# MRI GAN using U-Net Architecture - Summary Report

## Problem Statement

The goal is to build a Generative Adversarial Network (GAN) using a modified U-Net architecture to generate artificial MRI images with different contrast levels (T1-weighted and T2-weighted scans) from existing MRI scans. The GAN consists of two main components: a generator network (U-Net) and a discriminator network.

## Summary Report

### 1. Libraries Used

- NumPy
- TensorFlow
- Matplotlib
- scikit-image
- imageio
- glob

### 2. Data Loading

- MRI images from the dataset are loaded as T1 and T2 contrasts from two directories.
- A function `extract_images()` is defined to load and convert images into NumPy arrays.
- The dataset size is computed, showing a limited number of images for training.

### 3. Data Visualization

- Sample T1 and T2 images are visualized to confirm the successful loading of the dataset.

### 4. Data Preprocessing

- Normalization: Images are normalized to the range [-1.0, 1.0] using a normalization function.
- Resizing: Images are resized to 64x64 dimensions using bilinear interpolation.
- Batching: Images are shuffled and batched with a batch size of 32 for efficient training.

## 5. Model Architecture

### a. U-Net Generator

- The generator follows the U-Net architecture, consisting of:
  - Downsampling blocks: Convolutional layers that reduce spatial dimensions.
  - Upsampling blocks: Transposed convolutional layers to restore original dimensions, with skip connections.
- Instance normalization is used instead of batch normalization.

### b. Discriminator

- The discriminator network consists of convolutional layers for binary classification (real or fake).
- Instance normalization is applied after each convolution, with zero padding layers for dimensionality consistency.

## 6. Loss Functions

- Binary Cross-Entropy (BCE) Loss for both generator and discriminator.
- Cycle Consistency Loss to measure the difference between original and regenerated images.
- Identity Loss for enforcing consistency when input images are already in the target domain.

## 7. Optimizers

- Adam Optimizer with a learning rate of 2e-4 is used for both the generator and discriminator.

## 8. Model Training

- The GAN is trained for 300 epochs.
- Images are passed through both generators and cycle consistency is enforced.
- Losses are computed and optimized during each epoch.

## 9. Results Visualization

- Generated T1 to T2 images and vice versa are visualized after each epoch to monitor progress.

## 10. GIF Creation

- A GIF is created to visualize the transformation of images across training epochs.

## Conclusion

This project demonstrates the use of a CycleGAN with a U-Net-based generator to generate MRI images of different contrast levels. The model utilizes cycle consistency and identity losses to maintain the integrity of the generated images. The approach is a promising method for augmenting medical data from limited MRI datasets.