IMAGE GENERATION USING CYCLEGAN ARCHITECTURE.

CycleGAN is a deep learning model that can be used for image-to-image translation tasks such as converting images from one domain to another, for instance, converting horse images to zebra images. In this report, the process of training a CycleGAN model, including how the training and test data are processed, the model used, and the hyperparameters that are utilized will be discussed clearly.

The first step in training a CycleGAN model is to prepare the training and test data. The data is preprocessed to ensure that it is in a suitable format for training the model. This involves resizing images to a size of (64,64,3) and normalizing the pixel values to lie in the range [-1,1]. A large portion of the data is used for training, and for testing the model additional images were used. The images of cats and dogs are treated as one domain and the human image were the other domain. Before loading the data, files are arranged in a suitable folder structure so that the model can process the data easily using data generators.

The following hyperparameters were used while training the model. Batch_size is set to 1, *epoch* value of 10 is used and the model checkpoint helped in identifying the best model before the model starts overfitting. The generator followed *resnet* architecture with 9 residual blocks and 64 filters. For the discriminator, the convolution layer having filter size 64 and stride value (4,4) was used with *leaky Relu* as an activation function. Finally, two generators and two discriminators were combined in a model where mean squared error is used as an adversarial loss function. Additionally, all the sub-models were optimized using the Adam algorithm with a learning rate of 0.0002.

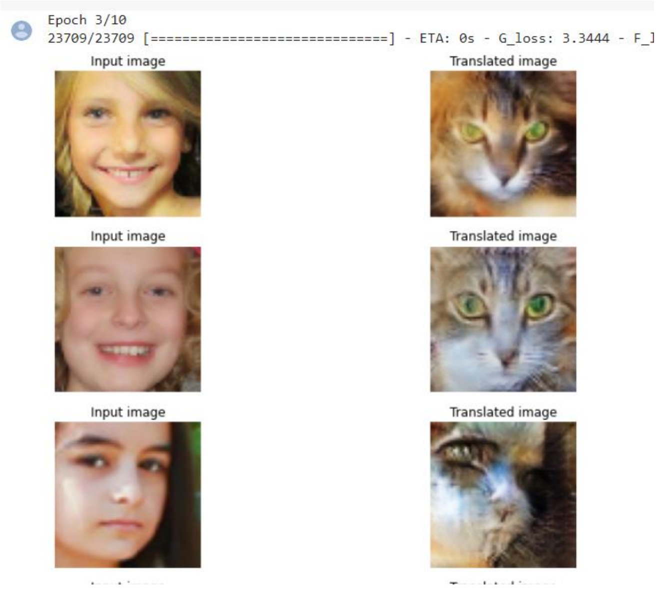
The model trained after 3rd epoch showed the best result and this can be used for generating images having the property of both classes.

In conclusion, training a CycleGAN model involved preparing the training, defining the model architecture, and setting the appropriate hyperparameters. By adjusting the hyperparameters during training, the model can be optimized to achieve the best performance on the task at hand.

Akhil Raj Rajan

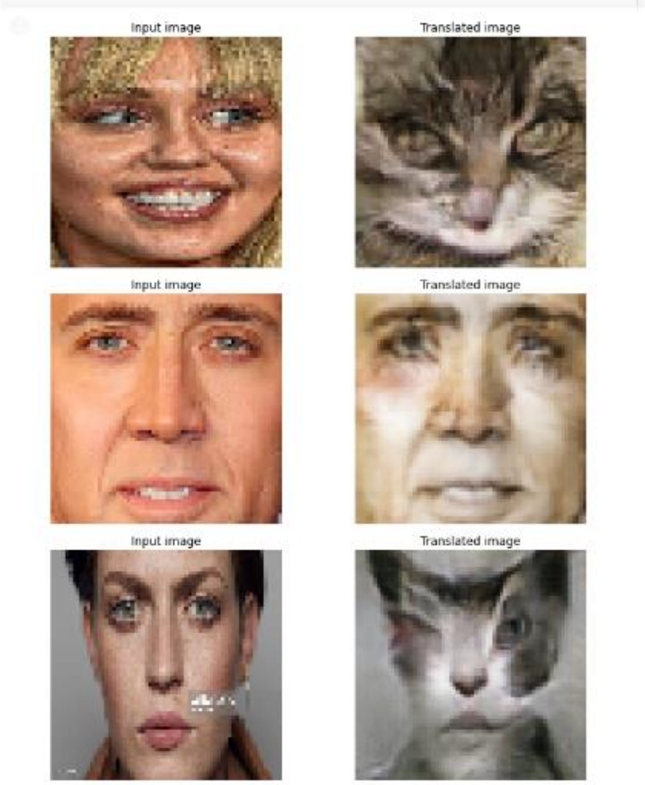
210409183

Images Generated after training.



```
Epoch 3/10
23709/23709 [==============================] - ETA: 0s - G_loss: 3.3444 - F_]
```

Images Generated During Testing



Input image
Translated image
Input image
Translated image
Input image
Translated image