Plan prévisionnel

Dataset retenu

L'option sélectionnée est de nous appuyer sur le jeu de données *Cityscapes*. C'est un dataset d'images et d'annotations pour l'entraînement sur des tâches de segmentation sémantique ou d'instance, de génération d'images, etc.

Pour notre étude nous nous en servirons pour de la segmentation sémantique.

Il est séparé en deux parties : les images, contenues dans `P8_Cityscapes_leftImg8bit_trainvaltest`, et les *masks* (notre target), contenus dans `P8_Cityscapes_gtFine_trainvaltest`.
Le jeu de données initial est composé de 4 types d'informations de *mask* :

- *Masks* couleur : chaque classe se voit attribuer une couleur différente.
- *Masks* d'identifiant d'instance : masques en niveaux de gris étiquetés avec des identifiants d'instance (chaque élément différent se voit attribuer une classe et un identifiant d'objet unique)
- Identifiants d'étiquette : *mask* en niveaux de gris étiqueté avec l'identifiant de classe. Comme nous souhaitons faire de la segmentation sémantique, ce sont ces fichiers que nous utilisons
- Polygones de *masks* : incluent les coordonnées détaillées de chaque segment.

Un découpage préalable *Train*/*Validation*/*Test* a été effectué :

- Entraînement : 2975 images / masks
- Validation : 500 images / masks
- Test : 1525 images / masks

Le souci est que les *masks* de Test ne sont pas exploitables, avec des identifiants d'étiquettes absents.
Nous utilisons donc seulement les données `train` et `val` pour notre travail.

Modèle envisagé

Le modèle envisagé est le *SegFormer*, un modèle de segmentation sémantique basé sur la technologie *Transformer*.

Ce choix est guidé par la **performance** : Dans l'étude comparative [1] datant de décembre 2023, cet algorithme est deuxième en termes de résultat sur le *dataset CityScapes* (indice de Jaccard - *Mean IoU*).

C'est aussi une solution **pragmatique** : Les modèles basés sur les *Transformers* sont coûteux à entraîner, là où le *SegFormer* est plus simple (comme présenté dans le chapitre *3 Méthode* de l'article initial [2]), le tout avec moins de paramètres. L'algorithme est disponible sur la plateforme *Hugging Face* et dispose d'une version compatible avec le *framework TensorFlow*. Il s'agit d'une compatibilité, et son utilisation nécessite tout de même d'adapter la modélisation sur plusieurs aspects (générateur d'images/*masks*, format des images, entraînement).

Références bibliographiques

[1] Semantic Segmentation using Vision Transformers : A survey
par Hans Thisanke, Chamli Deshan, Kavindu Chamith, Sachith Seneviratne, Rajith Vidanaarachchi, Damayanthi Herath (2023)
Cet article est une étude comparative de différents algorithmes utilisant les Vision Transformers (ViT). Celle-ci rappelle le principe général des ViT avec l'utilisation du mécanisme d'attention non pas sur chaque pixel mais sur des patchs d'images. L'approche de chaque modèle, ainsi que sa manière d'appréhender la segmentation sémantique, sont décrites.

Enfin **un tableau comparatif** permet de confronter les performances sur différents jeux de données, et notamment le *dataset Cityscapes*.
Une recherche des possibilités pratiques d'implémentation des meilleurs algorithmes a conduit au choix du *SegFormer*.

[2] SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers
par Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, Ping Luo (2021)
Il s'agit de l'article initial qui a présenté la technique. Celle-ci utilise la force du mécanisme d'attention, déjà adapté à la Computer Vision avec le Vision Transformer, et introduit une structure nouvelle :

- *Hierarchical Transformer Encoder* : Il permet de générer des *Feature maps* de différentes résolutions, pour mettre à disposition du *decoder* de segmentation des *features* de différentes tailles et reflétant des informations aussi bien locales que globales
- *Lightweight All-MLP Decoder* : Plutôt que de réaliser un *upscaling* successif classique, toutes les *features* de différentes échelles sont utilisée simultanément dans un Perceptron Multicouche classique, avec pour idée directrice de combiner l'attention locale et l'attention globale pour une représentation plus puissante.

Plus de détails seront présents dans la note méthodologique.

[3] MedAI #32: Simple & Efficient Design for Semantic Segmentation with Transformers
https://www.youtube.com/watch?v=Yf9fNn1fWy8&t=380s
par Enze Xie
Une présentation vidéo du *Segformer* par un de ses concepteurs.

[4] Semantic segmentation with SegFormer and Hugging Face Transformers
https://keras.io/examples/vision/segformer/
par Sayak Paul (2023)
Une implémentation du *Segformer* avec le *framework Tensorflow*. Cet article permet de comprendre l'utilisation du modèle mis à disposition par *Hugging Face* : `TFSegformerForSemanticSegmentation`. Il offre notamment la possibilité de bien appréhender la création du *dataset*.

[5] Semantic segmentation for Attribution methods tutorial
https://colab.research.google.com/drive/1AHq7KO1fCOX5nZLGZfxkZ2-DLPPdSfbX#scrollTo=0c7d9d33-2de5-4f03-a793-a76dd2b1e238
par Antonin Poché (2023)
Une mise en pratique pour une librairie spécialisée dans l'explicabilité des modèles de *computer vision* : `Xplique`.

Explication de la démarche de test du nouvel algorithme (la preuve de concept)

Afin de comparer le *Segformer*, nous utiliserons comme *baseline* un modèle classique n'utilisant pas de *ViT*. Nous prenons un *Unet* avec pour backbone un *Resnet*. Ce modèle avait obtenu les meilleurs résultats dans le projet précédent. L'objectif est de pouvoir comparer les deux modèles sur des bases assez proches, aussi un effort sera fait pour rapprocher les deux configurations :

- taille des images pour l'entraînement
- nombre de paramètres de l'algorithme
- fonction de perte
- nombre d'*epochs*

On évalue les performances sur une métrique commune, ainsi que visuellement avec des tests de segmentation sur des exemples d'images issus du *dataset*.

Afin de donner à voir les résultats, nous implémenterons une application simple sous la forme d'un *dashboard*, avec un parcours utilisateur simple :

- un onglet permettant d'explorer quelques images du *dataset CityScapes* et d'afficher :
    - l'image
    - son *mask*
    - un graphique interactif avec la répartition des pixels par macro-catégorie
    - un graphique similaire pour l'ensemble du *dataset*
- un onglet permettant de tester le *SegFormer* sur ces mêmes images de test :
    - l'image
    - son *mask*
    - le résultat de la segmentation avec la Baseline *Unet-Resnet*
    - le résultat de la segmentation avec le *SegFormer*
    - un graphique interactif avec les performances sur la métrique d'évaluation
- un troisième onglet sur l'explicabilité permettant, sur un ou deux exemples, d'afficher le résultat visuel d'une interprétation

Nous veillerons à prendre en compte certains critères d'accessibilité du WCAG (*Web Content Accessibility Guidelines*).