



INSTITUT FRANCOPHONE INTERNATIONAL (IFI)

**Rapport TP1 Détection Peau
Vision par Ordinateur**

**Rédigé par :
AZARIA SAIDI ALLY**

Promotion 22 Hanoï 2018

Introduction

Ce projet a pour but premier de nous immerger dans le concept de vision par ordinateur en implémentant un programme fait en c++ qui permettra de détecter la peau humaine dans une image. Et pour cela nous avons besoin de constituer une base de connaissance qui contient trois type d'images, à savoir des images des personnes qui constituent nos exemples de base, des images de masque de peaux (de peaux tout simplement) et enfin des images des non peaux.

Dans ce document nous mettrons en exergue l'ensemble des tâches exécutés à travers des exemples illustratifs, suggestions ainsi que les analyses et futur amélioration non exhaustifs de ce programme .

Composition du programme

Notre programme comme mentionné dans l'introduction est composé essentiellement des trois type d'images à savoir les images source qui sont tirés du web, les images peaux qui sont construite à partir des images sources, qui sont en effet des masques de peaux et enfin les images non peaux toutes situés dans des répertoires différents .

dans un premier temps nous commencerons à construire des histogramme , premièrement des peaux et ensuite des images non-peaux, tout ceci en convertissant l'espace de couleur RGB en un espace de couleur LAB. pour cela nous exécutons notre programme de la manière suivante:

`./detect_peau 34 0.4 4.jpg`

Explication des paramètres

le premier paramètre lorsqu'on appelle le programme représente d'abord le programme lui même (le nom du programme), le second paramètre représente l'échelle de réduction de l'espace de couleur RGB soit de 256 à 34, le troisième paramètre représente le seuil pour les histogrammes à construire et enfin le dernier paramètre représente le nom de l'image source que nous appelons

Histogrammes

Nous avons deux histogrammes dans notre programme à savoir l'histogramme peaux et l'histogramme non-peaux , construit sur un espace de couleur RGB réduit à 34 valeurs (34 étant un paramètre que nous pouvons modifier selon notre convenance et il représente aussi une échelle de réduction de l'espace de couleur RGB de 256 à 34)

le calcul de ces deux histogrammes se fait en boucle , c'est à dire en parcourant toutes les images de la base pour mettre à jour les histogrammes.

une fois le programme lancé , nous avons l'affichage des histogrammes suivant :

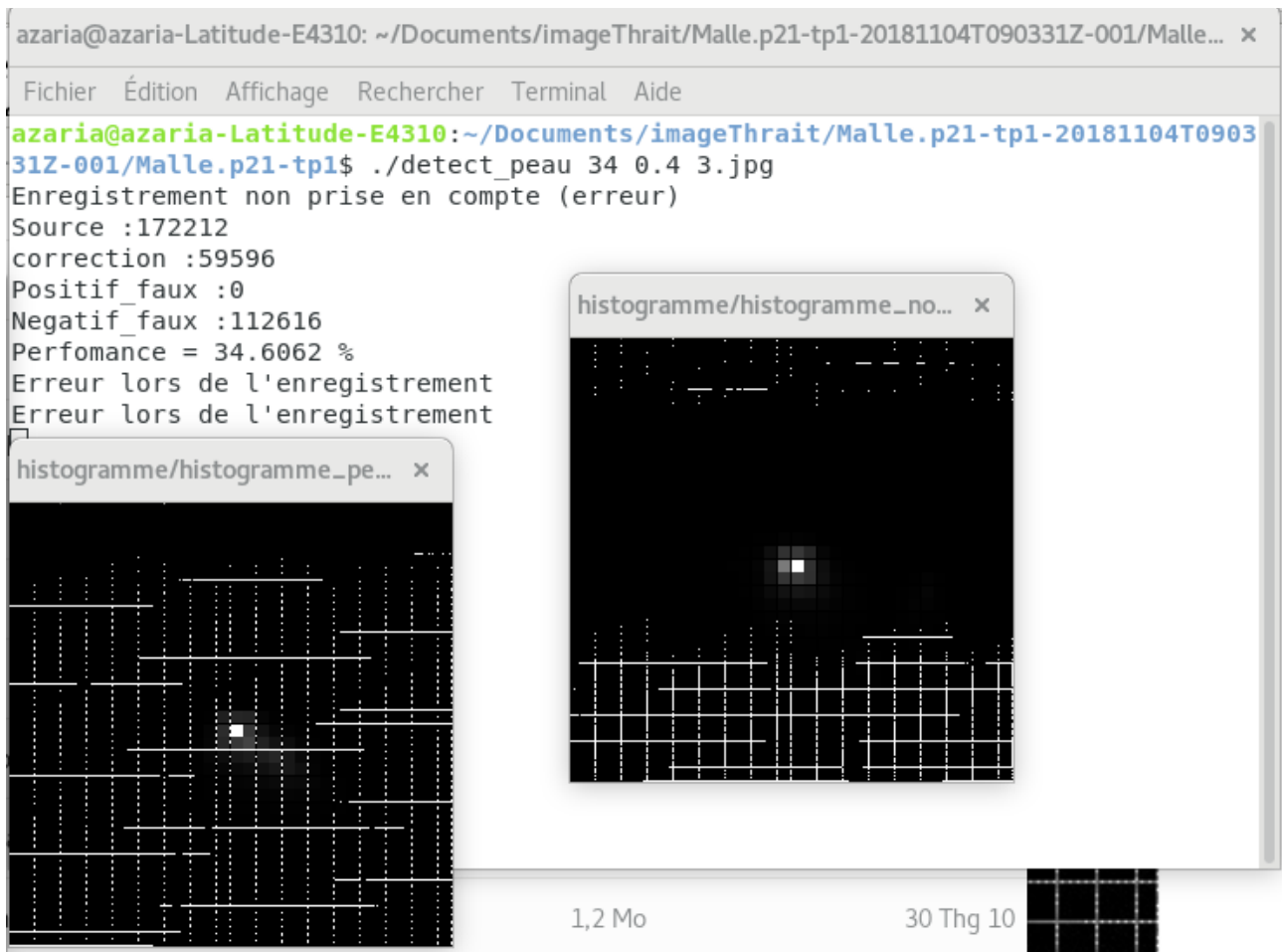


Illustration 1: histogramme peaux et non-peaux

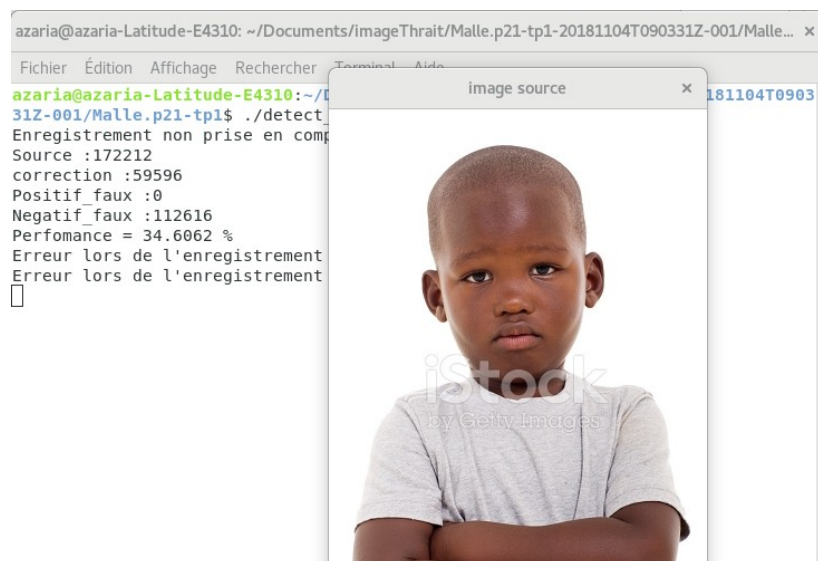


Illustration 2: Image Test

Étape de détection de la peau : cette étape consistera à prendre en compte les deux histogrammes de l'illustration1 afin de détecter les pixels peau dans l'image de test ci-dessus.

Après conversion de l'image en entrée de l'espace de couleur RGB vers l'espace de couleur Lab nous effectuons une réduction de l'espace des valeurs de pixels comme dans le cas de l'histogramme. Cette partie consiste à sélectionner chaque couple (a,b) d'un pixel de l'image d'entrée ainsi que celui correspondant dans les histogrammes ci-dessus.

Voici la formule utilisée pour la détection de la peau :

$$p(\text{peau} \mid a,b) = \frac{p(c \mid \text{peau}) p(\text{peau})}{p(c \mid \text{peau}) p(\text{peau}) + p(c \mid \neg \text{peau}) p(\neg \text{peau})}$$

Illustration 3: formule de détection de la peau

Post-traitement :sert pour l'amélioration lors de la détection de la peau; les nombres d'imperfections sont prises dans l'image concernée en entrée lors d'une application procédurale

successive de dilatation à l'image détectée.

Évaluation : cette étape nous permettra de connaître les performances d'efficacité de notre

programme.

résultats et analyse

nous effectuerons une série de plusieurs tests sur une même image avec des valeurs de seuil

différentes .

Notre test commencera avec l'image 3.jpg avec une variation du seuil comme le montre l'image ci-dessous :

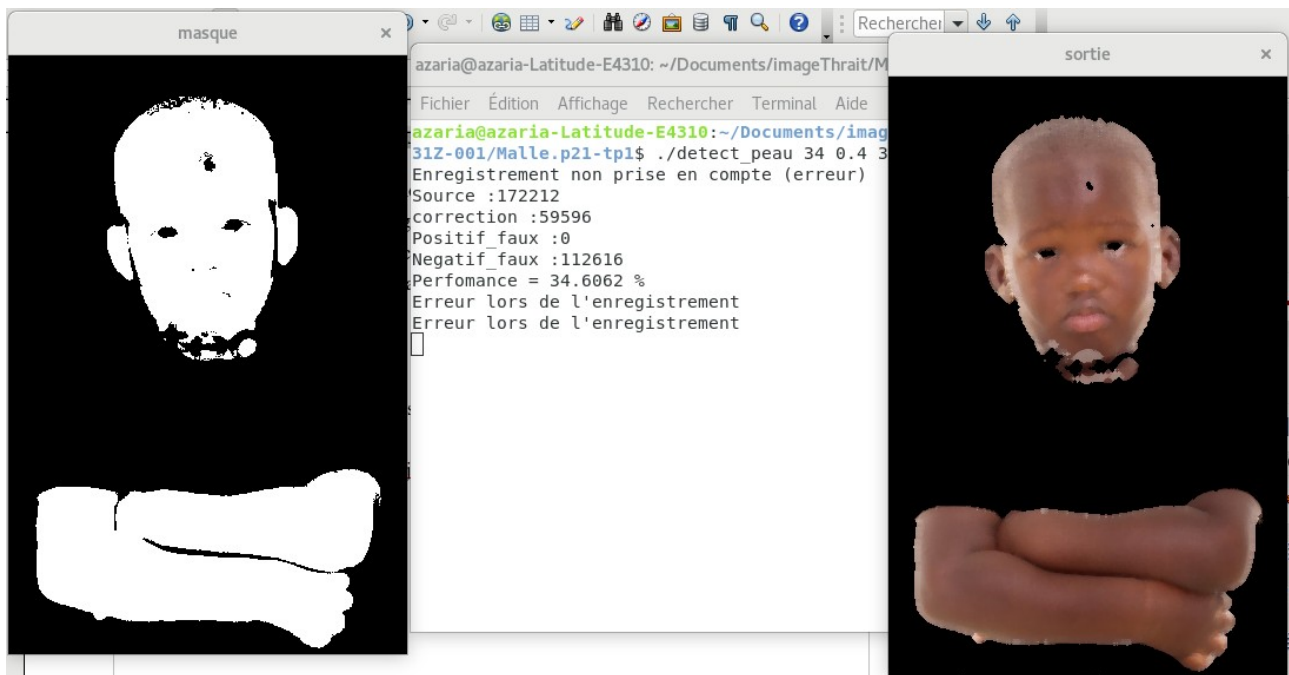


Illustration 4: test

Vous pouvez remarquer qu'avec un seuil de 0.4 nous avons une performances 34%

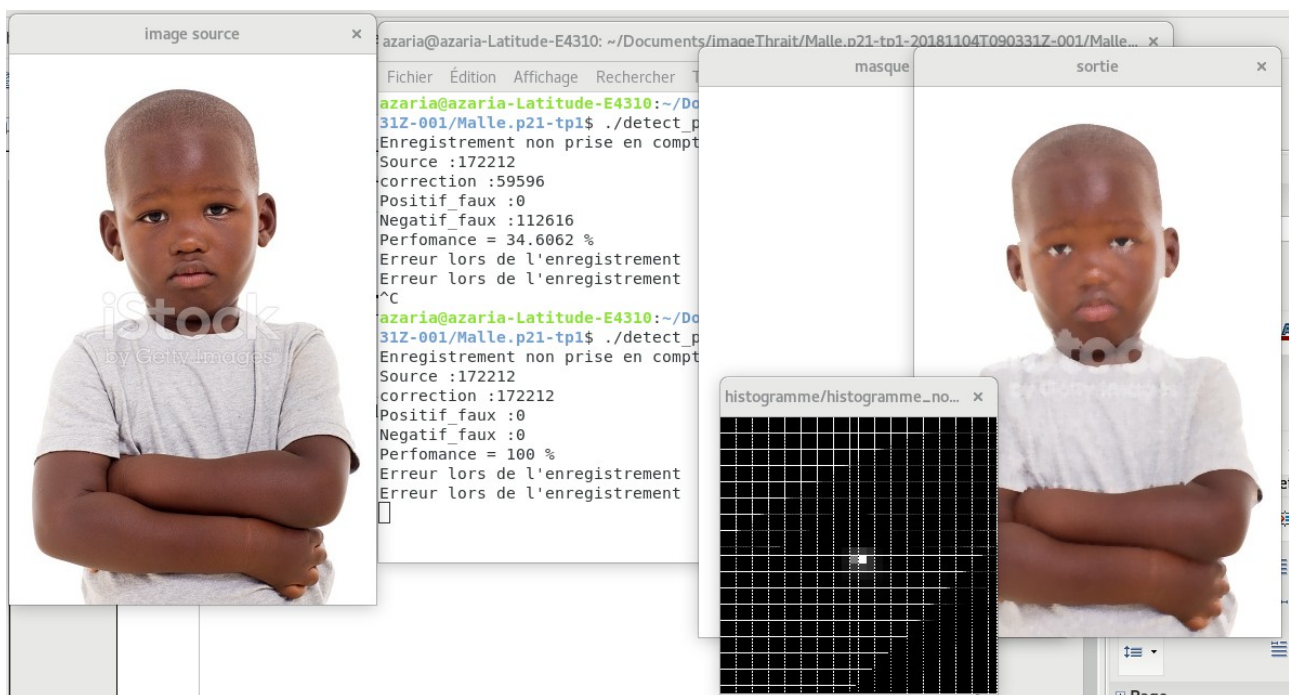


Illustration 5: test2

dans cette illustration nous mettons le seuil à 0.0, ce qui a pour conséquence d'une performance de 100%

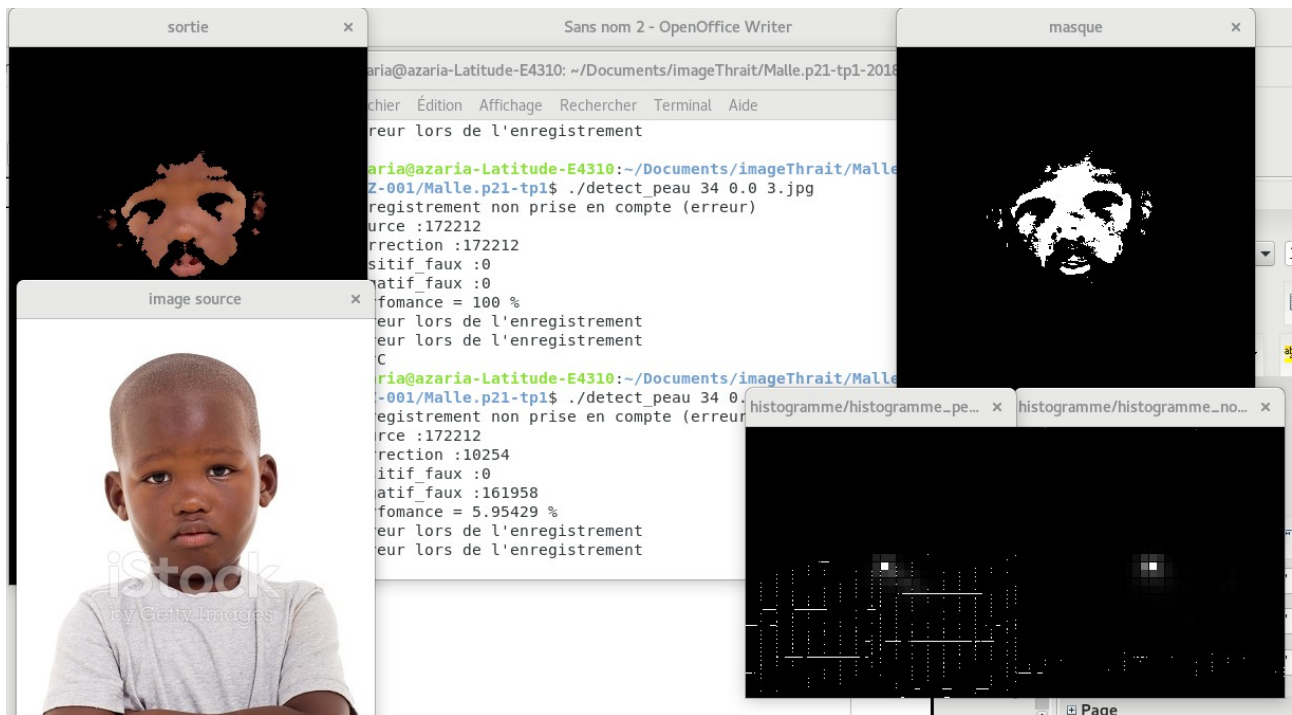


Illustration 6: test3

dans l'image de test ci-haut nous avons pris un seuil de 0.6 ce qui a entraîné une performance de 5.95%

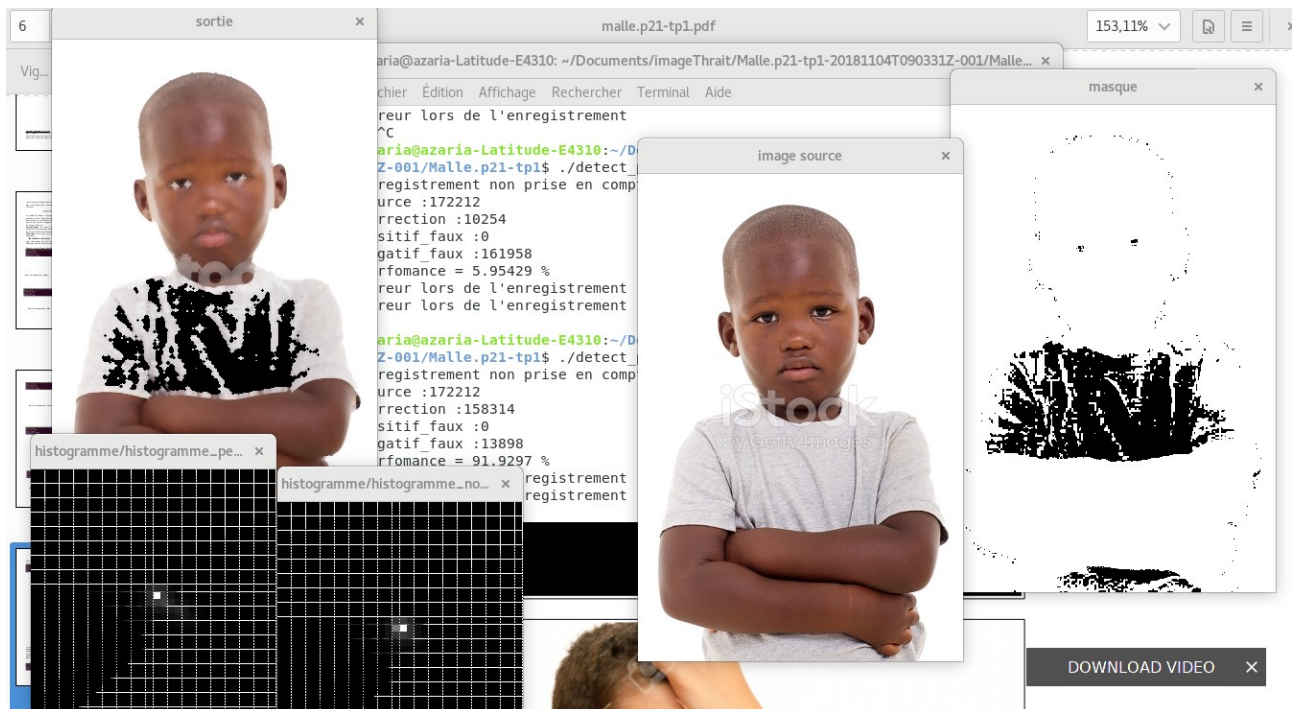


Illustration 7: test4

ici nous avons pris un seuil de 0.3 et avons eu une performance de 91%
maintenant nous allons faire un dernier test avec une autre image et afin conclure

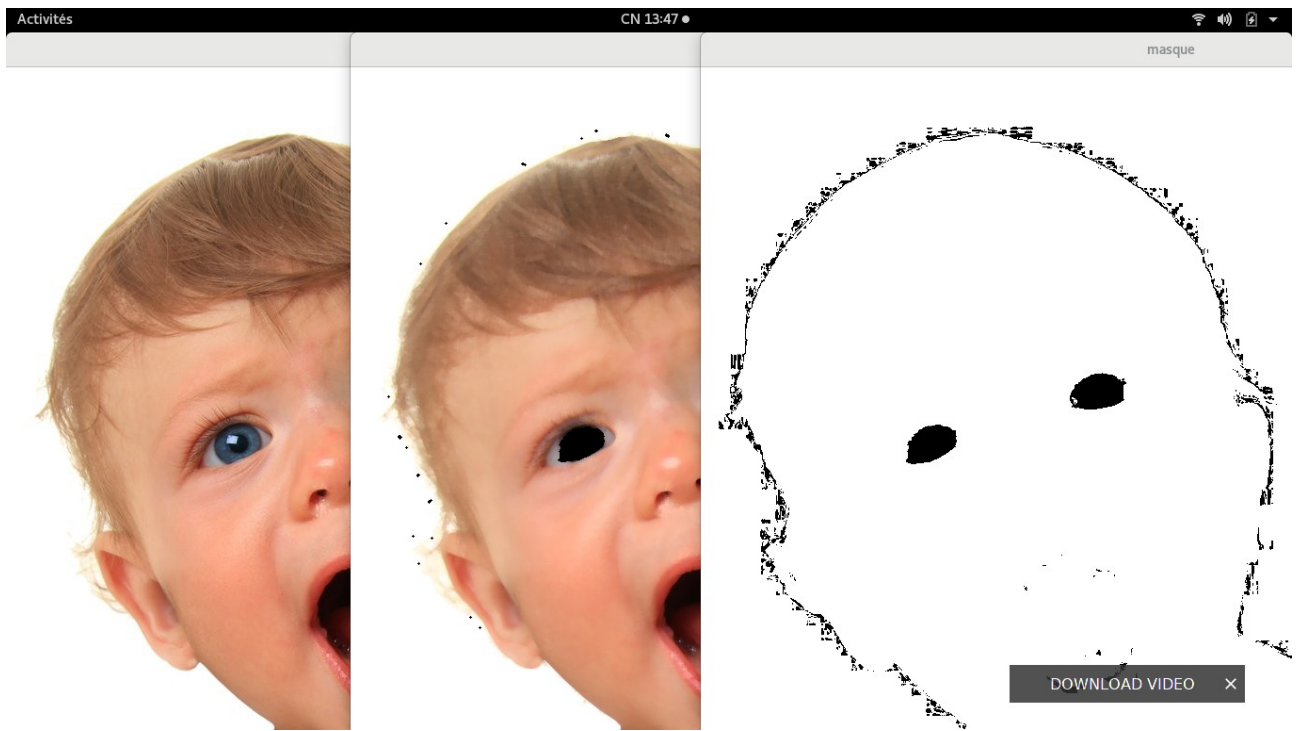


Illustration 8: test5

ici nous avons pris un seuil de 0.3 et avons eu une performance de 99%

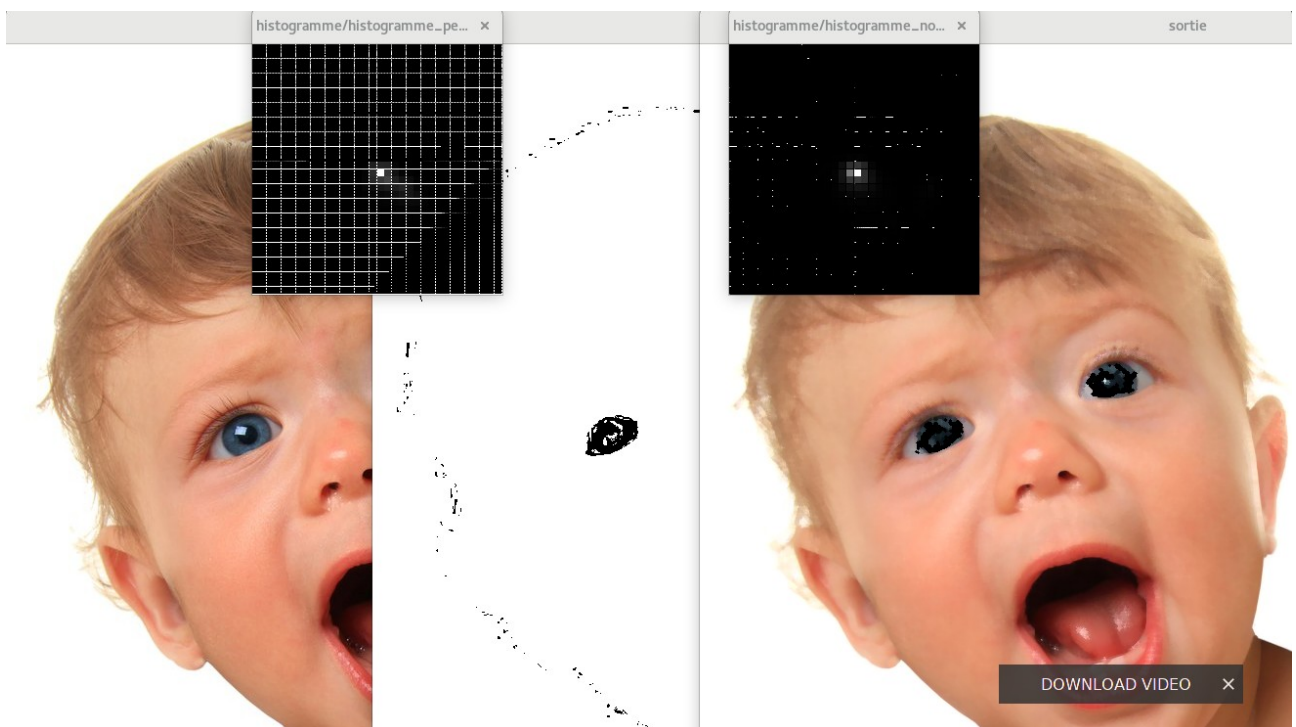


Illustration 9: test7

avec un seuil de 0.2 nous obtenons une performance de 99%, nous constatons qu'ici le résultat vraiment probant et cela avec le seuil 0.2 et 0.3



Illustration 10: test8



Illustration 11: test 9

avec les images ci-haut (illustrations 8 et 9) , nous remarquons que le programme détecte bien la peau de couleur noire , ce qui justifie la faible performance notée ,cela est du à la détection des pixels non-peau comme le t-shirt, tatouage sur les bras. Le programme détecte aussi les cheveux , la chaîne de collier comme pixels peau. Ces détections erronées peuvent se justifier par la faible présence d'éléments de couleur rouge dans la base d'apprentissage.

Conclusion

Lors de notre premier test sur l'image 3.jpg nous avons constater une nette amélioration de la détection de la peau avec le post-traitement (dilatation et l'érosion) en conséquence qui entraîne aussi une perte de niveau des pixels de la peau, des information considérable sur les images moins nettement détectée. Pour conclure nous pensons que ce programme pourrais être améliorer pour mieux apprendre la détection de la peau et il n'est pas exclu l'utilisation d'autres techniques de conversion et de détection d'images en vue d'accroître les performances de détection.