## TransGAN for Image

## Super-Resolution: A Case Study

Introduction to Deep Learning

Group 6 – Class 1

—

Nguyễn Thành Long - 22BI13264

Vũ Tuấn Minh - 22BI13317

Nguyễn Quang Minh - 22BI13301

Phan Nguyễn Tuấn Minh - 22BI13307

Hoàng Long - 22BI13259

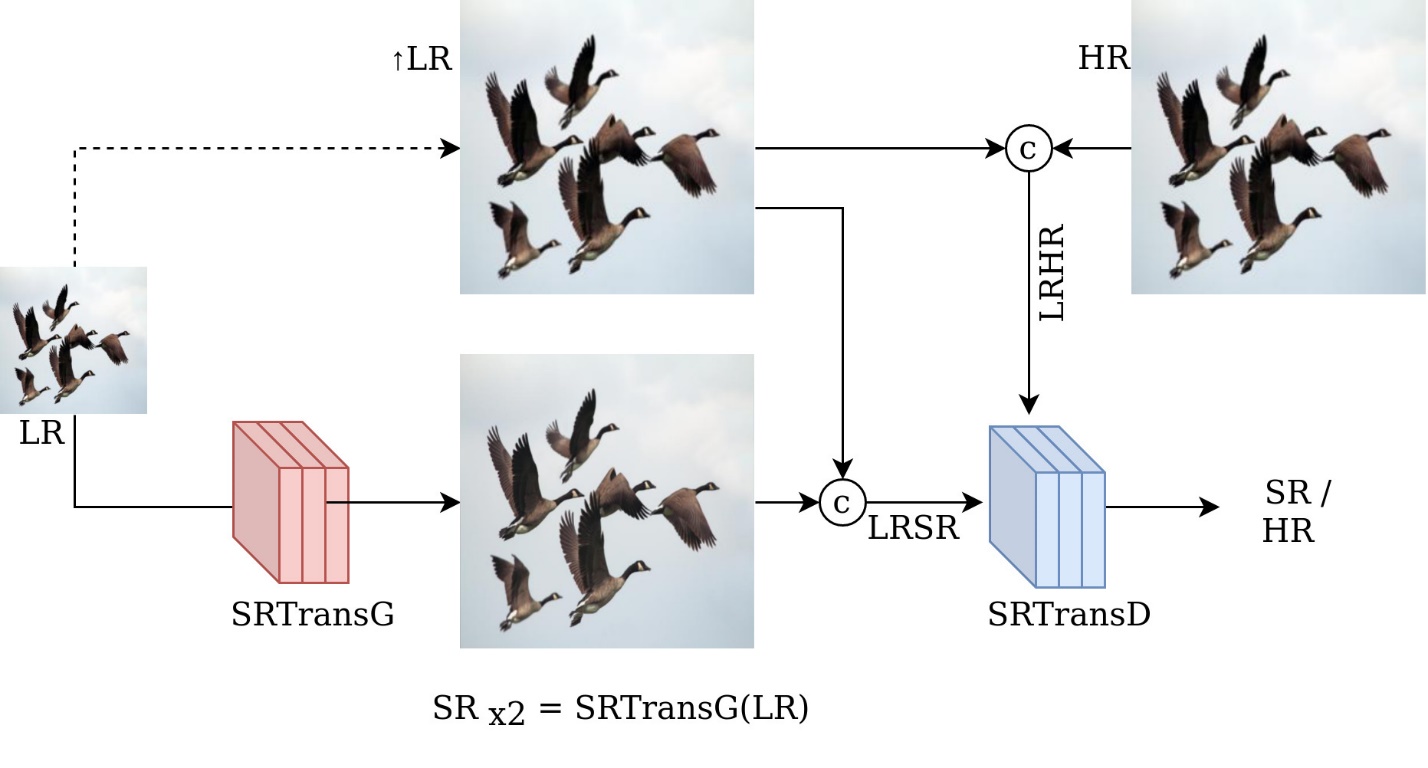
Nguyễn Hà My - 22BI13318

—

31/9/2024

ABSTRACT

Image Super-Resolution (SR), a machine learning task aims to enhance the resolution of low-resolution images by reconstructing high-frequency details. In recent years, Generative Adversarial Networks (GANs) and Transformer architectures were developed and have proven its effective in improving SR performance. This case study explores how a Transformer-based GAN can be applied to super-resolution tasks and presents the implementations of TransGAN.



*Figure 1: Proposed SRTransGAN framework for image super-resolution*.

SETUP

## Dataset and Preprocessing

- Dataset: DIV2K

- Images are downsampled, producing lower-resolution one.

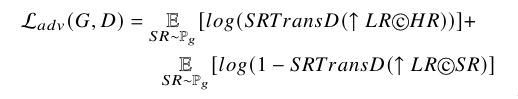
- The pixel values of the images are normalized to a range suitable for deep learning models.

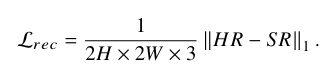
## Loss Function

To train the SRTransGAN model, a combination of two loss functions is employed: Adversarial loss and Reconstruction loss. Additionally, the Adversarial loss itself consists of both Generator loss and Discriminator loss. The overall Generator loss can be calculated by:



Where as is the Adversarial loss, is the Generator loss, and is the weight coefficients for Adversarial and Reconstruction loss, respectively.



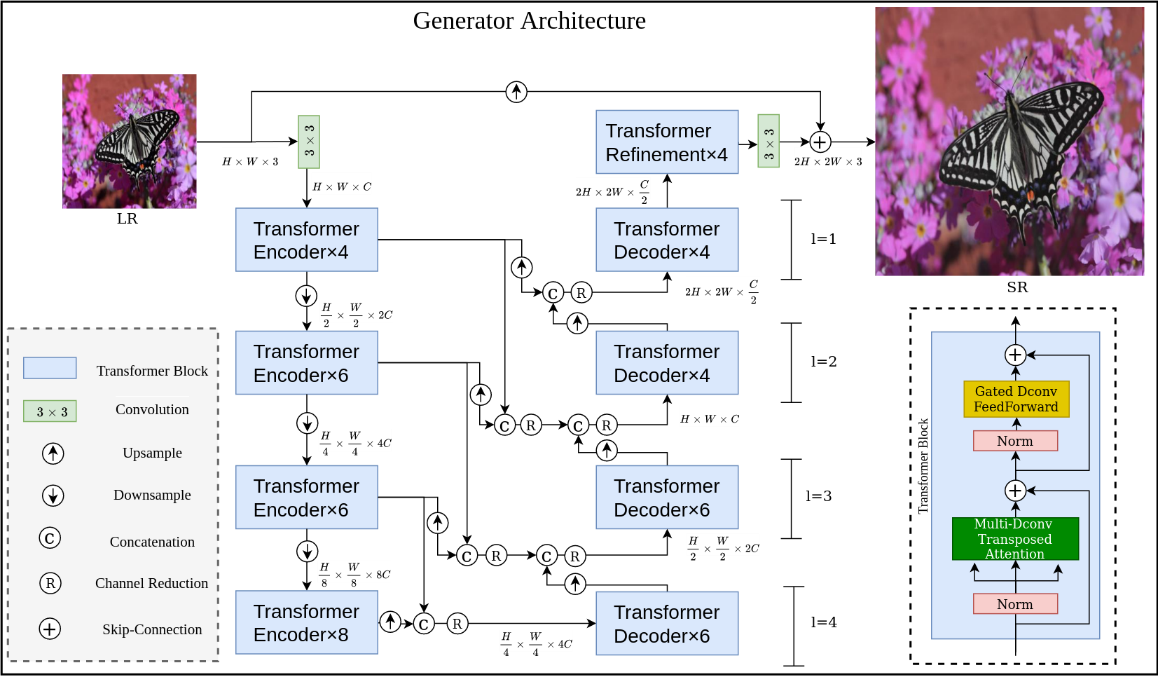


METHODOLOGY

## Model Architecture

**Generator (SRTransGAN):**

The proposed SRTransGAN uses a Transformer-based SRTransG generator network to transform low-resolution (LR) images into super-resolution (SR) images. SRTransG generates SR images progressively, using Transformer Encoder modules for downsampling and Transformer Decoder modules for upsampling. These modules effectively capture both global and local relationships, crucial for generating high-resolution images. Initially, SR images are produced at a 2x scale, and to achieve a 4x scale, the SRTransG network is applied twice with the same parameters, progressively enhancing the resolution.



*Figure 2:Proposed Super-Resolution Transformer Generator (SRTransG) Network used in SRTransGAN.*

**Input Image**

- The architecture starts with a low-resolution input image (), meaning height, width, and 3 color channels: RGB).

- This image is first processed by a 3 × 3 convolutional layer, which extracts initial features from the image and increases the number of channels ().

**Transformer Encoders**

- The architecture consists of multiple *Transformer Encoder Blocks*, each of which processes the image at different resolutions and depths.

- As the network goes deeper, it increases the number of channels, which allows it to extract more complex features

- *Downscaling* occurs at different points in the architecture, where the feature map is progressively reduced in spatial dimensions (height and width) but increased in the number of channels. This helps the model learn multi-scale features.

**Skip Connections**

- They are represented by the arrows moving across different levels in the network. These connections allow the network to retain important information from earlier layers and pass it to later stages without losing detail, a common technique in encoder-decoder architectures.

**Transformer Decoders**

- After downscaling the image features, the architecture uses a series of *Transformer Decoder Blocks* to reconstruct the image at higher resolutions.

- Each decoding step upsamples the feature map, increasing the spatial resolution back to the original dimensions.

**Upsampling and Channel Reduction**

- Upsampling happens progressively as the decoders increase the spatial resolution.

- At each stage, the number of channels is reduced, eventually returning to the 3 channels for the RGB image.

**Concatenation (C)**

- At various stages, *concatenation (C)* merges feature maps from the encoder and decoder paths, providing the decoder with detailed information from the corresponding encoder blocks.

**Transformer Refinement**

- At the end of the decoding process, a *Transformer Refinement Block* is applied. This is an additional block that refines the final upscaled image further, ensuring high-quality reconstruction.

**Final Convolution**

- The final step involves applying another 3 × 3 convolution to transform the upscaled feature map back to the original 3-channel high-resolution image ().

**Transformer Block Details**

- *Multi-Dconv Transposed Attention:* This block applies multi-head attention using transposed convolutions (deconvolutions), which helps upsample the feature map while attending to important regions.

- *Norm (Normalization):* Standard normalization layers are applied for stable training.

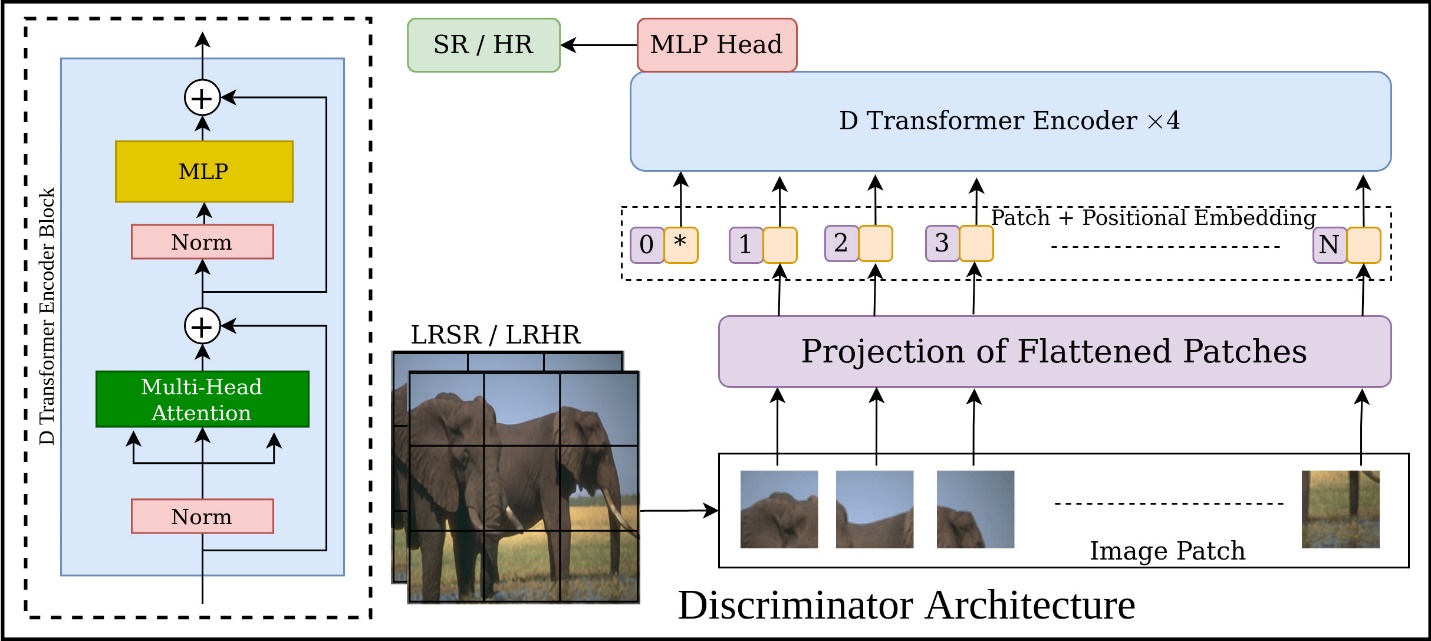
- *Gated Dconv FeedForward*: After attention, a Gated Deconvolutional Feed-Forward Network is applied, which helps to further refine the features.

**Output Image**

- The final output is a high-resolution image, which is likely ()in size, as the architecture performs image super-resolution.

**Discriminator:**

The discriminator's role is to distinguish between real high-resolution images and the super-resolution images generated by the SRTransG Generator By forcing the generator to improve over time, the discriminator enhances the visual quality of the generated SR images. It operates by evaluating both the pixel-level and structural fidelity, helping ensure that the final images look natural and high-quality.



*Figure 3:The Discriminator Network (SRTransD) used in SRTransGAN model. It is based on the Vision Transformer.*

**Input (LRSR/LRHR)**

- LRSR (Low-Resolution Super-Resolved) and LRHR (Low-Resolution High-Resolution) images are fed into the Discriminator. These are either the real high-resolution images or the generated super-resolved images.

- The images are divided into patches, which are smaller sections of the image, similar to the approach taken in vision transformer (ViT) models.

**Projection of Flattened Patches**

- Once the image is split into patches, each patch is flattened (converted into a 1D vector), where 2D image patches are treated like tokens in Natural Language Processing (NLP) models.

- These flattened patches are then projected into a higher-dimensional space, creating an initial embedding for each patch.

**Patch & Positional Embedding**

- After projection, positional embeddings are added to each patch. Since *Transformers* don’t inherently understand the spatial structure of images (unlike CNNs), positional embeddings are used to inform the model about the relative position of each patch in the original image.

**D Transformer Encoder × 4**

- The patches, along with their positional embeddings, are passed through multiple layers of Transformer encoders (in this case, 4 layers).

- *Multi-Head Attention*: This mechanism allows the model to focus on different parts of the image simultaneously, capturing long-range dependencies between different patches.

- *Normalization (Norm)*: This layer normalizes the inputs, helps stabilize training.

- *MLP (Multi-Layer Perceptron):* After attention, the data is passed through a fully connected neural network (MLP) to process and extract more complex patterns.

**MLP Head**

- After processing through the Transformer encoder blocks, the final feature representations of the patches are passed to an *MLP head*, which aggregates the information across patches.

- The MLP head outputs a probability indicating whether the input image (SR/HR) is “real” (high-resolution from the dataset) or “fake” (super-resolved by the Generator).

**SR/HR Output**

- The final output is a classification label (real or fake) for the input image. This is a binary classification task: if the image is super-resolved (SR) or high-resolution (HR), it helps the Discriminator decide whether the image is real or generated.

RESULTS







References

Z. Wang, J. Chen, and S. C. Hoi, “Deep learning for image super-resolution: A survey,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3365–3387, 2020.

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.

Neeraj Baghel, Shiv Ram Dubey, Satish Kumar Singh, “SRTransGAN: Image Super-Resolution using Transformer based Generative Adversarial Network”, 2023

K. Zhang, W. Zuo, and L. Zhang, “Learning a single convolutional super-resolution network for multiple degradations,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3262–3271.