

TP 2 : Détection et Reconnaissance de Visages

**Groupe : BIAKOTA BOMBIA Cephass Herbert
Milorme Pierre Rubens**

Ce travail a pour objectif de mettre en place un programme de détection puis de reconnaissance de visages dans une image requête fournie en entrée au système. Dans ce présent rapport nous résumons nos travaux et expérimentations en deux grandes parties subdivisés. La première partie traite uniquement la question de détection et la deuxième partie uniquement la reconnaissance.

1- Détection de visages

1-1- Description et fonctionnement du programme

Le programme implémenté, prend en entrée une image et avant tout traitement il la redimensionne. Les nouvelles tailles possibles sont 220*250 pour les images ne contenant que la tête d'un individu et 1200*1000 pour celles contenant plusieurs personnes. Ces tailles ont été retenues après plusieurs expérimentations. Nous utilisons le ratio largeur/hauteur pour identifier chaque type d'image. Pour une valeur du ratio supérieure à 1 on considère qu'il s'agit d'une image contenant plusieurs personnes dans le cas contraire l'image ne contient qu'un individu. Nos images sont donc choisies en conséquence. Après le redimensionnement, le programme applique le détecteur de visages de Viola et Jones, déjà présent dans opencv, à partir du modèle contenu dans le fichier « haarcascade_frontalface_alt.xml ». Nous retenons ce fichier car il offre les meilleurs résultats de détection. De plus la majorité de nos images contiennent des visages de face. Une fois le détecteur appliqué il est possible que l'on ait plusieurs zones considérées comme visage mais qui ne le sont pas toutes.

Ainsi pour discriminer les zones étant réellement visage de celles qui ne le sont pas, nous leur appliquons un détecteur d'yeux et un détecteur de nez qui utilisent aussi l'algorithme de Viola et Jones à partir des fichiers « haarcascade_eye_tree_eyeglasses.xml » et « haarcascade_mcs_nose.xml ». Dès lors nous ne considérons comme visage que les zones où sont au moins détecté un oeil et/ou un nez. Toutefois pour une efficacité optimale de notre méthode, il est également nécessaire de jouer sur les paramètres d'échelle, de nombre de voisins, de taille minimale et maximale des objets détectés. A la fin du processus de détection, nous normalisons l'histogramme de l'image obtenue afin de faciliter les opérations de reconnaissance.

Ainsi en résumé, Le programme que nous avons implémenté pour pouvoir faire la détection des visages dans les images prend seulement en entrée le chemin vers l'image à traiter. Notre programme renvoie l'image en entrée les visages encadrés (voire les rectangles colorés).

Pour pouvoir utiliser le programme, il suffit de se positionner dans le répertoire contenant le code source du programme depuis le terminal puis de lancer la commande « make ».

Exécution du programme:

pour exécuter le programme, il faut lancer la commande suivante: **./main « chemin_nom_image »**. Exemple : La commande « **./main images/rubens1.jpg** » permet de détecter les visages dans l'image 1.jpg situé dans le répertoire « **images** ».

1.3- Expérimentations

Récapitulatif de nos données de paramètre utilisé :

Paramètres Valeurs

Facteur d'échelle 1,01 ou 1,1

Nombre de voisins 1,2,3

Taille minimale des têtes 30*30

Taille maximale 200*200



Fig1 : image originale

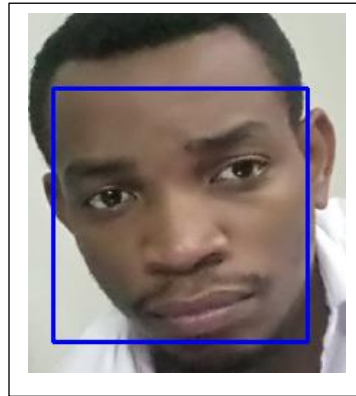


fig1.1 : image avec visage détecté

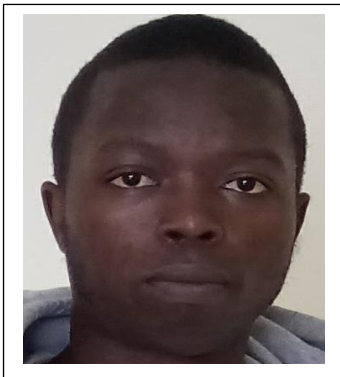


Fig2 : image originale

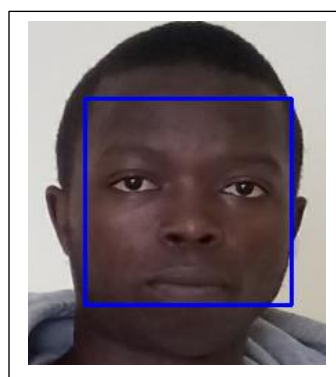


fig2.1 : image avec visage détecté



Fig3 : image originale

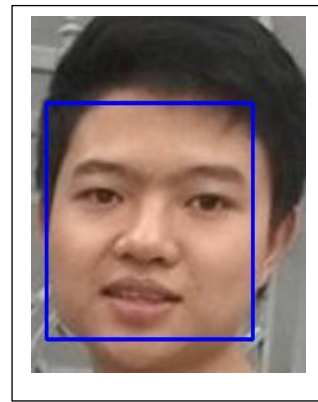


fig3.1 : image avec visage détecté



Fig4 : image originale

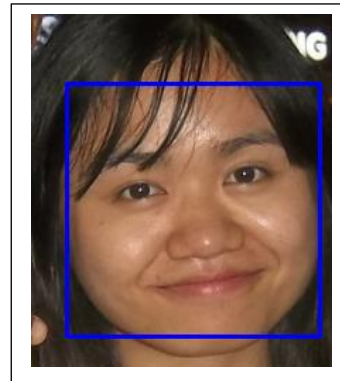


fig4.1 : image avec visage détecté



Fig5.1 : image originale

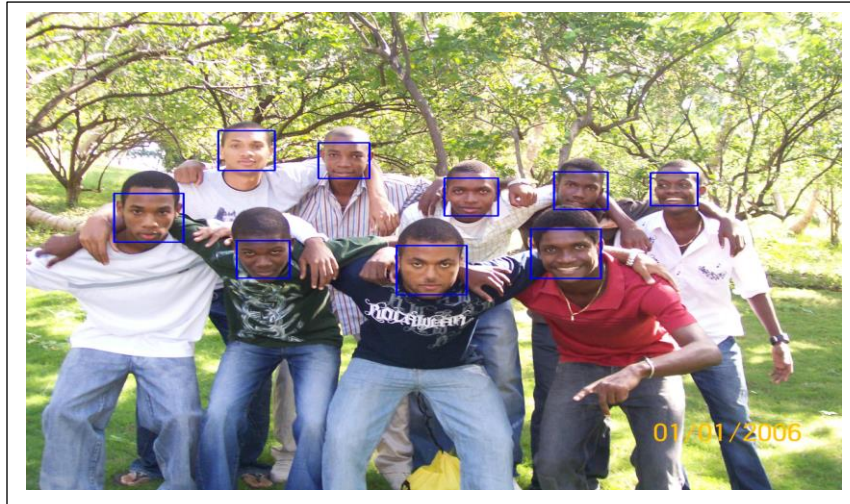


Fig5.1 : image avec visage détecté



Fig6.1 : image originale

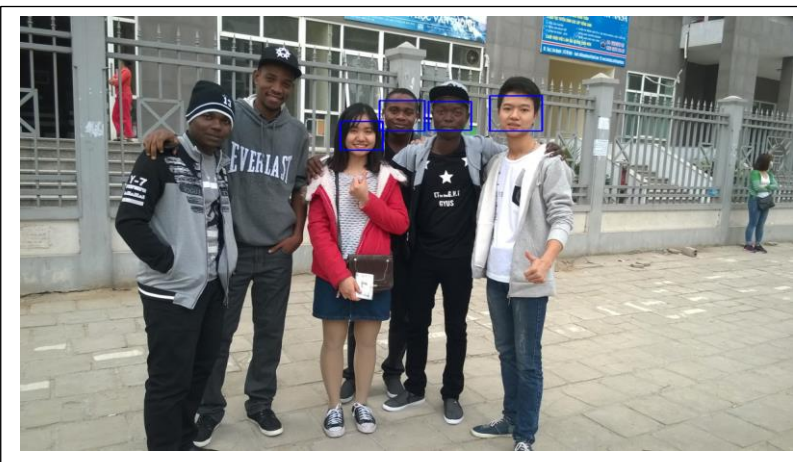


Fig6.1 : image avec visage détecté

Interprétation

À l'issu de cette expérimentation effectuée avec différente image prise dans des différentes position et conditions d'acquisition, Nous avons remarqué que notre programme à bien détecter les visages pour les images contenant uniquement la tête d'une personne (voire fig1à4) ainsi que pour les images contenant plusieurs personne (voir fig5) ce qui explique que le descripteur de visage a bien pris en compte les parties visages tandis que pour la dernière image (voire fig6) la qualité de la détection n'est pas très optimale pour les visages car tous les visages n'ont été pas détectés ce qui explique que les visages non détectés ont été considérés comme étant des faux négatifs c'est-à-dire les partie de visage qui sont considérés comme partie non visage. Ce fait est dû aux conditions d'éclairage et à la position du visage des individus. Pour ce fait nous n'avons pas besoin d'améliorer la détection car la position du visage et les conditions d'acquisition de l'image peuvent influencer les résultats de la détection. Ainsi, dans le cas où notre programme détectait des faux positifs (partie non visage détecté), dans ce cas une amélioration serait possible en utilisant la méthode de discriminant afin de remédier à ce problème

2- Reconnaissance de visages

2-1 Description et fonctionnement du programme

Les 2 programmes implémentés, prennent en entrée, le nom d'un fichier contenant notre base d'apprentissage et le nom d'un fichier contenant notre base de test ou le chemin vers une image test. Nous obtenons en sortie la matrice de confusion ou d'occurrence, le taux de précision de la reconnaissance et les étiquetés ou labels associés à chaque image.

Pour pouvoir utiliser le programme, il est important d'effectuer les actions décrites ci-dessous :

- **Bases d'images** : le programme doit être accompagné de deux répertoires contenant pour l'un les images d'apprentissage et pour l'autre celles de test.
- **Fichiers descriptifs** : ces fichiers qui sont de format «.txt »(apprentissage.txt et test.txt) décrivent le contenu des bases d'apprentissage et de test. Chaque ligne de ces fichiers nous donne le chemin vers une image avec le label de l'image, séparés par un point-virgule « ; ».
- **Compilation du programme** : pour compiler le programme, il suffit de se placer dans le répertoire contenant le code source du programme depuis le terminal puis de lancer la commande « make ».
- **Exécution du programme** : Pour pouvoir utiliser le programme, il suffit de se positionner dans le répertoire contenant le code source du programme depuis le terminal puis de lancer la commande « make ». et ensuite la commande suivante: pour la reconnaissance de visage simple : `./main <fichier_descr_base_apprentissage> <fichier_descr_base_test>` et pour ma reconnaissance multiple : `./main <fichier_descr_base_apprentissage> <chemin_image_test>` .

Exemple : La commande `./main apprentissage.txt test.txt` permettra de détecter les labels de toutes les images contenues dans la base décrite par test.txt en se basant sur les images décrites par apprentissage.txt.

2-2 Description de l'implémentation

- Création des bases d'images

Nos bases d'images (test et apprentissage) ont été créées à partir des images de huit (08) personnes et en prenant quatre (04) images de chaque personne ce qui fait un total de trente-deux (32) images. Après avoir traité les images pour ne conserver que la tête des individus, nous avons réparti équitablement et aléatoirement les images en deux groupes, le premier constituant notre base d'apprentissage et le second notre base de test.

- Détection des visages

A cette étape, notre programme, grâce au module de détection de visage présenté plus haut, détecte les visages dans toutes les images (apprentissage et test), en les redimensionnant à la taille 100*100 et en les normalisant. La normalisation consiste à égaliser l'histogramme des images ce qui permet d'atténuer les effets des changements de conditions d'illumination dans images et d'améliorer le contraste.

2-3 Expérimentation et résultats

Après avoir exécuté notre programme avec une base de test contenant des personnes également présentes dans la base d'apprentissage on obtient les résultats ci-dessous. Notons que nous avons retenu 5 composantes principales les vecteurs propres. Rappelons que nous avons utilisé le programme de reconnaissance des visages simples.

```
Image de perrault : Classe prédite = 2 / Classe réelle = 2.
Image de perrault : Classe prédite = 1 / Classe réelle = 2.
Image de rubens : Classe prédite = 3 / Classe réelle = 3.
Image de rubens : Classe prédite = 3 / Classe réelle = 3.
Image de hu : Classe prédite = 4 / Classe réelle = 4.
Image de hu : Classe prédite = 0 / Classe réelle = 4.
Image de thao : Classe prédite = 5 / Classe réelle = 5.
Image de thao : Classe prédite = 1 / Classe réelle = 5.
Image de jb : Classe prédite = 6 / Classe réelle = 6.
Image de jb : Classe prédite = 6 / Classe réelle = 6.
Image de zopoul : Classe prédite = 1 / Classe réelle = 7.
Image de zopoul : Classe prédite = 1 / Classe réelle = 7.

Matrice de confusion
0 1 0 0 1 0 0 0
0 1 1 0 0 0 0 0
0 1 1 0 0 0 0 0
0 0 0 2 0 0 0 0
1 0 0 0 1 0 0 0
0 1 0 0 0 1 0 0
0 0 0 0 0 0 2 0
0 2 0 0 0 0 0 0

Le taux de précision est :50%
```

Fig 7 : Résultats reconnaissance d'image, matrice de confusion et taux de précision

Les résultats sont structurés en trois parties. La première nous montre pour chaque image la classe qui a été prédite et celle originale en plus du nom réel de l'individu dont l'image a été utilisée. La seconde partie présente la matrice de confusion dans laquelle une case $M(i,j)$ indique le nombre d'images de la classe i qui ont été reconnues par le programme comme étant de la classe j . $0 \leq (i, j) \leq 7$

Enfin le taux de précision indique la performance du programme.

D'après nos résultats notre programme reconnaît moyennement bien les personnes dans les images. Ce fait s'explique par un taux de précision 50%

Les figures ci-dessous nous montrent les résultats sur les photos.



Img1:jb

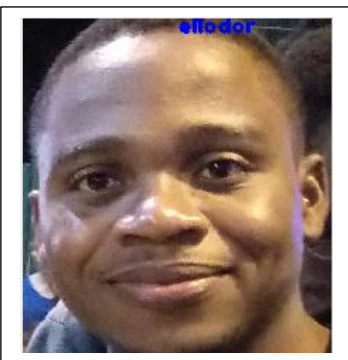


Img2 :thao



Img3:rubens

Figure 13 : Résultats montrant des images de personnes bien reconnues



Img4 : perrault



Img5 :cephas

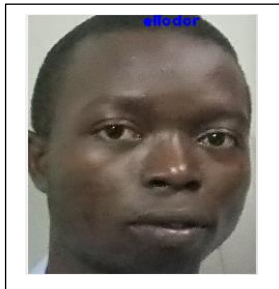


Img6:thu

Fig8 : Résultats montrant des images de personnes mal reconnues

Après l'analyse de nos résultats, nous remarquons que les personnes à peau claire sont globalement mieux reconnues que celles à peau sombre. Ce qui peut s'expliquer par le fait les images de personnes à peau claire sont plus nombreuses que celles des personnes à peau sombre. Ainsi lors du ACP les caractéristiques de ces personnes ne sont pas suffisamment prises en compte. Les problèmes de reconnaissance peuvent aussi s'expliquer par l'orientation des visages car suivant celle-ci la détection du visage d'une même personne peut différer fortement et donc affecter la reconnaissance car les personnes ayant une même orientation de la tête ont tendance à être confondues par le programme. De plus les particularités de certains visages comme la barbe et les lunettes sont aussi des facteurs d'échec dans la reconnaissance car ils n'apparaissent pas souvent dans les images de la base d'apprentissage. Un autre facteur est le nombre de composantes de l'espace ACP (vecteurs propres) considérées lors de la projection des images car chaque composante de l'espace ACP est non corrélée par rapport aux autres et donc apporte ses propres informations qui peuvent s'avérer indispensables à la reconnaissance. Outre les facteurs précédemment cités et qui affectent la reconnaissance des visages, nous devons indiquer que l'algorithme de classification 1NN choisi, est aussi une source de mauvaise classification car il n'est pas très optimal. En effet il se base sur la distance euclidienne pour classer hors cette mesure est une importante source d'erreur car elle ne prend pas en compte la variance. Ainsi pour une meilleure

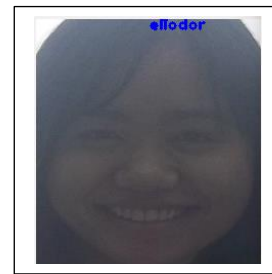
classification, ainsi nous préconisons l'utilisation de la **distance de Mahalanobis** ou alors l'utilisation **des réseaux de neurones ou des réseaux bayésiens** qui offrent de meilleurs résultats. Afin de mieux illustrer l'effet des composantes principales sur la reconnaissance nous exécutons à nouveau le programme en considérant 3 composantes principales. Les résultats obtenus sont alors :



Img4 : cephas



Img5 :perrault



Img6:thao

Fig9 : Résultats montrant des images de personnes mal reconnues (avec 3 composantes principales)

Nous notons qu'avec trois composantes principales, l'image img4 n'est pas reconnue comme **cephas** mais plutôt comme **eliador**. De plus le taux de précision est à 50%. Dans nos expérimentations nous avons aussi essayé de reconnaître les individus inconnus dans la base d'apprentissage. Pour ce faire nous nous sommes basés sur les distances calculées lors de l'application de la classification 1NN. L'idée développée consiste à fixer un seuil tel que si la plus petite des distances calculées est supérieure à ce seuil alors l'image est considérée comme appartenant à un inconnu. Toutefois, nos essais n'ont pas été fructueux car l'image était chaque fois associée à une personne de la base. Nous avons également essayé avec la distance de Mahalanobis mais sans plus de succès. En effet cela est dû au fait que une fois projeter dans l'espace ACP l'image perd un grand nombre de ses caractéristiques et est plus proche des images ayant servi à constituer la base d'apprentissage. Ce problème a aussi été relevé **M.Laforest, I. Jalliffier-Verne** de l'école polytechnique de Montréal dans leur document intitulé Application des mathématiques : reconnaissance de visages.

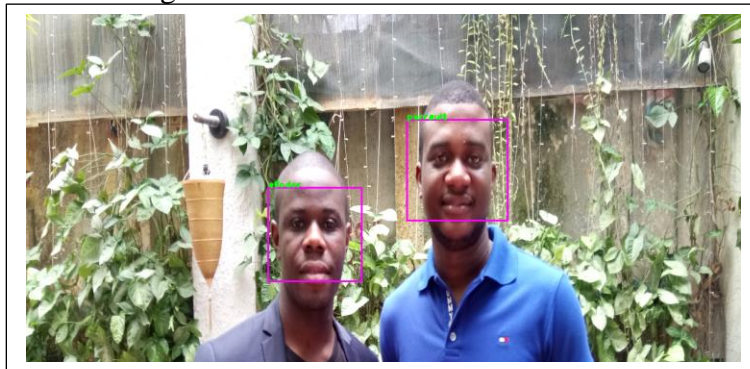


Fig10 : Résultats de la reconnaissance de personnes inconnues

A présent nous montrons les résultats des expérimentations sur les images contenant plusieurs personnes.

Cette expérimentation est réalisée grâce au programme **de la reconnaissance des visages multiples**. Les résultats obtenus présentent une mauvaise reconnaissance des visages par le programme. Les raisons sont d'abord la présence de plusieurs personnes inconnues, puis la position des visages dans l'image de test qui diffère souvent assez de celle des images de la base d'apprentissage. Toutefois certains visages au même moment sont bien reconnus car appartenant à la base d'apprentissage et orientées pratiquement de la même façon.

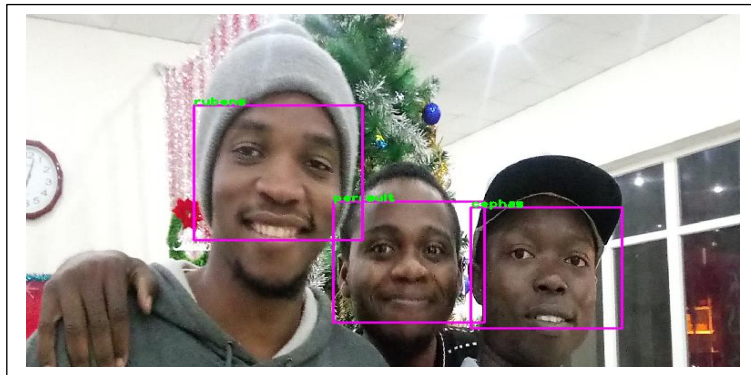


photo de : milorme, perrault et cephas

Fig11 : Résultats de la reconnaissance de plusieurs personnes dans une image

Conclusion

A l'issue de ce travail très instructif, nous avons pu nous familiariser et d'acquérir des connaissances et les compétences majeures qui sont la détection et la reconnaissance de visage. Les différentes expérimentations nous ont fait comprendre que, souvent il faut associer de plusieurs caractéristiques pour obtenir une bonne détection. Ainsi, pour la reconnaissance nous avons pu remarquer que l'ACP permet d'accélérer la comparaison des images en réduisant leur dimension mais que l'algorithme du plus proche voisin n'est pas très adapté à la tâche de classification des nouvelles images. Nous avons aussi essayé d'identifier les individus inconnus en nous basant sur la distance euclidienne et de Mahalanobis mais sans grand succès. En somme nous pouvons retenir de ces expérimentations que la reconnaissance serait mieux si l'on augmente la taille de nos bases d'image (apprentissage) tout en utilisant des algorithmes de classification les plus appropriés

Références

http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html, consulté le 20/05/2017

<http://www.genie-des-maths.polymtl.ca/exemples/ex-00090/ex11.pdf>, consulté le 20/05/2017