

# NFT Collections Generator

## with Computer Vision



**Vincenzo Imperati 1834930**



hi all, this is my project: "NFT collections generators with computer vision". the aim in the project is to generate new nfts starting from already existing nft collections

-----  
ciao a tutti, questo e' il mio progetto: "NFT collections generators with computer vision". lo scopo nel progetto e' quello di riuscire a generare nuovi nft partendo da collection nft gia' esistenti

## 1. Generative Adversarial Network

### 2. Generate NFTs

- **Model**
  - **DCGAN**
  - and
  - **SRGAN**
- **Final results**

### 3. Blend NFTs

- **Model**
  - **DCGAN-2D**
- **Final results**

This is possible thanks to the use of advanced computer vision techniques such as the use of gan, or generative adversarial networks, a type of unsupervised reinforcement learning model capable of generating new images starting from an initial dataset.

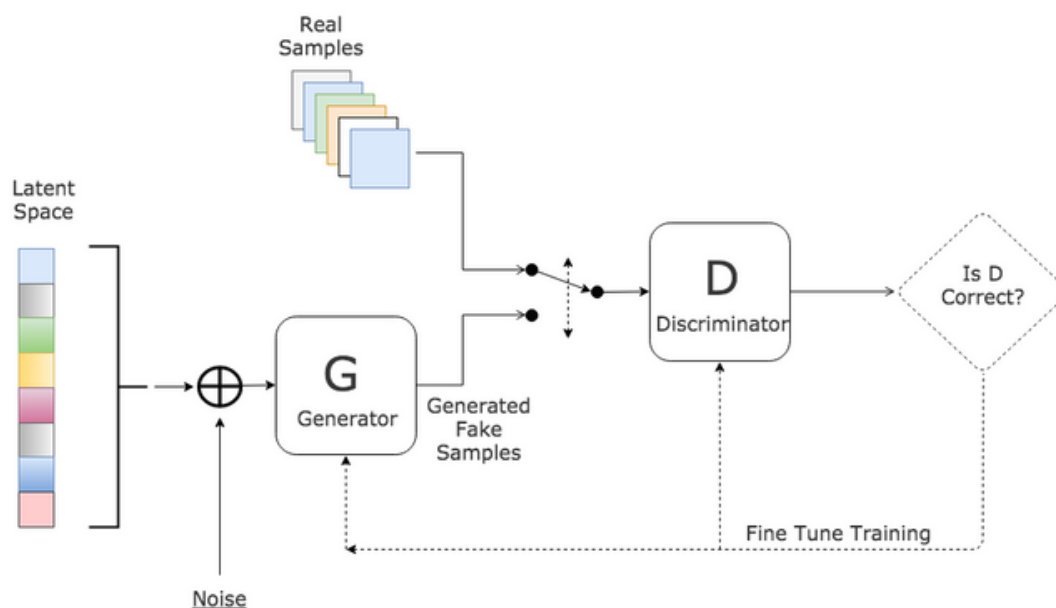
during the presentation I will explain the architecture of a gan, I will mainly talk about how I generated new nfts and finally I will talk about an attempt made to blend two different nfts

-----  
Questo e' possibile grazie all'uso di tecniche di computer vision avanzate come l'utilizzo delle gan, ovvero le generative adversarial network, un tipo di modello di apprendimento per rinforzo non supervisionato capace di generare nuove immagini partendo da un dataset iniziale.

nel corso della presentazione spieghero' l'architettura di una gan, parlero principalmente di come ho generato nuovi nfts e infine parlero' di un tentativo fatto per fare il blend di due nft diversi

# 1. Generative Adversarial Network





the generative adversarial network consists of two neural networks which are the generator and the discriminator which compete against each other. the generator has the task of generating an output as similar as possible to the real dataset and the discriminator must evaluate if his input came from real dataset or generator output. the generator is therefore indirectly trained from the discriminator.

as you can see from the image, the generator generates new outputs starting from the latent space. so, the generator learns to map any input into an output similar to the real dataset.

the discriminator then judges the output of the generator and finally the back forward is useful to train the models

la generative adversarial network si compone di due reti neurali che sono il generator e il discriminator che competono una contro l'altra. il generator ha in compito di generare un output quanto più possibile fedele al dataset reale e il discriminator deve giudicare se quello che gli viene dato in input è reale oppure fake, quindi generato dal generator. il generator quindi viene trainato indirettamente dal discriminator.

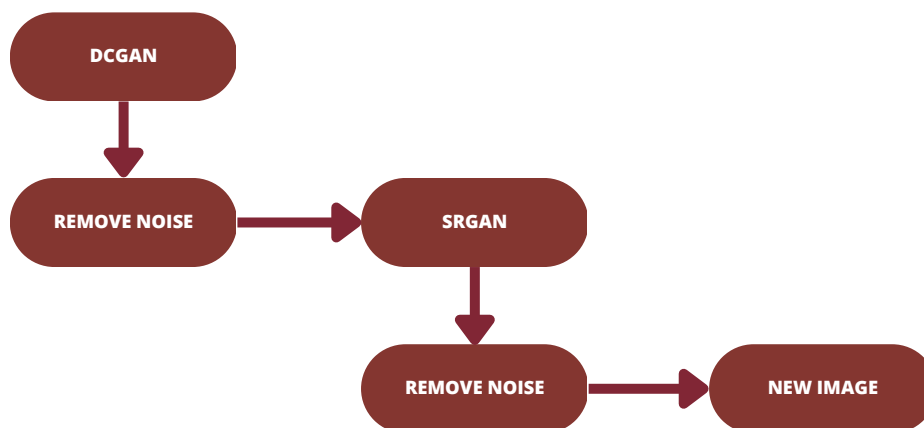
come si vede dall'immagine, il generator genera nuovi output partendo dallo spazio latente. questa cosa è possibile perché il generator impara a mappare qualsiasi input in un output simile al dataset reale.

il generato poi giudica l'output del generator e infine avviene il back forward per trainare i modelli

## 2. Generate NFTs

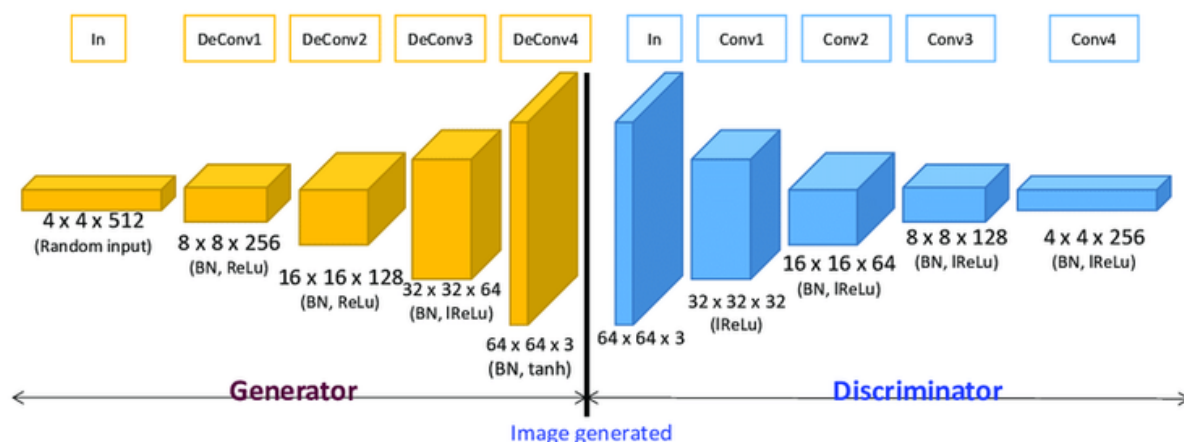
- Model
  - DCGAN
  - and
  - SRGAN
- Final results





to generate new nfts I chose to use a DCGAN+SRGAN. the output of the deep convolutional gan, after the denoising process, goes in input to a super resolution gan capable of increasing the input resolution by 2 and 4 times. after the srgan we denoising again in order to get the final image

-----  
per generare nuovi nfts ho scelto di usare una deep convolutional gan, a seguito di questa, l'output viene denoised per poi passare in input ad una super resolution gan capace di aumentare la risoluzione dell'input di 2 e 4 volte. dopo la srgan si passa nuovamente alla cancellazione del rumore per poi ottenere l'immagine finale



the dcgan is a gan where the generator is a convolutional neural network, so the architecture appears as in the figure.

in fact the generator generates an image of 64x64 pixels with three channels and the discriminator digests it.

the real image dataset must be 64x64 pixels

la dcgan e' una gan dove il generator e' una convolutional neural network, quindi l'architettura appare come in figura.

infatti il generator genera un'immagine di 64x64 pixels con tre canali e il discriminator la digerisce.

il dataset delle immagini reali deve essere di 64x64 pixels



these are the results of the dcgan trained on the BAYC collection. the dataset consists of 10k different images characterized by different traits.

the training took 1 hour for 1000 epochs.

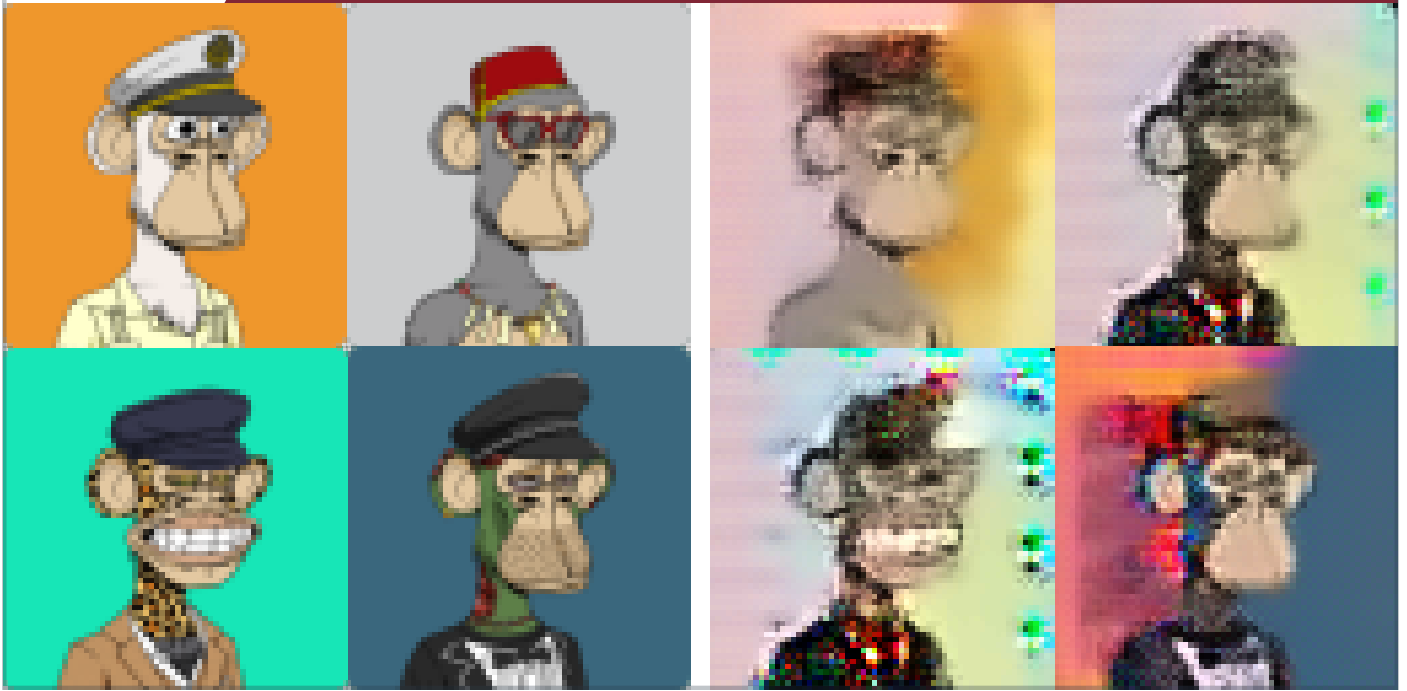
we can see how due to the remarkable contrast of the backgrounds the model did not learn well. this is because, for example, the model knows that blue and green are two good colors for the background, but he doesn't quite understand that one excludes the other.

-----questi sono i risultati della dcgan trainata sulla collection delle BAYC. il dataset e' di 10k immagini differenti caratterizzate da diversi tratti.

il training ha impiegato 1 ora per 1000 epoche.

possiamo notare come a causa nel netto contrasto degli sfondi il modello non ha imparato bene. questo perche' per il modello, ad esempio, sa che il blu e il verde sono due colori buoni per lo sfondo, ma non capisce bene che uno esclude l'altro.

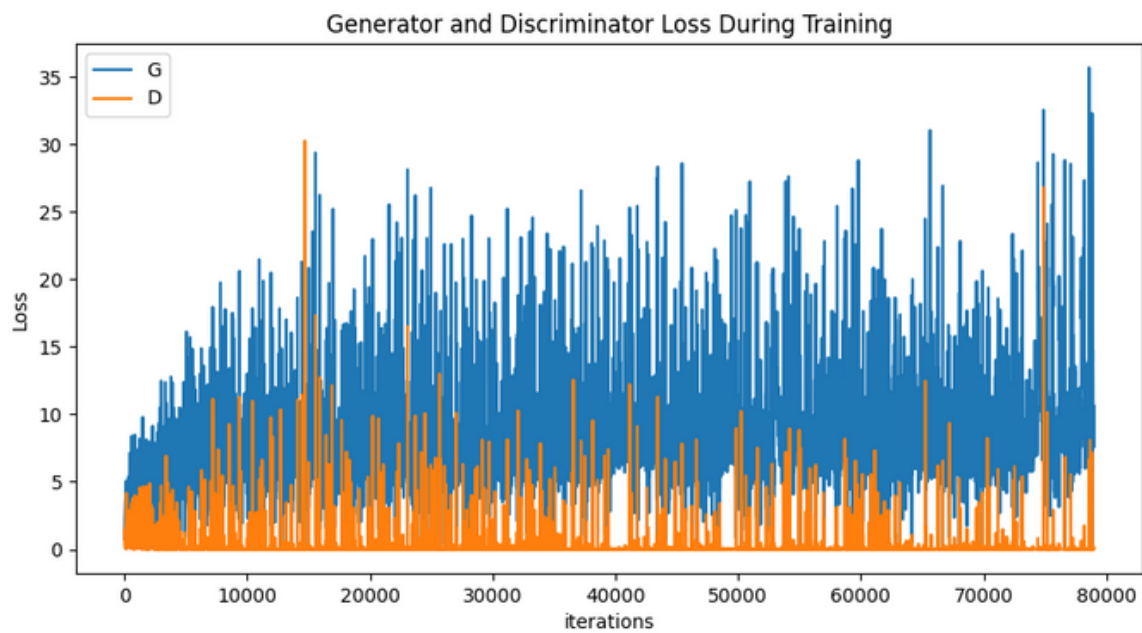




here we have a few more examples in detail

-----

qui abbiamo qualche altro esempio nel dettaglio



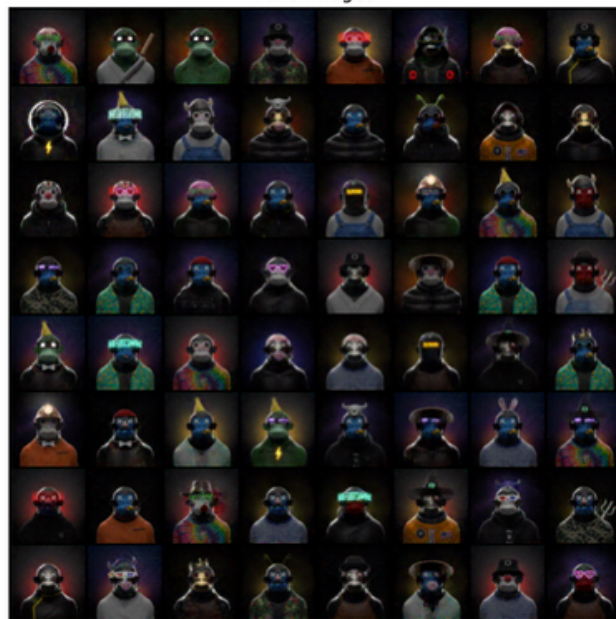
here we have the graph of the generator and discriminator loss

-----  
qui abbiamo il grafico del loss di generator e discriminator

Real Images



Fake Images

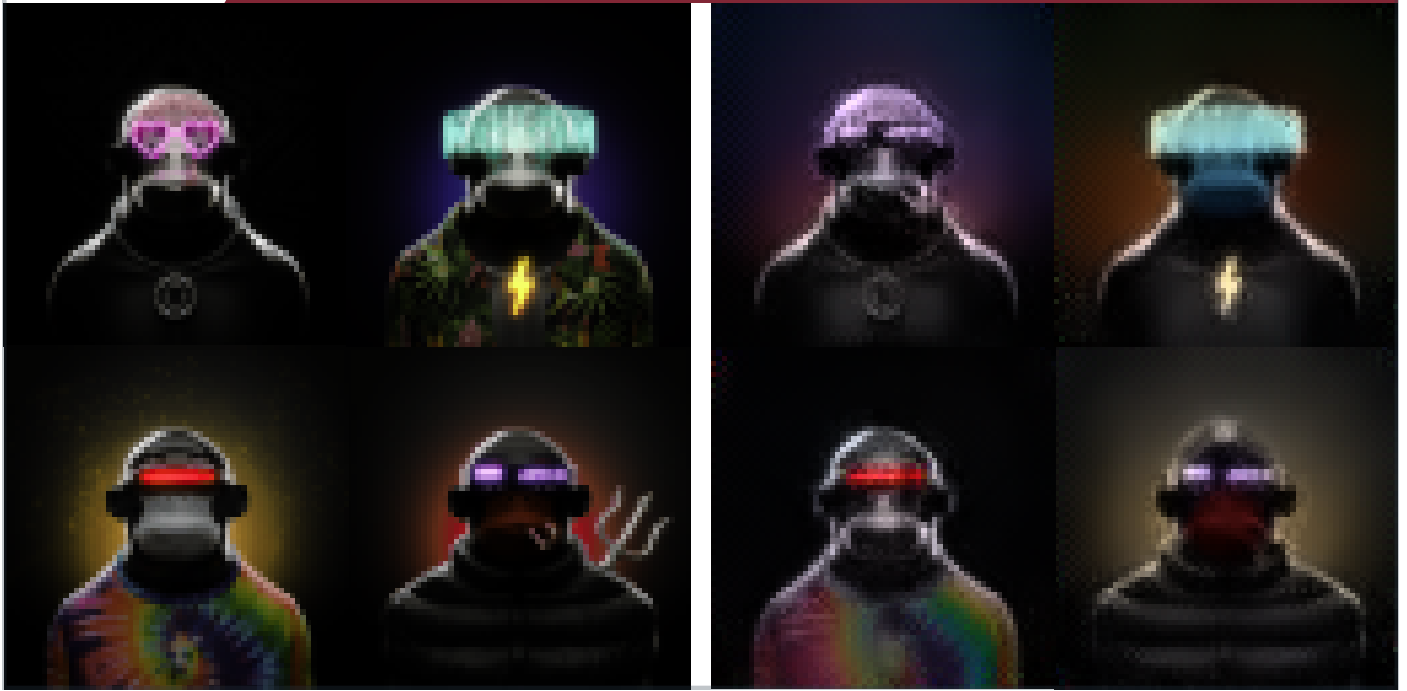


these instead are the results of the training of the model on another nft collection. these are the EAPES, they are always 10k different images and the model has always been trained in an hour for 1000 epochs.

here we note that since the backgrounds are always dark, the model learned earlier and better

-----  
questi invece sono i risultati del training del modello su un'altra collection nft. queste sono le EAPES, sono sempre 10k immagini differenti e il modello è sempre stato trainato in un'ora per 1000 epoche.

qui notiamo che siccome gli sfondi sono sempre scuri il modello ha imparato prima e meglio

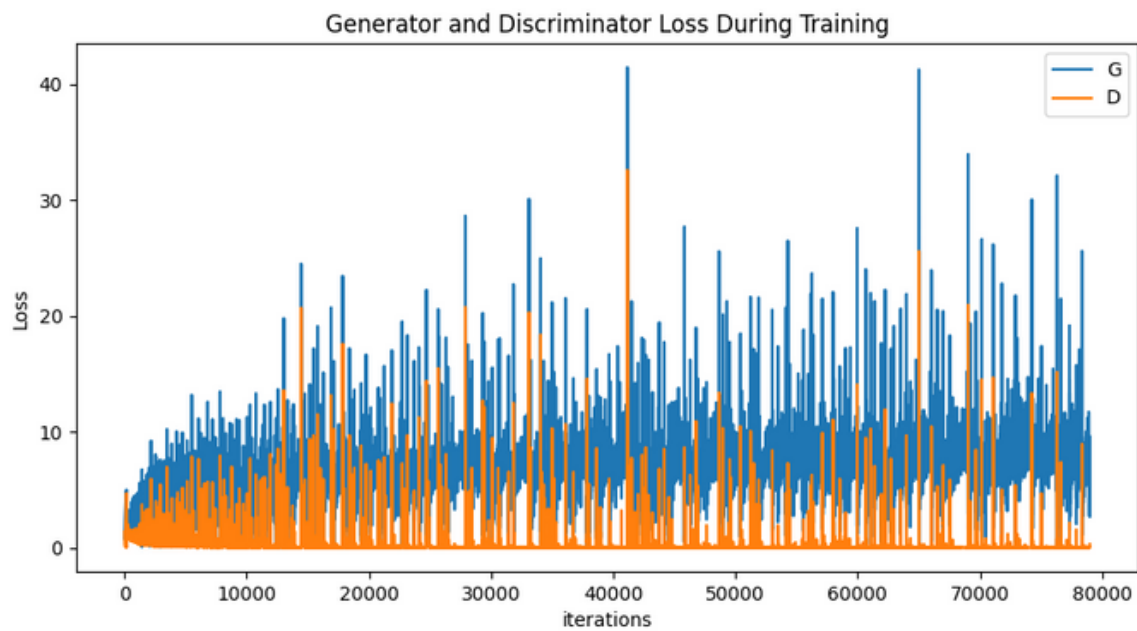


here we see in detail some results

-----

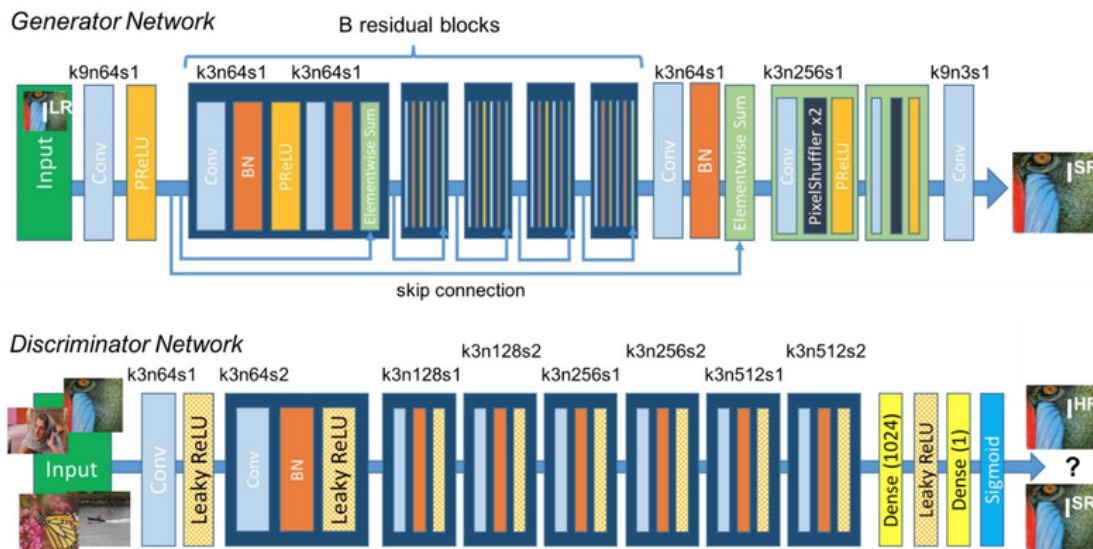
qui vediamo nel dettaglio alcuni risultati

## Generate NFTs: DCGAN results EAPES



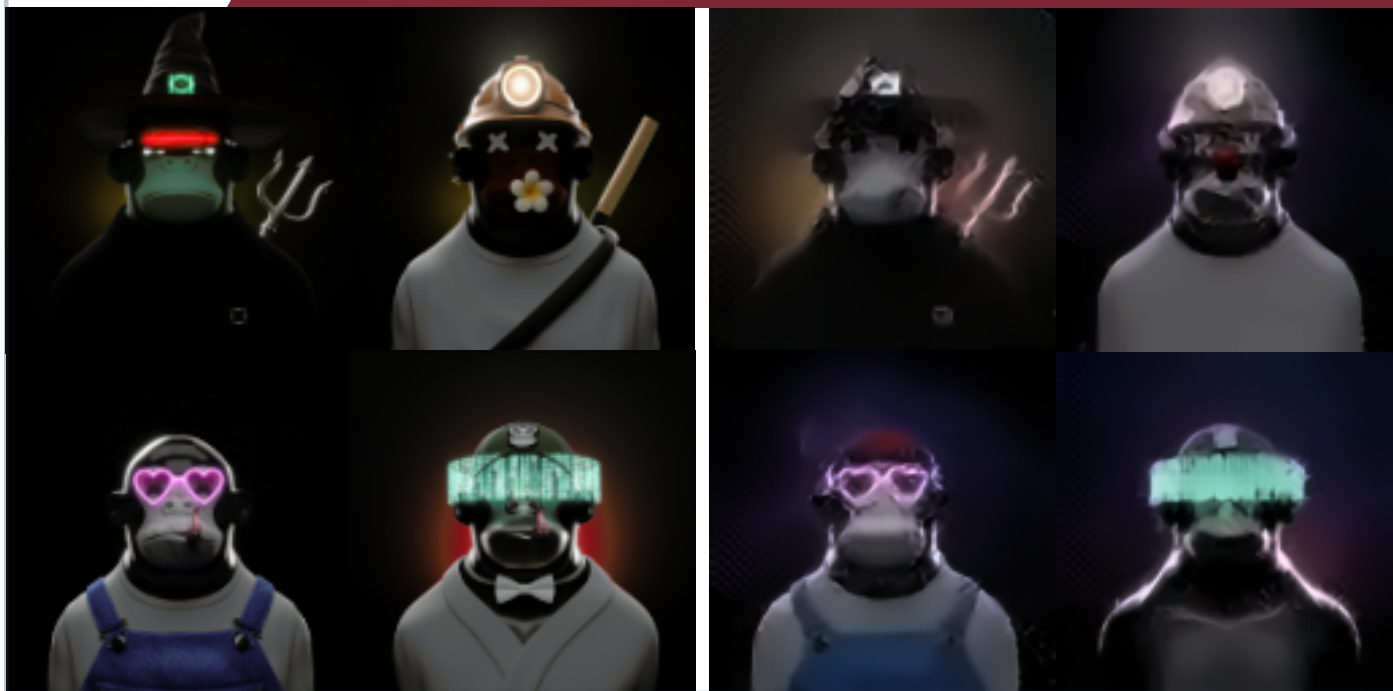
here we note the losses of the generator and of the discriminator, much more stable and low

-----  
qui notiamo i loss del generato e del discriminator, molto piu' stabili e bassi



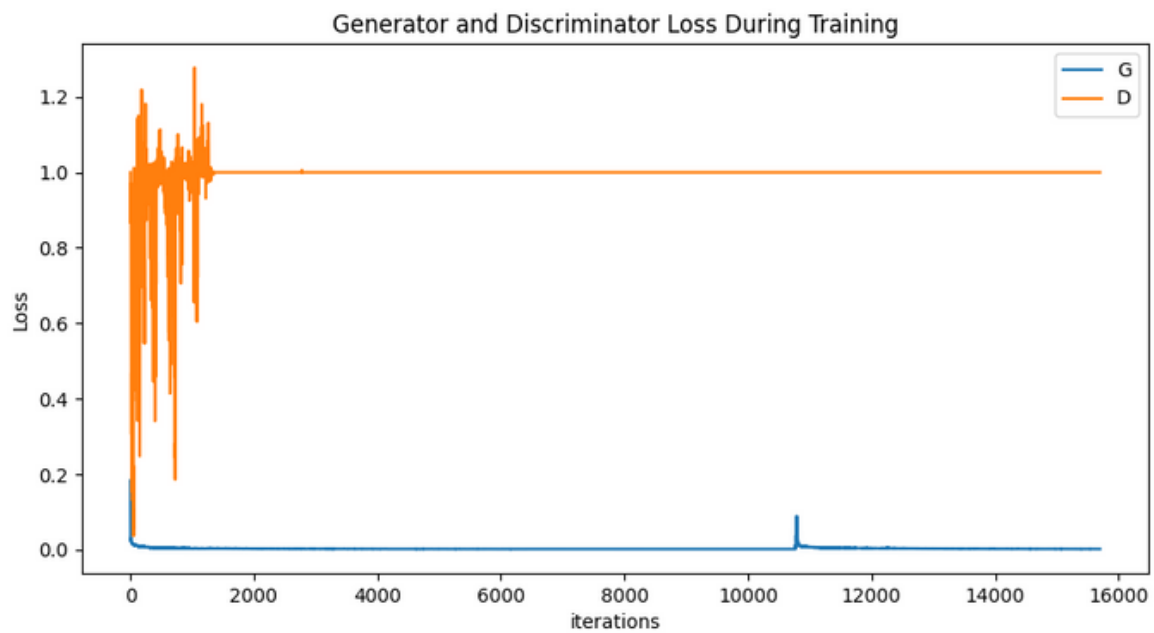
the second part of the model uses a srgan. this gan is a bit more complex. to train this model, the dataset images are reduced and given as input to the generator that will do the upscaling. finally the discriminator will judge the output of the generator to see how different they are from the original high resolution images.

la seconda parte del modello utilizza una srgan. questa gan e' un po' piu' complessa. per trainare questo modello le immagini del dataset vengono ridotte e date in input al generatore che fara l'upscaling. infine il discriminator giudichera' l'output del generator per vedere quanto sono diverse dalle immagini originali ad alta risoluzione.



I chose to train the srgan only for the EAPES that resulted in the best trained collection.  
here we see images at 128x128 pixels

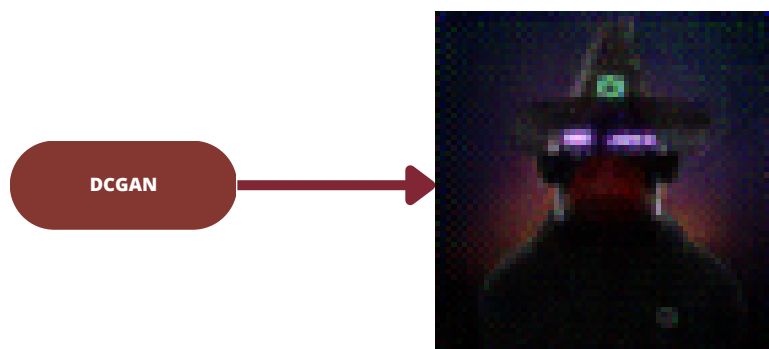
-----  
ho scelto di addestrare la srgan solo per le EAPES che sono risultate la collection migliore  
trained. qui vediamo delle immagini a 128x128 pixels



here we see the losses of the srgan.

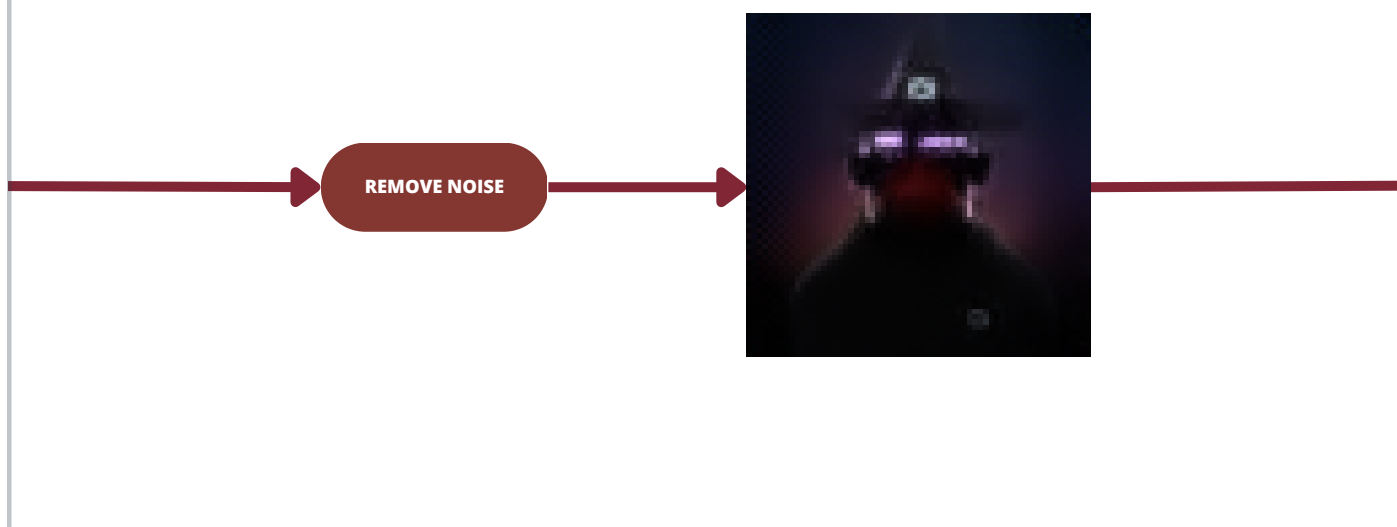
-----  
qui vediamo i loss della srgan.





in summary, the whole model works as follows:  
dcgan generates a 64x64 image

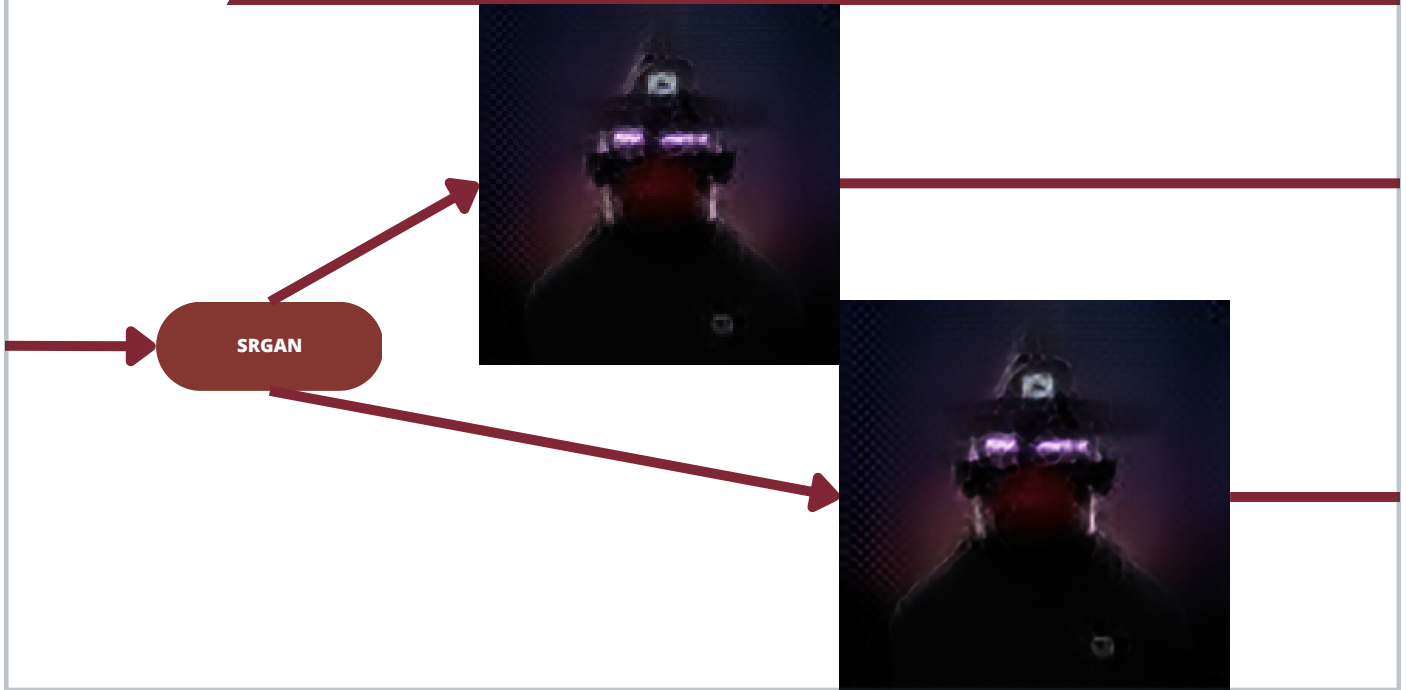
-----  
ricapitolando, l'intero modello funziona come segue:  
la dcgan genera un immagine 64x64



this image is denoised

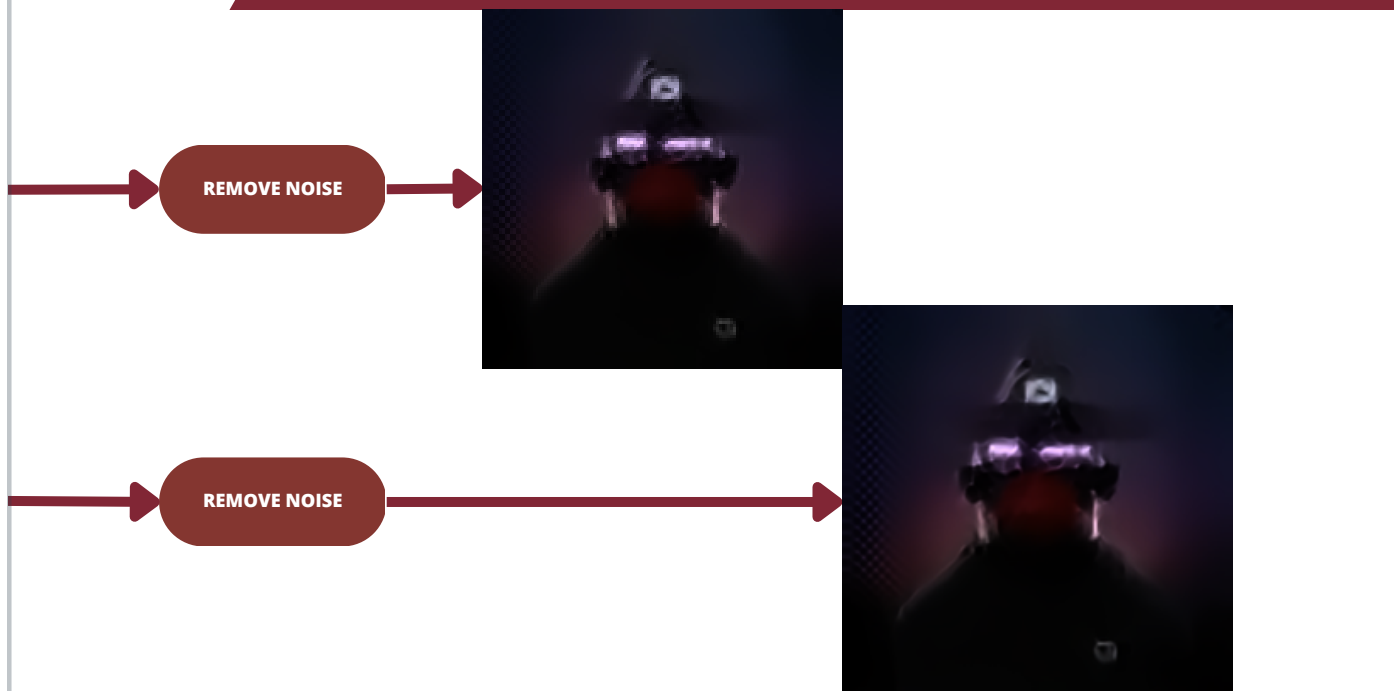
-----

questa immagine viene denoised



and then it is passed to the srgan which upscales the image returning an image with 128x128 and 256x256 pixels

-----  
e poi viene passata alla srgan che fa l'upscale dell'immagine restituendo un immagine a 128x128 e 256x256



finally the noise is removed

-----  
infine viene rimosso il rumore

### 3. Blend NFTs

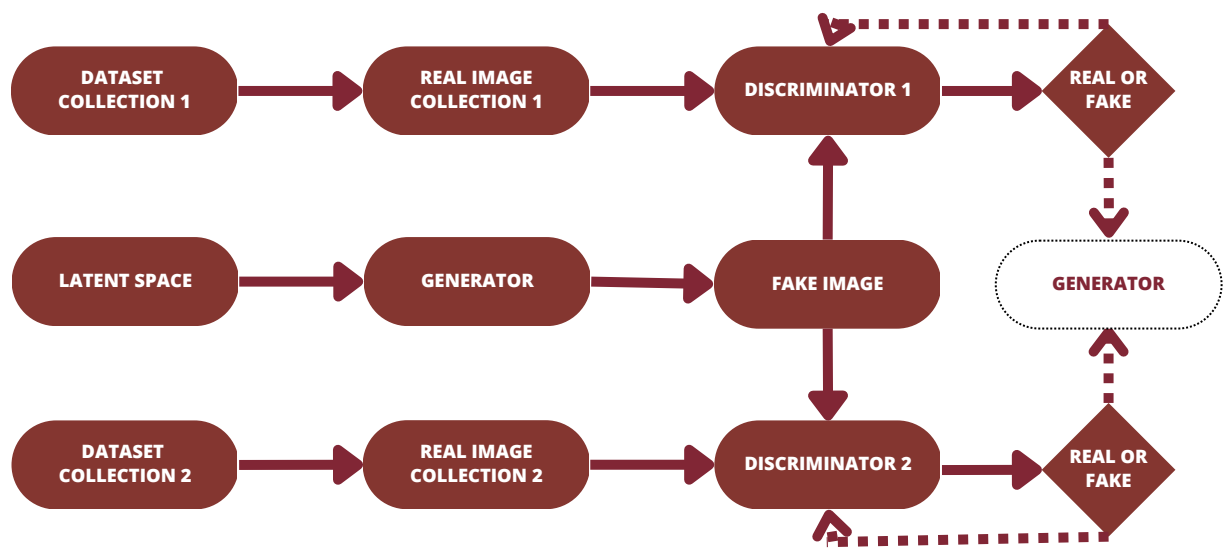
- Model
  - DCGAN-2D
- Final results



last thing, I tried to blend two nft collections

-----

ultima cosa, ho provato a fare blending di due collection nft



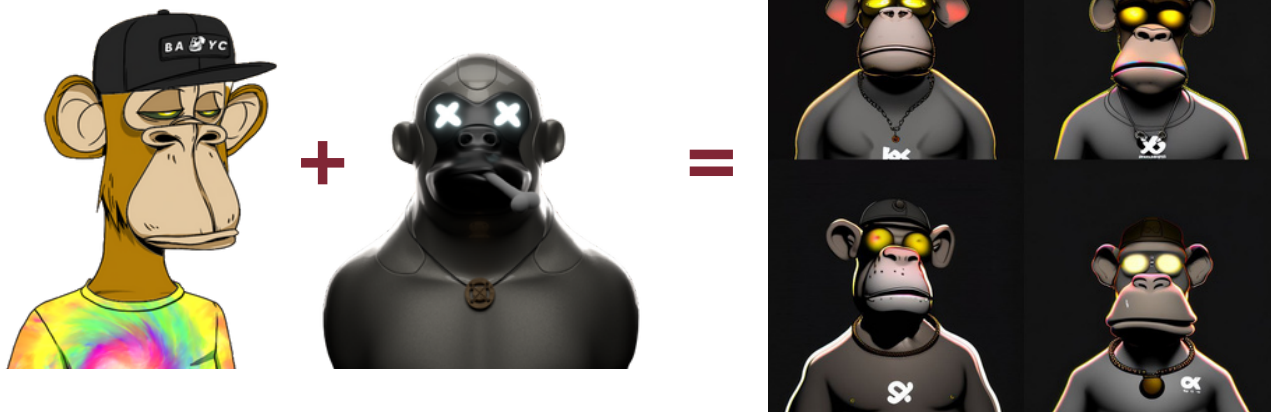
I then created a new dcgan with two discriminators. the fake image must be judged by the first discriminator who judges how similar it is to the first nft collection. then the fake image must also be judged by the second discriminator who does the same thing but on the second nft collection.

the two discriminators are in the AND therefore the max loss between the two losses is taken for each image.

ho quindi creato una nuova dcgan con due discriminators. l'immagine fake deve quindi essere giudicata dal primo discriminatore che giudica quanto e' somigliante alla prima collection nft. la fake image poi deve anche essere giudicata dal secondo discriminatore che fa la stessa cosa ma sulla seconda collection nft.

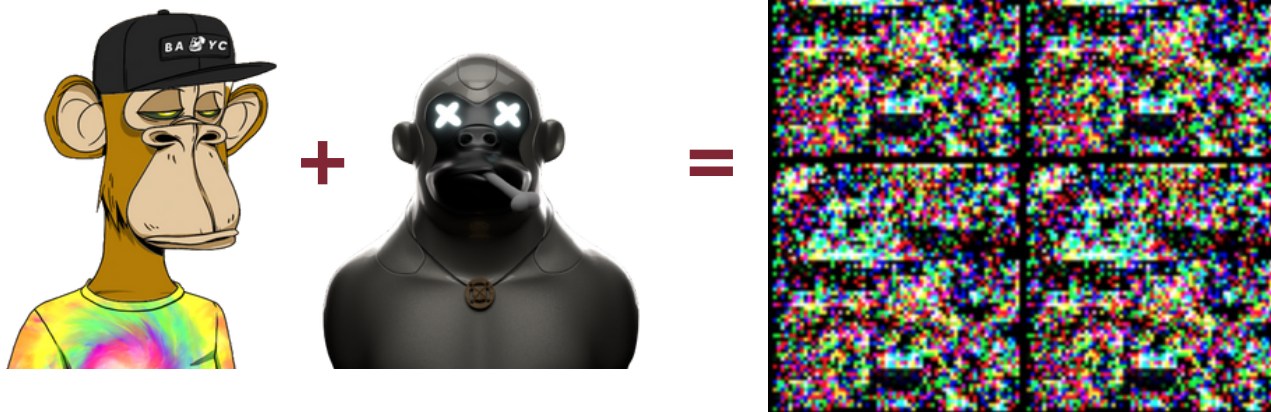
i due discriminatori sono in AND quindi viene presa per ogni immagine il max lost tra i due losses.

## Blend NFTs: Expectation :)



these were the expectations. this result was obtained by midjourney. good job

-----  
queste erano le aspettative. questo risultato e' stato ottenuto da midjourney. good job



this is the result of my homemade model. it just doesn't work. we do not blend with a dcgen. the recommended gan to do these things is the stylegan.

-----  
questo e' il risultato del mio modello fatto in casa. semplicemente non funziona. non si  
effettaumo i blend con una dcgen. la gan consigliata per fare queste cose e' la stylegan.



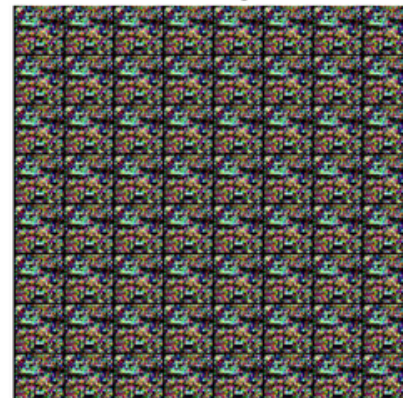
Real1 Images

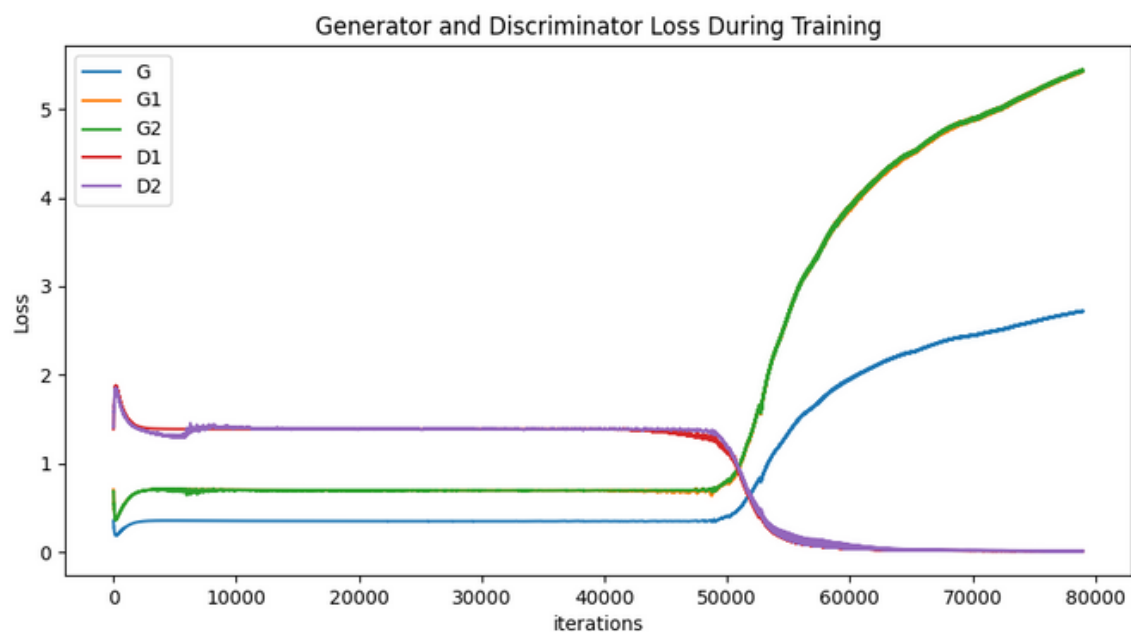


Real2 Images



Fake Images





**Thanks for the attention**



**Vincenzo Imperati 1834930**

