21
1	0.5	0.5	0.5	1	44.32
1	0.5	0.5	1	1	44.34
0.5	0.5	0.5	1	1	44.21
1	1	0.5	1	1	44.31
1	0.5	1	1	1	44.25
1	0.5	0.5	0.5	1	44.26
1	0.5	0.5	1	0.5	44.30

Table 9

Studies of stages. $MNet_{k_1, k_2, k_3}$ represent that there are k_1 U-Nets in layer 1, k_2 U-Nets in layer 2, and k_3 U-Nets in layer 3. #U-Nets refers to the total number of U-Nets. Experiments are conducted on the LOGO-H dataset. The best results are highlighted in **bold**.

Method	First stage	Second stage	#U-Nets	Parms ($\times 10^6$)	PSNR↑
Multi-stage network					
SplitNet [16]	✓	✓	N/A	32.62	40.05
SLBR [17]	✓	✓	N/A	21.35	40.56
DENet [18]	✓	✓	N/A	22.42	40.83
MultiNet ₁	MNet _{1,1,1}	MNet _{1,1,1}	6	21.18	41.70
MultiNet ₂	MNet _{1,1,1}	MNet _{4,1,1}	9	36.55	42.39
MultiNet ₃	MNet _{4,1,1}	MNet _{1,1,1}	9	36.55	44.31
Single-stage network					
BVMR [15]	✓	✗	N/A	20.51	36.51
SingleNet ₁	MNet _{1,1,1}	✗	3	10.59	40.01
SingleNet ₂	MNet _{4,1,1}	✗	6	25.96	43.82
SingleNet ₃	MNet _{5,2,1}	✗	8	33.43	44.34

even lead to performance degradation due to overfitting or optimization problems.

4.4.2. Study on Hyper-parameters

We modify the values of hyper-parameters in Eq. (8) and then record the PSNR values between the final generated watermark-free images and ground truth images. As shown in Table 8, when $\lambda_{L_1} = 1$, $\lambda_{perc} = 0.5$, $\lambda_{bce} = 0.5$, $\lambda_{iou} = 1$, and $\lambda_{dice} = 1$, the PSNR is the highest, thus, we set them as the hyper-parameter values in Eq. (8). Regarding the value of the hyper-parameter λ_{dice} , we first conduct an ablation study on dice loss (Table 7) to confirm that it should not be set to 0. Following this, through the study in this section, we determine the optimal values of λ_{dice} and other hyper-parameters. In addition, as shown in Table 8, MNet performs well under various combinations of the hyper-parameters.

4.4.3. Single-Stage and Two-Stage Network Analysis

In this part, we construct a two-stage network, MultiNet_{*}, where two standard MNets are served as the first and second stage networks. The output of the first stage is input to the second stage for refinement. Some comparison results between the two-stage networks (denoted as MultiNet_{*}), single-stage networks (denoted as SingleNet_{*}) and existing methods (BVMR [15], SplitNet [16], SLBR [17] and DENet [18]) are summarized in Table 9. Comparisons between MultiNet₁ and SingleNet₁, MultiNet₂ and SingleNet₁, as well as MultiNet₃ and SingleNet₂, demonstrate that adding a second stage for refinement can typically improve the visual quality of watermark-free images. However, there are more U-Nets and more learnable parameters in these two-stage networks than in these single-stage networks in general. When the same number of U-Nets is utilized, the visual quality of the watermark-free image obtained by a single-stage network is significantly better than that of a two-stage network. For example, the PSNR value of SingleNet₂ is **2.12 dB** higher than that of MultiNet₁. Moreover, SingleNet₃ has fewer U-Nets and fewer learnable parameters than MultiNet₃, however, it still can achieve higher PSNR value, which indicates that a well-optimized single-stage network can surpass a two-stage network in visible watermark removal task. In addition, as observed from Table 9, compared to the traditional single-stage network BVMR [15], multi-stage networks such as SplitNet [16], SLBR [17], and DENet [18] improve PSNR values by 3.54 to 4.32 dB. However, when compared to SingleNet₁, which adopts the proposed efficient multi-scale design, these multi-stage networks (i.e., SplitNet [16], SLBR [17], and DENet [18]) only achieve 0.04 to 0.82 dB higher PSNR, while the total number of model parameters in SingleNet₁ is only 32.5% to 49.6% of the total number of model parameters in these multi-stage networks. After upgrading SingleNet₁ to SingleNet₂, the model parameters in SingleNet₂ (25.96×10^6) become comparable to those in the multi-stage networks (22.42×10^6 to 32.62×10^6), but in terms of the PSNR value, SingleNet₂ outperforms these multi-stage networks by 2.99 to 3.77 dB. Overall, the experimental results in Table 9, as well as those in Table 2, Table 3 and Table 4, suggest that focusing on optimizing a single-stage network is more efficient than building a two-stage network, which challenges the recent trends [16, 17, 18] in visible watermark removal researches.

5. Conclusion

In this paper, we address the task of visible watermark removal from a new perspective and propose a novel single-stage and multi-scale network called MNet. As a single-stage network, MNet benefits from a simplified optimization process and avoids the error accumulation often seen in multi-stage networks. By utilizing the multi-scale framework and feature fusion schemes, MNet can significantly improve the visual quality of the generated watermark-free images, thus avoiding the problems of incomplete watermark localization and low quality of generated watermark-free images that

may exist in traditional single-stage networks. In addition, MNet has a flexible architecture, and its performance can be further improved by optimizing the number of stacked U-Nets in different layers.

6. Future Work

Future work can explore extending MNet to other image processing tasks, such as denoising and image restoration. These tasks share similarities in feature extraction and refinement, allowing them to benefit from MNet's multi-scale and feature fusion designs. Additionally, future work could focus on enhancing MNet's real-time application capabilities. While MNet has achieved excellent results, the complexity can be further reduced to support real-time deployment. As previously explored, reducing the number of U-Nets can help simplify the architecture. Furthermore, additional simplification of the network structure or leveraging hardware acceleration can make MNet more suitable for real-time applications.

Acknowledgments

This work is supported by the National Natural Science Foundation of China U2336208, 62072481 and 62472454, Guangdong Provincial Key Laboratory of Information Security Technology 2023B1212060026, and the 2024 Graduate Education Innovation Project (Project No. 2024_73120_B24747).

References

- [1] I. Cox, M. Miller, J. Bloom, C. Honsinger, Digital watermarking, Journal of Electronic Imaging 11 (2002) 414–414.
- [2] S. Katzenbeisser, F. Petitcolas, Digital watermarking, Artech House, London 2 (2000).
- [3] R. Hu, S. Xiang, Cover-lossless robust image watermarking against geometric deformations, IEEE Transactions on Image Processing 30 (2020) 318–331.
- [4] B. Zhang, Y. Wu, B. Chen, et al., Embedding guided end-to-end framework for robust image watermarking, Security and Communication Networks 2022 (2022).
- [5] J. Zhang, X. Li, Y. Zhao, A new robust watermarking algorithm based on intra-frame difference, in: 2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP), IEEE, 2022, pp. 1–5.
- [6] Y. Liu, Z. Zhu, X. Bai, Wdnet: Watermark-decomposition network for visible watermark removal, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 3685–3693.
- [7] C.-H. Huang, J.-L. Wu, Attacking visible watermarking schemes, IEEE transactions on multimedia 6 (2004) 16–30.
- [8] S.-C. Pei, Y.-C. Zeng, A novel image recovery algorithm for visible watermarked images, IEEE Transactions on information forensics and security 1 (2006) 543–550.
- [9] J. Park, Y.-W. Tai, I. S. Kweon, Identigram/watermark removal using cross-channel correlation, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 446–453.
- [10] T. Dekel, M. Rubinstein, C. Liu, W. T. Freeman, On the effectiveness of visible watermarks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2146–2154.
- [11] D. Cheng, X. Li, W.-H. Li, C. Lu, F. Li, H. Zhao, W.-S. Zheng, Large-scale visible watermark detection and removal with deep convolutional networks, in: Pattern Recognition and Computer Vision: First Chinese Conference, PRCV 2018, Guangzhou, China, November 23–26, 2018, Proceedings, Part III 1, Springer, 2018, pp. 27–40.

- [12] Y. Gandelsman, A. Shocher, M. Irani, "double-dip": unsupervised image decomposition via coupled deep-image-priors, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11026–11035.
- [13] Z. Cao, S. Niu, J. Zhang, X. Wang, Generative adversarial networks model for visible watermark removal, *IET Image Processing* 13 (2019) 1783–1789.
- [14] X. Li, C. Lu, D. Cheng, W.-H. Li, M. Cao, B. Liu, J. Ma, W.-S. Zheng, Towards photo-realistic visible watermark removal with conditional generative adversarial networks, in: Image and Graphics: 10th International Conference, ICIG 2019, Beijing, China, August 23–25, 2019, Proceedings, Part I 10, Springer, 2019, pp. 345–356.
- [15] A. Hertz, S. Fogel, R. Hanocka, R. Giryes, D. Cohen-Or, Blind visual motif removal from a single image, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6858–6867.
- [16] X. Cun, C.-M. Pun, Split then refine: stacked attention-guided resnets for blind single image visible watermark removal, in: Proceedings of the AAAI conference on artificial intelligence, volume 35, 2021, pp. 1184–1192.
- [17] J. Liang, L. Niu, F. Guo, T. Long, L. Zhang, Visible watermark removal via self-calibrated localization and background refinement, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 4426–4434.
- [18] R. Sun, Y. Su, Q. Wu, Denet: Disentangled embedding network for visible watermark removal, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, 2023, pp. 2411–2419.
- [19] S. Li, W. Ren, F. Wang, I. B. Araujo, E. K. Tokuda, R. H. Junior, R. M. Cesar-Jr, Z. Wang, X. Cao, A comprehensive benchmark analysis of single image deraining: Current challenges and future perspectives, *International Journal of Computer Vision* 129 (2021) 1301–1322.
- [20] J. Nie, J. Xie, J. Cao, Y. Pang, Context and detail interaction network for stereo rain streak and raindrop removal, *Neural Networks* 166 (2023) 215–224.
- [21] C. Zhao, W. Cai, C. Hu, Z. Yuan, Cycle contrastive adversarial learning with structural consistency for unsupervised high-quality image deraining transformer, *Neural Networks* (2024) 106428.
- [22] X. Zhang, R. Ng, Q. Chen, Single image reflection separation with perceptual losses, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4786–4794.
- [23] X. Cun, C.-M. Pun, C. Shi, Towards ghost-free shadow removal via dual hierarchical aggregation network and shadow matting gan, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 10680–10687.
- [24] B. Li, Y. Gou, S. Gu, J. Z. Liu, J. T. Zhou, X. Peng, You only look yourself: Unsupervised and untrained single image dehazing neural network, *International Journal of Computer Vision* 129 (2021) 1754–1767.
- [25] G. Fan, M. Gan, B. Fan, C. P. Chen, Multiscale cross-connected dehazing network with scene depth fusion, *IEEE Transactions on Neural Networks and Learning Systems* 35 (2022) 1598–1612.
- [26] H. Sun, B. Li, Z. Dan, W. Hu, B. Du, W. Yang, J. Wan, Multi-level feature interaction and efficient non-local information enhanced channel attention for image dehazing, *Neural Networks* 163 (2023) 10–27.
- [27] J. Wang, S. Wu, Z. Yuan, Q. Tong, K. Xu, Frequency compensated diffusion model for real-scene dehazing, *Neural Networks* 175 (2024) 106281.
- [28] X. Cun, C.-M. Pun, Improving the harmony of the composite image by spatial-separated attention module, *IEEE Transactions on Image Processing* 29 (2020) 4759–4771.
- [29] K. Kim, S. Lee, S. Cho, Mssnet: Multi-scale-stage network for single image deblurring, in: European Conference on Computer Vision, Springer, 2022, pp. 524–539.
- [30] S. Lee, H. Kume, H. Urakubo, H. Kasai, S. Ishii, Tri-view two-photon microscopic image registration and deblurring with convolutional neural networks, *Neural Networks* 152 (2022) 57–69.
- [31] J. Zhang, W. Zhai, Blind attention geometric restraint neural network for single image dynamic/defocus deblurring, *IEEE Transactions on Neural Networks and Learning Systems* 34 (2022) 8404–8417.
- [32] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, Z. Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1874–1883.
- [33] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14, Springer, 2016, pp. 694–711.
- [34] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 586–595.
- [35] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556* (2014).
- [36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.
- [37] F. Milletari, N. Navab, S.-A. Ahmadi, V-net: Fully convolutional neural networks for volumetric medical image segmentation, in: 2016 fourth international conference on 3D vision (3DV), Ieee, 2016, pp. 565–571.
- [38] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.
- [39] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.
- [40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, *Advances in neural information processing systems* 32 (2019).
- [41] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014).
- [42] I. Loshchilov, F. Hutter, Sgdr: Stochastic gradient descent with warm restarts, *arXiv preprint arXiv:1608.03983* (2016).
- [43] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE transactions on image processing* 13 (2004) 600–612.