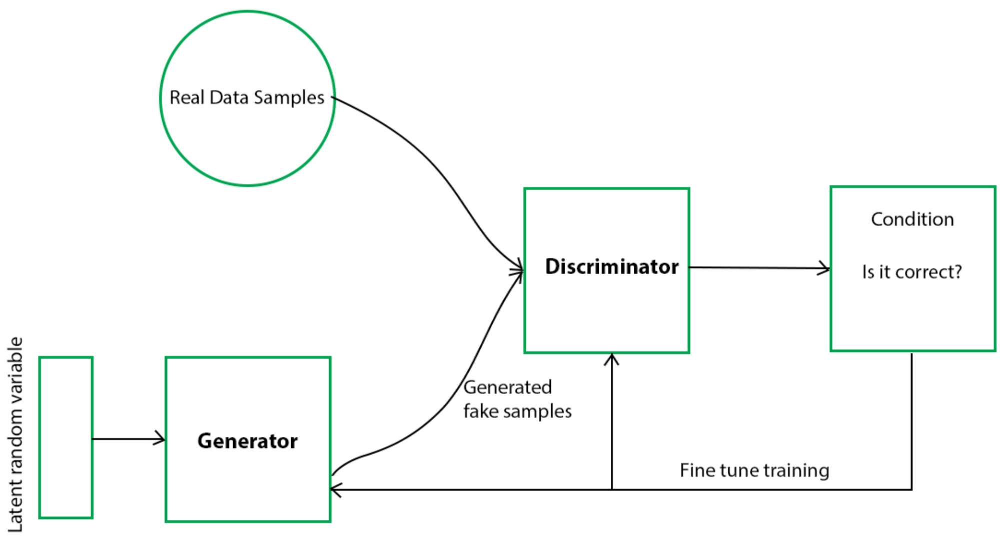
Final Report

Goal of the project

To generate synthetic iris images , that can later be used as training data for the MOSIP verification system.

Approach Used

GANs



Generative Adversarial Networks (GANs) are machine learning models consisting of a generator and a discriminator. The generator creates samples that resemble a target dataset, while the discriminator tries to distinguish between real and generated samples. Through an adversarial training process, the generator and discriminator compete and improve over time.

Deep Convolutional GANs (DCGANs) are a type of GAN specifically designed for image generation. They utilize convolutional neural networks (CNNs) in both the generator and discriminator networks to capture spatial features in images. DCGANs employ strided and transposed convolutions for downsampling and upsampling, respectively, allowing them to generate high-resolution images. Batch normalization is often used to stabilize training.

DCGANs have achieved remarkable results in generating realistic and high-quality images and are widely used in various image-related tasks.

In this project we have mainly focused on using DCGANs to generate new and synthetic iris images. We were able to generate iris images with resolution 64 x 64 and 128 x 128.

Datasets

CASIA - Thousand :

The CASIA1000 iris dataset contains iris images from 1,000 subjects, with each subject contributing multiple images. The images are captured under various conditions, including different illumination levels, occlusions, and rotations. The dataset provides a diverse range of iris patterns and image quality, making it suitable for assessing the performance of iris recognition algorithms under challenging scenarios.

IIT - Delhi :

The IIT Delhi Iris Database mainly consists of the iris images collected from the students and staff at IIT Delhi, New Delhi, India.

Architecture and Hyperparameters

The following link contains the code to generate 64 x 64 iris images . Please refer to it to understand the various hyperparameters used and the architecture of the generator and discriminator networks.
https://www.kaggle.com/code/soorajsathish/mosip-gans

This link contains the code for generating 128 x 128 images.

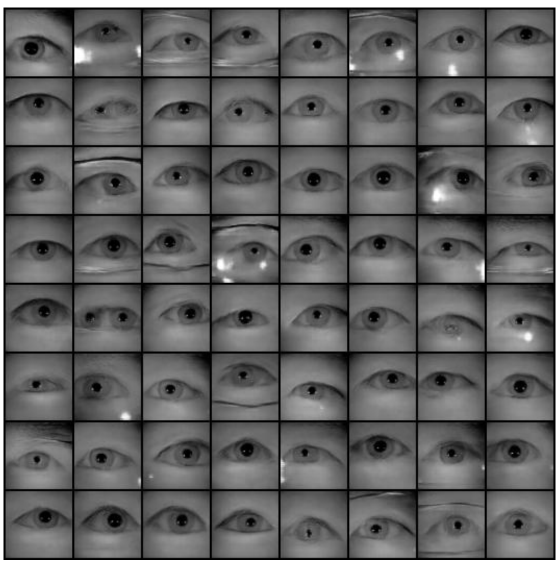
https://www.kaggle.com/code/sumanth2002iiitb/mosip-gans-128

Note: We have made some changes to the first code as we were trying out different loss functions . Please refer to the second link for the original loss functions.

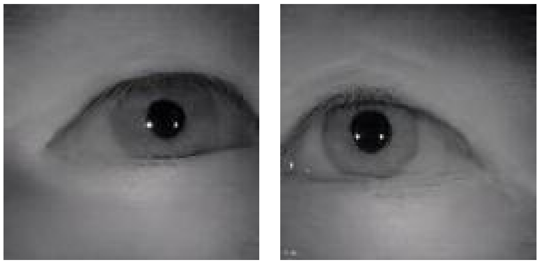
To run the code , open these in kaggle and press run all. You will be able to view the generated images in the output folder.

Results

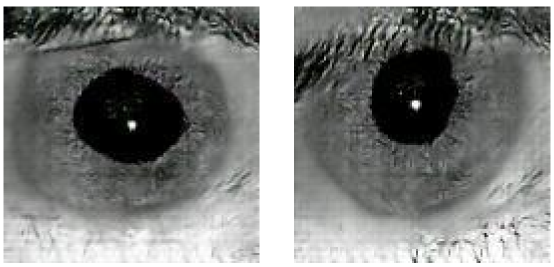
64 x 64 images : Networks trained on CASIA1000



128 x 128 images : Networks trained on CASIA1000



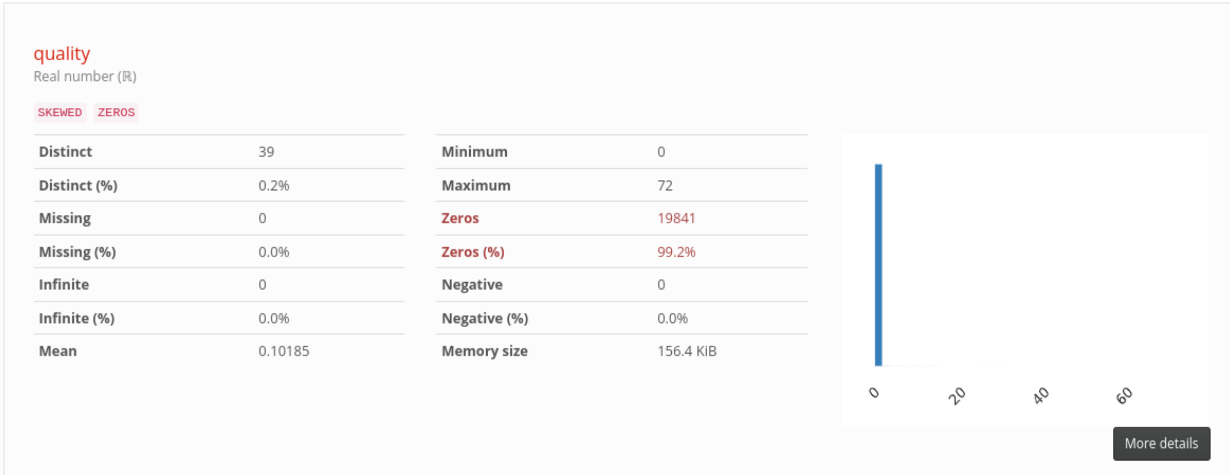
128 x 128 images : Networks trained on IIT Delhi dataset



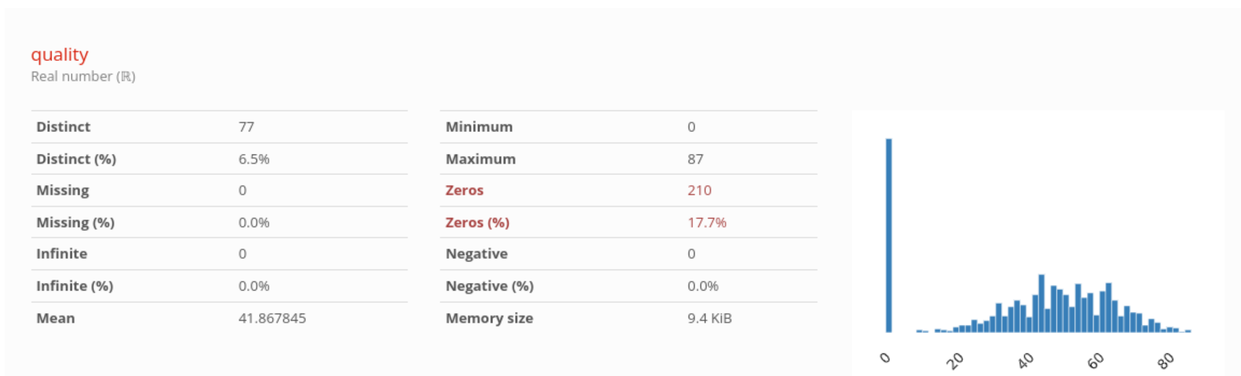
Analysis

For the analysis of the generated iris images we have used the BQAT api.(https://github.com/Biometix.)

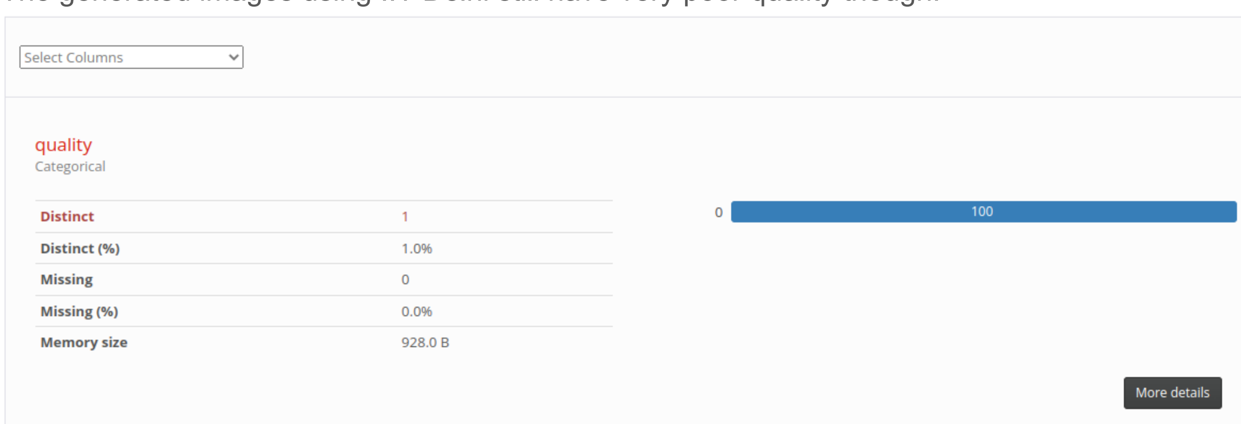
1. The original CASIA1000 dataset doesn't seem to have very high quality images. We realized this only later after running all the code using this



2. Thus we shifted to using the IIT Delhi dataset. This seems to have better quality images but still has a lot of low quality images.



3. The generated images using IIT Delhi still have very poor quality though.



4. Even after removing the poor quality images from the IIT Delhi dataset and then training the networks , we obtained poor results.

Future Scope / Next Steps

1. We can see in the analysis that the iris quality is 0 for all the images that are generated by a GAN trained using IITD-Dataset. Hence, we need to understand what metrics the biometix tool(or any other tool) is taking into account to measure the quality of the iris and train the model in a way that enhances the output quality.

2. One approach we experimented with is RASGAN(https://www.kaggle.com/code/soorajsathish/mosip-gans) which has been introduced in a CVPR paper(https://arxiv.org/abs/1812.04822) which actually modifies the loss function of the actual GAN. But our implementation isn't working because of the diminishing gradients problem. This could be made to work by choosing appropriate architecture like adding skip connections and experimenting with different activation functions.

3. The above-mentioned paper uses Frechet Inception distance(FID) to measure the iris quality. Studying FID and implementing RASGAN that generates iris with promising FID score will help the MOSIP to test the iris verifying SDKs.