

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDERRAHMANE MIRA DE BÉJAÏA



FACULTÉ DES SCIENCES EXACTES
DÉPARTEMENT D'INFORMATIQUE
MEMOIRE DE MASTER
OPTION : INTELLIGENCE ARTIFICIELLE

Thème

Identification des enfants disparus à partir des images faciales malgré l'avancement dans l'âge

Présenté par :

KENDJOUH DAHBIA

Soutenu devant le jury composé de :

<i>Président</i>	Dr MOKTEFI MOHAND	M.C.B	U. A/MIRA BÉJAÏA
<i>Examineur</i>	Dr AISSANI SOFIANE	M.C.A	U. A/MIRA BÉJAÏA
<i>Rapporteur</i>	Dr KHAMMARI MOHAMMED	M.C.A	U. A/MIRA BÉJAÏA

Promotion 2023 – 2024

Remerciements

Je tiens à exprimer ma gratitude envers le Bon Dieu qui m'a donné la force nécessaire pour réaliser ce modeste travail.

Je tiens à remercier mon encadrant Monsieur KHAMMARI MOHAMMED de m' avoir orienté et aidé à mener à bien ce projet. je lui exprime toute ma reconnaissance pour son soutien constant, ses conseils précieux et son expertise. Sa disponibilité et sa patience.

Je tiens à remercier tous mes enseignant sans exception depuis L3 jusqu'au M2 pour leur compréhension, leur soutien et leur encouragement.

Je tiens à remercier tous mes camarades étudiants sans exception depuis L3 jusqu'au M2 pour leur soutien inoubliable.

Je tiens à remercier tout le personnel de département informatique pour les efforts porté à mon égard.

Je tiens à remercier l'incroyable , la maman des étudiants Mme LAHDIRI GHANIA pour toutes ces années auprès de moi.

Je tiens à remercier mon grand génie RAMY.

Dédicaces

Je dédie ce travaille à mes parents. Ma mère qui a toujours cru en moi, qui m'encourageait tout au long de ma vie, qui m'a toujours poussé vers l'avant, que dieux tu protège.

À ma famille, pour tout ce que vous avez fait pour moi depuis mon plus jeune âge et jusqu'à maintenant, malgré les hauts et les bas de la vie. Un grand merci à mes sœurs qui m'ont toujours épaulée dans toutes les situations, LILA, SOUAD et SAFIA a mes frères NABIL, Hassen, BOUZID et YACINE qui a été toujours là pour moi.

Au lumières de mes yeux, mes merveilleux enfants RAMY et ILIAN que Dieu les protègent.

À celui qui me disait "Maman je veux que tu sois un Docteur des Ordinateurs" Cyril. À tous jamais dans mon cœurs.

Mme. KENDJOUH DAHBIA

Résumé

L'identification des enfants disparus est une préoccupation universelle qui transcende les frontières géographiques et culturelles. Chaque année, des milliers d'enfants disparaissent, laissant derrière eux des familles dévastées et des communautés bouleversées. Bien que les technologies de recherche et de sauvetage aient considérablement progressé au fil des ans, de nombreux défis subsistent, en particulier lorsqu'il s'agit d'enfants qui ont grandi et dont l'apparence a évolué avec le temps. L'âge est un défi complexe en raison des changements physiques qui surviennent au fil du temps tels que la perte des dents, la maturité de la peau, la puberté, etc.

Pour relever ce défi complexe, Nous proposons une nouvelle méthode qui consiste à réaliser un système d'identification robuste et efficace qui se base sur les réseaux de neurones pour la détection et l'alignement des visages avec la méthode MTCNN, deux méthodes puissantes pour l'extraction des caractéristiques locales LBP et HOG, deux méthodes très efficaces pour la réduction des dimensions PCA et LDA, un CNN pour l'apprentissage et la classification et Finalement la distance euclidienne pour la prise de décision. notre système sera appliqué sur la base de donnée FG net.

mot clé : Identification, réseaux de neurones, MTCNN, CNN, extraction des caractéristiques locales, LBP, HOG, réduction des dimensions, PCA, LDA, apprentissage, classification, CNN, prise de décision, distance euclidienne, FG net.

Table des matières

Table des matières	I
Table des figures	III
Liste des tableaux	V
Liste des abréviations	VI
Introduction générale	1
1 Identification des enfants disparus	3
1.1 Introduction	3
1.2 Identification des enfants disparus	4
1.3 Travaux connexes	6
1.4 Conclusion	13
2 La reconnaissance faciale	14
2.1 Introduction	14
2.2 Détection de visage	15
2.2.1 Méthodes de détection de visage	16
2.3 Alignement	22
2.4 Extraction des caractéristiques	23
2.4.1 Méthodes d'extraction des caractéristiques	23
2.5 Réduction de dimension	28
2.5.1 Méthodes de réduction de dimension	28
2.6 Apprentissage profond : Deep Learning	29
2.6.1 Principales caractéristiques de l'apprentissage profond	30
2.6.2 Les réseaux de neurones	31
2.6.3 CNN : Réseaux de neurones convolutifs	32
2.6.4 Principales caractéristiques des CNN	32
2.7 Prise de décision	35
2.7.1 La classification	35
2.7.2 La régression	35
2.8 Evaluation	36

2.9	Conclusion	37
3	Système d'identification des enfants disparus malgré l'avancement dans l'âge	38
3.1	Description du système proposé	38
3.2	Apprentissage de notre système :	38
3.3	Test de notre système :	39
3.4	Préparation de la base de données	40
3.5	Méthodes utilisées	40
3.5.1	Détection de visage	41
3.5.2	Alignement :	46
3.5.3	Extraction des caractéristiques	47
3.5.4	Réduction de dimensions	50
3.5.5	Réseau de Neurones Convolutifs : CNN	54
3.6	Comparaison et Prise de décision	56
3.6.1	Distance Euclidienne	57
3.7	Conclusion	58
4	Environnement matériel et logiciel	59
4.1	Environnement matériel	59
4.2	Environnement logiciel	59
4.2.1	Python	59
4.2.2	OpenCV	60
4.2.3	NumPy	60
4.2.4	Matplotlib	61
4.2.5	TensorFlow	61
4.2.6	keras	61
4.2.7	Google Colaboratory	62
4.3	Bases de données utilisées	62
4.4	Algorithme	63
4.4.1	MTCNN : Réseaux convolutifs en cascade multitâches	63
4.4.2	LBP : Motifs binaires locaux	64
4.4.3	HOG : Histogramme des gradients orientés	66
4.4.4	Concaténation des vecteurs de caractéristiques LBP et HOG	68
4.4.5	PCA : Analyse en Composantes Principales	68
4.4.6	LDA : Analyse discriminante linéaire	68
4.5	Conclusion	69
	Conclusion	70
	Bibliographie	72

Table des figures

1.1	Croissance et développement faciale	4
1.2	Évolution des caractéristiques du visage	4
1.3	Maturité de la peau	5
1.4	Perte de dents et changements buccaux	5
1.5	Puberté et changements hormonaux	5
1.6	Expression faciale	6
2.1	Système de reconnaissance faciale	15
2.2	Architecture d'un CNN	32
3.1	Système d'identification d'enfants - Partie Apprentissage	39
3.2	Système d'identification d'enfants - Partie Test	40
3.3	Architecture du réseau en cascade MTCNN [37]	42
3.4	Image organisée en pyramide	42
3.5	Fenêtre glissante	43
3.6	Analyse la sortie P-Net	44
3.7	Rejet des faux candidats	45
3.8	Les 5 points de repères faciaux	46
3.9	Image avant et après Alignement	47
3.10	Image donnée par le descripteur LBP	47
3.11	Un ensemble voisinage d'opérateur LBP	48
3.12	Image donnée par le descripteur HOG	50
3.13	Image donnée après l'application de PCA	51
3.14	Représentation de la dimension PCA	52
3.15	Représentation de la dimension LDA	54
3.16	Réseaux de neurones convolutifs (CNN) utilisé	56
4.1	Code python de la méthode MTCNN	63
4.2	visage détecté avec MTCNN	64
4.3	Code python de la méthode LBP	65
4.4	Image après application de LBP	66
4.5	Code python de la méthode HOG	67
4.6	Image après application de HOG	67

4.7	code python pour la Concaténation des vecteurs de caractéristiques	68
4.8	code python de la méthode PCA	68
4.9	code python de la méthode LDA	69

Liste des tableaux

- 1.1 Tableau comparatif des approches étudiées 12
- 2.1 Tableau de quelques réseaux de neurones et leurs domaines d'application 32
- 2.2 Tableau de quelques CNN et leurs domaines d'application 34

Liste des abréviations

API	<i>interface de programmation d'application</i>
BSIF	<i>caractéristiques des images statistiques binarisées</i>
CNN	<i>Réseau de Neurones Convolutifs</i>
DCT	<i>Transformation en cosinus discrète</i>
DL	<i>Apprentissage Profond</i>
DNN	<i>Réseaux de neurones profonds</i>
HOG	<i>Histogramme des Gradients Orientés</i>
ICA	<i>Analyse en composantes indépendantes</i>
LBP	<i>Modèles binaires locaux</i>
LDA	<i>Analyse Discriminante Linéaire</i>
LPQ	<i>La quantification de la phase locale</i>
ML	<i>Apprentissage Machine</i>
MTCNN	<i>Réseaux convolutifs en cascade multitâches</i>
R-CNN	<i>Regions avec CNN</i>
PCA	<i>Analyse en Composantes Principales</i>
ResNet	<i>Réseaux de neurones résiduels</i>
RNN	<i>Réseaux de neurones récurrents</i>
SIFT	<i>transformation de caractéristiques invariantes à l'échelle</i>
SSD	<i>Détecteur MultiBox à prise unique</i>
SSH	<i>Détecteur de visage sans tête a une étape</i>
YOLO	<i>Vous ne regardez qu'une seule fois</i>

Introduction générale

Dans notre société contemporaine, la tâche consistant à retrouver les enfants disparus, même à mesure qu'ils grandissent, pose un dilemme à multiples facettes et profondément émouvant. Chaque année, une multitude d'enfants sont portés disparus et malgré les efforts inlassables déployés pour les retrouver, une partie de ces jeunes échappe à la découverte pendant de longues périodes, voir des décennies.

L'identification des enfants disparus à partir d'images faciales est une question profondément sensible et cruciale dans le domaine de la sécurité publique et de la protection de l'enfance. Malgré les progrès technologiques et les avancées dans les techniques de surveillance et de sécurité, les cas de disparition d'enfants demeurent un problème mondial persistant. L'âge est un facteur critique dans ces situations, car les enfants grandissent et évoluent rapidement, ce qui rend leur identification de plus en plus complexe au fil du temps. Lorsqu'un enfant disparaît chaque instant compte, et la capacité à retrouver rapidement et précisément ces enfants est une priorité absolue.

Les images faciales, qu'elles soient fournies par des photographies récentes ou par des images d'archives, jouent un rôle central dans la recherche et la localisation des enfants disparus. Cependant, l'identification des enfants qui ont vieilli depuis leur disparition, parfois sur plusieurs années, représente un défi considérable. Cette triste vérité suscite de profondes interrogations sur les techniques d'identification.

Dans cette étude du sujet de l'identification des enfants disparus à partir d'images faciales malgré l'avancement dans l'âge, nous allons plonger dans les défis auxquels sont confrontés les familles et les enquêteurs, ainsi que dans les développements technologiques qui permettent de mieux répondre à cette problé-

matique. En fin de compte, l'objectif est de contribuer la technologie à réunir les familles avec leurs proches disparus, tout en protégeant les droits fondamentaux des individus.

Notre travail est organisé en quatre chapitres qui nous permettront de présenter les différents aspects de notre solution. Dans le premier chapitre, nous donnerons une Identification au problème que nous allons aborder qui est l'identification des enfants disparus, les différentes approches des chercheurs dans le domaine de la reconnaissance faciale, puis nous ferons une comparaison des méthodes apportées. Dans le deuxième chapitre, nous présenterons un système global de la reconnaissance faciale, puis détailler son état de l'art en présentant les différentes méthodes et outils dans le domaine. Dans le chapitre trois, nous détaillerons le fonctionnement des méthodes utilisées, voir MTCNN pour la détection de visage, LBP et HOG pour l'extraction des caractéristiques, PCA et LDA pour la réduction de dimension, ainsi que CNN pour l'apprentissage et pour finir la Distance Euclidienne pour la prise de décision. Dans le derniers chapitre, nous présenterons les outils matériels, les différents outils logiciels, la base de donnée Fgnet que nous utiliserons pour notre système et finalement, nous présenterons quelques codes python des méthodes utilisés et leurs images résultantes. Enfin, une conclusion générale qui résumera les différents travaux effectués.

Chapitre 1

Identification des enfants disparus

1.1 Introduction

L'identification des enfants disparus est une préoccupation humanitaire majeure qui touche les cœurs et les esprits du monde entier. Chaque année, des milliers de familles vivent l'angoisse de ne pas savoir où se trouvent leurs enfants bien-aimés. La disparition d'un enfant est une tragédie qui a des conséquences profondes, non seulement pour la famille concernée, mais aussi pour la société dans son ensemble. Dans ces moments de détresse, le temps devient un ennemi implacable, car chaque minute compte pour retrouver l'enfant disparu en toute sécurité. [15]

Pourtant, il existe un défi supplémentaire qui complexifie encore davantage cette quête désespérée : l'âge. Les enfants grandissent, et leurs visages évoluent avec eux. Les traits juvéniles qui pouvaient être caractéristiques sur une photo prise il y a quelques années peuvent se transformer, rendant l'identification encore plus difficile à mesure que le temps passe. Cette évolution naturelle du visage pose une question cruciale : comment identifier les enfants disparus malgré les années qui s'écoulent ?

Le présent chapitre constitue le point de départ de notre exploration dans le domaine de l'identification des enfants disparus, en mettant l'accent sur la problématique posée par le vieillissement facial et les défis complexes qu'elle engendre.

1.2 Identification des enfants disparus

L'identification des enfants disparus à partir des images faciales malgré l'avancement dans l'âge est un défi complexe en raison des changements physiques qui surviennent au fil du temps.

Voici comment le visage de l'enfant peut changer avec l'âge :

1. Croissance et développement facial

Au fur et à mesure que les enfants grandissent, leur visage subit une croissance et un développement continus. Les changements les plus évidents sont généralement observés dans la structure osseuse du visage, notamment la croissance du nez, des mâchoires et du front (voir Figure 1.1).

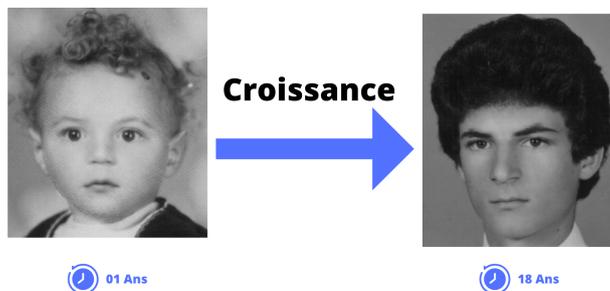


FIGURE 1.1 – Croissance et développement faciale

2. Évolution des caractéristiques du visage

Les caractéristiques du visage, telles que la forme des yeux, du nez et des lèvres, peuvent changer à mesure que l'enfant grandit. Par exemple, un enfant avec des yeux relativement grands peut voir la taille de ses yeux diminuer en proportion à mesure qu'il vieillit (voir Figure 1.2).



FIGURE 1.2 – Évolution des caractéristiques du visage

3. Maturité de la peau

La texture de la peau peut également évoluer avec l'âge. Les bébés ont souvent une peau plus douce et plus lisse, tandis que les adolescents peuvent développer de l'acné et des irrégularités cutanées (voir Figure 1.3).

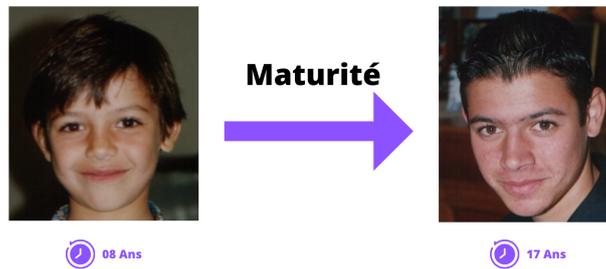


FIGURE 1.3 – Maturité de la peau

4. Perte de dents et changements buccaux

La perte de dents de lait et la croissance des dents permanentes modifient la morphologie de la bouche et de la mâchoire, affectant ainsi l'apparence du visage (voir Figure 1.4).



FIGURE 1.4 – Perte de dents et changements buccaux

5. Puberté et changements hormonaux

La puberté entraîne des changements hormonaux qui peuvent influencer la distribution des graisses dans le visage. Par exemple, certains adolescents développent des pommettes plus prononcées à mesure que leur visage s'amincit (voir Figure 1.5).

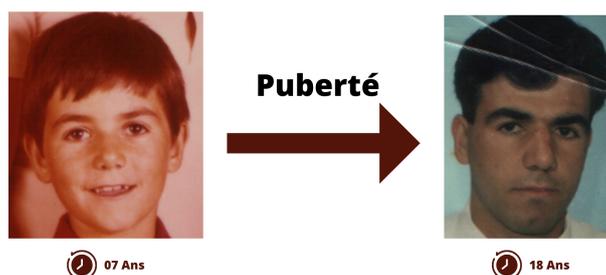


FIGURE 1.5 – Puberté et changements hormonaux

6. Expression faciale

Les expressions du visage, y compris les sourires, les froncements de sourcils et les habitudes de mouvements musculaires, peuvent également évoluer avec l'âge, affectant la manière dont le visage apparaît dans une image (voir Figure 1.6).

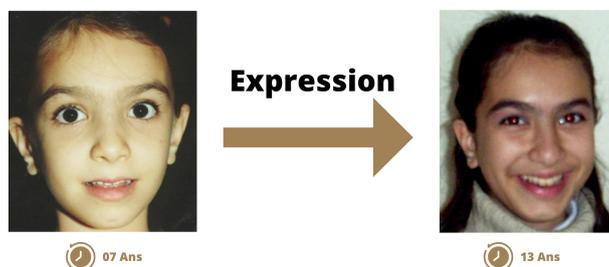


FIGURE 1.6 – Expression faciale

Tous ces éléments contribuent à la complexité de l'identification des enfants disparus à mesure qu'ils grandissent. C'est pourquoi il est essentiel de développer des approches et des technologies de reconnaissance faciale qui tiennent compte de ces variations pour maximiser les chances de réunir les enfants disparus avec leur famille, même après de nombreuses années.

Pour relever ce défi complexe, en tenant compte des changements physiques du visage, nous nous focalisons sur les caractéristiques locales du visage de l'enfant qui distingue un individu d'un autre, tout en utilisant des techniques de réduction de dimension et un réseau de neurones convolutifs (CNN).

1.3 Travaux connexes

Les travaux de recherche dans le domaine de la reconnaissance faciale ont connu une croissance exponentielle au cours des dernières décennies, stimulée par les avancées en matière de matériel informatique, de méthodes d'apprentissage automatique, et de collecte de données. Ces travaux ont permis de repousser les limites de la précision et de la fiabilité de la reconnaissance faciale, ouvrant ainsi la voie à de nouvelles applications et opportunités.

De nombreux chercheurs ont apporté des contributions significatives qui ont favorisé l'évolution de cette discipline. Cependant, il convient de noter que la reconnaissance faciale ne se limite pas aux adultes, mais aux enfants, ce qui est

un aspect crucial, notamment dans les applications de sécurité, de recherche d'enfants disparus et de protection de l'enfance. Malheureusement, cette dimension est souvent négligée dans les recherches et les développements technologiques, bien que la reconnaissance faciale pour les enfants présente des défis spécifiques et des enjeux importants en matière de vie privée et de protection des droits de l'enfant. Voici quelques-uns des auteurs et chercheurs notables qui ont laissé leur empreinte sur le domaine de la reconnaissance faciale :

Haibin Ling.al ont proposé en 2010 [23] une Approche discriminatoire pour la vérification du visage à travers les âges, Tout d'abord, un gradient (GOP) pour la représentation de visage et ils ont montré que lorsqu'elle est combinée avec la machine à vecteurs de support (SVM) la GOP démontre d'excellentes performances pour la vérification du visage avec des écarts d'âge. l'Approche a été testée sur deux jeux de données d'images de passeport réel, nommés Passport I et Passport II et la base de donnée FGNET et les expériences ont démontré que cette approche est nettement plus addaptée que les approches déjà proposées au préalable avec un taux d'erreur de 5.1% pour Passport I et 10.8% Passport II et aussi que l'approche n'est pas efficace dans l'intervalle d'âge 8 à 18 pour la base de donnée FGNET avec un taux d'erreur de 30.5% pour l'intervalle d'âge 0 à 8 et 38.6% pour l'intervalle de l'âge 8 à 18. Haibin Ling.al ont conclu que les méthodes discriminatoires ne sont pas efficaces dans l'intervalle d'âge 8 à 18 et il faut associer à l'avenir des méthodes génératives.

Zhifeng Li.al ont proposé en 2011 [22] Un modèle discriminant pour l'âge invariant pour traiter l'appariement des visages en présence de variation d'âge, ils ont représenté d'abord chaque visage en concevant un schéma de description de caractéristiques locales densément échantillonné, dans lequel la transformation de caractéristiques invariantes à l'échelle (SIFT) et les modèles binaires locaux multi échelles (MLBP) ont servé de descripteurs locaux qui ont couvert un espace d'entités de grande dimension, pour éviter le problème de surajustement est qu'ils ont développé un algorithme appelé analyse discriminante multi entités (MFDA) qui

est une extension et une amélioration de la (LDA) pour traiter ces deux espaces d'entités locales dans un cadre unifié par l'échantillonnage aléatoire en divisant les différents types de caractéristiques en tranches, puis en les mettant à l'échelle par la normalisation. (PCA) et le problème de surajustement est résolu. Le modèle a été testé sur deux jeux de données MORPH Album2 et FGNET et les expériences que ce modèle proposé dépasse de loin les autres approches avec un taux réussite de 83.9% à base de donnée MORPHAlbum2 et 47.50% pour la base de donnée FGNET. Zhifeng Li.al ont conclu que Le modèle proposé est vulnérable au changement de pose, Une méthode plus tolérante aux changements de pose devrait être étudiée dans des travaux futurs et puisque le modèle génératif et le modèle discriminatif offrent des informations quelque peu complémentaires, il est intéressant d'améliorer le cadre de fusion pour des performances améliorées.

Dihong Gong.al ont proposé en 2013 [8] une nouvelle méthode probabiliste appelée Hidden Factor Analysis (HFA) ou Analyse des facteurs cachés pour la reconnaissance faciale invariante selon l'âge avec deux facteurs latents : un facteur d'identité invariant avec l'âge et un facteur d'âge affecté par le processus de vieillissement, ils ont développé également un algorithme d'apprentissage qui estime conjointement les facteurs latents et les paramètres du modèle à l'aide d'une procédure EM (Algorithme de maximisation des attentes). les visages de formation sont d'abord regroupés en fonction de leurs identités et leurs âges, suivis de l'extraction de caractéristiques avec HOG (Histogramme des Gradients Orientés) puis la réduisons de la dimension de ces caractéristiques avec PCA (Analyse en Composantes Principales) et LDA (Analyse Discriminante Linéaire). Enfin, les modèles HFA sont adaptés indépendamment sur chacune des caractéristiques tranchées du jeu de données, obtenant un ensemble de paramètres de modèle pour chaque tranche. Le score d'appariement final est donné par la distance cosinus. La méthode a été testée sur deux jeux de données MORPH Album2 et FGNET, et les expériences que cette méthode proposée dépasse de loin les autres approches avec un taux réussite de 91.14% pour la base de donnée MORPHAlbum2 et 69.0% pour la base de donnée FGNET. Dihong Gong.al ont conclu que leur méthode est supérieure autres Approches déjà proposées.

Dihong Gong.al ont proposé en 2015 [9] une nouvelle approche pour surmonter les problèmes de représentation et d'appariement dans la reconnaissance faciale invariante selon l'âge. Tout d'abord, un nouveau descripteur de caractéristique d'entropie maximale (MEFD) basé sur un arbre de décision qui croît de manière gloutonne qui est utilisé pour extraire les caractéristiques expressives informatives. Contrairement aux descripteurs de caractéristiques existants, une nouvelle méthode d'appariement est également développée appelée Analyse Factorielle d'Identité (IFA) pour estimer la probabilité que deux visages aient la même identité sous-jacente. La méthode a été testée sur trois jeux de données MORPH Album2, FGNET et LFW, et les expériences que cette méthode proposée dépasse de loin les autres approches et ils ont amélioré leur score par rapport à leur méthode HFA en 2011 avec un taux de réussite de 94.59% pour la base de données MORPH Album2, 76.2% pour la base de données FGNET ET 88.86% pour la base de données LFW.

Zhifeng Li.al ont proposé en 2016 [21] un modèle hiérarchique (arbre d'encodage) basé sur un apprentissage à deux niveaux. Au premier niveau, les caractéristiques efficaces sont apprises à partir de microstructures de bas niveau, sur la base de leur nouveau descripteur de caractéristiques appelé Local Pattern Selection (LPS). Le descripteur LPS proposé sélectionne des modèles discriminants de bas niveau de manière à minimiser la dissemblance intra-utilisateur. Au deuxième niveau, les informations visuelles de niveau supérieur sont encore affinées sur la base de la sortie du premier niveau. PCA et USA (Universal Subspace Analysis) sont utilisés comme classificateur de base. Le modèle a été testé sur le jeu de données MORPH Album2 et les expériences que ce modèle proposé montre une amélioration significative de la précision par rapport aux méthodes de pointe avec un taux de réussite de 92.11%.

Debayan Deb.al ont proposé en 2020 [4] une nouvelle méthode qui peut vieillir les traits profonds du visage afin d'améliorer les performances de reconnaissance faciale longitudinale dans l'identification des enfants disparus. La méthode propo-

sée guide également le processus de vieillissement dans l'espace d'image, de sorte que les images vieilles synthétisées peuvent améliorer la précision de la reconnaissance faciale inter âge de n'importe quel comparateur de visage de produit, cette méthode englobe trois modules :

1. Module de vieillissement des fonctionnalités(FAM)

Ce module est une série de couches entièrement connectées apprennent les paramètres d'échelle W et de biais b Pour l'apprentissage, il reçoit une paire authentique de caractéristiques de visage extraites d'un encodeur d'identité (EID) de la même personne acquise respectivement aux deux âges différents et afin de s'assurer que les identités sont préservées et que les caractéristiques synthétisées progressent en fonction de l'âge jusqu'à l'âge souhaité t , FAM est entraîné en utilisant une perte d'erreur quadratique moyenne (MSE) qui mesure la qualité des caractéristiques prédites, Une fois le modèle formé, FAM peut faire évoluer une caractéristique du visage à l'âge désiré.

2. Encodeur de style (Encodeur de caractéristiques résiduelles)

Les caractéristiques résiduelles au niveau des pixels qui ne sont pas liées à l'identité d'une personne, mais qui sont nécessaires pour améliorer la qualité visuelle des images qui ne sont pas nécessairement liées à l'identité d'une personne, mais qui sont nécessaires pour améliorer la qualité visuelle de l'image de visage synthétisée, cependant Le décodage direct d'un vecteur de caractéristiques dans une image peut donc gravement affecter la qualité visuelle du visage synthétisé. C'est pourquoi ils ont utilisé un encodeur de style (Estyle) qui prend une image de visage comme entrée et produit un vecteur de caractéristiques k -dimensionnel qui encode les informations de style de l'image.

3. Décodeur

Afin de synthétiser une image de visage, ils ont proposé un décodeur D qui prend en entrée un style et un vecteur d'identification obtenus à partir d'Estyle et d'EID, respectivement, et produit une image. Leur objectif était de synthétiser des images de visages vieillis qui peuvent être utiles aux systèmes de reconnaissance faciale. Par conséquent, une perte au niveau des pixels est également adoptée pour maintenir la cohérence du contenu de l'image de bas niveau entre l'image d'entrée et l'image de sortie.

La méthode a été testée sur trois jeux de données ITWCC, FGNET et CACD-VS et les expériences que cette méthode proposée dépasse de loin les autres approches avec un taux de réussite de 95.91% pour FGNET et 99.58% pour CACD-VS et améliore la précision d'identification de l'ensemble FaceNet de 16,53% à 21,44% et de CosFace de 60,72% à 66,12% sur un ensemble de données ITWCC. Debayan Deb.al ont conclu que ces résultats suggèrent que le schéma de vieillissement proposé peut améliorer la capacité des appareilleurs de visages à localiser et à identifier les jeunes enfants qui sont perdus à un jeune âge afin de les réunir avec leurs familles. et ils prévoient d'étendre leur travail aux images de visage d'enfant sans contrainte, ce qui est typique dans les affaires de traite d'enfants.

Ces chercheurs ont contribué à façonner la reconnaissance faciale telle que nous la connaissons aujourd'hui. Leurs travaux ont permis d'améliorer la précision, la vitesse et la robustesse des systèmes de reconnaissance faciale, tout en relevant les défis éthiques et de confidentialité associés à cette technologie (voir Table 1.1).

Auteur	Approche	Base de donnée utilisée	Taux de réussite
Haibin Ling.al 2010 [23]	HOG + SVM	Passport I	94.9%
		Passport II	89.2%
		FGNET	30.5% age [0-8]
		FGNET	38.6% age [8-18]
Zhifeng Li.al 2011 [22]	MFLD	MORPH Album2	83.9%
		FGNET	47.50%
Dihong Gong.al 2013 [8]	HFA	MORPH Album2	91.14%
		FGNET	69.0%
Dihong Gong.al 2015 [9]	MEFD	MORPH Album2	94.59%
		FGNET	76.2%
		LFW	88.86%
Zhifeng Li.al 2016 [21]	Arbre d'encodage + LPS	MORPH Album2	92.11%
Debayan Deb.al 2020 [4]	FAM	FGNET	95.91%
		CACD-VS	99.58%

TABLE 1.1 – Tableau comparatif des approches étudiées

1.4 Conclusion

Dans ce chapitre, nous avons Identifié le problème que nous allons aborder qui est l'identification des enfants disparus, nous avons pu voir les différentes approches des chercheurs dans le domaine de la reconnaissance faciale, puis nous avons fait une comparaison des méthodes apportées et nous nous sommes inspirés de quelques-unes d'entre elles afin d'élaborer notre solution. Dans le chapitre suivant, nous présenterons un état de l'art sur la reconnaissance faciale.

Chapitre 2

La reconnaissance faciale

2.1 Introduction

La reconnaissance faciale est une discipline captivante de l'intelligence artificielle qui vise à doter les systèmes informatiques de la capacité à identifier et à authentifier des individus en se basant sur leurs caractéristiques faciales uniques. Cette technologie repose sur des concepts fondamentaux tels que la détection de visages, l'extraction de caractéristiques faciales, la création de modèles de reconnaissance et la prise de décision basée sur des données visuelles.

Au cours de cette discussion, nous allons plonger plus en profondeur dans les mécanismes sous-jacents de la reconnaissance faciale, en examinant les techniques de pointe et en mettant l'accent sur des aspects techniques tels que la précision, la vitesse de traitement, la gestion des données (voir Figure 2.1).

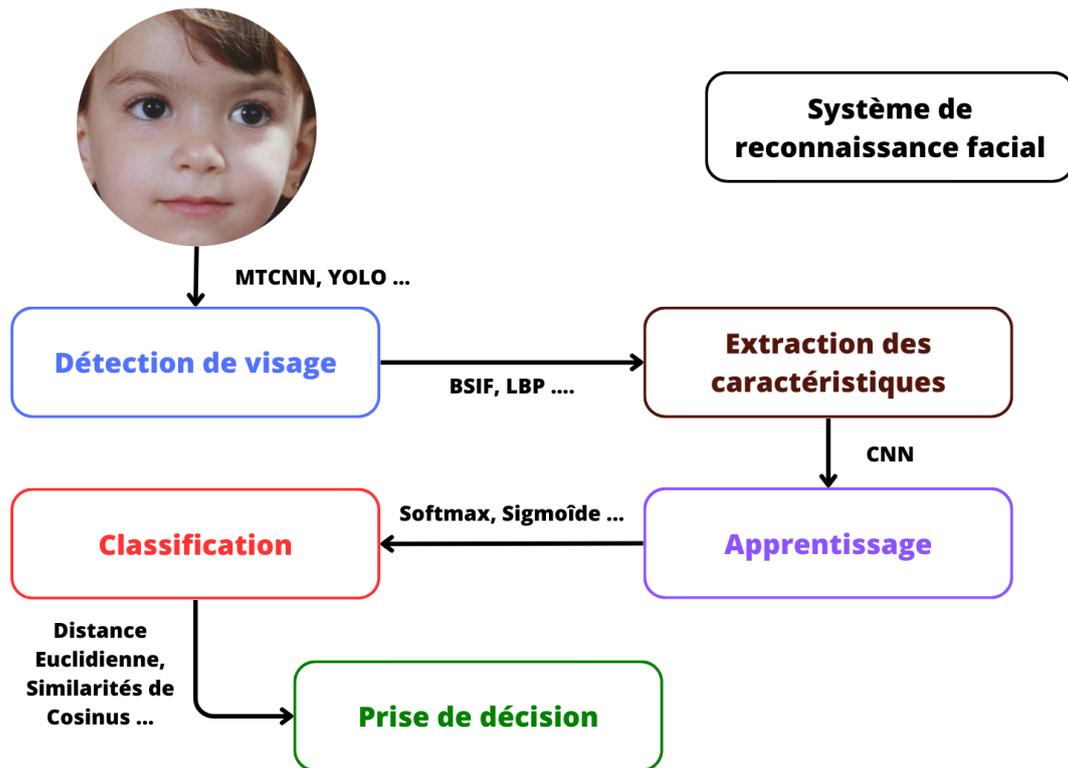


FIGURE 2.1 – Système de reconnaissance faciale

2.2 Détection de visage

La détection de visage est un processus fondamental de l'analyse d'images, qui consiste à localiser la région du visage et ses limites.

Le but de la détection du visage est de déterminer si un visage est présent ou non sur l'image et de localiser son emplacement.

La détection du visage a de très nombreuses applications directes en vidéo surveillance, biométrie, robotique, commande d'interface homme-machine, photographie, indexation d'images et de vidéos, recherche d'images par le contenu, etc. Elle permet également de faciliter l'automatisation complète d'autres processus comme la reconnaissance de visage ou la reconnaissance d'expressions faciales. Parmi les applications directes, la plus connue est sa présence dans de nombreux appareils photo numérique, ou elle sert à effectuer la mise au point automatique sur les visages. C'est également une technique importante pour les interfaces homme-machine évoluées, afin de permettre une interaction plus naturelle entre un humain et un ordinateur [19].

2.2.1 Méthodes de détection de visage

Il existe différentes méthodes pour la détection de visage

Méthode basée sur l'apparence

Cette méthode est une technique de traitement d'image et de vision par ordinateur qui vise à localiser automatiquement les visages dans une image ou une séquence vidéo en se basant principalement sur les caractéristiques visuelles des visages humains. Cette approche utilise des informations visuelles telles que la couleur, la texture, la forme et la disposition des pixels pour identifier et localiser les régions de l'image qui correspondent aux visages.

Viola et Jones : les filtres de Haar

La détection d'objets à l'aide de classificateurs en cascade, basée sur les fonctions de Haar est une méthode efficace de détection d'objet proposée par les chercheurs Paul Viola et Michael Jones [28] la fonction cascade est formée à partir de nombreuses images positives (images de visages) et négatives (images sans visage), qui est ensuite utilisée pour détecter des objets dans d'autres images. La méthode de Viola et Jones est l'une des méthodes les plus appréciées pour sa rapidité de détection et a été largement utilisée, dans des applications telles que la détection de visage en temps réel, sur les caméras de sécurité, les smartphones et les applications de photographie. Elle se démarque des autres méthodes par :

1. l'utilisation d'images intégrées qui permettent de calculer plus rapidement les caractéristiques.
2. la sélection par boosting des caractéristiques.
3. la combinaison en cascade de classificateurs boostés, apportant un gain de temps net d'exécution.

Méthodes basées sur les réseaux de neurones convolutifs CNN Ces méthodes sont des techniques de détection de visages qui reposent sur l'utilisation de réseaux de neurones convolutifs, elle exploitent la capacité des CNN à extraire automatiquement des caractéristiques discriminantes à partir d'images pour localiser et identifier des visages dans des images ou des vidéos.

R-CNN : Région avec CNN

Le modèle de détection de visage R-CNN a été proposé par Ross Girshick et al en 2014 [7], ce modèle se compose de trois modules :

1. Génération de propositions régionales : indépendantes de la catégorie, qui définissent l'ensemble des détections candidates disponibles pour le détecteur.
2. Extraction de caractéristiques : le deuxième module est un grand réseau de neurones convolutions qui extrait un vecteur de caractéristique de longueur fixe de chaque région.
3. Classification et localisation : Le troisième module est un ensemble de SVM linéaires spécifiques à chaque classe.

Le R-CNN atteint une précision moyenne (mAP) de 53,7% sur PASCAL VOC 2010 [7].

Fast R-CNN

En 2015, Ross Girshick [6] a présenté un nouvel algorithme d'apprentissage appelé Fast R-CNN, qui visait à remédier aux limites des méthodes précédentes telles que R-CNN et SPPNet, tout en améliorant leur vitesse et leur précision. Le nom « Fast R-CNN » reflète le fait que cet algorithme peut être entraîné et testé relativement rapidement. La première couche de sortie produit des estimations de probabilité Softmax pour K classes d'objets, ainsi qu'une classe « d'arrière-plan ». La deuxième couche de sortie génère quatre nombres à valeur réelle pour chacune des K classes, représentant les positions affinées des boîtes englobantes.

L'algorithme Fast R-CNN prend en entrée une image entière et un ensemble de propositions d'objets. Cela commence par traiter l'intégralité de l'image à travers plusieurs couches convolutives et une mise en commun maximale, ce qui aboutit à une carte de caractéristiques. Pour chaque proposition d'objet, une couche de regroupement de régions d'intérêt (RoI) extrait un vecteur de caractéristiques de longueur fixe de la carte de caractéristiques. Ces vecteurs de caractéristiques passent ensuite par une séquence de couches entièrement connectées, qui finissent par se diviser en deux couches de sortie. L'algorithme Fast R-CNN a obtenu des résultats impressionnants sur des ensembles de données de référence. Il a obtenu un résultat de 66,1 % sur l'ensemble de données VOC2010 et le meilleur résultat

sur l'ensemble de données VOC12 avec une précision moyenne (mAP) de 65,7 % (et 68,4 % avec des données supplémentaires).

Faster R-CNN

En 2016, Shaoqing Ren et al [30] ont proposé un réseau composé de deux modules. Le premier module est un réseau convolutif profond qui génère une carte de fonctionnalités. Ce module intègre un réseau de propositions de région (RPN) qui prend la carte caractéristique de n'importe quelle taille et génère un ensemble de propositions d'objets rectangulaires, chacune avec un score de précision. Le deuxième module est le détecteur Fast R-CNN, qui utilise les régions proposées comme le montre l'architecture Fast R-CNN.

Le résultat obtenu par ce modèle avec la base de données mscoco (COCO val) avec (mAP) est 41.5% et on COCO test-dev avec (mAP) @[0.5] est 42.7% [30].

G-CNN Détecteur d'objets itératif basé sur une grille

En 2016, Najibi et al [26] ont introduit G-CNN, qui utilise un CNN pour manipuler et ajuster une grille fixée à différentes échelles, allant des cadres de délimitation aux objets. L'objectif principal de G-CNN est de résoudre le problème de la détection d'objets en effectuant une recherche itérative dans l'espace de toutes les boîtes englobantes potentielles. Le processus commence avec G-CNN utilisant une pyramide spatiale composée de boîtes fixes multi-échelles. Le but de l'entraînement du réseau est de lui permettre de relocaliser avec précision ces cases initiales vers les objets présents dans l'image en plusieurs étapes, obtenant ainsi un comportement itératif, crucial pour l'efficacité de l'algorithme. L'évaluation des performances de ce modèle sur la base de données COV 2007 a donné une précision moyenne (mAP) de 57,2 %.

Mask R-CNN

En 2017, Kaiming He et al [12] ont introduit une nouvelle architecture appelée Mask R-CNN, qui s'appuie sur le modèle Faster R-CNN. Ils ont développé la branche existante pour la reconnaissance du cadre de sélection en ajoutant une branche distincte pour prédire simultanément les masques d'objets. Cette branche supplémentaire, appelée branche de masque, utilise un petit réseau entièrement

convolutif (FCN) appliqué à chaque région d'intérêt (RoI). Il prédit un masque de segmentation pixel par pixel, en traitant l'élément manquant dans les modèles Fast/Faster R-CNN. Les résultats obtenus sur la base de données fine MS COCO+ ont présenté une précision moyenne (mAP) de 36,4% lors de la validation lors de l'utilisation de ResNet-50-FPN. Le concept derrière Mask R-CNN est relativement simple. Dans Faster R-CNN, chaque objet candidat est associé à deux sorties : une étiquette de classe et un décalage de boîte englobante. Pour améliorer cela, une troisième branche a été incorporée pour générer le masque de l'objet.

SSH : Détecteur de visage sans tête a une étape

En 2017, Najibi et al [27] ont introduit SSH, une nouvelle technique de détection d'objets visant à réduire le temps d'inférence, à minimiser l'utilisation de la mémoire et à garantir l'invariance d'échelle. Contrairement aux méthodes traditionnelles qui séparent les tâches de détection et de classification du cadre de délimitation, SSH effectue les deux simultanément en utilisant les informations globales extraites des couches convolutives. Remarquablement, SSH a démontré une précision de détection de visage exceptionnelle, même sans avoir besoin d'une « tête » distincte dans son réseau sous-jacent. De plus, SSH est spécifiquement conçu pour être invariant à l'échelle et peut intégrer efficacement des informations contextuelles.

SSD : Détecteur MultiBox à prise unique

L'approche SSD (Single Shot MultiBox Detector) utilise un réseau convolutif à action directe pour générer des cadres de délimitation de taille fixe et des scores pour la détection de classe d'objets dans ces cadres. Ceci est suivi d'une étape de suppression non maximale pour produire les détections finales. Les couches initiales du réseau sont basées sur l'architecture VGG-16, couramment utilisée pour la classification d'images de haute qualité. Cependant, dans l'approche SSD, ces couches sont tronquées avant qu'une couche de classification ne soit appliquée, créant ainsi un réseau de base. Une structure auxiliaire est ensuite ajoutée au réseau pour produire des détections avec des caractéristiques spécifiques, notamment des cartes de caractéristiques multi-échelles, des prédicteurs convolutifs et des boîtes par défaut avec une apparence spécifique.

Le résultat obtenu du modèle SSD512 entraîné sur COCO trainval35k puis affiné en pascal voc 2007+2012 est le meilleur résultat : 81,6% mAP, et 68.0% mAP avec le modèle SSD300 et le résultat obtenu des modèles SSD300 et SSD512 entraînés sur la base de données COCO trainval35k est 41.2% mAP et 46.5% mAP [24].

YOLO : Vous ne regardez qu'une seule fois

C'est un système [29] qui peut être divisé en deux composants principaux : l'extracteur de caractéristiques et le détecteur. Les deux composants opèrent à plusieurs échelles. Lorsqu'une nouvelle image est introduite dans le système, elle subit d'abord une extraction de caractéristiques pour obtenir des intégrations de caractéristiques à trois échelles différentes ou plus.

Ces caractéristiques sont ensuite introduites dans des branches distinctes du détecteur, qui génèrent des boîtes englobantes et fournissent des informations sur les classes. Les couches du réseau YOLO peuvent être classées en fonction de leurs fonctionnalités. Les 20 couches convolutives initiales sont suivies d'une couche de pooling moyenne, et une couche pré-entraînée entièrement connectée est utilisée pour la classification ImageNet, qui se compose de 1000 classes.

La préformation à la classification est réalisée à l'aide d'un ensemble de données avec une résolution de 224x224. De plus, le réseau intègre des couches de réduction 1x1 et des couches convolutionnelles 3x3. Les 4 dernières couches convolutives sont accompagnées de 2 couches complètement connectées pour former le réseau de détection d'objets. Pour capturer des informations plus détaillées, la résolution de l'ensemble de données est augmentée à 448x448 à des fins de détection d'objets. La dernière couche du réseau est responsable de la prédiction des probabilités de classe et des boîtes englobantes [29].

En résumé, l'algorithme YOLO est une approche de détection d'objets largement reconnue et influente. Son architecture unique, inspirée de GoogLeNet, couplée à ses performances impressionnantes sur divers jeux de données, l'a propulsé à l'avant-garde du domaine. Les améliorations continues apportées aux versions ultérieures du modèle ont consolidé le statut de YOLO en tant qu'algorithme de détection d'objets de premier plan. L'architecture de ce réseau s'inspire du modèle GoogLeNet, réputé pour la classification d'images. Pour évaluer ses performances,

l'algorithme YOLO a été testé sur l'ensemble de données de détection COV PASCAL, atteignant une précision moyenne (mAP) impressionnante de 63,4% [29].

Tout au long de son évolution, le modèle YOLO a connu six versions, chaque itération améliorant la précédente en termes de vitesse et de précision.

Dans ce qui suit, nous allons définir la méthode que nous avons utilisée dans notre travail pour la détection de visage : MTCNN

MTCNN : Réseaux convolutifs en cascade multitâches

Cette méthode se distingue avec son intégration des tâches de détection de visage et d'alignement, obtenue grâce à l'utilisation de CNN unifiés en cascade en employant un apprentissage multitâche [37]. Essentiellement, MTCNN se compose de trois réseaux distincts travaillant en harmonie. Le premier réseau, connu sous le nom de Proposal Network (P-Net), génère principalement des fenêtres candidates et leurs vecteurs de régression de boîte englobante correspondants. Il utilise également la suppression non maximale (NMS) pour fusionner des boîtes qui se chevauchent fortement. Le deuxième réseau, appelé Refinement Network (R-Net), agit comme un filtre pour éliminer un grand nombre de faux candidats du P-Net. De plus, il affine le rectangle englobant grâce à des techniques de régression.

Enfin, le troisième réseau, connu sous le nom d'Output Network (O-Net), filtre davantage les faux candidats et produit les fenêtres de candidats finales ainsi que les positions précises de cinq repères faciaux, le tout obtenu grâce à la puissance des réseaux de neurones [20]. Pour améliorer la précision et la robustesse de la détection des visages, MTCNN utilise une technique connue sous le nom de pyramide d'images. Cela implique de redimensionner l'image d'entrée à différentes échelles, créant ainsi une pyramide d'images qui sont ensuite introduites dans le cadre en cascade en trois étapes. Cette approche multi-échelle permet à MTCNN de gérer efficacement des visages de différentes tailles et orientations, ce qui en fait un outil polyvalent et puissant [20]. MTCNN est un cadre très efficace qui a été développé pour relever le défi de la détection des visages. Il s'agit d'un réseau sophistiqué de neurones capables de détecter les visages et d'identifier les repères faciaux tels que les yeux, le nez et la bouche dans les images.

Ce cadre est basé sur un modèle de CNN proposé par Zhang et al [37]. Il a gagné en popularité et il est largement utilisé comme l'un des outils de détection de visage les plus précis disponibles aujourd'hui [20]].

L'objectif principal de cette approche est d'identifier et de localiser avec précision les cadres de délimitation des visages dans une image. De plus, le réseau vise à déterminer les positions précises de cinq points de référence sur le visage, soit les deux points oculaires, la pointe du nez et les deux points du bout de la bouche. Ces points de référence sont communément appelés « points de repère » et leur détection précise est cruciale pour diverses applications d'analyse et de reconnaissance faciale.

2.3 Alignement

L'alignement des visages est une étape cruciale dans les systèmes de reconnaissance faciale, car il permet de normaliser la position et l'orientation des visages dans les images, ce qui facilite leur comparaison. Une fois les visages détectés, on extrait les points de repère faciaux qui décrivent les caractéristiques clés du visage, tels que les yeux, le nez, la bouche, etc. Ces points de repère sont essentiels pour aligner le visage de manière précise. Des techniques telles que le modèle d'apparence active (AAM) ou les réseaux de neurones profonds (MTCNN) sont utilisés pour extraire ces points.

À l'aide des points de repère faciaux, on calcule une transformation géométrique (généralement une homographie ou une transformation affine) qui permet de normaliser la position, l'orientation et l'échelle du visage. Cette transformation aligne le visage de manière à ce qu'il soit centré, que les yeux soient à une distance fixe l'un de l'autre, et que l'orientation soit constante.

Après l'alignement, le visage peut être redimensionné pour avoir une taille constante. Cela garantit que tous les visages ont la même taille, ce qui simplifie la comparaison.

2.4 Extraction des caractéristiques

L'extraction des caractéristiques est une étape essentielle dans le domaine de la vision par ordinateur et de l'apprentissage automatique. Cette étape consiste à extraire des informations significatives à partir de données brutes, telles que des images ou des signaux, afin de les représenter de manière plus appropriée pour une analyse ultérieure. Dans le contexte de la vision par ordinateur, l'extraction des caractéristiques vise à identifier des motifs, des structures ou des propriétés visuelles spécifiques dans une image.

Dans le contexte de la reconnaissance faciale, l'extraction des caractéristiques revêt une importance particulière. Elle consiste à extraire des informations spécifiques du visage d'une personne afin de créer une représentation numérique qui peut être utilisée pour identifier cette personne ultérieurement.

2.4.1 Méthodes d'extraction des caractéristiques

Dans le domaine de l'extraction des caractéristiques, plusieurs méthodes existent :

SIFT : Transformation de caractéristiques invariantes d'échelle

David G. Lowe [25] a introduit la transformation de caractéristiques invariantes d'échelle (SIFT) comme technique utilisée pour analyser les caractéristiques visuelles des images. Cela implique le calcul de descripteurs pour les images SIFT, qui sont des caractéristiques locales qui restent cohérentes malgré les changements d'éclairage, de bruit d'image, de rotation, de mise à l'échelle et de petits changements de point de vue. Cela signifie que les descripteurs SIFT sont invariants par rapport aux différents facteurs qui définissent le contenu visuel d'une image. L'algorithme SIFT, développé en 1999 par David Lowe, comprend cinq étapes principales. Ces étapes comprennent :

1. la détection des extrema dans l'espace de l'échelle.
2. la localisation précise des points clés.
3. l'attribution d'orientations à ces points clés.
4. le calcul de descripteurs pour les points clés.
5. l'établissement de correspondances entre les points clés.

En suivant ces étapes, SIFT fournit une méthode robuste et fiable pour détecter et extraire les caractéristiques clés des images selon la formule(2.1).

$$H(x, y, \theta) = \sum_{p \in R(x, y)} w(p) \cdot \text{Mag}(p) \cdot \delta(\theta - \theta(p)) \quad (2.1)$$

- $H(x, y, \theta)$: Histogramme des orientations des gradients pour le point clé à la position (x, y) avec une orientation θ .
- $R(x, y)$: La région autour du point clé à la position (x, y) .
- $w(p)$: Un poids pour le pixel p dans la région $R(x, y)$.
- $(\text{Mag}(p))$: La magnitude du gradient du pixel p .
- $\theta(p)$: L'orientation du gradient du pixel p .
- $\delta(\theta - \theta(p))$: Une fonction delta qui compte les votes d'orientation dans l'histogramme.

Filtres de Gabor

Introduit par Dennis Gabor, le Filtre de Gabor est un filtre linéaire dont la réponse impulsionnelle est une sinusoïde modulée par une fonction gaussienne (également appelée ondelette de Gabor). Il porte le nom de son inventeur, d'origine hongroise Dennis Gabor (1940). Les filtres de Gabor sont des filtres passe-bande utilisés en traitement d'images pour l'extraction de caractéristiques, l'analyse de texture et l'estimation de disparité stéréo, etc. Les filtres de Gabor sont connus comme un moyen d'analyse espace-fréquence très robuste; tel qu'un ensemble de filtres de Gabor qui est utilisé avec 5 fréquences spatiales et 8 orientations distinctes, ce qui donne 40 filtres de Gabor différents. Cette spécificité a fait des filtres de Gabor un moyen puissant d'analyse de textures et de classification. Les filtres de Gabor analysent la texture d'un objet suivant différentes résolutions et différents angles dans le domaine spatial.

La représentation de Gabor d'une image de visage est obtenue par la convolution de l'image avec la famille des filtres de Gabor. Cette convolution est définie

par $IG(r,o) = I * G(r,o)$. $I * G(r,o)$ est le résultat de la convolution de l'image par le filtre de Gabor à une certaine résolution r et à une orientation o .

DCT : Transformation en cosinus discrète

La méthode de transformation en cosinus discrète (DCT), également connue sous le nom de transformation en cosinus discrète, Le processus lui-même est relativement simple. Chaque image de visage est représentée par un vecteur constitué des coefficients initiaux de la transformée DCT. Lorsqu'un visage est présenté, sa transformée est calculée et un nombre spécifique de coefficients est conservé pour comparaison avec les coefficients de la base de données existante selon la formule(2.2).

De même, pour chaque image de la base de données, sa transformée est calculée à l'aide de la DCT sur l'image normalisée, et les premiers coefficients de la DCT sont extraits pour former un vecteur unifié. Ces représentations sont ensuite enregistrées. Il est important de souligner que le processus d'apprentissage s'effectue indépendamment sur chaque image, l'un des avantages de cette méthode réside dans sa flexibilité permettant d'incorporer facilement de nouvelles images ou individus dans le système [10].

$$DCT(k) = \sqrt{\frac{2}{N}} \cdot C(k) \cdot \sum_{n=0}^{N-1} x[n] \cdot \cos\left(\frac{(2n+1)k\pi}{2N}\right) \quad (2.2)$$

ICA : Analyse en composantes indépendantes

L'Analyse en composantes indépendantes (ou ICA pour Independent Component Analysis) est une généralisation de PCA (Analyse en Composantes Principales). En outre, ICA permet une meilleure caractérisation des données dans un espace à n dimensions et les vecteurs de base trouvés par ICA ne sont pas nécessairement orthogonaux afin qu'ils réduisent également l'erreur de reconstruction. Deux architectures différentes pour l'ICA sont fournies : une première architecture (ICA I) qui construit une base d'images statistiquement indépendante et une

deuxième architecture (ICA II) qui fournit une représentation en code factoriel des données [14].

LPQ : La quantification de la phase locale

La quantification de la phase locale ou le descripteur LPQ a été désigné pour la première fois par Ojansivu et Heikkilä pour l'utiliser dans la classification de textures pour les images floues [1].

Le descripteur LPQ est construit de façon à ne retenir dans une image que l'information locale invariante à un certain type de flou. LPQ est basée sur le calcul de la transformée de Fourier à court terme (STFT : short term Fourier transform) sur la fenêtre d'image locale. À chaque pixel, le coefficient local de Fourier est calculé pour quatre points de fréquence. Ensuite, les signes de la partie réelle et imaginaire de chaque coefficient sont quantifiés à l'aide d'un quantificateur scalaire binaire, afin de calculer l'information de phase. Le coefficient binaire de huit bits qui en résulte est ensuite représenté sous forme d'entiers par codage binaire. Finalement, un vecteur de caractéristiques à 256 dimensions est obtenue [1].

BSIF : caractéristiques des images statistiques binarisées

BSIF est une méthode proposée par Kannala et Rahtuen en 2012 [16] pour construire des descripteurs d'images locaux qui codent efficacement les informations de texture et conviennent à la représentation basée sur un histogramme de régions d'image. Le procédé calcule un code binaire pour chaque pixel en projetant linéairement des zones d'image locales sur un sous-espace, dont les vecteurs de base sont appris à partir d'images naturelles via une analyse de composants indépendante, et en binarisant les coordonnées dans cette base via un seuillage. La longueur de la chaîne de code binaire est déterminée par le nombre de vecteurs de base. Les régions d'image peuvent être facilement représentées par des histogrammes des codes binaires des pixels.

La fonction correspondante est définie comme suit :

Dans ce qui suit, nous allons définir les méthodes que nous avons utilisées dans notre travail pour l'extraction des caractéristiques : LBP et HOG

LBP : Modèles binaires locaux

L'opérateur LBP, qui signifie « Local Binary Patterns », a été suggéré pour la première fois par Ojala et al en 1995 [11]. Il s'agit d'un moyen de décrire la texture d'une image. Il a été utilisé dans de nombreux systèmes de reconnaissance faciale pour aider à identifier et à analyser différentes parties d'un visage. Il est populaire, car il est à la fois puissant et facile à utiliser dans les programmes informatiques. Il peut également être utilisé dans d'autres domaines comme l'imagerie médicale, l'analyse des mouvements et des activités et la recherche d'images ou de vidéos dans une base de données. De nombreux chercheurs ont trouvé que la LBP était très utile pour la reconnaissance faciale. Il ne s'agit pas simplement d'un moyen simple de décrire la texture, mais cela a par ailleurs ouvert de nouvelles possibilités pour l'étude d'images et de vidéos à l'aide de motifs binaires.

Le concept du LBP est simple, il propose d'attribuer un code binaire à un pixel en fonction de son voisinage selon la formule(2.3). Ce code d'écriture de la texture locale d'une région est calculé par seuillage d'un voisinage avec le niveau de gris du pixel central selon la formule(2.4).

$$LBP_{P,R}(x_c, y_c) = \sum_{n=0}^{P-1} s(g_n - g_c) \cdot 2^n \quad (2.3)$$

$$s(x) = \begin{cases} 1, & \text{si } x \geq 0 \\ 0, & \text{sinon} \end{cases} \quad (2.4)$$

Pour générer un motif de 0 et de 1, un grand nombre de variations a été développée pour l'amélioration des performances LBP. Les avantages de ces variations sont la rapidité et une occupation de mémoire faible.

HOG : Histogramme des dégradient orientés

Dans le but d'améliorer les performances de la reconnaissance et de la détection des visages, les chercheurs explorent de nouvelles méthodes d'extraction de caractéristiques. L'une de ces techniques est HOG [3], qui vise à représenter les

visages en analysant les gradients présents dans l'image grâce à une analyse d'histogramme. HOG calcule des histogrammes locaux des orientations de gradient sur une grille dense, uniformément répartie sur l'image. Cette méthode partage des similitudes avec les techniques SIFT (Scale-Invariant Feature Transform). Plus précisément, la région dans laquelle le visage est détecté est divisée en blocs de tailles égales, eux-mêmes subdivisés en cellules. Chaque cellule subit une analyse des gradients de pixels pour générer un histogramme de gradient à neuf bandes. Dans la littérature, diverses techniques permettant de combiner ces histogrammes en vecteurs ont été proposées.

2.5 Réduction de dimension

la réduction de dimension est une technique essentielle en traitement de données et en apprentissage automatique, utilisée pour réduire le nombre de variables ou de caractéristiques dans un ensemble de données tout en préservant autant que possible les informations pertinentes. Cette approche est particulièrement utile lorsqu'on traite avec des ensembles de données de grande dimension, car elle peut simplifier les données, améliorer la visualisation, réduire le bruit et accélérer les algorithmes d'apprentissage automatique.

Dans le domaine de la reconnaissance faciale, la réduction de dimension joue un rôle crucial les techniques de réduction de dimension peuvent aider à extraire les caractéristiques les plus discriminatives du visage, telles que les contours, les textures, les formes et les points clés, tout en réduisant la dimension de la représentation des visages. Cela permet d'obtenir des systèmes de reconnaissance faciale plus performants et plus efficaces.

2.5.1 Méthodes de réduction de dimension

dans ce domaine plusieurs techniques sont utilisées :

PCA : Analyse en Composantes Principales

La PCA ou Analyse en Composantes Principales (ACP) en français est une technique d'analyse de données développée pour réduire la dimensionnalité d'un ensemble de données tout en préservant ses caractéristiques importantes.

L'objectif principal de la PCA est d'identifier les directions (ou axes) dans lesquelles les données varient le plus et de les utiliser pour projeter les données originales dans un nouvel espace de dimensions réduites.

L'auteur de la PCA est Karl Pearson en 1901, un statisticien britannique. Turk et Pentland [35] ont introduit cette méthode dans la reconnaissance faciale, elle consiste à identifier les principales composantes d'un ensemble de visages. Ce processus détermine les vecteurs propres de la matrice de covariance formée par toutes les images échantillons. En utilisant une combinaison de ces vecteurs propres.

LDA : Analyse discriminante linéaire

LDA est une méthode de réduction de dimensionnalité et de classification [5]. Elle est utilisée pour trouver une combinaison linéaire des caractéristiques originales qui maximise la séparation entre différentes classes dans un ensemble de données. LDA est souvent utilisée en reconnaissance de motifs et en classification pour améliorer la performance des algorithmes de classification.

En reconnaissance faciale, LDA effectue une véritable séparation de classe, et cela, en minimisant les variations entre les images d'un même individu tout en maximisant les variations entre les images d'individus différents. Néanmoins, lorsque le nombre d'individus est inférieur à la résolution de l'image, il est difficile d'appliquer la LDA qui peut faire apparaître des matrices de dispersions singulières. Afin de contourner ce problème, certains algorithmes basés sur la LDA ont été proposées, le plus connu est la RLDA (Regularized LDA).

2.6 Apprentissage profond : Deep Learning

L'apprentissage profond, également connu sous le nom de Deep Learning (DL), est une sous-discipline de l'intelligence artificielle (IA) qui vise à modéliser des données complexes et à résoudre des tâches d'apprentissage automatique en utilisant des réseaux de neurones artificiels profonds.

L'apprentissage profond est une approche d'apprentissage automatique basée sur l'idée de créer des modèles informatiques composés de nombreuses couches de neurones artificiels interconnectés, formant ainsi un réseau neuronal profond. Ces

réseaux sont conçus pour apprendre de manière autonome à partir de données brutes, en identifiant des structures et des caractéristiques complexes dans les données.

L'apprentissage profond, également connu sous le nom de Deep Learning, est une sous-discipline de l'intelligence artificielle (IA) qui vise à modéliser des données complexes et à résoudre des tâches d'apprentissage automatique en utilisant des réseaux de neurones artificiels profonds.

2.6.1 Principales caractéristiques de l'apprentissage profond

Les principales caractéristiques de l'apprentissage profond sont les suivantes

1. Réseaux de neurones profonds : Les réseaux de neurones profonds sont le pilier de l'apprentissage profond. Ils sont composés de nombreuses couches (d'où le terme "profond") qui permettent au modèle d'apprendre des représentations hiérarchiques des données.
2. Représentations de niveau supérieur : L'apprentissage profond vise à extraire automatiquement des caractéristiques de niveau supérieur à partir des données brutes, ce qui signifie que le modèle peut apprendre des caractéristiques abstraites et hiérarchiques.
3. Entraînement basé sur les données : L'apprentissage profond nécessite de grandes quantités de données pour entraîner efficacement les modèles. Plus les données d'entraînement sont diverses et représentatives, meilleures sont les performances du modèle.
4. Utilisation de fonctions d'activation non linéaires : Les fonctions d'activation non linéaires, telles que ReLU (Rectified Linear Unit) sont couramment utilisées dans les réseaux de neurones profonds pour introduire de la non-linéarité et améliorer la capacité de modélisation.
5. Apprentissage supervisé et non supervisé : L'apprentissage profond peut être utilisé à la fois pour des tâches supervisées (classification, régression, etc.) et non supervisées (clustering, réduction de dimension, etc.).
6. Applications diverses : L'apprentissage profond a trouvé des applications dans de nombreux domaines, notamment la vision par ordinateur, le trai-

tement du langage naturel, la reconnaissance vocale, la recommandation de contenu, la robotique, et bien d'autres.

7. Outils et frameworks : Les programmeurs en IA utilisent souvent des frameworks tels que TensorFlow, PyTorch et Keras pour développer, entraîner et déployer des modèles d'apprentissage profond.

En résumé, l'apprentissage profond est une approche puissante qui a révolutionné de nombreux domaines de l'IA en permettant aux machines d'apprendre des représentations de données complexes de manière autonome. Cette technologie est à la base de nombreuses avancées récentes dans l'IA et continue d'évoluer rapidement.

2.6.2 Les réseaux de neurones

Les réseaux de neurones, également connus sous le nom de réseaux de neurones artificiels (RNA) ou simplement de neurones, sont des structures de données et des modèles mathématiques inspirés du fonctionnement des neurones biologiques. En tant que programmeur en IA, voici comment je décrirais les réseaux de neurones.

Un réseau de neurones est un modèle computationnel composé de plusieurs unités de base appelées "neurones" interconnectées. Chaque neurone effectue des opérations mathématiques sur les données qu'il reçoit et transmet le résultat à d'autres neurones. Les neurones sont organisés en couches, généralement en trois types de couches : couche d'entrée, couches cachées et couche de sortie.

Quelques réseaux de neurones existants

Il existe plusieurs réseaux de neurones, chacun a son domaine d'application (voir Table 1.2) :

Réseaux de neurones	Domaines d'application
Réseaux de neurones convolutifs (CNN)	Classification d'images
Réseaux de neurones récurrents (RNN)	Traitement du langage naturel (NLP)
Réseaux de neurones bidirectionnels (Bi-RNN)	Reconnaissance de la parole
Réseaux de neurones générateurs adversaires (GAN)	Génération d'images et de vidéos
Réseaux de neurones autoencodeurs (AE)	Compression d'images, débruitage d'images
Réseaux de neurones siamois	Vérification de similarité

TABLE 2.1 – Tableau de quelques réseaux de neurones et leurs domaines d'application

2.6.3 CNN : Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs, abrégés en CNN ou ConvNets, sont une classe de réseaux de neurones artificiels spécialement conçus pour le traitement efficace des données structurées en grille, telles que les images et les vidéos. Les CNN sont largement utilisés en vision par ordinateur pour des tâches telles que la classification d'images, la détection d'objets, la segmentation sémantique et d'autres applications liées à la reconnaissance visuelle (voir Figure 2.2).

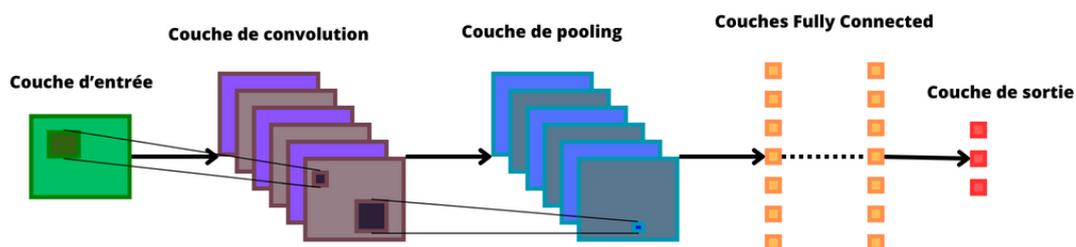


FIGURE 2.2 – Architecture d'un CNN

2.6.4 Principales caractéristiques des CNN

Les principales caractéristiques des CNN sont les suivantes :

1. Opération de convolution : L'opération de convolution est au cœur des CNN.

Elle consiste à appliquer un filtre (ou noyau) de convolution à l'image d'entrée. Le filtre est une petite matrice de poids qui glisse sur l'image en effectuant une multiplication entre ses valeurs et les pixels de l'image sous-jacente. L'opération de convolution permet de détecter des motifs locaux tels que des bords, des textures et des contours dans l'image. En appliquant plusieurs filtres de convolution, le réseau peut extraire des caractéristiques hiérarchiques et complexes à partir de l'image.

2. Dans un CNN, les filtres de convolution sont partagés entre différentes parties de l'image. Cela signifie que le même filtre est utilisé pour analyser différentes régions de l'image. Le partage de poids permet au modèle de généraliser efficacement les caractéristiques apprises à l'ensemble de l'image, ce qui réduit le nombre de paramètres à entraîner.
3. Couches de sous-échantillonnage (pooling) : Les CNN utilisent des couches de sous-échantillonnage (ou pooling) pour réduire la dimension spatiale des données tout en préservant les caractéristiques importantes. La couche de pooling réduit la résolution de l'image en agrégeant l'information à partir de régions voisines. La max-pooling est une technique courante où seule la valeur maximale dans chaque région est conservée, tandis que les autres valeurs sont ignorées. Cela permet de réduire la complexité du modèle tout en maintenant les caractéristiques clés.
4. Couches de neurones entièrement connectées : Après plusieurs couches de convolution et de sous-échantillonnage, les CNN peuvent inclure des couches de neurones entièrement connectées. Ces couches sont similaires à celles des réseaux neuronaux traditionnels et sont utilisées pour la classification ou d'autres tâches spécifiques. Les informations extraites des couches de convolution sont acheminées vers les couches entièrement connectées pour prendre des décisions finales, telles que la classe d'une image.
5. Fonctions d'activation : Les CNN utilisent généralement des fonctions d'activation non linéaires pour introduire de la non-linéarité dans le modèle. La fonction d'activation ReLU (Rectified Linear Unit) est largement utilisée en raison de sa simplicité et de son efficacité. Elle remplace toutes les valeurs négatives par zéro. L'ajout de non-linéarités permet au modèle de capturer des

relations complexes entre les caractéristiques et de modéliser des fonctions plus complexes.

Les CNN ont démontré une grande efficacité dans la reconnaissance visuelle et ont contribué de manière significative à des avancées importantes dans des domaines tels que la détection d'objets, la reconnaissance faciale, la vision par ordinateur médicale, etc. Ils sont devenus un élément essentiel de l'arsenal d'outils en intelligence artificielle pour le traitement des données visuelles.

Quelques réseaux de neurones convolutifs existants

Il existe plusieurs réseaux de neurones convolutifs, chacun a son domaine d'application (voir Table 1.3) :

Réseaux de neurones	Domaines d'application	Auteur	Année
LeNet	Reconnaissance de caractères manuscrits	Yann LeCun.al [18]	1998
AlexNet	Classification d'images	Alex Krizhevsky.al [17]	2012
VGGNet	Classification d'images	Karen Simonyan.al [32]	2014
GoogLeNet (Inception-Net)	Classification d'images, détection d'objets	Christian Szegedy.al [33]	2015
U-Net	Segmentation d'images médicales	Olaf Ronneberge.al [31]	2015
YOLO	Détection d'objets en temps réel	Joseph Redmon.al [29]	2016
MobileNet	Vision par ordinateur sur appareils mobiles et embarqués	Andrew G Howard.al [13]	2017
EfficientNet	Classification d'images	Mingxing Tan.al [34]	2019

TABLE 2.2 – Tableau de quelques CNN et leurs domaines d'application

Ces exemples illustrent comment différentes architectures de CNN sont adaptées à des domaines d'application spécifiques, de la reconnaissance de caractéristiques, à la détection d'objets en passant par la segmentation d'images médicales. Les CNN ont joué un rôle crucial dans de nombreuses avancées en vision par ordinateur et en traitement d'images.

2.7 Prise de décision

Lorsque de nouvelles données sont présentées au modèle, il applique la fonction de décision apprise pendant l'entraînement pour prendre une décision ou faire une prédiction. Cette décision peut être binaire (oui/non), une classification parmi plusieurs classes, une valeur numérique, etc., en fonction du type de problème.

Il existe différents types de décisions que les modèles d'apprentissage automatique peuvent prendre en fonction du type de problème auquel ils sont confrontés. Voici quelques-uns des types de décisions courants :

2.7.1 La classification

La classification est un domaine clé de l'apprentissage automatique où l'objectif principal est de catégoriser les données en classes ou en catégories prédéfinies.

Types de classification

1. Classification binaire : Il s'agit de la classification en deux catégories, souvent "positif" et "négatif". Par exemple, la détection de spam dans les e-mails est un problème de classification binaire.
2. Classification multi-classes : Dans ce cas, il y a plus de deux classes possibles. Par exemple, la classification d'images en tant que "Homme", "Femme" ou "Enfant" est un problème de classification multi-classes.
3. Classification multi-étiquettes : Les exemples peuvent appartenir à plusieurs classes simultanément. Par exemple, la catégorisation de documents qui peuvent appartenir à plusieurs catégories thématiques.

2.7.2 La régression

La régression est un autre aspect important de l'apprentissage automatique qui vise à prédire une valeur numérique continue plutôt que de catégoriser les données.

Types de régression

1. Régression linéaire : Elle est utilisée pour modéliser la relation linéaire entre une ou plusieurs variables d'entrée et une variable cible continue.
2. Régression polynomiale : Elle étend la régression linéaire en utilisant des polynômes pour ajuster des relations plus complexes.
3. Régression logistique : Bien qu'elle porte le nom de "logistique", elle est principalement utilisée pour la classification binaire. Cependant, elle peut également être utilisée pour la régression logistique en ajustant une courbe sigmoïde à des données.

Il est important de choisir la méthode de classification ou de régression appropriée en fonction de la nature de nos données et de notre problème spécifique, de bien comprendre les métriques d'évaluation, et d'optimiser les modèles pour obtenir les meilleures performances possibles. De plus, la validation croisée et la gestion de sur-apprentissage (overfitting) sont des aspects essentiels à prendre en compte lors de la création de modèles de classification ou de régression robustes.

2.8 Evaluation

L'évaluation dans le domaine de la reconnaissance faciale revêt une importance capitale, car elle permet de mesurer l'efficacité et la fiabilité des systèmes déployés. Ce processus complexe débute par une phase cruciale de collecte de données d'évaluation, sélectionnées avec soin pour qu'elles reflètent fidèlement la variabilité que l'on retrouvera dans des situations réelles. Une fois ces données obtenues, elles sont réparties en deux ensembles distincts : l'ensemble d'entraînement, utilisé pour former le modèle de reconnaissance faciale, et l'ensemble de test, réservé pour l'évaluation proprement dite.

Dans cette démarche, il est impératif de garantir que les données de test soient pré-traitées de la même manière que celles de l'ensemble d'entraînement, assurant ainsi une égalité lors de l'évaluation. Le choix des métriques de performance est une étape stratégique qui déterminera la qualité de l'analyse des résultats. Ces métriques peuvent être à la fois qualitatives, où l'inspection visuelle des résultats joue un rôle essentiel et quantitatif, basées sur des calculs précis.

L'analyse des résultats permet de dégager des tendances et des motifs, mettant en lumière les forces et les faiblesses du système. Si les performances ne sont pas

satisfaisantes, une étape cruciale consiste à ajuster les paramètres du modèle ou à envisager une révision plus profonde de l'approche adoptée. Une évaluation dans des conditions réelles, notamment en utilisant des caméras en temps réel dans des environnements variables, est souvent nécessaire pour valider les capacités du système.

L'évaluation continue et l'amélioration sont des aspects essentiels pour garantir que la reconnaissance faciale réponde aux exigences de précision et de sécurité dans des contextes variés.

2.9 Conclusion

Dans ce chapitre, nous avons présenté un système global de la reconnaissance faciale, puis détailler son état de l'art en présentant les différentes méthodes et outils dans le domaine. Dans le chapitre suivant, nous présenterons le système proposé.

Chapitre 3

Systeme d'identification des enfants disparus malgré l'avancement dans l'age

Le vieillissement des enfants disparus est un sujet complexe et délicat. Alors que les enquêteurs s'efforcent de les retrouver, le temps passe, modifiant leurs caractéristiques physiques et posant ainsi un défi supplémentaire à la reconnaissance et à l'identification. Cependant, malgré les avancées technologiques et les méthodes novatrices, il existe aujourd'hui très peu de travaux qui relèvent ce défi pour réunir les familles séparées depuis longtemps.

Ce chapitre se penchera sur notre système d'identification des enfants disparus malgré l'Avancement dans l'age, une approche qui combine la puissance de la technologie, de l'apprentissage automatique et de la reconnaissance faciale pour tenter de résoudre ce problème complexe. Nous explorerons les méthodes, les défis et les opportunités que cette approche offre,

3.1 Description du système proposé

Notre système se fait en deux étapes :

3.2 Apprentissage de notre système :

cette étape commence avec la détection et l'alignement de visage avec la méthode MTCNN de deux images d'un enfant a deux ages différent suivi d'une extraction de caractéristiques avec les descripteurs LBP et HOG qui génèrent chacun un vecteur de caractéristique, en suite les vecteurs de caractéristiques

LBP et HOG sont concaténés pour chaque image séparément, les deux vecteurs résultants sont concaténés et on leur applique la PCA suivi de la LDA pour la réduction de dimension et un CNN d'une Architecture standard sera appliqué pour avoir un vecteur de caractéristique Finale, ce vecteur (voir Figure 3.1).

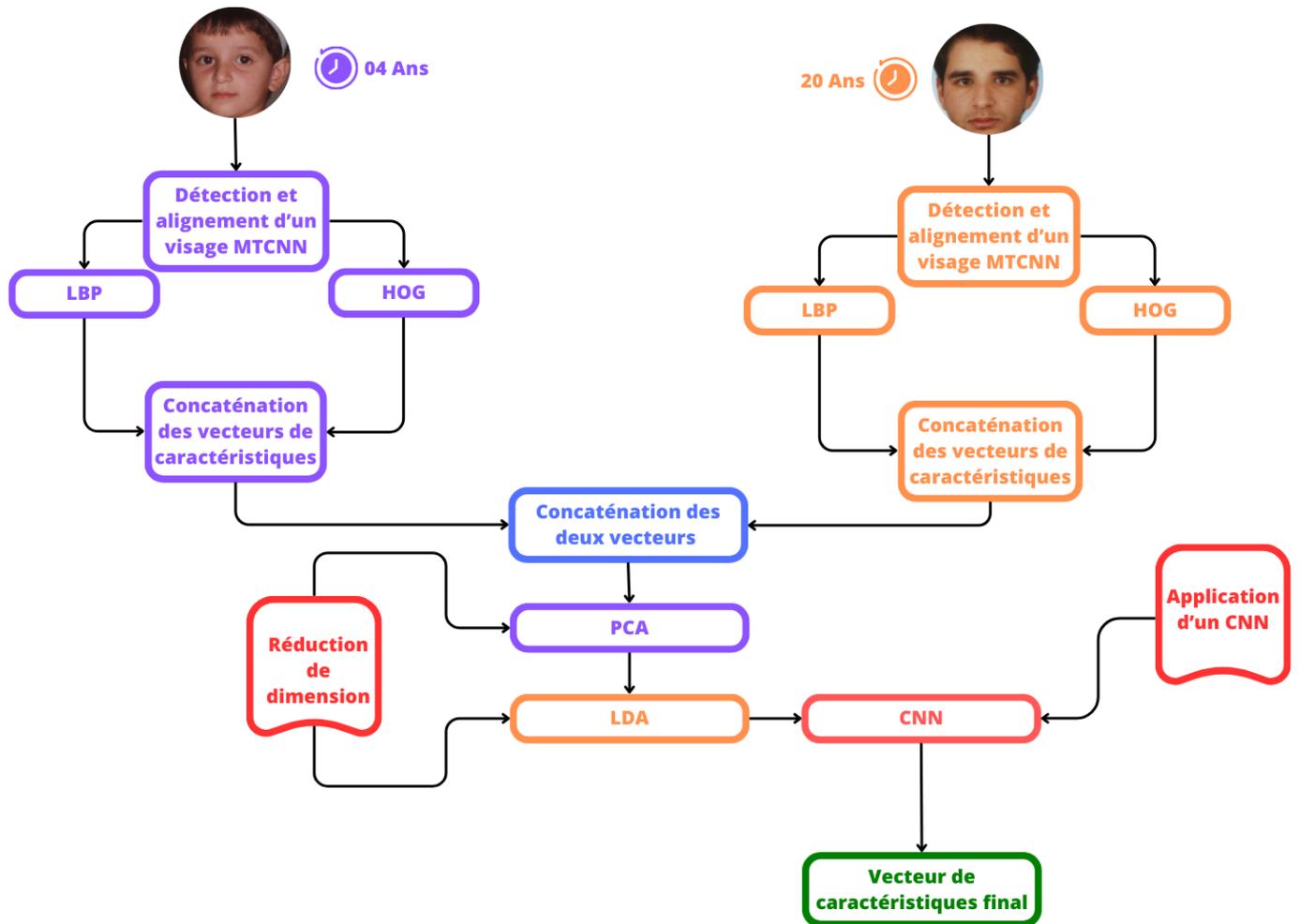


FIGURE 3.1 – Système d'identification d'enfants - Partie Apprentissage

3.3 Test de notre système :

l'image de test subira les mêmes opérations que les images de l'apprentissage, voir la détection et l'alignement, l'extraction des caractéristiques, la réduction de dimension pour avoir un vecteur de caractéristique qui sera comparé avec le vecteur caractéristique Finale résultant de l'étape de l'apprentissage en utilisant la Distance Euclidienne (voir Figure 3.2).

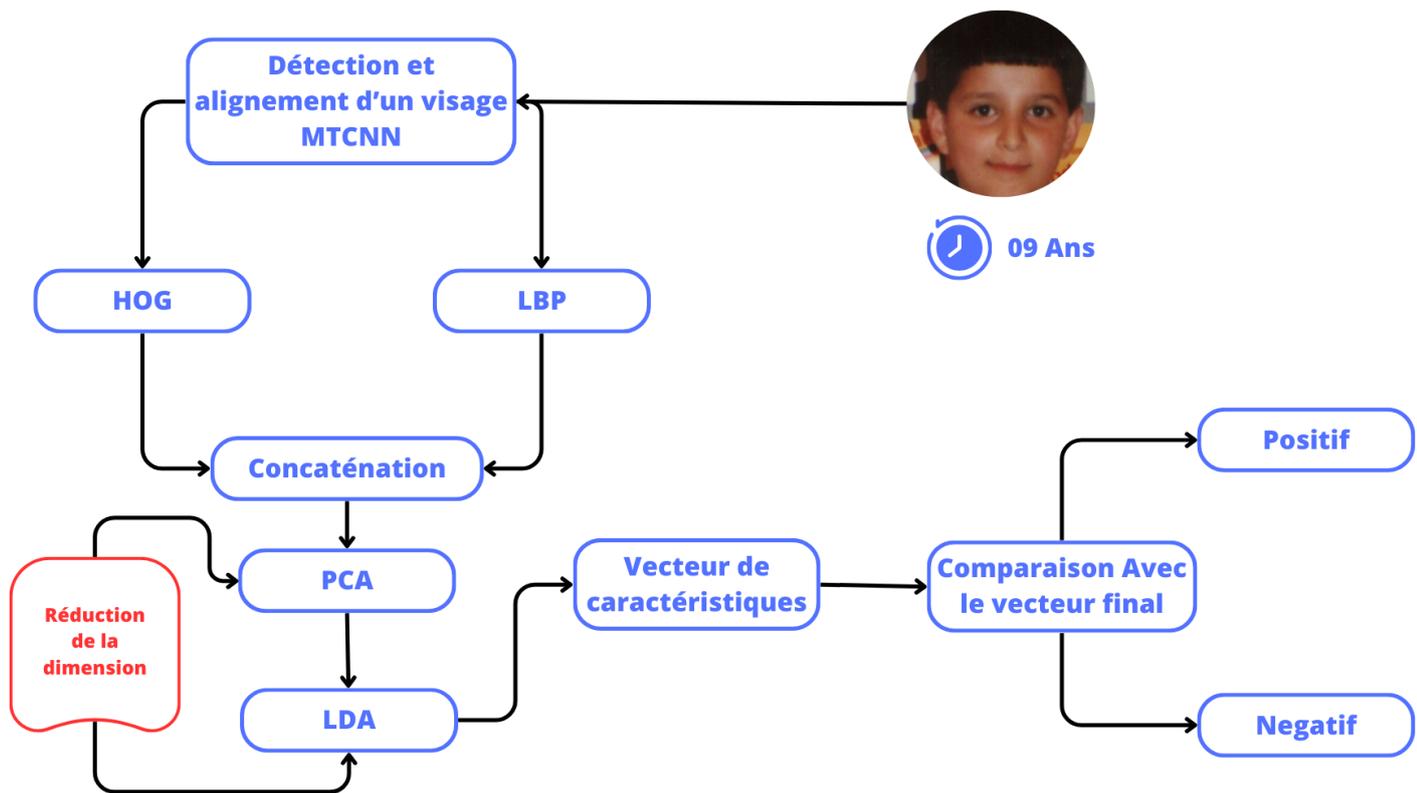


FIGURE 3.2 – Système d’identification d’enfants - Partie Test

3.4 Préparation de la base de données

La préparation de la base de données pour l’identification et la recherche d’enfants disparus, même des années après leur disparition, est une étape cruciale dans le domaine de l’apprentissage automatique et de la reconnaissance faciale. Elle consiste à rassembler, organiser et traiter les données nécessaires pour former et évaluer un modèle de reconnaissance faciale pour permettre au système de fonctionner de manière efficace et précise. Pour construire notre système nous avons utilisé la base de données à accès public FGNET qui a été partagée en deux classes Positive et Négative. Chaque classe est partagée en données d’entraînement, de validation et de test.

3.5 Méthodes utilisées

Nous avons sélectionné plusieurs méthodes pour construire notre système, ces méthodes ont fait leurs preuves des beaucoup de recherches précédentes. Citons-les comme suit ;

3.5.1 Détection de visage

C'est le processus automatisé de localisation et de repérage des visages humains dans des images numériques ou des séquences vidéo. Cette étape vise à détecter les régions de l'image qui correspondent à des visages humains et à les encadrer ou les marquer pour une identification ultérieure.

La détection de visage repose souvent sur des techniques d'apprentissage automatique, telles que les réseaux de neurones convolutifs (CNN), qui ont la capacité d'apprendre des caractéristiques faciales discriminantes à partir de grands ensembles de données d'entraînement. Parmi ses techniques, nous avons choisi les Réseaux convolutifs en cascade multitâches (MTCNN).

MTCNN : Réseaux convolutifs en cascade multitâches

Également connu sous le nom (Multi-task Cascaded Convolutional Networks), est un modèle de détection et d'alignement développé en 2016 [37]. L'objectif principal de ce modèle est de détecter les boîtes englobantes des visages, ainsi que d'identifier cinq des repères spécifiques sur le visage. Ce processus de détection s'effectue en trois étapes distinctes, comme le montre la figure. Dans un premier temps, un réseau CNN superficiel propose plusieurs fenêtres candidates pour une analyse plus approfondie. Ces fenêtres sont ensuite affinées dans un deuxième temps par un réseau CNN plus complexe, qui vise à éliminer un grand nombre de fenêtres ne contenant pas de visage. Enfin, dans la troisième étape, un puissant CNN est utilisé pour affiner davantage les résultats et afficher avec précision les positions des repères faciaux (voir Figure 3.3).

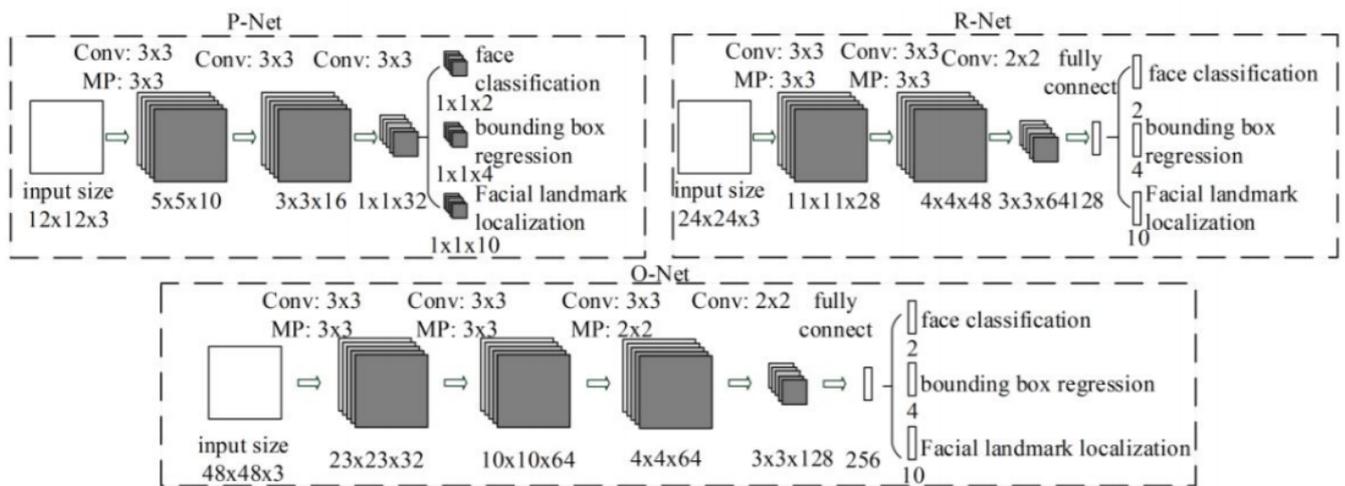


FIGURE 3.3 – Architecture du réseau en cascade MTCNN [37]

Les étapes de fonctionnement de MTCNN son comme suit :

Étape du bloc P-Net (Proposal Network)

La première étape du processus consiste à utiliser un réseau entièrement convolutif connu sous le nom de réseau de propositions (P-Net) pour identifier les régions proposées et déterminer les limites de leurs vecteurs de régression en boîte. Ces vecteurs de régression sont ensuite utilisés pour ajuster et affiner les régions. Pour résoudre le problème des régions fortement superposées, une technique appelée suppression non maximale (NMS) est appliquée pour fusionner ces régions. Le principe sous-jacent de cette étape est de créer une pyramide d'images à partir de l'image d'entrée, ce qui permet de détecter des visages de différentes tailles. Essentiellement, plusieurs copies redimensionnées de la même image sont générées pour rechercher des visages de différentes tailles dans l'image. Chaque copie mise à l'échelle de l'image subit un processus de recherche utilisant un filtre de noyau d'une taille de 12×12 pour identifier les visages [37] (voir Figure 3.4).

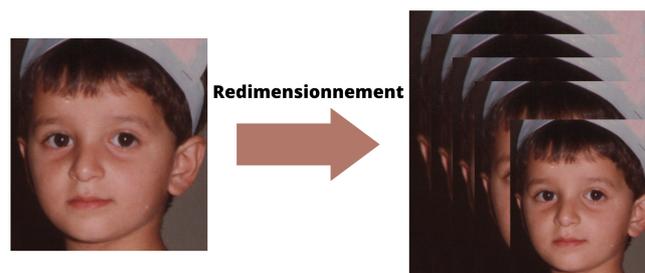


FIGURE 3.4 – Image organisée en pyramide

Le traitement commence par l'examen d'une région spécifique de l'image, située dans le coin supérieur gauche et s'étendant des coordonnées $(0,0)$ à $(12,12)$. Cette région est ensuite envoyée au bloc P-Net, où elle est analysée pour détecter la présence d'un visage. Si un visage est détecté, le bloc P-Net renvoie les coordonnées d'un cadre de délimitation. Le processus est ensuite répété avec les régions suivantes, chacune décalée d'un pas de deux pixels vers la droite ou vers le bas par rapport à la région précédente. En utilisant cette taille de pas, le temps de traitement est réduit. Cependant, un inconvénient de cette approche est que tous les indices liés doivent être recalculés à chaque étape de déplacement. Néanmoins, la méthode MTCNN gère efficacement ce problème en permettant la détection et l'alignement des visages à l'aide de l'étape de déplacement. Dans l'ensemble, la méthode MTCNN offre un moyen fiable de détecter et d'aligner les visages [37] (voir Figure 3.5).



FIGURE 3.5 – Fenêtre glissante

L'utilisation de noyaux plus petits permet la détection de visages plus petits dans des images d'entrée plus grandes. À l'inverse, des noyaux plus grands sont utilisés pour détecter des visages plus grands dans des images de plus petite taille. Pour ce faire, plusieurs copies de l'image de différentes tailles sont générées et introduites dans le réseau neuronal initial, P-Net. Ces images de différentes tailles convergent en sortie du réseau pour un traitement ultérieur. Les poids et biais de P-Net sont entraînés pour identifier et sélectionner avec précision les caractéristiques au sein de chaque noyau (12×12) , ce qui aboutit à un cadre de détection plus précis ([37]).

Cependant, le réseau présente différents niveaux de confiance selon les différentes boîtes. Par conséquent, il est essentiel d'analyser les résultats du P-Net afin d'obtenir une liste de niveaux de confiance pour chaque boîte englobante. Les

cadres avec des niveaux de confiance inférieurs, indiquant une incertitude quant à la présence d'un visage, sont supprimés. Une fois les boîtes avec le plus haut niveau de confiance sélectionné, les coordonnées des boîtes englobantes sont normalisées pour correspondre à celles de l'image originale non mise à l'échelle. Puisque la plupart des noyaux sont présents dans une image réduite, leurs coordonnées sont basées sur l'image plus petite. Cependant, il existe encore un nombre important de cadres de délimitation qui ont tendance à se chevaucher. Pour résoudre ce problème, la méthode de suppression non maximale (NMS) est utilisée pour réduire le nombre de boîtes englobantes. Le processus NMS consiste à trier les boîtes englobantes et leurs noyaux (12x12) correspondants en fonction de leur confiance ou de leur score. Dans certains modèles, le système NMS utilise le plus grand boîtier contenant un objet plutôt que de s'appuyer sur celui bénéficiant de la plus grande confiance sur le réseau.

Ensuite, des calculs sont effectués pour déterminer la surface de chaque noyau et la zone de chevauchement entre chaque noyau et celui qui a le score le plus élevé. Tous les noyaux chevauchant le noyau haute performance sont éliminés. En fin de compte, le traitement NMS génère une liste de cadres de délimitation restants. Ce processus NMS est effectué pour chaque image mise à l'échelle, puis répété avec tous les noyaux survivants de chaque échelle. Cela permet d'éliminer les cadres de délimitation redondants, permettant à la recherche de se concentrer sur un seul cadre précis par visage ([37]) (voir Figure 3.6).

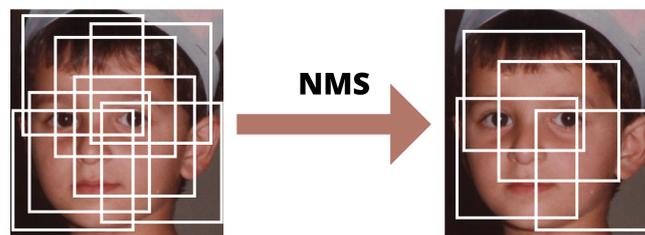


FIGURE 3.6 – Analyse la sortie P-Net

Étape du bloc R-Net (Reading Network)

Après le bloc P-Net, les candidats sont envoyés vers le R-Net, qui est un autre CNN. Le R-Net filtre en outre les faux candidats et affine les positions du cadre de sélection à l'aide de techniques de régression et de NMS. Il vise à améliorer

la précision de la détection des visages. Dans certains cas, une image peut ne capturer qu'une partie du visage, notamment si elle est observée depuis le côté du cadre. Dans de tels cas, le réseau peut fournir une zone de sélection partielle qui s'étend au-delà des limites de la trame. Pour traiter chaque cadre de délimitation, un tableau de même taille est créé. Les valeurs de pixels, correspondant à l'image dans la zone de sélection, sont ensuite copiées dans ce nouveau tableau. Si le cadre de délimitation s'étend au-delà des bords de l'image, seule la partie de l'image à l'intérieur du cadre de délimitation est copiée et la partie restante du tableau est remplie de zéros pour conserver la taille du tableau. Une fois que toutes les boîtes englobantes sont remplies, elles sont redimensionnées à une dimension standard de (24x24) pixels et normalisées pour avoir des valeurs comprises entre -1 et 1. Cette étape de normalisation de l'entrée aide à donner de meilleures performances [37].

Une fois que tous les tableaux d'images ont été redimensionnés à une dimension de (24x24), ce qui est égal au nombre de cadres de délimitation qui ont survécu à l'étape précédente, ces cadres sont mis à l'échelle et normalisés dans les noyaux. Par la suite, ces tableaux redimensionnés sont rassemblés et envoyés au réseau R-Net. Le résultat du R-Net est similaire à celui du P-Net, composé de coordonnées plus précises pour les nouveaux cadres de sélection, ainsi que du niveau de confiance associé à chacun d'eux. Comme auparavant, les cases avec une confiance moindre sont éliminées. Suite à la normalisation des coordonnées, les boîtes englobantes sont converties en carrés puis transmises au réseau O-Net ([37] (voir Figure 3.7).

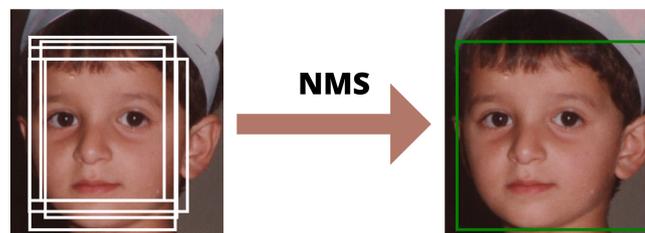


FIGURE 3.7 – Rejet des faux candidats

Étape du bloc O-Net (Output Network)

Cette étape est similaire à la deuxième étape, mais elle implique d'identifier les régions du visage qui contiennent les cinq repères faciaux (yeux, nez et extré-

mités de la bouche) . Avant de traiter les cadres de délimitation de R-Net, tous les cadres qui se trouvent en dehors des limites spécifiées sont remplis de zéros et redimensionnés à une taille de (48x48) pixels. Ces boîtes redimensionnées sont ensuite transmises au réseau O-Net. Les résultats obtenus à partir du bloc O-Net diffèrent légèrement de ceux du P-Net et du R-Net. Le bloc O-Net effectue trois types de traitement : il fournit les coordonnées du cadre englobant (out[0]), les coordonnées des cinq repères faciaux (out[1]) et le niveau de confiance de chaque image (out[2]). Une fois de plus, toutes les cases avec un faible niveau de confiance sont éliminées et les coordonnées des cases englobantes et des repères faciaux restants sont standardisés. Enfin, le traitement NMS est appliqué pour garantir qu'un seul cadre de délimitation existe pour chaque visage de l'image. La toute dernière étape consiste à regrouper toutes les informations dans un dictionnaire à trois clés : "box" contient les coordonnées du cadre englobant, "confidence" contient le niveau de confiance du réseau pour chaque case à cocher et "keypoints" inclut les coordonnées de chaque Repère facial (yeux, nez et extrémités de la bouche) ([37] (voir Figure 3.8).

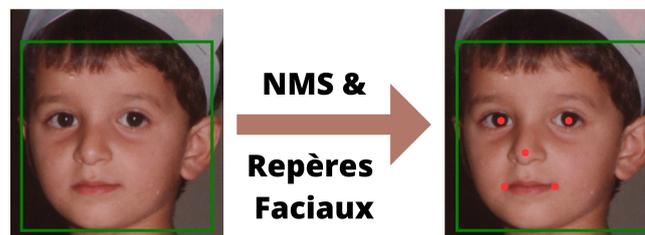


FIGURE 3.8 – Les 5 points de repères faciaux

3.5.2 Alignement :

L'alignement se fait avec le Réseaux convolutifs en cascade multitâches MTCNN (voir Figure 3.9)

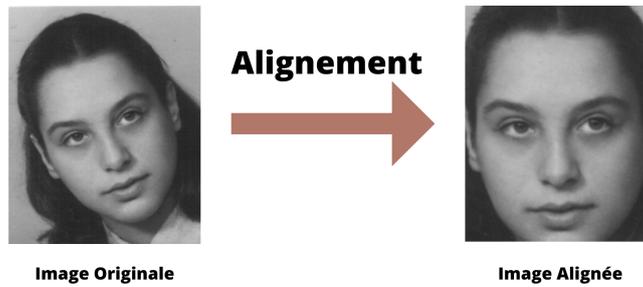


FIGURE 3.9 – Image avant et après Alignement

3.5.3 Extraction des caractéristiques

Après avoir localisé un visage dans une image, la prochaine étape est l'extraction de ces caractéristiques qui est l'étape la plus importante du système de reconnaissance faciale. Il existe plusieurs techniques d'extraction des données du visage, comme mentionner dans le chapitre précédent, nous avons opté pour deux méthodes LBP et HOG que nous allons développer comme suit :

LBP : Motifs binaires locaux

Le modèle binaire local (LBP) a été initialement introduit par Ojala et al [11]. Il s'agit d'une méthode très efficace pour décrire les textures (voir Figure 3.10).

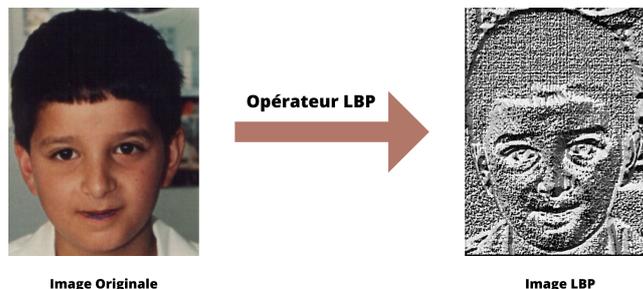


FIGURE 3.10 – Image donnée par le descripteur LBP

LBP basé est sur la séparation de l'image en plusieurs petites régions pour extraire la caractéristique de chaque point ou bien pixel. L'idée, c'est de prendre la valeur centrale selon la formule (3.1) comme seuil, donc nous obtenons 1 lorsque la valeur du pixel voisin est supérieure ou égale au pixel central et 0 lorsque la valeur du pixel voisin est inférieure au pixel central. Après l'obtention d'un code binaire, nous le convertissons en code décimal [11] selon la formule (3.2).

$$LBP_{P,R}(x_c, y_c) = \sum_{n=0}^{P-1} s(g_n - g_c) \cdot 2^n \quad (3.1)$$

$$s(x) = \begin{cases} 1, & \text{si } x \geq 0 \\ 0, & \text{sinon} \end{cases} \quad (3.2)$$

L'instruction donnée décrit un scénario dans lequel g_c représente la valeur d'un pixel situé aux coordonnées centrales (x_c, y_c) . La variable g_n désigne l'une des huit valeurs de pixels proches positionnées autour du rayon R . P représente le nombre total de ces pixels voisins. De plus, il existe une fonction de signe, qui est une fonction mathématique qui attribue une valeur positive ou négative en fonction de l'entrée (voir Figure 3.11).

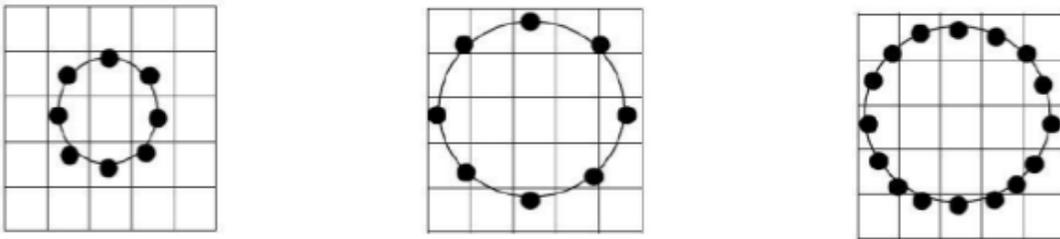


FIGURE 3.11 – Un ensemble voisinage d'opérateur LBP

HOG : Histogramme des gradients orientés

Le descripteur HOG [3], nommé aussi (Histogram of Oriented Gradients) est une technique avancée de représentation des caractéristiques utilisée en reconnaissance faciale et en vision par ordinateur pour la détection d'objets et la reconnaissance de formes. Il se base sur la distribution des gradients d'intensité dans une image pour capturer des informations essentielles sur la texture et la structure des objets selon les étapes suivante.

1. Prétraitement de l'image : Avant de calculer le descripteur HOG, il est important de préparer l'image suivant ces étapes :

- Conversion en niveaux de gris : L'image couleur est convertie en niveaux de gris pour simplifier le calcul des gradients.
 - Normalisation de la luminosité : La luminosité de l'image peut être normalisée pour réduire les variations d'éclairage locales.
 - Réduction du bruit : Le bruit dans l'image peut être réduit pour améliorer la qualité de la représentation.
2. Calcul des gradients : Les gradients d'intensité de l'image sont calculés à l'aide d'opérateurs de dérivation de manière rapide par des masques de dérivation simple de 1D centré. On applique un masque $-1 \ 0 \ 1$ puis un masque $1 \ 0 \ -1$ ou non centré $-1 \ 1$. Il y a d'autre type de masquage tel que filtre de Sobel , ou le filtre gaussien pour 2D pour obtenir les composantes horizontales (G_x) et verticales (G_y) du gradient à chaque pixel de l'image. Par la suite, l'amplitude (g) et la direction d'orientation (θ) sont calculées à partir du gradient comme suit
- La magnitude du gradient (G) : Elle mesure l'intensité du changement de concentration lumineuse à un pixel donné selon la formule (3.3).

$$G = \sqrt{G_x^2 + G_y^2} \quad (3.3)$$

- La direction du gradient (θ) : Elle indique la direction dans laquelle le changement d'intensité est le plus important selon la formule (3.4).

$$\theta = \arctan \left(\frac{G_y}{G_x} \right) \quad (3.4)$$

3. Création de l'histogramme des gradients :

- L'image est divisée en cellules, généralement des cellules de 8x8 pixels.
- Pour chaque cellule, un histogramme des orientations des gradients est construit. Cet histogramme compte combien de gradients tombent dans chaque plage d'orientation prédéfinie (par exemple, 0-20 degrés, 20-40 degrés, etc.).

4. Normalisation des blocs :

- Les cellules sont ensuite regroupées en blocs, par exemple, des blocs de 2x2 cellules
- Pour chaque bloc, les histogrammes des orientations des cellules sont normalisés. en divisant les valeurs de l'histogramme par la norme du vecteur des valeurs de l'histogramme dans le bloc. La normalisation permet de rendre le descripteur HOG insensible aux variations d'éclairage et de contraste locaux.

5. Création du descripteur HOG : Le descripteur HOG final est formé en concaténant tous les histogrammes normalisés des blocs de l'image (voir Figure 3.12).

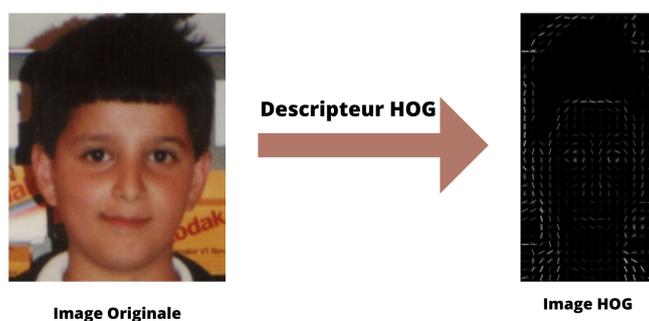


FIGURE 3.12 – Image donnée par le descripteur HOG

3.5.4 Réduction de dimensions

La réduction de dimensions dans le contexte de la reconnaissance faciale est une étape importante qui vise à réduire la complexité des données tout en préservant les caractéristiques essentielles du visage. Cette réduction de dimensions peut être réalisée pour diverses raisons, notamment l'efficacité de la reconnaissance, la réduction du temps de traitement et la réduction des besoins en stockage.

La réduction de dimensions peut aider à rendre les données d'entrée plus gérables pour les algorithmes de reconnaissance faciale tout en préservant la capacité du système à identifier et à comparer les visages de manière efficace. Cependant, il est important de trouver un équilibre pour éviter une perte d'informations critique pour la précision de la reconnaissance. ce domaine connaît beaucoup de méthodes, nous avons opté pour les plus efficaces et les plus utilisées dans ce domaine PCA et LDA.

PCA : Analyse en Composantes Principales

La PCA (Principal Component Analysis) [35] est une technique d'analyse de données et de réduction de dimension largement utilisée en statistiques et en apprentissage automatique. Turk et Pentland sont les premiers ayant mis en évidence l'efficacité de cet algorithme dans le domaine de reconnaissance faciale, elle permet de réduire la complexité d'un ensemble de données en projetant les données dans un nouvel espace de dimension inférieure tout en préservant autant d'information que possible. L'ensemble des vecteurs propres (eigenvectors) et valeurs propres, (eigenvalues) permet de représenter un nouveau plan, la visualisation des vecteurs propres introduit le concept d'eigenfaces illustré dans la (figure). C'est l'ensemble d'images de visages fantômes qui contiennent toutes les variations issues des images d'entraînement. Une projection de ces dernières sur le plan résultant, implique une nouvelle représentation sous forme d'une combinaison linéaire (voir Figure 3.13).

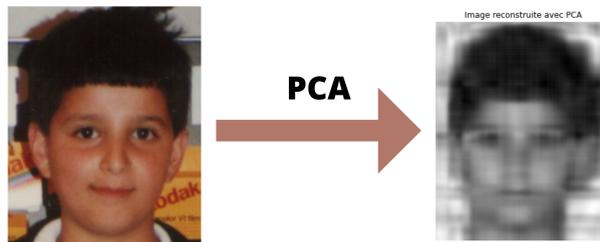


FIGURE 3.13 – Image donnée après l'application de PCA

Supposons que nous ayons un ensemble de données avec N observations, chacune ayant D dimensions. Pour effectuer une PCA, voici les étapes :

1. Centrer les données : Soustraire la moyenne de chaque dimension à partir de toutes les observations pour centrer les données autour de l'origine.
2. Calculer la matrice de covariance : Calculer la matrice de covariance des données centrées. La matrice de covariance mesure les relations linéaires entre les dimensions selon la formule (3.5).

$$\Sigma = \frac{1}{n-1} \sum_{i=1}^n (X_i - \mu)(X_i - \mu)^T \quad (3.5)$$

3. Sélectionner les composantes principales : Trier les vecteurs propres en fonction de leurs valeurs propres associées selon la formule (3.6).

$$\Sigma v = \lambda v \quad (3.6)$$

4. Sélectionner les K premiers vecteurs propres pour réduire la dimension de l'espace de données. Ces K vecteurs propres sont les nouvelles composantes principales.
5. Projeter les données dans le nouvel espace : Pour chaque observation, multiplier les données originales centrées par les K vecteurs propres sélectionnés pour obtenir la projection dans le nouvel espace (voir Figure 3.14).

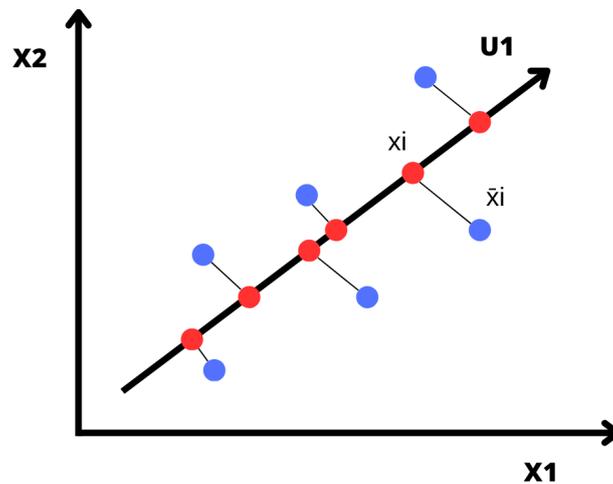


FIGURE 3.14 – Représentation de la dimension PCA

La PCA est utilisée pour diverses tâches telles que la réduction de dimension, la compression d'image, la détection d'anomalies et l'analyse de données. Elle permet de représenter les données de manière plus compacte tout en préservant l'information la plus importante contenue dans l'ensemble de données.

LDA : Analyse discriminante linéaire

LDA est une méthode de réduction de dimensionnalité et de classification [5] l'Analyse Discriminante Linéaire (Linear Discriminant Analysis) a pour but de

maximiser la distance entre les classes tout en minimisant la variance intra-classe.

Plus formellement, intra-classe :

Étapes de calcul de LDA :

1. Les vecteurs moyens de classe sont calculés pour chaque classe c . Supposons que vous ayez C classes et n_c échantillons dans chaque classe. Les vecteurs moyens de classe sont notés \mathbf{m}_c et sont calculés selon la formule (3.7).

$$\mathbf{m}_c = \frac{1}{n_c} \sum_{i=1}^{n_c} \mathbf{x}_{c,i} \quad (3.7)$$

2. Calcul de la matrice de dispersion entre classe (SB) : La matrice de dispersion entre classe mesure la variabilité entre les classes selon la fomule (3.8).

$$\mathbf{SB} = \sum_{c=1}^C n_c (\mathbf{m}_c - \mathbf{m})(\mathbf{m}_c - \mathbf{m})^T \quad (3.8)$$

3. Calcul de la matrice de dispersion infra-classe SW : La matrice de dispersion intraclasse mesure la variabilité à l'intérieur de chaque classe selon la formule (3.9).

$$\mathbf{SW} = \sum_{c=1}^C \sum_{i=1}^{n_c} (\mathbf{x}_{c,i} - \mathbf{m}_c)(\mathbf{x}_{c,i} - \mathbf{m}_c)^T \quad (3.9)$$

4. Sélection des composantes discriminantes : Triez les valeurs propres en ordre décroissant. Sélectionnez les $K-1$ premiers vecteurs propres correspondant aux $K-1$ plus grandes valeurs propres pour réduire la dimensionnalité, où K est le nombre de classes.
5. Projection des données dans le nouvel espace : Projetez les données d'entrée dans le nouvel espace en multipliant les données originales \mathbf{x} par les vecteurs propres sélectionnés \mathbf{w} selon la formule (3.10).

$$Y = X \cdot W \quad (3.10)$$

Où y est la projection dans le nouvel espace, x est le vecteur de données original et w est le vecteur propre sélectionné.

L'analyse discriminante linéaire est couramment utilisée pour la classification, en particulier lorsque les classes sont linéairement séparables. Elle permet de réduire la dimensionnalité des données tout en maintenant la capacité de discrimination entre les classes, ce qui en fait une technique puissante en apprentissage automatique et en reconnaissance faciale (voir Figure 3.15).

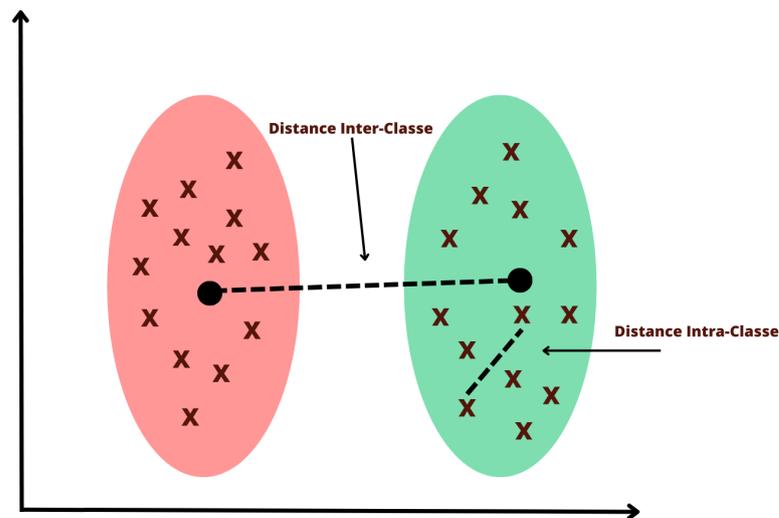


FIGURE 3.15 – Représentation de la dimensionnalité LDA

3.5.5 Réseau de Neurones Convolutifs : CNN

un CNN est une architecture de réseau de neurones profonds spécialement conçue pour la vision par ordinateur. Il extrait automatiquement des caractéristiques hiérarchiques à partir des données d'entrée et il est largement utilisé dans des applications telles que la classification d'images, la détection d'objets et la segmentation sémantique. La combinaison de couches de convolution, de pooling

et de couches entièrement connectées lui permet de modéliser des structures complexes dans les données d'image.

L'architecture spécifique du CNN permet d'extraire des caractéristiques de différentes complexités, des plus simples aux plus sophistiquées. L'information circule de bas en haut, le premier niveau inférieur agit comme détecteur de bord orienté.

Architecture d'un CNN Un CNN est composé de plusieurs couches différentes qui interagissent pour extraire des caractéristiques des données d'entrée et effectuer des tâches spécifiques, comme la classification d'images. Chaque couche cachée représente un niveau d'abstraction légèrement supérieur à celui du bas. Les couches cachées d'un CNN sont principalement composées de Couches de Pooling et Couches entièrement connectée (Fully-Connected).

1. Couches de convolution : Une convolution, est le fait d'appliquer un filtre mathématique à une image. Explicitement, il s'agit de faire glisser une matrice par-dessus d'une image, et pour chaque pixel, utiliser la somme de la multiplication de ce pixel par la valeur de la matrice. Cette technique nous permet de trouver des parties de l'image qui pourraient nous être intéressantes. Noté que lors de l'opération de convolution, la taille de la carte de caractéristique résultante est dépendante de deux paramètres : le padding et le stride (pas). Dans une couche de convolution Plusieurs filtres sont appliqués, chaque filtre possède une dimension égale à la dimension des données à convoluer. Les filtres sont appris par le CNN lors la phase d'entraînement. Les noyaux des filtres désignent les poids de la couche de convolution. Ils sont initialisés puis mis à jour par rétropropagation du gradient. La convolution produit autant de nouvelles cartes de caractéristiques que de filtres de la couche. Toutes les couches de convolution sont généralement suivies d'une fonction d'activation (par exemple ReLU)
2. Couche de Pooling : a pour objectif de sous-échantillonner (rétrécir) l'image d'entrée afin de réduire la charge de calcul, l'utilisation de la mémoire et le nombre de paramètres. Elle utilise un filtre sans valeurs, et qui se contente d'agrèger les entrées en utilisant une fonction d'agrégation, comme la valeur maximale ou la moyenne. Elle est très destructrice. Même avec un minuscule

filtre de 2×2 et un stride de 2, la sortie est deux fois plus petite dans les deux directions, ce qui élimine 75% des valeurs d'entrée.

3. Couches entièrement connectée (Fully-Connected) : Les couches entièrement connectées sont souvent ajoutées au sommet de la pile, elle combine les caractéristiques extraites par les couches précédentes pour classifier l'image, elle renvoie un vecteur de taille N , où N est le nombre de classes dans notre problème de classification d'images. Chaque élément du vecteur indique la probabilité pour l'image en entrée et la couche finale produit la prédiction (par exemple une couche softmax qui génère les probabilités de classes estimées). Les CNNs sont généralement formés de plusieurs couches de convolution et de pooling, suivies par une couche fully-connected qui d'appartenir à une classe.

Les couches de convolution et de Pooling forme un premier bloc qui extrait une hiérarchie de caractéristiques et les couches fully-connected forment un deuxième bloc qui utilise la hiérarchie de caractéristiques pour classer des images (voir Figure 3.16).

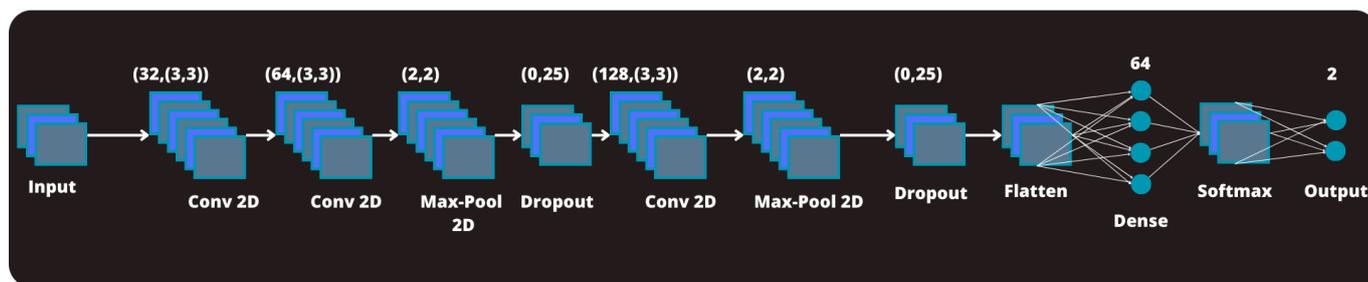


FIGURE 3.16 – Réseaux de neurones convolutifs (CNN) utilisé

3.6 Comparaison et Prise de décision

Dans le contexte de la reconnaissance faciale, la comparaison et la prise de décision sont deux étapes cruciales qui interviennent après que les visages ont été détectés et que leurs caractéristiques ont été extraites. Voici comment elles sont définies dans ce contexte :

La comparaison

La comparaison est le processus d'évaluation de la similitude ou de la correspondance entre les caractéristiques faciales extraites d'un visage donné et celles enregistrées dans une base de données ou une galerie.

Pour chaque visage détecté, les caractéristiques extraites, telles que les vecteurs de descripteurs ou les représentations faciales, sont comparées à celles stockées dans la base de données des visages connus. Les méthodes de comparaison peuvent varier, notamment en utilisant des mesures de similarité, telles que la distance euclidienne ou des techniques plus avancées, comme les réseaux de neurones siamois.

La prise de décision

Une fois que la comparaison a été effectuée, la prise de décision intervient pour déterminer si le visage détecté correspond à une personne connue ou non.

Un seuil de décision peut être défini pour déterminer la similitude minimale requise pour considérer qu'un visage est une correspondance positive avec une personne de la base de données.

En fonction des résultats de la comparaison et du seuil de décision, une décision est prise pour identifier ou non le visage détecté comme étant une correspondance avec une personne enregistrée.

Pour clôturer notre système, nous avons choisi la Distance Euclidienne pour la comparaison et la prise de décision.

3.6.1 Distance Euclidienne

En reconnaissance faciale, les vecteurs de caractéristiques extraits de deux visages sont comparés en utilisant la distance euclidienne. Si la distance euclidienne entre les vecteurs est inférieure à un seuil prédéfini, les visages sont considérés comme similaires (appartenant à la même personne). Sinon, ils sont considérés comme différents (appartenant à des personnes différentes).

La distance euclidienne est une mesure de la similarité ou de la dissimilarité entre deux points dans un espace euclidien. En contexte de reconnaissance faciale, elle est couramment utilisée pour comparer les caractéristiques extraites de deux

visages en calculant la distance entre les deux vecteurs de caractéristiques. Plus la distance est petite, plus les visages sont similaires selon la formule (3.11).

$$\text{Distance Euclidienne} = \sqrt{\sum_{i=1}^n (A_i - B_i)^2} \quad (3.11)$$

3.7 Conclusion

Dans ce chapitre, nous avons détaillé le fonctionnement des méthodes utilisées, voir MTCNN pour la détection de visage, LBP et HOG pour l'extraction des caractéristiques, PCA et LDA pour la réduction de dimension, ainsi que CNN pour l'apprentissage et pour finir la Distance Euclidienne pour la prise de décision.

Chapitre 4

Environnement matériel et logiciel

L'environnement matériel et logiciel puissant et efficace est parmi les clés de réussite d'un système d'identification des personnes.

4.1 Environnement matériel

La configuration de la machine utilisée est comme suit :

Processus : AMD Ryzen 5 5600H with Radeon Graphics 3.30

Memoire RAM GHz 16,0 Go (15,3 Go utilisable)

Système d'exploitation : Windows 10 Professionnel X64

Disque Dur : 500 Go SSD

Carte graphique accélératrice de traitements en temps réel (GPU) : AMD 4G.

4.2 Environnement logiciel

l'environnement logiciel utilisé est très riche :

4.2.1 Python

Python est un langage de programmation créé en 1989 par Guido van Rossum au centre de recherche CWI au pays bas. Il est connu par les caractéristiques suivantes [23] :

- Gratuit
- Multiplate-forme, il fonctionne sur plusieurs systèmes d'exploitation, à savoir Windows, Mac OS X, Linux, Android, iOS,

- Langage de haut niveau. Il demande relativement peu de connaissance sur le fonctionnement d'un ordinateur pour être utilisé.
- Langage interprété. Un script Python n'a pas besoin d'être compilé pour être exécuté, contrairement à des langages comme le C ou le C++. Il est possible de concevoir en Python des entités qui reproduisent celles du monde réel (une cellule, une protéine, un atome, etc.) avec un certain nombre de règles de fonctionnement et d'interactions.

4.2.2 OpenCV

OpenCV (Open Source Computer Vision Library) est une bibliothèque logicielle open source de vision par ordinateur et d'apprentissage automatique. OpenCV a été construit pour fournir une infrastructure commune pour les applications de vision par ordinateur et pour accélérer l'utilisation de la perception artificielle dans les produits commerciaux. La bibliothèque compte plus de 2500 algorithmes optimisés, ce qui inclut un ensemble complet d'algorithmes de vision par ordinateur et d'apprentissage automatique classiques et de pointe. Ces algorithmes sont spécialisés dans le traitement d'images en temps réel, ce qui les rend adaptés à une variété de tâches telles que la détection et la reconnaissance de visages, l'identification d'objets, la classification d'actions humaines dans des vidéos, le suivi des mouvements de la caméra, le suivi d'objets en mouvement, l'extraction de modèles 3D d'objets, etc.

4.2.3 NumPy

NumPy, également connu sous le nom de Numerical Python, est une bibliothèque Python très appréciée et largement utilisée à des fins de calcul numérique. Cette puissante bibliothèque offre un large éventail de structures de données, la plus notable étant les tableaux multidimensionnels, également appelés ndarrays. Ces ndarrays permettent aux utilisateurs de stocker et de manipuler efficacement de grands volumes de données numériques. De plus, NumPy fournit une collection complète de fonctions mathématiques qui peuvent être appliquées à ces tableaux, facilitant diverses opérations et calculs. Grâce à ses capacités exceptionnelles, NumPy a acquis une immense popularité dans les domaines de la science des don-

nées, de l'apprentissage automatique et de la recherche scientifique. L'efficacité de la bibliothèque dans la gestion de calculs complexes et de tâches de manipulation de données en a fait un outil indispensable pour les professionnels de ces domaines.

4.2.4 Matplotlib

Matplotlib est une bibliothèque puissante spécialement conçue pour Python, axée sur la création de tracés 2D visuellement attrayants et informatifs. Grâce à sa vaste gamme de fonctionnalités et d'options de personnalisation, les utilisateurs peuvent générer sans effort des graphiques, des diagrammes, des courbes et des visualisations de données personnalisés. Matplotlib est largement utilisé en conjonction avec NumPy, permettant la visualisation de calculs numériques d'une manière facilement interprétable.

4.2.5 TensorFlow

TensorFlow est une bibliothèque open source puissante et polyvalente qui a été méticuleusement conçue et développée par Google. Son objectif principal est de faciliter les processus de calcul numérique et d'apprentissage automatique. L'une des caractéristiques les plus remarquables de TensorFlow est sa remarquable capacité à construire et à entraîner des réseaux neuronaux profonds, incroyablement bénéfiques pour un large éventail d'applications d'apprentissage automatique. Ces applications englobent la vision par ordinateur, le traitement du langage naturel et de nombreux autres domaines. Grâce à sa flexibilité exceptionnelle, TensorFlow a acquis une immense popularité et est largement utilisé par les chercheurs et les professionnels dans le domaine de l'intelligence artificielle. Il est devenu un outil indispensable pour d'innombrables projets d'IA menés dans diverses institutions de recherche et industries à travers le monde.

4.2.6 keras

Keras est une bibliothèque open source pour l'apprentissage automatique, en particulier pour le deep learning. Elle fournit une interface haut niveau pour créer et entraîner des réseaux de neurones artificiels. L'objectif de Keras est de

rendre le développement de modèles de deep learning plus accessible et convivial en fournissant une API simple et cohérente. Keras est largement utilisée dans la communauté de l'apprentissage automatique pour sa facilité d'utilisation et sa flexibilité. Elle permet aux chercheurs et aux développeurs de créer rapidement des modèles de réseaux de neurones pour une variété de tâches, telles que la classification d'images, la génération de texte, la détection d'objets, etc.

4.2.7 Google Colaboratory

Google Colaboratory, également connu sous le nom de Colab, est une plateforme de blocs-notes interactifs développée par Google et basée sur le cloud. Cette plateforme permet aux utilisateurs d'exécuter du code Python directement dans un environnement de notebook appelé Jupyter, sans avoir besoin de procédures de configuration ou d'installation complexes. L'efficacité de Colab réside dans la fourniture d'un accès gratuit aux ressources informatiques, notamment aux puissants GPU et TPU de Google. Cette fonctionnalité remarquable a rendu Colab très recherché parmi les chercheurs et les praticiens dans les domaines de l'apprentissage automatique et de la science des données.

4.3 Bases de données utilisées

Pour notre système, nous avons utilisé La base de données à accès publique FG Net qui est une collection de données bien connue et largement utilisée pour tester et améliorer les algorithmes de reconnaissance faciale. elle contient un large éventail d'images de personnes d'origines, d'âges et de contextes variés, permettant aux chercheurs d'évaluer dans quelle mesure les systèmes de reconnaissance faciale peuvent identifier les individus à différentes étapes de leur vie.

La base de données contient 1002 image de 82 personnes agés de 0 à 69 ans, le nombre d'image par personne varie entre 9 à 13 images par personnes, nous l'avons soigneusement préparé manuellement en la divisant en deux classes : 779 images pour la classe positive et 223 images pour la classe négative. Chaque classe est divisée en ensembles d'apprentissage avec un taux de 80%, de validation avec un taux de 10%, et de tests avec un taux de 10%, Chaque image est redimensionnée à (64, 64), pour faciliter le travail.

Cette base de données constitue une ressource importante pour faire progresser la recherche d'identification des enfants disparus, car elle met en évidence les défis associés à la diversité des données et des scénarios réels.

4.4 Algorithme

Nous présentons quelques algorithmes des Méthodes Appliquées :

4.4.1 MTCNN : Réseaux convolutifs en cascade multitâches

La méthode MTCNN est utilisée pour détecter et aligner les visages, la Figure(4.1) montre le code python de la méthode et la Figure (4.2) montre le résultat du code.

```
[ ] 1 img = cv2.imread('/content/drive/MyDrive/PFC/DATA/train/child/041A04.JPG')
[ ] 1 print(img.shape)
(520, 479, 3)
##### MTCNN MTCNN #####
[ ] 1
1 detector = MTCNN()
2 faces = detector.detect_faces(img)
3
4 for face in faces:
5     x, y, w, h = face['box']
6
7     left_eyex, left_eyey = face['keypoints']['left_eye']
8     right_eyex, right_eyey = face['keypoints']['right_eye']
9     nosex, nosey = face['keypoints']['nose']
10    mouth_leftx, mouth_lefty = face['keypoints']['mouth_left']
11    mouth_rightx, mouth_righty = face['keypoints']['mouth_right']
12
13    cv2.circle(img, (left_eyex, left_eyey), color=(0,255,0), thickness=2, radius=2)
14    cv2.circle(img, (right_eyex, right_eyey), color=(0,255,0), thickness=2, radius=2)
15    cv2.circle(img, (nosex, nosey), color=(255,0,0), thickness=2, radius=2)
16    cv2.circle(img, (mouth_leftx, mouth_lefty), color=(0,0,255), thickness=2, radius=2)
17    cv2.circle(img, (mouth_rightx, mouth_righty), color=(0,0,255), thickness=2, radius=2)
18    cv2.rectangle(img, (x,y), (x+w, y+h), (255,0,0), 3)
19 plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
20 plt.show()
21
22 # Enregistrer l'image dans le répertoire
23 output_path = '/content/drive/MyDrive/PFC/DATA/IMAGE/image_mtcnn.jpg'
```

FIGURE 4.1 – Code python de la méthode MTCNN

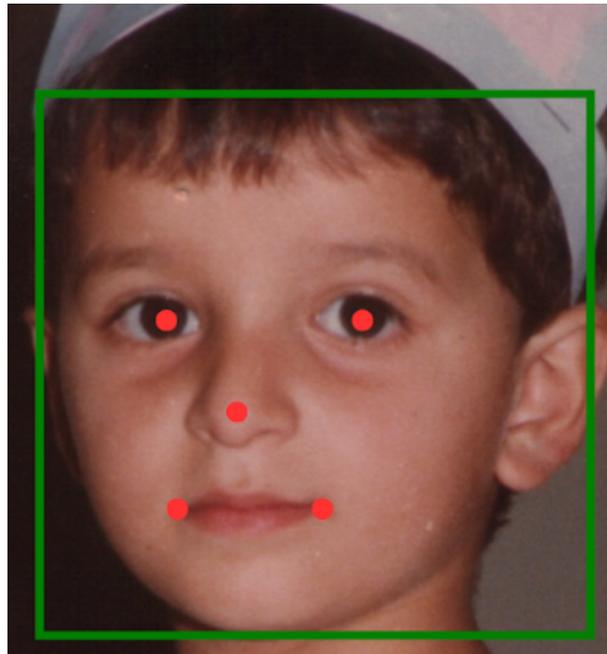


FIGURE 4.2 – visage détecté avec MTCNN

4.4.2 LBP : Motifs binaires locaux

la méthode LBP est la première méthode qu'on a utilisée pour l'extraction des caractéristiques, la Figure (4.3) montre le code python de la méthode et la Figure (4.4) montre le résultat du code.

```

1 #img = cv2.imread('/content/drive/MyDrive/PFC/DATA/train/041A04.JPG', cv2.IMREAD_GRAYSCALE)
2 method_lbp=['default','ror','uniform','var']
3 # Calculer l'image LBP
4 radius = 1
5 n_points = 8 * radius
6 LBP = np.zeros_like(image)
7 for i in range(radius, image.shape[0]-radius):
8     for j in range(radius, image.shape[1]-radius):
9         center = image[i,j]
10        val = 0
11        for n in range(n_points):
12            theta = n * (2 * np.pi / n_points)
13            x = int(np.round(i + radius * np.cos(theta)))
14            y = int(np.round(j - radius * np.sin(theta)))
15            if np.any(np.greater_equal(image[x,y], center)):
16                val += 2**n
17        LBP[i,j] = val
18 # Affichage de l'image
19 plt.imshow(image)
20 plt.show()
21 # Affichage de l'image LBP
22 plt.imshow(LBP, cmap='gray')
23 plt.show()
24
25 #transformer l'image LBP en vecteur
26 V_LBP = LBP.reshape(-1)
27 print(V_LBP.shape)
28 #V_LBP = LBP.Flatten()
29 print('C EST MOI LBP', V_LBP)
30
31 #Normaliser le vecteur LBP
32 V_LBP_NOR = (V_LBP - V_LBP.mean(axis=0)) / V_LBP.std(axis=0)
33 print(V_LBP_NOR)
34 print(V_LBP_NOR.shape)
35 LBP_resized = cv2.resize(LBP, hog_image_rescaled.shape)
36 output_path = '/content/drive/MyDrive/PFC/DATA/IMAGE/image_lbp.jpg'
37 plt.imsave(output_path, cv2.cvtColor(LBP_resized, cv2.COLOR_BGR2RGB))

```

FIGURE 4.3 – Code python de la méthode LBP

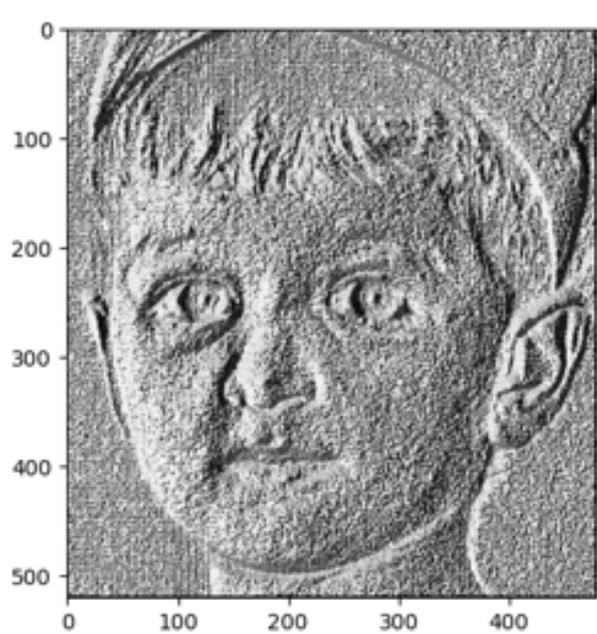


FIGURE 4.4 – Image après application de LBP

4.4.3 HOG : Histogramme des gradients orientés

la méthode HOG est la deuxième méthode qu'on a utilisée pour l'extraction des caractéristiques, la Figure (4.5) montre le code python de la méthode et la Figure (4.6) montre le résultat du code.

```

1 #img = cv2.imread('/content/drive/MyDrive/PFC/DATA/train/041A04.JPG')
2 image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
3 resized_img = resize(image, (128*4, 64*4))
4 print(resized_img.shape)
5 plt.imshow(resized_img )
6 plt.show()
7 V_HOG, hog_image = feature.hog(resized_img, orientations=9, pixels_per_cell=(8, 8),
8                               cells_per_block=(3, 3), block_norm='L2-Hys',
9                               visualize=True, transform_sqrt=False,
10                              feature_vector=True, multichannel = False)
11 print('C EST MOI HOG', V_HOG)
12 print(V_HOG.shape)
13
14 #Normaliser le vecteur HOG
15 V_HOG_NOR = (V_HOG - V_HOG.mean(axis=0)) / V_HOG.std(axis=0)
16 print(V_HOG_NOR)
17 print(V_HOG_NOR.shape)
18 plt.imshow(hog_image)
19 plt.show()
20
21 #Normaliser l'IMAGE HOG
22 hog_image_rescaled = exposure.rescale_intensity(hog_image, in_range=(0, 10))
23
24 output_path = '/content/drive/MyDrive/PFC/DATA/IMAGE/image HOG.jpg'
25 plt.imsave(output_path, hog_image_rescaled)

```

FIGURE 4.5 – Code python de la méthode HOG

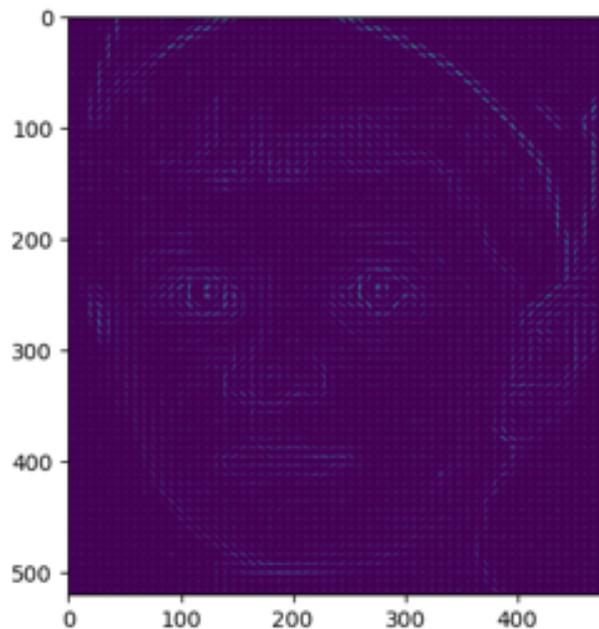


FIGURE 4.6 – Image après application de HOG

4.4.4 Concaténation des vecteurs de caractéristiques LBP et HOG

La Figure (4.7) montre le code python pour la Concaténation des vecteurs de caractéristiques LBP et HOG pour les préparer à la réduction de la dimension.

```
1 #Concaténer les deux images horizontalement
2 # Ajouter des pixels de remplissage à l'image de plus petite taille
3 if LBP_resized.shape[0] < hog_image_rescaled.shape[0]:
4     LBP_resized = cv2.copyMakeBorder(LBP_resized, 0, hog_image_rescaled.shape[0] - LBP_resized.shape[0], 0, 0, cv2.BORDER_CONSTANT, value=0)
5 else:
6     hog_image_rescaled = cv2.copyMakeBorder(hog_image_rescaled, 0, LBP_resized.shape[0] - hog_image_rescaled.shape[0], 0, 0, cv2.BORDER_CONSTANT, value=0)
7
8 #Concaténer les deux images
9 conca_img = np.concatenate((LBP_resized, hog_image_rescaled), axis=1)
10 plt.imshow(conca_img)
11 plt.title('HOG and LBP')
12 plt.show()
13 print(conca_img.shape)
14 #####
15 #Concaténer les deux vecteurs horizontalement
16 CONCAT_V = np.concatenate((V_LBP, V_HOG), axis=0)
17 V = CONCAT_V.reshape(-1, 1)
18 print(V)
19 print(V.shape)
```

FIGURE 4.7 – code python pour la Concaténation des vecteurs de caractéristiques

4.4.5 PCA : Analyse en Composantes Principales

la méthode PCA est la première méthode qu'on a utilisée pour la réduction de la dimension, la Figure (4.8) montre le code python de la méthode.

```
1 # Aplatir les images en vecteurs de caractéristiques
2 #images = np.load('conca_img.npy')
3 pca_components=100
4 n_samples = conca_img.shape[0]
5 X = conca_img.reshape((n_samples, -1))
6 pca = PCA(n_components = pca_components)
7 pca.fit(X)
8 X_pca = pca.transform(X)
9 print(X_pca)
10 print(X_pca.shape)
11 # Visualiser les résultats
12 plt.scatter(X_pca[:, 0], X_pca[:, -1] , color = 'red')#c=labels)
13 plt.xlabel('Composante principale 1')
14 plt.ylabel('Composante principale 2')
15 plt.show()
```

FIGURE 4.8 – code python de la méthode PCA

4.4.6 LDA : Analyse discriminante linéaire

la méthode LDA est la deuxième méthode qu'on a utilisée pour la réduction de la dimension, la Figure (4.9) montre le code python de la méthode.

```

1 # Convertir y en un tableau à une dimension
2 # Redimensionner X_pca si nécessaire
3
4 #n_samples = len(X_pca)
5 n_features = len(X_pca[0])
6 X_pca = X_pca.reshape((n_samples, n_features))
7 #n_samples = X_pca.shape[1]
8 #y = np.zeros(X_pca.shape[0])
9 #y = np.concatenate([np.zeros(X_pca.shape[0]),np.ones(X_pca.shape[1])])
10 y = np.random.randint(0, 4, X_pca.shape[0])
11 #print(y)
12 #y = y.reshape(-1)
13 #print(y)
14 #print(X_pca.shape)
15 #print(y.shape)
16
17 # Appliquer LDA sur les données réduites en dimension
18 lda = LinearDiscriminantAnalysis(n_components=1)
19 X_lda = lda.fit_transform(X_pca, y)
20 #X_lda =X_lda.reshape(-1,1)
21 print(X_lda.shape)
22 # Afficher les résultats
23 plt.scatter(X_lda, np.zeros(X_pca.shape[0]), c='red', alpha=0.5)
24 ()#plt.scatter(X_lda[y==0], np.zeros(X_pca.shape[0]), color='red', alpha=0.5, label='child')
25 #plt.scatter(X_lda[y==1], np.zeros(X_pca.shape[0]), color='blue', alpha=0.5, label='not_child')
26 plt.legend()
27 plt.title('Analyse discriminante linéaire')
28 plt.xlabel('LDA')

```

FIGURE 4.9 – code python de la méthode LDA

4.5 Conclusion

Dans ce chapitre, nous avons présenté les outils matériels et les différents outils logiciels tels que python et ces vastes bibliothèques comme Opencv et NumPy, nous avons présenté aussi la bibliothèque TensorFlow, l'API keras et la plateforme de blocs-notes (Google Colaboratory) et finalement la base de donnée FG net, la clé de notre système et pour clôturer, nous avons présenté quelques codes python des méthodes utilisés et leurs images résultantes.

Conclusion générale

En conclusion, l'identification des enfants disparus à partir d'images faciales malgré l'avancement dans l'âge est un domaine complexe et émotionnellement chargé qui a été profondément influencé par les progrès technologiques. Dans notre exploration de ce sujet, nous avons observé que l'identification rapide est essentielle pour augmenter les chances de réunir les familles. Les technologies de reconnaissance faciale offrent un nouvel espoir en permettant de comparer des images d'enfants disparus avec des photographies actuelles, même lorsque des années se sont écoulées depuis leur disparition. nous avons sélectionné quelques outils de ces technologies telles que MTCNN pour la détection et l'alignement des images, LBP et HOG pour l'extraction des caractéristiques, PCA et LDA pour la réduction des dimensions et un CNN pour l'apprentissage. La combinaison de tous ces outils donne une robustesse et une efficacité au système proposé.

Dans le premier chapitre, nous avons Identifié le problème que nous avons abordé qui est l'identification des enfants disparus, nous avons présenté les différentes approches des chercheurs dans le domaine de la reconnaissance facile, puis nous avons fait une comparaison des méthodes apportées, Dans le deuxième chapitre, nous avons présenté un système global de la reconnaissance faciale, puis détailler un état de l'art du domaine de la reconnaissance faciale en présentant les différentes méthodes et les outils existant, Dans le troisième chapitre, nous avons détaillé le fonctionnement des méthodes utilisées, voir MTCNN pour la détection de visage, LBP et HOG pour l'extraction des caractéristiques, PCA et LDA pour la réduction de dimension, ainsi que CNN pour l'apprentissage et pour finir la Distance Euclidienne pour la prise de décision, Et en fin dans le quatrième chapitre, nous avons présenté les outils matériels et les différents outils logiciels tels que python et ces vastes bibliothèques comme Opencv et NumPy, nous avons présente aussi TensorFlow,keras et la plate-forme de blocs-notes (Google Cola-

boratory) et finalement la base de donnée Fgnet la clé de notre système et pour clôturer, nous avons présenté quelques codes python des méthodes utilisés et leurs images résultantes.

l'association de LBP et HOG pour l'extraction des caractéristiques, PCA et LDA pour la réduction de dimension, ainsi que CNN pour l'apprentissage et la Distance Euclidienne pour la prise de décision à donner robustesse et efficacité à notre système, Nous prévoyons d'étendre notre travail à d'autres méthodes afin d'améliorer la précision de notre système et d'augmenter le taux d'identification des enfants disparus. Ce qui Offre l'espoir de transformer des vies et de restaurer l'espoir dans des moments de désespoir.

Bibliographie

- [1] Timo Ahonen, Esa Rahtu, Ville Ojansivu, and Janne Heikkila. Recognition of blurred faces using local phase quantization. In *2008 19th international conference on pattern recognition*, pages 1–4. IEEE, 2008.
- [2] Peter N. Belhumeur, Joao P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces : Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7) :711–720, 1997.
- [3] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.
- [4] Debayan Deb, Divyansh Aggarwal, and Anil K Jain. Child face age-progression via deep feature aging. *arXiv preprint arXiv :2003.08788*, 2020.
- [5] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2) :179–188, 1936.
- [6] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [8] Dihong Gong, Zhifeng Li, Dahua Lin, Jianzhuang Liu, and Xiaoou Tang. Hidden factor analysis for age invariant face recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 2872–2879, 2013.
- [9] Dihong Gong, Zhifeng Li, Dacheng Tao, Jianzhuang Liu, and Xuelong Li. A maximum entropy feature descriptor for age invariant face recognition. In

- Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5289–5297, 2015.
- [10] Ziad M Hafed and Martin D Levine. Face recognition using the discrete cosine transform. *International journal of computer vision*, 43 :167–188, 2001.
- [11] David Harwood, Timo Ojala, Matti Pietikäinen, Shalom Kelman, and Larry Davis. Texture classification by center-symmetric auto-correlation, using kullback discrimination of distributions. *Pattern Recognition Letters*, 16(1) :1–10, 1995.
- [12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [13] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets : Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv :1704.04861*, 2017.
- [14] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. Independent component analysis, a wiley-interscience publication, 2001.
- [15] Juho Kannala and Esa Rahtu. Bsif : Binarized statistical image features. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 1363–1366, 2012.
- [16] Juho Kannala and Esa Rahtu. Bsif : Binarized statistical image features. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, pages 1363–1366. IEEE, 2012.
- [17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998.
- [19] Xiaochao Li, Zhenjie Yang, and Hongwei Wu. Face detection based on receptive field enhanced multi-task cascaded convolutional neural networks. *IEEE Access*, 8 :174922–174930, 2020.

- [20] Xiaochao Li, Zhenjie Yang, and Hongwei Wu. Face detection based on receptive field enhanced multi-task cascaded convolutional neural networks. *IEEE Access*, 8 :174922–174930, 2020.
- [21] Zhifeng Li, Dihong Gong, Xuelong Li, and Dacheng Tao. Aging face recognition : a hierarchical learning model based on local patterns selection. *IEEE Transactions on Image Processing*, 25(5) :2146–2154, 2016.
- [22] Zhifeng Li, Unsang Park, and Anil K Jain. A discriminative model for age invariant face recognition. *IEEE transactions on information forensics and security*, 6(3) :1028–1037, 2011.
- [23] Haibin Ling, Stefano Soatto, Narayanan Ramanathan, and David W Jacobs. Face verification across age progression using discriminative methods. *IEEE Transactions on Information Forensics and security*, 5(1) :82–91, 2009.
- [24] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd : Single shot multibox detector. In *Computer Vision–ECCV 2016 : 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.
- [25] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.
- [26] Mahyar Najibi, Mohammad Rastegari, and Larry S Davis. G-cnn : an iterative grid based object detector. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2369–2377, 2016.
- [27] Mahyar Najibi, Pouya Samangouei, Rama Chellappa, and Larry S Davis. Ssh : Single stage headless face detector. In *Proceedings of the IEEE international conference on computer vision*, pages 4875–4884, 2017.
- [28] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1) :51–59, 1996.
- [29] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once : Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

- [30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn : Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.
- [31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net : Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015 : 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.
- [32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*, 2014.
- [33] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabino-vich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [34] Mingxing Tan and Quoc Le. Efficientnet : Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.
- [35] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1) :71–86, 1991.
- [36] Paul Viola, Michael Jones, et al. Robust real-time object detection. *International journal of computer vision*, 4(34-47) :4, 2001.
- [37] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10) :1499–1503, 2016.