

# MINI PROJET TRAITEMENT D'IMAGES

Elaboré par : Aymen Kooli & Maha Ben Arbia & Manel  
Reghima RT4/1

## A/Objectifs :

Le Travail à faire consiste à choisir deux approches de segmentation parmi les quatre approches vues en cours, et à les évaluer sur les 5 bases d'images fournies en utilisant deux critères d'évaluations pour enfin déduire quelle méthode est meilleure selon type d'image, le critère de performance, et le but derrière la segmentation.

## B/ Présentation des deux méthodes de segmentation choisies :

### 1/Segmentation par seuillage :

⇒ Methode de OTSU :

OTSU est un algorithme de segmentation non supervisé et adaptatif sans paramètres, qui est largement utilisé.

L'algorithme recherche de manière exhaustive le seuil qui minimise la variance intra-classe, définie comme la somme pondérée des variances des deux classes :

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t)$$

Les poids  $\omega_0$  et  $\omega_1$  sont les probabilités que les deux classes soient séparées par un seuil  $t$ , et  $\sigma_0^2$  et  $\sigma_1^2$  sont les variances de ces deux classes.

La méthode Otsu consiste à trouver la valeur qui minimise la quantité  $\sigma_w^2(t)$

C'est-à-dire trouver  $T = \operatorname{argmin}(\omega_0 \sigma_0^2(t) + \omega_1 \sigma_1^2(t))$ .

### 2/Segmentation par classification :

### ⇒ Algorithme des K-moyennes :

L'algorithme de clustering K -means est un algorithme non supervisé et il est utilisé pour segmenter la zone d'intérêt de l'arrière-plan. En effet, il affecte chaque pixel de l'image à l'un des K groupes ( clusters ). C'est l'une des méthodes les plus utilisées vu sa simplicité et sa rapidité d'implémentation.

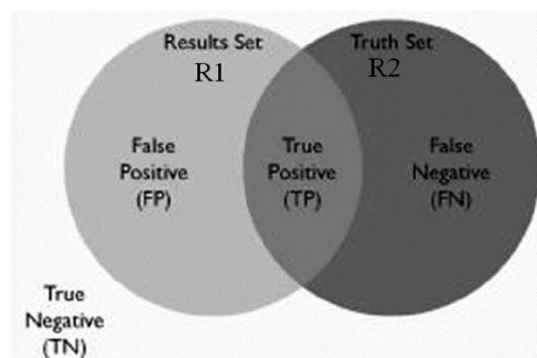
Voilà ses étapes :

- 1-Choisir le nombre de clusters K.
- 2-Sélectionner au hasard K points, les centroïdes.
- 3-Assigner chaque point de données au centroïde le plus proche → qui forme K clusters.
- 4-Calculer et placer le nouveau centroïde de chaque cluster.
- 5-Réaffecter chaque point de données au nouveau centroïde le plus proche. Si une réaffectation a eu lieu, passez à l'étape 4, sinon, le modèle est prêt.

### C/ Choix des critères d'évaluation de performance :

Avant tout on définit la matrice de confusion comme suit :

		Classe réelle	
		-	+
Classe prédite	-	<b>True Negatives</b> <i>(vrais négatifs)</i>	<b>False Negatives</b> <i>(faux négatifs)</i>
	+	<b>False Positives</b> <i>(faux positifs)</i>	<b>True Positives</b> <i>(vrais positifs)</i>



1/ L'**exactitude (accuracy)** : elle indique le pourcentage de bonnes prédictions. C'est un très bon indicateur parce qu'il est très simple à comprendre.

Elle est calculée comme suit :

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2/ La **precision** : c'est la proportion de prédictions correctes parmi les points que l'on a prédits positifs. C'est la capacité de notre modèle à ne déclencher d'alarme que pour un vrai incendie.

## D/ Presentation des résultats :

### 1-Dataset 1objet :

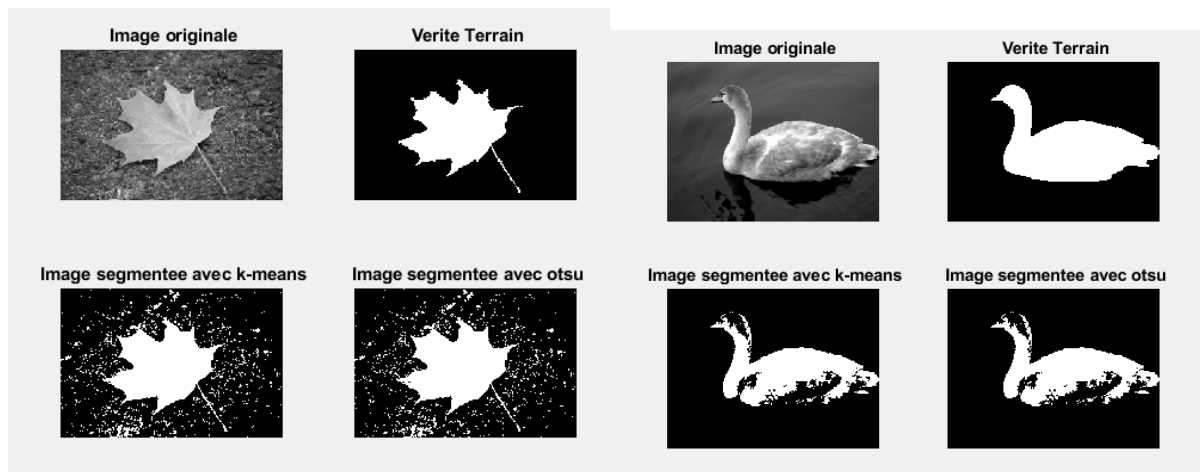


Cette dataset est composé d'images en composantes de gris contenant des objets centrés aux milieux.

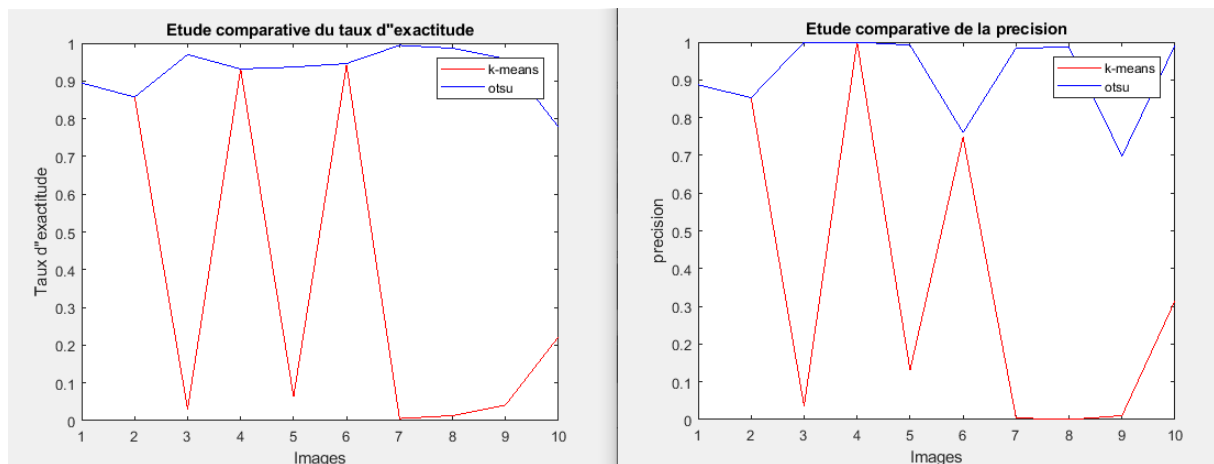
Critère : niveaux de gris.

Méthodes à essayer : k-means et otsu.

Et voilà quelques résultats :



Performances pour k-means on prend ( $K=2$ ), otsu est sans paramètres :



Moyenne des metriques :

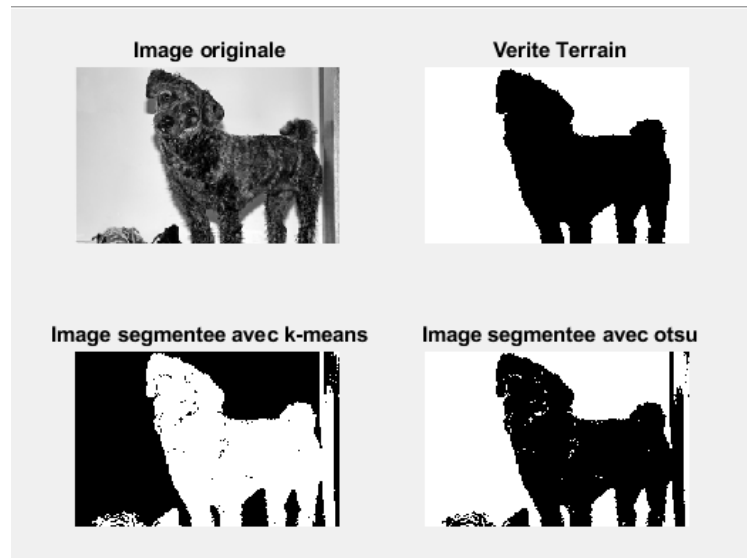
```
>> disp(mean(accuracy_rate(:,1)));
0.4948

>> disp(mean(accuracy_rate(:,2)));
0.9254

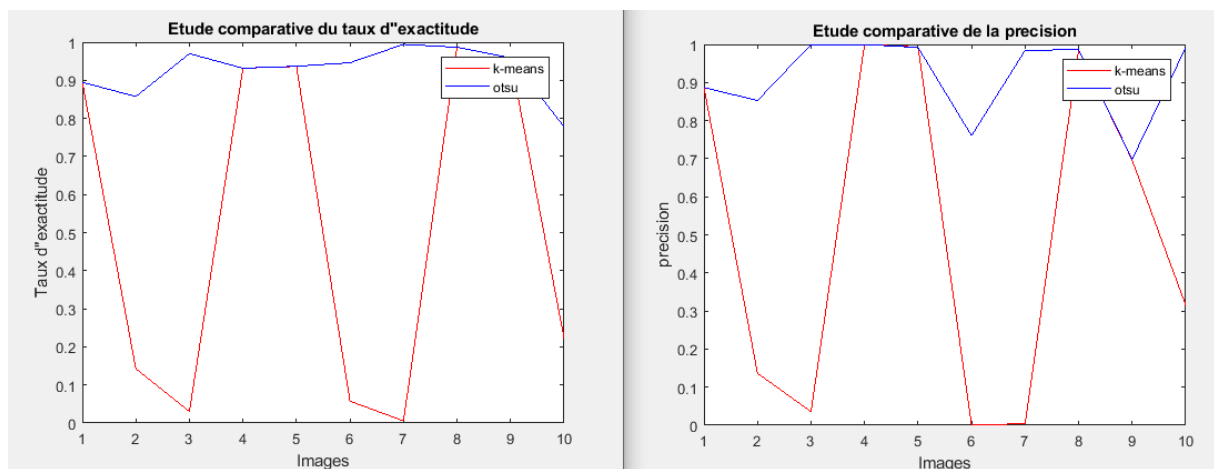
>> disp(mean(prec_rate(:,1)));
0.4937

>> disp(mean(prec_rate(:,2)));
0.9153
```

Pour les images ou l'exactitude pour k-means est presque nulle c'est pas un problème parce que les couleurs des clusters est inversée, exemple :



Performances pour k-means on prend ( $K=4$ ), otsu est sans paramètres :



moyenne des metriques :

```
>> disp(mean(accuracy_rate(:,1)));
    0.6471

>> disp(mean(accuracy_rate(:,2)));
    0.9254

>> disp(mean(prec_rate(:,1)));
    0.6611

>> disp(mean(prec_rate(:,2)));
    0.9153
```

- ⇒ K-means s'améliore lorsque on augmente K.
- ⇒ Otsu donne de meilleurs résultats avec une exactitude proche de 1.

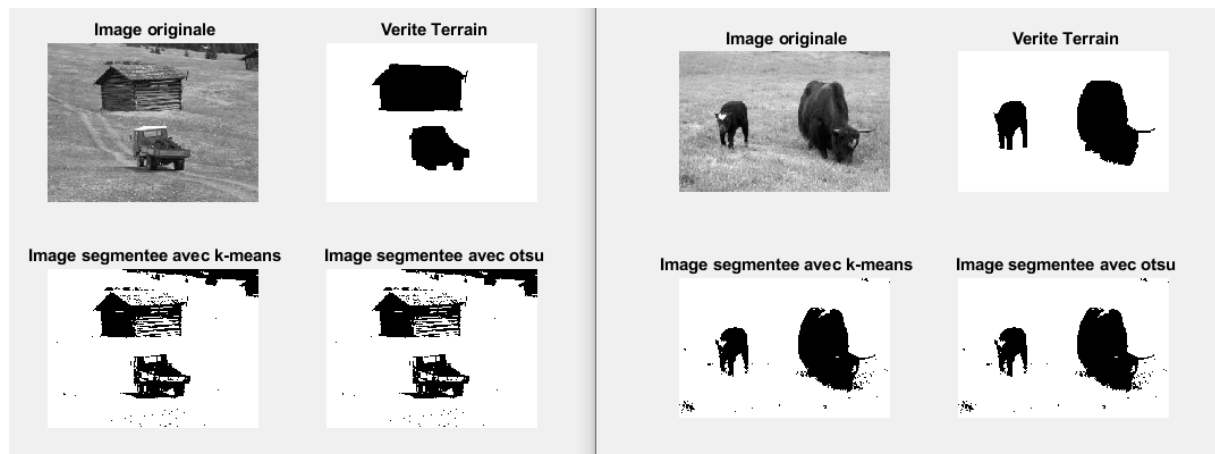
## 2-dataset 2objets :

Dans cette dataset on a des photos en nuances de gris aussi, chaque photo contient 2 objets.

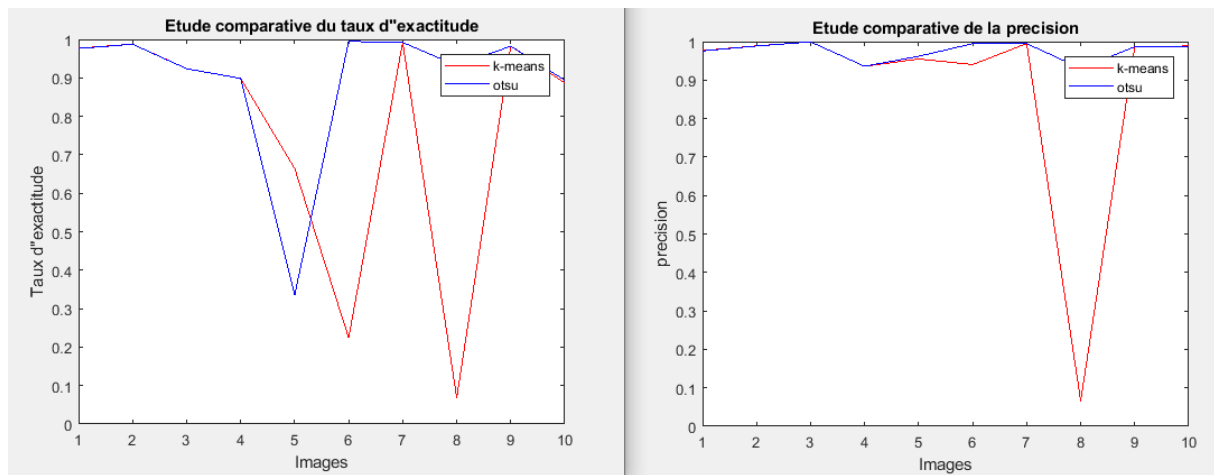
Critères : niveaux de gris.

Méthodes choisies : otsu et k-means.

Pour k-means k=2 et otsu sans paramètres :



Mesures des performances :



Moyenne des metriques :

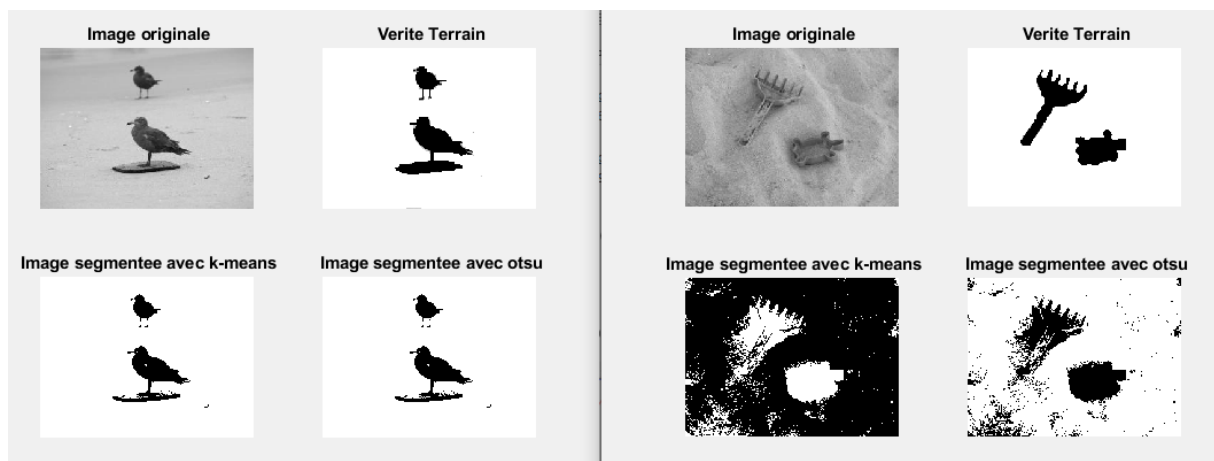
```
>> disp(mean(accuracy_rate(:,1)));
    0.7606

>> disp(mean(accuracy_rate(:,2)));
    0.8917

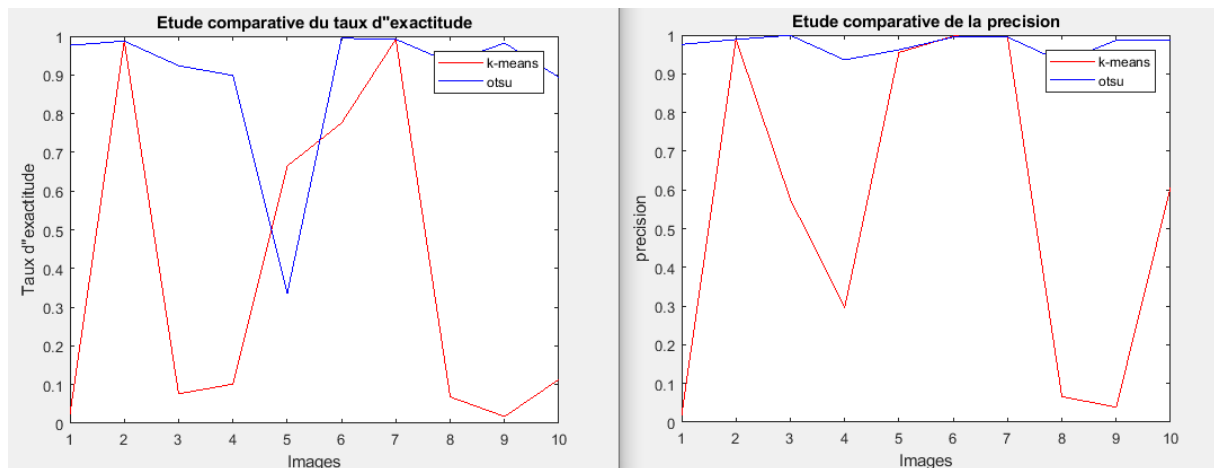
>> disp(mean(prec_rate(:,1)));
    0.8833

>> disp(mean(prec_rate(:,2)));
    0.9757
```

Pour k-means k=4 et otsu sans paramètres :



## Mesures des performances :



## Moyenne des métriques :

```
>> disp(mean(accuracy_rate(:,1)));  
0.3820  
  
>> disp(mean(prec_rate(:,1)));  
0.5537
```

- ⇒ Pour multiples objets k-means donne de meilleurs résultats avec k=2.
- ⇒ Otsu donne de meilleurs résultats.

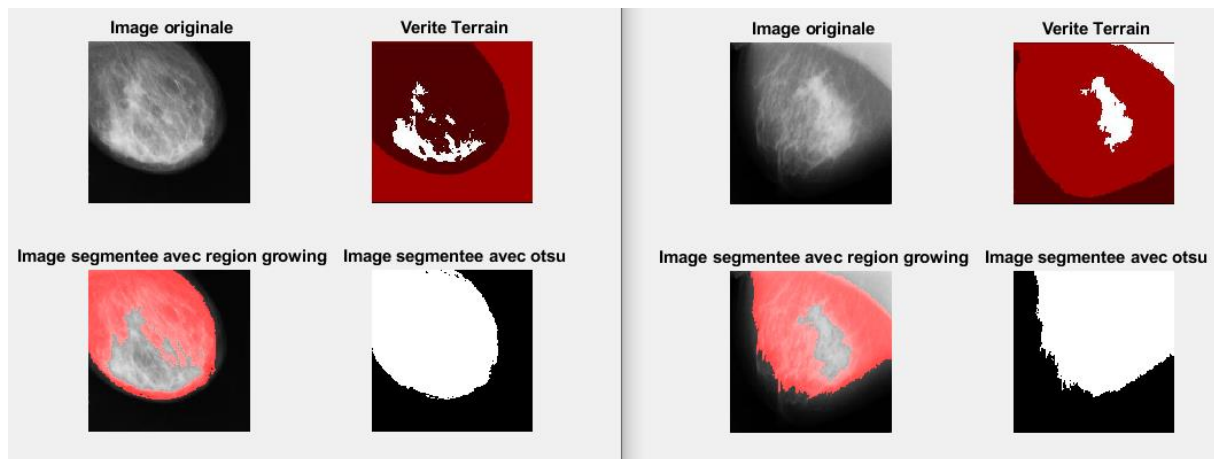
## 3-Dataset Breast-Cancer :

Cette dataset contient des images d'IRM, en niveaux de gris.

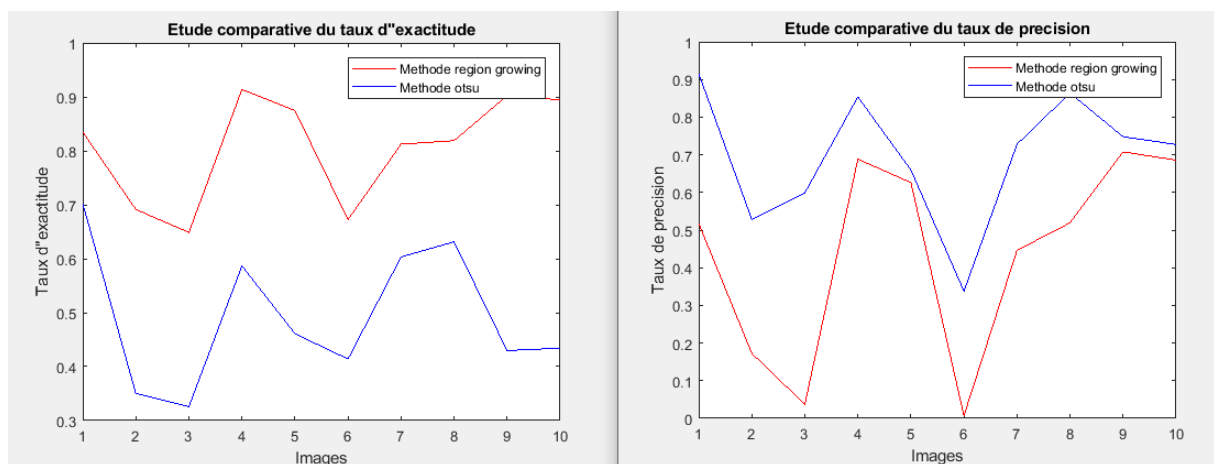
Critère : Difference d'intensité

Méthodes utilisées : Region growing et otsu.





Mesure des performances :



Moyenne des métriques :

```
>> disp(mean(accuracy_rate(:,1)));
0.8063

>> disp(mean(accuracy_rate(:,2)));
0.4938
```

⇒ La methode de région growing donne de meilleurs résultats pour cette dataset et aide à la détection d'anomalies.

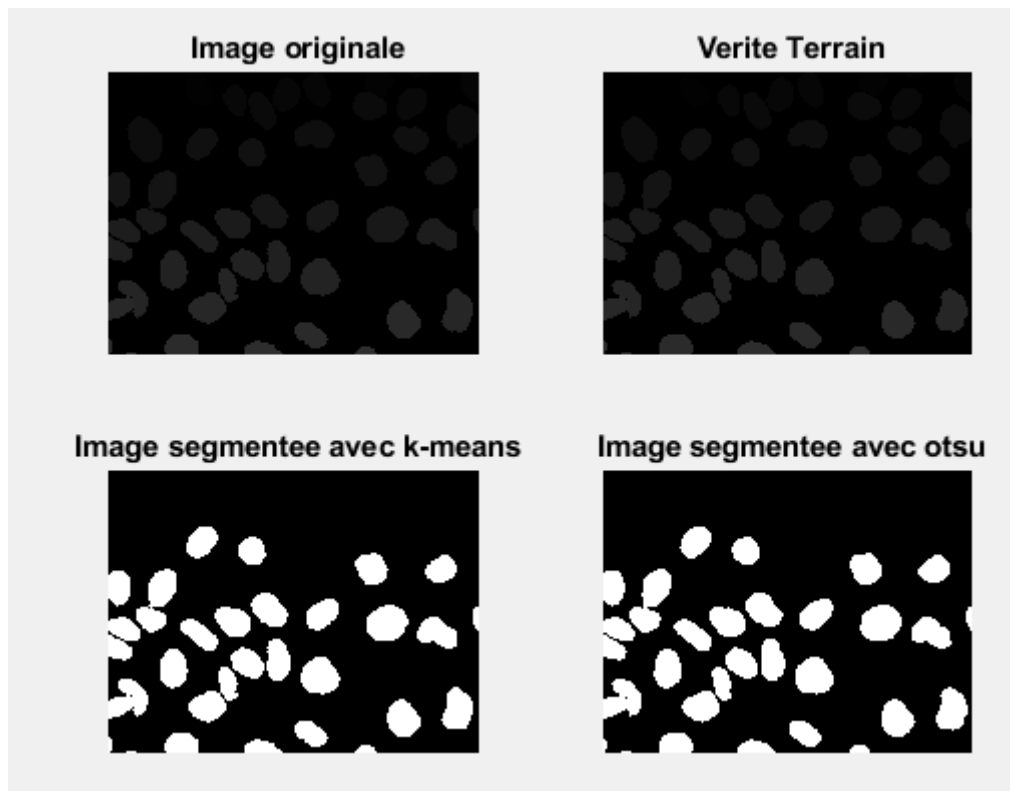
#### 4-Dataset Cellules :

Dataset en niveaux de gris contenant des images de cellules.

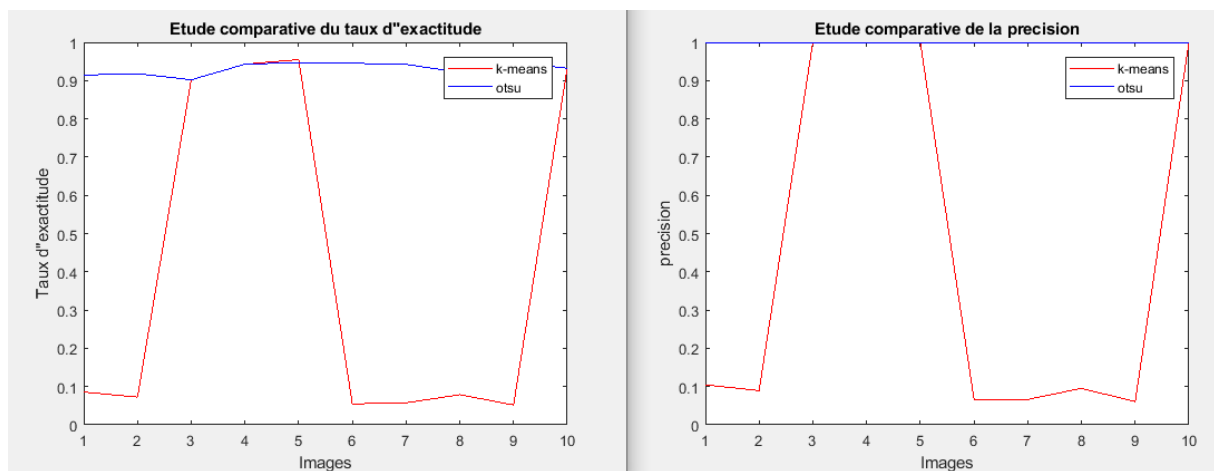
Critère : Contour

Méthodes : K-means et otsu

Pour k-means k=2 et otsu sans paramètres :



Mesures des performances :



Moyenne des métriques :

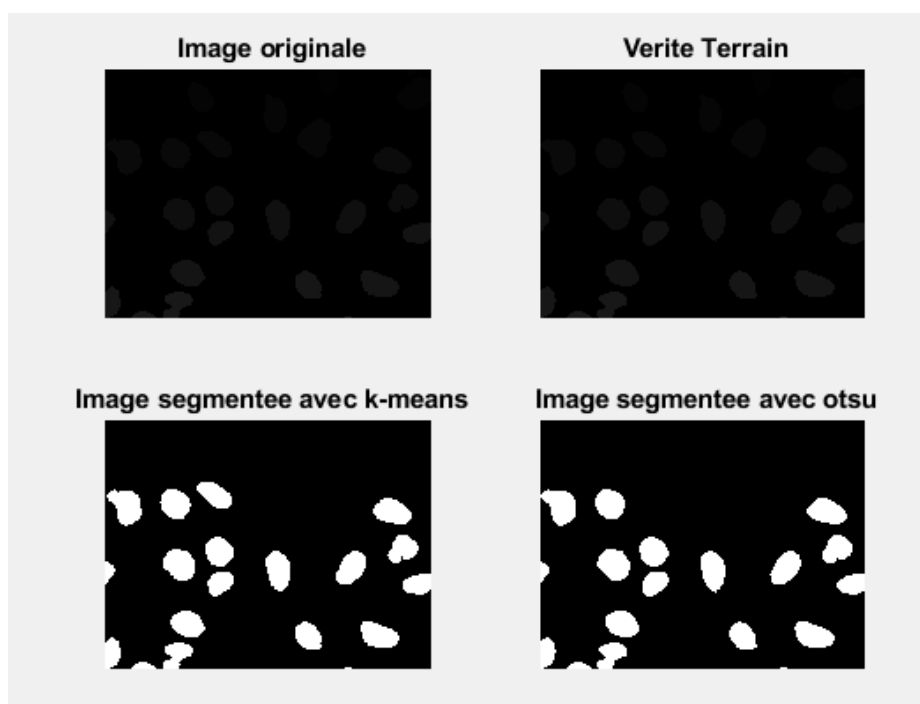
```
>> disp(mean(accuracy_rate(:,1)));
    0.4133

>> disp(mean(accuracy_rate(:,2)));
    0.9314

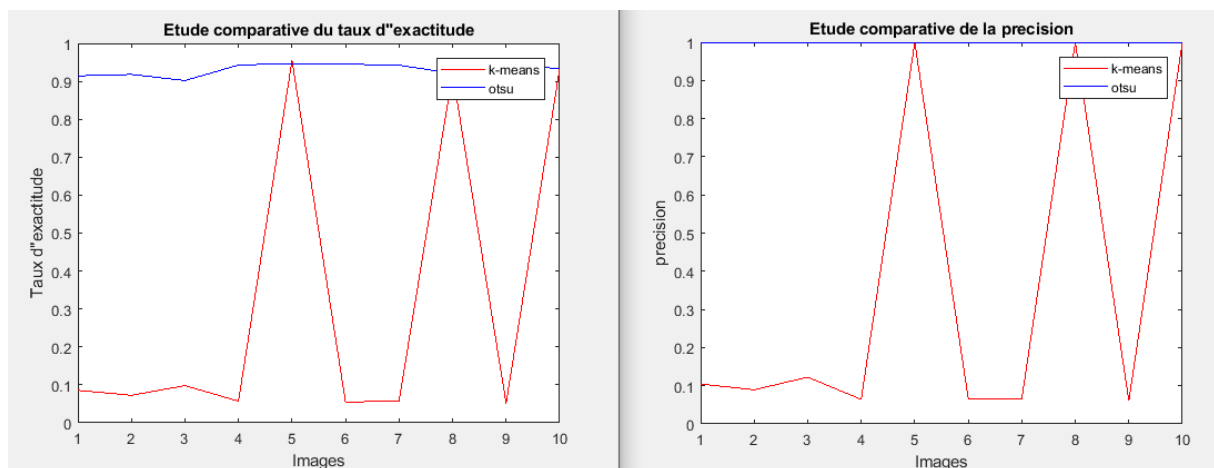
>> disp(mean(prec_rate(:,1)));
    0.4482

>> disp(mean(prec_rate(:,2)));
    1
```

Pour k-means k=3 et otsu sans paramètres :



Mesure des performances :



Moyenne des metriques :

```
>> disp(mean(accuracy_rate(:,1)));
    0.3286

>> disp(mean(prec_rate(:,1)));
    0.3574
```

⇒ OTSU donne les meilleurs résultats.

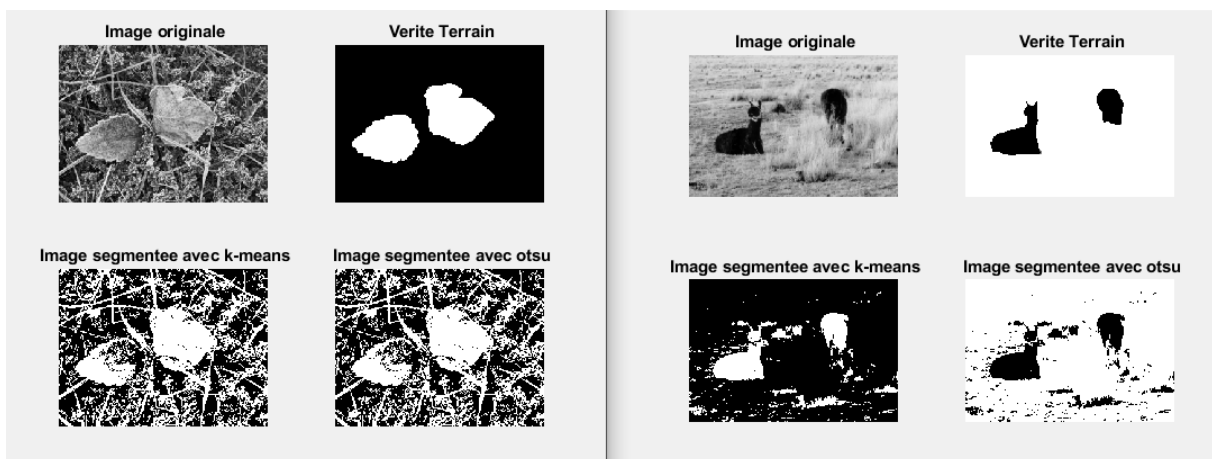
## 5- Dataset textfort :

C'est une dataset contenant des images à forte texture.

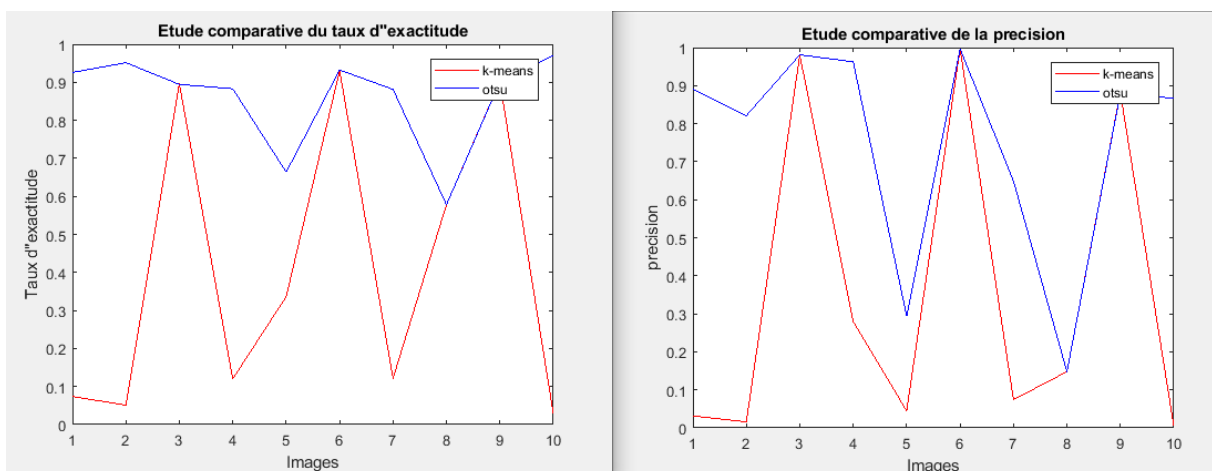
Critère : texture

Méthodes : otsu et k-means.

Pour k-means k=2 et otsu sans paramètres :



Mesure des performances :



Moyenne des metriques :

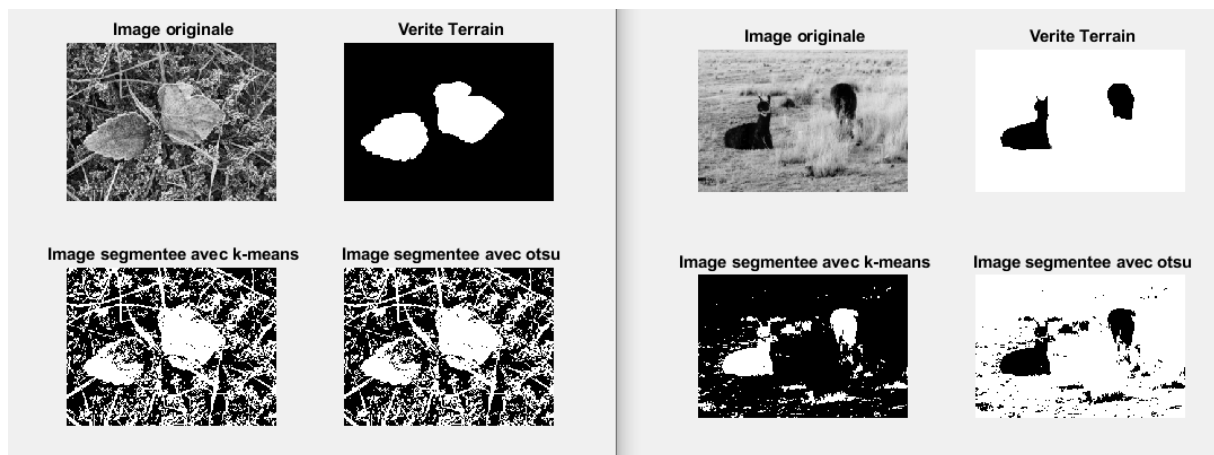
```
>> disp(mean(accuracy_rate(:,1)));
    0.4039

>> disp(mean(accuracy_rate(:,2)));
    0.8582

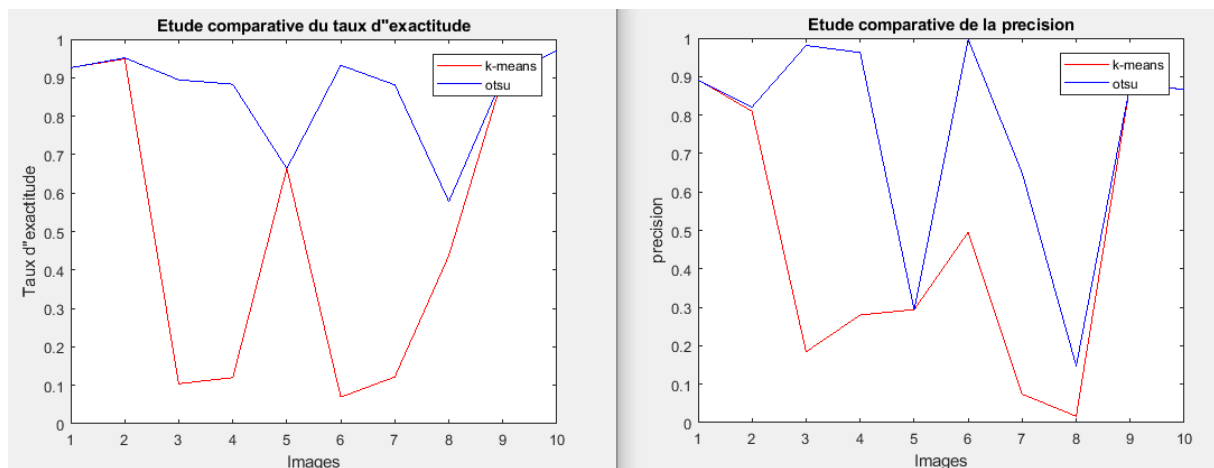
>> disp(mean(prec_rate(:,1)));
    0.3458

>> disp(mean(prec_rate(:,2)));
    0.7489
```

Pour k-means k=10 et otsu sans paramètres :



Mesure des performances :



Moyenne des metriques :

```
>> disp(mean(accuracy_rate(:,1)));
    0.5265

>> disp(mean(prec_rate(:,1)));
    0.4794
```

- ⇒ Pour les fortes textures si  $k$  augmente l'efficacité de la segmentation  $k$ -means augmente aussi
- ⇒ Mais otsu donne toujours de meilleurs résultats.

### E/ Conclusion :

Le choix de la méthode de segmentation n'est pas standard.

En effet, on doit ajuster le type de segmentation avec le type d'images sur lequel on va opérer.

On a vu que pour un même type d'image, un type de segmentation donne des valeurs d'exactitude meilleurs que pour un autre type.