Pacific Graphics 2019
C. Theobalt, J. Lee, and G. Wetzstein
(Guest Editors)

Volume 38 (2019), Number 7

Naturalness-Preserving Image Tone Enhancement Using Generative Adversarial Networks (Supplementary Material)

Hyeongseok Son, Gunhee Lee, Sunghyun Cho, and Seungyong Lee

POSTECH

1. Additional Component Analysis

In our work, we use popular distance metrics to design our loss functions. However, they may be less optimal than other metrics used in recent deep learning literature. In this supplementary material, we examine the possibility of further improvement using more recent metrics for our loss functions, especially the naturalness loss and the artifact suppression loss.

Regarding the naturalness loss, although we adopt the least-squares generative adversarial networks (LSGAN) [MLX*17] in our system, more recent generative adversarial network (GAN) methods may improve the overall quality. To examine this possibility, we tested Wasserstein GAN using gradient penalty (WGAN-GP) [GAA*17] with several different parameters. Both LSGAN and WGAN-GP showed similar tone enhancement results, but we found that the results of WGAN-GP are slightly worse with less enhanced details and more artifacts as shown in Fig. 1. Nevertheless, a more advanced GAN framework may still improve the overall quality in future.

Regarding the artifact suppression loss, while we adopt mean-squared-error (MSE) in our system, a metric that can measure structural similarity such as SSIM [ZBSS04] can be more effective as our goal is to minimize structural artifacts. To verify this idea, we tested an artifact suppression loss using SSIM. As shown in Fig. 2, the results of SSIM have cleaner surfaces with high local contrasts than the results of MSE. This result demonstrates that our system can be further improved by SSIM.

2. Network Architectures

The specific architectures of our three networks G, C, and D are shown in Tables 1, 2, and 3, respectively.

layer type(#)	size	stride	out	norm	act.
Encoder					
*Conv*1_1	3×3	(1, 1)	32	*BN*	-
*Conv*2_1	3×3	(2, 2)	64	*BN*	*relu*
*Conv*2_2	3×3	(1, 1)	64	*BN*	*relu*
*Conv*2_3	3×3	(1, 1)	64	*BN*	-
*Conv*3_1	3×3	(2, 2)	128	*BN*	*relu*
*Conv*3_2	3×3	(1, 1)	128	*BN*	-
ResBlocks $\times16$					
Conv	3×3	(1, 1)	128	*BN*	*relu*
Conv	3×3	(1, 1)	128	-	
Decoder					
*Conv*4_1	3×3	(1, 1)	128	*BN*	-
add	*Conv*4_1, *BN*3_2			-	-
Upsample	*Nearest neighbor upsampling*				
*Conv*5_1	3×3	(1, 1)	64	*BN*	-
add	*Conv*5_1, *Conv*2_3			-	*relu*
*Conv*5_2	3×3	(1, 1)	64	*BN*	*relu*
*Conv*5_3	3×3	(1, 1)	64	*BN*	*relu*
Upsample	*Nearest neighbor upsampling*				
*Conv*6_1	3×3	(1, 1)	32	*BN*	-
add	*Conv*6_1, *Conv*1_1			-	*relu*
*Conv*6_2	3×3	(1, 1)	32	*BN*	*relu*
*Conv*6_3	3×3	(1, 1)	3	-	-
add	*Conv*6_3, *Input*			-	-

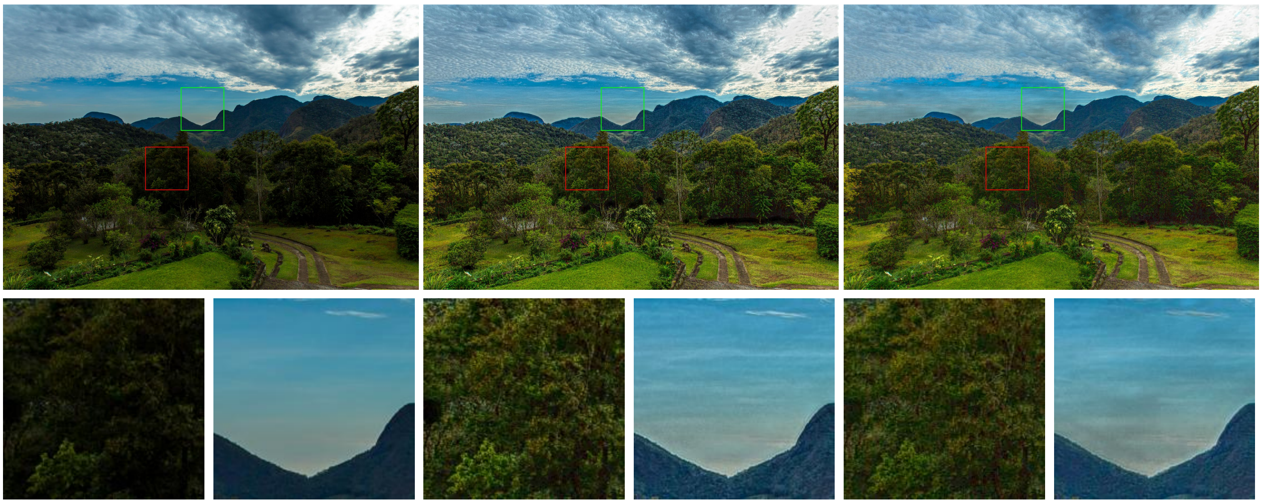
Table 1: *Architecture of our enhancement network G.*

References

[GAA*17] GULRAJANI I., AHMED F., ARJOVSKY M., DUMOULIN V., COURVILLE A. C.: Improved training of wasserstein GANs. In *Proc. NIPS*. 2017. 1

[MLX*17] MAO X., LI Q., XIE H., LAU R. Y., WANG Z., SMOLLEY S. P.: Least squares generative adversarial networks. *Proc. ICCV* (2017). 1

[ZBSS04] ZHOU WANG, BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P.: Image quality assessment: from error visibility to structural similarity. *IEEE TIP 13*, 4 (2004), 600–612. 1



(a) Input (b) Using LSGAN (c) Using WGAN

Figure 1: *Comparison with using WGAN instead of LSGAN for the naturalness loss.*

(a) Input (b) Using MSE (c) Using SSIM

Figure 2: *Comparison with using SSIM instead of MSE for the artifact suppression loss.*

layer type(#)	size	stride	out	norm	act.
*Conv*1	3×3	(2, 2)	64	*BN*	*relu*
*Conv*2	3×3	(1, 1)	64	*BN*	*relu*
*Conv*3	3×3	(1, 1)	64	*BN*	*relu*
*Conv*4	3×3	(2, 2)	64	*BN*	*relu*
*Conv*5	3×3	(1, 1)	64	-	-
add	*Conv*5, *Input*			-	-

Table 2: *Architecture of our inverse enhancement network C.*

layer type(#)	size	stride	out	norm	act.
*Conv*1_1	3×3	(1, 1)	64	*BN*	-
*Conv*2_1	3×3	(2, 2)	128	*BN*	*relu*
*Conv*2_2	3×3	(1, 1)	128	*BN*	*relu*
*Conv*3_1	3×3	(2, 2)	256	*BN*	*relu*
*Conv*3_2	3×3	(1, 1)	256	*BN*	*relu*
*Conv*4_1	3×3	(2, 2)	512	*BN*	*relu*
*Conv*4_2	3×3	(1, 1)	512	*BN*	*relu*
MeanPooling	24×24	(16, 16)	-	*BN*	*relu*
*Conv*5_1	3×3	(1, 1)	512	*BN*	*relu*
*Conv*5_2	3×3	(1, 1)	1	-	-
Flatten	-	-	-	-	*sigmoid*

Table 3: *Architecture of our discriminator network D.*