MaskGIT for Single Image Super-Resolution

Yifan Yang, Erik Huang, Sharon Xu

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

YIFAN ERIK SHARON .ai

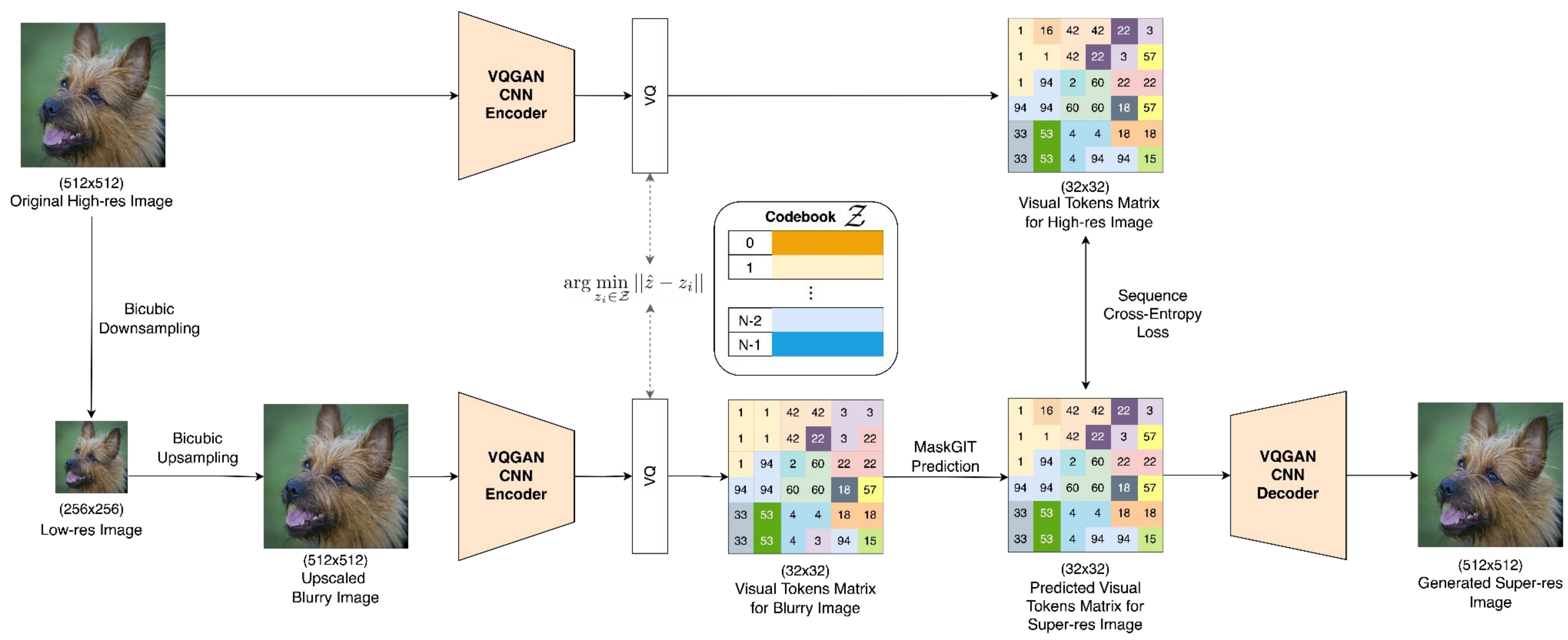
Data, Models, and Code: https://github.com/YES-ai/super-resolution

Introduction and methods

In the paper MaskGIT: Masked Generative Image Transformer, Chang et al. proposed a novel image synthesis paradigm using a bidirectional transformer decoder combined with the VQGAN architecture. Leveraging MaskGIT's strength in learning high-quality intermediary representation and image inpainting, we finetuned the MaskGIT model in the task of Simple Image Super Resolution.
Key contributions:

- Two novel methods to adapt MaskGIT with VQVAE in the task of single image super-resolution
- A novel decoding process along with the masking strategy which proved to be effective in generating super-resolution images

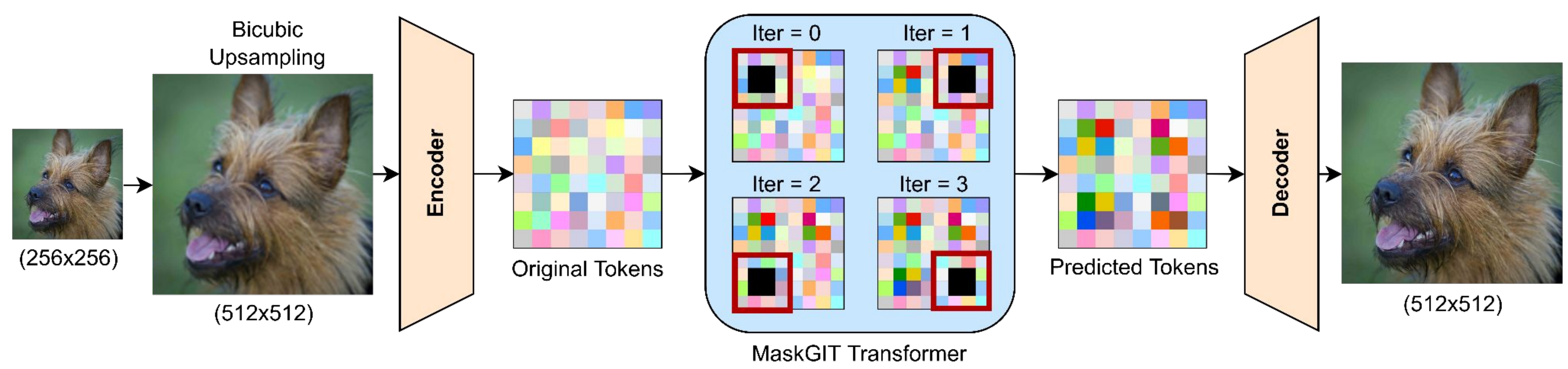
Image Preprocessing
The image preprocessing is the same for both methods. Given a low-res 256×256 input image, we first rescale it into size of 512×512. VQVAE will compress the image into by a fixed factor of 16, i.e. provides us with 32×32 input tokens matrix with codebook size = 1024.



Method 1: Token Correction in Latent Space

In method 1, we train the transformer decoder to predict visual tokens of the high-res image from that of the low-res image. We froze the weights of VQVAE and trained the transformer alone. The training objective is to minimize the negative log likelihood of the high-res image visual tokens, which is computed as the cross-entropy between the ground-truth one-hot token and predicted token. To decode visual tokens, we adapted a simple one-step prediction scheme using argmax. Lastly the token matrix is decoded using VQVAE to generate the super-res image.

In method 2, we used masking to perform conditional generation. For the task with 256×256 images, the input token matrix is uniformly divided into 8×8 grids. Within each grid, we mask out the center 4×4 tokens and make MaskGIT predict those masked tokens. This step is very similar to Image Inpainting, except that we are iteratively performing this step for the entire token matrix. Lastly, the predicted token matrix is decoded into the output image.



Method 2: Inpainting Based Super Resolution

Results

Dataset	Method	SR-MaskGIT Token Correction	CAR Sun et al.
DIV2K 2xUpscaling	PSNR	19.82	**38.26** (SOTA)
Agustsson et al.	SSIM	0.2021	**0.9599** (SOTA)

Here are example outputs for our two proposed methods. The leftmost image is the low-res 256x256 input image, the second image is the output of our Token Correction method, the third is the output of our Image Inpainting method, and the rightmost image is the original high-res 512x512 output image.

We have also performed quantitative analysis by evaluating our methods on the DIV2K 2xUpscaling Super-resolution task and measure the PSNR and SSIM metrics.

Observations

We proposed two approaches to demonstrate MaskGIT's potential in super-resolution task with rich details. However, we are also crippled by the intrinsic over-generalizing property of VQVAE, which we found to be:

- The original pretrained VQVAE model struggles with perfect image reconstruction and we observed poor SSIM & PSNR scores.
- Increasing the codebook size of VQVAE (e.g., from 1024 to 16384) does not necessarily yield better image embedding quality in terms of SSIM & PSNR metrics. However, larger codebook size tends to yield more visually pleasing and natural images.
- Our masking strategy is rather aggressive since we introduced some degree of manual interference.

We believe future work can be done to potentially improve our methods for better super-resolution results, with the most significant alternative being pretraining MaskGIT and VQVAE with a much larger codebook size (e.g., 16384 or 32768) on a large dataset such as ImageNet.