

AIIP

(mai.nguyen-verger@cyu.fr)

Partie 7. Segmentation des images

La segmentation d'une image consiste à regrouper les pixels de l'image qui partagent les mêmes propriétés. Cette opération permet de classer les pixels d'une image en sous groupes selon des critères prédéfinis (classification). Lorsque l'on souhaite séparer les pixels d'une image en deux sous-groupes (deux classes), on parle de binarisation de l'image.

I) Segmentation d'images par seuillage d'histogramme (Binarisation)

Une technique simple de binarisation de l'image consiste à évaluer visuellement un seuil grâce l'observation de l'histogramme de l'image. En effet, l'histogramme nous permet d'évaluer la répartition des intensités des pixels de notre image.

Lorsque les pixels de notre image sont regroupés en deux sous-groupes identifiables à l'œil nu, la binarisation est évidente mais cette opération peut se compliquer avec une image réelle.

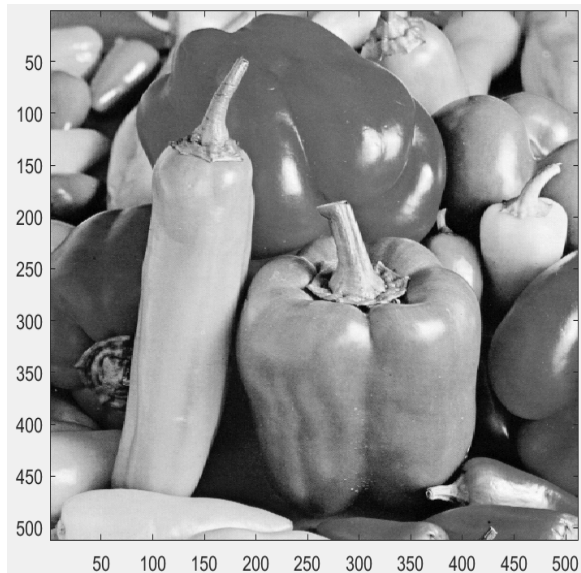


Fig. 1 Image originale

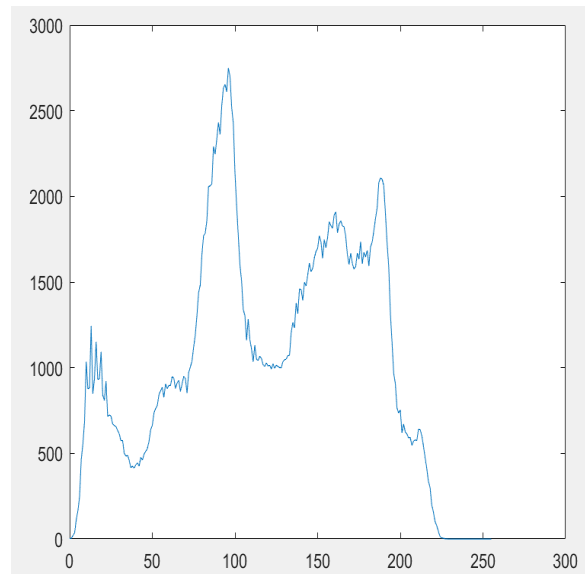


Fig. 2. Histogramme de l'image originale

Objectif : étude d'une méthode simple *d'évaluation de seuil* pour la binarisation d'une image à partir du seuil choisi de manière aléatoire.

Critère : équi-répartition du nombre de pixels par classe.

Algorithme :

1. Choisir un seuil S de manière aléatoire (compris entre 0 et 255). Ainsi, les intensités inférieures à S appartiennent au groupe 1 et les intensités supérieures appartiennent au groupe 2.
2. Calculer la moyenne de chacun des deux groupes de pixels S_1 et S_2 grâce à la répartition des pixels donnée par l'histogramme de l'image.
3. Calculer $S = \frac{S_1 + S_2}{2}$.
4. Répéter les étapes 2 et 3 jusqu'à ce que S soit constant.

Travail proposé :

1. Calculer l'histogramme de l'image « pepper.bmp » proposée.
2. Implémenter l'algorithme de seuillage proposé et vérifier la convergence de celui-ci.
3. Seuiller l'image de deux manières différentes :

- Proposition 1 :

$$Pixel(i, j) = \begin{cases} 0, & \text{si } Pixel(i, j) < S \\ 255, & \text{sinon.} \end{cases}$$

- Proposition 2 :

$$Pixel(i, j) = \begin{cases} S_1, & \text{si } Pixel(i, j) < S \\ S_2, & \text{sinon.} \end{cases}$$

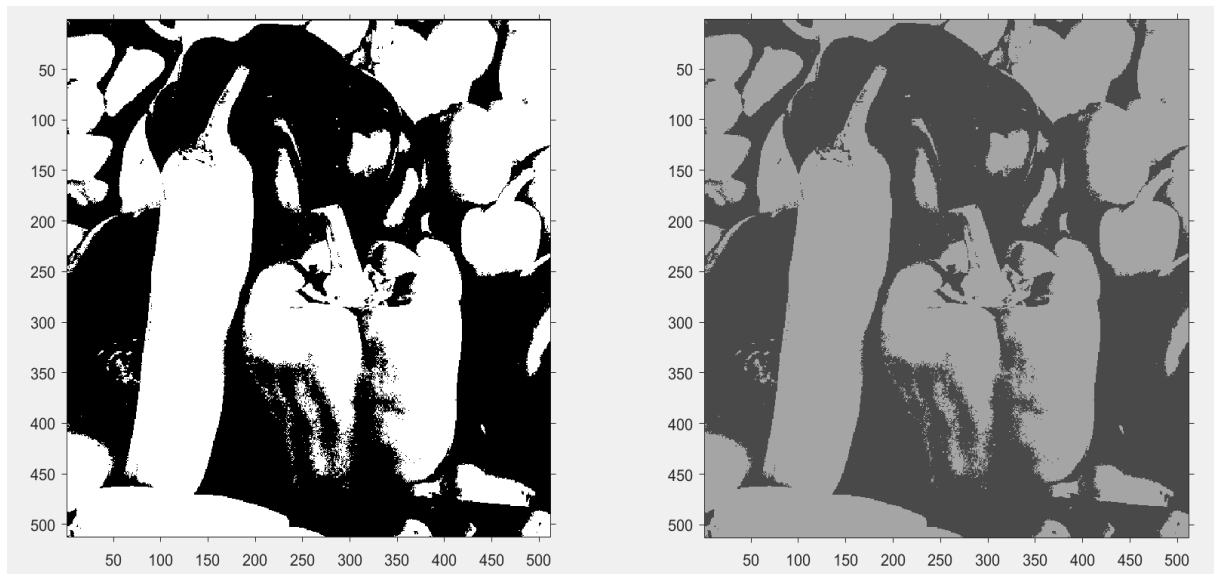


Figure 3 : Résultats obtenus avec la proposition 1 (à gauche) et la proposition 2 (à droite)

II) Segmentation d'images grâce à l'algorithme des K-means

Contexte :

La segmentation est une opération qui permet de classer les pixels d'une image en sous groupes. Une des méthodes de segmentation d'images est l'algorithme des K-means.

Etant donné un nombre de pixels n , on cherche à répartir ces n pixels en K classes C_1, C_2, \dots, C_K en minimisant la distance entre les points internes à chaque classe. Ensuite, après convergence de l'algorithme, les pixels se situant dans le même sous-groupe vont prendre la valeur de la moyenne de ces points.

Objectif : Ce projet propose de segmenter une image grâce à l'algorithme des k-means.

Algorithme :

1. Choisir le nombre de classes (K classes par exemple).
2. Choisir de manière aléatoire les moyennes des K classes et les ranger dans un vecteur par ordre croissant.
3. Affecter pour chaque intensité i allant de 0 à 255 de l'histogramme, la classe ayant la moyenne la plus proche de cette intensité i :

$$c(i) = \min_{k \in [1, K]} (|i - m_k|)$$

avec m_k la moyenne de la $k^{\text{ème}}$ classe

4. Calculer la moyenne de chaque nouvelle classe.
5. Répéter les étapes 3 et 4 jusqu'à convergence de l'algorithme.

Travail proposé :

4. Calculer l'histogramme de l'image « pepper.bmp » proposée.
5. Implémenter l'algorithme de k-means proposé et vérifier la convergence de celui-ci.
6. Segmenter l'image en affectant à chaque pixel, la moyenne de la classe auquel il appartient.
7. Tester l'image pour des nombres de classe différents.

Résultats attendus :

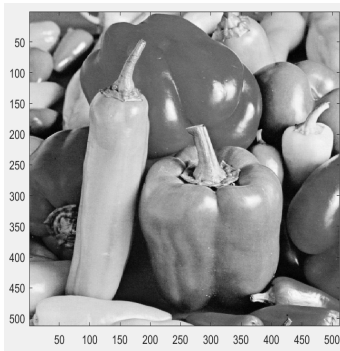


Figure1 : Image originale

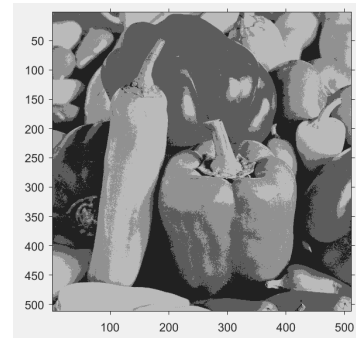


Figure 3 : Segmentation pour K = 4

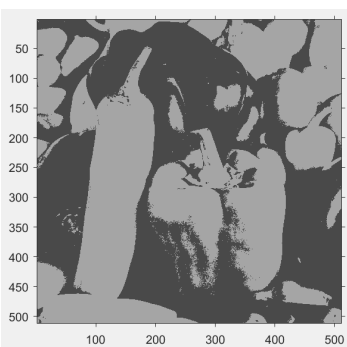


Figure 2 : Segmentation pour K = 2

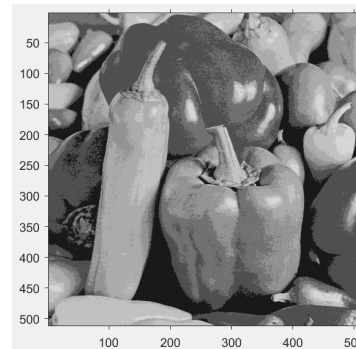


Figure 4 : Segmentation pour K = 8

