SRGAN
Prof. Sudipan Saha

Tushar Verma (2022AIY7514)
ScAI, IIT Delhi

1 Introduction

Super Resolution Generative Adversarial Networks (SRGAN) represent a significant advancement in the field of image super-resolution. Developed by Ledig et al. in 2016, SRGAN employs a generative adversarial network (GAN) architecture to enhance the resolution of images while preserving texture and details. Traditional upsampling methods, such as bicubic interpolation, often lead to blurred and unrealistic images, failing to capture the finer details that are crucial in many applications. SRGAN overcomes these limitations by leveraging the adversarial training process, where two networks—the generator and the discriminator—compete against each other.

The generator aims to create high-resolution images from low-resolution inputs, while the discriminator evaluates the authenticity of the generated images, distinguishing between real high-resolution images and those produced by the generator. This adversarial process enables the generator to improve its performance iteratively, resulting in visually appealing super-resolved images that retain essential details and textures. Application of SRGAN for Upsampling Satellite Images

2 Applications of SRGAN in Remote Sensing

In this mini project, I have trained an SRGAN model to upsample satellite images from zoom level 17 to a resolution four times greater. Satellite imagery plays a crucial role in various applications, including urban planning, environmental monitoring, and disaster management. However, the inherent challenges associated with low-resolution images can impede detailed analysis and decision-making.

By utilizing SRGAN, I aimed to enhance the spatial resolution of satellite images while maintaining the quality and authenticity of the original content. The training process involved feeding the model pairs of low-resolution and corresponding high-resolution satellite images, allowing it to learn the intricate patterns and features specific to satellite imagery.

The resulting model effectively generates high-resolution satellite images that reveal finer details, making it easier to interpret land use, infrastructure, and other critical features. This enhancement can significantly improve the utility of satellite data in various applications, offering more accurate insights and facilitating better-informed decisions in geographical and environmental contexts.

3 Model Architecture

Table 1: Architecture of the Generator and Discriminator

Layer	Type	Input Channels	Output Channels	Kernel Size	Stride
		Generator			
1	Conv Layer	3	64	9x9	1
2	Residual Block	64	64	3x3	1
3	Residual Block	64	64	3x3	1
4	Residual Block	64	64	3x3	1
5	Residual Block	64	64	3x3	1
6	Residual Block	64	64	3x3	1
7	Conv Layer	64	64	3x3	1
8	Upsample Block	64	64	3x3	1
9	Upsample Block	64	64	3x3	1
10	Conv Layer	64	3	9x9	1
		Discriminator			
11	Conv Layer	3	64	3x3	1
12	Conv Layer	64	64	3x3	2
13	Conv Layer	64	128	3x3	1
14	Conv Layer	128	128	3x3	2
15	Conv Layer	128	256	3x3	1
16	Conv Layer	256	256	3x3	2
17	Conv Layer	256	512	3x3	1
18	Conv Layer	512	512	3x3	2
19	Adaptive Avg Pooling	512	512	-	-
20	Conv Layer	512	1024	1x1	1
21	Conv Layer	1024	1	1x1	1

Generator: The Generator employs a sequence of convolutional layers and Residual Blocks to extract features and enhance the image quality. It begins with a convolution layer that processes the input image, followed by multiple Residual Blocks that enable the model to learn residual mappings, thereby improving the overall feature representation. The Generator also includes an Upsample Block, which utilizes Pixel Shuffle for efficient upsampling. The final layer produces the output image with three channels, normalized to the range [0, 1].

Discriminator: The Discriminator is a convolutional neural network that classifies images as real or generated. It consists of multiple convolutional layers with increasing depth and includes Leaky ReLU activations for non-linearity. The Discriminator down-samples the input image through strided convolutions and ends with an adaptive average pooling layer, which condenses the features into a single value for binary classification.

4 Loss Components

4.1 1. Adversarial Loss

$$\text{Adversarial Loss} = \text{mean}(1 - \text{out\_labels}) \tag{1}$$

Description: This loss measures how well the generator can fool the discriminator. In the context of GANs (Generative Adversarial Networks), the generator's goal is to produce images that are indistinguishable from real images. A lower value of this loss indicates that the generated images are more likely to be classified as real by the discriminator. Essentially, it incentivizes the generator to produce high-quality images that deceive the discriminator.

4.2 2. Perceptual Loss

$$\text{Perception Loss} = \text{MSE}(\text{loss\_network}(\text{out\_images}), \text{loss\_network}(\text{target\_images})) \tag{2}$$

Description: This loss compares the high-level feature representations of the generated images and the target images using a pre-trained VGG-16 network. By focusing on perceptual differences rather than pixel-wise differences, this loss helps the generator produce images that are visually similar to the target images. The VGG network extracts features that capture the essence of images, making this loss particularly effective for tasks like image super-resolution and style transfer.

4.3 3. Image Loss

$$\text{Image Loss} = \text{MSE}(\text{out\_images}, \text{target\_images}) \tag{3}$$

Description: This is a standard mean squared error (MSE) loss that measures the pixel-wise difference between the generated images and the target images. While useful, it can lead to blurry images if used alone. This loss encourages the generator to create images that closely resemble the target images at a pixel level.

4.4 4. Total Variation (TV) Loss

$$\text{TV Loss} = \frac{1}{\text{batch\_size}} \times 2 \times \left(\frac{\text{h\_tv}}{\text{count\_h}} + \frac{\text{w\_tv}}{\text{count\_w}} \right) \tag{4}$$

Description: Total Variation loss is used to reduce noise and promote spatial smoothness in the generated images. It penalizes rapid intensity changes in the image, which can help in creating more visually appealing images with fewer artifacts. By encouraging spatial continuity, this loss helps maintain the structure of the images while smoothing out unnecessary details.

4.5 Overall Loss Calculation

In the `forward` method of the `GeneratorLoss` class, the total loss for the generator is computed as a weighted sum of the individual losses:

$$\text{Total Loss} = \text{image\_loss} + 0.001 \times \text{adversarial\_loss} + 0.006 \times \text{perception\_loss} + 2 \times 10^{-8} \times \text{tv\_loss} \tag{5}$$

The weights (0.001, 0.006, and 2e-8) control the contribution of each loss term to the total loss, allowing for a balanced approach that considers the quality, realism, and visual characteristics of the generated images.

5 Result

Figure 1: Results of the SRGAN for High-Resolution Satellite Image