PROBLEM FORMULATION

SHIHAO PIAO

1. DATASET

Our dataset was split into four sections: trainX, trainY, testX, and testY. The trainX folder contains approximately 1800 authentic landscape images, while the trainY folder is populated with around 900 images in the Ghibli style, collected from various Ghibli movies such as Totoro and The Wind Rises, among others. The testX folder was filled with about 700 real landscape images for testing purposes, and the testY folder includes roughly 30 images in the Ghibli style. It's important to note that the CycleGAN dataset utilizes an unpaired format, leading to variability in the data volume across each folder.

2. ARCHITECTURE

The architecture of CycleGAN consists of two main components: generators and discriminators, with two sets of these networks to manage the translation between the two domains.

2.1. **Generator.** It starts with a 7x7 convolutional layer with reflection padding, instance normalization, and ReLU activation. Following the initial layer are the downsampling layers, each consisting of a convolutional layer that doubles the number of filters while reducing the spatial dimensions by half, using stride 2 and kernel size 3. These layers also include instance normalization and ReLU activation. Central to the generator are residual blocks. Each block consists of two convolutional layers, kernel size 3, with normalization and ReLU activation, optionally including dropout. Mirroring the downsampling, the generator includes upsampling layers that increase the spatial dimensions of the feature maps while reducing the number of filters by half, using transposed convolutional layers with stride 2 and kernel size 3, instance normalization, and ReLU activation. It concludes with another 7x7 convolutional layer and a Tanh activation function.

2.2. **Discriminator.** The architecture starts with a convolutional layer. This layer uses a kernel size of 4, a stride of 2 to reduce the spatial dimensions and padding to maintain the size of the output relative to the input. The number of filters in the first convolutional layer is 64, with the LeakyReLU activation function applied with a negative slope of 0.2. Then, followed by 3 convolutional layers. Each subsequent layer doubles the number of filters from the previous layer until a maximum is reached. The stride for these layers is set to 2 to further downsample the feature maps, except for the last layer, where the stride is reduced to 1 to adjust the output size. Each convolutional layer includes a normalization layer, defaulting to instance normalization, and is followed by LeakyReLU activation. The final layer of the discriminator is another convolutional layer that reduces the channel of the feature maps to 1 with kernel size 4 and stride 1, producing a score for each patch of the input image. This output can be interpreted as a matrix of scores, with each score indicating the likelihood that the corresponding patch is real or fake.

3. LOSS

3.1. **Adversarial Loss.** Adversarial loss makes the generated images indistinguishable from real images within each domain. This loss is applied separately for the two mappings by training two discriminators. We will use MSE in this part, the equation is shown below.

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[D(G(x))^2\right] + \mathbb{E}_{y\sim p_{\text{data}}(y)}\left[(D(y)-1)^2\right] \tag{3.1}$$

$$\mathcal{L}_{GAN}(F, D_X, Y, X) = \mathbb{E}_{y\sim p_{\text{data}}(y)}\left[D(F(y))^2\right] + \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[(F(x)-1)^2\right] \tag{3.2}$$

3.2. **Cycle Consistency Loss.** The cycle consistency loss is what differentiates CycleGAN from other GAN models, addressing the issue of unpaired training data. It ensures that an image can be translated from one domain to another and then back to the original domain without losing its original content. The equation is shown below.

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\|F(G(x)) - x\|_1\right] + \mathbb{E}_{y\sim p_{\text{data}}(y)}\left[\|G(F(y)) - y\|_1\right] \tag{3.3}$$

3.3. **Full Objective.** The total loss used to train CycleGAN is a weighted sum of the adversarial and cycle consistency losses.

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda\mathcal{L}_{cyc}(G, F) \tag{3.4}$$

4. OTHERS

For all the experiments, we will set $\lambda = 10$ in Equation 3.4. Our optimizer is Adam. All networks will train from scratch with a learning rate of 0.0002.