GAN-Based SAR-to-Optical Image Translation

Sameer Chakrawrti, Vinit Saini

1 Discriminator Comparison and Analysis

This section presents a comparative analysis of different discriminator architectures evaluated for the SAR-to-optical image translation task. Three discriminator designs were studied: a discriminator trained from scratch, a PatchGAN discriminator, and a Wasserstein GAN (WGAN) critic. The comparison focuses on architectural design, training behavior, and qualitative impact on generated outputs.

1.1 Discriminator Trained from Scratch

1.1.1 Architecture Overview

The baseline discriminator was implemented as a convolutional neural network trained from scratch to perform binary classification between real and generated optical images. The network accepts an input image of size $128 \times 128 \times 3$ and produces a single sigmoid-activated probability indicating real or fake.

Layer (Type)	**Output Shape**	**Parameters**	**Description**
Conv2D	$(32, 32, 64)$	1,792	Low-level feature extraction
LeakyReLU	$(32, 32, 64)$	0	Non-linear activation
Conv2D	$(16, 16, 128)$	73,856	Downsampling via stride $(2, 2)$
LeakyReLU	$(16, 16, 128)$	0	Prevents neuron inactivation
Conv2D	$(8, 8, 128)$	147,584	Intermediate feature learning
LeakyReLU	$(8, 8, 128)$	0	Activation
Conv2D	$(4, 4, 256)$	295,168	High-level feature extraction
LeakyReLU	$(4, 4, 256)$	0	Activation
Flatten	(4096)	0	Feature vector conversion
Dropout (0.4)	(4096)	0	Regularization
Dense (Sigmoid)	(1)	4,097	Real/fake probability output

Table 1: Architecture of the discriminator trained from scratch. Total parameters: 522,497 ($\approx$1.99 MB).

1.1.2 Training Setup

The discriminator was optimized using the binary cross-entropy loss:

$$L_D = -\left[y \log(D(x)) + (1-y) \log(1 - D(x))\right],$$

where real samples were labeled as 1 and generated samples as 0.

- Optimizer: Adam (learning rate = 0.001, $\beta_1 = 0.5$)
- Number of epochs: 20

1.1.3 Training Results

Epoch	Real Accuracy (%)	Fake Accuracy (%)
1	89.1	44.5
5	69.8	72.8
10	85.1	85.9
15	90.2	90.5
20	92.7	92.9

Table 2: Classification accuracy of the baseline discriminator.

1.1.4 Observation

The discriminator demonstrated steady and stable convergence, achieving high classification accuracy for both real and fake samples. This behavior indicates effective learning and reliable adversarial feedback to the generator.

1.2 PatchGAN Discriminator

The PatchGAN discriminator was evaluated to encourage local realism by classifying overlapping image patches rather than the entire image. The discriminator takes concatenated SAR and optical image pairs as input and produces a spatial map of real/fake predictions.

1.2.1 Observation

The PatchGAN discriminator converged more slowly and achieved moderate accuracy (approximately 54%). While it encouraged local texture consistency, it provided weaker global discrimination compared to the full-image discriminator.

Layer (Type)	Output Shape	Parameters	Description
Input (SAR)	$(128, 128, 3)$	0	SAR image
Input (Optical)	$(128, 128, 3)$	0	Optical image
Concatenate	$(128, 128, 6)$	0	Channel-wise fusion
Conv2D	$(64, 64, 32)$	3,104	Low-level feature extraction
LeakyReLU	$(64, 64, 32)$	0	Activation
Conv2D	$(32, 32, 64)$	32,832	Downsampling
BatchNorm	$(32, 32, 64)$	256	Stabilization
LeakyReLU	$(32, 32, 64)$	0	Activation
Conv2D	$(16, 16, 128)$	131,200	Deep feature learning
BatchNorm	$(16, 16, 128)$	512	Normalization
LeakyReLU	$(16, 16, 128)$	0	Activation
Dropout	$(16, 16, 128)$	0	Regularization
Conv2D	$(16, 16, 256)$	524,544	Local structure modeling
BatchNorm	$(16, 16, 256)$	1,024	Stability
LeakyReLU	$(16, 16, 256)$	0	Activation
Conv2D	$(16, 16, 1)$	4,097	Patch-wise output

Table 3: PatchGAN discriminator architecture (697,569 parameters).

1.3 Wasserstein GAN (WGAN) Critic

A WGAN-based critic was also evaluated to improve training stability by replacing the probabilistic discriminator with a real-valued critic.

1.3.1 Training Objective

The critic was optimized using the Wasserstein loss:

$$L_D = -\mathbb{E}[D(x_{\text{real}})] + \mathbb{E}[D(x_{\text{fake}})].$$

- Optimizer: RMSprop (learning rate = 0.00005)
- Weight clipping: $|w| \leq 0.01$

1.3.2 Observation

Training with the WGAN critic was stable and exhibited smooth convergence. However, the generated outputs tended to be overly smooth, resulting in blurred visual details.

1.4 Overall Comparison

Discriminator	Loss	Stability	Accuracy	Visual Impact
From scratch	BCE	High	∼92.8%	Strong global realism
PatchGAN	BCE	Moderate	∼54%	Improved local consistency
WGAN critic	Wasserstein	Very high	–	Coarse, blurred outputs

Table 4: Comparison of discriminator architectures.

2 Generator Architecture (U-Net)

The generator used across all experiments (basic GAN, PatchGAN, and WGAN) follows an encoder–decoder architecture similar to U-Net. The encoder compresses the input image into a lower-dimensional latent representation (latent space), capturing essential structural and contextual features. The decoder then reconstructs the image from this latent vector, with skip connections ensuring spatial detail preservation.

Layer (Type)	Output Shape	Parameters	Description
Conv2D + LeakyReLU	(64,64,64)	3,136	Feature extraction
Conv2D + BN + LeakyReLU	(32,32,128)	131,712	Deep encoding
Conv2D + BN + LeakyReLU	(16,16,256)	525,568	High-level encoding
Conv2D + BN + LeakyReLU	(8,8,512)	2,099,712	Latent feature space
Conv2DTranspose + BN + ReLU	(16,16,256)	2,098,432	Upsampling
Conv2DTranspose + BN + ReLU	(32,32,128)	1,049,216	Reconstruction
Conv2DTranspose + BN + ReLU	(64,64,64)	262,464	Fine details
Conv2DTranspose	(128,128,3)	6,147	RGB output

Table 5: U-Net generator architecture (6.17M parameters).

Generator Training and Results The generator was trained using a combined loss function consisting of:

- Binary Cross-Entropy (BCE) for adversarial feedback from the discriminator.
- Mean Absolute Error (L1 loss / MAE) for image reconstruction accuracy. These were weighted by a parameter (λ_{L1}), balancing realism and pixel-level fidelity.

Training Progress: Between epochs 10 and 130, the generator's total loss shows only a marginal decrease, dropping from approximately 33.0 to 28.4, while the L1 loss reduces slightly from about 0.295 to 0.250. This indicates slow convergence, with losses stabilizing rather than decreasing sharply.

Figure 1: At Epoch 10

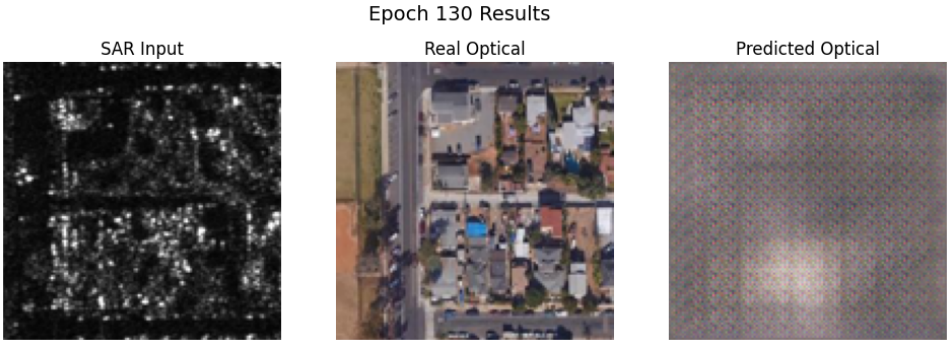

Figure 2: At Epoch 130

3 Pre-Trained Model Overview

The pre-trained model from `yuuIind/SAR2Optical` follows a Pix2Pix framework [**?**] using a U-Net generator and PatchGAN discriminator. The architecture was trained on the Sentinel-1/2 dataset, which consists of co-registered SAR and optical image pairs captured by the European Space Agency's Sentinel satellites.

3.1 Architecture Details

The generator follows an encoder-decoder structure with skip connections:

- **Encoder**: 8 downsampling blocks (Conv2d → BatchNorm → LeakyReLU)
- **Decoder**: 8 upsampling blocks (ConvTranspose2d → BatchNorm → Dropout → ReLU)
- **Skip connections**: Concatenation between corresponding encoder-decoder layers
- **Output**: $3 \times 256 \times 256$ RGB image via Tanh activation

The discriminator is a PatchGAN classifier that operates on 70×70 patches, determining whether each patch is real or fake.

3.2 Loss Functions

The training objective combines adversarial and reconstruction losses:

- **Adversarial loss**: Binary cross-entropy between discriminator predictions and target labels
- **Reconstruction loss**: L1 (Mean Absolute Error) between generated and ground truth images
- **Total generator loss**: $\mathcal{L}_G = \mathcal{L}_{adv} + \lambda \mathcal{L}_{L1}$, where $\lambda = 100$

3.3 Baseline Performance on Training Domain

To establish a baseline, we first evaluated the pre-trained model on images from its original training domain (Sentinel-1/2 dataset). As shown in Figure 3, the model produces high-quality optical reconstructions with accurate color representation, sharp structural details, and faithful reproduction of land cover features.

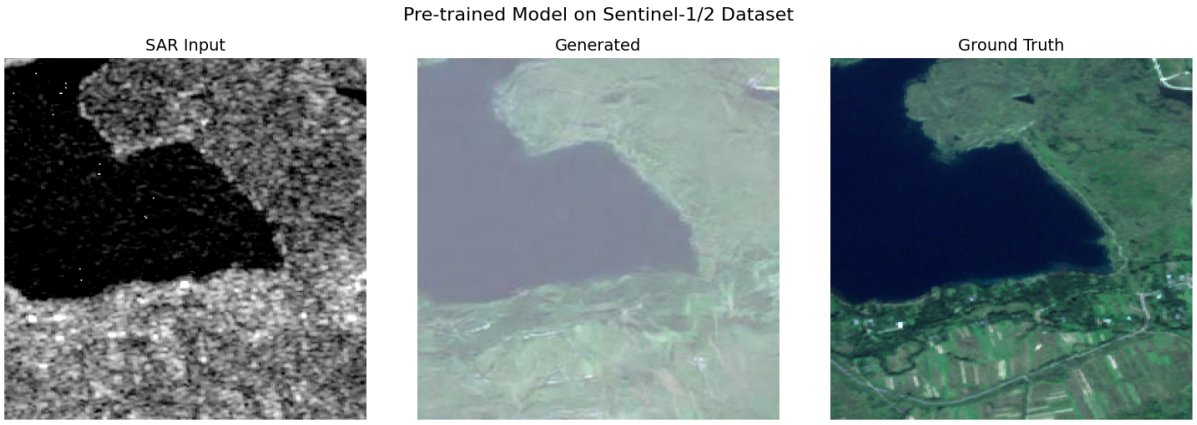

Figure 3: Pre-trained model performance on Sentinel-1/2 dataset (training domain).

3.4 Direct Inference on QXSLAB-SAROPT Dataset

When the same pre-trained model was applied directly to the QXSLAB-SAROPT dataset without any preprocessing, the results exhibited severe quality degradation, as illustrated in Figure 4.

Figure 4: Pre-trained model inference on raw QXSLAB-SAROPT images.

Analysis of Artifacts. The generated images exhibit several characteristic failure modes:

- **Crystalline/blocky artifacts**: The model produces unnatural geometric patterns resembling crystal structures. This occurs because the input SAR statistics fall outside the distribution the model was trained on, causing the convolutional filters to respond erratically and produce high-frequency artifacts.
- **Excessive brightness/whitening**: The generated images appear washed out with abnormally high intensity values. This is attributed to the different radiometric calibration between Sentinel-1 and QXSLAB SAR sensors, where the input dynamic range does not match the expected distribution.
- **Loss of structural coherence**: Buildings, roads, and field boundaries visible in the SAR input are not preserved in the generated output. The domain shift causes the encoder features to misrepresent the spatial structure.
- **Color bleeding and hallucination**: The model hallucinates colors that do not correspond to the actual ground cover, producing white, baige and other unrealistic hues.

Root Cause: Domain Mismatch. The fundamental issue is the significant domain gap between the Sentinel-1/2 training data and the QXSLAB-SAROPT dataset. Key differences include:

- **Sensor characteristics**: Different radar frequencies, incidence angles, and polarization modes
- **Speckle noise patterns**: Varying noise statistics due to different acquisition parameters
- **Radiometric calibration**: Different preprocessing pipelines applied to raw SAR data

- **Spatial resolution**: Differences in ground sampling distance and pixel spacing

It is worth noting that satellite imagery providers such as NASA and ESA apply extensive preprocessing pipelines to their distributed products, including radiometric terrain correction, speckle filtering, and calibration to backscatter coefficients. These processing steps are often proprietary or involve sensor-specific parameters that are difficult to replicate exactly for cross-sensor applications.

3.5 Inference with Preprocessing

To mitigate the domain mismatch, we applied preprocessing to the QXSLAB-SAROPT SAR images before inference. The preprocessing pipeline included:

- **Speckle filtering**: Median filter to reduce multiplicative noise
- **Window size**: 7×7 pixels for local smoothing
- **Gamma correction**: $\gamma = 7$ to adjust the dynamic range and contrast

The results after preprocessing are shown in Figure 5.

Figure 5: Pre-trained model inference on preprocessed QXSLAB-SAROPT images.

Observations. While preprocessing provides marginal improvements—reducing some of the crystalline artifacts and normalizing the brightness to some extent—the generated images still fall significantly short of the ground truth quality. The structural details remain blurred, colors are inconsistent, and the overall fidelity is inadequate for practical applications. With larger window sizes, more white or crystalline structures were observed.

Conclusion. These experiments demonstrate that preprocessing alone is insufficient to bridge the domain gap between Sentinel-1/2 and QXSLAB-SAROPT datasets. The pre-trained model's learned features are fundamentally misaligned with the new domain's characteristics, necessitating fine-tuning or retraining approaches, which are explored in subsequent sections.

4 Fine-Tuning of Pre-Trained Pix2Pix Generator

4.1 Fine-tuning with Full Parameter Update

I attempted full fine-tuning of the pre-trained model on the QXSLAB-SAROPT dataset. This approach updates all 54.5 million parameters of the generator during training.

4.1.1 Training Configuration

The fine-tuning was performed on Google Colab using an NVIDIA A100 GPU (80GB). Table 4.1.1 summarizes the training configuration.

Parameter	**Value**
Pre-trained Checkpoint	pix2pix_gen_180.pth
Number of Epochs	50
Batch Size	32
Learning Rate	5×10^{-5}
Optimizer	Adam ($\beta_1 = 0.5$, $\beta_2 = 0.999$)
L1 Loss Weight (λ)	100
Training Samples	16,000 (80%)
Validation Samples	2,000 (10%)
Data Augmentation	Horizontal/Vertical flip, 90° rotations

Table 6: Full Fine-tuning Configuration

4.1.2 Training Results

The model was trained for 50 epochs. Figure 6 shows the training curves for discriminator loss, generator loss, and L1 reconstruction loss.

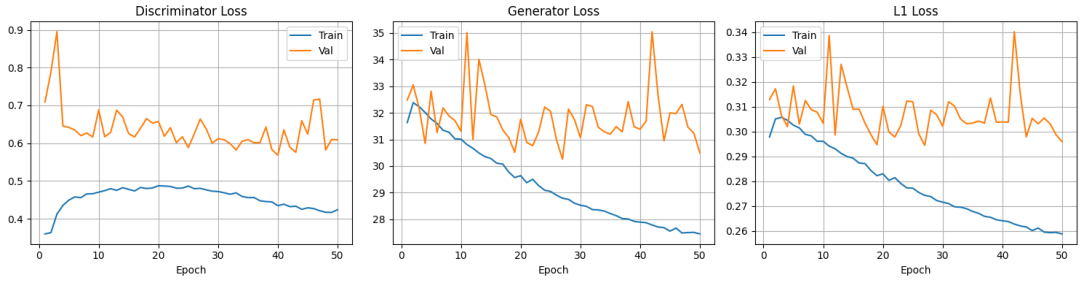

Figure 6: Training curves for full fine-tuning over 50 epochs.

Key observations from the training:

- **Discriminator**: Training loss stabilized around 0.42-0.48, while validation remained at 0.60-0.65, indicating the discriminator learned to distinguish real from fake on training data.

- **Generator**: Training loss decreased from 32.5 to 27.5, with validation fluctuating between 30-35.
- **L1 Loss**: Training L1 improved consistently from 0.34 to 0.26. Validation L1 showed improvement from 0.34 to approximately 0.30, with notable variance.

4.1.3 Qualitative Results

Figure 7 presents sample outputs from the fine-tuned model on the validation set.

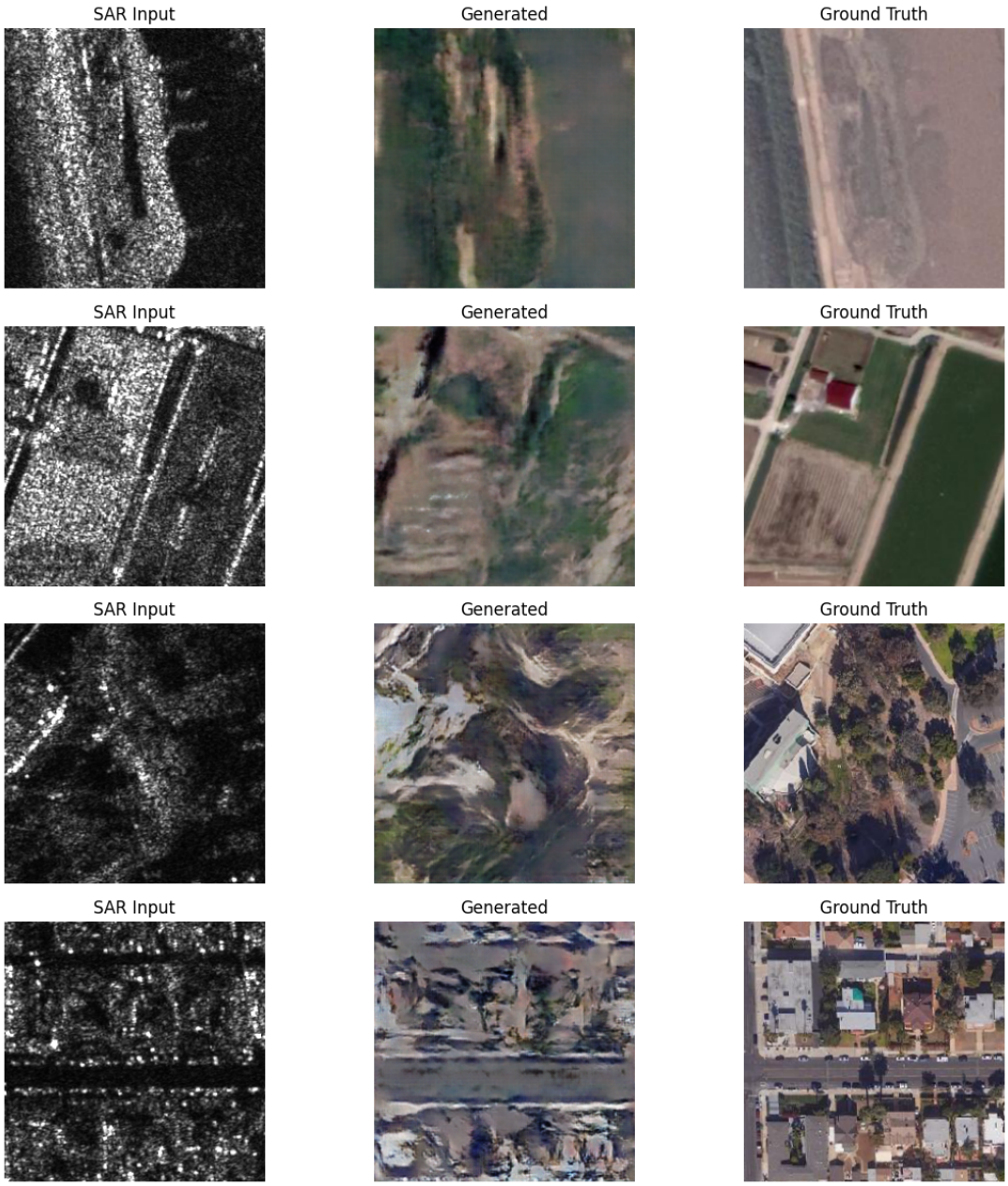

Figure 7: Qualitative results from full fine-tuning on QXSLAB-SAROPT. The model captures general color tones and some structural features but lacks fine details.

4.1.4 Analysis

While full fine-tuning achieved lower training losses compared to the LoRA approach, the results exhibit similar limitations:

- **Blurry outputs**: Generated images lack sharpness, particularly for building edges and fine textures.
- **Color approximation**: The model learns approximate color distributions but fails to accurately reconstruct specific land cover colors.
- **Structural loss**: Complex structures like buildings and road networks are not faithfully reproduced.
- **Overfitting tendency**: The gap between training and validation L1 loss (0.26 vs 0.30) suggests mild overfitting despite data augmentation.

The full fine-tuning approach, while computationally more expensive than LoRA, did not yield significantly better visual quality. This suggests that the fundamental domain gap between Sentinel-1/2 and QXSLAB-SAROPT datasets may require more sophisticated adaptation techniques or training from scratch on the target domain.

4.2 Fine-Tuning with LoRA Adapters

To adapt the pre-trained Pix2Pix generator to the QXSLAB_SAROPT dataset in a parameter-efficient manner, LoRA adapters were integrated into the generator network. Fine-tuning was performed by updating only the LoRA parameters while keeping the original pre-trained weights frozen.

4.2.1 Implementation Details

LoRA adapters were implemented within the U-Net generator by augmenting the convolutional layers of the encoder with low-rank adaptation modules. The specific configuration used in the experiments is summarized in Table 4.2.1.

Parameter	**Value**
LoRA Rank (r)	16
LoRA Alpha (α)	32
Dropout	0.1
Target Modules	Encoder convolutions
Learning Rate	0.0002
Batch Size	32
Number of Epochs	30
Optimizer	Adam ($\beta_1 = 0.5$, $\beta_2 = 0.999$)

Table 7: LoRA Configuration Parameters

The adapters were applied exclusively to the eight encoder convolutional layers (enc1–enc8), as these layers are primarily responsible for feature extraction and domain-specific representation learning. The decoder layers were left unchanged. The discriminator was trained from scratch to model the target domain distribution.

With this design, the number of trainable parameters was reduced to approximately 1.5% of the original generator size, decreasing from 54.5 million parameters to roughly 820,000 trainable LoRA parameters.

4.2.2 Training Results

The model was trained for 30 epochs on the QXSLAB_SAROPT dataset. Quantitative training and validation statistics at selected epochs are reported in Table 4.2.2.

Epoch	**Train $\mathcal{L}_D$**	**Train $\mathcal{L}_G$**	**Train $\mathcal{L}_{L1}$**	**Val $\mathcal{L}_D$**	**Val $\mathcal{L}_G$**	**Val $\mathcal{L}_{L1}$**
1	0.2156	38.0381	0.3513	0.6789	34.3414	0.3326
5	0.1020	38.5482	0.3445	0.8328	35.8182	0.3381
10	0.0427	40.4052	0.3508	0.6985	36.4843	0.3428
15	0.0717	40.0660	0.3472	0.8892	35.6108	0.3321
20	0.0528	40.2161	0.3435	1.1539	37.3918	0.3437
25	0.0015	42.6669	0.3509	1.2074	37.2212	0.3390
29	0.1026	41.9068	0.3446	0.8156	32.8147	**0.3123**
30	0.0554	39.3827	0.3414	1.3804	38.7455	0.3495

Table 8: LoRA Fine-Tuning Training Statistics (Selected Epochs)

Several observations can be drawn from the training behavior:

- **Discriminator instability**: The training discriminator loss rapidly decreased to near-zero values, while the validation loss remained high and unstable.
- **Generator behavior**: The generator loss increased over training, indicating ineffective adversarial learning.
- **L1 reconstruction**: The training and validation L1 losses showed limited improvement and high variance, suggesting unstable convergence.

Figure 8 presents the loss curves over all epochs, highlighting the divergence between training and validation behavior.

4.2.3 Qualitative Results

Representative inference results on the QXSLAB_SAROPT validation set are shown in Figure 9.

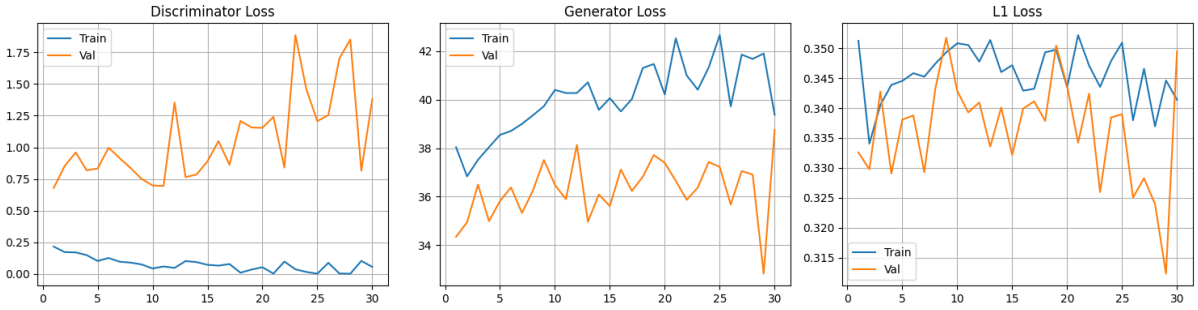

Figure 8: Training curves for LoRA fine-tuning showing (a) discriminator loss, (b) generator loss, and (c) L1 loss over 30 epochs. The large gap between training and validation discriminator loss indicates discriminator collapse.

4.2.4 Failure Analysis

The LoRA-based fine-tuning approach failed to produce meaningful SAR-to-optical translations on the QXSLAB_SAROPT dataset. Several failure modes were identified.

Discriminator Collapse. The discriminator rapidly memorized the training data, as evidenced by near-zero training loss (0.0015 at epoch 25) and persistently high validation loss (0.68–1.38). This collapse resulted in weak adversarial gradients, effectively reducing training to L1 loss minimization.

Limited Adaptation Capacity. The substantial domain gap between the Sentinel-based pre-training data and the QXSLAB_SAROPT dataset—including differences in resolution, speckle statistics, acquisition geometry, land-cover distribution, and radiometric calibration—could not be bridged by rank-16 LoRA adapters, which constituted only 1.5% of the total model parameters.

Training–Validation Divergence. Persistent divergence between training and validation losses indicates unstable optimization and poor generalization, with no consistent convergence observed.

Qualitative Artifacts. The generated images consistently exhibit:

- Loss of structural detail in buildings, roads, and field boundaries
- Severe color inconsistency and bleeding artifacts
- Repetitive, unrealistic texture hallucinations
- Spatial misalignment with SAR input geometry

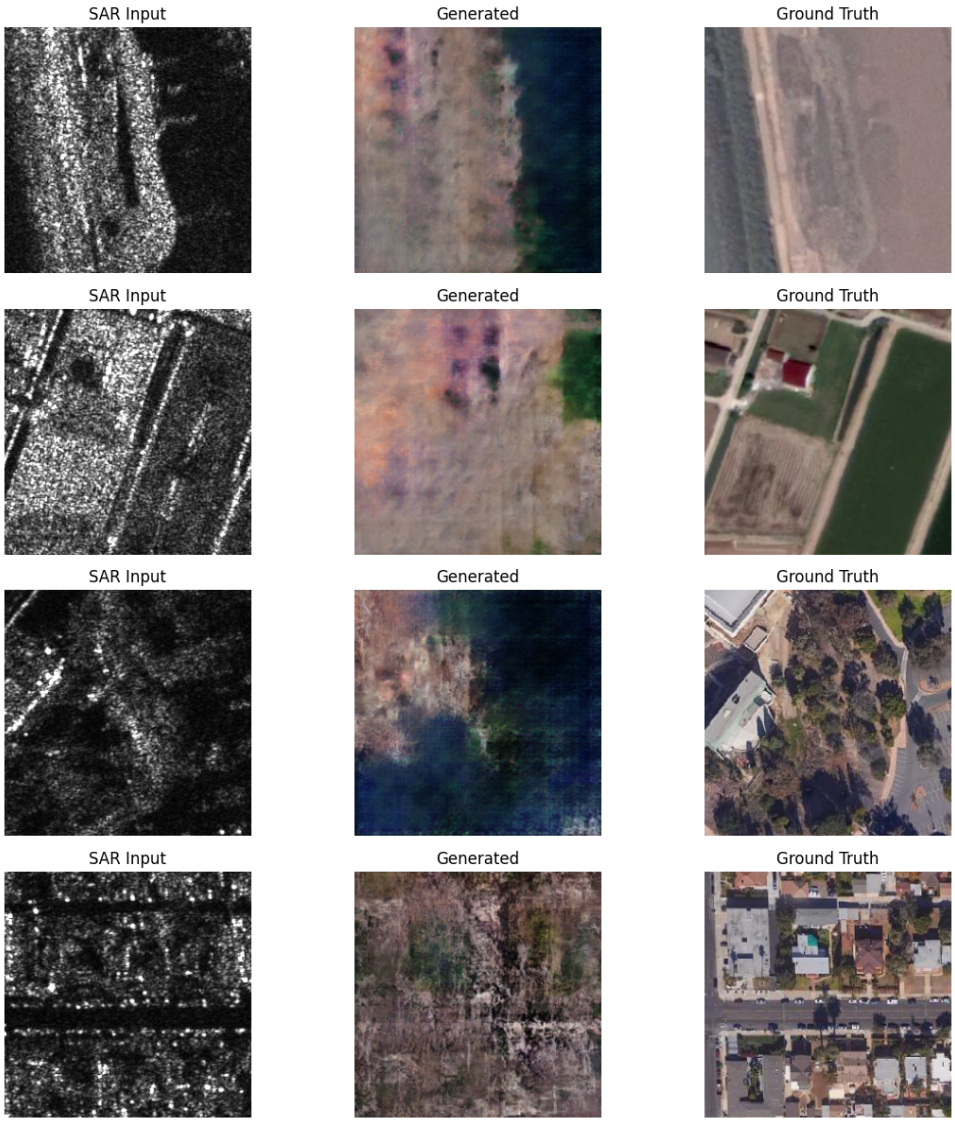

Figure 9: Qualitative results from LoRA fine-tuning on QXSLAB_SAROPT. The generated outputs lack structural fidelity.

4.2.5 Conclusion

Although LoRA-based fine-tuning significantly reduced the number of trainable parameters, it proved insufficient for adapting the pre-trained Pix2Pix model to the QXSLAB_SAROPT domain. The combination of discriminator collapse, limited adaptation capacity, and a large domain mismatch resulted in poor qualitative and quantitative performance. These findings suggest that SAR-to-optical translation under substantial domain shifts requires full retraining or more advanced domain adaptation strategies.