2022 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-9620-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIP46576.2022.9897229

CTGAN : CLOUD TRANSFORMER GENERATIVE ADVERSARIAL NETWORK

Gi-Luen Huang, Pei-Yuan Wu

National Taiwan University, Taiwan
Graduate Institute of Communication Engineering, Electrical Engineering
{r09942171, peiyuanwu}@ntu.edu.tw

ABSTRACT

Cloud occlusions obstruct some applications of remote sensing imagery, such as environment monitoring, land cover classification, and poverty prediction. In this paper, we propose the Cloud Transformer Generative Adversarial Network (CTGAN), taking three temporal cloudy images as input and generating a corresponding cloud-free image. Unlike previous work using generative networks, we design the feature extractor to maintain the weight of the cloudless region while reducing the weight of the cloudy region, and we pass the extracted features to a conformer module to find the most critical representations. Meanwhile, to address the lack of datasets, we collected a new dataset named Sen2_MTC from the Sentinel-2 satellite and manually labeled each cloudy and cloud-free image. Finally, we conducted extensive experiments on FS-2, the STGAN dataset, and Sen2_MTC. Our proposed CTGAN demonstrates higher qualitative and quantitative performance than the previous work and achieves state-of-the-art performance on these three datasets. The code is available at `https://github.com/come880412/CTGAN`

Index Terms— Cloud removal for multi-temporal cloudy images, generative adversarial network, conformer, Sentinel-2 satellite, FormoSat-2 satellite

1. INTRODUCTION

Remote sensing imagery has been used in many geoscience observation fields such as land cover classification [1, 2], environment monitoring [3], change detection [4, 5], forest canopy closure estimation [6], and poverty prediction [7, 8]. However, remote sensing imagery is inevitably affected by many factors, such as cloud occlusions, weather, and climate effects. Thick cloud occlusions will lose much of the information. Therefore, cloud removal is an indispensable preprocessing step before using remote sensing imagery in various applications.

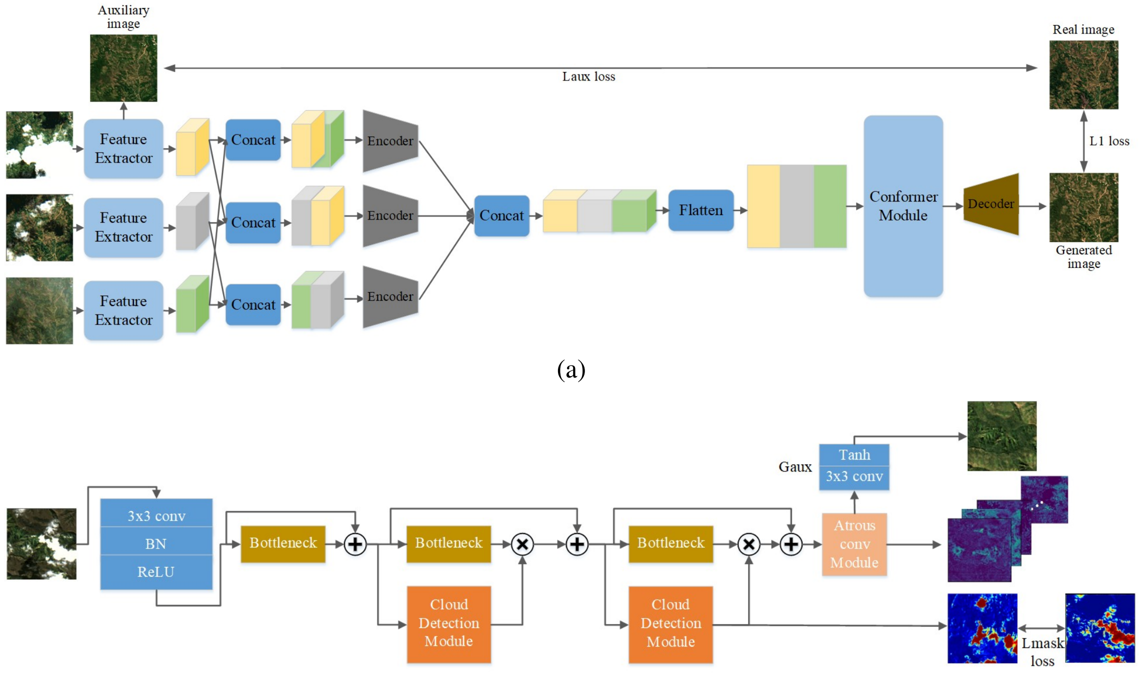
Cloud removal methods comprise single-image methods and multi-temporal methods. Single-image methods input one cloudy image to the network and generate a corresponding cloud-free image. Singh *et al.*[9] applied CycleGAN to remove cloud occlusions from synthetically generated cloudy images. Pan *et al.* [10] proposed a spatial-attention-based model for detecting the cloud's location and generating cloud-free images. Lee *et al.* [11] proposed a CNN-based model to synthesize realistic cloudy images and used the synthesized images to train the network for cloud removal. However, thick cloud occlusions will prevent single-image methods with only a few bands from restoring realistic cloud-free images [12].

To date, there is a great deal of research on single-image methods but comparatively little on multi-temporal methods. Multi-temporal methods can reconstruct thick cloudy images [10, 13]. Sintarasirikulchai *et al.* [14] designed an autoencoder-based model to fuse spectral information across multi-temporal data. Chen *et al.* [15] processed multi-temporal data by integrating feature maps of the spatial and temporal information. Sarukkai *et al.* [16] proposed the spatiotemporal generative network (STGAN) model for multi-temporal end-to-end training. Fig. 1 illustrates that the temporal cloudy images may have different visibility in the same region. However, [14, 15, 16] did not make additional processing of the features to differentiate between cloudy and cloud-free regions, which might hinder the model from restoring a realistic cloud-free image.

The main contributions of this paper are summarized as follows.

1. We propose the Cloud Transformer Generative Adversarial Network (CTGAN), a multi-temporal end-to-end training network. We focus on the design of the feature extractor and the processing of the downsampled features. The former uses the cloud mask to force the model to focus on the cloud-free region. The latter uses the attention mechanism in the conformer module to make the model find the most critical representations before restoring the cloud-free image. Meanwhile, our model can simultaneously detect cloud locations and restore the cloud-free image.

2. We collected a new dataset named Sen2_MTC for public use. The images in Sen2_MTC were gathered from the Sentinel-2 satellite, with manually labeled cloudy and cloud-free images.



(a)

(b)

Fig. 1: Generator of CTGAN (a) Generative network of CTGAN (b) Feature Extractor.

2. PROPOSED METHOD

Like [16], our CTGAN takes three cloudy images to recover a corresponding cloud-free image. Pairwise cloudy and cloud-free images are required to train CTGAN. We denote the input of temporal cloudy images as $x_c = \{x_1, x_2, x_3\}$ and the corresponding cloud-free image as y. Given x_c, the model learns how to generate the cloud-free image x_g, which should be similar to the corresponding cloud-free image y.

2.1. Generator

The overall CTGAN generative network is illustrated in Fig. 1(a). Our generator is based on STGAN [16]. However, unlike STGAN, we focus more on the design of the feature extractor and the processing of the downsampled multi-temporal features. The feature extractor structure is illustrated in Fig. 1(b). The bottleneck module, consisting of three convolutional layers followed by batch normalization and a rectified linear unit after each convolutional layer, extracts the feature representation of the image. This representation proceeds through the cloud detection module, consisting of three convolutional layers, passing through a sigmoid function before output. The cloud detection module detects the location of the cloud, and the generated cloud mask is multiplied by the feature map to keep the weight of the cloud-free region while reducing the weight of the cloudy region. In the previous layer of the feature extractor, we introduce the atrous convolution module [17] to enlarge the receptive field in the feature extractor. Moreover, inspired by [18], we include an auxiliary generator in the feature extractor to accelerate its convergence. In addition to the design of the feature extractor, we introduce the conformer module [19], which is the modified version of the original Transformer [20], to make the downsampled multi-temporal features find the most critical representations. Finally, the encoder and the decoder are convolutional layers with stride 2 to downsample the feature maps and the transposed convolutional layers with stride 2 to upsample the feature maps, respectively.

2.2. Discriminator

The CTGAN discriminator is a deep convolutional neural network. We utilize the conditional generative adversarial network (GAN). The network's input is the concatenation of the three cloudy images and the one generated or cloud-free image. In the prediction phase, the network carries out a binary classification to determine whether the concatenated image matches a generated or cloud-free image.

2.3. Loss function

In this work, the loss function can be defined as:

$$L = \min_G \max_D L_{cGAN}(G, D) + \lambda_G L_1(G) + L_{mask} + \lambda_{aux} L_{aux}, \quad (1)$$

where the parameters G and D represent the CTGAN generator and discriminator, and λ_G and λ_{aux} are the reconstruction quality weights of the loss, which are set to 100 and 50 in our model, respectively.

The loss function comprises four parts, where the first part is the loss function of the conditional GAN. We define the loss function of L_{cGAN} as:

$$L_{cGAN}(G, D) = E_{(x_c,y)}[\log D(x_c, y)] + E_{(x_c)}[\log(1 - D(x_c, G(x_c)))] \tag{2}$$

The second part is the standard L_1 loss function, defined as:

$$L_1(G) = \frac{1}{CWH} \sum_{c,w,h} \|y^{c,w,h} - G(x_c)^{c,w,h}\|_1, \tag{3}$$

where $G(x_c)^{c,w,h}$ denotes the pixels of the generated output image at coordinates (c, w, h). The third part is the cloud mask loss, defined as :

$$L_{mask} = \|M - M'\|_2^2, \tag{4}$$

where M and M' denote the ground-truth cloud mask and the predicted cloud mask, respectively. The fourth part is the auxiliary loss, defined as:

$$L_{aux} = \frac{1}{CWH} \sum_{c,w,h} \|y^{c,w,h} - G_{aux}(FE(x_c))^{c,w,h}\|_1, \tag{5}$$

where G_{aux} denotes the auxiliary generator in the feature extractor, and $FE(x_c)$ denotes the output of the feature extractor when feeding x_c into the network.

3. EXPERIMENTS

In this section, we employed our CTGAN on three different datasets and evaluated the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) [21] performance metrics.

3.1. Dataset and Implementation Details

FormoSat-2 (FS-2) is a decommissioned earth observation satellite formerly operated by the National Space Organization of Taiwan. The dataset contains 12 non-overlapping tiles, each with three cloudy images and a corresponding cloud-free image, $C = 4$ channels (R, G, B, NIR), and pixel value range $[0, 10000]$. Due to the lack of data in this dataset, we conducted 4-fold cross-validation to evaluate the performance and ensure the robustness of our model compared to previous works [14, 15, 16].

STGAN dataset [16] contains 945 distinct tiles, a total of 3101 images. The dataset was created using the publicly available Sentinel-2 images. [16] paired each cloud-free image with the three most recent cloudy images, each with size $(w, h) = (256, 256)$, $C = 4$ channels (R, G, B, NIR), and pixel value range $[0, 255]$. In addition, [16] randomly split the data into training/validation/testing sets with the ratio of 8:1:1 and kept the images from the same tile together.

Sen2_MTC was collected by us using publicly available Sentinel-2 images to annotate a new cloud removal dataset for multi-temporal training. The dataset contains 50 non-overlapping tiles, each with 70 images, pixel value range $[0, 10000]$, size $(w, h) = (256, 256)$, and $C = 4$ channels (R, G, B, NIR). We randomly split the data into training/validation/testing sets with a ratio of 7:1:2 and kept the images from the same tile together. [16] collected a satellite dataset from Sentinel-2 for multi-temporal training, but Fig. 2 shows that the images collected by [16] had low resolution and incorrect annotation, causing the model to have high quantitative but low qualitative performance in the early training stage. It also hindered the model from learning to generate a correct cloud-free image.

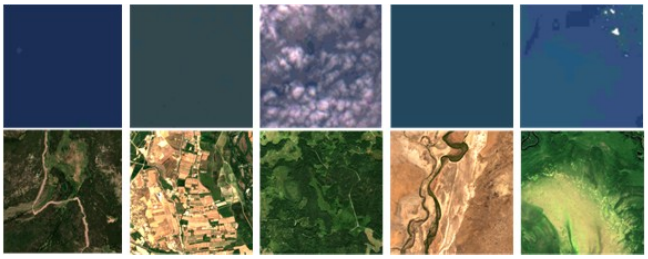


Fig. 2: Examples of cloud-free images in the STGAN_dataset (top row) and the Sen2_MTC dataset (bottom row)

Table 1: The performance evaluated by 4-fold cross validation on the FS-2 dataset.

PSNR	Fold1	Fold2	Fold3	Fold4	avg
AE [14]	16.85	17.04	17.42	15.57	16.72
ST_net [15]	18.27	18.21	18.15	16.57	17.80
STGAN [16]	18.28	17.32	18.28	17.40	17.82
CTGAN(ours)	**19.38**	**19.81**	**20.59**	**18.26**	**19.51(0.97)**
SSIM	Fold1	Fold2	Fold3	Fold4	avg
AE [14]	0.577	0.589	0.603	0.541	0.578
ST_net [15]	0.620	0.611	0.564	0.598	0.598
STGAN [16]	0.614	0.604	0.589	0.614	0.605
CTGAN(ours)	**0.666**	**0.662**	**0.689**	**0.657**	**0.669(0.012)**

Implementation details. Our proposed CTGAN was implemented via Pytorch and run on a server with two NVIDIA GeForce RTX 3090 GPUs, each with 24 GB of graphics memory. We first divided the pixel value by $10,000$ and normalized the pixel value range to $[-1, 1]$. In the training phase, we initially trained our model on the FS-2 dataset because we had the ground-truth cloud mask for the FS-2 dataset images. Next, we applied the semi-supervised learning technique to the Sentinel-2 dataset, using the feature extractor trained on the FS-2 dataset to generate the pseudo cloud mask on the Sentinel-2 dataset. In addition, we adopted the Adam optimizer with a learning rate of 5 x 10^{-4} and exponential decay rates (β_1, β_2) = (0.5, 0.999). We also used the CosineAnnealing scheduler to decay the learning rate per epoch and stopped training after 200 epochs.

3.2. Evaluation on FS-2

We reproduced [14, 15, 16] on the FS-2 dataset to compare the performance between our model and the previous works. When the authors provided the source code [16], we used the provided code to reproduce their model on the FS-2 dataset. Otherwise, we programmed it by ourselves from the description in their paper [14, 15]. We only evaluated their models' performance on our datasets because they did not release their datasets. The results are compared in Table 1, where the numbers in parentheses are the standard devi-

Table 2: Comparison of PSNR and SSIM results on the STGAN dataset [16].

Methods(PSNR/SSIM)	Validation	Test
Pix2Pix (RGB)	23.130/0.442	22.894/0.437
MCGAN (RGB + NIR)	21.352/0.485	21.146/0.481
Mean Filter	16.962/0.174	16.893/0.173
Median Filter	9.081/0.357	9.674/0.395
Raw Cloudy Images	7.926/0.389	8.289/0.422
STGAN U-Net (IR) [16]	25.142/0.651	25.388/0.661
STGAN ResNet(IR) [16]	25.628/0.724	26.186/0.734
CTGAN(ours)	**26.149/0.805** (0.438)/(0.017)	**26.264/0.808** (0.204)/(0.011)

Table 3: Comparison of PSNR and SSIM results on the Sen2_MTC dataset.

Methods(PSNR/SSIM)	Validation	Test
AE [14]	16.010/0.431	15.251/0.412
ST_net [15]	17.741/0.467	16.206/0.427
STGAN [16]	20.612/0.613	18.152/0.587
CTGAN(ours)	**21.259/0.662** (0.046)/(0.003)	**18.308/0.609** (0.089)/(0.007)

ation (STD) of our model. On the FS-2 dataset, the results shown in Table 1 demonstrate that the design of our model is effective. The improvement of SSIM significantly outperformed the previous state-of-the-art model STGAN, with a breakthrough gain of SSIM 0.064.

3.3. Evaluation on the STGAN dataset

We evaluated our CTGAN on the dataset collected by [16]. The performance is shown in Table 2. [16] did not describe their random seed to split the dataset in their paper, so we trained our model 10 times using the same data-splitting method with different random seeds and averaged these results to obtain the final result. The SSIM improvement of our CTGAN is considerable. The SSIM on the validation set and the testing set of the previous state-of-the-art STGAN are 0.724 and 0.734, respectively. Our CTGAN significantly outperformed the previous state-of-the-art STGAN, the gain of SSIM on the validation set and the testing set are 0.081 and 0.074, respectively. The experimental result on the STGAN dataset also demonstrates that our CTGAN can achieve state-of-the-art performance on the STGAN dataset.

3.4. Evaluation on Sen2_MTC

The method of reproducing STGAN [16], ST_net [15], and AE [14] is the same as described in section 3.2. We also evaluated CTGAN on the Sen2_MTC dataset. The results shown in Table 3 demonstrate that CTGAN achieves higher quantitative performance than [14, 15, 16]. The SSIM on the validation set and the testing set of the previous state-of-the-art STGAN are 0.613 and 0.587, respectively. Our CTGAN outperformed the previous state-of-the-art STGAN, the gain of SSIM on the validation set and the testing set are 0.049 and 0.022, respectively. The improvement for SSIM is usually shown that the model is more capable of restoring the visible structure. Therefore, CTGAN can reconstruct more information in the generated cloud-free image than the previous state-of-the-art model STGAN, as seen in Fig. 3.

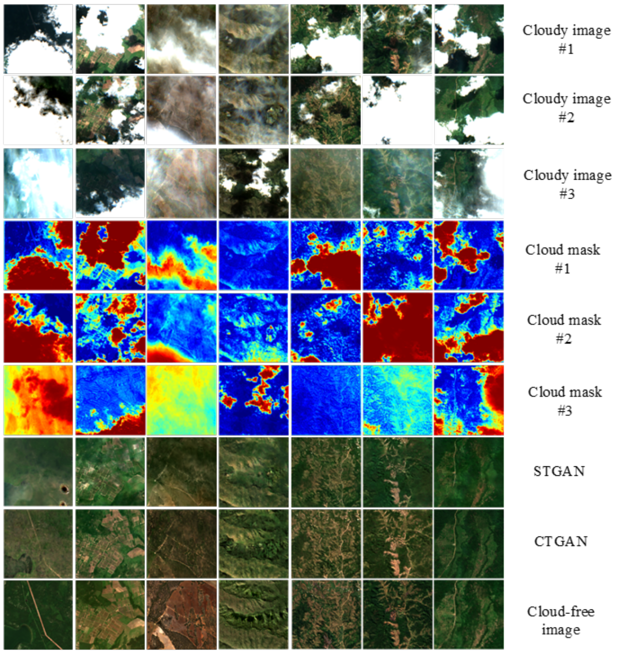


Fig. 3: Visualized results of the generated images on the Sen2_MTC dataset.

3.5. Visualization on Sen2_MTC

We visualized the generated results using CTGAN and the previous state-of-the-art STGAN [16]. The results are shown in Fig. 3. From top to bottom are the three cloudy input images, three cloud masks generated by CTGAN, the cloud-free image generated by STGAN, the cloud-free image generated by CTGAN, and the corresponding ground-truth cloud-free image. Fig. 3 shows that our CTGAN can restore a cloudy image more like the actual image, while the image generated by STGAN has many artifacts. Even if the details of the three images are nearly lost, our CTGAN can still roughly generate the shapes under the clouds (first column of Fig. 3).

4. CONCLUSION

We propose CTGAN for multi-temporal cloud removal. Unlike previous work, We focus more on the design of the feature extractor and processing of the downsampled multi-temporal features. In addition, to solve the lack of datasets for multi-temporal cloud removal, we collected a new dataset from Sentinel-2, which we named Sen2_MTC, and labeled each cloudy and cloud-free image. Finally, we experimentally demonstrated that CTGAN can achieve high qualitative and quantitative performance and significantly outperformed the previous state-of-the-art models. We will also release the Sen2 MTC dataset for public use.

5. REFERENCES

[1] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Journal*, 2019.

[2] Rodrigo Minetto, Maurício Pamplona Segundo, and Sudeep Sarkar, "Hydra: An ensemble of convolutional neural networks for geospatial land classification," *IEEE Transactions on GRS*, 2019.

[3] Ainong Li, Jinhu Bian, Guangbin Lei, Xi Nan, and Zhengjian Zhang, "Remote sensing monitoring and integrated assessment for the eco-environment along china-pakistan economic corridor," in *IGARSS*, 2019.

[4] Maria Papadomanolaki, Sagar Verma, Maria Vakalopoulou, Siddharth Gupta, and Konstantinos Karantzalos, "Detecting urban changes with recurrent neural networks from multitemporal sentinel-2 data," in *IGARSS*, 2019.

[5] Daifeng Peng, Yongjun Zhang, and Haiyan Guan, "End-to-end change detection for high resolution satellite images using improved unet++," *Remote Sensing*, 2019.

[6] Shanshan Sun, Zengyuan Li, Xin Tian, Zhihai Gao, Chongyang Wang, and Chengyan Gu, "Forest canopy closure estimation in greater khingan forest based on gf-2 data," in *IGARSS*, 2019.

[7] Partha Sarathi Das, Harsh Chhabra, and Sanjay Kumar Dubey, "Socio economic analysis of india with high resolution satellite imagery to predict poverty," in *International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2020.

[8] Arwa Okaidat, Shatha Melhem, Heba Alenezi, and Rehab Duwairi, "Using convolutional neural networks on satellite images to predict poverty," in *ICICS*, 2021.

[9] Praveer Singh and Nikos Komodakis, "Cloud-gan: Cloud removal for sentinel-2 imagery using a cyclic consistent generative adversarial networks," in *IGARSS*, 2018.

[10] Heng Pan, "Cloud removal for remote sensing imagery via spatial attention generative adversarial network," *arXiv*, 2020.

[11] Kyu-Yul Lee and Jae-Young Sim, "Cloud removal of satellite images using convolutional neural network with reliable cloudy image synthesis model," in *ICIP*, 2019.

[12] Andrea Meraner, Patrick Ebel, Xiao Xiang Zhu, and Michael Schmitt, "Cloud removal in sentinel-2 imagery using a deep residual neural network and sar-optical data fusion," *ISPRS Journal*, 2020.

[13] Meng Xu, Xiuping Jia, Mark Pickering, and Antonio J Plaza, "Cloud removal based on sparse representation via multitemporal dictionary learning," *IEEE Transactions on Geoscience and Remote Sensing*, 2016.

[14] Wassana Sintarasirikulchai, Teerasit Kasetkasem, Tsuyoshi Isshiki, Thitiporn Chanwimaluang, and Preesan Rakwatin, "A multi-temporal convolutional autoencoder neural network for cloud removal in remote sensing images," in *ECTI-CON*, 2018.

[15] Yang Chen, Qihao Weng, Luliang Tang, Xia Zhang, Muhammad Bilal, and Qingquan Li, "Thick clouds removing from multitemporal landsat images using spatiotemporal neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[16] Vishnu Sarukkai, Anirudh Jain, Burak Uzkent, and Stefano Ermon, "Cloud removal in satellite images using spatiotemporal generative networks," in *WACV*, 2020.

[17] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv*, 2017.

[18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[19] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al., "Conformer: Convolution-augmented transformer for speech recognition," *arXiv*, 2020.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017.

[21] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, 2004.