Team Astra

Shashank Patil, Ashwin Nayar

Data

Objective: Develop a deep learning model for SAR/EO Image-to-Image translation

Preprocessing Steps:

1. SAR Image - Values encode distance and intensity, preserved this information by using a global constant for normalization rather than per image minmax scaling.
2. EO image - Dropped 4th channel as it captured less information relative to other channels (Appendix A) and normalized to [0, 1]
3. Random Jittering: Random resize (286*286) followed by cropping (256*256) and mirroring - Helps in generalization
4. Data split: 80% (train) + 10% (validation) + 10% (test)

Methodology

We opted for the Pix2Pix GAN for its expertise in paired image-to-image translation, perfectly suited for our SAR to EO image conversion task.

Architecture (*Refer Appendix B*)

Generator - employs a U-Net architecture, incorporating an encoder-decoder structure with skip connections to preserve spatial details during the translation process

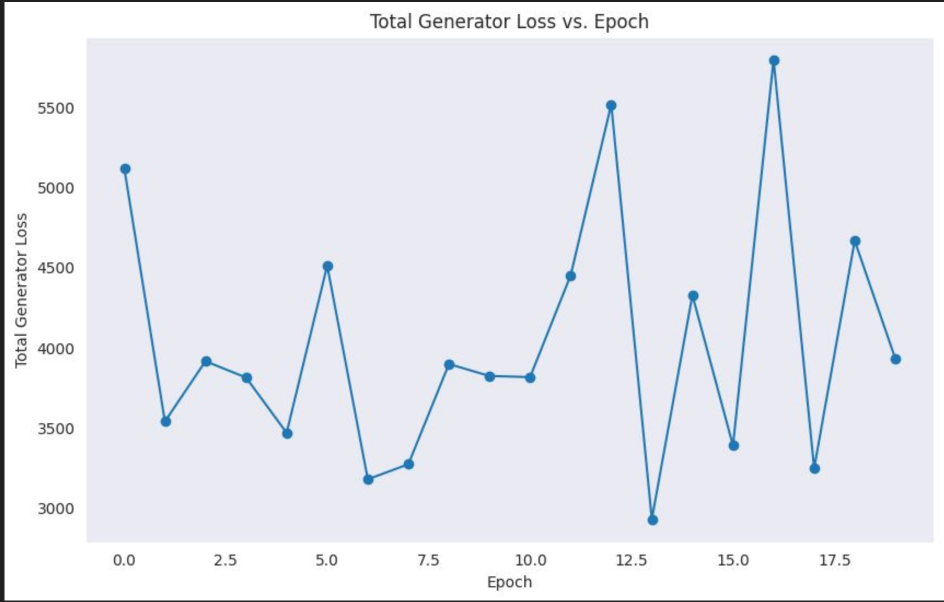
Discriminator - trained to differentiate between real pairs and translated pairs, guiding the generator to produce more realistic outputs through adversarial training

Loss Function

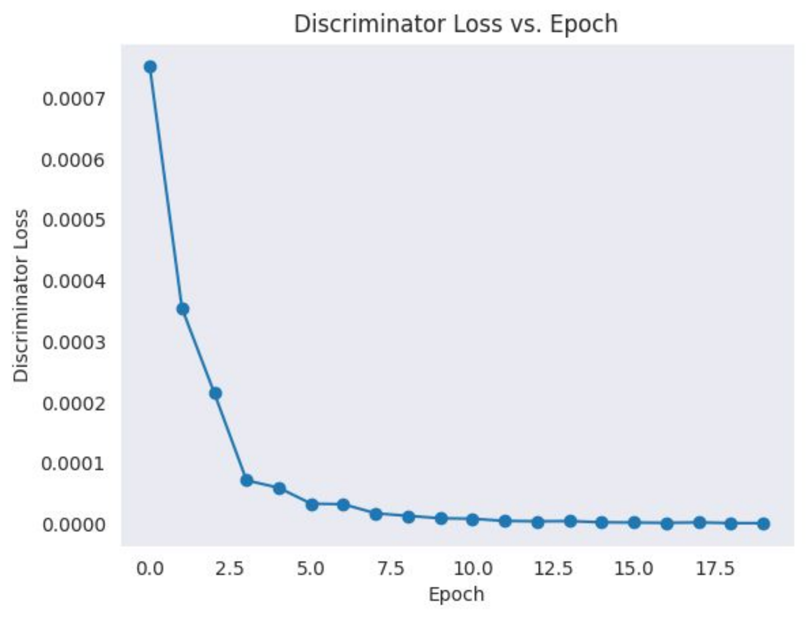
The Binary Cross-Entropy (BCE) loss is defined to model the objectives of the Generator and Discriminator networks. There is a second reconstruction loss LI Loss specifically for the generator, defined in the generator loss function. (*Refer Appendix C*)

Training

No of Epochs = 20, TIme taken = 1.5 hrs, Machine = T4 GPU (15gb RAM), Train,Val - 0.8, 0.1



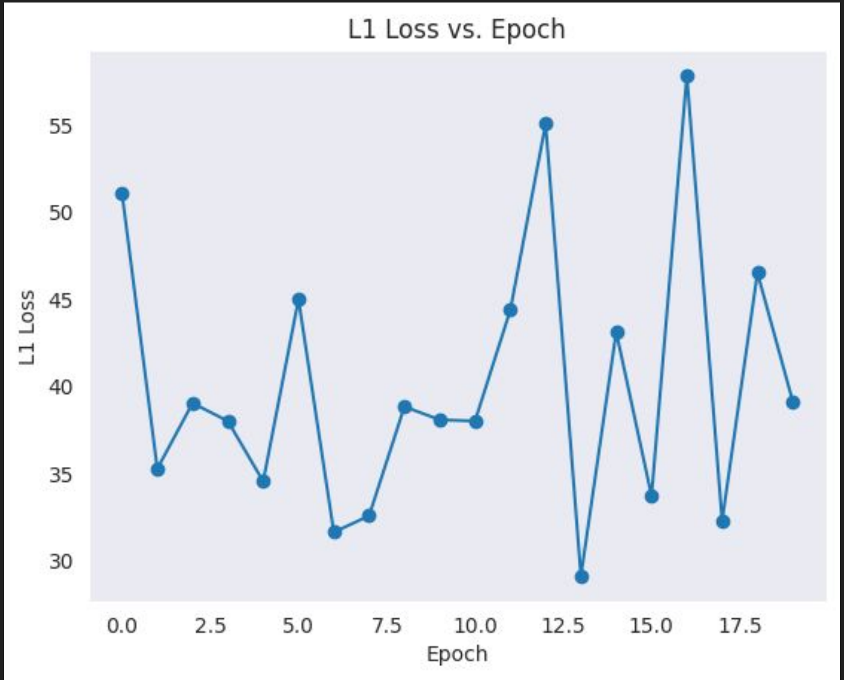
Generator loss stabilizes with epochs



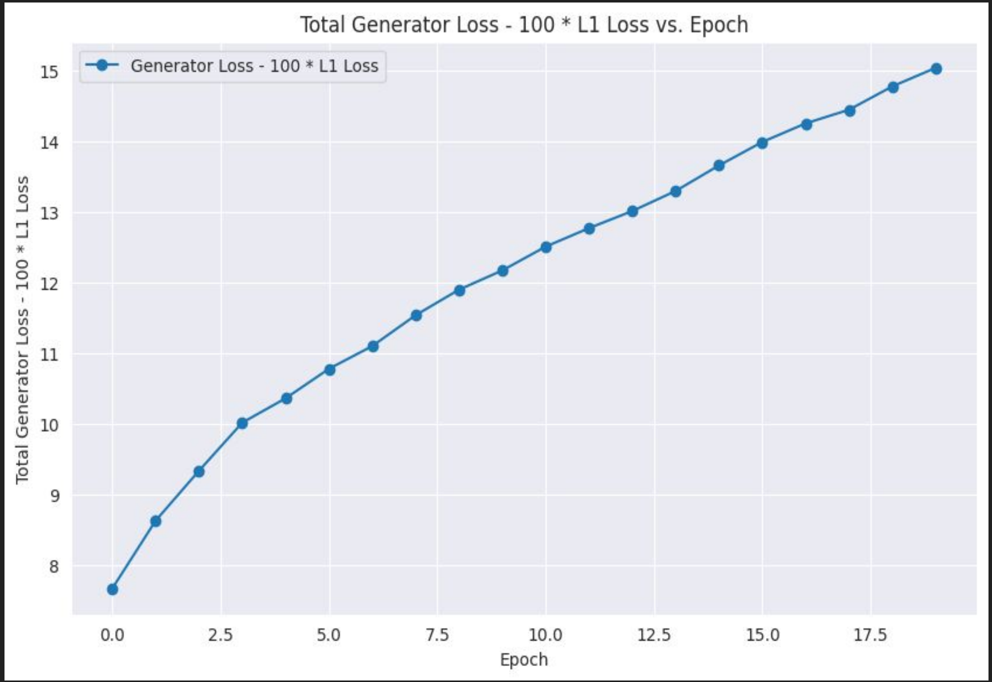
Discriminator loss decreases with epochs

Note: Default Hyperparameters were used

Training



L1 loss stabilizes with epochs

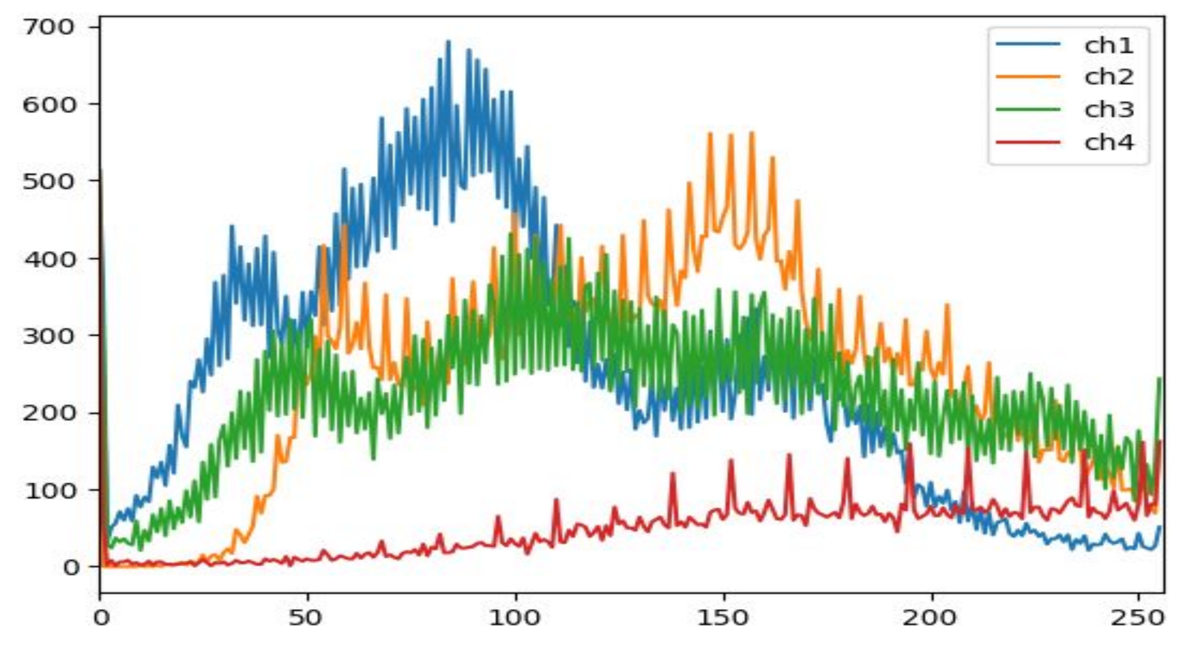


Total generator loss increases with epochs

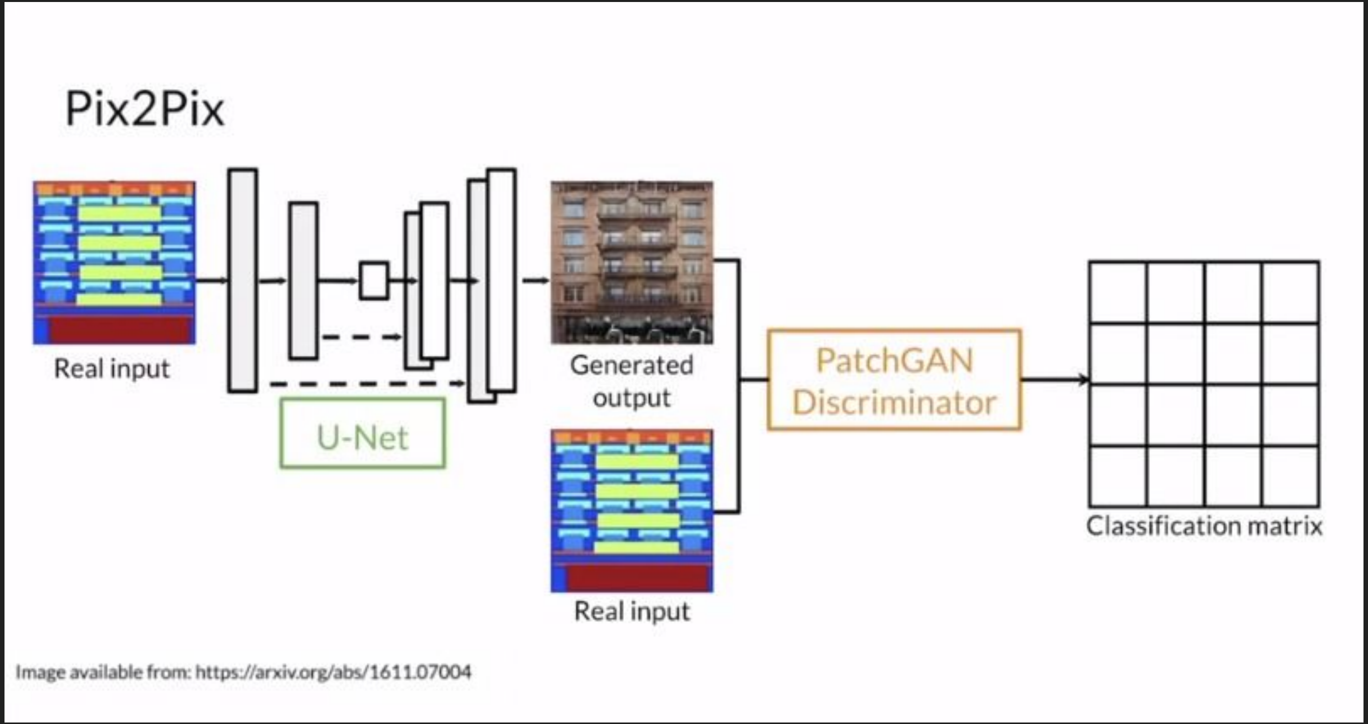
Test data examples

https://colab.research.google.com/drive/1QnVvT0eosHd6XXrvBa5KnPsEIFSCJt2j?authuser=2#scrollTo=5CRkbcMa6Os3

Appendix A : Channel Histogram for EO



Appendix B: Pix2Pix Architecture



Appendix C - Loss Function

The discriminator objective here is to minimize the likelihood of a negative log identifying real and fake images

$$\text{Discriminator Loss} = \frac{(\text{BCE Loss for Real Images} + \text{BCE Loss for Generated I}}{2} \quad (1)$$

$$\text{L1-Loss} = \sum_{i=1}^{n} |\text{Generated Output} - \text{Target Output}|$$

$$\text{Generator-Loss} = \text{BCE Loss with Real Labels} + \lambda \cdot \text{L1-Loss}$$

Reason for not using other popular loss functions -

1. MSE - highly sensitive to differences in pixel values, uniformity assumption of all pixels, lacks semantic understanding
2. SSIM - a good option, but restricting ourselves to quantifying structural similarity to only three metrics mean, variance