

Visible Watermark Removal with Dynamic Kernel and Semantic-aware Propagation

Xing Zhao
 1033874657@sjtu.edu.cn
 Li Niu*
 ustcnewly@sjtu.edu.cn
 Liqiang Zhang
 zhang-lq@cs.sjtu.edu.cn

MoE Key Lab of Artificial Intelligence
 Shanghai Jiao Tong University
 Shanghai, China

Abstract

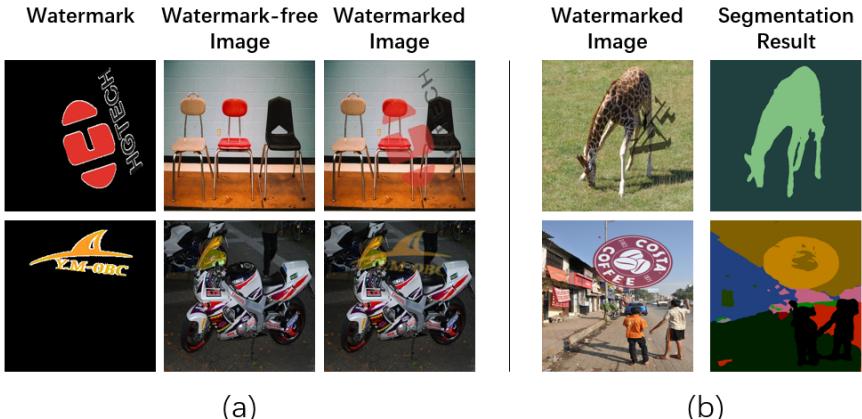
Visible watermarks are designed to protect the copyright of images. Inversely, visible watermark removal studies how to enhance the removal resistance of visible watermarks. The existing watermark removal methods have achieved competitive results, but they cannot cope with diverse watermarks and complicated semantics very well. For the problem of diverse watermarks, we propose watermark-specific dynamic kernel, which can detect and remove watermark adaptively considering the specific properties of different watermarks. For the problem of complicated semantics, we develop semantic-aware propagation, which aims to reconstruct the corrupted pixels by borrowing information from semantically similar pixels in the neighboring region. We conduct ample experiments on two benchmark datasets, demonstrating that our method outperforms previous methods.

1 Introduction

As a prevalent information carrier nowadays, images are ubiquitous in a wide range of applications, during which the copyright issue becomes critical for information security. To solve this problem, we can superimpose visible watermarks on images to clarify and protect the copyright. As an inverse process, visible watermark removal aims to erase the visible watermark and reconstruct the background image. Visible watermark removal could provide informative and useful hints for developing more robust watermarking techniques. In recent years, this research direction has attracted growing interest of many scholars [2, 4, 6, 8, 19, 25], but the state-of-the-art performances are still far below our expectation. Note that background image is also called watermark-free image, so two terms “watermark-free image” and “background image” are used interchangeably. In Figure 1(a), we show a pair of watermark and watermark-free image, as well as the obtained watermarked image.

Early visible watermark removal methods require either prior knowledge (*e.g.*, location) of the watermark [20, 30, 32] or strict assumption of the watermark (*e.g.*, the same watermark is present in different image) [8, 12]. To eliminate the requirement of prior knowledge

*Corresponding author.



(a)

(b)

Figure 1: (a) Given a pair of watermark and watermark-free image, we superimpose the watermark on the watermark-free image to yield a watermarked image. (b) The segmentation results using HRNet+OCR [42] pretrained on COCO-stuff dataset [27].

and strict assumption, many methods based on deep learning have been developed recently [2, 4, 6, 19, 25, 26, 28]. Several methods [2, 25, 28] employed generative adversarial networks to remove the watermark and recover realistic background images. Besides the output watermark-free image, a few methods [6, 26] designed multi-task framework with one encoder and multiple decoders, which have additional outputs (*e.g.*, watermark pattern and watermark mask). Although these methods have greatly advanced the research on visible watermark removal, they may produce low-quality images with blurring and distortion, when meeting diverse watermarks and complicated semantics.

In this paper, a novel visible watermark removal method called DKSP is developed, which is equipped with Dynamic Kernel and Semantic-aware Propagation. Following previous works [6, 26], we adopt a multi-task learning framework to cope with two tasks: watermark localization and background restoration. One encoder is shared by two tasks and two decoders are employed for two tasks separately. Specifically, the mask decoder produces watermark masks of different scales, and the background decoder takes these predicted watermark masks as auxiliary information. Previous watermark removal methods often fail to handle diverse watermarks and complicated semantics, which motivates us to make two contributions under this multi-task learning framework. On one hand, considering that the watermarks have a diversity of colors, shapes, and patterns, we adapt the network to different types of watermarks using dynamic convolutional filter [15, 21, 29, 35]. Specifically, we design a dynamic kernel module (DKM) to dynamically generate the convolutional kernel according to the watermark feature, so that the network can adaptively cope with different types of watermarks. We insert DKM to both decoders. On the other hand, we explore incorporating semantic information into the network, which has never been studied in the field of watermark removal. When using pretrained segmentation model to segment a watermarked image, we observe that the segmentation model usually ignores the watermark region, or classifies the watermark region as an isolated object, as illustrated in Figure 1(b). In either case, it is useful to transfer information from semantically similar pixels in the neighboring region to the watermarked pixels. To this end, we develop a semantic-aware propagation

module (SPM), which is inserted into the background decoder.

In this work, we have the following contributions: 1) We design a novel dynamic kernel module, which enables the network to tackle various types of watermarks adaptively; 2) We contribute a novel semantic-aware propagation module, which can help restore the watermarked region using semantic information; 3) We perform comprehensive experiments using two benchmark datasets, proving the advantage of our developed approach.

2 Related Work

Visible Watermark Removal: As the inverse process of adding watermark, watermark removal is of great significance to enhancing the robustness of watermark, so it has attracted considerable attention from many researchers. In [25] and [2], they leveraged generative adversarial network to remove the watermark. Compared with the above two works, [28] also used generative adversarial network, but in addition to the watermark-free image, the generator also outputs the watermark mask and watermark pattern. In [19], they used the form of a single encoder and multiple decoders to predict watermark-free images, watermarked images, and watermarked masks. The final output is constructed by concatenating the input and watermark-free images in light of the watermark mask. In [6], they also designed multi-task framework with ResUNet being the backbone network. In the recent work [26], on the basis of the multi-task ResUNet architecture, they designed three additional modules to promote the prediction quality of watermark masks and watermark-free images. The above methods still have difficulty in tackling diverse and complex watermarked images. Therefore, we design dynamic kernel to deal with diverse watermarks adaptively and leverage auxiliary semantic information to boost the performance.

Image Content Removal: Visible watermark removal belongs to a broad scope named image content removal, which includes myriads of tasks such as image deraining [11, 33, 37, 41], image dehazing [1, 5, 10, 18, 40, 43, 44], shadow removal [7, 9, 38]. To name a few, the deraining method [33] adopted generative adversarial network, in which the generator consists of a RNN and an auto-encoder. The RNN extracts the raindrop region attention of the image, which is injected into the auto-encoder and the discriminator to improve the quality of the derain. The shadow removal method [7] proposed the dual hierarchically aggregation network to fully exploit multi-scale features to promote the effect of shadow removal. The network uses one backbone network and two attentive aggregation heads to generate shadow masks and shadow-free images respectively.

Dynamic Convolutions: Dynamic convolution has been extensively used in distinct deep learning tasks, which has been discussed and classified in [16]. Here we mainly mean dynamic parameters. Dynamic parameters aim to produce input-dependent parameters, which can be further divided into parameter adjustment [13, 17], weight prediction [15, 21, 29, 35] and dynamic features [3, 24]. Our dynamic kernel module belongs to weight prediction in dynamic parameter methods. To our best knowledge, this work is the first one using dynamic kernel for visible watermark removal.

3 Our Method

Given a pair of watermark and watermark-free image I , we can overlay the watermark on the image using alpha blending to get the watermarked image J . As an reverse process,

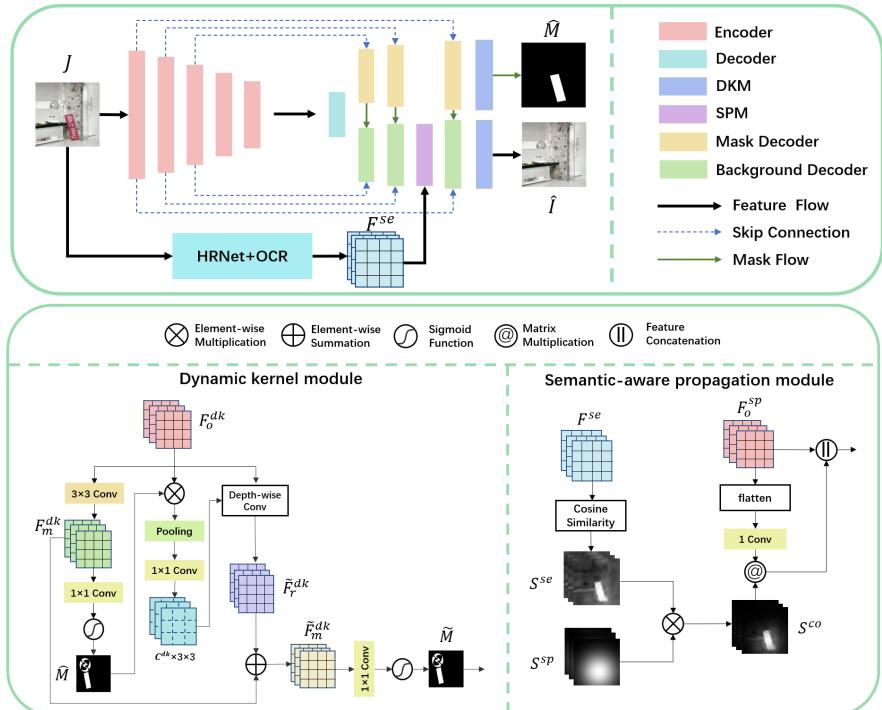


Figure 2: Top figure: our whole network structure with one encoder and two decoders, in which the mask decoder predicts watermark mask and the background decoder restores the background image. The dynamic kernel module (DKM) is inserted into the last block in both decoders and semantic-aware propagation module (SPM) is inserted before the penultimate block of background decoder. Bottom figure: the architectures of DKM and SPM. The detailed introductions to DKM and SPM can be found in Section 3.1 and 3.2 respectively.

visible watermark removal focuses on the goal of reconstructing I based on J . Since watermark mask M is not available, we need to predict the watermark mask while removing the watermark. As shown in Figure 2, we resort to the U-Net [34] architecture, in which we design the encoder and decoder block structures following [19]. Our network includes a shared encoder and two decoders responsible for watermark localization (mask decoder) and background image reconstruction (background decoder) respectively. The first block of the two decoders is shared, while the last three blocks are separated. In the mask decoder, we predict a mask in each block, which is delivered to the background decoder block corresponding to this mask decoder block. For mask prediction, we use binary cross-entropy loss: $\mathcal{L}_{mask} = -\sum_{i,j} (M_{i,j} \log \hat{M}_{i,j} + (1 - M_{i,j}) \log(1 - \hat{M}_{i,j}))$, where $M_{i,j}$ (resp., $\hat{M}_{i,j}$) is the (i, j) -th entry in M (resp., \hat{M}).

In each background decoder block, the encoder features from skip connection, the predicted mask from mask decoder, and the output from previous decoder block are concatenated as input. In the two decoders, we insert additional modules to enhance the performance. Specifically, we insert dynamic kernel module (DKM) to both decoders, and insert semantic-aware propagation module (SPM) to the background decoder. The required seman-

tic information in SPM is provided by an off-the-shelf HRNet+OCR [42] model pretrained on COCO-Stuff dataset [27]. The details of DKM will be introduced in Section 3.1 and the details of SPM will be shown in Section 3.2.

The background decoder produces the background image \hat{I} and we enforce \hat{I} to approach the ground-truth background image I by using L_1 loss and perceptual loss [22]: $\mathcal{L}_{bg} = ||I - \hat{I}||_1 + \lambda_{vgg} \sum_{k=1,2,3} ||\Phi_{vgg}^k(I) - \Phi_{vgg}^k(\hat{I})||_1$, where $\Phi_{vgg}^k(\cdot)$ represents the activation map of k -th layer in VGG16 [36]. In the end, we collect the losses above and arrive at: $\mathcal{L}_{all} = \mathcal{L}_{mask} + \lambda \mathcal{L}_{bg}$, where λ is a hyper-parameter.

3.1 Dynamic Kernel Module

Considering the diverse properties (*e.g.*, shape, color, pattern, transparency) of watermarks in different images, it is challenging for one set of convolution filters to localize and erase all types of watermarks. Inspired by previous works [15, 21, 35], we propose our Dynamic Kernel Module (DKM) with dynamic convolution filters generated based on watermark features. We insert DKM into the last block of two decoders, so that our network can deal with different types of watermarks adaptively.

In the following, we first introduce DKM in the mask decoder. As shown in the bottom left of Figure 2, in the last mask decoder block, we denote the last feature map as F_m^{dk} and the penultimate feature map as F_o^{dk} , in which F_m^{dk} is used to predict the mask \hat{M} . Although the estimated mask \hat{M} is not perfectly accurate, it can roughly capture the watermark feature, based on which we can generate the dynamic convolutional filters tailored to specific watermark. Specifically, this estimated mask \hat{M} is multiplied with the feature map F_o^{dk} . Then, we downsample the masked feature map to $3 \times 3 \times C^{dk}$ (C^{dk} is the channel dimension of F_o^{dk}) via spatial average pooling. The downsampled feature map passes through a 1×1 convolution layer to yield a dynamic kernel, which contains C^{dk} depth-wise convolutional filters with size 3×3 . Note that we predict depth-wise convolutional filters [14] for efficiency, with each convolutional filter acting upon each channel in the feature map. We apply the obtained dynamic kernel to F_o^{dk} by depth-wise convolution to produce residual feature map \tilde{F}_r^{dk} , which is added to F_m^{dk} to get the refined feature map \tilde{F}_m^{dk} . Finally, we get the refined mask \tilde{M} by passing \tilde{F}_m^{dk} through a 1×1 convolutional layer and a sigmoid layer. We adopt the mask loss \mathcal{L}_{mask} to supervise the refined mask. By using the dynamic kernel module, the predicted refined mask can spot some missed pixels and erase some misdetected pixels.

Recall that the dynamic kernel module is also used in the last block in background decoder. However, unlike the mask decoder, we directly use the refined mask \tilde{M} from mask decoder to supersede \hat{M} . Similar to mask decoder, we apply the generated dynamic kernel to obtain a residual feature map. Then, we add up residual feature map and original last feature map to get a refined feature map. Finally, refined feature map is delivered to a 1×1 convolutional layer to generate the background image \hat{I} . Although previous work [26] has also tried to adapt the network according to specific watermark, it simply concatenates watermark feature with original feature map. In contrast, we generate watermark-specific dynamic kernel to modulate the feature map more effectively.

3.2 Semantic-aware Propagation Module

The watermarked regions restored by previous watermark removal methods are often of low quality (*e.g.*, blurring, distortion, artifacts). We conjecture that propagating information from

semantically similar neighboring pixels can help recover the watermarked region. For instance, if a part of zebra is covered by watermark, we can borrow information from the remaining part of zebra to restore the texture and color of the watermarked region. Therefore, we propose a semantic-aware propagation module and insert it before the penultimate block of background decoder.

To find semantically similar regions, we resort to pretrained segmentation model HR-Net+OCR [42] pretrained on COCO-Stuff Dataset [27]. When applying this model to segment a watermarked image, one observation is that the segmentation model sometimes ignores the watermarked region, that is, the watermarked region is classified as the category of covered background (see top row in Figure 1(b)). In the other case, the segmentation model may misclassify the watermarked region as an isolated object (see bottom row in Figure 1(b)). In the former case, the information from semantically similar and nearby non-watermarked region can help recover the watermarked region. In the latter case, the information propagation within the watermarked region can also help reconstruct the watermarked region. To locate semantically similar and spatially near pixels for each target pixel, we first design semantic similarity map and spatial similarity map as follows.

Semantic similarity map: We use the pretrained segmentation model to extract the semantic feature map, based on which we can calculate the cosine similarity between the semantic feature vectors of two pixels. Considering efficiency, we downsample the semantic feature map to 32×32 and calculate pair-wise cosine similarities, resulting in a semantic similarity map $S^{se} \in \mathcal{R}^{1024 \times 1024}$.

Spatial similarity map: We additionally introduce a spatial similarity map S^{sp} , in which each spatial similarity value $s^{sp}(i, j) = \exp(-d^{sp}(i, j))$ with $d^{sp}(i, j)$ being L_2 distance between two pixels.

As illustrated in the bottom right of Figure 2, we denote the last feature map in the penultimate background decoder block as $F_o^{sp} \in \mathcal{R}^{H^{sp} \times W^{sp} \times C^{sp}}$. By denoting $N^{sp} = H^{sp} \times W^{sp}$, we resize S^{sp} and S^{se} to the same size $N^{sp} \times N^{sp}$, and multiple them element-wisely. Then, we perform row-wise L_1 normalization to get the normalized similarity map $S^{co} \in \mathcal{R}^{N^{sp} \times N^{sp}}$.

We reshape the feature map $F_o^{sp} \in \mathcal{R}^{H^{sp} \times W^{sp} \times C^{sp}}$ to $\bar{F}_o^{sp} \in \mathcal{R}^{N^{sp} \times C^{sp}}$ and pass it through a 1×1 convolutional layer to get $\bar{F}_v^{sp} \in \mathcal{R}^{N^{sp} \times C^{sp}}$. Similar to self-attention [39], we perform information propagation by attending relevant values for query pixels. Specifically, we treat the similarity map S^{co} as attention map and \bar{F}_v^{sp} as values. According to our motivation, we could only deem watermarked pixels as query pixels. However, the predicted watermark mask is not accurate and there might be some missed watermarked pixels. Besides, deeming all pixels as query pixels will bring no harm to the performance. Therefore, we deem all pixels as query pixels by multiplying similarity map S^{co} with \bar{F}_v^{sp} to acquire the attended values $\bar{F}_a^{sp} = S^{co} \bar{F}_v^{sp}$ for all pixels. Finally, we reshape $\bar{F}_a^{sp} \in \mathcal{R}^{N^{sp} \times C^{sp}}$ back to $F_a^{sp} \in \mathcal{R}^{H^{sp} \times W^{sp} \times C^{sp}}$ and concatenate it with F_o^{sp} channel-wisely. We decrease the channel dimension of concatenated feature map to C^{sp} and deliver it to the next decoder block.

4 Experiments

4.1 Experimental Setting

We use two datasets: Colored Large-scale Watermark Dataset (CLWD) [28] and LOGO30K Dataset (LOGO30K) [6]. Both datasets contain colored watermarks and backgrounds, which

Method	CLWD				LOGO30K			
	PSNR↑	RMSE _w ↓	IoU(%)↑	F ₁ ↑	PSNR↑	RMSE _w ↓	IoU(%)↑	F ₁ ↑
U-Net [34]	23.21	48.43	-	-	24.64	43.29	-	-
Qian <i>et al.</i> [33]	34.60	19.34	56.65	0.6910	36.89	17.26	62.68	0.7565
Cun <i>et al.</i> [7]	35.29	18.25	59.41	0.7122	37.67	16.88	65.13	0.7745
Li <i>et al.</i> [25]	27.96	46.80	-	-	30.51	39.11	-	-
Cao <i>et al.</i> [2]	29.04	41.21	-	-	32.18	35.16	-	-
WDNet [28]	35.53	17.27	61.20	0.7240	39.15	15.94	68.21	0.8010
BVMR [19]	35.89	18.71	70.21	0.7871	38.28	16.72	72.87	0.8305
SplitNet [6]	37.41	15.25	71.96	0.8027	41.27	14.85	74.14	0.8411
SLBR [26]	38.28	14.07	74.63	0.8234	41.50	14.69	78.58	0.8647
DKSP	38.84	12.16	77.30	0.8480	42.16	13.78	80.16	0.8770

Table 1: The comparison of various methods on CLWD [28] and LOGO30K [6]. PSNR and RMSE_w are used to evaluate the restored background image, while IoU and F₁ are used to evaluate the predicted watermark mask. “—” means that the method does not predict watermark mask. The best results are denoted in boldface.

are realistic and challenging. The details of two datasets are left to Supplementary.

Our method is implemented with Pytorch [31]. The input image size is set to 256 × 256. During training, we choose Adam [23] optimizer, in which the initial learning rate is 0.001, momentum parameters are $\beta_1 = 0.5$, $\beta_2 = 0.999$, batch size is 8. According to the quality of the predicted watermark-free images and watermark masks after several attempts, we empirically set λ_{vgg} to 0.001.

For baselines, we first compare with the following watermark removal methods: Li *et al.* [25], Cao *et al.* [2], WDNet [28], BVMR [19], SplitNet [6], SLBR [26]. In addition, following [26], we also select some typical image-to-image translation methods and image content removal methods for comparison. For typical image-to-image translation, we compare with U-Net [34]. Besides, we compare with Qian *et al.* [33] for deraining, and Cun *et al.* [7] for shadow removal.

In terms of evaluation metrics, we use Peak Signal-to-Noise Radio (PSNR), weighted Root-Mean-Square distance (RMSE_w) for the restored background images. We use Intersection over Union (IoU) and F1-score (F₁) for the predicted watermark masks.

4.2 Experimental Results

We list the results of our DKSP method and baseline methods in Table 1. One observation is that watermark removal methods [6, 19, 26, 28] and image content removal methods [7, 33] perform much better than the image-to-image translation methods [2, 25], which demonstrates the necessity of predicting watermark masks for watermark removal. In addition, baselines WDNet [28], BVMR [19], SplitNet [6], and SLBR [26], which are specifically designed for watermark removal, outperform other methods [7, 33] in the wider scope.

Our DKSP method beats all baseline methods, indicating the advantage of our designed modules. From Table 1, we can see that methods which can predict more accurate watermark mask tend to generate better background image, which again indicates the importance of mask prediction in watermark removal task.

For qualitative comparison, we show the reconstructed background images in Figure 3. We compare with the baselines [6, 19, 26, 28]. In each row, we exhibit the watermarked



Figure 3: Visualization of watermark-free images produced by various methods on CLWD [28] (first two rows) and LOGO30K [6] (last two rows). “Input” indicates the watermarked image and “GT” indicates the ground-truth background image. We zoom in the watermarked region in green bounding box for better observation.

image, the original background image, and the background images recovered using various approaches. Based on Figure 3, our method is able to recover the texture details and overall structure more accurately and coherently. For example, in the first two rows, baseline method struggles with removing the watermark color and recovering the background color (blue, gray), while our method is capable of removing the holistic watermark and restoring the texture/color of the background image better.

We also visualize the predicted watermark masks from our method and baselines (SLBR [26], SplitNet [6], BVMR [19], and WDNet [28]). From Figure 4, it can be observed that our mask decoder branch can predict the watermark mask more accurately, whereas the baselines are prone to have more missed detection and false alarms. For example, in the first and third rows, our predicted masks are closer to ground truth, whereas the predicted masks from the baseline methods are blurry or incomplete. In the fourth row, the baseline methods may include an extra region mistakenly (*e.g.*, SplitNet) or miss a large region (*e.g.*, WDNet), whereas our method can precisely localize the watermark in most cases.

4.3 Ablation Studies

We perform ablation studies on LOGO30K dataset to validate the necessity of each component in our method. At start, we discard the dynamic kernel module and semantic-aware propagation module. In this case, we obtain a basic network with one common encoder and two separate decoders for two tasks. The obtained results are listed in row 1 in Table 2.

Based on row 1, we add our dynamic kernel module (DKM) in two decoders, leading to the results in row 5. We can see that our DKM brings notable improvement for both

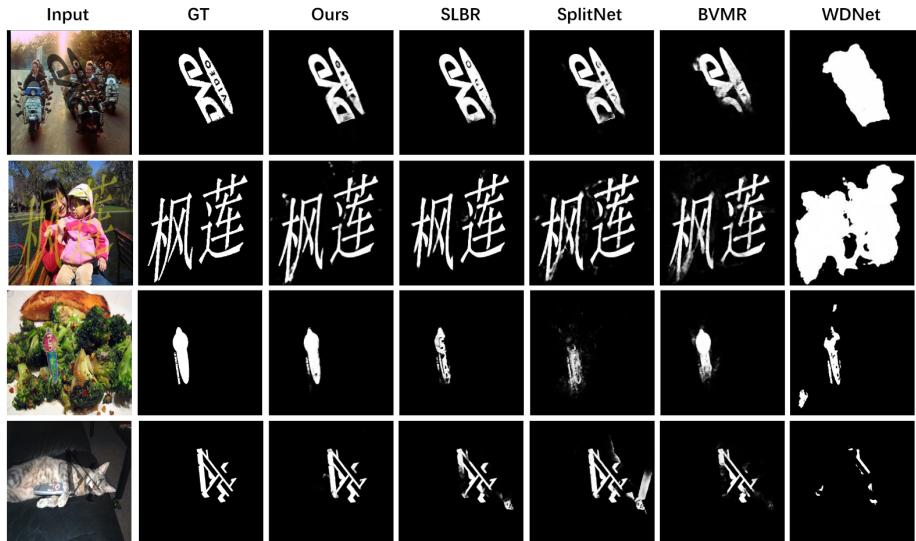


Figure 4: Visualization of predicted masks of various methods on CLWD [28] (the first two rows) and LOGO30K [6] (the last two rows). “Input” indicates the watermarked image and “GT” indicates the ground-truth watermark mask.

mask prediction and background reconstruction, which verifies the advantage of our proposed DKM. On the basis of row 5, we change the output of dynamic kernel module into non-residual form, that is, \tilde{F}_r^{dk} is directly used to predict mask or image without being added to F_m^{dk} . The results are placed in row 2, from which we can observe that residual design is indeed useful. Besides, we also use the whole feature map F_o^{dk} instead of masked feature map to generate dynamic kernel, corresponding to row 3 in Table 2. As mentioned in Section 3.1, [26] also considered adapting to specific watermark and proposed SMR module. We replace our DKM with SMR and list the results in row 4. The performances in row 5 are better than those in row 3 and row 4, which shows the advantage of generating dynamic kernel conditioned on watermark feature.

Then, we add our semantic-aware propagation module (SPM) based on row 5, leading to the results in row 9. The promotion demonstrates the effectiveness of our proposed SPM. Based on row 9, we remove the spatial similarity map S^{sp} (*resp.*, S^{se}) and report the results in row 8 (*resp.*, row 7), which indicates that both maps contribute to the final performance. Since our SPM is structurally similar to self-attention [39], we also replace our SPM by typical self-attention and report the results in row 6. We can conclude that our designed similarity map works better than the similarity map automatically learned in self-attention.

To intuitively show how the information propagation in SPM helps reconstruct the background, we further provide qualitative comparison between row 5 and row 9 in Figure 5. Two rows correspond to two cases mentioned in Section 1: 1) the segmentation model ignores the watermark; 2) the segmentation model classifies the watermark as an isolated object. In each case, with a specified query pixel (yellow dot in “Input”), we show the semantic similarity map S^{se} , the spatial similarity map S^{sp} , and their combination S^{co} , from which it can be seen that S^{co} can indicate the semantically similar and spatially near region of the query pixel. In both cases, row 5 underperforms row 9, which proves the effectiveness of propagating

#	DKM	SPM	Evaluation Metrics			
			PSNR↑	RMSE _w ↓	IoU(%)↑	F ₁ ↑
1	-	-	38.23	16.90	73.56	0.8308
2	w/o residual output	-	38.90	16.43	78.69	0.8634
3	w/o \hat{M} mul	-	38.88	16.54	76.58	0.8521
4	SMR [26]	-	38.94	15.99	79.15	0.8678
5	✓	-	39.41	15.90	79.71	0.8707
6	✓	self-attention [39]	39.97	15.21	79.82	0.8725
7	✓	w/o S^{se}	39.90	15.59	79.81	0.8721
8	✓	w/o S^{sp}	41.66	14.36	80.14	0.8748
9	✓	✓	42.16	13.78	80.16	0.8770

Table 2: The ablation study results on LOGO30K [6]. “✓” indicates using the whole module and “–” indicating discarding this module.

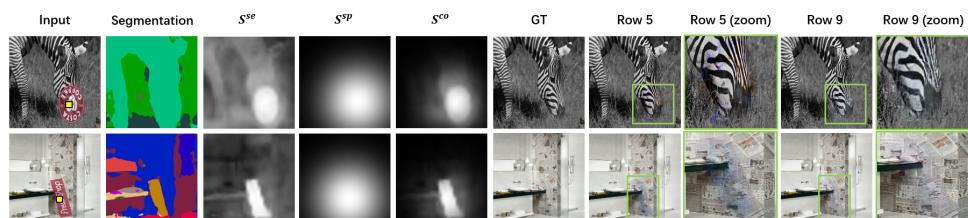


Figure 5: Visualization results of semantic-aware propagation. “Input” is the watermarked image. “Segmentation” is the segmentation result of HRNet+OCR [42] for input image. S^{se} , S^{sp} , S^{co} are the similarity maps for the query pixel (yellow dot in “Input”). “GT” indicates the ground-truth background image. Row 5 and Row 9 correspond to the rows in Table 2. We zoom in the region in green bounding box for better observation.

information from semantically similar and spatially near pixels.

5 Conclusion

In this work, we concentrate on watermark removal and develop a U-Net like model that incorporates a novel Dynamic Kernel Module and Semantic-aware Propagation Module, which can simultaneously predict watermark masks and reconstruct watermark-free images. The Dynamic Kernel Module can make our network better adapt to watermarks of different shapes, patterns and colors, while the Semantic-aware Propagation Module can make the restoration of the watermark region more accurate. We conduct comprehensive experiments on two large-scale datasets, which indicates that our approach performs more favorably than the existing approaches.

Acknowledgement

The work was supported by the Shanghai Municipal Science and Technology Major/Key Project, China (2021SHZDZX0102, 20511100300) and National Natural Science Foundation of China (Grant No. 61902247).

References

- [1] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016.
- [2] Z. Cao, S. Niu, J. Zhang, and X. Wang. Generative adversarial networks model for visible watermark removal. *IET Image Processing*, 13(10):1783–1789, 2019.
- [3] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic relu. In *ECCV*, 2020.
- [4] D. Cheng, L. Xiang, W. H. Li, L. Chan, F. Li, Z. Hua, and W. S. Zheng. Large-scale visible watermark detection and removal with deep convolutional networks. In *PRCV*, 2018.
- [5] Xiaofeng Cong, Jie Gui, Kai-Chao Miao, Jun Zhang, Bing Wang, and Peng Chen. Discrete haze level dehazing network. In *ACMMM*, 2020.
- [6] Xiaodong Cun and Chi-Man Pun. Split then refine: stacked attention-guided resunets for blind single image visible watermark removal. In *AAAI*, 2021.
- [7] Xiaodong Cun, Chi-Man Pun, and Cheng Shi. Towards ghost-free shadow removal via dual hierarchical aggregation network and shadow matting gan. In *AAAI*, 2020.
- [8] T. Dekel, M. Rubinstein, C. Liu, and W. T. Freeman. On the effectiveness of visible watermarks. In *CVPR*, 2017.
- [9] Bin Ding, Chengjiang Long, Ling Zhang, and Chunxia Xiao. Argan: Attentive recurrent generative adversarial network for shadow detection and removal. In *ICCV*, 2019.
- [10] Hang Dong, Jinshan Pan, Lei Xiang, Zhe Hu, Xinyi Zhang, Fei Wang, and Ming-Hsuan Yang. Multi-scale boosted dehazing network with dense feature fusion. In *CVPR*, 2020.
- [11] Zhiwen Fan, Huafeng Wu, Xueyang Fu, Yue Huang, and Xinghao Ding. Residual-guide network for single image deraining. In *ACMMM*, 2018.
- [12] Yosef Gandelsman, Assaf Shocher, and Michal Irani. "double-dip": Unsupervised image decomposition via coupled deep-image-priors. In *CVPR*, 2019.
- [13] Hang Gao, Xizhou Zhu, Steve Lin, and Jifeng Dai. Deformable kernels: Adapting effective receptive fields for object deformation. *arXiv preprint arXiv:1910.02940*, 2019.
- [14] Yunhui Guo, Yandong Li, Liqiang Wang, and Tajana Rosing. Depthwise convolution is all you need for learning multiple visual domains. In *AAAI*, 2019.
- [15] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.
- [16] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

- [17] Adam W Harley, Konstantinos G Derpanis, and Iasonas Kokkinos. Segmentation-aware convolutional networks using local attention masks. In *ICCV*, 2017.
- [18] Kaiming He, Jian Sun, and Xiaou Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2010.
- [19] Amir Hertz, Sharon Fogel, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. Blind visual motif removal from a single image. In *CVPR*, 2019.
- [20] C. H. Huang and J. L. Wu. Attacking visible watermarking schemes. *IEEE Transactions on Multimedia*, 6(1):16–30, 2004.
- [21] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *NeurIPS*, 2016.
- [22] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.
- [23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [24] HyunJae Lee, Hyo-Eun Kim, and Hyeonseob Nam. Srm: A style-based recalibration module for convolutional neural networks. In *ICCV*, 2019.
- [25] Xiang Li, Chan Lu, Danni Cheng, Wei-Hong Li, Mei Cao, Bo Liu, Jiechao Ma, and Wei-Shi Zheng. Towards photo-realistic visible watermark removal with conditional generative adversarial networks. In *ICIG*, 2019.
- [26] Jing Liang, Li Niu, Fengjun Guo, Teng Long, and Liqing Zhang. Visible watermark removal via self-calibrated localization and background refinement. In *ACMMM*, 2021.
- [27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.
- [28] Yang Liu, Zhen Zhu, and Xiang Bai. Wdnet: Watermark-decomposition network for visible watermark removal. In *WACV*, 2021.
- [29] Ningning Ma, Xiangyu Zhang, Jiawei Huang, and Jian Sun. Weightnet: Revisiting the design space of weight networks. In *ECCV*, 2020.
- [30] Jaesik Park, Yu-Wing Tai, and In So Kweon. Identigram/watermark removal using cross-channel correlation. In *CVPR*, 2012.
- [31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019.
- [32] Soo-Chang Pei and Yi-Chong Zeng. A novel image recovery algorithm for visible watermarked images. *IEEE Transactions on information forensics and security*, 1(4):543–550, 2006.

- [33] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *CVPR*, 2018.
- [34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.
- [35] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *CVPR*, 2017.
- [36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [37] Cong Wang, Yutong Wu, Zhixun Su, and Junyang Chen. Joint self-attention and scale-aggregation for self-calibrated deraining network. In *ACMMM*, 2020.
- [38] Jifeng Wang, Xiang Li, and Jian Yang. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *CVPR*, 2018.
- [39] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.
- [40] Dong Yang and Jian Sun. Proximal dehaze-net: A prior learning-based deep network for single image dehazing. In *ECCV*, 2018.
- [41] Youzhao Yang and Hong Lu. Single image deraining via recurrent hierarchy enhancement network. In *ACMMM*, 2019.
- [42] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. 2020.
- [43] He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *CVPR*, 2018.
- [44] Jing Zhang, Yang Cao, Zheng-Jun Zha, and Dacheng Tao. Nighttime dehazing with a synthetic benchmark. In *ACMMM*, 2020.