LISS4 Upsampling for Enhanced Visualization
Prof. Sudipan Saha

Tushar Verma (2022AIY7514)
ScAI, IIT Delhi

1 Introduction

LISS-4 (Linear Imaging Self-Scanning Sensor-4) is a high-resolution optical imaging sensor aboard ISRO's Resourcesat-2 and Resourcesat-2A satellites, designed for applications in land resource management, urban planning, agriculture, forestry, and disaster management. It provides a spatial resolution of approximately 5.8 meters per pixel and captures imagery in three spectral bands: Green (Band 2: 0.52–0.59 µm), Red (Band 3: 0.62–0.68 µm), and Near Infrared (NIR) (Band 4: 0.77–0.86 µm), covering a swath width of 23.9 km at nadir in high-resolution mode. Using SRGAN (Super-Resolution Generative Adversarial Networks) to upsample satellite images can be a promising approach to enhance spatial resolution and reveal finer details in imagery. SRGAN leverages a generator network to produce high-resolution (HR) images from low-resolution (LR) inputs and employs a discriminator to ensure the results resemble real HR images, alongside perceptual loss for improved visual fidelity. Upscaling LISS-4 data using methods like SRGAN could yield finer visual details but might not guarantee radiometric accuracy or meaningful enhancements for scientific applications.

Methodology Overview

Upscaling LISS-4 Imagery

The super-resolution process involves upscaling the LISS-4 imagery by a factor of four, yielding a target spatial resolution of approximately 1.445 meters per pixel. This enhances the details present in LISS-4 imagery while preserving its spectral characteristics.

Ground Truth Data

High-resolution imagery from Google Earth was downsampled from its original resolution of 0.6 meters per pixel to match the target resolution of 1.445 meters per pixel. This downsampled imagery served as the ground truth during training, providing a reliable reference for evaluating the performance of the super-resolution model.

Integration of Building Footprint Data

To further validate the results and ensure alignment with real-world structures, building footprint data from the Google Open Buildings Project was used during training.

These footprints provided additional spatial context and improved the model's ability to reconstruct fine details in urban and semi-urban areas.

2 Model Architecture

Generator			
Layer	**Input**	**Output**	**Details**
Conv2d + PReLU	`in_ch`	32	9×9, Pad: 4
Residual Block 1	32	32	3×3, BatchNorm
Residual Block 2	64	32	3×3, BatchNorm
Residual Block 3	96	32	3×3, BatchNorm
Residual Block 4	128	32	3×3, BatchNorm
Residual Block 5	160	32	3×3, BatchNorm
Residual Block 6	192	32	3×3, BatchNorm
Conv2d + BatchNorm	192	32	3×3, Pad: 1
Optional Mask Layer	32	1	9×9, Sigmoid
Upsample Block (Repeated)	32	32	Scale: $2\times$, SubPix
Final Conv2d	32	3	9×9, Pad: 4

Table 1: Generator Architecture

Discriminator			
Layer	**Input Channels**	**Output Channels**	**Details**
Conv2d + LeakyReLU	3	64	3×3, Padding: 1, Slope: 0.2
Conv2d + BatchNorm2d	64	64	3×3, Stride: 2
Conv2d + BatchNorm2d	64	128	3×3, Padding: 1
Conv2d + BatchNorm2d	128	128	3×3, Stride: 2
Conv2d + BatchNorm2d	128	256	3×3, Padding: 1
Conv2d + BatchNorm2d	256	256	3×3, Stride: 2
Conv2d + BatchNorm2d	256	512	3×3, Padding: 1
Conv2d + BatchNorm2d	512	512	3×3, Stride: 2
AdaptiveAvgPool2d	512	512	Output: 1×1
Conv2d + LeakyReLU	512	1024	1×1, Slope: 0.2
Conv2d	1024	1	1×1

Table 2: Discriminator Architecture

ResidualBlock Details: The ResidualBlock uses two convolutional layers with 3×3 kernels, followed by BatchNorm, and a PReLU activation. The input is added to the intermediate output (residual connection). A final 1×1 convolution ensures the output has 32 channels. This helps capture features efficiently while maintaining computational efficiency.

UpsampleBLock Details: The UpsampleBLock performs convolution, increases spatial resolution using PixelShuffle, and applies PReLU activation. Each block scales up by a factor of 2×, and multiple such blocks are stacked for the desired upscaling factor.

3 Loss Components

3.1 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.

3.2 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.

3.3 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.

3.4 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.

3.5 5. Dice Loss (Building Footprint)

Formula:

$$\text{Dice Loss} = 1 - \frac{2 \cdot (\text{inputs} \cdot \text{targets}).\text{sum}() + \text{smooth}}{\text{inputs}.\text{sum}() + \text{targets}.\text{sum}() + \text{smooth}}$$

Description: Used for binary masks of building footprints, this loss evaluates the overlap between predicted and ground truth masks. It ensures precise spatial reconstruction of building structures.

3.6 6. Overall Loss Calculation

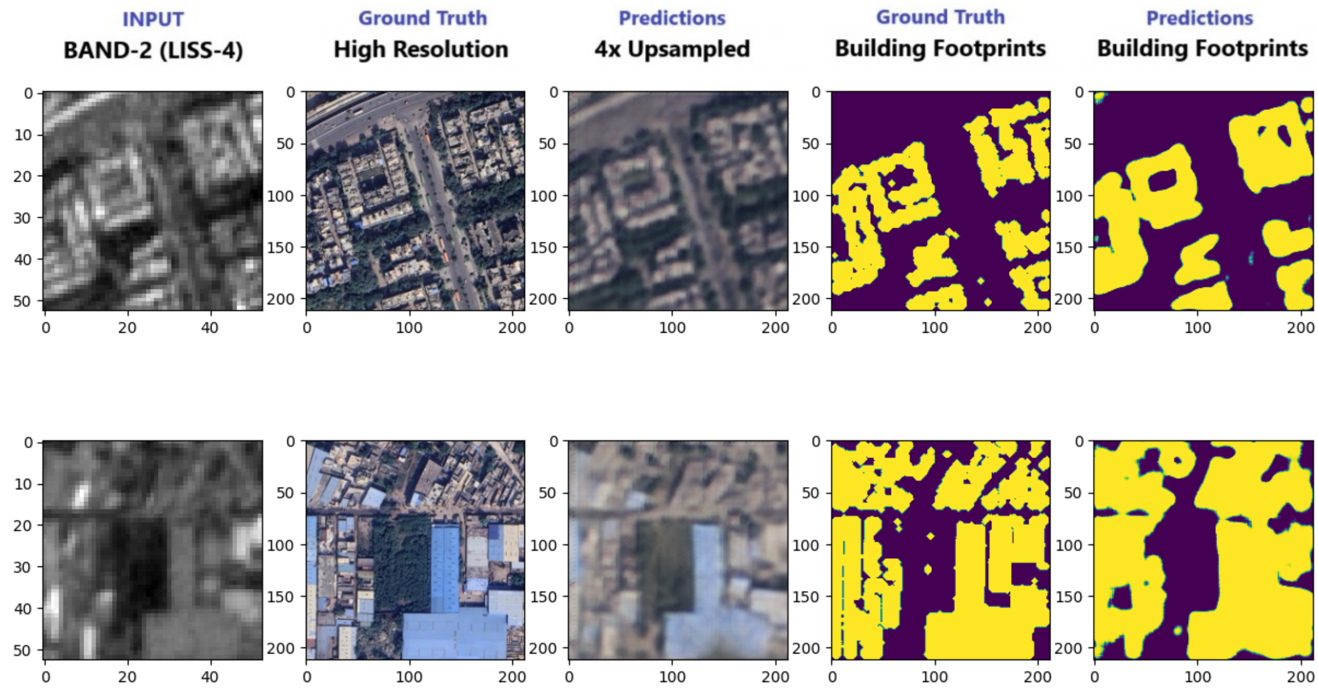
Formula:

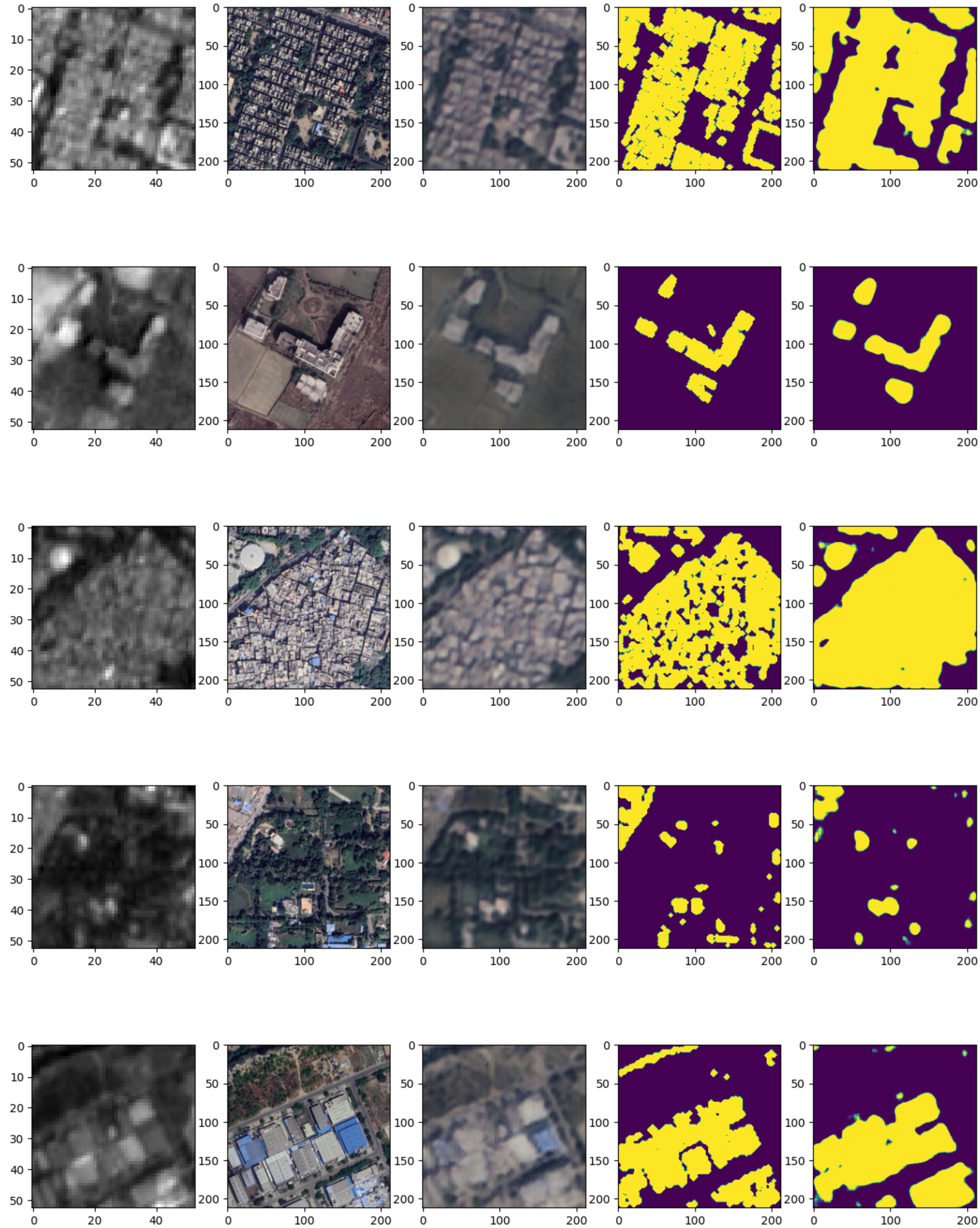
$$\begin{aligned}\text{Total Loss} = {} & \text{Image Loss} \\ & + 0.001 \times \text{Adversarial Loss} \\ & + 0.006 \times \text{Perception Loss} \\ & + 2 \times 10^{-8} \times \text{TV Loss} \\ & + 0.006 \times \text{Dice Loss}\end{aligned}$$

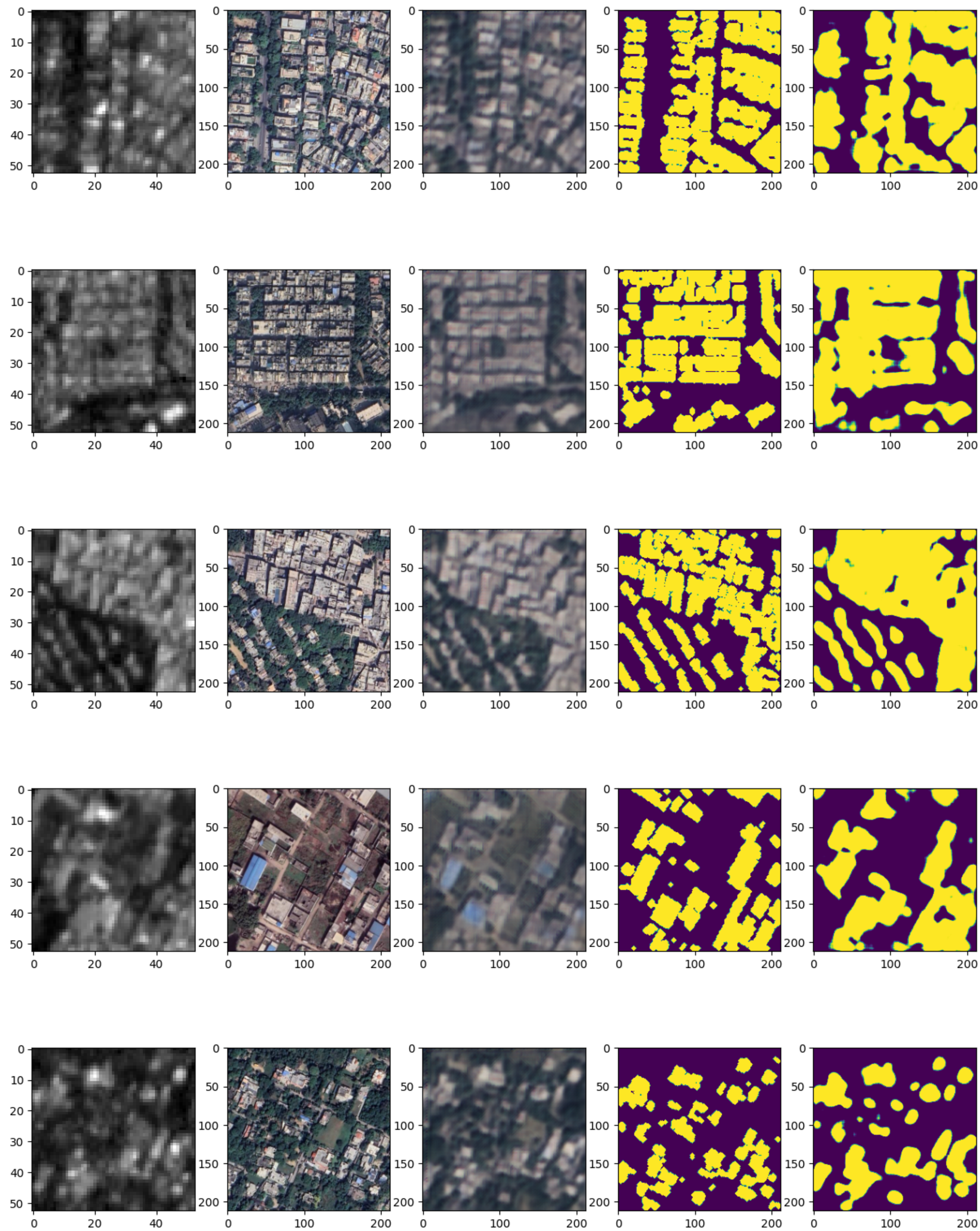
Description: A weighted sum of individual losses balances pixel-level accuracy, perceptual quality, realism, and spatial consistency while ensuring detailed building footprint prediction.

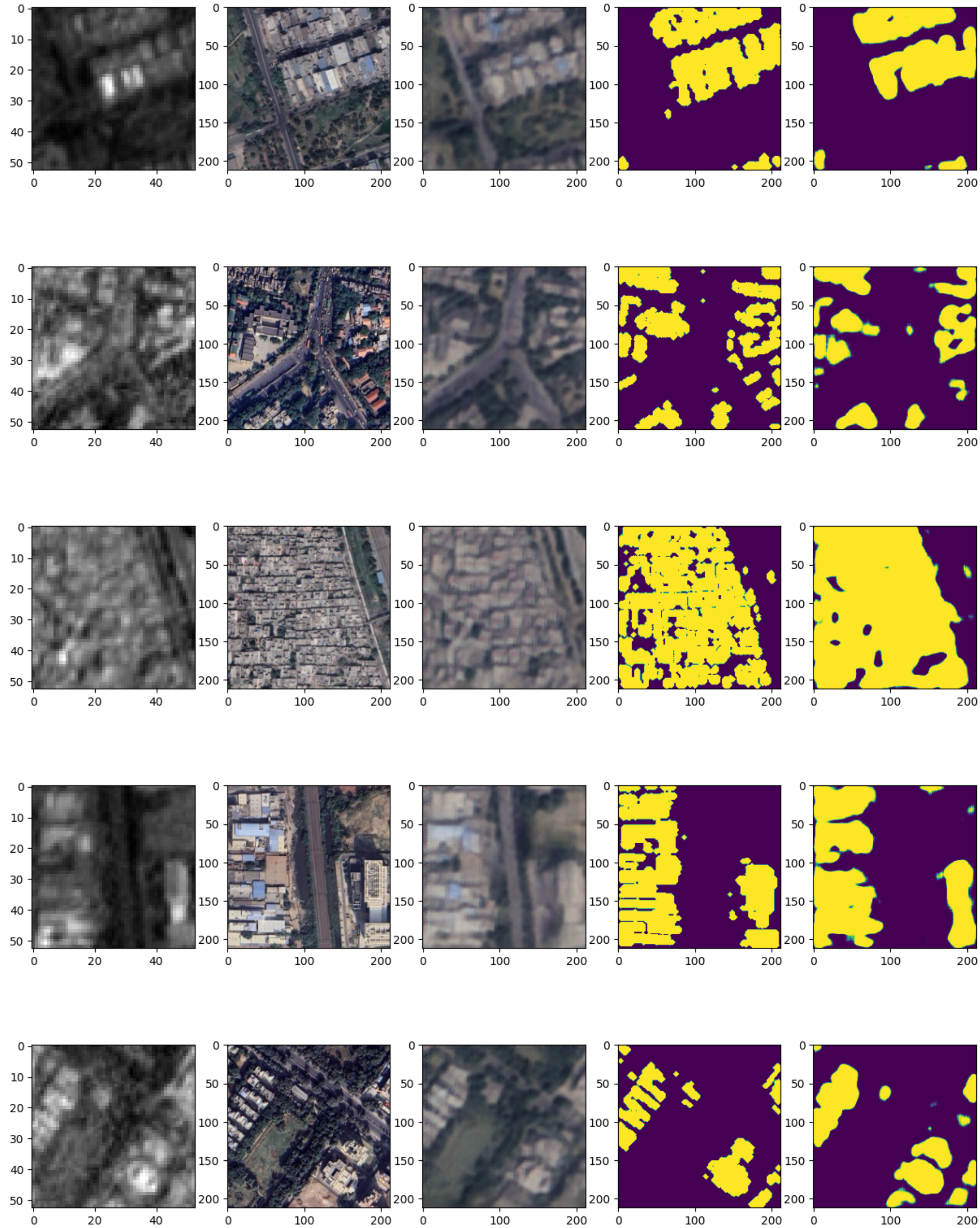
4 Result

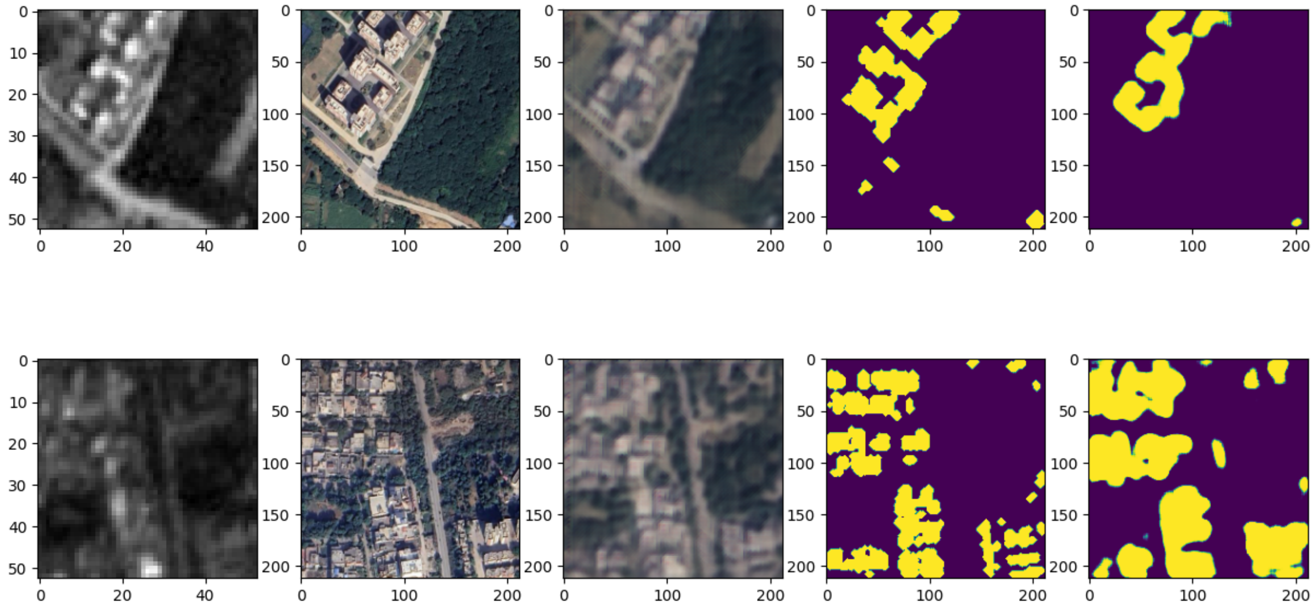
The results highlight remarkable visual clarity and strong alignment between predicted and actual building footprints, reflecting the precision and reliability of the model. The results of the test data set are presented below.




0 10 20 30 40 50
0 20 40
0 50 100 150 200
0 100 200



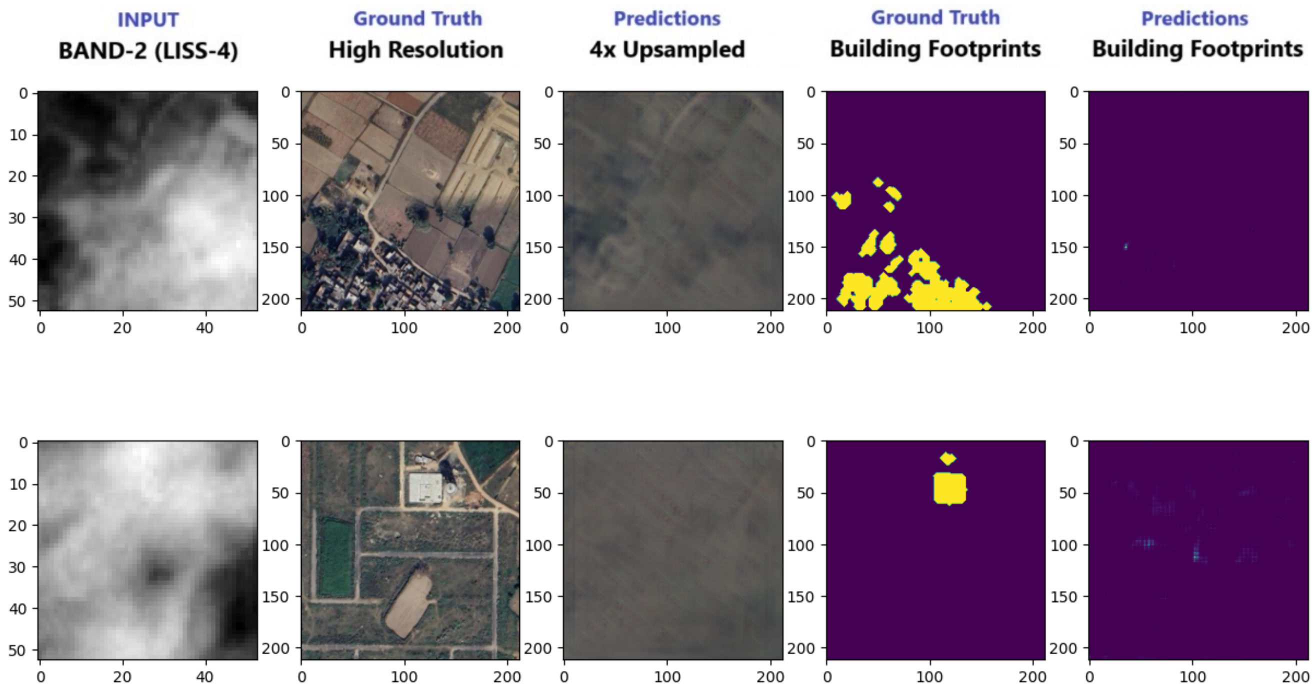


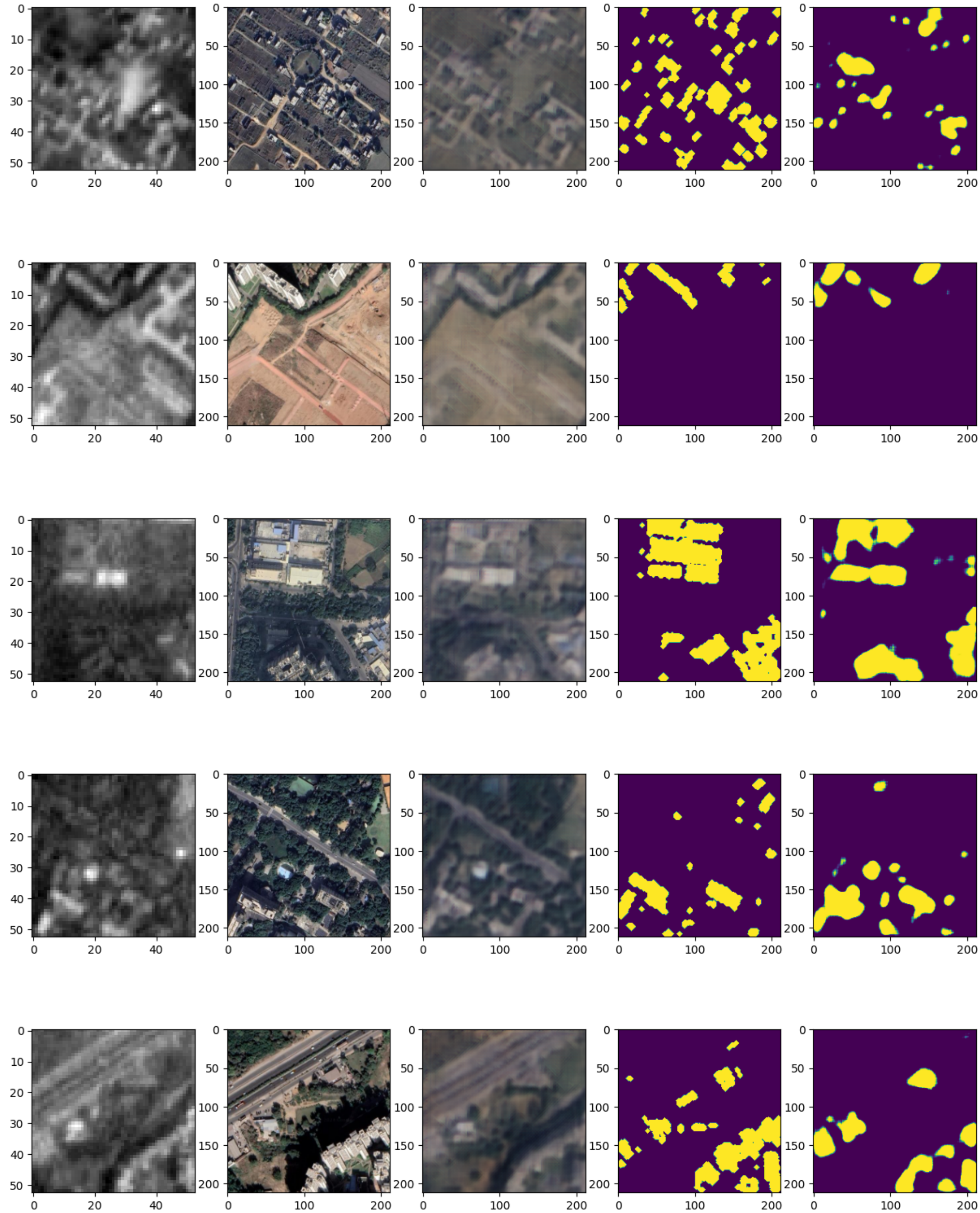


5 Challenges

The following results underscore the challenges of the model in accurately predicting building footprints, particularly for small structures and in regions affected by cloud cover. The NIR band, although useful for distinguishing vegetation and water bodies, does not always aid in mitigating the effects of cloud cover, as it does not directly contribute to resolving occluded building structures.

In addition, in cases involving very tall buildings, the model demonstrates a lower degree of visual clarity, suggesting that the model may struggle to accurately capture large-scale structures in complex urban environments.





Conclusion

In this project, we successfully applied advanced superresolution techniques to upscale LISS-4 satellite imagery using high-resolution Google Earth images. The process involved leveraging multiple loss functions, such as adversarial, perceptual, and image loss, alongside Dice Loss to accurately predict the footprint of the building. Our model showed promising results, with high visual clarity and substantial overlap between the predicted and actual footprints. This validates the effectiveness of our approach in enhancing low-resolution imagery and extracting meaningful spatial information.

Future Scope for Improvement

- **Enhanced Model for Tall Buildings:** Refining the model's ability to handle tall buildings—perhaps through multi-scale approaches or by integrating more contextual data—could improve footprint overlap.
- **Data Augmentation and Multi-Source Training:** Expanding the training dataset by incorporating more varied urban environments and satellite data from different sources would help improve model robustness and accuracy.
- **Real-Time Application and Deployment:** Moving forward, this technique could be adapted for real-time urban monitoring and disaster management systems, where timely and accurate building footprint detection is critical.