

# TD/Mini-projet

## Classification bayésienne

Image 3 : De l'image à l'information

Reconnaissance de formes

Télécom Saint-Étienne

Sacha GRELET-RIOUT et Hugo BRUHIER

### 1 Introduction

Nous allons, lors de ce TD, travailler sur la classification bayésienne et faire l'application des formules et principes vus en cours sur des cas pratiques sur *MatLab*. Pour bien poser certaines notations on a :

$$p(w_i|x) = \frac{p(x|w_i)\mathbb{P}(w_i)}{p(x)} \quad (1)$$

avec :

- $p(w_i|x)$  la probabilité à postériori,
- $p(x|w_i)$  la vraisemblance,
- $\mathbb{P}(w_i)$  la probabilité à priori,
- $p(x)$  la probabilité totale.

On dispose de la formule suivante <sup>1</sup> qui nous permettra de faire des calculs plus tard dans le code :

$$p(x) = p(x|w_1)\mathbb{P}(w_1) + p(x|w_2)\mathbb{P}(w_2) \quad (2)$$

### 2 Exercice 1 : Décision bayésienne

Ici on regarde un problème à deux classes que l'on va modéliser par deux lois gaussiennes, la variance  $\sigma^2$  sera la même et les moyennes respectives seront 0 et 1 :

$$p(x|w_1) = \frac{1}{\sqrt{\pi}} \exp[-x^2] \quad (3)$$

$$p(x|w_2) = \frac{1}{\sqrt{\pi}} \exp[-(x-1)^2] \quad (4)$$

On cherche ici à déterminer la valeur de la variance  $\sigma^2$ , pour cela on rappelle la formule d'une loi normale de paramètres  $\mu$  et  $\sigma$  :

$$\mathcal{N}(\mu, \sigma) = \frac{1}{\sqrt{2\sigma^2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

---

1. Vraie dans le cas d'un problème à deux classes.

On procède à l'identification suivante :

$$\begin{aligned} 2\sigma^2 &= 1 \\ \Leftrightarrow \sigma^2 &= \frac{1}{2} \\ \Leftrightarrow \sigma &= \frac{1}{\sqrt{2}} \end{aligned}$$

Dans un premier temps nous allons prendre l'équiprobabilité comme probabilités à priori, donc nous avons :

$$\begin{cases} \mathbb{P}(w_1) = 0,5 \\ \mathbb{P}(w_2) = 0,5 \end{cases}$$

Traçons sur la figure 2 les densités de probabilité des deux classes, les probabilités à postériori ainsi que le seuil bayésien.



FIGURE 1 – Données de probabilités dans le cas initial

Nous cherchons le seuil de classification, tout d'abord nous allons le trouver en déroulant de manière littérale les calculs puis nous allons coder sa recherche pour vérifier expérimentalement le résultat. On définit le seuil  $x_B$  par :

$$g_1(x_B) = \log [p(x_B|w_1)\mathbb{P}(w_1)] = g_2(x_B) \quad (5)$$

On cherche donc  $x$  tel que l'égalité suivante soit vérifiée :

$$\log [p(x|w_1)\mathbb{P}(w_1)] = \log [p(x|w_2)\mathbb{P}(w_2)]$$

On remplace  $p(x|w_1)$  et  $p(x|w_2)$  par leurs expressions respectives :

$$\log \left[ \frac{1}{\sqrt{\pi}} \exp [-x^2] \mathbb{P}(w_1) \right] = \log \left[ \frac{1}{\sqrt{\pi}} \exp [-(x-1)^2] \mathbb{P}(w_2) \right]$$

On utilise une propriété bien connue du logarithme<sup>2</sup> pour séparer les termes :

$$\log \left[ \frac{1}{\sqrt{\pi}} \right] + \log [\exp [-x^2]] + \log [\mathbb{P}(w_1)] = \log \left[ \frac{1}{\sqrt{\pi}} \right] + \log [\exp [-(x-1)^2]] + \log [\mathbb{P}(w_2)]$$

Avec l'égalité  $\mathbb{P}(w_1) = \mathbb{P}(w_2)$  on peut simplifier aisément notre expression :

$$\log [\exp [-x^2]] = \log [\exp [-(x-1)^2]]$$

Les fonctions log et exp sont réciproques l'une à l'autre, on simplifie donc par :

$$-x^2 = -(x-1)^2$$

On développe le second terme de l'expression :

$$\begin{aligned} -x^2 &= -x^2 + 2x - 1 \\ \Leftrightarrow x &= 0,5 \end{aligned}$$

On trouve donc notre seuil bayésien :

$$x_B = 0,5$$

En utilisant le code ci-dessous on recherche expérimentalement cette valeur du seuil bayésien. On trouve bien en sortie une valeur de seuil de 0,5.

---

```

1 i = 1;
2 g1 = log(pxw1(1) * Pw1);
3 g2 = log(pxw2(1) * Pw2);
4 for k = 2 : length(x)
5     g1m = log(pxw1(k) * Pw1);
6     g2m = log(pxw2(k) * Pw2);
7
8     if (g1 - g2)*(g1m - g2m) < 0
9         xB(i) = x(k);
10        i = i + 1;
11    end
12
13    g1 = log(pxw1(k) * Pw1);
14    g2 = log(pxw2(k) * Pw2);
15
16 end

```

---

2.  $\log(x.y) = \log(x) + \log(y)$

On recherche maintenant à quantifier la probabilité de l'erreur. On rappelle la formule de cette probabilité dans le cas à deux régions :

$$\mathbb{P}(\text{erreur}) = \int_{\mathcal{R}_2} p(x|w_1)\mathbb{P}(w_1)dx + \int_{\mathcal{R}_1} p(x|w_2)\mathbb{P}(w_2)dx \quad (6)$$

En utilisant notre seuil  $x_B$  on peut réécrire l'erreur comme :

$$\mathbb{P}(\text{erreur}) = \int_{x_B}^{+\infty} p(x|w_1)\mathbb{P}(w_1)dx + \int_{-\infty}^{x_B} p(x|w_2)\mathbb{P}(w_2)dx$$

On va calculer les intégrales par la méthode des rectangles via la portion de code suivante :

---

```

1 indices = find(x <= xB);
2 M = length(indices);
3 Perreur = Pas * sum(pxw2(indices) * Pw2);
4 Perreur = Perreur + Pas * sum(pxw1(indices(M) + 1:L) * Pw1);

```

---

On trouve avec la bonne valeur du seuil une probabilité d'erreur de 0,2398. On trace sur la figure la probabilité d'erreur en fonction de la position du seuil  $x_B$ , on voit que pour d'autres valeurs du seuil que celle que l'on a trouvé la probabilité de faire une erreur augmente.

Nous allons maintenant changer les probabilités à priori pour nous placer dans un cas non-équi-probable pour voir les changements que cela va apporter sur notre classifieur. Maintenant on a :

$$\begin{cases} \mathbb{P}(w_1) = 0,9 \\ \mathbb{P}(w_2) = 0,1 \end{cases}$$

La figure suivante montre les mêmes informations que la figure 2 avec les nouvelles probabilités à priori :

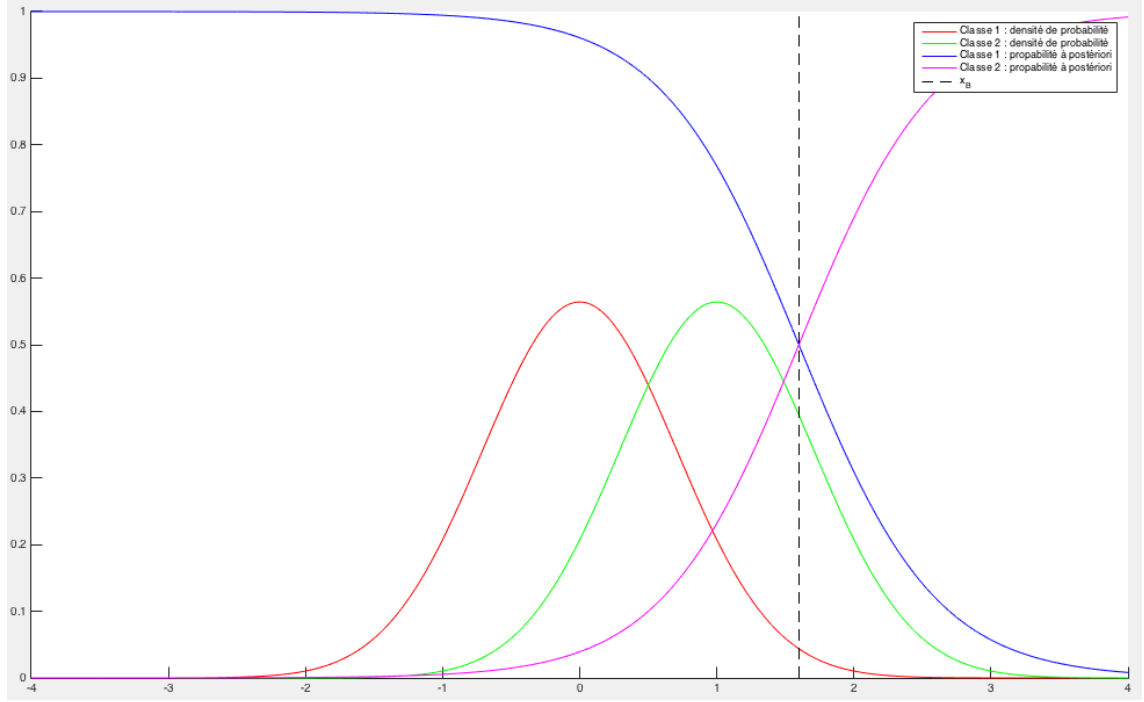


FIGURE 2 – Données de probabilités dans le cas des nouvelles probabilités à priori

On réitère ici le calcul du seuil de décision bayésienne, défini à l'équation 5 page 2, on recherche donc  $x$  tel que :

$$\log [p(x|w_1)\mathbb{P}(w_1)] = \log [p(x|w_2)\mathbb{P}(w_2)]$$

On remplace encore  $p(x|w_1)$  et  $p(x|w_2)$  par leurs expressions respectives :

$$\log \left[ \frac{1}{\sqrt{\pi}} \exp [-x^2] \mathbb{P}(w_1) \right] = \log \left[ \frac{1}{\sqrt{\pi}} \exp [-(x-1)^2] \mathbb{P}(w_2) \right]$$

On sépare les termes par la propriété du logarithme :

$$\log \left[ \frac{1}{\sqrt{\pi}} \right] + \log [\exp [-x^2]] + \log [\mathbb{P}(w_1)] = \log \left[ \frac{1}{\sqrt{\pi}} \right] + \log [\exp [-(x-1)^2]] + \log [\mathbb{P}(w_2)]$$

On simplifie :

$$\log [\exp [-x^2]] + \log [\mathbb{P}(w_1)] = \log [\exp [-(x-1)^2]] + \log [\mathbb{P}(w_2)]$$

On passe toutes les constantes du même côté et on simplifie les exponentielles via les logarithmes :

$$-x^2 = -(x-1)^2 + \log [\mathbb{P}(w_2)] - \log [\mathbb{P}(w_1)]$$

On développe l'expression pour supprimer les termes au carré :

$$-x^2 = -x^2 + 2x - 1 + \log \left[ \frac{\mathbb{P}(w_2)}{\mathbb{P}(w_1)} \right]$$

On modifie un peu l'expression en remplaçant  $\mathbb{P}(w_1)$  et  $\mathbb{P}(w_1)$  par leurs valeurs :

$$2x = 1 - \log \left( \frac{1}{9} \right)$$

Ce qui nous donne :

$$x_B \simeq 1,5986$$

Le calcul dans le programme nous donne pour valeur expérimentale du seuil 1,6 ce qui est cohérent<sup>3</sup>.

On va ne pas prendre en compte le dénominateur dans le calcul des probabilités à postériori, cela nous donne la figure 2. On constate que le seuil de classification bayésienne  $x_B$  n'est pas modifié que l'on prenne en compte ou non la probabilité totale au dénominateur car cette quantité n'est pas discriminante dans la classification.

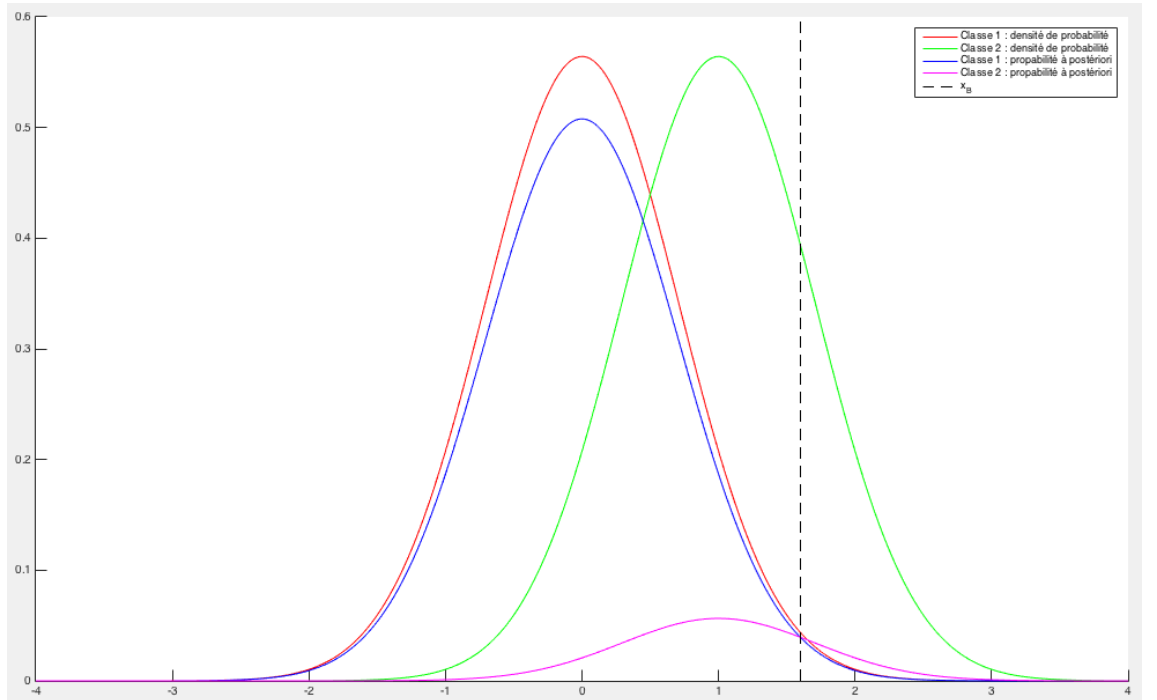


FIGURE 3 – Données de probabilités sans le dénominateur

3. De plus nous utilisons un pas entre chaque valeur du vecteur  $x$  de 0,01 pour diminuer le temps de calcul et l'espace mémoire nécessaire.

## 3 Exercice 2 : Application à la détection de visages automatique

### 3.1 Partie 1 : Apprentissage du modèle de peau

#### 3.1.1 Sélection des échantillons d'apprentissage

On considère que le modèle de peau peut être modélisé par un modèle gaussien multi-varié. On pose  $\mathbf{x}$  le vecteur contenant les données de couleurs d'un pixel, si on pose  $\omega_1$  le classe "peau" on écrit :

$$p(\mathbf{x}|\omega_1) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_1|^{\frac{1}{2}}} \exp \left[ -\frac{1}{2}(\mathbf{x} - \mu_1)^T \Sigma_1^{-1}(\mathbf{x} - \mu_1) \right] \quad (7)$$

On va chercher à trouver les paramètres  $(\mu_1, \Sigma_1)$  inconnus par une phase d'apprentissage. On veut tout d'abord prendre des échantillons d'images représentant de la peau, pour cela on a sur le centre de toutes nos images le visage de George W. Bush que l'on va extraire pour déterminer les caractéristiques de notre modèle.

On copie ici le code de *SelectPixelsCenter.m* que nous devons essayer de comprendre :

---

```
1 function cim=SelectPixelsCentre (image , p)
2
3 I=imread (image);
4 [M N O]=size (I);
5
6 minx = floor ((M-p*M)/2);
7 miny = floor ((N-p*N)/2);
8 maxx= minx+ floor (p*M);
9 maxy= miny+ floor (p*N);
10
11 cim = I (minx : maxx , miny : maxy , :);
```

---

Cette fonction est celle utilisée pour prendre seulement une zone centrale de l'image. L'attribut  $p$  détermine la taille de cette zone. Cette approche a l'avantage d'être économe en temps de calcul. Cependant, elle ne fonctionne que si la zone d'intérêt (ici, le visage de l'ex-président) se situe dans au centre de l'image. Dans le cas contraire, il faudra faire appel à une technique différente.

#### 3.1.2 Choix des caractéristiques

Pour cette étude, il est proposé d'utiliser l'information chromatique de la peau. Cette approche a l'avantage d'être un vecteur de faible dimension. Ce plus, cette information permet de bien séparer la peau du reste de l'image car la chromaticité de la peau restera très différente de celle du fond.

Nous nous placerons dans l'espace  $Y C_r C_b$ . Sachant que l'information de luminance de nous intéresse pas, nous ne travaillerons que sur deux informations ce qui est plus facile à gérer.

L'histogramme obtenu suite à l'exécution du code *plot\_hist2d.m* représente les valeurs de  $C_r$  et  $C_b$  de chaque pixels des 20 images de la base d'échantillon. On observe que les valeurs de  $C_r$  et  $C_b$  sont localisées en deux pics<sup>4</sup>.

---

4. Dont un principal vers (150,150) et un secondaire vers (80,80)

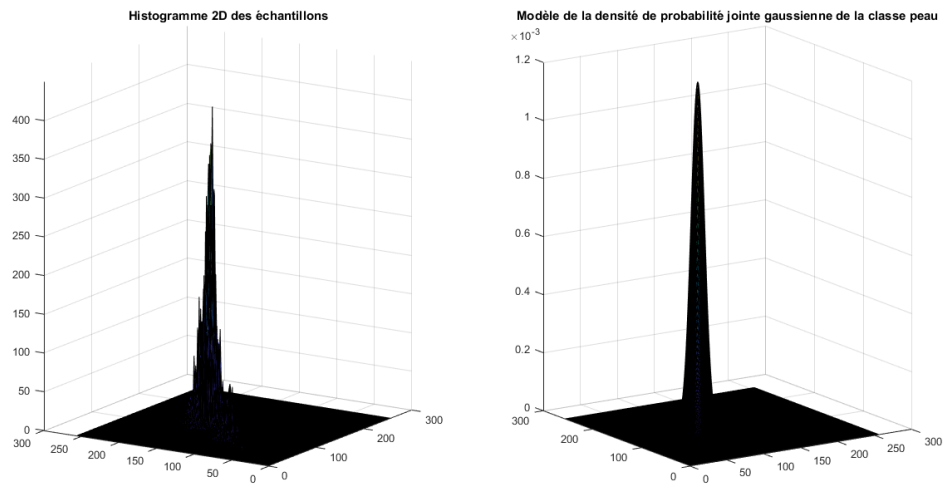


FIGURE 4 – Histogramme 2D des composantes  $C_r$  et  $C_b$  sur 20 images comparé au modèle gaussien - vue en 3D

Sur la figure suivante on regarde ces histogrammes vus du dessus pour avoir une vision un peu plus claire des choses.

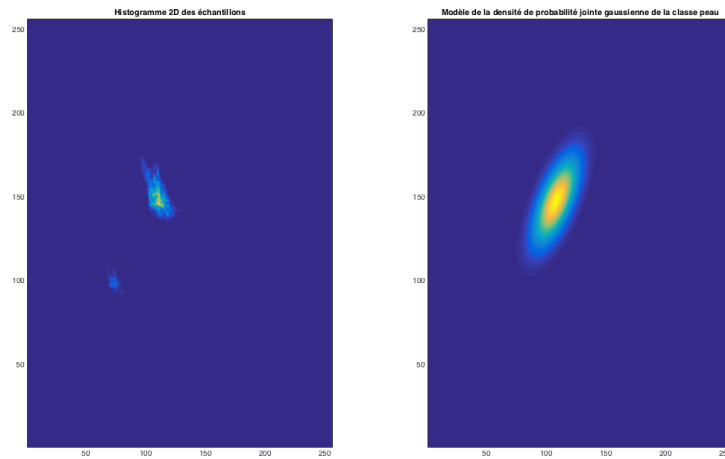


FIGURE 5 – Histogramme 2D des composantes  $C_r$  et  $C_b$  sur 20 images comparé au modèle gaussien - vue du dessus

Comme modèle de probabilité théorique que l'on va utiliser pour modéliser l'information de couleur de la peau on va prendre un modèle *gaussien multi-varié*.



### 3.1.3 Calcul des paramètres du modèle

On rappelle l'équation 7 du modèle gaussien multi-varié :

$$p(\mathbf{x}|\omega_1) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_1|^{\frac{1}{2}}} \exp \left[ -\frac{1}{2}(\mathbf{x} - \mu_1)^T \Sigma_1^{-1} (\mathbf{x} - \mu_1) \right]$$

On a un modèle où le vecteur de données est un vecteur contenant deux paramètres<sup>5</sup> donc la matrice de covariance doit être une matrice à deux dimensions, ce qui est bien le cas ici :

$$\Sigma_1 = \begin{pmatrix} 243,17 & 115,27 \\ 115,27 & 128,89 \end{pmatrix}$$

On voit sur les figures 4 et 5 que le modèle gaussien ne possède qu'un seul mode. Le modèle semble toutefois proche des observations avec un maximum correspondant au maximum de l'histogramme des images d'apprentissage.

### 3.1.4 Application du modèle à une image test

Le modèle de probabilité de la classe peau est bien pertinent avec les résultats que l'on a, comme sur la figure suivante<sup>6</sup> :

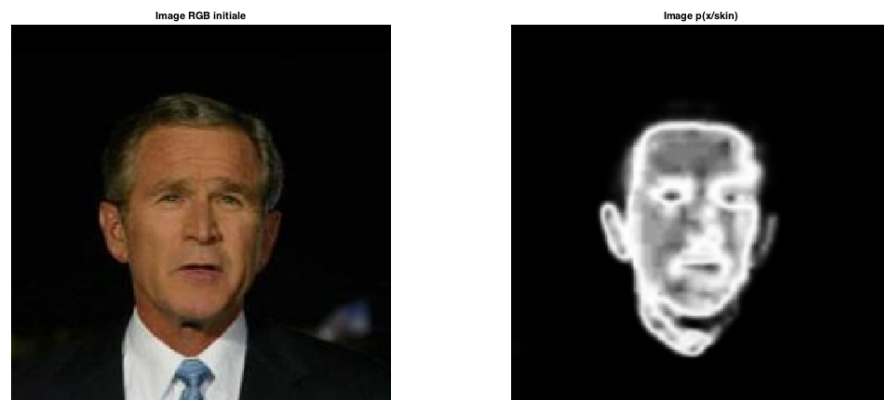


FIGURE 6 – Image originale et reconnaissance de la peau

---

5. Un vecteur 2x1

6. D'autres résultats peuvent être trouvés en annexe C.1.

Cette approche nous permet, de plus, de faire de la segmentation de visage sur les images. On le fait et on trouve le résultat suivant<sup>7</sup> :



FIGURE 7 – Image originale et image segmentée du visage

Ce modèle présente tout de même certaines limites que l'on illustre dans les figures en annexe C.3 où parfois on ne détecte pas tout le visage ou alors le fond est aussi reconnu comme de la peau alors que cela n'est pas le cas.

## 3.2 Partie 2 : Vers un classifieur bayésien

### 3.2.1 Classe "non-peau"

Nous allons ici travailler au développement d'un classifieur bayésien. Pour cela, il convient de rappeler l'expression suivante :

$$p(\omega_1|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_1)\mathbb{P}(\omega_1)}{p(\mathbf{x})}$$

Nous allons considérer que  $\mathbb{P}(\omega_1) = \mathbb{P}(\omega_2) = \frac{1}{2}$ .

L'expression précédente devient alors :  $p(\omega_1|\mathbf{x}) = \frac{1}{2} \frac{p(\mathbf{x}|\omega_1)}{p(\mathbf{x})}$ .

Afin d'obtenir  $p(\mathbf{x}|\omega_2)$ , on prend les bords des images de la banque d'échantillon. Pour cela on utilise la fonction *SelectPixelsNonCentre* dont le code est donné ci-dessous. Cette fonction ; contrairement qui travaillait sur l'image brute, travaille uniquement sur les composantes  $C_b$  et  $C_r$  d'un image, qui font partie des paramètres d'entrée de la fonction.

---

7. On peut voir d'autres résultats de segmentation en annexe C.2.

---

```

1  function cim=SelectPixelsNonCentre (cb , cr , p)
2
3  [M N]=size (cb );
4
5  minx = floor ((M-p*M)/2);
6  miny = floor ((N-p*N)/2);
7  maxx= minx+ floor (p*M);
8  maxy= miny+ floor (p*N);
9
10 n_sup = (maxy-miny+1)*(maxx-minx+1);
11 cim = zeros (M*N-n_sup ,2);
12
13 index = 1;
14 for x = 1:M
15     for y = 1:N
16         if (x < minx)
17             cim(index ,1) = cb(x,y);
18             cim(index ,2) = cr(x,y);
19             index = index + 1;
20         elseif (x > maxx)
21             cim(index ,1) = cb(x,y);
22             cim(index ,2) = cr(x,y);
23             index = index + 1;
24         else
25             if (y < miny | y > maxy)
26                 cim(index ,1) = cb(x,y);
27                 cim(index ,2) = cr(x,y);
28                 index = index + 1;
29             end
30         end
31     end
32 end

```

---

On va pouvoir faire un apprentissage sur de la classe "non-peau" comme nous l'avons fait sur la classe "peau" précédemment.

### 3.2.2 Classifieur optimal

## A Compléments et figures annexes de l'exercice I

### A.1 Tableau de la probabilité de l'erreur

On donne dans le tableau ci-dessous les différents résultats que nous donne la classification bayésienne en fonction des paramètres du modèle de la classe 2.

$\mu_2$	$\sigma_2$	$x_B$	$\mathbb{P}(\text{erreur})$
1	$1/\sqrt{2}$	0,5	0,2398
0,5	$1/\sqrt{2}$	0,25	0,3818
2	$1/\sqrt{2}$	1	0,0787
1	0,25	0,59 et 1,71	0,0275
1	1	-2,64 et 0,65	0,3188
1	2	-1,3 et 1,02	0,2750

TABLE 1 – Probabilité de l'erreur avec de nouveaux paramètres pour la classe 2

### A.2 Graphiques avec les nouveaux paramètres

On modifie certains paramètres de la classe 2 et on trace les graphiques correspondant.



FIGURE 8 – Données de probabilités avec  $\mu_2 = 0,5$  et  $\sigma_2 = 1/\sqrt{2}$



FIGURE 9 – Données de probabilités avec  $\mu_2 = 2$  et  $\sigma_2 = 1/\sqrt{2}$

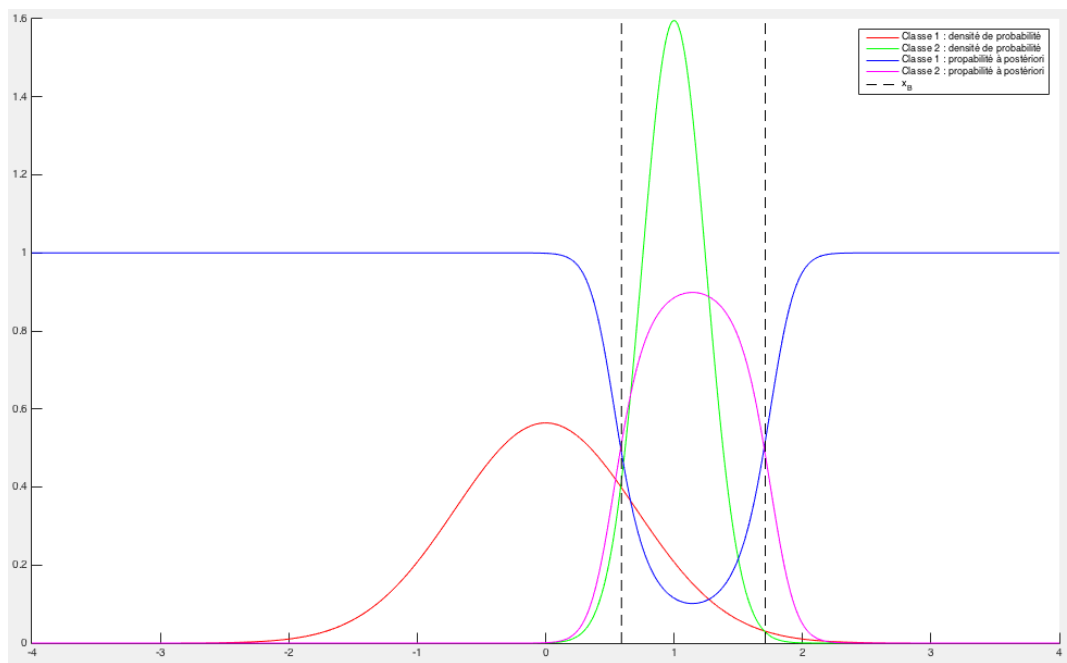


FIGURE 10 – Données de probabilités avec  $\mu_2 = 1$  et  $\sigma_2 = 0,25$

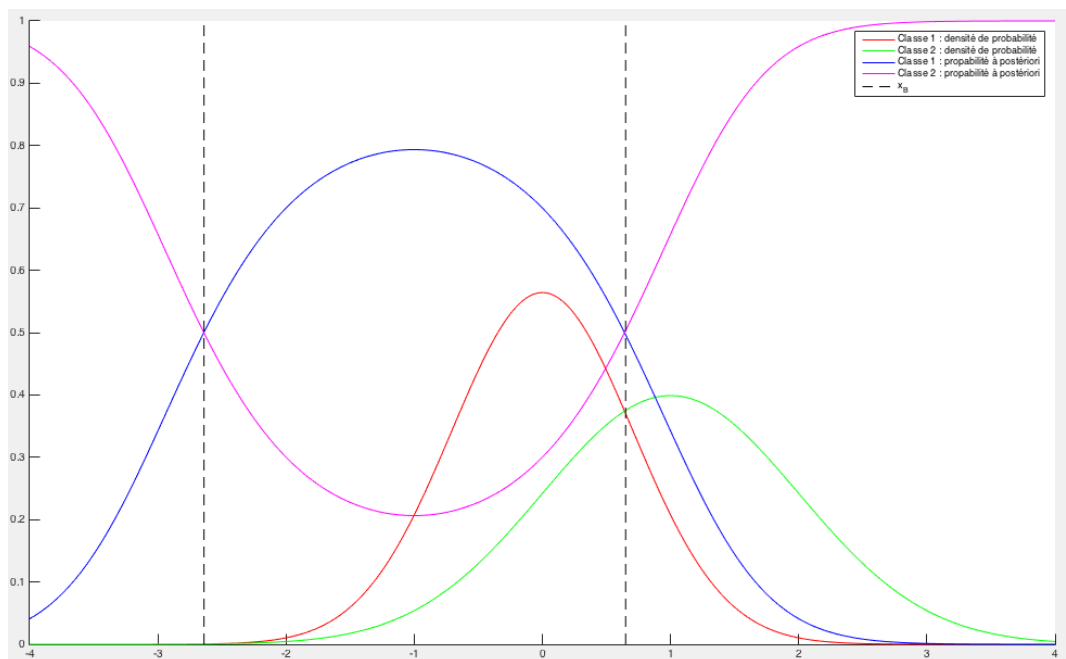


FIGURE 11 – Données de probabilités avec  $\mu_2 = 1$  et  $\sigma_2 = 1$

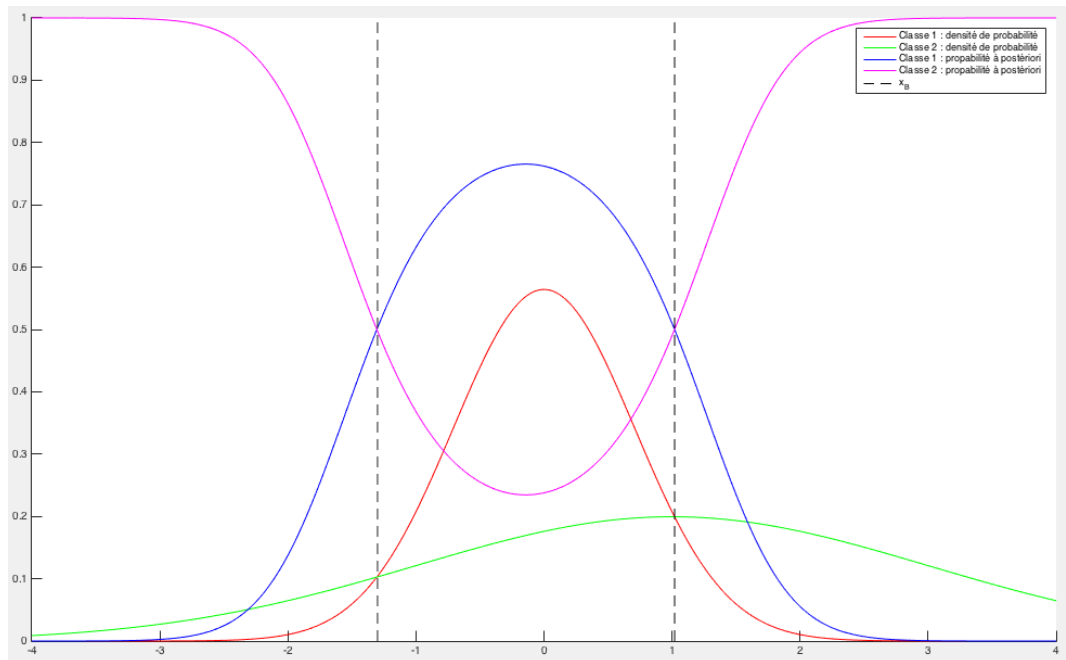


FIGURE 12 – Données de probabilités avec  $\mu_2 = 1$  et  $\sigma_2 = 2$

## B Code *MatLab* de l'exercice I

```
1 %% TD Reconnaissance de Formes — Exercice 1
2
3 clc;
4 close all;
5 clear;
6
7 Max_x = 4;
8 Min_x = -4;
9 Pas = 0.01;
10 x = [Min_x : Pas : Max_x];
11 L = length(x);
12
13 % Considerons que 2 classes sont modelisees par les densites de
14 % probabilite gaussiennes suivantes :
15 pxw1 = exp(-(x.*x))./ sqrt(pi);
16 mu2 = 1;
17 sigma2 = sqrt(0.5); % ecart-type
18 %sigma2 = 2;
19 pxw2 = exp(-(x-mu2).*(x-mu2)/(2*sigma2^2))./ sqrt(2*pi*sigma2^2);
20
21 %% 1. 1er cas: Pw1=Pw2=0,5
22 Pw1 = 0.5;
23 Pw2 = 1 - Pw1;
24
25 px = pxw1 * Pw1 + pxw2 * Pw2; % calcul de la probabilite totale
26
27 Pw1x = pxw1 * Pw1 ./ px;
28 Pw2x = pxw2 * Pw2 ./ px;
29
30
31 % Seuil
32 i = 1;
33 g1 = log(pxw1(1) * Pw1);
34 g2 = log(pxw2(1) * Pw2);
35 for k = 2 : length(x)
36     g1m = log(pxw1(k) * Pw1);
37     g2m = log(pxw2(k) * Pw2);
38
39     if (g1 - g2)*(g1m - g2m) < 0
40         xB(i) = x(k);
41         i = i + 1;
42     end
43
44     g1 = log(pxw1(k) * Pw1);
45     g2 = log(pxw2(k) * Pw2);
46
47 end
48
49 % Calcul de l'erreur
50 %indices = find(x <= xB(1) | xB(2) <= x);
51 indices = find(x <= xB(1));
52 M = length(indices);
53 Perreur = Pas * sum(pxw2(indices) * Pw2);
```

```

54 Perreur = Perreur + Pas * sum(pxw1(indices(M) + 1:L) * Pw1);
55
56 % Traces des densites de probabilites des 2 classes
57 figure(1)
58 hold on;
59 plot(x,pxw1,'color','red');
60 plot(x,pxw2,'color','green');
61 plot(x,Pw1x,'color','blue');
62 plot(x,Pw2x,'color','magenta');
63 plot([xB(1) xB(1)],[0 1],'—','color','black');
64 %plot([xB(2) xB(2)],[0 1],'--','color','black');
65 hold off;
66 legend('Classe 1 : densite de probabilite','Classe 2 : densite de probabilite', ...
67        'Classe 1 : propabilite a posteriori', ...
68        'Classe 2 : propabilite a posteriori','x_B');
69
70
71
72 %% 2. Modification des valeurs de probabilite a priori
73 Pw1 = 9/10;
74 Pw2 = 1/10;
75
76 px = pxw1 * Pw1 + pxw2 * Pw2; % calcul de la probabilite totale
77
78 Pw1x = pxw1 * Pw1 ./ px;
79 Pw2x = pxw2 * Pw2 ./ px;
80
81
82 % Seuil
83 i = 1;
84 g1 = log(pxw1(1) * Pw1);
85 g2 = log(pxw2(1) * Pw2);
86 for k = 2 : length(x)
87     g1m = log(pxw1(k) * Pw1);
88     g2m = log(pxw2(k) * Pw2);
89
90     if (g1 - g2)*(g1m - g2m) < 0
91         xB(i) = x(k);
92         i = i + 1;
93     end
94
95     g1 = log(pxw1(k) * Pw1);
96     g2 = log(pxw2(k) * Pw2);
97
98 end
99
100 % Calcul de l'erreur
101 indices = find(x <= xB);
102 M = length(indices);
103 Perreur = Pas * sum(pxw2(indices) * Pw2);
104 Perreur = Perreur + Pas * sum(pxw1(indices(M) + 1:L) * Pw1);
105
106 % Traces des densites de probabilites des 2 classes
107 figure(2)
108 hold on

```



```

109 plot(x,pxw1,'color','red');
110 plot(x,pxw2,'color','green');
111 plot(x,Pw1x,'color','blue');
112 plot(x,Pw2x,'color','magenta');
113 plot([xB xB],[0 1],'—','color','black');
114 hold off;
115 legend('Classe 1 : densite de probabilite','Classe 2 : densite de probabilite', ...
116        'Classe 1 : propabilite a posteriori', ...
117        'Classe 2 : propabilite a posteriori','x_B');
118
119
120 % Prise en compte du numerateur uniquement
121 Pw1x_modif = pxw1 * Pw1;
122 Pw2x_modif = pxw2 * Pw2;
123
124
125 figure(3)
126 hold on
127 plot(x,pxw1,'color','red');
128 plot(x,pxw2,'color','green');
129 plot(x,Pw1x_modif,'color','blue');
130 plot(x,Pw2x_modif,'color','magenta');
131 plot([xB xB],[0 0.6],'—','color','black');
132 hold off;
133 legend('Classe 1 : densite de probabilite','Classe 2 : densite de probabilite', ...
134        'Classe 1 : propabilite a posteriori', ...
135        'Classe 2 : propabilite a posteriori','x_B');

```

## C Figures annexes de la partie I de l'exercice II

### C.1 Reconnaissance de la peau

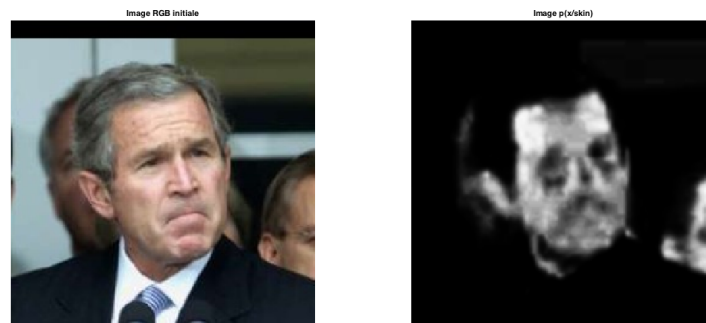


FIGURE 13 – Image originale et reconnaissance de la peau

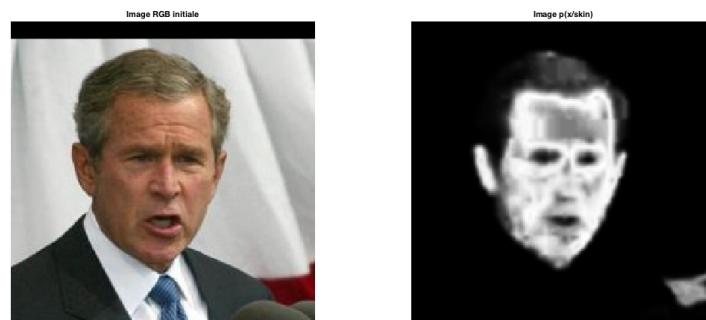


FIGURE 14 – Image originale et reconnaissance de la peau

## C.2 Images segmentées

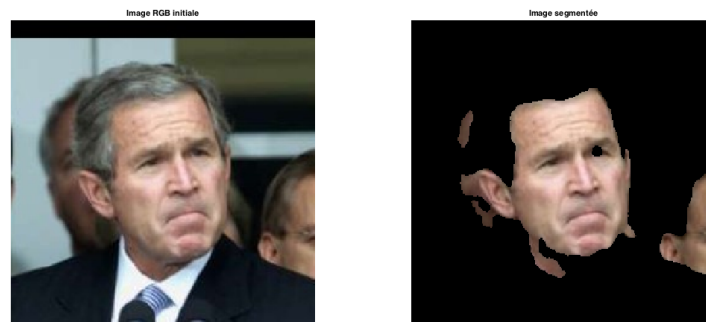


FIGURE 15 – Image originale et image segmentée du visage



FIGURE 16 – Image originale et image segmentée du visage

### C.3 Limites du modèle

On voit de la mauvaise segmentation pour les deux figures suivantes en utilisant notre modèle :



FIGURE 17 – Mauvaise segmentation de la peau



FIGURE 18 – Mauvaise segmentation de la peau

## D Code *MatLab* de l'exercice II

### D.1 Code de la fonction *get\_cbc*

```
1 function [cb, cr] = get_cbc(im)
2 % appel de la fonction de conversion de Matlab RGB to YCbCr
3 imycc = rgb2ycbcr(im);
4 % filtrage passe bas de chaque plan couleur
5 % definition du filtre
6 lpf = 1/9 * ones(3);
7 % extraction des 2 composantes chromatiques
8 cb = imycc(:, :, 2);
9 cr = imycc(:, :, 3);
10 % filtrage
11 cb = filter2(lpf, cb);
12 cr = filter2(lpf, cr);
```

### D.2 Code de la fonction *plot\_hist2d*

```
1 function plot_hist2d(cr, cb)
2 %creation d'une matrice 256 x 256 de zeros pour initialisation d'une
3 %matrice quantifiee
4 hist2d = zeros(256);
5 %approximation entiere des valeurs de chrominance
6 cr = round(cr);
7 cb = round(cb);
8 %calcul de l'histogramme 2D
9 for i = 1:length(cr)
10     hist2d(cr(i), cb(i)) = hist2d(cr(i), cb(i)) + 1;
11 end
12 %representation 2D de l'histogramme
13 surf(hist2d);
```

### D.3 Code de la fonction *GaussLikelihood*

```
1 function pxlikelihood=GaussLikelihood(xs, mu, sig_inv, d_sig)
2
3 d = length(mu);
4 coef_norm = (1/((2*pi)^(d/2)*d_sig^0.5));
5 pxlikelihood = coef_norm*exp(-(xs-mu)'*sig_inv*(xs-mu)/2);
```

### D.4 Code de la fonction *SelectPixelsCentre*

```
1 function cim=SelectPixelsCentre(image, p)
2
3 I=imread(image);
4 [M N O]=size(I);
5
6 minx = floor((M-p*M)/2);
7 miny = floor((N-p*N)/2);
8 maxx= minx+ floor(p*M);
9 maxy= miny+ floor(p*N);
10
11 cim = I(minx:maxx, miny:maxy, :);
```

## D.5 Code de la fonction *SelectPixelsNonCentre*

```
1 function cim=SelectPixelsNonCentre(cb,cr,p)
2
3 [M N]=size(cb);
4
5 minx = floor((M-p*M)/2);
6 miny = floor((N-p*N)/2);
7 maxx= minx+ floor(p*M);
8 maxy= miny+ floor(p*N);
9
10 n_sup = (maxy-miny+1)*(maxx-minx+1);
11 cim = zeros(M*N-n_sup,2);
12
13 index = 1;
14 for x = 1:M
15     for y = 1:N
16         if(x < minx)
17             cim(index,1) = cb(x,y);
18             cim(index,2) = cr(x,y);
19             index = index + 1;
20         elseif(x > maxx)
21             cim(index,1) = cb(x,y);
22             cim(index,2) = cr(x,y);
23             index = index + 1;
24         else
25             if(y < miny | y > maxy)
26                 cim(index,1) = cb(x,y);
27                 cim(index,2) = cr(x,y);
28                 index = index + 1;
29             end
30         end
31     end
32 end
```

## D.6 Code principal

```
1 Toiiiiiiiiiii
```