

TP : Segmentation d'images

24 octobre 2023

Table des matières

| | | |
|----------|---|-----------|
| 1 | Introduction | 2 |
| 1.1 | Segmentation en imagerie médicale | 2 |
| 1.2 | Smart Agriculture | 2 |
| 1.3 | Portée du TP | 2 |
| 1.4 | Objectifs du TP | 3 |
| 1.5 | Installation des outils nécessaires | 3 |
| 2 | Segmentation basée sur l'intensité (Thresholding) | 4 |
| 3 | Segmentation par croissance de région (Region Growing) | 5 |
| 4 | Segmentation par Clustering | 7 |
| 5 | Segmentation basée sur la couleur | 8 |
| 6 | Exploitation de Datasets Médicaux | 10 |
| 6.1 | Datasets recommandés et leur pertinence | 10 |
| 6.2 | Rendu et Collaboration | 11 |
| 7 | Comparatif des Méthodes de Segmentation | 11 |
| 7.1 | Objectifs | 11 |
| 7.2 | Protocole | 11 |
| 7.3 | Rapport Final | 11 |
| 8 | Conclusion | 12 |

1 Introduction

La segmentation d'images est l'une des étapes fondamentales du traitement d'images et de la vision par ordinateur. Elle consiste à subdiviser une image en différents segments, où chaque segment représente un ensemble de pixels ayant des caractéristiques communes. L'objectif est de simplifier et/ou de modifier la représentation de l'image pour la rendre plus significative et plus facile à analyser. Chaque segment obtenu peut correspondre à une région d'intérêt spécifique ou à un objet individuel présent dans l'image.

L'importance de la segmentation réside dans sa capacité à mettre en évidence des structures pertinentes dans une image, facilitant ainsi l'extraction d'informations utiles ou la compréhension de l'image. En segmentant une image, on peut mieux identifier et isoler des objets ou des zones d'intérêt, ce qui est crucial pour de nombreuses applications.

1.1 Segmentation en imagerie médicale

En imagerie médicale, la segmentation est indispensable pour séparer des structures anatomiques, identifier des zones pathologiques, mesurer des volumes d'organes ou suivre l'évolution d'une maladie. Par exemple, sur une image IRM du cerveau, la segmentation peut aider à distinguer les différentes structures cérébrales et à détecter des anomalies telles que des tumeurs.

1.2 Smart Agriculture

Dans le domaine de l'agriculture intelligente, la segmentation d'images permet de distinguer des éléments tels que des plantes individuelles, des zones de stress hydrique, ou des zones infectées par des maladies. En utilisant des images aériennes capturées par des drones, les agriculteurs peuvent, grâce à la segmentation, obtenir des informations précises sur l'état de leurs cultures et ainsi prendre des décisions éclairées pour optimiser leurs rendements.

1.3 Portée du TP

Tout au long de ce TP, nous explorerons différentes techniques de segmentation d'images, en commençant par des méthodes simples comme le seuillage, pour progresser vers des méthodes plus avancées. Bien que la segmentation puisse être abordée sous de nombreux angles, avec une variété de méthodes disponibles, notre objectif est de vous fournir une base solide sur laquelle vous pourrez construire et élargir vos connaissances dans ce domaine fascinant.

1.4 Objectifs du TP

Au terme de ce TP, les étudiants devraient être en mesure de :

- Comprendre l'importance et les applications pratiques de la segmentation d'images dans divers domaines tels que l'imagerie médicale et l'agriculture intelligente.
- Maîtriser différentes méthodes de segmentation basées sur l'intensité, notamment le seuillage global, le seuillage adaptatif, le seuillage d'Otsu, le seuillage multi-niveaux et le seuillage par hystérésis.
- Appréhender la technique de segmentation par croissance de région et en reconnaître les avantages et inconvénients.
- Comprendre le principe du clustering pour la segmentation d'images et être capable d'implémenter des méthodes basiques comme k-means.
- Distinguer les différentes approches de segmentation basée sur la couleur, et savoir quand et comment les utiliser.
- Développer une aptitude à choisir la technique de segmentation la plus adaptée en fonction des caractéristiques et des exigences d'une image donnée.
- Acquérir une expérience pratique dans la mise en œuvre de ces techniques à l'aide d'outils de traitement d'images tels que OpenCV.

Étapes du TP

1.5 Installation des outils nécessaires

Assurez-vous d'avoir installé Python et les bibliothèques nécessaires : 'numpy', 'matplotlib' et 'scikit-image'.

```
1 pip install numpy matplotlib scikit-image
```

Chargement et affichage de l'image

```
1 import matplotlib.pyplot as plt
2
3 image = plt.imread('votre_image.jpg')
4 plt.imshow(image)
5 plt.axis('off')
6 plt.show()
```

2 Segmentation basée sur l'intensité (Thresholding)

Objectif :

- Comprendre les différentes techniques de seuillage.
- Apprendre à appliquer ces techniques sur des images diverses.
- Comparer les avantages et inconvénients de chaque technique.

Théorie :

1. Seuillage global (Binarisation globale) :

Il s'agit d'une approche simple où l'on fixe un seuil pour toute l'image. Tous les pixels ayant une intensité supérieure au seuil sont mis en blanc, et les autres en noir.

2. Seuillage adaptatif (Binarisation adaptative) :

Dans cette méthode, le seuil est déterminé localement pour chaque pixel en fonction d'une petite région autour de lui. Elle est utile pour des images présentant des variations d'éclairage.

3. Seuillage d'Otsu :

Il s'agit d'une méthode automatisée qui cherche à trouver un seuil qui minimise la variance intra-classe des pixels noirs et blancs.

4. Seuillage multi-niveaux :

Cette technique segmente l'image en plus de deux classes. Plutôt que d'avoir un seul seuil divisant l'image en objets et fond, plusieurs seuils sont utilisés pour segmenter l'image en différentes intensités.

5. Seuillage par hystérésis :

Deux seuils (un haut et un bas) sont utilisés pour déterminer l'appartenance des pixels. Les intensités entre les deux seuils sont déterminées en fonction de leur connectivité avec les intensités sûres.

Pratique :

1. Seuillage global :

```
1 seuil = 127
2 image_binarisee = image > seuil
3 plt.imshow(image_binarisee, cmap='gray')
```

2. Seuillage adaptatif :

```
1 from skimage.filters import threshold_local
2 seuil_adaptatif = threshold_local(image, block_size=35,
    offset=10)
```

```

3 image_binarisee = image > seuil_adaptatif
4 plt.imshow(image_binarisee, cmap='gray')

```

3. Seuillage d'Otsu :

```

1 from skimage.filters import threshold_otsu
2 seuil_otsu = threshold_otsu(image)
3 image_binarisee = image > seuil_otsu
4 plt.imshow(image_binarisee, cmap='gray')

```

4. Seuillage multi-niveaux :

```

1 seuil1 = 85
2 seuil2 = 170
3 mask1 = (image <= seuil1)
4 mask2 = (image > seuil1) & (image <= seuil2)
5 mask3 = (image > seuil2)
6 image_segmentee = mask1 + 2*mask2 + 3*mask3
7 plt.imshow(image_segmentee, cmap='nipy_spectral')

```

5. Seuillage par hystérésis :

```

1 from skimage import feature
2 edges = feature.canny(image, low_threshold=30, high_threshold
    =100)
3 plt.imshow(edges, cmap='gray')

```

Questions :

1. Comment choisir le seuil approprié pour le seuillage global ?
2. Pourquoi le seuillage adaptatif est-il préférable pour les images avec des variations d'éclairage ?
3. Comment le seuillage d'Otsu détermine-t-il le seuil optimal ?
4. Dans quels scénarios le seuillage multi-niveaux serait-il utile ?
5. Expliquez comment le seuillage par hystérésis fonctionne pour éliminer les fausses détections de bords.

3 Segmentation par croissance de région (Region Growing)

Objectif :

- Comprendre la méthode de segmentation par croissance de région.
- Implémenter et tester cette technique sur diverses images.

Théorie :

La croissance de région est une méthode basée sur le pixel. Elle commence à partir d'un point initial (ou plusieurs points) et essaie d'inclure les pixels voisins sur la base de certains critères de similarité, tels que l'intensité, la couleur ou la texture. Cette méthode continue jusqu'à ce que plus aucun pixel ne puisse être ajouté à la région.

Étapes de la croissance de région :

1. Sélectionner un point initial.
2. Définir un critère de similarité.
3. Ajouter les pixels voisins qui satisfont le critère à la région.
4. Répéter l'étape 3 jusqu'à ce que plus aucun pixel ne puisse être ajouté.

Pratique :

Implémentation de base de la croissance de région :

```
1 def region_growing(image, seed):
2     height, width = image.shape
3     visited = np.zeros_like(image, dtype=np.bool)
4     segmented = np.zeros_like(image, dtype=np.bool)
5     list_pixels = [seed]
6     intensity_threshold = 20 # Ajuster selon l'image
7
8     while len(list_pixels) > 0:
9         current_pixel = list_pixels.pop()
10        x, y = current_pixel
11
12        if visited[y, x]:
13            continue
14
15        visited[y, x] = True
16        segmented[y, x] = True
17
18        for dx in [-1, 0, 1]:
19            for dy in [-1, 0, 1]:
20                nx, ny = x + dx, y + dy
21                if nx >= 0 and nx < width and ny >= 0 and ny
22                < height:
23                    difference = abs(int(image[y, x]) - int(
24                    image[ny, nx]))
25                    if not visited[ny, nx] and difference <
26                    intensity_threshold:
27                        list_pixels.append((nx, ny))
28
29    return segmented
```

Application sur une image :

```
1 seed_point = (50, 50) # Choisir en fonction de l'image
2 segmented_image = region_growing(grayscale_image, seed_point)
3 plt.imshow(segmented_image, cmap='gray')
```

Questions :

1. Comment le choix du point initial (germe) influence-t-il le résultat final ?
2. Quels sont les avantages et inconvénients de la segmentation par croissance de région par rapport à d'autres méthodes de segmentation ?
3. Comment pouvez-vous améliorer la méthode pour qu'elle soit plus robuste face aux variations d'intensité dans une région ?
4. Dans quels types d'images ou situations cette méthode est-elle particulièrement utile ?

4 Segmentation par Clustering

Objectif :

- Comprendre la segmentation d'images via la méthode de clustering.
- Implémenter et tester le clustering, en particulier l'algorithme K-means, pour segmenter une image.

Théorie :

Le clustering, ou regroupement, est une technique d'apprentissage non supervisée utilisée pour diviser un ensemble de données en plusieurs groupes homogènes. En matière de traitement d'images, le clustering peut être utilisé pour segmenter une image en plusieurs régions sur la base de la similarité des pixels. Chaque cluster formé représente alors une région distincte de l'image.

La technique de clustering la plus couramment utilisée pour la segmentation d'images est le k-means. Le principe du k-means est simple : on initialise k centres (avec k représentant le nombre de clusters souhaité), puis on affecte chaque point (pixel) au centre le plus proche. Ensuite, on recalcule le centre de chaque cluster comme étant la moyenne de tous les points qui lui sont associés. Ce processus est répété jusqu'à convergence.

Pratique :

Pour segmenter une image en utilisant le k-means, nous pouvons utiliser la bibliothèque scikit-learn.

Segmentation d'une image avec K-means :

```
1 from sklearn.cluster import KMeans
2
3 # Redimensionnement de l'image en une liste de pixels
4 image_resaped = image.reshape((-1, 3)) # pour une image
   couleur
5 k = 3 # Nombre de clusters, ajuster selon les besoins
6 kmeans = KMeans(n_clusters=k)
7 kmeans.fit(image_resaped)
8 clustered = kmeans.cluster_centers_[kmeans.labels_]
9 segmented_image = clustered.reshape(image.shape)
10 plt.imshow(segmented_image.astype(np.uint8))
```

Questions :

1. Comment déterminer la valeur optimale de k (nombre de clusters) pour une image donnée ?
2. Quels sont les avantages et inconvénients de la segmentation par clustering par rapport aux autres méthodes de segmentation ?
3. Comment le clustering basé sur la couleur peut-il être combiné avec d'autres caractéristiques, comme la texture ou la position spatiale, pour obtenir une meilleure segmentation ?
4. Le k-means est sensible à l'initialisation des centres. Comment cela peut-il affecter le résultat de la segmentation et comment peut-on remédier à ce problème ?

5 Segmentation basée sur la couleur

Objectif :

- Comprendre comment la couleur peut être utilisée pour segmenter une image.
- Explorer différentes représentations de couleur et leurs impacts sur la segmentation.

Théorie :

La couleur est une caractéristique importante des objets dans une image. La segmentation basée sur la couleur utilise les valeurs de couleur des pixels pour les classer en différents segments ou régions. Il est important de noter que le choix du modèle de couleur (par exemple RGB, HSV, Lab, etc.) peut influencer la qualité de la segmentation.

Modèles de couleur courants :

- RGB (Rouge, Vert, Bleu) : modèle basé sur les couleurs primaires de la lumière.
- HSV (Teinte, Saturation, Valeur) : sépare l'information de couleur (teinte) de l'intensité/luminosité.
- Lab : espace couleur perceptuellement uniforme, séparant les axes de luminance et de chrominance.

Pratique :

Segmentation d'une image en utilisant le modèle de couleur HSV :

```
1 import cv2
2 import numpy as np
3
4 # Fonction pour segmenter une image basée sur la couleur HSV
5 def segment_by_color_hsv(image, lower_bound, upper_bound):
6     # Convertir l'image de BGR à HSV
7     hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
8
9     # Appliquer un masque basé sur les limites de couleur
10    # spécifiées
11    mask = cv2.inRange(hsv_image, lower_bound, upper_bound)
12
13    # Utiliser le masque pour extraire la région d'intérêt
14    # de l'image originale
15    segmented_image = cv2.bitwise_and(image, image, mask=mask)
16
17    return segmented_image
18
19 # Définir les limites de couleur pour la teinte verte
20 lower = np.array([30, 50, 50])
21 upper = np.array([60, 255, 255])
22
23 # Segmenter l'image originale en utilisant les limites
24 # définies
```

```

22 segmented_image = segment_by_color_hsv(original_image, lower,
    upper)
23
24 # Afficher l'image segment e
25 plt.imshow(segmented_image)

```

Questions :

1. Comment le choix du modèle de couleur influence-t-il la segmentation ?
2. Quels sont les avantages de l'utilisation du modèle HSV pour la segmentation basée sur la couleur ?
3. Comment déterminer les limites inférieure et supérieure pour la segmentation d'une couleur spécifique ?
4. Comment la segmentation basée sur la couleur peut-elle être améliorée en utilisant d'autres caractéristiques, comme la texture ?

6 Exploitation de Datasets Médicaux

Afin de mettre en pratique et de comprendre en profondeur les techniques de segmentation abordées dans ce TP, nous vous encourageons à travailler avec des images médicales issues de bases de données publiques. Vous pourrez télécharger ces images, les intégrer dans Google Colab, et y appliquer les méthodes de segmentation.

6.1 Datasets recommandés et leur pertinence

Segmentation basée sur l'intensité

Suggestion : Utilisez des images de mammographie de <https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=22516629DDSM>. Ces images montrent des structures ayant des variations d'intensité, idéales pour cette technique.

Segmentation par croissance de région

Suggestion : Essayez des images cérébrales de <https://brainweb.bic.mni.mcgill.ca/brainweb/>. Ce dataset est excellent pour segmenter diverses structures du cerveau.

Segmentation Clustering

Suggestion : Exploitez les images de tumeurs cérébrales du <https://wiki.cancerimagingarchive.net/display/NBIA/Downloading+TCIA+ImagesTCIA>. Ces images sont adaptées au clustering en raison de leurs variations de densité.

Segmentation basée sur la couleur

Suggestion : Utilisez des images rétinienne de <https://www5.cs.fau.de/research/data/fun-images/RIGA>. Ces images contiennent des structures colorées distinctes, ce qui en fait une excellente candidate pour cette méthode.

6.2 Rendu et Collaboration

Une fois vos expériences terminées, veuillez préparer un compte-rendu détaillant les méthodes utilisées, les challenges rencontrés et les résultats obtenus. Pour faciliter la collaboration et l'évaluation, partagez également votre code et vos observations sur Google Colab.

7 Comparatif des Méthodes de Segmentation

Pour mieux saisir la portée et les nuances de chaque technique de segmentation, il est essentiel d'effectuer une comparaison systématique sur les différents datasets.

7.1 Objectifs

- Mesurer l'efficacité de chaque technique sur des types d'images variés.
- Discerner les avantages et limites propres à chaque méthode.
- Souligner l'importance de choisir la technique la plus pertinente selon le type de données.

7.2 Protocole

1. Implémentez chaque technique sur l'ensemble des datasets suggérés.
2. Évaluez qualitativement la segmentation : précision des contours, détails, gestion du bruit, etc.
3. Utilisez des métriques telles que le coefficient de Jaccard ou l'indice de Dice pour une évaluation quantitative.
4. Consignez vos observations et conclusions.

7.3 Rapport Final

Préparez un rapport synthétisant vos observations. Mettez en avant les forces et faiblesses de chaque méthode, leurs pertinences selon les types d'images, et les particularités des structures segmentées.

Rappel

La segmentation, souvent initiale dans le traitement d'images, influe sur les étapes ultérieures (mesure, classification, détection d'anomalies). Assurez-vous donc de choisir judicieusement la méthode adaptée à chaque situation.

8 Conclusion

La segmentation d'images joue un rôle fondamental dans de nombreux domaines d'application, allant de l'imagerie médicale à l'agriculture intelligente. Au cours de ce TP, nous avons exploré diverses techniques et méthodes pour segmenter des images, soulignant l'importance de choisir la méthode la plus adaptée en fonction des spécificités et des besoins de l'image en question.

Nous avons débuté avec des techniques de seuillage, avant de progresser vers des méthodes plus avancées telles que la croissance de région et le clustering. La segmentation basée sur la couleur nous a également permis d'appréhender l'importance des caractéristiques chromatiques dans la distinction des objets ou des régions d'intérêt.

Il est essentiel de comprendre que, si chaque méthode a ses propres avantages, elle peut également présenter des limites dans certaines situations. C'est pourquoi une approche combinée ou l'adaptation de certaines techniques peut parfois s'avérer nécessaire pour obtenir des résultats optimaux.

Enfin, nous espérons que ce TP aura non seulement renforcé vos compétences techniques, mais qu'il vous aura également inspiré à approfondir davantage les vastes possibilités offertes par la segmentation d'images dans des contextes réels et variés.