



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**

---

*Réalisé par :*

Sami KAMMOUN      Yosser BATITA

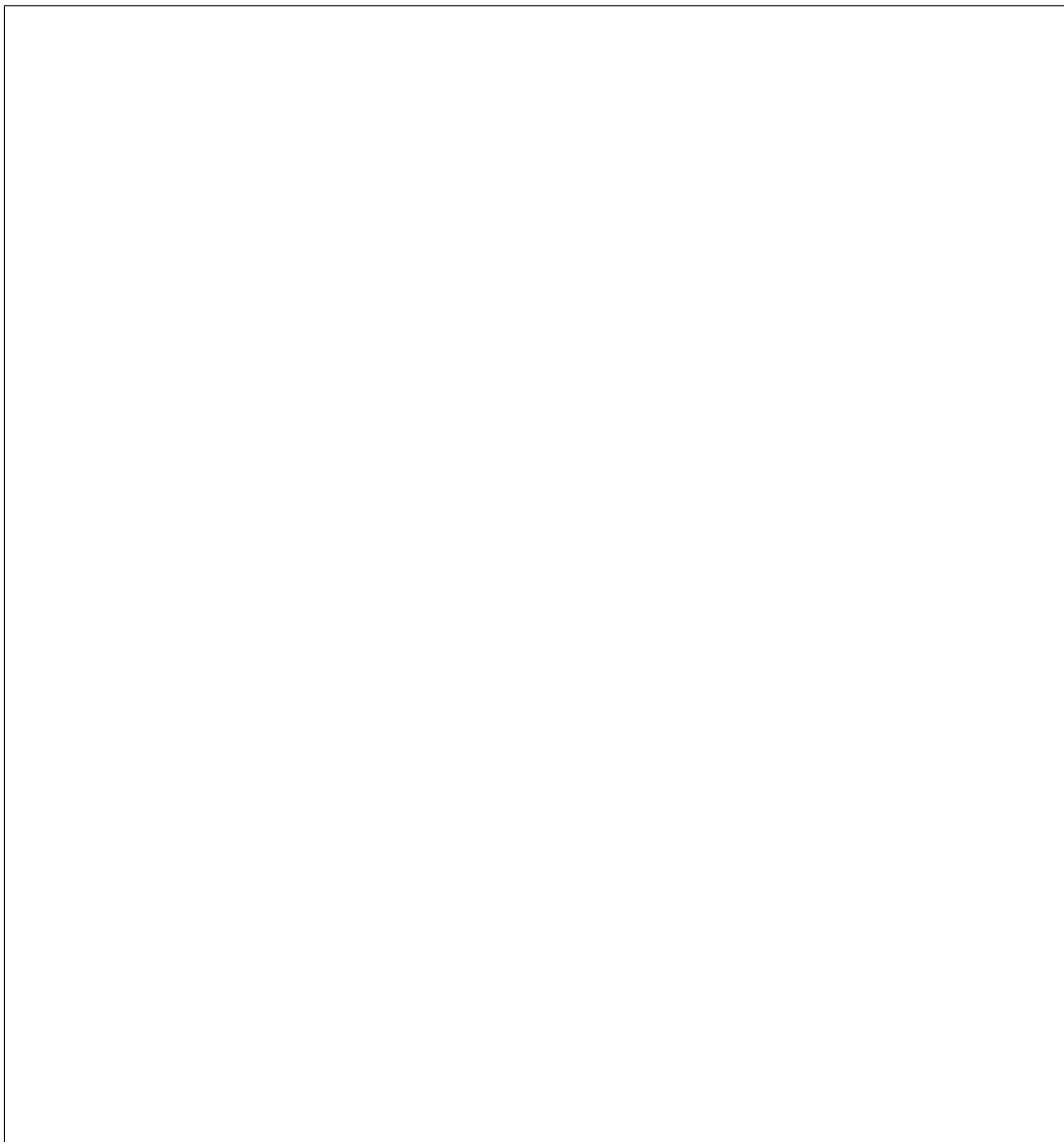
*Encadré par :*

Dr. Inés HAMDI

---

Année Universitaire :2019 /2020

## **Appréciations et signature de l'encadrant**

A large, empty rectangular box with a thin black border, occupying most of the page below the title. It is intended for handwritten text, such as appreciations or signatures.

# Remerciements

*C'est un honneur autant qu'un devoir de remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de notre projet .*

*Tout d'abord, nous tenons à remercier Mme Hamdi Inés, qui nous a encadré, et qui n'a épargné aucun effort pour notre conseil pour que nous puissions mener à bien notre projet.*

*Ainsi, nous exprimons notre profonde gratitude et nous tenons à remercier vivement les membres de jury d'avoir accepter d'évaluer notre travail.*

*Enfin, nous remercions affectueusement tous ceux qui nous ont soutenu dans nos études.*

---

## ملخص

تم تطوير هذا المشروع كجزء من وحدة "مشروع التصميم والتطوير" ، للسنة الثانية من التدريب على هندسة الإعلامية في المدرسة الوطنية لعلوم الإعلامية (ENSI).  
يهدف مشروعنا على تصميم وتطوير أداة ذكية للتعرف على الوجوه.  
**الكلمات المفاتيح:** الذكاء الاصطناعي ، التعلم العميق .

---

### Résumé

Ce projet est élaboré dans le cadre du module de "Projet de Conception et de Développement" du deuxième année de la formation d'ingénieur en informatique à l'École Nationale des Sciences de L'informatique (ENSI).Notre projet avait pour objectif de concevoir et développer un outil intelligent de reconnaissance faciale .

**Key words :** Intelligence artificielle , Reconnaissance faciale , Deep-Learning

---

### Abstract

This project is being developed as part of the "Conception and Developpement project" for the second year of Computer-Science engineering at the National School of Computer Science (ENSI). Our project is to make an intelligent tool of facial recognition.

**Key words :** Artificial intelligence, facial recognition, Deep-Learning

---

# Table des matières

<b>Introduction</b>	<b>1</b>
<b>Chapitre 1 :Etude de l'existant</b>	<b>3</b>
1.    Introduction . . . . .	3
2.    Etude de l'existant . . . . .	3
2.1    L'application FindFace . . . . .	3
2.2    L'application BioID . . . . .	3
2.3    L'application ClearView . . . . .	4
3.    Synthèse . . . . .	4
4.    conclusion . . . . .	4
<b>Chapitre 2 :Analyse et specification de besoins</b>	<b>6</b>
1.    Introduction . . . . .	6
2.    Besoins fonctionnels . . . . .	6
3.    Besoins non fonctionnels . . . . .	7
4.    Diagramme des cas d'utilisation . . . . .	7
5.    Diagramme de séquence . . . . .	9
6.    Conclusion . . . . .	10
<b>Chapitre 3 :Conception</b>	<b>11</b>
1.    Introduction . . . . .	11
2.    Modèle conceptuel proposé . . . . .	11
3.    Module de prétraitement . . . . .	12
4.    Détection de visage . . . . .	13
4.1    Fonction de classification . . . . .	14
4.2    Image intégrale . . . . .	14
4.3    Bloc de décision . . . . .	14
5.    Reconnaissance de visage . . . . .	14
5.1    Détection des composantes faciales . . . . .	15
5.2    Extraction des paramètres . . . . .	16
5.3    Signature . . . . .	17
5.4    Identification . . . . .	17
6.    Conclusion . . . . .	17

<b>Chapitre 4 :Réalisation</b>	<b>18</b>
1. Introduction . . . . .	18
2. Environnement de travail . . . . .	18
3. Présentation de la bibliothèque OpenCV . . . . .	18
3.1 Classifieur OpenCV . . . . .	19
3.2 Implémentation . . . . .	19
3.3 Expérience . . . . .	20
4. Reconnaissance de visage . . . . .	22
4.1 Base de données . . . . .	22
5. Interface utilisateurs . . . . .	26
5.1 Fenêtre principale . . . . .	26
5.2 Apprentissage des photos . . . . .	2
5.3 Fenêtre résultat . . . . .	27
6. Résultat . . . . .	28
6. Conclusion . . . . .	30
<b>Conclusion et perspectives</b>	<b>31</b>
<b>Bibliographie</b>	<b>33</b>
<b>Netographie</b>	<b>34</b>

# Table des figures

2.1	Diagramme de cas d'utilisation « Choisir la meilleure image » . . . . .	15
2.2	Diagramme de cas d'utilisation « Reconnaître les visages » . . . . .	16
2.3	Diagramme de cas d'utilisation « Gérer la base de données » . . . . .	17
2.4	Diagramme de séquence . . . . .	18
3.1	Schéma fonctionnel . . . . .	21
3.2	Filtre Médian . . . . .	22
3.3	Étapes de la détection . . . . .	23
3.4	Étapes de reconnaissance faciale . . . . .	24
4.1	Dossier « Photos :Paul Wilsey» . . . . .	31
4.2	Résultat détection « Photos :Paul Wilsey» . . . . .	31
4.3	Dossier « Photos :Phoebe Tonkin» . . . . .	32
4.4	Résultat détection « Photos :Phoebe Tonkin» . . . . .	32
4.5	Dossier « Photos :Lan Somerhalder» . . . . .	33
4.6	Résultat détection « Photos :Lan Somerhalde» . . . . .	33
4.7	Dossier « Photos :Nina Dorbev» . . . . .	34
4.8	Résultat détection « Photos :Nina Dorbev» . . . . .	34
4.9	Menu principale . . . . .	36
4.10	Apprentissage des photos . . . . .	36
4.11	Fenêtre résultat . . . . .	38

# Liste des sigles et acronymes

[**OPENCV**] *Open Source Computer Vision Library*

[**XML**] *eXtensible Markup Language*

[**UML**] *Unified Modeling Language*

[**PIN**] *Personal Identification Number*

[**BioID**] *proximity-dependent biotin identification*

[**NYT**] *The New York Times*

[**JPG**] *Joint Photographic Experts Group ( Le type d'images utilisé)*

[**HighGUI**] *high-level graphical user interface*

[**CvMat**] *Computer Vision Matrix*

[**Haar**] *High Altitude Aerial Reconnaissance*

[**Ipl**] *Intel Processing Library*

# Introduction

La reconnaissance d'une personne à partir de son visage est une tâche facile à réaliser pour un être humain. Est-ce que c'est le cas pour la machine ? Comme réponse à la question de l'automatisation de la reconnaissance beaucoup de travaux de recherches sont générés au cours des dernières années. Actuellement, la surveillance et la sécurité sont devenus un enjeu majeur c'est pourquoi l'identification de la personne de façon efficace et exacte est devenu une nécessité. Pour identifier quelqu'un, il existe plusieurs outils et méthodes, prenons comme exemple les méthodes « Token-Based » et « Knowledge-based ». Ces deux cas sont les plus répandues dans le monde de la reconnaissance faciale. Pour obtenir une sécurité accrue ces deux manières d'identification se complètent. Néanmoins, elles ont chacune leurs impuissances par exemple pour le premier cas le mot de passe peut être deviné par une autre personne ou aussi l'utilisateur peut oublier son mot de passe .Pour le second cas, l'outil d'identification peut être perdu ou volé par exemple le badge, la pièce d'identité ou la clef .

Comme réponse à ce problème, plusieurs alternatives sont envisageables, notamment les caractéristiques biométriques. L'être humain est caractérisé par plusieurs traits physiques propres et uniques, c'est pour cela qu'on trouve plusieurs applications utilisant comme paramètres les données biométriques de l'individu. Ces données, étant universelles, permettent l'identification et la reconnaissance de toutes les personnes. D'autre part, elles sont mesurables et particulières : deux personnes ne peuvent pas avoir exactement les mêmes caractéristiques. Elles sont de plus permanentes ce qui implique qu'elles ne varient pas ou peu au cours du temps.

En outre, la biométrie permet une exploitation plus facile de plusieurs données personnelles. Dans notre quotidien, le visage constitue un ensemble de données qui permettent, à eux seules, de différencier tout couple de personnes avec une forte probabilité. Il est donc naturel de l'utiliser, spécialement qu'il est facilement identifiable.

Jusqu'à aujourd'hui, la reconnaissance faciale demeure encore le leader des structures dans le domaine de Computer-Vision, spécialement celui de la sécurité et la surveillance, et qui devient de plus en plus viral créant une perpétuelle innovation. Le passage de la détection manuelle à l'automatique, citons comme exemples : la télésurveillance, le contrôle d'accès, La sécurité des bâtiments, etc est le but d'un système de reconnaissance de visages. Le présent rapport est constitué de quatre chapitres : Le premier, intitulé « Etude de l'existant », vise à avoir une idée claire et précise sur l'existant quel qu'il soit. Le second est l'« Analyse et spécification des besoins » Il a pour objectif l'analyse de spécifier les besoin

fonctionnelles et non-fonctionnelles du projet, donnant ainsi une idée sur la possibilité de réalisation du projet, le troisième, au titre de « Conception », va traiter aussi bien la conception globale que détaillée. Cette dernière sera mieux expliquée dans le dernier chapitre qui va englober les réalisations effectuées ainsi que l'ensemble des tests d'exécution. Finalement une conclusion clôturera le rapport.

# Chapitre 1

## Etude de l'existant

### 1.1 Introduction

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.

### 1.2 Étude de l'existant

#### 1.2.1 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 satisfaire les deux conditions : la bonne qualité de photo et la disposition d'un compte VKontakte pour la personne photographiée. Ensuite, c'est un algorithme qui prend le relai fondé sur la technologie de réseau de neurones artificiels (le même modèle que pour nos neurones qui servent à apprendre). Cet algorithme présente 80 caractéristiques comme le sexe, l'âge ou la forme du visage. Ainsi, l'algorithme peut faire la reconnaissance de l'ensemble des caractéristiques du visage humain. Puis, le scan de la photographie est analysé après il est comparé avec le réseau social russe VKontakte utilisé comme base de données de photos pour l'application FindFace. D'après les réalisateurs du projet, leur algorithme serait également cohérent avec les bases de données de Facebook ou Instagram ! D'après le fondateur de la start-up, pour la base de 299 millions de photos, le taux de réussite est de 70 %.

#### 1.2.2 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.

### 1.2.3 L'application ClearView

L'application créée par l'Américain Hoan Ton-That, a partir d'une simple photo, l'application peut retrouver le nom, et aussi dans la majorité des cas l'âge, des photos des proches, de vos compétitions sportives, parfois même des photos que vous aviez oubliées ! En fait, l'application est capable de naviguer dans les contenus que vous postez sur les réseaux sociaux, et ce, sans avoir votre autorisation. Extrêmement efficace, mais également redoutable puisque ClearView est utilisée par pas moins de 599 agences gouvernementales, selon le NYT, qui a révélé la présence de cette application au grand public

## 1.3 Synthèse

Ce tableau résume les avantages et les inconvénients des applications de reconnaissance faciale précédemment cité

Application	Avantages	Inconvénients
FindFace	- facilite le 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.

## 1.4 Conclusion

La recherche dans le domaine de la détection et reconnaissance de visages subsiste à être innovante et très riche, nous trouvons ainsi une multitude de méthodes et d'outils spécialisés dans la détection et reconnaissance faciale. Quant à la détection de visage, comme on cherche à être plus efficace et on tend d'optimiser le temps nécessaire pour traiter un à plusieurs visages par image et dans le but d'éliminer tout enchevêtrement afin

de respecter les exigences de notre projet, nous avons opté pour l'utilisation la bibliothèque OpenCV de Python qui contient un algorithme de détection qui demeure un outil meilleur dans le monde de la reconnaissance faciale par sa rapidité et sa crédibilité au niveau de la détection de visage. Quant à la reconnaissance de visage, on a conclu qu'il serait meilleur d'adopter une approche basée sur un alliage entre la méthode de Paul Viola and Michael Jones et la méthode SVM qui sera plus détaillé dans les parties à venir.

# Chapitre 2

## Analyse et spécification des besoins

### 2.1 Introduction

Dans ce chapitre, on va mettre les points sur l'ensembles de besoins fonctionnels et non-fonctionnels de ce projet. Ceci est mieux élaborer par l'intermédiaire de diagrammes de séquence, en plus de cas d'utilisation.

### 2.2 Besoins fonctionnels

Dans notre projet, on a deux parties qui présentent les piliers :

- La détection de visage
- La reconnaissance de visage

**La détection de visage :** Le but de cette partie est de délimiter les visages dans des rectangles pour faciliter la partie suivante en donnant un intervalle plus étroci pour la tâche de détections.

**La construction d'une base de descripteurs associés aux visages :** Ceci signifie qu'on veut mathématiser les traits de la personne pour pouvoir donner aux machines un moyen pour faire la reconnaissance des différentes personnes. Par suite, optimiser le nombre de trait est important pour avoir un meilleur rapport Temps-Taux.

**La reconnaissance de visage :** Cherchons un match dans la base de données, notre application finit par donner le meilleur Match pour une photo ou un frame de vidéo donnée. Cette partie doit pouvoir nous donner un taux de fiabilité pour la personne qu'elle identifie.

## 2.3 Besoins non fonctionnels

### **Interface Graphique de l'Utilisateur :**

L'application doit avoir une interface qui facilite l'utilisation des méthodes et facilite l'ajout des personnes et la détection. Un utilisateur doit pouvoir utiliser l'application dans un maximum de 5minutes.

### **Documentation :**

Accompagnant le projet, des commentaires et des explications qui clarifie le fonctionnement de l'application.

### **Rapidité :**

Chaque photo ne doit dépasser 2 secondes pour donner un résultat, et pour chaque vidéo, le système doit être capable de détecter en temps réel une vidéo de 10fps avec un délai de 2secondes.

Dans cette partie nous avons détaillé les principaux besoins fonctionnels et non-fonctionnelles, et maintenant, nous passons au diagrammes UML de notre application qui vont clarifier de plus la construction et les services offertes.

## 2.4 Diagramme des cas d'utilisation

Les diagrammes de cas d'utilisation présentent une intermédiaire pour mieux comprendre les besoins fonctionnels du système de coté pratique. Ils définissent aussi les interactions entres les acteurs du système et donne la façon de l'application à donner les résultats.

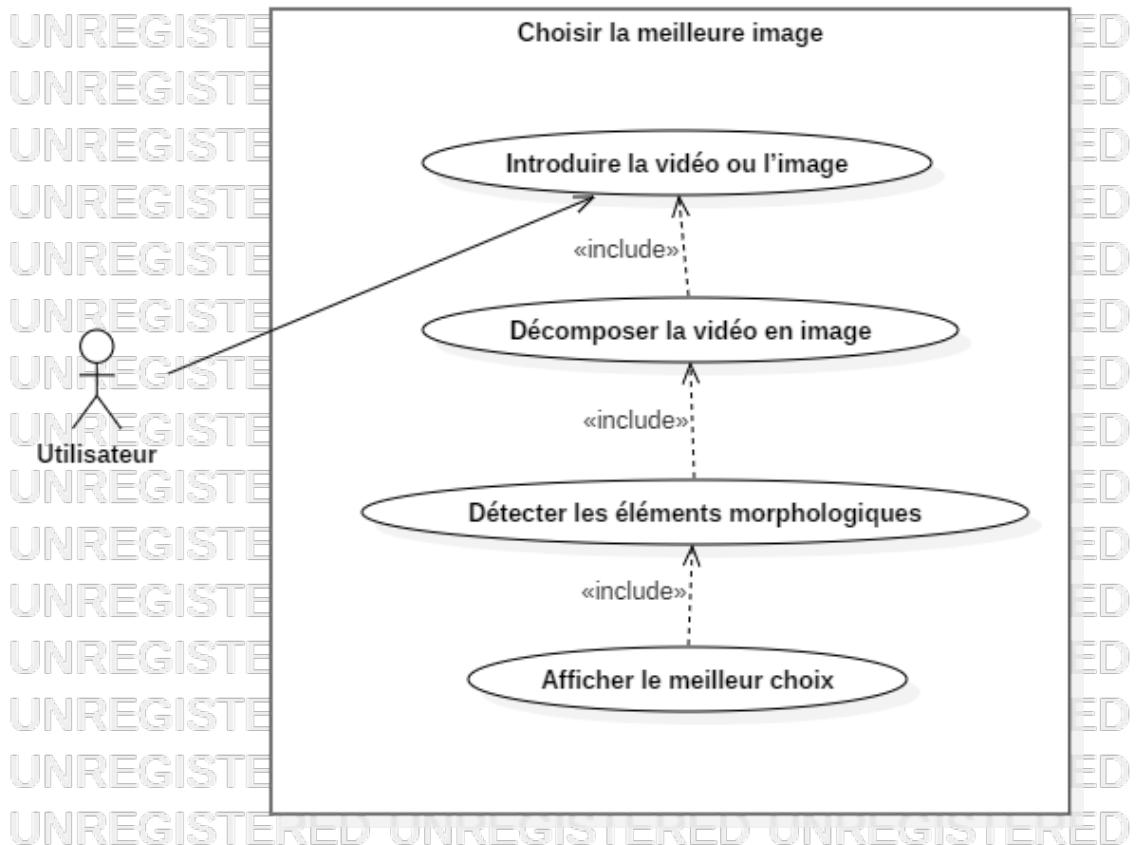


FIGURE 2.1 – Diagramme de cas d'utilisation « Choisir la meilleure image »

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.

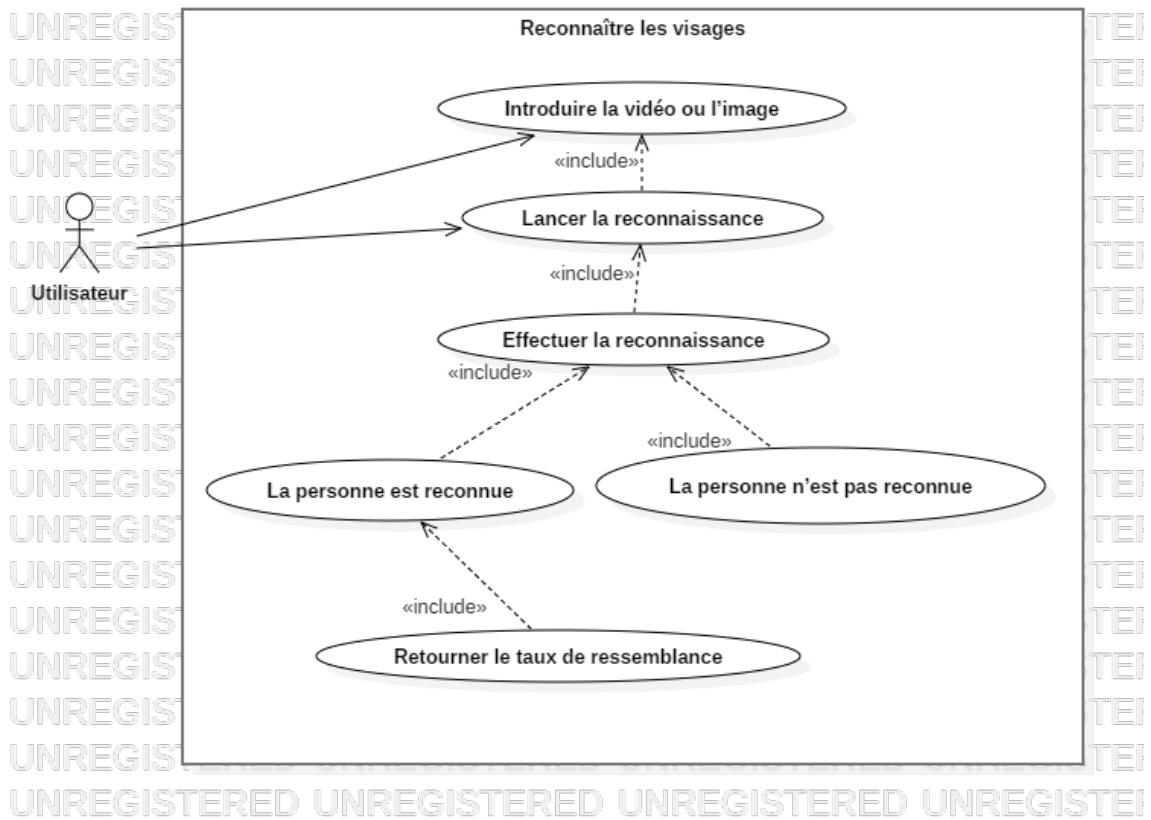


FIGURE 2.2 – Diagramme de cas d'utilisation « Reconnaître les visages »

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

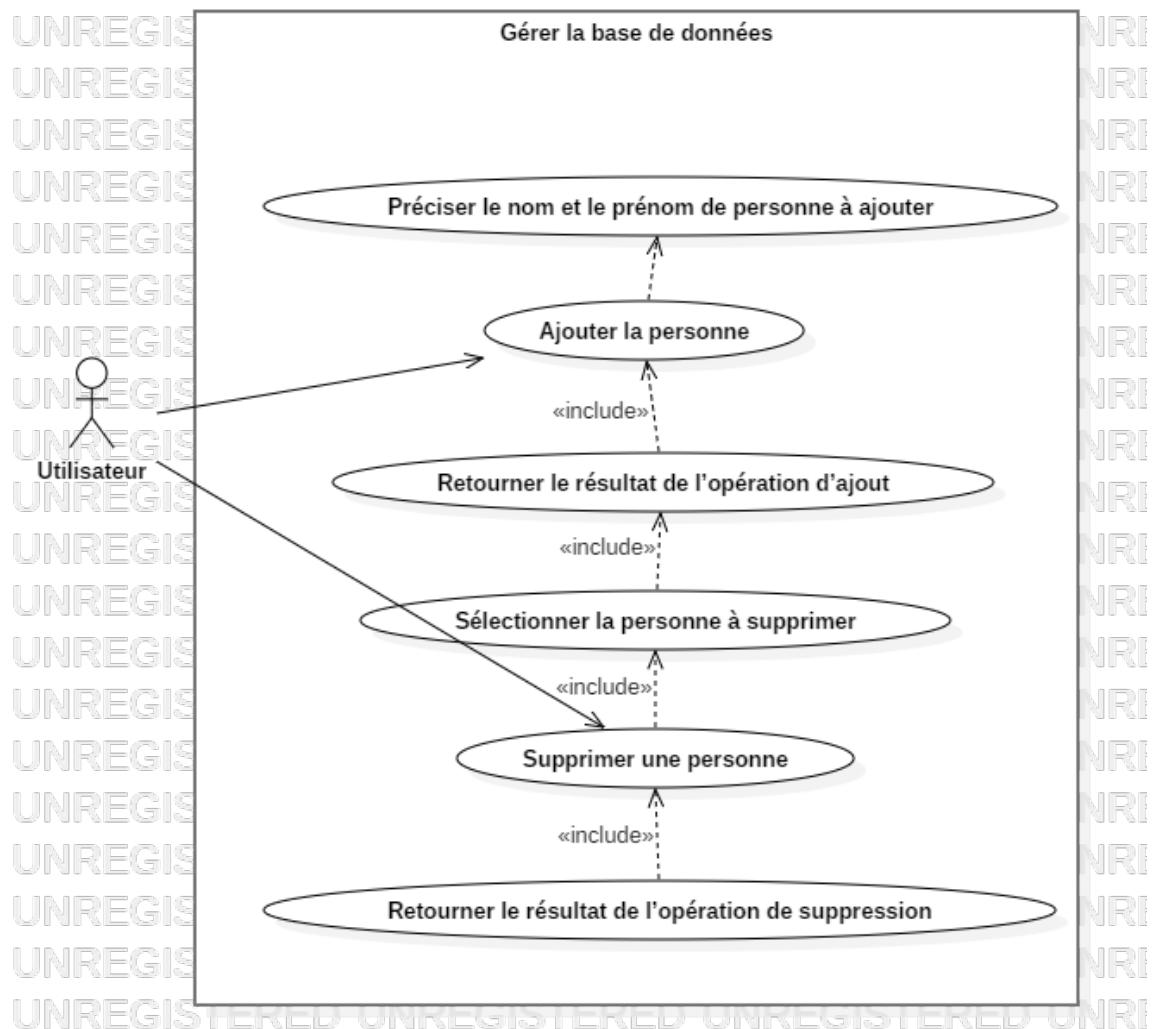


FIGURE 2.3 – Diagramme de cas d'utilisation « Gérer la base de données »

## 2.5 Diagramme de séquence

Dans ce diagramme de séquence, l'utilisateur introduit l'image ou la vidéo à analyser, après il y a lancement de 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.

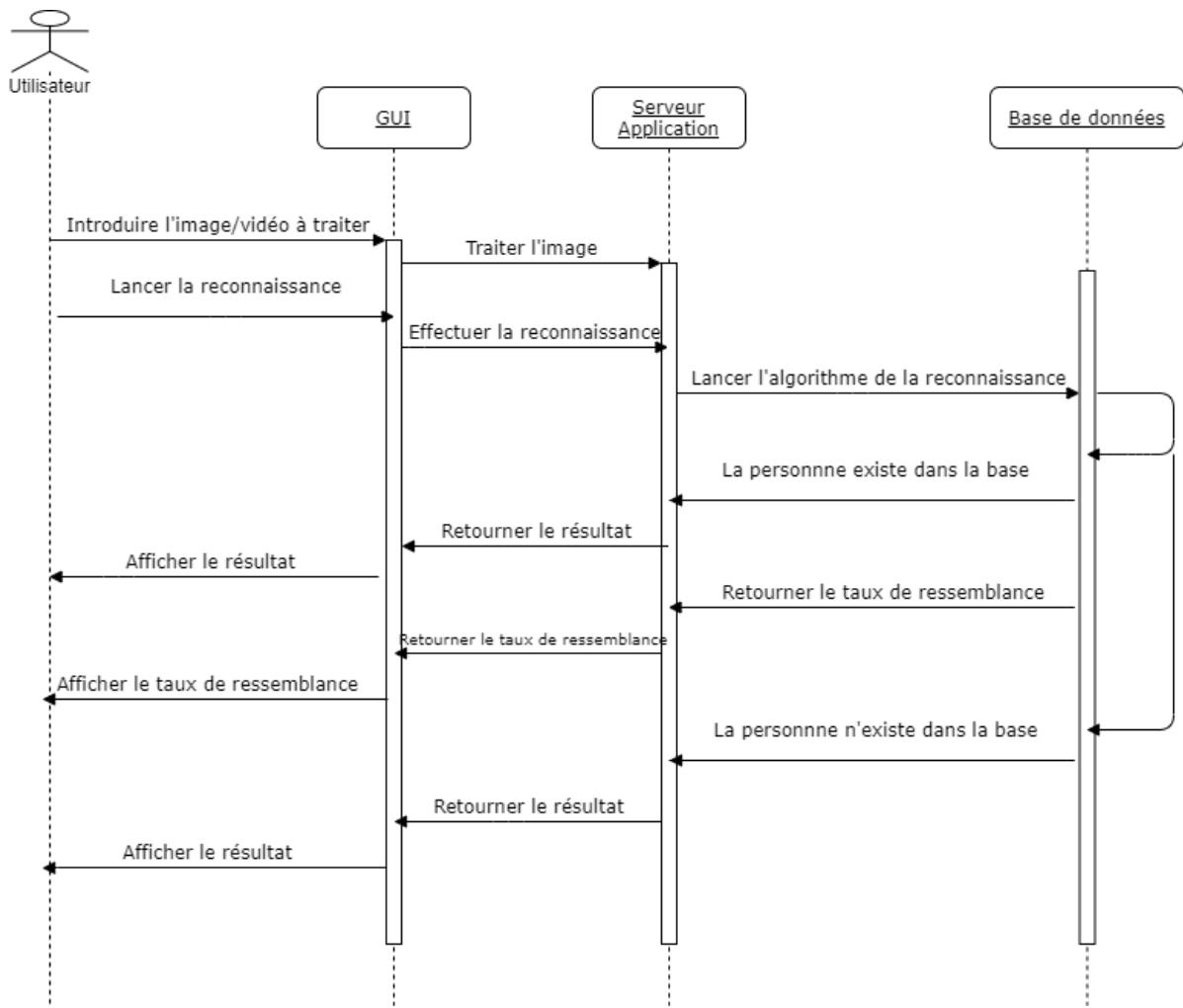


FIGURE 2.4 – Diagramme de séquence

## 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 chapitre suivant, nous allons aborder la conception de l'application développée.

# **Chapitre 3**

## **Conception**

### **3.1 Introduction**

Dans le chapitre précédent on a bien détaillé les besoins fonctionnels et non fonctionnels de notre projet. Pour la partie qui suit nous allons nous focaliser sur la conception de notre application.

### **3.2 Architecture proposée**

Afin de décompliquer le projet, on a adopté une organisation par module. A cet effet, le projet est constitué en 3 modules : « Module de pré-traitement », « Module de détection » et « Module de reconnaissance ». Leurs missions consistent à standardiser chaque photo, détecter le visage et les yeux, calculer les paramètres et finalement la reconnaître.

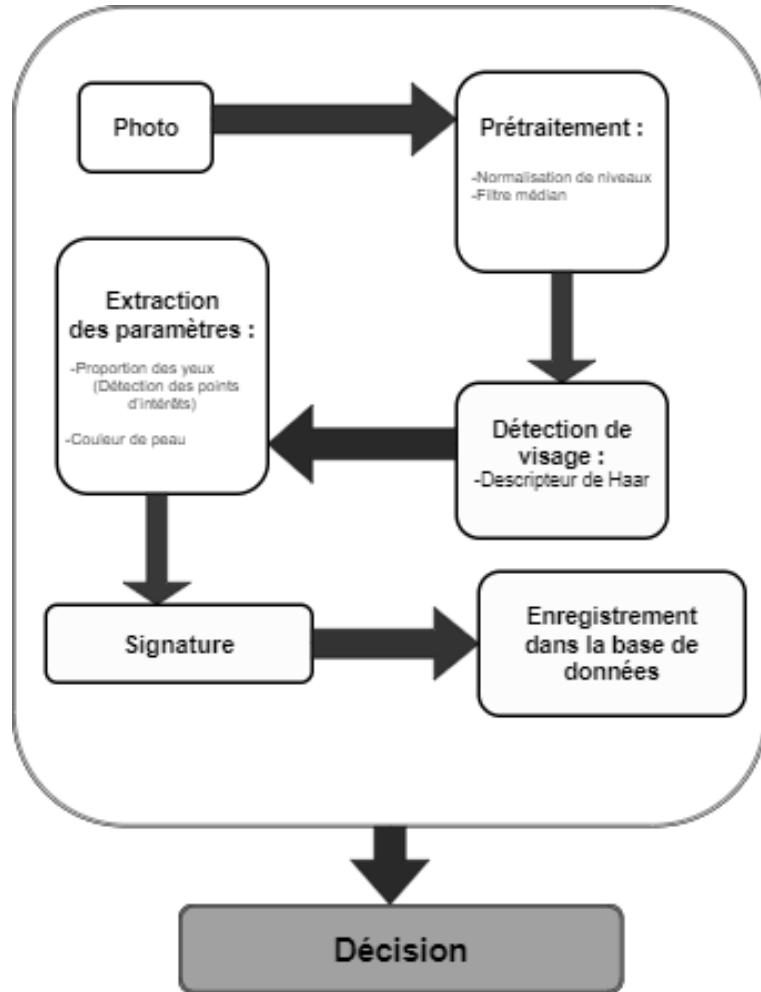


FIGURE 3.1 – Schéma fonctionnel

### 3.3 Module de pré-traitement

L'éclairage, les ombres ou encore les résolutions peuvent changer d'une image à une autre. De ce fait, une normalisation des données s'impose et l'ajustement des variations de contraste devient nécessaire pour qu'elles représentent ainsi les caractéristiques du visage et non de son environnement.

Après avoir détecté les images, la normalisation se fait en deux étapes : Dans un premier temps, on va réaliser l'étirement d'histogramme, dans un second temps, on va appliquer un filtre médian.

- **Étirement d'histogramme :**

les pixels seront ajusté sur la largeur de l'histogramme. Pour ceci, on va augmenter le contraste et les intensités sur l'échelle des valeurs disponibles. Ceci revient à transformer 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 est présent pour éliminer le « bruit ». Ceci revient à éliminer les pixels dont l'intensité est trop faible ou trop forte ce qui peut endommager la détection.

Le principe est le suivant : pour chaque pixel, il le ré-évalue par la valeur la moyenne des niveaux présent dans les pixels lui-entourant, c'est-à-dire les pixels avec lesquelles il forme un cube. Cette méthode est particulièrement pertinente quand le bruit est intense.

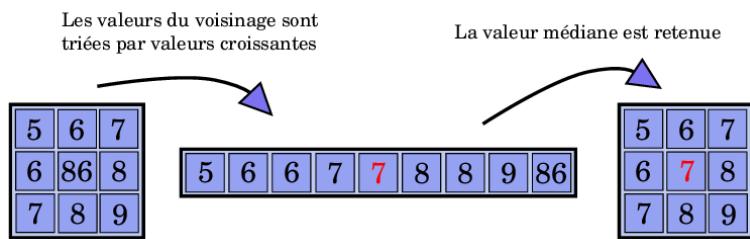


FIGURE 3.2 – Filtre Médian

### 3.4 Détection de visage

Dans la partie qui suit, on aura recours à la méthode de « Viola Jones ». Grâce à sa phase d'apprentissage, elle permet de donner une classification meilleure dans la détection. En réalité la phase d'apprentissage est déjà bel et bien faite par la bibliothèque « OpenCV », elle se présente alors sous la forme d'un fichier « XML ».

L'algorithme de classification de détection (visage ou non) va, par suite, exploiter les résultats de ce fichier pour la réalisation d'une classification en visages et non-visages.

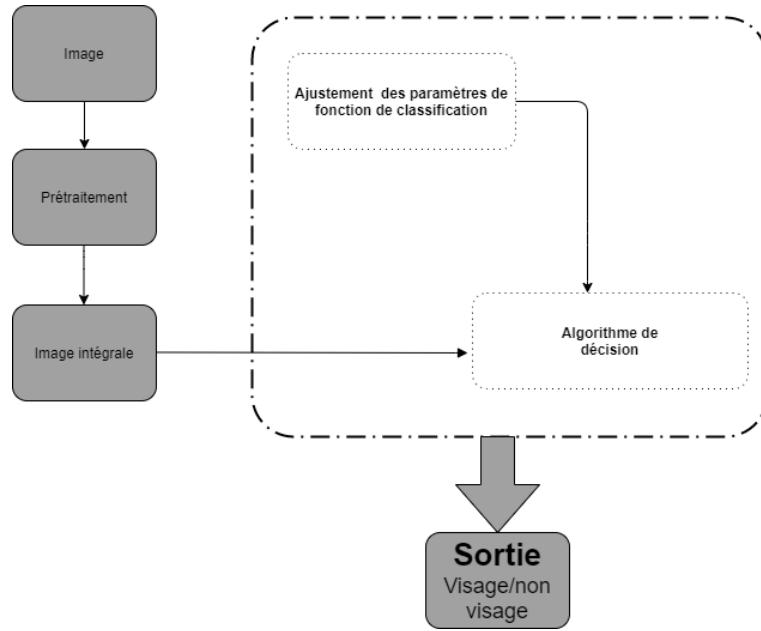


FIGURE 3.3 – Étapes de la détection

### 3.4.1 Classification

La classification va se faire par un algorithme qui utilise le Machine Learning CNN en plus de celui de Viola Jones. Elle utilise une sorte d’arbre de décision pour chaque étape de classification selon le nombre de points d’intérêt.

### 3.4.2 Image intégrale

« ImageIntégrante » est une technique développée par Viola Jones qui permet de détecter si oui ou pas les descripteur de l’algorithme Haar sont présent. La valeur intégrante est donnée par la somme des pixels formant dans le coin gauche en haut. Ceci est fait de haut en bas et de gauche à droite.

### 3.4.3 Bloc de décision

Les deux parties précédentes forment les entrés d’une boucle. Pour cela, la partie balayé s’agrandi incrémentalement à chaque boucle, ceci par un facteur déjà choisi auparavant. Par suite, la fonction de classification se transforme dans chaque itération. Cette transformation propose de redimensionner les descripteurs. Le but intégral de cette étape est de trouver tout visage présent dans une photo ou frame donnée.

## 3.5 Reconnaissance de visage

Dans cette partie nous définissons la méthodologie d'identification de visage. Nous n'avons pas au préalable un modèle ou des trait calculé. Toutefois, notre « Modèle » est en fait un vecteur défini et calculé pour chaque visage pour enregistrer les traits qui lui caractérisent. De suite, on va décrire deux étapes, la préparation, suivi par la phase de reconnaissance.

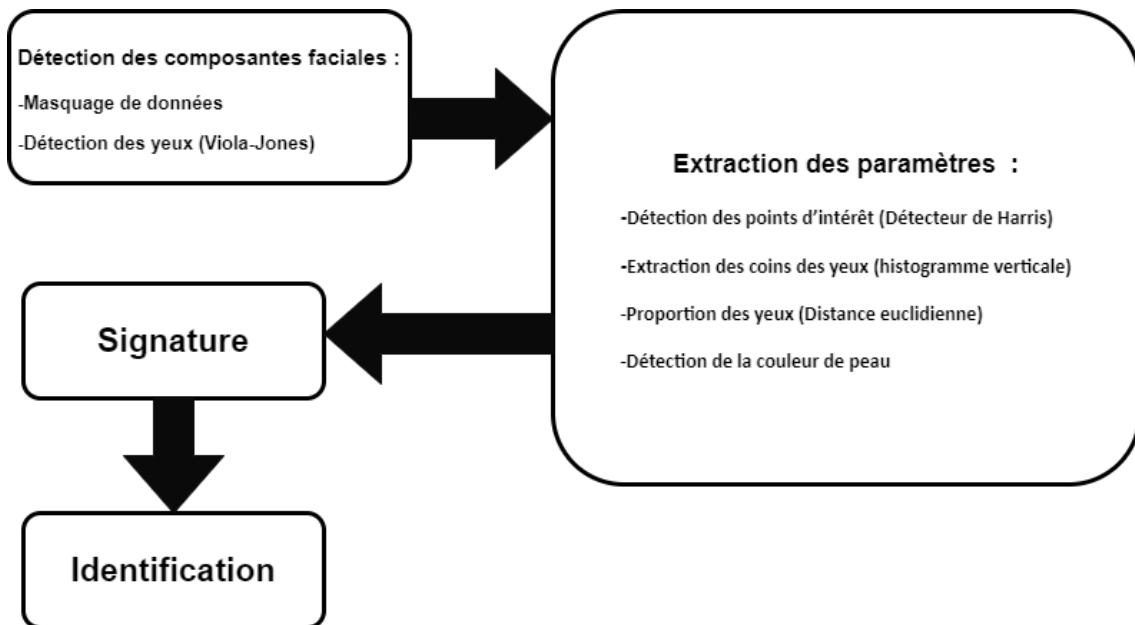


FIGURE 3.4 – Étapes de reconnaissance faciale

### 3.5.1 Détection biométrique du visage

Cette phase se traduit en deux étapes :

- **Masquage de donnée** : Ayant une forme plus ou moins ovale, le visage ne peut être que dans des parties spécifiques, donc le masquage est une technique qui élimine des parties de la photo, notamment les quatre cotés puisque ces parties seront dans la plupart des cas une partie du décor ou du cheveu.
- **Détection des yeux** : De la même façon que la détection faciale, l'algorithme de Viola Jones détecte les yeux et le nez.

### **3.5.2 Extraction des paramètres**

Ayant créé des signatures pour toute personne à identifier, il vient alors le moment de test. Le test se lance lors de l'implantation d'une image au système où on va maintenant calculer le vecteur associer à la nouvelle photo par extraction de paramètre. Puis, on va calculer la différence, en valeur absolue, les données de ce vecteur par rapport à ceux enregistré dans notre base de données et la personne avec le minimum de différence est présenté comme la personne présente dans la photo donnée.

## **3.6 Conclusion**

A la fin de ce chapitre nous avons pu expliquer le fonctionnement de notre projet. 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

### 4.1 Introduction

Dans ce chapitre, nous allons détailler notre chemin et les méthodes que nous avons utilisé pour réaliser soit la détection, soit l'identification du visage. Naturellement, nous allons aussi figurer les résultats de notre travail ainsi que les limites de notre projet. OpenCV était notre point de départ puisqu'il offre une bibliothèque d'outils incomparable et facile à comprendre, notamment la méthode de « Viola Jones ». CNN à lui a été utilisé pour améliorer le taux et la qualité de détection puisqu'il utilise le DeepLearning. En outre, SVM était notre classifieur de choix pour la classification et la reconnaissance de visage, toujours utilisant le Machine Learning.

### 4.2 Environnement de travail

#### 4.2.1 Environnement de travail matériel

Pour ce projet, nous nous avions comme machine un PC avec les caractéristiques suivantes : Processeur : Intel R Core TM i7-7200U CPU @ 2.70GHz 2.90GHz RAM : 8.00 Gø Windows 10 Webcam intégrée 4.3

#### 4.2.2 Environnement de travail logiciel

On a, dans ce projet, opté pour la version 3.4 de Python :

- Python est un langage de programmation flexible et extensible qui peut être utilisé dans plusieurs domaines avec un taux haut d'efficacité
- C'est un langage simple qui peut être facilement compris et qui, à un niveau expert, est largement utilisé

- Les programmes écrit en Python sont facilement décomposés en module, ce qui encourage la réutilisabilité et la réutilisation des codes. Ceci est très bénéfique dans la totalité des domaines de l'informatique
- Il offre un ensemble de bibliothèque incomparable avec les autres logiciels de programmation avec une spécialisation importante ce qui facilite tellement son utilisation

### 4.2.3 Implémentations

#### Installation

L'une des attractions de python est la facilité de son installation et son utilisation, ceci dit, on a opté pour la version 3.4, et avec la commande pip, il est simple de télécharger les bibliothèques dont on a besoin via l'invite de commande de Windows ou le terminal sur Linux. Exemple : pip install opencv-python

## 4.3 Présentation des bibliothèques

Dans ce projet, nous avons utilisé une multitude de bibliothèques, dont les utilisations sont diverses, allons de la création de l'interface graphique à la détection de visage, à la reconnaissance de personne, à l'affichage des résultats.

### 4.3.1 La bibliothèque OpenCV

OpenCV est une bibliothèque « Open Source » de traitement d'image qui, implémenté par Intel, viens comme outil important sur Windows et Linux. Elle est bien optimisée pour l'usage temps-réel. Cette bibliothèque offre une variété d'outils pour le traitement d'image allons de la lecture et l'écriture des images et leurs affichages, vers le calcul et l'affichage des histogramme soit en couleur, soit en niveau de gris. De plus, elle offre des options de lissage et de filtrage. Elle demeure aussi une option importante pour sa variété dans le traitement vidéo qui, en plus des outils de photo, ajoute un ensemble de méthode de joindre les vidéos. Dans les deux cas, elle offre, par l'utilisation de la méthode de « Viola Jones », un outil de détection faciale.

### 4.3.2 La bibliothèque CNN

Une approche infaillible pour la reconnaissance faciale basée sur Deep-Learning utilisant le Convolutional Neural Networks (CNN) avec Dlib.

#### Dlib

c'est une bibliothèque C ++ open source, implémenté en Python, implémentant une variété d'algorithmes d'apprentissage automatique, y compris la classification, la régression, le clustering, la transformation de données et la prédiction.

L'avantage de ceci est que les besoins en mémoire sont réduits et le nombre de paramètres à entraîner est réduit en conséquence. La performance de l'algorithme est donc améliorée. En même temps, dans d'autres algorithmes d'apprentissage automatique, les images nécessitent que nous effectuons un pré-traitement ou une extraction de fonctionnalités. Cependant, nous avons rarement besoin de faire ces opérations lorsqu'on utilise CNN pour le traitement d'image. C'est quelque chose que les autres algorithmes d'apprentissage automatique ne peuvent pas faire.

Il existe également des lacunes dans l'apprentissage en profondeur. L'un d'eux est qu'il nécessite beaucoup d'échantillons pour construire un modèle de profondeur, ce qui limite l'application de cet algorithme. Aujourd'hui, de très bons résultats dans le domaine de la reconnaissance faciale.

### 4.3.3 La bibliothèque KERAS

KERAS est une bibliothèque populaire pour l'apprentissage en profondeur en Python, mais l'objectif de la bibliothèque est l'apprentissage en profondeur. En fait, il vise le minimalisme, en se concentrant uniquement sur ce dont vous avez besoin pour définir rapidement et simplement et construire des modèles d'apprentissage en profondeur.

### 4.3.4 La bibliothèque SVM

Les machines à vecteurs de support (SVM) sont formulées pour résoudre un modèle classique à deux classes problème de reconnaissance. Nous adaptons SVM pour faire face à la reconnaissance en modifiant l'interprétation de la sortie d'un classificateur SVM et de concevoir une représentation des images faciales qui est concordant avec un problème à deux classes. Le SVM traditionnel renvoie une valeur binaire. Pour former notre algorithme SVM, nous formulons le problème dans un espace de différence, qui capture explicitement les différences entre deux images faciales. Ceci est un départ à partir d'un espace facial traditionnel ou d'approches basées sur la vue, qui encode chaque image faciale à une vue séparée d'un visage.

### 4.3.5 La bibliothèque Tkinter

Tkinter est un module de base intégré dans Python. Python offre plusieurs options pour développer GUI (Graphical User Interface). De toutes les méthodes GUI, tkinter est la méthode la plus utilisée. Il s'agit d'une interface Python standard vers la boîte à outils Tk GUI livrée avec Python. Python avec tkinter est le moyen le plus rapide et le plus simple de créer les applications GUI. La création d'une interface graphique à l'aide de tkinter est une tâche facile.

## 4.4 Détection de visage

### Détection de visage en utilisant le «Classifieurs de Haar»

Dans les parties antérieures, on a spécifié que la première partie de la reconnaissance faciale est la détection de visage.

La bibliothèque OpenCV offre la méthode de Haar, qui est mieux connue comme la méthode de Viola Jones comme outil de classification et détection de visage dans une photo ou une vidéo.

Étant donné un fichier ou une vidéo en direct, il examine recouvre chaque partie de l'image et la classifie comme visage ou non. Le classifieur fait recours à un fichier XML pour décider comment classifier toute partie de l'image reçue.

OpenCV est offert avec divers classifieurs pour la détection de visage sous toutes ses faces, la détection des yeux, détection des corps, etc. Du coup, on a utilisé CNN comme outil pour mieux faire la détection, en effaçant les fausse positive, mais ceci a un coût puisque parfois il enlève une vraie détection.

Afin de traiter et résoudre ce problème, nous avons implémenté une étape de pré-traitement de l'image préalablement à celle de détection.

### Pré-traitement :

Si nous tentons de faire l'identification d'un visage dans une image non éditée, nous aurons une efficacité  $<20\%$ .

Il est primordiale d'appliquer des techniques différentes de pré-traitement d'image pour les standardiser.

Il devient alors très intéressant d'appliquer des filtres et des outils de pré-détection pour donner un ton plus standardisé à la photo, ce qui fait que les classifieur spécialisé puisse fonctionner à leur meilleur.

L'extrême majorité des algorithmes de reconnaissance de visage sont fortement sensibles aux conditions d'éclairage de manière que s'il a été formé pour reconnaître une personne dans une pièce sombre, il ne saura probablement pas la reconnaître dans une salle lumineuse.

C'est extrêmement important puisque la quasi-totalité des algorithmes de reconnaissance faciale souffre, en termes d'efficacité de détection, même par les plus faibles changements

d'éclairage.

Le pré-traitement des images s'effectue en deux étapes :

- **Étirement d'histogramme** : C'est le fait de délimiter les bordures des personnes dans une photo, et ceci se fait en augmentant l'intensité de contraste entre les pixels.
- **Filtre Médian** : Ceci est une étape intelligente qui permet d'éliminer les bruit (ou « noise » en anglais) des photos en appliquant un algorithme qui calcule le pixel en fonction du carré qui l'entoure.

La combinaison de ces deux filtres augmente le taux de succès de détection vers un pourcentage >95%.

Ceci implique que le pré-traitement est aussi bel et bien important pour avoir une bonne détection et non pas l'algorithme de détection lui-même.

## 4.5 Reconnaissance de visage

A ce stade du processus de reconnaissance, on a une photo ou frame qui a été nettoyée de tout bruit et qui met l'axe sur le visage (avec un encadrement pour la photo à afficher dans l'écran).

### 4.5.1 Base de données

La base de données est définie comme suit :

Le répertoire racine contient :

Database : elle contient les photos de la base de données.

Output : il comprend les visages détectés dans chaque photo.

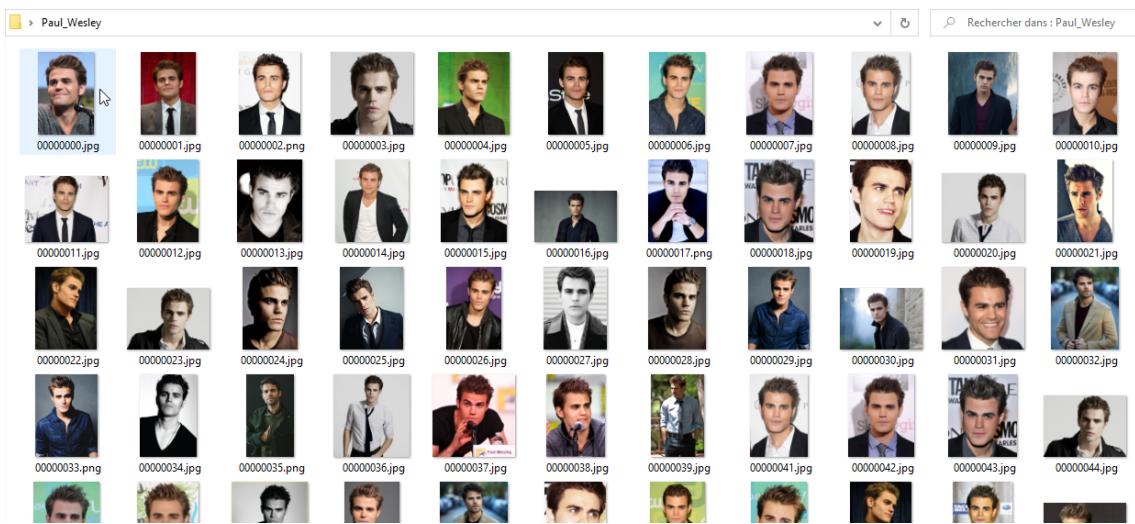


FIGURE 4.1 – Dossier « Photos :Paul Wilsey»

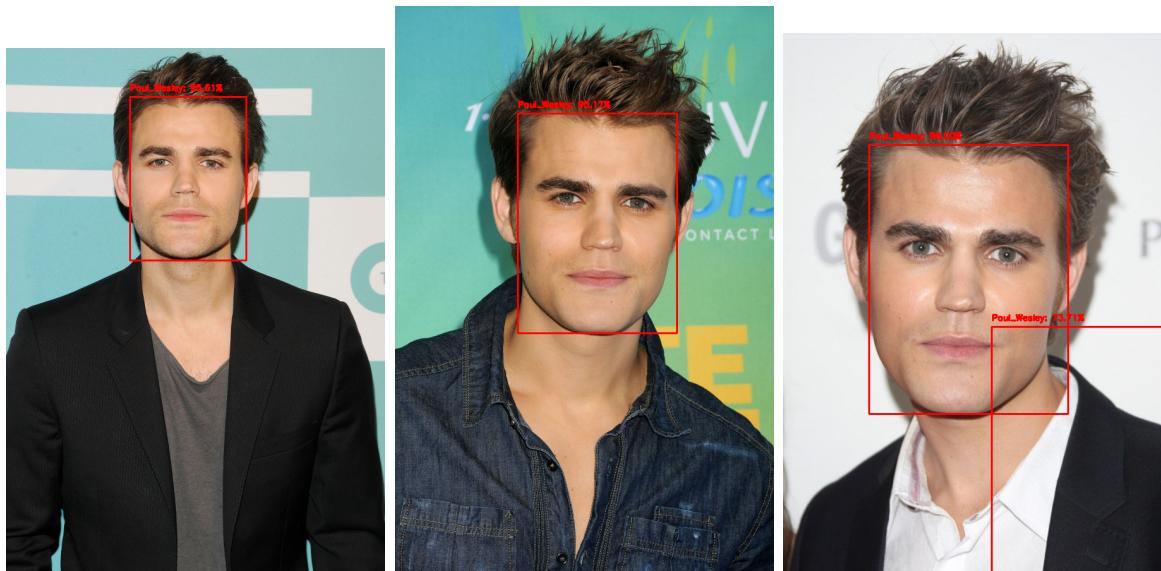


FIGURE 4.2 – Résultat détection « Photos :Paul Wilsey»

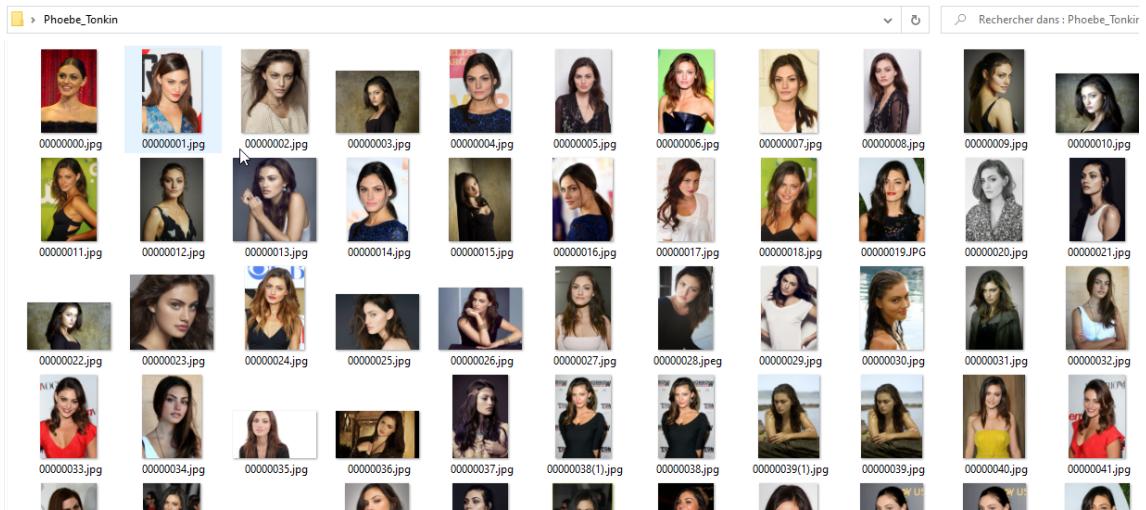


FIGURE 4.3 – Dossier « Photos :Phoebe Tonkin»



FIGURE 4.4 – Résultat détection « Photos :Phoebe Tonkin»

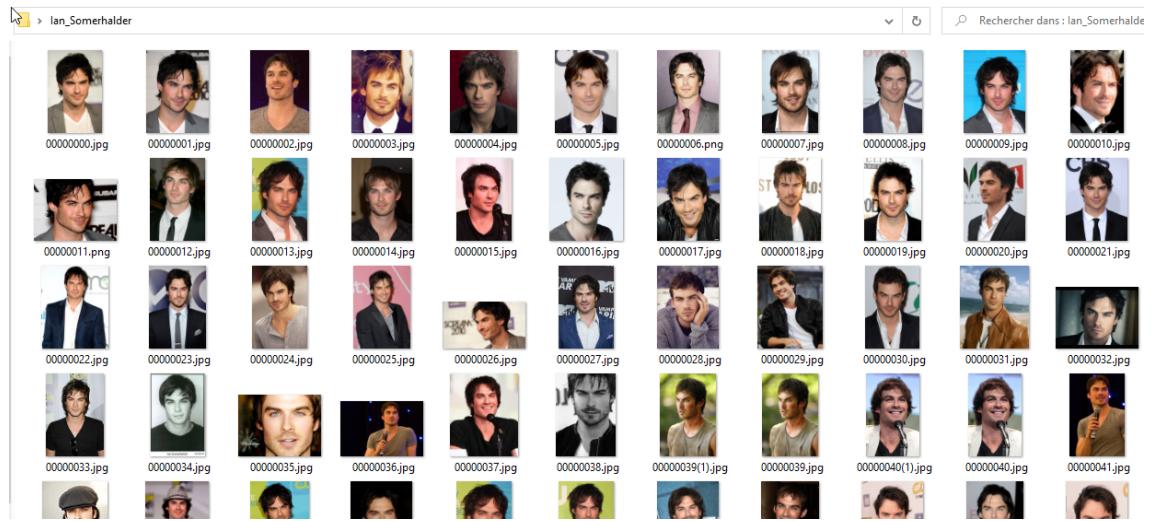


FIGURE 4.5 – Dossier « Photos :Lan Somerhalde»

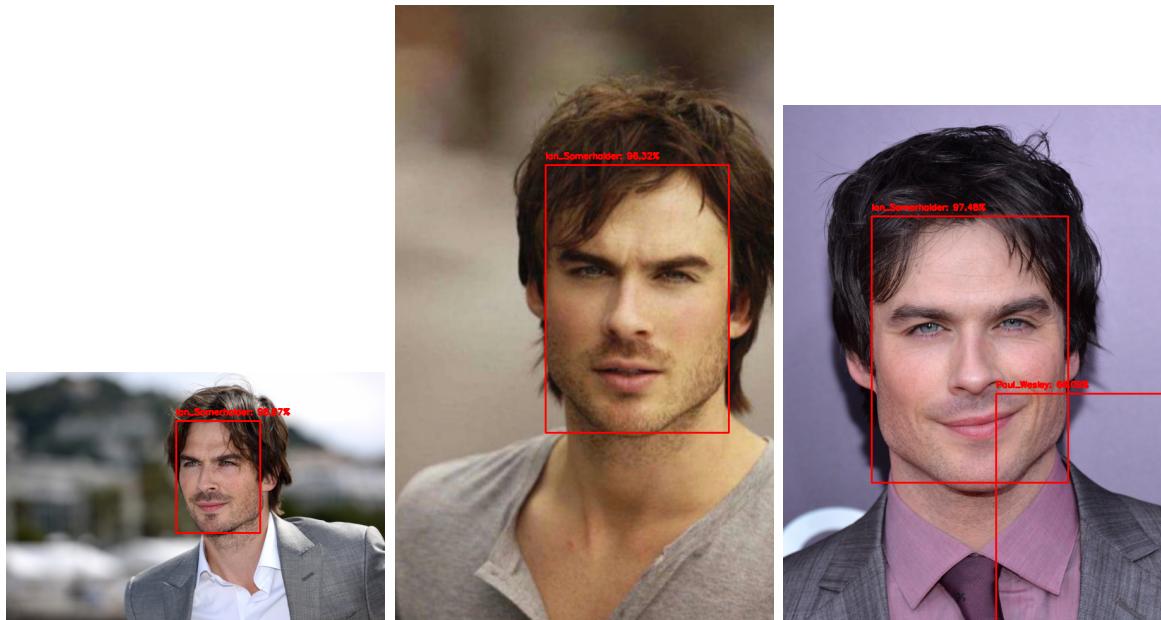


FIGURE 4.6 – Résultat détection « Photos :Lan Somerhalde»

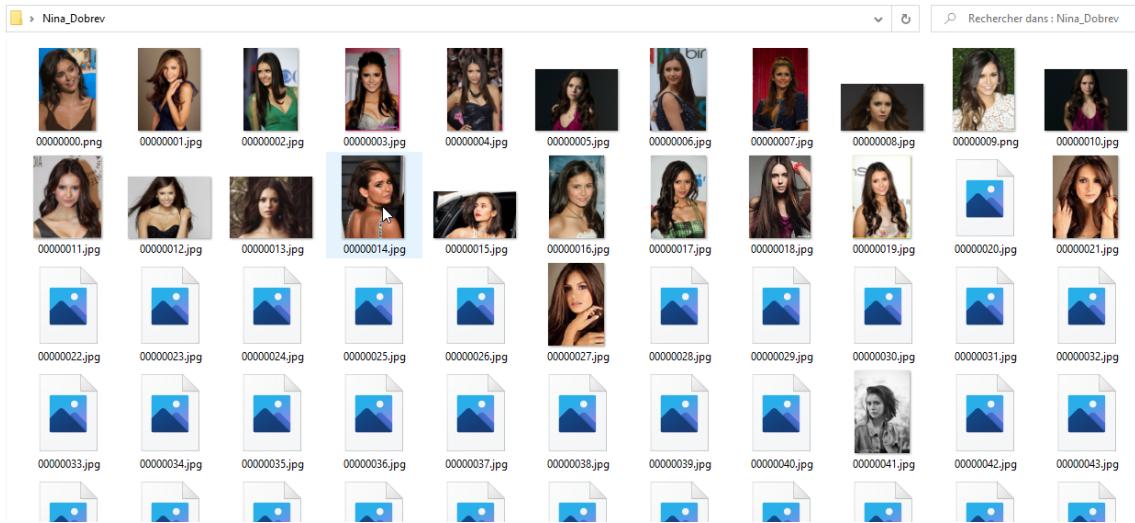


FIGURE 4.7 – Dossier « Photos :Nina Dorbev»

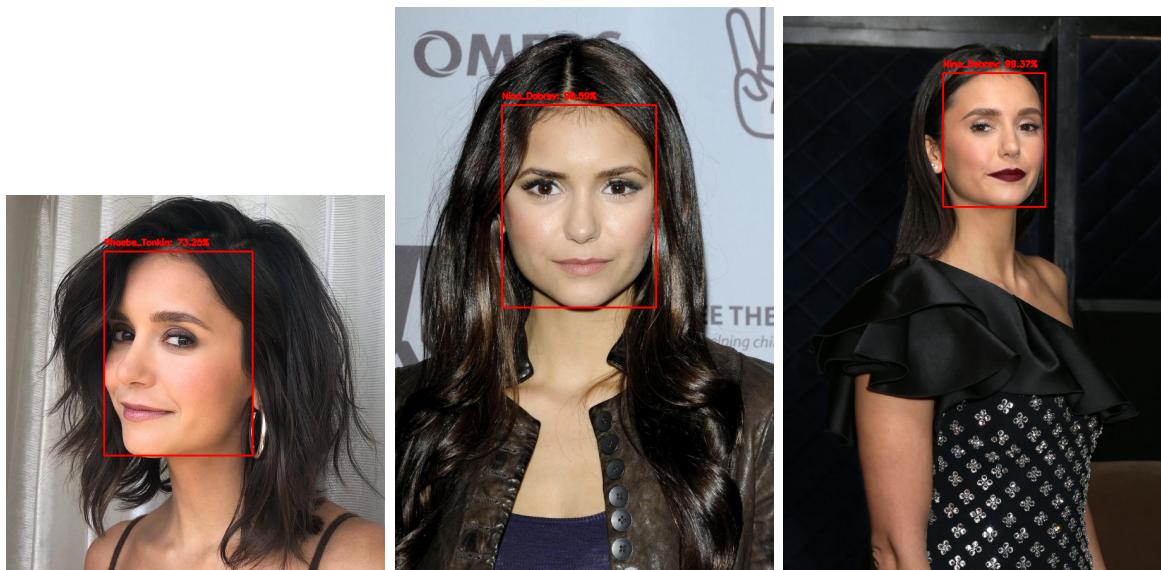


FIGURE 4.8 – Résultat détection « Photos :Nina Dobrev»

Information générales de la set d'images :

- **Nombre total des individus :** 4
- **Nombre des images par individu :** 100 pour la reconnaissance et 30 à 60 pour les tests
- **Total des images :** 600
- **Genre :** des images contenant mâles et femelles
- **Origine :** Des célébrités de la série « The Vampire Diaries »
- **Format des images :** 24 bits en couleur
- **Type des images :** JPG
- **Éclairage :** artificiel et naturel

Les tests élaborés en dessus ont permis une bonne configuration de la fonction « cvGoodFeaturesToTrack » de la bibliothèque OpenCV dans le but d'assurer une bonne qualité de reconnaissance.

Après plusieurs essais, il s'est avéré que le meilleur nombre maximal de points d'intérêt à trouver est égal à 100, et que la qualité minimale des points d'intérêts est égale à 0.09, quant à la distance minimale entre chaque deux points d'intérêt, sa valeur optimale est égale à 4, finalement, la taille de voisinage doit être de 3.

lorsqu'il s'agit de temps d'exécution, la partie apprentissage pris de l'ordre de 2 à 3 secondes qui est considéré trop !

La reconnaissance, par contre, est très rapide voire même quasiment immédiate.

## 4.6 Interface utilisateurs

A ce stade-là, nous allons procéder à la complétude notre travail par une interface GUI, pour ceci, nous avons continué avec python en utilisant la bibliothèque Tkinter. Celui-ci fournit des outils performants aussi bien que simple et facile à utiliser. Il est aussi bien efficace puisqu'on s'est familiariser avec Python.

### 4.6.1 Fenêtre principale

Ci-dessous est une représentation de l'interface graphique initiale lors de lancement de l'application

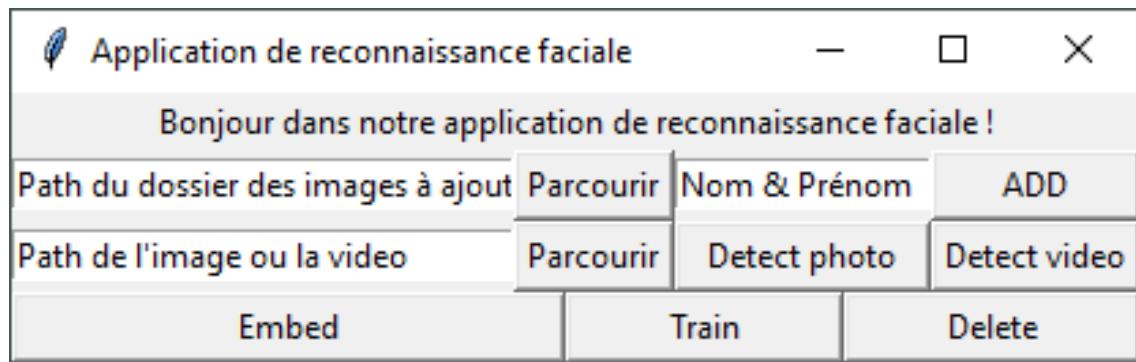


FIGURE 4.9 – Menu principale

#### 4.6.2 Apprentissage des photos

Pour apprendre une nouvelle personne, on doit chercher le chemin où se présentent les photos de cette personne, puis on donne son nom, et comme le présente la figure ci-dessous l’application copie le contenu de ce dossier dans sa base de données.

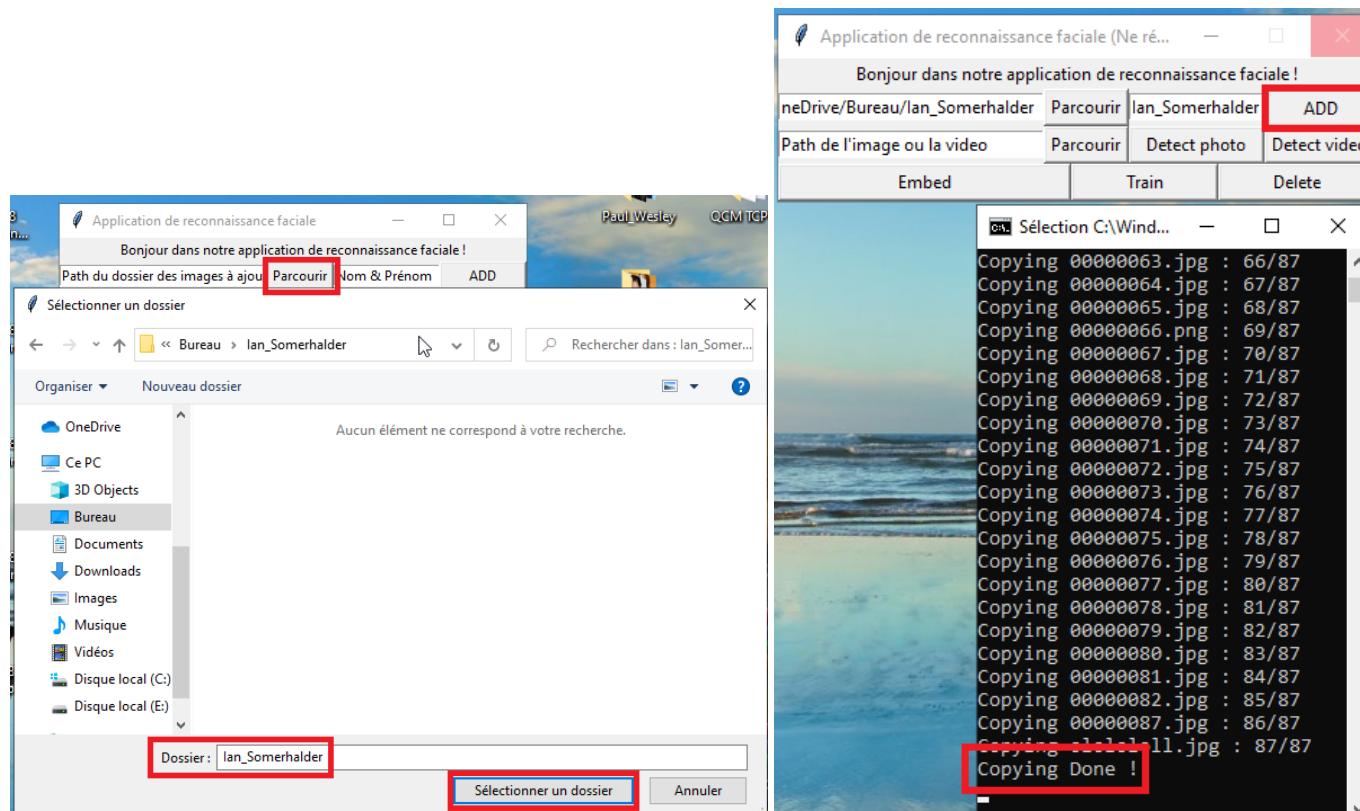
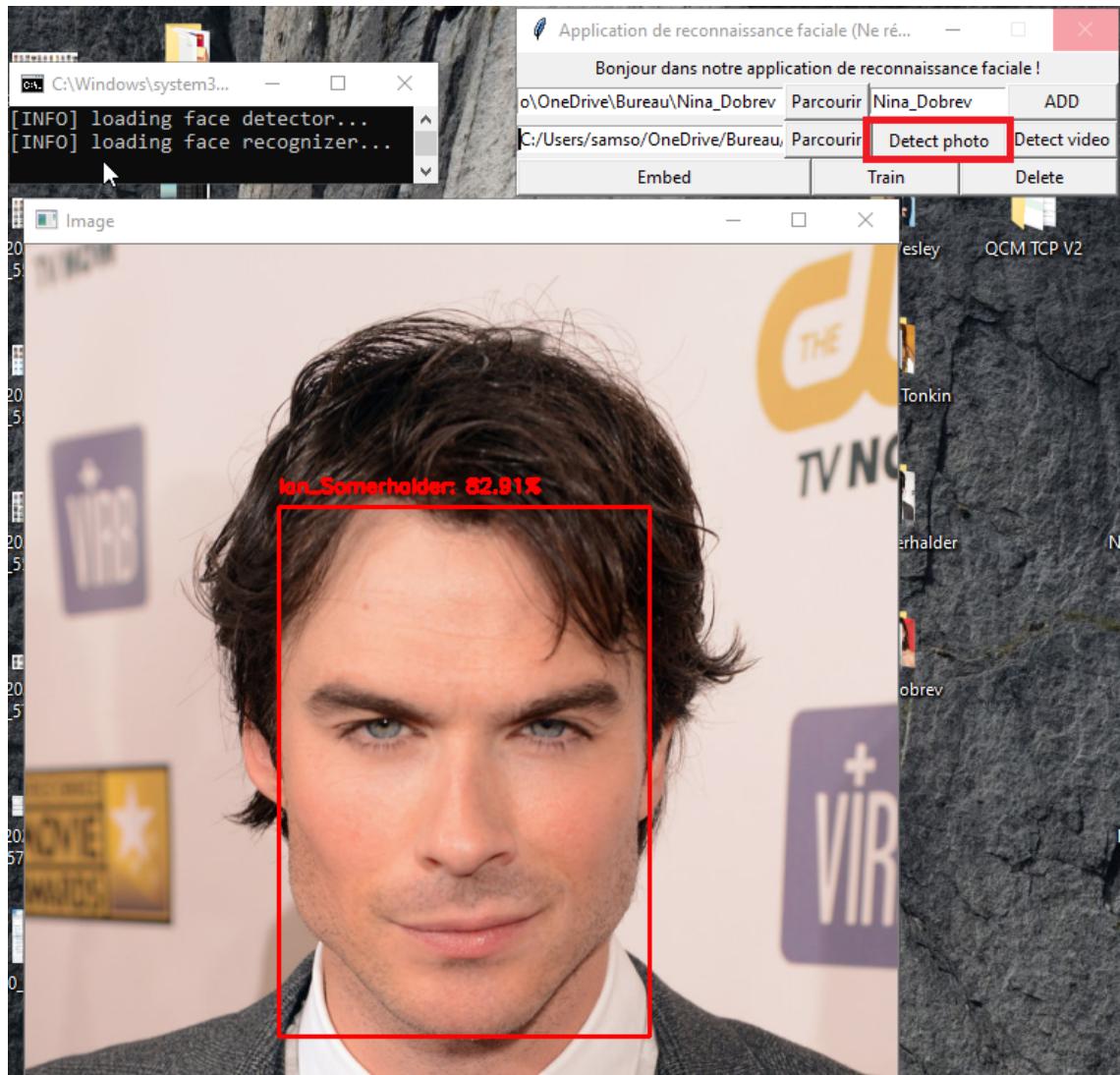


FIGURE 4.10 – Apprentissage des photos

### 4.6.3 Fenêtre résultat

Ci-dessous le résultat d'exécution de l'outil de détection, une fois pour la photo et une fois pour la vidéo.



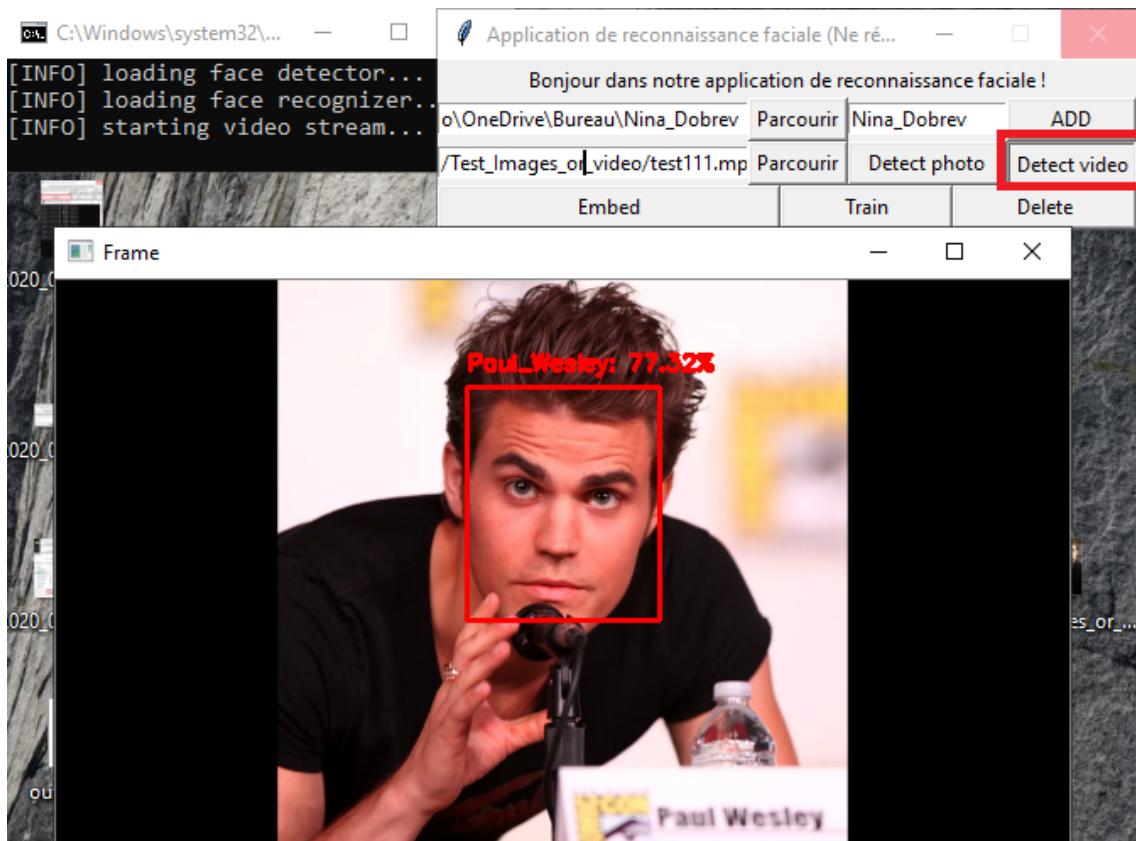


FIGURE 4.11 – Fenêtre résultat

## 4.7 Résultat

Les tests réalisés ont été élaborés afin d’observer et d’interpréter l’effet des conditions d’illuminations sur la marge d’erreur de prédiction, et l’influence de la distance des visages objets de détection ainsi que leur position vis à vis le système de capture.

En effet, les variabilité de luminosité est minimisée voire remédiée par le pré-traitement des images, quand à la position des visages et l'espace qui leur sépare du système, ces deux facteurs influencent en premier lieu la qualité du visage extrait et mettent l'accent sur l'impossibilité de la reconnaissance de profil.

Notre système peut néanmoins reconnaître des visages détectés avec un angle assez léger mais ce avec une marge d’erreur beaucoup plus élevée.

## 4.8 Conclusion

Dans une première étape, nous avons suivi le processus de détection de visage et ce en se basant sur les outils offerts par la bibliothèque OpenCV, notre objectif était d'éliminer les fausses alarmes pour éventuellement améliorer le taux de détection.

Ensuite, nous avons passé à la phase d'identification des visages dans laquelle nous nous sommes basés sur l'extraction des détails qui caractérisent chaque visage d'un autre. exemple : proportion des coins des yeux, la couleur de la peau, la taille du nez, etc.

# Conclusion et perspectives

Ce projet avait comme but de concevoir et réaliser une application qui pourrait, en un temps faible, de faire la reconnaissance d'une personne. De plus, l'utilité d'une telle application est en hausse vu son importance notamment pour les réseaux sociaux, de plus de son utilité dans les caméras de surveillances. La reconnaissance est, toujours, un obstacle continu qui, avec tout avancement, pose de nouvelles difficultés malgré l'expansion de la recherche dans ce domaine. De ce fait, les difficultés à surpasser sont toujours évolutionnaires et ne sont dans aucun contexte faciles à combattre. Il serait à noter que la performance d'un système affecte largement le pouvoir de reconnaissance dans le sens de rapidité. La tâche de reconnaissance de visage n'est pas directe, mais plutôt précédé par une étape de détection. Nous avons, pour cela, mis en forme une approche utilisant la méthode de CNN comme outil de détection et pré-traitement. L'utilisation de cet outil nous a permis de minimiser les fausses détections causées par la librairie OpenCV, mais il a, toutefois, éliminé parfois de vrais visages. Ceci fait, passons à l'étape d'identification. Nous nous sommes inspirés tellement et principalement par la méthode de « Viola Jones », donc nous avons procédé par des approches heuristiques pour mieux délimiter la position du visage, puis on a ajouté les vecteurs de détection de la géométrie de la tête avec SVM pour faire la classification. Il est indéniable qu'il y aura toujours de nouvelles approches et recherche pour améliorer le domaine de reconnaissance faciale. A travers notre projet, nous avons essayé de réaliser une application qui utilise les avancements dans l'AI en reconnaissance faciale dans le cadre réel.

# Netographie

- [1]<https://opencvlibrary.sourceforge.net/> - consulté le 02/05/2020.
- [2]<https://www.biometrie-online.net/> - consulté le 05/05/2020.
- [3]<https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>, consulté le 20/05/2020.
- [4]<https://www.nouvelobs.com/>, consulté le 22/05/2020.
- [5]<https://www.medecinesciences.org/>, consulté le 22/04/2020.
- [6]<https://www.futura-sciences.com/tech/actualites/tech-clearview-application-capable-tout-savoir-votre-vie-privee-partir-image-79246/>, consulté le 24/05/2020.
- [7]<https://www.pyimagesearch.com/2018/04/16/keras-and-convolutional-neural-networks-cnns/>, consulté le 27/05/2020.
- [8]<https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition>, consulté le 01/06/2020.
- [9]<https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning>, consulté le 12/06/2020.