Sommaire

Table des figures iv
Liste des tableaux v

Introduction générale 1
CHAPITRE 1 : La reconnaissance faciale 2
1.1Introduction 2
1.2 La reconnaissance faciale 2
1.3 Domaines de la Reconnaissance Faciale 3
1.4 Historique 4
1.5 Les techniques de détection et de reconnaissance faciale 4
1.6 Problématique 5
1.7 Conclusion 7
Chapitre 2 spécifications et conceptions 8
2.1 introduction 8
2.2 Spécification de la technique et langage utilisés 8
2.2.1 Techniques de détection utilisées 9
2.2.1 Choix du langage de programmation 16
2.2.2 Les diagrammes de cas d'utilisation 17
2.3 Conception 18
2.3.1 Conception global 18
2.3.2 Conception détaillée 20
2.3.2.1 Diagramme de séquence « identification » 20
2.4 Conclusion 22
Chapitre 3 : Réalisation 23
3.1 Introduction 23
3.2 Environnement du Travail 23
3.3 Présentation 23
3.3.1 Détection de visage : 23
3.3.3 Reconnaissance faciale : 26
3.4 Détection de visage par open cv 28
3.4.1 Introduction : 28
3.4.2 Les étapes pour que OpenCV détecte un visage : 29
3.5 Prétraitement pour la reconnaissance faciale fisher (lda) 31

3.6 Conclusion 33
Conclusion Générale et perspectives 34

Table des figures

Figure 1 : La procédure de DeepFace pour la reconnaissance.

Figure 2 : Changement d'angle (poses).

Figure 3 : Changement d'éclairage.

Figure 4 : Changement des expressions faciales.

Figure 5 : Schéma général d'un système de reconnaissance des visages.

Figure 6 : Caractéristiques de Haar.

Figure 7 : Les propriétés sélectionnées par Haar.

Figure 8 : Attribution d'une étiquette à chaque pixel d'une image.

Figure 9 : LBP Différents exemples de points d'échantillonnage et de rayons.

Figure 10 : résultat de l'application du LBP sur image a.

Figure 11 : Simple image « face ».

Figure 12 : Application du LBP sur figure 11.

Figure 13 : histogramme de figure 3.

Figure 14 : Diagramme de cas d'utilisation pour la détection.

Figure 15 : Diagramme de cas d'utilisation pour la reconnaissance.

Figure 16 : Diagramme de séquence « identification ».

Figure 17 : Diagramme de séquence « reconnaissance ».

Figure 18 : La zone de choix de classifier.

Figure 19 : Menu principale

Figure 20 : Capture de l'application en fonctionnement.

Figure 21: Capture sur interface Heidi SQL.

Figure 22 : Capture sur les dossiers crée par l'application

Figure 23 : Reconnaissance échouée.

Figure 24 : L'ajout d'un utilisateur

Figure 25 : Reconnaissance réussie.

Figure 26 Transformation vers niveau de gris

Figure 27 : Comparaison de 2 histogrammes

Figure 28 : Chargement de fichier XML pour la sélection de Haar

Figure 29: Chargement de fichier XML pour la sélection de LBP

Figure 30 : Lancement de la détection

Figure 31 : Traçage de triangle

Figure 32 : Procédure de découpage d'un visage

Figure 33 : Procédure de décimation d'un visage

Liste des tableaux

Tableau 1: Resultat par detection Haar

Tableau 2: Resultat par detection LBP

Tableau 3: Tableau comparative entre langage de programmation

Introduction générale

La reconnaissance de visages est une technologie biométrique en vogue, elle est très utilisée dans les applications d'authentification, de contrôle d'accès et de vidéo de surveillance, on trouve plusieurs méthodes globales, locales et hybrides de reconnaissance de visages.

Ces derniers temps et avec l'avancement de la technologie, les mots de passe et les clés qu'on utilise dans tous les domaines de sécurité et de contrôle d'accès sont devenus facilement falsifiables et franchissables. C'est pour ça, que les chercheurs de différents domaines ont orientés leurs travaux sur des clés et des mots de passe impossible à falsifier, surs et surtout efficaces. Et c'est ainsi que fut inventée la biométrie qui est devenue à la mode dans les domaines qui requiert un niveau élevé de sécurité et de contrôle.

Et parmi toutes les technologies biométriques qui existent, la reconnaissance faciale est l'une des technologies les plus utilisées et les plus adaptées car elle nous permet d'exploiter de nombreuses informations relatives à une personne.

Malgré l'énorme progression en informatique surtout dans le domaine du traitement d'images on trouve toujours des difficultés concernant les systèmes appliquant les techniques de reconnaissance faciale, c'est pour cela qu'elle a engendré un grand nombre de travaux de recherche au cours des 50 dernières années et ceci dans différents domaines tels que l'identification de modèles, réseaux de neurones, vision par ordinateur, infographie …

Dans ce PFE, on s'est justement intéressé à cette thématique fort utile et très demandée dans le domaine de la sécurité des réseaux, la sécurité nationale et même internationale. On s'est proposé d'implémenter une application de reconnaissance faciale utilisant la technique la plus performante de l'état de l'art.

CHAPITRE 1 : La reconnaissance faciale

1.1 Introduction

Durant la vie quotidienne chacun de nous identifie tout au long de la journée différents visages. Ainsi lorsque nous rencontrons une personne, notre cerveau va chercher dans notre mémoire et vérifie si cette personne est répertoriée ou non, c'est une tâche aisée pour les humains. En est-il de même pour une machine ?

Dans ce premier chapitre, nous allons justement présenter les grandes lignes de notre travail, nous allons ainsi expliquer des notions générales sur la reconnaissance faciale, le fonctionnement d'un système de reconnaissance faciale, On va aussi aborder quelques techniques utilisées, des exemples et les domaines d'application.

.

1.2 La reconnaissance faciale

La reconnaissance faciale est un procédé biométrique au même titre que la reconnaissance d'empreintes digitales, d'iris, ou vocale. Cela consiste donc à déterminer l'identité d'une personne. Le système de reconnaissance faciale est une application logicielle visant à reconnaître une personne grâce à son visage de manière automatique.

A l'aide d'algorithmes, cette application analyse toutes les caractéristiques faciales telles que l'écartement des yeux, des arêtes du nez, des commissures des lèvres, des oreilles, du menton, à partir d'une image de son visage qui peut provenir à la fois d'une photo ou d'une vidéo. Donc de nombreux formats d'image peuvent être utilisés qu'ils soient statiques (JPG, bmp, png, GIF …) ou en mouvement (MPEG4 Part 2) [1]

Plusieurs méthodes de reconnaissance de visages ont été proposées durant ces 30 dernières années, suivant deux grands axes : la reconnaissance à partir d'images fixes et la reconnaissance à partir de séquence d'images (vidéo). La reconnaissance de visages basée sur la vidéo est préférable à celle basée sur des images fixes, puisque l'utilisation simultanée des informations temporelles et spatiales aide dans la reconnaissance.

Dans notre projet, on s'est focalisé sur la reconnaissance basée sur la vidéo, puisqu'elle est plus demandée et plus pertinente [2].

1.3 Domaines de la Reconnaissance Faciale

Aujourd'hui la reconnaissance faciale est utilisée principalement pour des raisons sécuritaires. Elle peut être utilisée à des fins très diverses. Par exemple, l'authentification, le contrôle d'accès (autorisation) et la vidéo de surveillance.

Un bon exemple de l'usage des applications d'identification, est le nouveau tunnel qui sera installé d'ici la fin de l'été, situé à Dubaï premier de ce type dans le monde. Il s'agit d'un système biométrique qui permet aux passagers d'être identifiés en traversant un tunnel dans le but d'augmenter l'efficacité des points de contrôle de sécurité. Ils n'ont même pas besoin de montrer leur passeport. L'outil fonctionne grâce à la reconnaissance de l'iris et du visage. La procédure dure environ 15 secondes. [3]

La reconnaissance faciale est aussi utilisée dans les Applications militaires. Un bon exemple de ce domaine est l'utilisation des lunettes de style « Robocop » munies d'une petite caméra d'une portée de 12 milles (19,3 km) par la marine américaine, la caméra peut aussi faire partie de l'optique d'un soldat sur son arme. Grâce à cet équipement, les soldats peuvent identifier des ennemis en quelques secondes sur le terrain, et cela sans réseau à large bande. [4]

En revanche, on distingue un autre domaine d'application de ces systèmes qui est l'assistance à l'utilisateur. Les systèmes de reconnaissance faciale sont de plus en plus présents au quotidien. Ils sont par exemple utilisés sur les réseaux sociaux sur internet pour identifier quelqu'un sur une photo, sur les Smartphones pour les déverrouiller...

Facebook a la plus grande collection et exposition de photos au monde et a maintenant une chance d'unifier tous les visages. Oui, même les personnes qui ne sont pas identifiées. Facebook a annoncé qu'il a développé un programme appelé DeepFace, qui selon les chercheurs, peut déterminer si deux visages appartiennent à la même personne avec une précision de 98%. Selon Facebook, dans 97,53% des cas, les photos à différents âges de même personnes ont la même réponse au test, comme le prouve la procédure de DeepFace pour la reconnaissance du célèbre acteur Sylvester Stallone. [5]

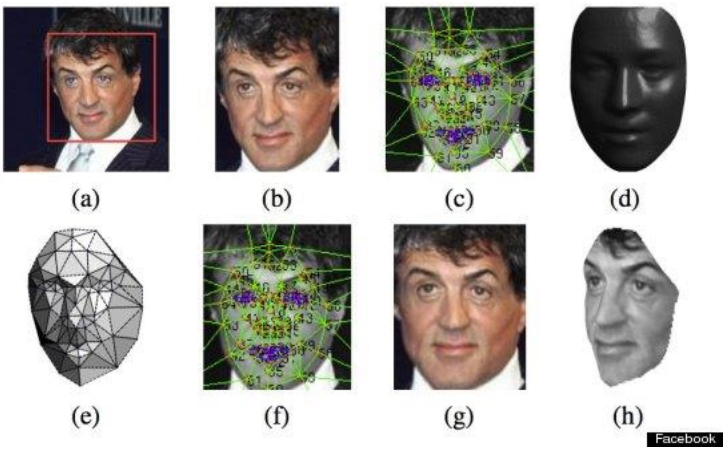


Figure 1 La procédure de DeepFace pour la reconnaissance

La nouveauté dans la reconnaissance faciale arrive grâce au développement de nouvelles caméras de type 3D. Ces caméras obtiennent de meilleurs résultats que les caméras classiques, parce qu'elles acquièrent une image tridimensionnelle de chaque visage (perspectives)pour identifier une personne lorsqu'elle passe par le portail d'authentification.

1.4 Historique

La reconnaissance faciale est une technique biométrique relativement récente. Si l'empreinte digitale est la technique biométrique la plus ancienne inventée en 1903 pour rechercher les criminels, la reconnaissance des visages a été développé par "Benton et Van Allen" en 1968 pour évaluer la capacité d'identification des visages non familiers. Il ne s'agit pas d'un test de reconnaissance ménisque de visages familiers ou non familiers, mais d'une épreuve consistant à apparier des photographies de visages non familiers présentés sous différents éclairages et selon des angles différents et nécessitant une bonne capacité d'intégration Visio-spatiale [6].

L'utilisation des techniques de reconnaissance faciale a connu un développement à grande échelle depuis le milieu des années 90, avec l'utilisation efficace de nouvelles technologies, notamment l'ordinateur et sa capacité de traitement d'images. L'utilisation de ces techniques existe depuis qu'une machine est capable de comprendre ce qu'elle « voit » lorsqu'on la connecte à une ou plusieurs caméras, c'est à dire que les premiers essais datent du début des années 70 (Benton et Van Allen en 1968), et sont basés sur des méthodes à bases d'heuristiques, basés sur des attributs faciaux mesurables comme l'écartement des yeux, des sourcils, des lèvres, la position du menton, la forme, etc. Ces méthodes sont très peu robustes, car elles font de nombreuses suppositions en se plaçant dans des cas très simples (visage de face, bonnes conditions d'illuminations, etc. L'une des premières tentatives de reconnaissance de visage est faite par Takeo Kanade en 1973 lors de sa thèse de doctorat à l'Université de Kyoto. [7][8]

1.5 Les techniques de détection et de reconnaissance faciale

Il existe une multitude de techniques consacrées à la détection du visage. Mais pour la plupart il est d'intérêt que ces techniques se basent sur des éléments du visage qui sont le moins susceptibles aux changements :

Les grands traits supérieurs des orbites, les secteurs entourant les pommettes, les côtés de la bouche et d'autres caractéristiques similaires de façon à ignorer les changements comme la coupe de cheveux.

Les méthodes de base les plus utilisées sont :

- *Le traitement automatique du visage*

C'est une technologie rudimentaire, elle caractérise les visages par des distances et des proportions entre des points particuliers comme les deux yeux, le nez, les coins de la bouche. Aussi éprouvé que les autres technologies, le traitement automatique du visage est le plus efficace dans des situations de capture d'image avec peu d'éclairage [9].

Rapport-gratuit.com

- *Eigenface*

Les « Eigenfaces » furent le premier type de caractérisation utilisé avec succès dans des traitements faciaux tels que la détection et la reconnaissance du visage [10]

Elle utilise une représentation des éléments caractéristiques d'une image de visage à partir d'images modèles en niveau de gris. Des variantes d'Eigenface sont fréquemment utilisées comme base pour d'autres méthodes de reconnaissance [11].

- *L'analyse des points particuliers*

Elle est la technique d'identification faciale la plus utilisée. Cette dernière se rapproche de Eigenface, mais elle est capable de s'adapter à des changements d'aspect facial (sourire, froncement des sourcils, ...). Les ingénieurs numériques l'utilisent souvent [12].

- *LDA (Linear discriminant analysis) fisher*

Elle fait partie des techniques d'analyse discriminante prédictive. Il s'agit d'expliquer et de prédire l'appartenance d'un individu à une classe (groupe) prédéfinie à partir de ses caractéristiques mesurées à l'aide de variables prédictives [13].

- *Filtre de Haar*

La détection du visage dans ce filtre est réalisée par un filtre multi-échelles de Haar. Les propriétés d'un visage sont décrites dans un fichier XML. Elles ne sont pas choisies au hasard et reposent sur un échantillon de quelques centaines d'images tests [14]. Nous reviendrons plus en détails sur cette méthode.

- *Méthode LBP (Local Binary Patterns)*

Les LBP (Modèles binaires locaux) ont également été utilisés pour la détection du visage ou ce dernier est subdivisé en sous–régions carrées de taille égale sur lesquelles sont calculées les caractéristiques LBP. Les vecteurs obtenus sont ensuite concaténés pour obtenir le vecteur de caractéristiques final.

1.6 Problématique

Le problème de la reconnaissance des visages peut être formulé comme suit :

Étant donnée une ou plusieurs images d'un visage, la tâche est de trouver ou de vérifier l'identité d'une personne par comparaison de son visage à l'ensemble des images de visages stockées dans une base de données. En général, un système de biométrie faciale est constitué de deux modules : un module de segmentation de visages (détection ou localisation de visage), et un module de reconnaissance qui se déroule en trois étapes : normalisation ou prétraitement, extraction de caractéristiques faciales, classification et reconnaissance

La reconnaissance faciale pose de nombreux défis car les visages sont des objets déformable 3D. Nous nous limitons dans ce travail à une reconnaissance à partir d'une image 2D de visage en environnements. De tels systèmes doivent pouvoir s'affranchir des problèmes suivants :

• **variations des poses**: Les changements d'orientation et les changements de l'angle d'inclination du visage engendrent de nombreuses médications d'apparence dans les images collectées. Provoquant des déformations qui font varier la forme globale du visage.

Figure 2 Changement d'angle (poses).

• **Changer l'éclairage:** L'intensité et la direction de l'éclairage pendant la prise de vue affectent grandement l'apparence du visage dans l'image. Donc il faut éviter surtout de collecter des vues à différents moments, à l'intérieur et à l'extérieur. Étant donné une forme particulière du visage humain, ces changements dans l'éclairage peuvent révéler des ombres qui mettent en évidence ou cachent certaines caractéristiques du visage.

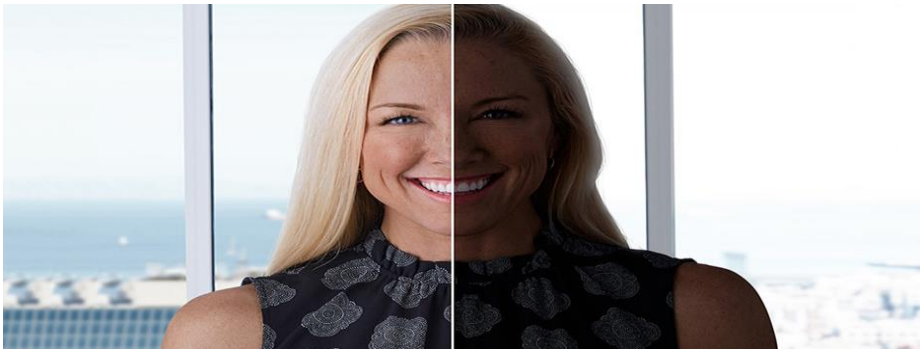


Figure 3 Changement d'éclairage

• **Changement des expressions faciales:** Le visage est un élément non-typique. Le sentiment qui exprime le visage, en plus de la transformation résultant de la parole, se traduit par un changement significatif perceptible et le nombre de structures possibles ne peut être étendu de façon réaliste.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Figure 4 Changement des expressions faciales

• **Influence des occultations :** Un visage peut être partiellement masqué par des objets ou par le port d'accessoires tels que les lunettes, un chapeau, une écharpe. Les occultations peuvent être intentionnelles ou non. Dans le contexte de la reconnaissance, il peut s'agir d'une volonté délibérée d'empêcher la reconnaissance. Il est clair qu'elle sera d'autant plus difficile que peu d'éléments discriminants seront simultanément visibles

D'autre part, relever des défis spécifiques à la reconnaissance:

• Images prises à des années d'intervalle: l'apparence du visage peut changer de manière significative au fil des ans, conduisant à des fausses erreurs de rejet qui peuvent provoquer le refus d'accès des utilisateurs légitimes.

• Identités similaires: les systèmes de reconnaissance biométrique sont suspectés d'erreurs d'acceptation fictive lorsqu'un imposteur ayant une identité similaire peut être accepté en tant qu'utilisateur légitime.

Dans certains cas, les données d'entraînement utilisées pour les systèmes de détection et de reconnaissance de visage sont des images de visages frontaux, mais évidemment, certaines difficultés apparaissent lorsqu'on essaie d'analyser des visages avec des angles de pose différents. Par conséquent, certains systèmes de traitement introduisent des images ayant subi une rotation dans leur ensemble de données de formation et d'autres tentent d'identifier les caractéristiques particulières du visage comme les yeux, la bouche et le nez, afin d'améliorer les ratios positifs.

1.7 Conclusion

Dans ce chapitre nous avons présenté les grandes lignes de notre travail, comme les notions de base et les techniques utilisées (état de l'art). Nous avons ainsi mis en valeur les points critiques à soigner et dégager les parties à aborder lors des étapes suivantes de la réalisation de notre projet. Dans le chapitre suivant, on abordera l'analyse de notre logiciel pour spécifier les étapes du travail et en dégager les principales fonctionnalités de l'application.

Chapitre 2 spécifications et conceptions

2.1 introduction

Dans ce chapitre, On va parler des spécifications et de la conception de notre application qui consiste à détecter des visages des personnes à partir d'une séquence vidéo. L'entrée de notre application est une séquence vidéo. Tout d'abord, on décompose la vidéo en un ensemble d'images, et à partir de cet ensemble on cherche le meilleur choix qui correspond à l'image dans laquelle l'algorithme de recherche a détecté la correspondance entre l'image courante et l'ensemble d'image enregistré dans la base de donnée donc l'algorithme de reconnaissance vérifie l'existence de la personne recherchée dans la base.

2.2 Spécification de la technique et langage utilisés

L'objectif de notre travail consiste à développer une application permettant la reconnaissance des visages des personnes à partir d'une séquence vidéo. L'entrée de notre application est une séquence vidéo qui sera décomposée en un ensemble d'images. Ensuite, à partir de cet ensemble on choisit l'image dans laquelle l'algorithme de recherche a détecté le maximum d'objets morphologiques du visage (les deux yeux, le nez et la bouche) puis on calcule les distances caractéristiques et on les compare avec celles des images de la base afin d'obtenir un score de ressemblance, ce dernier est comparé à un seuil déjà fixé pour vérifier l'existence de la personne recherchée dans la base.

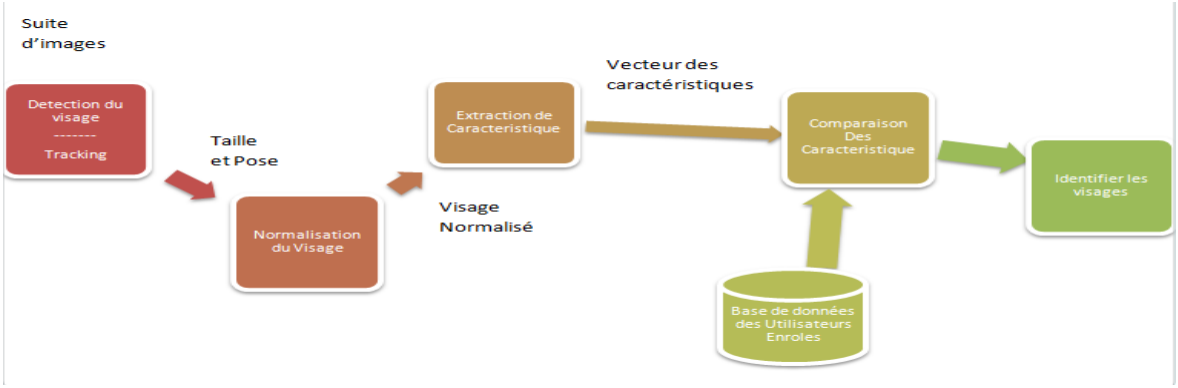


Figure 5: Schéma général d'un système de reconnaissance des visages.

2.2.1 Techniques de détection utilisées

On va passer en revue brièvement les techniques choisies pour détecter les visages, et la raison pour laquelle nous les avons choisies. Précédemment, nous avons brièvement présenté quelques techniques et méthodes de résolution spécifiques à notre problématique, mais nous avons choisi deux d'entre elles, les plus populaires, ayant donnés des résultats pertinents selon l'état de l'art la méthode Haar et BLP, que nous allons un peu plus détaillées dans ce chapitre.

A) Détection de Face à l'aide de la fonction de Haar

La détection d'objets à l'aide de classificateurs de cascade, basée sur les fonctions de Haar est une méthode efficace de détection d'objet proposée par Paul Viola et Michael Jones dans leur article "Détection rapide d'objets utilisant une cascade de fonctions simples" en 2001 .[15] Il s'agit d'une approche ou la fonction cascade est formée à partir de nombreuses images positives et négatives, qui est ensuite utilisée pour détecter des objets dans d'autres images.

Pour notre cas nous allons travailler avec la détection de visage. Initialement, l'algorithme a besoin de beaucoup d'images positives (images de visages) et d'images négatives (images sans visages) pour former le classificateur. Ensuite, nous devons extraire des fonctionnalités de celui-ci. Pour cela, les caractéristiques de Haar présentées dans l'image ci-dessous sont utilisées. Ils sont juste comme notre noyau convolutif. Chaque caractéristique est une valeur unique obtenue en soustrayant la somme des pixels sous un rectangle blanc de la somme des pixels sous un rectangle noir**.**

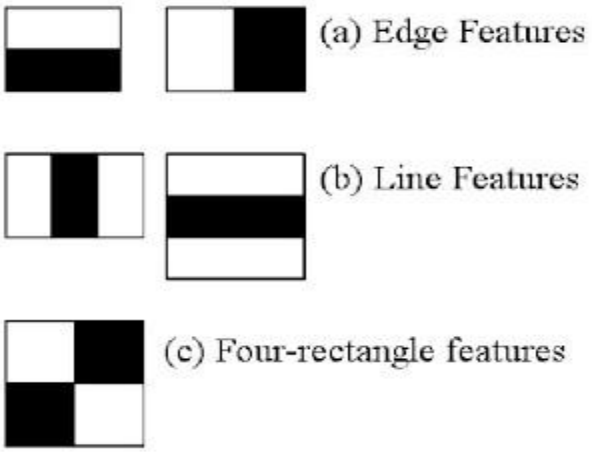


Figure 6 : Caractéristiques de Haar

Maintenant, toutes les tailles et les emplacements possibles de chaque noyau sont utilisés pour calculer beaucoup de fonctionnalités. (Imaginez de combien de calculs il a besoin? Même une fenêtre 24x24 donne plus de 160000 fonctions). Pour chaque calcul d'entité, nous devons trouver la somme des pixels sous les rectangles blancs et noirs. Pour résoudre cela, ils ont introduit les images intégrales. Il simplifie le calcul de la somme des pixels, quelle que soit la taille du nombre de pixels, pour une opération impliquant seulement quatre pixels. Mais parmi toutes ces caractéristiques que nous avons calculées, la plupart d'entre elles ne sont pas pertinentes. Par exemple, considérez l'image ci-dessous. La rangée supérieure montre deux bonnes caractéristiques. La première caractéristique sélectionnée semble se concentrer sur la propriété que la région des yeux est souvent plus sombre que la région du nez et des joues. La deuxième caractéristique sélectionnée repose sur la propriété que les yeux sont plus sombres que le pont du nez. Mais les mêmes fenêtres appliquées sur les joues ou tout autre endroit est sans importance. Alors, comment pouvons-nous sélectionner les meilleures fonctionnalités sur 160000+ fonctionnalités?

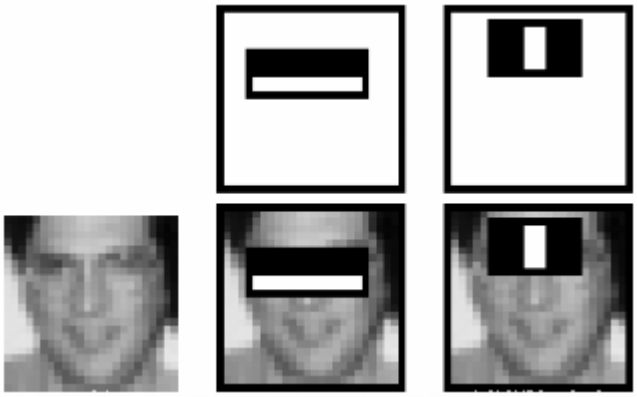


Figure 7 : les propriétés sélectionnées par Haar

Pour cela, nous appliquons chaque fonctionnalité sur toutes les images d'entraînement. Pour chaque caractéristique, il trouve le meilleur seuil qui classifiera les faces positives et négatives. Mais évidemment, il y aura des erreurs ou des erreurs de classification. Nous sélectionnons les fonctions avec un taux d'erreur minimum, ce qui signifie que ceux sont les caractéristiques qui classent le mieux les images de visage et de non-visage. Le processus n'est pas aussi simple que cela: chaque image reçoit un poids égal au début. Après chaque

classification, les poids des images mal classées sont augmentés, le même processus est effectué, les nouveaux taux d'erreurs sont calculés. Le processus est poursuivi jusqu'à ce que la précision requise ou le taux d'erreur soit atteint ou que le nombre requis de caractéristiques soit atteint.

Le classificateur final est une somme pondérée de ces classificateurs faibles. Il est appelé faible parce que seul il ne peut pas classer l'image, mais avec d'autres forme un classificateur fort. Le papier indique que même 200 caractéristiques fournissent la détection avec l'exactitude de 95%. Leur installation finale a environ 6000 fonctionnalités. (Imaginez une réduction de 160000+ fonctionnalités à 6000. C'est un gros gain).

Alors maintenant, prenons une image. Prenons chaque fenêtre 24x24. Appliquons 6000 fonctionnalités à celui-ci. Vérifions si c'est face ou non. C'est bien sur inefficace et prend du temps. Mais les chercheurs ont une bonne solution pour cela.

Dans une image, la majeure partie de la région d'image est une région sans visage. C'est donc une meilleure idée d'avoir une méthode simple pour vérifier si une fenêtre n'est pas une région de visage. Si ce n'est pas le cas, jetons-le en un seul coup. Ne pas le traiter à nouveau. Au lieu de se concentrer sur la région où il peut y avoir un visage. De cette façon, nous pouvons trouver plus de temps pour vérifier une région de visage possible.

Pour cela, ils ont introduit le concept de cascade de classificateurs. Au lieu d'appliquer toutes les fonctionnalités de 6000 sur une fenêtre, regrouper les fonctions dans différentes étapes de classification et appliquer une par une, est plus intelligent. (Normalement, les premières étapes contiennent très peu de fonctionnalités). Si une fenêtre échoue à la première étape, nous la jetons. Nous ne considérons pas les fonctionnalités restantes. Si cela réussit, nous appliquons la deuxième étape des fonctionnalités et poursuivons le processus. La fenêtre qui passe toutes les étapes est une région de visage.

Le Détecteur d'auteurs avait 6000+ caractéristiques avec 38 étapes avec 1, 10, 25, 25 et 50 caractéristiques dans les cinq premières étapes. (Deux caractéristiques dans l'image ci-dessus sont réellement obtenues comme les deux meilleures caractéristiques d'Adaboost). Selon les auteurs, en moyenne, 10 caractéristiques sur 6000+ sont évaluées par sous-fenêtre.

Il s'agit donc d'une simple explication intuitive de la manière dont fonctionne la détection de visages Viola-Jones.

B) Détection de Face à l'aide de fonction de LBP

LBP présente un espace caractéristique discriminant qui peut être appliqué à des problèmes de détection des visages et de reconnaissance, motivé par son invariance par rapport aux transformations d'échelle de gris monotones (par exemple, tant que l'ordre des valeurs de gris reste le même, la sortie de l'opérateur LBP continue est constante, et le fait qu'il peut être extrait en un seul balayage à travers l'image entière.

Les motifs binaires locaux (LBP) sont des descripteurs de texture qui peuvent également être utilisés pour représenter des visages, puisqu'une image de visage peut être vue comme une composition de motifs de micro-texture.

En bref, la procédure consiste à diviser une image faciale en plusieurs régions où les caractéristiques LBP sont extraites et concaténées dans un vecteur de caractéristiques qui sera plus tard utilisé comme descripteur facial.

Le LBP est apparu à l'origine comme un descripteur de texture générique. L'opérateur attribue une étiquette à chaque pixel d'une image en limitant un voisinage 3x3 avec la valeur de pixel central et en considérant le résultat comme un nombre binaire. Dans différentes publications, les valeurs résultantes circulaires 0 et 1 sont lues soit dans le sens des aiguilles d'une montre, soit dans le sens inverse des aiguilles d'une montre. Dans ce PFE, le résultat binaire sera obtenu en lisant les valeurs dans le sens des aiguilles d'une montre, à partir du voisin supérieur gauche, comme on peut le voir sur la figure suivante.

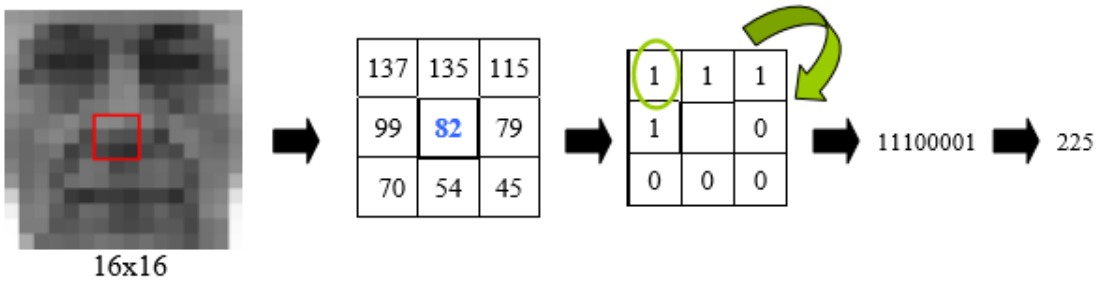


Figure 8 : Attribution d'une étiquette à chaque pixel d'une image

En d'autres termes, étant donné une position de pixel (xc, yc), LBP est défini comme un ensemble ordonné de comparaisons binaires d'intensités de pixels entre le pixel central et ses pixels environnants. La valeur d'étiquette décimale résultante du mot de 8 bits est calculée.

.Afin de traiter les textures à différentes échelles, l'opérateur LBP a été étendu pour utiliser des quartiers de tailles différentes. En utilisant des voisins circulaires et une interpolation bilinéaire des valeurs de pixels, n'importe quel rayon et nombre d'échantillons dans le voisinage peuvent être manipulés. Par conséquent, la notation suivante est définie:

(P, R) ce qui signifie P points d'échantillonnage sur un cercle de rayon R.

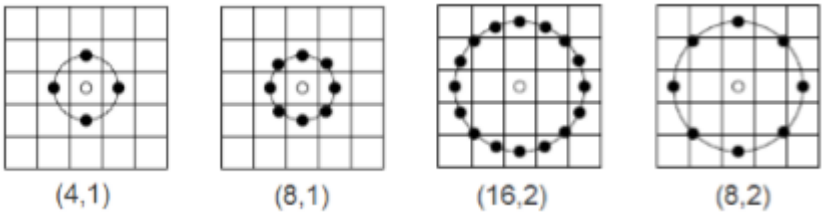


Figure 9 : LBP, Différents exemples de points d'échantillonnage et de rayons

La raison pour laquelle les quatre points sélectionnés correspondent à ceux verticaux et horizontaux, est que les faces contiennent plus d'arêtes horizontales et verticales que diagonales.

Lors du calcul des opérations de pixel prenant en compte les voisinages NxN à la limite d'une image, une partie du masque NxN se trouve hors du bord de l'image. Dans de telles situations, différentes techniques de remplissage sont généralement utilisées, telles que le remplissage à zéro, la répétition d'éléments de bordure ou l'application d'une réflexion miroir pour définir les bordures d'image. Cependant, dans le cas de l'opérateur LBP, la limite critique, définie par le rayon R de l'opération circulaire, n'est pas résolue en utilisant une technique de remplissage, plutôt que cela, l'opération est démarrée au pixel d'image (R, R). L'avantage est que l'histogramme final des étiquettes LBP ne sera pas affecté par les bordures, bien que la taille d'image des étiquettes LBP résultante sera réduite à (Largeur-R) x (Hauteur-R) pixels.

Les images suivantes montrent le résultat de l'application de modèles binaires locaux LBP sur l'image. Dans les images résultats (a) et (b), on peut clairement observer pourquoi les LBP sont appelés descripteur de texte.

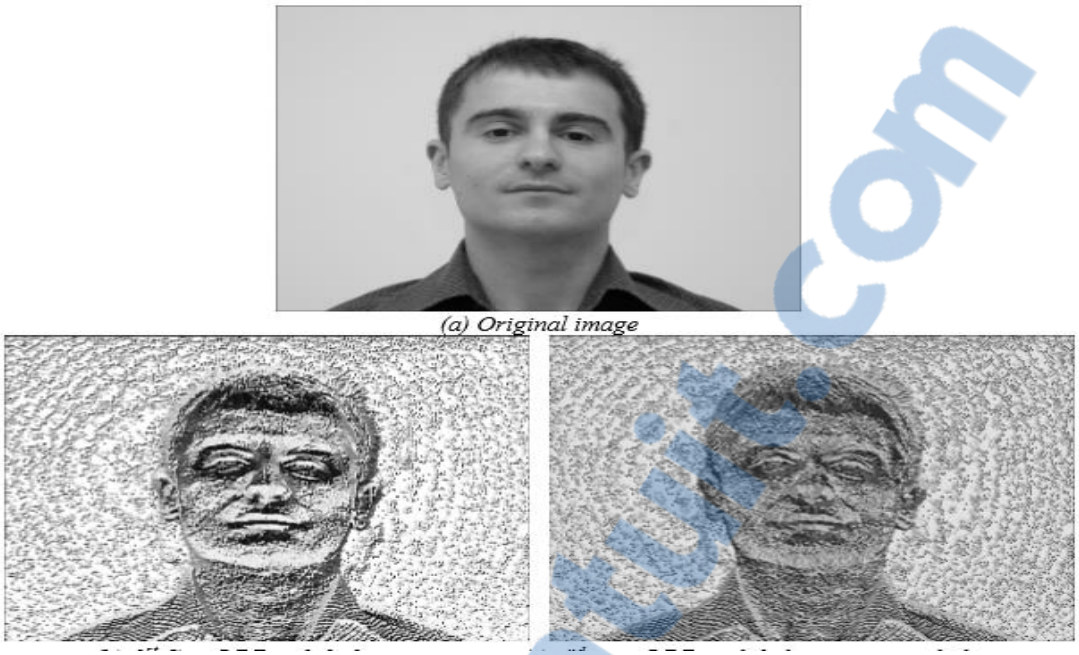


Figure 10 : Résultat de l'application du LBP sur l'image a

Comme le montre la figure10, les contours des traits du visage (yeux, bouche, nez, parenté, sourcils ...) sont clairement remarqués. Dans l'image des étiquettes de premier niveau, les contours sont constamment mis en évidence, ce qui donne une vue d'ensemble générale de la netteté de l'image, ce qui est utile pour distinguer les images sans visage dans la première phase rapide. D'un autre côté, dans l'image des étiquettes du deuxième étage, les informations de texture locales sont plus détaillées, ce qui est utile pour la décision finale, où le facteur de l'image est évalué localement.

Les figures suivantes sont destinées à comparer les histogrammes des faces, des non-faces et des histogrammes des deux étapes. La figure 11montre un échantillon d'image de visage de 16x16 pixels, la figure 12 montre le résultat de l'image LBP et la Figure 13 son histogramme avec des motifs non uniformes accumulés dans le premier casier.

Figure 11 : Simple image « face »

Figure 12 : Application du LBP sur figure 11

Rapport-Gratuit.com

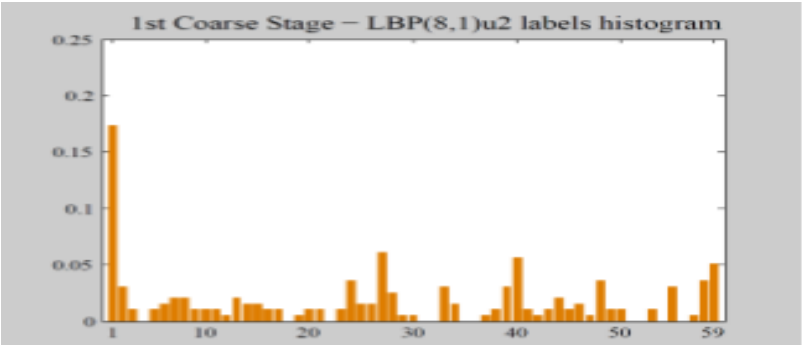


Figure 13 : Histogramme de la figure 11

C)Comparaison entre LBP et Haar :

Une cascade de LBP peut être entraînée de manière similaire (ou meilleure) que la cascade de Haar, mais la cascade de Haar est environ 3 fois plus lente, et selon nos données, environ 1-2% de plus pour détecter avec précision l'emplacement d'un visage. Cette augmentation de la précision est assez importante étant donné que la détection du visage peut fonctionner dans la plage de précision de +95%.

Voici quelques résultats lorsque nous utilisons le jeu de données MUCT. La base de données MUCT comprend 3635 visages avec 76 points de repère manuels. La base de données a été créée pour fournir une plus grande diversité d'éclairage, d'âge et d'origine ethnique que les bases de données de visages 2D. [16]

Une détection correcte est notée lorsqu'il y a un chevauchement d'au moins 50% entre la vérité au sol et les coordonnées détectées par OpenCV.

Cascade: Haarcascade_frontalface_alt2.xml

Fichier de données: muct.csv

frappe	**Non Detecté**	**Faux détecté**	**Double-detecté**
3635	**55**	**63**	**5**
		Temps: 4m2.060s	

Tableau 1: Les Résultats par détection Haar

Cascade: lbpcascade_frontalface.xml

Fichier de données: muct.csv

frappe	Non Detecté	Faux détecté	Double-detecté
3635	**106**	**77**	**3**
		Temps: 1m12.511s	

Tableau 2: Les Résultats par détection LBP

On remarque que la cascade Haar est beaucoup plus précise que LBP. Cependant, LBP est beaucoup plus rapide. Donc nous avons décidé de les utiliser conjointement dans un esprit de comparaison et de multiplicité.

D) Conclusion

Après avoir donné un aperçu général des différentes méthodes de détection du visage humain, il fut décidé que notre travail sera basé sur une technique utilisant des opérateurs LBP et Haar (qui peuvent être utilisés aussi bien pour la détection des visages et la reconnaissance), motivés par leurs robustesses à supporter les changements d'éclairage et aussi en raison de leurs faibles complexités de calcul.

2.2.1 Choix du langage de programmation

C#	JAVA	MATLAB	C++
Langage interprété	Langage intermédiaire	Langage interprété	Langage compilé
Utilisé pour les applications avec interfaces graphique. .	Utilisé pour les applications avec interfaces graphiques et la programmation orientée internet	Utilisé pour le Calcul mathématique et les solutions nécessitant très grande puissance de calcul	utilisé pour développer des applications graphiques et pour la programmation système objet.
Disponible sur plusieurs plateformes	Disponible sur plusieurs plateformes	Disponible sur plusieurs plateformes	Disponible sur plusieurs plateformes
Parmi les langages les plus rapides	Rapide	Rapide	Très rapide
orientée objet	orientée objet	orientée prototype et orientée objet	orientée objet

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Nombreuse bibliothèques	Nombreuse bibliothèques	Nombreuses fonctionnalités mathématique	Nombreuse bibliothèques
Plateforme de développement à usage facile. (Visual studio)	Plateforme de développement à usage moyennement facile.	Plateforme de développement à usage moyen.(Calcul)	Plateforme de développement à usage facile. (Visual studio)

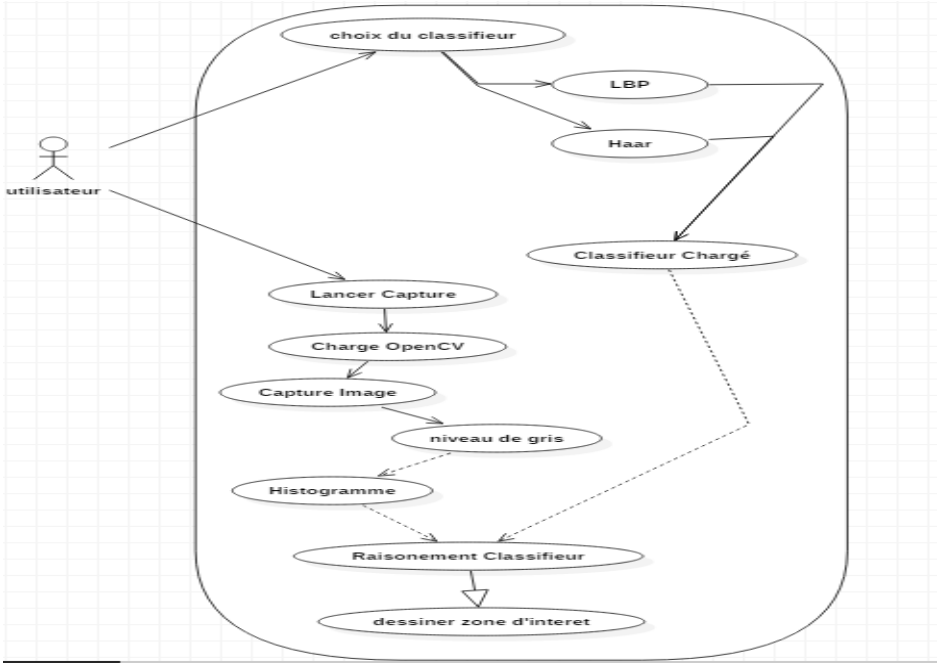
Tableau 3: Tableau comparative entre langage de programmation

D'après le tableau comparatif ci-dessus, les deux langages de programmation les plus adapté au métier d'imagerie sont JAVA et C++. Cependant, le langage JAVA possède plus de souplesse, plus de sources d'informations pour le développement que le langage C++ dans le domaine de l'imagerie.

- Java est aujourd'hui un langage aussi rapide que le c++ pourvu qu'on ne l'utilise pas pour une application très lourde.

Java est connu et donc il y a plus de chance de trouver des documentations, des développeurs java. En plus du fait que notre maitrise de Java est plus convaincante que celle de c++, donc c'est la raison de notre choix pour implémenter notre application.

2.2.2 Les diagrammes de cas d'utilisation



Dans ce cas d'utilisation, l'utilisateur lance la capture, OpenCV récupère l'ensemble des utilisateurs qui se trouvent dans la base de donnée, ensuite le système effectue la reconnaissance. Deux résultats existent: soit la personne n'est pas reconnue (n'existe pas dans la base) soit elle est reconnue (existe dans la base). Pour le deuxième cas l'application affiche les coordonnées de la personne reconnue.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

2.3 Conception

Dans ce qui suit, nous allons expliquer les raisons de nos choix quant aux outils de développement les plus adéquats avec nos différents besoins, ce qui va nous permettre d'atteindre au mieux et le plus rapidement nos objectifs, aussi de bien mener la phase d'implémentation, C'est la Conception.

2.3.1 Conception global

2.3.1.1 OpenCV

OpenCV (Open Source Computer Vision Library) est une bibliothèque open source de logiciels de vision artificielle et d'apprentissage automatique, initialement développée par Intel. Dans un langage simple, elle est spécialisée dans le traitement d'images en temps réel. Elle est optimisée pour les processeurs multi-cœurs, comme elle est multiplateforme c'est-à-dire disponible sur Linux, Windows, Mac OS X) et gratuite sous licence BSD.

OpenCV (la bibliothèque) a été construite pour fournir une infrastructure commune pour les applications de vision par ordinateur et pour accélérer l'utilisation de la perception de la machine dans les produits commerciaux. Adoptée dans le monde entier, OpenCV compte plus de 47 000 utilisateurs et un nombre de téléchargements estimé à plus de 14 millions. La bibliothèque dispose de plus de 2500 algorithmes optimisés, Ces algorithmes peuvent être utilisés pour détecter et reconnaître des visages, identifier des objets, classer des actions humaines, suivre des mouvements de caméra, suivre des objets en mouvement, extraire des modèles 3D d'objets, trouver des images similaires à partir d'une base de données, etc. En tant que produit sous licence BSD (Berkeley Software Distribution License), OpenCV facilite l'utilisation et la modification du code par les entreprises. [5]

Parmi les caractéristiques d'Open CV on peut citer :

• Analyse du mouvement (flot optique, segmentation de mouvement, suivi).

• Reconnaissance d'objets (Eigen-méthodes, HMM).

• Diverses structures de données dynamiques (listes, les files d'attente, ensembles, arbres, graphes).

• Traitement de l'image de base (filtrage, détection de contour, détection d'angle, l'échantillonnage et l'interpolation, de conversion des couleurs, des opérations morphologiques, histogrammes, pyramides d'images).

• Image et vidéo I/O (fichier et l'appareil photo d'entrée en fonction, image/fichier de sortie vidéo).

2.3.1.2JavaFx

JavaFX est une famille de produits et de technologies d'Oracle Corporation (initialement Sun Microsystems), pour la création d'applications Internet riches (RIA),y compris les applications

multimédias interactives.il contient des outils très divers, notamment pour les médias audio et vidéo, le graphisme 2D et 3D, la programmation Web, la programmation multi-fils etc..

JavaFx étend la puissance de Java en permettant aux développeurs d'utiliser n'importe quelle bibliothèque Java dans les applications JavaFX. De cette façon, les développeurs peuvent augmenter leurs capacités Java et utiliser la technologie de présentation fournie par JavaFX pour créer un attrait visuel.

Parmi ses principales caractéristiques, JavaFx permet aux développeurs d'intégrer des graphiques vectoriels, des animations, des contenus Web audio et vidéo dans une application interactive, complète et attrayante, et aussi de maintenir un flux de travail efficace entre concepteur et développeur dans lequel les concepteurs peuvent travailler sur les outils qu'ils souhaitent tout en collaborant avec les développeurs.

On note que les applications écrites à l'aide de cette bibliothèque peuvent s'exécuter de manière cohérente sur plusieurs plates-formes. [18]

2.3.1.3 Maven

Le déploiement d'applications est devenu aujourd'hui un vrai casse-tête. En effet, à chaque phase du projet, les développeurs doivent gérer un environnement différent : intégration, recette, pré-production, production. Chaque environnement possède ses propres caractéristiques : adresses IP, serveurs de bases de données, etc.

Aujourd'hui, les applications sont la plupart du temps générées à l'aide de scripts Ant, de Shell scripts ou même à la main. Cela revient souvent à faire des multitudes de copier/coller de scripts et à les réadapter à chaque projet. Maven permet donc de s'affranchir de ces contraintes et d'uniformiser le déploiement des applications. [19]

Maven est un outil de construction de projets (build) open source développé par la fondation Apache, initialement pour les besoins du projet Jakarta Turbine. Un élément clé relativement spécifique de Maven est son aptitude à fonctionner en réseau. Il utilise un paradigme connu sous le nom de Project Object Model (POM) afin de décrire un projet logiciel, ses dépendances avec des modules externes et d'ordre à suivre pour sa production. Il est livré avec un grand nombre de tâches prédéfinies, comme la compilation de code Java ou encore sa modularisation.

Maven permet de faciliter et d'automatiser certaines tâches de la gestion d'un projet Java. Il permet notamment :

* de gérer des dépendances vis à vis des bibliothèques nécessaires au projet ;

* de générer des documentations concernant le projet.

* de gérer le build et le reporting de l'application [20].

2.3.2 Conception détaillée

2.3.2.1 Diagramme de séquence « identification »

Dans ce diagramme de séquence, l'utilisateur est l'acteur dans notre application, il choisit le classifieur (LBP ou Haar) qui convient et puis il lance la capture. Cette dernière sera décomposée en un ensemble d'images dans chacune d'elles, nous détectons le visage à l'aide de OpenCV qui applique un algorithme de détection du visage.

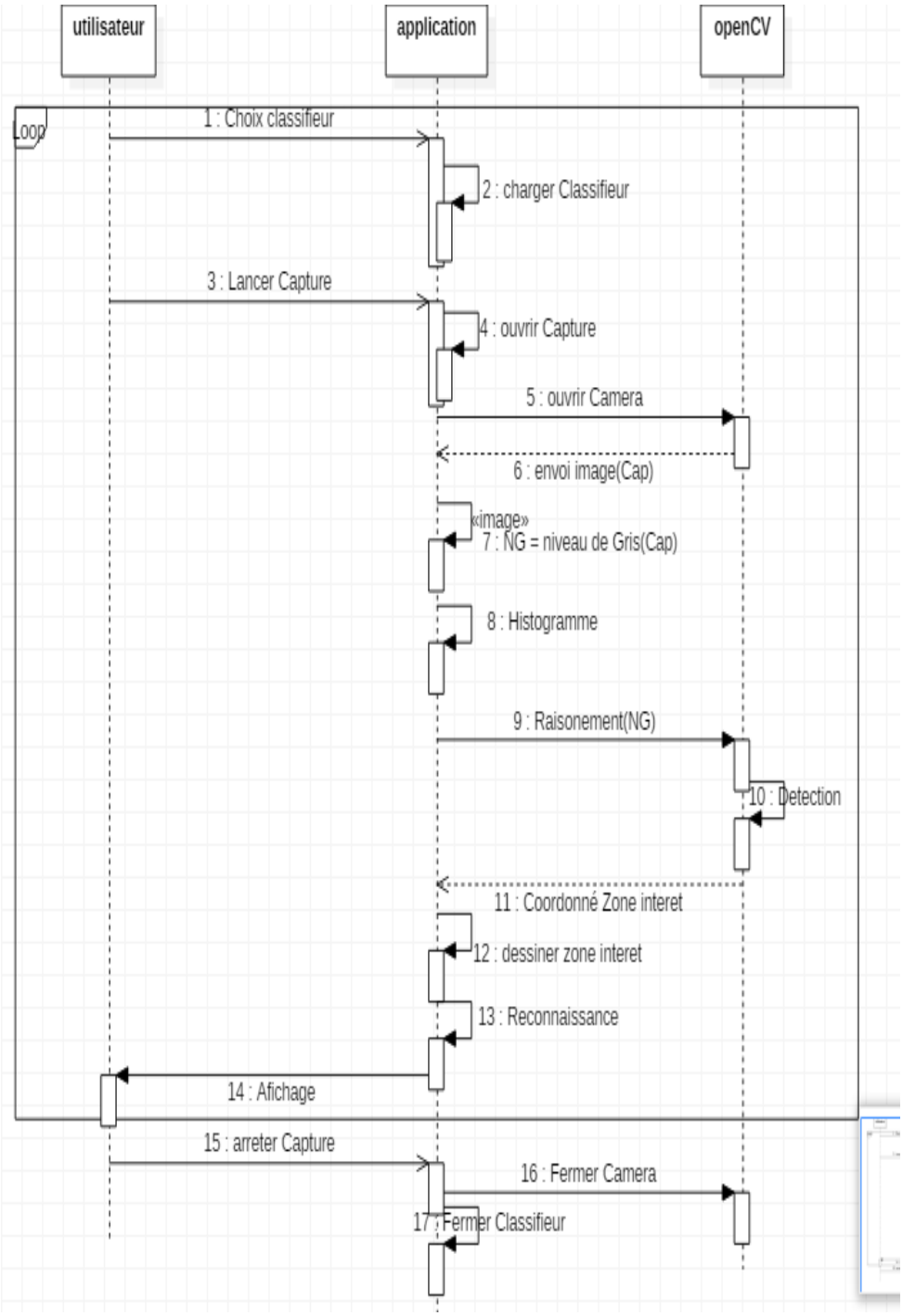


Figure 16 : Diagramme de séquence « identification »

2.3.1.2 Diagramme de séquence « Reconnaissance » :

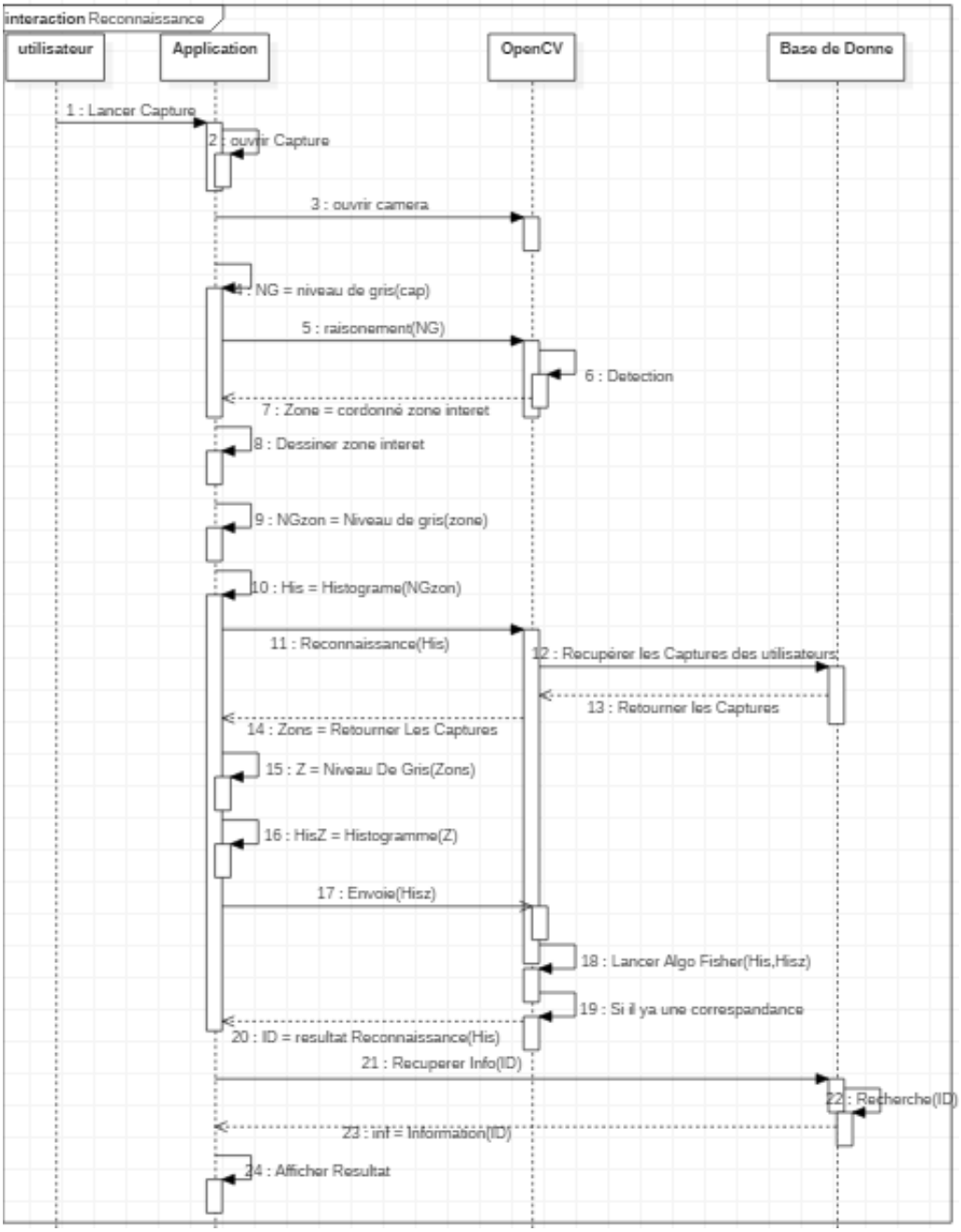


Figure 17 : Diagramme de séquence « reconnaissance »

2.4 Conclusion

Dans ce chapitre, nous avons énumérer les différents besoins fonctionnels, en proposant des diagrammes des cas d'utilisation qui nous ont aidé à mieux comprendre les rôles et les fonctionnalités de notre application.

Nous avons aussi pu modéliser le fonctionnement du système. Ceci est fait à travers différents diagrammes de séquences qui nous permettent de bien spécifier la composition et le comportement de l'application. Notre choix s'est fait sur les deux techniques de détection et reconnaissance de visages Haas et LBP pour solutionner notre problème. Puis l'utilisation de Java s'est imposée, place maintenant à l'implémentation dans le prochain chapitre.

Chapitre 3 : Réalisation

3.1 Introduction

La dernière étape du processus de développement concerne l'implémentation de l'application en fonction des technologies choisies.

Cette étape correspond à la phase de construction du service de synchronisation et de l'application principale. Nous commencerons tout d'abord, par la présentation de l'environnement de travail utilisé pour développer notre projet de fin d'études. Puis nous donnerons quelques screeshots de notre application démontrant ainsi son niveau de performance.

3.2 Environnement du Travail

Afin de bien mener ce projet, nous avons utilisé un PC dont les principales caractéristiques sont les suivantes :

Processeur : Intel(R) CoreTM i7-7500u @2.7 2,9 GHZ

RAM : 8.00 GO

OS : windows 10 famille

Webcam intégrée

3.3 Présentation

Cette interface présente le menu principal de notre application, une sorte de passerelle pour accéder aux applications du PFE à partir duquel nous choisissons le type de reconnaissance à utiliser :

3.3.1 Détection de visage :

A. *Choix du Classifieur :*

L'utilisateur peut choisir entre deux classifieur LBP ou Haar pour le raisonnement de détection, ce choix est réalisé en utilisant deux Champs « Check Box »:

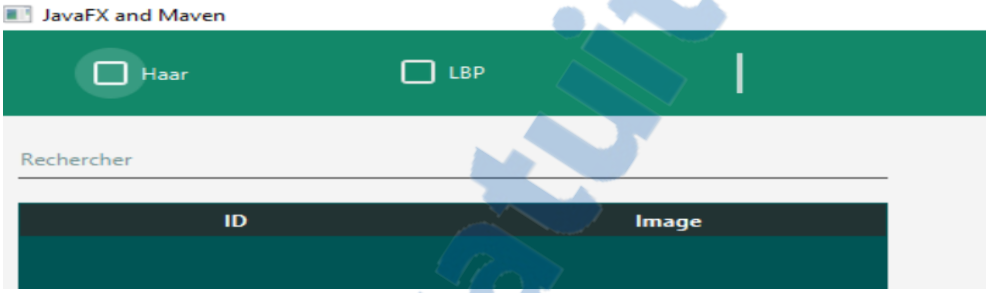


Figure 18 : La zone de choix de classifieur

B. Lancement de capture

L'utilisateur lance la capture en cliquant sur le bouton « Lancer Camera ».

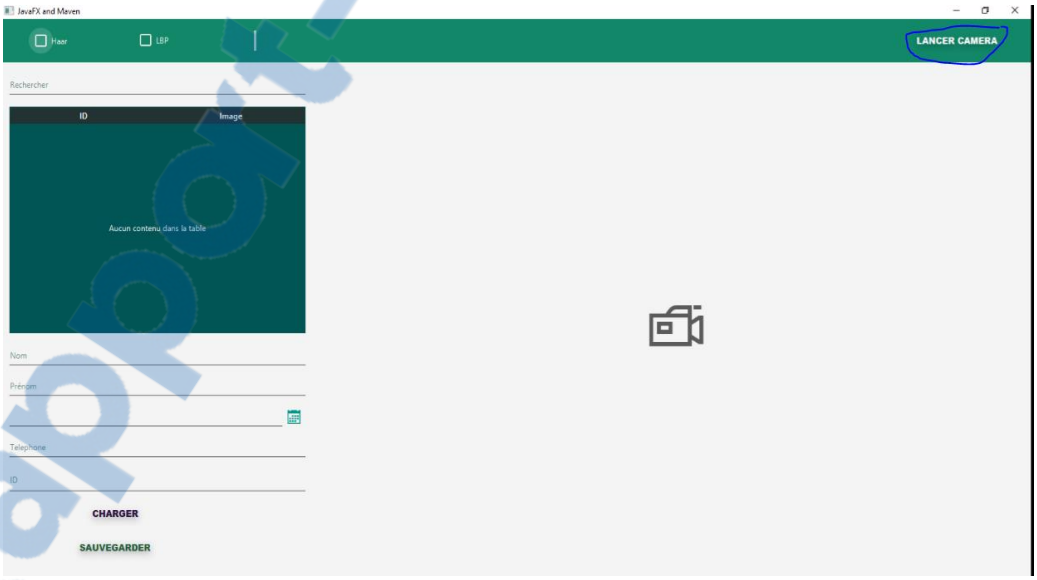


Figure 19 : Menu principal

C. La Détection :

Rapport-Gratuit.com

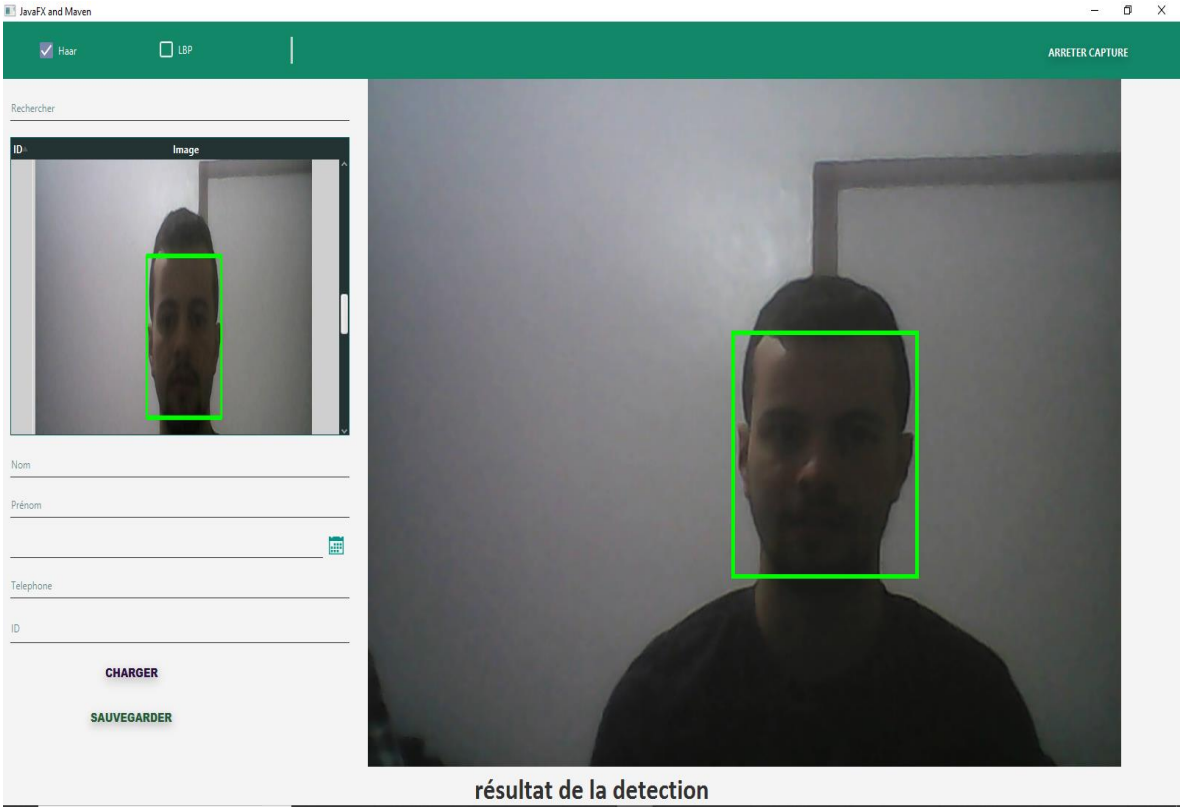


Figure 20 : Capture de l'application en fonctionnement.

Si le visage a été détecté, le Programme va dessiner la zone d'intérêt « l'emplacement du visage »

Dans notre Application, on a implémenté deux techniques de détection de visage : LBP et Haar. Ce sont deux méthodes différentes qui utilisent des classifieurs différents.

On a utilisé Heidi SQL qui est un outil d'administration de base de données possédant un éditeur sql et un constructeur de requête.

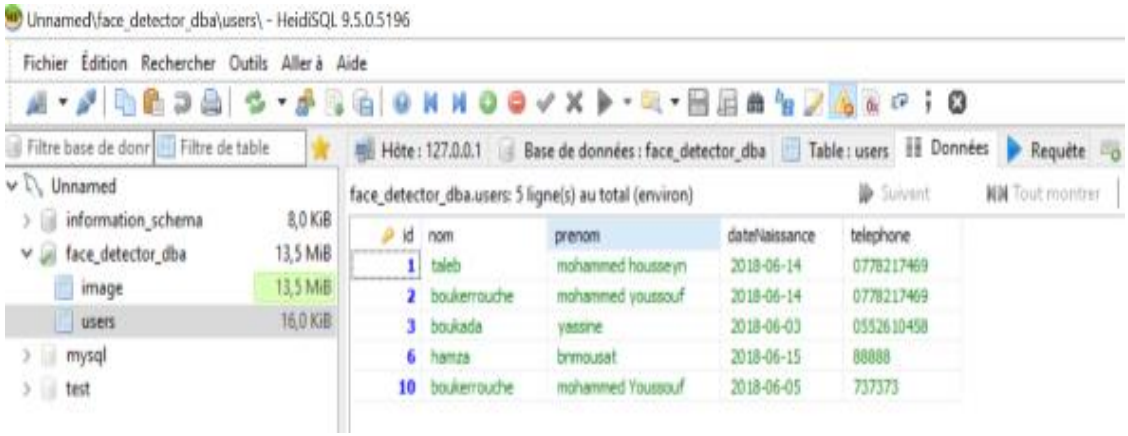


Figure 21: Capture sur interface HeidiSQL

On a rempli notre base de données de photos de personnes et ces photos sont déplacées vers un dossier temporaire (trainingDir) qui sera créé sur le disque dure C.

Figure 22 : Capture des dossiers créés par l'application

3.3.3 Reconnaissance faciale :

Dans notre application, on a utilisé la technique d'Analyse Discriminante Linéaire (LDA).

Une fois la personne que l'on souhaite reconnaitre ne correspond à aucune personne dans notre base de données, un message s'affichera pour indiquer que cette personne n'est pas fichée.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

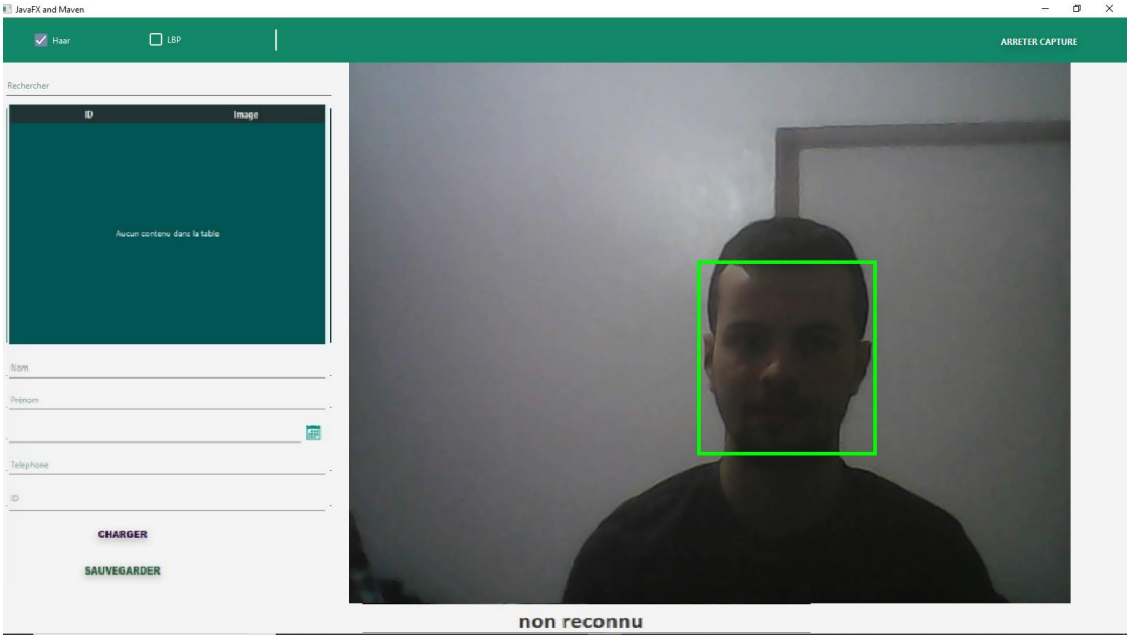


Figure 23 : Reconnaissance échouée

Dans le cas où la personne n'existe pas dans la base de données (n'est pas reconnue), nous avons prévu la possibilité de l'y ajouter en écrivant son nom, prénom …

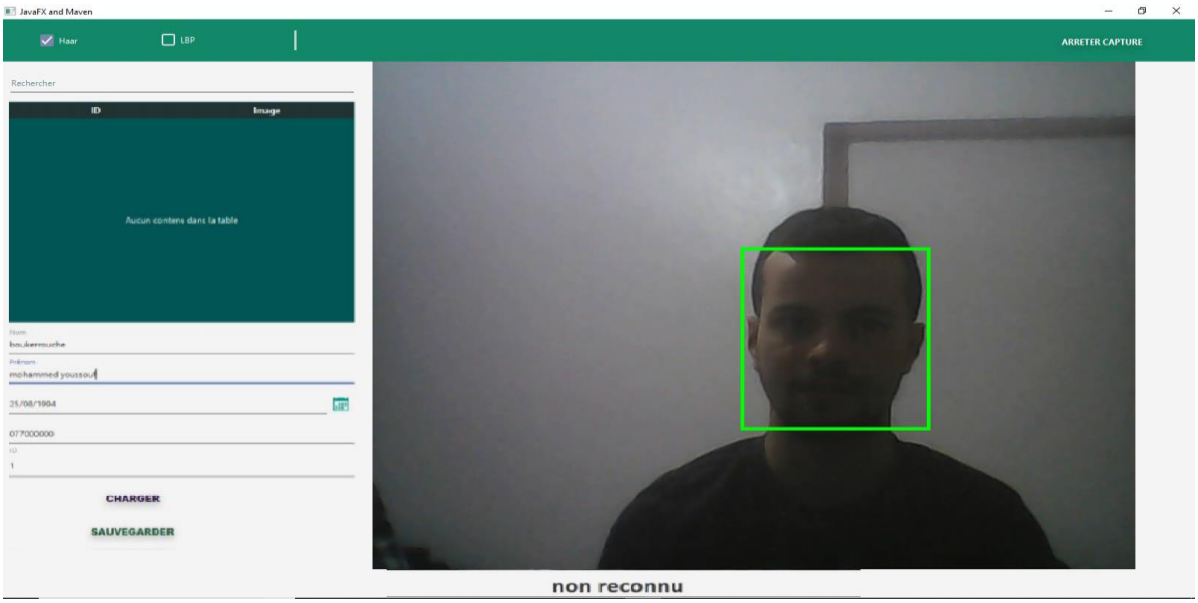


Figure 24 : L'ajout d'un utilisateur

Si la personne que l'on souhaite reconnaitre existe dans la base de données, l'application va afficher son nom et prénom.

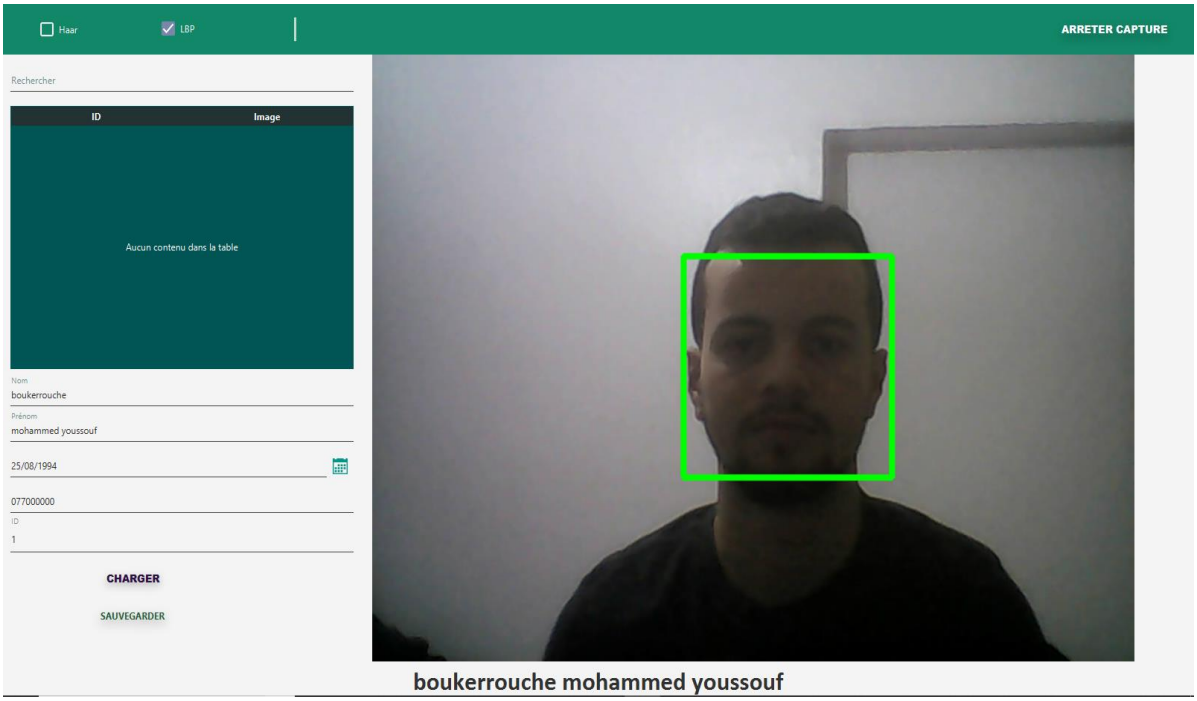


Figure 25 : Reconnaissance réussie

3.4 Détection de visage par open cv

3.4.1 Introduction :

Comme il a déjà été mentionné, la première étape dans la reconnaissance des visages est la détection des visages. Avec la bibliothèque OpenCV, il est assez facile de détecter un visage de face dans une image en utilisant son détecteur de visage Haar Cascade ou LBP Cascade étant donné un fichier ou une vidéo en direct, le détecteur de visage examine chaque emplacement d'image et classifie comme visage ou non visage. Le classifieur utilise des données stockées dans un fichier XML pour décider comment classifier chaque localisation image.

OpenCV est livré avec plusieurs classifieurs différents pour la détection de visage dans des poses frontales, ainsi que la détection de certains visages de profil, la détection des yeux, détection des corps...etc.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

3.4.2 Les étapes pour que OpenCV détecte un visage :

**** ouvrir le flux de camera***

On ouvre le flux caméra avec cvQueryframe

**** transformer le frame courante en image niveau de gris :***

```
// convert the frame in gray scale
Imgproc.cvtColor(frame, grayFrame, Imgproc.COLOR_BGR2GRAY);
// equalize the frame histogram to improve the result
```

Figure 26Transformation vers le niveau de gris

**** égaliser l'histogramme du cadre pour améliorer le résultat :***

```
// equalize the frame histogram to improve the result
Imgproc.equalizeHist(grayFrame, grayFrame);

// compute minimum face size (20% of the frame height
```

Figure 27 : Comparaison de 2 histogrammes

**** charger les données XML***

Pour charger des données XML, nous utilisons la fonction cvLoad() avec le chemin vers le fichier XML contenant cascade de Haar comme son premier paramètre d'entrée :

```
protected void haarSelected(Event event) {
    // check whether the lpb checkbox is selected and deselect it
    if (this.lbpClassifier.isSelected()) {
        this.lbpClassifier.setSelected(false);
    }

    this.checkboxSelection("/haarcascades/haarcascade_frontalface_alt.xml");
}
```

Figure 28 : Chargement de fichier XML pour la sélection de Haar

```
protected void lbpSelected(Event event) {
    // check whether the haar checkbox is selected and deselect it
    if (this.haarClassifier.isSelected()) {
        this.haarClassifier.setSelected(false);
    }

    this.checkboxSelection("/lbpcascades/lbpcascade_frontalface.xml");
}
```

Figure 29: Chargement de fichier XML pour la sélection de LBP

**** la détection avec le classifieur choisi***

Il ne reste plus qu'à effectuer une détection de visage sur l'image capturée, en utilisant Haar Cascade et retourner un rectangle pour la région détectée dans l'image donnée. La Fonction utilise en paramètre l'image récupérer pendant la boucle de traitement. C'est sur cette image que toutes les opérations vont être effectuées par la suite.

```
// detect faces
this.faceCascade.detectMultiScale(grayFrame, faces, 1.1, 2, 0 | Objdetect.CASCADE_SCALE_IMAGE,
        new Size(this.absoluteFaceSize, this.absoluteFaceSize), new Size());
```

Figure 30 : Lancement de la détection

****traçage de rectangle***

Nous avons maintenant l'ensemble des visages qui ont été détectés, et afin d'afficher le résultat à l'écran nous allons simplement dessiner un carré autour du visage détecté.

```
// each rectangle in faces is a face: draw them!
Rect[] facesArray = faces.toArray();
for (int i = 0; i < facesArray.length; i++) {
    Imgproc.rectangle(frame, facesArray[i].tl(), facesArray[i].br(), new Scalar(0, 255, 0), 1);
}
```

Figure 31 : Traçage de triangle

3.5 Prétraitement pour la reconnaissance faciale fisher (lda)

La reconnaissance de visage est un problème difficile en vision par ordinateur, la plupart des algorithmes de reconnaissance de visage sont extrêmement sensibles aux conditions d'éclairage. Pour simplifier la reconnaissance, il est extrêmement important d'appliquer diverses techniques de prétraitement d'image. Le prétraitement atténue les effets d'une différence des conditions lors des prises de vues.

C'est une phase importante dans le domaine globale d'identification. Elle augmente en général les performances du système. Il y a aussi d'autres problèmes, tels que le visage doit être dans une position très cohérente dans les images (tels que les yeux se trouvant dans les mêmes coordonnées de pixels), de taille uniforme, angle de rotation, émotion (sourire, colère, etc.)

Pour cela une réduction d'image est nécessaire dont l'opération est d'extraire seulement les paramètres essentiels pour l'identification et qui changent très peu avec le temps.

Les étapes de la réductions sont :

**** découpage :***

Le découpage de l'image consiste à conserver les maximums des variations intrinsèques du visage, et de supprimer les autres informations comme l'arrière plan, les cheveux, les cols de chemise, les oreilles et toutes les informations qui sont changeantes avec le temps.

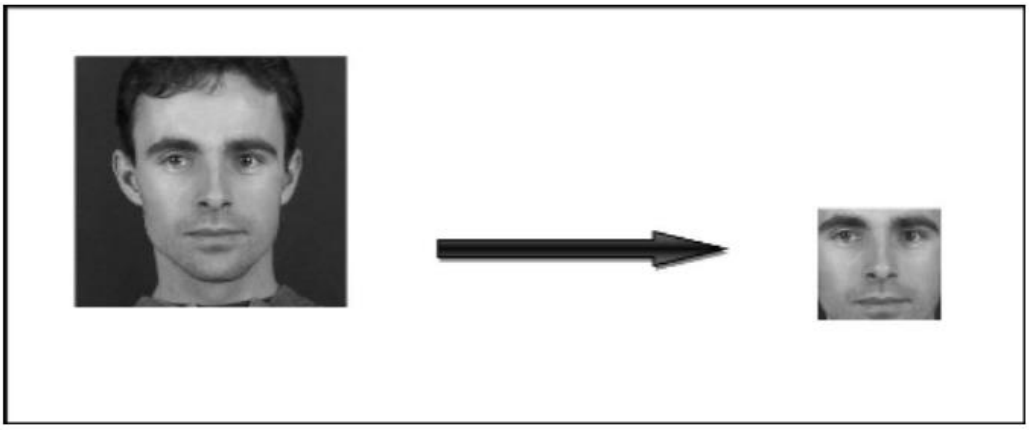


Figure 32 : Procédure de découpage d'un visage

**** Filtrage***

Pour améliorer la qualité visuelle de l'image, on doit éliminer les effets des bruits (parasites) en lui faisant subir un traitement appelé filtrage. Le filtrage consiste à modifier la distribution fréquentielle des composantes d'un signal selon des spécifications données.

****Décimations***

La décimation consiste à ne prendre qu'un pixel sur deux par exemple .Cela réduit bien entendu la résolution des images. Cette opération est précédée d'un filtrage passe bas, détruisant les hautes fréquences, de manière à respecter les conditions d'échantillonnage.

Figure 33 : Procédure de décimation d'un visage

**** Normalisation***

La normalisation permet d'assurer l'homogénéité des données. La photo-normalisation s'applique à une seule image. Alors que la normalisation s'applique à un groupe d'images, pour chaque composante, nous retirons la moyenne de cette composante pour toutes les images et nous divisons par dérivation standard.

Donc le prétraitement est une étape qui mène à une première réduction de donnée avant d'utiliser une deuxième étape de réduction comme par exemple la méthode d'analyse discriminante linéaire de fisher « LDA ».

3.6 Conclusion

L'objectif de ce mémoire est de concevoir et d'implémenter une application de reconnaissance faciale capable, en temps réel, de reconnaître les visages. Vu la quantité de logiciels potentiels (sécurité, réseaux sociaux,...) pouvant se baser sur cette application, celle- ci doit répondre à des exigences de rapidité et de robustesse des résultats.

Il existe une variété de techniques consacrées à la détection de visage. Les systèmes de suivi de visage se sont beaucoup développés ces dernières années grâce à l'amélioration du matériel et à la forte demande industrielle, par exemple pour la recherche d'individu par le biais d'une caméra de surveillance. Parmi les méthodes de base les plus utilisées on site :

Le traitement automatique du visage, Eigenface , L'analyse de points particuliers

LDA fisher (Linear discriminant analysis) ,filtre de Haar , lbp

Conclusion Générale et perspectives

L'objectif de ce mémoire est de concevoir et d'implémenter une application de reconnaissance faciale capable, en temps réel, de reconnaître les visages. Vu la quantité de logiciels potentiels (sécurité, réseaux sociaux,...) pouvant se baser sur cette application, celle- ci doit répondre à des exigences de rapidité et de robustesse des résultats.

Il existe une variété de techniques consacrées à la détection de visage. Les systèmes de suivi de visage se sont beaucoup développés ces dernières années grâce à l'amélioration du matériel et à la forte demande industrielle, par exemple pour la recherche d'individu par le biais d'une caméra de surveillance. Parmi les méthodes de base les plus utilisées on site :

Le traitement automatique du visage , Eigenface , L'analyse de points particuliers

LDA fisher (Linear discriminant analysis) ,filtre de Haar , lbp

Malgré tous les progrès qui ont été réalisés, les problèmes de pose et d'éclairage, et de l'identification dans des environnements extérieurs restent des challenges qui susciteront les efforts des chercheurs.

Parmi les perspectives ouvertes à ce projet, l'utilisation d'autres méthodes de reconnaissance, et la combinaison avec d'autre technologie biométriques comme l'empreinte digitale ou l'iris pour finaliser l'application multimodale et fiabiliser le système en diminuant la sensibilité aux conditions d'éclairage par de nouvelles normalisations

Rapport-Gratuit.com

Bibliographie

[1] reconnaissance faciale [en ligne] < https://labiometrie.wordpress.com/2017/02/12/reconnaissance-faciale/ >

[2] A. Nefian and M. Hayes. "Face Detection and Recognition Using Hidden Markov Models". In: International Conference on Image Processing (ICIP), pp. 141–145, 1998.

[3] Air journal>Actualitè [en ligne] < http://www.air-journal.fr/2017-10-15-bientot-des-tunnels-a-reconnaissance-faciale-a-dubai-5188994.html>

[4] New military applications for facial recognition technology [en ligne] < http://www.speroforum.com/a/HCESHCICAJ39/73073-Military-facial-recognition-technology-advances-in-Maryland#.WuMCn4huY2x >

[5] ("DeepFace", le nouveau système de reconnaissance faciale de Facebook qui fait froid dans le dos <https://www.huffingtonpost.fr/2014/03/20/deepface-reconnaissance-faciale-facebook_n_5000872.html>

[6] Cabal (Christian), 2003. Rapport sur les méthodes scientifiques d'identification

des personnes à partir de données biométriques et les techniques de mise en oeuvre.

Office parlementaire d'évaluation des choix scientifiques et technologiques.

Enregistré à la présidence de l'Assemblée nationale Le 16 juin 2003 sous N° 938.

70 pages R. Brunelli and T. Poggio, "Face recognition: Features vs. templates," IEEE Trans.

Pattern Anal. Mach. Intell., vol. 15, no. 10, pp. 1042–1053, Oct. 1993.

[7] reconnaissance faciale [en ligne]

< https://labiometrie.wordpress.com/2017/02/12/reconnaissance-faciale/ >)

[8] F. Perronnin and J.-L. Dugelay. "Introduction à la biométrie – Authentification des

individus par traitement audio-vidéo". Traitement du signal, Vol. 19, No. 4, 2002.)

[9] P. Jonathon Phillips and Hyeonjoon Moon and Syed A. Rizvi and Patrick J. Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 22, number 10, pages 1090-1104, 2000.

[10] M. Turk, A. Pentland, Eigenfaces for recognition. J. of Cognitive Neuroscience 3, 72–86,1991.

[11] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in Proc. IEEE Conf. Computer Vision Pattern Recognition, Jun. 1991, pp.586–591.

[12] P. Jonathon Phillips and Hyeonjoon Moon and Syed A. Rizvi and Patrick J. Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 22, number 10, pages 1090-1104, 2000

[13] S. Der Phillips J., P. J. Rauss. Feret recognition algorithm development and test results. Army Research Laboratory technical report, ARL-TR-995, 1996.

[14] P. Jonathon Phillips and Hyeonjoon Moon and Syed A. Rizvi and Patrick J. Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 22, number 10, pages 1090-1104, 2000

[15] Méthode de Viola et Jones https://fr.wikipedia.org/wiki/M%C3%A9thode_de_Viola_et_Jones

[16] http://www.milbo.org/muct/

[17] about OpenCv [en ligne] < https://opencv.org/about.html >

[18] Oracle Corporation [en ligne] < https://www.java.com/es/download/faq/javafx.xml >

[19] présentation Maven [institut d'électronique et d'informatique gaspard Monge ,Paris] < http://www-igm.univ-mlv.fr/~dr/XPOSE2003/site_maven/html/home.html >

[20] Apache Maven Project [site officielle] < http://maven.apache.org/ >

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES