



Développez une preuve de concept

Deep Learning

Génération d'images

StyleGAN2-ADA sous PyTorch



[source](#)



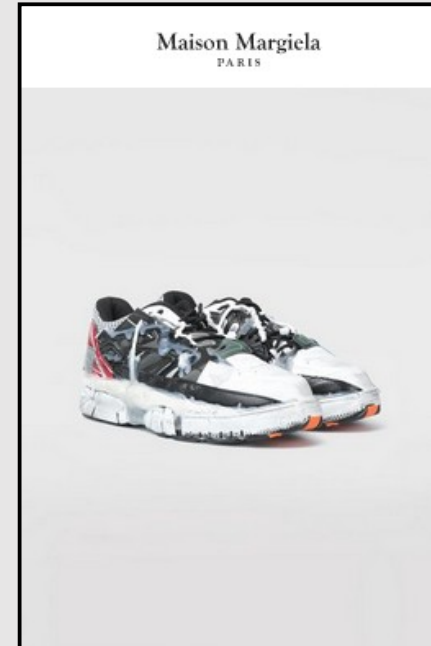
Compétences évaluées

- Identifier des sources d'informations fiables et pertinentes
- Identifier les méthodes "état de l'art" pour résoudre un problème de Data science
- Développer une preuve de concept pour résoudre un problème de Data science
- Réaliser une veille sur les évolutions de la Data Science



Client

- Marque de prêt à porter haut de gamme
- Renouvellement de collection tous les 6 mois
- Nouveaux designs de baskets





Objectif : GAN

- Recherche d'une méthode récente de génération d'images
- Méthode de production actuelle : **DCGAN**
 - Images de sortie petites : 128x128
 - Images de faible résolution
- Méthode nouvelle : **StyleGAN2 ADA**





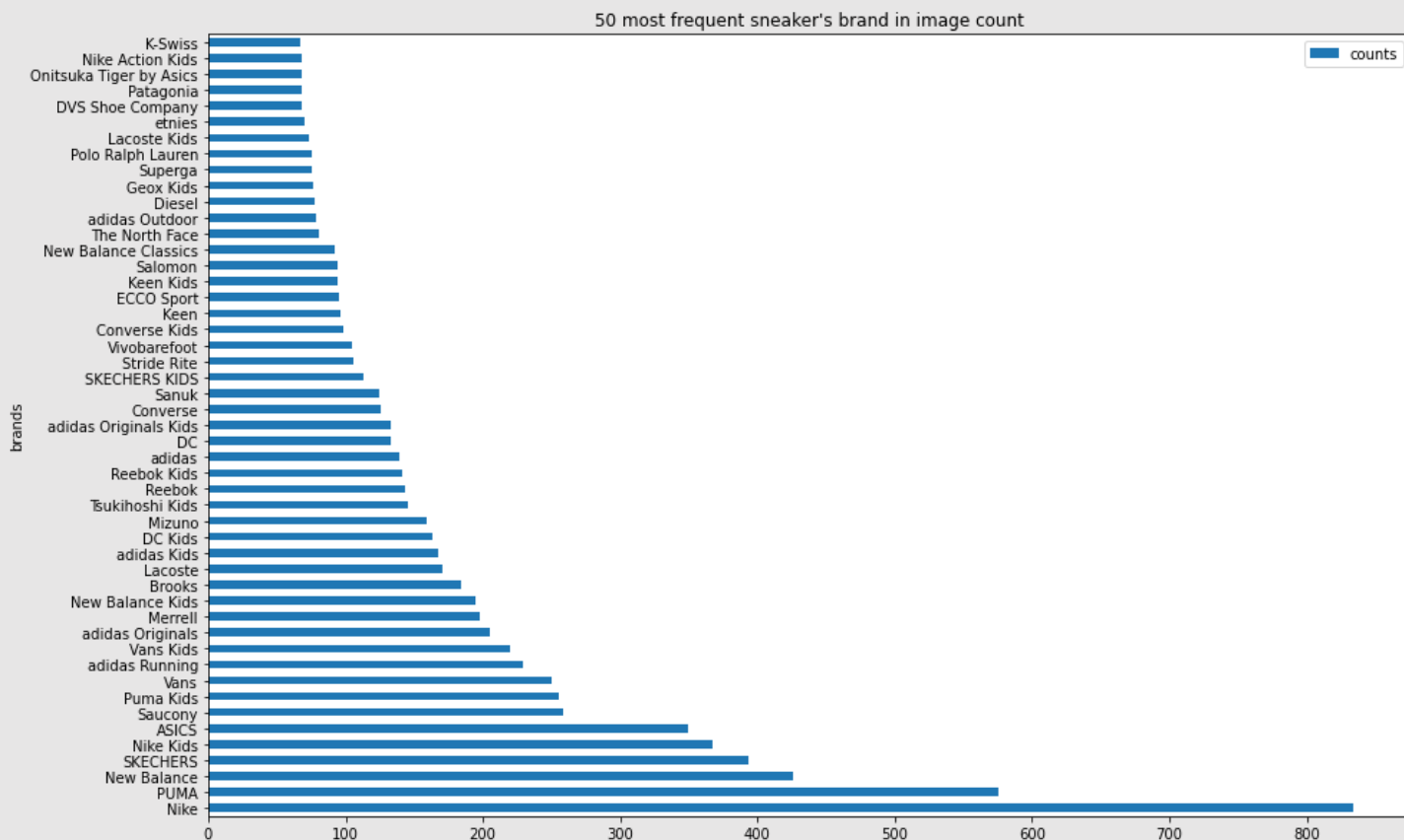
Présentation des données

- Dimensions 136 * 136 pixels
- Poids 3.8 kB
- Format JPEG
- Centrées sur un fond blanc
- Même orientation
- 12 859 images
- sneakers_and_Athletic_Shoes





Exploration des données





Préparation des données

DCGAN

- Re-dimension de 64x64 à 128x128
- Normalisation
- CenterCrop

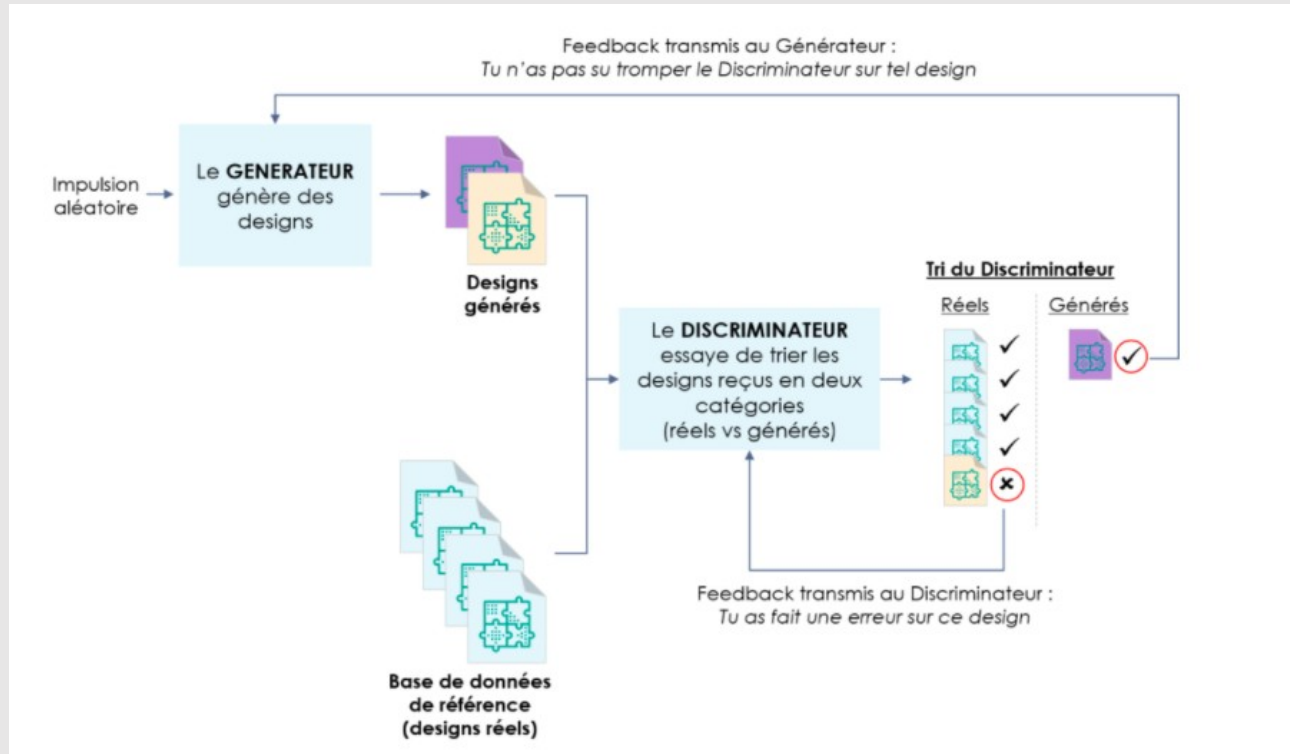
StyleGAN2 ADA

- Re-dimension de 136x136 à 256x256
- dataset_tool.py
- Data augmentation ADA



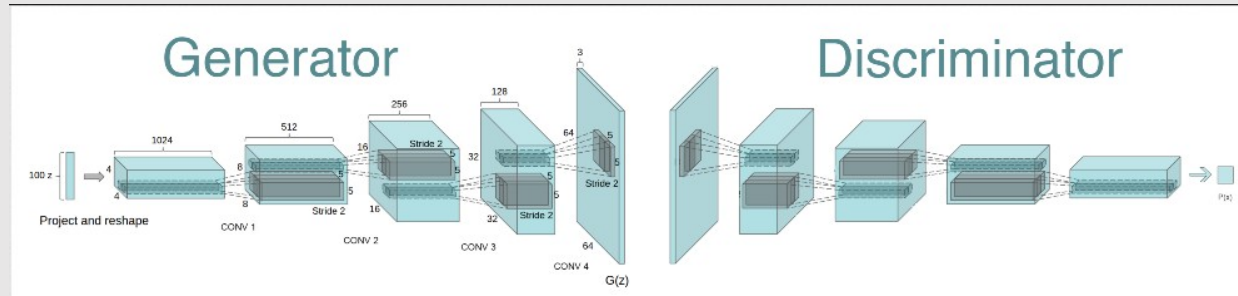


Modélisation GAN





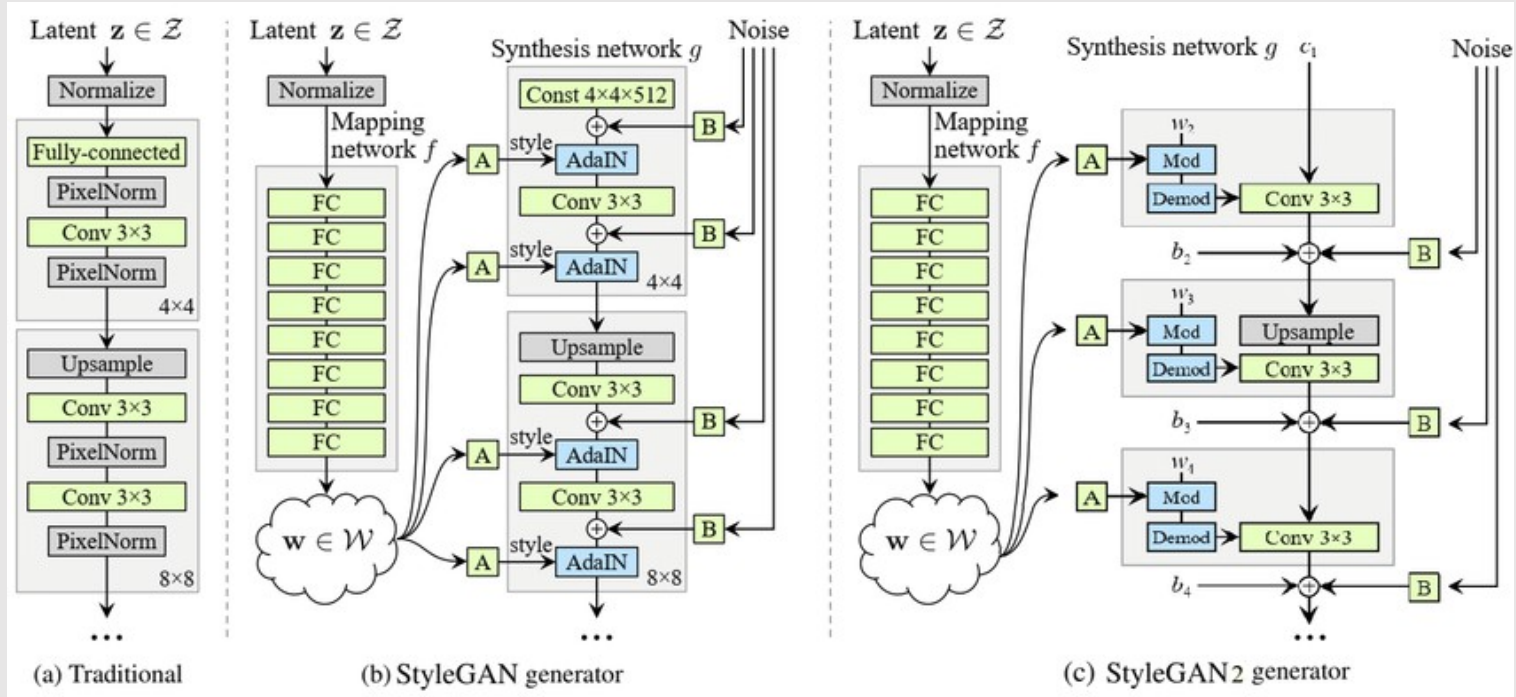
Modélisation DCGAN



```
netG(  
  (main): Sequential(  
    (0): ConvTranspose2d(100, 1024, kernel_size=(4, 4), stride=(1, 1), bias=False)  
    (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (2): ReLU(inplace=True)  
    (3): ConvTranspose2d(1024, 512, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)  
    (4): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (5): ReLU(inplace=True)  
    (6): ConvTranspose2d(512, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)  
    (7): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (8): ReLU(inplace=True)  
    (9): ConvTranspose2d(256, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)  
    (10): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (11): ReLU(inplace=True)  
    (12): ConvTranspose2d(128, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)  
    (13): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (14): ReLU(inplace=True)  
    (15): ConvTranspose2d(64, 3, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)  
    (16): Tanh()  
  )  
)
```

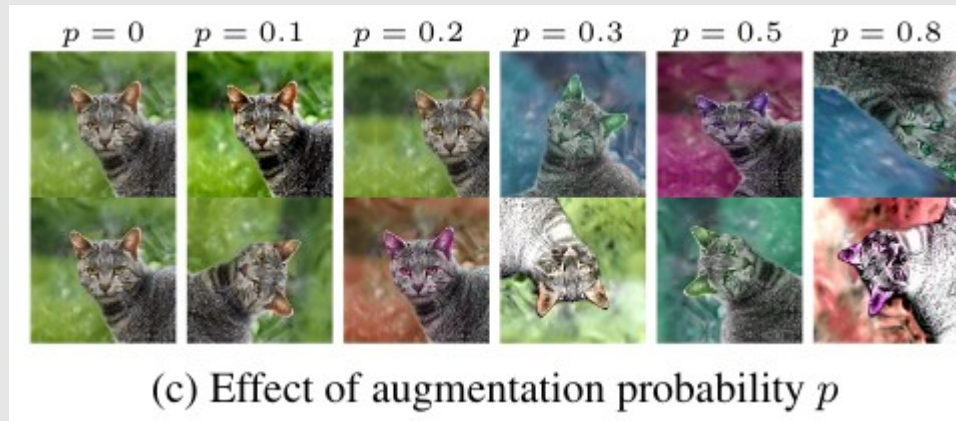


Modélisation StyleGAN



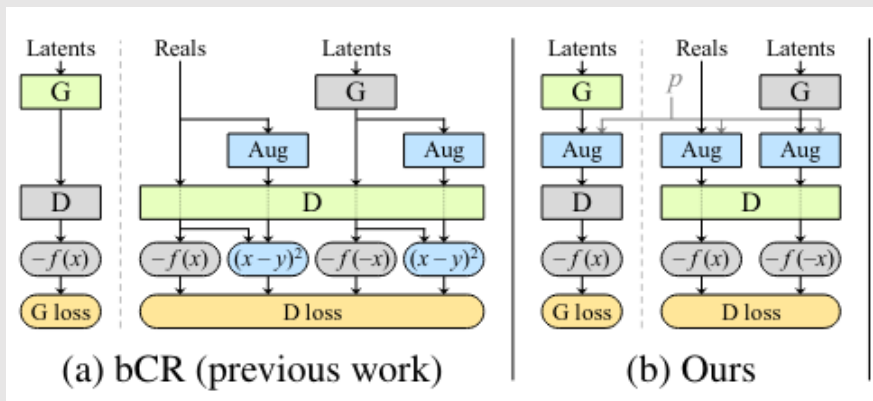


Modélisation StyleGAN2 ADA





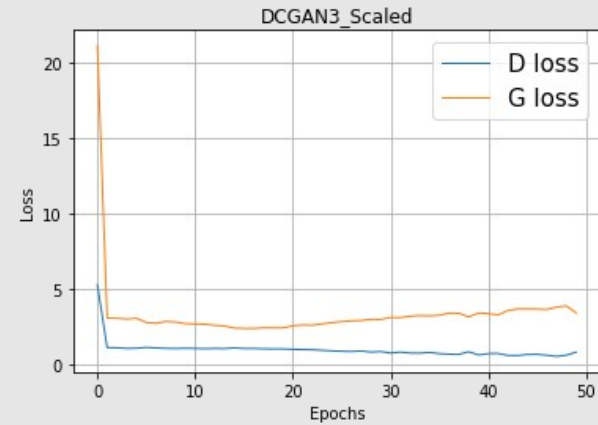
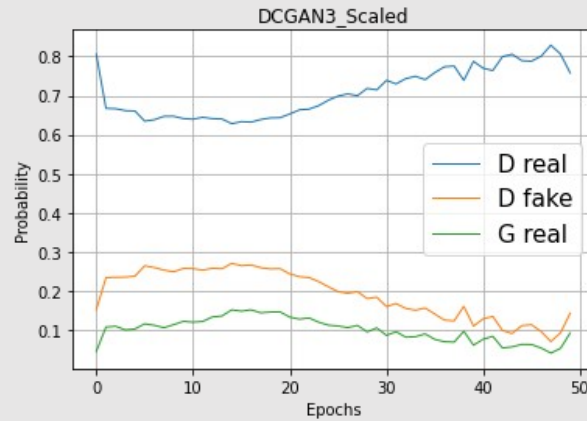
Modélisation StyleGAN2 ADA



```
"augment_kwargs": {  
  "class_name":  
    "training.augment.AugmentPipe",  
  "xflip": 1,  
  "rotate90": 1,  
  "xint": 1,  
  "scale": 1,  
  "rotate": 1,  
  "aniso": 1,  
  "xfrac": 1,  
  "brightness": 1,  
  "contrast": 1,  
  "lumaflip": 1,  
  "hue": 1,  
  "saturation": 1  
}
```



Résultats DCGAN



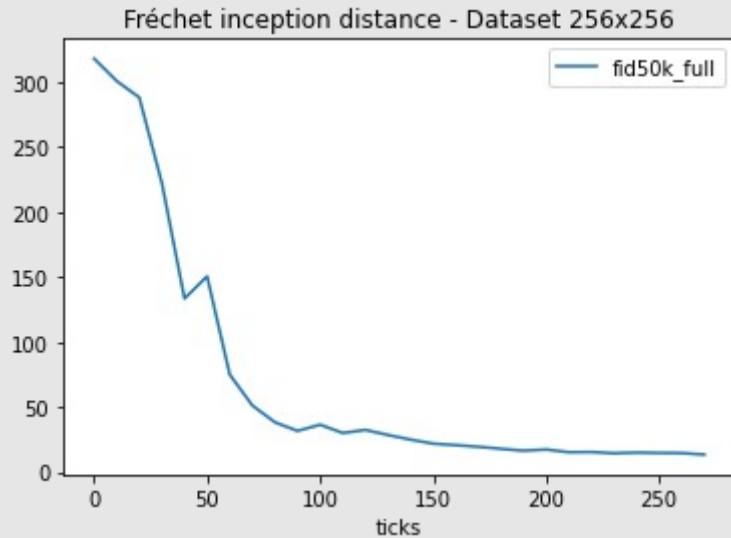
real



fake



Résultats StyleGAN2 ADA



real

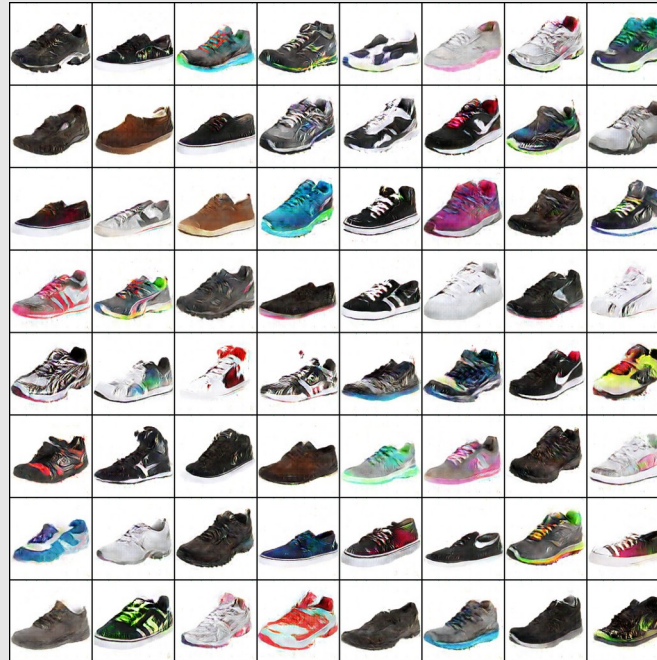


fake



Synthèse des résultats

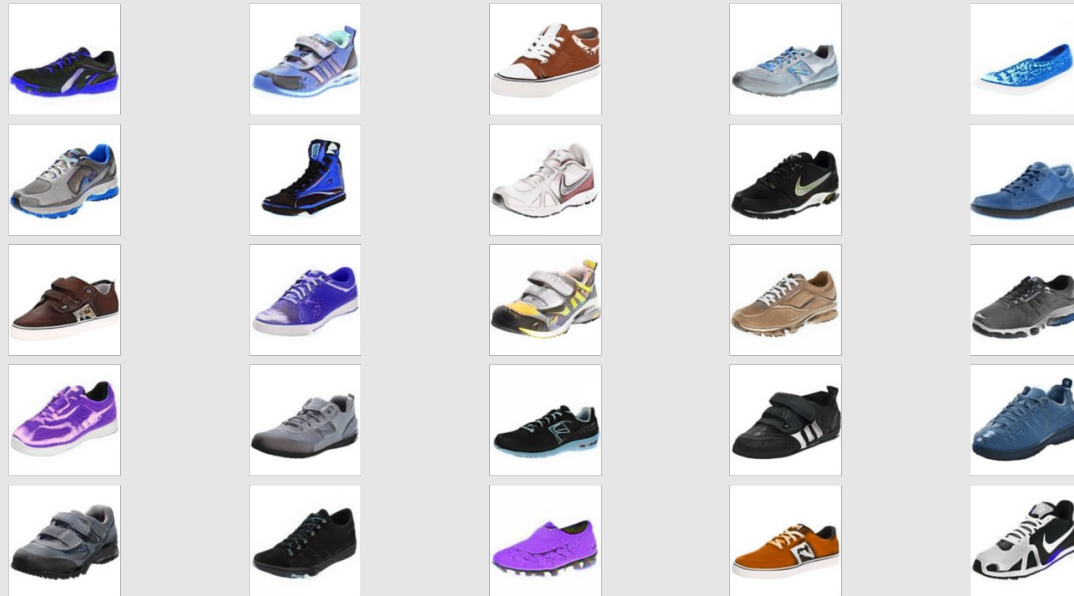
DCGAN





Synthèse des résultats

StyleGAN2 ADA





Synthèse des résultats

	Avantage	Inconvénient
DCGAN	Entraînement rapide: 1h	Images floutés Images petites
StyleGAN2 ADA	Entraînement long: 23h	Images fidèles à la réalité Images plus grandes



----- MERCI! -----



Source