



# Hina Eye: Recherche de Personne Disparue en appliquant les techniques de Traitement d'Images

Mathématique du signal et traitement de signal : Image

Etudiant :

**François-Xavier Kobon,**  
Master 1 Génie Logiciel (spé.  
Bigdata et IA)

Enseignant

**Dr. Ghislain Pandry,**  
Chercheur Traitement Signal et  
Image

Septembre, 2024

# Table des matières

|  |          |
|--|----------|
| <b>Introduction</b>  | <b>4</b> |
| Contexte et Problématique . . . . .                                | 4        |
| Enjeux Sociaux . . . . .   | 4        |
| Enjeux Économiques . . . . .                                       | 4        |
| Enjeux Problématique . . . . .                                     | 5        |
| Objectif du projet . . . . .                                       | 5        |
| Plan rapport du projet . . . . .                                   | 6        |
| Fondements Théoriques du Traitement de Signal et d'Images .        | 6        |
| Techniques Utilisées pour la Recherche de Personnes Disparues      | 6        |
| Méthodologie de l'Étude . . . . .                                  | 7        |
| Résultats et Analyses . . . . .                                    | 7        |
| Conclusion et Perspectives . . . . .                               | 7        |
| <b>1 Fondements Théoriques du Traitement de Signal et d'Images</b> | <b>8</b> |
| 1.1 Concepts de base du traitement de signal . . . . .             | 8        |
| 1.1.1 Transformée de Fourier et filtrage fréquentiel . . . . .     | 8        |
| 1.1.2 Transformée de Fourier et filtrage fréquentiel . . . . .     | 9        |
| 1.1.3 Convolution et filtrage spatial . . . . .                    | 9        |
| 1.2 Traitement d'images numériques . . . . .                       | 9        |
| 1.2.1 Acquisition et représentation des images . . . . .           | 9        |
| 1.2.2 Prétraitement des images . . . . .                           | 10       |
| 1.2.3 Filtrage spatial . . . . .                                   | 10       |
| 1.2.4 Normalisation de l'éclairage . . . . .                       | 10       |
| 1.2.5 Redimensionnement et conversion en niveaux de gris .         | 10       |
| 1.2.6 Segmentation et détection des contours . . . . .             | 10       |
| 1.3 Applications du traitement d'images à la reconnaissance de     |          |
| personnes . . . . .  | 11       |
| 1.3.1 Détection des visages . . . . .                              | 11       |
| 1.3.2 Extraction des caractéristiques faciales . . . . .           | 11       |
| 1.3.3 Reconnaissance faciale . . . . .                             | 11       |
| 1.4 Conclusion . . . . .   | 12       |

|          |  |           |
|----------|--|-----------|
| <b>2</b> | <b>Techniques Utilisées pour la Recherche de Personnes Disparues</b>   | <b>13</b> |
| 2.1      | Prétraitement des images . . . . .   | 13        |
| 2.1.1    | Techniques de filtrage et de réduction de bruit . . . . .  | 13        |
| 2.1.2    | Ajustement du contraste et de la luminosité . . . . .  | 13        |
| 2.2      | Détection des visages et extraction des caractéristiques . . . . .   | 14        |
| 2.2.1    | Utilisation de <code>face_recognition</code> pour la détection des visages . . . . .                                 | 14        |
| 2.2.2    | Encodage des visages . . . . .   | 15        |
| 2.3      | Reconnaissance faciale . . . . .   | 15        |
| 2.3.1    | Modèles utilisés (HOG et CNN) . . . . .  | 15        |
| 2.3.2    | Validation des résultats et scores obtenus . . . . .   | 15        |
| 2.4      | Gestion des fichiers et organisation des données . . . . .   | 16        |
| 2.5      | Conclusion . . . . .   | 16        |
| <b>3</b> | <b>Méthodologie de l'Étude</b>   | <b>17</b> |
| 3.1      | Choix des outils et algorithmes . . . . .  | 17        |
| 3.1.1    | Justification des outils choisis . . . . .   | 17        |
| 3.1.2    | <code>Face_recognition</code> . . . . .  | 17        |
| 3.1.3    | Pillow (PIL) . . . . .   | 18        |
| 3.1.4    | OpenCV . . . . .   | 18        |
| 3.1.5    | NumPy . . . . .  | 18        |
| 3.1.6    | Flask et Flask-CORS . . . . .  | 18        |
| 3.2      | Comparaison des performances attendues des différentes techniques . . . . .  | 18        |
| 3.3      | Description des données . . . . .  | 19        |
| 3.3.1    | Type de données utilisées . . . . .  | 19        |
| 3.3.2    | Source des données . . . . .   | 19        |
| 3.4      | Mise en œuvre des techniques . . . . .   | 19        |
| 3.4.1    | Explication des étapes de la mise en œuvre des techniques choisies . . . . .   | 20        |
| 3.4.2    | Détails sur les algorithmes appliqués (prétraitement, extraction de caractéristiques, reconnaissance. . .) . . . . . | 20        |
| 3.5      | Conclusion . . . . .   | 21        |
| <b>4</b> | <b>Résultats et Analyses</b>   | <b>22</b> |
| 4.1      | Résultats obtenus . . . . .  | 22        |
| 4.1.1    | Jeu de données personnalisé (photos d'enfants) . . . . .   | 22        |
| 4.1.2    | Dataset Kaggle (7 Celebrity Images) . . . . .  | 22        |
| 4.2      | Analyse des performances . . . . .   | 23        |
| 4.2.1    | Précision de la reconnaissance faciale . . . . .   | 23        |

|          |  |           |
|----------|--|-----------|
| 4.2.2    | Temps de traitement . . . . .  | 23        |
| 4.3      | Limites et contraintes . . . . .                                       | 24        |
| 4.3.1    | Taille du dataset . . . . .  | 24        |
| 4.3.2    | Qualité des images . . . . .   | 24        |
| 4.3.3    | Contraintes matérielles . . . . .                                      | 24        |
| 4.4      | Améliorations et perspectives . . . . .                                | 24        |
| 4.4.1    | Augmentation de la taille des données . . . . .                        | 24        |
| 4.4.2    | Optimisation du temps de traitement . . . . .                          | 25        |
| 4.4.3    | Développement d'une interface utilisateur . . . . .                    | 25        |
| 4.5      | Conclusion . . . . .   | 25        |
| <b>5</b> | <b>Conclusion et Perspectives</b>                                      | <b>26</b> |
| 5.1      | Résumé des résultats . . . . .   | 26        |
| 5.2      | Conclusion sur les performances des techniques utilisées . . . . .     | 27        |
| 5.3      | Limites de l'étude . . . . .   | 27        |
| 5.4      | Perspectives futures . . . . .   | 28        |
| 5.4.1    | Augmentation et diversification du dataset . . . . .                   | 28        |
| 5.4.2    | Intégration de la reconnaissance à partir de séquences vidéo . . . . . | 28        |
| 5.4.3    | Optimisation des algorithmes et des ressources matérielles . . . . .   | 28        |
| 5.4.4    | Développement d'une plateforme intégrée . . . . .                      | 29        |
| 5.4.5    | Recherche collaborative et partenariats . . . . .                      | 29        |
| <b>A</b> | <b>Exemples</b>  | <b>30</b> |
| A.1      | Code source . . . . .  | 30        |
| A.2      | Test après entraînement . . . . .                                      | 33        |

# Introduction

## Contexte et Problématique

La disparition d'une personne, qu'elle soit volontaire ou involontaire, représente une source de détresse pour les familles, les communautés et les autorités concernées. Chaque année, des millions de personnes disparaissent à travers le monde, que ce soit en raison d'enlèvements, de catastrophes naturelles, de conflits, ou de situations accidentelles. En Côte d'Ivoire, tout comme dans d'autres pays, ce phénomène constitue un véritable enjeu de sécurité publique. Le processus de recherche d'une personne disparue mobilise d'importantes ressources humaines et financières et nécessite souvent l'intervention de multiples institutions, telles que les forces de l'ordre, les organismes humanitaires, et les services de santé.

## Enjeux Sociaux

D'un point de vue social, la disparition d'une personne crée un vide au sein des familles et des communautés. Les proches des personnes disparues sont confrontés à l'incertitude, à l'angoisse, et à la douleur émotionnelle, ce qui peut affecter la cohésion sociale et la stabilité des foyers. En outre, les disparitions ont un impact direct sur la sécurité de la population, car elles sont souvent liées à des situations criminelles (trafic d'êtres humains, enlèvements, violences) ou à des événements tragiques (accidents, catastrophes). Le nombre croissant de personnes disparues dans certains contextes, comme les zones de conflit ou les migrations forcées, met en évidence le besoin urgent de renforcer les capacités de recherche et de gestion des crises.

## Enjeux Économiques

Sur le plan économique, les efforts de recherche de personnes disparues peuvent engendrer des coûts considérables. Ces coûts incluent non seulement le déploiement de forces de sécurité et de secours, mais aussi l'utilisation de

technologies coûteuses, comme les systèmes de surveillance, les drones, ou encore les logiciels de reconnaissance faciale. Les pertes de productivité associées aux disparitions, qu'elles soient liées à des employés ou à des membres actifs de la communauté, représentent également un fardeau économique. En effet, l'absence d'une personne active dans le milieu professionnel ou familial peut entraîner des perturbations dans les chaînes de production ou dans les dynamiques économiques des communautés locales.

## Problématique

La recherche de personnes disparues représente un défi majeur pour les autorités et les familles, car les méthodes traditionnelles sont souvent lentes, coûteuses et peu efficaces. L'application des techniques de traitement d'images et des mathématiques du signal offre une opportunité d'améliorer la rapidité et la précision de ces recherches. Le problème consiste donc à déterminer comment utiliser ces technologies pour localiser et identifier efficacement les personnes disparues, tout en respectant les contraintes économiques, sociales et éthiques liées à leur mise en œuvre.

## Objectif du projet

L'objectif principal de ce projet est d'explorer et de mettre en œuvre des techniques de traitement d'images pour la recherche de personnes disparues. Dans un contexte où la disparition de personnes représente un enjeu majeur tant pour les autorités que pour les familles, l'utilisation des technologies avancées s'avère cruciale pour optimiser les processus de recherche. Le traitement d'images, en particulier, est devenu un outil incontournable pour extraire des informations pertinentes à partir de données visuelles, qu'il s'agisse de photographies, de vidéos de surveillance, ou de flux vidéo en temps réel.

Notre travail vise à démontrer comment les techniques mathématiques du traitement de signal, appliquées aux images, permettent d'améliorer la reconnaissance, la localisation et le suivi de personnes à partir de données visuelles. Plus spécifiquement, il s'agira de :

- **Préparer les images** pour en faciliter l'analyse, en utilisant des méthodes de filtrage, de réduction de bruit et d'ajustement de contraste.
- **Extraire des caractéristiques pertinentes** des images, telles que des contours, des points d'intérêt ou des traits faciaux distinctifs, afin d'améliorer la détection des individus recherchés.

- **Mettre en œuvre des algorithmes de reconnaissance faciale** pour identifier ou vérifier l'identité des personnes disparues à partir d'une base de données d'images existantes.
- **Appliquer des techniques de suivi d'objets dans des séquences vidéo**, permettant ainsi de suivre une personne à travers plusieurs cadres visuels, en particulier dans le contexte de la vidéosurveillance.
- **Comparer les performances des différentes méthodes** de traitement d'images en termes de précision, de rapidité et de robustesse, afin de proposer des solutions efficaces et adaptées aux contraintes du terrain.

À travers ces objectifs, ce projet propose une approche intégrée où les mathématiques du signal et les algorithmes de traitement d'images convergent pour répondre à une problématique pratique et complexe : celle de la recherche de personnes disparues. L'analyse des résultats obtenus permettra de mesurer l'efficacité des différentes approches et de proposer des améliorations futures dans l'utilisation de ces technologies dans le domaine de la sécurité publique, des enquêtes criminelles, et des opérations de secours en cas de catastrophe.

## Plan rapport du projet

Ce document s'articule autour de cinq chapitres, chacun contribuant à l'élaboration de la problématique principale de la recherche de personnes disparues à l'aide des techniques de traitement d'images.

### Fondements Théoriques du Traitement de Signal et d'Images

Ce premier chapitre introduit les concepts clés qui sous-tendent les techniques de traitement de signal et d'images. Nous y détaillerons les bases mathématiques du traitement de signal, telles que la transformée de Fourier et les méthodes de filtrage, avant de passer aux principes du traitement d'images, notamment la segmentation, la détection de contours et les méthodes de reconnaissance.

### Techniques Utilisées pour la Recherche de Personnes Disparues

Le deuxième chapitre présente les différentes techniques spécifiques que nous utiliserons pour résoudre le problème de recherche de personnes disparues. Nous explorerons les algorithmes de prétraitement d'image, la détection

de caractéristiques, la reconnaissance faciale et le suivi d'objets dans des séquences vidéo, tout en justifiant le choix de chaque méthode pour l'application à notre sujet d'étude.

## **Méthodologie de l'Étude**

Dans ce chapitre, nous décrirons la méthodologie suivie pour la réalisation de ce projet. Nous aborderons le choix des outils, des algorithmes ainsi que des données utilisées, puis nous expliquerons les étapes d'implémentation des différentes techniques retenues pour la recherche de personnes disparues à partir d'images fixes et de vidéos.

## **Résultats et Analyses**

Ce quatrième chapitre présentera les résultats obtenus après l'application des techniques de traitement d'images sur les données recueillies. Nous y analyserons les performances des algorithmes sélectionnés en termes de précision, de robustesse et de rapidité. Une comparaison des méthodes sera également effectuée pour évaluer les avantages et limites de chaque approche.

## **Conclusion et Perspectives**

Enfin, le dernier chapitre résume les résultats principaux et évalue l'efficacité des techniques utilisées pour la recherche de personnes disparues. Il ouvre également la discussion sur les pistes d'amélioration possibles et propose des perspectives d'évolution du projet, tant sur le plan technique que sur les applications potentielles dans des contextes réels.



# Chapitre 1

## Fondements Théoriques du Traitement de Signal et d'Images

### 1.1 Concepts de base du traitement de signal

Le traitement de signal est une discipline qui consiste à analyser, modifier et synthétiser des signaux, afin d'extraire des informations utiles ou de les transformer. Les signaux peuvent être de nature variée : audio, vidéo, ou encore des images numériques. Dans le cadre de ce projet, nous nous intéressons spécifiquement aux **signaux d'images**. Voici quelques concepts fondamentaux du traitement de signal qui sont pertinents pour l'analyse des images.

#### 1.1.1 Transformée de Fourier et filtrage fréquentiel

Un signal peut être représenté dans le **domaine temporel**, où chaque point représente une valeur (ou intensité) en fonction du temps, ou dans le **domaine fréquentiel**, où l'on considère les différentes fréquences qui composent le signal. Pour passer du domaine temporel au domaine fréquentiel, on utilise la **transformée de Fourier**. Cette technique est couramment utilisée dans le traitement des images pour analyser les variations d'intensité lumineuse et pour détecter les motifs récurrents dans une image.

Mathématiquement, la transformée de Fourier discrète d'une image  $I(x, y)$  est donnée par :

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} K(x, y) e^{-2\pi i \left( \frac{ux}{M} + \frac{vy}{N} \right)}$$

Où  $M$  et  $N$  sont les dimensions de l'image, et  $u$  et  $v$  représentent les fréquences spatiales.

La **transformée de Fourier inverse (TFI)** permet de revenir du domaine fréquentiel au domaine spatial, ce qui est essentiel pour certaines techniques de filtrage et de compression d'images.

### 1.1.2 Transformée de Fourier et filtrage fréquentiel

La **transformée de Fourier** est une opération mathématique qui décompose un signal en une somme de sinusoides de différentes fréquences. Pour le traitement d'images, elle permet d'identifier et de supprimer les fréquences indésirables, telles que le bruit. Le **filtrage fréquentiel** est une technique utilisée pour supprimer ou atténuer certaines fréquences dans une image, permettant ainsi de réduire le bruit ou de faire ressortir certains détails.

### 1.1.3 Convolution et filtrage spatial

En traitement d'images, la **convolution** est une opération qui combine deux signaux pour produire un signal de sortie. Elle est utilisée pour appliquer des filtres sur une image, par exemple pour flouter l'image ou pour détecter les contours. Le **filtrage spatial** est une technique qui consiste à appliquer un filtre (ou noyau) à une région locale de l'image pour modifier sa luminosité, son contraste, ou pour faire ressortir certains motifs. Les filtres couramment utilisés incluent les filtres de lissage (pour réduire le bruit) et les filtres de détection de contours (pour détecter les transitions de luminosité). La convolution  $2D$  pour une image  $I$  avec un filtre  $K$  est exprimée par :

$$S(x, y) = \sum_{i=-a}^b \sum_{j=-b}^b I(x+i, y+j) K(i, j)$$

où  $S(x, y)$  est l'image résultante, et  $K(i, j)$  est le filtre appliqué, avec  $a$  et  $b$  étant les dimensions du noyau.

## 1.2 Traitement d'images numériques

### 1.2.1 Acquisition et représentation des images

Le traitement d'images numériques est une sous-discipline du traitement de signal qui se concentre spécifiquement sur la manipulation et l'analyse des images numériques. Le processus de traitement d'images implique plusieurs

étapes clés qui permettent d'améliorer la qualité des images et d'extraire des informations utiles pour des tâches telles que la reconnaissance faciale.

### 1.2.2 Prétraitement des images

Le **prétraitement** est une étape essentielle dans le traitement d'images. Il consiste à améliorer la qualité des images avant de les soumettre aux algorithmes d'analyse.

### 1.2.3 Filtrage spatial

Utilisé pour éliminer le bruit dans les images. Des filtres comme le filtre gaussien ou le filtre médian sont souvent appliqués pour lisser l'image tout en conservant les détails importants.

### 1.2.4 Normalisation de l'éclairage

L'éclairage peut varier d'une image à l'autre, ce qui peut compliquer la reconnaissance faciale. Des techniques comme la **limitation de contraste adaptative à l'histogramme (CLAHE)** permettent de corriger ces variations.

### 1.2.5 Redimensionnement et conversion en niveaux de gris

Les images sont souvent redimensionnées et converties en niveaux de gris pour réduire la quantité de données à traiter sans perte d'informations pertinentes pour la reconnaissance faciale.

### 1.2.6 Segmentation et détection des contours

La **segmentation** consiste à diviser une image en plusieurs régions significatives, souvent en fonction de la variation des intensités de pixels. Cette technique permet d'isoler les objets ou les visages dans une image. La **détection de contours**, quant à elle, est utilisée pour localiser les transitions abruptes d'intensité dans une image, et donc identifier les bords des objets. Des algorithmes comme celui de **Canny** sont souvent utilisés pour détecter les contours des visages dans les systèmes de reconnaissance faciale.

## 1.3 Applications du traitement d'images à la reconnaissance de personnes

Le traitement d'images joue un rôle clé dans la reconnaissance faciale, un domaine qui s'appuie sur des techniques avancées pour détecter, identifier et suivre des visages dans des images. La reconnaissance de personnes à partir d'images peut être divisée en plusieurs étapes.

### 1.3.1 Détection des visages

La première étape consiste à détecter les visages dans une image. Les algorithmes de détection de visages, comme les réseaux neuronaux convolutifs (CNN), sont capables de localiser les visages avec une grande précision, même dans des environnements complexes. Les modèles basés sur **HOG (Histograms of Oriented Gradients)** sont aussi couramment utilisés pour la détection des visages en raison de leur rapidité.

### 1.3.2 Extraction des caractéristiques faciales

Une fois les visages détectés, il est nécessaire d'extraire des caractéristiques uniques qui permettent d'identifier une personne. Ces caractéristiques peuvent inclure la distance entre les yeux, la forme du nez, ou encore la structure des pommettes.

### 1.3.3 Reconnaissance faciale

La dernière étape est la reconnaissance faciale proprement dite, où les caractéristiques extraites sont comparées à celles d'une base de données existante pour identifier la personne. Les algorithmes comme les réseaux neuronaux convolutifs (CNN) ou les techniques de réduction de dimensionnalité comme la PCA (Principal Component Analysis) permettent d'associer des visages à des identités, même dans des images de faible qualité.

## 1.4 Conclusion

Ce chapitre a présenté les fondements théoriques du traitement de signal et du traitement d'images, qui sont essentiels à la reconnaissance faciale. En combinant des techniques comme la détection des visages, l'extraction des caractéristiques et la reconnaissance, il est possible d'appliquer ces concepts à des problèmes concrets comme la recherche de personnes disparues. Les avancées technologiques dans ces domaines offrent des solutions de plus en plus précises et efficaces, capables de traiter de grandes quantités de données visuelles en temps réel. Ces concepts seront appliqués dans les chapitres suivants pour la mise en œuvre du projet « Hina Eye ».

## Chapitre 2

# Techniques Utilisées pour la Recherche de Personnes Disparues

### 2.1 Prétraitement des images

Le prétraitement est une étape clé dans le projet, visant à améliorer la qualité des images avant leur analyse par les algorithmes de reconnaissance faciale. Le code implémente plusieurs étapes de prétraitement, dont le redimensionnement, la conversion en niveaux de gris, la réduction du bruit et la normalisation de l'éclairage.

#### 2.1.1 Techniques de filtrage et de réduction de bruit

La méthode `fastNlMeansDenoising` d'OpenCV a été utilisée pour réduire le bruit présent dans les images. Le bruit peut résulter de variations dans les conditions de prise de vue ou de la qualité des caméras utilisées. Cette méthode permet de lisser les images tout en conservant les détails essentiels, notamment les contours des visages.

- **Filtre de réduction du bruit** : Cette technique supprime les artefacts visuels tout en préservant les contours des objets et des visages dans l'image. Cela est essentiel pour améliorer la qualité des encodages de visages dans les étapes suivantes.

#### 2.1.2 Ajustement du contraste et de la luminosité

Le projet utilise l'algorithme CLAHE (Contrast Limited Adaptive Histogram Equalization) pour normaliser l'éclairage des images et ajuster le

contraste. CLAHE permet d'améliorer les détails des images dans des conditions d'éclairage variables, telles que des ombres ou des surexpositions, qui pourraient fausser la détection des visages.

- **Normalisation de l'éclairage** : CLAHE corrige les variations d'éclairage en ajustant la distribution des intensités lumineuses, rendant les visages plus visibles dans les images sombres ou à fort contraste.
- **Amélioration de la netteté** : Un processus supplémentaire d'amélioration de la netteté a été appliqué à l'aide de la bibliothèque Pillow pour rendre les contours du visage plus nets après le prétraitement.

Ces étapes de prétraitement sont essentielles pour garantir que les images traitées soient de bonne qualité et prêtes pour l'encodage des visages.

## 2.2 Détection des visages et extraction des caractéristiques

Dans le projet *Hina Eye*, la bibliothèque `face_recognition` a été utilisée pour détecter et extraire les encodages des visages dans les images prétraitées. Cette étape consiste à localiser les visages dans les images puis à générer des encodages qui permettent de reconnaître les visages.

### 2.2.1 Utilisation de `face_recognition` pour la détection des visages

La bibliothèque `face_recognition` permet de détecter les visages dans des images en utilisant deux modèles : **HOG (Histograms of Oriented Gradients)** et **CNN (Convolutional Neural Networks)**. Le projet utilise principalement le modèle HOG, qui fonctionne rapidement sur les processeurs (CPU).

- **Modèle HOG** : HOG est un modèle performant pour la détection des visages lorsque l'accès à des GPU est limité. Il permet de détecter les visages en analysant les gradients d'intensité lumineuse dans une image. Le modèle HOG a été choisi pour sa rapidité et son efficacité sur des systèmes à ressources limitées.
- **Modèle CNN** : Le modèle CNN, bien qu'il soit plus précis, est utilisé de manière optionnelle dans le projet lorsqu'un GPU est disponible. CNN est plus lent mais offre une meilleure performance, particulièrement pour les grandes bases de données d'images.

## 2.2.2 Encodage des visages

Après la détection, les visages sont encodés en utilisant la fonction `face_encodings` de la bibliothèque `face_recognition`. Ces encodages sont des représentations numériques des caractéristiques du visage (comme la distance entre les yeux, la forme du nez, etc.) qui permettent de comparer des visages entre eux.

- **Encodage des visages** : Chaque visage détecté dans une image est converti en un vecteur de caractéristiques numériques, qui est ensuite stocké dans une base de données pour être comparé aux autres visages dans les étapes de reconnaissance.

Les encodages permettent ensuite de comparer des visages inconnus avec des visages déjà encodés pour identifier des correspondances.

## 2.3 Reconnaissance faciale

La reconnaissance faciale repose sur la comparaison des encodages des visages extraits d'une nouvelle image avec ceux stockés dans la base de données des visages connus. Le projet utilise la méthode `compare_faces` de la bibliothèque `face_recognition` pour cette tâche.

### 2.3.1 Modèles utilisés (HOG et CNN)

Le projet *Hina Eye* permet de choisir entre deux modèles pour la reconnaissance faciale.

- **HOG** : Ce modèle est utilisé par défaut en raison de sa rapidité et de sa faible demande en ressources matérielles. Il permet de détecter et d'encoder des visages en se basant sur les gradients d'intensité lumineuse dans l'image.
- **CNN** : Ce modèle est utilisé dans les cas où un GPU est disponible. Bien que plus lent, le modèle CNN offre une précision plus élevée et est capable de mieux gérer les variations d'éclairage, les rotations ou les obstructions partielles du visage.

### 2.3.2 Validation des résultats et scores obtenus

Le projet a utilisé deux jeux de données pour valider et évaluer la performance des modèles.

- **Dataset personnalisé** : Avec un score d'entraînement de 60%, ce jeu de données limité a permis de tester le modèle sur un petit ensemble de photos.



- ● 7 Celebrity Images (Kaggle) : Ce dataset a permis d'entraîner le modèle sur un ensemble plus grand et diversifié, atteignant une précision de 90,61% avec le modèle HOG.

La précision des résultats est largement influencée par la taille du dataset et la qualité des images. Une base de données plus grande, comme celle de Kaggle, permet d'obtenir de meilleurs résultats en termes de reconnaissance faciale.

## 2.4 Gestion des fichiers et organisation des données

Le projet intègre également des fonctions pour organiser et gérer les fichiers d'images et d'encodages. Le prétraitement des images est effectué dans un dossier temporaire (**processed**), et les encodages des visages sont sauvegardés sous forme de fichiers pickle.

- **Suppression automatique des fichiers** : Avant chaque nouvel entraînement, le contenu du dossier **processed** est automatiquement supprimé pour éviter les conflits entre les anciens et les nouveaux fichiers d'entraînement.
- **Encodage des visages** : Les encodages des visages connus sont stockés dans des fichiers **encodages.pkl**, permettant une récupération rapide lors de la phase de reconnaissance.

## 2.5 Conclusion

Ce chapitre a détaillé les techniques appliquées dans le projet *Hina Eye*, qui se concentre principalement sur l'utilisation de bibliothèques telles que **face\_recognition**, **OpenCV** et **Pillow** pour la détection, l'encodage et la reconnaissance faciale. Les étapes de prétraitement et les modèles HOG/CNN offrent une base solide pour reconnaître les visages dans des images fixes, avec des résultats prometteurs qui varient en fonction de la taille et de la qualité des datasets utilisés.

# Chapitre 3

## Méthodologie de l'Étude

Ce chapitre aborde la méthodologie employée pour la réalisation du projet de recherche sur les personnes disparues à l'aide des techniques de traitement d'images. Il présente le choix des outils et des algorithmes, la description des données, ainsi que la mise en œuvre des techniques choisies. L'objectif est de détailler le processus qui permet de transformer les données visuelles brutes en informations exploitables pour la recherche de personnes.

### 3.1 Choix des outils et algorithmes

Le choix des outils et des algorithmes est une étape essentielle pour la réussite de ce projet. Plusieurs bibliothèques et Framework de traitement d'images et d'apprentissage automatique sont disponibles. Chacun offre des avantages particuliers pour l'exécution des tâches liées à la détection, à la reconnaissance et au suivi des personnes.

#### 3.1.1 Justification des outils choisis

#### 3.1.2 Face\_recognition

Cette bibliothèque est essentielle pour la reconnaissance faciale. Elle est simple à utiliser et repose sur les algorithmes DLib pour l'encodage des visages et la détection des points d'intérêt. Elle supporte aussi bien les modèles HOG (Histograms of Oriented Gradients) que CNN, rendant cette bibliothèque polyvalente selon les besoins en ressources.

### 3.1.3 Pillow (PIL)

Pillow est utilisé pour la manipulation des images, notamment le redimensionnement, la conversion en niveaux de gris, et l'application de filtres d'amélioration des images. Il permet de préparer les images pour l'étape d'encodage.

### 3.1.4 OpenCV

Utilisé principalement pour le filtrage du bruit et la normalisation de l'éclairage dans les images, OpenCV est une bibliothèque puissante pour le traitement d'images en temps réel. La fonction CLAHE (Contrast Limited Adaptive Histogram Equalization) permet d'améliorer le contraste des images.

### 3.1.5 NumPy

Utilisé pour manipuler les matrices d'images et effectuer des calculs sur les pixels, notamment lors des étapes de prétraitement.

### 3.1.6 Flask et Flask-CORS

Ces bibliothèques sont utilisées pour implémenter une interface web, permettant d'uploader des images et d'appliquer la reconnaissance faciale via une interface simple.

## 3.2 Comparaison des performances attendues des différentes techniques

Les outils choisis présentent des performances variées en fonction des tâches spécifiques en fonction des ressources disponibles et de la taille des datasets, le modèle HOG a été principalement utilisé pour garantir une exécution rapide.

- **Modèle HOG** : Le modèle HOG, qui utilise uniquement le CPU, est rapide mais moins précis que le modèle CNN. Il est adapté aux machines sans GPU et peut être utilisé dans des environnements où la vitesse est primordiale.
- **Modèle CNN** : Le modèle CNN, quant à lui, est plus précis mais nécessite des ressources plus importantes (GPU). Il est conseillé lorsque

la précision est critique, par exemple pour l'analyse de grandes quantités de données ou des images complexes.

## 3.3 Description des données

### 3.3.1 Type de données utilisées

Le projet se concentre sur l'utilisation d'images fixes. Le choix d'éviter les séquences vidéo a été motivé par des contraintes de temps et la nécessité d'obtenir des autorisations pour accéder à des infrastructures publiques, telles que les caméras de vidéosurveillance gérées par la Police ou d'autres autorités. Les images fixes facilitent l'application de techniques comme le prétraitement, la détection de contours et l'extraction de points d'intérêt sans avoir à gérer les variations temporelles propres aux vidéos. Cela permet d'explorer en profondeur les algorithmes de détection et de reconnaissance faciale.

### 3.3.2 Source des données

Deux datasets ont été utilisés pour l'entraînement et la validation des algorithmes :

Deux datasets ont été utilisés pour l'entraînement et la validation des algorithmes :

- un jeu de données personnel constitué de quelques images d'enfants résidant dans l'immeuble de mon appartement a été créé. En raison du petit nombre de photos disponibles, ce dataset a principalement servi pour les tests internes avec un taux de reconnaissance d'environ 60%.
- 7 Celebrity Images (Kaggle) : Ce dataset, largement utilisé pour l'entraînement des modèles de reconnaissance faciale, a permis d'entraîner le modèle sur un ensemble plus grand et plus varié. Le modèle a atteint une précision de 90,61% sur ce dataset.

## 3.4 Mise en œuvre des techniques

La mise en œuvre des techniques de traitement d'images dans le cadre de la recherche de personnes disparues s'est déroulée en plusieurs étapes, allant du prétraitement des images à l'extraction des caractéristiques et à la reconnaissance faciale. En raison des contraintes de temps et du type de

données utilisées (images fixes), les algorithmes ont été adaptés pour répondre à cette spécificité tout en permettant d’obtenir des résultats significatifs.

### 3.4.1 Explication des étapes de la mise en œuvre des techniques choisies

La méthodologie suivie pour la mise en œuvre des techniques choisies s’est organisée autour de plusieurs étapes essentielles.

#### Prétraitement des images

Les images provenant des datasets ont été redimensionnées (*800x800 pixels*) et converties en niveaux de gris. Cette étape facilite l’analyse en éliminant les informations de couleur qui ne sont pas essentielles pour la reconnaissance faciale. Un **filtrage du bruit** a été appliqué à l’aide de la fonction `fastNlMeansDenoising` d’OpenCV pour réduire le bruit visuel. Cela permet d’obtenir des images plus propres, améliorant la précision des algorithmes de reconnaissance faciale. L’éclairage des images a été **normalisé** avec l’algorithme CLAHE pour égaliser l’histogramme et améliorer le contraste.

#### Entraînement et encodage des visages

Les visages ont été détectés et encodés en utilisant la bibliothèque `face_recognition`. Le modèle HOG a été principalement utilisé pour la détection et l’encodage, en raison de sa rapidité sur les systèmes ne disposant pas de GPU. Chaque visage encodé a été associé à un nom (classe) correspondant à la personne sur la photo.

### 3.4.2 Détails sur les algorithmes appliqués (prétraitement, extraction de caractéristiques, reconnaissance...)

Les algorithmes appliqués ont été soigneusement choisis pour maximiser l’efficacité du traitement des images fixes.

#### Prétraitement des images

L’application des filtres de réduction du bruit et de normalisation de l’éclairage a permis de maximiser la qualité des images, particulièrement celles ayant des conditions d’éclairage variées. L’utilisation de `Pillow` et `OpenCV` a permis de redimensionner, convertir et améliorer les images.

## Détection et encodage des visages

`face_recognition` a été utilisé pour détecter les visages et générer des encodages numériques uniques pour chaque visage détecté. Cette étape a été réalisée en utilisant à la fois les modèles HOG et CNN, selon les ressources disponibles.

## Reconnaissance faciale

Une fois les encodages créés, le système compare les encodages des visages inconnus avec ceux stockés dans la base de données d'entraînement. Cette comparaison est effectuée via la méthode `compare_faces` de la bibliothèque `face_recognition`. Si une correspondance est trouvée, le visage est identifié ; sinon, il est marqué comme "inconnu".

## 3.5 Conclusion

La méthodologie utilisée dans le projet *Hina Eye* repose sur l'application de techniques de traitement d'images et de reconnaissance faciale à partir d'images fixes. Le choix des outils et des algorithmes a été guidé par la nécessité de maximiser l'efficacité tout en prenant en compte les contraintes de temps et de ressources matérielles. L'utilisation de bibliothèques comme `face_recognition`, `OpenCV` et `Pillow` a permis de mettre en place un pipeline efficace pour la détection, l'encodage et la reconnaissance des visages. Le modèle HOG a été privilégié pour ses performances en termes de rapidité sur des systèmes CPU, tandis que le modèle CNN a été utilisé de manière optionnelle pour améliorer la précision, notamment avec des datasets plus grands. Le prétraitement des images a joué un rôle crucial dans l'amélioration de la qualité des données, garantissant une meilleure détection des visages et un encodage plus précis. La méthodologie adoptée a permis de structurer efficacement le projet, facilitant ainsi l'évaluation des performances des modèles et l'analyse des résultats.

# Chapitre 4

## Résultats et Analyses

Ce chapitre présente les résultats obtenus après la mise en œuvre des techniques décrites dans les sections précédentes. Il analyse la performance des modèles de reconnaissance faciale en fonction des différents datasets utilisés et des méthodes de traitement d'images appliquées. L'accent est mis sur l'évaluation de la précision des algorithmes en fonction de la taille des datasets et des conditions d'entraînement.

### 4.1 Résultats obtenus

L'entraînement des modèles de reconnaissance faciale a été réalisé à partir de deux datasets distincts, ce qui a permis de comparer les performances des algorithmes dans des contextes variés.

#### 4.1.1 Jeu de données personnalisé (photos d'enfants)

Le premier dataset, constitué de photos d'enfants dans l'immeuble de l'utilisateur, a servi de base pour un test initial des algorithmes. En raison de la petite taille de ce dataset, les résultats obtenus lors de l'entraînement et de la validation ont été plus modestes. Le modèle basé sur HOG a permis d'atteindre un score d'entraînement d'environ **60%**. Ce score illustre les limites d'un jeu de données réduit en termes de diversité et de nombre d'images, où la précision est largement affectée.

#### 4.1.2 Dataset Kaggle (7 Celebrity Images)

Le second dataset, constitué de photos de célébrités tiré de Kaggle, a été utilisé pour entraîner le modèle sur un plus grand volume de données. En

utilisant ce dataset plus riche et diversifié, les performances du modèle se sont significativement améliorées, atteignant une précision d'environ 90,61% sur l'ensemble des images. Ce résultat démontre que l'augmentation de la taille et de la diversité des données permet une meilleure reconnaissance faciale, et que les modèles HOG et CNN peuvent être utilisés de manière efficace pour des ensembles de données plus grands. Les résultats montrent également que le modèle CNN, bien que plus lent, offre une meilleure performance que le modèle HOG, en particulier lorsque le dataset est large et contient des images de qualité variable.

## 4.2 Analyse des performances

Les performances des modèles ont été évaluées selon deux critères principaux : **la précision de la reconnaissance faciale** et **le temps de traitement**.

### 4.2.1 Précision de la reconnaissance faciale

- **Modèle HOG** : Le modèle HOG, utilisé principalement en raison de la simplicité de son implémentation et de sa rapidité, a montré des résultats satisfaisants pour des datasets de taille modeste. Toutefois, sa précision baisse lorsque les conditions d'éclairage sont mauvaises ou lorsque les images sont partiellement obstruées.
- **Modèle CNN** : Le modèle CNN, bien que plus exigeant en termes de ressources, a produit de meilleurs résultats en termes de précision, particulièrement avec le dataset Kaggle. La capacité du CNN à extraire des caractéristiques complexes et à s'adapter à des variations d'échelle et de rotation a permis d'améliorer considérablement la reconnaissance faciale.

### 4.2.2 Temps de traitement

- Le **modèle HOG** est plus rapide, notamment parce qu'il est exécuté sur le CPU, ce qui en fait un choix approprié pour des systèmes ne disposant pas de GPU. Le temps de traitement moyen pour une image est d'environ 1 à 2 secondes sur un processeur classique.
- Le **modèle CNN**, en revanche, demande plus de temps de calcul, particulièrement lorsque l'on travaille sur des datasets volumineux. Cependant, avec l'utilisation d'un GPU compatible CUDA, ce modèle est capable de traiter des images plus complexes en moins de temps



que sur un CPU. Le temps de traitement pour une image avec CNN varie de 5 à 10 secondes selon les ressources.

## **4.3 Limites et contraintes**

### **4.3.1 Taille du dataset**

L'un des principaux défis rencontrés lors de cette étude est la taille limitée du dataset personnalisé (photos d'enfants). Les résultats obtenus avec ce jeu de données montrent que la reconnaissance faciale est beaucoup plus difficile lorsque les algorithmes sont formés sur un faible nombre d'images. Le manque de diversité des photos (en termes de conditions d'éclairage, d'angles, etc.) a également affecté la performance des modèles.

### **4.3.2 Qualité des images**

La qualité des images joue un rôle essentiel dans la reconnaissance faciale. Les images provenant du dataset Kaggle sont généralement de bonne qualité, avec des visages clairement visibles, ce qui a facilité l'entraînement et la reconnaissance. En revanche, les photos du jeu de données personnalisé présentaient des conditions d'éclairage et des résolutions variables, ce qui a compliqué le processus d'encodage et réduit la précision.

### **4.3.3 Contraintes matérielles**

L'utilisation du modèle CNN, bien que plus précis, a été limitée par l'accès aux ressources matérielles (GPU). En l'absence d'un GPU performant, le temps de traitement pour chaque image a été plus long, limitant l'applicabilité de ce modèle dans un environnement où le temps réel est crucial.

## **4.4 Améliorations et perspectives**

Les résultats obtenus montrent un potentiel prometteur pour l'application de ces techniques à la recherche de personnes disparues, mais plusieurs améliorations pourraient être apportées.

### **4.4.1 Augmentation de la taille des données**

Une augmentation de la taille et de la diversité du dataset, par exemple en intégrant des photos issues de plusieurs sources (bases de données publiques,

images de surveillance, etc.), permettrait d'améliorer encore la précision des modèles. La diversité en termes d'angles, d'éclairage et de qualité des images serait cruciale pour développer un système robuste capable de fonctionner dans des situations réelles.

#### 4.4.2 Optimisation du temps de traitement

Le modèle CNN, bien que performant, nécessite une optimisation des ressources pour pouvoir être utilisé efficacement. L'accès à des GPU plus puissants ou l'optimisation des algorithmes pour fonctionner sur des architectures de calcul parallèle permettrait d'accélérer le temps de traitement, rendant le système applicable dans des contextes de recherche en temps réel.

#### 4.4.3 Développement d'une interface utilisateur

Une autre perspective est l'intégration d'une interface utilisateur simple (via une application web ou mobile) permettant de faciliter l'utilisation du système par des équipes de secours ou des enquêteurs. L'intégration avec une plateforme de surveillance en temps réel pourrait également être envisagée pour détecter automatiquement les visages dans des séquences vidéo. (*Voir le fichier README.md*)

### 4.5 Conclusion

Les résultats obtenus dans le projet "Hina Eye" ont démontré l'efficacité des algorithmes de reconnaissance faciale implémentés, avec des scores d'entraînement variés selon la taille et la qualité des jeux de données utilisés. Le modèle HOG, plus rapide, a permis d'obtenir des performances satisfaisantes avec des datasets limités, atteignant un score de 60% avec le jeu de données personnalisé. En revanche, l'utilisation du modèle CNN, avec des datasets plus volumineux et diversifiés tels que le dataset Kaggle, a permis d'atteindre une précision plus élevée, avec un score de 90,61%. L'analyse des performances a également révélé que la qualité des images, notamment en termes de contraste et de bruit, ainsi que la taille du dataset, jouent un rôle déterminant dans la précision de la reconnaissance faciale. Bien que des limitations aient été identifiées, notamment l'absence de gestion de vidéos et des ressources GPU limitées, les résultats obtenus montrent le potentiel de l'approche adoptée pour la recherche de personnes disparues. Des perspectives d'amélioration sont envisagées, notamment l'intégration de séquences vidéo et l'optimisation du modèle CNN pour des environnements en temps réel.

# Chapitre 5

## Conclusion et Perspectives

Le présent mémoire avait pour objectif de développer une approche basée sur le traitement d'images et la reconnaissance faciale dans le cadre de la recherche de personnes disparues. En s'appuyant sur des techniques de traitement d'images modernes et des algorithmes de machine learning, nous avons exploré l'efficacité de différentes méthodes pour la détection et la reconnaissance de visages dans des images fixes. Ce chapitre récapitule les principaux résultats obtenus, tire les conclusions de l'étude et propose des perspectives pour les futures améliorations du projet *Hina Eye*.

### 5.1 Résumé des résultats

Les résultats obtenus montrent que l'utilisation de techniques de reconnaissance faciale, comme les modèles HOG et CNN, permet d'atteindre des niveaux de précision intéressants, en fonction de la taille et de la qualité des données.

- Le **modèle HOG** a offert de bonnes performances pour la reconnaissance faciale avec des temps de calcul rapides, ce qui en fait un choix adapté pour des systèmes disposant de ressources matérielles limitées. Cependant, sa précision diminue dans des conditions d'éclairage difficiles ou pour des visages partiellement obstrués.
- Le **modèle CNN** a montré des performances de reconnaissance supérieures, particulièrement avec des datasets de grande taille, atteignant une précision de 90,61% sur le dataset "7 Celebrity Images" de Kaggle. En revanche, ce modèle nécessite des ressources matérielles importantes (GPU), ce qui limite son applicabilité dans des environnements sans infrastructure dédiée.
- Le **prétraitement des images** a été une étape cruciale pour amélio-

rer la qualité des données d'entrée. L'application de techniques comme la réduction du bruit et la normalisation de l'éclairage a permis d'optimiser les images pour la phase d'encodage et de reconnaissance. Les deux modèles ont cependant montré leurs limites lorsque le jeu de données est petit ou de qualité variable. L'utilisation d'un petit dataset personnalisé (photos d'enfants) a atteint une précision de 60%, ce qui met en lumière l'importance de disposer d'un dataset riche et diversifié pour améliorer la performance des algorithmes.

## 5.2 Conclusion sur les performances des techniques utilisées

L'étude a démontré que les algorithmes de reconnaissance faciale basés sur HOG et CNN, couplés à des méthodes de prétraitement d'images, offrent un potentiel réel pour la recherche de personnes disparues à partir d'images fixes. Ces techniques sont capables d'identifier et de reconnaître des visages avec une précision satisfaisante, à condition de disposer de données de qualité et d'une infrastructure matérielle appropriée. Le modèle HOG, plus léger, offre une solution rapide et utilisable même avec des ressources limitées. Le modèle CNN, bien que plus exigeant, représente une option robuste pour des environnements où la précision est critique, comme dans le cadre d'enquêtes ou de secours. Les résultats obtenus montrent qu'il est possible d'améliorer la précision de la reconnaissance faciale en augmentant la taille du dataset et en optimisant les étapes de prétraitement.

## 5.3 Limites de l'étude

- Plusieurs limitations ont été identifiées au cours de cette étude :
- **Taille des datasets** : Le petit dataset personnalisé n'a pas permis d'atteindre des résultats optimaux. La diversité des images, tant en termes d'angles, d'éclairage que de qualité, est essentielle pour garantir une reconnaissance faciale robuste.
  - **Ressources matérielles** : L'absence d'accès à un GPU performant a limité l'utilisation du modèle CNN dans les phases de test. Bien que le modèle HOG soit suffisant pour certains scénarios, l'optimisation du système pour des environnements en temps réel nécessitera l'utilisation de GPU pour accélérer les calculs.
  - **Images fixes** : Le projet s'est concentré sur des images fixes, limitant ainsi l'applicabilité dans des contextes où la vidéo ou les flux en temps

réel sont nécessaires. L'intégration de séquences vidéo dans le cadre d'un suivi dynamique des personnes disparues représente une avancée importante.

## 5.4 Perspectives futures

Plusieurs pistes d'amélioration et de développement futurs peuvent être envisagées pour perfectionner le projet *Hina Eye* et augmenter son efficacité dans la recherche de personnes disparues.

### 5.4.1 Augmentation et diversification du dataset

Une des principales améliorations consiste à augmenter la taille du dataset en intégrant des images provenant de différentes sources, comme des bases de données publiques, des photos de vidéosurveillance, ou des photos diffusées par les familles des personnes disparues. Cela permettrait d'améliorer la diversité des images utilisées pour l'entraînement des modèles, augmentant ainsi la robustesse de la reconnaissance faciale.

### 5.4.2 Intégration de la reconnaissance à partir de séquences vidéo

Le passage des images fixes aux séquences vidéo serait une amélioration majeure pour ce projet. Cela permettrait d'utiliser des flux vidéo captés par des caméras de surveillance ou des drones pour suivre les déplacements de personnes disparues en temps réel. Des techniques comme le suivi d'objets par **Optical Flow** ou **Kalman Filter** pourraient être intégrées pour analyser les vidéos.

### 5.4.3 Optimisation des algorithmes et des ressources matérielles

Une autre perspective est d'optimiser le modèle CNN pour qu'il puisse être utilisé de manière plus efficace sur des machines dépourvues de GPU. L'optimisation des algorithmes pourrait inclure l'utilisation de versions plus légères des réseaux neuronaux, comme les **MobileNets**, qui sont conçus pour fonctionner sur des appareils mobiles avec des ressources limitées.

#### **5.4.4 Développement d'une plateforme intégrée**

Le développement d'une interface web complète pourrait rendre le système accessible aux équipes de recherche et de secours. Une application web ou mobile permettrait aux utilisateurs de télécharger des photos et de rechercher des correspondances en temps réel à partir d'une base de données. Ce type de plateforme pourrait être connecté à un système de surveillance pour automatiser la détection et la reconnaissance de visages dans des environnements urbains.

#### **5.4.5 Recherche collaborative et partenariats**

Le développement de partenariats avec des institutions publiques et privées (forces de l'ordre, ONG, centres de recherche) permettrait d'accéder à des données plus riches et diversifiées, ainsi qu'à des infrastructures de calcul plus performantes. Une collaboration avec des autorités locales faciliterait l'accès à des images de vidéosurveillance dans les enquêtes réelles, renforçant l'applicabilité de la solution.

# Annexe A

## Exemples

### A.1 Code source

Tout le code source du projet est disponible [ici](#)

```
1 import argparse
2 import pickle
3 from collections import Counter
4 from pathlib import Path
5 import shutil
6 import face_recognition
7 from PIL import Image, ImageDraw, ImageEnhance
8 import cv2
9 import numpy as np
10
11 # Chemin par défaut pour les encodages
12 CHEMIN_ENCODAGES_PAR_DEFAULT = Path("output/encodages.pkl")
13 COULEUR_BOITE_DELIMITATION = "blue"
14 COULEUR_TEXTE = "white"
15
16 # Creation des repertoires s'ils n'existent pas déjà
17 Path("output").mkdir(exist_ok=True)
18 Path("processed").mkdir(exist_ok=True)
19
20 # Configuration des arguments en ligne de commande
21 parser = argparse.ArgumentParser(description="Reconnaitre des
    visages dans une image")
22 parser.add_argument("--train", nargs="+", help="Chemins vers
```

```

    les repertoires des datasets d\'entrainement")
23 parser.add_argument("--validate", action="store", nargs="?",
    const="validation", help="Valider le modele entrane
    (chemin du repertoire facultatif)")
24 parser.add_argument("--test", action="store_true",
    help="Tester le modle avec une image inconnue")
25 parser.add_argument(
26     "-m",
27     action="store",
28     default="hog",
29     choices=["hog", "cnn"],
30     help="Quel modle utiliser pour l\'entrainement : hog (CPU),
    cnn (GPU)",
31 )
32 parser.add_argument("-f", action="store", help="Chemin vers
    une image avec un visage inconnu")
33 args = parser.parse_args()
34
35 # Fonction pour supprimer le contenu du dossier 'processed'
36 def supprimer_contenu_dossier(dossier="processed"):
37     pass
38
39 # Fonction pour normaliser l\'eclairage d\'une image
40 def normaliser_eclairage(image_np):
41     pass
42
43 # Fonction pour pretraiter les images et les sauvegarder dans
    le dossier 'processed'
44 def pretraiter_images(dossiers_entree,
    dossier_sortie="processed", taille=(800, 800)):
45     pass
46
47 # Fonction pour encoder les visages connus dans les images du
    dossier 'processed'
48 def encoder_visages_connus(
49     modele: str = "hog", emplacement_encodages: Path =
    CHEMIN_ENCODES_PAR_DEFAUT, dossier_entree="processed"
50 ) -> None:
51     pass
52
53 # Fonction pour reconnaitre les visages dans une image donnee

```



```

54 def reconnaitre_visages(
55     emplacement_image: str,
56     modele: str = "hog",
57     emplacement_encodages: Path = CHEMIN_ENCODAGES_PAR_DEFAUT,
58     extraire_points_interet=False
59 ) -> None:
60     pass
61
62 # Fonction pour reconnaitre un visage partir de son encodage
63 def _reconnaitre_visage(encodage_inconnu, encodages_charges):
64     pass
65
66 # Fonction pour afficher le visage avec une bote de
67 # delimitation et une legende
68 def _afficher_visage(dessin, boite, nom):
69     pass
70
71 # Fonction pour valider le modle sur un ensemble d'images
72 def valider(modele: str = "hog",
73     dossier_validation="validation"):
74     pass
75
76 # Point d'entree principal
77 if __name__ == "__main__":
78     if args.train:
79         supprimer_contenu_dossier("processed")
80         pretraiter_images(dossiers_entree=args.train,
81             dossier_sortie="processed", taille=(800, 800))
82         encoder_visages_connus(modele=args.m,
83             dossier_entree="processed")
84     if args.validate:
85         valider(modele=args.m,
86             dossier_validation=args.validate)
87     if args.test:
88         reconnaitre_visages(emplacement_image=args.f,
89             modele=args.m, extraire_points_interet=True)

```

Listing A.1 – Extrait du code source du projet

## A.2 Test après entraînement



Identification d'une personne censée être portée disparu reconnu par notre modèle entraîné.