Terrain Scene Generation Using A Lightweight Vector Quantized Generative Adversarial Network

Yan Wang, Huiyu Zhou, and Xinghui Dong, *Member, IEEE*

***Abstract*—Natural terrain scene images play important roles in the geographical research and application. However, it is challenging to collect a large set of terrain scene images. Recently, great progress has been made in image generation. Although impressive results can be achieved, the efficiency of the state-of-the-art methods, e.g., the Vector Quantized Generative Adversarial Network (VQGAN), is still dissatisfying. The VQGAN confronts two issues, i.e., high space complexity and heavy computational demand. To efficiently fulfill the terrain scene generation task, we first collect a Natural Terrain Scene Data Set (NTSD), which contains 36,672 images divided into 38 classes. Then we propose a Lightweight VQGAN (Lit-VQGAN), which uses the fewer parameters and has the lower computational complexity, compared with the VQGAN. A lightweight super-resolution network is further adopted, to speedily derive a high-resolution image from the image that the Lit-VQGAN generates. The Lit-VQGAN can be trained and tested on the NTSD. To our knowledge, either the NTSD or the Lit-VQGAN has not been exploited before[1]. Experimental results show that the Lit-VQGAN is more efficient and effective than the VQGAN for the image generation task. These promising results should be due to the lightweight yet effective networks that we design.**

***Index Terms*—Terrain scenes, natural terrains, image generation, super-resolution, lightweight networks.**

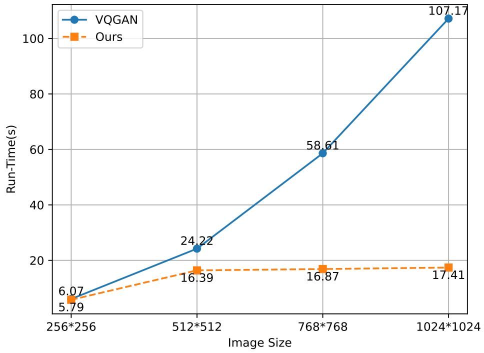

Fig. 1. Comparison of the time used for generating terrain images of different sizes using the VQGAN [11] and our method. As can be seen, our method performs image generation more efficiently than the VQGAN. In particular, this is the case when the image is larger than 256×256 pixels. Given that 256×256 terrain images are generated using our method and the VQGAN, the FID scores computed between the images generated and those in the testing set are 24.36 and 30.83 respectively. It is suggested that our method generates images with the higher quality than those produced by the VQGAN.

I. Introduction

Natural terrain scene images are key to the geographical research and application, for example, terrain scene recognition and understanding. However, it is hard to collect a large number of terrain scene images, in particular, covering diverse categories. In recent years, great progress has been made in the field of Artificial Intelligence Generated Content (AIGC) [1, 2, 3], such as dialog generation, image generation and cross-modal generation. In general, image generation has been studied on top of Generative Adversarial Networks (GANs) [4, 5], Variational Autoencoders (VAEs) [6, 7, 8] and Autoregressive (AR) models [9, 10].

The GAN [4] normally contains a generator network and a discriminator network. The generator is used to produce fake images while the discriminator attempts to distinguish the real images from the fake images. Each network trains the other by continuously improving their performance. In contrast, the VAE encodes the input image into a vector in the latent space and decodes this vector to a novel image. Unlike the GAN, the VAE is trained by minimizing the reconstruction error and the regularization term in the latent space. Besides, the AR model generates images in a pixel-by-pixel manner. Thus, this method is usually slower than the other methods. But the AR model can generate high quality and high-resolution images.

Recently, Esser et al. [11] proposed a Vector Quantized Generative Adversarial Network (VQGAN) by exploring the long-range dependencies modeled by Transformers in the discrete space in order to improve the AR model. The training process of the VQGAN can be divided into two stages. First, the VQGAN uses an encoder to map the input image to a set of vectors in the latent space. The latent vectors are decoded to a new image using the decoder. These images are used to train a vector quantizer. The quantizer maps the latent vectors to a set of discrete vectors, which are comprised of a codebook. Second, the AR model samples a set of latent vectors from a uniform distribution. Each vector is mapped to the closest code vector in the codebook. The code vectors are fed into the decoder. As a result, an image is generated by decoding.

Although the VQGAN can generate diverse high-resolution images, it still encounters two issues, i.e., high space complexity and heavy computational demand. To generate high-resolution images, in particular, the sampling and decoding

This study was in part supported by the National Natural Science Foundation of China (NSFC) (No. 42176196) and was in part supported by the Young Taishan Scholars Program (No. tsqn201909060) (Corresponding author: Xinghui Dong).

Y. Wang and X. Dong are with the School of Computer Science and Technology, Ocean University of China, Qingdao, 266100. (e-mail: wangyan6183@stu.ouc.edu.cn, xinghui.dong@ouc.edu.cn). H. Zhou is with the School of Computing and Mathematical Sciences, University of Leicester, LE1 7RH Leicester, U.K. (e-mail: hz143@leicester.ac.uk).

[1] The data set and code are available at: https://indtlab.github.io/projects/Lit-VQGAN.

operation has to be performed in a moving-window manner. In other words, this operation needs to be conducted in each window. Due to the computational complexity of the sampling process, the speed of high-resolution image generation is slow.

Since it is challenging to obtain a large set of natural terrain scene images, we aim to overcome this challenge by exploiting image generation techniques. To efficiently perform the natural terrain scene generation task, we first collect a Natural Terrain Scene Data Set (NTSD) which covers diverse terrain categories. In total, 36,672 images which are divided into 38 terrain classes are comprised of this data set. The data set can be used to train an image generation network.

Then we introduce a Lightweight Vector Quantized Generative Adversarial Network (Lit-VQGAN), to address the two issues. Specifically, two lightweight blocks are designed, including a Local Feature Extraction Block (LFEB) and an Efficient Feature Fusion Block (EFFB). The LFEB is used to extract local features while the EFFB can capture both the local and global information. The Lit-VQGAN is built on top of the two blocks. Thus, it utilizes fewer parameters, compared with the VQGAN. Also, we adopt a lightweight super-resolution network using the Complex Attention Block (CAB) that we design. This network is used to derive a high-resolution image from the image generated by the decoder. In contrast to the moving-window sampling and decoding scheme that the VQGAN uses, our solution is able to perform the high-resolution image generation task more efficiently (see Fig. 1).

To our knowledge, either the NTSD or the Lit-VQGAN has not been explored before. The contributions of this study are summarized as threefold.

- We collect a new Natural Terrain Scene Data Set (NTSD). This data set can be used to train an image generation network in order to derive more terrain images for the geographical research and application.
- We propose a lightweight VQGAN [11] which is built using two lightweight blocks that we deliberately design, including the Local Feature Extraction Block (LFEB) and the Efficient Feature Fusion Block (EFFB). As a result, both the training and the inference of this network are more efficient, compared with the VQGAN [11].
- To overcome the speed bottleneck of high-resolution image generation, we adopt a lightweight super-resolution network. The sampling and decoding operation is only conducted once instead of being performed in a moving-window style [11]. Hence, the speed of high-resolution image generation is greatly accelerated.

The rest of this paper is organized as follows. The related work is reviewed in Section II. We introduce our Natural Terrain Scene Data Set (NTSD) in Section III. The proposed Lit-VQGAN is proposed in Section IV. In Sections V and VI, experimental setup and results are reported respectively. We draw our conclusions in Section VII.

II. Related Work

A. Image Synthesis

Deep generative models have achieved many successes in different image synthesis tasks. Although high-fidelity images can be generated using GAN-based methods, likelihood-based methods, such as Variational Autoencoders (VAEs) [6, 7, 8, 12], diffusion models [13, 14, 15] and AR models [9, 10], usually provide more diverse images.

To generate high-quality, realistic images, a diffusion model normally combines the diffusion process with a generative model. Rombach et al. [16] proposed a diffusion model based on the latent representation space, which reduced the computational complexity and increased the training speed. In [17], the DiT method was developed by applying the transformer architecture to a diffusion model. For the purpose of enhancing the quality of the images generated and the stability of the training process, Nichol and Dhariwal [18] utilized an improved Variational Lower Bound (VLB) as the training objective.

The generation process of the VQVAE [7] contains two stages. First, images are quantized into the latent space. Second, sampling and decoding are conducted in this space. Many studies [11, 13, 19] were inspired by the two-stage approach. The Masked Generative Image Transformer (MaskGIT) [19] method was also built based on the two-stage approach. Particularly, this method was focused on improving the efficiency of the second stage in which an AR model was used by introducing a bidirectional Transformer. To accelerate the image generation task, the VQ-Diffusion [13] approach brought the two-stage method and the diffusion model together.

In contrast, we utilize the lightweight encoder-decoder network and a lightweight super-resolution network to address the two issues that the VQGAN [11] encountered.

B. Efficient Neural Networks

Convolutional neural networks (CNNs) have dominated various computer vision tasks until now. Due to the increasing demand for application of neural networks to mobile and robotic systems, efficient network design has been given much attention. Howard et al. [20] built the MobileNetV1 by proposing the Depthwise Separable Convolution, which greatly reduced the number of parameters and the computational cost. Specifically, the standard convolution was decomposed into a Depthwise Convolution and a Pointwise Convolution. To enhance the representation ability of the network, Howard et al. [21] further developed the MobileNetV2 using inverted residual blocks and linear bottlenecks. In [22], the MobileNetV3 was introduced on top of the Adaptive Width and Automated Neural Architecture Search techniques in order to improve the accuracy and efficiency of the network.

The MobileNeXt [23] was designed based on the Sandglass block, which improved the Inverted Residuals block used in the MobileNetV2 [21]. In [24], a lightweight convolutional neural network, namely, ShuffleNet, was proposed on top of the channel shuffling operation, which effectively decreased both the model size and the computational complexity. Tan and Le [25] adopted the EfficientNet by balancing the scaling in the network depth, width and resolution.

On the other hand, Transformers have been attracting more and more attention because they can model long-range dependencies. However, the computational cost is prohibitive for mobile and robotic applications. To save the computational cost, Li et al. [26] built the SepViT by designing a deep separable self-attention module. In [27], a Light Vision Transformer, i.e., Light ViT, was developed. A global information token was added to the self-attention unit, which was embedded into the local information for interaction. Wang et al. [28] further reduced the computational cost by replacing the standard Multi-head Attention (MHA) with the Spatial-Reduction Attention (SRA). The key point of the SRA is to reduce the number of key-value pairs used in the attention layer.

It has been shown that a hybrid architecture which combines the convolution and Transformer is able to capture both local and global features better than a CNN or Transformer network. Li et al. [29] developed the Next Convolution Block and the Next Transformer Block in order to encode the local and global information respectively. In [30], the bidirectional fusion of local and global features was used to build the Mobile-Former by exploiting the advantage of the MobileNet [20] in the local attention and the merit of the Transformer in the global interaction. To eliminate inefficient frequent reshape operations, Li et al. [31] designed a dimensionally consistent network, namely, Efficient-Former, using the 4D feature implementation and 3D Multi-head Self-Attention (MHSA).

Inspired by these studies, we design a Lightweight VQ-GAN (Lit-VQGAN), which efficiently exploits both the local and global features through the lightweight blocks that we adopt. This network involves fewer parameters and the lower computational complexity, compared with the VQGAN [11].

C. Efficient Super-Resolution Networks

To improve the efficiency of models, lightweight and efficient super-resolution networks become popular. The computational burden was reduced using post-upsampling [32]. Hui et al. [33] developed a lightweight multi-distillation network for the sake of performing fast and accurate image super-resolution. In [34], the performance was improved by using multiple attention mechanisms to refine and extract features. Sun et al. [35] adopted the ShuffleMixer by introducing the channel split and channel shuffle operations, which can be used to effectively perform feature fusion.

Recently, large kernel convolutions have received much attention. As an early convolutional neural network (CNN) based on large kernels, AlexNet [36] involved a large number of parameters and encountered the high computational cost. In [37], the 7×7 depthwise convolution was used for the ConvNeXt. As a result, this network outperformed the corresponding ViT network. The better results were derived using the RepLKNet [38] which enlarged the size of the convolution kernel to 31×31. Guo et al. [39] demonstrated that large convolution kernels can be effectively decomposed into a set of convolutions, including the depthwise convolution, dilated convolution and pointwise convolution, by experimentation.

To efficiently perform high-resolution image generation, we propose a lightweight super-resolution network. This network uses both the effective decomposition of large convolutional kernels and channel shuffling to fuse both the local and global features. Our network can be used to generate a high-resolution image after only applying the sampling and decoding operation once. In contrast, the VQGAN [11] uses a moving-window sampling and decoding scheme in which the sampling and decoding operation has to be applied to each window.

III. Natural Terrain Scene Data Set

Although many terrain data sets have been collected, they normally contain remote sensing images, e.g., aerial or satellite images. In this study, we particularly pay attention to natural terrain scene images, which can be used for the geographical research and application. To this purpose, we collected a Natural Terrain Scene Data Set (NTSD). This data set covers diverse geographic scenes and presents various landforms. Compared with the existing scene data sets [40, 41], the NTSD is specifically focused on natural terrain categories.

A. Data Collection and Copyright

We collected images from Unsplash [42], Pixabay [43], Pexels [44], Flickr [45] and Google Search [46]. The images downloaded from Unsplash can be used freely. All images provided by Unsplash can be used for both the commercial and non-commercial purposes. The images downloaded from Pixabay and Pexels are subject to the Creative Commons Zero license. This license allows the copying, modification, distribution and commercial use of the work. When we downloaded images from Flickr and Google Search, we followed the licensing guidelines for each image.

B. Statistics of the Data Set

In total, we collected 36,672 terrain scene images, which were divided into 38 classes. In terms of each class, an example image is shown in Fig. 2. As can be seen, not only both terrestrial and marine terrains, including mountains, farmland, terraces, beaches, islands, reefs, sandbars, etc., but also unique landforms, such as Adarce, Danxia and Yardang, are covered. To demonstrate the statistics of the images included in the 38 classes, we present the number of the images contained in each class in Fig. 3. It can be observed that the distribution of different classes is relatively uniform except that the Mangrove class comprises a relatively large number of images. In average, each class consists of 965 images. This number suggests a large intra-class variation.

IV. The Lightweight Vector Quantized Generative Adversarial Network

Considering the existing image generation networks usually encounter the challenges of high space complexity and heavy computational demand, we introduce a Lightweight Vector Quantized Generative Adversarial Network (Lit-VQGAN). The architecture of this network is shown in Fig. 4. Specifically, the Lit-VQGAN comprises a VQGAN [11] network which is built on top of a series of lightweight blocks and a lightweight super-resolution network. Compared with the original VQGAN, our network uses fewer parameters and performs the image generation task more efficiently.

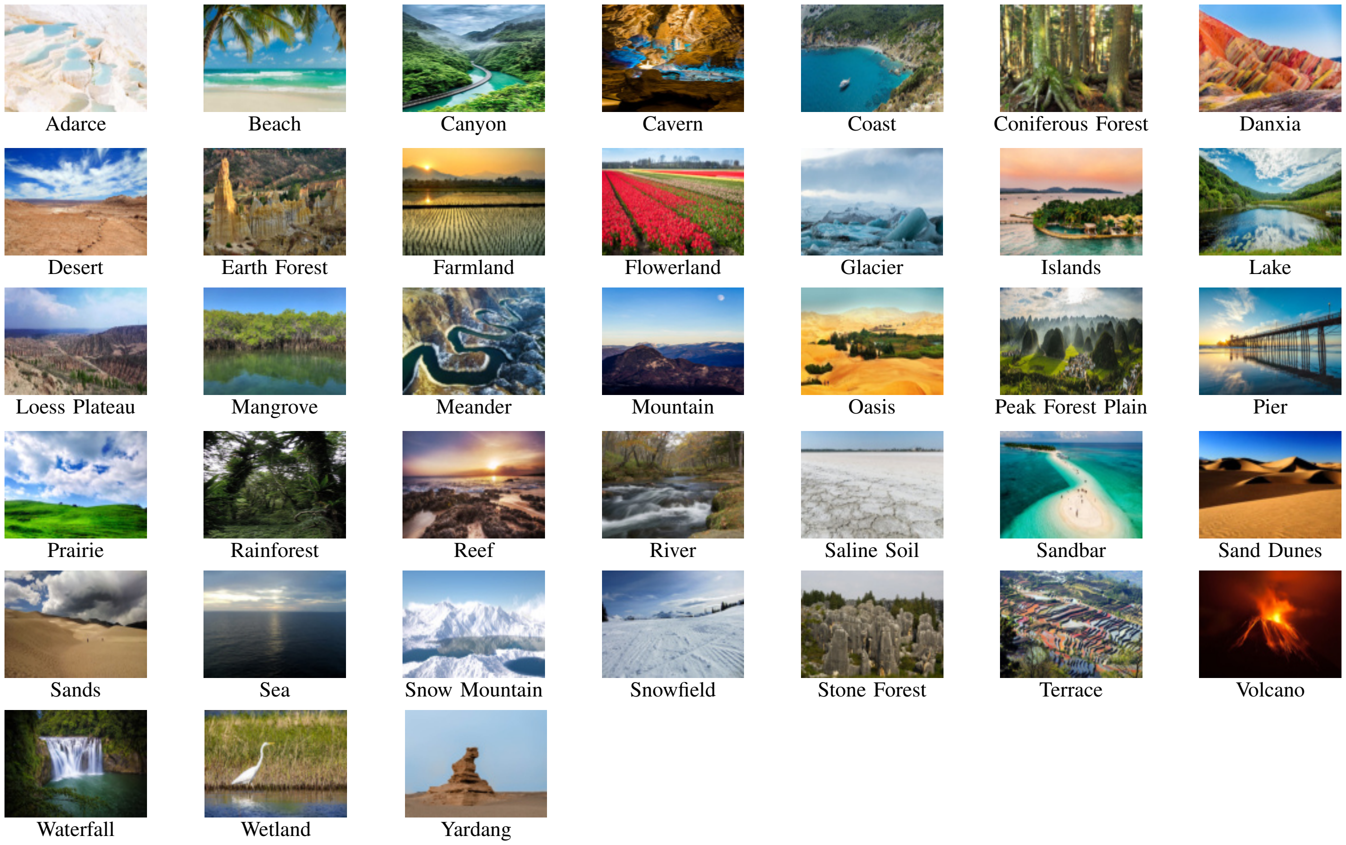

Fig. 2. An example is shown in terms of each of the 38 classes contained in the Natural Terrain Scene Data Set (NTSD).

A. *Overview*

The Lit-VQGAN consists of a lightweight encoder-decoder with a Vector Quantization (VQ) module, a discriminator and a lightweight super-resolution network. The encoder contains a series of lightweight Local Feature Extraction Blocks (LFEBs) and Efficient Feature Fusion Blocks (EFFBs). The decoder has a symmetrical structure to the encoder. Both the VQ module and the discriminator maintain the same structure as that the VQGAN [11] utilizes. On top of the architecture of the ShuffleMixer [35], the lightweight super-resolution network is built using a set of Complex Attention Blocks (CABs). The computational complexity of the proposed Lit-VQGAN is lower than that of the original VQGAN due to the applications of the lightweight blocks and the large kernel convolution decomposition technique.

The training process of the Lit-VQGAN is described as follows. Given that an image is fed into the encoder, the feature maps produced are sent to the VQ module. As a result, they are mapped into a discrete feature space. The quantized feature maps are then fed into the decoder. The result is a reconstructed image in terms of the input image. This image is determined by the discriminator. The loss function that the VQGAN [11] employs is used here.

The image generation process is different from that used by the VQGAN [11]. Instead of performing the sampling and decoding operation in a moving-window style [11], we only conduct this operation once and then use the super-resolution network which is pre-trained in advance to derive a high-resolution image. Specifically, the sampling procedure is conducted using an AR model in the discrete feature space. The result is fed into the decoder and a 256×256 image is generated. The image generated is sent to the pre-trained super-resolution network to derive a high-resolution image.

B. *Local Feature Extraction Blocks (LFEBs)*

Considering that the resolution of the feature maps extracted at the shallow part of the encoder is relatively high, global characteristics may greatly increase both the number of parameters and the computational complexity. To balance the efficiency and effectiveness, we introduce a Local Feature Extraction Block (LFEB) which is a lightweight unit in essence. The LFEB consists of a Sandglass Block (SG Block) [23] and a Multilayer Perceptron (MLP) layer, each of which utilizes a residual connection. The SG Block contains two depthwise convolutions in order to encode the spatial information. For the purpose of capturing the relationship between different channels, the MLP layer is appended to the SG Block. The LFEB can be formulated as follows:

$$\tilde{x}^{l} = \mathrm{SG}(\tilde{x}^{l-1}) + \tilde{x}^{l-1}, \tag{1}$$

$$x^{l} = \mathrm{MLP}(\tilde{x}^{l}) + \tilde{x}^{l}, \tag{2}$$

where $\tilde{x}^{l-1}$ denotes the feature maps generated by the $(l$-1)-th block, and $\tilde{x}^{l}$ and x^{l} are the feature maps produced by the SG Block and the MLP respectively. Compared with the popular lightweight blocks, such as the inverted residual structure of

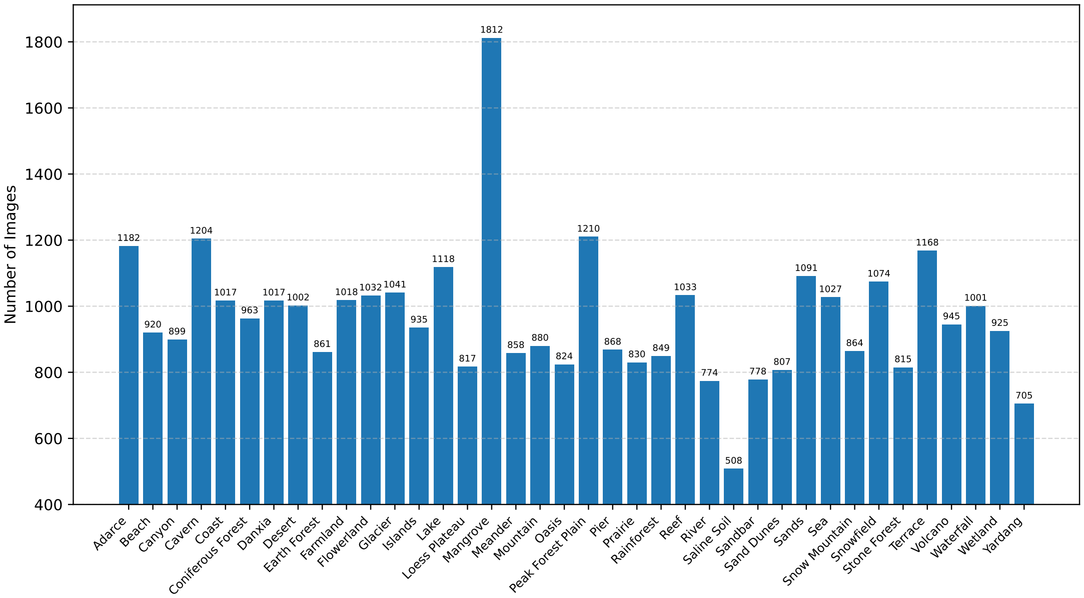

Fig. 3. The number of the images contained in each of the 38 classes of the Natural Terrain Scene Data Set (NTSD).

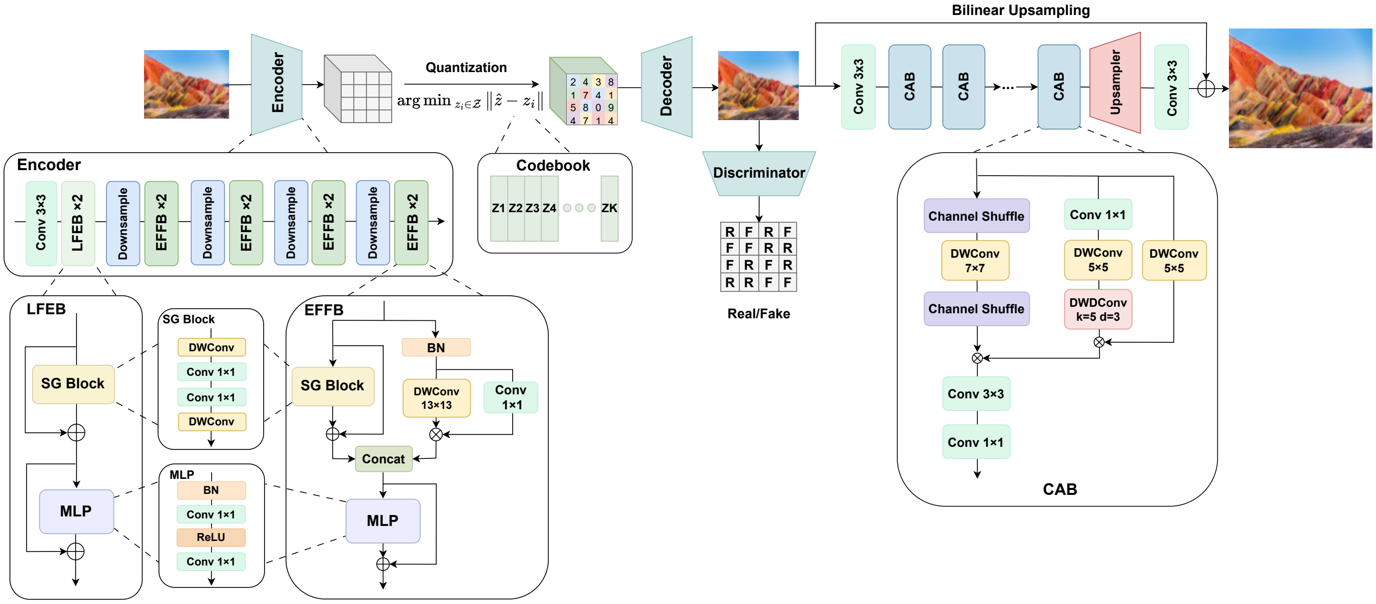

Fig. 4. The architecture of the proposed Lit-VQGAN, which consists of a lightweight VQGAN [11] built on top of two types of lightweight blocks, including the LFEB and the EFFB, and a lightweight super-resolution network, which is adopted using a set of Complex Attention Blocks (CABs).

MobileNetV2 [21] and the Fire Module of SqueezeNet [47], the LFEB is able to capture the more complex image characteristics. This is particularly important to image reconstruction.

C. *Efficient Feature Fusion Blocks (EFFBs)*

The core of the encoder-decoder network in the VQGAN [11] comprises a series of convolutional layers, which lack the representation of the global information. However, this information is useful for improving the quality of the images generated by the VQGAN [11]. To exploit both the global information and local characteristics, we introduce a Efficient Feature Fusion Block (EFFB). The left branch of the EFFB comprises an SG Block, which captures local characteristics. For the sake of capturing the global information, the right branch of the EFFB is adopted on top of a large kernel convolution and a 1×1 convolution, which is an attention unit in essence. Compared with the self-attention mechanism that Transforms utilize, both the computational complexity and the number of parameters of this branch are less.

Therefore, the utilization of the large kernel convolution not only enhances the computational efficiency but also captures the global information. Due to the trade-off between the efficiency and effectiveness, we use 13×13 convolutional kernels. The local and global features are fused using a concatenation operation. An MLP layer is further applied to the features fused for the purpose of modeling the relationship betwee different channels. The EFFB can be formulated as follows:

$$\text{Attention}(\tilde{x}^{l-1}) = \text{DConv}_{k\times k}(\tilde{x}^{l-1}) \times \text{Conv}_{1\times 1}(\tilde{x}^{l-1}), \quad (3)$$

$$\tilde{x}^{l} = \text{Concat}((SG(\tilde{x}^{l-1}) + \tilde{x}^{l-1}), \text{Attention}(\tilde{x}^{l-1})), \quad (4)$$

$$x^{l} = \text{MLP}(\tilde{x}^{l}) + \tilde{x}^{l}, \quad (5)$$

where $\tilde{x}^{l-1}$ denotes the feature maps produced by the $(l\text{-}1)$-th block, k is the size of the kernel, $Attention(\cdot)$ represents the right branch of the EFFB, $SG(\cdot)$ stands for the left branch of the EFFB and x^{l} is the feature maps generated by the MLP.

D. The Lightweight Super-Resolution Network

Regarding the image generation process, the VQGAN [11] first uses an AR model to perform the sampling operation in the discrete feature space mapped by the VQ module and then feeds the mapped feature maps to the decoder in order to generate an image. However, AR models normally encounter two problems. First, the sampling operation at each pixel location relies on the codes sampled at the prior pixel locations, which may lead to an accumulated error. This error will impair the generation process. Second, the AR-based sampling is conducted at each pixel location, which is time-consuming. Since the generation of a high-resolution image is performed in a moving-window style, in which the sampling and decoding operation is conducted in each window, this process is time-consuming and requires the large memory.

To address the above issues, we aim to perform the image generation operation using a sampling, decoding and super-resolution scheme instead of the moving-window sampling and decoding scheme. To this end, we first adopt a lightweight block and then build a super-resolution network on top of these blocks by referring to the ShuffleMixer [35]. In this case, a 256×256 image is first generated using the AR-based sampling and decoding operation. Then, the super-resolution network is used to upscale the image to the higher resolution. As a result, the high-resolution image generation can be conducted in the faster and more parameter-efficient manner.

As shown in Fig. 4, the image generated by the Lit-VQGAN is first fed into a 3×3 convolutional layer in order to extract the shallow-level features. Then these features are passed through a series of Complex Attention Blocks (CABs). The result is a set of deep-level feature maps. These maps are sent to an Upsampler and a 3×3 convolutional layer. The resultant image is added with the upsampled image of the generated image. As a result, the high-resolution image is derived.

The original ShuffleMixer [35] network lacks exploration of global features. However, both the local features and long-range dependencies are useful for the super-resolution task. Therefore, the CAB is adopted using two branches, which are used to extract the two types of information respectively. The left branch of the CAB contains a ShuffleMixer [35] layer which comprises a channel shuffling, a 7×7 depthwise convolution and a channel shuffling. This branch enables the extraction of informative features across different channels and the relatively large spatial area.

On the other hand, the decomposed large kernel convolution is utilized in the right branch, to capture long-range dependencies. Specifically, a 17×17 convolutional kernel is decomposed into a 1×1 pointwise convolution, a 5×5 depthwise convolution and a 5×5 depthwise dilation convolution with the dilation rate of 3. Compared with the large convolutional kernel, this decomposition greatly reduces both the number of parameters and the computational complexity. An attention style operation is carried out by performing an element-wise multiplication between the feature maps produced by a single 5×5 depthwise convolution and those generated using the decomposed large kernel convolution. The two sets of feature maps produced by the left and right branches are fused using a second attention computation. The fused feature maps are passed through a 3×3 convolution and a 1×1 convolution.

The CAB can be formulated as follows:

$$\tilde{x}_{L}^{l} = \text{Shuffle}(\text{DConv}_{7\times 7}(\text{Shuffle}(\tilde{x}^{l-1}))), \quad (6)$$

$$\tilde{x}_{K}^{l} = \text{DConv}_{5\times 5, d=3}(\text{DConv}_{5\times 5}(\text{Conv}_{1\times 1}(\tilde{x}^{l-1}))), \quad (7)$$

$$\tilde{x}_{R}^{l} = \text{DConv}_{5\times 5}(\tilde{x}^{l-1}) \times \tilde{x}_{K}^{l}, \quad (8)$$

$$x^{l} = \text{Conv}_{1\times 1}(\text{Conv}_{3\times 3}(\tilde{x}_{L}^{l} \times \tilde{x}_{R}^{l})), \quad (9)$$

where $\tilde{x}^{l-1}$ denotes the feature maps produced by the $(l\text{-}1)$-th block, $\tilde{x}_{L}^{l}$ denotes the output of the left branch, $\tilde{x}_{K}^{l}$ stands for the output of the large kernel convolution decomposition module, $\tilde{x}_{R}^{l}$ denotes the output of the right branch and x^{l} is the output of the CAB. In essence, the CAB is a lightweight block which extracts not only local features but also long-range dependencies.

In contrast to other attention modules, the CAB contains a unique dual-branch structure and a different attention mechanism. Specifically, one branch uses channel mixing operations to focus on the inter-channel information, while the other branch captures long-range dependencies through the decomposed large-kernel convolutions. However, existing attention modules, such as the SE module [48] and CBAM [49], typically generate attention maps using the single-branch global pooling or convolution operations. In addition, the CAB exploits a novel attention mechanism which performs the element-wise multiplication between the features produced by a depthwise convolution and those generated by the decomposed large-kernel convolution. This design enables the CAB to effectively integrate both the local and global information. As a result, the feature representation ability of the CAB is enhanced while the low computational complexity is obtained.

V. Experimental Setup

We will introduce the experimental setup in this section, including the baselines, data sets, evaluation metrics and implementation details.

A. Baselines

Regarding both the image reconstruction and generation tasks, we compared the proposed method with state-of-the-art approaches, including VQVAE-2 [8] and VQGAN [11].

Three additional lightweight VQGAN networks built using the blocks that MobileNetV2 [21], MobileNetV3 [22] and Next-ViT [29] utilized were compared with the proposed LiT-VQGAN for image reconstruction task. For the super-resolution task, our lightweight super-resolution network was compared against 14 state-of-the-art lightweight super-resolution networks, including SRCNN [50], FSRCNN [32], VDSR [51], LapSRN [52], DRCN [53], CARN [54], EDSR-baseline [55], FALSR-A [56], IMDN [33], LAPAR [57], ECBSR [58], SMSR [59], LBNet [60] and ShuffleMixer [35].

B. Data Sets

During the training processes of image reconstruction and image generation, our Natural Terrain Scene Data Set (NTSD) was utilized. We did not discriminate the class of images during the training process. Given a class, 2/3 of the images were randomly selected, which were used as the training images, while the remaining images were utilized as the testing images. As a result, the training set and the testing set contained 24,437 and 12,235 images respectively. The same data pre-processing was used as that utilized in [11]. We resized the training images to the resolution of 256×256 pixels. For the semantic image synthesis task, a semantic segmentation mask was obtained from each image using the DeepLabV2 [61] trained on the COCO-Stuff [62] data set.

In the super-resolution stage, we trained our network using the DF2K data set, which comprised the DIV2K [63] and Flickr2K [55] data sets. In total, 3,450 high-quality images were included. To derive low-resolution (LR) images, we followed the standard protocol in which high-resolution (HR) reference images were downsampled using bicubic interpolation. The model that we trained was assessed using five publicly available benchmark data sets, including Set5 [64], Set14 [65], B100 [66], Urban100 [67] and Manga109 [68].

C. Evaluation Metrics

Regarding the image reconstruction task, we computed the average reconstruction loss (Rec.) and the average perceptual loss (Perc.) across the testing set, to evaluate the performance of image reconstruction. For the image generation task, we used the Frechet Inception Distance (FID) to measure the quality of the image generated. For the purpose of assessing the performance of the super-resolution network, both the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) measures were employed. These measures were calculated using the Y channel in the YCbCr color space.

D. Implementation Details

The codebook in the VQ module contained 1,024 tokens. Either the encoder or the decoder of the Lit-VQGAN comprised 10 blocks. The network was trained using the Adam optimizer where β_1 and β_2 were set to 0.5 and 0.9 respectively. We set the initial learning rate to $4.5e$-6. The mini-batch size was set to 4. The Lit-VQGAN was trained for 200 epochs using two GeForce RTX 3090 GPUs.

We trained the super-resolution network using the RGB channels. Given a mini-batch, we randomly cropped 32 64×64 patches from all the LR images. These patches were augmented by applying the random horizontal flip and rotation operations. The patches were used as the input of the network. Our network was trained using the Adam optimizer for a total of $1{\times}10^6$ iterations. During the training process, we minimized both the L1 loss and the frequency loss. The learning rate was set to $5{\times}10^{-4}$ constantly. The super-resolution experiment was conducted on a GeForce RTX 3090 GPU.

TABLE I

Comparison of the number of parameters, FLOPs, and the FID, KID and IS values obtained using our network and five state-of-the-art networks for the unconditional image generation task. Here, ↓ indicates that the lower value corresponds to the higher image quality while ↑ indicates that the higher value corresponds to the higher image quality. In terms of each metric, the bold font indicates the best result.

Network	Params. (M)	FLOPs (G)	FID ↓	KID ↓	IS ↑
VQVAE-2 [8]	196.61	488.57	54.33	0.0298	3.13
BigGAN [69]	**149.43**	**342.53**	34.06	0.0140	3.93
VQGAN [11]	466.69	1320.51	30.83	0.0141	**7.21**
LDM [16]	353.40	1005.05	**18.50**	**0.0065**	5.03
Improved-Diffusion [18]	424.05	965.41	28.45	0.0090	6.09
Ours	374.13	804.55	24.36	0.0096	**7.21**

VI. Experimental Results

The experiment was conducted using the setup introduced in Section V along with the NTSD. We report the experimental results in this section.

A. Unconditional Natural Terrain Scene Generation

In this subsection, we report the results obtained using the proposed method for unconditional natural terrain scene generation. Both the quality and speed of the task are evaluated.

1) Quality: Given that the training settings of the VQGAN [11] was used, the unconditional image generation task was conducted to derive 256×256 images. As shown in Table I, our network used fewer parameters while achieving the better image generation quality, compared with the VQGAN [11]. In addition, we compared our method with VQ-VAE-2 [8], BigGAN [69] and two state-of-the-art diffusion models [16, 18]. The results are also reported in Table I. It can be seen that our approach achieves a proper trade-off between the accuracy and the complexity. We further show six sets of images generated using five state-of-the-art approaches and our method in Fig. 5. In contrast to the images produced by the baselines, the images generated using our method manifest the better, or at least the comparable, image quality.

2) Speed: Given that the NTSD were used for training both our method and the VQGAN [11] using the same settings, the time required in average is around 39 minutes and 49 minutes per epoch, respectively. In other words, our method could speed up the training process by about 20 percent. We also compared the time required for generating terrain scene images of different sizes using the VQGAN [11] and our method. As shown in Fig. 1, our method conducted image generation faster than the VQGAN, in particular, this was the case when the image was larger than 256×256 pixels.

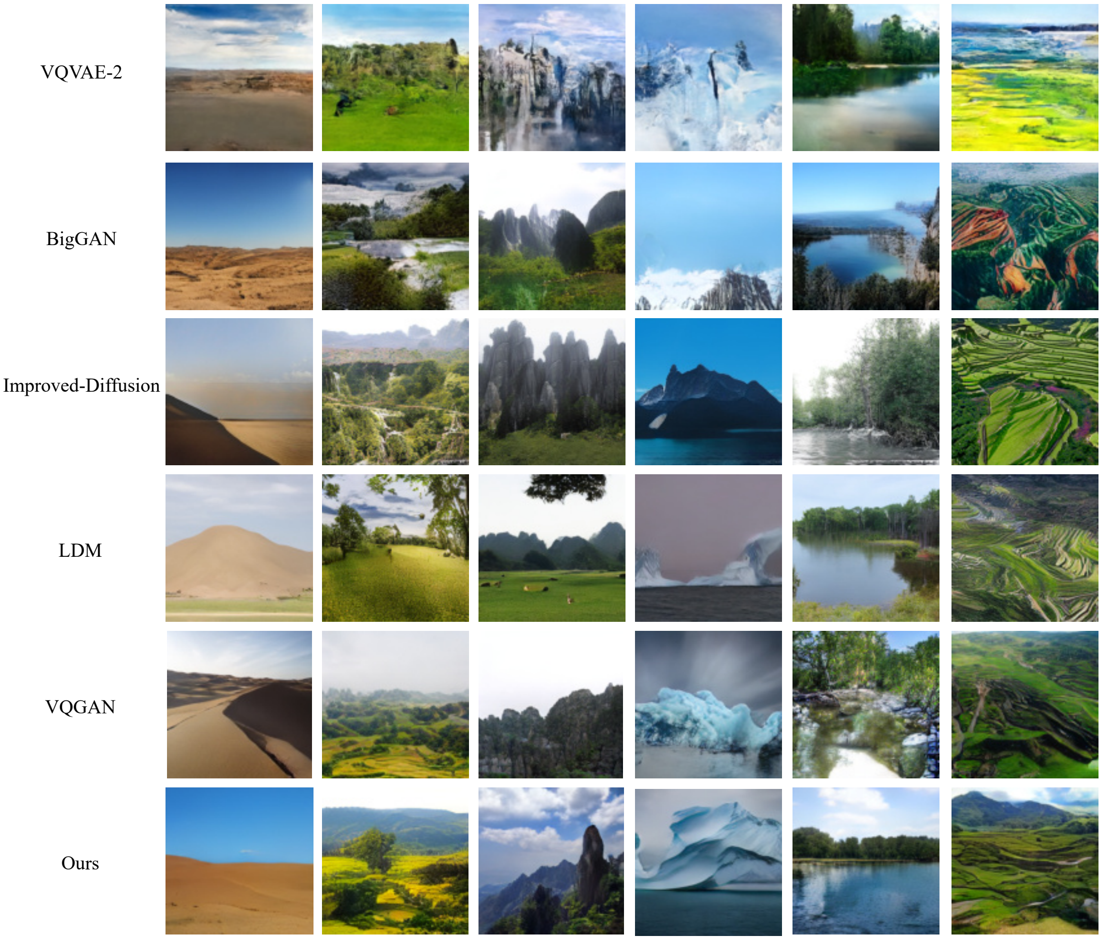

Fig. 5. Comparison of the images unconditionally generated using our method and five state-of-the-art approaches [8, 11, 16, 18, 69].

B. Semantic Image Synthesis

The semantic image synthesis experiment was performed in the case that a semantic segmentation map produced by the DeepLabV2 [61] was used as the conditional input. Intuitively, the segmentation map provides the semantic information for each pixel. As a result, the additional information was introduced which was useful for guiding the generation process. Fig. 6 presents eight sets of results. As can be seen, our method was able to generate realistic terrain scene images with the guidance of the semantic maps.

C. Super-Resolution

We compared our lightweight super-resolution network with 14 state-of-the-art baselines for the super-resolution task with three different upscaling factors, including ×2, ×3 and ×4, on five different benchmark data sets in Table II. In terms of the PSNR and SSIM measures, our method performed better than, or at least comparably to, the baselines across the five data sets. It should be noted that our method achieved a good trade-off between the number of parameters and the performance with a proper computational speed. In addition, five high-resolution terrain scene images derived by applying our super-solution network to the images that we generated using our image generation network are displayed in Fig. 7. As can be seen, these images present rich details and the high-fidelity.

D. Ablation Studies

To further understand and demonstrate the effectiveness of our Lit-VQGAN, we ablate it by evaluating the impact of each key component for the terrain scene image reconstruction task.

1) Impact of Lightweight Blocks: Three additional VQGAN [11] networks were built on top of the blocks used by three lightweight networks, including MobileNetV2 [21], MobileNetV3 [22] and NextViT [29], respectively. These networks were compared with our Lit-VQGAN for the image reconstruction task. Table III shows the number of parameters, FLOPs, and the reconstruction loss and perceptual loss calculated between the original and reconstructed images. It can be seen that the Lit-VQGAN was superior to its counterparts for the image reconstruction task. Our network used the fewer parameters and had the faster computational speed than the VQGAN networks crafted using the MobileNetV2 [21] and NextViT [29] blocks. These results suggest that the proposed Lit-VQGAN owns a good trade-off between the model size or efficiency and the effectiveness. Besides, Fig. 8 presents three sets of images reconstructed using the four VQGAN networks. As can be seen, the image reconstructed using our method shows the higher similarity to the ground-truth image according to the SSIM value, than the images reconstructed using the other networks.

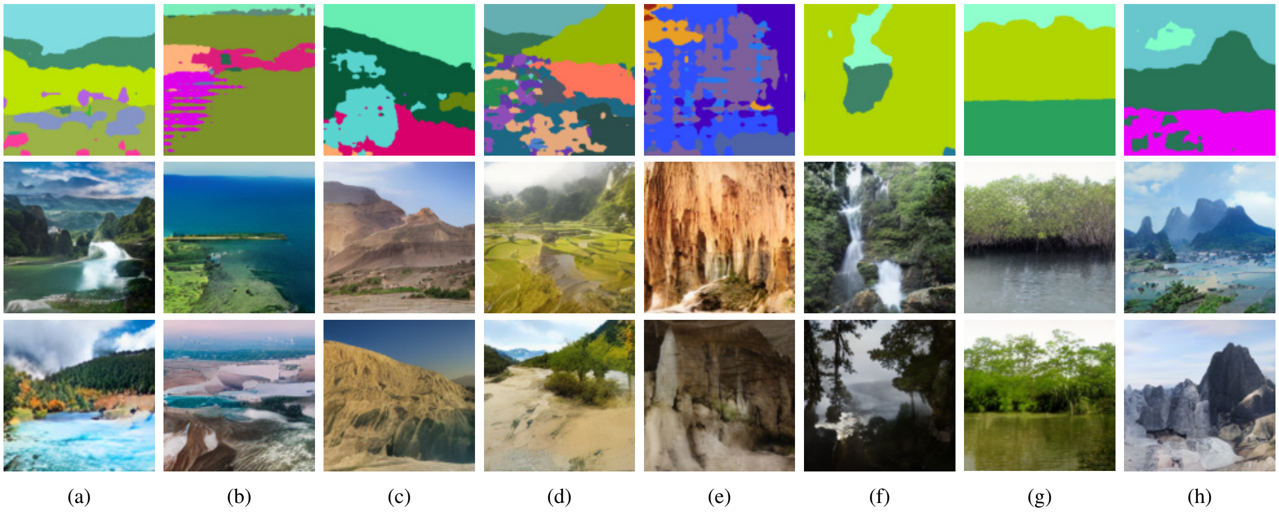

Fig. 6. Eight sets of results obtained in the semantic image synthesis experiment. In each column, a semantic segmentation map derived by applying the DeepLabV2 [61] model to an NTSD image (top) and two images generated using our method from the semantic map (middle and bottom) are shown in turn.

TABLE II

Comparison of our lightweight super-resolution network and 14 baselines on five benchmark data sets. Specifically, the number of parameters, FLOPs and the PSNR/SSIM values are reported. The FLOPs values were computed using a 1280×720 HR image. The PSNR/SSIM values were calculated using the Y channel. The best and second best PSNR/SSIM values are highlighted in the **Red** and *Blue* fonts respectively. The sign "-" indicates that the result was not reported in the original literature. Here, ↑ indicates that the higher value corresponds to the higher image quality.

Scale	Network	Params (M)	FLOPs (G)	PSNR ↑ / SSIM ↑				
				Set5	Set14	B100	Urban100	Manga109
×2	SRCNN [50]	0.057	53	36.66/0.9542	32.42/0.9063	31.36/0.8879	29.50/0.8946	35.74/0.9661
	FSRCNN [32]	0.012	6	37.00/0.9558	32.63/0.9088	31.53/0.8920	29.88/0.9020	36.67/0.9694
	VDSR [51]	0.665	613	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140	37.22/0.9729
	DRCN [53]	1.774	17,974	37.63/0.9588	33.04/0.9118	31.85/0.8942	30.75/0.9133	37.63/0.9723
	LapSRN [52]	0.813	30	37.52/0.9590	33.08/0.9130	31.80/0.8950	30.41/0.9100	37.27/0.9740
	CARN [54]	1.592	223	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256	-
	EDSR-baseline [55]	1.37	316	37.99/0.9604	33.57/0.9175	32.16/0.8994	31.98/0.9272	38.54/0.9769
	FALSR-A [56]	1.021	235	37.82/0.9595	33.55/0.9168	32.12/0.8987	31.93/0.9256	-
	IMDN [33]	0.694	161	38.00/0.9605	*33.63*/0.9177	*32.19*/0.8996	*32.17/0.9283*	**38.88**/*0.9774*
	LAPAR-A [57]	0.548	171	*38.01*/0.9605	33.62/*0.9183*	*32.19*/0.8999	32.10/*0.9283*	38.67/0.9772
	ECBSR-M16C64 [58]	0.596	137	37.90/**0.9615**	33.34/0.9178	32.10/**0.9018**	31.71/0.9250	-
	SMSR [59]	0.985	132	38.00/0.9601	**33.64**/0.9179	32.17/0.8990	**32.19/0.9284**	38.76/0.9771
	LBNet [60]	-	-	-	-	-	-	-
	ShuffleMixer [35]	0.394	91	*38.01/0.9606*	*33.63*/0.9180	32.17/0.8995	31.89/0.9257	*38.83/0.9774*
	Ours	0.381	88	**38.02**/*0.9606*	**33.64/0.9187**	**32.21**/*0.9001*	32.05/0.9281	*38.83*/**0.9777**
×3	SRCNN [50]	0.057	53	32.75/0.9090	29.28/0.8209	28.41/0.7863	26.24/0.7989	30.59/0.9107
	FSRCNN [32]	0.012	5	33.16/0.9140	29.43/0.8242	28.53/0.7910	26.43/0.8080	30.98/0.9212
	VDSR [51]	0.665	613	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279	32.01/0.9310
	DRCN [53]	1.774	17,974	33.82/0.9226	29.76/0.8311	28.80/0.7963	27.15/0.8276	32.31/0.9328
	LapSRN [52]	-	-	-	-	-	-	-
	CARN [54]	1.592	119	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493	-
	EDSR-baseline [55]	1.555	160	34.37/0.9270	30.28/0.8417	29.09/0.8052	28.15/0.8527	33.45/0.9439
	FALSR-A [56]	-	-	-	-	-	-	-
	IMDN [33]	0.703	72	34.36/0.9270	30.32/0.8417	29.09/0.8046	*28.17*/0.8519	33.61/0.9445
	ECBSR-M16C64 [58]	-	-	-	-	-	-	-
	LAPAR-A [57]	0.594	114	34.36/0.9267	*30.34*/0.8421	29.11/*0.8054*	28.15/0.8523	33.51/0.9441
	SMSR [59]	0.993	68	*34.40*/0.9270	30.33/0.8412	29.10/0.8050	**28.25/0.8536**	*33.68*/0.9445
	LBNet [60]	0.407	-	34.33/0.9264	30.25/0.8402	29.05/0.8042	28.06/0.8485	33.48/0.9433
	ShuffleMixer [35]	0.415	43	*34.40/0.9272*	**30.37**/*0.8423*	*29.12*/0.8051	28.08/0.8498	**33.69**/*0.9448*
	Ours	0.402	42	**34.42/0.9273**	**30.37/0.8431**	**29.13/0.8058**	28.16/*0.8529*	33.66/**0.9453**
×4	SRCNN [50]	0.057	53	30.48/0.8628	27.49/0.7503	26.90/0.7101	24.52/0.7221	27.66/0.8505
	FSRCNN [32]	0.012	5	30.71/0.8657	27.59/0.7535	26.98/0.7150	24.62/0.7280	27.90/0.8517
	VDSR [51]	0.665	613	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524	28.83/0.8809
	DRCN [53]	1.774	17,974	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510	28.98/0.8816
	LapSRN [52]	0.813	149	31.54/0.8850	28.19/0.7720	27.32/0.7280	25.21/0.7560	29.09/0.8845
	CARN [54]	1.592	91	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	-
	EDSR-baseline [55]	1.518	114	32.09/0.8938	28.58/0.7813	27.57/0.7357	26.04/0.7849	30.35/0.9067
	FALSR-A [56]	-	-	-	-	-	-	-
	IMDN [33]	0.715	41	*32.21*/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838	30.45/0.9075
	LAPAR-A [57]	0.659	94	32.15/0.8944	28.61/0.7818	*27.61*/0.7366	*26.14/0.7871*	30.42/0.9074
	ECBSR-M16C64	0.603	35	31.92/0.8946	28.34/0.7817	27.48/**0.7393**	25.81/0.7773	-
	SMSR [59]	1.006	42	32.12/0.8932	28.55/0.7808	27.55/0.7351	26.11/0.7868	30.54/0.9085
	LBNet [60]	0.41	-	32.08/0.8933	28.54/0.7802	27.54/0.7358	26.00/0.7819	30.37/0.9059
	ShuffleMixer [35]	0.411	28	*32.21/0.8953*	*28.66/0.7827*	*27.61*/0.7366	26.08/0.7835	**30.65**/*0.9090*
	Ours	0.398	27	**32.23/0.8957**	**28.69/0.7835**	**27.63**/*0.7376*	**26.16/0.7879**	*30.61*/**0.9101**

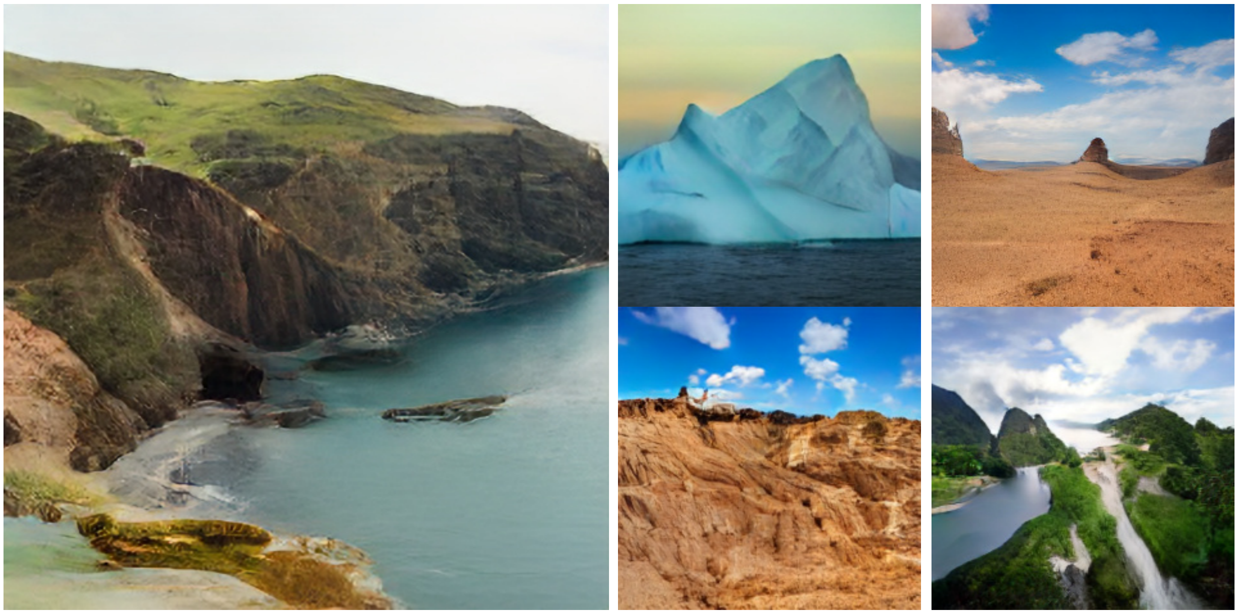

Fig. 7. Examples of the high-resolution images generated by our method. The image shown at the left part was generated at the resolution of 1024×1024 pixels while the four images displayed at the middle and right parts were generated at the resolution of 512×512 pixels.

TABLE III
COMPARISON OF OUR METHOD WITH THREE VQGAN [11] NETWORKS BUILT USING DIFFERENT LIGHTWEIGHT BLOCKS FOR THE NATURAL TERRAIN SCENE IMAGE RECONSTRUCTION TASK.

Network	Params. (M)	FLOPs (G)	Rec. ↓	Perc. ↓
VQGAN-MobileNetV2	45.56	71.07	0.588	0.429
VQGAN-MobileNetV3	**34.76**	**32.32**	0.571	0.423
VQGAN-Next-ViT	46.60	60.31	0.503	0.364
Ours	43.34	52.74	**0.472**	**0.342**

TABLE IV
COMPARISON OF DIFFERENT CONVOLUTIONAL MODULES.

Module	Params. (M)	FLOPs (G)	Rec. ↓	Perc. ↓
MobileNetV2 [21]	45.21	58.73	0.516	0.377
NCB [29]	44.15	59.08	0.507	0.363
FasterNet [70]	49.04	70.35	0.506	0.354
LFEB (Ours)	**43.34**	**52.74**	**0.472**	**0.342**

TABLE V
COMPARISON OF DIFFERENT ATTENTION BLOCKS.

Attention Block	Params. (M)	FLOPs (G)	Rec. ↓	Perc. ↓
SRA [28]	43.46	**51.43**	0.48	0.349
HA [71]	43.46	52.34	0.578	0.429
EFFB (Ours)	**43.34**	52.74	**0.472**	**0.342**

TABLE VI
COMPARISON OF DIFFERENT CONVOLUTIONAL KERNEL SIZES.

Kernel Size	Params. (M)	FLOPs (G)	Rec. ↓	Perc. ↓
11×11	**43.20**	**52.38**	0.492	0.36
13×13	43.34	52.74	**0.472**	**0.342**
15×15	43.50	53.18	0.478	0.348
17×17	43.68	53.65	0.493	0.359

2) Impact of the Local Feature Extraction Block: To verify the effectiveness of the proposed Local Feature Extraction Block (LFEB), we replaced it by three modules, including the bottleneck in the MobileNetV2 [21], the Next Convolution Block (NCB) in the Next-ViT [29] and the base module in the FasterNet [70], to build three networks. For the purpose of fair comparison, the experimental setting was kept the same. Table IV presents the number of parameters, FLOPs and two loss values computed for each network. As can be seen, our network which was built using the LFEB outperformed its three counterparts while this network utilized fewer parameters and run faster than the other networks.

3) Impact of the Efficient Feature Fusion Block: We also replaced the right branch of the proposed Efficient Feature Fusion Block (EFFB), which was an attention module in essence, by two attention modules, including the Spatial-Reduction Attention (SRA) [28] and the Hydra Attention (HA) [71] modules, to build two networks. Given that the same experimental settings were used, the comparison is shown in Table V. It can be observed that the network built using the EFFB performed the better than its two counterparts. However, our network used fewer parameters. Furthermore, we examined the effect of the size of the convolutional kernel used in the EFFB. As shown in Table VI, the 13×13 kernel achieved a proper trade-off between the model size and performance.

4) Impact of the Complex Attention Block: Regarding the ShuffleMixer [35], we adopted a variant by removing a ShuffleMixer layer from each block. This variant is referred to as ShuffleMixer−. The two parts in the right branch of the Complex Attention Block (CAB), i.e., the decomposed 17×17 convolution and the 3×3 depthwise convolution, were added into the ShuffleMixer− in turn. As a result, two additional variants were derived. The first is referred to as +Conv$_{17\times17}$ and the second is the lightweight super-resolution network that we adopted. The experiment was performed on the Urban100

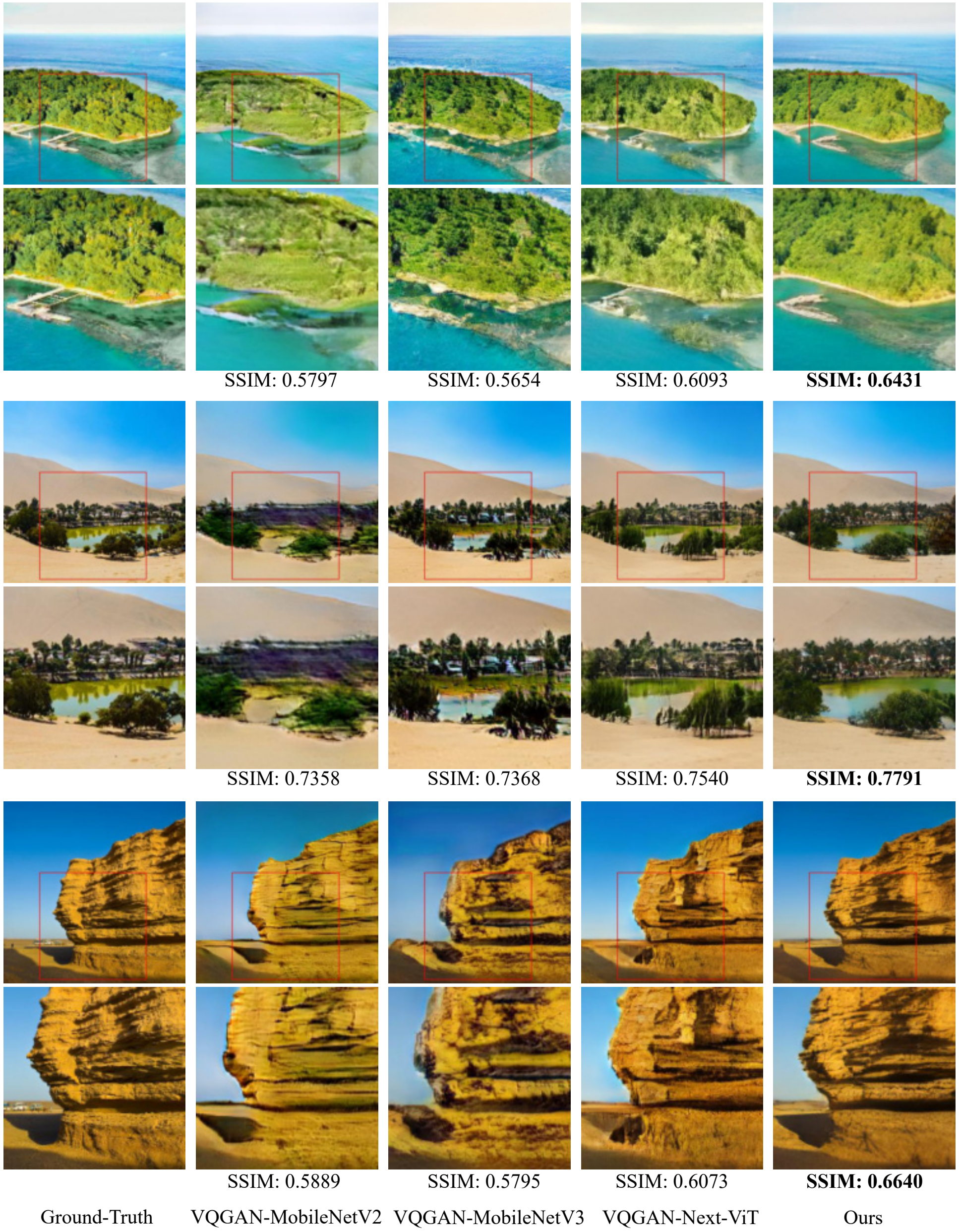

Fig. 8. The natural terrain scene image reconstruction results derived using three VQGAN [11] networks which were built using the blocks utilized by three different lightweight networks, including MobileNetV2 [21], MobileNetV3 [22] and NextViT [29], respectively, and our Lit-VQGAN. Below each ground-truth or reconstructed image, a magnified image of the sub-region in this image and the SSIM value computed between this image and the ground-truth image are displayed in turn.

TABLE VII
THE EFFECT OF THE COMPONENTS OF OUR LIGHTWEIGHT SUPER-RESOLUTION NETWORK.

Network	Params. (M)	FLOPs (G)	PSNR/SSIM
ShuffleMixer−	**0.352**	**24.42**	26.03/0.7833
+Conv$_{17\times17}$	0.389	26.52	26.09/0.7855
Ours	0.397	26.98	**26.16/0.7879**

[67] data set with the upscaling factor of ×4. The FLOPs was computed using the 1280×720 HR image. As reported in Table VII, our method produced the best super-resolution result with a slight sacrifice of the model size and computational speed.

VII. Conclusion

Since it is difficult to collect sufficient natural terrain scene images, we aimed to overcome this challenge using image generation techniques. To this end, we first collected a Natural Terrain Scene Data Set (NTSD), which contains 36,672 images divided into 38 classes. This data set can be used to train and test image generation networks. Although state-of-the-art image generation networks can be used for synthesizing terrain scene images, high space complexity and heavy computational demand are usually confronted. To address these issues, we then proposed a Lightweight Vector Quantized Generative Adversarial Network (Lit-VQGAN). This network was built on top of two types of lightweight blocks. As a result, the parameters of the Lit-VQGAN were greatly reduced. In addition, a lightweight super-resolution network was adopted using Complex Attention Blocks (CABs). Compared with the moving-window sampling and decoding scheme that the VQGAN used, this network was able to perform the high-resolution image generation task more efficiently. To our knowledge, none of the NTSD and the Lit-VQGAN had been exploited before. Our results demonstrated that the Lit-VQGAN conducted the image generation task more efficiently and effectively, in contrast to the VQGAN. We believe that the promising results should be due to the lightweight but effective blocks that we deliberately designed.

References

[1] Z. Wang, M. Jiang, and J. Wang, "Phaed: A speaker-aware parallel hierarchical attentive encoder-decoder model for multi-turn dialogue generation," *IEEE Transactions on Big Data*, 2023.

[2] I. Zinno, M. Bonano, S. Buonanno, F. Casu, C. De Luca, M. Manunta, M. Manzo, and R. Lanari, "National scale surface deformation time series generation through advanced dinsar processing of sentinel-1 data within a cloud computing environment," *IEEE Transactions on Big Data*, vol. 6, no. 3, pp. 558–571, 2018.

[3] C. Zhuo, D. Gao, and L. Liu, "Pkdgan: Private knowledge distillation with generative adversarial networks," *IEEE Transactions on Big Data*, 2022.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Neural Information Processing Systems*, 2014.

[5] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[6] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv.org*, 2014.

[7] A. Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," 2017.

[8] A. Razavi, A. Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," 2019.

[9] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *International conference on machine learning*. PMLR, 2018, pp. 4055–4064.

[10] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *International conference on machine learning*. PMLR, 2016, pp. 1747–1756.

[11] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," 2020.

[12] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," 2014.

[13] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, and B. Guo, "Vector quantized diffusion model for text-to-image synthesis," *arXiv preprint arXiv:2111.14822*, 2021.

[14] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[15] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[16] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[17] W. Peebles and S. Xie, "Scalable diffusion models with transformers," *arXiv preprint arXiv:2212.09748*, 2022.

[18] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International conference on machine learning*. PMLR, 2021, pp. 8162–8171.

[19] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, "Maskgit: Masked generative image transformer," 2022.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[21] A. Howard, A. Zhmoginov, L.-C. Chen, M. Sandler, and M. Zhu, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," 2018.

[22] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[23] D. Zhou, Q. Hou, Y. Chen, J. Feng, and S. Yan, "Rethinking bottleneck structure for efficient mobile network design," *ECCV, August*, 2020.

[24] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.

[25] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[26] W. Li, X. Wang, X. Xia, J. Wu, X. Xiao, M. Zheng, and S. Wen, "Sepvit: Separable vision transformer," *arXiv preprint arXiv:2203.15380*, 2022.

[27] T. Huang, L. Huang, S. You, F. Wang, C. Qian, and C. Xu, "Lightvit: Towards light-weight convolution-free vision transformers," *arXiv preprint arXiv:2207.05557*, 2022.

[28] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[29] J. Li, X. Xia, W. Li, H. Li, X. Wang, X. Xiao, R. Wang, M. Zheng, and X. Pan, "Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios," *arXiv preprint arXiv:2207.05501*, 2022.

[30] Y. Chen, X. Dai, D. Chen, M. Liu, X. Dong, L. Yuan, and Z. Liu, "Mobile-former: Bridging mobilenet and transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5270–5279.

[31] Y. Li, G. Yuan, Y. Wen, J. Hu, G. Evangelidis, S. Tulyakov, Y. Wang, and J. Ren, "Efficientformer: Vision transformers at mobilenet speed," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12 934–12 949, 2022.

[32] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 391–407.

[33] Z. Hui, X. Gao, Y. Yang, and X. Wang, "Lightweight image super-resolution with information multi-distillation network," in *Proceedings of the 27th acm international conference on multimedia*, 2019, pp. 2024–2032.

[34] A. Muqeet, J. Hwang, S. Yang, J. H. Kang, Y. Kim, and S.-H. Bae, "Ultra lightweight image super-resolution with multi-attention layers," *arXiv preprint arXiv:2008.12912*, vol. 2, no. 5, 2020.

[35] L. Sun, J. Pan, and J. Tang, "Shufflemixer: An efficient convnet for image super-resolution," *arXiv preprint arXiv:2205.15175*, 2022.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[37] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 976–11 986.

[38] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31x31: Revisiting large kernel design in cnns," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 963–11 975.

[39] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, "Visual attention network," *arXiv preprint arXiv:2202.09741*, 2022.

[40] X. Dong and J. Dong, "Oceanic scene recognition using graph-of-words (gow)," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[41] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[42] "Unsplash," https://unsplash.com/.

[43] "Pixabay," https://pixabay.com/.

[44] "Pexels," https://www.pexels.com/zh-cn/.

[45] "Flickr," https://www.flickr.com/.

[46] "Google," https://www.google.com/.

[47] A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, and K. Keutzer, "Squeezenext: Hardware-aware neural network design," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1638–1647.

[48] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[49] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “Cbam: Convolutional block attention module,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[50] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[51] J. Kim, J. K. Lee, and K. M. Lee, “Accurate image super-resolution using very deep convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654.

[52] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, “Deep laplacian pyramid networks for fast and accurate super-resolution,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 624–632.

[53] J. Kim, J. K. Lee, and K. M. Lee, “Deeply-recursive convolutional network for image super-resolution,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1637–1645.

[54] N. Ahn, B. Kang, and K.-A. Sohn, “Fast, accurate, and lightweight super-resolution with cascading residual network,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 252–268.

[55] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, “Enhanced deep residual networks for single image super-resolution,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.

[56] X. Chu, B. Zhang, H. Ma, R. Xu, and Q. Li, “Fast, accurate and lightweight super-resolution with neural architecture search,” in *2020 25th International conference on pattern recognition (ICPR)*. IEEE, 2021, pp. 59–64.

[57] W. Li, K. Zhou, L. Qi, N. Jiang, J. Lu, and J. Jia, “Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 343–20 355, 2020.

[58] X. Zhang, H. Zeng, and L. Zhang, “Edge-oriented convolution block for real-time super resolution on mobile devices,” in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4034–4043.

[59] L. Wang, X. Dong, Y. Wang, X. Ying, Z. Lin, W. An, and Y. Guo, “Exploring sparsity in image super-resolution for efficient inference,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 4917–4926.

[60] G. Gao, Z. Wang, J. Li, W. Li, Y. Yu, and T. Zeng, “Lightweight bimodal network for single-image super-resolution via symmetric cnn and recursive transformer,” *arXiv preprint arXiv:2204.13286*, 2022.

[61] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[62] H. Caesar, J. Uijlings, and V. Ferrari, “Coco-stuff: Thing and stuff classes in context,” in *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018.

[63] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, “Ntire 2017 challenge on single image super-resolution: Methods and results,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 114–125.

[64] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, “Low-complexity single-image super-resolution based on nonnegative neighbor embedding,” 2012.

[65] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” in *Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7*. Springer, 2012, pp. 711–730.

[66] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2010.

[67] J.-B. Huang, A. Singh, and N. Ahuja, “Single image super-resolution from transformed self-exemplars,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5197–5206.

[68] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, “Sketch-based manga retrieval using manga109 dataset,” *Multimedia Tools and Applications*, vol. 76, pp. 21 811–21 838, 2017.

[69] A. Brock, J. Donahue, and K. Simonyan, “Large scale gan training for high fidelity natural image synthesis,” *arXiv preprint arXiv:1809.11096*, 2018.

[70] J. Chen, S.-h. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H. G. Chan, “Run, don’t walk: Chasing higher flops for faster neural networks,” *arXiv preprint arXiv:2303.03667*, 2023.

[71] D. Bolya, C.-Y. Fu, X. Dai, P. Zhang, and J. Hoffman, “Hydra attention: Efficient attention with many heads,” in *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*. Springer, 2023, pp. 35–49.

Yan Wang received the bachelor’s degree in Computer Science and Technology from Jining Medical College, Shandong Province, China in 2021. She is currently a post-graduate student at Ocean University of China working toward her master’s degree in Computer Science. Her research interests include computer vision, deep learning, image generation, and image super-resolution.

Huiyu Zhou received the B.Eng. degree in radio technology from the Huazhong University of Science and Technology, Wuhan, China, in 1990, the M.Sc. degree in biomedical engineering from the University of Dundee, Dundee, U.K., in 2002, and the Ph.D. degree in computer vision from Heriot-Watt University, Edinburgh, U.K., in 2006. He is currently a Full Professor with the School of Computing and Mathematical Sciences, University of Leicester, Leicester, U.K. His research interests include medical image processing, computer vision, intelligent systems, and data mining.

Xinghui Dong received the PhD degree from Heriot-Watt University, U.K., in 2014. He worked with the Centre for Imaging Sciences, the University of Manchester, U.K., between 2015 and 2021. Then he jointed Ocean University of China in 2021. He is currently a professor at the Ocean University of China. His research interests include computer vision, defect detection, texture analysis, and visual perception.