



Université de la Manouba
École Nationale des Sciences de l'Informatique



RAPPORT DU
PROJET DE CONCEPTION ET DE DÉVELOPPEMENT

Sujet : Conception et développement d'un outil intelligent de reconnaissance faciale

Mahdié Far
Auteurs :

Sami KAMMOUN

Yosser BATITA

Encadrants: Enseignants

Pr. Inés HAMDI

Année Universitaire :2019 /2020

Remerciements

Table des matières

Introduction

1 Etude de l'existant

chapter.1

I. Une section

section.1

II. Une autre section

section.1

III conclusion

section.1

2 Analyse et specification de besoins

chapter.1

1.1 Une section

section.1.1

1.2 Une autre section

section.1.2

1.3 conclusion

section.1.3

3 Conception

chapter.1

1.1 Une section

section.1.1

1.2 Une autre section

section.1.2

1.3 conclusion

section.1.3

4 Réalisation

chapter.1

1.1 Une section

section.1.1

1

2

2

2

2

3

2

2

2

2

4

2

2

2

5

2

1.2 Une autre section	
section.1.2	
1.3 conclusion	
section.1.3	

Conclusion et perspectives	
chapter*.9	3

Bibliographie	
chapter*.10	4

Netographie	
Item.1	5

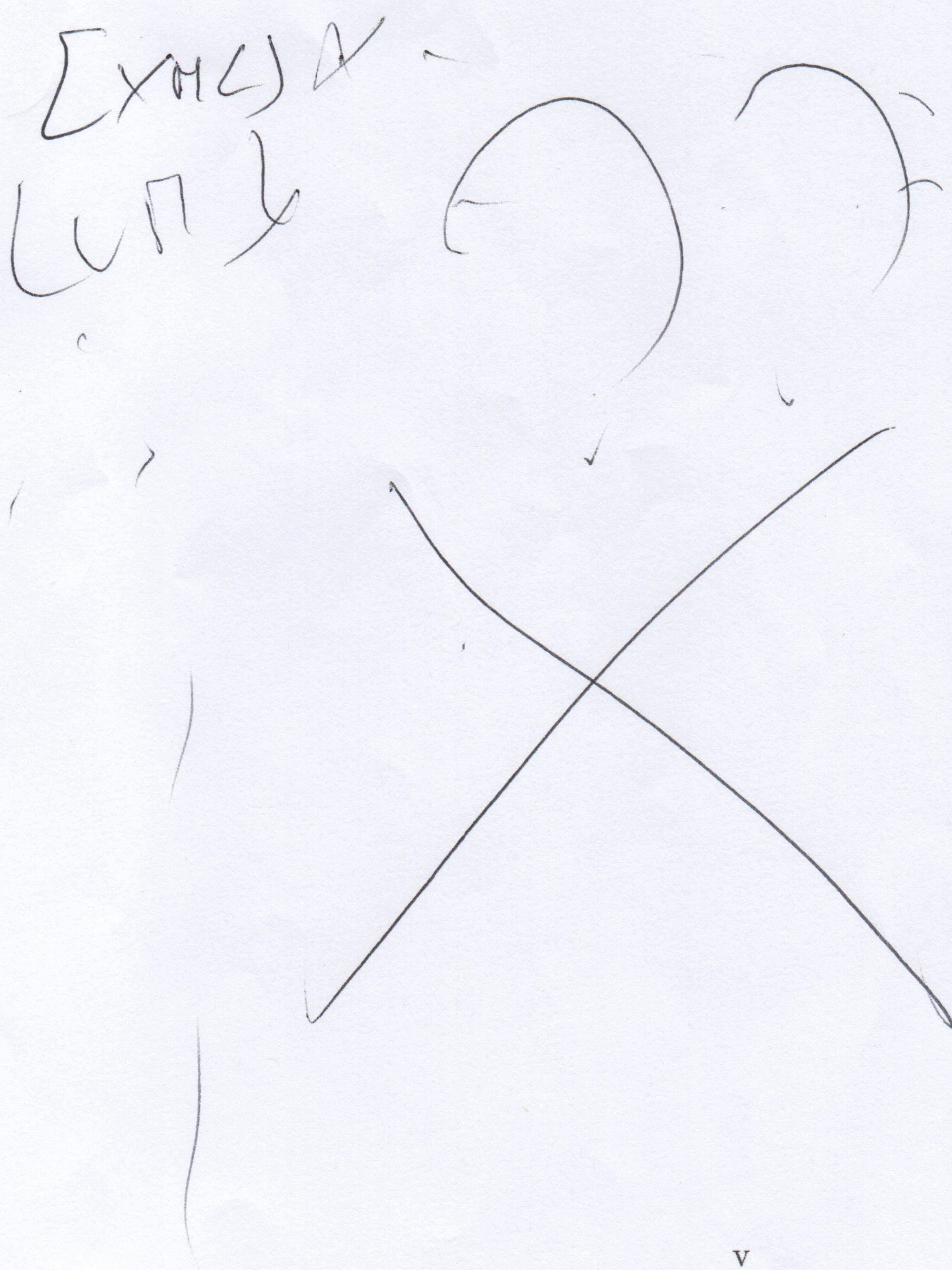
Table des figures

1	Diagramme de cas d'utilisation « Choisir la meilleure image »	xi
2	Diagramme de cas d'utilisation « Reconnaître les visages »	xii
3	Diagramme de cas d'utilisation « Gérer la base de données »	xii
4	Processus de reconnaissance faciale	xiv
5	Schéma fonctionnel	xv
6	Filtre Médian	xvi
7	Schéma du système de détection	xvii
8	Étapes de reconnaissance faciale	xviii

Liste des tableaux

Liste des sigles et acronymes

[ACRO5] ACRO1ACRONYME1 ACRO2ACRONYME2 ACRO3ACRONYME3 ACRO4ACRONYME4
ACRO5ACRONYME5



Introduction

Identifier une personne à partir de son visage est une tâche simple pour les êtres humains. En est-il de même pour la machine ? Ceci détermine la problématique de la reconnaissance automatique de visages, qui a engendré un grand nombre de travaux de recherche au cours des dernières années. Savoir déterminer de manière à la fois efficace et exacte l'identité d'un individu est devenu un problème critique car de nos jours l'accès sécurisé et la surveillance constituent un sujet de très grande importance. 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. 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. 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'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. Le visage peut être considéré comme une donnée biométrique. Une donnée biométrique est une caractéristique physique ou biologique permettant d'identifier une personne. Les données biométriques sont devenues des données inéluctables pour le problème de l'identification sécurisée et de la reconnaissance de personnes. Les méthodes d'identification ou de vérification d'identité fondées sur une donnée biométrique offrent plus d'avantages par rapport aux méthodes fondées sur un mot de passe ou un code PIN puisque , les données biométriques sont des données individuelles, alors que les mots de passe peuvent être utilisés ou volés par quelqu'un d'autre que l'utilisateur autorisé.. Cela veut dire entre autre que la biométrie faciale nous permet d'exploiter de nombreuses informations relatives à une personne. Dans la vie quotidienne, le visage est probablement le trait biométrique le plus utilisé par les humains afin de reconnaître les autres, le visage

a de grands avantages par rapport aux autres biométries, parce qu'il est naturel, et facile à acquérir. La reconnaissance de visages est par conséquent devenue l'une des branches de la vision par ordinateur qui remporte un grand succès et qui est en perpétuel développement. Le but d'un système de reconnaissance de visages est de simuler le système de reconnaissance humain par la machine pour automatiser certaines applications telles que : la télésurveillance, le contrôle d'accès à des sites, l'accès à des bâtiments sécurisés, etc. Le présent rapport est constitué de quatre chapitres : Le premier, intitulé « Etude de l'existant », a pour objectif d'avoir une idée claire et précise sur l'existant ~~quel qu'il soit~~. Le ~~deuxième chapitre est l'~~ « Analyse et spécification des besoins ». Il vise à analyser la faisabilité organisationnelle et technique du projet. Le troisième chapitre, intitulé « Conception », présentera la conception globale et détaillée qui sera traitée. Le dernier chapitre présentera les réalisations effectuées tout au long du projet ainsi que l'ensemble des tests d'exécution. Enfin une conclusion générale clôturera le rapport.

Q (mb)

mritwé

Chapitre 1 :Etude de l'existant

Dans ce chapitre, nous présentons l'étude préalable qui doit être élaboré avant d'entamer la mise en place de notre projet. Pour cela, nous présenterons l'étude des applications de reconnaissance faciale.

0.1 I.Etude de l'existant

0.1.1 L'application FindFace

C'est une application développée par la start-up russe N-Tech qui utilise la technologie des réseaux de neurones artificiels et le réseau social russe VKontakt Considérée comme le Shazam des visages, pour le bon fonctionnement de l'application, il faut que deux conditions soient remplies : une photo d'une bonne qualité et que la personne photographiée dispose d'un compte VKontakte. Ensuite, c'est un algorithme qui prend le relai basé sur la technologie de réseau de neurones artificiels (le même modèle que pour nos neurones qui servent à apprendre). Cet algorithme dispose de 80 caractéristiques comme le sexe, l'âge ou la forme du visage. Ainsi, l'algorithme peut reconnaître l'ensemble des caractéristiques du visage humain. Ensuite, le scan de la photographie est analysé pour être comparé avec le réseau social russe VKontakte utilisé comme base de données de photos pour l'application FindFace. D'après les auteurs du projet, leur algorithme serait également compatible avec les bases de données de Facebook ou Instagram ! D'après le fondateur de la start-up, pour la base de 300 millions de photos, le taux de réussite est de 70 %.

0.1.2 L'application BioID

BioID est une application de reconnaissance faciale pour les utilisateurs Androïde et iOS. C'est une application d'authentification d'utilisateur multifactorielle qui permet de sécuriser les Smartphones et les données. En ajoutant une authentification biométrique sur le Smartphone avec les dernières fonctionnalités afin que personne d'autre ne puisse y accéder sans votre permission. Il réduit le type de mot de passe long et déverrouille facilement le Smartphone en utilisant votre BioID comme visage, scanner d'iris d'empreintes digitales et autres.

0.1.3 L'application ClearView

L'application créée par l'Américain Hoan Ton-That, a partir d'une simple photo, l'application est capable de retrouver le nom, mais aussi dans la plupart des cas l'âge, des photos des proches, de vos compétitions sportives, Bref, parfois même des photos que vous aviez oubliées ! En fait, l'application est capable de puiser dans les contenus que vous postez sur les réseaux sociaux, et ce, sans avoir votre autorisation. Extrêmement efficace, donc, mais également redoutable puisque ClearView est utilisée par pas moins de 600 agences gouvernementales, selon le New York Times, qui a révélé l'existence de cette application au grand public

0.2 Tableau récapitulatif

Dynithé

Application	Avantages	Inconvénients
FindFace	- facilite la rencontre. - efficacité énorme. - taux d'identification de 73%.	- taux d'identification faible pour les personnes âgées.
BioID	- déverrouillage facile du Smartphone.	- baisse générale de la sécurité des smartphones.
ClearView	- outil utilisé pour résoudre des dossiers de vol. - capacité énorme à identifier une personne même si elle porte un chapeau ou des lunettes.	- les pratiques de l'application sont basées sur : - l'extraction non-autorisée des données. - fraude informatique. - vol d'information.

0.3 Conclusion

L'étude des applications de la détection et reconnaissance de visages est très riche, nous trouvons des multitudes applications qui sont utilisés et développés dans les systèmes commerciaux et dans les projets de recherche. Pour la détection de visage, et pour avoir un meilleur taux de détection de visage avec un minimum de temps d'exécution pour le traitement d'un seul visage par image et afin d'éviter les règles de calcul complexe pour répondre aux exigences de notre projet nous choisirons la méthode de Viola-Jones qui contient un algorithme de détection d'objet AdaBoost devenu une référence de détection d'objet par ses qualités de rapidité et d'efficacité. Pour la reconnaissance de visage, en nous basant sur l'approche locale nous utiliserons une méthode basée sur quelques points caractéristiques du visage.

Qui sont ces auteurs ?

Chapitre 2 : Analyse et spécification des besoins

Dans ce chapitre, nous passerons à l'analyse des besoins fonctionnels et non fonctionnels du module à développer. Cette analyse sera appuyée par les diagrammes de cas d'utilisation et de séquence ~~système~~.

0.4 Besoins fonctionnels

Notre application se répartit en deux modules fondamentaux :

- détection de visage
- reconnaissance de visage

Ces modules représentent les fonctionnalités les plus importantes de notre application dont les besoins fonctionnels sont :

La détection de visage : Le système doit détecter le visage dans une image. La détection consiste à délimiter une zone rectangulaire contenant la partie de l'image reconnue comme un visage.

La construction d'une base de descripteurs associés aux visages : Dans cette étape nous extrairons de l'image les informations qui seront sauvegardées en mémoire pour être utilisées plus tard dans la phase de reconnaissance. Les informations doivent être choisies de manière à être discriminantes et non redondantes.

La reconnaissance de visage : La reconnaissance de visage permet de savoir si un nouvel individu est déjà enregistré dans la base ou non. Cette phase est l'aboutissement du processus, elle peut être valorisée par un taux de fiabilité qui est déterminé par le taux de justesse de la décision.

0.5 Besoins non fonctionnels

Interface utilisateur : L'application devra être cohérente du point de vue de l'ergonomie. La qualité de l'ergonomie sera un facteur essentiel, étant donné l'utilisation intensive qui sera faite de l'application.

Documentation : Une documentation écrite devra accompagner le système de reconnaissance faciale, qui décrit clairement son fonctionnement.

Rapidité : Le temps d'exécution de l'application ne devra pas dépasser 5 secondes.

Après avoir identifié les besoins fonctionnels et non fonctionnels, nous allons identifier les principaux cas d'utilisation du système, afin de bien concevoir notre solution.

0.6 Diagramme des cas d'utilisation

Les cas d'utilisation sont un moyen qui permet de capturer les exigences fonctionnelles d'un système. Ils décrivent les interactions entre les acteurs et le système et constituent une méthode pour recueillir et décrire les besoins des acteurs.

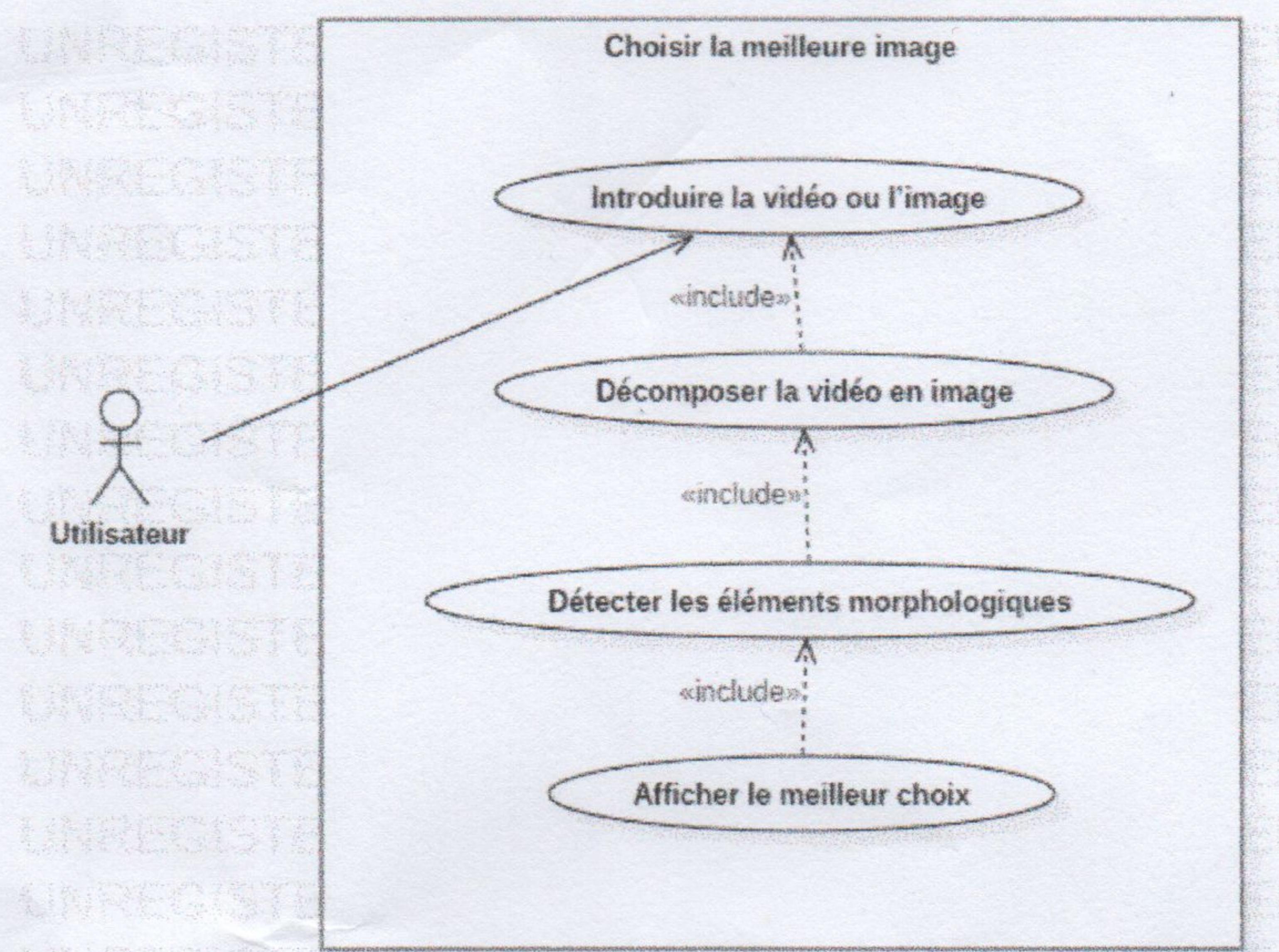


FIGURE 1 – Diagramme de cas d'utilisation « Choisir la meilleure image »

Dans ce cas d'utilisation, il y a lancement de la reconnaissance une fois l'utilisateur ouvre la vidéo à analyser . 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.

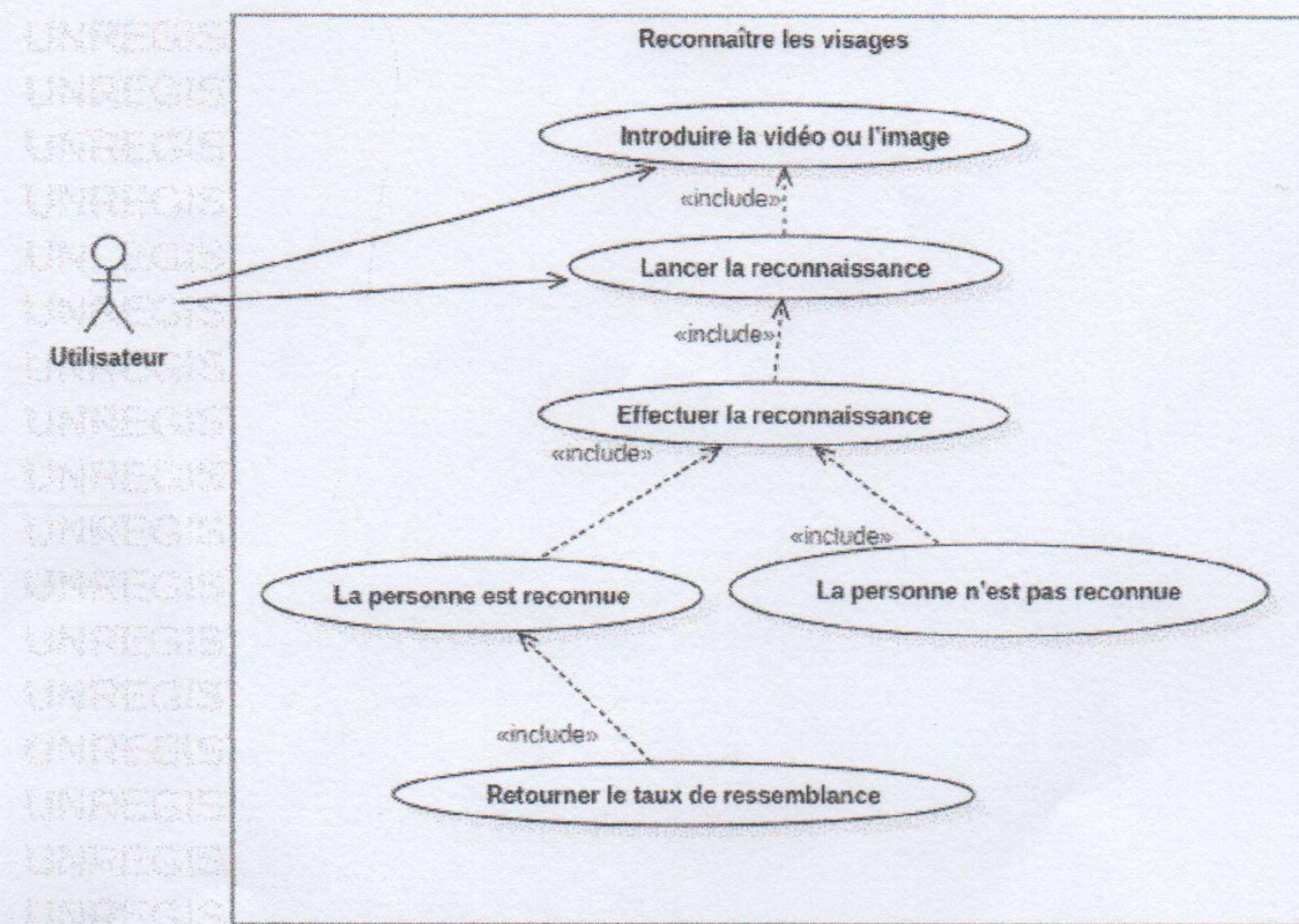


FIGURE 2 – Diagramme de cas d'utilisation « Reconnaître les visages »

Dans ce cas d'utilisation, l'utilisateur peut gérer la base de données en modifiant (ajout ou suppression) des données.

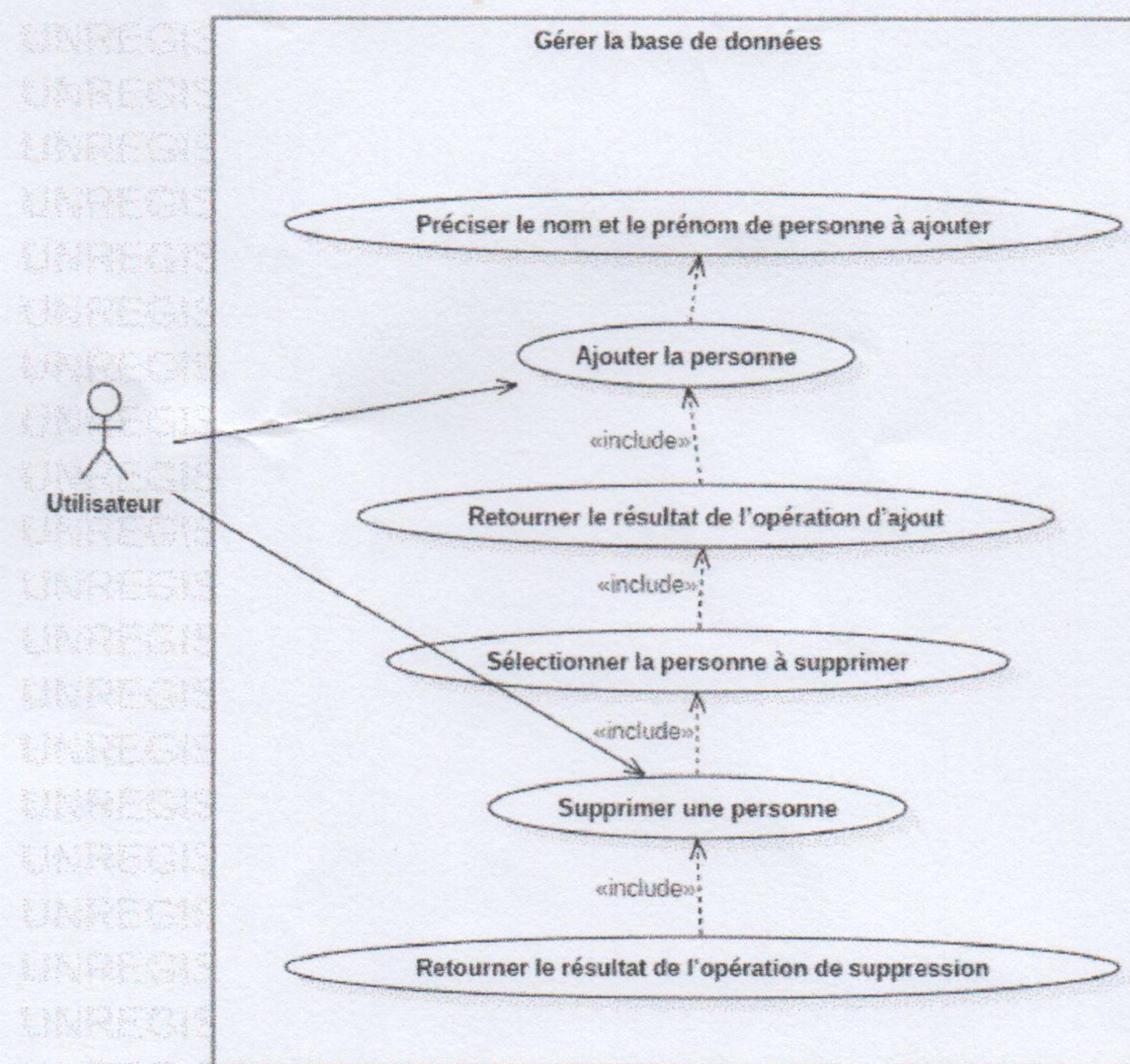


FIGURE 3 – Diagramme de cas d'utilisation « Gérer la base de données »

0.7 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~~ *suivant* chapitre, nous allons aborder la conception de l'application développée.

Après avoir déjoué les besoins
fonct. et non fonct. de notre projet
nous allons proposer
ce présent chapitre de modèle conceptuel.
Qui sera détaillé par la suite.

Chapitre3 :Conception

Introduction.

Pour simplifier le projet, une division en module s'imposait. Ainsi, le projet a été subdivisé en 3 modules (voir figure 1) : « Module de prétraitement », « Module de détection » et « Module de reconnaissance ». Leurs rôles sont respectivement de normaliser l'image capturée, détecter le visage et l'analyser pour enfin pouvoir le reconnaître.

I. ??.

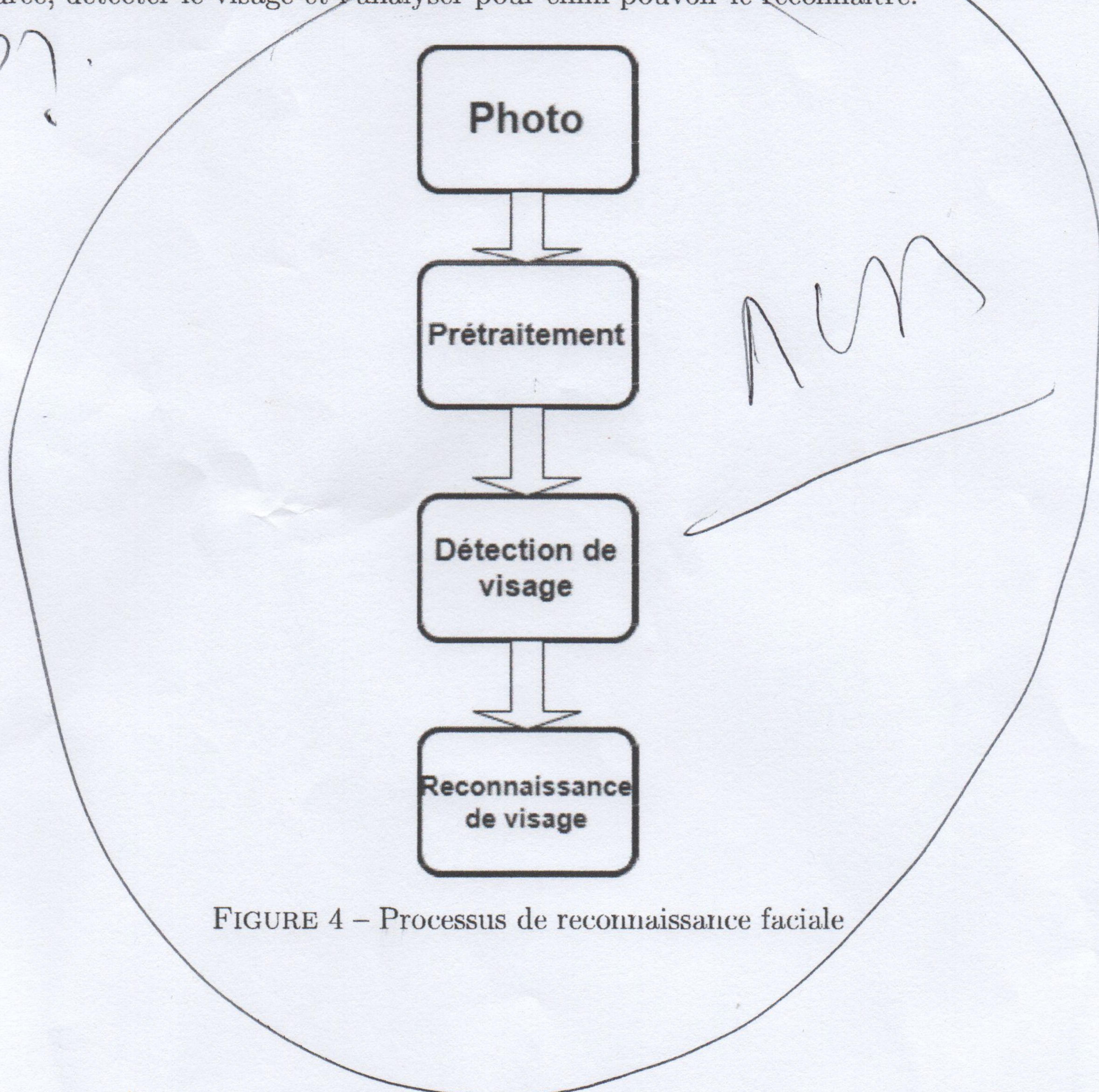


FIGURE 4 – Processus de reconnaissance faciale

I. Modèle Conceptuel Mapot

0.8 Schéma fonctionnel

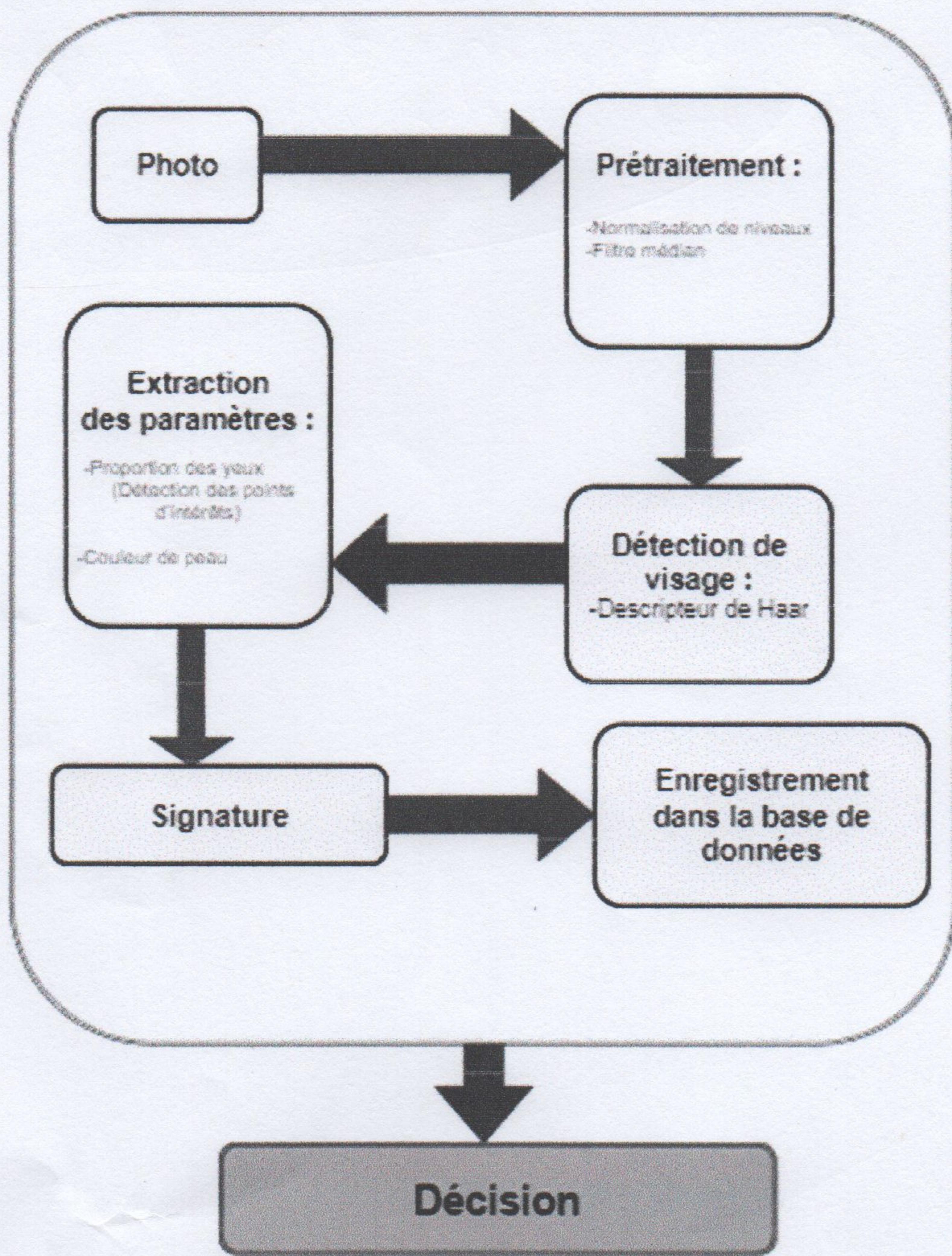


FIGURE 5 – Schéma fonctionnel

0.9 Module de prétraitement

Les images peuvent se présenter avec un éclairage et des ombres différentes ou encore avec des résolutions différentes. Il est donc nécessaire que les données soient normalisées et que les variations de contraste soient ainsi représentatives des caractéristiques du visage et non de son environnement. La normalisation des images reçues et détectées s'effectue en deux étapes : La première est l'étirement d'histogramme, la seconde consiste en l'application d'un filtre médian.

- **Etirement d'histogramme (Normalisation des niveaux de gris)** : L'étirement d'histogramme consiste à répartir les fréquences d'apparition des pixels sur la largeur de l'histogramme. Ainsi il s'agit d'une opération consistant à modifier l'histogramme de telle manière à répartir au mieux les intensités sur l'échelle des valeurs

disponibles. Ceci revient dans la plupart des cas à étendre l'histogramme afin que la valeur d'intensité la plus faible soit à zéro et que la plus haute soit à la valeur maximale.

- **Filtre Médian :** Le filtre médian nous permet d'éliminer un type particulier de bruits, dit « Salt and Pepper noise » qui consiste en des tâches dispersées d'une très faible ou forte intensité. Le principe du filtre est le suivant : il remplace le niveau de gris de chaque pixel par la valeur en la position médiane des niveaux de gris de son voisinage ordonné. Cette méthode est particulièrement pertinente quand la configuration du bruit est forte.

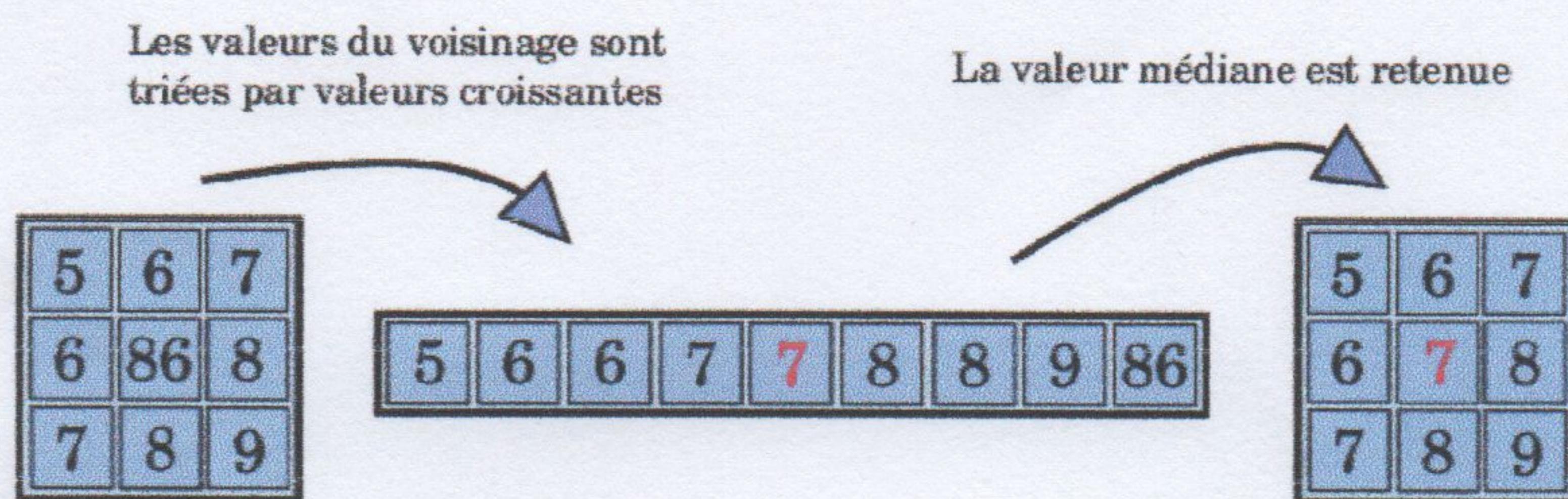


FIGURE 6 – Filtre Médian

0.10 Détection de visage

La réalisation de cette partie consiste à choisir la méthode de Viola-Jones qui nécessite une phase d'apprentissage à l'issue de laquelle une fonction de classification robuste est formée. En réalité la phase d'apprentissage a déjà été réalisée et les données qui en découlent nous sont proposées par la bibliothèque open source dédiée au traitement d'images « OpenCV » ; sous la forme d'un fichier « XML ». La fonction de classification offerte par cette bibliothèque sera exploitée par l'algorithme de détection pour la réalisation d'une classification en visages et non-visages.

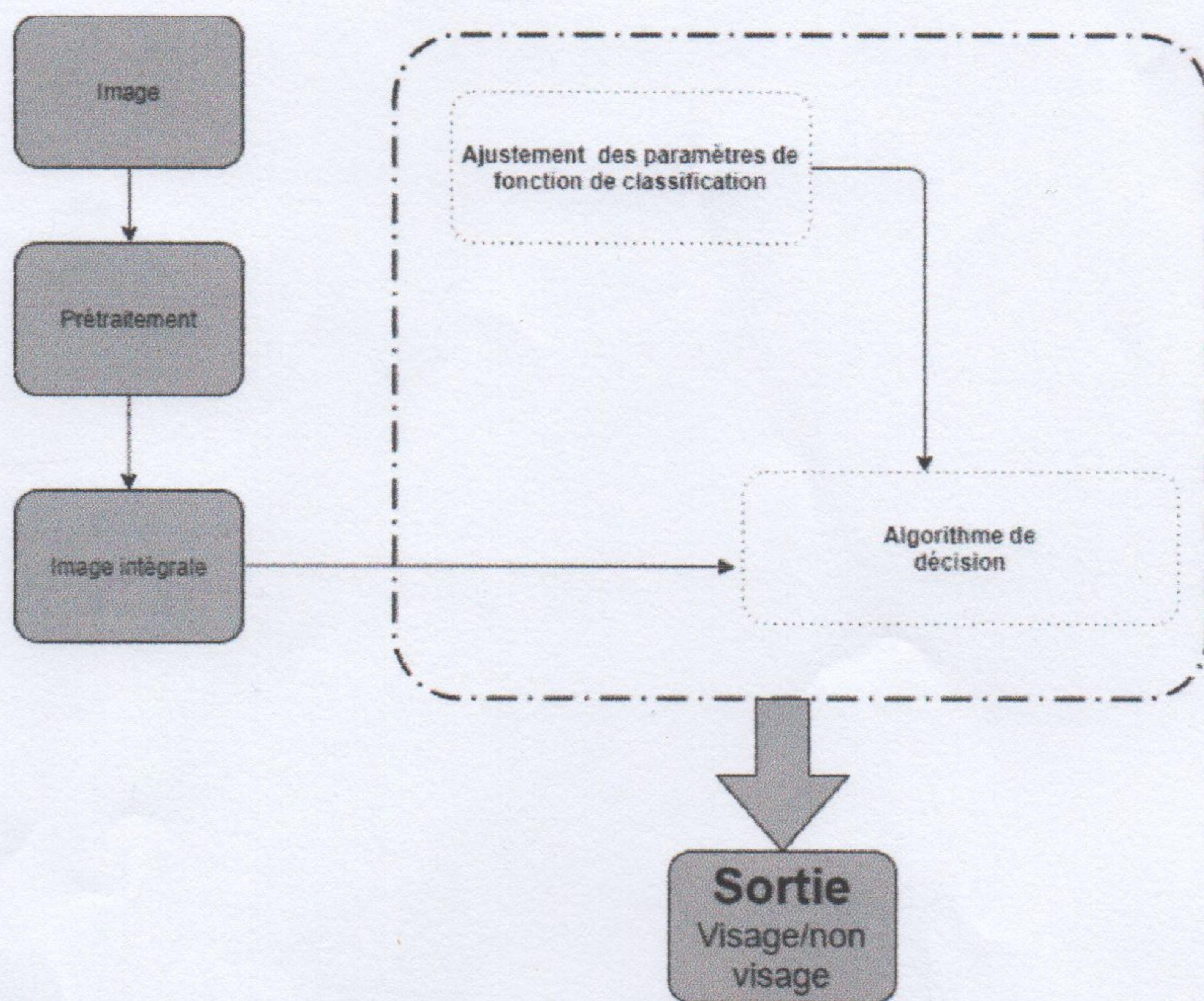


FIGURE 7 – Schéma du système de détection

0.10.1 Fonction de classification

La fonction de classification utilise comme entrée une cascade de classificateurs qui est un arbre de décisions où à chaque étape un classificateur est formé pour la détection de tous les objets d'intérêt (visages dans notre cas) et rejeter certaines fractions des objets qui n'appartiennent pas au modèle. Dans [LIENHART 2002], l'algorithme discret d'AdaBoost [FREUND 1996], permet l'élimination de 50% de fausses détections, mais il élimine 0.2% des modèles frontaux de visage.

0.10.2 Image intégrale

Pour identifier la présence ou l'absence de centaines de descripteurs de Haar à chaque endroit d'image et à plusieurs échelles efficacement, Viola et Jones ont utilisé une technique nommée « Image Intégrante ». En général, c'est l'ajouter d'ensemble de petites unités. Dans notre exemple, les petites unités sont des valeurs de pixel. La valeur intégrante pour chaque pixel est la somme de tous les pixels au-dessus et à gauche du pixel donné. En commençant en haut et en passant vers la droite.

0.10.3 Bloc de décision

La fonction de classification et les images intégrales sont les entrées d'une boucle. À chaque itération, la fenêtre de balayage sera redimensionnée, plus précisément elle sera agrandie d'un facteur d'échelle bien choisi. Par conséquent, les données de la

fonction de classification doivent subir une adaptation à ce changement. L'adaptation consiste principalement à redimensionner les descripteurs de Haar proportionnellement à la taille de la fenêtre. La boucle que nous venons de citer, se charge de garantir la détection de visage à toute échelle. Son paramètre principal est le facteur d'échelle. Ce paramètre est généralement entre 1.1 et 1.4. Afin de pouvoir repérer les visages à tout emplacement de l'image, notre fenêtre doit balayer toute l'image en utilisant un algorithme de décision à chaque itération .

0.11 Reconnaissance de visage

Dans cette partie nous définissons la méthode qui permette d'identifier un visage. Il est à noter que nous n'avons aucun modèle des différents visages à détecter. Cependant nous utilisons le terme « modèle » pour désigner une représentation de chaque visage par un vecteur contenant des informations pertinentes et non redondantes afin de bien caractériser le visage. Nous allons donc décrire dans la première étape la phase de préparation des modèles. Nous décrirons ensuite la phase d'identification. Le schéma ci-dessous, décrit la méthode de reconnaissance de visage

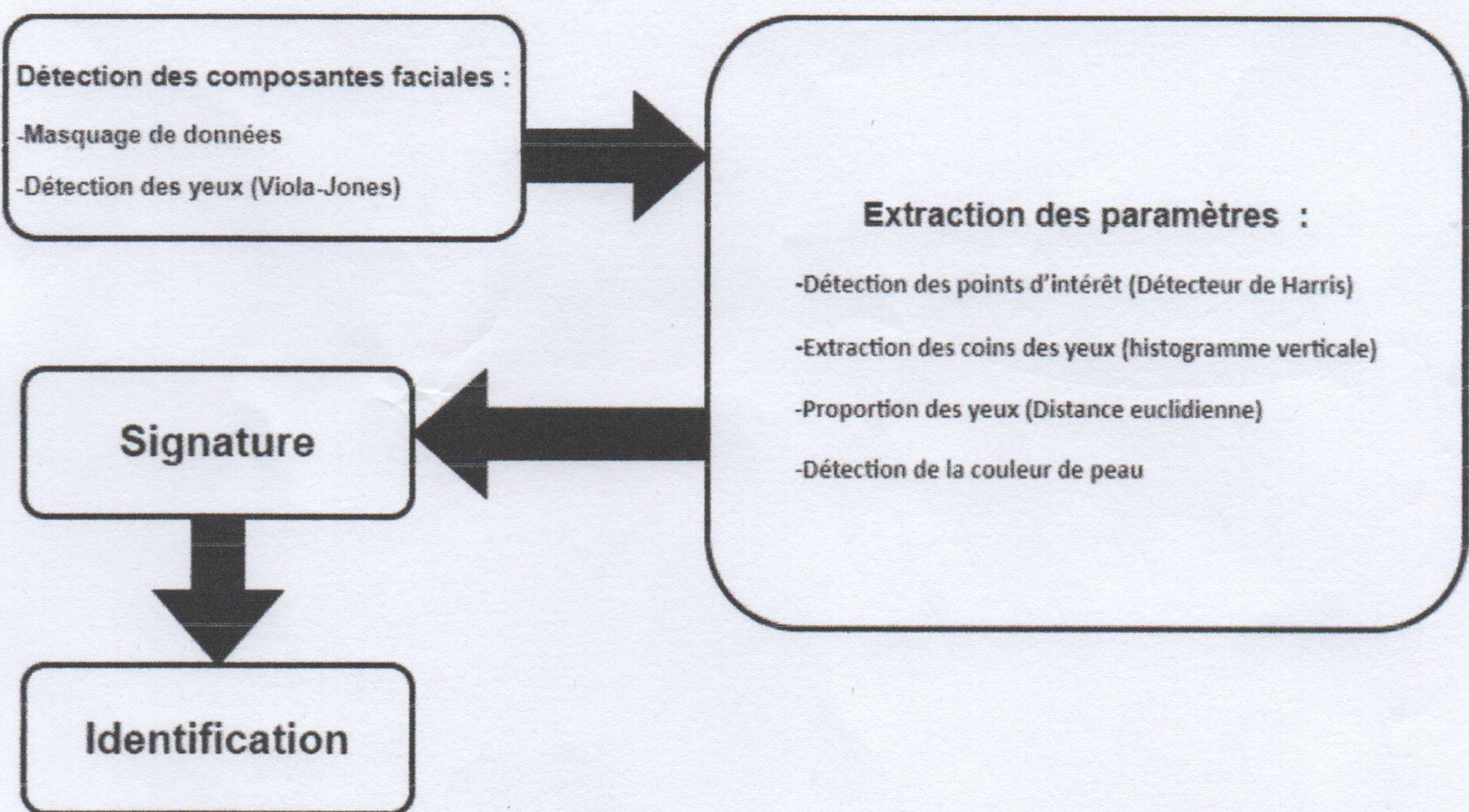


FIGURE 8 – Étapes de reconnaissance faciale

0.11.1 Détection des composantes faciales

Cette phase se traduit en deux étapes :

- **Masquage de donnée :** Afin de cibler l'analyse sur l'image, un visage ayant une forme ovale et verticale, les quatre coins de la fenêtre seront ignorés, ainsi

qu'une bande de chaque côté, dans la mesure où ces données représenteront principalement un décor, des habits, ou des cheveux, données très variables et inutiles pour la détection des yeux et du nez. Cette technique permet donc de définir un masque de taille $T \times T$ pixels (T a été fixé à 200 pixels) pour cacher des données. La forme exacte du masque sera déterminée de façon heuristique en effectuant des tests sur la base d'images disponible.

- **Détection des yeux :** De la même manière que celle de la détection de visage, nous utiliserons la méthode de Viola-Jones pour détecter les yeux et ce en utilisant l'image masquée et le classifieur de cascade spécifique aux yeux.

0.11.2 Extraction des paramètres

Après l'extraction de la région qui contient les yeux dans un visage, nous nous servons des informations situées dans ces régions pour définir des modèles de visages. Donc, un modèle de visage est représenté par deux paramètres : -
- La proportion des yeux qui est calculée après détection des points d'intérêt.
- La couleur de peau.

0.11.3 Signature

Une fois les points d'intérêts et la couleur de peau déterminés, nous commençons la génération de la signature de visage en construisant un vecteur qui contient deux caractéristiques, à savoir la proportion entre les coins des yeux et la couleur de peau. La précision des paramètres qui constituent la signature de visage, permet d'avoir une identification efficace

0.11.4 Identification

Une fois toutes les signatures créées et enregistrées dans la base de données, il s'agit maintenant de comparer un nouvel élément avec ceux déjà présents dans la base de données. Dans notre cas, nous proposons d'afficher les 12 visages qui ressemblent le plus au visage recherché. La comparaison est réalisée par le calcul de la distance entre la signature décrivant le nouvel élément candidat et les signatures enregistrées dans la base de données. Cette distance est obtenue en utilisant deux paramètres : - Couleur de peau - Proportions des coins des yeux. On calcule la différence en valeur absolue entre ces paramètres et les paramètres des visages enregistrés dans la base de données. Les modèles ayant le plus bas score seront les plus proches du visage recherché, pour cela on fait le tri des modèles à l'ordre croissant vis-à-vis ces paramètres pour afficher les 10 premiers. La décision finale sera alors effectuée par

l'utilisateur en choisissant ou non un visage parmi les visages donnés par le système.

Quel système ?

0.12 Conclusion

A la fin de ce chapitre nous avons pu expliquer le fonctionnement du système. Ceci est fait à travers une subdivision en module permettant de bien spécifier la composition et le comportement de l'application. Pour le chapitre suivant, on procède à la réalisation afin de mettre en œuvre la solution proposée.

Chapitre 4 :Réalisation

Dans ce chapitre, nous définirons les méthodes que nous avons développées, pour la détection et la reconnaissance d'un visage. Nous présenterons aussi les résultats obtenus à la suite des tests effectués, ainsi que nos propositions pour améliorer la méthode proposée. Pour la détection du visage nous avons utilisé des outils existants dans la bibliothèque OpenCV car elle possède pleins d'outils pour le traitement d'image déjà implémentés, citons particulièrement la méthode de « Viola-Jones » que nous avons choisie pour la détection de visage. Nous décrivons ici le fonctionnement et les services offerts par OpenCV.

0.13 II. Environnement de travail

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

Processeur : Intel®Core™ i7-7200U CPU @ 2.70GHz 2.90GHz

RAM : 8.00 GO

OS : Windows 10

Webcam intégrée

0.14 II. Présentation de la bibliothèque OpenCV

OpenCV [OPENCV, 2007] (Open source Computer Vision Library), est une bibliothèque de traitement d'images et de vision par ordinateur , optimisée, proposée par Intel pour Windows et Linux. Elle est « Open Source », Elle comprend plusieurs solutions pour le traitement d'image et de l'analyse du mouvement. OpenCV (Open Source Computer vision) est une bibliothèque de traitement d'images et de vision par ordinateur, optimisée ,initialement développée par Intel Windows et Linux et maintenant soutenu par la société de robotique Willow Garage, spécialisée dans le traitement d'images en temps réel [47]. Elle est « Open Source ». La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées, elle propose la plupart des opérations classiques en traitement d'images :

- lecture, écriture et affichage d'image ;
- calcul de l'histogramme des niveaux de gris ou d'histogramme de couleur

- lissage, filtrage ;

Cette bibliothèque s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils en traitement vidéo :

- lecture, écriture et affichage d'une vidéo depuis un fichier ou une caméra ;
- détection de visage par la méthode de Viola et Jones, détection de mouvement

1. Classifieur OpenCV

Nous nous servons, dans nos travaux de détection du visage, du classifieur OpenCV appelé. Il s'applique sur des régions de l'image tout en faisant subir des transformations d'échelle, afin de reconnaître si un objet d'une région rassemble à un visage.

2. Implémentation

Installation d'OpenCV sous Windows

L'installation sous Windows est simple. OpenCV est distribué en tant que package binaire prêt à l'emploi. Nous commençons par télécharger le fichier d'installation d'OpenCV : <http://sourceforge.net/projects/opencvlibrary/files/hyperref/CTAN> Puis nous installons le programme dans le chemin qui est proposé par défaut : C :

Environnement de programmation :

Pour réaliser notre application nous utiliserons le langage de programmation Python en se basant sur la bibliothèque OpenCV. Cette bibliothèque se compose de 5 modules :

- **cxcore** : Contient les structures de base et elle fournit de nombreuse fonction de dessin (lignes, cercles, ellipses, arcs, ...).
- **cv** : les fonctions principales.
- **cvaux** : Contient les fonctions en cours d'implémentation, ou dépréciées
- **highgui** : Contient les fonctions de création d'interfaces utilisateurs (matrices, tableaux, listes, files, graphes, arbres...), ainsi les structures élémentaire (matrices, tableaux, listes, files, graphes, arbres...).

- **mll (sorti en 2009)** : Elle intègre de nombreux algorithmes d'apprentissage automatisé.

Le module CV contient les fonctions nécessaires pour notre application, dont les plus importantes sont :

cvHaarDetectObjects : C'est une fonction pratique pour détecter des objets dans une images. Les paramètres d'entrée sont :

- o Une image
- o Base d'apprentissage (fichier XML)
- o Espace mémoire
- o Facteur d'échelle

cvGoodFeaturesToTrack : C'est une fonction pratique pour détecter les points d'intérêts. Les paramètres d'entrée sont :

- o Une l'image dont on cherche les points d'intérêt, elle doit avoir un seul canal et avoir une profondeur de 8bit ou de 32bits en virgule flottante.
- o Deux images temporaires utilisées par la fonction, elles sont de type IPLo Un vecteur contenant les points d'intérêt, les éléments de ce vecteur sont de type CvPoint2D32f.
- o Le nombre maximal de points d'intérêt à trouver. Une fois la fonction exécutée, cette variable pointe vers le nombre de points d'intérêts trouvés.
- o La qualité minimum des points d'intérêt.
- o La distance (euclidienne) minimum entre deux points.
- o Un masque représenté par une matrice (de type CvMat) spécifiant les pixels de l'image à traiter représentés par les éléments non-nuls de la matrice. Si la valeur est NULL (valeur par défaut), toute l'image est traitée.
- o La taille du voisinage (valeur par défaut = 3).
- o Variable qui indique l'utilisation de l'opérateur Harris (valeur par défaut = 0).
- o Paramètre pour l'opérateur Harris.

3. Expérience

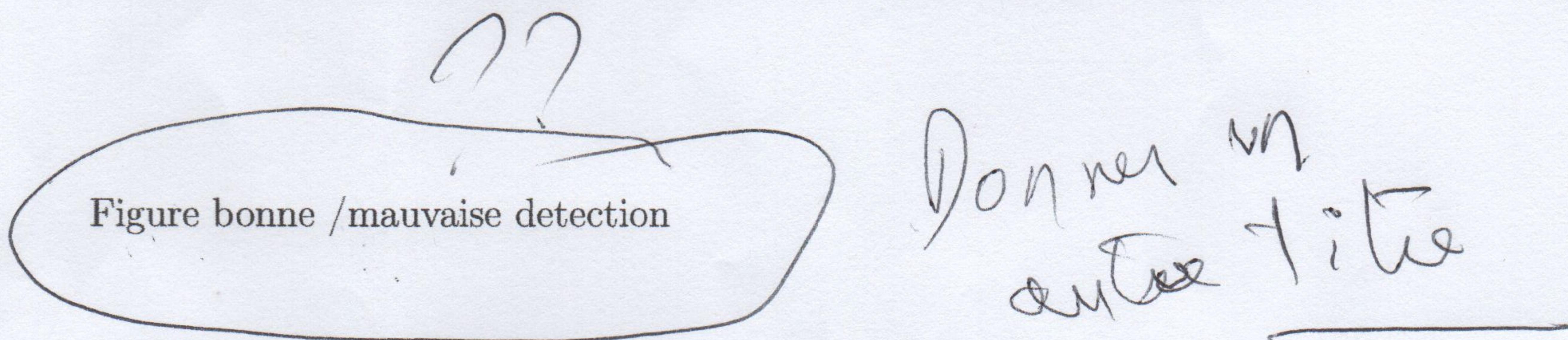
3.1. Détection à l'aide des «Classifieurs de Haar»

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 (connu comme la méthode de Viola-Jones). Etant donné un fichier ou une vidéo en direct, le détecteur de visage examine chaque emplacement de l'image et classifie comme visage ou non visage. Le classifieur utilise des données stockées dans un fichier XML pour décider comment

classifieur chaque localisation image. OpenCV est livré avec plusieurs classifieur 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. Nous pouvons utiliser la fonction cvHaarDetectObjects avec l'un de ces autres détecteur, mais le détecteur des visages de face est le seul qui est très fiable. Pour la détection des données XML, nous pouvons choisir l'un de ces classificateurs Haar Cascade d'OpenCV (dans le répertoire "opencv/data/haarcascade") :

- **haarcascade_frontalface_default.xml**
- **haarcascade_frontalface_alt.xml**
- **haarcascade_frontalface_alt2.xml**
- **haarcascade_frontalface_alt_tree.xml**

Nous avons choisi le fichier "*haarcascade_frontalfacealt.xml*" pour l'application car il offre les meilleures performances pour la détection de visages selon notre expérience. La figure 5-1 présente des échantillons de résultats



Nous verrons maintenant comment ces scores peuvent être améliorés, en effet, pour résoudre le souci des fausses détections, nous avons ajouté un module de prétraitement de l'image avant la phase de détection.

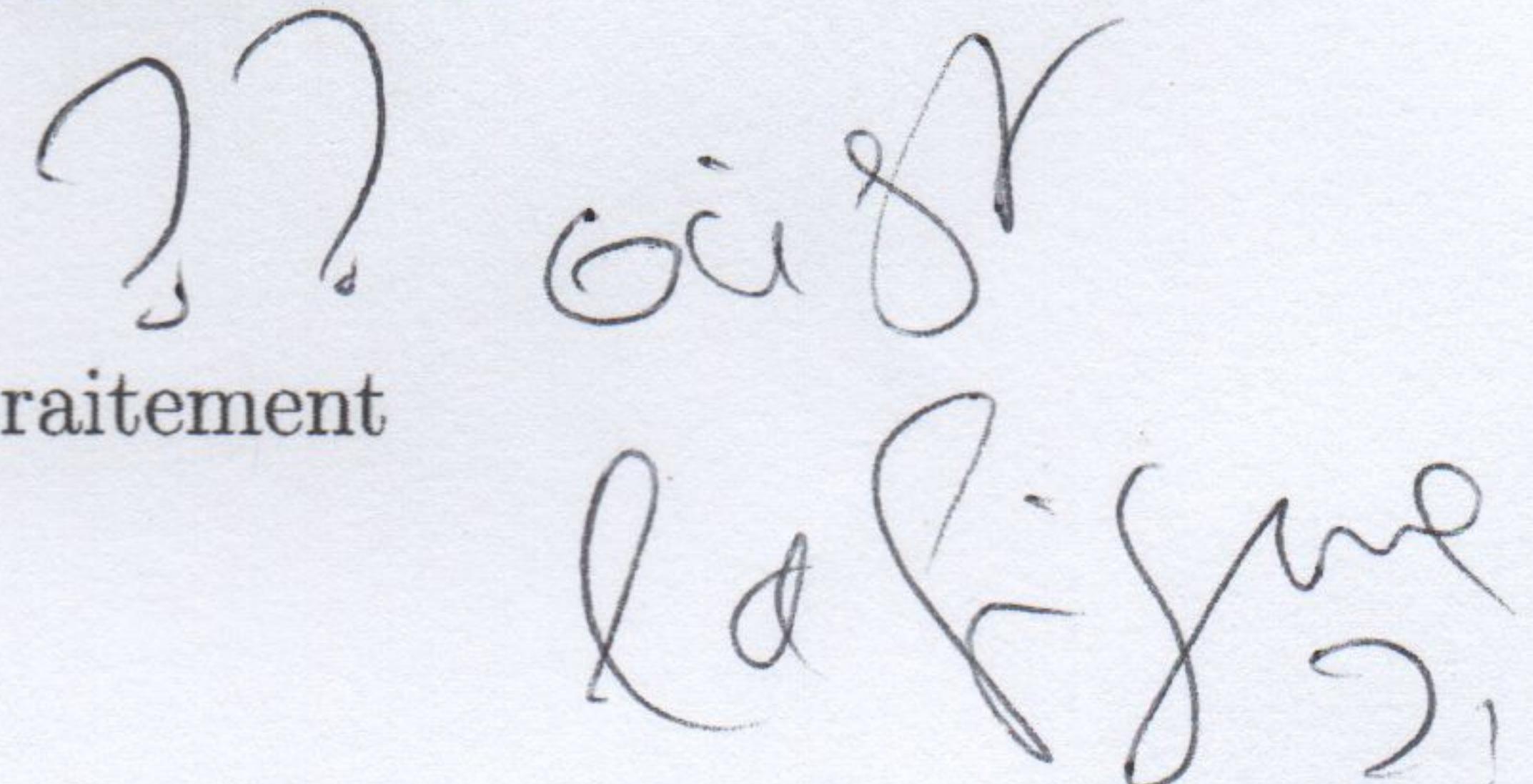
3.2. Prétraitement

Maintenant que nous avons détecté un visage, nous pouvons utiliser cette image de visage pour la reconnaissance faciale. Toutefois, si nous avons essayé de simplement effectuer une reconnaissance de visage directement sur une image normale de photo, nous aurons une précision inférieure à 10%. Il est extrêmement important d'appliquer diverses techniques de prétraitement d'image pour standardiser les images que nous fournissons à un système de reconnaissance faciale. La plupart des algorithmes de reconnaissance de visage sont extrêmement sensibles aux conditions d'éclairage de sorte que s'il a été formé pour reconnaître une personne dans une pièce sombre, il ne sera probablement pas la reconnaître dans une salle lumineuse. 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). C'est pourquoi il est si important d'utiliser un prétraitement d'image avant d'appliquer la reconnaissance faciale. Comme nous l'avons déjà décrit dans le chapitre 4 (conception) le module prétraitement nous permettra d'améliorer l'efficacité de détection. Le prétraitement des images s'effectue en deux temps :

- **Etirement d'histogramme** : Cette action nous a permis d'améliorer le taux de bonne détection à 92% avec un nombre d'images égal à 782 qui contient des visages détectés.
- **Filtre Médian** : L'ajout de filtre médian avec l'action précédente nous a permis d'améliorer le taux de bonne détection à 95,1 % avec un nombre d'images égal à 782 qui contient des visages détectés.

Figure 5.2 : Exemple d'image après prétraitement



Les résultats obtenus sont assez bons, et dans nos tentatives d'amélioration par la création d'un module de prétraitement pour les images, nous nous sommes aperçus que la qualité d'une image est au moins aussi importante que l'algorithme qui l'utilise. Sachant qu'une détection efficace est indispensable pour obtenir de bons résultats de reconnaissance, Nous nous sommes servis d'OpenCV comme d'un outil clé-en-main pour effectuer cette partie. Dans la partie suivante, nous détaillerons la réalisation de la reconnaissance de visage.

0.15 III. Reconnaissance de visage:

A ce niveau du processus, nous sommes en possession d'une image propre, c'est-à-dire libérée du bruit et ne contenant que l'information utile à analyser.

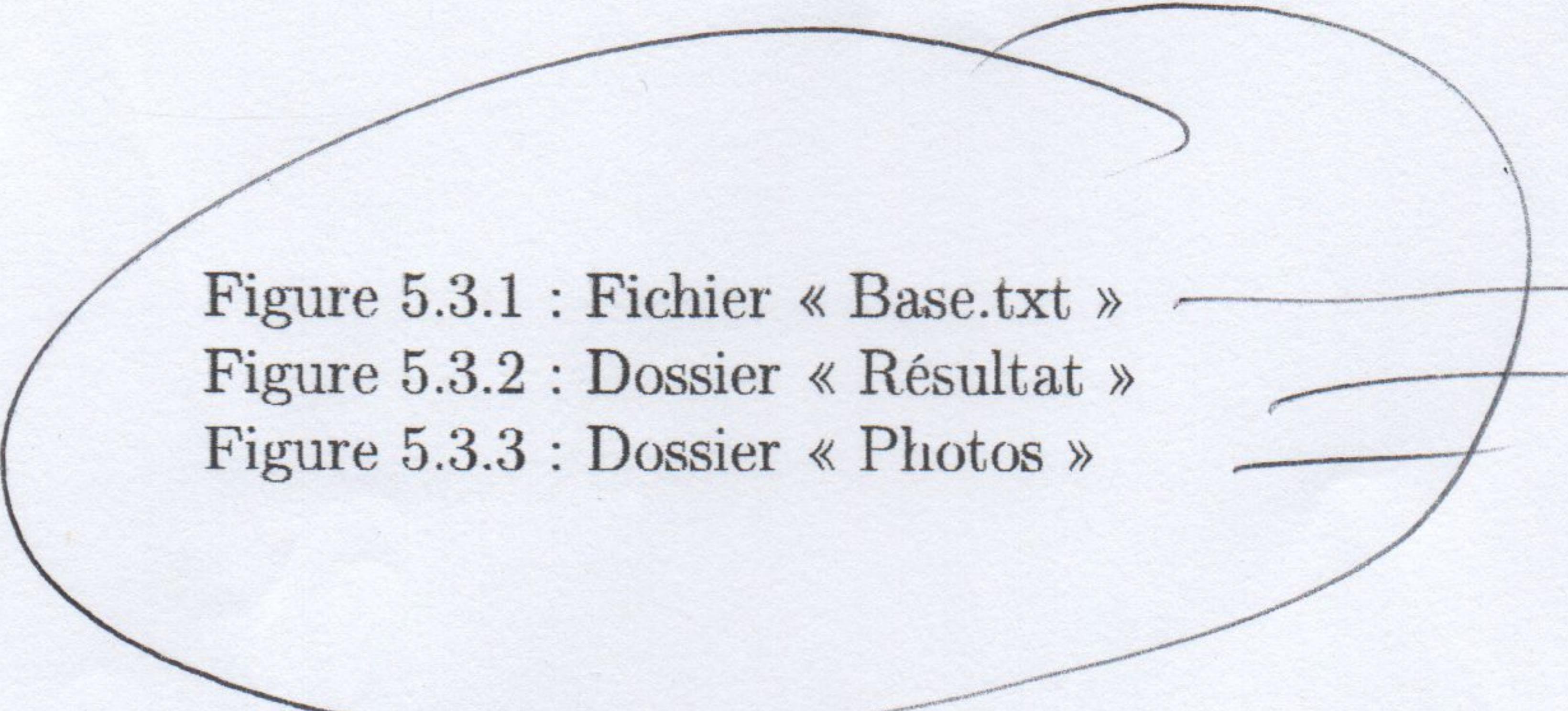
1. Base de données

La base de données est définie de la façon suivante. Le répertoire racine contient deux dossiers : Photos : elles contiennent les photos de la base

de données. Résultat : il contient les images de visage issu après la détection. Un fichier texte : Base.txt : il contient les vecteurs caractéristiques des visages, chaque vecteur représente des informations exclusives pour, à savoir la proportion des yeux et la couleur de peau, lesquels sont iné-luctables pour la phase de décision.

Figure 5.3.1 : Fichier « Base.txt »

de données. Résultat : il contient les images de visage issu après la détection. Un fichier texte : Base.txt : il contient les vecteurs caractéristiques des visages, chaque vecteur représente des informations exclusives pour, à savoir la proportion des yeux et la couleur de peau, lesquels sont inévitables pour la phase de décision.

- 
- Figure 5.3.1 : Fichier « Base.txt »
 - Figure 5.3.2 : Dossier « Résultat »
 - Figure 5.3.3 : Dossier « Photos »

Les caractéristiques de la base sont :

- **Nombre total des individus :** XXX
- **Nombre des images par individu :** XXX
- **Nombre total des images :** XXX
- **Genre :** Contient des images des males et des femelles
- **Origine :** Contient des images de différentes origine
- **Format des images :** 24 bit couleur
- **Type des images :** JPEG
- **Éclairage :** artificiel, un mélange de tungstène et fluorescent

Les tests appliqués à ces bases ont permis de bien configurer la fonction « cvGoodFeaturesToTrack » fournis par la bibliothèque OpenCV afin d'avoir une bonne qualité de reconnaissance. Nous avons bien conclu que le nombre maximal de points d'intérêt à trouver est à égale 100, la qualité minimum des points d'intérêts est égale à 0.09, la distance minimale entre deux points d'intérêt est égale à 4, finalement, la taille de voisinage est égale 3. En ce qui concerne le temps d'exécution, c'est la partie apprentissage qui prend beaucoup de temps (de l'ordre de deux ou trois secondes), la reconnaissance, elle, est rapide, le résultat est quasi immédiat.

0.16 IV.Interface utilisateurs :

A ce niveau, nous nous proposons de compléter notre travail par une interface utilisateur, et constatons que la bibliothèque OpenCV ne fournit

pas des outils performants pour la construction des interfaces graphiques. D'où l'idée d'utiliser le langage Python , pour nous l'intérêt majeur d'utiliser ce langage réside dans la simplicité et l'efficacité de créer les interfaces graphiques.

1. Fenêtre principale :

Figure 5.4 : Menu principale

2. Apprentissage des photos :

Figure 5.5 : Apprentissage des photos

3. Fenêtre résultat :

Figure 5.6 : Apprentissage des photos

0.17 V.Résultat

Les tests réalisés avaient pour but d'observer l'influence des conditions d'illuminations des sujets, de leurs distances par rapport au système de capture d'image et à la position du visage. En réalité, l'effet du premier cas est minimisé par le prétraitement des images et le second par le détecteur de visage mais il y a une influence sur la qualité du visage extrait. Quant au dernier, le système développé n'est pas adapté à la reconnaissance de profil. Il peut tout de même reconnaître des visages tournés d'un léger angle mais avec un taux d'erreur plus élevé.

0.18 Conclusion

Nous avons tout d'abord abordé le processus de détection de visage à l'aide de la bibliothèque OpenCV, et nous avons essayé d'éliminer les fausses alarmes afin d'améliorer le taux de détection. Ensuite, nous avons traité le processus d'identification du visage en proposant une méthode qui consiste à extraire quelques caractéristiques de visage (proportion des coins des yeux et la couleur de peau).

Ok validation ?

Conclusion et perspectives

L'objectif de ce projet 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 robustesse et de rapidité des résultats. La reconnaissance des personnes demeure un problème complexe et non parfaitement résolu, malgré tous les travaux réalisés au cours des dernières années. Plusieurs problèmes incombent à cette tâche d'identification et chacun d'eux est non trivial. De nombreuses conditions réelles affectent la performance d'un système, cependant la détection automatique des visages influence énormément la performance du module d'identification. A travers ce projet nous avons mis en œuvre une approche d'identification du visage, et pour aboutir à ce but, il fallait au préalable aborder un travail de détection du visage. Pour cela nous avons détaillé dans notre projet nos travaux de détection et notre approche pour améliorer le résultat obtenu en ajoutant un module de prétraitement. Bien que notre méthode d'amélioration ait montré de bons résultats, au niveau de l'interpolation de visages non détectés par la librairie OpenCV, elle élimine parfois de vrais visages. Après la phase de détection, nous avons pu aborder la tâche de reconnaissance. Notre apport dans cette tâche délicate, est d'utiliser la notion des points d'intérêt pour reconstruire un modèle de visage. Ce projet ne manque pas de perspectives : pour la tâche de détection, et à partir des visages détectés par la librairie OpenCV, il est intéressant de trouver d'autres méthodes d'élimination des fausses alarmes et de détecter en contre-partie les visages oubliés par la méthode « Viola-Jones ». Nous proposons d'utiliser des approches heuristiques, pour prévoir si une telle détection correspond à un visage ou non, en tenant compte des positions des autres visages. Pour l'identification, nous proposons d'ajouter ou d'améliorer d'autres paramètres aux vecteurs qui caractérisent le visage comme par exemple : la géométrie de la tête et les distances entre les composantes faciale. La reconnaissance des visages fait partie de la biométrie qui est sans doute un domaine d'avenir. Au cours des prochaines décennies, de plus en plus de systèmes verront probablement le jour afin de réaliser une surveillance.

accrue. A travers ce projet, nous avons apporté une modeste contribution dans ce domaine, dont l'intérêt réside dans la mise en place directe de l'application dans le cadre réel.

Bibliographie

[Auteur1 and Auteur2, 2018] Auteur1 and Auteur2 (2018). Titre du papier. *Le titre de la revue*, page 24.

Ex) Auteur, ou nom, année -

Ex:

Ex) Albert Camus, "L'art et l'absurde", 1946