Text To Image Using DCGAN

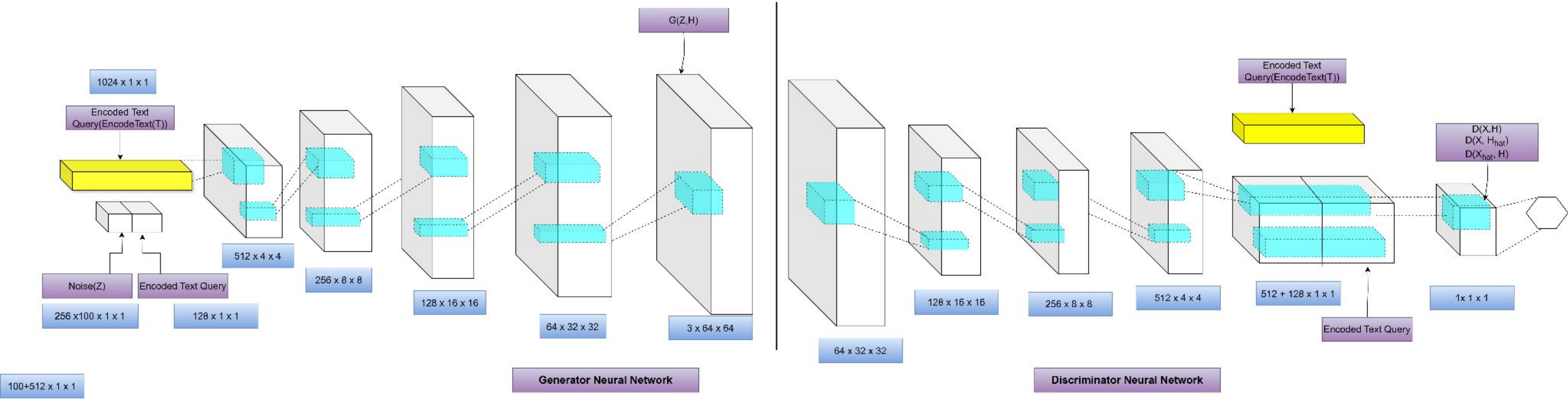
Input

Preprocessed images with embeddings in h5py format

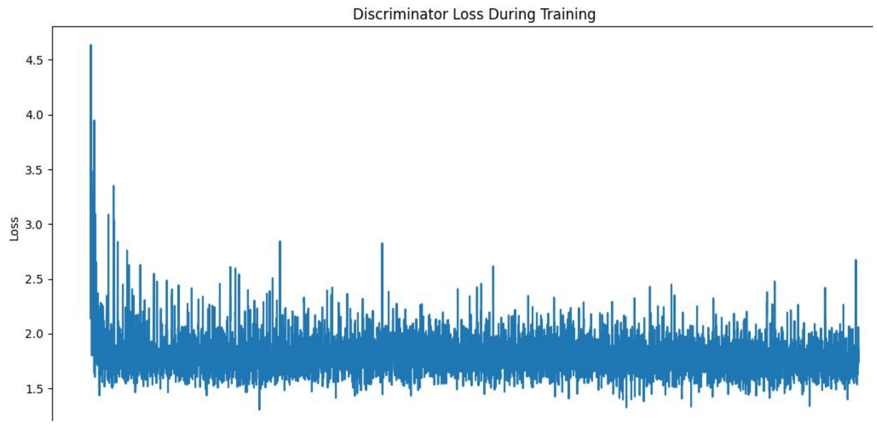
Output

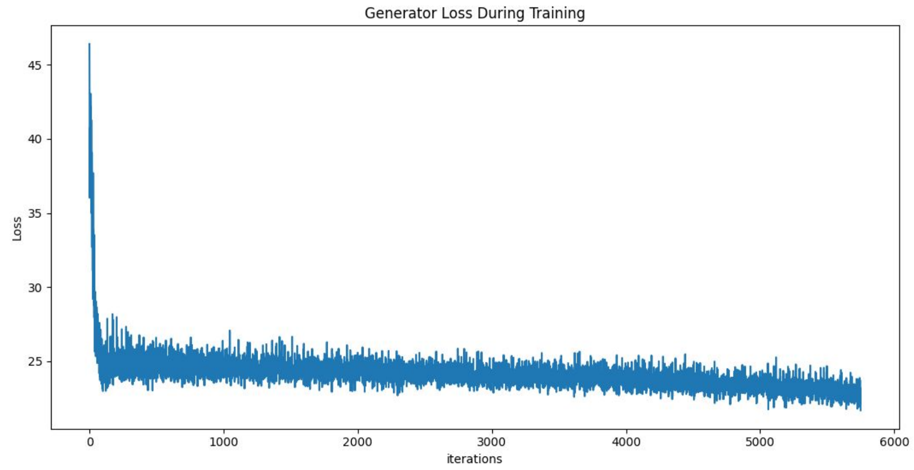
Generate images of flower that match the description present in embeddings

Architecture



Algorithm & Graphs





Algorithm : GAN Training For Text-to-Image Generation

Input: Image batch X, matching text T, mismatching text T_{hat}, batch size B, learning rate η

Output: Trained generator and discriminator

for $n = 1$ **to** B **do**
- **Encode** matching text: $H \leftarrow \text{EncodeText}(T)$
- **Encode** mismatching text: $H_{\text{hat}} \leftarrow \text{EncodeText}(T_{\text{hat}})$
- **Generate** noise: $Z \sim \text{Gaussian}(0, I)$
- **Generate** fake images: $\text{G}(Z, H)$
- **Compute** discriminator scores: $\text{D}(X, H)$ (real image with correct text)
- **Compute** discriminator scores: $\text{D}(X, H_{\text{hat}})$ (real image with incorrect text)
- **Compute** discriminator scores: $\text{D}(X_{\text{hat}}, H)$ (fake image with correct text)
- **Compute** discriminator loss: $L_D \leftarrow \log(D(X,H)) + (\log(1 - D(X_{\text{hat}}, H)) + \log(1 - D(X, H_{\text{hat}})))/2$
- **Update** discriminator parameters: $\rho_D \leftarrow \rho_D - \eta \cdot \frac{\partial L_D}{\partial \rho_D}$
- **Compute** generator loss: $L_G \leftarrow \log(D(X_{\text{hat}}, H))$
- **Update** generator parameters: $\rho_G \leftarrow \rho_G - \eta \cdot \frac{\partial L_G}{\partial \rho_G}$

end for

Reference : Generative Adversarial Text to Image Synthesis by REEDSCOT1, AKATA2, XCYAN1, LLAJAN1 SCHIELE2,HONGLAK