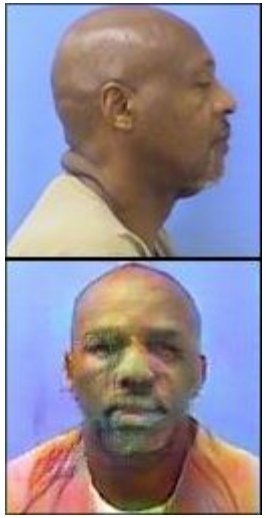
Projet Image :
Détection et création d'un mugshot à partir d'une caméra de surveillance



[HAI927I] - M2 IMAGINE

Mathis Duban

Paul Deligne

Contexte et objectifs du projet

- détecter la présence d'une personne depuis un flux vidéo temps réel

- extraire le visage issu d'une capture vidéo

- générer le mugshot correspondant

Plan

- état de l'art sur les méthodes de génération de mugshot
- détail de notre implémentation
- résultats et analyses
- démonstration et conclusion

État de l'art

Synthèse de visages & édition de visages avec GANs

++

- exploitation d'un espace latent pour naviguer dans les caractéristiques du visage

- adapté pour de la génération de contenu

- permettent d'éditer certaines caractéristiques bien précises

--

- extrêmement sensible au contexte d'entraînement

- pas facile à mettre en place

État de l'art

Modèles hybrides / diffusion + GAN

++

- contrôle plus profond sur les résultats générés

- amélioration sémantiques des images générées

--

- assez nouveaux dans le domaine de la recherche

- entraînement lent

État de l'art

Quelques papiers de références

RigFace [1] : utilise un encodeur d'attributs + un modèle de diffusion pré-entraîné pour éditer des portraits tout en conservant l'identité

StyleDit [2] : explorent la synthèse de visages en combinant les "prior facial" de StyleGAN et des modèles de diffusion pour permettre plus de diversité tout en gardant un bon réalisme

[1] RigFace Wei et al.

[2] StyleDit Chiu et al.

Méthode proposé #1 : utilisation d'un transfert de style entre 2 datasets

image issue de celebA

image issue d'ONOT

Tenter d'extraire les caractéristiques des visages et poses depuis l'espace latent et essayer d'aligner entre les 2 datasets

Méthode proposé #1 : utilisation d'un transfert de style entre 2 datasets

Capture source

Première itération du modèle

Méthode proposé #1 : utilisation d'un transfert de style entre 2 datasets

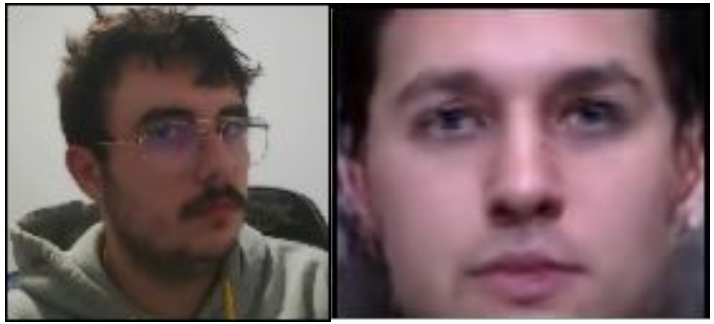
Style GAN

VAE + GAN

Méthode proposé #2 : utilisation d'un modèle déjà entraîné

- utilisation du SEUL modèle trouvable en ligne de frontalization **(scaleway) [3]**

- résultats pas très convaincants malgré la reconstruction partiel d'un visage



[3] face-frontalization - scaleway

Méthode proposé #3 : élaboration de notre propre modèle

- création de notre propre modèle basé sur le papier suivant : **Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis proposé par Rui Huang et al. [4]**

- création de l'architecture

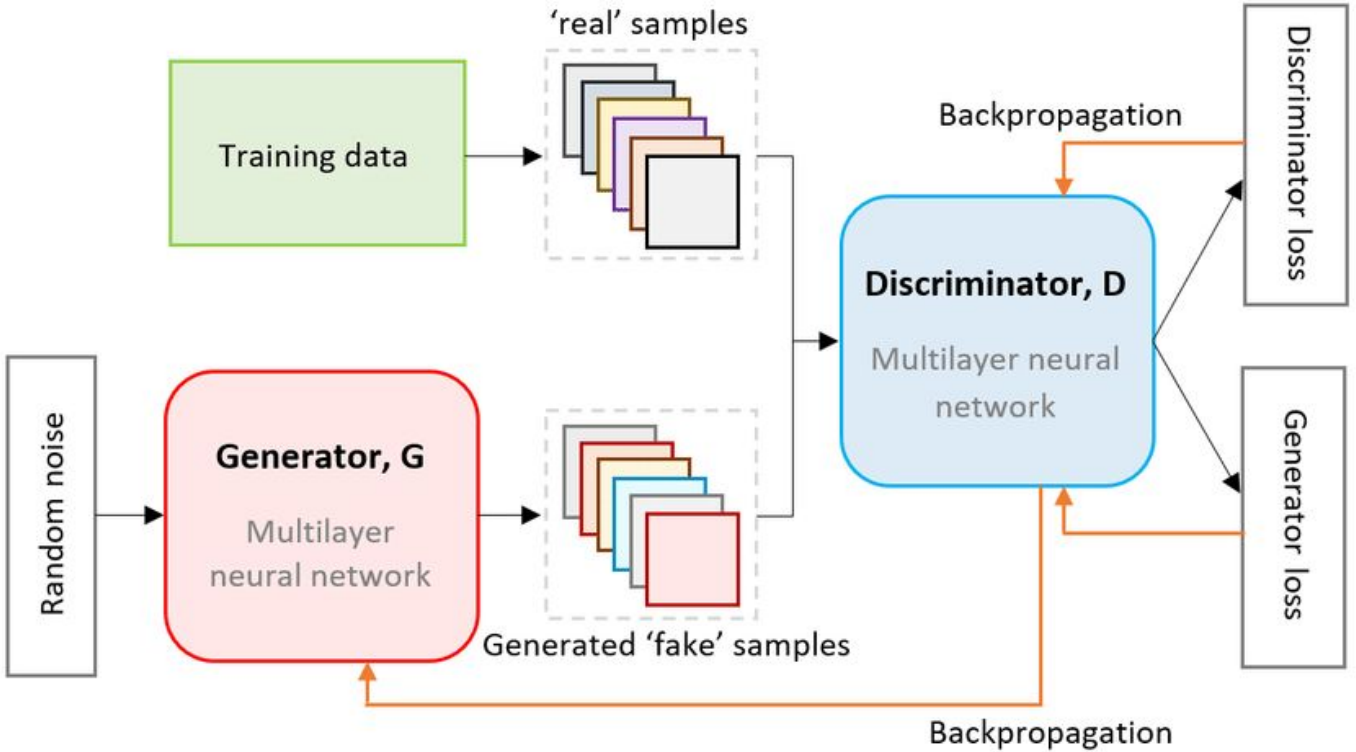
- phase d'entraînement et ajustement des poids

[4] Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis. Rui Huang et al.

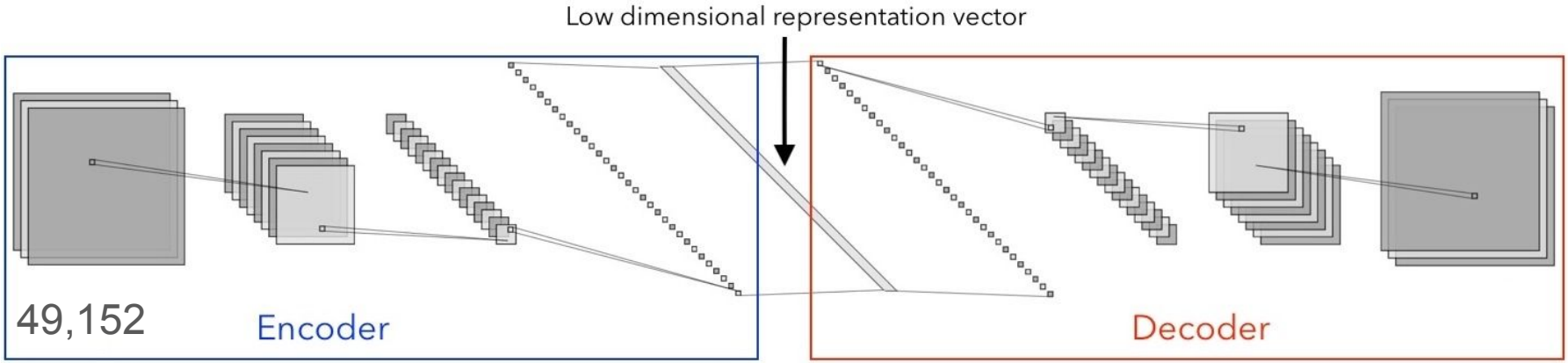
Méthode proposé #3 : élaboration de notre propre modèle

Générateur (G)
Rôle : Transformer les images de profil en images frontales
Architecture : Encoder-Decoder (Auto-encoder)

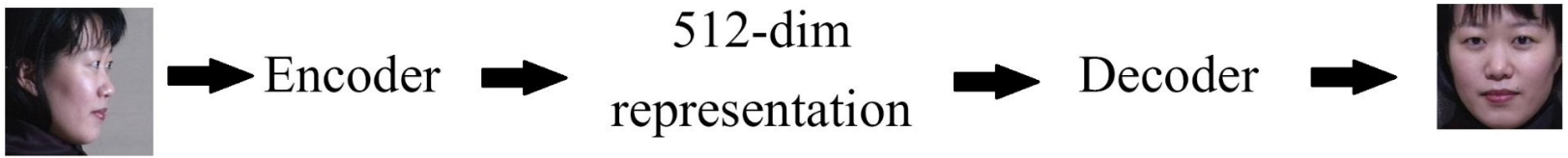
Discriminateur (D)
Rôle : Distinguer les vraies images frontales des images générées
Architecture assez similaire au Générateur



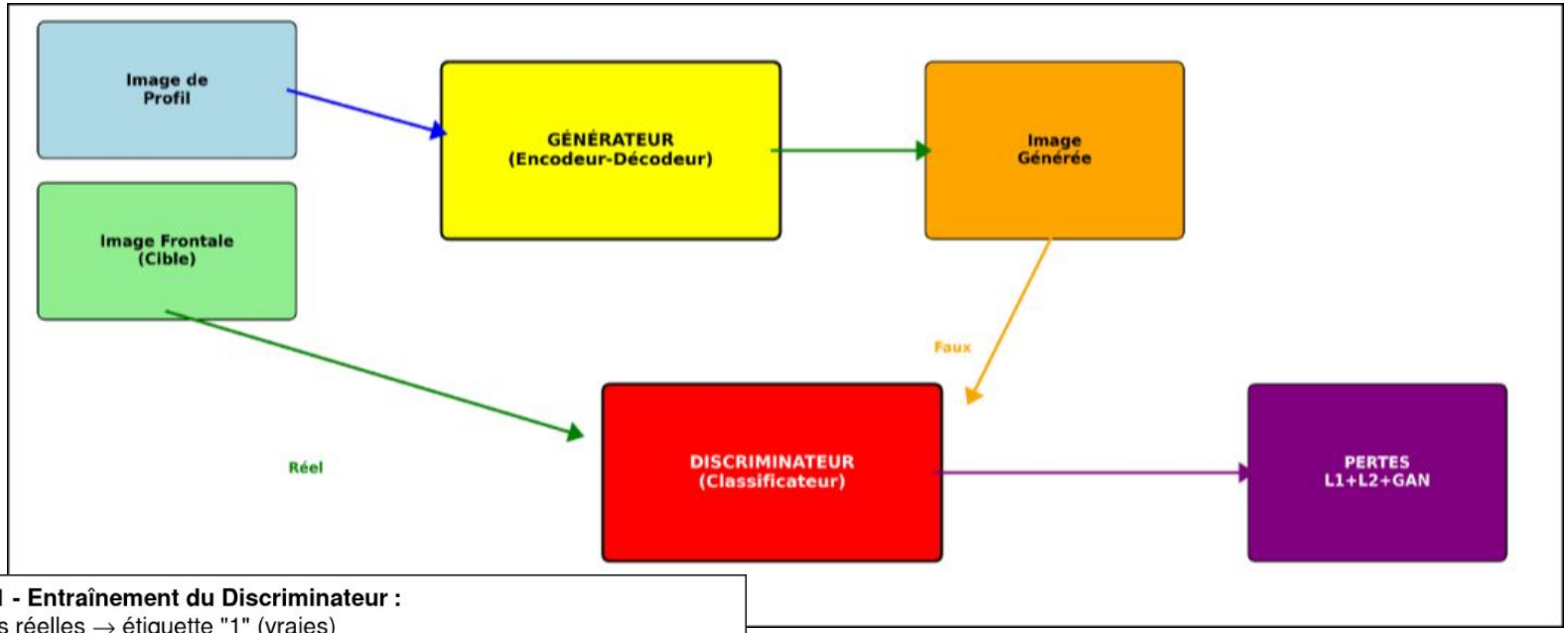
Architecture du Générateur



≃ réduction de 98.9% des paramètres



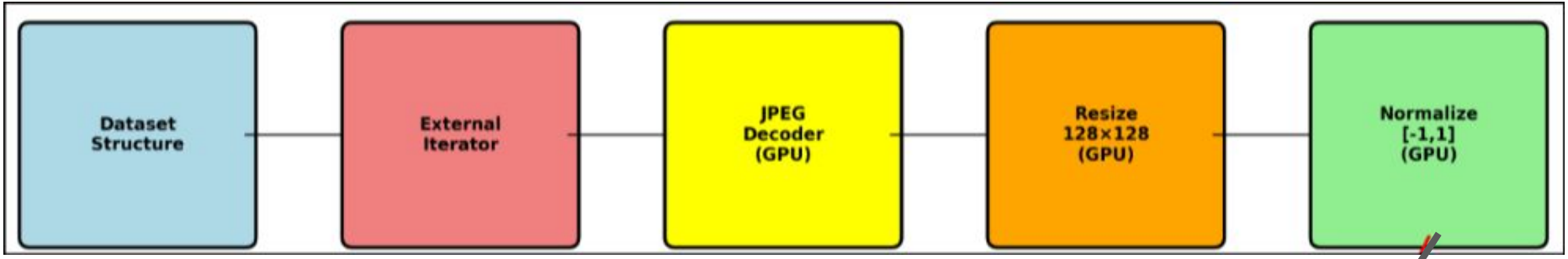
Flux d'entraînement du GAN



Phase 1 - Entraînement du Discriminateur :
- Images réelles → étiquette "1" (vraies)
- Images générées → étiquette "0" (fausses)
- Optimisation via Binary Cross Entropy Loss

Phase 2 - Entraînement du Générateur :
- Objectif double : tromper le discriminateur ET ressembler aux images cibles
- Fonction de perte hybride : $L_{total} = \alpha \times L_{GAN} + \beta \times L_{L1} + \gamma \times L_{L2}$
- Mise à jour des poids via rétropropagation

Préparation du dataset pour l'entraînement



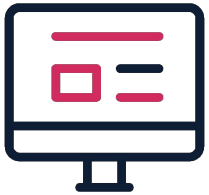
IDOC Mugshots

Structure du Dataset:

```
dataset/
    ├── 001.jpg  (frontal)
    ├── 002.jpg  (frontal)
    ├── 001/
    |    ├── profile1.jpg
    |    └── profile2.jpg
    └── 002/
```

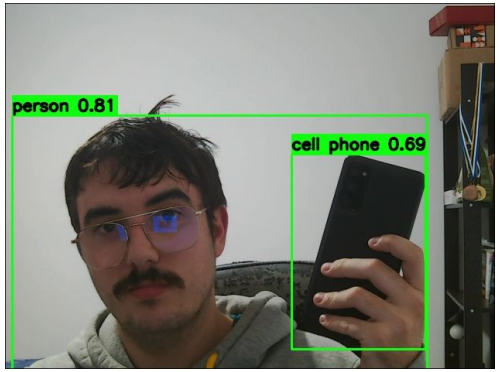
[5] IDOC Mugshots dataset

Implémentation de l'architecture de notre projet



Interface

Lecture Webcam



Demande d'un mugshot

Passage dans notre classifieur pour détecter si profil ou face

Envoi de l'image dans notre modèle pré-entraîné

Comparaison de nos résultats via différentes architectures

Première architecture issue du papier

VAE + GAN

GAN + UNET

Comparaison de nos résultats via différentes architectures

StyleGAN

Résultats via notre architecture

Input

Output

Ground truth

Comprendre comment adapter notre modèle

L_GAN (Perte adversariale)

- binary Cross Entropy pour tromper le discriminateur
- force la génération photoréaliste

L_L1 (Mean Absolute Error)

- comparaison pixel/pixel avec l'image cible
- sert à préserver les détails structurels

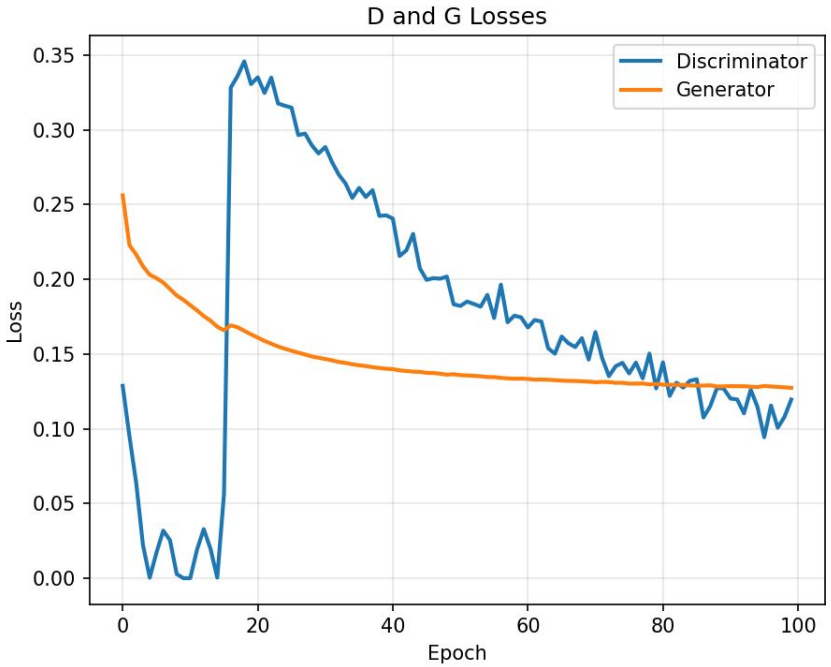
L_L2 (Mean Squared Error)

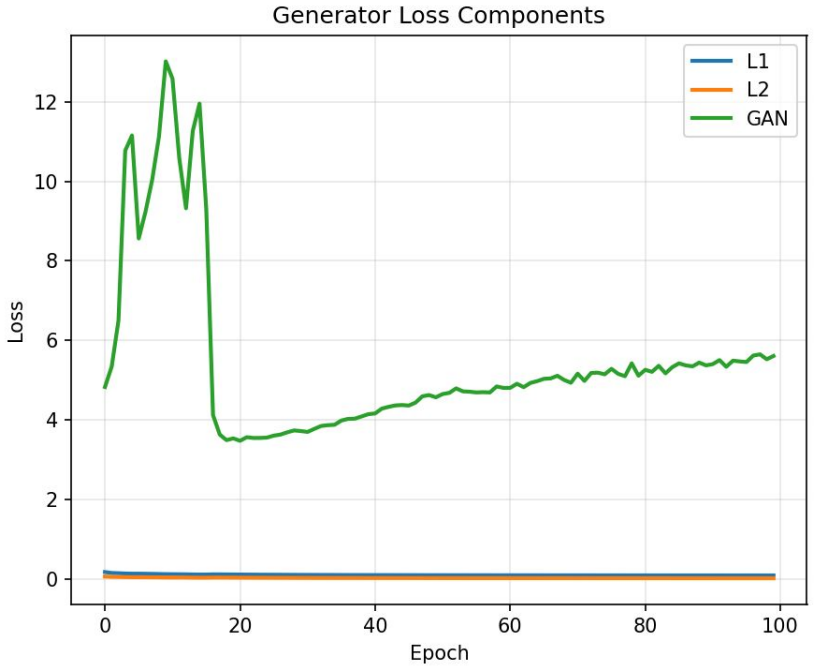
- pénalise fortement les grandes erreurs
- sert à améliorer la fidélité globale

Utilisé dans notre fonction de perte hybride :

$$L\_total = \alpha \times L\_GAN + \beta \times L\_L1 + \gamma \times L\_L2$$

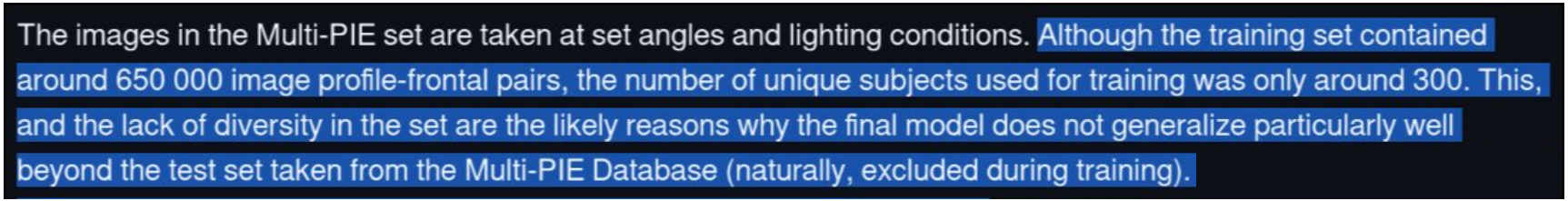
Comprendre comment adapter notre modèle





Pourquoi ca ne marche pas ?

- Modèle de démonstration de leur services de prestation d'IA
- Leur résultats proviennent d'une entraînement sur plusieurs jours complet avec 8 MILLIONS d'images issu du **Multi-PIE dataset [6]** (dataset payant distribué par une société)
- Le modèle présente lui aussi des limitations comme l'indique scalway


The images in the Multi-PIE set are taken at set angles and lighting conditions. Although the training set contained around 650 000 image profile-frontal pairs, the number of unique subjects used for training was only around 300. This, and the lack of diversity in the set are the likely reasons why the final model does not generalize particularly well beyond the test set taken from the Multi-PIE Database (naturally, excluded during training).

[6] Multi-PIE dataset

Pourquoi ca ne marche pas ?

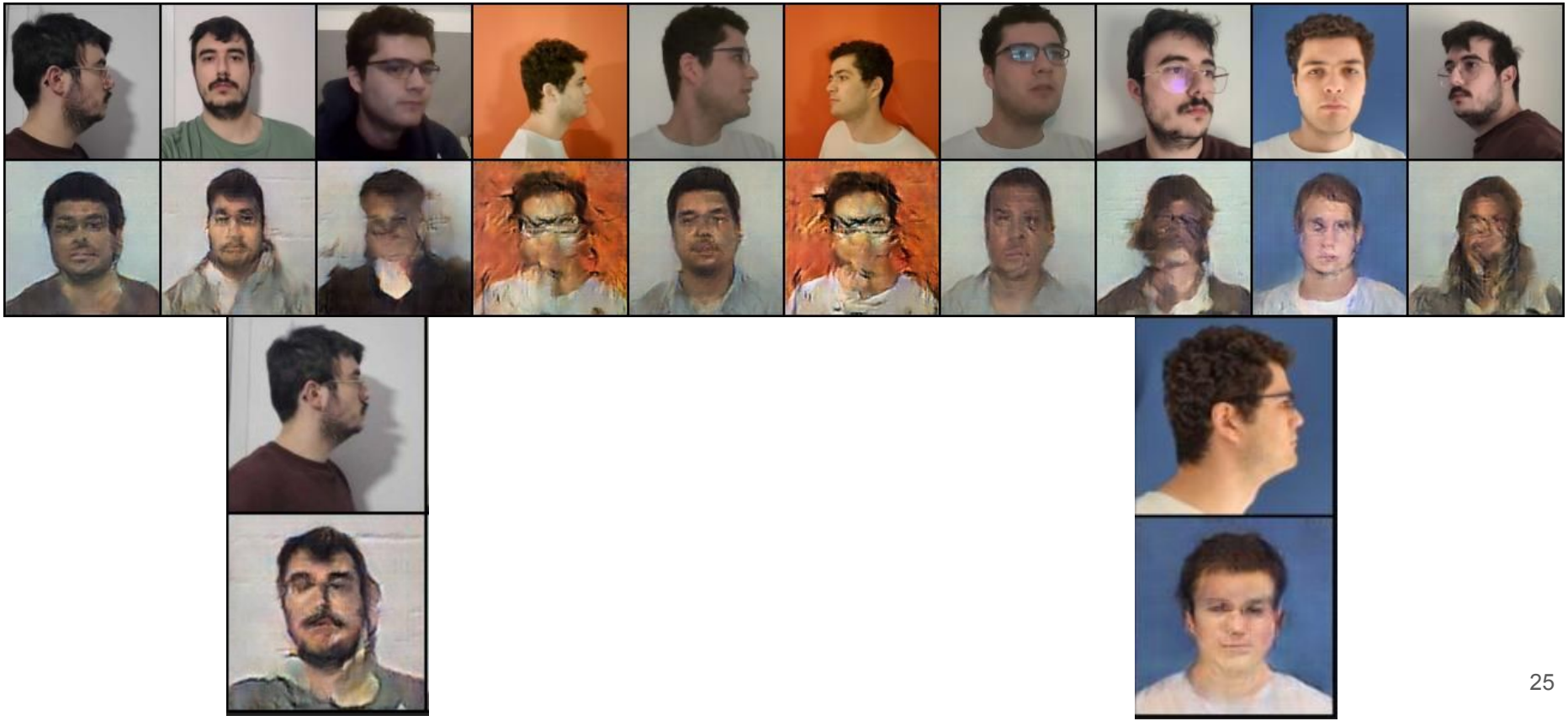
- Temps d'entraînement faramineux requis
- Des limitations énoncées dans le papier
- Sujet complexe et en cours de recherche

Expected training times for the default configuration using Tesla V100 GPUs:

GPUs	1024×1024	512×512	256×256
1	41 days 4 hours	24 days 21 hours	14 days 22 hours
2	21 days 22 hours	13 days 7 hours	9 days 5 hours
4	11 days 8 hours	7 days 0 hours	4 days 21 hours
8	6 days 14 hours	4 days 10 hours	3 days 8 hours

Tentative avec notre propre Multi-PIE dataset

Tentative avec notre propre Multi-PIE dataset



Démonstration

Conclusion et pistes d'amélioration

- Projet vraiment complexe

- Peu de ressources en ligne, donc obligé de tout faire maison

- Entraînement des modèles de génération d'images vraiment fastidieux

- Création de visage +/- réaliste

Pistes d'amélioration

- Essayer de se procurer le dataset **[4]** utilisé dans notre papier

- Ajuster l’architecture et si pas de signe de sur/sous apprentissage laisser tourner pendant un temps conséquent : **Pix2Pix [7]**

- Continuer notre piste de notre propre Multi-PIE dataset en standardisant les prises de vue

[7] Alternative Face

Merci de votre écoute !