

# UNIVERSITÉ NATIONAL DU VIETNAM (UNV)

## INSTITUT FRANCOPHONE INTERNATIONAL (IFI)



Promotion 24  
Système Intelligent et Multimédia

Rapport Tp1 Vision par Ordinateur  
Sujet: Détection de la peau  
Professeure: Nguyen Thi Hoan  
Année Académique: 2019 - 2020

Rédigé par: Altema Emaude, Bertelot Peterson et Janvier fabienne

Novembre 2020

## Introduction

La Détection de la peau révèle du domaine de la vision par ordinateur, un domaine de l'intelligence artificielle très connu pour sa capacité à faire fonctionner un ordinateur comme un humain exerçant le sens de la vue chez l'homme.

Dans le cadre de ce travail il nous a été demandé d'implémenter un modèle de détection de peau en utilisant une base d'image recueillie manuellement par notre équipe.

L'implémentation de ce programme a été faite pour le compte du premier tp concernant le Cours vision par ordinateur, il prend en compte l'évaluation de l'image saisie en entrée des arguments : échelle de réduction des valeurs de l'espace de couleurs, la valeur du seuil de décision et l'image à évaluer. A la sortie des différents traitements, le programme produit les histogrammes de peau et non peau de toutes les images de la base d'apprentissage. Il retourne aussi l'image de peau détectée, le masque de détection et l'image de référence trouvée dans la base d'image de peau.

## Pour exécuter ce code

Pour exécuter ce programme, deux actions sont considérées. Premièrement, il faut compiler le programme en exécutant la commande « make » sans les guillemets, ensuite, il faudra exécuter la commande `./détection_peau « échelle » « seuil » « nom image à évaluer »`. D'abord pour arriver à ces actions, il faut d'un premier temps se placer dans le répertoire du programme, dans mon cas « vision\_par\_ordi\_TP1\_Master\_2 ». Ce répertoire contient un sous répertoire « base » qui contient les sous répertoires « peau », « non\_peau » et « test » ; un autre sous répertoire « histogramme », créer pour sauvegarder les histogrammes peau et non peau et un autre « résultat » destiné à recevoir la sortie qui dans notre cas est l'image détectée après traitements.

En guise d'exemple : `./détection_peau 32 0.2 17.jpg`

## Implementation

Ce travail concernant la détection de la peau est implémenté en plusieurs grandes étapes, soit cinq au total. En tout premier lieu, on a abordé la création de la base de données image pour enchaîner avec la construction des histogrammes, la détection de la peau, le post-traitement et l'évaluation.

### ❖ Création de la base de données

Cette base de données image qu'on a créée est composée de vingt (20) images de différentes qualité téléchargées sur internet. On s'est assuré que dans notre base de données les images sont de différentes couleurs, et que leur exposition aux sources lumineuses étaient toutes différentes afin de s'assurer que la base de données image d'apprentissage est assez diversifiée. Donc, après avoir recueilli nos images, nous avons regroupé dans un premier temps toutes les images qui cachent les pixels

illustrant ce qui n'est pas peau (valeur de pixels réduite à zéro) et, la valeur des autres pixels était montrée. Puis dans un second temps, effectué l'inverse de l'opération précédente, ce qui veut dire, de mettre à zéro les valeurs des pixels montrant la peau et faire ressortir les pixels non peau. D'une part, les images montrant la peau ont été regroupées dans un dossier « peau » et dans un autre dossier « non\_peau » pour celles cachant la peau. Cependant, les images originales, elles, sont regroupées dans un autre dossier « test », tous dans le répertoire « base ».

#### ❖ Construction des histogrammes

Pour construction les histogrammes de peau et non peau, On converti les images de notre base d'apprentissage de l'espace RGB en espace de couleur Lab.

Par la suite on a fait une réduction de valeurs de pixels de  $256 \times 256 \times 256$  à échelle x échelle Rappelons-nous qu'échelle est la première valeur à fournir comme argument au programme. Alors que dans notre cas l'échelle qui est demandée est 32, donc nous passons de  $256 \times 256 \times 256$  à  $32 \times 32 \times 32$ . N.B qu'en associant toutes les valeurs de pixels comprises entre deux multiples du facteur «  $256/\text{échelle}$  » à la valeur du plus petit des multiples considérés.

Ensuite la réduction, le comptage du nombre d'occurrences de chaque couple de valeur (a,b) , dans le répertoire d'image considéré. C'est ainsi que nous avons obtenu un tableau contenant le nombre d'occurrences des différents couples et ce tableau n'est ni plus, ni moins que l'histogramme dont on souligne.

En fait, après que l'histogramme soit finalement exécuté dans notre programme nous l'appliquons un lissage, histoire de porter certaines améliorations à la qualité de la détection. Sans oublier que le lissage consiste à calculer pour chaque pixel, l, les valeurs d'intensité sur son voisinage. Enfin, une dernière étape à notre processus de construction des histogrammes est de normaliser ces valeurs. Une normalisation qui consiste à diviser la valeur de chaque case de l'histogramme par le nombre total des pixels de celle-ci.

Maintenant, pour afficher nos histogrammes, nous reconsidérons l'échelle  $256 \times 256 \times 256$  juste pour garantir une meilleure visualisation, notons que l'on procède inversement à la façon de procéder pour la réduction de l'espace des valeurs. Ci-dessous, deux figures illustrant deux histogrammes (« peau » et « non\_peau ») de notre base de données image.

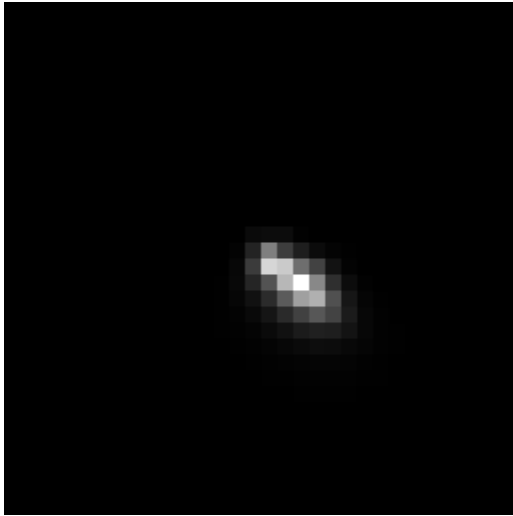


Figure 1 : Histogramme « peau »

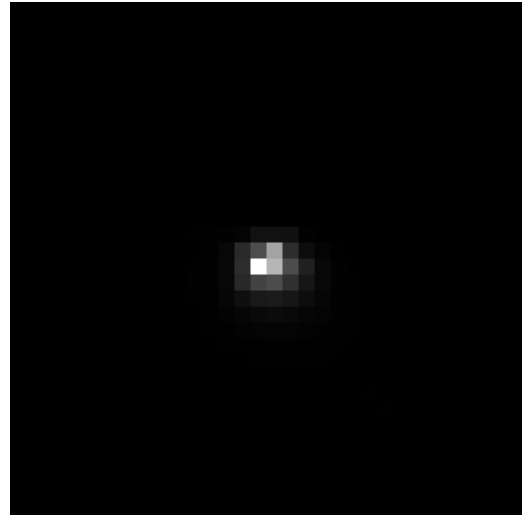


Figure 2 :Histogramme « non\_peau »

## Détection de la peau dans les images

A ce stade, nous utilisons des histogrammes que nous avons construits auparavant pour détecter tous les pixels de peau dans l' image que l'on passe en paramètre à l'exécution du programme. Pour ce, notre modèle charge l'image fournie en paramètre dans l'espace RGB et la convertit dans l'espace Lab. Cette conversion nous permet d'effectuer une sorte de réduction de l'espace de valeurs de pixels comme on a fait lors des histogrammes. A cet instant la phase detection commence et pour chaque couple(a, b) d'un pixel produit une valeur identique à ce même couple dans les histogrammes peau et non-peau.

Maintenant, en utilisant la formule de Bayes nous allons calculer la probabilité pour qu'un pixel soit considéré comme un pixel peau. Puis nous faisons une comparaison entre cette valeur obtenue par bayes et la valeur fournie en paramètre lors de l'exécution. Si la valeur de la probabilité de Bayes est supérieure au seuil c'est qu'on a un pixel de peau autrement c'est un pixel de non-peau. C'est à ce moment qu'il est généré un masque dont les pixels « peau » ont tous la valeur 255 et ceux de « non\_peau » celle de 0. Ainsi nous pouvons restituer l'image détectée, grâce au masque trouvé.

## Traitement apres détection

Pour améliorer les images issues de la détection, nous faisons une dilatation puis une érosion.

## Evaluation

En guise d'évaluation de la performance de notre programme, nous utilisons la formule de contours corrects ou dans notre cas nbre\_pixels\_corrects qui le suivant :

$$P = \text{contours\_corrects} / (\text{contours\_corrects} + \text{faux\_positifs} + \text{faux\_negatifs})$$

$$TFP = \text{faux\_positifs} / (\text{contours\_corrects} + \text{faux\_positifs} + \text{faux\_negatifs})$$

$$TFN = \text{faux\_negatifs} / (\text{contours\_corrects} + \text{faux\_positifs} + \text{faux\_negatifs})$$

Où : P = Performance ; TFP = Taux de Faux Positifs ; TFN = Taux de Faux Négatifs

Ce qui équivaut à :

$$P = \text{nbre\_pixels\_corrects} /$$

$$(\text{nbre\_pixels\_corrects} + \text{nbre\_pixels\_faux\_pos} + \text{nbre\_pixels\_faux\_neg})$$

Nous vous rappelons que nous utilisons des images qu'on ne peut pas trop fier donc il se peut que nos résultats ne soient pas trop excellents.

## Expérimentation et Résultats

Nous vous présentons dans cette partie quelques screenshots découlant de nos propres textes. A noter qu'on a trois points différents traités ici, les résultats des tests dans la base de données d'apprentissage, résultats des tests avec différents seuils sur les images et l'analyse.

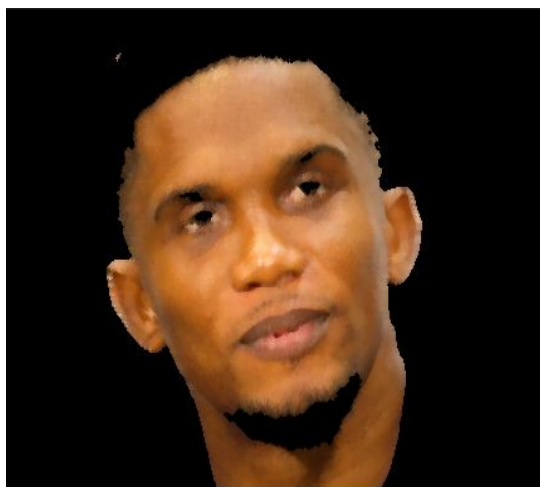
•



seuil 0.1 performance 35.1172 %



seuil 0.2 performance 77.6196 %



seuil 0.3 performance 82.5065 %

seuil 0.4 performance 82.484 %

Ces quatre images traduisent de la manière la plus simple la relation qui existe entre le seuil et la qualité de l'image de sortie, cette dernière qui est l'image détectée construite à partir du masque. En performance, nous atteignons jusqu'à 82.5065% pour un seuil de 0,3, On remarque sur la première image que plus le seuil est bas plus l'image est bonne pour les yeux humains mais donne une très mauvaise performance soit de 35,1172%.

- Détection de la peau dans le cas des tests avec les images de la base :



seuil 0.3 performance 67.1731 %



seuil 0.3 image de sortir

référence :43568

correct :35063

faux\_positif :8630

faux\_negatif :8505

Performance du programme = 67.1731 %



déTECTÉE seuil 04 performance 80.114 % image de référence dans la base

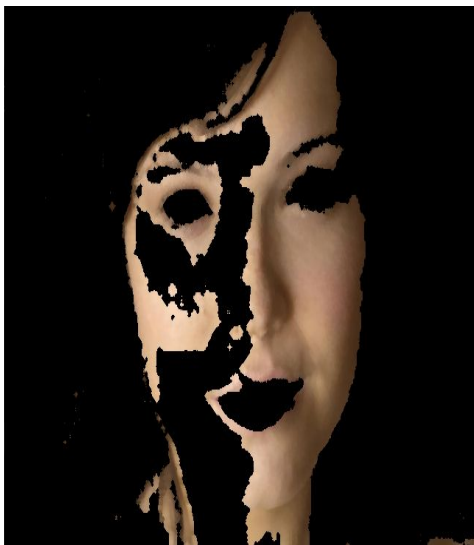
Référence = 51.285

Correct = 50.078

Faux\_positif = 40.593

Faux\_negatif = 12.07

Performance du programme = 54.5049%



déTECTÉE seuil 0.5 performance 44.5165 % image de référence 44.5165 %

référence : 11.4843

correct : 57.230

faux\_positif : 13.716

faux\_negatif : 57.613

Performance du programme = 44.5165 %



détectée performance 53.4895 %



image de référence 53.4895 %

référence :52857

correct :30136

faux\_positif :3483

faux\_negatif :22721

Performance du programme = 53.4895 %

## Conclusion

Pour conclure, . Nous avons développé un modèle utilisant une base d'images élaborée par nous même à partir des images prises sur internet. On souligne que pas toutes nos images étaient de bonne qualité mais nous les avons améliorées par des opérations de dilatation et d'érosion. Enfin notre modèle n'est pas le meilleur car il reste encore des travaux a faire pour améliorer afin d'éviter tout éventuel biais possible de subvenir lors de l'exécution.

Ce travail a été un moyen de pratiquer certaines notions apprises lors du cours de traitement d'images ajoutées à celles nouvellement acquises en Vision par ordinateur.