

Travail pratique #4

Pipelines de traitement d'images

INF600F - Traitement d'images

Hiver 2022

Table des matières

Pondération	1
Exercice 1 : Reconnaissance de texte (15 pts)	1
Partie 1 : Extraction des lettres (10 pts)	2
Partie 2 : Partitionnement non supervisé des lettres (5 pts)	2
Exercice 2 : Compter des cellules (10 pts)	3
Références	3

Ce travail comporte 2 exercices et il vaut pour 12.5% de la note finale. Les données et le notebook à utiliser pour effectuer ce travail pratique se trouvent dans l'archive ZIP de ce TP disponible sur le site web du cours. Consultez également le document Instructions générales pour les travaux pratiques sur Moodle pour connaître les exigences du rapport de laboratoire.

Pondération

- **Exercice 1** : Reconnaissance de texte (15 pts)
- **Exercice 2** : Compter des cellules (10 pts)
- **Total** : 25 points

Exercice 1 : Reconnaissance de texte (15 pts)

Utilisez l'image *tp4_ex1.png* pour cet exercice.

Le but est de mettre en place un pipeline de traitement d'images afin d'extraire toutes les lettres d'une image contenant du texte. Chaque lettre doit être placée dans une image séparée et conservée dans une liste Python. Pour effectuer cet exercice, des notions de segmentation d'images et de morphologie mathématique seront utilisées.

Partie 1 : Extraction des lettres (10 pts)

Pour extraire les lettres dans une image, implémentez le pipeline de traitement d'images suivant.

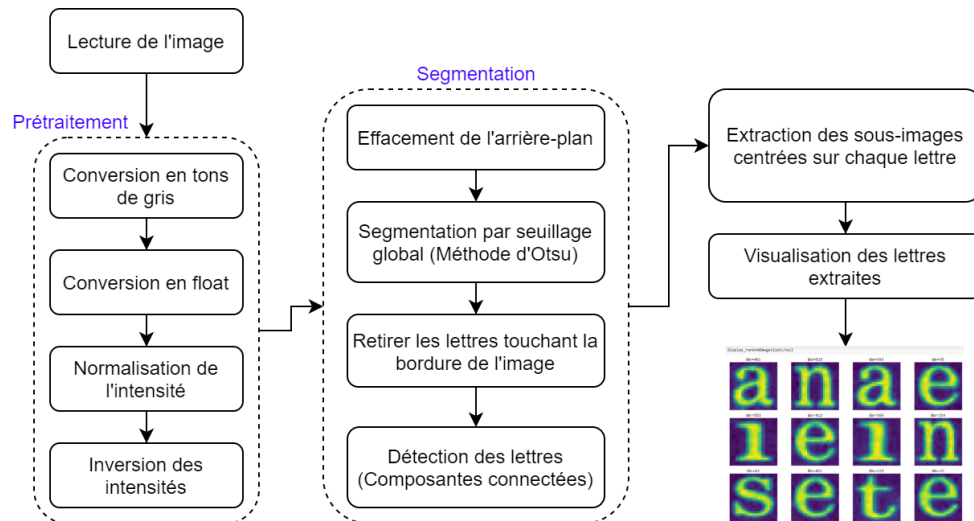


FIG. 1 : Pipeline pour l'extraction des lettres

- Pour l'effacement de l'arrière-plan, vous devez utiliser la morphologie mathématique en tons de gris. Utilisez les fonctions du module **skimage.morphology**
- Afin de s'assurer que votre méthode extrait des lettres complètes, vous devez retirer les lettres touchant la bordure de l'image à l'aide d'opérations morphologiques binaires.
- Pour l'extraction d'une image centrée sur chacune des lettres détectées dans l'image, vous devez effectuer une boucle sur toutes les étiquettes (*labels*) détectées à la dernière étape de la segmentation. Vous devez ensuite rogner l'image pour ne conserver que la lettre associée à chaque étiquette. Ajoutez une marge de 3px autour de la lettre, et assurez-vous que la sous-image soit centrée sur celle-ci. Finalement, vous devez ré-échantillonner les sous-images pour qu'elles aient une taille de 24x24 pixels.
- Pour visualiser vos résultats, créez une fonction nommée **display_randomImages** qui reçoit en entrée la liste des images des lettres extraites et qui affiche 12 lettres au hasard. Utilisez le module **numpy.random** pour choisir au hasard des lettres à afficher.
- Commentez votre code et expliquez votre démarche.
- **Question** : Combien de lettres avez-vous détectées dans cette image ?
- **Question** : En observant les lettres segmentées, pouvez-vous identifier des situations pour lesquelles le pipeline de traitement d'images proposé plus haut échoue à extraire une lettre isolée ? Que suggérez-vous pour améliorer les performances de l'extraction des lettres ?

Partie 2 : Partitionnement non supervisé des lettres (5 pts)

- Utilisons l'algorithme K-moyennes pour regrouper les lettres en classes similaires. Pour ce faire, utilisez le module python **scikit-learn** et sa fonction **sklearn.cluster.KMeans** pour effectuer le partitionnement. Partitionnez les images en 26 classes, observez ensuite les résultats de partitionnement en affichant les images associées à chaque classe et l'image moyenne par classe. Discutez des résultats obtenus.

- Utilisez l'histogramme des orientations de gradient (`skimage.feature.hog`) en tant que descripteur pour le partitionnement.
- **Note** : pour effectuer le partitionnement avec `sklearn.cluster.KMeans`, vous devez créer une matrice `X` de taille (nImages x nCaractéristiques). Ensuite, vous pouvez appliquer la méthode `KMeans` de la façon suivante : `modele = KMeans(n_clusters=26).fit(X)`. Après le partitionnement, la classe associée à chaque image est donnée dans la liste `modele.labels__`

Exercice 2 : Compter des cellules (10 pts)

Utilisez l'image `tp4_ex2.tif` pour cet exercice.

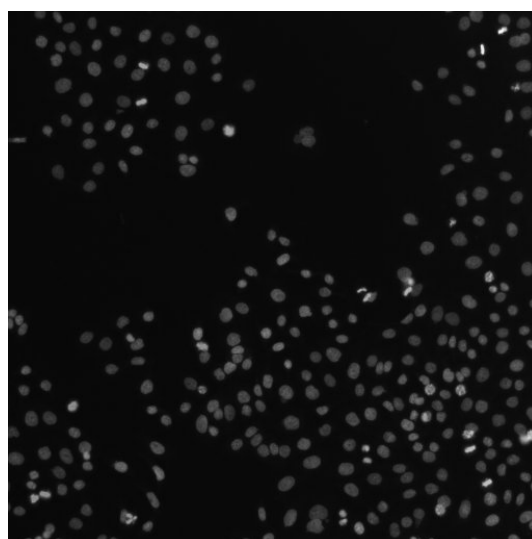


FIG. 2 : Image de cellules humaines HT29 pour l'exercice 2

Vous avez une connaissance qui travaille dans un laboratoire de biologie. Elle souhaite obtenir une application pour faciliter le dénombrement des cellules dans une image de microscopie. Vous disposez d'images des cellules prises sur fond sombre. Développez une méthode pour estimer le nombre de cellules présentes dans une image. Expliquez votre démarche.

Note : Il n'est pas nécessaire de trouver la position exacte de chaque cellule, seulement le nombre est explicitement demandé. Pour vérifier que votre méthode fonctionne bien, vous pouvez la tester avec une petite sous image pour laquelle vous pouvez manuellement compter le nombre de cellules.

Références

- L'image utilisée pour l'exercice 1 est une photographie de l'article : Booth, J. M., & Schroeder, J. B. (1977). Design Considerations for Digital Image Processing Systems. *Computer*, 10(8), 15–20. <https://doi.org/10.1109/C-M.1977.217814>
- L'image utilisée pour l'exercice 2 représente des cellules humaines HT29 (une lignée cellulaire de cancer du colon). Elle provient de <https://cellprofiler.org/examples>.