**Synthetic Iris Image Generation Using GANs**

Overview

This project aims to explore the potential of Generative Adversarial Networks (GANs) in generating realistic synthetic iris images for the purpose of biometric spoofing. The generator network learns to create synthetic iris images, while the discriminator network classifies them as real or fake. The goal is for the generator to improve over time, generating more realistic images that can fool the discriminator into classifying them as real.

**Installation**

To run this project, you'll need to set up the following environment:

**Prerequisites**

* Python 3.x
* TensorFlow / Keras for deep learning
* OpenCV for image processing
* NumPy for numerical operations
* Matplotlib for visualizations
* scikit-learn for evaluation metrics

**Installation Steps**

Install the required dependencies using pip:  
  
pip install tensorflow opencv-python numpy matplotlib scikit-learn

1. Upload your dataset (a zip file) containing iris images to Google Colab or your local environment. The dataset should consist of images organized into subfolders, where each subfolder represents a different iris class with 6 images per class.

**Dataset**

The dataset for this project should consist of iris images in a folder structure, where:

* Each subfolder represents a unique class of iris images.
* Each subfolder contains 6 iris images.

The dataset is expected to be uploaded in a ZIP file (e.g., dataa.zip). Upon extraction, it will be used for training the GAN to generate synthetic iris images.

**Execution**

**Data Preprocessing**

Before training the GAN, the iris images are loaded and preprocessed using the DataGenerator class:

* Resized to 64x64 pixels.
* Converted to grayscale.
* Normalized to a range of [-1, 1].

This ensures that the images are ready for the GAN model.

**Model Architecture**

The project uses the standard GAN architecture:

* Generator: A neural network that generates synthetic iris images from random noise.
* Discriminator: A neural network that classifies images as real (from the dataset) or fake (generated by the generator).
* GAN: Combines the generator and discriminator for adversarial training, where the generator tries to fool the discriminator, and the discriminator tries to classify the images correctly.

The models are built using Keras with TensorFlow backend.

**Training**

The training process involves the following:

1. Train the Discriminator: The discriminator is trained on both real images (from the dataset) and fake images (generated by the generator).
2. Train the Generator: The generator is trained to create fake images that fool the discriminator into classifying them as real.

**Evaluation**

Once training is complete, the generator is evaluated based on how well it can fool the discriminator. We generate synthetic images using random noise and then classify them using the discriminator. The generator’s performance is assessed, which provides the following metrics:

* Precision
* Recall
* F1-Score
* Accuracy

These metrics help evaluate the quality of the generated images.

The discriminator’s performance is evaluated separately based on its ability to classify real and fake images, and the loss and accuracy are tracked throughout the training.

**Results**

After training, the GAN will generate synthetic iris images. These images will be stored in the generated\_images/ folder, and the quality of the generated images will be visually inspected at regular intervals.