Supplementary Material for "WAN: Watermarking Attack Network"

Seung-Hun Nam[1]
shnam1520@gmail.com
In-Jae Yu[2]
myhome98304@gmail.com
Seung-Min Mun[2]
qkqhd222@gmail.com
Daesik Kim[1]
daesik.kim@webtoonscorp.com
Wonhyuk Ahn[†1]
whahnize@gmail.com

[1] NAVER WEBTOON Corp.
Seongnam, South Korea
[2] Samsung Electronics
Suwon, South Korea

1 Introduction

In this supplementary material, we provide:

- experimental details about the MW methods and datasets,
- additional quantitative and qualitative results of the proposed WAN,
- additional quantitative and qualitative results of the proposed AoW.

2 Experimental Details

2.1 Details of MW Methods

We present properties and details of each MW method performing blind extraction. **M1**, **M2**, **M3**, and **M4** embed a watermark in the minimum unit, and aggregate the results extracted from the minimum unit using majority voting on the extraction process. Details of parameter setting of each WM method are listed in Table 1.

- **M1** [5] is designed to robust against desynchronization attacks. This method aligns the watermark synchronization using the template inserted in the image, and embeds the watermarks in the 1D DCT coefficients of each minimum unit (1×64) based on SS embedding.
- **M2** [6] embeds multiple watermarks, which are designed to minimize interference between each watermark, into minimum unit of size 16×16. The method applies DCT to each minimum unit, and then inserts a watermark into the coefficients using ISS embedding.

† Corresponding author

- **M3** [10] first obtain non-overlapping 3×3 pixels from minimum unit (8×8), and then QRD is applied to obtained pixels. This method embeds a watermark by adjusting the relation between the coefficients located in the first column of the second and third rows of orthogonal matrix Q.

- **M4** [7] selects significant minimum units (8×8) based on entropy map as HVS characteristics. After first-level DWT decomposition on the selected units, a watermark is inserted by applying subtle deformation to the U matrix of the SVD according to a predefined condition.

- **M5** [8] uses two-level NSCT decomposition and embeds watermark in the low coefficients of NSCT subbands using QT embedding for ensuring robustness against pixel-level shift. It improves the imperceptibility by adjusting the embedding strength based on computed perceptual masking value.

MW method	Watermarking domain / Embedding algorithm	Size of minimum unit	Extraction approach / Key characteristic	Parameter and value
M1 [6]	DCT / SS	1×64 (MV)	Blind / Template-based	$N = 1, M = 1, \alpha = 0.5, WM_{len} = 20, EM_{pos} = 15, \alpha_t = 5, \beta_t = 50$
M2 [6]	DCT / ISS	16×16 (MV)	Blind / Multiple watermarks	$\alpha = 1, \lambda = 1, K_{16\times16} = 80, EM_{pos} = 10, t_x = 0,$
M3 [10]	QRD / DIF	8×8 (MV)	Blind / Low false positive rate	$T = 0.03$, *Size of Matrix* $= 3 \times 3$, $x = y = \{1, 2, 3\}$
M4 [7]	DWT, SVD / DIF	8×8 (MV)	Blind / Considering HVS	$T = 0.02$, *Size of Matrix* $= 4 \times 4$
M5 [8]	NSCT / QT	–	Perceptual masking	$N = 1, M = 1, \alpha = 1, \beta = 0.7, \theta = 0.5, \varepsilon_1 = 0.2, \varepsilon_2 = 0.8, \eta_1 = 0.6,$ $\eta_2 = 0.4, \mathbb{L} = 8, \mathbb{S} = 2, \Delta = 2, \omega_1 = \omega_2 = 600, \mu = 0.1522$

† MV is abbreviation of the majority voting that aggregates extraction results of minimum units.
† *Details of parameters are set based on the notation of each paper [6, 6, 6, 7, 8, 10].*

Table 1: List of attributes and parameters of MW methods.

2.2 Details of Datasets

2.2.1 Grey-scale image dataset

As mentioned in our manuscript, we employ BOSSbase [2] and BOWS [1] datasets to generate 20,000 original grey-scale images with a size of 512×512. For base experiment, we resize them to 64×64 (i.e., $W = H = 64$) using the default settings in MATLAB R2018a, and the resized images are divided into three sets for training, validation, and test (with a $14 : 1 : 5$ ratio). The block-based MW methods [5, 6, 7, 8, 10] are used to generate watermarked images, and the images are generated by embedding watermark bits (0 or 1) into the original images given for each MW method. For further evaluation on scalability according to watermark capacity, we additionally generate test images sized 128×128 for the test set. Watermarked images with resolutions of 128×128 have a watermark capacity of 4 bits. In the experiment, watermark embedding, the WAN-based attacks, and watermark bit extraction proceed for each 64×64 patch. Furthermore, in this supplementary material, we present the results of WAN on test images sized 256×256 with 16 bits of watermark capacity. The detailed description of datasets is listed in Table 2.

MW method	1 bit (64×64)			4 bits (128×128)			16 bits (256×256)		
	# of B_o	# of B_{w_0}	# of B_{w_1}	# of B_o	# of B_{w_0}	# of B_{w_1}	# of B_o	# of B_{w_0}	# of B_{w_1}
M1 [6]	20,000	20,000	20,000	20,000	20,000	20,000	20,000	20,000	20,000
M2 [6]	20,000	20,000	20,000	20,000	20,000	20,000	20,000	20,000	20,000
M3 [10]	20,000	20,000	20,000	20,000	20,000	20,000	20,000	20,000	20,000
M4 [7]	20,000	20,000	20,000	20,000	20,000	20,000	20,000	20,000	20,000
M5 [8]	20,000	20,000	20,000	20,000	20,000	20,000	20,000	20,000	20,000
Total	100,000	100,000	100,000	100,000	100,000	100,000	100,000	100,000	100,000

† *Details on block images (e.g., $\tilde{B}_{w_0}$ or $\tilde{B}_{w_0}$) based on WAN's output are excluded from this table.*

Table 2: Details of grey-scale image datasets for our WAN.

2.2.2 Color image dataset

In this supplementary material, we present further quantitative and qualitative results of the WAN on color images. To do this, color image dataset, introduced in [9], consisting of single-compressed images based on RAISE [3] and Dresden [4] are exploited (https://github.com/plok5308/DJPEG-torch). Based on random sampling, we select 20,000 original color images with a size of 256×256. We resize them to 64×64 (i.e., $3 \times 64 \times 64$) using the default settings in MATLAB R2018a, and the resized images are divided into three sets for training, validation, and test (with a $14:1:5$ ratio). The number of the generated color images is the same as that of the grey-scale image with 1 bit of capacity (see left part of Table 2). After converting the RGB domain to YCbCr domain, watermarks are inserted into the Y-channel of given image for each MW method. Likewise, watermark extraction is performed on the Y-channel of given image. However, in the process of training and testing the WAN for color image dataset, RGB images are provided to the WAN as input. That is, the WAN is guided to invert the watermark bit inserted into a specific channel (i.e., Y-channel).

3 Additional Results of WAN

3.1 Additional Results of WAN on 16 Bits of Watermark Capacity

In this section, we further test for 16 bits watermark capacity scenario with the trained WAN model with stride 64. As listed on the right side of Table 3, the average PSNR, SSIM, and BER values for the attacked images over the WAN are 37.30 dB, 0.978, and 0.942, respectively. Compared with the results for 1 bit and 4 bits of watermark capacity, we can confirm that the WAN's overall performance is maintained even when the watermark capacity is increased.

MW method	Non-attack			WAN		
	PSNR	SSIM	BER	PSNR	SSIM	BER
M1 [5]	37.73	0.957	0.043	35.98	0.971	0.882
M2 [6]	43.11	0.985	0	38.30	0.982	0.990
M3 [10]	38.33	0.973	0	35.31	0.976	0.994
M4 [7]	39.88	0.979	0.003	38.44	0.980	0.991
M5 [8]	41.63	0.989	0.032	38.48	0.985	0.851
Average	40.14	0.977	0.015	37.30	0.978	0.942

Table 3: Quantitative evaluation results of the proposed WAN on the test set with 16 bits of watermark capacity.

3.2 Additional Results of WAN on Color Images

We present further quantitative and qualitative results of our WAN on color images in Table 4 and Fig. 1. To this end, we utilize the color image datasets, which is already introduced in Sec. 2.2. For each MW method, our WAN is trained with the hyperparameters $\lambda_c = 0.4$ and $\lambda_{wa} = 0.3$ during 50 epochs. Except for the channel of the input image (i.e., $C = 3$), the training settings are equal to the training methodology in our manuscript. In this experiment, the WAN is provided an RGB image as input and is trained to invert a watermark bit, which is embedded in the y-channel of given watermarked images.

Table 4 shows the performance results of our WAN on the grey-scale and color images with 1 bit of watermark capacity generated through each MW method. In non-attack situations, for color image, each method has a low BER value of 0.048 or less, while the average

MW method	Grey-scale image						Color image					
	Non-attack			WAN			Non-attack			WAN		
	PSNR	SSIM	BER	PSNR	SSIM	BER	PSNR	SSIM	BER	PSNR	SSIM	BER
M1 [8]	35.55	0.938	0.026	34.04	0.956	0.893	42.63	0.968	0.048	40.14	0.966	0.805
M2 [8]	41.86	0.988	0	37.47	0.979	0.996	43.14	0.979	0	39.51	0.971	0.942
M3 [48]	36.59	0.974	0	33.05	0.96	1.000	40.89	0.972	0	38.19	0.963	0.987
M4 [8]	38.98	0.986	0.002	37.70	0.985	0.988	41.03	0.969	0	38.41	0.965	0.977
M5 [8]	39.21	0.987	0.013	36.54	0.980	0.947	43.07	0.988	0.037	41.97	0.981	0.802
Average	38.44	0.974	0.008	35.76	0.972	0.965	42.15	0.975	0.017	39.64	0.969	0.903

Table 4: Quantitative evaluation results of the proposed WAN on color images with 1 bit.

BER value increases dramatically to 0.903 after WAN is applied. In case of color image, the average PSNR and SSIM values in non-attack situation are 42.15 dB and 0.975, respectively. After the WAN attacks images, average PSNR and SSIM decrease by 2.51 dB and 0.006, respectively. Next, Fig. 1 shows the qualitative results of our WAN on the color images. As shown in the figure, for each MW system, the proposed WAN hardly causes visual degradation in the process of inverting watermark bits (see the residual images in Fig. 1). In this experiment, WAN achieve acceptable performance for color image datasets in terms of quantitative and qualitative evaluation.

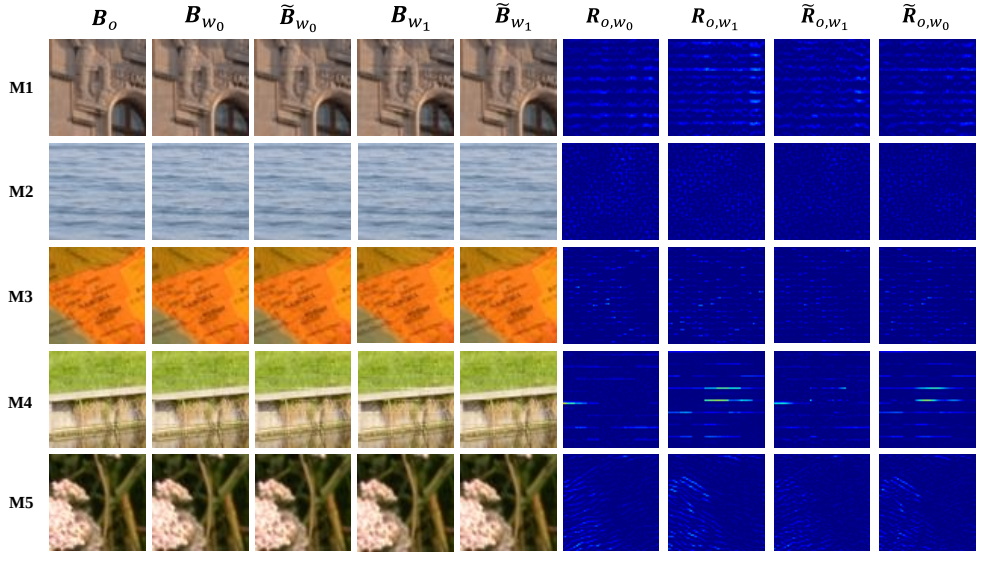

Figure 1: Qualitative evaluation results of the proposed WAN on color images with 1 bit.

4 Additional Results of AoW

In this section, we elaborate more on Add-on Watermarking (AoW) that WAN can be used to adjust the robustness or imperceptibility of pre-defined rule-based watermarking. As illustrated in Fig. 2, we introduce AoW-min and AoW-max, aiming for improving imperceptibility and robustness, respectively. When WAN attacks watermarked block with 1 bit embedded B_{w_1}, it tries to invert bit to 0 bit with $\tilde{R}_{o,w_1}$ by imitating residual between original block B_o and B_{w_0}, which is R_{o,w_0}. If attack is succeeded, $\tilde{R}_{o,w_1}$ would be similar to the R_{o,w_0}. The motivation of AoW is that $\tilde{R}_{o,w_1}$ can be used for fine-tuning B_{w_0}. AoW simply

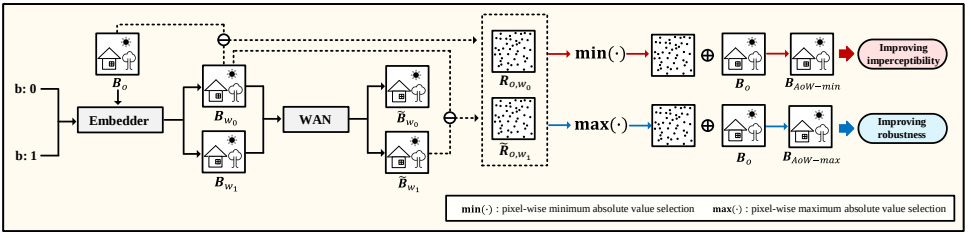

Figure 2: Schematic illustration of the proposed AoW framework.

compares residuals between $\tilde{R}_{o,w_1}$ and R_{o,w_0} element-wisely, and AoW-min and AoW-max add the minimum or maximum value to B_o, respectively.

Table 5 shows BER of five watermarking methods (B_w) and their AoW-min ($B_{AoW-min}$) and AoW-max ($B_{AoW-max}$) on JPEG, median blur, and noise addition attacks, and attack parameters of StirMark are specified. AoW-max shows lower BER than original methods as expected, and the improvements are more acquired in strong attacks than week attacks. Specifically, the improvements in JPEG quality factor of 60, MB parameter 4, and NA parameter 3 are notable. The BER of AoW-min are more degraded in strong attacks similarly. Next, we present the visualized results of residual images ($\bar{R}_{o,AoW-max}$ and $\bar{R}_{o,AoW-min}$) in Fig. 3, where $\bar{R}_{o,AoW-max}$ and $\bar{R}_{o,AoW-min}$ are defined as $B_o - B_{AoW-max}$ and $B_o - B_{AoW-min}$, respectively. As WAN inverts embedded bit successfully, we can find the R_{o,w_j} and $\bar{R}_{o,w_{j-1}}$ are similar for both bit ($j \in 0,1$). Although the patterns of residual are different according to watermarking methods, $B_{AoW-max}$ shows more distinct patterns than the R_{o,w_j}, and $\overline{R}_{o,AoW-max}$ leaves less footprints in images than R_{o,w_j}. Based on the results of Table 5 and Fig. 3, we can confirm that $B_{AoW-min}$ and $B_{AoW-max}$ can be alternative to original watermarking methods for adjusting their imperceptibility and robustness.

Type	WM method	JPEG param.				Median blur param.				Noise addition param.			
		60	70	80	Avg.	2	3	4	Avg.	1	2	3	Avg.
B_w	**M1** [8]	0.118	0.117	0.110	0.115	0.093	0.110	0.196	0.133	0.098	0.102	0.144	0.114
	M2 [8]	0	0	0	0	0.016	0.020	0.486	0.174	0	0	0.01	0.003
	M3 [16]	0.008	0.004	0	0.004	0.286	0.293	0.368	0.316	0	0.014	0.028	0.014
	M4 [4]	0.038	0.022	0.012	0.024	0.228	0.231	0.515	0.324	0.002	0.004	0.016	0.007
	M5 [8]	0.092	0.067	0.049	0.069	0.166	0.175	0.320	0.220	0.013	0.084	0.174	0.090
$B_{AoW-max}$	**M1** [8]	0.066	0.070	0.062	0.066	0.071	0.075	0.144	0.096	0.046	0.056	0.128	0.76
	M2 [8]	0	0	0	0	0.010	0.013	0.423	0.148	0	0	0	0
	M3 [16]	0.007	0.003	0	0.003	0.285	0.288	0.364	0.312	0	0.002	0.010	0.004
	M4 [4]	0.034	0.018	0.012	0.021	0.192	0.218	0.496	0.302	0	0	0.014	0.004
	M5 [8]	0.086	0.065	0.044	0.065	0.152	0.173	0.306	0.210	0.013	0.079	0.170	0.087
$B_{AoW-min}$	**M1** [8]	0.197	0.206	0.190	0.197	0.198	0.230	0.337	0.255	0.193	0.236	0.265	0.231
	M2 [8]	0.006	0	0	0.002	0.036	0.038	0.475	0.183	0	0	0.040	0.013
	M3 [16]	0.018	0.006	0	0.008	0.288	0.297	0.373	0.319	0	0.012	0.018	0.010
	M4 [4]	0.092	0.056	0.035	0.061	0.263	0.286	0.517	0.355	0.010	0.034	0.055	0.033
	M5 [8]	0.170	0.135	0.122	0.142	0.246	0.280	0.356	0.294	0.066	0.168	0.242	0.158

Table 5: Comparison of robustness of the proposed AoW against StirMark attacks of each parameter.

Figure 3: The qualitative results of AoW-min and AoW-max.

References

[1] Patrick Bas and Teddy Furon. Bows-2, 2007.

[2] Patrick Bas, Tomáš Filler, and Tomáš Pevnỳ. Break our steganographic system: the ins and outs of organizing boss. In *International workshop on information hiding*, pages 59–70. Springer, 2011.

[3] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM multimedia systems conference*, pages 219–224, 2015.

[4] Thomas Gloe and Rainer Böhme. The'dresden image database'for benchmarking digital image forensics. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1584–1590, 2010.

[5] Wook-Hyung Kim, Jong-Uk Hou, Han-Ul Jang, and Heung-Kyu Lee. Robust template-based watermarking for dibr 3d images. *Applied Sciences*, 8(6):911, 2018.

[6] Yu-Hsun Lin and Ja-Ling Wu. A digital blind watermarking for depth-image-based rendering 3d images. *IEEE transactions on Broadcasting*, 57(2):602–611, 2011.

[7] Nasrin M Makbol, Bee Ee Khoo, and Taha H Rassem. Block-based discrete wavelet transform-singular value decomposition image watermarking scheme using human visual system characteristics. *IET Image processing*, 10(1):34–52, 2016.

[8] Seung-Hun Nam, Seung-Min Mun, Wonhyuk Ahn, Dongkyu Kim, In-Jae Yu, Wook-Hyung Kim, and Heung-Kyu Lee. Nsct-based robust and perceptual watermarking for dibr 3d images. *IEEE Access*, 8:93760–93781, 2020.

[9] Jinseok Park, Donghyeon Cho, Wonhyuk Ahn, and Heung-Kyu Lee. Double jpeg detection in mixed jpeg quality factors using deep convolutional neural network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 636–652, 2018.

[10] Qingtang Su, Gang Wang, Xiaofeng Zhang, Gaohuan Lv, and Beijing Chen. An improved color image watermarking algorithm based on qr decomposition. *Multimedia Tools and Applications*, 76(1):707–729, 2017.