

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique Université
Abderrahmane Mira de Béjaia
Faculté Des Science Exacts
Département Informatique



Thème : MÉMOIRE DE FIN D'ÉTUDE MASTER 2
INFORMATIQUE
THÈME :

Conception et Réalisation d'une Application Intelligente pour Retrouver des Personnes Disparues

Présenté par : *M^{elle} HAMLAOUI Houa* **Spécialité :** Intelligence Artificielle
M^{elle} MOUZAOUI Cylia Génie Logiciel

Soutenu le 07/07/2022 devant le jury composé de :

| | | | |
|----------------------|--------------------|-----|----------------------|
| Présidente | BOUCHELAGHEM Siham | MCB | Université de Béjaia |
| Examineur | MOKTEFI Mohand | MCB | Université de Béjaia |
| Encadrant | AKILAL Abdellah | MAA | Université de Béjaia |
| Co-Encadrante | ZEBBOUDJ Sofia | MCB | Université de Béjaia |

Année universitaire 2021/2022

Table des matières

| | |
|--|-------------|
| Table des matières | i |
| Table des figures | iv |
| Liste des tableaux | vi |
| Remerciement | viii |
| Dédicace | ix |
| Liste des abréviations | x |
| Introduction Générale | 1 |
| 1 État de l'art sur la reconnaissance faciale | 2 |
| 1.1 Introduction | 2 |
| 1.2 Détection du visage | 2 |
| 1.3 Méthodes de détection du visage | 2 |
| 1.3.1 Algorithme de Viola Jones | 2 |
| 1.3.2 Classifieur SVM (Support Vector Machine) | 4 |
| 1.3.3 Histogramme de gradients orientés : HOG | 8 |
| 1.4 Reconnaissance faciale | 10 |
| 1.5 Historique sur la reconnaissance faciale | 10 |
| 1.6 Processus de reconnaissance faciale | 11 |
| 1.6.1 Monde physique | 11 |
| 1.6.2 Acquisition | 11 |
| 1.6.3 Détection du visage | 12 |
| 1.6.4 Le pré-traitement | 12 |
| 1.6.5 Extraction des paramètres | 12 |
| 1.6.6 Classification | 12 |
| 1.6.7 Apprentissage | 12 |
| 1.6.8 Décision | 12 |

| | | |
|----------|---|-----------|
| 1.7 | Principales difficultés liées à la reconnaissance faciale | 13 |
| 1.8 | Méthodes pour la reconnaissance faciale | 13 |
| 1.8.1 | Méthodes globales | 13 |
| 1.8.1.1 | L'Analyse en Composantes Principales (ACP) | 14 |
| 1.8.1.2 | Réseau neuronal convolutif (CNN – Convolutional Neural Network) | 14 |
| 1.8.1.3 | Modèles de Markov Cachés (HMM – Hidden Markov Model) : | 14 |
| 1.8.2 | Méthodes locales | 14 |
| 1.8.3 | Méthodes hybrides | 15 |
| 1.9 | Comparaison des méthodes de reconnaissance et détection de visages | 15 |
| 1.10 | Conclusion | 16 |
| 2 | Problématique et systèmes de notifications de personnes perdues | 17 |
| 2.1 | Introduction | 17 |
| 2.2 | Systèmes de notification de personnes perdues | 17 |
| 2.2.1 | Système AMBER Alert | 17 |
| 2.2.2 | À la recherche d'un ange (Looking For An Angel). | 18 |
| 2.2.3 | Nous aidons les disparus (We Help the Missing) | 18 |
| 2.2.4 | Le Croissant-Rouge algérien (CRA) | 18 |
| 2.3 | Rôle des réseaux sociaux dans la recherche de personnes perdues | 18 |
| 2.3.1 | En Algérie | 19 |
| 2.4 | Problématique | 19 |
| 2.5 | Conclusion | 20 |
| 3 | Analyse et Conception | 21 |
| 3.1 | Introduction | 21 |
| 3.2 | Cahier des charges | 21 |
| 3.2.1 | Définition des besoins fonctionnels | 21 |
| 3.2.2 | Définition des besoins non fonctionnels | 21 |
| 3.3 | Analyse des besoins | 22 |
| 3.3.1 | Identification des acteurs | 22 |
| 3.3.2 | Diagramme de contexte | 22 |
| 3.3.3 | Identification des cas d'utilisation | 23 |
| 3.3.3.1 | Diagramme globale de cas d'utilisation | 23 |
| 3.3.3.2 | Description des cas d'utilisation | 24 |
| 3.3.4 | Modélisation des diagrammes de séquences | 25 |
| 3.3.4.1 | Diagramme de séquence du cas d'utilisation « S'authentifier » | 26 |
| 3.3.4.2 | Diagramme de séquence du cas d'utilisation « Poster un avis de recherche pour une personne perdue » | 27 |

| | | |
|----------|--|-----------|
| 3.4 | Diagramme d'interaction | 27 |
| 3.4.1 | Diagramme d'interaction « S'authentifier » | 28 |
| 3.5 | Diagramme de navigation | 28 |
| 3.6 | Diagramme de classe | 29 |
| 3.6.1 | Diagramme de classe de domaine | 29 |
| 3.6.2 | Classe, attributs et responsabilités | 30 |
| 3.7 | Schéma relationnel | 31 |
| 3.7.1 | Règles de passage au modèle relationnel | 31 |
| 3.7.2 | Le modèle relationnel de notre application Web | 31 |
| 3.8 | Schéma récapitulatif du système de reconnaissance faciale abordé | 31 |
| 3.9 | Conclusion | 32 |
| 4 | Réalisation de l'application | 33 |
| 4.1 | Introduction | 33 |
| 4.2 | Les technologies utilisées | 33 |
| 4.2.1 | L'API Twitter | 33 |
| 4.2.1.1 | Utilisation de l'API twitter | 34 |
| 4.2.2 | La reconnaissance faciale | 35 |
| 4.2.2.1 | Exemple de code pour la reconnaissance faciale | 35 |
| 4.2.3 | Langages utilisés | 37 |
| 4.2.3.1 | HTML | 37 |
| 4.2.3.2 | JavaScript | 37 |
| 4.2.3.3 | CSS | 38 |
| 4.2.3.4 | MySQL | 38 |
| 4.2.3.5 | Python | 38 |
| 4.2.4 | Environnement de développement utilisés | 38 |
| 4.2.4.1 | WampServer | 38 |
| 4.2.4.2 | PhpMyAdmin | 38 |
| 4.2.4.3 | Visual Paradigme for UML | 39 |
| 4.2.4.4 | Adobe Photoshop | 39 |
| 4.2.4.5 | Visual studio code | 39 |
| 4.2.4.6 | Spyder | 39 |
| 4.2.5 | Frameworks utilisés | 39 |
| 4.2.5.1 | Bootstrap | 40 |
| 4.2.5.2 | Flask | 40 |
| 4.3 | Présentation des interfaces de notre application Web | 40 |
| 4.3.1 | Interface « Page d'Accueil » | 40 |
| 4.3.2 | Interface « À propos » | 41 |

| | | |
|--------|---|----|
| 4.3.3 | Interface « Avis de recherche » | 42 |
| 4.3.4 | Interface « Inscription » | 42 |
| 4.3.5 | Interface « Authentification » | 43 |
| 4.3.6 | Interface « Compte utilisateur » | 43 |
| 4.3.7 | Interface « Poster un avis de recherche » | 44 |
| 4.3.8 | Interface « Éditer mes avis » | 44 |
| 4.3.9 | Interface « Résultats » | 45 |
| 4.3.10 | Interface « Dashboard » | 45 |
| 4.4 | Conclusion | 46 |

Conclusion Générale et perspectives 47

Table des figures

| | | |
|------|---|----|
| 1.1 | Les 4 types de caractéristiques utilisées par Viola et Jones [1]. | 3 |
| 1.2 | Les points/vecteurs extrêmes qui aident à créer l'hyperplan dans SVM [2]. | 4 |
| 1.3 | Classification des paires de coordonnées (x1,x2) en vert ou en bleu [2]. | 5 |
| 1.4 | Séparation de paires de coordonnées (x1,x2) en vert ou en bleu [2]. | 5 |
| 1.5 | Hyperplan optimal avec SVM[2]. | 5 |
| 1.6 | SVM non linéaire [2]. | 5 |
| 1.7 | SVM non linéaire après l'ajout d'une 3e dimension [2]. | 7 |
| 1.8 | Division des ensembles de données en classes avec SVM après l'ajout de la 3e dimension [2]. | 7 |
| 1.9 | Classification avec un SVM non linéaire dans un plan à 2 dimensions. [2] | 7 |
| 1.10 | Division de l'image en blocs avec HOG [3]. | 8 |
| 1.11 | Construction de l'histogramme d'un bloc avec HOG [3] | 9 |
| 1.12 | Calcul de l'histogramme global avec HOG [3]. | 9 |
| 1.13 | Détection vs. reconnaissance du visage [4]. | 10 |
| 1.14 | Diagramme de processus de reconnaissance faciale [5]. | 11 |
| 1.15 | Classification des algorithmes principaux utilisés en reconnaissance faciale [5] | 13 |
| 2.1 | Plan d'action lors de la disparition/kidnapping d'une personne. | 19 |
| 3.1 | Diagramme de contexte de notre application Web. | 23 |
| 3.2 | Diagramme globale de cas d'utilisation. | 24 |
| 3.3 | Diagramme de séquence du cas d'utilisation «S'authentifier» | 26 |
| 3.4 | Diagramme de séquence du cas d'utilisation «Poster un avis pour une personne perdue». | 27 |
| 3.5 | Diagramme d'interaction du cas d'utilisation « S'authentifier » | 28 |
| 3.6 | Diagramme de navigation de notre application web. | 29 |
| 3.7 | Diagramme de classe de domaine | 29 |
| 3.8 | Schéma récapitulatif de notre système de reconnaissance faciale | 32 |
| 4.1 | Compte développeur. | 34 |
| 4.2 | Les clés API du projet. | 34 |
| 4.3 | Le dossier qui contient les images des personnes perdus | 35 |

| | | |
|------|--|----|
| 4.4 | Exemple de code pour faire la reconnaissance faciale. | 36 |
| 4.5 | Résultat d'exécution de code sur le terminale de notre éditeur spyder | 37 |
| 4.6 | L'image trouver sur twitter | 37 |
| 4.7 | Interface de la « Page d'Accueil » de notre application Web. | 40 |
| 4.8 | Interface « A propos » de notre application Web. | 41 |
| 4.9 | Interface « Avis de recherche » de notre application Web. | 42 |
| 4.10 | Interface « Inscription » de notre application Web. | 42 |
| 4.11 | Interface « Authentification » de notre application Web. | 43 |
| 4.12 | Interface « Compte utilisateur » de notre application Web. | 43 |
| 4.13 | Interface « Poster pour un avis de recherche » de notre application Web. | 44 |
| 4.14 | Interface « Éditer mes avis » de notre application Web. | 44 |
| 4.15 | Interface « Notification » de notre application Web. | 45 |
| 4.16 | Interface « Dashboard » de notre application Web. | 46 |

Liste des tableaux

| | | |
|-----|--|----|
| 1.1 | Avantages et Inconvénients des méthodes de classification des visages. | 16 |
| 3.1 | Description des acteurs de notre application Web. | 22 |
| 3.2 | Cas d'utilisation associés aux acteurs de notre application Web. | 23 |
| 3.3 | Description textuelle du cas d'utilisation « S'authentifier » | 24 |
| 3.4 | Description textuelle du cas d'utilisation « Poster pour une personne perdue » . . . | 25 |
| 3.5 | Description des attributs de classes de notre application Web | 30 |

Remerciement

Nous tenons à remercier en premier temps le dieu tout puissant qui nous a donné le courage la volonté pour accomplir ce modeste travail.

Un remerciement particulier à nos promoteurs Monsieur AKILAL Abdellah,
Madame ZEBBOUDJ Sofia.

Pour son encadrement, sa disponibilité, sa positivité et sa motivation, et surtout pour la confiance qu'il nous a toujours accordée.

Nous tenons à remercier les membres du jury qui ont consacré leur temps et leur attention afin d'évaluer notre travail.

On remercie aussi toute personne ayant contribué de près ou de loin, directement ou indirectement à la réalisation de ce travail.

Dédicace

C'est avec un grand plaisir qu'on dédie ce travail, fruit de nos études en exprimant notre profonde gratitude à tous nos proches particulièrement :

A nos pères H. lahlal et M. Nordine pour leurs patiences, leurs encouragements et Leurs sacrifices.

A nos mères O. Kahina et I. Linda Pour l'éducation qu'elles nos à inculquée et tous les efforts qu'elles Ont donnée pour notre réussite

A nos très chères sœurs Lydia, Dydia, Djedjiga, Nawel.

A nos très chères amies de cœur F.Fatima, S.Zahra, B.Yamina

A nos très chère frère Saïd et Massi

A nos mariés Mahmoud et Nassim.

..... Cylia et Houa.....

Liste des abréviations

| | |
|---------------|---|
| AAM | Active Appearance Model |
| ACP | Analyse en Composantes Principales |
| ANN | Artificial Neural Networks |
| API | Application Programming Interface |
| BIC | Bayesian Information Criterion |
| CNN | Réseau neuronal convolutif |
| CRA | Croissant Rouge Algérien |
| CRM | Customer Relationship Management |
| CSS | Cascading Style Sheets |
| CSV | Comma Separated Values |
| EBGM | Elastic Buch Graph Matching |
| GNU | GNU's Not Unix |
| HMM | Hidden Markov Model |
| HOG | Histogram of Oriented Gradients |
| HTML | HyperText Markup Language |
| HTTP | HyperText Transfert Protocol |
| IA | Intelligence Artificielle |
| ICMEC | International Centre for Missing and Exploited Children |
| IDE | Integrated Development Environment |
| IHM | Interactions homme-machine |
| JSON | JavaScript Object Notation |
| LFA | Local Feature Analysis |
| LG-PCA | Log Gabor PCA |
| LBP | Local Binary Pattern |
| MIT | Massachusetts Institute of Technology |
| MySQL | My Structured Query Language |
| PDF | Portable Document Format |
| PHP | HyperText Preprocessor |
| RGB | Red Green Blue |
| SGBD | Système de Gestion de Base de Données |
| SSL | Secure Socket Layer |
| SVM | Support Vector Machine |
| UML | Unified Modeling Language |
| URL | Uniform Resource Locator |
| W3C | World Wide Web Consortium |
| WHATWG | Web HyperText Application Technology Working Group |
| XML | eXtensible Markup Language |

Introduction Générale

Identifier une personne à partir de son visage est une tâche aisée pour les humains, est-il de même pour une machine? Ceci définit la problématique de la reconnaissance automatique de visages, qui a engendré un grand nombre de travaux de recherche au cours des dernières années. La reconnaissance du visage est un domaine très actif dans la vision par ordinateur. Elle a été fortement étudiée et son but final est de produire des applications pour la sécurité, des applications en robotique, des interfaces homme-machine, des applications pour les appareils photo numériques et des jeux, etc.

La reconnaissance du visage implique généralement la détection de visage qui consiste à chercher et à détecter un ou plusieurs visages en lisant une image numérique ou une vidéo. Le visage détecté peut être comparé à un autre visage se trouvant dans une base de données de visages reconnus afin d'identifier une personne.

La disparition des personnes s'augmente de plus en plus dans le monde, ce qui nous a poussé à chercher un moyen développé pour les trouver.

En Algérie on utilise beaucoup les réseaux sociaux comme Facebook pour diffuser une alerte de disparition d'une personne mais à l'étranger, ils utilisent des systèmes plus développés dans la recherche des disparus.

Notre travail s'intéresse au problème général de la reconnaissance de visages dans le contexte de la recherche des personnes disparues.

Dans ce mémoire, nous concevons et implémenterons une application Web permettant de télécharger des médias (photos et vidéos) depuis le réseau social Twitter grâce à une API Twitter. Après avoir collecté quelques médias, une reconnaissance de visage des personnes disparues sur ces images et vidéos sera effectuées.

Notre application permettra de retourner l'identité de la personne disparue et l'URL de l'image où l'on a trouvé la personne en plus du pourcentage de similarité entre le visage de la personne perdue et celui de la personne retrouvée.

Notre mémoire est organisé comme suit :

- Le chapitre 1 : est un état de l'art sur la reconnaissance faciale dans lequel nous présentons le domaine de la reconnaissance faciale et les méthodes qu'elle utilise.
- Le chapitre 2 : est une description de notre problématique et quelques exemples de systèmes de notifications des personnes perdues.
- Le chapitre 3 : présente la conception et la réalisation de notre application Web.
- Le chapitre 4 : présente l'implémentation de notre application Web.

Chapitre 1

État de l'art sur la reconnaissance faciale

1.1 Introduction

Dans ce chapitre nous introduirons le domaine de la reconnaissance faciale et la détection de visages. Ensuite, nous présenterons le mode, les techniques, le processus de la reconnaissance faciale et les détecteurs de visage les plus populaires.

1.2 Détection du visage

La détection du visage est un terme de technologie informatique utilisé lorsqu'un logiciel intelligent est capable de déterminer l'existence, l'emplacement et la taille d'un visage humain sur une photo particulière, tout en ignorant d'autres objets sur la photo comme les arbres, les bâtiments et les corps aux alentours [4].

1.3 Méthodes de détection du visage

Dans ce qui suit nous allons introduire certaines méthodes de la détection de visage.

1.3.1 Algorithme de Viola Jones

L'algorithme de Viola Jones est nommé d'après deux chercheurs du domaine de la vision par ordinateur : Paul Viola et Michael Jones. Ces deux derniers ont proposé la méthode en 2001 dans leur article «*Rapid Object Detection using a Boosted Cascade of Simple Features*». Sa particularité repose sur le fait que les fonctionnalités utilisées sont très simples mais très nombreuses.

La méthode de Viola et Jones est une méthode basée sur l'apparence. Elle consiste à numériser l'image entière en calculant plusieurs caractéristiques dans des régions rectangulaires qui se chevauchent. Les caractéristiques utilisées pour détecter les visages dans cet algorithme sont de type *Haar* [Haar Cascade est une approche basée sur l'apprentissage automatique où de nombreuses images positives et négatives sont utilisées pour former le classifieur. Images positives – Ces images contiennent les images que nous voulons que notre classifieur identifie.]. Le premier apport de cette méthode est l'introduction d'images intégrales, qui permettent de calculer rapidement ces caractéristiques. Le deuxième apport important est d'interpréter ces caractéristiques comme un classifieur en boostant la sélection de proposer ces caractéristiques. Enfin, la méthode propose une architecture

qui combine des classificateurs boostés en cascades, ce qui augmente significativement le temps de détection.

C'est une méthode d'apprentissage supervisé. Elle se décompose donc en deux étapes : entraîner un classifieur à partir d'un grand nombre d'exemples positifs et négatifs, et appliquer le classifieur dans la phase de détection.

Viola et Jones suggèrent d'utiliser de nombreuses fonctionnalités au lieu de traiter directement les valeurs de pixel, ce qui est plus efficace et plus rapide. Ce sont des caractéristiques quasi-Haar, qui sont les différences dans la somme des pixels calculés dans une zone rectangulaire. La figure 1.1 montre les quatre types de caractéristiques proposés par Viola et Jones, où la somme des pixels gris est soustraite de la somme des pixels blancs. Leur nom vient de la similitude avec les ondelettes de Haar comme caractéristiques proposées par Papageorgiou et al. [1], dont Viola et Jones se sont inspirés.

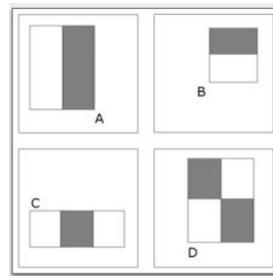


FIGURE 1.1 – Les 4 types de caractéristiques utilisées par Viola et Jones [1].

Pour calculer efficacement ces caractéristiques sur une image, les auteurs proposent également une méthode rapide, qu'il appellent « image intégrale ». C'est une représentation sous la forme d'une image, de même taille que l'image d'origine, qui en chacun de ses points contient la somme des pixels situés en dessus et à gauche de ce point.

Le deuxième élément clé de la méthode de Viola et Jones est l'utilisation de méthodes de boosting pour sélectionner les meilleures caractéristiques. Le boosting est un principe qui consiste à construire un classifieur « fort » à partir d'une combinaison pondérée de classifieur « faibles », c'est-à-dire, légèrement meilleurs que le hasard. Viola et Jones adaptent ce principe en identifiant les entités avec un classifieur faible, en construisant un classifieur faible qui n'utilise qu'une seule entité. Le classifieur est alors réduit à un couple (caractéristique, seuil). L'algorithme de boosting utilisé en pratique est une version modifiée d'AdaBoost pour effectuer la sélection et l'entraînement des classifieurs forts. Les classifieurs faibles utilisés sont généralement des arbres de décision. Un cas courant est lorsque l'arbre a une profondeur de 1, ce qui réduit l'opération de classification à un simple seuillage.

L'algorithme est de type itératif, avec un nombre d'itérations donné. À chaque itération, l'algorithme sélectionne une caractéristique qui est utilisée pour construire le classifieur fort final. Cette sélection est effectuée en apprenant des classifieurs faibles pour toutes les fonctionnalités et en choisissant le classifieur avec la plus petite erreur sur l'ensemble d'apprentissage. L'algorithme maintient également une distribution de probabilité sur l'ensemble d'apprentissage, qui est réévaluée à chaque itération en fonction des résultats de la classification. En particulier, il faudra accorder plus de poids aux exemples difficiles à classer. Le classifieur fort final construit par AdaBoost est une somme pondérée des classifieurs sélectionnés.

Bien qu'il s'agisse d'un framework obsolète, l'algorithme de Viola-Jones est assez puissant et son application s'est avérée exceptionnellement remarquable dans la détection de visages en temps réel[1].

1.3.2 Classifieur SVM (Support Vector Machine)

Une machine à vecteurs de support (SVM), est une technique d'apprentissage machine qui dépend d'algorithmes d'apprentissage supervisé. L'objectif de SVM est de classer correctement les données non visibles. Les classifieurs SVM sont utilisés pour les problèmes de reconnaissance d'image, particulièrement en reconnaissance de forme et en classification de couleur. SVM joue également un rôle essentiel dans de nombreux domaines de la reconnaissance manuscrite des symboles, tels que les services d'automatisation postale.

SVM sélectionne les points/vecteurs extrêmes qui aident à créer l'hyperplan. Ces cas extrêmes sont appelés vecteurs de support, d'où le nom de l'algorithme. Sur la figure 1.2, les limites de décision (ou les hyperplans) sont utilisés pour classer deux classes différentes.

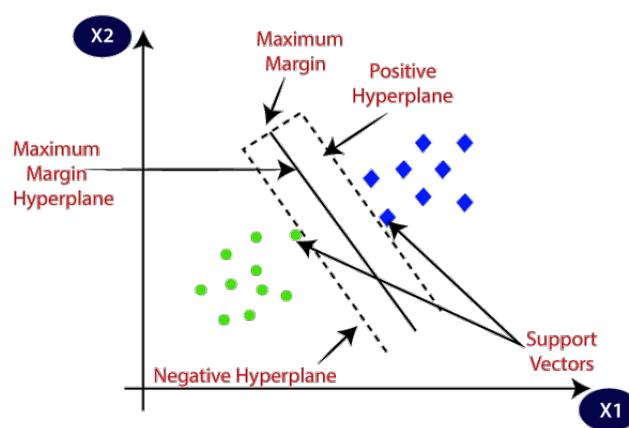


FIGURE 1.2 – Les points/vecteurs extrêmes qui aident à créer l'hyperplan dans SVM [2].

Il existe deux types de classifieurs SVM : des SVM linéaires et des SVM non linéaires.

SVM linéaire

Le fonctionnement de l'algorithme SVM peut être compris avec un exemple. Soit un jeu de données avec deux étiquettes (verte et bleue) et deux caractéristiques x_1 et x_2 . Le but est d'obtenir un classificateur capable de classer les paires de coordonnées (x_1, x_2) en vert ou en bleu tel qu'illustré sur la figure 1.3.

Puisqu'il s'agit d'un espace à deux dimensions, il est facile de séparer les deux classes en utilisant simplement une ligne droite. Néanmoins, il est possible d'avoir plusieurs lignes tel qu'illustré sur la figure 1.4.

L'algorithme SVM aide à trouver la meilleure ligne ou limite de décision; cette meilleure limite ou région est appelée hyperplan. L'algorithme SVM permet de trouver le point le plus proche des lignes des deux classes. Ces points sont appelés vecteurs supports. La distance entre les vecteurs et l'hyperplan est appelée marge. Le but de SVM est de maximiser cette marge. L'hyperplan avec une marge maximale est appelé l'hyperplan optimal. Ceci est illustré sur la figure 1.5.

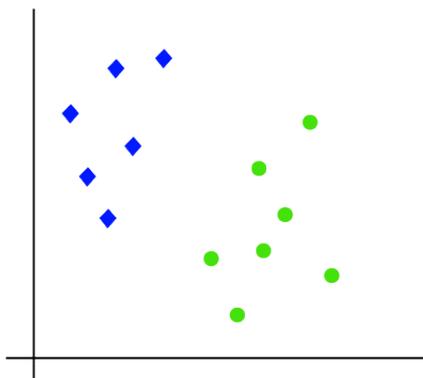


FIGURE 1.3 – Classification des paires de coordonnées (x_1, x_2) en vert ou en bleu [2].

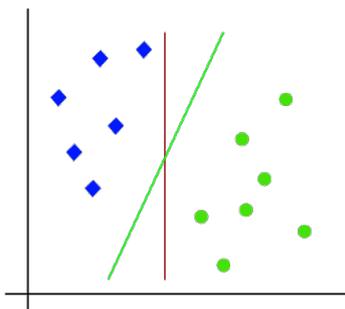


FIGURE 1.4 – Séparation de paires de coordonnées (x_1, x_2) en vert ou en bleu [2].

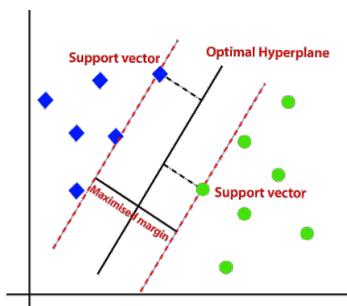


FIGURE 1.5 – Hyperplan optimal avec SVM[2].

SVM non linéaire

Quand les données sont disposées non linéairement, il devient impossible de les séparer en utilisant une ligne droite, tel que dans l'exemple illustré sur la figure 1.6.

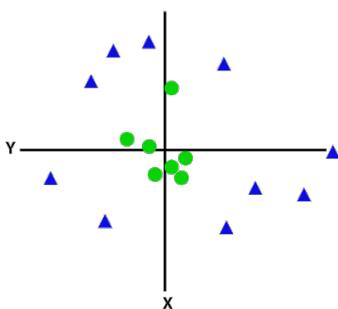


FIGURE 1.6 – SVM non linéaire [2].

Pour pouvoir séparer ces points de données, il faut ajouter une dimension supplémentaire : (x,y,z) . Cette troisième dimension z peut être obtenue comme suit : $z = x^2 + y^2$. En ajoutant la troisième dimension, l'espace échantillon deviendra comme illustré sur la figure 1.7.

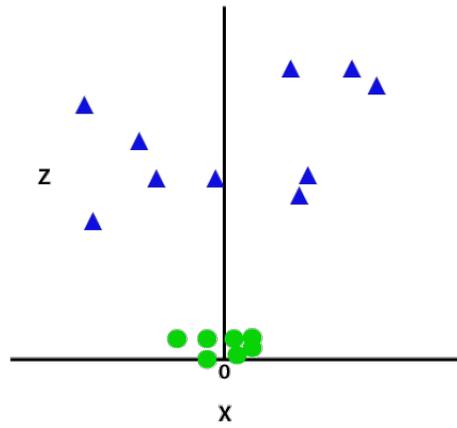


FIGURE 1.7 – SVM non linéaire après l'ajout d'une 3e dimension [2].

Il devient ainsi possible de diviser les ensembles de données en classes comme illustré sur la figure 1.8.

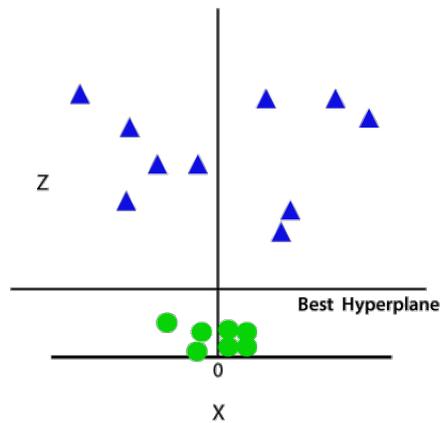


FIGURE 1.8 – Division des ensembles de données en classes avec SVM après l'ajout de la 3e dimension [2].

La figure 1.9 représente l'espace à 3 dimensions quand $z = 1$. la classe verte est délimitée par un cercle de rayon 1.

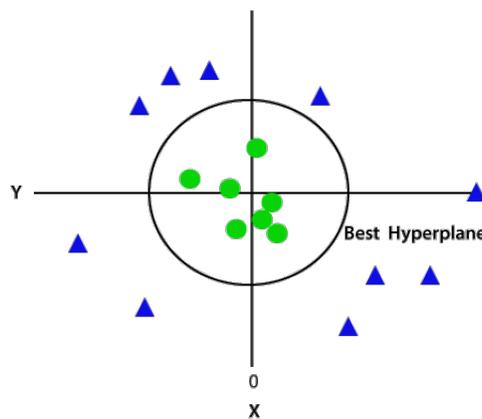


FIGURE 1.9 – Classification avec un SVM non linéaire dans un plan à 2 dimensions. [2]

1.3.3 Histogramme de gradients orientés : HOG

L'histogramme de gradients orientés (HOG – Histogram of Oriented Gradients) est une méthode d'extraction de caractéristiques utilisée en vision par ordinateur pour la détection d'objets. Cette méthode est particulièrement efficace pour la détection humaine. La technique consiste à calculer des histogrammes locaux de directions de gradient sur une grille dense. Ce calcul se fait en plusieurs étapes. [3].

Étape 1 : le pré-traitement

Une étape de pré-traitement peut être effectuée avant le calcul du gradient afin que les couleurs de l'image soient normalisées et une correction gamma correcte. Il y a plusieurs espaces de couleurs, le meilleur étant le RGB (Red Green Blue) [3].

Pour calculer un descripteur HOG, il faut d'abord calculer les gradients horizontaux et verticaux. Ceci est réalisé en filtrant l'image avec les noyaux comme Sobel, Prewitt, Roberts, etc. Par la suite, l'amplitude (g) et la direction d'orientation (O) sont calculées à partir du gradient comme suit [3] :

$$g = \sqrt{d_x^2 + d_y^2} \quad (1.1)$$

$$O = \tan^{-1}(d_x \times d_y) \quad (1.2)$$

Étape 2 : Construction de l'histogramme

La seconde étape consiste à créer des histogrammes de l'orientation des gradients. Ceci est fait sur des cellules carrées de petite taille (de 4x4 à 12x12 pixels) comme le montre la figure 1.10.

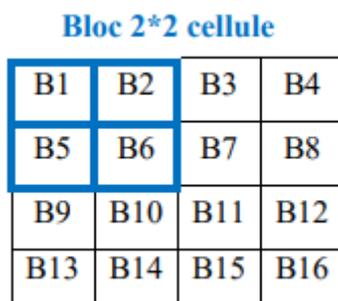


FIGURE 1.10 – Division de l'image en blocs avec HOG [3].

Chaque pixel de la cellule vote alors pour une classe de l'histogramme, en fonction de l'orientation du gradient par rapport à ce point. Le vote du pixel est pondéré par l'intensité du gradient en ce point. Les histogrammes sont uniformes de 0 à 180° ou de 0 à 360°. Mais de bien meilleurs résultats sont obtenus avec un histogramme à 9 classes. La figure 1.11 illustre la construction d'un histogramme à partir d'un bloc [3].

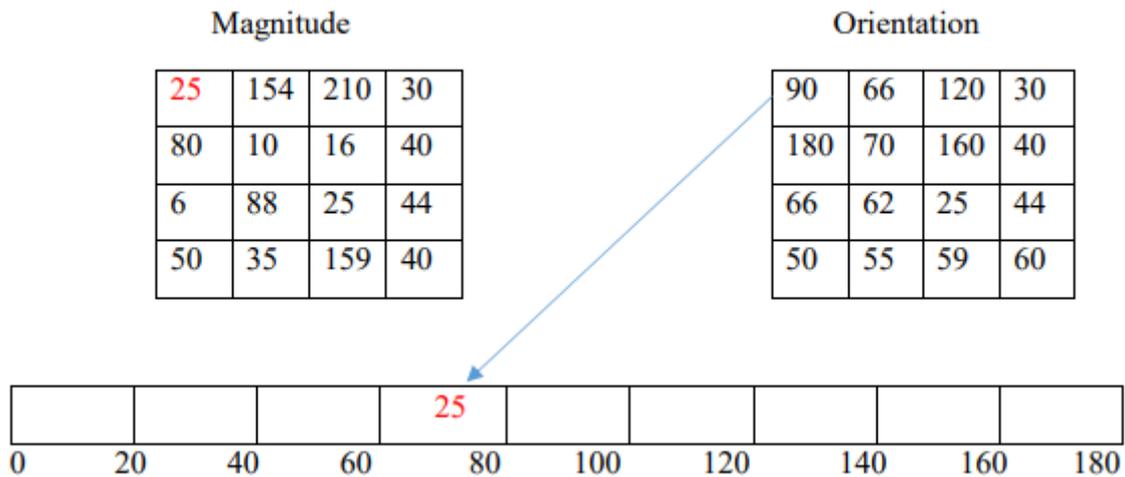


FIGURE 1.11 – Construction de l’histogramme d’un bloc avec HOG [3]

Étape 3 : Normalisation des blocs

Une étape importante est la normalisation des descripteurs afin d’éviter les disparités dues aux variations d’illumination. Cette étape introduit également de la redondance dans le descripteur. Pour cela, il est important de regrouper plusieurs cellules dans un bloc, qui est l’unité sur laquelle est effectuée la normalisation. Les blocs se recouvrent, donc une même cellule participe plusieurs fois au descripteur final, comme membre de blocs différents [3].

Étape 4 : Formation des blocs

Dans cette étape, plusieurs blocs sont regroupés dans un mini histogramme. Un histogrammes global est ensuite obtenu par concaténation de tous les mini histogrammes comme illustré sur la figure 1.12 [3].

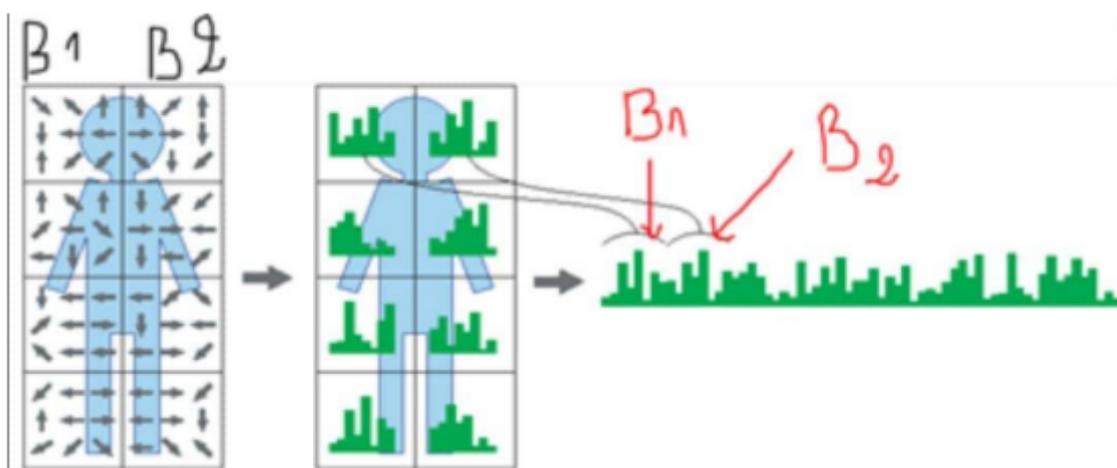


FIGURE 1.12 – Calcul de l’histogramme global avec HOG [3].

1.4 Reconnaissance faciale

La reconnaissance faciale est un domaine de la vision par ordinateur qui consiste à construire une intelligence artificielle qui vise à identifier une personne ou à vérifier son identité à partir de la structure faciale d'une personne détectée dans une image d'elle même.

La figure 1.13 illustre la différence entre la détection (gauche) et la reconnaissance (droite) du visage.

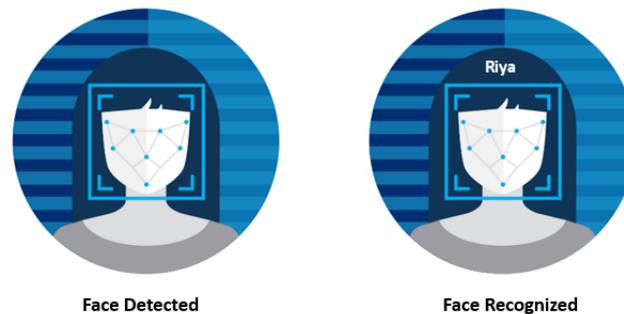


FIGURE 1.13 – Détection vs. reconnaissance du visage [4].

La reconnaissance faciale est utilisée dans trois principales tâches, selon le matériel à la disposition et l'objectif visé [6] :

1. **En mode de vérification** : comparaison de deux images entre elles, dite (1-v-1)
2. **En mode d'identification** : comparaison d'une image aux images d'une base de données d'images, dite (1-v-N).
3. **En mode de supervision** : suivi de l'image d'une personne dans une séquence vidéo, dite (1-v-0).

1.5 Historique sur la reconnaissance faciale

La technologie de la reconnaissance faciale est née dans les années 1960, suite à la conception d'une nouvelle Intelligence Artificielle (IA) qui va imiter toutes les actions humains.

En 1964 l'équipe de Woodrow Bledsoe [7], ont utilisé 20 marqueurs clés du visage pour localiser les yeux, les oreilles, le nez et la bouche sur une photo. Ils ont ensuite fourni ces valeurs à une IA de reconnaissance faciale semi-automatisée afin qu'elle procède à des calculs pour reconnaître deux photos d'une même personne. Une fois le système éprouvé et amélioré, Goldstein, Harmon et Lesk [8] ont rajouté 21 marqueurs tels que la couleur des cheveux et l'épaisseur de la lèvre pour automatiser la reconnaissance.

En 1988, Kirby et Sirovich [9] sont passé à l'amélioration des outils théoriques par l'application de l'Analyse en Composantes Principales (ACP), une technique mathématiques standard de l'algèbre linéaire. Cela a été considéré en quelque sorte comme une étape importante car elle a montré qu'au moins une centaine de valeurs ont été nécessaires pour coder convenablement et avec précision une image alignée et normalisée.

C'est ainsi qu'en 1991, Turk et Pentland [10] ont découvert la première vraie IA de reconnaissance faciale, en utilisant la technique Eigenfaces (visages propres), une découverte qui a permis de faire de la reconnaissance faciale automatique en temps réel. En effet, la méthode de reconnaissance faciale Eigenfaces emploie la technique de l'analyse en composantes principales, qui marque

une différence notable avec les méthodes plus classiques, appelées méthodes locales, qui se basent sur les particularités du visage analysé, et dont les défauts résident dans son manque de précision, ainsi que sa sensibilité aux informations qui ne sont pas pertinentes.

Dans [11], la méthode de Viola et Jones a été utilisée pour la détection de visages car 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.

Aujourd'hui la reconnaissance faciale est utilisée dans plusieurs domaines telle que La sécurité qui reste le domaine d'application principal et qui est nécessaire à l'identification et à l'authentification. Un bon exemple de cet usage est l'aéroport de Franc-fort où les passeports sont placés sur une borne qui lit la puce et récupère la photo de son détenteur qui est enregistrée dedans. Puis, une caméra prend une photo le passager et l'IA vérifie s'il s'agit de la bonne personne.

1.6 Processus de reconnaissance faciale

Lors de la reconnaissance faciale, une image suit un processus spécifique contenant plusieurs étapes qui peuvent être illustrées dans le diagramme de la figure 1.14.

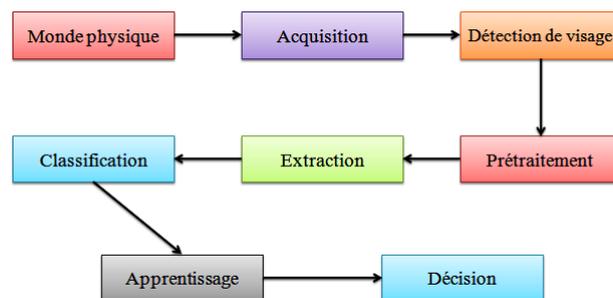


FIGURE 1.14 – Diagramme de processus de reconnaissance faciale [5].

1.6.1 Monde physique

C'est le monde réel (l'image). Dans cette étape, on tient compte des paramètres suivants : l'éclairage, la variation et l'échelle. La variation de l'un de ces trois paramètres peut conduire à une distance entre deux images du même individu supérieure à celle séparant deux images de deux individus différents, et par conséquent une fausse identification [5].

1.6.2 Acquisition

Le système d'acquisition est généralement équipé d'un capteur qui permet aux utilisateurs d'obtenir une fonction spécifique, une caméra pour capter une photo par exemple, de la même manière qu'un appareil photo permet d'avoir une image 2D d'un visage à partir d'une scène 3D [5].

1.6.3 Détection du visage

La détection du visage est l'étape essentielle de la reconnaissance faciale. Cette étape consiste en la localisation précise des régions caractéristiques du visage, telles que les yeux, le nez, les sourcils, la bouche, les lèvres, les oreilles, etc. [5].

1.6.4 Le pré-traitement

Les données fournies par le capteur primaire ne sont que sa représentation initiale et doivent donc être traitées au préalable. L'image originale peut être affectée par divers facteurs qui peuvent contenir des informations erronées dues à un équipement optique ou électronique. Cette étape supprime le bruit qui accompagne l'image et qui est causé par la qualité des appareils. Ceci est nécessaire car une image ne peut jamais être exempte de bruit ; l'arrière-plan et la lumière sont souvent inconnus [5].

1.6.5 Extraction des paramètres

L'extraction des paramètres est le coeur des systèmes de reconnaissance faciale. Lors de cette étape, les informations sur l'image sont extraites, ensuite stockées dans la mémoire pour une utilisation ultérieure dans l'étape de décision. Le choix de cette information utile réside dans la création d'un modèle du visage, qui doit être capable de le distinguer. Cette analyse est appelée propriétés d'indexation, de représentation, de modélisation ou d'extraction [5].

1.6.6 Classification

Elle consiste à modéliser les paramètres extraits d'un visage ou d'un ensemble de visages d'un individu en se basant sur leurs caractéristiques communes. Un modèle est un ensemble d'informations utiles, uniques et non récurrents qui caractérisent un ou plusieurs individus ayant des similarités. Ces derniers seront regroupés dans la même classe, et ces classes varient selon le type de décision [5].

1.6.7 Apprentissage

L'apprentissage consiste à retenir les modèles calculés pendant la phase d'analyse des personnes connues. Ce modèle est une représentation intégrée d'images pour faciliter l'identification, mais aussi la quantité de données stockées sous une forme ou une autre. Cette étape correspond aux références interactives réelles qui seront enregistrées dans la base de données [5].

1.6.8 Décision

La décision fait partie du système d'identification qui consiste à trouver le modèle correspondant au visage pris à partir de ceux stockés dans la base de données . À partir de cela, il est possible de décider si un visage de la base de donnée correspond à l'individu recherché [5].

1.7 Principales difficultés liées à la reconnaissance faciale

Pour le cerveau humain, le processus de la reconnaissance faciale est une tâche visuelle de haut niveau. Bien que les êtres humains puissent détecter et identifier des visages dans une scène facilement, construire un système automatique qui accomplit ces tâches représente un sérieux défi. Ce défi est d'autant plus grand lorsque les conditions d'acquisition des images sont très variables.

Il existe deux types de variations associées aux images de visages :

1. La première variation est attribuée au facteur de la ressemblance physique entre les individus.
2. La deuxième variation est plus vaste. Elle peut être attribuée à plusieurs facteurs, à savoir [12] :
 - Les changements d'illumination. En effet, les variations d'éclairage rendent la tâche de reconnaissance de visages très difficile.
 - Les variations de la pose qui est considérée comme un problème majeur pour les systèmes de reconnaissance faciale. Cependant, lorsque la rotation est supérieure à 30°, la normalisation géométrique n'est plus possible.
 - Le changement des expressions faciales. La déformation du visage qui est due aux expressions faciales est localisée principalement sur la partie inférieure du visage. Elle entraîne forcément une diminution du taux de reconnaissance faciale.
 - Dans le cas de vrais jumeaux, il est peu probable que la vérification automatique du visage puisse détecter les différences très subtiles qui existent entre eux.

1.8 Méthodes pour la reconnaissance faciale

Il existe trois principales méthodes pour la reconnaissance faciale : les méthodes globales, les méthodes locales et les méthodes hybrides, comme indiqué dans le diagramme de la figure 1.15.

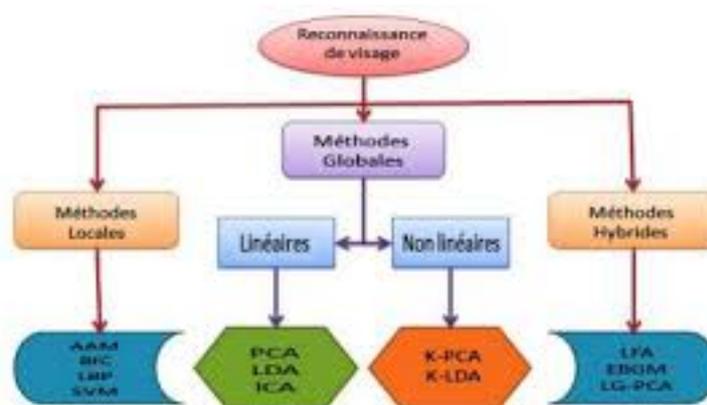


FIGURE 1.15 – Classification des algorithmes principaux utilisés en reconnaissance faciale [5]

1.8.1 Méthodes globales

Les méthodes globales sont basées sur des techniques d'analyse statistique connues. Il n'est pas nécessaire de repérer certains points caractéristiques du visage (comme les centres des yeux, le centre de la bouche, etc.) à part pour normaliser les images. Dans ces méthodes, les images de visages sont traitées de manière globale et sont généralement transformées en vecteurs [13]. Il existe plusieurs méthodes globales, les plus connues sont cités ci-après.

1.8.1.1 L'Analyse en Composantes Principales (ACP)

C'est l'approche la plus investie en reconnaissance de visages. Elle est connue également comme étant l'extension de Karhunen-Loève, vecteurs propres, ou bien l'Analyse en Composantes Principales ACP ou PCA (Principal Component Analysis) [13].

Les étapes suivantes résument le processus de reconnaissance par ACP [13] :

1. Initialisation : Acquérir les M images de visages constituant l'ensemble d'apprentissage et calculer les visages propres qui définissent l'espace des visages.
2. Pour l'image d'un nouveau visage, calculer un ensemble de poids basé sur l'image en entrée et les M visages propres par projection de cette image sur chaque visage propre.
3. Déterminer si l'image contient un visage en vérifiant si elle est suffisamment proche de l'espace des visages.
4. Si l'image contient un visage, on le classe selon les poids précédemment calculés comme étant une personne connue ou inconnue.

1.8.1.2 Réseau neuronal convolutif (CNN – Convolutional Neural Network)

La première technique utilisant les réseaux de neurones artificiels (ANN – Artificial Neural Networks) pour la reconnaissance des visages humains était un réseau à une seule couche adaptative nommé WISCAD de Stonham [?] traitant les images binaires des visages de 16 individus (vue de face et avec variation d'expressions). WISCAD était capable de reconnaître avec succès l'identité et l'expression du visage. Lors de l'apprentissage près de 400 images étaient utilisées pour les 16 individus [13].

Un autre exemple est le réseau de neurones basé sur une décision probabiliste qui est hérité de la structure modulaire de son prédécesseur. Ce réseau a prouvé son efficacité pour [13] :

1. La détection du visage dans l'image.
2. Localisation des yeux et génération du vecteur des caractéristiques
3. La reconnaissance faciale des visages.

1.8.1.3 Modèles de Markov Cachés (HMM – Hidden Markov Model) :

Les visages sont intuitivement divisés en régions (yeux, nez, etc.) qui peuvent être associées à des états du HMM ; vu que les HMMs requièrent une séquence d'observations unidimensionnelles, et que les images de visages sont bidimensionnelles, l'auteur a proposé de convertir ces dernières en une séquence temporelle ou spatiale unidimensionnelle. [13].

1.8.2 Méthodes locales

Les approches locales de reconnaissance faciale sont basées sur un traitement séparé des différentes régions de l'image du visage. Le modèle utilisé est basé sur la connaissance de la morphologie du visage. La plupart du temps, il s'agit de détecter/extraire des traits faciaux locaux.

Les avantages de ces méthodes sont qu'elles prennent en compte les particularités des visages comme des formes naturelles à reconnaître, en plus elles utilisent un nombre réduit de paramètres. Par rapport aux méthodes globales, elles sont plus robustes aux problèmes posés par les changements d'éclairage, de pose et d'expressions faciales [5]. Parmi ces méthodes, on peut citer [5] :

1. AAM (Active Appearance Model).
2. BIC (Bayesian Information Criterion).
3. LBP (Local Binary Pattern).
4. SVM (Support Vector Machines).

Toutes ces méthodes ont l'avantage de pouvoir modéliser plus facilement les variations de pose, d'illumination et d'expression par rapport aux méthodes globales. Cependant, ils sont plus lourds à utiliser car il est souvent nécessaire de placer manuellement un grand nombre de points sur le visage, alors que les méthodes globales ont seulement besoin de connaître la position des yeux pour normaliser les images, ce qui peut être fait automatiquement et de manière assez fiable à partir d'un algorithme de détection [5].

1.8.3 Méthodes hybrides

La méthode hybride combine la détection avec l'extraction des caractéristiques structurelles pour obtenir des informations supplémentaires utiles pouvant améliorer les performances de reconnaissance lors des changements de pose, d'éclairage et les expressions faciales [5]. Parmi ces méthodes on trouve :

1. LFA (Local Feature Analysis).
2. EBGM (Elastic Buch Graph Matching).
3. LG-PCA (Log Gabor PCA).

1.9 Comparaison des méthodes de reconnaissance et détection de visages

Le tableau 1.1 résume les avantages et limites de méthodes de reconnaissance et détection des visages [13].

| APPROCHE | AVANTAGES | LIMITES |
|--|--|--|
| APPROCHE LOCALE | <ul style="list-style-type: none"> - Requiert moins d'espace mémoire. - Plus rapide et simple à implémenter | <ul style="list-style-type: none"> - Dépendent étroitement de l'exactitude d'extraction des caractéristiques - Prise en charge de la variation de la pose, de l'éclairage. |
| Les Réseaux de Neurones (CNN) | <ul style="list-style-type: none"> - Temps de classification - Résistent aux bruits et au manque de fiabilité des données - Prise en compte des variations de poses/éclairage | <ul style="list-style-type: none"> - Sur-apprentissage - Extension à des problèmes plus larges - Apprentissage long |
| Les HMMs | <ul style="list-style-type: none"> - Algorithme d'apprentissage très robuste | <ul style="list-style-type: none"> - Temps d'apprentissage et classification long. - Choix des états basé sur l'intuition subjective |
| Les Machines à Vecteurs de Support (SVM) | <ul style="list-style-type: none"> - Capacité d'extraction des informations discriminantes pertinentes du visage d'une manière entièrement automatique. | <ul style="list-style-type: none"> - Temps d'apprentissage long et algorithme d'apprentissage compliqué |
| Haar cascade | <ul style="list-style-type: none"> -Détection rapide des objets - Faible taux de faux positifs | <ul style="list-style-type: none"> -Faible avec la rotation -Processus d'apprentissage lent -moins robuste à l'occlusion |

TABLE 1.1 – Avantages et Inconvénients des méthodes de classification des visages.

1.10 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur la détection et la reconnaissance faciale et les techniques utilisées dans les systèmes de reconnaissance faciale pour l'identification de personnes. Cette étude nous a permis de constater que la reconnaissance de visage suscite de plus en plus l'intérêt de la communauté scientifique.

Dans le chapitre suivant nous verrons quelle est la démarche à suivre lors de la disparation d'une personne et quels systèmes il est possible d'utiliser.

Chapitre 2

Problématique et systèmes de notifications de personnes perdues

2.1 Introduction

Dans ce chapitre nous commencerons par présenter quelques systèmes de notifications de personnes perdus, les réseaux sociaux et leur rôle dans la recherche des personnes perdus. Nous présenterons ensuite notre problématique et les raisons qui nous ont amenée à nous lancer dans ce travail. Enfin nous introduirons la solution que nous proposons à cette problématique.

2.2 Systèmes de notification de personnes perdues

Une personne portée disparue est une personne dont il est impossible de déterminer si elle est vivante ou décédée; on ne sait pas où elle se trouve ou dans quel état elle est. Des personnes peuvent être portées disparues après une disparition volontaire, ou à la suite d'un accident, d'un crime, d'un décès dans un lieu où elles ne sont pas retrouvées (dans la mer par exemple), ou bien pour d'autres raisons. L'enlèvement criminel est l'un des cas de disparitions de personnes les plus fréquemment signalés.

Dans la plupart des régions du monde, les personnes disparues sont généralement retrouvées rapidement. De nombreuses organisations, ont établi des réseaux, partagé les meilleures pratiques, diffusé des informations et des images d'enfants portés disparus. Parmi ces organisations et sociétés, nous citons les suivantes.

2.2.1 Système AMBER Alert

Le système AMBER Alert est un système d'alerte à grande échelle nommé à la mémoire de Amber Hagerman. Lorsqu'un enfant de moins de 18 ans est déclaré kidnappé, une alerte AMBER est lancée via une technologie de diffusion de l'information. Les plans d'alerte AMBER sont des partenaires volontaires comprenant des organismes d'application de la loi, les départements des autoroutes et les entreprises de communication qui fournissent des alertes d'urgence [14].

La décision de lancer l'alerte est prise par le service de police autorisé à la déclencher dès que tous ces critères sont réunis. Parmi Les principaux critères d'alerte pour déclencher une alerte AMBER sont :

- L'enfant disparu doit être âgé de moins de 18 ans.

- Le policier doit avoir des motifs raisonnables de croire que l'enfant disparu a été victime d'enlèvement.
- Le policier doit avoir les motifs raisonnables de croire que la vie de cet enfant est sérieusement en danger.
- Le policier doit posséder suffisamment de renseignements qui permettront de localiser l'enfant.

Lorsque l'alerte AMBER est lancée, des messages sont diffusés à la radio, à la TV, sur les téléphones portables, sur des sites internet, par e-mail, etc. Les informations distribuées comprennent : la description, la photographie et le nom de l'enfant enlevé, une description, la photographie du ou des suspect(s) et, lorsque c'est possible, la description et la plaque d'immatriculation du véhicule suspecté.

2.2.2 À la recherche d'un ange (Looking For An Angel).

C'est une organisation à but non lucratif qui se consacre à réunir des personnes disparues avec leurs proches. Ils aident les familles à signaler la disparition de leurs proches et les aident tout au long du processus de recherche. Cette organisation s'efforce de servir de groupe de soutien solide, et organisent des activités et des événements spéciaux pour accroître la sensibilisation aux personnes disparues. Certains de leurs efforts comprennent la distribution de dépliants/littérature et des allocutions. En outre, ils assistent les opérations de recherche et de sauvetage et collaborent avec d'autres organisations pour lutter contre l'épidémie de disparitions [15].

2.2.3 Nous aidons les disparus (We Help the Missing)

C'est une société nationale à but non lucratif. Elle est composée d'une équipe d'enquêteurs privés et de citoyens qui se consacrent de manière désintéressée à sensibiliser, enquêter, localiser, défendre les personnes, identifier les personnes décédées non identifiées et sauver les victimes de la traite de leur ravisseur, et ce, peu importe l'âge de la personne perdue ou depuis combien de temps elle a disparu. Ils s'efforcent d'offrir un soutien aux familles qui traversent cette période difficile [16].

2.2.4 Le Croissant-Rouge algérien (CRA)

C'est une association d'aide humanitaire algérienne. Elle a pour objectif de venir en aide aux personnes en difficulté en Algérie et à l'étranger. Ses missions fondamentales sont l'urgence, le secourisme, l'action sociale, la formation, la santé et l'action internationale [17].

2.3 Rôle des réseaux sociaux dans la recherche de personnes perdues

Les réseaux sociaux sont des services permettant de regrouper diverses personnes dans le but d'échanger sur un sujet particulier. Les origines des réseaux sociaux remontent aux forums, groupes de discussion et salons de chat introduits dès les premières heures d'internet. Parmi les premiers réseaux sociaux, il y a Myspace et Facebook qui sont des services généralistes sur lesquels chacun peut partager le contenu de son choix, quelque soit le sujet, avec ses contacts, à condition de disposer d'un compte sur ces réseaux sociaux.

Les réseaux sociaux comme Facebook et Twitter permettent de retrouver de nombreuses connaissances perdues de vue. La solution la plus simple est de tenter de la retrouver grâce aux moteurs de recherche internet. En tapant simplement le nom de la personne, il est possible de la retrouver et de la contacter, comme il est possible d'accéder à un certains nombre d'informations qu'il la concerne et sont plus ou moins personnelles.

Les progrès technologiques des moyens de communication ont grandement facilité la tâche des autorités et des proches des personnes disparues. Par conséquence, les enquêteurs comptent beaucoup sur la diffusion de l'information, grâce au phénomène instantané fourni par des sites comme Facebook ou Twitter, où les gens peuvent partager des photos et des informations plus rapidement qu'avec les méthodes traditionnelles telles que le bouche-à-oreille.

Un exemple de cas de disparition où les réseaux sociaux ont démontré leur efficacité, est le cas d'une jeune fille qui a été déclarée disparue auprès de la fondation Cédrika Provencher. Son fondateur, Henry Provencher, a déclaré : « *Aux débuts de la Fondation, on a eu un avis d'une disparition d'une jeune fille. Avec les médias sociaux, en quelques heures, on a eu 1,6 million de partages sur les réseaux sociaux. Le lendemain, j'ai reçu un coup de téléphone d'un témoin disant avoir vu la jeune fille, je lui ai dit d'appeler la police. J'ai moi-même appelé la police et la jeune fille a été localisée saine et sauf quelques heures plus tard* ».

2.3.1 En Algérie

Lorsqu'une personne est portée disparue en Algérie, la première chose à faire est d'alerter le quartier dans lequel la personne habite. Dans les petits villages par exemple, la population part à la recherche de la personne disparue. Mais si la personne n'est pas retrouvée, sa famille doit se présenter au commissariat pour déposer une plainte pour perquisition. À partir de là, la famille et les proches commencent à publier des avis de recherche dans la rue mais surtout sur de nombreux réseaux sociaux comme Facebook, Instagram, YouTube, Twitter, etc., où l'on trouve de nombreuses pages spécialisées dans la recherche de personnes disparues. La figure 2.1 représente le plan d'action général dans le cas de la disparition d'une personne.

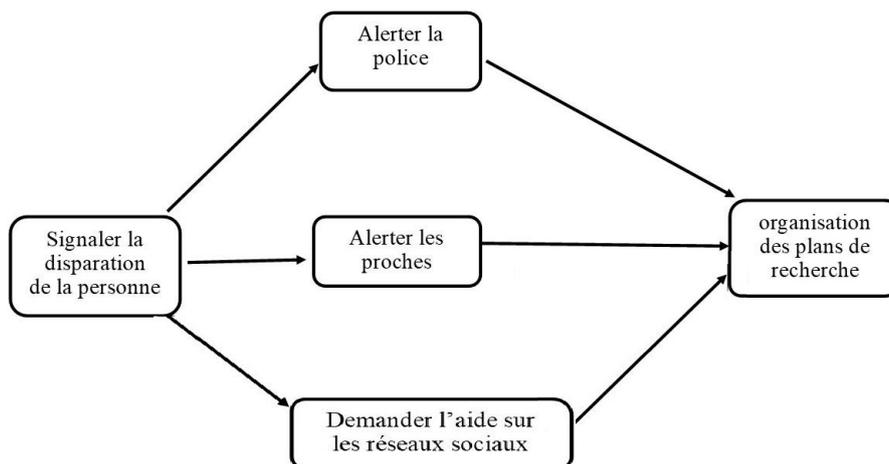


FIGURE 2.1 – Plan d'action lors de la disparition/kidnapping d'une personne.

2.4 Problématique

Chaque année, des centaines des milliers des personnes dans le monde disparaissent sans laisser de traces et dans des circonstances mystérieuses. Meurtre, évasion, enlèvement, suicide, toutes

ces hypothèses sont de bonnes tentatives pour expliquer ce qui a pu se passer. Le cas de certaines personnes disparues reste non résolu depuis des années. Dans de nombreuses juridictions, les parents et les tiers ne peuvent pas gérer les biens personnels à moins qu'un certificat de décès officiel ne soit délivré. Les lois relatives à ces décès sont souvent compliquées. Les situations, les incertitudes, les fermetures ou l'absence de funérailles en raison de la disparition d'une personne sont extrêmement pénibles et peuvent avoir un impact durable sur la famille et les amis.

Ce sont les autorités étatiques qui sont les premières chargées de prévenir les disparitions et de révéler le sort de la personne disparue. Mais à mesure que de plus en plus de personnes se perdent dans le monde, les autorités perdent le contrôle de tout cela.

Il est à noter que le développement de la technologie Internet a entraîné de nombreux changements dans la vie humaine non seulement en temps ordinaire mais aussi en temps anormal. Plus particulièrement, les réseaux sociaux sont de plus en plus utilisés dans tous les secteurs de la vie humaine.

L'Algérie utilise beaucoup les réseaux sociaux comme Facebook pour diffuser une alerte de disparition d'une personne mais à l'étranger, on utilise des systèmes plus développés dans la recherche des personnes perdues comme le système d'alerte AMBER. C'est pourquoi nous proposons dans notre mémoire de résoudre cette problématique par la création d'un portail Web dédiée à la recherche de personnes disparues. Pour arriver à cela, nous nous sommes posés les questions suivantes :

1. Comment utiliser les réseaux sociaux pour aider dans la recherche des personnes perdues ou kidnappées ?
2. Comment l'IA peut aider ?
3. Qui aura accès à notre système ?

En utilisant les réseaux sociaux et en se basant sur les nouvelles technologies de la reconnaissance faciale et les APIs, nous nous sommes lancées dans la conception et la réalisation d'une application Web permettant aux autorités de réduire la charge de travail à laquelle elles sont confrontées. Notre Application Web a les objectifs suivants :

1. Les personnes qui peuvent accéder à notre application Web sont la famille et les proches de la personne disparue/kidnappée. Leur rôle sur le application Web est la création et la soumission d'avis de recherche. Quand une personne est retrouvée, le système affiche les résultats de recherche à l'utilisateur qui a posté un avis de recherche.
2. Le rôle de l'intelligence artificielle dans notre système est de vérifier si les personnes disparues enregistrées dans la base de données de notre application existent sur des images de tweets publiés sur twitter, et ceci, en utilisant des APIs. Notre application doit trouver avec quel pourcentage la personne perdue et la personne trouvée sur les tweets sont similaires.

2.5 Conclusion

Dans ce chapitre avons présenté des systèmes de diffusion d'alerte de personnes perdues et le rôle des réseaux sociaux pour les trouver. Nous avons aussi présenté notre problématique et introduit une solution.

Dans le prochain chapitre nous définirons la partie de conception et analyse en présentant le cadre fonctionnel et le cadre dynamique de notre système.

Chapitre 3

Analyse et Conception

3.1 Introduction

Ce chapitre sera consacré à l'analyse des besoins et des exigences. Il se focalisera sur l'identification des acteurs et l'analyse des besoins. Puis, il traduira les besoins fonctionnelles et non fonctionnelles, qui seront également modélisés à l'aide d'un diagramme de cas d'utilisation et des diagrammes de séquence. Ensuite, nous entamerons la phase de conception, nous présenterons les diagrammes d'interaction, ainsi que le diagramme de navigation. Ensuite, nous concevrons un diagramme de classes suivi d'un schéma récapitulatif que nous avons utilisé.

3.2 Cahier des charges

Notre cahier des charges est sous forme d'un ensemble de besoins fonctionnels et un ensemble de besoins non fonctionnels présenté comme ci-dessous.

3.2.1 Définition des besoins fonctionnels

Le site à réaliser doit satisfaire l'ensemble des fonctionnalités définies par les utilisateurs. Cette plate-forme doit répondre aux besoins fonctionnels suivants :

1. Permettre la consultation des avis de recherches des personnes perdus (afficher la liste des avis).
2. Mettre à disposition de nombreuses fonctionnalités (accès aux comptes utilisateur, créer un avis de recherche, éditer les avis).
3. Permettre aux utilisateurs de consulter les résultats quand une personne cherchée est retrouvée.
4. Donner un accès au système pour faire la recherche et la reconnaissance faciale d'une manière automatique.

3.2.2 Définition des besoins non fonctionnels

Ce sont des besoins spécifiques au système. Ils doivent être pris en compte tout au long du processus de développement de l'application Web, à savoir :

1. La sécurité : tous les accès des utilisateurs doivent être protégés par un login et un mot de passe.
2. Validité : l'application Web doit réaliser exactement les tâches spécifiées dans le cahier des charges.
3. Performance : le temps de réponse doit être court.
4. L'ergonomie et la convivialité : il est primordial de faciliter l'interaction entre les différents utilisateurs du site et de l'interface.
5. Le code doit être clair pour permettre un développement ou une amélioration future.

3.3 Analyse des besoins

3.3.1 Identification des acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Autrement dit, un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données [18].

Dans le cadre de notre étude nous avons distingué trois acteurs principaux qui sont décrits dans le tableau 3.1.

| Acteur | Description |
|------------------------|---|
| Visiteur | Une visiteur est toute personne qui accède à notre application sans s'authentifier. Il peut consulter quelques interfaces de notre application et s'inscrire pour devenir membre. |
| Membre | C'est la personne inscrite sur notre site et qui a accès à certaines fonctionnalités (poster un avis de recherche). Il est en général un proche d'une personne disparue. |
| Agent de police | L'agent de police a accès à toutes les fonctionnalités de notre application Web qui permettent d'aider à la localisation d'une personne perdue. |
| Administrateur | C'est la personne ou un groupe de personnes qui fait la gestion de notre application Web et qui peut être un simple programmeur ou un policier. |

TABLE 3.1 – Description des acteurs de notre application Web.

3.3.2 Diagramme de contexte

Le diagramme de contexte est l'un des premiers diagrammes pouvant être réalisé afin de décrire à haut niveau le futur système en mode boîte noire. Il permet de se concentrer sur l'identification des différents acteurs interagissant avec lui [19].

La figure 3.1 montre le diagramme de contexte de notre application Web.

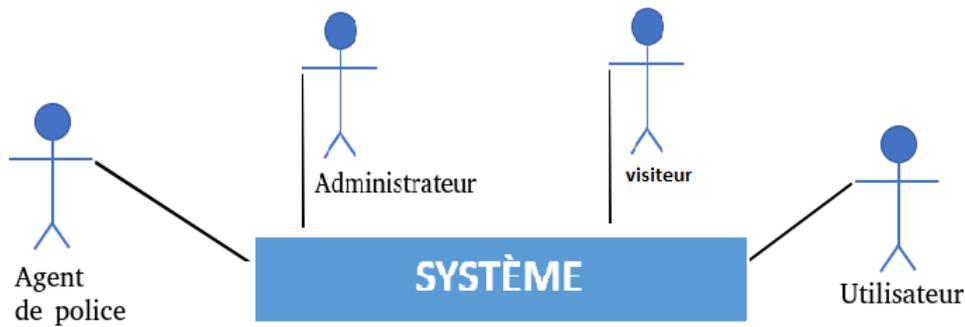


FIGURE 3.1 – Diagramme de contexte de notre application Web.

3.3.3 Identification des cas d'utilisation

Les cas d'utilisation sont des outils formels utilisés pour identifier et modéliser les besoins des utilisations. Une unité cohérente représente une fonction visible de l'extérieur [18].

Le tableau 3.2 illustre l'ensemble d'actions réalisées par les acteurs définis précédemment.

| Acteur | Cas d'utilisation |
|---------------------------------------|--|
| Visiteur | Créer un compte Consulter la liste des personnes perdues |
| Membre/Agent de police/Administrateur | S'authentifier |
| Membre/Agent de police | Poster un avis de recherche Consulter la liste des personnes perdues Consulter les résultats reçus Gérer son compte |
| Agent de police | Archiver un avis de recherche |
| Administrateur | Gestion de l'application |

TABLE 3.2 – Cas d'utilisation associés aux acteurs de notre application Web.

3.3.3.1 Diagramme globale de cas d'utilisation

Le diagramme de cas d'utilisation est une vision globale du comportement fonctionnel des systèmes logiciels. Il est présenté aux managers ou aux acteurs du projet [18].

Pour notre système, le diagramme de cas d'utilisation global est représenté par la figure 3.2.

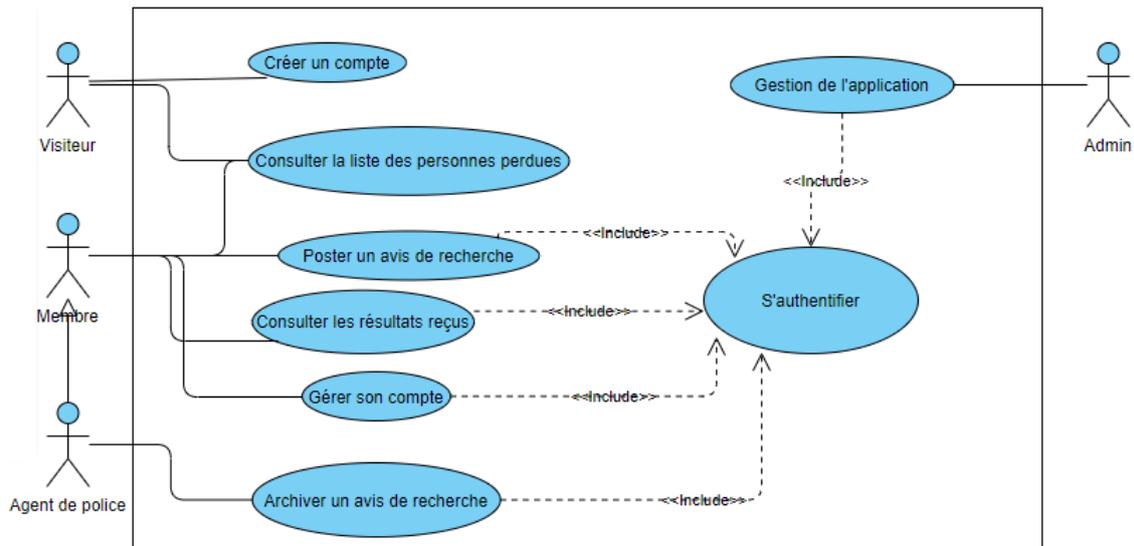


FIGURE 3.2 – Diagramme globale de cas d'utilisation.

3.3.3.2 Description des cas d'utilisation

Dans ce qui suit, nous décrivons textuellement (tableau 3.3 et tableau 3.4) les cas d'utilisation les plus importants de notre application Web. Il s'agit plus précisément de [18] :

1. Identifier le cas : en résumant son objectif et en donnant les acteurs impliqués.
2. Décrire un scénario nominal : par un ensemble de messages échangés entre les acteurs et le système. Il s'agit ici d'expliquer l'approche idéale où tout fonctionne parfaitement.
3. Décrire un scénario alternatif : par un ensemble d'actions qui s'exécutent si les conditions dans le scénario nominal ne sont pas validées .

| Cas d'utilisation n°1 | S'authentifier |
|-------------------------------|---|
| Description | Il s'agit de vérifier l'identité d'un utilisateur avec un login et un mot de passe. |
| Acteur | Membre, Agent de police, Administrateur |
| Pré-condition | Application disponible et utilisateur déjà inscrit |
| Scénario nominal | <ol style="list-style-type: none"> 1. L'acteur saisit son login et son mot de passe dans le formulaire d'authentification. 2. L'acteur valide les données 3. Le système vérifie la validité des champs saisis 4. Le système affiche l'interface adéquate à l'acteur authentifié |
| Scénario alternatif | <ol style="list-style-type: none"> 4a. Des champs n'ont pas été saisis ou les données sont non conformes. 4b. Le système demande à l'utilisateur de saisir les champs non valides. 4c. Le scénario reprend à l'étape 1. du scénario nominal. |
| Enchaînement d'erreurs | <ol style="list-style-type: none"> 4a. Le login saisi ne correspond à aucun utilisateur inscrit ou le mot de passe ne correspond pas au login. 4b. Le système affiche un message d'erreur. |
| Post-condition | L'acteur est authentifié et à accès à ses fonctionnalités |

TABLE 3.3 – Description textuelle du cas d'utilisation « S'authentifier »

| | |
|------------------------------|--|
| Cas d'utilisation n°2 | Poster un avis de recherche sur une personne perdue |
| Résumé | Poster un avis de recherche par un membre ou un un agent de police sur une personne perdue. |
| Acteur | Membre, Agent de police |
| Pré-condition | Application disponible et utilisateur déjà inscrit |
| Scénario nominal | <ol style="list-style-type: none"> 1. L'acteur saisit les informations. 2. L'acteur envoie les informations d'avis de recherche. 3. Le système vérifie la validité des informations. 4. Le système affiche un message de confirmation à l'utilisateur. |
| Scénario alternatif | <ol style="list-style-type: none"> 4a. Des champs n'ont pas été saisis ou les données sont non conformes. 4b. Le système demande à l'utilisateur de saisir les champs non valides. 4c. Le scénario reprend à l'étape 1. du scénario nominal. |

TABLE 3.4 – Description textuelle du cas d'utilisation « Poster pour une personne perdue »

3.3.4 Modélisation des diagrammes de séquences

Un diagramme de séquence est un diagramme d'interaction qui détaille comment une opération est effectuée. La formulation UML représente graphiquement l'interaction entre les acteurs et les systèmes dans le temps [18].

Dans ce qui suit, nous présenterons les diagrammes de séquences de fonctionnalités de notre application Web.

3.3.4.1 Diagramme de séquence du cas d'utilisation « S'authentifier »

Pour s'authentifier, l'utilisateur, l'agent de police ou l'administrateur devra saisir son login et son mot de passe. Le système devra d'abord vérifier la conformité des données saisies (dans un loop), ensuite si l'utilisateur existe et enfin si le mot de passe correspond à cet utilisateur (dans un alt). L'utilisateur est authentifié si toutes les vérifications sont réussies. Ceci est représenté dans le diagramme de la figure 3.3.

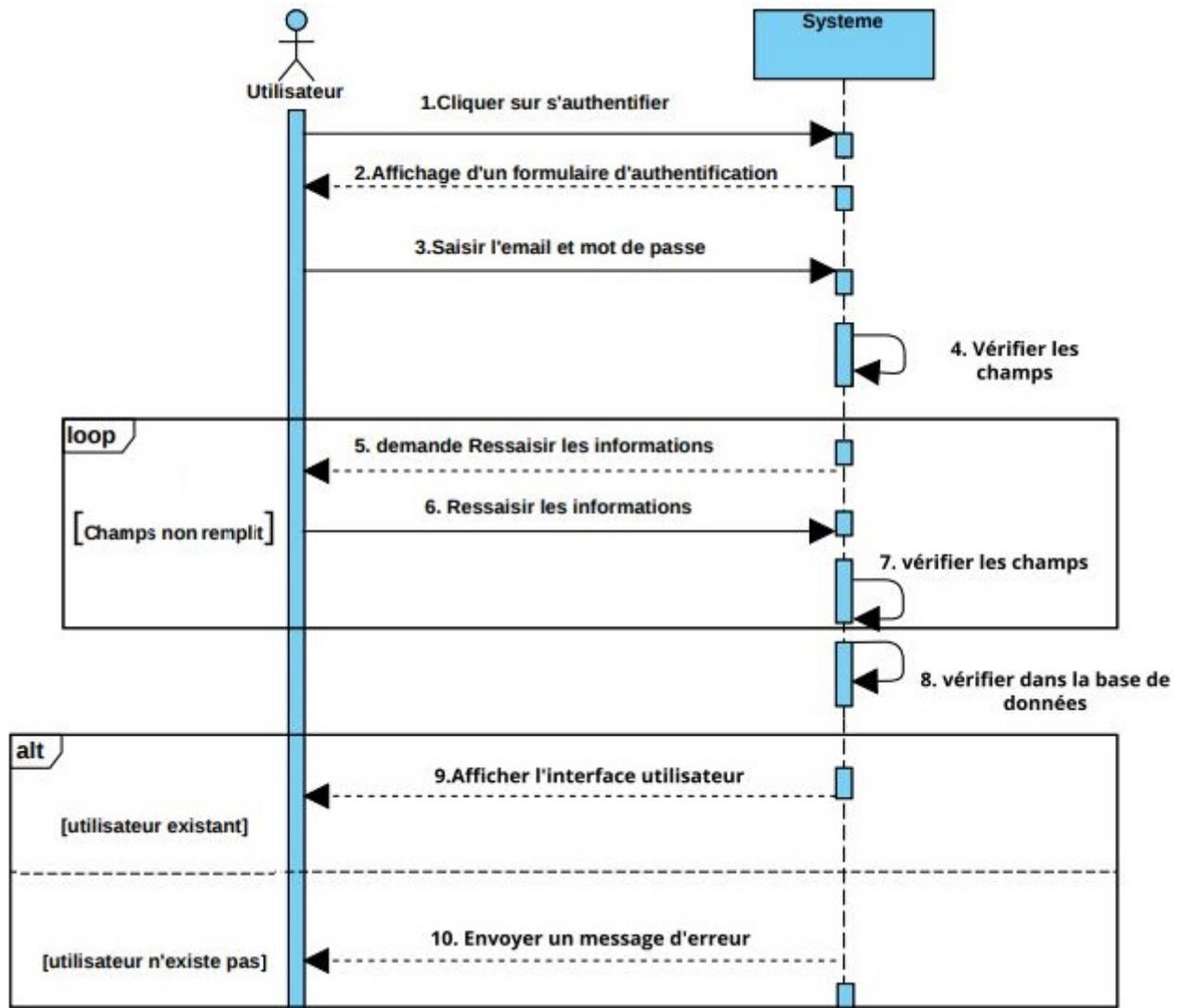


FIGURE 3.3 – Diagramme de séquence du cas d'utilisation «S'authentifier»

3.3.4.2 Diagramme de séquence du cas d'utilisation « Poster un avis de recherche pour une personne perdue »

Pour poster un avis de recherche, l'utilisateur doit être authentifié. Il remplira ensuite le formulaire de publication. Le système doit vérifier si tous les champs (obligatoires) ont bien été saisis. Il sauvegardera enfin l'avis dans la base de données.

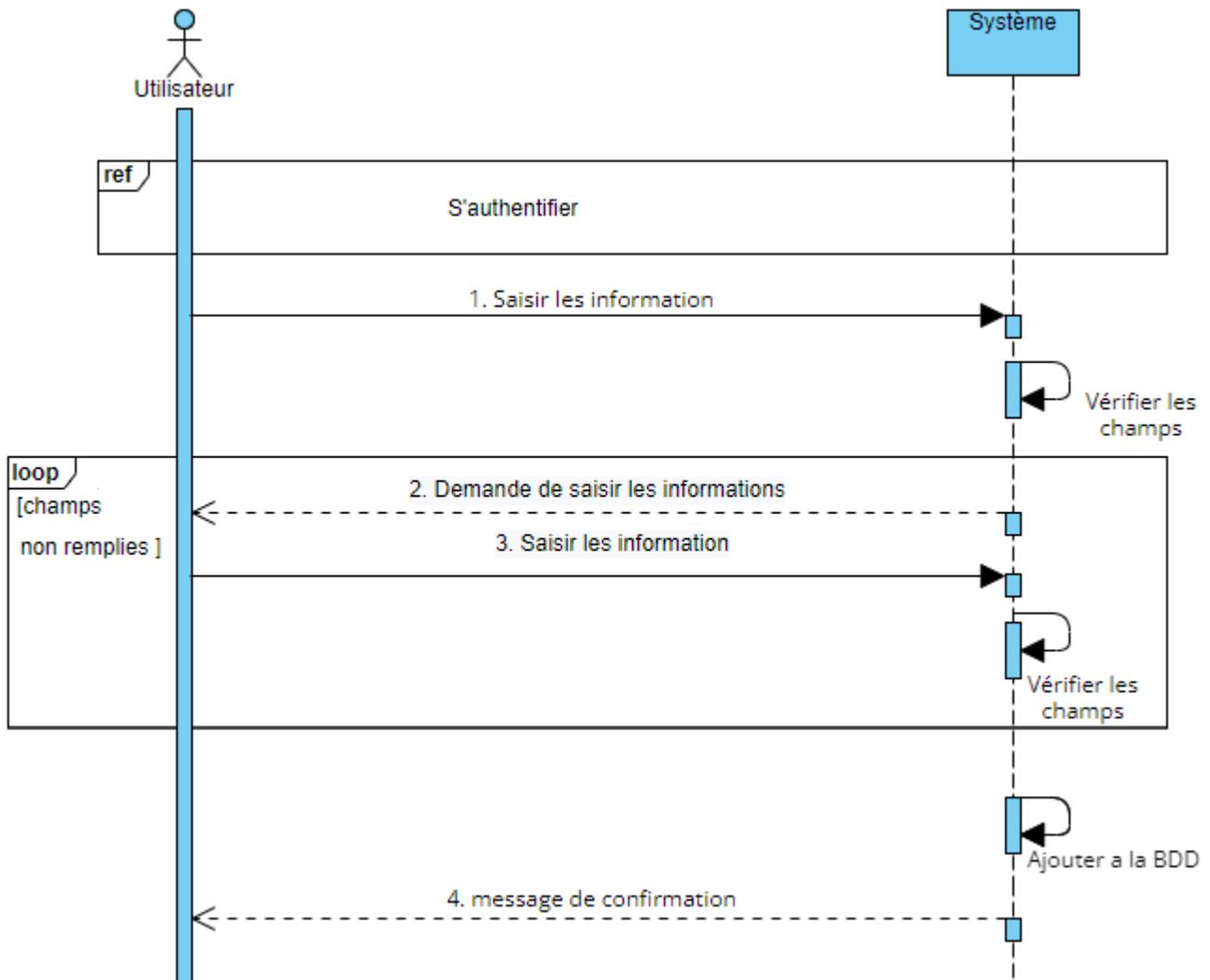


FIGURE 3.4 – Diagramme de séquence du cas d'utilisation «Poster un avis pour une personne perdue».

3.4 Diagramme d'interaction

Chaque diagramme d'interaction représente un ensemble d'objets de différentes classes qui fonctionnent ensemble dans le cadre d'un scénario d'exécution du système. Dans ce type de diagramme, les objets communiquent en s'envoyant des messages les uns aux autres en invoquant l'opération de l'objet récepteur. Ces objets sont des instances de trois classes d'analyse différentes (interfaces, contrôles, entités) [20].

3.4.1 Diagramme d'interaction « S'authentifier »

L'utilisateur doit saisir ses identifiants dans le formulaire d'authentification. Suivant le résultat du contrôle effectué par le système, il se retrouve soit sur la page d'accueil, soit de nouveau sur le formulaire avec un message d'erreur. Le scénario qui s'exécute est représenté par la figure 3.5.

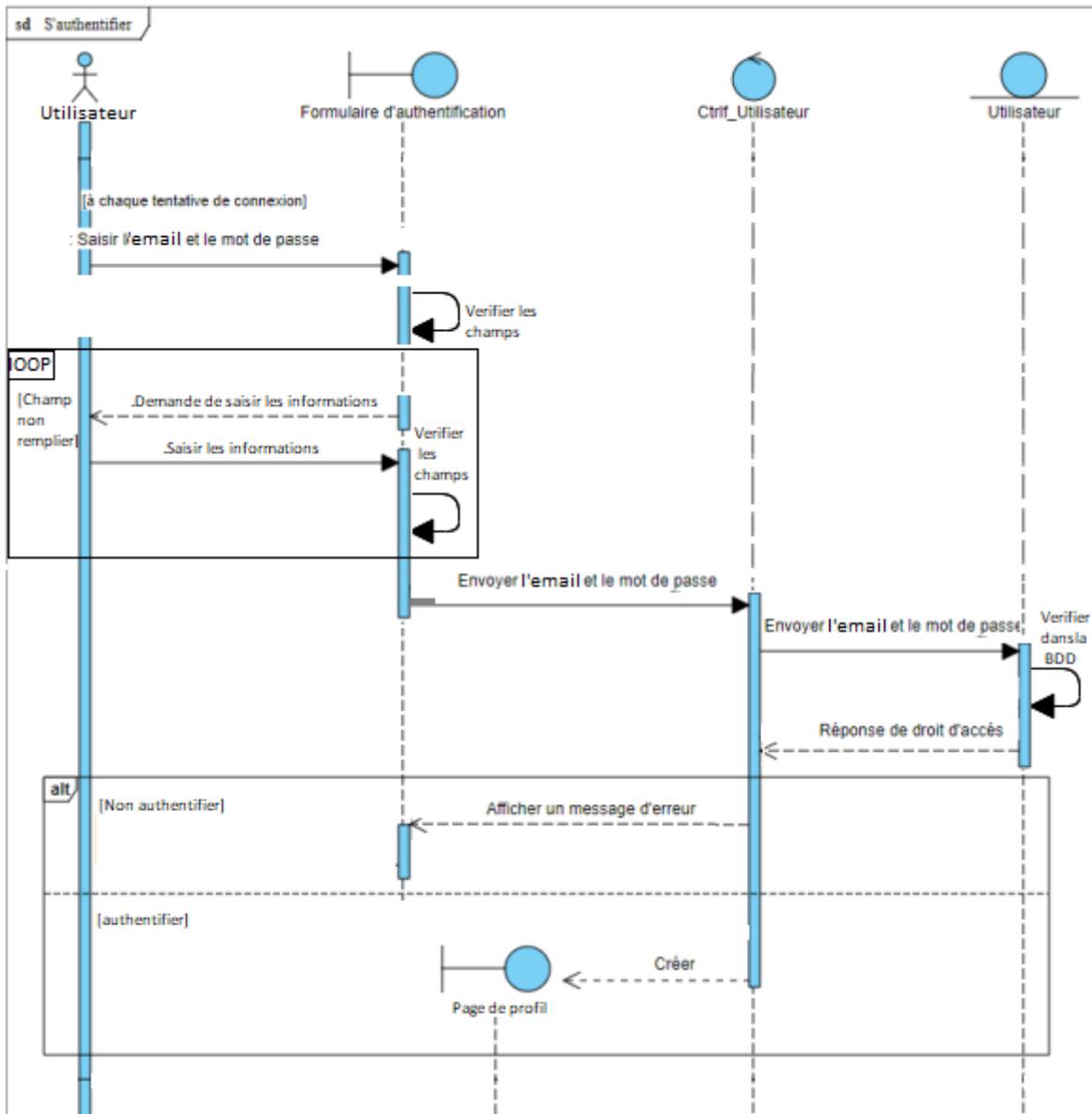


FIGURE 3.5 – Diagramme d'interaction du cas d'utilisation « S'authentifier »

3.5 Diagramme de navigation

Le diagramme de navigation est un diagramme dynamique représentant de manière formelle l'ensemble des chemins possibles entre les principales interfaces proposées à l'utilisateur. Il est considéré comme une vue concrète mais non définitive de l'IHM [20].

La figure 3.6 représente le diagramme de navigation de notre système.

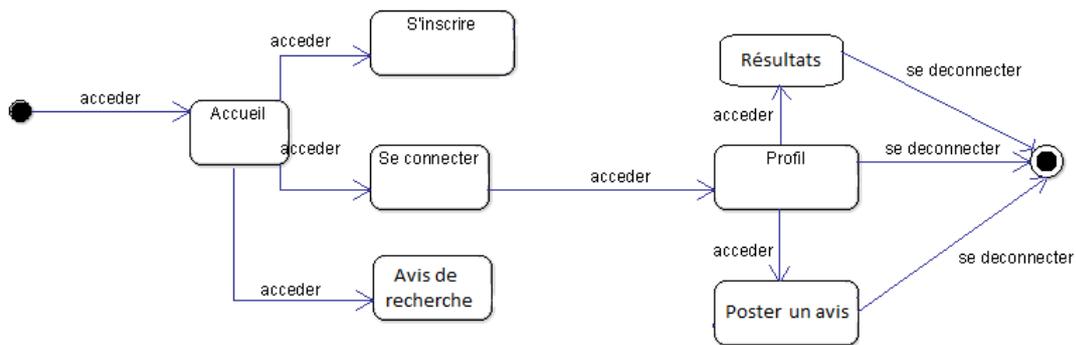


FIGURE 3.6 – Diagramme de navigation de notre application web.

3.6 Diagramme de classe

Une classe est un ensemble de fonctions et de données (attributs) liées les unes aux autres. par le champs sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser des programmes et donc de décomposer des tâches complexes en quelques petites tâches faciles.

Les diagrammes de classes sont l'un des types de diagrammes UML les plus utiles. Il permet de décrire clairement la structure d'un système particulier, les attributs, leurs opérations et les relations entre leurs objets [20].

3.6.1 Diagramme de classe de domaine

La figure 3.7 montre le diagramme de classe de domaine de notre application Web.

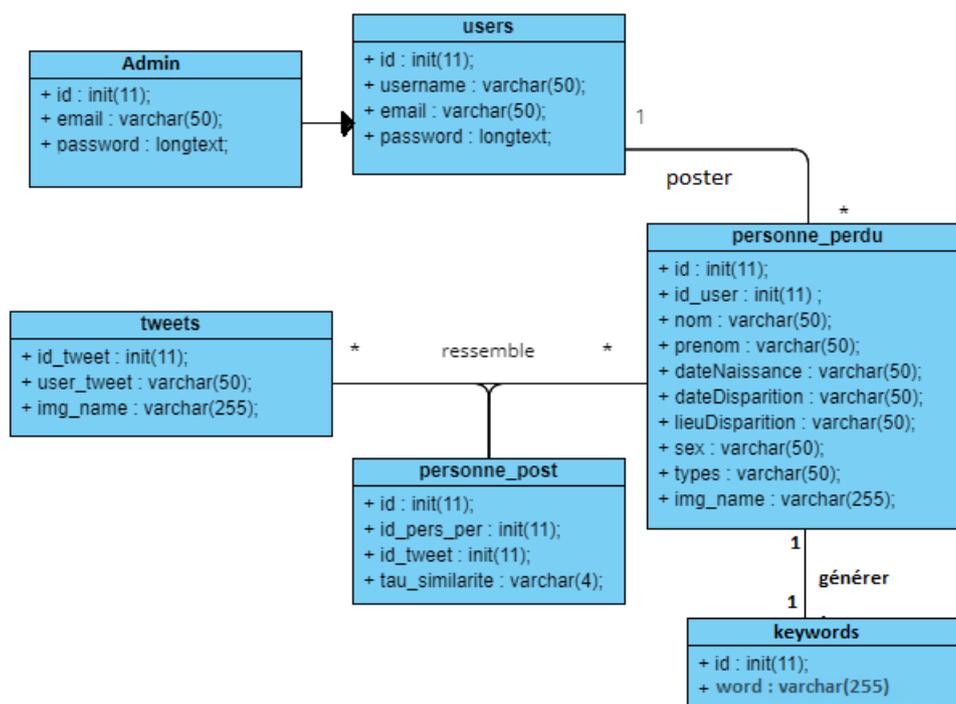


FIGURE 3.7 – Diagramme de classe de domaine

3.6.2 Classe, attributs et responsabilités

Le tableau 3.5 décrit la responsabilité de chaque classe et définit les attributs qu'elle contient.

| Classe | Responsabilité | Attributs | | |
|-----------------------|---|-----------------|---|----------------------|
| | | Intitulé | Définition | Type |
| Users | Classe qui représente les personnes inscrites dans notre application Web | id | Identifiant | Numérique |
| | | username | Nom d'utilisateur | Chaîne de caractères |
| | | email | É-mail de l'utilisateur | Chaîne de caractère |
| | | password | Mot de passe de l'utilisateur | Chaîne de caractères |
| Personneperdue | Classe qui représente la liste d'avis de recherche des personnes perdues de notre application Web | id | identifiant | Numérique |
| | | idUser | Identifiant de l'utilisateur | Numérique |
| | | nom | Nom de personne perdu | Chaîne de caractères |
| | | prenom | Prénom de personne perdu | Chaîne de caractères |
| | | dateNaissance | Date de naissance de personne perdu | Chaîne de caractères |
| | | dateDisparition | Date de disparition de personne perdu | Chaîne de caractères |
| | | lieuDisparition | Lieu de disparition de personne perdu | Chaîne de caractères |
| | | sex | Sexe de personne perdu | Chaîne de caractères |
| | | type | Type de publication (privé, publique) | Chaîne de caractères |
| | | imgname | Nom de l'image de la personne perdu | Chaîne de caractères |
| tweets | Représente la liste des personnes retrouvées | id_tweet | Identifiant obtenu par l'API twitter | Numérique |
| | | user_tweet | Nom d'utilisateur obtenu par l'API twitter | Chaîne de caractères |
| | | img_name | L'url de l'image obtenu par l'API twitter | Chaîne de caractères |
| Personne_post | Classe association entre la classe Personneperdue et tweets | id | identifiant | Numérique |
| | | id_pers_per | Identifiant de personne perdu | Numérique |
| | | id_tweet | Identifiant de personne retrouver | Numérique |
| | | tau_similiarité | Tau de similarité entre personne perdue/retrouvée | Chaîne de caractères |
| Admin | Représente les administrateurs de notre application Web. | id | identifiant | Numérique |
| | | email | É-mail de l'administrateur | Chaîne de caractères |
| | | password | Mot de passe de l'admin | Chaîne de caractères |
| Keywords | Représente les mot qu'on doit rechercher sur twitter. | id | identifiant | Numérique |
| | | word | Le nom de la personne perdue | Chaîne de caractères |

TABLE 3.5 – Description des attributs de classes de notre application Web

3.7 Schéma relationnel

À partir du diagramme de classe, nous passons au modèle relationnel qui est le modèle de données logique. Ce modèle décrit comment les données sont représentées dans une base de données. Pour décrire une relation, nous indiquons tout simplement son nom, suivi du nom de ses attributs entre parenthèses.

Un identifiant de relation consiste en un ou plusieurs attributs qui forment la clé primaire. Une relation peut en référencer une autre via une clé étrangère qui correspond à la clé primaire de la relation référencée [21].

3.7.1 Règles de passage au modèle relationnel

Les règles de passage au modèle relationnel sont les suivantes [21] :

1. Relation (1..*) : la clé primaire de la table parent doit être ajoutée comme clé étrangère dans la table enfant.
2. Relation (1..1) : la relation doit être ajoutée comme table ou la clé primaire est la combinaison des clés primaire des deux classes (qui sont aussi clés étrangères).
3. Relation d'héritage : selon les restrictions existantes, trois décompositions sont possibles pour transformer les associations d'héritage :
 - Décomposition par distinction : il faut convertir chaque sous-classe en relation. La clé primaire de la classe parent va à la relation qui est le résultat de la sous-classe et devient à la fois la clé primaire et la clé étrangère.
 - Décomposition descendante (push-down) : tous les attributs de la classe parent doivent être migrés vers les relations résultant de la sous-classe.
 - Décomposition ascendante (push up) : dans ce cas, il faut supprimer les relations issues des sous classes et faire migrer tous les attributs dans la relation issue de la classe mère.

3.7.2 Le modèle relationnel de notre application Web

Après avoir appliqué toutes les règles de passage au modèle relationnel, nous avons obtenu le schéma suivant :

1. Users (id, username, email, password)
2. Personneperdu (id, nom, prenom, dateNaissance, dateDisparition, lieuDisparition, sex, types, imgname, #iduser)
3. Tweets (id_tweet, user_tweet, img_name)
4. Personne_post (id, tau_similarite, #id_pers_pos, #id_pers_per)
5. Admin (id, email, password)
6. Keywords (id, word)

3.8 Schéma récapitulatif du système de reconnaissance faciale abordé

La figure 3.8 montre le schéma récapitulatif de la reconnaissance faciale suivi dans notre application Web.

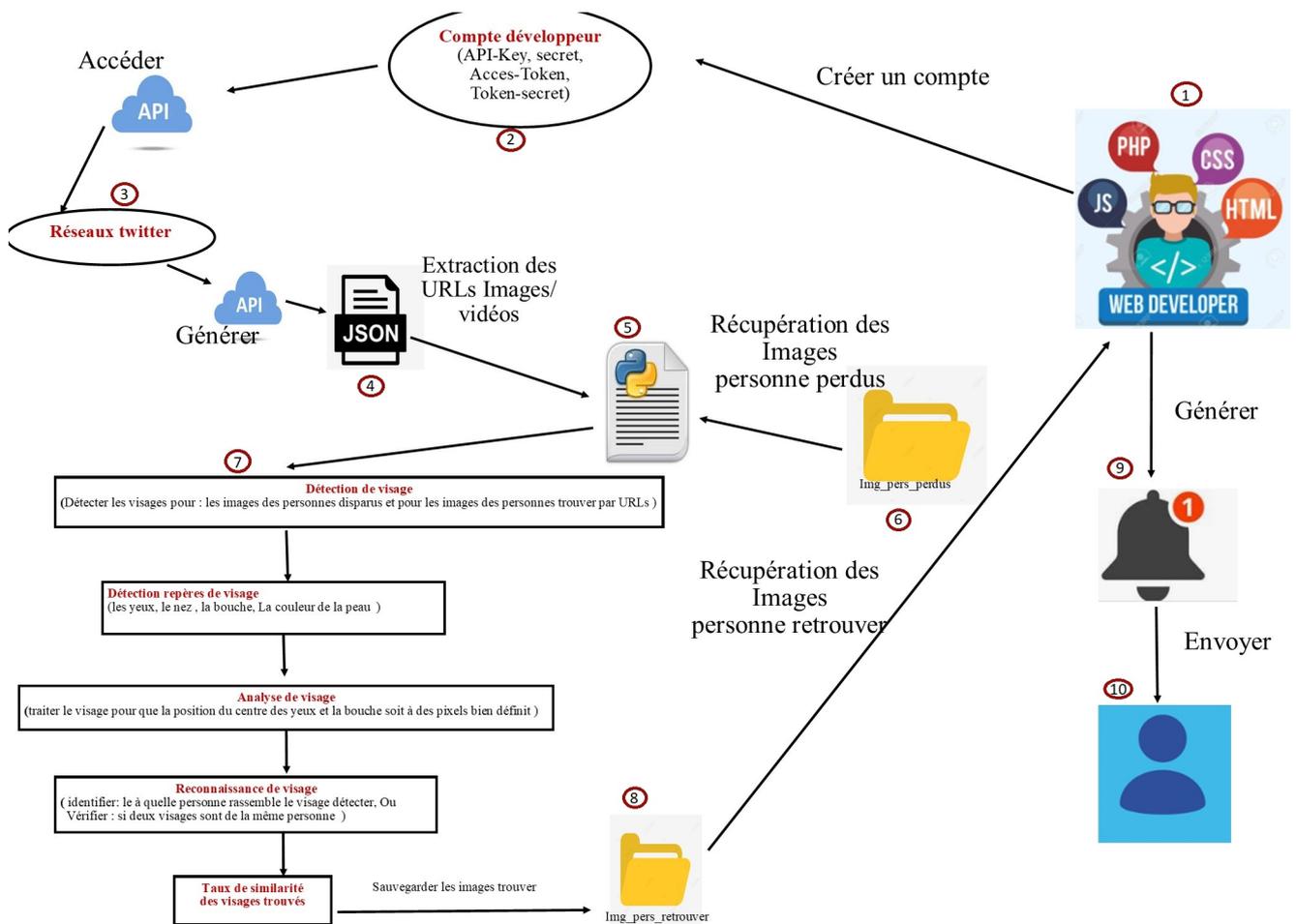


FIGURE 3.8 – Schéma récapitulatif de notre système de reconnaissance faciale

3.9 Conclusion

Nous avons présenté dans ce chapitre une vue générale du formalisme UML. Nous avons expliqué le fondement de fonctionnement et les relations qui existent entre les différents acteurs et leur interaction avec le système. Nous avons aussi élaboré les diagrammes de cas d'utilisation et de séquences adopté par notre système.

Dans le dernier chapitre nous présenterons la partie réalisation de notre projet.

Chapitre 4

Réalisation de l'application

4.1 Introduction

Dans ce dernier chapitre, nous décrivons la phase d'implémentation et réalisation qui est une étape très importante dans le cycle de vie d'une application. Nous allons d'abord définir les outils de développement et les langages de programmation utilisés, puis décrire la mise en œuvre de l'entrepôt de données. Nous allons également présenter les étapes suivies pour la modélisation des prédictions à l'aide de l'apprentissage automatique. Nous terminons le chapitre par la présentation de quelques interfaces graphiques et jeux de tests.

4.2 Les technologies utilisées

Durant la réalisation de notre application Web on a utilisé plusieurs technologies, et les plus importantes sont décrites ci-bas.

4.2.1 L'API Twitter

Une API est une interface constituée d'un ensemble de méthodes qu'un site Web expose aux développeurs et leur permet d'utiliser des fonctions et d'accéder à des données de site Web spécifiques. Cette interface simplifie la communication entre les différents types de clients (sites web, ordinateurs, applications mobiles, logiciels CRM, etc.) et les serveurs qui s'exécutent à la demande du client [22].

Pour notre application, nous avons utilisé l'API Twitter. Cette dernière est une passerelle ou interface de programmation permettant de se connecter aux données Twitter de façon automatisée. L'API Twitter peut être utilisée pour afficher automatiquement des tweets sur un site web ou pour extraire des données sur les réseaux sociaux. Dans ce dernier cas, on utilise la Search API de Twitter. Une API Twitter peut également par exemple être utilisée pour extraire et faire défiler à l'écran les tweets utilisant le hashtag lié à une émission de télévision [23]. Plus précisément, l'API Twitter possède les caractéristiques suivantes [24] :

- Il utilise les quatre objets principaux : Tweets, Entités, Lieux et Utilisateurs.
- Il existe des restrictions quotidiennes : les appels et les modifications apportées à l'API sont limités par des jetons d'accès pour protéger la plate-forme contre les abus.
- Elle est basée sur HTTP (plutôt que SSL).

- Il existe des mesures spécifiques pour adapter le fonctionnement de l'API au réseau social, y compris les restrictions de bibliothèque, la pagination générée et des paramètres spécifiques.

4.2.1.1 Utilisation de l'API twitter

Pour commencer, il nous a fallu d'abord créer un compte développeur sur "<https://developer.twitter.com/>". La figure 4.1 représente un capteur d'écran de notre compte développeur twitter.

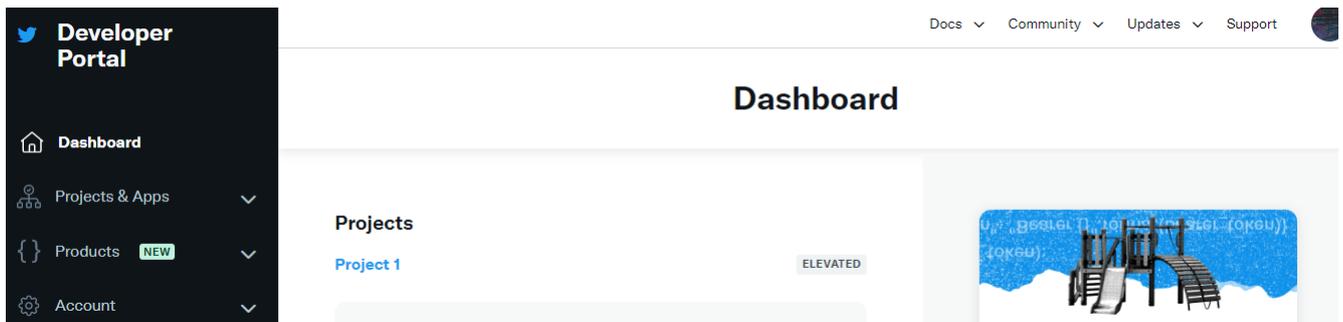


FIGURE 4.1 – Compte développeur.

Puis, nous avons créé un nouveau projet pour récupérer les clés du consommateur, et jetons d'authentification, comme illustré dans La figure 4.2 :

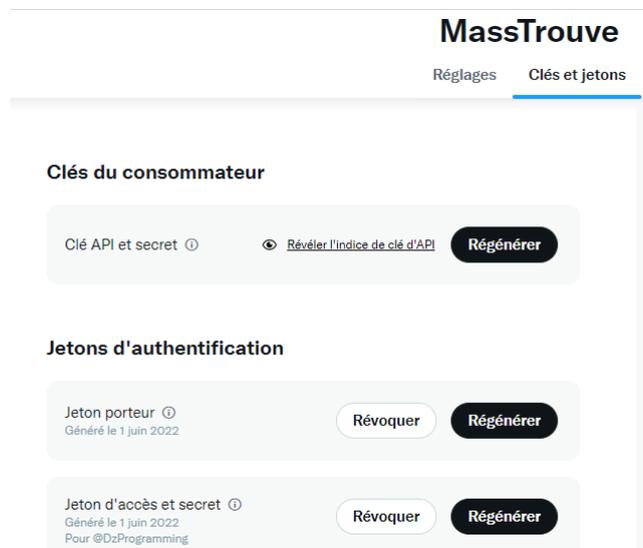


FIGURE 4.2 – Les clés API du projet.

Après la création du compte développeur, nous avons pu intégrer les APIs twitter dans notre projet. Nous avons créé un fichier *config.ini* pour configurer le compte twitter développeur et grâce à la bibliothèque "*tweepy*" que nous avons pu installer par la commande "*pip install tweepy*", nous avons pu avoir accès à axu différentes fonctionnalités de twitter comme [24] :

- Envoyer un tweet.
- Télécharger un fichier multimédia.
- Répondre à un tweet.

- Récupérer des tweets sous un hashtag spécifique.
- Trouver des sujets tendances localement.
- Créer des bots.

4.2.2 La reconnaissance faciale

Dans le cadre de ce projet, nous avons utilisé la bibliothèque de reconnaissance faciale *face_recognition* pour reconnaître et manipuler les images de visages depuis Python. Pour son installation, nous avons utilisé la commande `pip3 install face_recognition`. Nous avons ainsi pu utiliser les paquets suivants :

- `face_recognition.load_image_file("nom du fichier image")` qui permet de charger un fichier image pour la reconnaissance faciale.
- `face_recognition.face_locations(image)` qui renvoie une liste d'encodages des visages humains dans une image.
- `face_recognition.face_encodings(image)` qui renvoie une liste d'encodages de visage à 128 dimensions (pour chaque visage de l'image).
- `face_recognition.face_distance(face_encodings , face_to_compare)`. Étant donné une liste d'encodages de visage et un encodage de visage connu, on peut avoir la distance euclidienne pour chaque visage de comparaison. La distance indique à quel point les visages sont similaires.
- `face_recognition.compare_faces(codages_face_connus, codage_face_à_vérifier)` qui renvoie une liste de valeurs Vrai/Faux indiquant quels `codages_face_connus` correspondent au codage de visage à vérifier.

4.2.2.1 Exemple de code pour la reconnaissance faciale

Le code de la figure 4.4 montre comment reconnaître un visage dans une image qu'on a sur notre ordinateur (figure 4.3) avec une image qui existe sur Twitter (Figure 4.6).

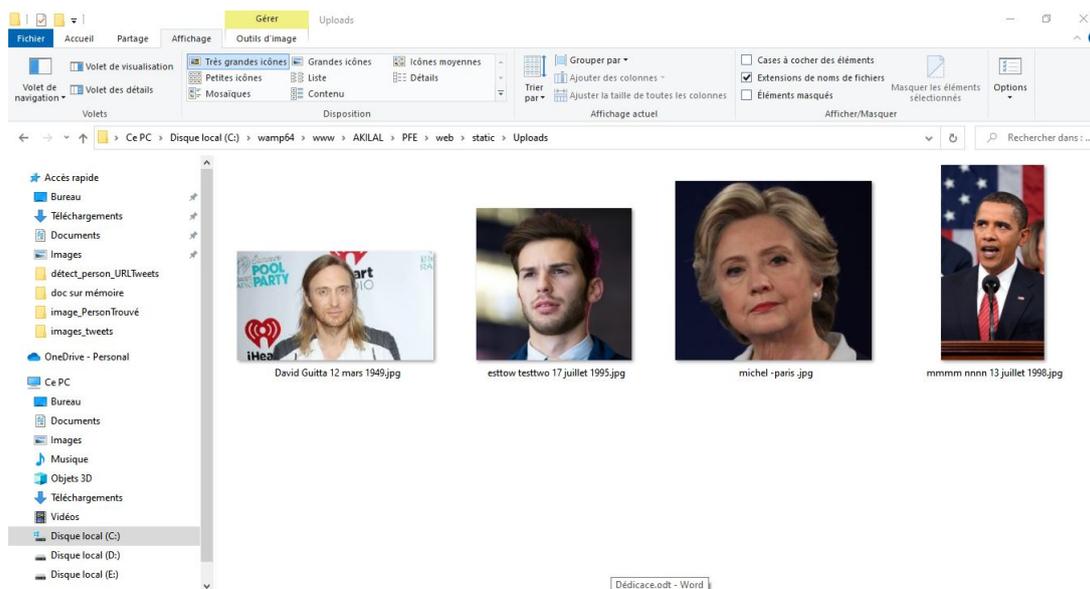


FIGURE 4.3 – Le dossier qui contient les images des personnes perdus

```

1  """ importation des module """
2  import os, re
3  from PIL import Image, ImageDraw
4  import urllib.request
5  import face_recognition
6  import numpy as np
7  """chargement des photos des personnes perdu"""
8  encodage_visage_connu = []
9  known_face_names = []
10 known_faces_filenames = []
11 for (dirpath, dirnames, filenames) in os.walk("ImgpersPerdu/"):
12     known_faces_filenames.extend(filenames)
13     break
14 for filename in known_faces_filenames:
15     faces = face_recognition.load_image_file("ImgpersPerdu/" + filename)
16     known_face_names.append(re.sub("[ ]", '', filename[:-4]))
17     encodage_visage_connu.append(face_recognition.face_encodings(faces)[0])
18     nom_personne_perdu =[]
19 """Charger une urls image tweet comme visage inconnu"""
20 url =('https://pbs.twimg.com/media/FSu91K2WQAAw_Tl.jpg')
21 dir=os.path.abspath( "image_tweets/FSu91K2WQAAw_Tl.jpg")
22 media =urllib.request.urlretrieve(url, dir)
23 image_inconnu = face_recognition.load_image_file(dir)
24 """Trouver tous les visages et encodages de visage dans l'image inconnue"""
25 emp_visage_inconnu = face_recognition.face_locations(image_inconnu)
26 encodage_visage_inconnu = face_recognition.face_encodings(image_inconnu, emp_visage_inconnu)
27 image_pil = Image.fromarray(image_inconnu)
28 draw = ImageDraw.Draw(image_pil)
29 # Traverser chaque visage trouve dans L'image inconnue
30 for (haut, droite, bas, gauche), encodage_visage in zip(emp_visage_inconnu, encodage_visage_inconnu):
31     # Voir si le visage correspond au visage connu
32     corresp = face_recognition.compare_faces(encodage_visage_connu, encodage_visage)
33     nom = "Inconnu"
34     # utilisez le visage connu avec la plus petite distance par rapport au nouveau visage
35     distances_visages = face_recognition.face_distance(encodage_visage_connu, encodage_visage)
36     meilleur_indice = np.argmin(distances_visages)
37     if corresp[np.argmax(distances_visages)] >= 0.3 :
38         nom = known_face_names[np.argmax(distances_visages)]
39         #Afficher le cnom de la personne sur L'image
40         nom_personne_perdu.append(nom)
41         #- Enregistrer L'image
42         image_pil.save("Image_PersonTrouvé/imagetweet.jpg")
43         #score similarité
44         score_similarité = distances_visages * 100
45         ss = max(np.array(score_similarité))
46         print("le score de similarité entre le visage de : " ,nom ,"\n",
47               "est le Visage trouvé dans URL : " ,url,"\n",
48               "et le visage trouvé:" , ss , "%","\n","\n",)
49         # Dessinez une boîte autour du visage a l'aide du module Pillow
50         draw.rectangle((gauche, haut), (droite, bas), outline=(0, 0, 255))
51         # Dessinez une étiquette avec un nom sous le visage
52         largeur_texte, hauteur_texte = draw.textsize(nom)
53         draw.text((gauche + 6, bas - hauteur_texte - 5), nom, fill=(255, 255, 255, 255))
54 image_pil.show()

```

FIGURE 4.4 – Exemple de code pour faire la reconnaissance faciale.

Le résultat de l'exécution de ce code est la figure 4.5 représente les images des personnes qu'on souhaite trouver sur Twitter :

```
le score de similarité entre le visage de : michel-paris
est le Visage trouvé dans URL : https://pbs.twimg.com/media/
FSu91K2wQAAw_T1.jpg
et le visage trouvé: 88.90991302671111 %
```

FIGURE 4.5 – Résultat d’exécution de code sur le terminale de notre éditeur spyder



FIGURE 4.6 – L’image trouver sur twitter

4.2.3 Langages utilisés

4.2.3.1 HTML

HTML est un langage informatique descriptif. Plus précisément, il s’agit d’un format de données utilisé dans le monde Internet pour formater des pages Web. Il permet non seulement l’écriture d’hypertexte, mais permet également l’introduction de ressources multimédias dans le contenu [25]. Le format ou langage HTML développé par le W3C (World Wide Web Consortium) et le WHATWG (Web HyperText Application Technology Working Group) est apparu dans les années 1990 et a été progressivement révisé, et une version HTML5 plus complète est fournie depuis 2014 [25].

HTML est basé sur un système de balises, qui permet d’ajouter des titres, des sous-titres, des caractères gras, etc., du texte et l’introduction d’éléments interactifs, tels que des images, des liens, des vidéos... HTML en tant que langage est plus facile pour les robots des moteurs de recherche pour comprendre JavaScript, et il est également utilisé pour rendre la page plus interactive [25].

4.2.3.2 JavaScript

JavaScript est un langage de programmation qui permet d’implémenter des mécanismes complexes sur des pages Web. Chaque fois qu’une page Web affiche plus que du contenu statique mis à jour, des cartes interactives, des animations 2D/3D, des menus vidéo défilants, etc., à un moment donné, JavaScript est susceptible d’être impliqué. Il s’agit de la troisième couche de la technologie Web standard, les deux premières couches étant (HTML et CSS) [26].

4.2.3.3 CSS

Les feuilles de style ("cascade style Sheets" en anglais, CSS en abrégé) sont des feuilles de style qui permettent de gérer l'affichage des pages web. CSS, comme HTML et XML, est la norme recommandée par le World Wide Web Consortium (W3C). Le but de CSS est de séparer la structure d'un document HTML de son affichage [27].

En fait, il est possible d'utiliser HTML pour définir des structures (structures hiérarchiques entre différentes parties du contenu et des documents) et des présentations. Cependant, cela pose quelques problèmes. Il est possible d'utiliser un couple HTML / CSS pour créer une page Web où la structure du document est dans un fichier HTML et la présentation est dans un fichier CSS [27].

4.2.3.4 MySQL

MySQL est un système de gestion de base de données relationnelle (SGBDR) unique, gratuit, performant, largement utilisé, multithread et multi-utilisateurs. MySQL est avant tout un serveur de base de données. Le client est utilisé pour se connecter localement ou à distance. Cela peut être une commande MySQL ou un script PHP normal. Dans ce cas, il faut installer le module PHP-MySQL qui permet à PHP de communiquer avec le serveur MySQL [28].

4.2.3.5 Python

Python est un langage de programmation polyvalent à usage général. C'est un langage interprété et peut être exécuté de manière interactive à partir de la console. Le temps de développement a tendance à être plus court car aucun compilateur n'est requis [29].

Il est disponible en téléchargement gratuit et peut être installé sur de différents systèmes d'exploitation. Il est particulièrement populaire pour les applications scientifiques et mathématiques et bénéficie donc d'un grand nombre de bibliothèques complémentaires utiles développées par sa grande communauté. Python est facile à apprendre par rapport aux autres langages[29].

4.2.4 Environnement de développement utilisés

4.2.4.1 WampServer

WampServer est une plate-forme de développement Web Windows pour les applications Web dynamiques qui utilisent des serveurs Apache2, des langages de script PHP et des bases de données MySQL. Il existe également PHPMyAdmin qui aide à gérer plus facilement la base de données [25].

WampServer peut lancer des sites Web dynamiques, qu'ils soient connectés à Internet ou localement. Il peut être utilisé dans un simple navigateur et inclut un serveur Apache pour le transfert de fichiers en ligne. Ce logiciel prend en charge la gestion des banques d'informations [25].

4.2.4.2 PhpMyAdmin

PhpMyAdmin est l'interface de gestion du SGBD MySQL. Écrit en langage PHP, il fournit à un administrateur de base de données de serveur une interface utilisateur intuitive. Il est possible de gérer avec les éléments suivants [26] :

- Base de données.
- Les tables et leurs champs (ajout, suppression, définition de type).
- Index, clé primaire et clé étrangère.

- Les utilisateurs de la base de données et leurs autorisations.
- Exporter des données dans divers formats (CSV, XML, PDF, Open Document, Word, Excel, LaTeX).

4.2.4.3 Visual Paradigme for UML

Visual Paradigme est un logiciel permettant de créer des diagrammes dans un programme. Tout en un, il dispose de plusieurs options qui permettent différentes possibilités de modélisation en UML [27].

4.2.4.4 Adobe Photoshop

Adobe Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur sur Windows. Publié par Adobe, il est principalement utilisé pour l'édition de photos numériques, mais il est également utilisé pour créer des images à partir de nihilo. L'image se compose d'une grille de points appelés pixels, elle fonctionne donc essentiellement avec des images raster. L'intérêt de ces images est de reproduire de subtils dégradés de couleurs [28].

4.2.4.5 Visual studio code

Visual Studio Code est un éditeur de code simplifié prenant en charge les opérations de développement telles que le débogage, l'exécution de tâches et le contrôle de version. Il est développé par Microsoft pour Windows, Linux et MacOS. Il vise à fournir uniquement les outils dont un développeur a besoin pour un cycle de création de code-débogage rapide et laisse les flux de travail plus complexes à des IDE plus complets [29].

4.2.4.6 Spyder

Est un environnement de développement pour Python. Libre (Licence MIT) et multiplateforme (Windows, Mac OS, GNU/Linux), il intègre de nombreuses bibliothèques d'usage scientifique comme Matplotlib, NumPy, SciPy et IPython.

Créé et développé par Pierre Raybaut en 2008, Spyder est maintenu, depuis 2012, par une communauté de développeurs qui ont pour point commun d'appartenir à la communauté Python scientifique.

En comparaison avec d'autres IDE pour le développement scientifique, Spyder a un ensemble unique de fonctionnalités - multiplateforme, open-source, écrit en Python [30].

4.2.5 Frameworks utilisés

Un Framework est un ensemble d'outils et de composants logiciels organisés selon un plan architectural et un modèle de conception. Le tout forme un cadre de programmation. Il est généralement fourni sous la forme d'une bibliothèque logicielle avec un plan pour l'architecture cible du Framework. Dans la programmation orientée objet, les Framework sont généralement constitués de superclasses, qui sont dérivées et étendues par héritage en fonction des besoins spécifiques de chaque logiciel qui utilise le Framework [31]. Ci-après, les Frameworks que nous avons utilisés dans notre application.

4.2.5.1 Bootstrap

Bootstrap est un Framework de développement frontal gratuit et open source pour la création de sites Web et d'applications Web. Le Framework Bootstrap est construit sur HTML, CSS et JavaScript pour faciliter le développement de sites et d'applications réactifs et mobiles [31].

La conception réactive permet aux pages Web ou aux applications de percevoir la taille et l'orientation de l'écran du visiteur, et d'ajuster automatiquement l'affichage en conséquence ; l'approche axée sur le mobile suppose que les Smartphones, les tablettes et les applications mobiles pour des tâches spécifiques sont des employés qui terminent leur travail et se rencontrent ces technologies dans la conception Le principal outil requis [32].

Bootstrap comprend des composants d'interface utilisateur, une mise en page et des outils Javascript et un cadre de mise en œuvre. Le logiciel peut être pré-compilé ou utilisé comme code source [32].

4.2.5.2 Flask

Flask est un micro framework open-source de développement web en Python. Il est classé comme microframework car il est très léger [33]. Flask a pour objectif de garder un noyau simple mais extensible. Il n'intègre pas de système d'authentification, pas de couche d'abstraction de base de données, ni d'outil de validation de formulaires. Cependant, de nombreuses extensions permettent d'ajouter facilement des fonctionnalités [34].

4.3 Présentation des interfaces de notre application Web

4.3.1 Interface « Page d'Accueil »

La figure 4.7 représente la page d'accueil de notre application Web. Elle offre une présentation globale de notre site, un espace pour le contact pour d'éventuelles informations et contient des liens vers les différentes pages du site .

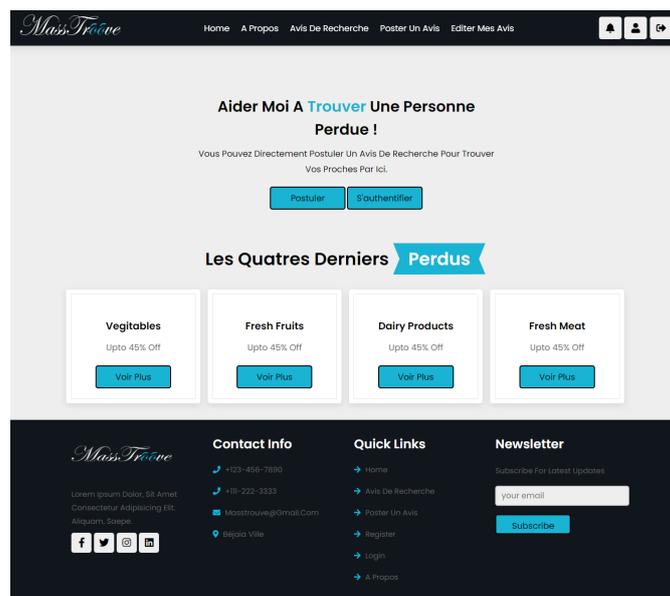


FIGURE 4.7 – Interface de la « Page d'Accueil » de notre application Web.

4.3.2 Interface « À propos »

La figure 4.8 met en évidence les objectifs du centre ainsi que ceux du site et son fonctionnement et sa géo position.

A Propos De Notre Application

Notre Site Web Pour La Recherche Des Personnes Perdus, Qui Mettra En Relation Les Personnes Perdus Est Ce Qui Les Recherche Ou Ce Dernier Peut Créer Son Propre Compte.

Après La Création De Compte L'utilisateur Peut Ajouter Un Avis De Recherche Qui Peut Être Soit Public Qui Être Afficher Dans La Page « Avis De Recherche » Ou Bien Privé Qui Ne Va Pas Être Publier Mais Qui Va Être Afficher Dans Son Compte Avec Tous Les Avis Qui A Publié Ou Il Peut-Il Les Modifier Ou Les Supprimer, Et Peut Aussi Voir Les Autres Avis Publier En Public.

De Coté Admin Notre Site Ramené Tous Les Vidéos Et Image Qui Se Trouve Dans Les Réseaux Sociaux Grâce Au API Pour Les Comparer Avec Les Image Des Personnes Perdus Et On Cas Ou Il Trouve Cette Personne Une Notification Sera Envoyer Au Utilisateur.

Comment Ça Marche :

1. Vous Crée Un Compte.
2. Vous Crée Un Avis De Recherche Dans La Page « Poster Un Avis ».
3. Vous Pouvez Soit Publier Un Avis Public Ou Privé.
4. Vous Attendez Une Notification On Cas Ou Le Site Trouve La Personne.
5. Vous Pouvez Voir Les Autres Avis Dans « Avis De Recherche ».

Caractéristique De L'Application

1. Site Web 100% Sécurisé.
2. Réduire Les Coûts Et Les Délais.
3. Accéder A Des Candidats Inconnus.
4. Motivation.
5. Effort De Recherche Pour Trouver Vos Proche.
6. Plus Des Experts.

Notre Position

FIGURE 4.8 – Interface « A propos » de notre application Web.

4.3.3 Interface « Avis de recherche »

La figure 4.9 représente la page avis de recherche où on trouve tous les avis de recherche postés en public.

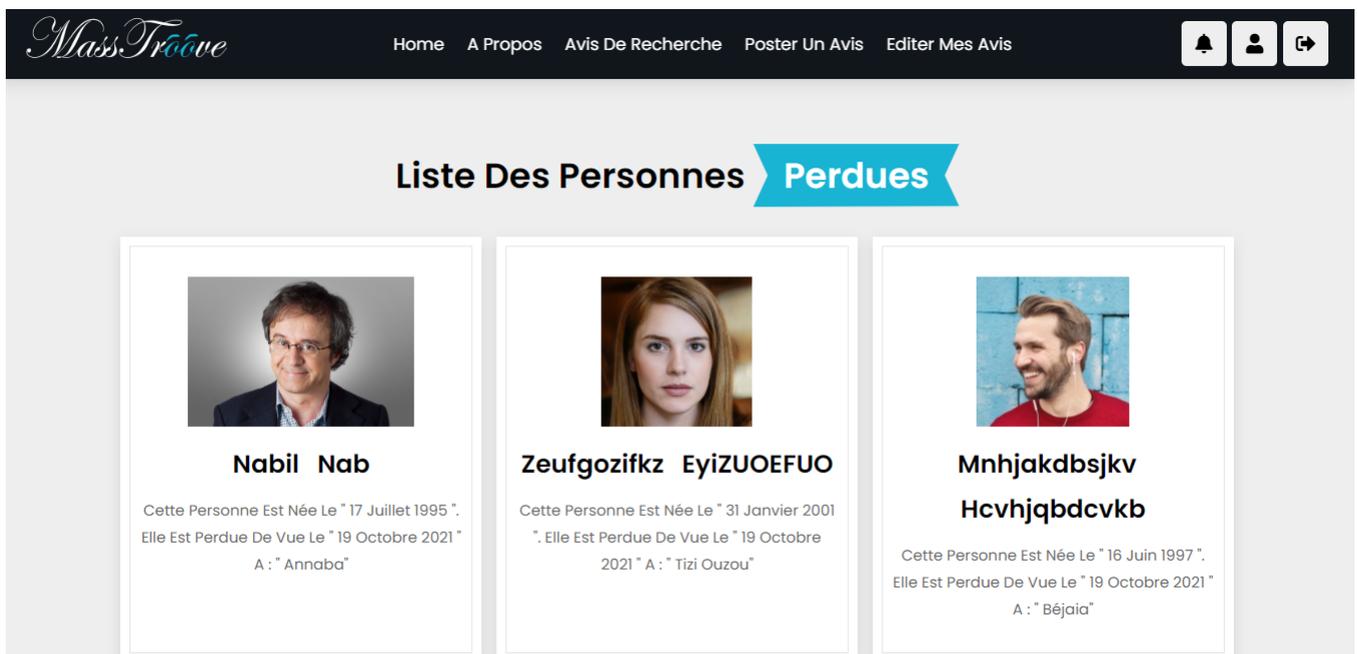


FIGURE 4.9 – Interface « Avis de recherche » de notre application Web.

4.3.4 Interface « Inscription »

La figure 4.10 représente l'interface d'inscription où un visiteur remplit le formulaire pour pouvoir créer un compte.

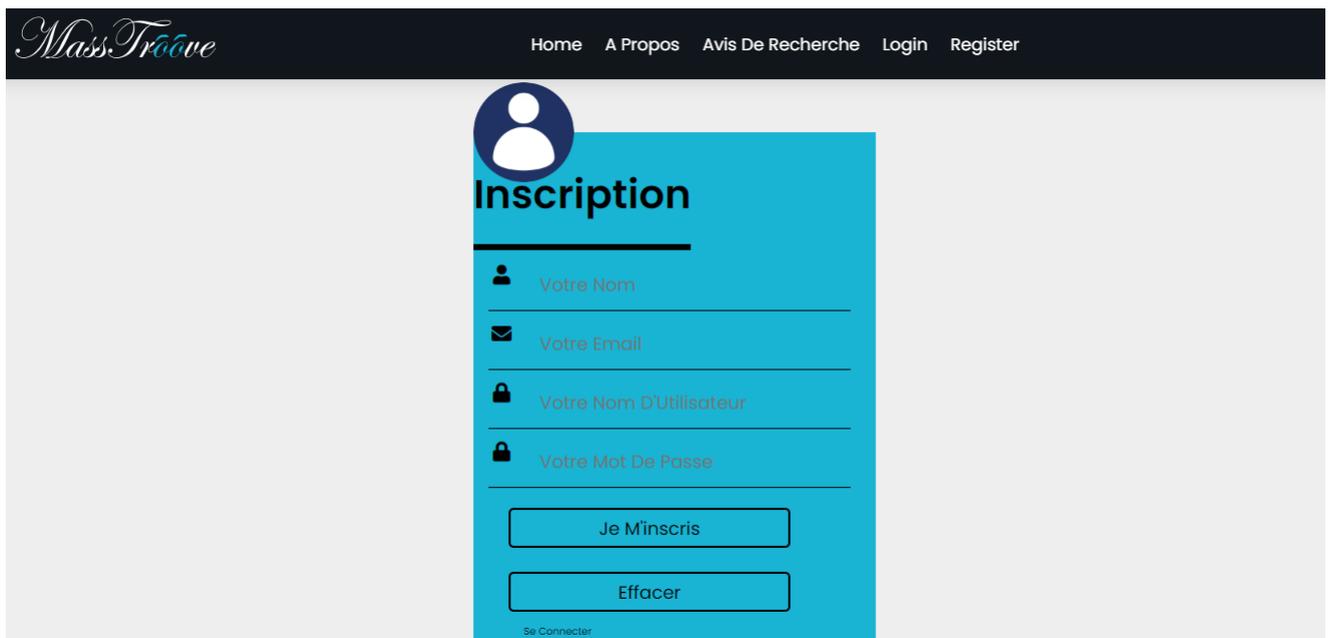


FIGURE 4.10 – Interface « Inscription » de notre application Web.

4.3.5 Interface « Authentification »

La figure 4.11 représente l'interface d'authentification où l'utilisateur saisi son e-mail et son mot de passe pour pouvoir accéder aux fonctionnalités du système.

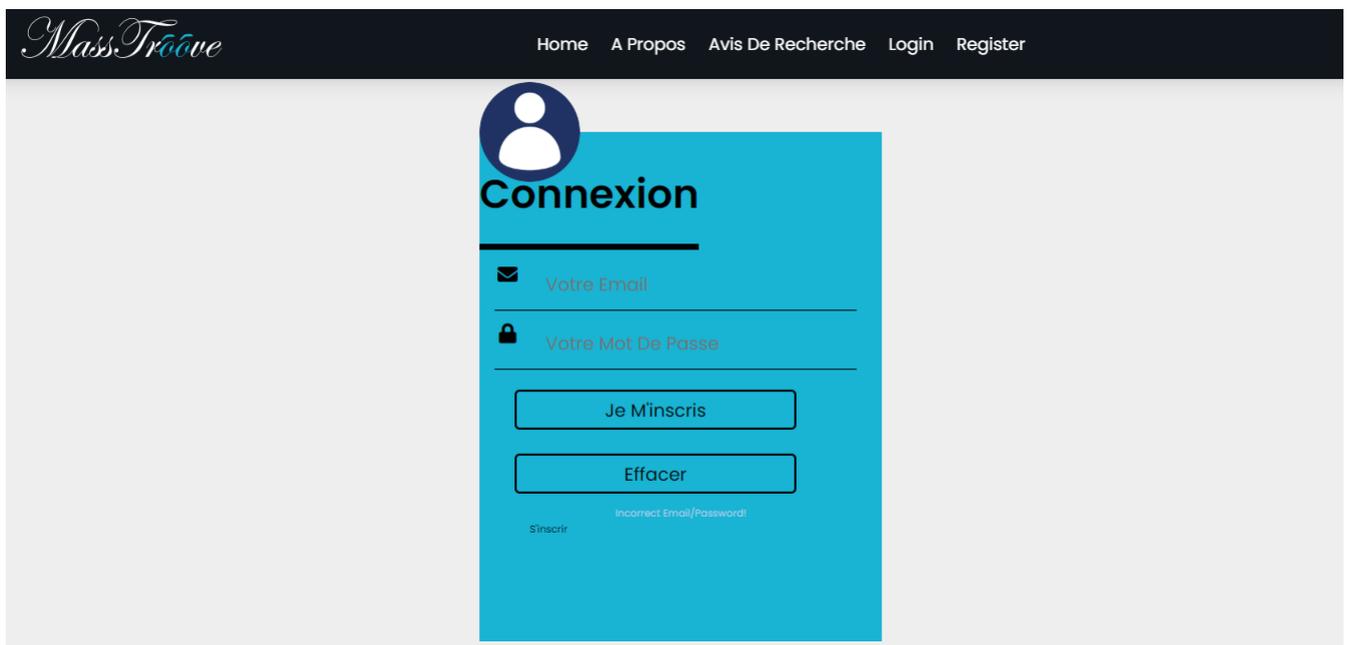


FIGURE 4.11 – Interface « Authentification » de notre application Web.

4.3.6 Interface « Compte utilisateur »

La figure 4.12 représente l'interface du compte où l'utilisateur va être dirigé une fois connecté et où il trouve plusieurs fonctionnalités qu'il peut exécuter.

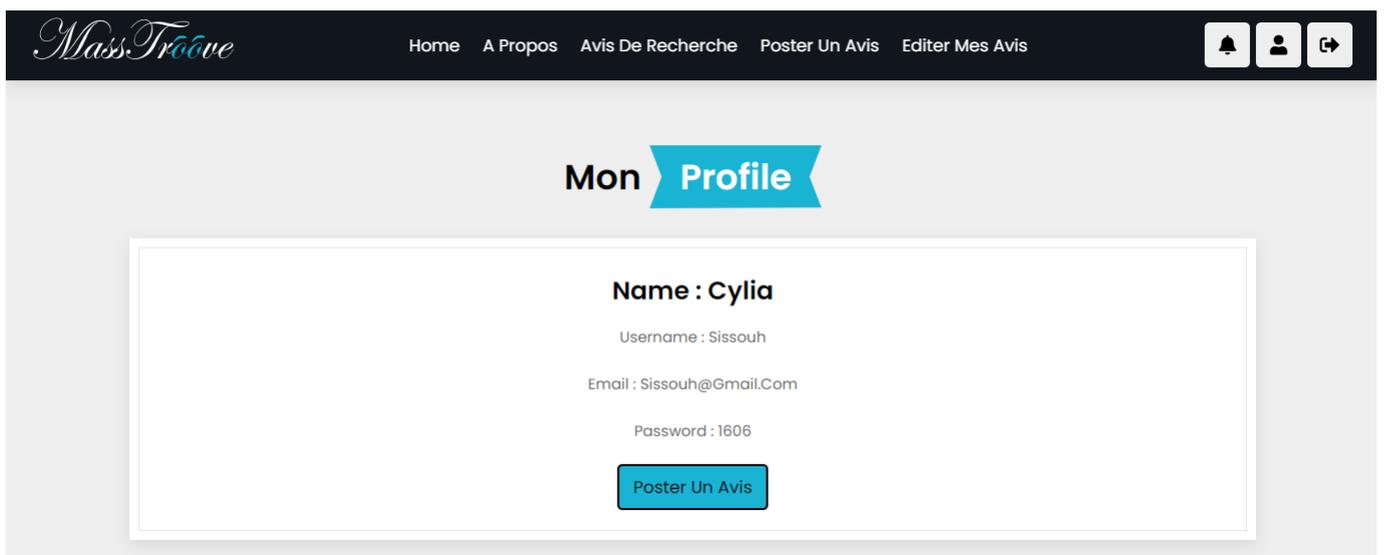


FIGURE 4.12 – Interface « Compte utilisateur » de notre application Web.

4.3.7 Interface « Poster un avis de recherche »

La figure 4.13 représente l'interface « Poster un avis de recherche », où l'utilisateur peut faire une annonce de recherche pour une personne perdue.

Mass Trêve Home A Propos Avis De Recherche Poster Un Avis Editer Mes Avis

Poster Un Avis De Recherche

Nom Prénom

DateNaissance Choisir un fichier Aucun Fichier Choisi

DateDisparition LieuDisparition

Homme Public

Postuler!

FIGURE 4.13 – Interface « Poster pour un avis de recherche » de notre application Web.

4.3.8 Interface « Éditer mes avis »

La figure 4.14 représente l'interface « Éditer mes avis », où l'utilisateur peut accéder à ses avis de recherche comme il peut les supprimer.

Mass Trêve Home A Propos Avis De Recherche Poster Un Avis Editer Mes Avis

Ma Liste Des Personnes Perdus

Donald Trump
Cette Personne Est Née Le " 12 Mars 1949 ".
Elle Est Perdue De Vue Le " 15 Aout 2017 " A :
" Usi "

Cristiano Ronaldo
Cette Personne Est Née Le " 20 Avril 1982 ".
Elle Est Perdue De Vue Le " 18 Mars 2022 " A
: " Italie "

Lamine Moumouh
Cette Personne Est Née Le " 13 Juillet 1998 ".
Elle Est Perdue De Vue Le " 05 Mars 2012 " A
: " Sidi Aich "

FIGURE 4.14 – Interface « Éditer mes avis » de notre application Web.

4.3.9 Interface « Résultats »

La figure 4.15 représente l'interface résultats, où l'utilisateur reçoit des pages de résultats lorsqu'on trouve la personne qu'il cherche.

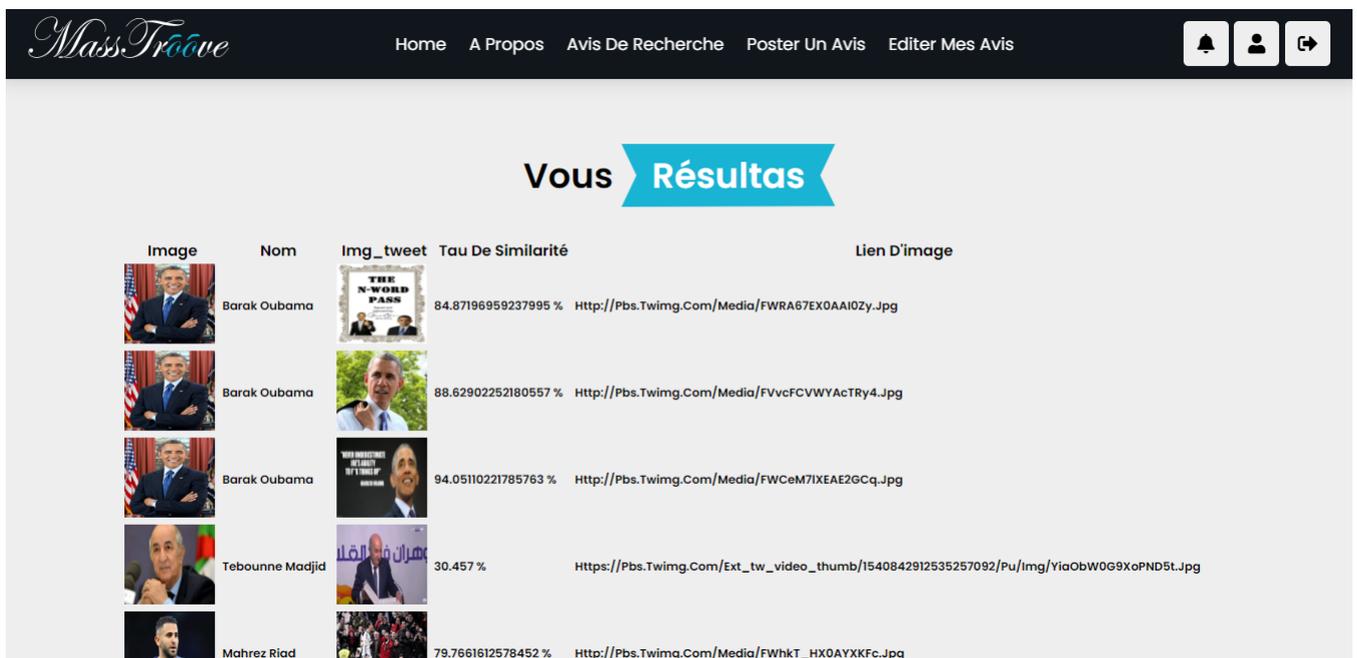


FIGURE 4.15 – Interface « Notification » de notre application Web.

4.3.10 Interface « Dashboard »

La figure 4.16 représente l'interface « Dashboard », où seulement l'administrateur peut accéder pour gérer l'application.

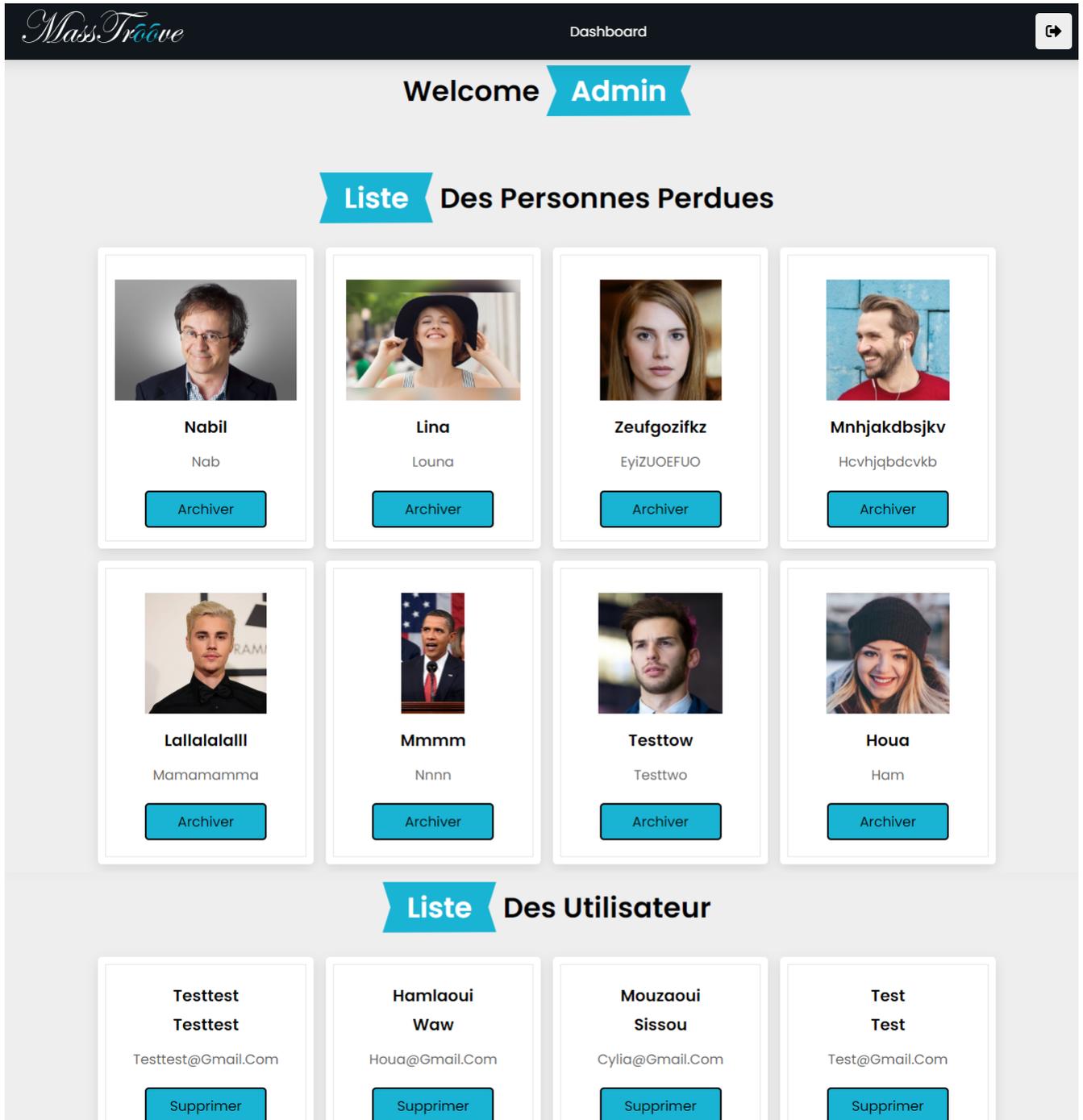


FIGURE 4.16 – Interface « Dashboard » de notre application Web.

4.4 Conclusion

Tout au long de ce chapitre nous avons expliqué les différentes technologies utilisées pour implémenter notre application, en suite on a parlé sur les langages et environnements qu'on a utilisé, puis on a défini les plus importants Frameworks qu'on a utilisé, en fin nous avons présenté les différentes interfaces de notre application d'où nous atteignons la fin de notre projet.

Conclusion Générale et perspectives

L'objectif de notre mémoire était de concevoir et d'implémenter une application de reconnaissance faciale capable de reconnaître les visages. Pour ce faire, nous avons remplis plusieurs tâches. La première tâche de l'application consistait à générer des photos et vidéos dans le réseaux sociaux Twitter grâce à une APIs Twitter et de les stocker dans un fichier JSON. La deuxième tâche était de poster une avis de recherche d'une personne perdue sur notre l'application. La troisième tâche consiste à vérifier si les visages des personnes perdues existants dans les photos et les vidéos publiées sur Twitter en utilisant la méthode de reconnaissance faciale.

Le résultat de notre application est l'image de personnes perdues, leurs URLs et le pourcentage de correspondance entre le visage de la personne perdue et le visage de la personne retrouvée sur Twitter.

Ce projet a été une expérience très riche pour nous, car il nous a permis de mettre en application nos connaissances informatiques et d'en acquérir de nouvelles, notamment dans le domaine de l'intelligence artificielle et du génie logiciel.

La solution que nous avons présentée dans ce mémoire répond au problème des personnes perdues en utilisant uniquement les APIs de twitter. Cette solution n'est en réalité qu'une ouverture vers d'autres travaux car notre système peut encore évoluer. Pour cela, plusieurs perspectives se présentent :

- Compléter les fonctionnalités de l'administrateur.
- Ajouter un espace pour un agent de police et ses fonctionnalités.
- Donner la main à l'administrateur pour ajouter des agents de police.
- Améliorer l'application en ajoutant d'autres méthodes de reconnaissance faciale.
- Appliquer la reconnaissance faciale sur des séquences vidéos.
- Étendre notre application pour faire de la reconnaissance faciale avec des APIs d'autres réseaux sociaux comme Facebook, Instagram, Youtube, ect.
- Élargir le champ d'application de notre proposition en intégrant une authentification biométrique.

Bibliographie

- [1] <https://www.techno-science.net/glossaire-definition/methode-de-viola-et-jones.html/>. (consulté le 14/06/2022).
- [2] <https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm/>. (consulté le 14/06/2022).
- [3] Zakarya Boukemoum. Détection des véhicules par histogramme orienté gradient. Université Guelma,Algerie.
- [4] [https://rassouldev.medium.com/introduction à la détection des visages/](https://rassouldev.medium.com/introduction-à-la-détection-des-visages/). (consulté le 7/06/2022).
- [5] Guerroudj Bencherki and Brahmin Moustafa. Implémentation d un système de reconnaissance de visages à base de pca. 2018.
- [6] Maëlig Jacquet and Lionel Grossrieder. Enjeux et perspectives de la reconnaissance faciale en sciences criminelles. *Criminologie*, 54(1) :135–170, 2021.
- [7] Philippe Cudré-Mauroux, Philippe Gilliéron, Jean Lassègue, Patrik Manzoni, René Bühler, Daniel Fink, Jens Piesbergen, Manon Jendly, Jörg Arnold, and Monika Simmler. Inhaltsverzeichnis/table des matières.
- [8] Nirmalya Kar, Mrinal Kanti Debbarma, Ashim Saha, and Dwijen Ru-dra Pal. Study of implementing automated attendance system using face recognition technique. *International Journal of computer and communication engineering*, 1(2) :100, 2012.
- [9] Agnité Maxim Wilfrid Straïker EDOH and Marc Kokou Assogba. Reconnaissance de visages de personnes de peau noire dans les images numériques par la méthode des eigenfaces. 2016.
- [10] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE computer society conference on computer vision and pat-tern recognition*, pages 586–587. IEEE Computer Society, 1991.
- [11] Mehul K Dabhi and Bhavna K Pancholi. Face detection system based on viola-jones algorithm. *International Journal of Science and Research (IJSR)*, 5(4) :62–64, 2016.
- [12] NADJETTE ASSADI. Mise au point d’une application de reconnaissance faciale. Université Mohamed Khider BISKRA ,Algerie.
- [13] Labandji Bisma Baghdadi Zineb. Système de vidéosurveillance avec la reconnaissance faciale. 2017/2018.
- [14] Timothy Griffin and Monica K Miller. Child abduction, amber alert, and crime control theater. *Criminal justice review*, 33(2) :159–176, 2008.
- [15] <https://lookingforanangel.org/>. (consulter le 14 juin 2022).
- [16] <https://www.wehelpthemissing.org/>. (consulter le 15 juin 2022).
- [17] https://fr.m.wikipedia.org/wiki/croissant-rouge_alg/. (consulter le 15 juin 2022).
- [18] BENZAADA Assia ARAB Majda. Conception et réalisation d’un site web pour le centre d’imagerie médicale du chu khelil amrane bejaia. Université Béjaia ,Algerie.

- [19] <https://blog.ippon.fr/2020/10/16/diagramme-de-contexte-vs-hexagone/>. (consulter le 07 juillet 2022).
- [20] LOUBAR Nacera AIROUCHE Cylia. Conception et réalisation d'un système d'information d'aide à la décision pour les prédictions de ventes (cas : Sarl ifri). Université Béjaia, Algérie.
- [21] MOKHTARI Nazim AIT BRAHAM Mouloud. Conception et réalisation d'un portail web pour parc de loisirs - cas parc de loisirs de Béjaia - Alivava. Université Béjaia, Algérie.
- [22] <https://mobapi.com/fr/apis-introduction-et-contexte/>. (consulter le 13 mai 2022).
- [23] <https://www.definitions-marketing.com/definition/api-twitter/> : :text=l'api. (consulter le 24 juin 2022).
- [24] https://developer.mozilla.org/fr/docs/learn/javascript/client-side_web_apis/introduction. (consulter le 15 mai 2022).
- [25] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203255-html-hypertext-markup-langage-definition-traduction/>. (consulter le 11 juin 2022).
- [26] https://developer.mozilla.org/fr/docs/learn/javascript/first_steps/what_is_javascript. (consulter le 11 juin 2022).
- [27] <https://www.futura-sciences.com/tech/definitions/internet-css/>. (consulter le 11 juin 2022).
- [28] <https://www.mysql.com/fr/>. (consulter le 24 juin 2022).
- [29] <https://www.clicours.com/langages-interpretes-ou-compiles-cours-visual-basic/>. (consulter le 11 juin 2022).
- [30] [https://fr.wikipedia.org/wiki/spyder_\(logiciel\)](https://fr.wikipedia.org/wiki/spyder_(logiciel)). (consulter le 29 juin 2022).
- [31] <https://www.techno-science.net/glossaire-definition/framework.html>. (consulter le 11 juin 2022).
- [32] <https://www.techtarget.com/whatis/definition/bootstrap>. (consulter le 11 juin 2022).
- [33] <https://flask.palletsprojects.com/en/1.1.x/foreword/>. (consulter le 24 juin 2022).
- [34] <https://pypi.org/search/?c=framework>. (consulter le 24 juin 2022).
- [35] <https://www.exemplaire.com.ulaval.ca/>. Rechercher une personne disparue, options et tapes (consulter le 03/06/2022).
- [36] <https://fr.acervolima.com/introduction-aux-api/>. (consulter le 12 mai 2022).
- [37] <https://www.wampserver.com/>. (consulter le 11 juin 2022).
- [38] <https://www.encyclopedie.fr/definition/phpmyadmin/>. (consulter le 11 juin 2022).
- [39] <https://www.commentcamarche.net/telecharger/bureautique/23283-visual-paradigm/>. (consulter le 11 juin 2022).
- [40] <https://www.techno-science.net/glossaire-definition/adobe-photoshop.html/>. (consulter le 11 juin 2022).
- [41] <https://code.visualstudio.com/>. (consulter le 11 juin 2022).

Résumé

Ce travail s'intéresse au problème général de la reconnaissance de visages en contexte de la recherche des personnes perdus. L'objectif est de concevoir et d'implémenter une application permettant la reconnaissance des visages des personnes à partir des photos publiées sur le réseau social Twitter. La 1ère tâche de l'application consiste à générer des photos dans le réseau social Twitter grâce à des APIs Twitter et de les stocker dans un fichier JSON. La 2ème tâche, est de poster un avis de recherche d'une personne perdue sur notre application. La 3ème tâche consiste à vérifier si les visages des personnes perdus existent dans les photos publiées sur Twitter en utilisant la méthode la reconnaissance faciale. Notre application Web retourne l'image de la personne perdues de la base de données, L'URL d'une image lui correspondant sur Twitter et le pourcentage de correspondance entre les deux images.

Mots-clés : Reconnaissance faciale, API Twitter, Python, Application Web.

Abstract

Our work focuses on the general problem of face recognition in the context of the search for lost people. Thus, our objective is to design and implement an application allowing the recognition of people's faces from photos published on the social networks Twitter. The 1st task of our Web application is to download photos from Twitter thanks to the Twitter API, and store them in a JSON file. The 2nd task is to post a notice of search for a lost person on our Web application. The 3rd task is to check if the faces of the lost people in our system exist in the photos posted on Twitter using the facial recognition method. Our application returns the image of the lost person from our database, the url of a photo of the lost person from Twitter and the percentage of correspondence between the two images.

keywords : Face recognition, Twitter API, Python, Web Application