Generation of iris strips patterns using Gang

1st Jose Corona
dept. Robotics and Computer Vision
Innopolis University
Kazan, Russia
j.coronaboch@innopolis.university

***Abstract*—This paper focus using GAN based methods to generate iris strips to allow better training of models. In this work it were implemented the popular StyleGAN2 framework for iris strip image generation in Google Colabs, to check that the GAN model could generate fake iris patterns with quality and diversity it was calculated the FID-Score, and the model get a high FID-Score. Also it were implemented a Super Resolution GAN to scale up x4 the iris images.**

***Index Terms*—GAN, Iris strips patterns, FID-Score, StyleGAN2, Google Colab**

I. Introduction

Biometric authentication now a days is more common on devices like smartphones, is based on physiological characteristics of the person such as fingerprints, face and iris patterns. Biometric security systems refer to an recognition algorithm to get access to a system or devices. Security systems based on Iris recognition has been developed because iris images has unique patterns,they are complex, it can easily be sampled and it is protected from the environment. [1]

Some problems to deal with while acquiring high quality iris image are: reflected light in the eyes, the irradiation near infrared light and external noise. Filtering techniques like smoothing and sharpening are very efficient to remove noise in the iris image. The iris image after the filtering process may vary in some information on thickness of the iris strips and some patterns may disappear. [2]

To increase iris recognition rates after noise reduction filtering super-resolution techniques are applied. A super resolution image could be archive using Generative adversarial networks (GANs) to generate the image. GAN is a class of machine learning framework, use two neural network models, one of them is a generator of data withe the same distribution as the training set, and the other one is a discriminator that learns to classify between the original data and the fake one generated. GANs could learns to generate new data, given a training set, with the same characteristics and distribution as the original. [3]

The objective of this study was to increase the amount of iris strip images available using GAN based methods; generate new iris patterns, to allow training the model with more data to get better accuracy of iris recognition techniques. Also to implement a Super Resolution GAN to scale iris images to x4 and with this overcome the problem of images taken not near the eye.

II. Related Work

The idea of using GANs for generating more data from a given set, to train models with a bigger number of samples, is not new, and since the invention of GANs there have been several works trying to improve them. Some works are try to improve GANs by changing the Loss funtion, other ones try to change the architecture of the neural networks.

Leding present a generative adversarial network (GAN) for image super-resolution (SR), know as SRGAN. The basic idea is to increase the resolution of a image, the super-resolution image looks very similar to the original. The framework generate photo-realistic natural images with a up-scaling factors of 4x. They focus on the use of a perceptual loss function, which is a combination of an adversarial loss and a content loss. [3]

To get many other works about GAN frameworks and models refer to [5] for a detailed list.

A related work on biometrics Iris recognition algorithm using Iris images and GAN networks was develop by Shervin Minaee. They develop a machine learning framework using GANs. The framework is able to generate iris images sampled from two databases, CASIA-1000 and IIT-Delhi. The images generated where very similar to the images in this databases. The model of convolution network they use for the generator and the discriminator contain 5 layers, each layer was followed by a batch normalization non-linearity layer. To measure the diversity of the generated iris image, they use the Frechet Inception distance (FID), and they get a high FID score, showing they have good diversity. [2]

Other work related to iris recognition using deep convolutional neural networks (DCNNs) and SRGANs was developed by Koji Kashihara. He focused in the effecs of SRGANs using iris images with low signal-to-noise ratio and other types of external noise and some prefiltering process. Because biometrics systems get the images by camera, and some cameras and photographic environment could cause a low signal-to-noise ratio. He use iris images from the UBIRIS.v1 databese. He develop a SRGANs framework that was able to restore the images, and a DCNNs that could accurately predict the individual iris patterns. He use a pixel-based differences for the classifier for biometrics. [4]

III. Implemented Model

Git Hub link of the Proyect. Project was developed in Google Colabs, the 4 databases where uploaded to my google drive.
https://github.com/Jose-R-Corona/Iris-GAN-ProyectCV

In this project where used 4 databases. Three databases from the Center for Biometrics and SecurityResearch (CASIA). And the other databases is form IIT Delhi.

From CASIA I get access to 'CASIA Iris Image Database (version 1.0)', 'CASIA 2 Device 1' and 'CASIA 2 Device 2'. CASIA Iris v1 (Fig No.1) contain 756 images with a size of 320x280. CASIA 2 Device 1 (Fig No.2) contain 1213 images and CASIA 2 Device 2 (Fig No.3)contain 1200 images with a dimension of 640x480.

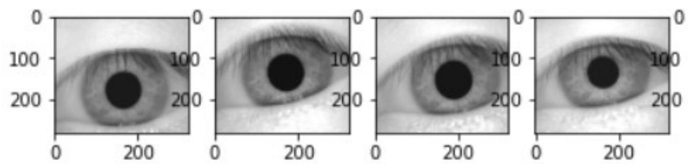

Fig. 1. CASIA Iris Image Database (version 1.0)

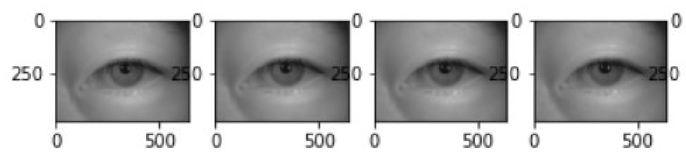

Fig. 2. CASIA 2 Device 1

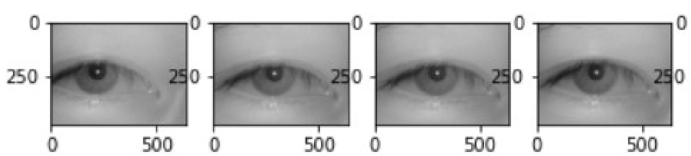

Fig. 3. CASIA 2 Device 2

The IIT Delhi v1 (Fig No.4) database contain 2240 images with a size of 320x240.

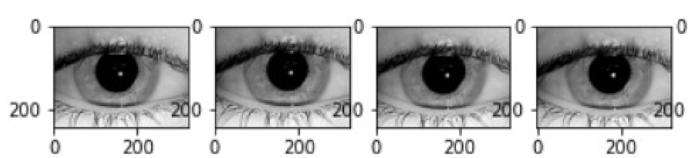

Fig. 4. IIT Delhi v1

To implement GAN to generate fake images, the Style-GAN2 framework from Nvidia [7] was used. It was introduced in 2019 and it main purpose was to generate portraits of fake human faces (Fig No.5). But the framework could be trained to generate other types of fake images. There are plenty of examples in which they train the framework to generate fake images of cars, watches, drawing characters, etc.

Fig. 5. Fake portrays generated by StyleGaN2 [7]

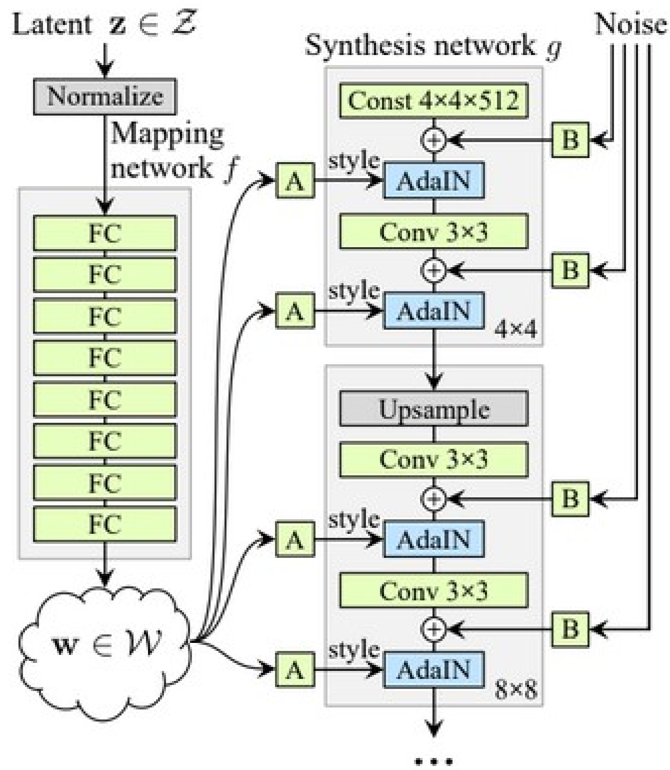

Fig. 6. Structure of StyleGAN2 [7]

The framework is free and is available in GitHub [6]. To use this framework is need to resize all images and covert them to JPEG format (Databases images have BMP format). Them is need to use a python function provided by NVIDIA to transform JPEG images to "tfrecords" format, this is a format special of tenserflow to train models.

After implement StyleGan2 framework in Google Colab with out any error, the model was trained just with the custom dataset, iris images. It were need at least 1 day of training to produce fake images very similar to the original ones.

For the super resolution GAN (SRGAN) it where need to downgrate the images to a scale of 64x64, and the high resolution was 256x256 pixels. And the dataset where divided into two parts, one for training and other for testing the accuracy and loss of the model.

The Super Resolution GAN use the same architecture as a GAN, with a generator and a discriminator, as in Fig No. 7. The generator produces a output from a given input donwgreated, it means the input layer of the generator is of size 64. The discriminator receives two types of data, the original high resolution and the generated image of the generator, and the discriminator classify as 1 if is real the image or 0 if is a fake image. After training the generator improves and the discriminator performance gets worse because there little difference between the real and generated fake image.

In Fig No. 8 were presented the model of the generator and discriminator for the SRGAN Generator implemented in Google Colab. In this implementation of SRGAN, in the generator section, were added a pretrained VGG19 before the discriminator to extract features from the image. For training the generator and discriminator take turns, not but at the same time, since both model were attached we have to freeze the discriminator. The VGG19 model is always frozen. In Fig No. 9 were presented the model implemented in tensorflow - keras.

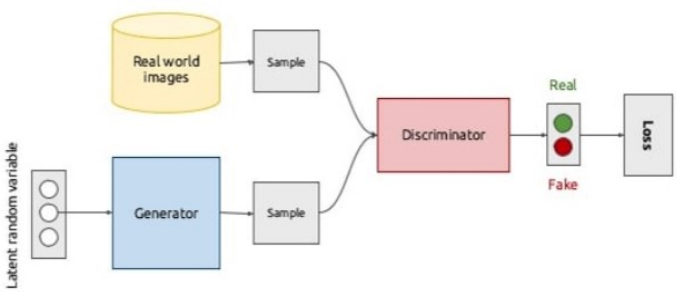

Fig. 7. SRGAN Basic model

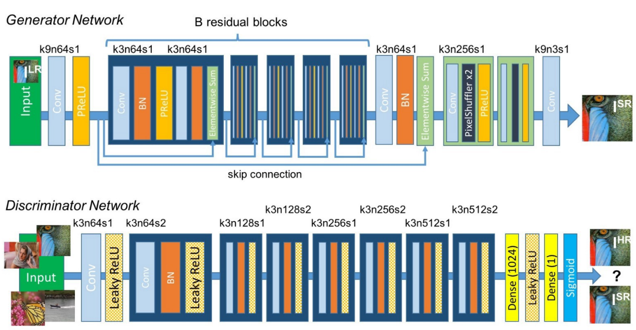

Fig. 8. Model of the SRGAN Generator and discriminator

IV. Experiments and Evaluation

In Fig No.5 we could see the data images after converting to ”tfrecords” format. To train the StyleGan2 framework in Google Colab, we have to select the option of number of gpus equal to 1. For this project it was select a number of iteration of 20000 for each ”tick”. We could see in Fig No.11 fakes images generated after the first tick, this model is very slow

```
################################################################################
Model: "model_3"
________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
================================================================================
input_1 (InputLayer)            [(None, 64, 64, 3)]  0
________________________________________________________________________________
model (Functional)              (None, 256, 256, 3)  2044291     input_1[0][0]
________________________________________________________________________________
input_2 (InputLayer)            [(None, 256, 256, 3) 0
________________________________________________________________________________
model_1 (Functional)            (None, 1)            138912577   model[0][0]
________________________________________________________________________________
model_2 (Functional)            (None, 64, 64, 256)  1735488     model[0][0]
================================================================================
Total params: 142,692,356
Trainable params: 2,040,067
Non-trainable params: 140,652,289
________________________________________________________________________________
```

Fig. 9. Model of the SRGAN implement in tensor flow - keras

to learn the patterns, the first tick take 36 minutes and the fakes images generated were no good.

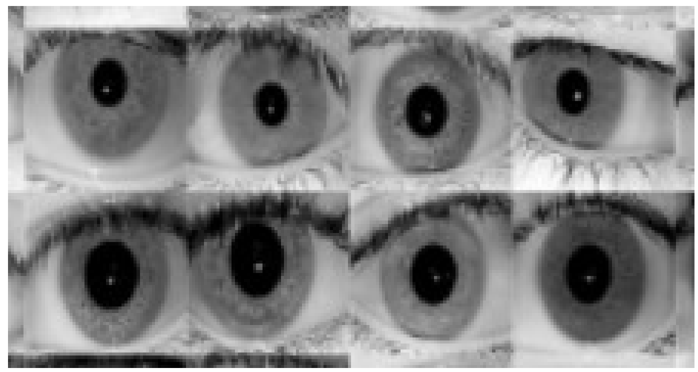

Fig. 10. Training data after converting to tfrecords

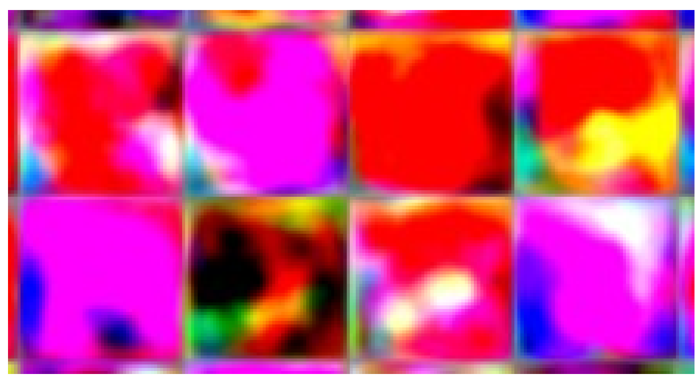

Fig. 11. Fakes images using StyleGAN2 model after 1 tick of 20000 kimg

After 15 ticks we could see the StyleGAN2 model drastically improve the fake images generated. In Fig No.12 were presented the generated fake images after 50 ticks, each tick take 36 to 30 minutes. The fake images generated were very similar to the original dataset, in the Figure No.13 we could see a image generated using a seed of 3875999, we could see that the model could generated the iris strip patterns, so we could use this fake images to train CNN models for biometric authentication.

After training the model for 50 ticks, it could garnered very good images with iris strips. To measure numerically the quality and diversity of the model, we could use the Frechet

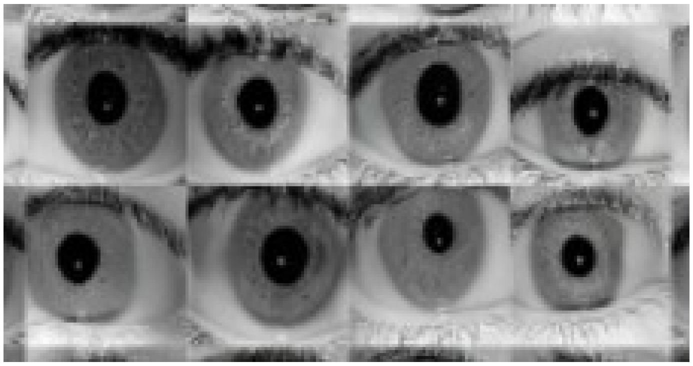

Fig. 12. Fakes images using StyleGAN2 model after 50 ticks of 20000 kimg

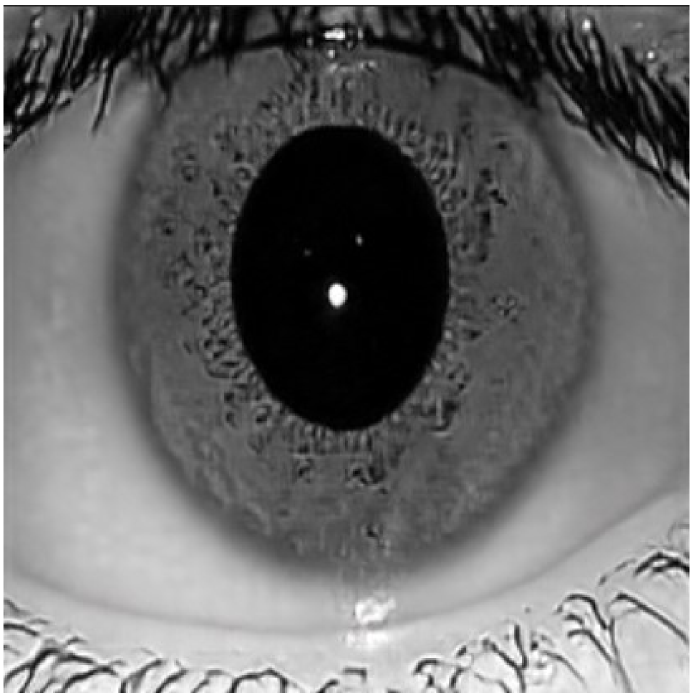

Fig. 13. Fake image generated with StyleGAN2 model using seed 3875999

Inception Distance (FID) on the fake iris images generated by the model. This score is an extension of Inception Score (IS), but FID is more popular to be applied in GAN models. FID use and compares the statistics of generated samples to real samples, using the Frechet distance for two multivariate Gaussians distributions. [8]. Very good GAN models could get a value of 50 FID-Score. In Fig No.14 is presented the value of FID-score achieved with the model, it was 39 and take 11 minutes to calculated it, so the StyleGAN2 model could generate fake iris patterns with quality and diversity.

```
network-snapshot-010547        time 11m 59s      fid50k 39.8445
```

Fig. 14. StyleGAN2 FID-Score after 50 ticks of 20000 kimg

For the Super Resolution GAN, it also implemented in Google Colab with tensorflow keras. In Fig No.15 we could see the comparison between the input low resolution image, the generated image and the original high resolution training image. We could see that the generated is not so bluer as the low resolution image but is not like the high resolution image. So the model need more training and more data to perform better.

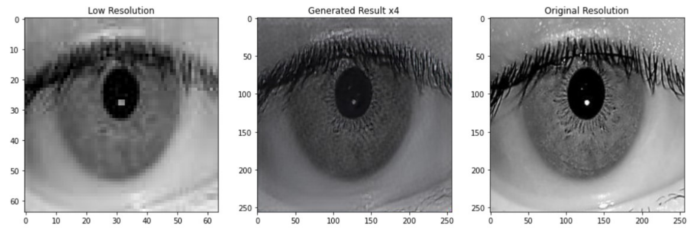

Fig. 15. Comparison between the input low resolution, the generated and the original high resolution training image

To evaluate numerically the performance of the model, it were calculated the accuracy and the loss on the test dataset, in Fig No.16 we could see the values achieved. And in Fig No.17 we could see another comparison between the input low resolution, the generated and the original high resolution using images from the training datasset.

```
Test dataset loss and accuracy: [14.092989921569824, 2.1272333583510772e-07, 14.092989921569824]
```

Fig. 16. SRGAN loss and accuracy achieved in the test dataset

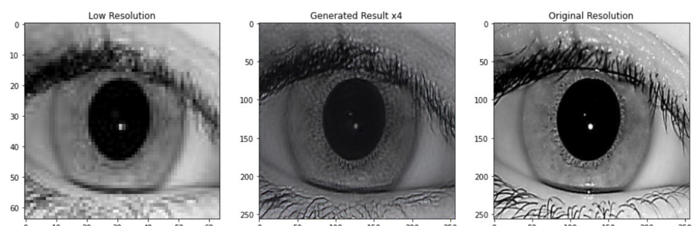

Fig. 17. Comparison between the input low resolution, the generated and the original high resolution training image

V. Analysis and Observations

In this work it were implemented the StyleGAN2 framework for iris strip image generation, this model is a deep convolutional generative adversarial networks (GAN). The model were trained on popular iris databases, and the model after 50 ticks could generated realistic iris strip patterns that look similar to real iris images.The model could achieve a FID-Score of 39, meaning that the model could generate fake iris patterns with quality and diversity.

Super Resolution GAN could achieve scaling factors up to 4x having good resolution. To train the generator we need to decrease to a low resolution the images and for the discriminator model is trained with the original high-resolution image and the output of the generator. And is need a pretrained VGG19 model for features extraction while training in the generator.

VI. LINK PROYECT GITHUB

https://github.com/Jose-R-Corona/Iris-GAN-ProyectCV

ACKNOWLEDGMENT

I would like to thank Ian Goodfellow for his work on GANs and the team of Nvidia for his work on StyleGAN2 framework. Also I would like to thank Center for Biometrics and Security Research (CASIA) and IIT Delhi for the provided Iris image database.

REFERENCES

[1] R. Saini "Comparison of various biometric methods," International Journal of Advances in Science and Technology, vol. 2, no. 1, 2, 2014.
[2] S. Minaee, A. Abdolrashidi, Iris-GAN: Learning to Generate Realistic Iris Images Using Convolutional GAN, CoRR, abs/1812.04822, 2018.
[3] C. Ledig, L. Theis, "Photo-realistic single image super-resolution using a generative adversarial network," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4681–4690, 2017.
[4] K. Kashihara, "Iris Recognition for Biometrics Based on CNN with Super-resolution GAN," 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), Bari, Italy, 2020, pp. 1-6.
[5] https://github.com/hindupuravinash/the-gan-zoo
[6] https://github.com/NVlabs/stylegan2.git
[7] T. Karras, "A Style-Based Generator Architecture for Generative Adversarial Networks", 2019, arXiv, eprint 1812.04948
[8] T. Salimans, I. Goodfellow, W. Zaremba and X Chen. "Improved techniques for training gan", Advances in Neural Information Processing Systems, pp. 2234-2242, 2016.