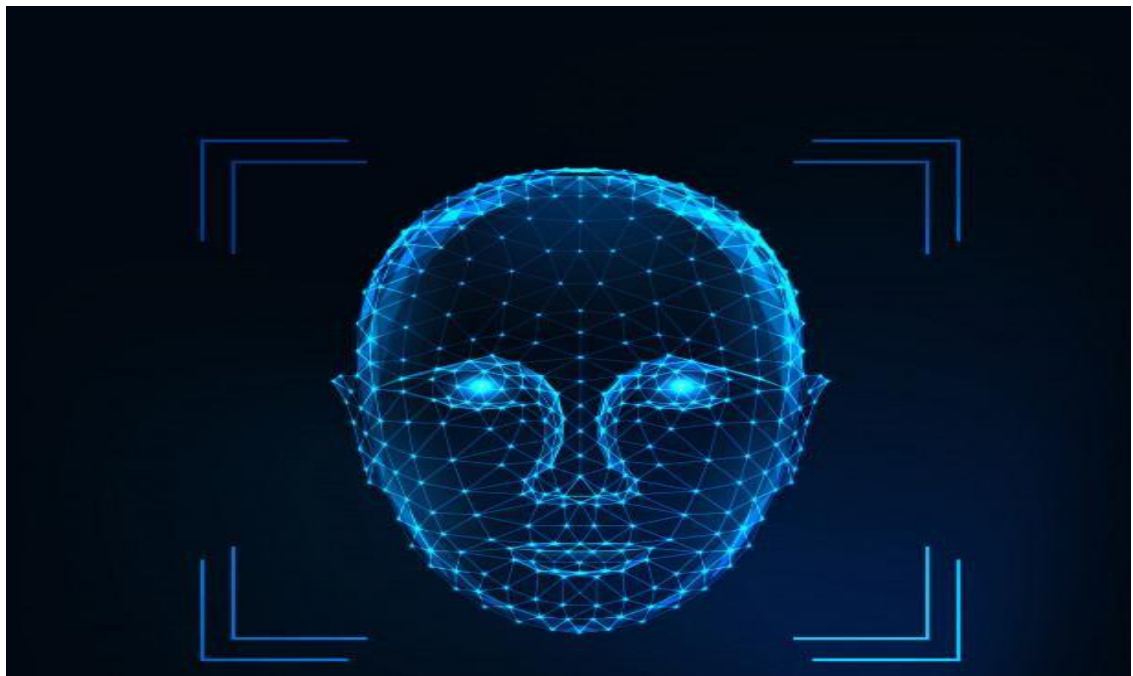


Projet de fin d'année

Filière : Génie informatique

Titre

Sujet : Réalisation d'une application de reconnaissance faciale : techniques de reconnaissance et enjeux de la sécurité informatique.



Encadrant :

Pr Bahri Abdelmalek
Pr ADDAM Mohamed

Réalisé par :

Asmae EL arrassi
Manal Lagnaoui
Mory moussa camara

Sujet : Réalisation d'une application de reconnaissance faciale : techniques de reconnaissance et enjeux de la sécurité informatique.

REMERCIEMENTS

Nous rendons tout d'abord grâce à notre seigneur, ALLAH de nous avoir donné la forme mentale que physique de pouvoir réaliser ce projet non négligeable. Dans un contexte, où la pandémie fait actualité.

Nous tenons à remercier notre encadrant M.Bahri Abdelmalek qui fut pour nous un encadrant attentif et disponible malgré ses charges nombreuses. Sa compétence, sa clairvoyance, sa ponctualité et son dynamisme nous ont beaucoup appris.

Je n'oserais oublier de remercier tout le corps professoral de l'école national des sciences appliquées Al hoceima ,pour le travail énorme qu'il effectue pour nous créer les conditions les plus favorables pour le déroulement de nos études .

Nous avons bénéficié d'une grande liberté dans notre travail et il nous a offert une ouverture vers beaucoup de domaines autre que l'informatique et le développement, ouverture que nous apprécions beaucoup, surtout sur l'imagerie. En bref, j'ai beaucoup appris de son contact.

Résumé

L'objectif de ce travail collectif que nous avons effectué dans ce cadre du projet de fin d'année, est le développement d'une application de reconnaissance faciale.

Alors notre application contient de différentes techniques de la détection des visages, des objets sur des images en temps réel. Due le développement de l'internet et la technologie, plusieurs administrations, entreprises et associations cherchent de plus en plus d'avoir une présence sur l'internet via des sites Web ou des applications Web. Cependant, les vulnérabilités de ces applications Web sont désormais le vecteur le plus important des attaques dirigées contre la sécurité des systèmes informatiques. On peut décrire notre application comme étant une porte intermédiaire pour rendre l'accès à un site de plus en plus sécurisé et rétreint.

Ce rapport s'articule autour de quatre chapitres principaux :

Dans le premier chapitre nous présenterons une étude bibliographique sur les aspects ciblés tant du côté théorique que pratique. Dans le deuxième chapitre, on exposera la spécification de notre application ainsi que les besoins auxquels elle doit répondre. Après la spécification détaillée des principaux besoins nous passerons à l'étape de conception. C'est une forme de synthèse de la spécification. Elle va nous permettre d'accéder à l'étape de réalisation en organisant les idées et en structurant le processus de codage suivant des diagrammes adéquats. Et après cette étape on pourra, aussi, mieux étudier la phase de développement et ainsi définir un plan de réalisation qui sera le plus conforme et convenable à nos différents besoins, ce qui nous permettra d'atteindre au mieux et le plus rapidement nos objectifs. Le dernier chapitre sera alors celui de Réalisation, dans ce chapitre, on décrira les outils de développement matériels et logiciels. Ensuite on présentera les principales interfaces de notre application ainsi que des interfaces.

Abstract

The objective of this collective work that we carried out as part of the end-of-year project, is the development of a facial recognition application.

This report is organized around four main chapters: In the first chapter we will present a bibliographic study on the targeted aspects both on the theoretical and practical side. In the second chapter, we will describe the specification of our application as well as the needs it must meet. After the detailed specification of the main requirements we will move on to the design stage. It is a form of summary of the specification. It will allow us to reach the realization stage by organizing the ideas and by structuring the coding process according to suitable diagrams. And after this stage we will also be able to better study the development phase and thus define an implementation plan that will be the most consistent and suitable to our different needs, which will allow us to achieve our objectives as quickly and as quickly as possible. The last chapter will then be that of Realization, in this chapter, we will describe the hardware and software development tools. Then we will present the main interfaces of our application as well as interfaces.

TABLE DES MATIERES

REMERCIEMENTS.....	4
RESUME.....	5
ABSTRACT.....	6
INTRODUCTION GENERALE.....	10
CHAPITRE 1 : ÉTAT DE L'ART.....	12
1.1 Introduction.....	13
1.2 La reconnaissance faciale.....	13
1.3 Historique.....	13
1.4 Les techniques de reconnaissance faciale.....	15
1.5 Objectif	16
1.6 Conclusion.....	16
CHAPITRE 2 : SPECIFICATION.....	18
2.1 Introduction.....	19
2.2 Besoins fonctionnels.....	19
2.3 Besoins non fonctionnels.....	20
2.4 Choix du langage de programmation.....	21
2.5 Les diagrammes des cas d'utilisation.....	22
2.6 Conclusion.....	24
CHAPITRE 3: CONCEPTION.....	25
3.1 Introduction.....	26
3.2 Conception globale.....	26
3.2.1 Eclipse	26
3.2.2 OpenCV.....	26

3.3	Conception détaillée.....	27
3.3.1	Diagramme de séquence <<Choisir la meilleure image>>.....	27
3.3.2	Diagramme de séquence <<Reconnaître les visages>>.....	28
3.3.3	Diagramme de séquence <<Gérer la base de données>>.....	29
3.3.4	Le diagramme de classe.....	30
3.4	Conclusion.....	31

CHAPITRE 4 : REALISATION.....32

4.1	Introduction.....	33
4.2	Environnement du travail.....	33
4.2.1	Environnement matériel.....	33
4.2.2	Environnement logiciel.....	33
4.3	Interfaces de l'application.....	34
4.3.1	Détection du visage.....	34
4.3.2	Enregistrement d'un visage.....	40
4.3.3	Reaconnnaissance d'un visage.....	40
4.3.4	Extraction des donnees et activation des reseaux sociaux.....	40
4.3.5	Détection des yeux.....	40
4.3.6	Détection du mouvement.....	41
4.3.7	Détection d'une forme rectangulaire.....	42
4.4	Conclusion.....	43

CONCLUSIONS ET PERSPECTIVES.....44

NETOGRAPHIE.....46

BIBLIOGRAPHIE.....47

ANNEXE.....50

TABLE DES FIGURES

Fig1. Principe de fonctionnement de base d'un système de reconnaissance faciale.....	19
Fig2. Diagramme de cas d'utilisation « Choisir la meilleure image ».....	22
Fig3. Diagramme de cas d'utilisation « Reconnaître les visages ».....	23
Fig4. Diagramme de cas d'utilisation « Gérer la base de données ».....	24
Fig5. Diagramme de séquence <<Choisir la meilleure image>>.....	28
Fig6. Diagramme de séquence <<Reconnaître les visages>>.....	29
Fig7. Diagramme de séquence <<Gérer la base de données>>.....	30
Fig8. Diagramme de classes.....	31
Fig9: Menu principal.....	34
Fig10. Détection du visage.....	36
Fig11. Enregistrement d'un visage.....	38
Fig12. Reconnaissance d'un visage.....	39
Fig13.Extraction des données.....	40
Fig14.Détection des yeux.....	41
Fig15.Détection des mouvements.....	42
Fig16.Détection d'une forme rectangulaire.....	43

INTRODUCTION GENERALE

De nos jours, on parle de plus en plus de l'insécurité dans divers secteurs ainsi que des moyens informatiques à mettre en œuvre pour contrer cette tendance : le contrôle d'accès aux ordinateurs, l'e-commerce, les opérations bancaires basées sur l'identification du demandeur, etc. Il existe traditionnellement deux manières d'identifier un individu. La première méthode est basée sur une connaissance a priori "knowledge-based" de la personne telle que, par exemple, la connaissance de son code PIN qui permet d'activer un téléphone portable.

La seconde méthode est basée sur la possession d'un objet "token-based". Il peut s'agir d'une pièce d'identité, d'une clef, d'un badge, etc. Ces deux modes d'identification peuvent être utilisés de manière complémentaire afin d'obtenir une sécurité accrue. Cependant, elles ont chacune leurs faiblesses. Dans le premier cas, le mot de passe peut être oublié par son utilisateur ou bien deviné par une autre personne. De plus, une personne sur quatre seulement fait l'effort d'appliquer les consignes de sécurité avant de retirer de l'argent (regarder derrière soi, cacher le clavier avec sa main lors de la saisie du code secret, etc.). Dans le second cas, le badge (ou la pièce d'identité ou la clef) peut être perdu ou volé. Les caractéristiques biométriques sont une solution alternative aux deux modes d'identification précédents [B2].

La biométrie humaine est la partie de la biologie qui analyse statistiquement les mesures faites sur l'Homme. Etymologiquement, biométrie humaine est synonyme d'anthropométrie. Les techniques appliquant la biométrie actuellement visent à établir l'identité d'une personne en mesurant une de ses caractéristiques physiques. Ces dernières sont variables, les unes plus fiables que d'autres, mais toutes doivent être infalsifiables et uniques pour pouvoir être représentatives d'un seul et même individu. D'autre part, comme nous allons le voir, les caractéristiques physiques sont loin d'être aussi parfaites et précises, qu'on pourrait le penser et l'on atteint très vite des limites pour ces techniques [B5].

Il existe plusieurs caractéristiques physiques uniques pour un individu, ce qui explique la diversité des systèmes appliquant la biométrie, citons quelques uns : L'empreinte digitale, La dynamique des signatures, l'iris, la rétine, la reconnaissance vocale et celle du visage. L'intérêt des applications

utilisant la biométrie se résume en trois classes : faciliter le mode de vie, éviter la fraude et contrôler la population [B10].

L'avantage de ces caractéristiques biométriques est d'être universelles, c'est-à-dire présentes chez toutes les personnes à identifier. D'autre part, elles sont mesurables et uniques : deux personnes ne peuvent posséder exactement la même caractéristique. Elles sont aussi permanentes ce qui signifie qu'elles ne varient pas ou peu au cours du temps [B18].

L'intérêt des applications utilisant la biométrie se résume en deux classes : faciliter le mode de vie, éviter la fraude [B11].

CHAPITRE 1 : ÉTAT DE L'ART

1.1 Introduction

En dépit de l'énorme progression en informatique et plus spécifiquement le traitement d'images, les systèmes appliquant les techniques de reconnaissance faciale sont considérés comme un gros « challenge » très intéressant, qui a attiré les chercheurs des différents domaines : psychologie, identification de modèles, réseaux de neurones, vision par ordinateur, infographie ...

1.2 La reconnaissance faciale

S'investir dans le domaine de la reconnaissance de visages est sans doute motivé par la multiplicité et la variété des champs d'applications de celui-ci (contrôle d'accès, télésurveillance et identification des criminels...). L'intérêt pour ce domaine s'explique aussi par le fait que par rapport aux autres filières de reconnaissance (reconnaissance de l'empreinte digitale, de la voix, etc.), la reconnaissance de visages n'a pas atteint un niveau aussi avancé que ces autres filières, elle souffre de plusieurs anomalies. Ceci s'explique avant tout par la complexité de la forme en question (le visage) comparée avec les autres formes [B1].

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 ce projet, on s'est focalisé sur la reconnaissance basée sur la vidéo, puisqu'elle est plus demandée [B3].

1.3 Historique

La reconnaissance faciale est une technique biométrique qui a une longue histoire. Si l'empreinte digitale est la technique biométrique la plus ancienne et la plus répandue et qui a été adoptée par Scotland Yard dès 1901 et par la préfecture de Police de Paris, dès 1903 pour rechercher les criminels, la reconnaissance des visages a été développée 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

mnésique de visages familiers ou non familiers, mais d'une épreuve consistant à appairer des photographies de visages non familiers présentés sous différents éclairages et selon des angles différents et nécessitant une bonne capacité d'intégration Visuo-spatiale [B12].

Actuellement cette technique est utilisée principalement pour des raisons sécuritaires lors des vols, d'identification et de contrôle d'accès : la comparaison se fait alors entre deux photos : l'une est extraite de l'enregistrement des caméras de surveillance, l'autre dite « image de référence ou image cible » correspondant au suspect, elle a été prise dans des conditions idéales et de bonne qualité [B20].

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 ce qui lui a donné des nouvelles applications, le contrôle de l'identité sous toutes ses formes [B4]:

- Contrôle d'accès des utilisateurs aux réseaux informatiques et à l'Internet.
- Contrôle d'accès de personnels aux zones contrôlées comme les aéroports.
- La recherche de criminels et des personnes disparues.

En plus, la reconnaissance faciale va permettre l'établissement de l'interaction entre machines et humains soit par la détection de l'identité, comme on l'a vu dernièrement avec le téléphone portable muni d'un système capable de reconnaître son utilisateur. Cette variabilité des applications a donné lieu à la disponibilité de produits commerciaux. Ces produits permettent [B15]:

- L'identification, par comparaison d'un visage avec ceux mémorisés à partir d'une base.
- La vérification, par comparaison des identités déclarées avec les identités associées aux visages mémorisés.
- La supervision, qui permet de suivre l'image d'une personne dans une séquence vidéo.
- La surveillance, qui permet de retrouver, en temps réel, une personne dans une séquence vidéo à partir d'une liste de visages.

L'exemple le plus connu d'utilisation à grande échelle de la reconnaissance faciale est celui de la ville de Newham, dans la banlieue de Londres. Il s'est vu décerner le trophée Big Brother Award en 1998 par l'organisation non gouvernementale Privacy International [B16].

Un autre exemple d'utilisation ayant eu un certain retentissement est celui de la surveillance et l'identification de personnes recherchées, par reconnaissance des visages dans une foule lors de la finale de foot-ball (Super Bowl) des USA, qui a eu lieu à Tampa en Floride, en janvier 2001. Cette utilisation a été fortement critiquée par l'ACLU (American Civil Liberty Union), qui a qualifié l'événement de Snooper Bowl, autrement dit de championnat d'espionnage [B16].

1.4 Les techniques de reconnaissance faciale

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.

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 de visage est la plus efficace dans des situations de capture d'image avec peu d'éclairage [B9].

- **Eigenface**

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 [B13].

- **L'analyse de points particuliers**

Elle est la technologie d'identification faciale la plus largement utilisée. Cette technologie se rapproche de Eigenface, mais elle est capable de s'adapter à des changements d'aspect facial (sourire, froncer des sourcils,...).

Les ingénieurs numériques l'utilisent souvent. Elle est comparable à Eigenface sauf qu'elle s'adapte à des changements d'expressions comme le sourire [B9].

- **Les réseaux de neurones**

Ils permettent en théorie de mener à une capacité accrue d'identification dans des conditions difficiles de capture. Les réseaux de neurones emploient un algorithme pour déterminer la similitude entre des captures d'images de visage. Par des algorithmes, le logiciel peut à partir d'une image de mauvaise qualité définir un visage le plus proche possible à partir d'une base de données [B17].

- **LDA (Linear discriminant analysis)**

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 [B9].

- **Filtre de Gabor**

Les filtres de Gabor déterminent l'ensemble des similarités perceptives entre les visages. Ils minimisent aussi néanmoins les incertitudes en transformant l'inégalité en égalité. Ainsi, on dit souvent que ce filtre propose un compromis temps-fréquence optimal [B9].

- **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 images tests [B9].

➤ La méthode la plus utilisée c'est la classification par caractéristiques de Haar

1.5 Objectif

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. Le but de ce projet de fin d'année est de détecter la forme des visages, les yeux, des formes rectangulaires, des mouvements et enfin de pouvoir reconnaître le visage déjà enregistré dans la base de données.

1.6 Conclusion

Dans ce chapitre nous avons présenté les grandes lignes de notre travail. Nous avons ainsi mis en valeur les points critiques à soigner et dégagé 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 : SPECIFICATION

2.1 Introduction

L'objectif de notre travail consiste à développer une application permettant la reconnaissance des visages des personnes à partir d'une séquence vidéo.

Tout d'abord, on décompose la vidéo en un ensemble d'images. Ensuite, à partir de cet ensemble on cherche le meilleur choix qui correspond à l'image dans laquelle l'algorithme de recherche a détecté le maximum des 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 on le compare à un seuil déjà fixé pour vérifier l'existence de la personne recherchée dans la base.

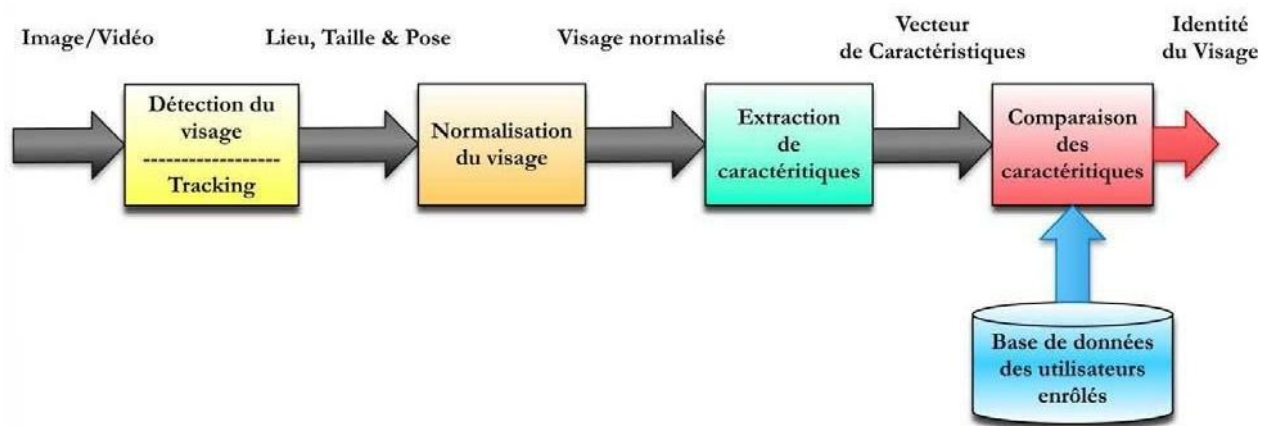


Fig1. Principe de fonctionnement de base d'un système de reconnaissance faciale

→ L'application que nous allons développer doit prendre en considération des besoins fournis en entrée. Dans ce qui suit, nous allons décoder les différents besoins fonctionnels et non fonctionnels que notre application doit satisfaire.

2.2 Besoins fonctionnels

- Récupérer la vidéo à analyser.
- Découper la séquence vidéo en un ensemble d'images.
- Détecter les différents éléments morphologiques dans chaque image de la vidéo.
- Enregistrer chaque image (avec ses objets morphologiques encadrés) dans l'entrepôt.
 - Choisir la meilleure image (qui contient le maximum des éléments morphologiques)
- Calculer les distances caractéristiques des images cibles et celles de test.

- Comparer l'image cible avec les images de test.
- Retourner le résultat de reconnaissance.
- Retourner le taux de ressemblance

2.3 Besoins non fonctionnels

- Fixer le seuil de reconnaissance.
- Configurer les chemins de nos ressources.
- Gérer la base.
- Rapidité du calcul des scores.
- Organisation et indexation de la base.
- L'utilité et l'utilisabilité de l'interface homme-machine.
- L'extensibilité de l'application.
- L'ergonomie de l'interface :
 - Les couleurs des zones de saisie, couleur de fond, images de fond...
 - La présentation des menus...
- La gestion des erreurs :

L'application doit gérer mieux ses exceptions par l'apparition d'un message d'alerte qui permettra de filtrer les données et de ne prendre en considération que les données qui correspondent aux types adéquats.
- Respect de l'ergonomie dans le choix des couleurs de l'IHM (Interface homme machine).

2.4 Choix du langage de programmation

C#	JAVA	Perl	MATLAB	C++
Langage interprété	Langage intermédiaire	Langage interprété	Langage interprété	Langage compilé
Utilisé pour les applications avec interfaces graphique.	Utilisé pour les applications avec interfaces graphiques et la programmation orientée internet	L'un des langages les plus utilisés pour la programmation de sites Web interactifs.	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.	Disponible sur plusieurs plateformes
Parmi les langages les plus rapides	Rapide	Souffrir d'une certaine lourdeur.	Rapide	Très rapide
orientée objet	procédural	orientée prototype et orientée objet	orientée prototype et orientée objet	orientée objet
Nombreuse bibliothèques	Nombreuse bibliothèques	Nombreuse fonctionnalités	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 facile.	Plateforme de développement à usage facile.	Plateforme de développement à usage facile. (Visual studio)
usage des pointeurs.	Pas de pointeur.	Pas de pointeur	Pas de pointeur	usage des pointeurs.
Permet les « partial class ».	Ne les permet pas.	Ne les permet pas.	Ne les permet pas.	Ne les permet pas.
Surcharge des opérateurs	Pas de surcharge des opérateurs	Surcharge des opérateurs	Surcharge des opérateurs	Surcharge des opérateurs

D'après le tableau comparatif ci-dessus, les deux langages de programmation les plus privilégiés sont JAVA et C++. Cependant, le langage C++ possède plus de souplesse pour la maintenance et le développement que le langage JAVA dans le domaine de l'imagerie et de l'embarqué.

Vu que nous sommes plus en lèse avec java, nous allons l'utiliser dans le projet

2.5 Les diagrammes des cas d'utilisation

Dans ce cas d'utilisation, l'utilisateur introduit la vidéo contenant la personne à reconnaître. Ensuite, cette dernière sera décomposée en images. Puis, à chaque image, nous appliquons une suite de traitements (filtre de Haar [A3]) afin de détecter les différents éléments morphologiques. Enfin, nous parcourons tous ses images et nous affichons la meilleure image (meilleur choix).

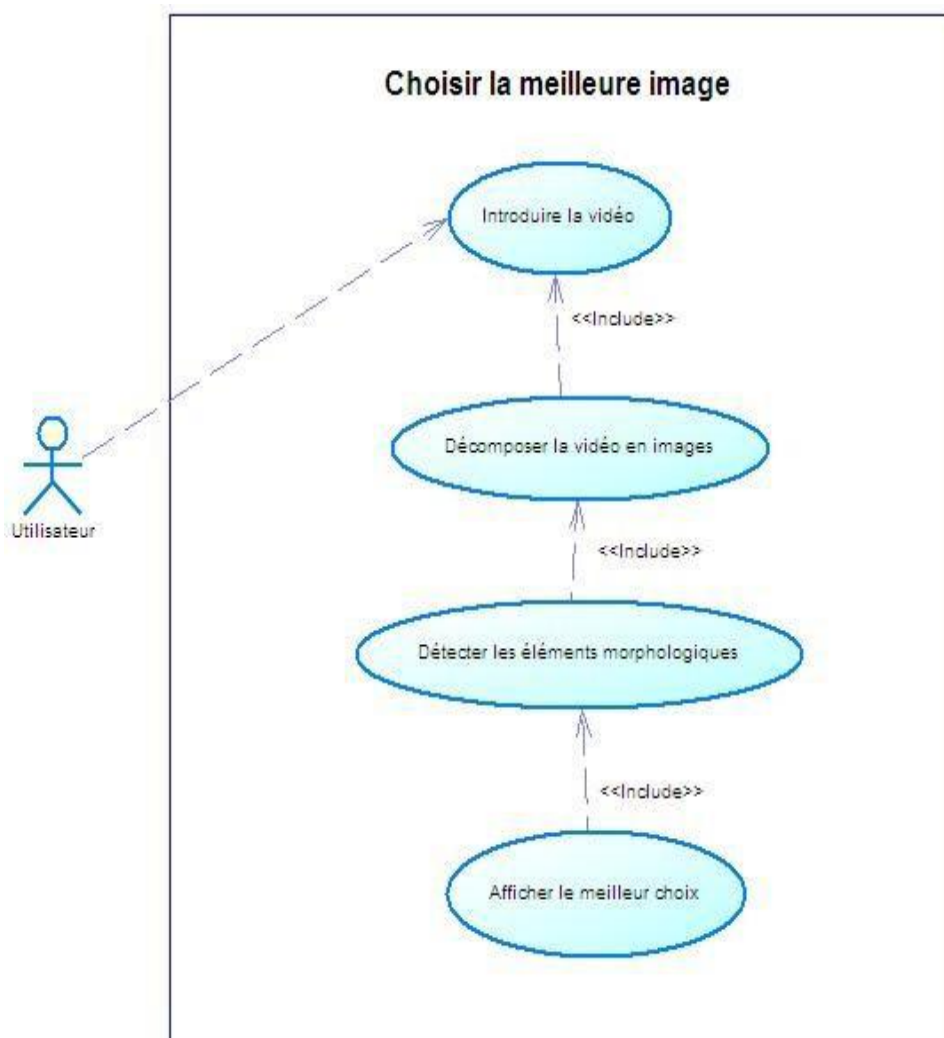


Fig2. Diagramme de cas d'utilisation « Choisir la meilleure image »

Dans ce cas d'utilisation, l'utilisateur ouvre la vidéo à analyser et lance la reconnaissance. 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 ces deux cas, l'application devrait retourner le taux de ressemblance.

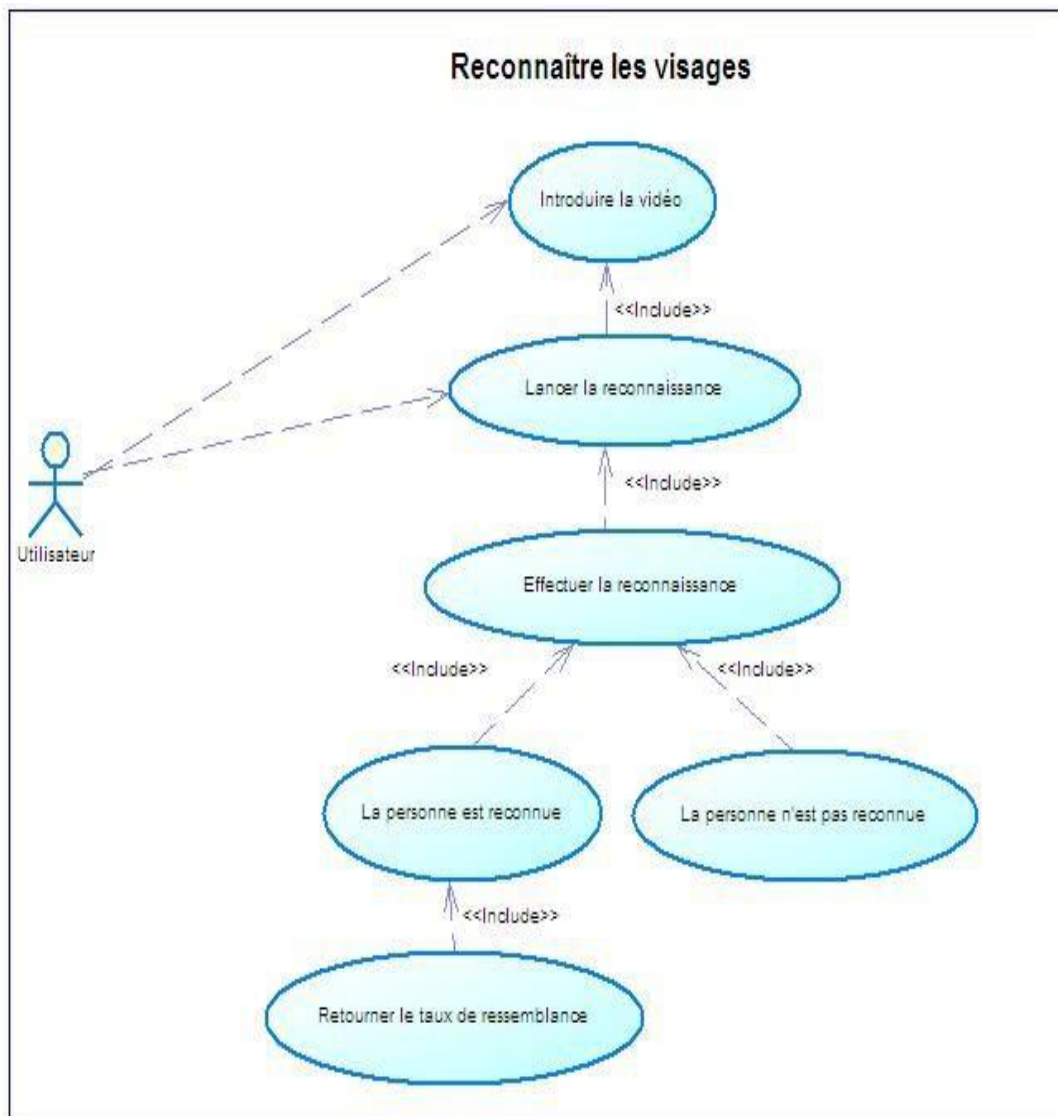


Fig3. Diagramme de cas d'utilisation « Reconnaître les visages »

Dans ce cas d'utilisation, l'utilisateur peut gérer la base de données en ajoutant ou en supprimant des données.

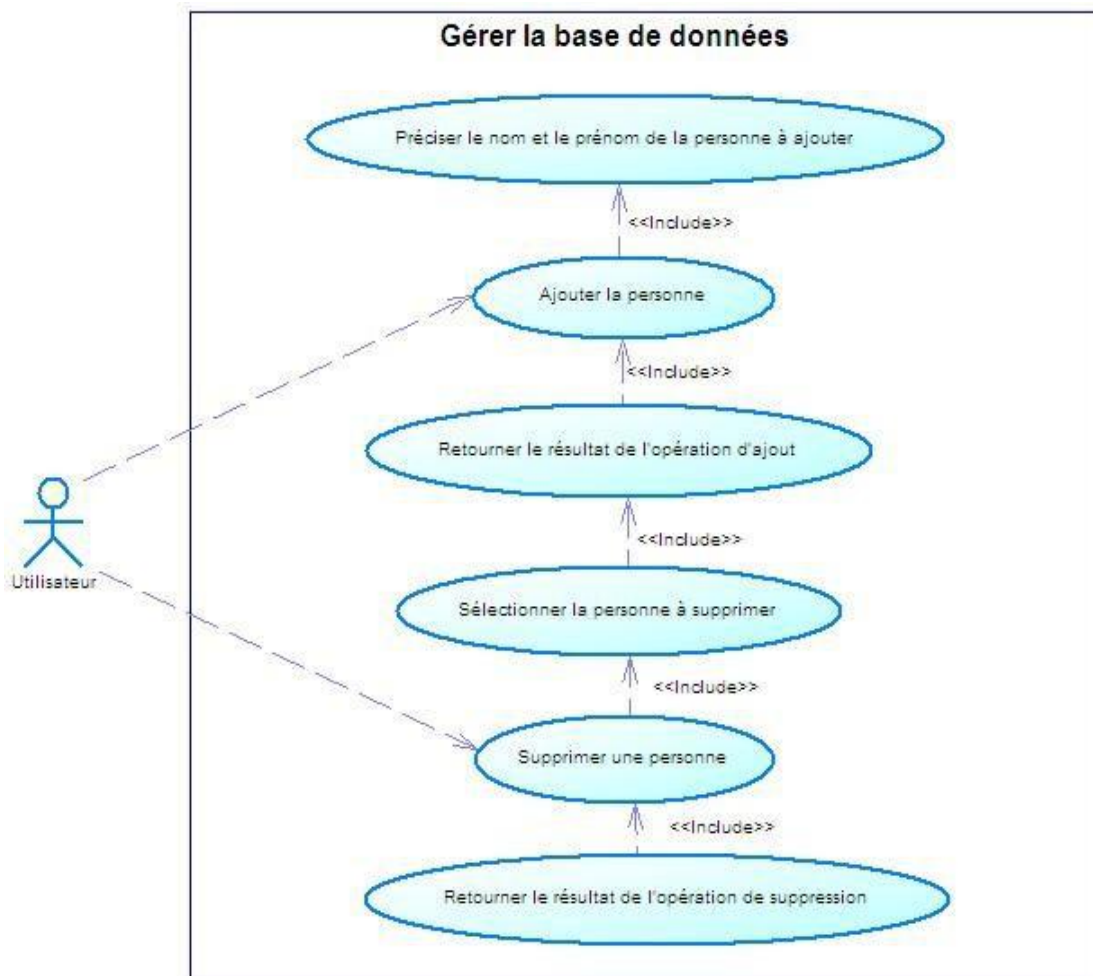


Fig4. Diagramme de cas d'utilisation « Gérer la base de données »

2.6 Conclusion

Préciser les objectifs et les fonctionnalités du système avant d'entamer la phase de conception semble une étape primordiale visant à reconnaître les frontières de l'utilisation, les services ainsi que les solutions que présente ce système à l'utilisateur.

Dans ce chapitre, nous avons détaillé la spécification en énumérant les différents besoins fonctionnels, en proposant des diagrammes de cas d'utilisation et les différents scénarios pour mieux comprendre le rôle et les fonctionnalités de notre application. Dans le prochain chapitre, nous allons aborder la conception de l'application développée.

CHAPITRE 3 : CONCEPTION

3.1 Introduction

Après la spécification détaillée des principaux besoins, nous passons à l'étape de conception qui permet de bien mener la phase d'implémentation. Nous allons, donc, choisir les outils de développement qui sont les plus adéquats avec nos différents besoins, ce qui va nous permettre d'atteindre au mieux et le plus rapidement nos objectifs.

Dans ce qui suit, nous décrirons les différentes orientations conceptuelles adoptées. Nous expliquerons l'architecture globale de notre application à travers des diagrammes.

3.2 Conception globale

Parmi les outils utilisés on cite :

3.2.1 Eclipse

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la [Fondation Eclipse](#) visant à développer un environnement de production de logiciels [libre](#) qui soit extensible, universel et polyvalent, en s'appuyant principalement sur [Java](#).

Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment [environnement de développement intégré](#) et [Framework](#)) mais aussi d'[AGL](#) recouvrant [modélisation](#), [conception](#), [test](#), [gestion de configuration](#), [reporting](#)... Son [EDI](#), partie intégrante du projet, vise notamment à supporter tout [langage de programmation](#) à l'instar de [Microsoft Visual Studio](#).

Bien qu'Eclipse ait d'abord été conçu uniquement pour produire des environnements de développement, les utilisateurs et [contributeurs](#) se sont rapidement mis à réutiliser ses briques logicielles pour des applications clientes classiques. Cela a conduit à une extension du périmètre initial d'Eclipse à toute production de logiciel : c'est l'apparition du Framework Eclipse [RCP](#) en 2004.

Figurant parmi les grandes réussites de l'[open source](#), Eclipse est devenu un standard du marché des logiciels de développement, intégré par de grands [éditeurs logiciels](#) et [sociétés de services](#). Les logiciels commerciaux [Lotus Notes](#) 8, [IBM Lotus Symphony](#) ou [WebSphere Studio Application Developer](#) sont notamment basés sur Eclipse.

3.2.2 OpenCV

OpenCV est une bibliothèque graphique libre, développée à la base par Intel, spécialisée dans le traitement d'image temps réel [A1].

3.2.3 OCR

La **reconnaissance optique de caractères** (ROC), en anglais *optical character recognition* (**OCR**), ou *océrisation*, désigne les procédés informatiques pour la traduction d'images de textes imprimés ou dactylographiés en fichiers de texte.

Un ordinateur réclame pour l'exécution de cette tâche un [logiciel](#) d'OCR. Celui-ci permet de récupérer le [texte](#) dans l'image d'un texte imprimé et de le sauvegarder dans un fichier pouvant être exploité dans un [traitement de texte](#) pour enrichissement, et stocké dans une base de données ou sur un autre support exploitable par un système informatique.

3.3 Conception détaillée

3.3.1 Diagramme de séquence <<Choisir la meilleure image>>

Dans ce diagramme de séquence, l'utilisateur, qui est l'acteur dans notre application, introduit la vidéo à analyser. Cette dernière sera décomposée en un ensemble d'images. Dans chacune d'elles, nous détectons les éléments morphologiques que nous enregistrons dans l'entrepôt avant de leur appliquer un algorithme pour récupérer la meilleure image.

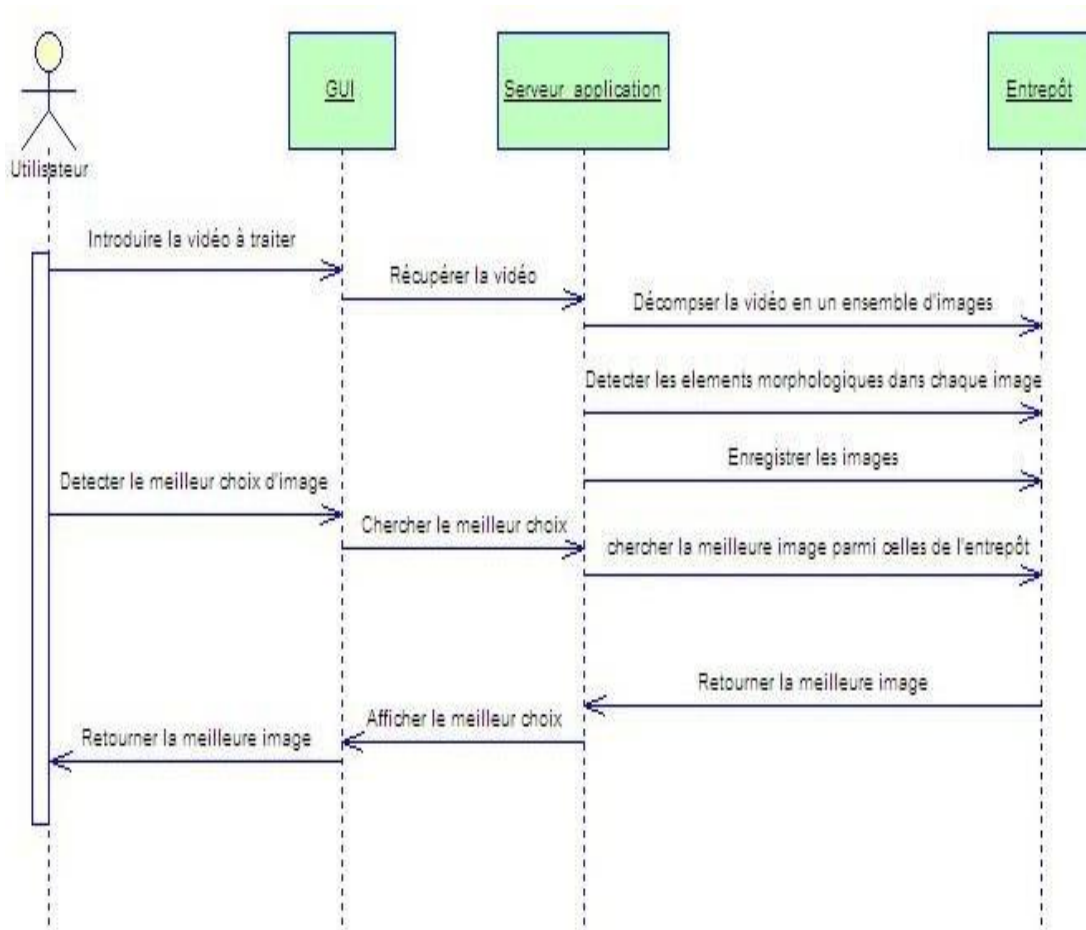


Fig5. Diagramme de séquence <<Choisir la meilleure image>>

3.3.2 Diagramme de séquence <<Reconnaître les visages>>

Dans ce diagramme de séquence, l'utilisateur introduit la vidéo à analyser, lance l'algorithme de reconnaissance et l'application décide, suivant la valeur de différence comparée au seuil déjà fixé, si la personne existe dans la base de données ou pas. Si oui, il affiche le taux de ressemblance.

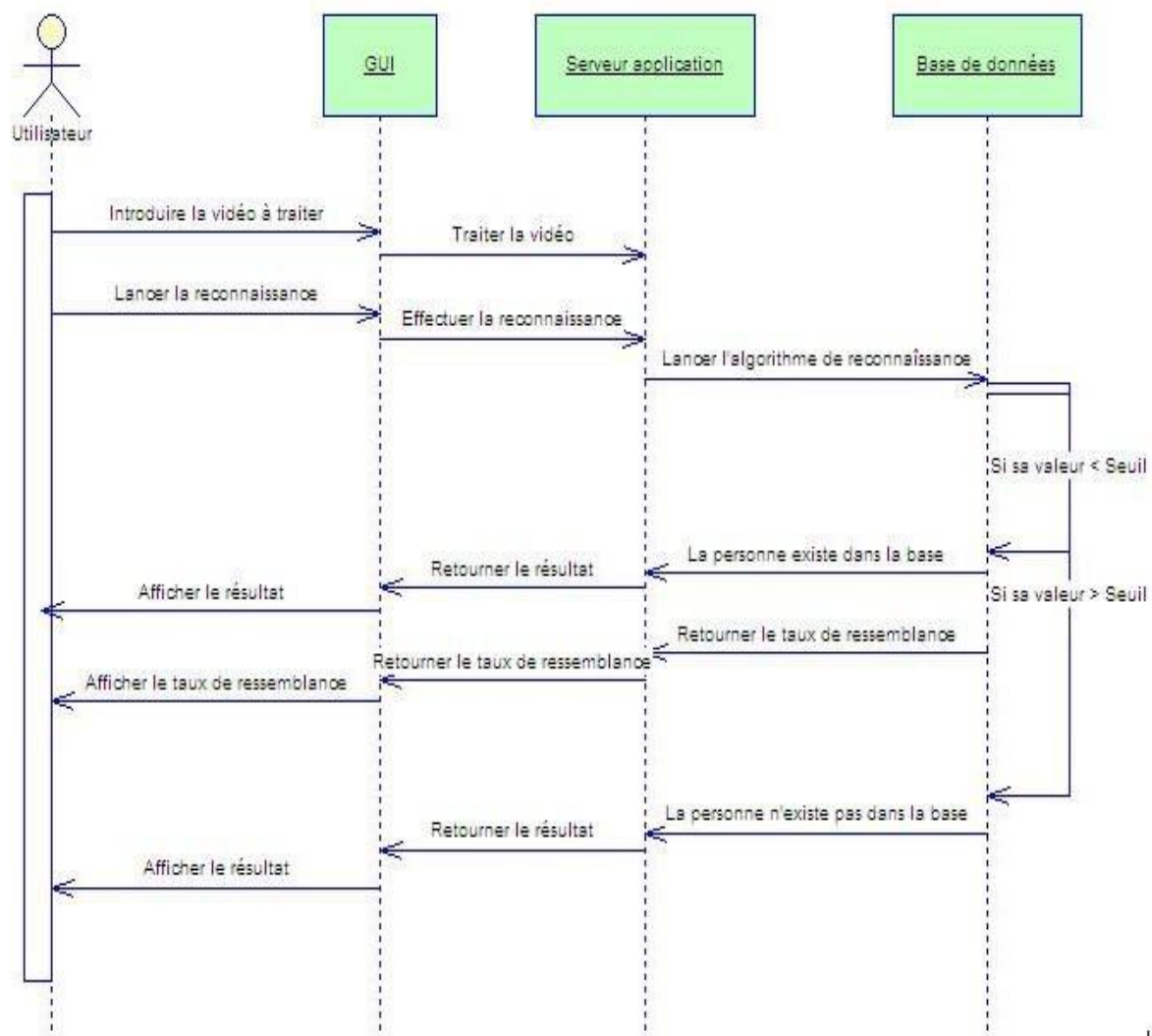


Fig6. Diagramme de séquence <<Reconnaître les visages>>

3.3.3 Diagramme de séquence <<Gérer la base de données>>

Dans ce diagramme de séquence, l'utilisateur peut accéder à la base et modifier son contenu en ajoutant ou en supprimant des personnes selon les besoins de l'application.

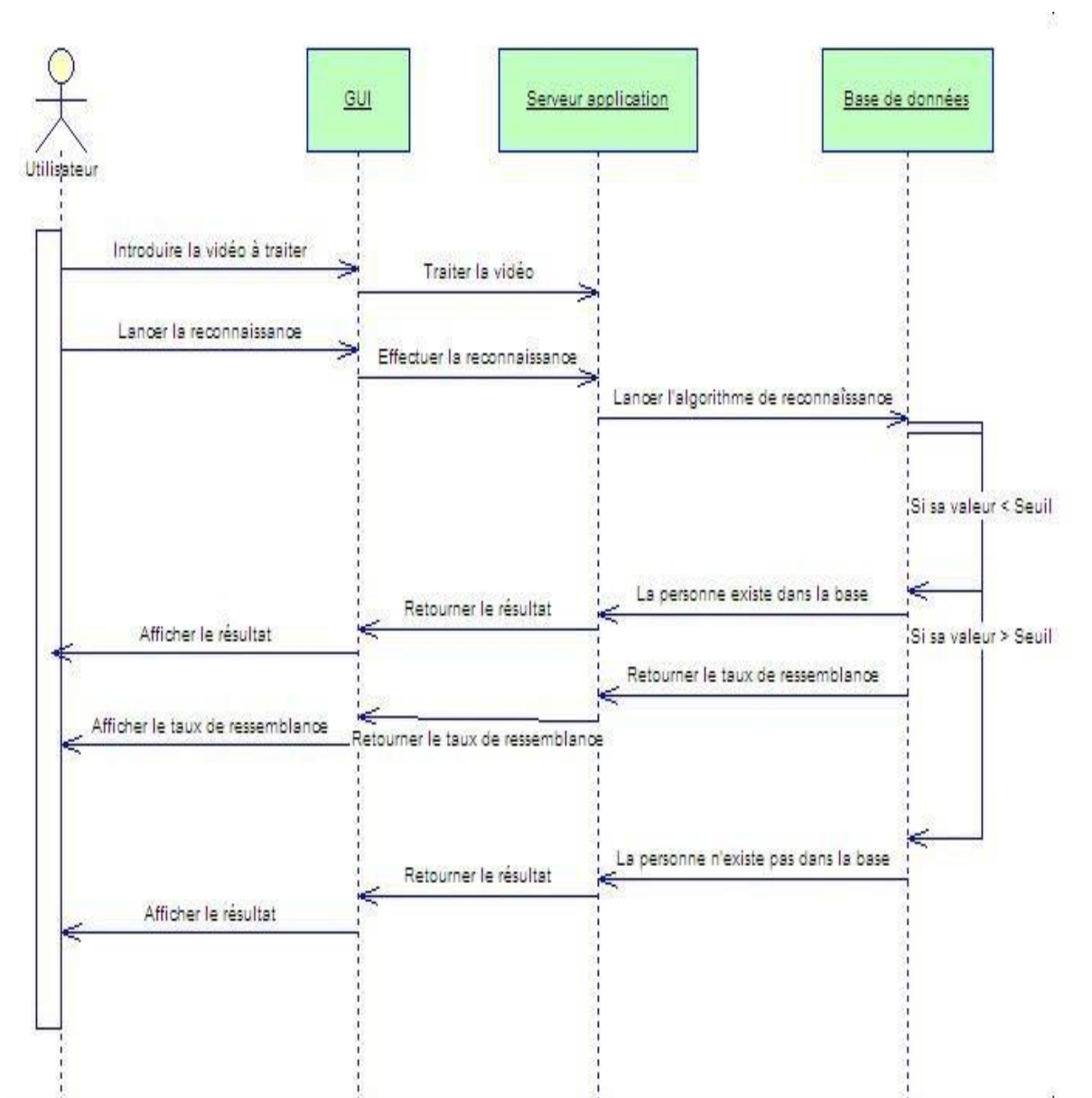


Fig7. Diagramme de séquence <<Gérer la base de données>>

3.3.4 Le diagramme de classe

Le diagramme de classe contient illustre la conception de l'application comme l'indique le schéma ci-dessous.

Diagramme de classes

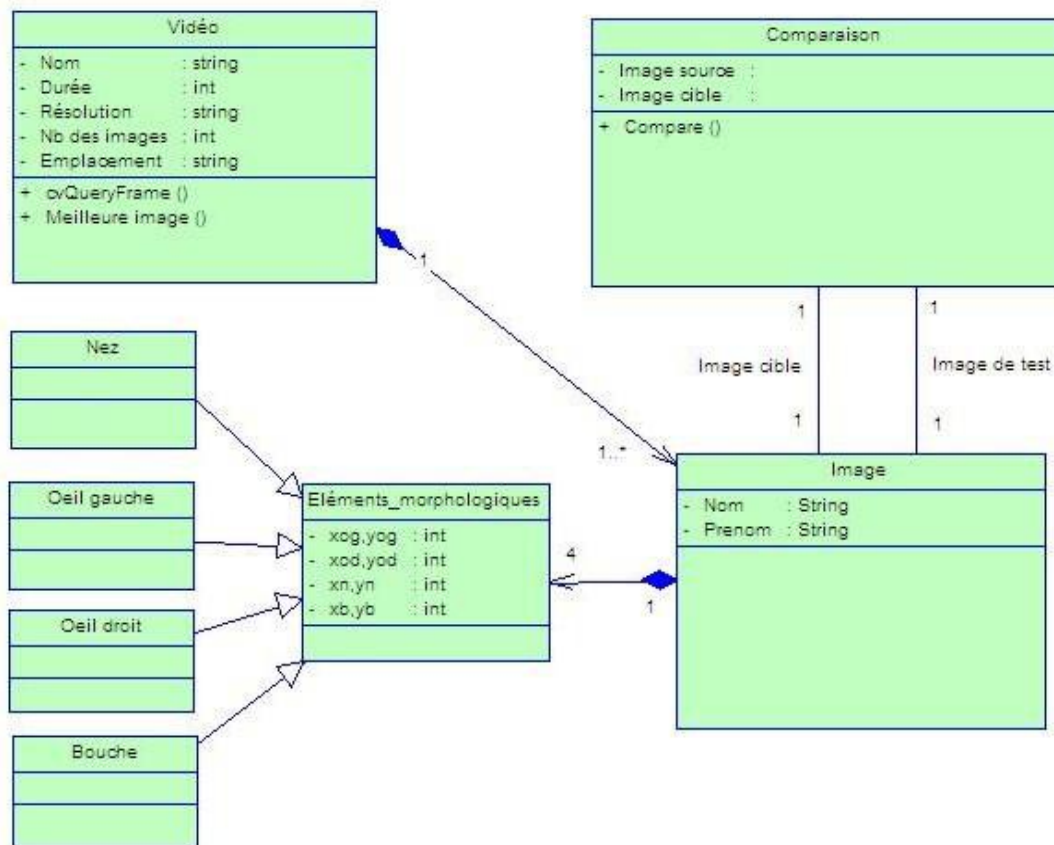


Fig8. Diagramme de classes

3.4 Conclusion

À la fin de ce chapitre nous avons pu modéliser le fonctionnement du système. Ceci est fait à travers différents diagrammes permettant de bien spécifier la composition et le comportement de l'application. Pour le chapitre suivant, on procède à la réalisation et la transformation de ces modèles afin de mettre en œuvre la solution proposée.

CHAPITRE 4 : REALISATION

4.1 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 matériel et logiciel utilisés pour développer notre projet de fin d'études.

4.2 Environnement du travail

Dans cette section, nous présenterons les environnements matériel et logiciel de notre travail.

4.2.1 Environnement matériel

Afin de mener à bien ce projet, il a été mis à notre disposition un ensemble de matériels dont les caractéristiques sont les suivantes :

Un ordinateur pour chacun des 3 membres du groupe avec des caractéristiques différentes.

4.2.2 Environnement logiciel

–Logiciel de conception :

Power AMC Evaluation

StarUml

–Logiciel de développement:

Eclipse

MySQL (base donnée)

JavaFX

Zoom

Github

–Bibliothèque graphique:

Open CV (Open Computer Vision) [A1]

OCR

4.3 Interfaces de l'application

Cette interface présente le menu principal de notre application, il contient plusieurs buttons, chacun a son propre fonctionnalité. Seul le bouton demander (en vert) qui est cliquable. pour pouvoir démarrer automatiquement la détection du visage .

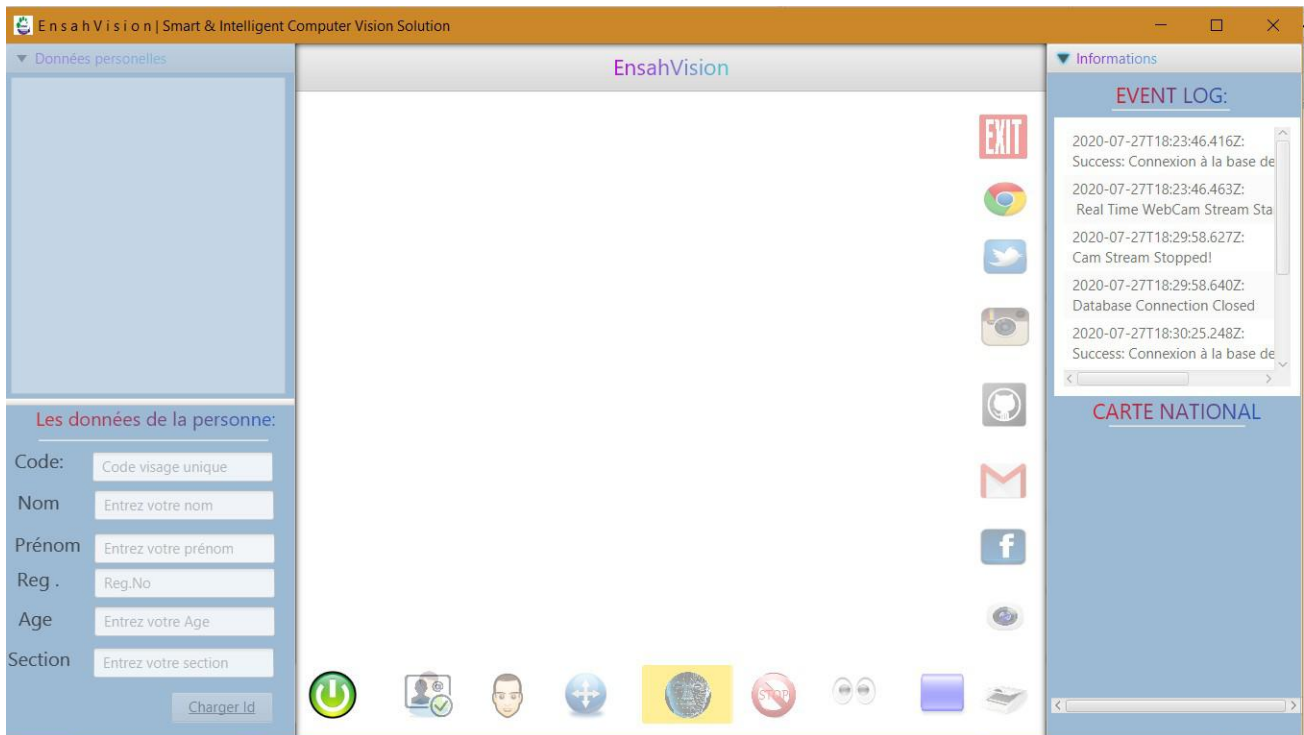


Fig9 : Menu principal

4.3.1 Détection du visage

Une fois le bouton démarrer est appuyé, la camera du pc s'annule automatiquement, du coût notre visage s'affiche dans le rectangle situé au milieu, et la détection de visage est déclenchée. Donc si on est face à la webcam, notre visage est détecté avec un rectangle vert.

-Pour ouvrir la camera, l'application fait appelle à la méthode « **startCamera** » :

@FXML

```
protected void startCamera()
{
    originalFrame.setFitWidth(600);
    originalFrame.setPreserveRatio(true);
    if (!this.cameraActive)
    {
        this.haarClassifier.setDisable(true);
        this.lbpClassifier.setDisable(true);
        this.capture.open(0);

        if (this.capture.isOpened())
        {
            this.cameraActive = true;
            Runnable frameGrabber = new Runnable()
            { @Override
              public void run()
              {
                  Image imageToShow = grabFrame();
                  originalFrame.setImage(imageToShow);
              }
            };

            this.timer = Executors.newSingleThreadScheduledExecutor();
            this.timer.scheduleAtFixedRate(frameGrabber, 0, 33,
TimeUnit.MILLISECONDS);

            this.cameraButton.setText("Stop Camera");
        }
        else
        {
            System.err.println("Failed to open the camera connection...");
        }
    }
}
```

-Pour la détection de visage :

```
if (faces != null) {

    g2.setColor(Color.green);
    g2.setStroke(new BasicStroke(2));
    int total = faces.total();

    for (int i = 0; i < total; i++) {

        //printing rectangle box where
        face detected frame by frame
        CvRect r = new
CvRect(cvGetSeqElem(faces, i));
(r.y() * 4), (r.width() * 4), (r.height() * 4));

        CvRect re = new CvRect((r.x()
* 4), r.y() * 4, (r.width() * 4), r.height() * 4);

        cvSetImageROI(temp, re);

        // File f = new

        org = new CvPoint(r.x(),
r.y());
    }
}
```

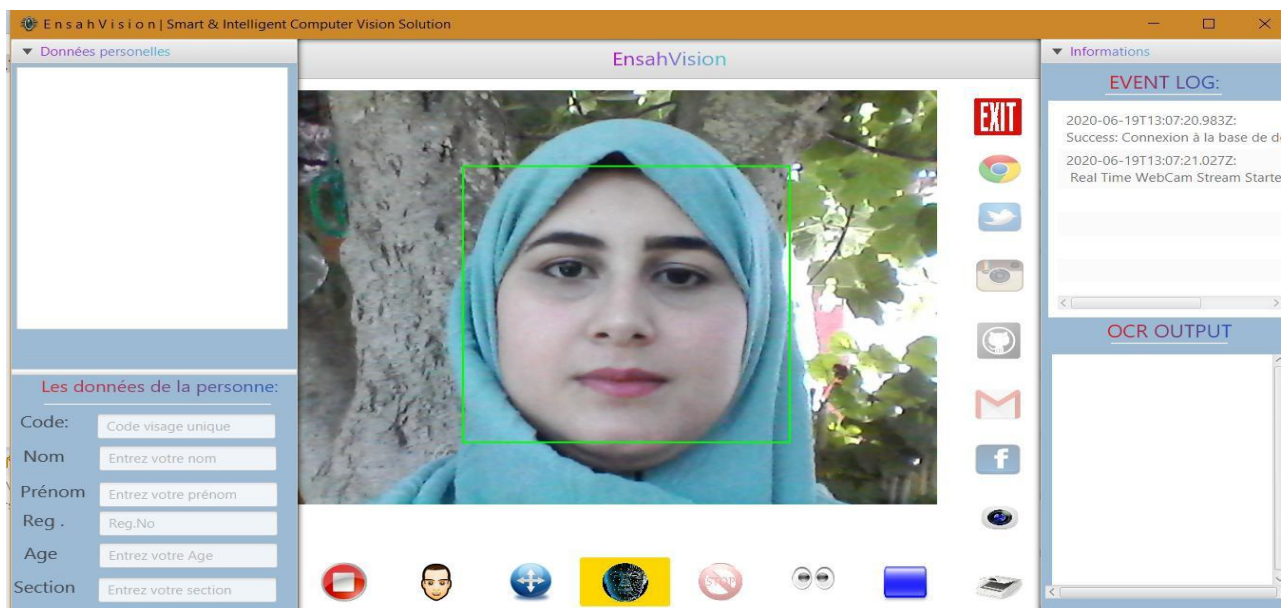


Fig10. Détection du visage

4.3.2 Enregistrement d'un visage

Cette partie concerne l'enregistrement d'un visage. La première chose à faire c'est de remplir le champ des informations personnelles et charger la carte d'identité, sinon un message d'erreur est affiché pour vous dire de remplir toutes les données personnelles et charger l'Id avant de faire la capture. Après le remplissage des informations, on prend des captures. On clique sur valider l'inscription si les informations sont toutes compatibles donc l'inscription est terminée sinon un message d'erreur va être affiché.

Nous vous conseillons de prendre plus de 5 photos pour que la reconnaissance soit plus précise.

-Pour enregistrer le visage, l'application fait appel à la méthode « **saveFace** »:

```

@FXML
protected void saveFace() throws SQLException
{
    supprimeruser.setVisible(false);
    //Input Validation

    System.out.println(fname.getText()+" "+lname.getText()+" "+faceDetect.getRecogniseCode());

    System.out.println(faceDetect.getNom()+" "+faceDetect.getPrenom()+" "+faceDetect.getCode1());

    if ( faceDetect.getNom().equalsIgnoreCase(fname.getText()) &&
        faceDetect.getPrenom().equalsIgnoreCase(lname.getText()) &&
        faceDetect.getCode1()==Integer.parseInt(code.getText())) {
        new Thread() -> {
            try {
                //Progressbar
                pb.setVisible(true);
                savedLabel.setVisible(true);
                File ancien = new File(path+"\\ID\\id"+cpt+".jpg");
                File nouveau = new File(path+"\\ID\\id"+faceDetect.getCode1()+".jpg");

                if(ancien.renameTo(nouveau))
                    System.out.println("Le fichier "+ancien+" a bien été renommé.");
                else
                    System.out.println("Echec! Le fichier "+ancien+" n'a pas pu être renommé.");

                database.setFname(fname.getText());
                database.setLname(lname.getText());
                database.setAge(Integer.parseInt(age.getText()));
                database.setCode(Integer.parseInt(code.getText()));
                database.setSec(sec.getText());
                database.setReg(Integer.parseInt(reg.getText()));
                database.insert();
                faceDetect.setCount(faceDetect.getCount()+1);
                javafx.application.Platform.runLater(new Runnable(){

                    @Override
                    public void run() {
                        pb.setProgress(100);
                    }
                });
                savedLabel.setVisible(true);
                Thread.sleep(2000);
                javafx.application.Platform.runLater(new Runnable(){

                    @Override
                    public void run() {
                        pb.setVisible(false);
                    }
                });

                javafx.application.Platform.runLater(new Runnable(){

```

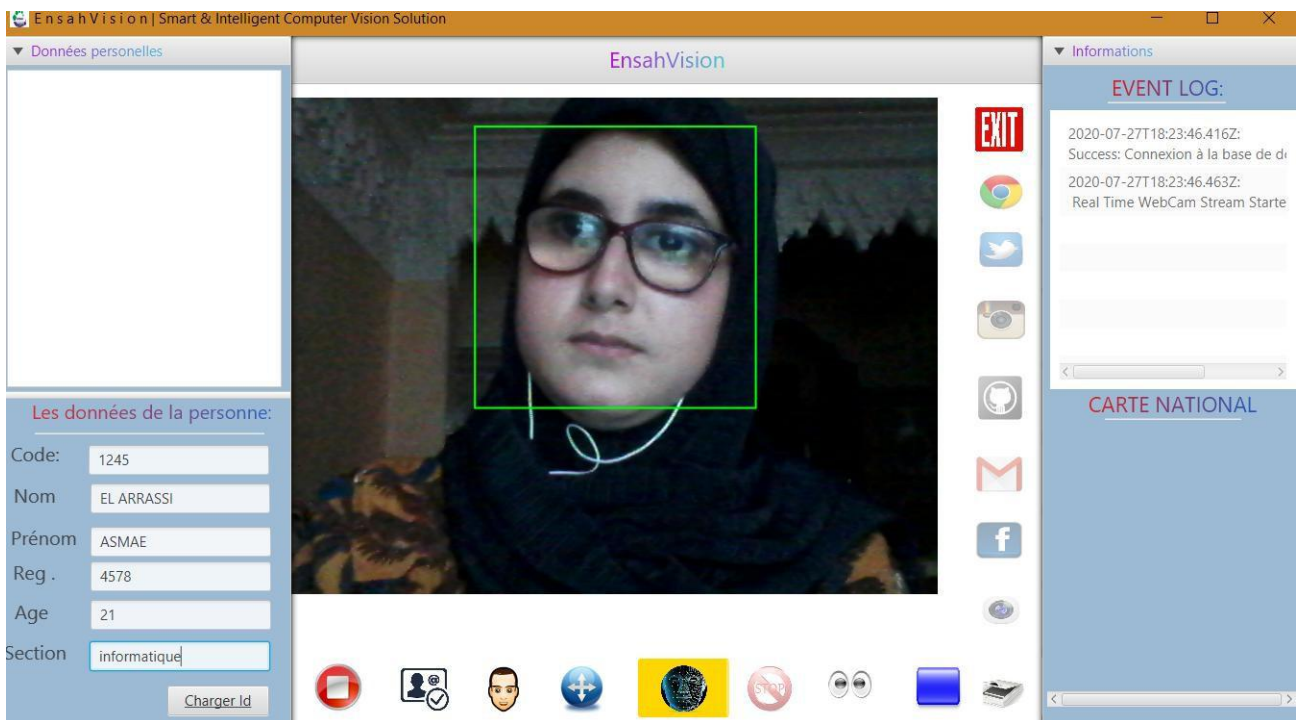


Fig11. Enregistrement d'un visage

4.3.3 Reconnaissance d'un visage

Après avoir faire l'enregistrement d'un visage, l'application est capable maintenant de reconnaitre automatique le visage, est votre nom et prénom seront afficher dans le rectangle vert.

-Pour la reconnaissance voici le code :

```
if(recogniseCode != -1)
{
    database.init();
    user = new ArrayList<String>();
    user = database.getUser(this.recogniseCode);
    this.output = user;
    names = user.get(1) + " " + user.get(2);
    if (showId) {
        try {
            file = new File(path+"\\ID\\id"+recogniseCode+".jpg");
            image2 = ImageIO.read(file);
            SampleController.imageID.setFitHeight(240);
            SampleController.imageID.setFitWidth(251);
            WritableImage showFrame1 = SwingFXUtils.toFXImage(image2,
            null);
            SampleController.imageID.setImage(showFrame1);
            SampleController.imageID.setVisible(true);

            showId=false;
        }
    }
}
```

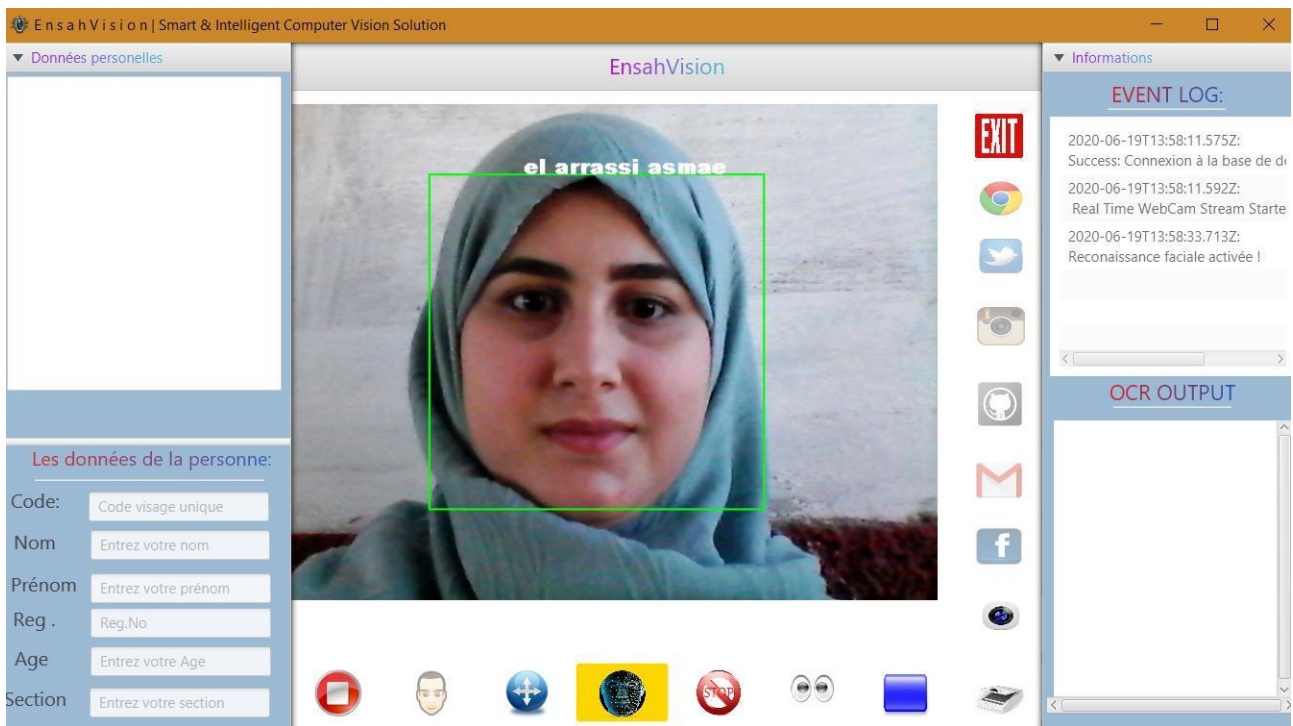


Fig12. Reconnaissance d'un visage

4.3.4 Extraction des données et activation des réseaux sociaux

Une fois la personne est connue par l'application, on peut extraire ses données en cliquant sur le même bouton, on peut les consulter dans la case « **Données personnelles** ». En plus de ça les boutons réseaux sociaux et celui de supprimer sont accessibles par l'utilisateur.

-Pour le code qui génère ce bouton :

```

if (saveId) {
    //extraire données de la carte national
    File file = new File(path+"\\ID\\id"+SampleController.cpt+".jpg");
    BufferedImage img1=ImageIO.read(file);
    ToIplImage iplConverter = new OpenCVFrameConverter.ToIplImage();
    Java2DFrameConverter java2dConverter = new Java2DFrameConverter();
    iplImage = iplConverter.convert(java2dConverter.convert(img1));
    IplImage grabId=cvCreateImage(cvGetSize(iplImage), 8, 1);
    cvCvtColor(iplImage,grabId,CV_BGR2GRAY);
    cvSaveImage("IDgray/idgray.jpg",grabId);
    String[] dRecuperer=OCR.init(path+"\\IDgray\\idgray.jpg").split("\n");
    //Recuperer nom et prenom

    String[] dRecupererPrenom=dRecuperer[3].split(" ");
    String[] dRecupererNom=dRecuperer[5].split("[ \\W]");
    dRecupererNom1 = dRecupererNom[1].replaceAll("[^A-Za-z]+", "");
    Prenom=dRecupererPrenom[0];
    Nom="";

    //comparer nom et prenom

    if(dRecupererNom[1].equals(dRecupererNom1))
    { Nom=dRecupererNom[0]+"
+dRecupererNom[1];

    else
    { Nom=dRecupererNom[0]
];

    System.out.println(fname+" "+Lname);

```

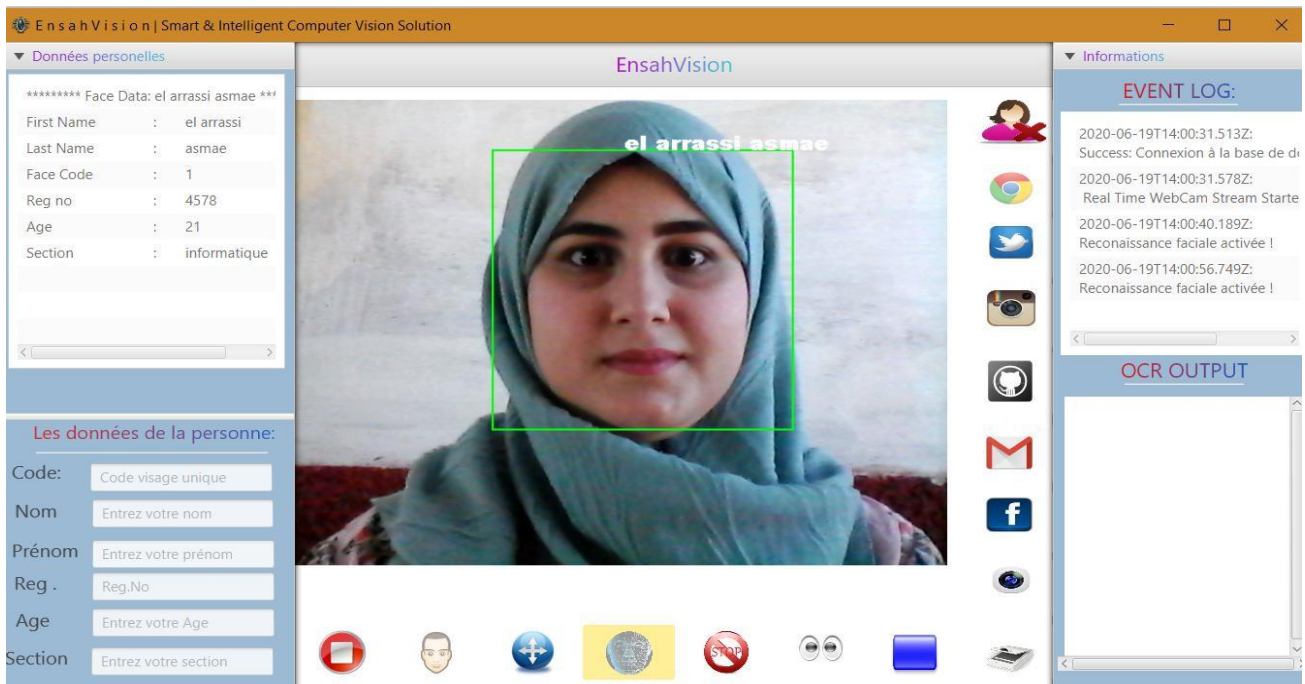



Fig13. Extraction des données

4.3.5 Détection des yeux

Cette fonction permet de détecter les yeux d'une personne, avec deux cercle entourant nos yeux .On peut le désactiver en cliquant sur le même bouton.

-Pour le code qui génère cette fonctionnalité :

```
if (isEyeDetection) { //eye detection logic
    eyes = cvHaarDetectObjects(smallImage, classifierEye, storage, 1.1, 3,
    CV_HAAR_DO_CANNY_PRUNING);
    if (eyes.total() == 0) {
        eyes = cvHaarDetectObjects(smallImage, classifierEyeglass, storage,
        1.1, 3, CV_HAAR_DO_CANNY_PRUNING);
    }
    printResult(eyes, eyes.total(), g2);
}
```

```
@FXML
protected void startEyeDetect()
{
    supprimeruser.setVisible(false);
    if(eye_count > 0)
    {
        faceDetect.setEyeDetection(false);
        eye_count = 0;
    }
    else {
        faceDetect.setEyeDetection(true);
        eye_count++;
    }
}
```

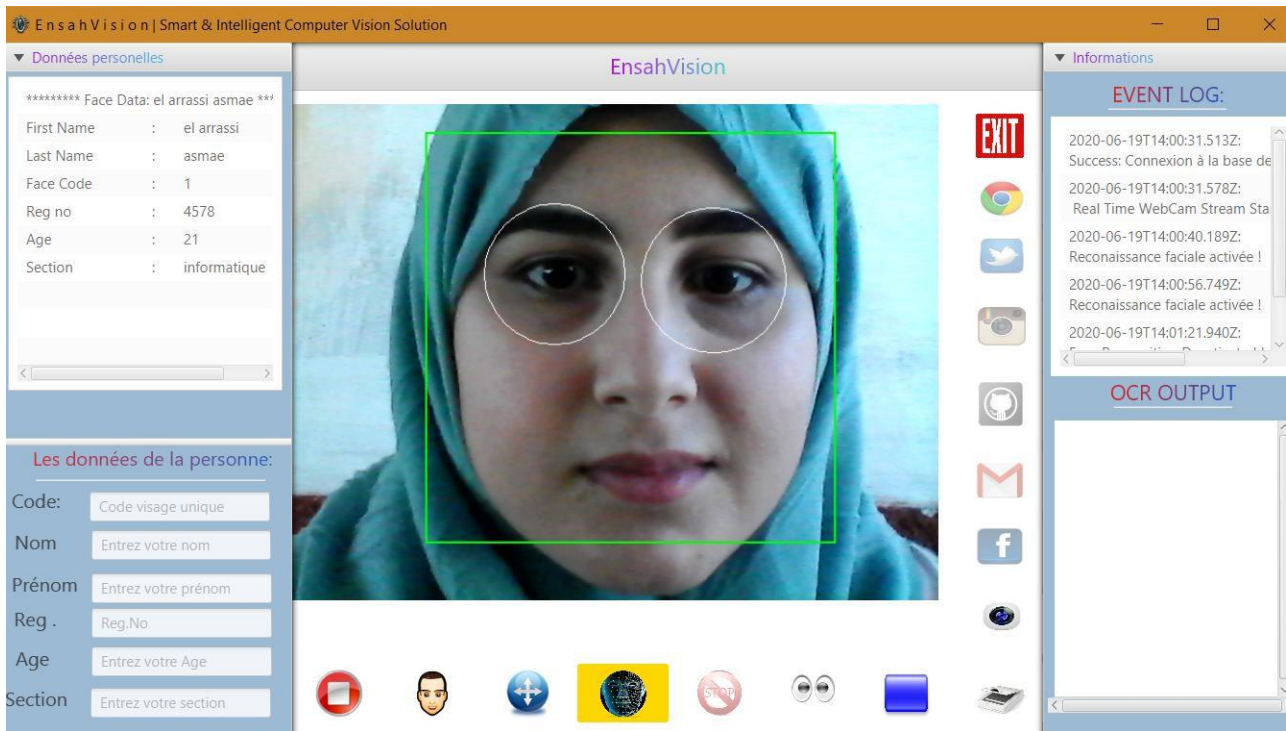



Fig14.Détection des yeux

4.3.6 Détection du mouvement

Si on déclenche le bouton bleu avec des flèches, une autre fenêtre noire ouvrira. Ce dernier détecte tous les mouvements de la personne ou d'un objet.

-Pour le code qui génère ce bouton :

```
if (isMotion) {
    new Thread(() -> {
        try
        { motionDetector.init(grabbedImage,
            g2);

            } catch (InterruptedException ex) {
            } catch (Exception e) {

                e.printStackTrace();
            }
        }).start();
    }
}
```

```
@FXML
protected void startMotion() {

    supprimeruser.setVisible(false);
    faceDetect.setMotion(true);
    putOnLog("motion Detector Activated !");
}
```

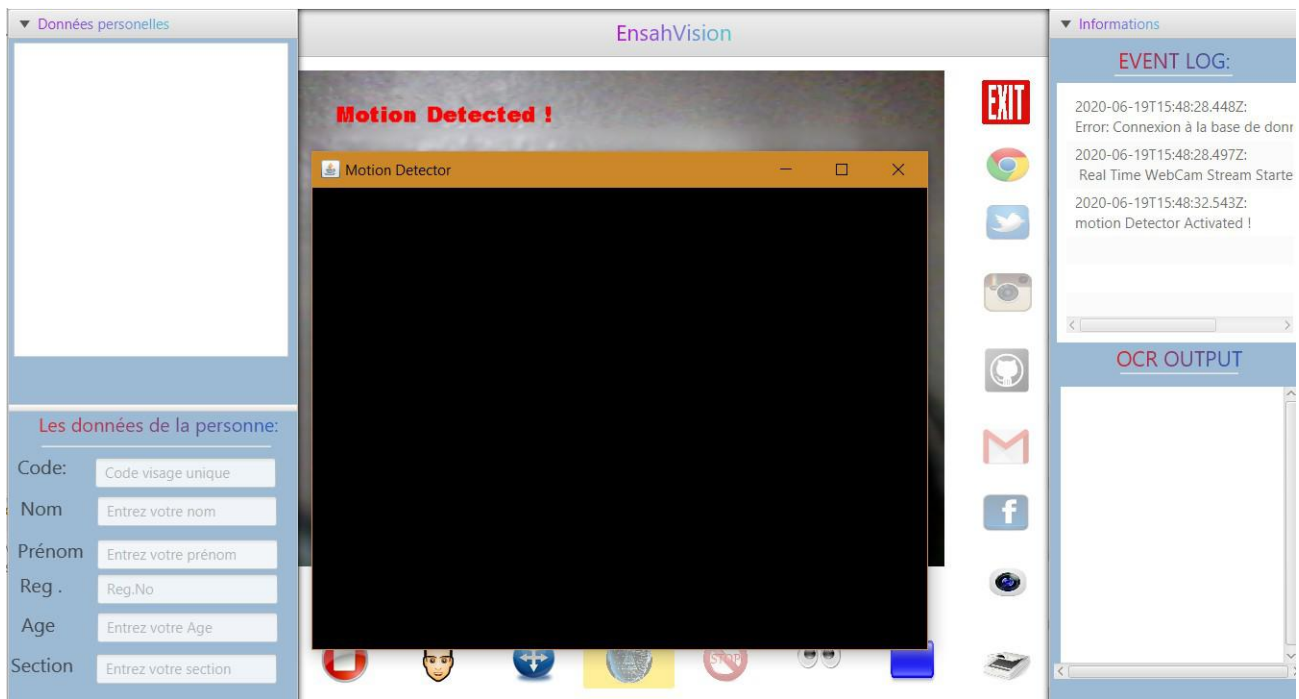


Fig15.Détection des mouvements

4.3.7 Détection d'une forme rectangulaire

C'est le même principe que dernier, fausse qu'ici il détecté seulement les objets en forme rectangulaire en les encadrants en vert.

-Pour le code qui génère ce bouton :

```
public void loop() {
    storage2 = AbstractCvMemStorage.create();
    try {
        storage = cvCreateMemStorage(0);
        try {
            grabber.start();
        } catch (org.bytedeco.javacv.FrameGrabber.Exception e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
        // get framerate
        double frameRate = grabber.getFrameRate();
        long wait = (long) (1000 / (frameRate == 0 ? 10 : frameRate));
        // keep capturing
        while (true) {
            Thread.sleep(wait);
            grabbedImage = converter.convert(grabber.grab());
            cvClearMemStorage(storage2);
            cvClearMemStorage(storage);
            if (grabbedImage != null)
            { drawSquares(grabbedImage, findSquares4(grabbedImage,
            storage));
            }
            // show stack trace
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
@FXML
protected void shapeStart() {

    faceDetect.stop();
    SquareDetector shapeFrame = new SquareDetector();
    shapeFrame.loop();

}
```

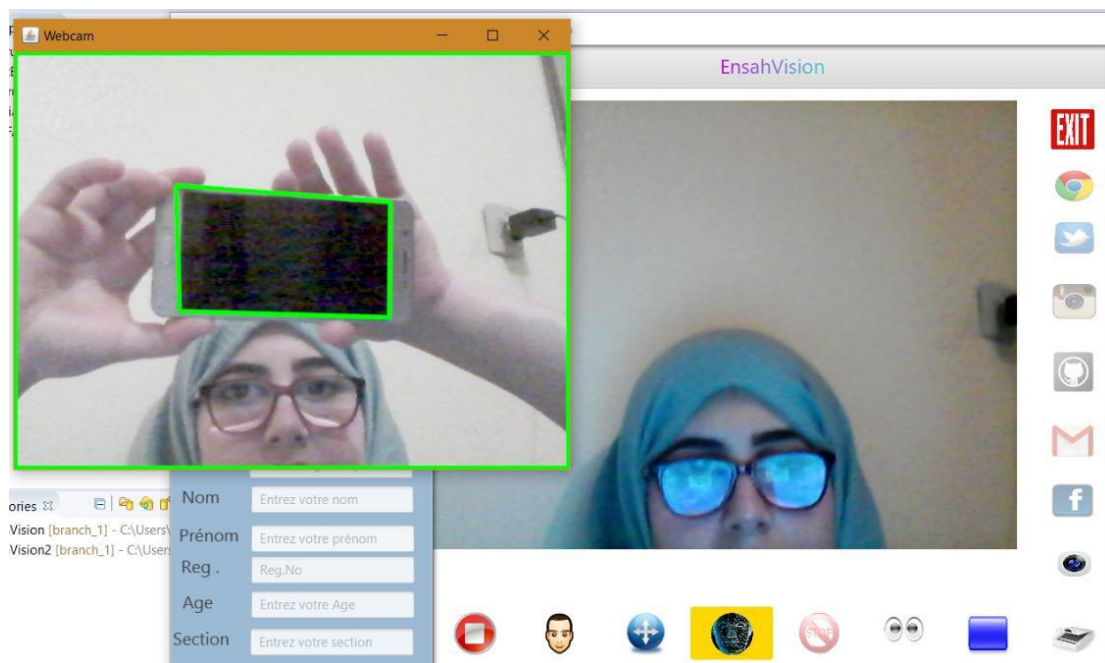


Fig16.Détection d'une forme rectangulaire

4.4 Conclusion

Dans ce chapitre, on a bien spécifier les outils de développement matériels et logiciels, et on a donnée une description totale sur les différentes fonctionnalités de notre application, qui se divisent sur deux parties : une partie liée a la détection de visage et yeux, et l'autre partie comprend l'activation des réseaux sociaux et détection des mouvements et formes rectangulaire.

CONCLUSIONS ET PERSPECTIVES

Dû à sa facilité d'utilisation, la reconnaissance des visages, considérée comme étant les fruits de recherches de plusieurs experts dans le domaine de l'imagerie, restera un outil puissant malgré l'existence d'autres méthodes biométriques de reconnaissance.

En dépit de l'énorme progression en informatique et plus spécifiquement le traitement d'images, le traitement des données et l'élaboration des bases des données, les systèmes appliquant les techniques de la reconnaissance de visages restent relativement peu convenables. Comme on a pu le voir à travers ce projet de fin d'études, le problème ne réside ni dans l'unicité du visage humain et dans les différents traits et éléments morphologiques qu'il contient, ni dans la capacité des systèmes informatiques à effectuer les différentes tâches de traitement d'images et de comparaison de données. D'ailleurs il reste encore des recherches à mener et des améliorations à effectuer dans ces deux domaines, mais dans les critères choisis pour effectuer le processus de reconnaissance. Pour surmonter les difficultés de la modulation de la morphologie faciale, avec tout ce qu'elle contient comme éléments, il est important que les anthropologues participent plus activement dans le développement de ce genre de systèmes, ce que nous n'avons pas pu constater dans la littérature anthropologique et médico-légale. En effet, la plupart des systèmes qui essayent d'intégrer la morphologie du visage dans leurs algorithmes de reconnaissance sont développés par les informaticiens suivant les méthodes de modulation des objets en général, sans prendre en compte la spécificité du visage humain, ce qui constitue un facteur défavorable pour l'avancement de cette technique. Nous proposons, donc, de fortifier la collaboration entre les spécialistes des deux sciences, bien que très différentes l'une de l'autre dans leur nature et leurs méthodes de recherches, pour aboutir aux meilleurs résultats.

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. Actuellement, il y'a une nouvelle tendance qui arrive et qui commence à susciter les efforts c'est le multimodale, dans lequel on combine plusieurs technologies biométriques, ou plusieurs algorithmes de reconnaissance, et dans lequel on utilise divers classificateurs pondérés pour essayer d'améliorer les performances de reconnaissance et réaliser des campagnes de test complètes pour valider le concept.

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.

NETOGRAPHIE

- [1] <http://www.developpez.net/>
- [2] <http://fr.wikipedia.org/>
- [3] <http://www.commentcamarche.net/>
- [4] <http://korben.info/>
- [5] <http://www.siteduzero.com/>
- [6] <http://www.irit.fr/>
- [7] <http://biometrics.it-sudparis.eu/>
- [8] <http://www.cppfrance.com/>
- [9] <http://opencv.willowgarage.com/>
- [10] <http://uml.free.fr/>
- [11] <http://www.automobileevaluation.com/>
- [12] <http://www.developpez.net/>
- [13] <http://www.dicofr.com/>
- [14] <http://www.cplusplus.com/forum/>
- [15] <http://www.programmez.com/>

BIBLIOGRAPHIE

[B1] L. Png, "*Morphological Shared-Weight Neural Network For Face Recognition*", A dissertation submitted to the University of Manchester Institute of Science and Technology for the degree of MSc, August 2004.

[B2] John D. Woodward, Jr., Christopher Horn, Julius Gatune, and Aryn Thomas "*Biometrics A Look at Facial Recognition*", documented briefing by RAND Public Safety and Justice for the Virginia State Crime Commission, 2003

[B3] A. Nefian and M. Hayes. "*Face Detection and Recognition Using Hidden Markov Models*". In: International Conference on Image Processing (ICIP), pp. 141–145, 1998.

[B4] 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.

[B5] Dr. Andrzej Drygajlo, ELE 233, "*BIOMETRICS*". Document available at : <http://scgwww.epfl.ch/courses>.

[B6] Bernd Heisele, Purdy Ho, Jane Wu, and Tomaso Poggio, "*Face recognition component based versus global approaches*", *Computer Vision and Image Understanding* Volume 91, Issue 1-2 (July 2003), Special issue on Face recognition Pages: 6–21, ISSN: 1077-3142.

[B7] "*Resources for Face Detection*". Document available at: <http://vision.ai.uiuc.edu/mhyang/face-detection-survey.html>.

[B8] Chennng Jye Liou, "*A Real Time Face Recognition System*", DSP/IC Design Lab, Department of Electrical Engineering, National Taiwan University, June 1997.

[B9] 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.

[B10] Mohamed Tayeb Laskri and DjallelChefrour, “*Who_Is : système d’identification des visages humains*”, A R I M A – Volume 1 – 2002, pages 39 à 61.

[B11] “*Biometricssystem – IDTECK*” Document available at:
<http://www.idteck.com/technology/biometrics.jsp>

[B12] Cabal (Christian), 2003. Rapport sur les méthodes scientifiques d’identification des personnes à partir de données biométriques et les techniques de mise en œuvre. 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.

[B13] M. A. Turk and A. P. Pentland, “*Face recognition using eigenfaces,*” in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, Jun. 1991, pp.586–591.

[B14] Y. Su, S.G. Shan, X.L. Chen, and W. Gao. Hierarchical ensemble of global and local classifiers for face recognition. In ICCV07, pages 1_8, 2007. AnastasiosTefas, Constantine Kotropoulos, and Ioannis Pitas. Face verificationusingelastic graph matchingbased on morphological signal decomposition. *Signal Process.* 82(6):833_851, 2002.

[B15] J. Li, S. Zhou, and C. Shekhar, “*A comparison of subspaceanalysis for face recognition,*” in *Proc. IEEE Int. Conf. Acoustics, Speech, SignalProcessing*, vol. 3, Apr. 2003, pp. 121–124.

[B16] Jonathon Phillips and Patrick J Flynn. Overview of the face recognition grand challenge. *Computer Vision and Pattern Recognition*, june 2005.

[B17] S. Der Phillips J., P. J. Rauss. Feret recognition algorithmdevelopment and test results. *ArmyResearchLaboratorytechnical report, ARL-TR-995*, 1996.

[B18] Olivier (G.) 1961. '*Morphologie et types humains*'. Vigot Frères, 171 pages. F.S.

Samaria and S. Young. Hmm-based architecture for face identification.

[B19] IVC, 12(8) :537_543, October 1994. L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. J. Opt. Soc. Am. A, 4(3) :519, 1987.

[B20] Dr. Andrzej Drygajlo, ELE 233, "biometrics". Document available at :
<http://scgwww.epfl.ch/courses>.

[A1] Présentation d'OpenCV :

OpenCV (Open Computer Vision) est une bibliothèque libre de visualisation en temps réel pour le langage C/C++, elle idéalise les optimisations *Integrated Performance Primitives (IPP)* d'Intel mais fonctionne aussi très bien sans. Les domaines d'utilisation sont variés : IHM, robotique, détection et reconnaissance d'objets ou visage ainsi que suivi et étude de leurs mouvements.

OpenCV est optimisée pour les processeurs multi-cœurs, comme elle est multiplateformes c'est-à-dire disponible pour Linux, Windows, Mac OS X) et gratuite sous licence BSD.

Caractéristiques d'OpenCV :

- Image de manipulation de données (répartition, les sorties, la copie, la création, conversion).
- Image et vidéo I/O (fichier et l'appareil photo d'entrée en fonction, image/fichier de sortie vidéo).
- La manipulation de matrices et vecteurs et routines d'algèbre linéaire (valeurs propres, produits, solveurs, SVD).
- Diverses structures de données dynamiques (listes, les files d'attente, ensembles, arbres, graphes).
- Traitement de l'image de base (filtrage, détection de contour, de détection d'angle, l'échantillonnage et l'interpolation, de conversion des couleurs, des opérations morphologiques, histogrammes, pyramides d'images).
- L'analyse structurale (composantes connexes, le traitement du contour, transformée de distance, des moments différents, correspondant modèle, transformée de Hough, approximation polygonale, montage en ligne, ellipse montage, triangulation de Delaunay).
- Calibration caméra (et de repérage mires de calibrage, le calibrage, estimation de la matrice fondamentale, l'estimation homographie, la correspondance stéréo).
- Analyse du mouvement (flot optique, segmentation de mouvement, suivi).
- Reconnaissance d'objets (Eigen-méthodes, HMM).
- Basic GUI (l'image d'affichage / vidéo, le clavier et la manipulation de la souris, des barres de défilement).
- Image d'étiquetage (ligne, coniques, des polygones, dessin de texte)

Les principaux modules d'OpenCV :

Cv : La fonction principale d'OpenCV pour Traitement d'image, calibrage caméra Géométrie algorithmique (triangulation, ...)

Cvaux : pour la reconnaissance de visage et code expérimental et obsolète.

CxCore : pour structures dynamiques (listes, files, graphes, arbres) et persistantes (L/E) manipulations de matrices, méthodes de l'algèbre linéaire et statistiques dessin en sur impression (overlay) (ex. étiquetage)

HighGUI : E/S vidéos, curseurs, fenêtrage.

ML : classification, regroupement (clustering).

Les structures de base de données OpenCV :

➤ Chargement d'un fichier image :

-IplImage : structure de base pour les « images » la fonction qui permet de charger l'image.

-cvShowImage : la fonction permettant d'afficher l'image.

-cvReleaseImage : la fonction permettant de libérer la mémoire image.

➤ Conversion et binarisation d'une image :

-cvConvertImage : convertir l'image couleur d'entrée en image à niveau de gris.

-cvThreshold : binarise l'image de sortie.

[A2] Une **interface de programmation** (*Application Programming Interface* ou *API*) est une interface fournie par un programme informatique. Elle permet l'interaction des programmes les uns avec les autres, de manière analogue à une interface homme-machine, qui rend possible l'interaction entre un homme et une machine.

Du point de vue technique une API est un ensemble de fonctions, procédures ou classes mises à disposition par une bibliothèque logicielle, un système d'exploitation ou un service. La connaissance des API est indispensable à l'interopérabilité entre les composants logiciels.

[A3] Une méthode bien connue de détection d'objets complexes tels que les visages est l'utilisation de « classifieurs de Haar » montés en cascade (boostés) au moyen d'un algorithme AdaBoost. Cette méthode est implémentée nativement dans la bibliothèque OpenCV et a été présentée initialement dans Viola et Jones **[A8]**. Le principe de cette méthode est obtenir un algorithme complexe de classification, composé de classifieurs élémentaires qui éliminent au fur et à mesure les zones de l'image qui ne sont pas compatibles avec l'objet recherché. Ces classifieurs binaires reposent sur des primitives visuelles qui dérivent des fonctions de Haar (Haar- like features).

```
Detection::Detection() {  
  
std::string cascadeNameFace = "C:\\OpenCV2.1\\data\\haarcascades\\haarcascade_frontalface_alt.xml";  
std::string cascadeNameEyes = "C:\\OpenCV2.1\\data\\haarcascades\\haarcascade_eye.xml"; std::string  
cascadeNameMouth = "C:\\OpenCV2.1\\data\\haarcascades\\haarcascade_mcs_mouth.xml"; std::string  
cascadeNameNose = "C:\\OpenCV2.1\\data\\haarcascades\\haarcascade_mcs_nose.xml"; .....  
}
```

[A4]Erosion :

- les objets de taille inférieure à celle de l'élément structurant vont disparaître,
- les autres seront "amputés" d'une partie correspondant à la taille de l'élément structurant,
- S'il existe des trous dans les objets, c'est à dire des "morceaux" de fond à l'intérieur des objets, ils seront accentués.
- les objets reliés entre eux vont être séparés.

[A5] La dilatation :

La dilatation est l'opération duale (ou inverse) de l'érosion.

- tous les objets vont "grossir" d'une partie correspondant à la taille de l'élément structurant. - s'il existe des trous dans les objets, c'est à dire des "morceaux" de fond à l'intérieur des objets, ils seront comblés.
- si des objets sont situés à une distance moins grande que la taille de l'élément structurant, ils vont fusionner.

[A6] L'ouverture :

Une érosion suivie d'une dilatation s'appelle une ouverture.

L'ouverture a pour propriété d'éliminer toutes les parties des objets qui ne peuvent pas contenir l'élément structurant.

[A7] La fermeture :

Une dilatation suivie d'une érosion s'appelle une fermeture.

La fermeture a pour propriété de combler tout ce qui est de taille inférieure à l'élément structurant.

[A8] La méthode de Viola et Jones est une méthode de détection d'objet dans une image numérique, proposée par les chercheurs Paul Viola et Michael Jones en 2001. Elle fait partie des toutes premières méthodes capables de détecter efficacement et en temps réel des objets dans une image. Inventée à l'origine pour détecter des visages, elle peut également être utilisée pour détecter d'autres types d'objets comme des voitures ou des avions. La méthode de Viola et Jones est l'une des méthodes les plus connues et les plus utilisées, en particulier pour la détection de visages et la détection de personnes.

En tant que procédé d'apprentissage supervisé, la méthode de Viola et Jones nécessite de quelques centaines à plusieurs milliers d'exemples de l'objet que l'on souhaite détecter, pour entraîner un classifieur. Une fois son apprentissage réalisé, ce classifieur est utilisé pour détecter la présence éventuelle de l'objet dans une image en parcourant celle-ci de manière exhaustive, à toutes les positions et dans toutes les tailles possibles.

Considérée comme étant l'une des plus importantes méthodes de détection d'objet, la méthode de Viola et Jones est notamment connue pour avoir introduit plusieurs notions reprises ensuite par de nombreux chercheurs en vision par ordinateur, à l'exemple de la notion d'image intégrale ou de la méthode de classification construite comme une cascade de classifieurs boostés.

Cette méthode bénéficie d'une implémentation sous licence BSD dans OpenCV, une librairie très utilisée en vision par ordinateur.