Dimitri Charneux
Rémi Kroll

Rendu tp: Segmentation automatique par analyse d'histogramme

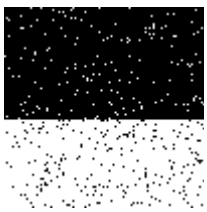
Introduction

Durant ce TP, nous allons voir comment concevoir des macros R permettant de réaliser un seuillage automatique des niveaux de gris d'une image par règle de Bayes.

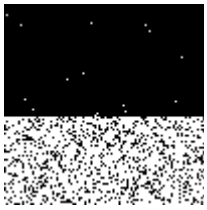
Q1. Seuillage fixe

L'argument seuil correspond au seuil de binarisation de l'image. Cet argument permet de transformer une image en niveaux de gris, en une image en noir et blanc. Ce seuil fonctionne de la manière suivante : les pixels dont le niveau de gris est supérieur au seuil sont mis à 1 et ceux inférieur à 0. Si le seuil est à 0, nous obtiendrons une image toute blanche. Par contre, si il est à 1, l'image obtenu sera noire.

Binarisation de l'image 2classes_100_100_8bits avec le seuil à 0,5



Binarisation de l'image 2classes_100_100_8bits avec le seuil à 0,55



Binarisation de l'image 2classes_100_100_8bits avec le seuil à 0,6

Pour l'image proposée, nous n'arrivons pas à obtenir une binarisation parfaite. Pour le premier seuil, celui de 0,5, nous obtenons trop de points blancs dans le carré noir. Pour le troisième, nous obtenons un constat opposé, il y a beaucoup trop de points noirs dans le carré blanc. Enfin, pour la seconde image, nous obtenons une meilleur binarisation mais elle n'est toujours pas parfaite car il reste du bruit dans l'image : des points blancs et des points noirs dans les carrés noir et blanc. Cela s'explique par le fait qu'il existe des points de niveaux de gris équivalent dans les deux parties de l'image.

Q2. Seuillage automatique (Bayes) – Probabilité a priori des classes

```
nbins <- 256
h1 <- hist (as.vector (omega1), freq=FALSE, breaks = seq (0, 1, 1 / nbins))
p_omega1= sum(h1$counts[0:255])/ sum(h$counts[0:255])

h2 <- hist (as.vector (omega2), freq=FALSE, breaks = seq (0, 1, 1 / nbins))
p_omega2= sum(h2$counts[0:255])/ sum(h$counts[0:255])
```

Avec le code ci-dessus, nous obtenons une probabilité P(omega1) = 0.57 et P(omega2) = 0.43. Ce code calcule l'histogramme des deux images puis, pour calculer la probabilité de la classe h1 (respectivement h2), il divise le nombre de pixels de h1 (respectivement h2) par le nombre de pixels de h. Cette probabilité nous indique le pourcentage de chance, pour un pixel de h, d'être dans l'image h1 (respectivement h2).

2classes_100_100_8bitsomega1

2classes_100_100_8bitsomega2

Q3. Seuillage automatique (Bayes) – Probabilité conditionnelle

```
pi141 = h$counts[142]/sum(h$counts[0:255])
pomega1141 = h1$counts[142]/sum(h$counts[0:255])
pomega2141 = h2$counts[142]/sum(h$counts[0:255])
```

Nombres de pixels caractérisés par le niveau de gris 141 dans l'image :
I = h$counts[142] = 118
omega1 = h1$counts[142] = 51
omega2 = h2$counts[142] = 67

Probabilité pour un pixel caractérisé par le niveau de gris 141 d'être dans une image :

P(141/I) = 0,0118
P(141/omega1) = 0.0051
P(141/omega2) = 0.0067

Q4. Seuillage automatique (Bayes) – probabilité d'erreur

Pour trouver la valeur du seuil de binarisation optimale ainsi que le taux d'erreur de classification, nous utilisons le code ci-dessous :

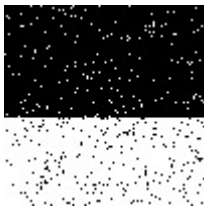
```
#  pour le seuil X calcul de l'erreur d'assignation
somme1 = 0:255
somme2 = 0:255
erreur = 0:255
# recherche du minimum
minimum_erreur = 1;
seuil_minimum_erreur = 0;

for (X in 1:255)
  {
# (\sum_{\mathbf{X} \in \hat{\omega_2}} P(\mathbf{X} / \omega_1). P(\omega_1)
  somme1[X+1]=sum(h1$density[(X+1):256])/sum(h1$density[1:256])
  somme1[X+1]=somme1[X+1]*p_omega1
#\sum_{\mathbf{X} \in \hat{\omega_1}} P(\mathbf{X}  / \omega_2). P(\omega_2)
  somme2[X+1]=sum(h2$density[1:(X+1)])/sum(h2$density[1:256])
  somme2[X+1]=somme2[X+1]*p_omega2

  erreur[X+1] = somme1[X+1] + somme2[X+1]
# seuil corrrespondant à l'erreur minimale
  if (erreur[X+1] < minimum_erreur ) seuil_minimum_erreur = X
  if (erreur[X+1] < minimum_erreur ) minimum_erreur = erreur[X+1]
  }

seuil = seuil_minimum_erreur/255
binaire_Bayes <- (image - seuil) >= 0
display (binaire_Bayes, "image binaire Bayes")
```

Grâce à ce code, nous obtenons un seuil de binarisation de 139, ainsi qu'un taux d'erreur de 0.0481. Nous obtenons ainsi l'image suivante :

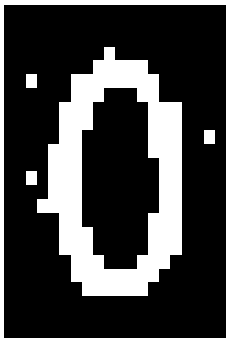


Seuillage automatique de l'image 2classes_100_100_8bits

Q5. Extraction de la région représentant le 0 par seuillage automatique (Bayes)

Afin de réaliser l'extraction de la région représentant le 0 par seuillage automatique. Nous devons modifier le code en remplaçant le nom des images par celui des images représentant le '0'.

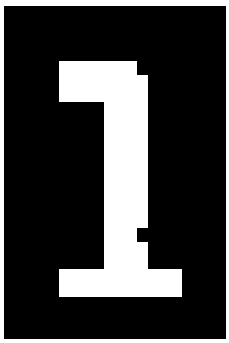
Nous obtenons ainsi un seuil de binarisation de 142 et l'image binarisée suivante :



E*xtraction de région de l'image rdf-chiffre-0-8bits par seuillage automatique*

Dans cette image, on voit clairement le '0' en blanc et le fond en noir, la binarisation est donc convaincante. Cependant, nous constatons encore quelques pixels blanc de bruit sur le fond.

Maintenant, si nous binarison l'image 'rdf-chiffre-1-8bits.png' avec le seuil trouver pour le chiffre zéro, nous obtenons l'image suivantes :



E*xtraction de région de l'image rdf-chiffre-0-8bits par seuillage automatique*

Le résultat de cette binarisation est excellent, il y a très peu de bruit et le chiffre '1' est vraiment reconnaissable.

Conclusion

Ce TP nous a permis de comprendre comment binariser une image en utilisant le seuillage automatique des niveaux de gris par la règle de Bayes. Cette méthode, très efficace, nous donne le seuil de binarisation optimal pour l'image donnée.