PatchWiper: Leveraging Dynamic Patch-Wise Parameters for Real-World Visible Watermark Removal

Zihao Mo
School of Computer Science and Engineering,
Sun Yat-Sen University
Guangzhou, China
mozh6@mail2.sysu.edu.cn

Junye Chen
School of Computer Science and Engineering,
Sun Yat-Sen University
Guangzhou, China
chenjy856@mail2.sysu.edu.cn

Chaowei Fang*
School of Artificial Intelligence,
Xidian University
Xi'an, China
chaoweifang@outlook.com

Guanbin Li*
School of Computer Science and Engineering,
Sun Yat-Sen University
Guangzhou, China
liguanbin@mail.sysu.edu.cn

Figure 1: Visual comparison on a real-world watermarked image. Our PatchWiper completely removes artifacts (thin white lines) with fine-grained restoration. While SLBR [17] trained on CLWD [20] fails, the version trained on our PRWD successfully erases the primary watermark components, demonstrating PRWD's superior generalization to real-world scenarios.

Abstract

Visible watermark removal is crucial for evaluating watermark robustness and advancing more resilient protection techniques. Current methods face challenges in real-world scenarios due to architectural constraints in multi-task frameworks and limited dataset diversity. To address these challenges, we first propose a novel two-stage framework, **PatchWiper**, consisting of an independent watermark segmentation network and a highly dynamic patch-wise restoration network. This framework decouples watermark localization from background restoration, allowing each network to focus on its designated task. Our restoration network dynamically generates unique parameters for each image patch, enabling fine-grained adaptation to different watermark distortions. Second, we construct the **P**ixabay **R**eal-world **W**atermark **D**ataset (**PRWD**), which incorporates diverse background images and over 1,000 distinct watermark types, providing a more comprehensive benchmark for evaluating watermark removal methods. Extensive experiments on PRWD, ILAW, and real-world testing images demonstrate our method's superior performance over existing approaches, particularly in handling complex real-world cases.

*Corresponding Author.

MM '25, Dublin, Ireland.

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755107

CCS Concepts

• **Computing methodologies → Image processing**.

Keywords

watermark removal, dynamic neural network

ACM Reference Format:
Zihao Mo, Junye Chen, Chaowei Fang, and Guanbin Li. 2025. PatchWiper: Leveraging Dynamic Patch-Wise Parameters for Real-World Visible Watermark Removal. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3746027.3755107

1 Introduction

Visible watermarking, a widely used method for asserting ownership of multimedia assets, plays a critical role in copyright protection. As a key method for evaluating watermark reliability, visible watermark removal has attracted considerable research attention, driving the development of more robust watermarking strategies.

Recently, benefiting from multiple open benchmarks [4, 5, 20], blind visible watermark removal has made a significant stride [2, 5, 10, 17, 20, 22, 26, 34]. The key to effective watermark removal from a watermarked image lies in two aspects: precise watermark localization and high-quality background restoration. Current multi-task learning frameworks [5, 17, 20, 22, 34] use shared encoders that simultaneously locate the watermark and restore the background content. However, the joint optimization of different targets will cause mutual interference, resulting in imprecise watermark localization. Meanwhile, these coarse-to-fine methods rely on static refinements [5, 17, 26] or simplistic dynamic mechanisms [22, 34], which either apply uniform parameters across regions or offer limited adaptability to a fixed number of watermark segments. These approaches often result in inferior performance, producing blur and artifacts in the restored images. These shortcomings prevent existing methods from completely removing watermarks.

In this paper, we propose **PatchWiper**, a novel two-stage framework consisting of an independent watermark localization network and an adaptively dynamic patch-wise restoration network. To accurately locate the watermark, we utilize a dedicated network focused solely on watermark localization, without interference from background restoration. For the restoration process, we address the varying levels of distortion across different patches by developing a dynamic, patch-wise restoration network, which includes a Representation Generation Network (RGN) and a Patch Query Network (PQN). Watermarked patches in the central region often experience severe distortion and require global context features for effective restoration. To handle these, the RGN dynamically generates adaptive representations for each patch. The PQN then processes each patch using these adaptive representations, incorporating sinusoidal positional encoding and watermark masks to enable coordinate-aware recovery. For patches near the watermark boundaries with only partial distortion, adjacent undistorted pixels provide valuable prior information for restoring the affected areas. By combining the RGN and PQN, our network dynamically assigns representations to the restoration process for each patch, enabling fine-grained restoration of patch-wise features.

In addition to our technical contributions, we address the limitations of existing datasets by developing the **P**ixabay **R**eal-world **W**atermark **D**ataset (**PRWD**). Existing methods often struggle with generalization due to training on datasets like CLWD [20] with limited scale and simple watermark patterns. While ILAW [13] introduced a more complex dataset, its scale and diversity remain limited. PRWD offers a diverse and extensive benchmark with 223,278 images—nearly four times larger than existing datasets [13, 29]. It includes a variety of backgrounds from natural, human-designed, and AI-generated sources, along with 1,129 distinct watermarks across three major types. This dataset better reflects real-world scenarios, enhancing model generalization and advancing watermark removal research. Extensive experiments are conducted on PRWD, ILAW [13], CLWD [29], and real-world testing images, and the quantitative and qualitative results demonstrate that our method outperforms existing methods significantly.

The main contributions of this paper are summarized as follows:

- We propose **PatchWiper**, a novel two-stage method consisting of an independent watermark localization network and a dynamic patch-wise restoration network that can generate adaptive representations for each image patch to achieve finer-grained recovery.
- We introduce a comprehensive benchmark for visible watermark removal, named **P**ixabay **R**eal-world **W**atermark **D**ataset (**PRWD**), which provides a more diverse and larger-scale dataset that facilitates model generalization and comprehensive evaluation of watermark removal methods.
- Extensive experiments show that our method achieves state-of-the-art performance and superior generalization on real-world watermark removal.

2 Related Works

2.1 Dynamic Neural Network

Dynamic neural networks have emerged as a powerful paradigm in image processing tasks, achieving adaptive computation by dynamically adjusting architectures or parameters based on input data [8, 9]. In super-resolution, dynamic neural networks have been utilized to adaptively generate upscaling parameters for different pixel positions, achieving arbitrary scale upscaling [3, 7, 11, 25, 30]. In image restoration such as denoising and deblurring, dynamic neural networks have been applied to adjust newtork parameters or architectures to better handle varying noise levels or blur intensities across different parts of an image [15, 21]. The adaptability of dynamic networks ensures that they can generalize well across a wide range of degradation types. In image inpainting, dynamic neural networks have shown promise in filling in missing or corrupted regions of an image [19, 29]. By dynamically generating parameters that consider the surrounding context, these networks can produce more coherent and visually pleasing inpainted results. This is particularly useful in scenarios where the missing regions have complex textures or patterns.

2.2 Visible Watermark Removal

The advent of deep learning has revolutionized watermark removal by enabling data-driven solutions. Early GAN-based methods [1, 16] treated it as image-to-image translation but often produce unsatisfactory removal results. Subsequent works [5, 10, 14, 17, 20] highlight the importance of watermark localization, adopting multi-task learning frameworks to localize watermarks and reconstruct backgrounds simultaneously. For example, SplitNet [5] introduced a two-stage architecture to decouple task-specific features and refine masked regions. However, it struggles when detection fails or when the watermark and background textures are similar. To improve accuracy, SLBR [17] proposed Self-calibrated Mask Refinement (SMR) for better localization and multi-stage refinement for enhanced restoration.

Recent approaches like DKSP [34] adapts convolutional kernel parameters for different watermarked images. Li *et al.* [22] enhanced this approach through a part-aware feature modulation module,

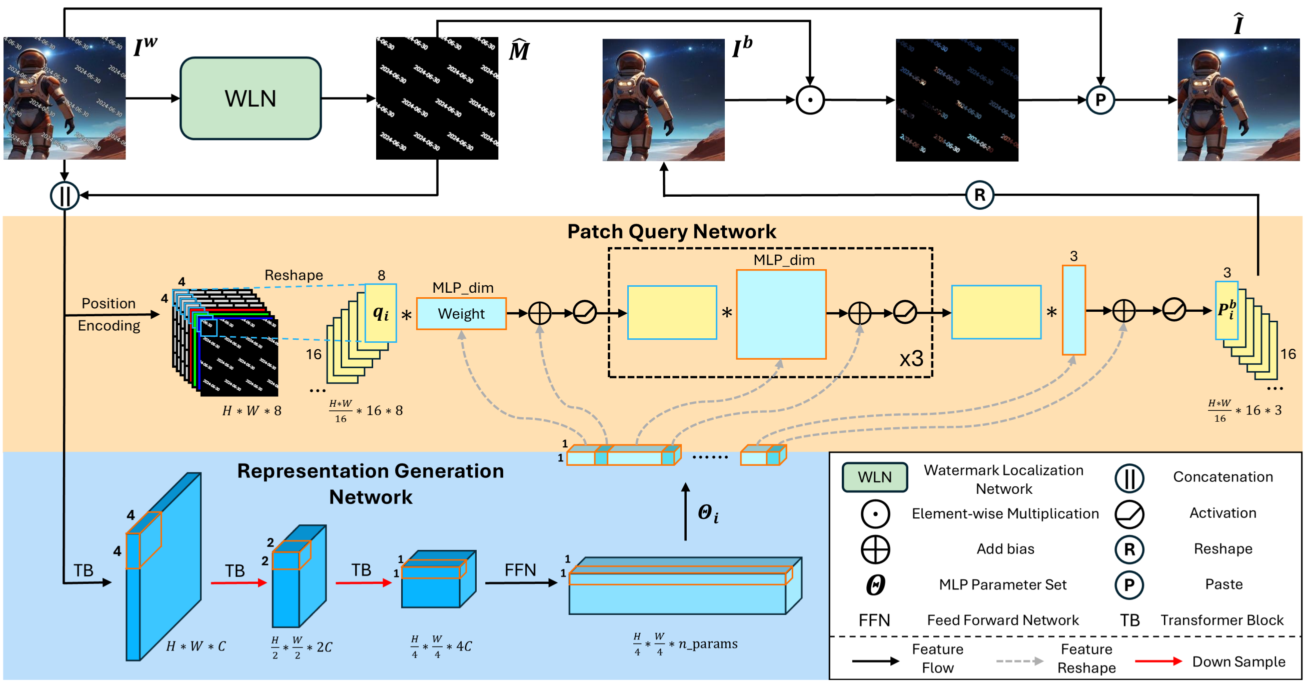

Figure 2: Architecture of PatchWiper. The watermarked image and predicted mask are taken as the input of both Representation Generation Network (RGN) and Patch Query Network (PQN). The PQN first partitions the encoded features into patch-based queries, each of which is processed using the corresponding unique MLP parameters generated by the RGN. The restored patches are reshaped to a complete image and pasted to the original input.

enabling fine-grained region-specific image processing. Although these dynamic methods demonstrate improved adaptability to diverse watermark patterns, their limited adaptive parameters hinder flexibility in local patch processing, particularly for handling heterogeneous pixel-level characteristics. Moreover, the shared encoder architecture employed across these methods [5, 10, 14, 17, 20, 22, 34] may introduce feature extraction conflicts, ultimately degrading localization performance.

Our work addresses this via a decoupled design that separates watermark localization and background restoration into distinct networks, and a dynamic patch-wise restoration network to handle varying watermark distortions.

3 Method

We introduce PatchWiper, a two-stage method that ingests a watermarked image as input and produces a watermark-free image as output. We first provide an overview of our method in Sec. 3.1, then introduce the Watermark Localization in Sec. 3.2, and finally the Dynamic Background Restoration in Sec. 3.3.

3.1 Overview

Given a watermarked image $I^w \in \mathbb{R}^{H\times W\times 3}$, PatchWiper predicts the target watermark-free image $\hat{I} \in \mathbb{R}^{H\times W\times 3}$ along with a watermark mask $\hat{M} \in \mathbb{R}^{H\times W\times 1}$, where H and W are the image height and width.

As illustrated in Fig. 2, our PatchWiper consists of two stages: precise watermark localization and background content restoration. For watermark localization, we utilize a dedicated network with a well-designed training strategy to precisely locate the watermarks and predict a binary watermark mask $\hat{M}$. For background restoration, we first capture the global representations of each patch with RGN, which are used to generate unique representations for each patch dynamically. Then, the PQN performs adaptive patch-based query processing to assign the generated representations to each patch and recover the background content using the original pixels, coordinates, and watermark mask as priors.

3.2 Watermark Localization

We observed that joint prediction and restoration in [5, 17, 26] leads to suboptimal results in mask prediction, thereby compromising the background content restoration. To this end, we decouple the watermark localization and background restoration, and introduce a dedicated segmentation network with a specific training strategy for the prediction of watermark masks. Specifically, this network has an UNet-like encoder-decoder structure [24], comprising a 5-layer encoder and a 4-layer decoder. The final 3 layers of the decoder incorporate the Self-Calibrated Mask Refinement (SMR) modules from SLBR [17]. This design leverages attention mechanisms to effectively expand the receptive field, which is particularly crucial for capturing watermark patterns that often exhibit varying scales and irregular shapes.

Inspired by [36], to calibrate mask predictions sequentially at multi-scales, our training strategy jointly supervises the outputs of each decoder layer and the final mask prediction. Specifically, each SMR layer yields a primary mask M^p and a self-calibrated mask M^{sc}, resulting in 6 intermediate outputs (2×3 layers) that collectively participate in the supervision along with the final mask prediction. For the primary masks, we employ a binary cross-entropy loss to supervise the predictions:

$$\mathcal{L}_{\mathrm{p}} = -\sum_{i=1}^{3} \gamma^{i-1} \sum_{j=1}^{N} \left[M_j \log(M^p_{i,j}) + (1 - M_j) \log(1 - M^p_{i,j}) \right], \quad (1)$$

where N denotes the number of pixels, M_j denotes the ground-truth value (0 or 1) for pixel j, $\hat{M}_j$ is the predicted probability for pixel j, and γ^i serves as the decay factor of i-layer to control the supervision intensity. To enhance the quality of intermediate representations, we introduce a self-calibrated mask loss that combines binary cross-entropy and intersection-over-union (IoU) metric:

$$\mathcal{L}_{\mathrm{sc}} = -\sum_{i=1}^{3} \gamma^{i-1} \sum_{j=1}^{N} \left[M_j \log(M^{sc}_{i,j}) + (1 - M_j) \log(1 - M^{sc}_{i,j}) \right] + \lambda_{\mathrm{iou}} \sum_{i=1}^{3} \gamma^{i-1} \left[1 - \frac{\sum_{j=1}^{N} M_j M^{sc}_{i,j}}{\sum_{j=1}^{N} (M_j + M^{sc}_{i,j} - M_j M^{sc}_{i,j})} \right], \quad (2)$$

where γ^i serves as a decay factor, and λ_{iou} controls the relative importance of the IoU, particularly beneficial for improving boundary precision. Additionally, the final mask prediction $\hat{M}$ is supervised by a standard binary cross-entropy loss $\mathcal{L}_{\mathrm{f}}$. Finally, the complete loss $\mathcal{L}_{\mathrm{mask}}$ is composed by the three losses mentioned above,

$$\mathcal{L}_{\mathrm{mask}} = \mathcal{L}_{\mathrm{f}} + \mathcal{L}_{\mathrm{sc}} + \lambda_p \mathcal{L}_{\mathrm{p}}, \quad (3)$$

where λ_p is a balanced weight.

This design of the watermark localization network yields two significant advantages. First, by decoupling localization from restoration, the encoder learns feature representations that are specifically discriminative for watermark patterns without interference from other objectives. Second, the multi-level self-calibration mechanism effectively leverages hierarchical features across different scales, significantly improving performance at challenging boundary regions where traditional segmentation methods often fail. Progressive refinement through multiple SMR layers enables the network to iteratively correct and enhance its predictions throughout the decoding process.

3.3 Dynamic Background Restoration

Watermarks often exhibit complex patterns with significant color variations while background textures show a similar complexity and variety. The dual complexity of watermarks and background textures makes restoring the background content highly challenging. Current dynamic restoration approaches [22, 34], which rely on limited adaptive convolution kernels, still struggle to address such complex variability. Inspired by CoordFill [19], we propose a patch-wise restoration paradigm consisting of two branches: the Representation Generation Network (RGN) and the Patch Query Network (PQN). Specifically, the RGN generates patch-specific representations that capture both global features and local spatial distributions. The PQN then uses these adaptive representations to perform customized restoration for each patch. This approach effectively handles the complexity of watermark patterns and background textures, enabling precise, location-aware removal and high-quality restoration of the background content.

Representation Generation Network (RGN). Instead of having a predefined set of filters like conventional CNNs or limited dynamic kernels such as prior works [22, 34], our RGN generates unique representations for each input patch. These representations encode both global features and local spatial distributions, enabling precise patch-specific restoration during the subsequent patch querying process. Concretely, we feed the concatenation of watermarked image I^w and the predicted mask $\hat{M}$ from the localization network described in Sec. 3.2 into the hierarchical Restormer blocks [32] to model global dependencies and extract contextual features. The derived features are then reshaped and processed through a Feed-Forward Network (FFN) to yield the comprehensive representation sets $\Theta = \{\theta_i | i = 1, 2, ..., \frac{HW}{16}\}$ for the subsequent PQN, which is formally defined as follows:

$$\Theta = \mathrm{FFN}(\mathcal{R}(f([I^w, \hat{M}]))), \quad (4)$$

where $f(\cdot)$ represents the stack of hierarchical Restormer blocks [32], and $\mathcal{R}(\cdot)$ denotes the reshape operation.

Patch Query Network (PQN). Unlike previous approaches [10, 17, 20], which apply uniform processing to all patches, our PQN performs adaptive, patch-wise query processing. It dynamically assigns the representations generated by PQN as path-wise network parameters. These parameters are used for recovering each image patch.

Specifically, to endow the network with explicit location awareness, we incorporate sinusoidal positional encoding [28] into each input watermarked image I^w according to the following formulation:

$$E_p = \left(\sin\left(\frac{2\pi(p_x \bmod H_p)}{H_p}\right), \cos\left(\frac{2\pi(p_x \bmod H_p)}{H_p}\right), \sin\left(\frac{2\pi(p_y \bmod W_p)}{W_p}\right), \cos\left(\frac{2\pi(p_y \bmod W_p)}{W_p}\right) \right), \quad (5)$$

where E_p is the embedding of position encoding, p_x and p_y denote the 2D pixel coordinate, and H_p and W_p represent the height and width of the patches, respectively. This embedding is then concatenated with the input I^w and the corresponding watermark mask $\hat{M}$ along the channel to form the patch embedding $E = concat(E_p, I^w, \hat{M})$. Following this, we divide E into a groups of non-overlapping 4×4 patches to serve as query inputs $\mathbf{Q} = \{q_i | i = 1, 2, ..., \frac{HW}{16}\}$. Consequently, each query $q_i \in \mathbf{Q}$ is associated with a distinct representation θ_i which is retrieved from Θ according to the location index of the i-th patch. The i-th restored patch P^b_i is then computed as follows:

$$P^b_i = \mathrm{MLP}(q_i, \theta_i). \quad (6)$$

Finally, we gather all querying results and reassemble them to restore the background image I^b. The target watermark-free image $\hat{I}$ can be obtained as follows:

$$\hat{I} = I^b \odot \hat{M} + I^w \odot (1 - \hat{M}), \quad (7)$$

where $\odot$ denotes element-wise multiplication. And we supervise the final output with $\mathcal{L}_1$ loss.

Combining RGN and PQN forms the final watermark removal model, effectively addressing the challenges of restoring watermark regions by adapting to local patch variations. Specifically,

1) Central Watermark Patches: For central, boundary-remote watermark regions, RGN's global features assist in reconstructing plausible backgrounds, while PQN ensures consistency across adjacent patches. 2) Edge-Adjacent Watermark Regions: For watermark regions near boundaries, intact original pixels serve as strong priors, aiding restoration and reducing blur or artifacts at the transitions. These examples highlight our approach's adaptability in handling diverse local patch characteristics within watermarked regions.

4 The PRWD Benchmark

Existing popular watermark datasets, such as LVW [4], LOGO-Series [5], and CLWD [20], exhibit discrepancies in data distribution compared to real-world watermark removal scenarios, manifesting in background diversity, watermark variety, and data scale. Models trained on these datasets show inferior generalization in real-world tasks, while the evaluation benchmarks struggle to objectively reflect actual performance.

Specifically, the background images in these datasets are predominantly sourced from real-world photos, such as PASCAL VOC2012 [6] for CLWD and LVW, and MS COCO [18] for LOGO-Series. However, real-world watermark removal scenarios involve diverse backgrounds, including human-designed content and AI-generated synthetic images. Moreover, existing datasets mainly feature single-pattern watermarks, whereas practical applications may encounter full-screen watermarks with repetitive elements or long-strip text watermarks like URLs and timestamps. These issues hinder further advancement in the field of watermark removal. To address these limitations and enhance the generalization capability of watermark removal models, we propose the **P**ixabay **R**eal-world **W**atermark **D**ataset (**PRWD**), an unprecedentedly diverse and large-scale dataset that establishes a more reliable benchmark for evaluating real-world performance, paving the way for further advancements in real-world visible watermark removal.

4.1 Data Collection

To align data distribution with real-world tasks, we developed a dedicated pipeline for collecting diverse images and watermarks from the Internet.

Background Collection. The diversity of background images is critical to the generalization of models to real-world watermark removal task. Therefore, we collected relevant images by utilizing 720 keywords covering various content categories, with each keyword crawling 600 images via Pixabay's API[1]. Duplicate and grayscale images were automatically removed, and a manual filtering process was conducted to exclude low-quality or inappropriate content. Subsequently, all images were standardized to a resolution of 256×256 pixels, aligning with established benchmarks and aiding in model training. Totally, we collected 223,278 different background images.

Watermark Collection. The watermarks are collected from public image websites and previous datasets or manually synthesized. These watermarks can be categorized into three types: single logo watermarks, long-strip text watermarks, and full-screen watermarks. 1129 different watermarks are collected in total.

- **Single logo watermarks** are individual brand symbols or icons used to assert ownership or brand identity.

[1]https://pixabay.com/

Figure 3: Visualization of watermarked images from PRWD dataset. Rows show (top to bottom) real-world photographs, human-designed graphics, and AI-generated content backgrounds. Columns show (left to right) single logo, long-strip text, and full-screen watermarks.

- **Long-strip text watermarks** are horizontal or vertical overlays, such as URLs or timestamps, which mark copyright but do not completely obscure the content.
- **Full-screen watermarks** consist of tiled patterns that cover the entire image, serving as a deterrent to unauthorized use while maintaining major content visibility.

Real-world Watermarked Collection. To evaluate the generalization of the watermark removal models in real-world scenarios, we also collect a real-world watermarked dataset specifically for evaluation, containing 49 various watermarked images of size 512×512 sourced from the Internet. These watermarked images have been meticulously selected by human experts to ensure diversity and representativeness, and the corresponding background images are not available.

4.2 Watermarked Data synthesis

The watermarked image $I^w \in \mathbb{R}^{H\times W\times 3}$ can be generated by blending the watermark $w \in \mathbb{R}^{H\times W\times 3}$ with the background image $I^b \in \mathbb{R}^{H\times W\times 3}$ followed by a post-processing $\mathcal{P}(\cdot)$:

$$I^w = \mathcal{P}(I^b \odot (1-\alpha M) + w \odot \alpha M), \tag{8}$$

where $M \in \mathbb{R}^{H\times W\times 1}$ represents the binary mask of watermark, $\alpha \in (0,1]$ is the opacity level of the watermark, and $\odot$ denotes element-wise multiplication. The post-processing $\mathcal{P}(\cdot)$ applies standard JPEG compression to the blended image. We synthesize the watermarked data adhering to a distribution ratio of 70% single logo watermarks, 20% long-strip text watermarks, and 10% full-screen watermarks. Specifically, for the single logo, we scale the watermark by the original ratio, with a factor within [0.3, 0.6]. The position

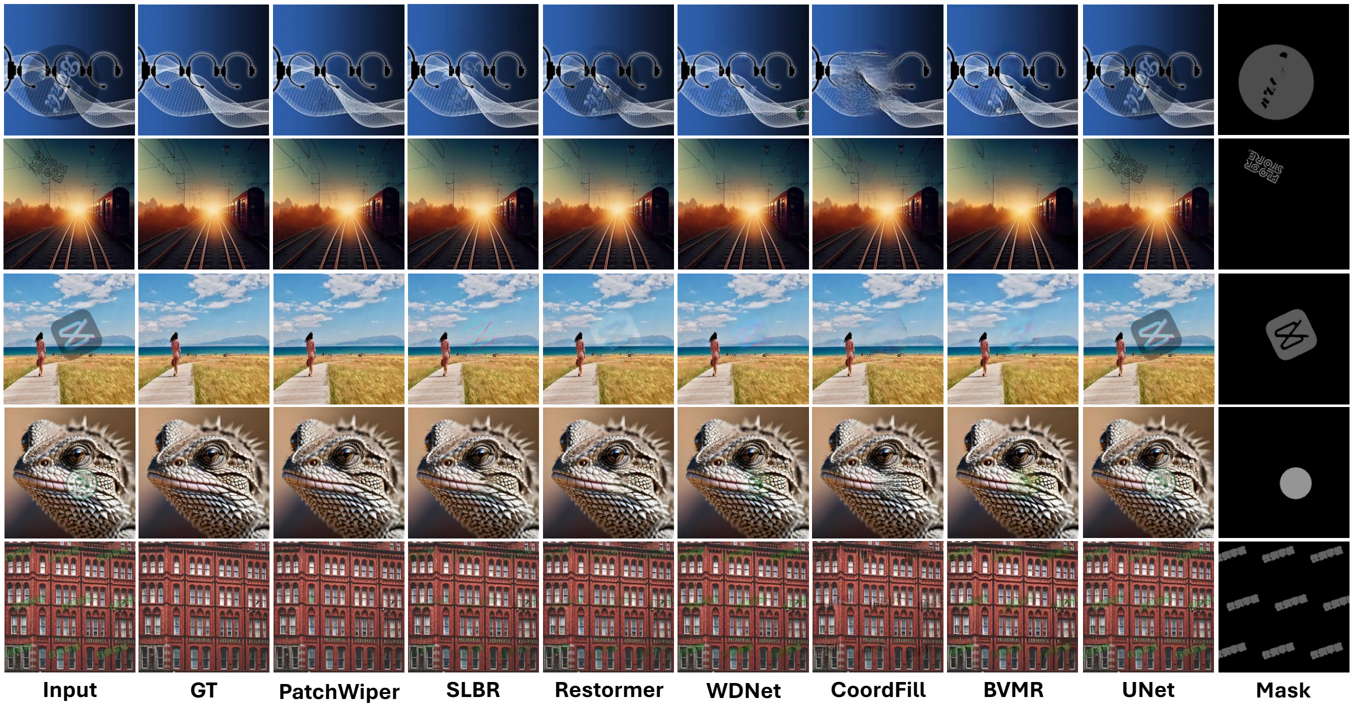

Figure 4: Visualization results of different methods on PRWD dataset. From left to right: the input watermarked image, the ground-truth watermark-free image, the output images of different methods, and the watermark mask.

and rotation are randomized, with the opacity randomly set within the range of (0.3, 0.7), akin to the CLWD [20]. For the long-strip text, to simulate real-world tasks, we scale the watermark by a factor within [0.5, 1] according to the original scale, with opacity randomly set in the range of (0.5, 1.0), and then randomly place it on the background image to enhance the diversity of the dataset. Since full-screen watermarks are not accounted for in other benchmarks, we manually craft 85 distinct watermarks using Photoshop and various online resources. In the synthesis process, we adjust the watermark size to match that of the background image and set the opacity level within the range of [0.3,1.0].

The whole PRWD benchmark encompasses a total of 223,278 distinct synthesized watermarked images, with a real-world sub-dataset of 49 watermarked images sourced from the Internet. For synthesized watermarked images, our PRWD benchmark provides corresponding ground truth background images and opacity masks. By combining more comprehensive watermarks with a wide range of background images, our PRWD addresses key limitations in existing datasets. This diversity not only better captures real-world complexity but also facilitates a more comprehensive evaluation of generalization capabilities, overcoming the limitations of small-scale homogeneous datasets in the field. A few example images from the PRWD dataset with watermarks are shown in Figure 3.

5 Experiments

In this section, we first introduce experiment datasets, implementation details, and evaluation metrics. Then, we compare our method with existing watermark removal methods and image content removal methods. We also provide visualization results of all methods to demonstrate the effectiveness and superiority of our method. Furthermore, we conduct comprehensive ablation studies to investigate the benefit of each design in our network.

5.1 Datasets and Implementation Details

To demonstrate the exceptional generalization capability of our method, we conduct experiments on three benchmark datasets: Colored Large-scale Watermark Dataset (CLWD) [20], Images with Large Area Watermarks (ILAW) dataset [13], and our proposed Pixabay Real-world Watermark Dataset (PRWD). These datasets represent the most complex challenges in visible watermark removal. Furthermore, to highlight the superiority in localization of our two-stage framework, we also report the results of mask prediction on the Colored Large-scale Watermark Dataset (CLWD).

CLWD [20]. CLWD comprises 60,000 training images featuring 160 unique watermarks and 10,000 test images with 40 distinct watermarks. Watermarks in CLWD are sourced from open-source logo image websites, while background images for both sets are randomly selected from the PASCAL VOC2012 [6] dataset. In the watermarking process, opacity is varied within $(0.3, 0.7)$, while the size, location, and rotation angle of each watermark are randomly adjusted across different images.

ILAW [13] In ILAW, the training set has 60,000 images of size 256×256 with 1,087 different watermarks, and the test set has 10,000 images of size 512×512 with 160 different watermarks. The background images are sourced from the Places365 [35] Challenge dataset [35], and the watermarks are collected from the Internet. The watermarks have a larger area and higher opacity than those datasets [4, 5, 20] used in previous watermark removal works.

Method	PSNR↑	SSIM↑	RMSE↓	RMSEw↓
CoordFill [19]*	35.90	0.9732	5.51	23.04
U-Net [24]	28.16	0.9354	12.16	51.67
Restormer [32]	37.69	0.9817	4.09	16.08
Li *et al.* [16]	26.36	0.8847	13.31	40.82
BVMR [10]	29.88	0.9294	8.79	16.50
WDNet [20]	36.68	0.9803	4.60	14.95
SLBR [17]	38.74	0.9805	3.46	12.80
PatchWiper(Ours)	**39.38**	**0.9839**	**3.27**	**12.20**

Table 1: The results of different methods on PRWD dataset. The method marked with * is provided with precise masks. The best results are denoted in boldface.

Method	Params(M)	PSNR↑	SSIM↑	LPIPS↓
LaMa [27]*	88.09	17.97	0.677	0.326
CoordFill [20]*	34.50	22.66	0.819	0.149
DENet [26]	22.42	19.66	0.814	0.236
WDNet [20]	21.00	24.37	0.887	0.166
SplitNet [5]	32.61	25.72	0.892	0.156
SLBR [17]	21.39	25.02	0.890	0.154
Leng *et al.* [13]	97.95	25.77	0.916	0.100
PatchWiper(Ours)	33.25	**32.00**	**0.947**	**0.068**

Table 2: The results of different methods on ILAW dataset. The method marked with * is provided with coarse masks. The best results are denoted in boldface.

PRWD. In PRWD, the training set has 189,768 images with 994 different watermarks, and the test set has 33,492 images with 135 different watermarks. All images are 256 × 256 in size. We utilize the real-world subdataset to evaluate the generalization of different methods on unseen data. Since the background images corresponding to these watermarked images are unavailable, we conduct the performance comparison through a user study.

Training details. We implement our method using Pytorch [23] and traind our model on CLWD [20], ILAW [13], and PRWD, respectively. For watermark localization, we choose Adam [12] optimizer with initial learning rate 1e-3, batch size 64, and momentum parameters: β_1 = 0.9, β_2=0.999. For background restoration, we choose Adam [12] optimizer with initial learning rate 3e-4, batch size 6, and momentum parameters: β_1 = 0.9, β_2=0.999. The hyper-parameters γ, λ_{iou}, and λ_p are empirically set as 0.5, 0.25, and 0.01, respectively.

5.2 Evaluation Metrics

Following [17], we employ a suite of full-reference evaluation metrics, including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) [31], Root-Mean-Square Error (RMSE) distance, and weighted Root-Mean-Square Error distance (RMSE$_w$), as well as Learned Perceptual Image Patch Similarity (LPIPS) [33]. The difference between RMSE and RMSE$_w$ lies in that RMSE$_w$ is only computed within the watermarked area.

5.3 Comparison with Baselines

We compare our method against three groups of baselines: (1) specialized watermark removal methods, (2) general image restoration approaches, and (3) image inpainting methods. For the first group, we compare with existing open-source visible watermark removal techniques, including cGAN-based methods [16], blind visual motif removal (BVMR) [10], SplitNet [5], WDNet [20], DENet [26], and SLBR [17]. Notably, SplitNet [5] and DENet [26] failed to converge on our PRWD benchmark due to their reliance on the watermark input, which is not preserved in our dataset. Therefore, we evaluate these two methods separately on the ILAW benchmark. The second group includes general image restoration methods such as U-Net [24] and Restormer [32]. Finally, the third group includes CoordFill [19], an efficient image inpainting method that can handle high-resolution images.

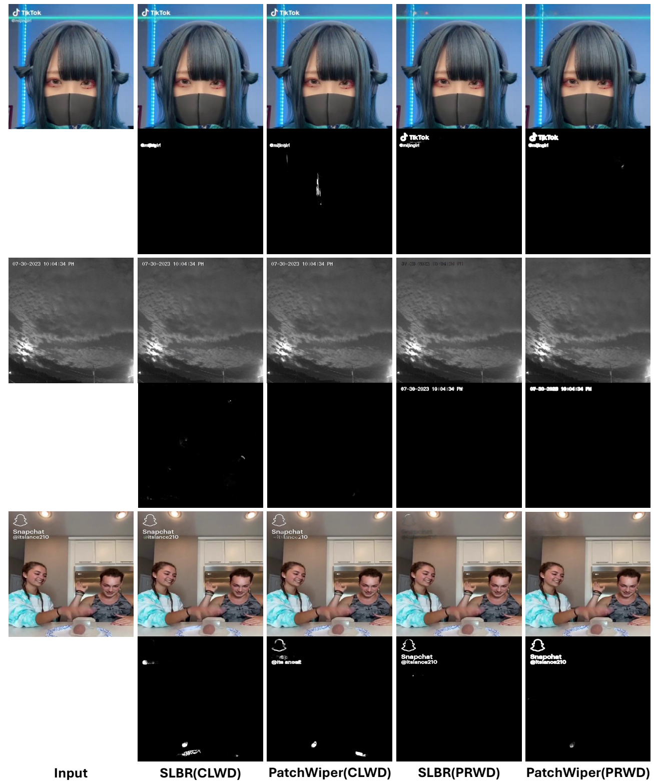

Figure 5: Visualization results on real-world watermark removal. Odd rows show restoration results while even rows display the corresponding predicted watermark masks.

Quantitative and Qualitative Evaluation. We evaluate the watermark removal performance of our method on two full-reference benchmarks, as shown in Tables 1 and 2. Our approach achieves state-of-the-art results on both, with significant improvements on ILAW, which includes heavily watermarked backgrounds with high opacity. This performance gap underscores the effectiveness of our dynamic recovery network in fine-grained watermark removal. As shown in Figure 4, we compare our method with other baselines [10, 17, 19, 20, 24, 32]on the PRWD benchmark, which demonstrates that our method effectively removes watermarks while preserving the highest fidelity of background details. For example, in the fourth row, baseline methods fail to entirely remove the watermark texture, leaving noticeable green pixels on the lizard's face. In the second row, while baseline methods remove the dark watermark, they also wipe out the overhead wires.

Encoding	Patch Size	Backbone	Decoder	Params(M)	PSNR↑	SSIM↑	RMSE↓	RMSEw↓
Position	8	AttFFC [19]	PQN	51.50	35.28	0.9725	5.87	24.83
Position	8	Transformer Block	PQN	40.30	38.70	0.9819	3.52	13.33
\	\	Transformer Block	CNN	55.46	28.23	0.9393	12.11	51.91
\	\	Transformer Block	Transformer Block	33.41	38.28	0.9837	3.87	15.22
Position	4	Transformer Block	PQN	33.19	39.07	0.9830	3.37	12.65
Position + Mask	4	Transformer Block	PQN	33.20	39.17	0.9831	3.33	12.48
Position + RGB	4	Transformer Block	PQN	33.24	39.21	0.9835	3.30	12.39
Position + RGB + Mask	4	Transformer Block	PQN	33.25	**39.38**	**0.9839**	**3.27**	**12.20**

Table 3: Ablation study results of our proposed PatchWiper with different combinations of components on PRWD dataset. The best results are in bold.

Method	PRWD		CLWD	
	IoU↑	F1↑	IoU↑	F1↑
WDNet [20]	0.7688	0.8688	0.6120	0.7240
BVMR [10]	0.7571	0.8612	0.7021	0.7871
SplitNet [5]	\	\	0.7196	0.8027
SLBR [17]	0.8083	0.8798	0.7463	0.8234
DKSP [34]	\	\	0.7730	0.8480
Li *et al.* [22]	\	\	0.7909	0.8634
PatchWiper(Ours)	**0.8177**	**0.8996**	**0.8042**	**0.8914**

Table 4: Quantitative evaluation of watermark masks predicted by our method and baselines on PRWD and CLWD [20] datasets.

Method & Dataset	Vote	Top-Ranked Count
SLBR [17] trained on CLWD [29]	180	0
PatchWiper trained on CLWD [29]	316	3
SLBR [17] trained on PRWD	1521	10
PatchWiper trained on PRWD	**2183**	**36**

Table 5: User study on the real-world dataset.

Watermark Localization Evaluation. We also evaluate the quality of watermark mask prediction across methods in Table 4. Our localization network outperforms all baselines in mask prediction precision, which can be attributed to its decoupled architecture, specifically designed to optimize localization precision for watermark localization.

Real-world Watermark Removal. To further assess the generalization of our PRWD benchmark and PatchWiper, we trained PatchWiper and SLBR [17] on PRWD and CLWD [20], respectively. We then evaluated their performance on the real-world watermarked dataset in PRWD through a user study involving 61 participants, who voted on the most effective model for each of 49 images. Table 5 summarizes the total votes and the number of images where each model was preferred. Figure 5 presents a visual comparison across three representative cases. The results reveal that when trained on PRWD, both PatchWiper and SLBR outperform their CLWD-trained counterparts in watermark localization. Furthermore, PatchWiper delivers more realistic restorations with sharper details and fewer artifacts than SLBR [17], further proving its robustness in real-world scenarios.

5.4 Ablation Studies

We conduct systematic ablation studies to validate our key design choices for the dynamic restoration network on PRWD dataset. The ablation study results are summarized in Table 3.

Effect of transformer backbone. We firstly begin with the design from CoordFill [19], using AttFFC [19] as the backbone to generate parameters (Row 1). When we replace AttFFC with transformer blocks from Restormer [32] (Row 2), the significant improvement demonstrates that transformer blocks serve as superior feature extractors and parameter generators, confirming that our transformer-based backbone better captures the long-range dependencies crucial for watermark removal.

Effect of PQN decoder and 4×4 patch size. We investigate decoder selection by replacing PQN (Row 2) with CNN (Row 3) and transformer blocks (Row 4). Comparisons show that our dynamic PQN design outperforms both CNN and transformer alternatives, offering more adaptive restoration. Additionally, reducing the patch size to 4 (Row 5) improves restoration quality and reduces computational costs, confirming that smaller patches better capture fine-grained watermark distortions.

Effect of RGB & Mask encoding. We first evaluate the impact of incorporating mask information (Row 6), which provides explicit guidance on watermark locations, improving restoration at boundary regions. Further enhancement comes from including original RGB values in the encoding (Row 7), enabling the network to use background pixel information as strong priors. The complete configuration (Row 8), combining positional encoding, RGB values, and mask information, achieves the best performance across all metrics. The progressive improvements from Rows 6-8 highlight how these cues work synergistically: the mask offers spatial guidance, while RGB values preserve background texture. This comprehensive encoding strategy is particularly effective for handling real-world watermarks with complex spatial distributions and varying opacity across diverse backgrounds.

6 Conclusions

In this work, we introduce a comprehensive framework for real-world visible watermark removal, comprising a two-stage method, PatchWiper, and the PRWD benchmark. PatchWiper features a decoupled architecture that separates watermark localization from background restoration, ensuring precise removal of diverse watermarks. The key innovation is our patch-wise restoration network, which dynamically generates unique representations for each patch, adapting to varying distortion areas while preserving fine details. Experimental results show that our method outperforms existing approaches, especially in complex real-world scenarios with diverse watermarks and backgrounds. These contributions advance watermark removal techniques and offer insights for developing more robust watermarking methods.

Acknowledgments

This work is supported in part by the National Key R&D Program of China under Grant No.2024YFB3908503, in part by the National Natural Science Foundation of China under Grant NO. 62322608, 62376206 and in part by the CCF-Kuaishou Large Model Explorer Fund (NO. CCF-KuaiShou2024007).

References

[1] Zhe Cao, Shuzi Niu, Jian Zhang, and Xiaogang Wang. 2019. Generative Adversarial Networks Model for Visible Watermark Removal. *IET Image Processing* 13, 10 (2019), 1783–1789.

[2] Junye Chen, Chaowei Fang, Jichang Li, Yicheng Leng, and Guanbin Li. 2025. Decouple and Couple: Exploiting Prior Knowledge for Visible Video Watermark Removal. *IEEE Transactions on Image Processing* 34 (2025), 1192–1203. doi:10.1109/TIP.2025.3534033

[3] Yinbo Chen, Sifei Liu, and Xiaolong Wang. 2021. Learning Continuous Image Representation With Local Implicit Image Function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8628–8638.

[4] Danni Cheng, Xiang Li, and Wei-Shi Zheng. 2018. Large-Scale Visible Watermark Detection and Removal with Deep Convolutional Networks. In *Pattern Recognition and Computer Vision (PRCV)*. Springer, 27–40.

[5] Xiaodong Cun and Chi-Man Pun. 2021. Split Then Refine: Stacked Attention-Guided ResUNets for Blind Single Image Visible Watermark Removal. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

[6] Mark Everingham, S.M. Ali Eslami, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. 2015. The PASCAL Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision* 111, 1 (2015), 98–136. doi:10.1007/s11263-014-0733-5

[7] Chaowei Fang, Guanbin Li, Xiaoguang Han, and Yizhou Yu. 2020. Self-Enhanced Convolutional Network for Facial Video Hallucination. *IEEE Transactions on Image Processing* 29 (2020), 3078–3090. doi:10.1109/TIP.2019.2955640

[8] David Ha, Andrew Dai, and Quoc V. Le. 2016. Hypernetworks. *arXiv preprint arXiv:1609.09106* (2016).

[9] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. 2021. Dynamic Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[10] Amir Hertz, Sharon Fogel, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. 2019. Blind Visual Motif Removal from a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[11] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. 2019. Meta-SR: A Magnification-Arbitrary Network for Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[12] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014). https://arxiv.org/abs/1412.6980

[13] Yicheng Leng, Chaowei Fang, Junye Chen, Yixiang Fang, Sheng Li, and Guanbin Li. 2025. Bridging Knowledge Gap Between Image Inpainting and Large-Area Visible Watermark Removal. arXiv:2504.04687 [cs.CV] https://arxiv.org/abs/2504.04687

[14] Yicheng Leng, Chaowei Fang, Gen Li, Yixiang Fang, and Guanbin Li. 2023. Removing Interference and Recovering Content Imaginatively for Visible Watermark Removal. arXiv:2312.14383 [cs.MM] https://arxiv.org/abs/2312.14383

[15] Fei Li, Lingfeng Shen, Yang Mi, and Zhenbo Li. 2022. DRCNet: Dynamic Image Restoration Contrastive Network. In *Computer Vision – ECCV 2022*, Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer Nature Switzerland, Cham, 514–532.

[16] Xiang Li, Chan Lu, Danni Cheng, Wei-Hong Li, Mei Cao, Bo Liu, Jiechao Ma, and Wei-Shi Zheng. 2019. Towards Photorealistic Visible Watermark Removal with Conditional Generative Adversarial Networks. In *Proceedings of the International Conference on Image and Graphics (ICIG)*.

[17] Jing Liang, Li Niu, Fengjun Guo, Teng Long, and Liqing Zhang. 2021. Visible Watermark Removal via Self-Calibrated Localization and Background Refinement. In *Proceedings of the ACM International Conference on Multimedia (ACMMM)*.

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV) (Lecture Notes in Computer Science, Vol. 8693)*. Springer, 740–755. doi:10.1007/978-3-319-10602-1_48

[19] Weihuang Liu, Xiaodong Cun, Chi-Man Pun, Menghan Xia, Yong Zhang, and Jue Wang. 2023. CoordFill: Efficient High-Resolution Image Inpainting via Parameterized Coordinate Querying. In *AAAI*.

[20] Yang Liu, Zhen Zhu, and Xiang Bai. 2021. WDNet: Watermark-Decomposition Network for Visible Watermark Removal. In *2021 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE.

[21] Chong Mou, Jian Zhang, and Zhuoyuan Wu. 2021. Dynamic Attentive Graph Learning for Image Restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 4328–4337.

[22] Li Niu, Xing Zhao, Bo Zhang, and Liqing Zhang. 2023. Fine-grained Visible Watermark Removal. In *ICCV*.

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv preprint arXiv:1912.01703* (2019).

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI) (Lecture Notes in Computer Science, Vol. 9351)*. Springer, 234–241. doi:10.1007/978-3-319-24574-4_28

[25] Yukai Shi, Guanbin Li, Qingxing Cao, Keze Wang, and Liang Lin. 2020. Face Hallucination by Attentive Sequence Optimization with Reinforcement Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 11 (2020), 2809–2824. doi:10.1109/TPAMI.2019.2915301

[26] Rui Sun, Yuxin Su, and Qi Wu. 2023. DENet: Disentangled Embedding Network for Visible Watermark Removal. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. AAAI Press, 2411–2419. doi:10.1609/aaai.v37i2.25337

[27] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. 2022. Resolution-robust Large Mask Inpainting with Fourier Convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2149–2159. doi:10.1109/WACV51458.2022.00220

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc., 5998–6008. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[29] Gourav Wadhwa, Abhinav Dhall, Subrahmanyam Murala, and Usman Tariq. 2021. Hyperrealistic Image Inpainting With Hypergraphs. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 3912–3921.

[30] Xiaohang Wang, Xuanhong Chen, Bingbing Ni, Hang Wang, Zhengyan Tong, and Yutian Liu. 2023. Deep Arbitrary-Scale Image Super-Resolution via Scale-Equivariance Pursuit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1786–1795.

[31] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. doi:10.1109/TIP.2003.819861

[32] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. 2022. Restormer: Efficient Transformer for High-Resolution Image Restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[33] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 586–595. doi:10.1109/CVPR.2018.00068

[34] Xing Zhao, Li Niu, and Liqing Zhang. 2022. Visible Watermark Removal with Dynamic Kernel and Semantic-Aware Propagation. In *Proceedings of the British Machine Vision Conference (BMVC)*.

[35] Bolei Zhou, Aditya Lapedriza, Aude Khosla, Aude Oliva, and Antonio Torralba. 2017. Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 6 (2017), 1452–1464. doi:10.1109/TPAMI.2017.2723009

[36] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. 2019. UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation. *IEEE Transactions on Medical Imaging* (2019).