

# **MLOPS: GAN CELEB FACES**

Eliott BEERNAERT  
Thomas PEREIRA  
Eliott VANCOPPENOLLE





# SOMMAIRE



Architecture globale



Conception du modèle de génération d'images



Collaboration Git et versionning



Intégration d'une API



Résultats



# Utilisation du DDD (Domain-Driven Design)

## Thomas :

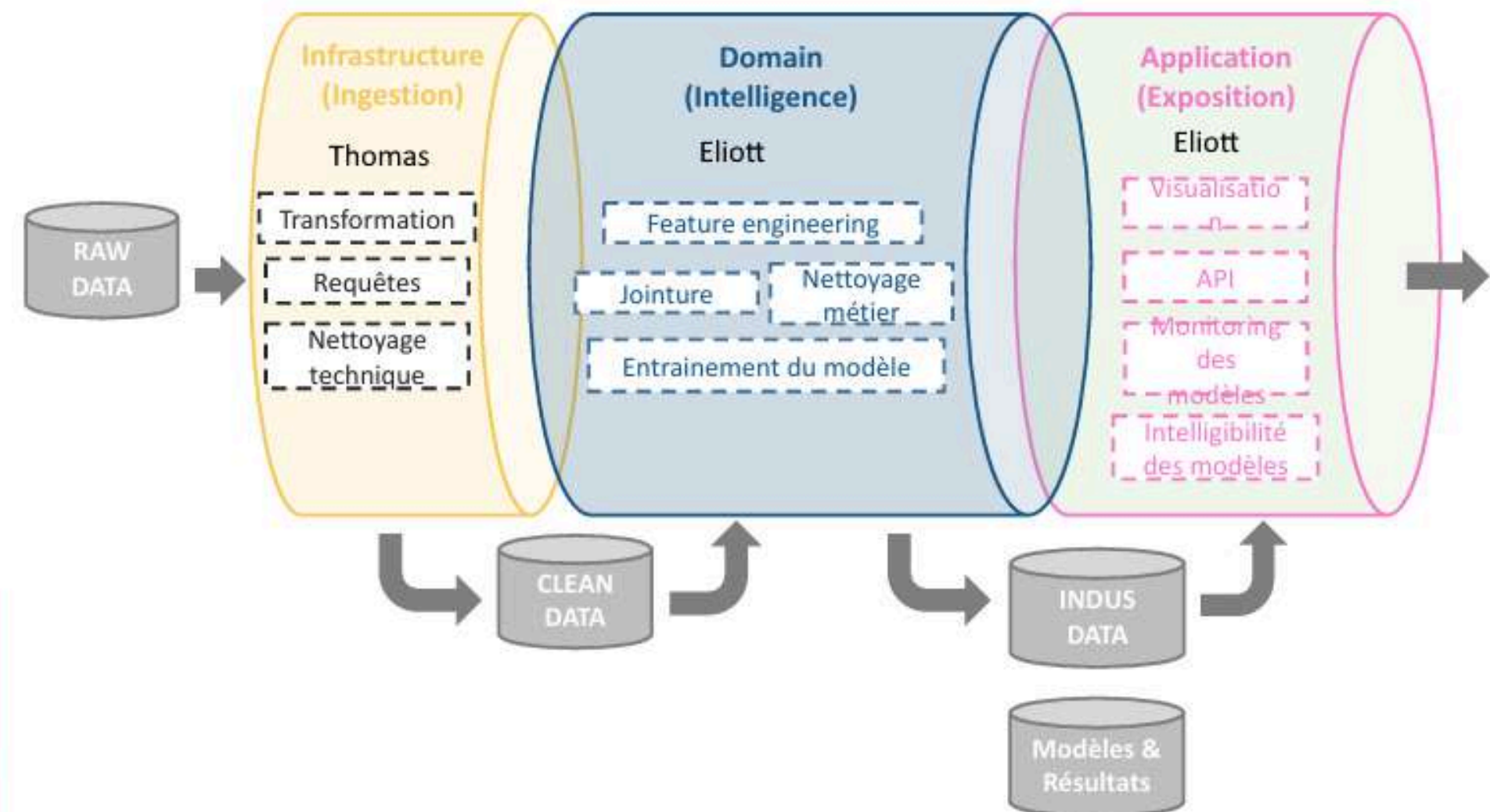
- Téléchargement des données et unzip
- Transformation des images pour retirer le background et ajouter un fond vert
- Préparation des données pour le modèle

## Eliott V. :

- Création et perfectionnement du GAN
- Création du pipeline d'entraînement et de reprise de l'entraînement
- Préparation de la démo

## Eliott B. :

- Création de l'API qui affiche l'image générée
- Création d'un code permettant la génération d'un GIF sur l'évolution des epochs

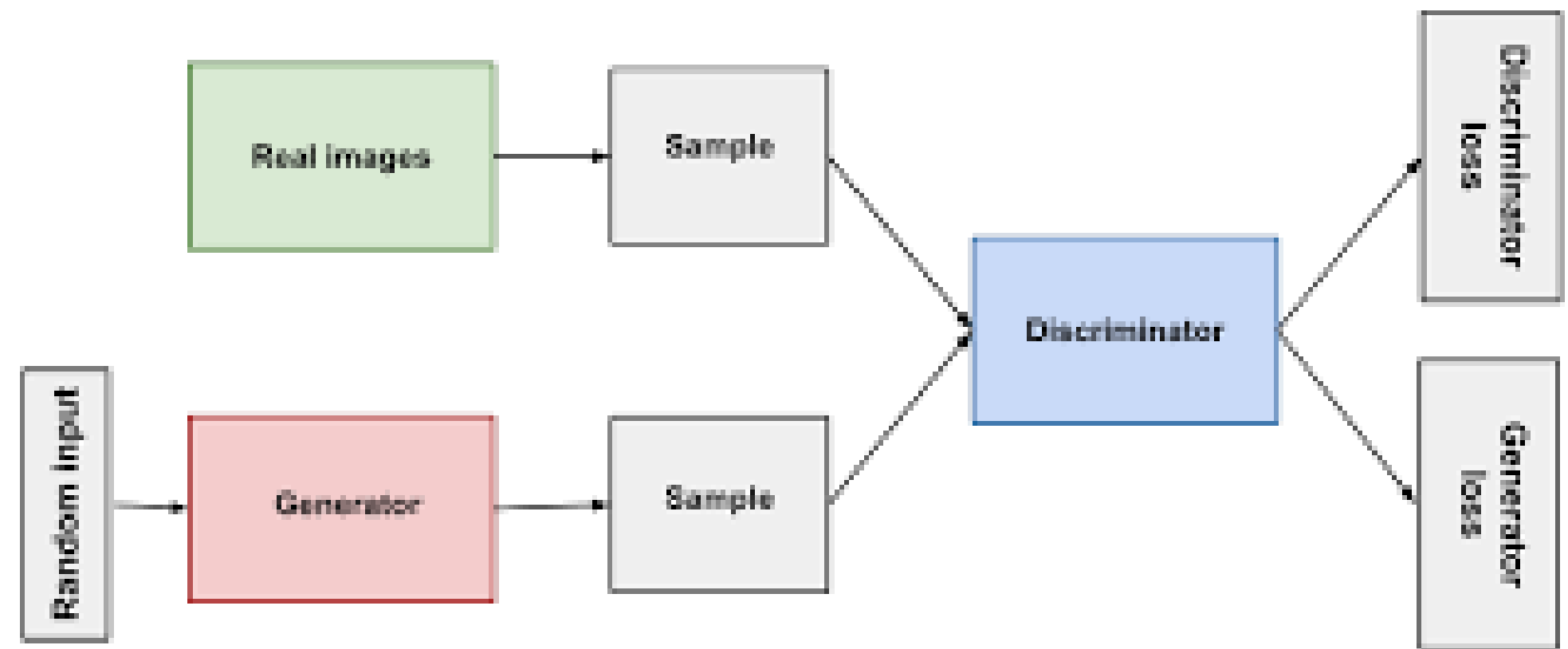


# Dataset : CelebFaces Attributes (CelebA)

Le dataset **CelebFaces Attributes (CelebA)** est une vaste collection d'images de visages de célébrités, largement utilisée dans la recherche en vision par ordinateur et en apprentissage profond. Il contient plus de **200 000 images**, accompagnées d'annotations détaillées sur 40 attributs faciaux, tels que les expressions, la présence de lunettes ou la couleur des cheveux. Les images sont disponibles en différentes résolutions, et pour ce projet, elles sont redimensionnées à une taille standardisée de **128 x 160 pixels**, ce qui permet une utilisation efficace pour des tâches comme la génération ou la reconnaissance d'images tout en réduisant la complexité computationnelle. Ce dataset est apprécié pour sa diversité et sa qualité, rendant possible l'entraînement de modèles complexes comme les **GAN**.

# Fonctionnement des GAN

Les **GAN (Generative Adversarial Networks)** reposent sur deux réseaux neuronaux : un **générateur**, qui crée des données artificielles à partir de bruit aléatoire, et un **discriminateur**, qui distingue ces données des vraies. Les deux réseaux s'entraînent en compétition : le **générateur** s'efforce de tromper le **discriminateur**, tandis que ce dernier affine sa capacité à détecter les faux. Cet affrontement permet au générateur de produire des données de plus en plus réalistes.



# Choix techniques

- Limitation du déséquilibre générateur - discriminateur

$$\text{Écart relatif} = \frac{\text{Perte Générateur} - \text{Perte Discriminateur}}{\max(\text{Perte Générateur}, \text{Perte Discriminateur}, \epsilon)}$$

Si  $|\text{Écart relatif}| > 30\%$  (paramètre personnalisable), détecter le modèle le plus performant (perte minimale).  
Mettre en pause l'entraînement du modèle le plus performant pour le prochain batch.

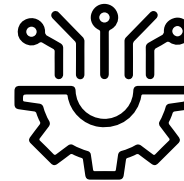
- Possible d'enregistrer des images à chaque epoch (intervalle personnalisable)

- Possible d'enregistrer des les modèles à chaque epoch (intervalle personnalisable)

# GAN Celeb Faces : roadmap fonctionnelle



## Chargement et transformation des données



## Entraînement du GAN

- Entraînement du GAN de 0
- Entraînement du GAN à partir d'une certaine Epoch.



## Monitoring

Monitoring des fichiers de logs, choix possible d'un fichier de log particulier.

## GIF

### Création de GIF

Création d'un GIF qui permet de voir l'évolution des Epochs.

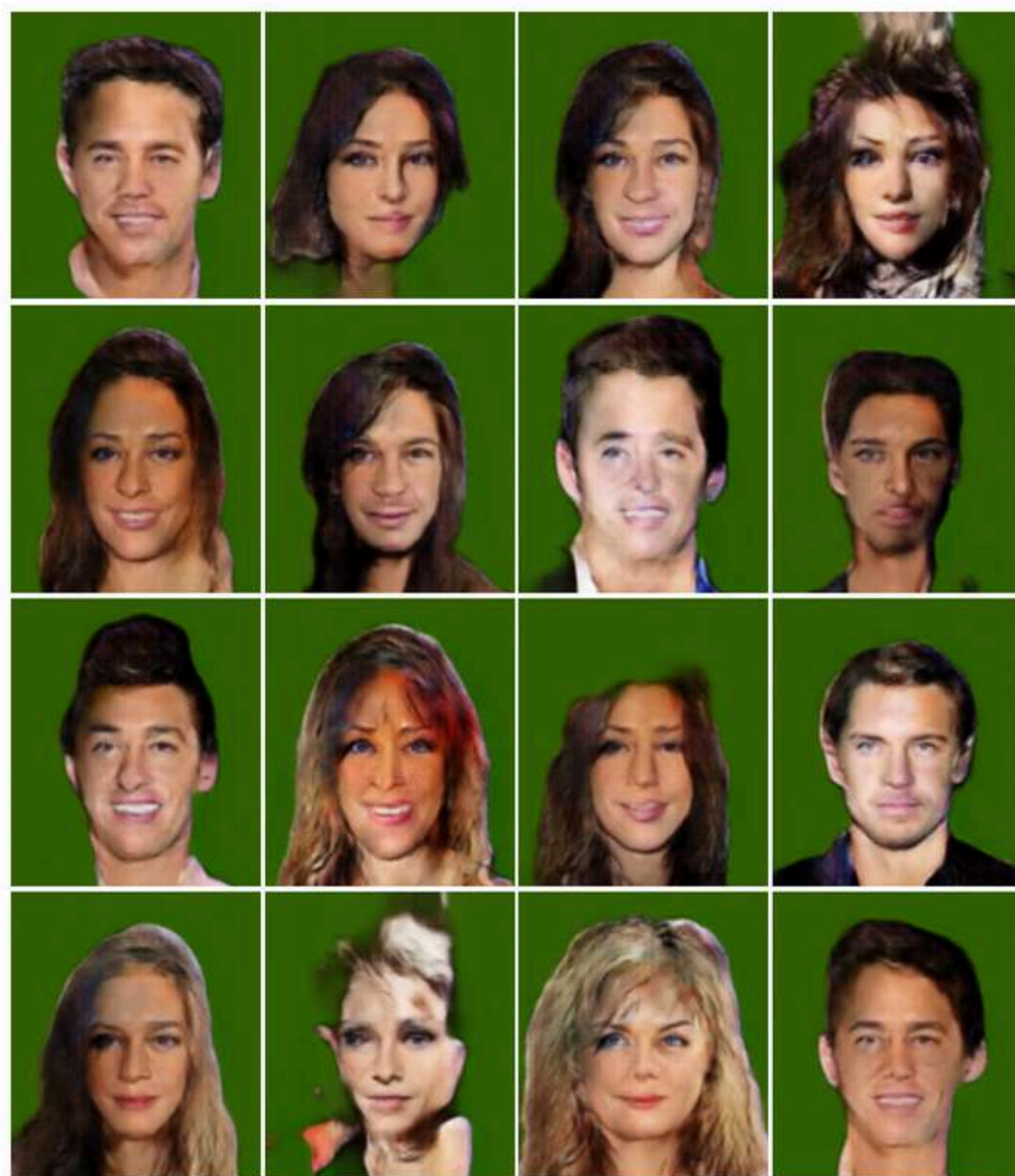
## API

### Lancement de l'API

Permet de lancer la génération d'image sur un modèle précis ou non.



# Résultats



L'entraînement d'un **GAN** peut nécessiter un **grand nombre d'epochs** pour produire des résultats **convaincants**, en fonction de la complexité des données et de l'architecture du réseau. Par exemple, lors d'un test d'entraînement **1452 epochs** ont été réalisés à titre de **démonstration**, les résultats peuvent être limités, car le modèle n'a pas encore eu suffisamment de temps pour **converger**. Généralement, des résultats acceptables commencent à apparaître après **10 000 à 20 000 epochs**, lorsque le **générateur** parvient à créer des échantillons suffisamment réalistes pour tromper le **discriminateur** de manière cohérente.



# Tutoriel : Utilisation du modèle

## Chargement des packages

```
from infrastructure import data_load_transform
from domain import train_gan
from application import run_app, create_gif, monitoring
```

✓ 5.3s

## Entraînement du GAN de 0 à 5 epochs

```
train_gan(epochs=5, batch_size=32)
```

✓ 32.0s

Starting new training session...

Directory '../data/processed\_data' contains 100 files.

Found 100 files.

2024-12-21 00:32:38 : Epoch 1 completed in 7.58 seconds with gen\_loss=0.6943 and disc\_loss=1.2915.

2024-12-21 00:32:42 : Epoch 2 completed in 4.40 seconds with gen\_loss=0.7174 and disc\_loss=0.9175.

2024-12-21 00:32:49 : Epoch 3 completed in 5.39 seconds with gen\_loss=0.6663 and disc\_loss=0.8840.

2024-12-21 00:32:55 : Epoch 4 completed in 6.04 seconds with gen\_loss=0.8549 and disc\_loss=0.7569.

2024-12-21 00:33:00 : Epoch 5 completed in 4.41 seconds with gen\_loss=1.0353 and disc\_loss=0.6024.

## Téléchargement et preparation des données

```
data_load_transform(n_image=100, n_jobs=-1)
```

✓ 4m 7.8s

Starting pipeline execution...

Data is not present. Downloading...

Download completed.

Extracting files...

Data has been extracted to: ../data/raw\_data

Selected 100 images to process.

Processing 100 additional images...

Processing 0 additional images...

Pipeline successfully executed.

## Reprise de l'entraînement du GAN pour 5 epochs

```
train_gan(epochs=5, batch_size=32, resume_epoch=5)
```

✓ 30.1s

Resuming training from epoch 5...

WARNING:tensorflow:From c:\Users\eliot\Desktop\ML\_Ops\_CelebFaces\_GAN\.venv\lib\site-packages\keras\src\backend\tensorflow\core.py:222

Models loaded from ../training/saved\_models/generator\_epoch\_5.keras and ../training/saved\_models/discriminator\_epoch\_5.keras

Directory '../data/processed\_data' contains 100 files.

Found 100 files.

2024-12-21 00:34:21 : Epoch 6 completed in 5.76 seconds with gen\_loss=1.0424 and disc\_loss=0.8150.

2024-12-21 00:34:26 : Epoch 7 completed in 4.90 seconds with gen\_loss=1.0106 and disc\_loss=0.7295.

2024-12-21 00:34:32 : Epoch 8 completed in 4.59 seconds with gen\_loss=1.1365 and disc\_loss=0.8284.

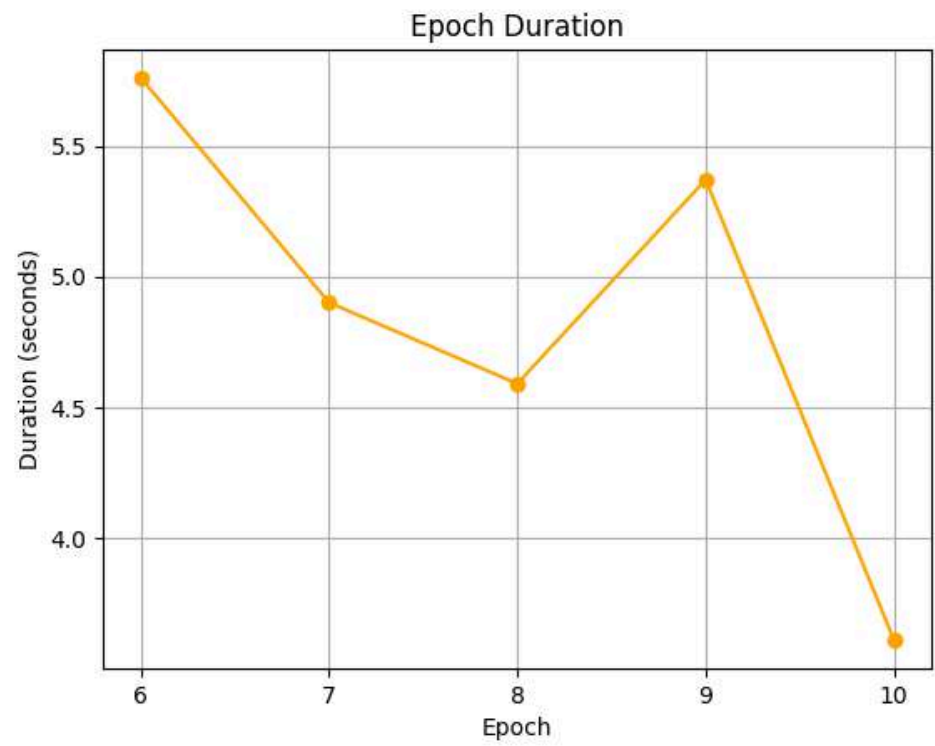
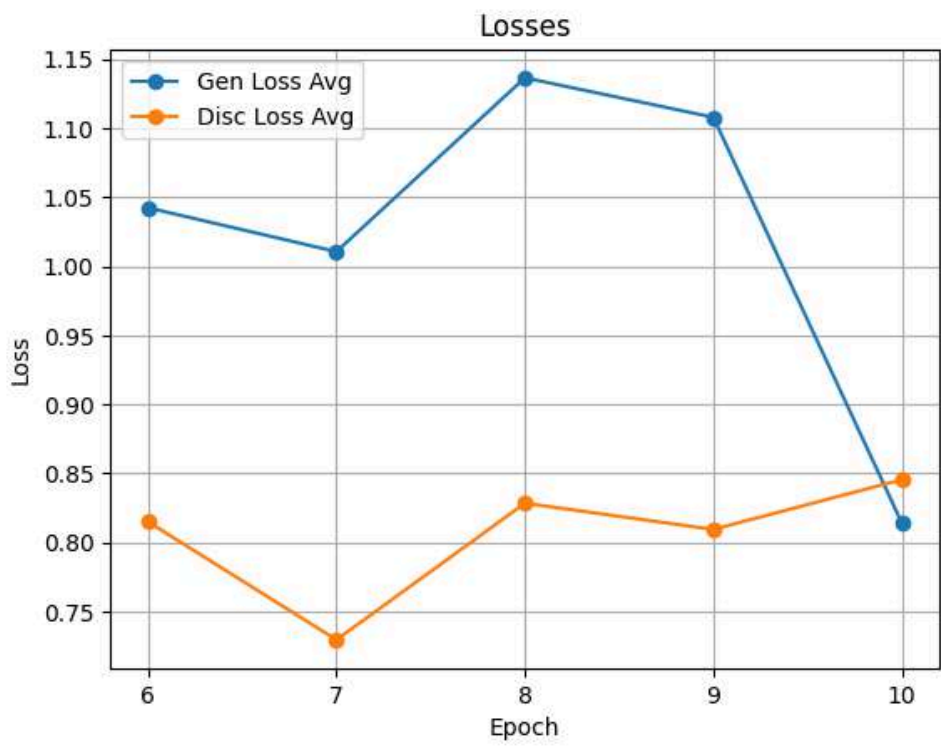
2024-12-21 00:34:38 : Epoch 9 completed in 5.37 seconds with gen\_loss=1.1079 and disc\_loss=0.8094.

2024-12-21 00:34:42 : Epoch 10 completed in 3.61 seconds with gen\_loss=0.8141 and disc\_loss=0.8455.

# Tutoriel : Utilisation du modèle

## Monitoring des epochs

```
monitoring()  
✓ 0.1s  
monitoring("log_20241220_222321.csv")
```



## Monitoring du modèle en générant des stats sur les epochs

Stats:

	Epoch_Duration	Gen_Loss_Avg	Disc_Loss_Avg
mean	4.846000	1.022300	0.805560
std	0.822514	0.126745	0.044738
min	3.610000	0.814100	0.729500
max	5.760000	1.136500	0.845500

# Tutoriel : Utilisation du modèle

## Création d'un GIF à partir des images générés pendant l'entraînement

```
create_gif("../generated_images.gif", duration = 300)
```

✓ 1.2s

GIF généré avec succès : ../generated\_images.gif

## Lancement de l'API dernier générateur

```
run_app()
```

6.1s

Chargement du modèle : ../training/saved\_models\generator\_epoch\_1450.keras

\* Serving Flask app 'application.app'

\* Debug mode: off

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

\* Running on all addresses (0.0.0.0)

\* Running on <http://127.0.0.1:8000>

\* Running on <http://192.168.1.42:8000>

Press CTRL+C to quit



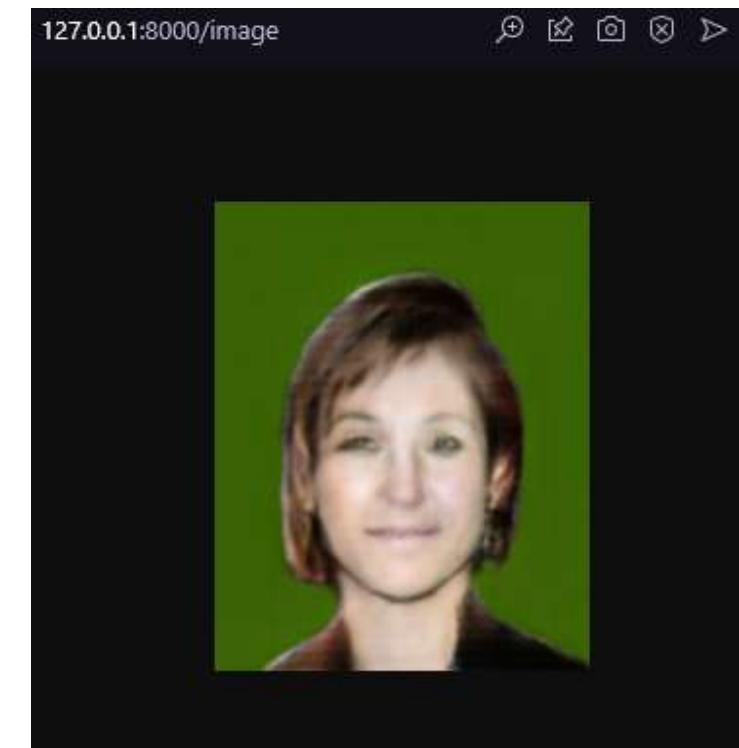
# Tutoriel : Utilisation du modèle

## Création d'un GIF à partir des images générés pendant l'entraînement

```
create_gif("../generated_images.gif", duration = 300)
✓ 12s
GIF généré avec succès : ../generated_images.gif
```

## Lancement de l'API dernier générateur

```
run_app()
6.1s
Chargement du modèle : ../training/saved_models\generator_epoch_1450.keras
* Serving Flask app 'application.app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8000
* Running on http://192.168.1.42:8000
Press CTRL+C to quit
```



## Lancement de l'API avec un modèle défini

```
run_app("generator_epoch_1440.keras")
```

# MLOPS: GAN CELEB FACES