

Approche collaborative et répartition des tâches

Projet : Reconstruction du modèle Fader Networks

1. 1. Organisation générale du projet

Le projet est divisé en deux sous-groupes travaillant en parallèle et en coordination :

- **Équipe A — Données et Validation** : responsable de la collecte, du prétraitement, de la préparation des données et de la validation sur un modèle préentraîné.
- **Équipe B — Implémentation et Modélisation** : responsable de la reconstruction complète du modèle Fader Networks, de son entraînement et de la comparaison avec le modèle officiel.

Une phase finale de synchronisation permettra de tester, comparer et analyser les performances entre le modèle officiel et celui réimplémenté par l'équipe.

2. 2. Équipe A — Données et Validation

Objectif général : Préparer un pipeline de données propre, exploitable et validé par un test initial sur un modèle Fader Networks préentraîné.

Sous-tâche A1 — Téléchargement et organisation du dataset

- Télécharger le dataset **CelebA**.
- Créer une structure de dossiers claire :

```
data/  
  img_align_celeba/  
  list_attr_celeba.txt  
  processed/
```

- Vérifier l'intégrité des fichiers (nombre, cohérence).
- Créer un script d'importation automatique.

Sous-tâche A2 — Prétraitement des images

- Rogner les visages pour centrer la région d'intérêt.
- Redimensionner toutes les images à 256×256 .
- Normaliser les valeurs des pixels dans l'intervalle $[-1, 1]$.
- Enregistrer les images traitées dans le dossier `processed/`.

Sous-tâche A3 — Traitement des attributs

- Charger le fichier `list_attr_celeba.txt`.
- Sélectionner un sous-ensemble d'attributs pertinents :
 - *Smiling, Eyeglasses, Male, Young, etc.*
- Convertir les labels $\{-1, 1\}$ en $\{0, 1\}$.
- Sauvegarder un fichier de correspondance : `image_name` \rightarrow `attributs`.

Sous-tâche A4 — Création du DataLoader PyTorch

- Implémenter une classe `CelebADataset`.
- Gérer les partitions `train` / `val` / `test`.
- Intégrer les transformations d'image (`Resize`, `ToTensor`, `Normalize`).
- Tester le DataLoader et afficher un batch d'images avec attributs.

Sous-tâche A5 — Test sur un modèle préentraîné

- Télécharger le modèle Fader Networks préentraîné : <https://github.com/facebookresearch/FaderNetworks>
- Charger les poids fournis par les auteurs.
- Tester le pipeline de données préparé :
 - Reconstruction d'images (sans modification d'attributs).
 - Modification d'un attribut (ex. ajout/suppression de lunettes).
- Évaluer la cohérence visuelle et la naturalité des résultats.
- Enregistrer les images avant/après modification.

Sous-tâche A6 — Analyse et documentation

- Étudier la distribution des attributs (histogrammes).
- Identifier les biais éventuels dans les données.
- Documenter :

- les étapes de traitement,
- les problèmes rencontrés,
- les choix effectués (attributs retenus, formats, etc.).
- Créer un rapport technique sur la qualité des données.

Livrables de l'équipe A

- Dataset prêt à l'emploi et bien documenté.
- Pipeline complet de chargement et de prétraitement.
- Résultats obtenus avec le modèle préentraîné (images, observations).
- Rapport d'analyse sur la qualité du dataset et la validité du pipeline.

3. 3. Équipe B — Implémentation et Modélisation

Objectif général : Recréer à partir de zéro le modèle Fader Networks, l'entraîner sur les données préparées par l'équipe A, puis comparer les résultats avec le modèle officiel.

Sous-tâche B1 — Mise en place du projet

- Créer la structure du projet :

```
FaderNetworks/
  models/
  training/
  utils/
  outputs/
  main.py
```

- Configurer l'environnement Python (PyTorch, Numpy, Matplotlib, etc.).
- Tester la disponibilité du GPU et la compatibilité CUDA.

Sous-tâche B2 — Implémentation des modules principaux

- **Encoder :**
 - Réseau CNN descendant (C16→C512).
 - Activation LeakyReLU et BatchNorm.
- **Decoder :**
 - Réseau de transposed convolutions symétrique.
 - Intégration des attributs dans chaque couche.

- Activation finale : Tanh.
- **Discriminator :**
 - Couche conv + 2 couches fully connected.
 - Dropout (0.3) pour régularisation.

Sous-tâche B3 — Fonctions de perte

- Reconstruction loss :

$$L_{AE} = ||D(E(x), y) - x||^2$$

- Discriminator loss :

$$L_{dis} = -\log P(y|E(x))$$

- Adversarial loss (encoder) :

$$L_{enc} = L_{AE} - \lambda \log P(1 - y|E(x))$$

- Implémenter la progression de λ ($0 \rightarrow 0.0001$).

Sous-tâche B4 — Entraînement adversarial

- Boucle d'apprentissage :
 1. Entraîner le discriminateur à prédire les attributs.
 2. Entraîner l'encodeur/décodeur à le tromper tout en reconstruisant.
- Sauvegarder régulièrement les checkpoints.
- Afficher et enregistrer les pertes L_{AE} , L_{dis} , L_{enc} .

Sous-tâche B5 — Évaluation et comparaison

- Utiliser les données validées par l'équipe A.
- Effectuer :
 - Reconstruction d'images.
 - Swap d'attributs (ex. lunettes, sourire).
- Comparer avec le modèle préentraîné :
 - Réalisme visuel.
 - Fidélité de la reconstruction.
 - Pertinence du changement d'attribut.
- Générer des figures comparatives (avant/après, officiel vs nouveau).

Sous-tâche B6 — Documentation technique

- Documenter :
 - l’architecture des réseaux (schémas),
 - les hyperparamètres d’entraînement,
 - les configurations de tests,
 - les résultats obtenus.
- Produire un rapport technique clair pour la section “Implémentation”.

Livrables de l’équipe B

- Code complet du modèle Fader Networks réimplémenté.
- Scripts d’entraînement et d’évaluation.
- Modèles sauvegardés (checkpoints).
- Comparaison graphique avec le modèle officiel.
- Rapport technique détaillant architecture, pertes et résultats.

4. 4. Phase commune — Intégration et comparaison

Objectif général : Connecter le pipeline de données (équipe A) au modèle réimplémenté (équipe B), comparer les performances et rédiger la synthèse commune.

Sous-tâche C1 — Intégration

- Tester le modèle réimplémenté sur le pipeline de données.
- Vérifier la compatibilité entre le DataLoader et le modèle.
- Corriger les éventuelles incohérences de formats.

Sous-tâche C2 — Comparaison

- Comparer les résultats :
 - Fader Networks officiel vs Réimplémentation.
 - Reconstruction (fidélité), swap (efficacité), réalisme (subjectif).
- Analyser les écarts et interpréter les différences.

Sous-tâche C3 — Rapport final et présentation

- Rédiger le rapport complet :
 1. Introduction et contexte.

2. Préparation des données.
 3. Implémentation et entraînement.
 4. Comparaison et analyse.
 5. Conclusion et perspectives.
- Créer une présentation synthétique :
 - Pipeline de données (équipe A).
 - Modèle et résultats (équipe B).
 - Comparaison finale (travail commun).

Livrables finaux

- Dataset final et pipeline validé.
- Modèle reconstruit et entraîné.
- Comparaison complète (visuelle et analytique).
- Rapport final et présentation commune.

Résumé global

- Équipe A — Données et Validation :
 - Préparation, nettoyage, test sur modèle préentraîné.
 - Validation du pipeline de données.
- Équipe B — Implémentation et Modélisation :
 - Reconstruction du modèle Fader Networks.
 - Entraînement et évaluation complète.
- Phase commune :
 - Intégration des deux travaux.
 - Comparaison et rédaction finale.