

Exercices Semaine 18

Cours Turing

1 Mini-projet

Le squelette du mini-projet contient 5 fichiers python :

- **train.py** : Script principal qui permet d'entraîner le modèle.
- **dataset.py** : Script qui définit le dataset en utilisant les classes de Python.
- **utils.py** : Script qui contient des fonctions utilitaires pour alléger le code.
- **UNET_model.py** : Script qui contient la définition du modèle U-Net. Tiré de [ce dépôt](#).
- **UNET_parts.py** : Script qui contient les parties du modèle U-Net. Tiré de [ce dépôt](#).
- **Commande.pour.installer.les.librairies.txt** : Contient la commande à exécuter dans le terminal pour installer les librairies nécessaires.

Mais également 4 dossiers :

- **TRAIN_images** : Contient des images en couleur. Les images de ce dossier seront utilisées pour l'entraînement du modèle.
- **TEST_images** : Contient des images en couleur. Les images de ce dossier seront utilisées pour tester le modèle après chaque époque d'entraînement.
- **OUTPUT** : Contendra les résultats de l'entraînement du modèle : graphe de la perte, images colorisées et modèle entraîné.
- **Images** : Contient des images en couleur qui ne sont pas exploitées

Chaque image de TEST_IMAGES sera colorisée après chaque époque d'entraînement et sauvegardée dans le dossier OUTPUT dans une image constituée de 3 sous-images :

- L'image en noir et blanc
- L'image colorisée par le modèle
- L'image originale

Questions

1. Commencez par répondre aux questions suivantes :

- (a) Dans le dossier **TRAIN_images** :
 - Combien d'images sont présentes ?
 - Quelle est la taille des images ?
 - Quel est le point commun entre toutes les images ?
- (b) Dans le dossier **TEST_images** :
 - Combien d'images sont présentes ?
 - Quelle est la taille des images ?
 - Anticipez-vous un problème ?
- (c) Dans le script **train.py** :
 - Quel est le taux d'apprentissage (LEARNING_RATE) ?
 - Quelle est la taille des paquets (BATCH_SIZE) ?
 - Quelle est l'inertie (MOMENTUM) ?
 - Quelle est la fonction de perte utilisée pour l'entraînement ?
 - Combien d'époques sont prévues pour l'entraînement ?

2. Lancez le script **train.py** et observez les résultats dans le dossier **OUTPUT** :

- Combien de temps a duré l'entraînement ?
- Quelle est la valeur finale de la fonction de coût ?

- Quelle est la qualité des images colorisées ?

3. **Que pouvez-vous faire pour améliorer les performances du modèle ?**

Il s'agira maintenant de trouver un bon compromis entre la qualité des images colorisées et le temps d'entraînement du modèle.

1. Quelle est le plus grand taux d'apprentissage (LEARNING_RATE) que vous pouvez utiliser sans que le modèle diverge ?
2. Quelle la plus grande taille de paquet (BATCH_SIZE) que vous pouvez utiliser ? Comment cela impact la vitesse d'entraînement ?
3. Quel est l'effet de l'inertie (MOMENTUM) sur l'entraînement du modèle ?
4. Arrivez-vous à obtenir des images colorisées satisfaisantes pour les images initialement dans le dossier **TEST_images** ?
5. Que se passe-t-il si vous utilisez des images de taille différente dans le dossier **TEST_images** ?
6. Que se passe-t-il si vous utilisez des images très différentes des données d'entraînement dans le dossier **TEST_images** ? Vous pouvez essayer avec images de bande dessinée, visages, véhicules, etc...