

TP 1 : DÉTECTION DE PEAU

MEDOU Daniel Magloire, IFI-Promo 21

Niveau : Master 2, option SIM

Implémenter un programme permettant de détecter la peau humaine dans les images, tel est l'objectif de ce TP1. Ici, le fait est l'identification des pixels qui représentent la peau dans les images fournies au programme en entrée. Ceci devrait bien se passer quelque soit la couleur de celle-ci et le type d'exposition de la lumière.

Dans le présent rapport, nous trouverons le rendu de notre effort, les résultats et commentaires.

I. Description et fonctionnement du programme

L'échelle de réduction des valeurs de l'espace de couleurs, la valeur de seuil de décision et l'image à évaluer sont les différents éléments que le programme prend en entrée comme argument. L'image entrée, l'image obtenue après détection de la peau, le masque de détection, l'image de référence correspondante dans la base d'images de peau et les histogrammes de peau et de non-peau de sa base d'apprentissage sont là les résultats que le programme nous donne en sortie.

les actions décrites ci-dessous sont essentielles à effectuer pour utiliser notre programme.

- **La base d'images** : C'est un répertoire accompagnant le programme appelé « base » dans lequel, nous retrouvons aussi trois sous-répertoire à savoir « **peau** » les images de ce dernier ne laissent apparaître que les pixels décrivant la peau, « **non-peau** » ici, pas de pixels peau et en fin « **test** », contient les images d'origine ayant été traitées pour l'obtention de nos répertoires peau et non-peau . Rappelons que, 23 est le nombre d'images de chacun.
- **Compilation et exécution du programme**: Pour compiler ce dernier, nous devons nous rassurer d'être dans le dossier où se trouve le code source de là, ouvrons le terminal et tapons la commande « **make** ». Une fois la compilation terminée avec succès, nous pouvons exécuter notre programme en tapant la commande donc la syntaxe est la suivante :
./detection_peau « echelle » « seuil » « nom_image_test »
« echelle » correspond à l'échelle de réduction dans l'espace des couleurs,
« seuil » est le seuil de décision permettant de classer un pixel comme peau ou non et
« nom_image_test » est le nom de l'image fournie en entrée au programme.

Illustration : **./detection_peau 32 0.4 imag1_peau.jpg**

Permet de détecter les pixels peau dans l'image 1.jpg avec une échelle de 32 et un seuil de décision de 0.4 .

II. Description de l'implémentation

Notre programme de détection de la peau est un processus se décomposant en cinq étapes à savoir : La création de la base d'images, la construction des histogrammes, la détection de la peau dans les images, le post-traitement et l'évaluation. Trouvons ci-dessous la description de ces différentes étapes.

1. Création de la base d'images

Notre base d'images a été créée à partir de vingt-trois (23) images obtenues sur internet. Ces images, contiennent des personnes de différentes couleurs, étant différemment habillées et dans des décors divers. L'exposition aux différentes sources lumineuses de ces images a été prise en compte afin, d'assurer une base d'apprentissage suffisamment diversifiée à notre programme.

Une fois les images sélectionnées, nous avons créer un répertoire « peau » qui, contient les images faisant ressortir que les pixels décrivant la peau des individus et la valeur des autres pixels étant mise à 0. Un second répertoire « non-peau » contenant des images dans lesquelles seule la peau est

occultée et la valeur des autres pixels mise à 0. Le dossier « test » contient les images originales nous permettant de créer les deux autres dossiers.

La construction des histogrammes, réservée à l'étape suivante utilisera seulement les images des répertoires « peau » et « non-peau ».

2. Construction des histogrammes

La construction des histogrammes est une étape très importante dans le processus de détection des pixels peau. Deux histogrammes devaient être construits. Un pour les images « peau » et l'autre pour les images « non-peau ». Dans chaque cas, la conversion des images de l'espace de couleurs RGB à l'espace de couleur Lab était une opération primordiale. Dans un souci de diminuer les calculs à effectuer, nous avons pensé à réduire l'espace des valeurs des pixels de $256 \times 256 \times 256$ à $\text{echelle} \times \text{echelle} \times \text{echelle}$. « echelle » étant la première valeur fournie en paramètre au programme. Par exemple si l'échelle fournie est 32, nous réduirons l'espace des valeurs de $256 \times 256 \times 256$ à $32 \times 32 \times 32$. Pour ce faire nous associons toutes les valeurs de pixels comprises entre deux multiples du facteur « $256/\text{echelle}$ » à la valeur du plus petit des multiples considérés. A titre d'exemple, pour une échelle de réduction égale à 32, toutes les valeurs comprises entre 0 et 8 (exclu) seront associées à la valeur 0 dans le nouvel espace. Après la réduction, nous comptons le nombre d'occurrences de chaque couple de valeurs (a, b), dans le répertoire d'images considéré. Ainsi à la fin de cette étape, nous obtenons un tableau contenant le nombre d'occurrences des différents couples. Ce tableau n'est rien d'autre que l'histogramme. Il est à noter que nous ne prenons pas en compte les parties en noir de l'image et la composante L de l'image convertie. Une fois l'histogramme construit, nous lui appliquons un lissage afin d'améliorer la qualité de la détection. Rappelons que le lissage consiste à calculer pour chaque pixel la moyenne des valeurs de ses huit (08) voisins et de sa propre valeur.

La normalisation des valeurs de notre histogramme est la fin de cette étape. Cette dernière consisterait à diviser la valeur de chaque case de l'histogramme par la somme des valeurs de toutes les cases de l'histogramme.

Nous devons tout de même connaître que pour l'affichage de l'histogramme celui-ci est remis à l'échelle $256 \times 256 \times 256$ pour une bonne visualisation. L'opération est exactement l'inverse de celle ayant permis la réduction de l'espace des valeurs. Les figures 1 et 2 montrent les histogrammes « peau » et « non-peau » obtenus à partir de notre base d'images.

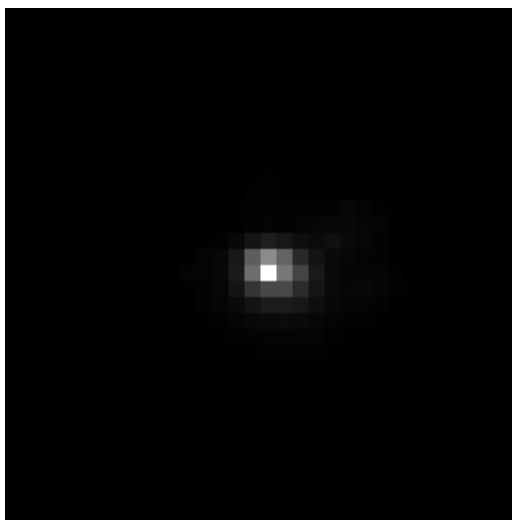


Figure1 : Histogramme « non-peau »

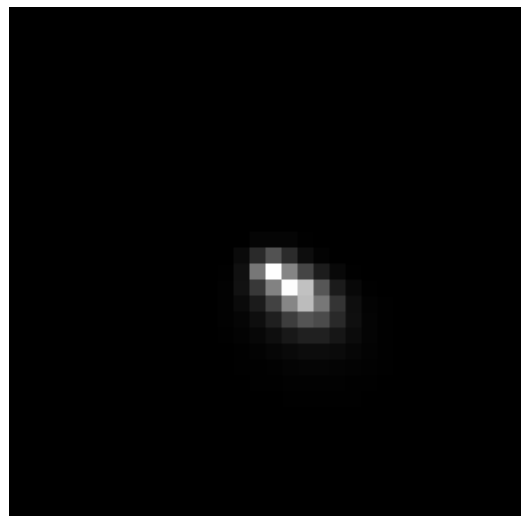


Figure2 : Histogramme « peau »

3. Détection de peau

Dans cette étape, il est question pour nous d'exploiter les histogrammes ci-dessus afin de détecter les pixels peau dans les images qui fournies au programme.

De prime à bord, nous chargeons l'image fournie en paramètre dans l'espace de couleurs RGB ensuite la convertir dans l'espace Lab. Nous poursuivons par une réduction de l'espace des valeurs de pixels comme dans le cas de la construction de l'histogramme et puis nous enchaînons avec la détection proprement dite.

Celle-ci consiste à sélectionner pour chaque couple (a,b) d'un pixel de l'image fournie, les valeurs correspondantes à ce couple dans les histogrammes « peau » et « non-peau ». Calculer la probabilité que le pixel considéré soit un pixel peau en se basant sur la formule de Bayes décrite dans l'énoncé du TP. Par la suite la probabilité obtenue est comparée à la valeur de seuil fournie en paramètre au programme. Si la probabilité est supérieure au seuil le pixel est classé comme « peau ». Dans le cas contraire il est considéré comme « non-peau ». Il est alors construit un masque où les pixels peau ont la valeur 255 et ceux non-peau la valeur 0. Ce masque sera, pour finir, utilisé pour restituer l'image détectée.

4. Post traitement

Dans cette étape, nous expliquons comment améliorer la détection de la peau qui a été faite précédemment. En effet, les imperfections sont notées dans les images de détection de peau et pour les corriger, nous appliquons les opérations de dilatation et d'érosion aux images détectées.

5. Évaluation

Cette étape nous permet de connaître ou d'évaluer les performances de notre programme. La performance nous est donnée par la formule (confer TP3 du cours de traitement d'image).

$$\text{Performance} = \frac{\text{nb_pixels_corrects}}{(\text{nb_pixels_corrects} + \text{nb_pixels_faux_pos} + \text{nb_pixels_faux_neg})}$$

nb_pixels_corrects : Nombre pixels correctement détectés

nb_pixels_faux_pos : Nombre pixels détectés comme peau mais qui sont non-peau

nb_pixels_faux_neg : Nombre pixels détectés comme non-peau mais qui sont peau

Il est important de noter que cette mesure est entachée d'un certain biais étant donné que nous n'avons pas pu établir des vérités parfaites c'est à dire des images de référence (images « peau ») parfaites. Nous l'utilisons donc à titre indicatif seulement.

III. Test, résultats et analyse

Ici nous présentons nos tests, les résultats sont donnés et ensuite analysés. Nos tests sont subdivisés en deux parties à savoir les **tests sur les images de la base d'apprentissage**, les **tests sur les images hors base**. Pour décider sur la valeur de seuil qui nous permettra de continuer nos expériences, il est préférable d'effectuer une série de tests basée sur une image sur laquelle sera appliquée ces valeurs de seuil. D'où le rôle du seuil sur la détection de la peau.

1. Rôle du seuil sur la détection de la peau

Les valeurs comprises entre 0 et 0.6, sont les seuils qui seront appliqués sur les images. la performance de notre programme sera aussi relevé afin de déterminer le meilleur seuil. Les résultats ci-dessous nous permettrons prendre ensemble le bon seuil ainsi que la performance du programme. Avant les tests, nous présentons respectivement l'images originale de la base test et l'image de la base peau qui est utilisée est 23.jpg et est la suivante :

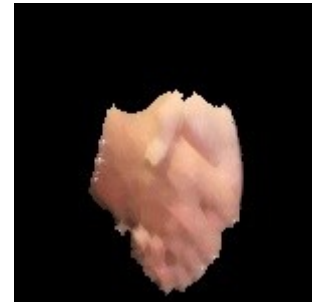


23.jpg (Image originale)

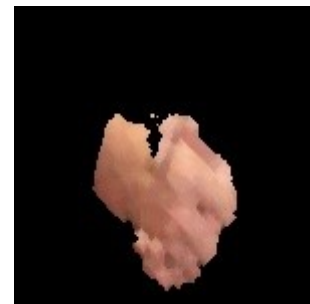
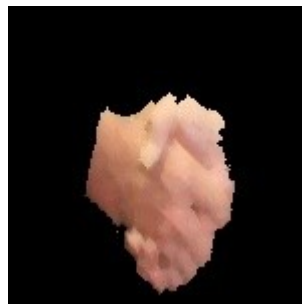
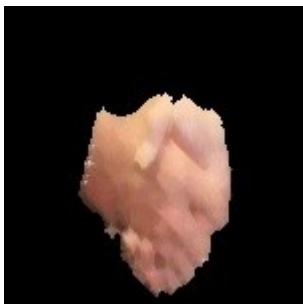


23.jpg

Les résultats aux différents tests sont des images ci-dessous :



Seuil = 0.1, performance = 31.27 Seuil = 0.2, performance = 33.11 Seuil = 0.3, performance = 73.63



Seuil = 0.4, performance = 73.76 Seuil = 0.5, performance = 71.23 Seuil = 0.6, performance = 59.47

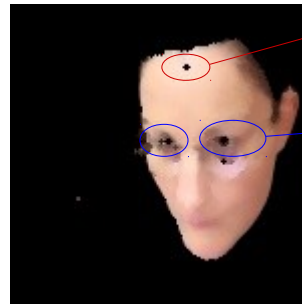
Nous constatons après les différents tests effectués, à des seuils différents et avec la même échelle de réduction « 32 » que, les seuils 0.3, 0.4 et 0.5 nous donne de bons ce qui n'est pas le cas pour les seuils 0.1 et 0.2 qui, détecte le « peau » et la « non-peau » en même temps. De ce constat, pour la suite des tests de notre projet, nous **utiliserons « 0.4 » comme seuil des autres tests car la en plus de la détection de la peau, il nous donne la meilleure performance « 73.76 »**

2. Détection dans les images de la base d'apprentissage :

Cas de peau blanche



Figure3 : Image 17.jpg



Faux négatif

Faux positif

Figure4 : rendu_image_17.jpg

`./detection_peau 32 0.4 17.jpg` et Performance du programme = 62.5914 %

Au vue du rendu de ce test, nous pouvons dire que l'image « Figure4 » est très bien pour la grande partie de pixels peau détecté et la performance largement au-dessus de 50%. Notons des imperfections sur notre image à savoir les faux négatifs (un point noir sur le visage encerclé en rouge) et les faux positifs (les points noirs présents sur les yeux de la femme et encerclé en bleu). Nous concluons que le programme détecte bien la peau de couleur blanche en plus le programme ne sachant dans quelle catégorie classer les yeux, les considère comme peau.

Cas de peau noir



Figure5 : Image 16.jpg



Figure6 : rendu_image_16.jpg

`./detection_peau 32 0.4 16.jpg` et Performance du programme = 62.838 %

La peau a été détectée « Figure6 » mais nous remarquons des faux positifs sur les yeux comme sur l'image précédente ainsi que les faux négatifs sur la peau. Constat de plus, les dents sur cette image sont considérées comme peau car elles ont été aussi détecté par notre programme qui ne sait pas aussi où classer les dents soit comme « peau » ou « non-peau ».

Autre cas de peau



Figure7 : Image 14.jpg

Faux négatifs



Figure8 : rendu_image_14.jpg

`./detection_peau 32 0.4 14.jpg` et Performance du programme = 61.2177 %

La Figure8 nous présente un résultat satisfaisant car presque tous les pixels de la peau ont été détecté. Nous remarquons deux tâches noirs représentant les faux négatifs encadrées en rouge. En plus des dents que notre programme a considéré comme « peau », il le fait aussi pour cette image et une éventuelle détection des cheveux que le programme considérerait comme peau.

Nous pouvons retenir que la peau a été bien détectée. Sans pour autant négliger les imperfections mentionnées par les différents cercles pour certains images en raison de l'insuffisance de caractéristiques dans les images de la base et de la proximité de certains pixels non-peau avec des pixels peau.

3. Détection dans des images hors de la base d'apprentissage

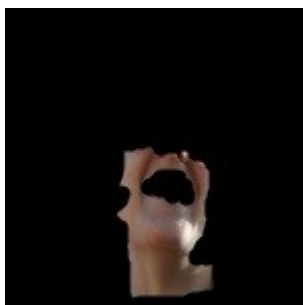


Figure9 : Image reference1

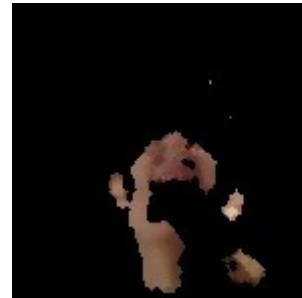


Figure10 : Image détectée

`./detection_peau 32 0.4 hors_base1.jpg` et Performance du programme = 35.8901 %

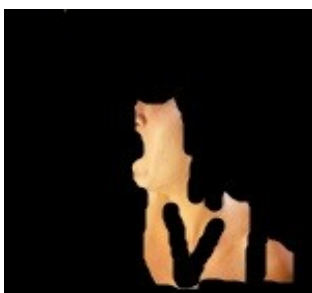


Figure11 : Image reference2



Figure12 : Image détectée

`./detection_peau 32 0.4 hors_base2.jpg` et Performance = 53,25%

Sur les différentes images testées ont notes que la peau est relativement bien détectée. Par ailleurs les mêmes imperfections (détection de cheveux...) sont également présente. Notre détecteur se comporte donc également bien sur les images en dehors de la base. Avec ces images, la performance du programme est moins bien que sur d'autres images.

Conclusion

En définitive, la réalisation du programme du TP1 pour la détection de la peau a été faite selon les consignes indiquées sur la sujet. Nous avons monter une base d'images qui nous a permis de passer des tests de détection de la peau à partir de notre programme. Cette dernière est constituée de 23 images de différentes couleurs, de forme et de degré de lumière différent. Les images, divisées en trois répertoires donc un pour celles faisant ressortir la peau, un autre les occultant et le dernier constitué des images originales. Les tests effectués avec ces images à partir du programme nous permettent de détecter la peau quelque soit la couleur de celle-ci. Ce qui nous laisse dire que le programme tourne bien. Il est important de noter que quelques éléments n'étant pas peau ont été détecté comme s'ils étaient des pixels peau (dent, cheveux,...).

En perspective, nous comptons améliorer notre programme avec l'évolution du cours tout en ajoutant ou bien essayant d'autres méthodes de détection de la peau sur les images.