

Republique Algerienne Democratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université A/Mira de Bejaia
Faculté des Sciences Exactes
Département d'Informatique

MEMOIRE DE MASTER RECHERCHE

En
Informatique

Option
intelligence artificielle

Thème

Vérification de la parenté à partir des images faciale
par HSV , BSIF , LPQ , VGG19 et CS avec CNN

Présenté par :
Belaid Sabine
Sahli Imene

Soutenu le 06 Juillet 2022 devant le jury composé de :

Président	Dr M.MOKETFI	MCB	U. A/Mira Bejaia.
Rapporteur	Dr M.KHEMMARI	MCA	U. A/Mira Bejaia.
Examineur	Dr k.SOUADIH	Docteur	Entreprise Sonatrash Bejaia

Bejaia, Juillet 2022.

Remerciements

On remercie particulièrement nos familles et nos amis pour le soutien qu'ils nous ont apporté et la motivation qu'ils ont su nous donner lorsque on en avait le plus besoin.

On tient aussi a remercier notre encadrant monsieur kHAMMARI Mohammed
sans qui , on aurait pas pu réaliser ce mémoire .

Nos sincères remerciements aux Dr MOKETFI Mohand et Dr SOUADIH Kamel

Pour avoir accepté de juger notre travail .
veuillez assurée notre entière reconnaissance.

On remercie sincèrement toute personne qui nous a apporté son soutien que soit
de près ou de loin.

Dédicaces

On dédie ce mémoire à nos chères parents qui ont toujours été là pour nous.

nous vous sommes reconnaissantes pour les sacrifices consentis
pour notre bien être. Votre soutien et votre éducation ont fait de nous
ce qu'on est aujourd'hui.

A nos grands parents, nos frères et soeurs et toutes nos familles
pour leur soutiens inconditionnels.

A nos amis qui nous ont apporté soutien et consolation dans les moment de besoin.

Table des matières

Table des matières	ii
Table des figures	vi
Liste des tableaux	viii
Notations et symboles	ix
Introduction générale	1
1 Concepts de base pour vérification de parenté	3
1.1 Introduction	3
1.2 Qu'est-ce qu'on appelle la parenté	3
1.3 Parenté dans la vision par ordinateur	4
1.4 Motivation et application de la parenté	4
1.5 Problèmes de parenté et défis	5
1.6 Travaux connexe	6
1.6.1 Vérification de la parenté basée sur l'apprentissage des caractéristiques	6
1.6.2 Vérification de la parenté basée sur l'apprentissage de métrique	12
1.6.3 Vérification de la parenté basée sur l'apprentissage profond convolutif	16
1.7 Conclusion	19
2 Méthodes de vérification de parenté	20
2.1 Introduction	20
2.2 Détection de visage	20
2.2.1 méthodes de détection de visage	21
2.3 Les espaces de représentation de la couleur :	26
2.3.1 L'espace colorimétrique RGB :	26
2.3.2 L'espace colorimétrique YCbCr	26
2.3.3 Niveaux de gris	27
2.3.4 L'espace colorimétrique HSV :	27
2.4 Extraction de caractéristiques	28

2.4.1	les méthode basé sur les caractéristiques artisanales (handcraft features)	29
2.4.2	les méthode basé sur les caractéristiques profond(Deep features)	32
2.5	Réduction de caractéristique	34
2.5.1	L'analyse discriminante linéaire (LDA)	35
2.5.2	Analyse en Composantes Principales (ACP)	36
2.6	Apprentissage profond (Deep Learning)	37
2.6.1	réseau de neurones convolutif CNN :	38
2.6.2	Keras	40
2.7	Algorithme de classification	40
2.7.1	K-plus proches voisins	41
2.8	Arbre de décision	42
2.9	Machines à vecteur de support (SVM)	43
2.10	conclusion	44
3	Système de vérification de parenté	45
3.1	Introduction	45
3.2	Structure du système proposé	45
3.2.1	Detection de visage	46
3.2.2	Conversion d'espace de couleurs de RGB vers HSV	48
3.2.3	Extraction de caractéristiques	49
3.2.4	Réduction de caractéristique avec ACP	53
3.3	L'apprentissage	54
3.3.1	Cosine similarity	60
3.4	Conclusion	61
4	Conception et implémentation	62
4.1	Introduction	62
4.2	Environnement de développement	62
4.2.1	Jupyter notebook	63
4.2.2	Tkinter	63
4.3	Les Bases de Donnees Utilisées	63
4.4	Architecture de l'application	64
4.4.1	Accueil	65
4.4.2	A propos de nous	66
4.4.3	Choisir la méthode	66
4.5	Test et Résultat	72
4.5.1	LPQ test	73
4.5.2	BSIF test	73
4.5.3	VGG19 test	74
4.5.4	Concatenation test	74

4.5.5	COSINE Similarity test	75
4.5.6	Interprétation et discussion des résultats	75
4.6	conclusion	76
Conclusion et perspectives		77
Bibliographie		77

Table des figures

1.1	Une illustration des grands défis de la vérification de la parenté dans une scène réelle [Fu et al. (2016)][37]	6
1.2	Le réseau hybride d'apprentissage à distance (HDL) proposé [72]	10
1.3	La méta-vue de la méthode KML avec une architecture profonde soigneusement conçue KinNet [88]	13
1.4	Schéma de principe du système de comparaison de paires de visages MSIDA [62]	15
2.1	architecture de yolo	22
2.2	Application du filtre YOLO sur une image	22
2.3	L'image d'entrée est redimensionnée à différentes échelles pour construire une pyramide d'images	23
2.4	Balayage des caractéristiques de type Haar dans une image	25
2.5	Cube colorimétrique RGB	26
2.6	Cylindre colorimétrique HSV	28
2.7	Algorithme de l'opérateur LBP	30
2.8	Algorithme de l'opérateur LPQ	31
2.9	Algorithme de l'opérateur BSIF	32
2.10	Architecture du réseau neuronal VGG	33
2.11	Projections LDA et non LDA[12]	36
2.12	Interprétation géométrique de l'analyse en composantes principales[38]	37
2.13	Les Couche convolutionnelle et couche de mise en commun	39
2.14	Couche entièrement connectée	40
2.15	Classification par K plus proches voisins	41
2.16	Exemple d'un arbre de décision construit à partir de l'exemple de Quinlan [31]	42
2.17	Les vecteurs de support, marqués par des carrés gris[17]	43
3.1	Notre systeme proposé	46
3.2	Balayage des caractéristiques de type Haar dans l'image	47
3.3	Detection de visage	48
3.4	Transformations d'espace RGB vers HSV	49
3.5	différents types de filtres [39]	50
3.6	BSIF sur les trois images H ,S et V	50

3.7	les étapes de l'opérateur LPQ de base	51
3.8	LPQ sur les trois images H ,S et V	51
3.9	Système de verification basé sur la méthode d'extraction de caractéristiques LPQ . .	55
3.10	Système de verification basé sur la méthode d'extraction de caractéristiques BSIF .	56
3.11	Système de verification basé sur la méthode d'extraction de caractéristiques VGG19	57
3.12	Modèle de reseau de neurone utilisé	59
3.13	Système de vérification basé sur le calcul de similarité en utilisant cosine similarity	61
4.1	Exemples alignés et recadrés des ensembles de données KinFaceW-I et KinFaceW-II, respectivement. De haut en bas, on trouve les relations de parenté F-S, F-D, M-S et M-D, et les deux images voisines de chaque rangée présentent la relation de parenté, respectivement.	64
4.2	Structure générale de notre application	64
4.3	Interface accueil	65
4.4	Interface A propos de nous	66
4.5	Importer deux photos	66
4.6	Importer deux photos	68
4.7	Application de la transformation HSV	68
4.8	Application de l'extraction des caractéristiques LPQ	69
4.9	Application de la méthode de reduction des caractéristiques PCA	69
4.10	Application du modele cnn	70
4.11	Interface de la méthode BSIF	70
4.12	Interface de la méthode VGG	71
4.13	Interface de la méthode concatination	71
4.14	Interface de la méthode cosine semilarty	72

Liste des tableaux

4.1	test LPQ	73
4.2	test LPQ	73
4.3	test BSIF	73
4.4	test BSIF	74
4.5	VGG-19	74
4.6	test VGG-19	74
4.7	test Concatenation	74
4.8	test LPQ	75
4.9	test cosine similarity	75
4.10	test cosine similarity	75

Notations et symboles

	<i>3DMDA</i>	Three-dimensional Modular Discriminant Analysis.
A	<i>AAM</i>	Modèle Apparence Active.
	<i>AGES</i>	AGing pattErn Subspace.
	<i>AFAD</i>	Asian Face Age Dataset.
	<i>AdaBoost</i>	Adaptive Boosting.
B	<i>BRIEF</i>	Binary Robust Independent Elementary Features.
	<i>BSIF</i>	Binarized Statistical Image Features.
C	<i>CNN</i>	Convolutional Neural Network.
	<i>CEA</i>	Conformal Embedding Analysis.
	<i>CS</i>	Cumulative Score.
	<i>CIF</i>	Common Intermediate Format .
	<i>CCA</i>	Canonical Correlation Analysis.
	<i>CoALBP</i>	Co-occurrence of Adjacent Local Binary Patterns.
	<i>CV</i>	Computer Vision.
D	<i>DCT</i>	Discrete Cosine Transform.
	<i>DL</i>	Deep Learning.

E	<i>ELM</i>	Extreme Learning Machine.
	<i>EDA</i>	Exponential Discriminant Analysis
F	<i>FRGC</i>	Face Recognition Grand Challenge.
	<i>FIW</i>	Families In the Wild
	<i>FLD</i>	Fisher's Linear Discriminant
	<i>FM – D</i>	Father-Mother-Daughter
	<i>FM – S</i>	Father-Mother-Son
	<i>FCN</i>	Fully Convolutional Network.
H	<i>HSV</i>	Hue Saturation Value.
	<i>HOG</i>	Histogram Oriented Gradients.
I	<i>IA</i>	Intelligence Artificielle
	<i>IBR</i>	Iteration Bayesian Reweighed .
	<i>IFDB</i>	Iranian Face DataBase.
	<i>ICA</i>	Independant Component Analysis.
	<i>IoU</i>	Intersection over Union.
K	<i>KRR</i>	Kernel Ridge Regression.
	<i>KNN</i>	K Nearest Neighbors.
	<i>KL</i>	Kullback Leibler
L	<i>LBP</i>	Local Binary Patterns.
	<i>LDA</i>	Linear Discriminant Analysis.
	<i>LARR</i>	Locally Adjusted Robust Regression.
	<i>LPQ</i>	Local Phase Quantisation.
M	<i>MAE</i>	Mean Absolute Error.
	<i>ML</i>	Machine Learning.
	<i>MLR</i>	Multiple Linear Regression
	<i>MLP</i>	Multilayer Perceptron
	<i>MTCNN</i>	Multi Task Cascaded CNN .
N	<i>NMS</i>	Non maximu Suppression.
O	<i>ODFL</i>	Ordinal Deep Feature Learning.
	<i>O – Net</i>	Output Network.

P	<i>PCA</i>	Principal Component Analysis.
	<i>PLS</i>	Partial Least Squares.
	<i>P – Net</i>	Proposal Network.
R	<i>RESNet</i>	Remote Sensing Neural Network.
	<i>RNN</i>	Recurrent Neural Networks.
	<i>RCNN</i>	Regions Convolutional Neural Network.
	<i>R – Net</i>	Refinement network.
	<i>ROI</i>	Region of Interest.
S	<i>SVR</i>	Support Vector Regression.
	<i>SOM</i>	Self Organizing Map.
	<i>SPF</i>	Spatially Patch Flexible.
	<i>SURF</i>	Speeded Up Robust Features.
	<i>SVM</i>	Support Vecteur Machine.
	<i>SSH</i>	Single Stage Headless.
	<i>SSD</i>	Single Shot multibox Detector.
	<i>SIFT</i>	Scale Invariant Feature Transform.
U	<i>UV</i>	UltraViolet.
V	<i>VGG</i>	Visual Geometry Group.
W	<i>WIT – DB</i>	Waseda Interaction Technology DataBase .
Y	<i>YOLO</i>	You Only Look Once.
	<i>YGA</i>	Yahama Gender et Age

Introduction générale

L'analyse automatique des visages à partir des images a toujours été un sujet d'étude important dans les communautés de la reconnaissance des formes et de l'intelligence artificielle. Les images faciales contiennent de nombreuses informations sur la personne à laquelle elles appartiennent, comme l'identité, l'âge, le sexe, l'origine ethnique et l'expression.

Pour cette raison, l'analyse des images faciales a de nombreuses applications dans les problèmes du monde réel tels que la reconnaissance des visages, l'estimation démographique, la reconnaissance des expressions faciales et la vérification de la parenté.

La vérification de la parenté entre les personnes s'est principalement concentrée sur l'étude de la similarité entre les visages humains.

En effet, il existe quatre principaux types de relations de parenté : les relations père-fille (F-D), mère-fils (M-S), père-fils (F-S) et mère-fille (M-D). Récemment, d'autres types de relations de parenté sont apparus, comme la relation entre les grands-parents et les petits-enfants.

Actuellement, la vérification de parenté est devenue un domaine de recherche très actif avec de nombreuses applications, comme l'organisation d'un album de photos, l'annotation d'images et l'identification de personnes perdues ou recherchées. En outre, la détermination de la parenté ne se limite plus exclusivement à l'analyse génétique elle s'étend au domaine de la biométrie et de la télésurveillance. Le problème de la vérification de la parenté présente de nombreuses difficultés, telles que les variations d'âge, de sexe, d'origine ethnique, les ressemblances entre des personnes sans lien de parenté réel et les différences d'attributs faciaux entre des personnes de la même famille. Pour remédier à cela les chercheurs ont mené de nombreux travaux afin d'introduire un système de vérification de parenté efficace. Dans ce mémoire nous proposons un système de vérification de parenté basé sur quatre étapes principales (1)-la détection de visage (2)-la conversion d'espace de couleur vers HSV (3)-L'extraction des caractéristiques (4) la réduction de dimensionnalité (5)-l'apprentissage basé sur un réseau de neurones convolutif. À la fin nous avons exposé un autre type de système de vérification basé sur le calcul de similarité. Ce mémoire est organisé comme suit : Chapitre 1 : Dans ce chapitre nous allons introduire les concepts de base de la vérification de parenté, ensuite nous allons aborder quelques travaux réalisés dans l'année précédente qui traitent cette problématique. Chapitre 2 : Dans ce chapitre nous allons parler sur les méthodes utilisées, dans les systèmes de vérification de parenté ou on va voir au début, les méthodes de détection de visage les plus utilisées, l'espace colorimétrique, ensuite nous allons énumérer les méthodes

d'extraction et de réduction des caractéristique, pour enfin terminer par définir l'apprentissage profond et les algorithmes de classification. Chapitre 3 dans ce chapitre nous allons expliquer et détaille l'approche qu'on a proposé pour la vérification de parenté, tout en expliquant les méthodes et les techniques utilisé. Chapitre 4 comme dernier chapitre nous allons expliquer la conception et l'implémentation de notre système de vérification de parentées,les testes qu'on a réalisé et les résultat que nous avons obtenu.

Concepts de base pour vérification de parenté

1.1 Introduction

La vérification de la parenté à partir d'images de visages, l'un des nouveaux sujets de la vision par ordinateur, a été étudié et utilisé depuis plusieurs années et peut être appliqué à des applications potentielles telles que la création d'arbres généalogiques, l'organisation d'albums de famille, l'annotation d'images, la recherche d'enfants disparus et la médecine légale. Apprendre et extraire les similitudes de visage entre les membres d'une famille est un défi. De nombreux résultats encourageants ont été obtenus au cours des dernières années. Bien que le test ADN soit le moyen le plus fiable pour la vérification de la parenté, il ne peut pas être utilisé dans de nombreuses situations. La vérification automatique de la parenté à partir d'images faciales peut être effectuée, par exemple : sur des scènes de vidéosurveillance, réseaux sociaux,... En plus de l'obstacle auquel est généralement confrontée la vérification de visage dans des environnements non contraints (c'est-à-dire des images faciales capturées dans des environnements non contrôlés, sans aucune restriction en termes de pose, d'éclairage, d'arrière-plan, d'expression et d'occlusion partielle), la vérification de la parenté ajoute une autre couche d'obstacles qui est loin d'être facile. Elle traite des images faciales qui appartiennent inévitablement à des personnes différentes, avec une différence d'âge considérable et, dans certains cas, avec une différence d'âge considérable et, dans certains cas, un sexe différent. Plus loin, les traits des visages de personnes de la même famille peuvent présenter une grande dissemblance alors que les visages de personnes n'ayant aucun lien de parenté peuvent se ressembler. Tous ces défis augmentent considérablement les difficultés du problème de la vérification automatique de la parenté. A travers ce chapitre, nous allons définir des concepts de base de la vérification de la parenté qui vont nous permettre de mieux comprendre la suite de notre approche.

1.2 Qu'est-ce qu'on appelle la parenté

La parenté est la plus universelle et la plus fondamentale de toutes les relations humaines et se fonde sur les liens du sang, du mariage ou de l'adoption. Il existe deux types fondamentaux de

liens de parenté : Ceux basés sur le sang qui permettent de retracer la descendance. Ceux basés sur le mariage, l'adoption ou d'autres liens. Le système de parenté maintient l'unité, l'harmonie et la coopération entre les relations. La parenté établit des lignes directrices pour la communication et les interactions entre les personnes. Lorsqu'il existe un tabou conjugal, elle décide qui peut épouser qui. La parenté régit le comportement des différents membres de la famille.

1.3 Parenté dans la vision par ordinateur

L'objectif de la vérification de la parenté par la biométrie est de déterminer si une paire donnée d'images de visages de deux personnes a une relation de parenté. Des preuves récentes en psychologie ont démontré que l'apparence du visage est un indice clé pour identifier la l'identification de la parenté , parce qu'en général deux personnes qui sont biologiquement liées ont une plus grande similarité faciale que les autres. Malheureusement, il est même difficile pour les êtres humains de prédire si deux personnes sont parentes ou non en se basant uniquement sur leurs visages, mais ce n'est plus le cas avec les machines. La classification de la parenté à partir d'images de visages est un nouveau problème de reconnaissance des formes et de vision par ordinateur, et elle a de nombreuses applications potentielles dans le monde réel.

1.4 Motivation et application de la parenté

La vérification automatique de la parenté à l'aide d'images faciales a plusieurs applications telles que localiser des parents dans des bases de données publiques, déterminer la parenté d'une victime ou d'un suspect par les forces de l'ordre, le filtrage des demandes d'asile où les liens de parenté doivent être déterminés, l'organisation et la résolution d'identités dans des albums photo. La vérification de la parenté comporte plusieurs aspects sécuritaires : les parents de personnes identifiées comme une menace pour la sécurité peuvent être identifiés en utilisant un cadre de vérification automatique de la parenté. La détermination automatique des informations de parenté peut également être utilisée pour renforcer les capacités de reconnaissance automatique des visages en utilisant les caractéristiques de la parenté comme biométrie douce. D'autre part, la vérification automatique de la parenté dans les vidéos est un domaine de recherche relativement inexploré et peut être très utile dans divers contextes tels que la sécurité, la surveillance et le contrôle de l'immigration. Par exemple, au cours de l'enquête sur des vidéos de surveillance de l'attentat du marathon de Boston, deux suspects masculins ont été identifiés comme étant les auteurs de l'attentat. Plus tard, il a été établi que les deux hommes étaient frères ce qui a conduit à leur identification. Un système de vérification automatique de la parenté qui détermine les liens de parenté dans une vidéo aurait pu accélérer cette enquête. Une autre application de la vérification de la parenté est le contrôle des frontières. à l'aide de vidéos de surveillance qui peuvent être utilisées pour valider le lien de parenté entre un adulte et un enfant, ce qui permet d'éviter le trafic illégal d'enfants. En outre, la vérification de la parenté par vidéo peut valider ou infirmer les demandes

de parenté des réfugiés et des demandeurs d'asile. Actuellement, dans le cadre de son programme de réunification, le département d'État américain effectue des tests ADN pour permettre aux personnes qui ont des parents aux États-Unis d'entrer sur le territoire américain en tant que réfugiés. Rapid-DNA est utilisé à cette fin, mais un algorithme de vérification automatique de la parenté peut produire des résultats rentables en temps réel. Les informations relatives à la parenté peuvent également être utilisées pour gérer le multimédia sur les sites Web de médias sociaux tels que Facebook et Youtube. Dans de nombreux cas, les membres de la famille ont différentes chaînes Youtube où ils mettent en ligne des vidéos quotidiennes. Les informations de parenté peuvent être appliquées pour étiqueter automatiquement ces vidéos et identifier les membres de la famille présents dans ces vidéos. Le contexte de parenté dans les vidéos peut également être utilisé pour l'indexation automatique et l'organisation des vidéos, les rendant ainsi facilement consultables.

1.5 Problèmes de parenté et défis

En particulier, la vérification de la parenté faciale est très difficile. Il existe au moins deux défis majeurs. Le premier est lié à l'environnement de la base de données. Le deuxième est lié à la parenté elle-même [9].

- Le visage des paires (parent-enfant) peut être différent en raison des variations d'âge, de sexe et d'ethnicité mixte, ainsi qu'à d'autres facteurs tels que les variations de pose, l'illumination et l'expression, en particulier lorsque les images sont capturées dans des environnements non contraints.

- Deuxièmement, l'extraction des caractéristiques peut être considérée comme un problème principal pour déterminer les relations. Nous devons d'abord déterminer quelles caractéristiques faciales sont les plus pertinentes pour déterminer les relations parents-enfants. Il est nécessaire de décrire et d'extraire les caractéristiques les plus héritées afin de réaliser un système robuste de vérification de la parenté.

La figure 1.1 montre les différents défis rencontrés lors de la vérification de la parenté.



FIGURE 1.1 – Une illustration des grands défis de la vérification de la parenté dans une scène réelle [Fu et al. (2016)][37]

1.6 Travaux connexe

Dans cette section, nous citons et illustrons quelques articles qui traitent notre problématique. Nous avons ordonné ces travaux en fonction de la méthode utilisée. Comme premier titre, nous avons les méthodes qui apprennent des caractéristiques/traits de parenté pour décrire les images de visage, et comme deuxième titre, il serait basé sur l'apprentissage de métrique, et le dernier titre nous illustrons les articles basés sur un apprentissage profond.

1.6.1 Vérification de la parenté basée sur l'apprentissage des caractéristiques

Fang et al [24] en 2010 ont été les premiers à proposer la vérification de la parenté sur des images faciales. Ils ont utilisé un ensemble de caractéristiques de bas niveau. Ils testent et évaluent la performance individuelle de nombreuses caractéristiques de bas niveau d'images faciales, puis ils sélectionnent les meilleures caractéristiques. Nous divisons ces meilleures caractéristiques en trois parties : couleur, distances des parties du visage, et les histogrammes de gradient. Pour la base de données, ils ont effectué une recherche contrôlée en ligne pour collecter des images du visage frontal de 150 paires de personnages publics et de célébrités qui ont nommé Cornell KinFace et ils ont obtenu une précision de classification de 70,67 % sur leur base de données.

Xia et al. [85] en 2012 ont construit deux bases de données : la première, nommée UB KinFace Ver2.0, qui consiste en des images d'enfants, de leurs jeunes parents et de leurs parents âgés. Parents âgés, et la seconde nommée FamilyFace. De plus, ils ont tout d'abord subdivisé chaque image du visage en régions en cinq parties. L'image totale du visage constitue la première partie. En outre, la

deuxième partie comprend de nombreuses régions : régions supérieure, inférieure, gauche, droite et centrale de l'image du visage. Le front, le nez, les yeux, la bouche et les zones de la joue comprennent la troisième partie et leurs sous-parties plus fines forment la quatrième partie. Enfin, une collection de sous-parties basées sur les quatre points de repère forme la cinquième partie. Après cela, ils ont utilisé les filtres de Gabor sur chacune des parties locales susmentionnées. Les résultats expérimentés sur FamilyFace montrent que la précision moyenne de l'humain sur la relation "fille-père" est de 58,75%, "fille-mère" de 56,25%, "fils-père" de 56,25%, "fils-père" 55,00%, et "fils-mère" 57,50%, respectivement les résultats de la vérification de la parenté ne sont pas fiables de haute qualité, car l'appareil photo numérique n'était pas populaire à l'époque.

Contrairement aux approches susmentionnées, Yan et al [89] en 2015 ont utilisé des caractéristiques de niveau intermédiaire par le biais de caractéristiques de bas niveau au lieu des pixels bruts, où le vecteur de trait appris comprend diverses valeurs de décision d'un hyperplan SVM (machine à vecteurs de support). En outre, un grand ensemble de données faciales non étiquetées et un très petit ensemble de paires de visages étiquetées avec des relations de paires de parenté sont utilisés. Pour augmenter une fonction cible de sorte que les visages avec relation de parenté sont susceptibles d'avoir des scores de décision identiques à partir de l'hyperplan utilisé. Le travail de Yan et al. [89] a proposé une nouvelle approche d'apprentissage discriminatif de caractéristiques basée sur des prototypes (PDFL), pour la vérification de la parenté faciale. Contrairement à la plupart des travaux antérieurs sur la vérification de la parenté qui utilisent des descripteurs de bas niveau fabriqués à la main (peu profonds), notamment binaire local (LBP) et les traits de Gabor pour la représentation faciale, leur travail a pour but d'apprendre une nouvelle méthode plus discriminante pour la vérification de la parenté. L'objectif de leur travail est d'apprendre de nouvelles caractéristiques de niveau intermédiaire plus discriminantes pour décrire parfaitement la relation de parenté des images faciales pour la vérification de parenté. Pour ce faire, ils collectent un groupe d'images faciales avec une relation de parenté non étiquetée provenant du jeu de données LFW (labeled face in the wild) comme ensemble de base. Ensuite, chaque visage de la base de données de formation de la parenté faciale est présenté comme un vecteur de caractéristiques de niveau intermédiaire (trait), où chaque entrée est le score de décision identique d'un hyperplan SVM. Par la suite, ils conçoivent une fonction cible en diminuant les visages intra-classes (avec une relation de parenté) et en augmentant les visages voisins inter-classes (sans relation de parenté) avec les traits de niveau intermédiaire. Enfin, ils utilisent plusieurs traits de bas niveau pour l'apprentissage des caractéristiques de niveau intermédiaire. Par conséquent, ils ont également proposé une PDFL multi-vues (MPDFL) pour apprendre plusieurs traits de niveau intermédiaire afin d'améliorer la performance de la vérification de la parenté. Ils ont comparé leur méthodes (PDFL) et (MPDFL) avec des différents descripteurs des caractéristiques sur les quatre bases de données et le résultat de (PDFL) est comme le suivant : 64.8% KinFaceW-I, 70.2% KinFaceW-II, 70.5% Cornell KinFace, et 63.6% UB KinFace. Le résultat de (MPDFL) : 70.1% KinFaceW-I, 77.0% KinFaceW-II, 71.9% Cornell KinFace, et 67.3% UB KinFace

Plus récemment, Lopez et al [52] en 2016 ont proposé de faire de la prédiction en utilisant

la distance de chrominance. Ils ont proposé d'utiliser la métrique de la distance de chrominance entre chaque paire d'images de visage comme le score de confiance. Ces types d'approches peuvent atteindre une certaine précision, mais il y a l'existence de nombreux problèmes, parmi lesquels l'hypothèse selon laquelle chaque paire de visages relatifs apparentés est découpée à partir de la même photo et que la variation de la pose des images de visages est relativement simple sont généralement des problèmes marquants. Leur résultat a été testé sur les deux bases de KinFaceW. La précision de classification obtenue par leur méthode de notation simple sur KinFaceW-I atteint environ 70 %. Plus important encore, dans l'ensemble de données KinFaceW-II la précision de classification de leur méthode simple atteint 80%.

Wu et al. [?] en 2016 ont réalisé que la généralité des approches proposées pour la vérification de la parenté faciale reposent essentiellement sur le traitement de la luminance (c'est-à-dire de l'échelle de gris) des images faciales, excluant ainsi les données d'information de chrominance (c'est-à-dire de couleur) qui peuvent être une caractéristique supplémentaire puissante pour vérifier la parenté à partir des visages. Leur travail montre pour la première fois que l'information de couleur contient un trait supplémentaire dans la vérification de la parenté à partir d'images de visages. Pour cet objectif, ils ont utilisé des traits conjoints couleur-texture pour décrire à la fois la luminance et la chrominance dans les images faciales en couleur. La performance de la vérification de la parenté utilisant à la fois l'analyse de la couleur et de la texture est comparée aux méthodes équivalentes utilisant uniquement les informations de données en échelle de gris. Parmi leur test en site les résultats test sur le modèle BSIF avec les différents espaces de couleur : 75.78% Gray, 77.94% RGB, 76.96% YCbCr, et 81.47% HSV.

Le résultat du modèle LPQ : 73.53% Gray, 76.67% RGB, 76.57% YCbCr, et 80.10% HSV.

Le résultat du modèle NRML : 73.50% Gray, 74.51% RGB, 77.84% YCbCr, 81.27% HSV.

Wang et al. [83] en 2017 ont utilisé les deux caractéristiques, l'apparence et la géométrie. Pour le trait d'apparence, les images faciales pyramidales sont d'abord construites sur chaque visage pour obtenir des blocs qui se chevauchent, puis l'extraction de caractéristiques avec les trois modèles : local binary pattern (LBP), scale-invariant feature transform (SIFT), Visual Geometry Group face (VGG-FACE) est effectuée dans ces blocs. Un modèle de mélange gaussien est utilisé pour trouver les paires de blocs similaires dans les emplacements correspondants de deux images faciales. Ensuite, l'écart absolu entre deux blocs similaires est calculé en tant que trait d'apparence. Pour le trait géométrique, les points de repère du visage sont d'abord calculés puis projetés dans un nouveau sous-espace appelé le collecteur de Grassmann. Enfin, la métrique topographique entre deux cadres faciaux est calculée comme trait géométrique. Wang et al. [83] ont testé leur méthode sur la base Families In the Wild (FIW) et leur résultat est comme le suivant : LBP 54.69%, SIFT 56.34%, VGG 64.02%

Lan et al. [48] en 2017 ont proposé une approche plus simple et efficace appelée descripteur local quaternionique de Weber (QWLD) pour l'extraction des traits des images de visage en couleur. En combinant la représentation quaternionique (QR) de l'image du visage en couleur et la loi de Weber (WL), le QWLD possède leurs deux propriétés. Il utilise QR pour traiter tous les canaux

de couleur de l'image faciale de manière globale tout en sauvegardant leurs relations (voisinage des différents canaux), et applique la loi de Weber pour garantir que les descripteurs combinés sont plus robustes et plus discriminants. En utilisant l'approche QWLD ils ont découvert le descripteur de Weber basé sur l'incrément quaternionique et le descripteur de Weber basé sur la distance quaternionique, en termes de perspectives multiples. A la fin leur résultat obtenu sur KinFaceW-I est de 76.6% et sur KinFaceW-II est environ 73.6%.

Le travail de Moujahid et al. [5] en 2018 a présenté une nouvelle approche pour la vérification de parenté à base d'images, qui permet de fusionner efficacement les informations locales et globales sur les traits du visage de divers descripteurs. Le cadre proposé repose sur deux étapes principales : (1) ils modélisent les images faciales à l'aide d'une description multi-niveaux pyramidale (PML), où les descripteurs locaux du visage sont extraits par de nombreux blocs à diverses échelles de taille ; (2) ils calculent la covariance (statistiques de second ordre) de diverses caractéristiques locales décrivant chaque bloc individuel dans la description pyramidale PML. Cela permet d'aboutir à un descripteur facial avec deux propriétés efficaces : (i) la description PML, les différentes échelles et les régions du visage sont explicitement fusionnées dans la description finale sans qu'il soit nécessaire de détecter les points de repère du visage ; (ii) le descripteur de covariance décrit les caractéristiques spatiales de tout type en accordant la combinaison de diverses caractéristiques de couleur et de texture de pointe. La précision de classification atteinte par leur méthode sur KinFaceW-I et KinfaceW-II est d'environ 88%.

Mahpod et al. [72] en 2018 ont proposé un réseau hybride d'apprentissage à distance symétrique et asymétrique (CSADL) pour la vérification de la parenté des visages. Les deux descriptions discriminantes sont combinées pour les parents et les enfants à l'aide d'un cadre d'apprentissage par maximisation de la marge, tandis que la vérification de la parenté est considérée comme une tâche de classification résolue par un SVM. La figure 1.2 décrit le réseau hybride d'apprentissage par la distance (HDL) proposé pour la vérification de la parenté faciale. (1) Un ensemble d'apprentissage composé de paires d'images de visages de parents et de leurs enfants. (2) Calcul des descripteurs d'images. (3) Formation du HDL par caractéristique. (4) Application de la projection HDL. (5) Concaténation des représentations multiples apprises. (6) Formation du HDL en utilisant les caractéristiques fusionnées. (7) Représentation fusionnée de la paire d'images d'entrée. (8) Classification SVM à noyau. (9) Résultat de la vérification Kin. Ils ont testé leur méthode sur les deux bases de données Cornell KinFace et KinFaceW I - II

La figure 1.2 représente la méthode utilisée par Mahpod et al. [72] en 2018

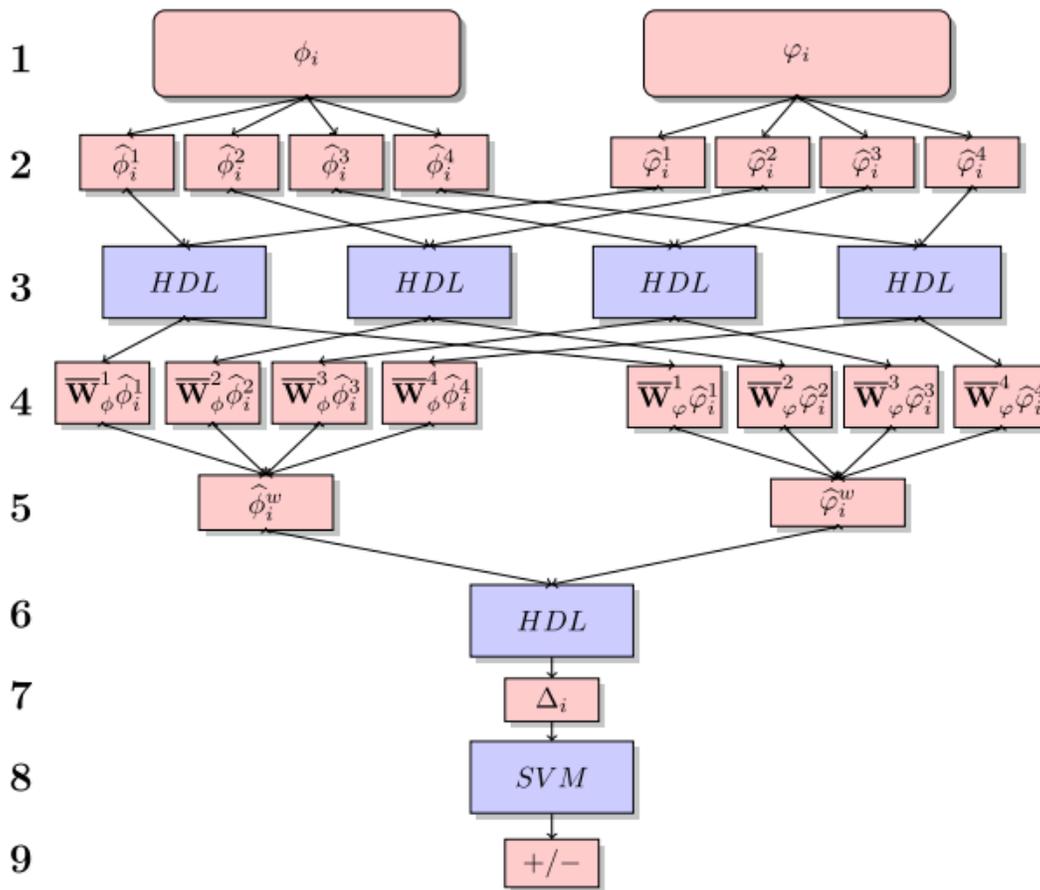


FIGURE 1.2 – Le réseau hybride d’apprentissage à distance (HDL) proposé [72]

Aliradi et al. [67] en 2018 ont proposé un nouveau cadre qui utilise des informations discriminantes sur les données, qui est axée sur l’analyse discriminante exponentielle (DIEDA) fusionnée avec diverses descriptions d’échelle. Les histogrammes de plusieurs blocs sont assemblés pour obtenir un vecteur de caractéristiques de haute dimension, qui démontre un descripteur spécifique de l’échelle. Les histogrammes basés sur les caractéristiques projetées pour chaque région ont utilisé la distance de similarité en cosinus pour minimiser la dimension du vecteur de données des caractéristiques. Enfin, les scores des régions en fonction de plusieurs descripteurs extraits à plusieurs échelles sont ensuite fusionnés et comparés à l’aide d’un classificateur. Leur travail permet d’obtenir des données d’information latérale efficaces pour la comparaison des visages et la vérification de la parenté dans des conditions sauvages (pour décider si les paires d’images faciales sont prises de la même personne ou non). Pour résoudre ce problème, ils prennent des échantillons d’images de visages avec des images de parenté non étiquetées à partir du visage étiqueté dans l’ensemble de données sauvages comme l’ensemble de référence. Ils ont créé une fonction cible optimisée en diminuant les échantillons intra-classe (avec une relation de parenté) et en augmentant les échantillons inter-classe (sans relation de parenté) avec le cadre proposé. KinFaceW-I. Nous avons comparé

les résultats de notre système avec ceux des sept autres systèmes. Nous avons remarqué que leur système proposé avait une précision égale à 80,60% sur KinFaceW-I, 88.60%, sur KinFaceW-II, et $94.50 \pm 0.9\%$ sur la base de données LFW.

Yan et al. [26] en 2019 ont proposé une nouvelle approche d'apprentissage de caractéristiques faiblement supervisée (semi-supervisée) appelée descripteur de visage binaire compact discriminatif (D-CBFD) pour la vérification de la parenté des visages. Contrairement aux approches précédentes de vérification de la parenté où des caractéristiques artisanales (peu profondes) sont utilisées pour la description du visage, leur D-CBFD effectue une description efficace du visage à partir d'un ensemble d'échantillons de données faiblement étiquetés. Étant donné une image faciale, ils ont d'abord calculé les vecteurs de différence de pixels (PDV) dans différentes régions locales. Ensuite, ils apprennent un espace de projection efficace pour cartographier chaque PDV et les projeter dans un nouvel espace discriminant de caractéristiques binaires à faible dimension, où l'information totale des données d'énergie totale du PDV devrait être bien sauvegardée et la métrique des paires positives est plus petite et celle des paires négatives est plus grande. Enfin, ils regroupent tous les vecteurs de caractéristiques binaires sur chaque visage en un histogramme d'extraction de caractéristiques comme description finale. Ils ont testé leur méthode sur les ensembles de données KinFaceW-I, KinFaceW-II et KVFV. Les auteurs ont constaté que leur D-CBFD surpasse les autres méthodes (LBP, SDIFT, HOG, LPQ, CBFD) de représentation des caractéristiques comparées sur tous les ensembles de données car il s'agit d'une approche d'apprentissage de caractéristiques discriminantes, ce qui permet d'exploiter au mieux les informations discriminantes et adaptées aux données.

Goyal et al. [3] en 2020 ont proposé une nouvelle approche de vérification de la parenté faciale basée sur l'excentricité (EKV) pour montrer la puissance des régions dominantes du visage pour la vérification de la parenté. L'approche EKV proposée utilise l'excentricité des régions dominantes du visage approchées par une ellipse comme paramètre efficace pour décrire les images faciales pour la vérification de la parenté. Elle présente deux cadres essentiels, appelés excentricité simple (SE) et excentricité fusionnée (FE). Le cadre SE pour l'approche EKV a construit une formulation unique en utilisant une seule région de visage. Pour chaque région de visage utilisée, elle est approximée comme une ellipse pour calculer le paramètre d'excentricité et mettre en œuvre la vérification. Ensuite, le cadre FE pour l'approche EKV a utilisé une description multi-vues en utilisant deux ou plusieurs régions de visage. L'excentricité des différentes régions du visage approchées sous forme d'ellipse est calculée et combinée pour former un paramètre converti et mettre en œuvre la vérification. Des expériences approfondies sont sur des bases de données de parenté (Cornell kinface, KinFaceW-I et II, UB KinFace, TsKinFace, FIW), et les résultats montrent que le schéma FE pour EKV a atteint une précision compétitive sur toutes les bases de données comme Cornell kinface 86.27%, KinFaceW-I 90.30%, KinFaceW-II 90.15%, UB KinFace 85.03%, TsKinFace 90.85%, FIW 87.87%. Plus précisément, les meilleurs résultats sont obtenus pour la fusion de deux sections du visage, à savoir la bouche et l'œil gauche (FMLE2).

Comme deuxième titre dans cette section nous montrons et décrivons les méthodes de pointe

qui apprennent une distance métrique par une stratégie d'apprentissage métrique proposée pour la tâche de vérification de la parenté.

1.6.2 Vérification de la parenté basée sur l'apprentissage de métrique

Lu et al. [55] en 2014 ont construit deux bases de données de parenté, nommées KinFaceW-I et KinFaceW-II, à partir des recherches sur Internet, où des images de visages ont été capturées dans des conditions non contrôlées et ils ont proposé une nouvelle approche d'apprentissage métrique repoussé par le voisinage (NRML) pour la vérification de la parenté faciale. Encouragés par la conviction que les échantillons de visages interclasses (sans relation de parenté) qui présentent une grande similarité se situent en fait dans un voisinage et sont plus facilement mal classés que par rapport à celles qui ont moins de similarité. Leur but est de découvrir une distance métrique par laquelle les échantillons d'images de visages intra-classes (avec une relation de parenté) sont rapprochés le plus possible et les échantillons d'images de visages inter-classes se trouvant dans un voisinage sont repoussés et éloignés aussi loin que possible. En outre, de telle sorte que des informations plus efficaces peuvent être extraites pour la vérification de la parenté. Pour montrer une meilleure utilisation de la description de caractéristiques multiples et d'extraire des informations intégrales, ils ont proposé une méthode NRML multivues (MNRML) pour calculer une distance métrique commune pour combiner des caractéristiques multiples dans un sous-espace de fusion afin d'améliorer la performance de la vérification de la parenté faciale. Ils ont mené des expériences de vérification de parenté sur deux autres ensembles de données : Cornell KinFace et UB KinFace. Les méthodes NRML et MNRML que ils ont proposée sont plus performantes que les autres méthodes d'apprentissage métrique comparées pour la tâches de vérification de parenté.

Zhou et al. [88] en 2019 ont proposé une nouvelle approche d'apprentissage métrique de la parenté (KML) avec un modèle de réseau neuronal profond (DNN) fusionné. Comme mentionné dans la figure 1.3, le KML modélise clairement la contradiction entre générations inhérente aux images de visages de paires parents-enfants, et apprend une métrique de similarité profonde fusionnée de telle sorte que les paires d'images faciales avec une relation de parenté sont rapprochées, tandis que celles qui n'ont pas de relation de parenté (mais avec une grande similarité d'apparence) sont rejetées (poussées aussi loin que possible).

En outre, en évaluant l'assortiment intra-connexion et l'uniformité inter-connexion sur le DNN fusionné, ils présentent la propriété de compacité hiérarchique dans le réseau fusionné pour faciliter l'apprentissage métrique profond avec un nombre fini de connexions.

Pour évaluer l'efficacité de la méthode de vérification de la parenté, ils ont réalisé des expériences sur quatre jeux de données largement utilisés : KinFaceW-I, KinFaceW-II, Cornell KinFace, et UB KinFace.

Le résultat de la méthode KML est comme le suivant 82.8% sur KinFaceW-I, 85.7% KinFaceW-II, 81.4% Cornell KinFace, 75.5% UB KinFace.

Comme l'illustre la figure 1.3, leur algorithme de classification KinNet est composé d'une paire

de réseaux neuronaux (Parent, Child) qui sont respectivement conçus pour les parents et les enfants avec des paramètres de modèle et des sorties différents.

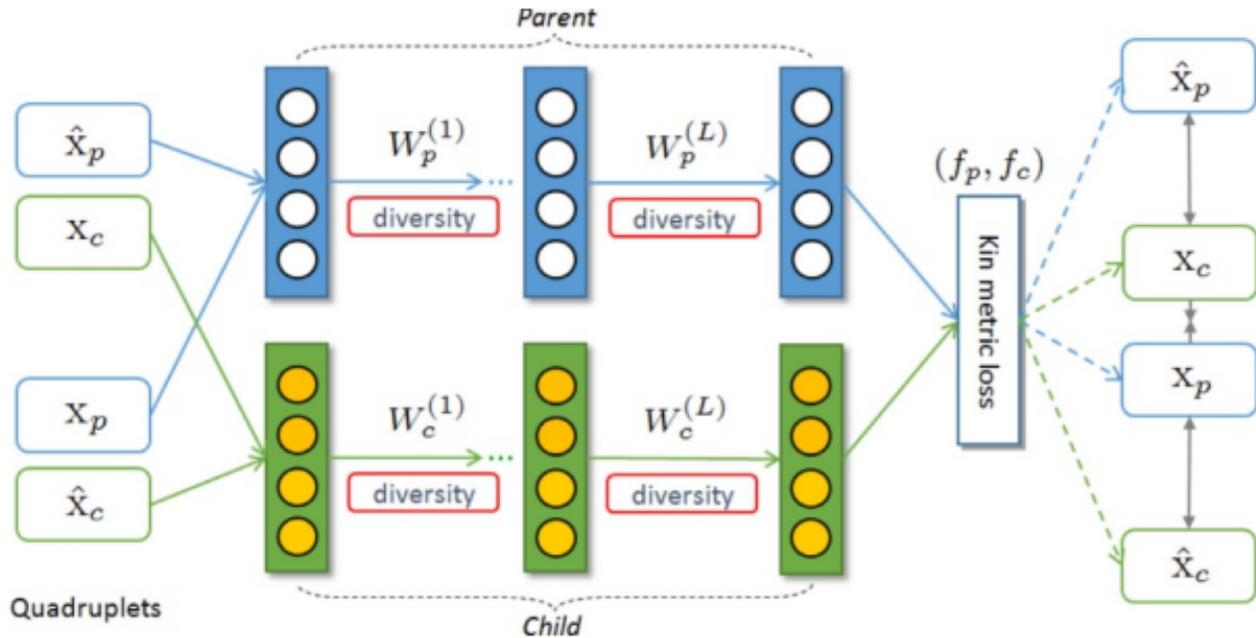


FIGURE 1.3 – La méta-vue de la méthode KML avec une architecture profonde soigneusement conçue KinNet [88]

Le travail de Bessaoudi et al. [59] en 2019 a proposé un cadre basé sur l'apprentissage métrique par tenseur (tenseur d'ordre élevé) pour la conception d'images faciales. Le tenseur facial est structuré à partir des descripteurs de texture locaux extraits de multi-échelles. En outre, ils ont proposé une nouvelle analyse discriminante basée sur l'information latérale multilinéaire (MSIDA) pour traiter la réduction et la classification semi-supervisée des projections de sous-espaces multilinéaires. En utilisant uniquement les données d'information faiblement étiquetées, MSIDA permet de projeter le tenseur facial d'entrée dans un sous-espace discriminant défini par l'analyse des sous-espaces tensoriels dans laquelle la discrimination est améliorée et la taille (dimension) de chaque mode de tenseur est minimisée simultanément.

Comme le montre la figure 1.4, le schéma fonctionnel de l'approche MSIDA est constitué de trois composants fondamentaux : l'extraction de caractéristiques, la transformation du sous-espace tensoriel et la comparaison.

Chaque image faciale apparaît par deux descripteurs de texture locaux, MSLPQ et MSBSIF, exécutés à différentes échelles en soumettant de nombreux vecteurs de caractéristiques de chaque image de visage. Les vecteurs de caractéristiques de l'ensemble des visages formés sont empilés sous la forme d'un tenseur de troisième ordre (i_1, i_2, i_3) , où i_1 représente la longueur du vecteur de caractéristique unique, i_2 représente les divers descripteurs de texture locale extraits de chaque image de visage et i_3 représente les échantillons d'images faciales contenus dans l'ensemble de données d'entraînement. Le tenseur construit est d'abord projeté par MPCA [35] pour minimiser

les dimensions du sous-espace à $j_1 \times j_2 \times i_3$, où $j_1 \times j_2 \ll i_2 \times i_2$. La cause de l'effet MPCA avant MSIDA est de traiter le problème de la petite taille de l'échantillon dans plusieurs modes tensoriels.

Ce problème apparaît lorsque la longueur de la dimension du vecteur de caractéristiques est plus grande que le nombre d'images de visage des échantillons de formation, ce qui entraîne la singularité des différentes matrices de diffusion MSIDA. La minimisation de la longueur de dimension de chaque mode tensoriel est donc appliquée en premier lieu.

Après avoir effectué la MPCA, le tenseur des données d'information d'entraînement a été divisé en deux sous-tenseurs selon les paires positives (paires correspondantes) et les paires négatives (paires non correspondantes), respectivement. La subdivision a été effectuée en fonction du troisième mode tenseur i_3 . Le tenseur positif a été utilisé pour calculer la matrice de dispersion intra-classe $(S^{msida})_w$ et le tenseur négatif a été utilisé pour calculer la matrice de diffusion entre classes $(S^{msida})_b$ de la méthode MSIDA.

Le tenseur des données d'information a été projeté à travers les sous-espaces MSIDA pour obtenir un tenseur inférieur et plus discriminant $k_1 \times k_2$, où $k_1 \times k_2 \ll j_2 \times j_2$. Dans la phase de test, chacune des images faciales de la paire a été vérifiée ; la correspondance a été représentée comme un tenseur de second ordre défini par l'empilement des descripteurs de texture locaux de l'image faciale. De plus, les deux tenseurs ont été projetés par la méthode MPCA et ensuite par la méthode MSIDA. Enfin, la distance de similarité en cosinus entre la paire de test a été calculée et utilisée pour décider si la paire est positive (appartenant à la même personne/famille) ou non. ils ont comparé les performances de vérification obtenues par l'approche proposée avec les méthodes de leur état de l'art sur la base de données Bosphorus 3D. Le meilleur taux de vérification de leur approche a atteint 92.12%.

La figure 1.4 représente le schéma fonctionnel de l'approche MSIDA.

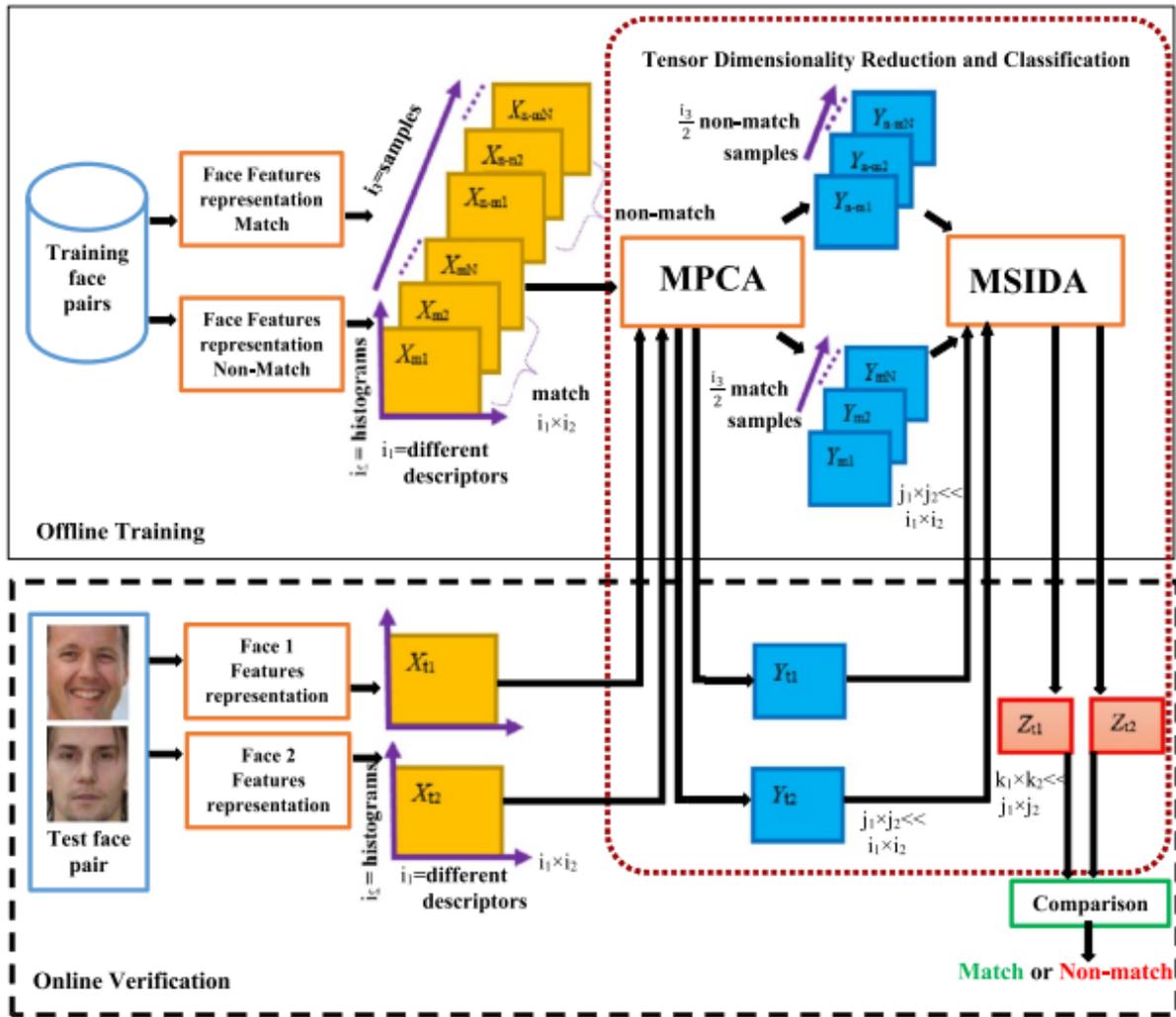


FIGURE 1.4 – Schéma de principe du système de comparaison de paires de visages MSIDA [62].

Le travail de Dornaika et al. [23] en 2020 a introduit un nouveau schéma qui extrait les traits profonds du visage pour la vérification de la parenté. L'approche fusionne une sélection efficace des caractéristiques et une projection de l'information des données proéminentes orientée vers la parenté. Le cadre présenté comprend trois étapes de fusion : (1) une fusion précoce des descripteurs de caractéristiques où le filtre de sélection choisit les caractéristiques profondes les plus discriminantes, (2) une fusion de niveau intermédiaire qui utilise une méthode d'apprentissage métrique multi-vues basée sur la parenté (MNRML), et (3) une fusion de niveau tardif qui combine les réponses des classificateurs (SVM). Dans leur travail, les caractéristiques faciales sont obtenues par le réseaux de neurones convolutifs profonds pré-entraînés VGG-F et VGG-Face, qui ont été essentiellement proposés pour la classification de groupes d'objets et d'identités, respectivement. Ils se concentrent sur quatre relations de parenté. Ces quatre relations sont les suivantes : Père-Fils (F-S), père-fille (F-D), mère-fils (M-S) et mère-fille (M-D). L'entrée est une paire d'images faciales. La première

image a été donnée pour décrire un enfant et la seconde pour le parent. En outre, l'image faciale d'un enfant correspond soit à un enfant, soit à un adulte. Le parent image faciale selon un jeune adulte ou une personne âgée. La confirmation que l'ordre de positionnement des paires dans les images faciales n'est pas important. Par conséquent, le résultat de leur cadre proposé était une décision binaire qui vérifie la relation de parenté faciale. Étant donné une paire d'images faciales (parent-enfant), les descriptions des caractéristiques faciales profondes par VGG-F et VGG-Face sont définies comme les vecteurs p , c et p' et c' , respectivement. Chaque vecteur était $\in R^{4096}$. Le traitement a montré que les informations de données jetées sur un pipeline. Résultats du cadre proposé comparé aux approches citées dans leur état de l'art sur les bases de données KinFace-I et KinFace-II est de 84.55% , 86.90% respectivement, de mieux que les méthodes comparées .

Dans le titre qui suit, nous avons présenté les méthodes basées essentiellement sur des réseaux de neurones convolutifs profonds au problème de la vérification de la parenté.

1.6.3 Vérification de la parenté basée sur l'apprentissage profond convolutif

Lu et al. [54] en 2017 ont proposé une approche DDML pour entraîner un réseau neuronal profond qui peut apprendre un groupe de sous-espaces de transformations non linéaires hiérarchiques pour projeter des images faciales dans le même espace de caractéristiques implicites, dans lequel la métrique de chaque paire positive est minimisée et celle de chaque paire négative est maximisée. positive est minimisée et celle de chaque paire négative est maximisée, respectivement. Pour mieux exploiter les points communs des différents descripteurs de caractéristiques et de rendre toutes les caractéristiques plus efficaces pour la vérification faciale de parenté, ils ont mis au point une méthode efficace d'apprentissage multi-métrique profond (DDMML) pour combiner les caractéristiques de chaque paire pour apprendre ensemble plusieurs réseaux neuronaux, ce qui permet d'augmenter la corrélation entre les différentes caractéristiques de chaque échantillon, la métrique de chaque paire positive est diminuée et celle de chaque paire négative est maximisée, respectivement. Le DDML apprend un réseau neuronal à partir d'une description de caractéristiques monoculaires et ne peut pas traiter directement diverses descriptions de caractéristiques. Dans la vérification du visage et de la parenté, il suffit d'extraire diverses caractéristiques pour chaque image faciale afin de procéder à diverses fusions de caractéristiques. Cependant, les caractéristiques extraites de la même image faciale sont généralement extrêmement corrélées entre elles, même si elles peuvent décrire les images faciales sous plusieurs aspects [91]. Pour la fusion de diverses caractéristiques, ces données d'information extrêmement corrélées doivent être conservées car elles reflètent généralement les caractéristiques fondamentales de l'image. Un principe important pour l'utilisation de la fusion de plusieurs caractéristiques est d'apprendre simultanément plusieurs métriques de distance en sauvegardant la corrélation entre diverses paires de caractéristiques d'images faciales.

Duan et al. [21] en 2017 ont introduit un cadre d'apprentissage de transfert grossier pour la vérification de la parenté et ont montré une amélioration sur trois bases de données de parenté différentes (UB ,Cornell KinFace,KinfaceW I-II). Le réseau initial grossier est entraîné sur la base

de données CASIA-WebFace pour la reconnaissance des visages avec 10 575 classes. Il est important de noter que le réseau formé est affiné pour la classification de la parenté, chaque paire positive étant étiquetée comme une classe distincte sur les plus petits ensembles de données de parenté. Ils ont eu comme résultat avec la méthode (Coarse to fine transferring) CFT : Cornell KinFace 78.6% UB 72.3%, KinFaceW I 75.9% , KinFaceW II 76.1% .

Zhang et al. [40] en 2015 ont proposé d'apprendre des descriptions de caractéristiques de haut niveau pour la vérification de parenté en se basant sur des réseaux neuronaux convolutifs profonds. Leur méthode est de bout en bout, avec un prétraitement non complexe souvent utilisé dans les approches traditionnelles. Les descriptions des caractéristiques de haut niveau sont issues des activations des neurones de la dernière couche cachée, puis introduites dans un classificateur softmax pour vérifier la parenté de deux images faciales. En considérant l'importance des points clés du visage, ils ont également extrait des descriptions de fonctionnalités pour la vérification de la parenté. Deux ensembles de données de parenté sont utilisés pour évaluer leur méthode (KinFaceW-I, KinFaceW-II) , et ils ont eu comme résultats avec la méthode CNN-Basic 74.8% sur la base KinFaceW-I, et 85.3% sur KinFaceW-II , et pour la méthode CNN-Points 77.5% KinFaceW-I, 88.4% KinFaceW-II.

Li et al [90] en 2017 ont introduit une approche d'apprentissage par transfert où un réseau a été formé pour la classification des visages, puis affiné pour la vérification de la parenté. Quatre réseaux résiduels différents ont été entraînés sur la base de données MS-Celeb pour la reconnaissance des visages et l'ensemble a été ré entraîné pour la vérification de la parenté en utilisant une perte triplée douce. Ils ont montré des résultats de 74.85% sur Families in the Wild database .

Kohli et al. [44] en 2017 ont réalisé une étude humaine visant à comprendre les capacités de perception humaine et d'identifier les régions discriminées d'un visage qui lisent les indices de parenté. Les stimuli visuels proposés aux participants permettent de définir leur capacité d'identifier les relations de parenté en utilisant l'ensemble du visage ainsi que des parties particulières du visage. L'influence de l'âge et du sexe des participants et des paires de parenté du stimulus est étudiée en utilisant des mesures quantitatives telles que la précision, l'indice de discrimination et l'entropie de l'information perceptive. En utilisant les données d'information acquises de l'étude humaine, une approche hiérarchique de vérification de la parenté par apprentissage de représentation (KVRL) a été utilisée pour mieux apprendre la description de diverses parties du visage d'une manière non supervisée. Elles sont présentes dans les images faciales en utilisant des filtres et une pénalité de régularisation contractive. Une description compacte des images de visages de la parenté a été utilisée comme une sortie du modèle appris et un réseau neuronal multicouche a été utilisé pour vérifier la parenté avec précision. Une nouvelle base de données de parenté WVU a été collectée, qui comprend plusieurs images faciales par personne pour faciliter la vérification de la parenté. Les résultats concluent que l'approche d'apprentissage profond proposée (KVRL-fcDBN) surpasse la précision de vérification de la parenté de l'état de l'art sur la base de données de parenté de la WVU, et sur quatre bases de données de référence existantes. En outre, les données d'information sur la parenté ont été utilisées comme une qualité biométrique douce pour augmenter

la performance de la vérification faciale par le produit du rapport de vraisemblance et des méthodes basées sur la machine à vecteurs de support. Ils ont proposé une nouvelle méthode de description des caractéristiques, appelée réseaux de croyance profonds (fcDBN). La description des caractéristiques proposée combine des données d'informations relationnelles. ils ont testé leur méthode sur les base de donnees suivante : Cornell KinFace, UB, KinFace-I, KinFace-II, WVU. Ils ont eu comme résultat : 89.5 % Cornell KinFace, 91.8% UB, 96.1 % KinFace-I, 96.2 % KinFace-II, 90.8 % avec la base WVU.

Yan et al. [28] en 2019 ont présenté une approche pour la vérification de la parenté des visages, qui utilise un réseau d'attention pour se concentrer sur l'obtention d'informations discriminantes des régions locales du visage. Contrairement à la plupart des méthodes existantes qui utilisent des descriptions de caractéristiques de bas niveau pour la vérification de la parenté, ils ont introduit un mécanisme d'attention dans le réseau profond pour obtenir des caractéristiques de haut niveau pour la description du visage. Ils ont également proposé une méthode auto-supervisée pour orienter le réseau d'attention. En outre, ils incluent de manière aléatoire un masque dans cinq régions de chaque visage pour aider le réseau à se concentrer sur l'obtention d'informations plus efficaces dans ces groupes de régions. ils ont testé leur méthode sur deux base de donnee et ils ont eu comme résultat 81.0 % sur KinFaceW-I, et 91.6% sur la base KinFaceW-II.

Kohli et al. [45] en 2019 ont proposé une nouvelle approche d'apprentissage profond pour la vérification de la parenté faciale dans des vidéos sans contrainte en utilisant un nouvel autoencodeur de normes mixtes supervisées (SMNAE). Cette nouvelle formalisation de l'autoencodeur présente la sparsité spécifique à la classe dans le poids de la matrice. L'approche proposée de vérification de la parenté faciale basée sur le SMNAE en trois étapes utilise la description spatio-temporelle apprise dans les images vidéo fixes pour vérifier la parenté dans une paire de vidéos. Un nouvel ensemble de vidéos de parenté (KIVI) de plus de 500 personnes avec des variations dues à l'occlusion, la pose, l'illumination, l'ethnicité et l'expression pour leur recherche. Il comprend essentiellement 355 paires de vidéos de parenté positives avec plus de 250 000 images. L'efficacité de l'approche proposée a été appliquée et réalisée sur le jeu de données KIVI et six autres jeux de données. de parenté faciale existants. Sur le jeu de données KIVI, SMNAE Ils ont obtenu une précision de vérification de la parenté basée sur la vidéo de 83,18 %, soit au moins 3,2 % de mieux que les méthodes existantes. L'approche a également été testée sur six bases de données publiques de parenté faciale et a obtenu les meilleurs résultats.

Zhang et al. [93] en 2021 montrent que la plupart des approches existantes pour la vérification de la parenté faciale peuvent être subdivisées en méthodes d'apprentissage superficiel basées sur les caractéristiques artisanales et en méthodes d'apprentissage profond basées sur les réseaux neuronaux convolutifs (CNN). En outre, ces approches traitent toujours le problème difficile de l'identification des traits de parenté à partir d'images de visage. En outre, la raison en est que toutes les données d'information sur les identifiants familiaux et la distribution diverse de la parenté par paires à partir de visages sont rarement prises en considération dans les problèmes de vérification de la parenté faciale. Par conséquent, une approche de réseau convolutif adversatif de données basée sur l'identification de la famille (AdvKin), principalement axée sur les traits de parenté

efficaces, a été proposée pour les deux types de bases de données de parenté (vérification de la parenté faciale à grande échelle et à petite échelle). Les avantages du cadre qu'ils proposent sont de quatre ordres : 1) pour les relations de parenté pour la détection des relations de parenté, un paradigme auto-adversarial plus simple mais efficace basé sur une perte maximale négative négative (NMMD) a été proposé comme attaque dans la première couche entièrement connectée (FC-1). ; 2) une perte contrastive par paire et une perte softmax basée sur l'ID de la famille sont alors ensemble formulées dans les deuxième et troisième couches entièrement connectées (FC-2 et FC3), respectivement, pour l'apprentissage supervisé de stratégies. 3) un schéma de réseau à deux flux avec des connexions résiduelles a été proposé dans AdvKin 4) pour une augmentation plus fine des traits de parenté, ensemble de réseaux AdvKin parcellaires a été proposé (E-AdvKin). ils ont testé leur méthode sur les deux base de donnée KinFaceW-I et KinFaceW-II et ils ont eu comme résultat 78.7% sur le jeu de data KinFaceW-I et 88.0 % sur KinFaceW-II.

1.7 Conclusion

Dans ce chapitre, nous avons donné un aperçu de la vérification automatique de la parenté à partir de visages. Nous avons également présenté des travaux connexes et un aperçu des approches de vérification automatique de la parenté. Nous avons discuté de certaines méthodes utilisées, divisées en deux catégories : les méthodes basées sur des caractéristiques artisanales, et les méthodes basées sur des caractéristiques d'apprentissage métrique et celles basées sur des caractéristiques profond. Dans le prochain chapitre, nous allons présenter quelques notions et plusieurs définitions de la conception des systèmes de vérification de la parenté.

Méthodes de vérification de parenté

2.1 Introduction

Dans ce chapitre, nous nous intéressons à les définitions des espaces de couleurs et des méthodes de détection de visage, d'extraction de caractéristiques et de classification sont abordées. Une image faciale est un indice crucial qui contient de nombreuses caractéristiques humaines utiles, comme l'identité, la race, le sexe, l'âge, l'expression, l'ethnicité, etc. Quels types de caractéristiques aident à identifier les relations et comment les caractéristiques sont représentées pour déterminer la relation ?, cela devient donc un problème majeur et un défi important dans la vérification de la parenté. Il est nécessaire de se concentrer sur l'étape d'extraction des caractéristiques car elle a un impact notable sur la performance de vérification de parenté. Ce chapitre commence par présenter les espaces de couleur et les méthodes de détection de visage qui représente la première étape de notre système . Ensuite nous concentrons nos recherches sur les méthodes d'extraction de caractéristique notamment les méthode basé sur les caractéristiques artisanales (handcraft feautres) et les méthode basé sur les caractéristiques profond(Deep features). En suit nous avons les Méthodes de réduction de caractéristiques qui vont nous permettre de réduire le nombre de caractéristiques sans perdre d'informations importantes , Enfin nous avons abordé l'apprentissage profond, les réseaux de neurones convolutif et les méthodes de classification qui sont considéré comme dernières étapes de notre système qui vont nous mener à atteindre notre objectif qui est la vérification de parenté

2.2 Detection de visage

La détection des visages également appelée détection faciale est une technologie informatique basée sur l'intelligence artificielle (IA) utilisée pour trouver et identifier des visages humains dans des images numériques. Elle joue désormais un rôle important en tant que première étape dans de nombreuses applications clés, notamment le suivi des visages, l'analyse des visages et la reconnaissance faciale. La détection des visages a un effet significatif sur la façon dont les opérations séquentielles se dérouleront dans l'application. Une grande variété de techniques a été proposée, allant des algorithmes simples basés sur les bords aux approches composites de haut niveau utilisant des méthodes avancées de reconnaissance des formes. Dans la section qui suit nous allons

définir quelques méthodes de détection de visage existantes.

2.2.1 méthodes de détection de visage

La détection du visage est un problème trivial qui a été résolu de manière satisfaisantes par une multitude de technique tel que Le traitement automatique du visage, Méthode LBP (Local Binary Patterns) , Eigenface , L'analyse des points particuliers et les trois techniques les plus utilisé quand vas détailler par la suite : YOLO, MTCNN, les filters de haar.

2.2.1.1 YOLO

YOLO [69] est un cadre d'apprentissage profond à la pointe de la technologie pour détection d'objets en temps réel. Il s'agit d'un modèle amélioré qui a obtenu de meilleures performances par rapport au détecteur à base de régions qui a surpassé les performances des jeux de données de détection standard comme PASCAL VOC [56] et COCO [78]. Il traite la détection d'objets comme un problème de régression plutôt que comme un problème de classification. Il réalise toutes les étapes essentielles à la détection d'un objet à l'aide d'un seul réseau neuronal unique. Par conséquent, il atteint non seulement de très bonnes performances de détection, mais aussi une vitesse en temps réel. En outre, il possède une excellente capacité de généralisation et peut être facilement entraîné pour détecter différents objets. Le fonctionnement de YOLO est le suivant : nous prenons une image et la divisons en une grille $S \times S$. Dans chacune des grilles, nous prenons m boîtes englobantes. Pour chaque boîte de délimitation, le réseau produit une probabilité de classe et des valeurs de décalage pour la boîte de délimitation. Les boîtes de délimitation dont la probabilité de classe est supérieure à une valeur seuil sont sélectionnées et utilisées pour localiser l'objet dans l'image. YOLO est beaucoup plus rapide (45 images par seconde) que les autres algorithmes de détection d'objets. La limitation de l'algorithme YOLO est qu'il a du mal avec les petits objets dans l'image, par exemple il pourrait avoir des difficultés à détecter une volée d'oiseaux. Ceci est dû aux contraintes spatiales de l'algorithme.

La figure 2.1 représente architecture de yolo

YOLO: You Only Look Once

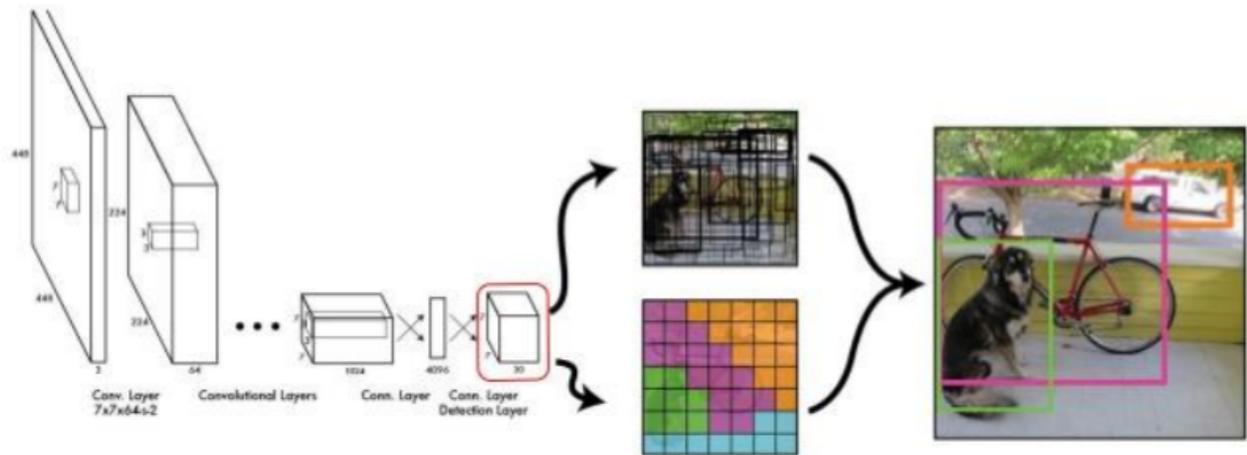


FIGURE 2.1 – architecture de yolo

La figure 2.2 représente l'application du filtre YOLO sur une image contenant des objet : chien ,vélo et camion.

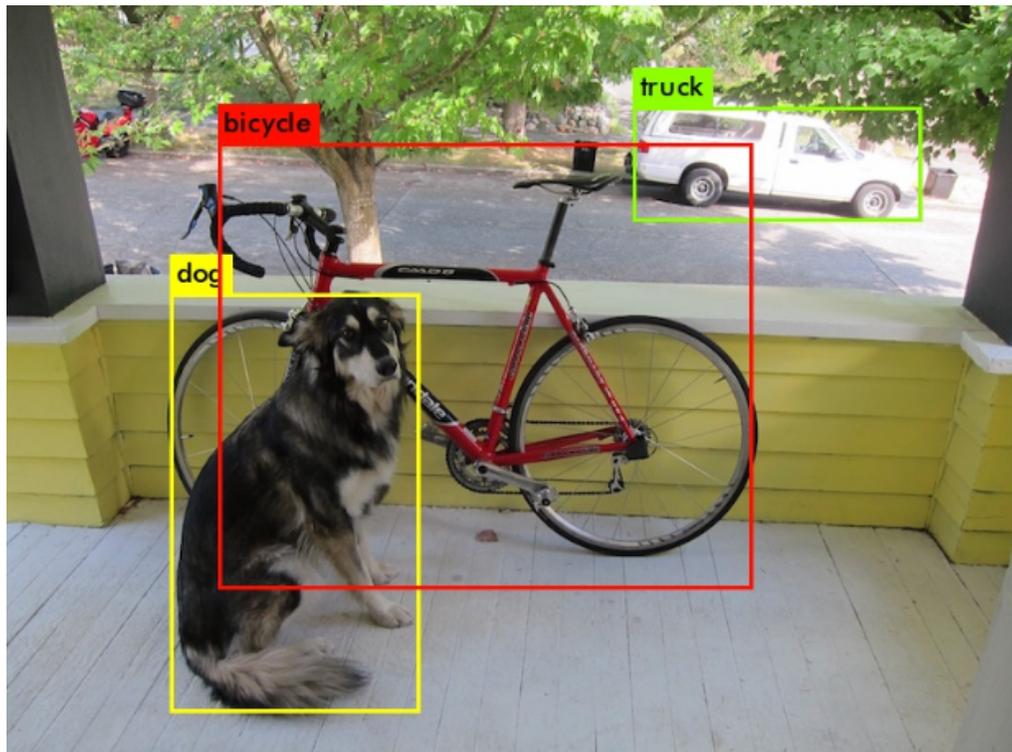


FIGURE 2.2 – Application du filtre YOLO sur une image

2.2.1.2 MTCNN

Les réseaux convolutifs multi-tâches en cascade (MTCNN) [51] sont un cadre développé comme solution pour la détection et l'alignement des visages. Le processus consiste en trois étapes de réseaux convolutifs qui sont capables de reconnaître les visages et l'emplacement des points de repère tels que les yeux, le nez et la bouche. Dans la première étape, il utilise un CNN peu profond pour produire rapidement des fenêtres candidates. Dans la deuxième étape, il affine les fenêtres candidates proposées à l'aide d'un CNN plus complexe. Et enfin, dans la troisième étape, il utilise un troisième CNN, plus complexe que les autres, pour affiner encore le résultat et produire des positions de repères faciaux. La tâche du réseau est de produire trois résultats : classification des visages et des non-visages, régression de la boîte de délimitation et localisation des points de repère du visage. Le MTCNN se compose de trois cascades de réseaux neuronaux, à savoir le P-Net, le R-Net et l'O-Net. Afin de parvenir à une reconnaissance des visages à une échelle unifiée, l'image originale doit être mise en différentes échelles pour former une pyramide avant d'utiliser ces réseaux comme montré dans la figure 2.3.

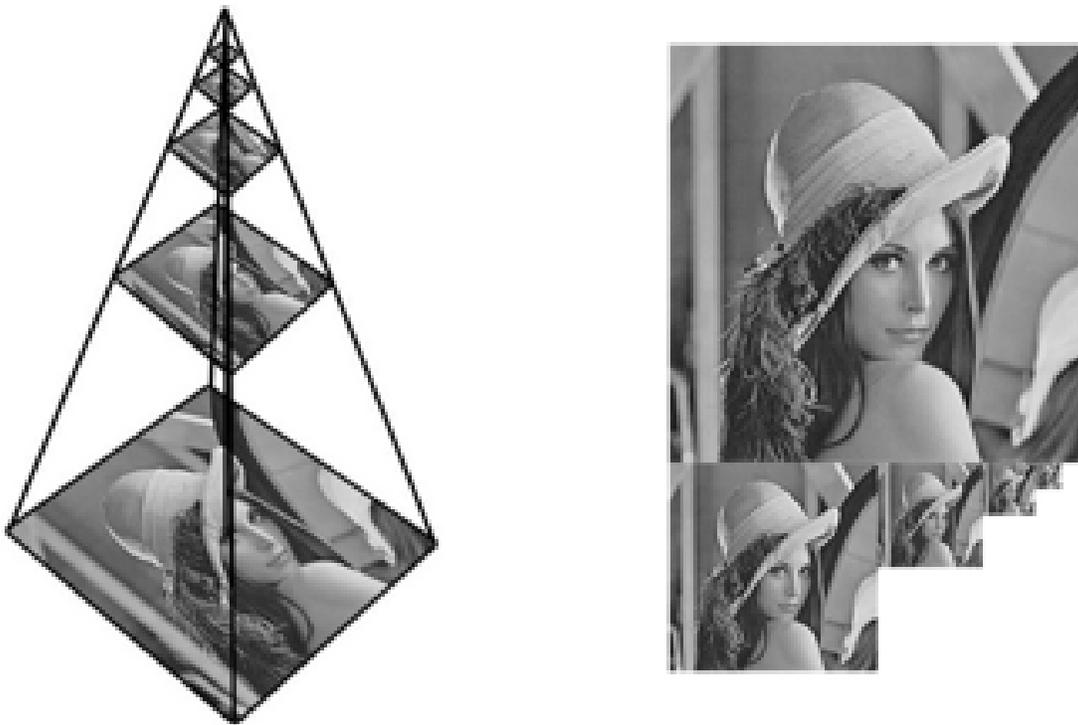


FIGURE 2.3 – L'image d'entrée est redimensionnée à différentes échelles pour construire une pyramide d'images

Le premier réseau, P-Net, est un réseau convolutif complet utilisé pour générer des vecteurs de régression de fenêtre et de frontières candidats. La régression par boîte englobante est utilisée pour corriger les boîtes candidates, puis les non-maxima sont utilisés pour supprimer ces boîtes candidates combinées qui se chevauchent. Les résultats du P-Net sont relativement grossiers, de sorte

qu'un réglage supplémentaire est effectué à l'aide du R-Net. Cette structure est très similaire à celle du P-Net. Il entrera dans le R-Net par la fenêtre candidate de P-Net, rejette la plupart des fausses fenêtres et continue à utiliser la régression par boîte englobante et la fusion des NMS. Enfin, O-Net est utilisé pour produire le cadre final du visage et les positions des points caractéristiques similaires aux deux premières étapes, la différence est de générer 5 positions de points caractéristiques.

Dans notre travail nous allons utiliser la détection de visage comme une étape préliminaire afin de localiser et d'extraire la région du visage de l'arrière-plan ce qui nous servira dans la suite de notre procédure vu la vitesse et la précision de détection des cascades de haar nous avons choisi de l'utiliser comme méthode de détection de visage.

2.2.1.3 Filtre de Haar

Viola et Jones [80] ont fourni une méthode rapide et efficace pour détecter un visage dans une image donnée. Elle est basée sur des caractéristiques de type Haar et un classificateur AdaBoost en cascade. La détection du visage dans ce filtre est réalisée par un filtre multi-échelles de Haar. Les propriétés d'un visage sont décrites dans un fichier XML. Elles ne sont pas choisies au hasard et reposent sur un échantillon de quelques centaines d'images tests [64]. Il s'agit du premier cadre de détection de visage capable de fournir des performances en temps réel, de nombreuses applications de traitement d'images qui requièrent des visages en entrée sont construites à l'aide de cet algorithme. C'est l'algorithme de détection de visages le plus utilisé. Cet algorithme se compose principalement de trois étapes de base. Dans la première étape, un classificateur Haar est entraîné en utilisant des milliers d'images positives et négatives qui servent de modèles pour détecter les visages. La formation du classificateur Haar est suivie de la formation du classificateur en cascade qui permet une détection efficace des visages pendant l'exécution. Enfin, une fenêtre est balayée à travers l'image et sa version mise à l'échelle pour détecter les visages à différents emplacements et tailles. Comme montrée dans la figure 2.4



FIGURE 2.4 – Balayage des caractéristiques de type Haar dans une image

Pour détecter les visages de différentes tailles il y a deux possibilités, qui sont soit de réduire l'image par une échelle fixe en gardant la taille de la fenêtre de balayage, soit augmenter la taille de cette dernière, en gardant l'image originale telle quelle. Ces deux méthodes ont été discutées plus en détail ci-dessous :

1- Approche de l'échelle de fenêtre

La mise à l'échelle des fenêtres est un moyen simple et rapide de détecter un visage dans une image. Dans cette méthode la fenêtre de taille 24x24 glissent sur l'image entière de gauche à droite et ensuite de haut en bas. A la fin d'un balayage complet, la taille de cette fenêtre est augmentée d'un certain facteur (de préférence 1.2) et ce processus se poursuit jusqu'à ce que la taille de la fenêtre devienne égale à la taille de l'image d'entrée. Cette approche est utilisée pour détecter des visages de différentes tailles dans une image.

2- Approche de la mise à l'échelle de l'image

L'implémentation actuelle d'OpenCV (Open Computer Vision)[14] de l'algorithme de détection de visage de Viola et Jones utilise une approche de mise à l'échelle de l'image pour détecter des visages de différentes tailles dans une image. Cependant, la réduction de l'image entraîne souvent la perte de détails importants ce qui peut finalement entraîner une perte de caractéristiques et affecter le taux de détection. Ainsi, afin d'assurer une perte minimale de caractéristiques et le taux de détection, nous devons avoir une bonne qualité de scaler d'image .

Dans le titre suivant nous allons exposer les espaces de couleur les plus utilisées et les mieux notés dans la vision par ordinateur

2.3 Les espaces de représentation de la couleur :

Un espace couleur décrit une gamme spécifique, mesurable et fixe de couleurs et de valeurs de luminance possibles. Sa fonction pratique la plus élémentaire est de décrire les capacités d'un dispositif de capture ou d'affichage à reproduire les informations de couleur. De nombreuses techniques de traitement d'images utilisées en vision par ordinateur consistent à passer du RVB à un autre espace couleur on site l'espace HSL, HSI, YIQ, YUV, XYZ, TSL, L^*a^*b , I1I2I3 et les trois espaces les plus utilisé quand vas détailler par la suite : l'espace RGB ,YCbCr et HSV .

2.3.1 L'espace colorimétrique RGB :

Parce que l'œil humain ne possède que des récepteurs sensibles à la couleur pour le rouge, le vert et le bleu, il est théoriquement possible de décomposer chaque couleur visible en combinaisons de ces trois «couleurs primaires». Les moniteurs couleur, par exemple, peuvent afficher des millions de couleurs simplement en mélangeant différentes intensités de rouge, vert et bleu. Il est plus courant de placer la plage d'intensité pour chaque couleur sur une échelle de 0 à 255 (un octet)[25]. La plage d'intensité est également connue sous le nom de «profondeur de couleur».

Les possibilités de mélanger les trois couleurs primaires ensemble peuvent être représentées comme un plan de coordonnées tridimensionnelles avec les valeurs de R (rouge), G (vert) et B (bleu) sur chaque axe. Ce plan de coordonnées produit un cube appelé espace colorimétrique RVB : La figure 2.5 représente la cube colorimétrique RGB.

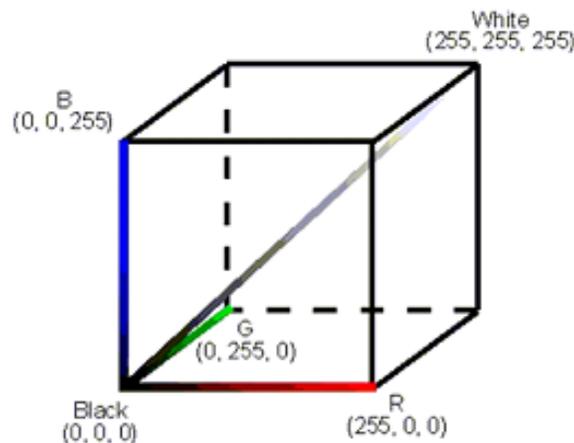


FIGURE 2.5 – Cube colorimétrique RGB

2.3.2 L'espace colorimétrique YCbCr

Est l'un des deux espaces colorimétriques primaires utilisés pour représenter la vidéo composante numérique il est défini dans les normes UIT-R BT.601-5 et UIT-R BT.709-5 de l'UIT

(Union internationale des télécommunications) comme un espace colorimétrique pour les systèmes de télévision numérique.

Ces documents donnent des définitions concrètes des coefficients de conversion entre les espaces colorimétriques RGB et YCbCr, pour la normalisation et la quantification des signaux numériques. Les composantes chromatiques individuelles de l'espace colorimétrique YCbCr sont le luma Y, le chroma Cb et le chroma Cr.

Une simple transformation linéaire permet de passer d'un système RGB au système YCbCr, Les principales transformations sont données par les équations suivantes [49] :

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2568 & 0.5041 & 0.0979 \\ -0.1482 & -0.2910 & 0.4392 \\ 0.4392 & -0.3678 & -0.0714 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (2.1)$$

2.3.3 Niveaux de gris

Dans les images en niveaux de gris, chacun des pixels d'une image ne représente qu'une quantité de lumière. En d'autres termes, chaque pixel ne contient que des informations d'intensité, qui sont décrites comme des informations de gris. Il n'existe que deux couleurs de base : le noir et le blanc, avec de nombreuses nuances de gris entre les deux [31]

En général, la forme, les bords et les caractéristiques du visage sont plus sombres, par rapport aux régions environnantes. Cette dichotomie peut être utilisée pour délimiter les différentes parties du visage et les visages du fond de l'image ou du bruit. Les informations en gris sont traitées en deux dimensions (2D), tandis que les informations en couleur sont traitées en 3D. Par conséquent, il est moins complexe sur le plan informatique (il nécessite moins de temps de traitement). Cependant, le traitement des informations grises est moins efficace et le rapport signal/bruit n'est pas à la hauteur

Après avoir lu plusieurs articles [84] [22] [61] qui abordent le sujet, il s'est avéré que l'espace de couleur le plus performant qui va nous aider dans la suite de notre projet est l'espace colorimétrique HSV que nous allons détailler ci-dessous.

2.3.4 L'espace colorimétrique HSV :

Le modèle HSB également connu sous le nom de modèle HSV (Hue, Saturation, Value) [29]. Il s'agit d'une transformation non linéaire de l'espace colorimétrique RVB. En d'autres termes, la couleur n'est pas définie comme une simple combinaison (addition / soustraction) de couleurs primaires mais comme une transformation mathématique [7].

HSL et HSV sont deux représentations liées de points dans un espace colorimétrique RVB, qui tentent de décrire les relations chromatiques perceptuelles avec plus de précision que RVB, tout en restant simples sur le plan des calculs. HSL signifie teinte, saturation, luminosité, tandis que HSV signifie teinte, saturation, valeur. HSI et HSB sont des noms alternatifs pour ces concepts,

en utilisant l'intensité et la luminosité ; leurs définitions sont moins standardisées, mais elles sont généralement interprétées comme synonymes de HSL. HSL et HSV décrivent les couleurs comme des points dans un cylindre dont l'axe central va du noir en bas au blanc en haut avec des couleurs neutres entre elles, où l'angle autour de l'axe correspond à la «teinte», la distance de l'axe correspond à la «saturation », et la distance le long de l'axe correspond à« légèreté », « valeur »ou« luminosité ».

La figure 2.6 représente le cylindre des couleurs HSV.

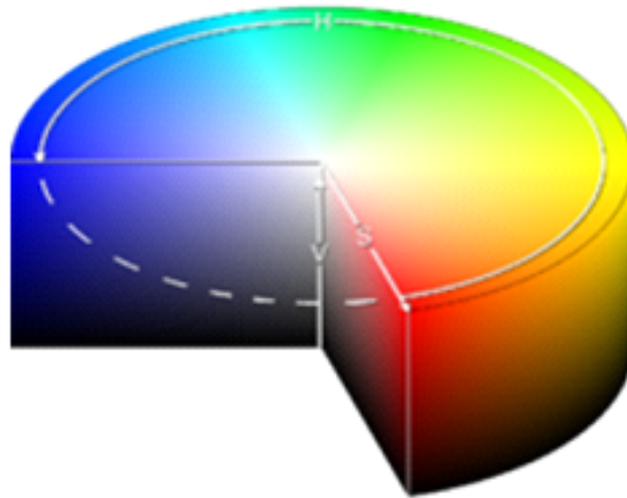


FIGURE 2.6 – Cylindre colorimétrique HSV

Dans la section suivante nous expliquons les types de méthode d'extraction des caractéristiques les plus pertinentes et les plus utilisés.

2.4 Extraction de caractéristiques

L'extraction des caractéristiques est connue pour être l'une des opérations les plus cruciales dans les systèmes de vérification des visages. L'étape d'extraction de caractéristiques nous donne un vecteur de caractéristiques ou une matrice de caractéristiques pour l'image, qui est considéré comme la signature biométrique de cet élément. Il existe deux types de techniques qui peuvent être utilisées pour construire la représentation du visage : les méthodes basées sur les caractéristiques artisanales (handcraft features) et les méthodes basées sur les caractéristiques profondes (Deep features). PCA, LDA, BSIF, LPQ et LBP, par exemple, sont des techniques artisanales d'extraction de caractéristiques où les méthodes spécifiques ont été conçues pour extraire

les caractéristiques des images de visage tout en maintenant des variations au sein de la classe aussi faibles que possible. Pour les descripteurs artisanaux, nous n'avons pas besoin d'entraîner la caractéristique avec un grand ensemble de données. Contrairement au méthode cité auparavant VGG-Face est une technique d'extraction de caractéristiques profond. En outre, un réseau neuronal convolutif hiérarchique profond et un grand ensemble de données d'images d'entraînement doivent être construits afin d'obtenir meilleure représentation de l'image. Dans ce qui suit nous allons définir les méthodes d'extraction de caractéristique les plus utilisés.

2.4.1 les méthode basé sur les caractéristiques artisanales (handcraft features)

2.4.1.1 Motifs binaires locaux (LBP)

Le LBP a été introduit pour la première fois par Ojala et al. dans [60]. Il s'agit d'une méthode puissante de description de texture basée sur l'analyse statistique et montre son utilisation pratique dans la description des textures. Bien que de nombreuses variantes de LBP soient largement utilisées pour l'analyse des visages en raison de leurs performances de classification satisfaisantes, elles n'ont pas encore prouvée compactes. La fusion de caractéristiques est donc considérée comme une approche efficace. LBP utilisant la maximisation de l'information mutuelle montre une meilleure performance dans l'analyse du visage [13]. L'opérateur étiquette les pixels d'une image en seillant le voisinage 3x3 de chaque pixel avec le pixel central et en considérant le résultat comme un nombre binaire. Le résultat LBP peut être exprimé comme suit :

$$LBP(x_c, y_c) = \sum_{n=0}^p s(i_n - i_c)2^n \quad (2.2)$$

où i_c est une valeur de pixel central positionnée à (x_c, y_c) , i_n est l'une des huit valeurs de pixels centraux au environs du rayon R , P est le nombre total de voisins, et une fonction de signe est définie de sorte que :

$$s(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{sinon} \end{cases} \quad (2.3)$$

L'histogramme des étiquettes LBP peut être utilisé comme descripteur de texture. descripteur de texture. Voir la figure 2.7 , pour une illustration de l'opérateur LBP de base. Le LBP original avec P voisinages a p^2 motifs binaires différents. Cependant, Ojala et al. [60] ont observé que les images naturelles contiennent généralement un petit nombre de codes LBP, qui sont appelés LBP uniformes. Le LBP uniforme contient un maximum de deux transitions bit à bit de 0 à 1 ou vice-versa lorsque le motif binaire est considéré comme circulaire. Ces modèles uniformes représentent la majorité des microstructures de texture.

La figure 2.7 représente les étapes de l'algorithme LBP

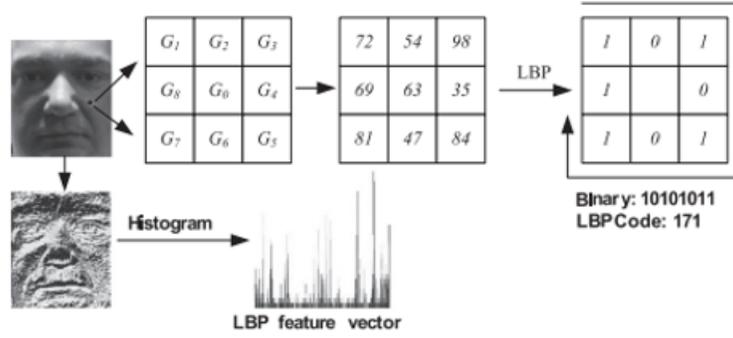


FIGURE 2.7 – Algorithme de l’opérateur LBP

2.4.1.2 Quantification de la phase locale (LPQ)

La quantification de la phase locale (Local Phase Quantization- LPQ) a été introduite pour la première fois par Ojansivu et al [79]. Le descripteur de quantification de phase locale est basé sur la quantification de la phase de la transformée de Fourier dans les voisinages locaux. La fréquence locale peut être calculée en utilisant une transformée de Fourier à court terme sur des voisinages locaux $M \times M N_x$ à chaque pixel x de l’image défini par

$$F(u, x) = \sum_{y \in N_x} f(x - y) e^{-j2\pi u^T y} \quad (2.4)$$

La transformée est évaluée efficacement pour toutes les positions de l’image $x \in x_1, x_2, \dots, x_N$ en utilisant simplement des convolutions 1-D pour les lignes et les colonnes successivement.

En LPQ, seuls quatre coefficients complexes sont considérés, correspondant à des fréquences 2-D Fréquences $u_1 = [a, 0]^T$, $u_2 = [0, a]^T$, $u_3 = [a, a]^T$, et $u_4 = [a, a]^T$, où a est un scalaire suffisamment petit pour satisfaire $H(u_i) > 0$. Soit

$$F_x^c = [F(u_1, x), F(u_2, x), F(u_3, x), F(u_4, x)], \text{ and} \quad (2.5)$$

$$F_x = [Re[F_x^c], Im[F_x^c]]^T \quad (2.6)$$

où Re - et Im - renvoient les parties réelles et imaginaires d’un nombre complexe, respectivement. La matrice de transformation correspondante de 8 par M2 correspondante est

$$W = [Re[w_{u1}, w_{u2}, w_{u3}, w_{u4}], Im[w_{u1}, w_{u2}, w_{u3}, w_{u4}]]^T \quad (2.7)$$

de sorte que

$$F_x = W f_x \quad (2.8)$$

L’information de phase dans les coefficients de Fourier est enregistrée en observant les signes des parties réelles et imaginaires de chaque composante de F_x . Cette opération est réalisée en utilisant

un simple quantificateur scalaire

$$q_j = \begin{cases} 1, & g_j \geq 0 \\ 0, & g_j < 0 \end{cases} \quad (2.9)$$

où est la jème composante du vecteur $G(x) = [\text{Re}F(x), \text{Im}F(x)]$.

Les huit coefficients binaires $q_j(x)$ qui en résultent sont représentés sous la forme de valeurs entières entre 0 et 255 en utilisant le codage binaire

$$f_{LPQ}(x) = \sum_{j=1}^8 q_j 2^{j-1} \quad (2.10)$$

Par conséquent, nous obtenons l'image d'étiquettes fLP Q dont les valeurs sont les étiquettes LPQ invariantes au flou. La figure 2.8 montre un exemple de calcul LPQ.

La figure 2.8 représente les étapes de l'algorithme LPQ.

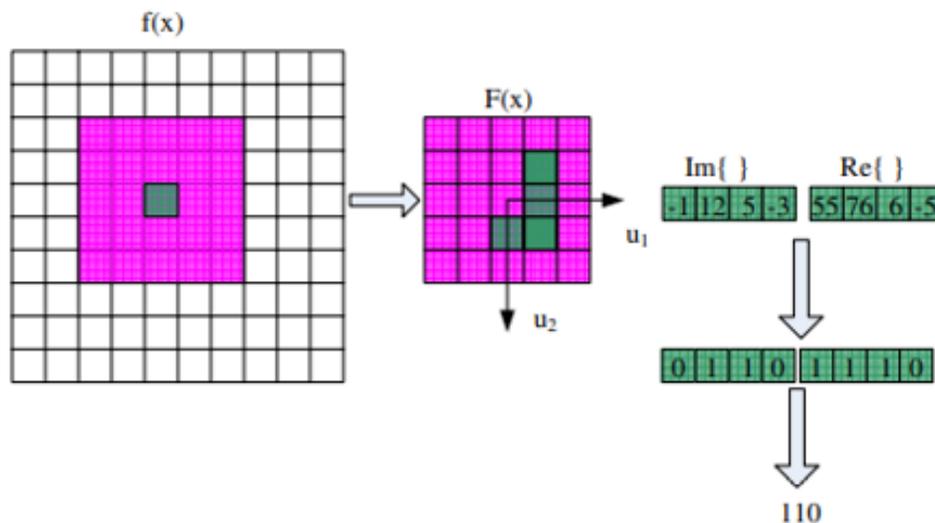


FIGURE 2.8 – Algorithme de l'opérateur LPQ

2.4.1.3 Fonctionnalités d'image statistique binaires (BSIF)

Contrairement à la LBP et la LPQ qui peuvent être utilisées pour calculer les statistiques d'étiquettes dans les voisinages des pixels locaux, le descripteur local appelé BSIF (Binarized Statistical Image Features), qui a été récemment proposé par Kannlaand et Rahtu [39], utilise un ensemble prédéfini manuellement des filtres linéaires et binarisation des réponses du filtre [2] statistiques des taches. La valeur de chaque élément (bit) dans notre chaîne de code binaire est calculée en binarisant la réponse d'un filtre linéaire avec un seuil à zéro. Chaque bit est associé à un filtre différent et la longueur souhaitée de la chaîne de bits détermine le nombre de filtres utilisés. L'ensemble des filtres est appris à partir d'un ensemble d'apprentissage de patches d'images naturelles en maximisant l'indépendance statistique des réponses aux filtres [2]. Ainsi,

les propriétés statistiques des taches d'images naturelles déterminent les descripteurs. Nous les appelons donc caractéristiques statistiques binarisées d'image (BSIF).

Étant donné un patche d'image X de taille $| \times |$ pixels et un filtre linéaire W_i de la même taille, la réponse du filtre s_i est obtenue par :

$$s_i = \sum_{u,v} W_i(u,v)X(u,v) = w_i^T x \quad (2.11)$$

où la notation vectorielle est introduite dans la dernière étape, c'est-à-dire que les vecteurs w et x contiennent les pixels de W_i et X . La caractéristique binarisée b_i est obtenue par :

$$b_i = \begin{cases} 1 & \text{si } s_i > 0 \\ 0 & \text{sinon} \end{cases} \quad (2.12)$$

Où b_i est l'élément de b . De cette manière, on peut calculer un code binaire de n bits série b pour chaque pixel et par la suite la région de l'image peut être représentée par des histogrammes de code binaire du pixel.

La figure 2.9 représente les étapes de l'algorithme BSIF

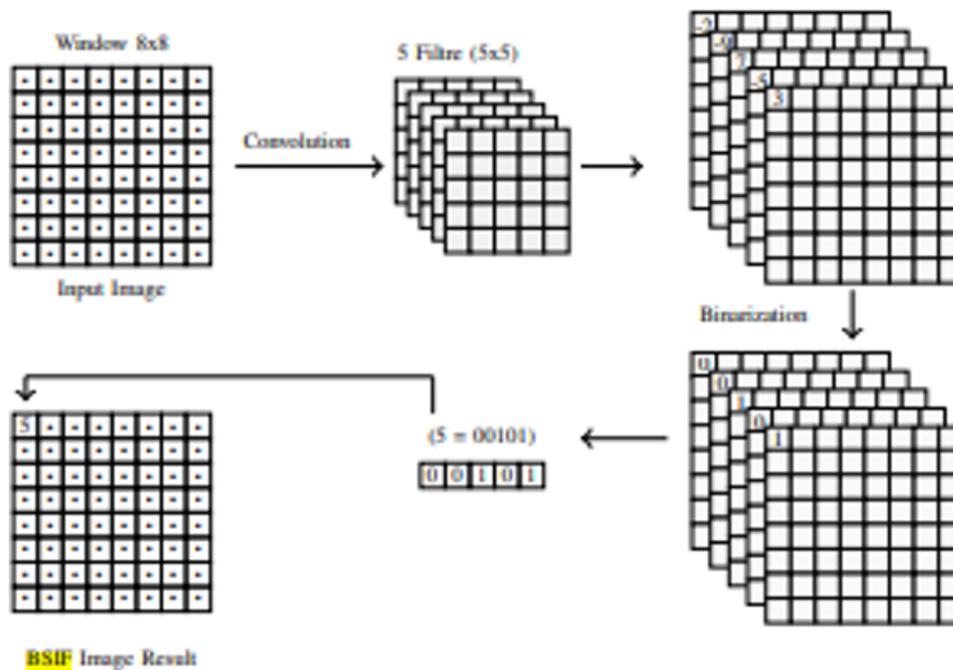


FIGURE 2.9 – Algorithme de l'opérateur BSIF

2.4.2 les méthode basé sur les caractéristiques profond(Deep features)

2.4.2.1 Extraction profonde de caractéristiques avec VGG16 et 19

Cette méthode se base initialement sur une architecture de réseaux convolutifs profond (Conv-Net) proposée pour la première fois par K. Simonyan et A. Zisserman[41] du Visual Geometry

Group de l'université d'Oxford en 2014 . L'architecture VGG est à la base de modèles de reconnaissance d'objets révolutionnaires. Développé en tant que réseau neuronal profond, le VGGNet surpasse également les bases de référence dans de nombreuses tâches et ensembles de données au-delà du réseau. En outre, il reste aujourd'hui l'une des architectures de reconnaissance d'images les plus populaires.

La figure 2.10 représente l'architecture du VGG

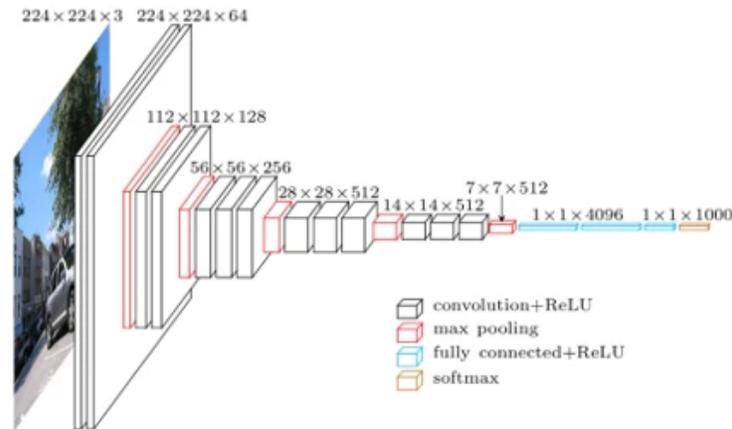


FIGURE 2.10 – Architecture du réseau neuronal VGG

VGG16 :

Le modèle VGG, ou VGGNet, qui prend en charge 16 couches est également appelé VGG16. Il s'agit d'un modèle de réseau neuronal convolutif proposé par A. Zisserman et K. Simonyan de l'université d'Oxford.

Le modèle VGG16 atteint une précision de près de 92,7 % dans le top 5 du test ImageNet. ImageNet est un jeu de données composé de plus de 14 millions d'images appartenant à près de 1000 classes. En outre, il s'agit de l'un des modèles les plus populaires soumis à l'ILSVRC-2014. Il remplace les grands filtres de la taille d'un noyau par plusieurs filtres de la taille d'un noyau 3×3 l'un après l'autre, apportant ainsi des améliorations significatives par rapport à AlexNet. Comme indiqué plus haut, le modèle VGGNet-16 prend en charge 16 couches et peut classer les images en 1000 catégories d'objets, dont le clavier, les animaux, le crayon, la souris, etc. De plus, le modèle a une taille d'entrée d'image de 224 par 224.

VGG19 :

Le concept du modèle VGG19 (également VGGNet-19) est le même que celui du VGG16, sauf qu'il prend en charge 19 couches. Les chiffres "16" et "19" représentent le nombre de couches de poids dans le modèle (couches convolutives). Cela signifie que VGG19 a trois couches convolutionnelles de plus que VGG16.

Le réseau VGG est construit avec de très petits filtres convolutifs. Le VGG-16 se compose de 13 couches convolutionnelles et de trois couches entièrement connectées.

Architecture de VGG :

Entrée : Le VGGNet prend en entrée une image de 224×224 . Pour la compétition ImageNet, les créateurs du modèle ont recadré le patch central 224×224 de chaque image afin de conserver une taille d'entrée cohérente. Couches convolutives : Les couches convolutives du VGG exploitent un champ réceptif minimal, c'est-à-dire 3×3 , la plus petite taille possible qui permet de capturer le haut/bas et la gauche/droite. De plus, il existe également des filtres de convolution 1×1 qui agissent comme une transformation linéaire de l'entrée. Ils sont suivis d'une unité ReLU, une innovation majeure d'AlexNet qui réduit le temps de formation. ReLU est l'abréviation de "rectified linear unit activation function" (fonction d'activation de l'unité linéaire rectifiée) ; il s'agit d'une fonction linéaire par morceaux qui produit l'entrée si elle est positive ; sinon, la sortie est nulle. Le pas de convolution est fixé à 1 pixel afin de préserver la résolution spatiale après convolution (le pas est le nombre de décalages de pixels sur la matrice d'entrée). Couches cachées : Toutes les couches cachées du réseau VGG utilisent ReLU. Le réseau VGG n'utilise généralement pas la normalisation locale de la réponse (LRN), car elle augmente la consommation de mémoire et le temps d'apprentissage. De plus, elle n'améliore pas la précision globale. Couches entièrement connectées : Le VGGNet comporte trois couches entièrement connectées. Sur les trois couches, les deux premières ont 4096 canaux chacune, et la troisième a 1000 canaux, 1 pour chaque classe.

L'extraction de caractéristiques est une étape très importante dans notre approche. Pour cela nous avons choisi de travailler avec les méthodes les plus utilisées et les plus avantageuses des deux types qui sont LPQ, BSIF et VGG19.

L'application d'une méthode d'extraction de caractéristique sur une image nous donne une matrice avec une dimension importante ce qui peut rendre le processus de vérification plus long pour cela une réduction de caractéristiques s'impose.

2.5 Réduction de caractéristique

La réduction des caractéristiques, également appelée réduction de la dimensionnalité, est le processus qui consiste à réduire le nombre de caractéristiques dans un calcul lourd en ressources sans perdre d'informations importantes. Réduire le nombre de caractéristiques signifie que le nombre de variables est réduit, ce qui facilite et accélère le travail de l'ordinateur. La réduction des caractéristiques peut être divisée en deux processus : la sélection des caractéristiques et l'extraction des caractéristiques. Il existe plusieurs méthodes de réduction de caractéristique citons Factorisation matricielle non négative (NMF), Filtre à haute corrélation, Filtre à faible variance, Rapport des valeurs manquantes et les trois les plus utilisées : Analyse Discriminante Linéaire (LDA), Décomposition en valeur singulière (SVD) et Principal Component Analysis (ACP) qui seront détaillés ci-dessous :

2.5.1 L'analyse discriminante linéaire (LDA)

L'analyse discriminante linéaire (LDA) est une méthode bien connue pour la réduction de dimensionnalité et de classification qui projette des données à haute dimension sur un espace à basse dimension dans lequel où les données atteignent une séparabilité maximale des classes [[70], [42], [77]]. Les caractéristiques dérivées dans LDA sont des combinaisons linéaires des caractéristiques originales, où les coefficients proviennent de la matrice de transformation [34]. L'analyse discriminante linéaire est une technique de classification de caractéristiques binaires et non binaires utilisant un algorithme linéaire pour apprendre la relation entre les caractéristiques dépendantes et indépendantes. Elle utilise la formule de Fischer pour réduire la dimensionnalité des données de manière à les faire entrer dans une dimension linéaire. LDA est un algorithme multifonctionnel, c'est un classificateur, un réducteur de dimensionnalité et un visualisateur de données. L'objectif du LDA est le suivant :

- Minimiser la variabilité interclasse, qui consiste à classer autant de points similaires que possible dans une classe. Cela permet de réduire le nombre de mauvaises classifications.
- Maximiser la distance entre la moyenne des classes, la moyenne est placée aussi loin que possible pour assurer une confiance élevée pendant la prédiction.

La dispersion entre classes S_B et la dispersion à l'intérieur de la classe S_W sont définies comme suit

$$S_b = \sum_{i=1}^c N_i (m_i - m)(m_i - m)^T \quad (2.13)$$

$$S_w = \sum_{i=1}^c \sum_{j=i}^{N_i} (x_{ij} - m_i)(x_{ij} - m_i)^T \quad (2.14)$$

où x_{ij} désigne le j ème échantillon d'apprentissage de la i ème classe, m_i est la moyenne du échantillon d'apprentissage de la i ème classe et m est la moyenne de tous les échantillons d'apprentissage. La fonction objective de LDA est définie comme suit

$$\max_w \frac{w^T S_B w}{w^T S_W w} \quad (2.15)$$

Dans La figure 2.11 Chaque couleur représente une classe. m_1 et m_2 sont les moyennes de la classe1 et de la classe2 respectivement. Le graphique de gauche montre les projections des données sur la ligne joignant m_1 et m_2 . Il y a beaucoup de chevauchements entre les échantillons des deux classes. Le graphique de droite montre la projection des données en utilisant LDA, ce qui minimise les chevauchements entre les deux classes.

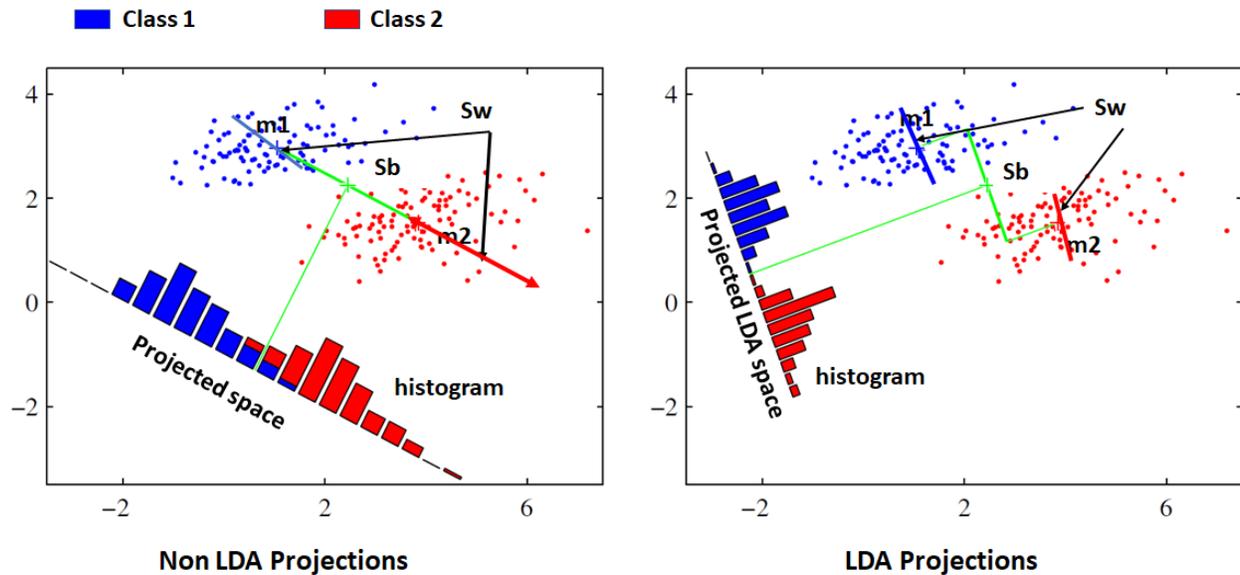


FIGURE 2.11 – Projections LDA et non LDA[12]

2.5.2 Analyse en Composantes Principales (ACP)

L'objectif de l'ACP ((PCA) Principal Component Analysis) est d'effectuer une réduction de la dimensionnalité. L'ACP trouve une transformation qui réduit la dimensionnalité des données tout en tenant compte d'autant de variance que possible. L'ACP est la plus ancienne technique d'analyse multi variée. Le concept fondamental de l'ACP est le mécanisme basé sur la projection. Une Analyse en Composantes Principales (ACP) permet de définir, à partir d'un jeu de données d'apprentissage, un sous espace permettant de simultanément conserver l'information discriminante et supprimer les informations secondaires (non informatives). Cette méthode consiste à trouver une nouvelle base de l'espace des données dont tous les vecteurs sont orthogonaux entre eux. Le premier de ces vecteurs correspond à la direction de variance maximale des données d'apprentissage. Les autres composantes sont déterminées par la contrainte d'orthogonalité entre les vecteurs tout en respectant une direction de variance maximum. Ainsi au lieu de stocker une vectrice image de taille Résolution N , on stocke un vecteur de taille $R \times M$ nombre d'image d'apprentissage. Dans l'approche PCA la normalisation d'éclairage est toujours indispensable. La PCA est une technique rapide, simple et populaire dans l'identification de modèle, c'est l'une des meilleures techniques. Les projections de la PCA sont optimales pour la reconstruction d'une base de dimension réduite [29]. L'ACP peut nous aider à améliorer les performances à un coût très faible pour la précision du modèle. Parmi les autres avantages de l'ACP, citons la réduction du bruit dans les données, la sélection des caractéristiques (dans une certaine mesure) et la possibilité de produire des caractéristiques indépendantes et non corrélées des données. C'est ces avantage qui nous a persuadé de choisir e PCA comme méthode de réduction pour notre projet $R^m \rightarrow S_p \rightarrow S_r$ ($S_p = l$ $dim(S_r) = (m - l)$). Le modèle ACP partitionne l'espace m des mesures en un sous-espace des composantes principales, S_p , où les variations normales auront lieu et un sous-espace des résidus, S_r , où les défauts doivent

apparaître. Notons que $\dim(S_p) = \ell$ et $\dim(S_r) = (m - \ell)$. Donc, le vecteur des mesures peut être décomposé en deux parties : La figure 2.12 montre une interprétation de ACP

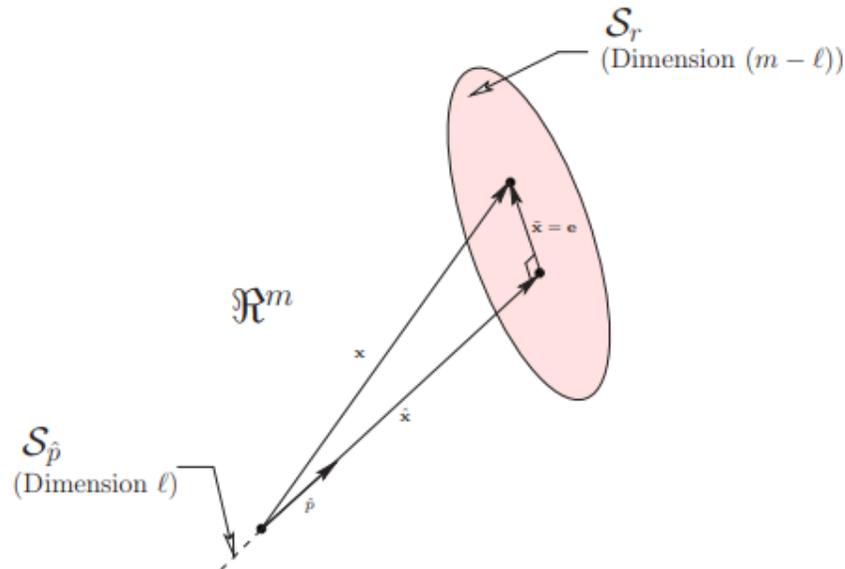


FIGURE 2.12 – Interprétation géométrique de l’analyse en composantes principales[38]

Parmi les meilleures méthodes utilisées dans les systèmes de vérification de parenté est l’apprentissage en utilisant un réseau de neurone profond.

2.6 Apprentissage profond (Deep Learning)

L’apprentissage profond est un sous-ensemble de l’apprentissage automatique, qui consiste essentiellement en un réseau neuronal à trois couches ou plus. Ces réseaux neuronaux tentent de simuler le comportement du cerveau humain, même s’ils sont loin d’en avoir la capacité, ce qui leur permet ”d’apprendre” à partir de grandes quantités de données. Si un réseau neuronal à une seule couche peut toujours faire des prédictions approximatives, des couches cachées supplémentaires peuvent aider à optimiser et à affiner la précision. Le Deep Learning, qui s’est imposé comme un outil efficace pour l’analyse des big data, utilise des algorithmes complexes et des réseaux neuronaux artificiels pour former des machines/ordinateurs afin qu’ils puissent apprendre par expérience, classer et reconnaître des données/images, tout comme le fait un cerveau humain. Dans le cadre du Deep Learning, un réseau neuronal convolutif ou CNN est un type de réseau neuronal artificiel, largement utilisé pour la reconnaissance et la classification des images/objets. Les CNN jouent un rôle majeur dans diverses tâches/fonctions telles que les problèmes de traitement d’images, les tâches de vision par ordinateur comme la localisation et la segmentation, l’analyse vidéo, la reconnaissance des obstacles dans les voitures à conduite autonome, ainsi que la reconnaissance vocale dans le traitement du langage naturel. Comme les CNN jouent un rôle important dans ces domaines émergents et à croissance rapide, ils sont très populaires dans l’apprentissage profond.

2.6.1 réseau de neurones convolutif CNN :

Un réseau de neurones convolutif [46] (ConvNet/CNN) est un algorithme d'apprentissage profond qui peut prendre une image en entrée, attribuer de l'importance (poids et biais apprenables) à divers aspects/objets de l'image et être capable de les différencier les uns des autres. Le prétraitement requis dans un ConvNet est beaucoup plus faible que dans d'autres algorithmes de classification. Alors que dans les méthodes primitives, les filtres sont conçus à la main, avec un entraînement suffisant, les ConvNet ont la capacité d'apprendre ces filtres/caractéristiques.

L'architecture d'un ConvNet est analogue à celle du modèle de connectivité des neurones dans le cerveau humain et s'inspire de l'organisation du cortex visuel. Les neurones individuels ne répondent aux stimuli que dans une région restreinte du champ visuel, appelée champ de réception. Un ensemble de ces champs se chevauchent pour couvrir l'ensemble de la zone visuelle. Le réseau neuronal convolutif CNN fonctionne en obtenant une image, en lui attribuant un certain poids en fonction des différents objets de l'image, puis en les distinguant les uns des autres. Le CNN nécessite très peu de données de prétraitement par rapport aux autres algorithmes d'apprentissage profond. L'une des principales capacités du CNN est qu'il applique des méthodes primitives pour former ses classificateurs, ce qui le rend suffisamment performant pour apprendre les caractéristiques de l'objet cible. Il existe trois types de couches qui composent le CNN, à savoir les couches de convolution, les couches de mise en commun et les couches entièrement connectées (FC). Lorsque ces couches sont empilées, une architecture CNN est formée.

2.6.1.1 Couche convolutionnelle

Cette couche est la première couche qui est utilisée pour extraire les différentes caractéristiques des images d'entrée. Dans cette couche, l'opération mathématique de convolution est effectuée entre l'image d'entrée et un filtre d'une taille particulière $M \times M$. En faisant glisser le filtre sur l'image d'entrée, le produit scalaire est pris entre le filtre et les parties de l'image d'entrée par rapport à la taille du filtre ($M \times M$).

La sortie est appelée carte de caractéristiques qui nous donne des informations sur l'image telles que les coins et les bords. Plus tard, cette carte de caractéristiques est introduite dans d'autres couches pour apprendre plusieurs autres caractéristiques de l'image d'entrée.

2.6.1.2 Couche de mise en commun

Dans la plupart des cas, une couche de convolution est suivie d'une couche de mise en commun (pooling). L'objectif principal de cette couche est de diminuer la taille de la carte de caractéristiques convolutionnelle afin de réduire les coûts de calcul. Ceci est réalisé en diminuant les connexions entre les couches et en opérant indépendamment sur chaque carte de caractéristiques. Selon la méthode utilisée, il existe plusieurs types d'opérations de mise en commun.

Dans la mise en commun maximale, l'élément le plus grand est extrait de la carte de caractéristiques. La mise en commun moyenne calcule la moyenne des éléments dans une section

d'image de taille prédéfinie. La somme totale des éléments dans la section prédéfinie est calculée dans Sum Pooling. La couche de mise en commun sert généralement de pont entre la couche de convolution et la couche FC.

La figure 2.13 represent extraire des informations de l'image avec Les Couche convolutionnelle et couche de mise en commun.

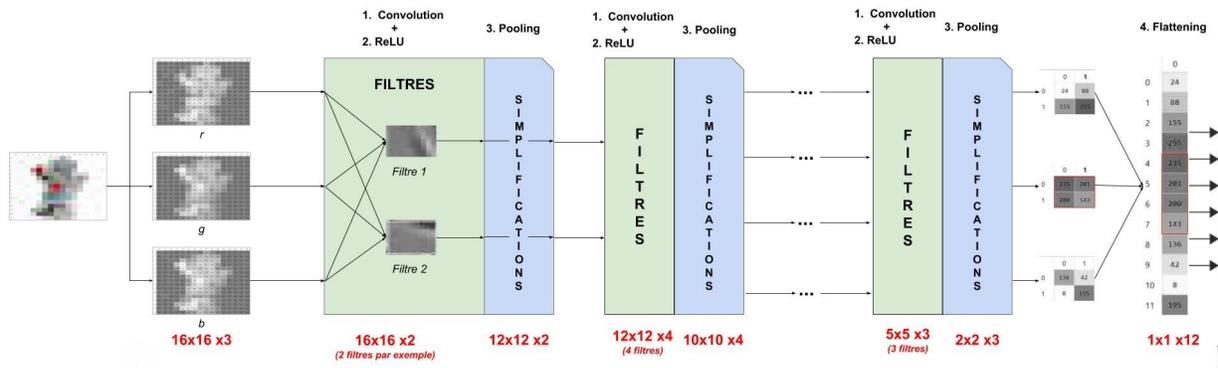


FIGURE 2.13 – Les Couche convolutionnelle et couche de mise en commun

2.6.1.3 Couche entièrement connectée

La couche entièrement connectée (FC) se compose des poids et des biais avec les neurones et est utilisée pour connecter les neurones entre deux couches différentes. Ces couches sont généralement placées avant la couche de sortie et forment les dernières couches d'une architecture CNN.

Dans cette couche, l'image d'entrée des couches précédentes est aplatie et transmise à la couche FC. Le vecteur aplati passe ensuite par quelques couches FC supplémentaires où les opérations de fonctions mathématiques ont généralement lieu. À ce stade, le processus de classification commence à avoir lieu.

La figure 2.14 represent couche entièrement connectée.

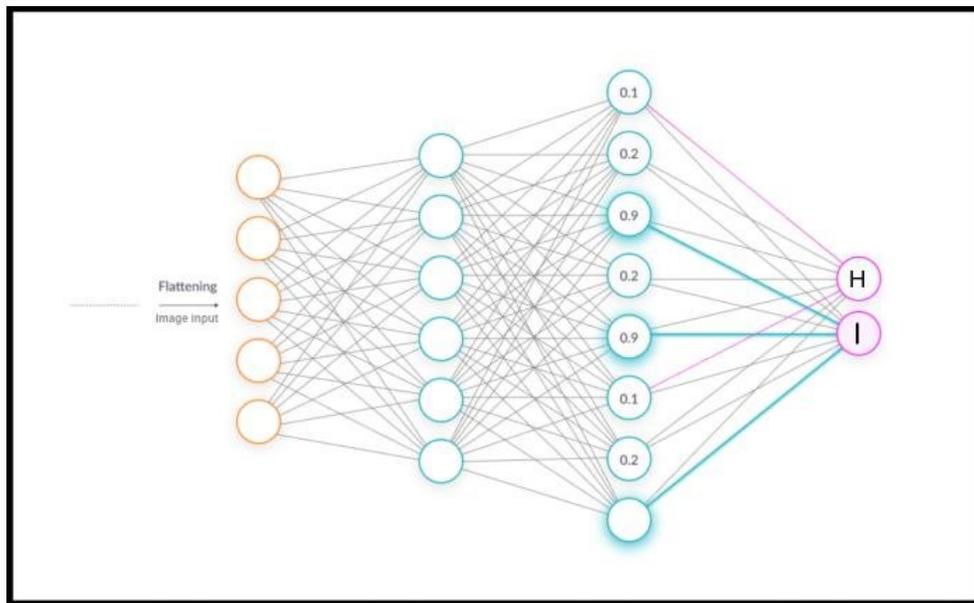


FIGURE 2.14 – Couche entièrement connectée

2.6.2 Keras

Keras est une API d'apprentissage profond de haut niveau développée par Google pour la mise en œuvre de réseaux neuronaux. Elle est écrite en Python et est utilisée pour faciliter la mise en œuvre de réseaux neuronaux. Elle prend également en charge le calcul de plusieurs réseaux neuronaux en arrière-plan.

Keras est relativement facile à apprendre et à utiliser parce qu'il fournit un frontal Python avec un haut niveau d'abstraction tout en ayant la possibilité d'utiliser plusieurs backends pour le calcul. Cela rend Keras plus lent que d'autres cadres d'apprentissage profond, mais extrêmement convivial pour les débutants. Keras dispose de deux principaux types de modèles : Les modèles séquentiels et fonctionnels. Vous choisissez le type de modèle que vous voulez, puis vous définissez le flux de données entre eux.

2.7 Algorithme de classification

L'algorithme de classification est une technique d'apprentissage supervisé qui est utilisée pour identifier la catégorie de nouvelles observations sur la base de données d'apprentissage. Dans la classification, un programme apprend à partir d'un ensemble de données, puis classe les nouvelles observations dans un certain nombre de classes. Par exemple, Oui ou Non, 0 ou 1, Spam ou Non Spam, chat ou chien, etc. Les classes peuvent être appelées cibles/étiquettes ou catégories.

Contrairement à la régression, la variable de sortie de la classification est une catégorie, et non une valeur, comme "vert ou bleu", "fruit ou animal", etc. L'algorithme de classification étant une technique d'apprentissage supervisé, il prend des données d'entrée étiquetées, ce qui signifie qu'il contient des entrées avec la sortie correspondante.

Dans l'algorithme de classification, une fonction de sortie discrète (y) est mise en correspondance avec une variable d'entrée (x). L'objectif principal de l'algorithme de classification est d'identifier la catégorie d'un ensemble de données donné, et ces algorithmes sont principalement utilisés pour prédire la sortie des données catégoriques. L'algorithme qui met en œuvre la classification sur un ensemble de données est connu comme un classificateur. Dans ce qui suit nous citons quelques algorithmes de classification les plus utilisées .

2.7.1 K-plus proches voisins

Les classificateurs K-plus proches voisins [16] sont basés sur l'apprentissage par analogie. Les échantillons d'apprentissage sont décrits par des attributs numériques à n dimensions ou chaque échantillon représente un point. De cette façon, tous les échantillons d'apprentissage sont stockés dans un espace de modèle à n dimensions. Lorsque on lui attribue un échantillon inconnu, un classificateur à k plus proches voisins recherche dans l'espace des échantillons d'apprentissage qui sont les plus proches . (classif aussi) La "proximité" est définie en termes de Distance euclidienne, où la distance euclidienne entre deux points est calculée comme suit :

$$X = (x_1, x_2, \dots, x_n) \text{ and } Y = (y_1, y_2, \dots, y_n) \text{ is } d(X, Y) = \sqrt{\sum_{t=1}^n (x_t - y_t)^2} \quad (2.16)$$

La figure 2.15 représente un exemple de classification par K plus proches voisins

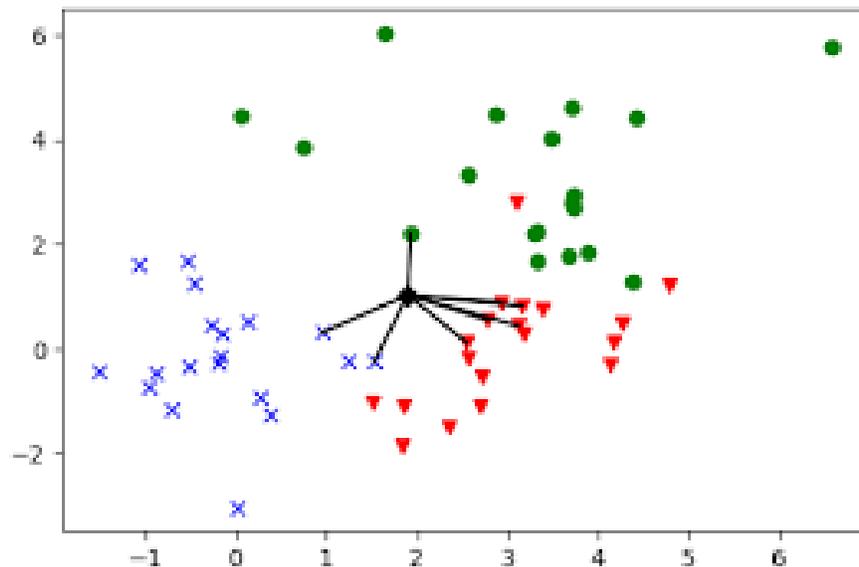


FIGURE 2.15 – Classification par K plus proches voisins

2.8 Arbre de décision

Un arbre de décision de décision est un classificateur exprimé comme une partition récursive de l'espace des instances. L'arbre de décision [71] est constitué de nœuds qui forment un arbre à racines, ce qui signifie qu'il s'agit d'un arbre dirigé avec un nœud appelé "racine" qui n'a pas d'arêtes entrantes. Tous les autres nœuds ont exactement une arête entrante. Un nœud avec des bords sortants est appelé nœud interne ou nœud test. Tous les autres nœuds sont appelés feuilles (également appelés nœuds terminaux ou de décision). Dans un arbre de décision, chaque nœud interne divise l'espace d'instance en deux sous-espaces ou plus selon une fonction discrète des attributs d'entrée. La racine et les nœuds internes sont associés aux attributs, les nœuds feuilles sont associés aux classes. Fondamentalement, chaque nœud non-feuille possède une branche sortante pour chaque valeur possible de l'attribut associé au nœud. Pour déterminer la classe pour une nouvelle instance en utilisant un arbre de décision, en commençant par la racine, les nœuds internes successifs sont visités jusqu'à ce qu'un nœud feuille soit atteint. Au nœud de la racine et à chaque nœud interne, un test est appliqué. Le résultat du test détermine la branche traversée, et le prochain nœud visité. La classe de l'instance est la classe du dernier nœud de la feuille. Le critère d'estimation [30] dans l'algorithme de l'arbre de décision est la sélection d'un attribut à tester à chaque nœud de décision de l'arbre. L'objectif est de sélectionner l'attribut le plus utile pour classer les exemples. Une bonne mesure quantitative de la valeur d'un d'un attribut est une propriété statistique appelée gain d'information qui mesure la capacité d'un attribut donné sépare les exemples d'apprentissage en fonction de leur classification cible. Cette mesure est utilisée pour sélectionner parmi les attributs candidats à chaque étape pendant la croissance de l'arbre[74].

La figure 2.16 d'un arbre de décision qui s'agit de prédire "oui" ou "non" en réponse à la question "Le temps permet-il de jouer à l'extérieur?"

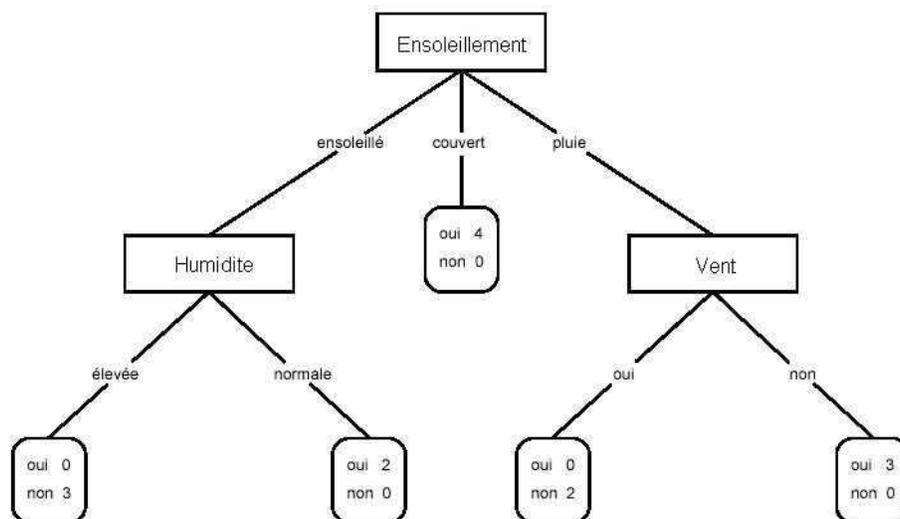


FIGURE 2.16 – Exemple d'un arbre de décision construit à partir de l'exemple de Quinlan [31] ”

2.9 Machines à vecteur de support (SVM)

Le SVM [17] est une méthode très efficace pour la régression, la classification et la reconnaissance générale des formes. Il est considéré comme un bon classificateur en raison de sa haute performance de généralisation sans avoir besoin d'ajouter a priori, même lorsque la dimension de l'espace d'entrée est très élevée. L'objectif des SVM est de trouver la meilleure fonction de classification pour distinguer les membres des deux classes dans les données d'apprentissage. La métrique pour le concept de la "meilleure" fonction de classification peut être réalisée de manière géométrique. Pour des données linéairement séparables, une fonction de classification linéaire correspond à un hyperplan de séparation $f(x)$ qui passe par le milieu des deux classes, et séparant les deux. Une fois cette fonction déterminée, une nouvelle instance de données $f(x_n)$ peut être classée en testant simplement le signe de la fonction $f(x_n)$; x_n appartient à la classe positive si $f(x_n) > 0$. Parce qu'il existe de tels hyperplans linéaires, les SVM garantissent que la meilleure fonction de ce type soit trouvée en maximisant la marge entre les deux classes. Intuitivement, la marge est définie comme la quantité d'espace, ou la séparation entre les deux classes, comme définies par l'hyperplan. Géométriquement, la marge correspond à la distance la plus courte entre les points de données les plus proches d'un point de l'hyperplan.

la figure 2.17 représenté un exemple de problème séparable dans un espace à 2 dimensions.

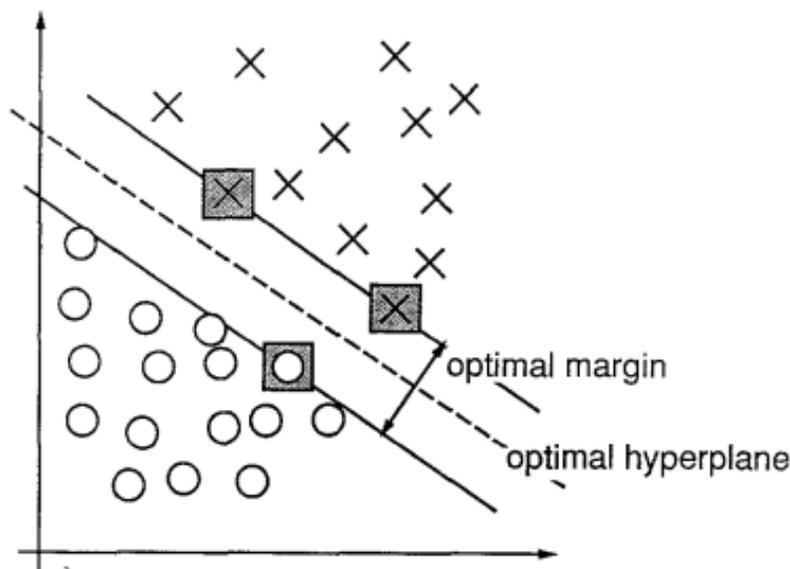


FIGURE 2.17 – Les vecteurs de support, marqués par des carrés gris[17]

2.10 conclusion

Dans ce chapitre nous avons présenté quelques notions et définitions de base sur les différentes méthodes utilisées dans la vérification de parenté afin dans le but d'obtenir de meilleurs résultats. Comme première partie nous avons vu la conversion d'espaces de couleur suivie par les méthodes de détection de visage et d'extraction des caractéristiques les plus utilisées et on a clôturé le chapitre en définissant les méthodes de réduction de caractéristique les algorithmes d'apprentissage et de classification les plus pertinentes.

Système de vérification de parenté

3.1 Introduction

La vérification de la parenté à partir d'images faciales est un problème émergent dans le domaine de la vision par ordinateur. Du point de vue de la reconnaissance des visages, la parenté nous offre une opportunité précieuse et opérationnelle de construire des relations utiles entre les personnes sur la base de leurs signaux visuels, approfondissant ainsi notre compréhension sur leur sémantique. Cependant, le problème de la vérification de la parenté présente de nombreuses difficultés, telles que les variations d'âge, de sexe, d'origine ethnique, les ressemblances entre des personnes sans lien de parenté réel et les différences d'attributs faciaux entre des personnes de la même famille. Afin d'arriver à résoudre ce défi de nombreux chercheurs en vision par ordinateur se sont penchés sur le problème de la vérification de la parenté. De nombreuses approches ont été proposées sur ont été proposées qui ont été citée auparavant. L'objectif de ce chapitre, est de déployer un système de vérification de parenté capable de fournir de meilleurs résultats La figure 3.1 représente un schéma général de l'approche. Présente une description globale De notre système.

3.2 Structure du système proposé

Notre système comporte cinq etape en general qui sont : la detection de visage, la conversion d'espace de couleurs, l'extraction des caractéristiques ,la réduction de nombre de caractéristiques , l'apprentissage pour enfin pouvoir déterminer s'il existe une relation de parenté entre la paire d'image qu'on avait comme entrée. La figure 3.1 représente une description globale de notre système de vérification de parenté qui utilise trois méthode d'extraction de caractéristique par la suite nous allons décomposer ce système en 3 sous système dont chacun utilise une seule méthode d'extraction de caractéristiques.

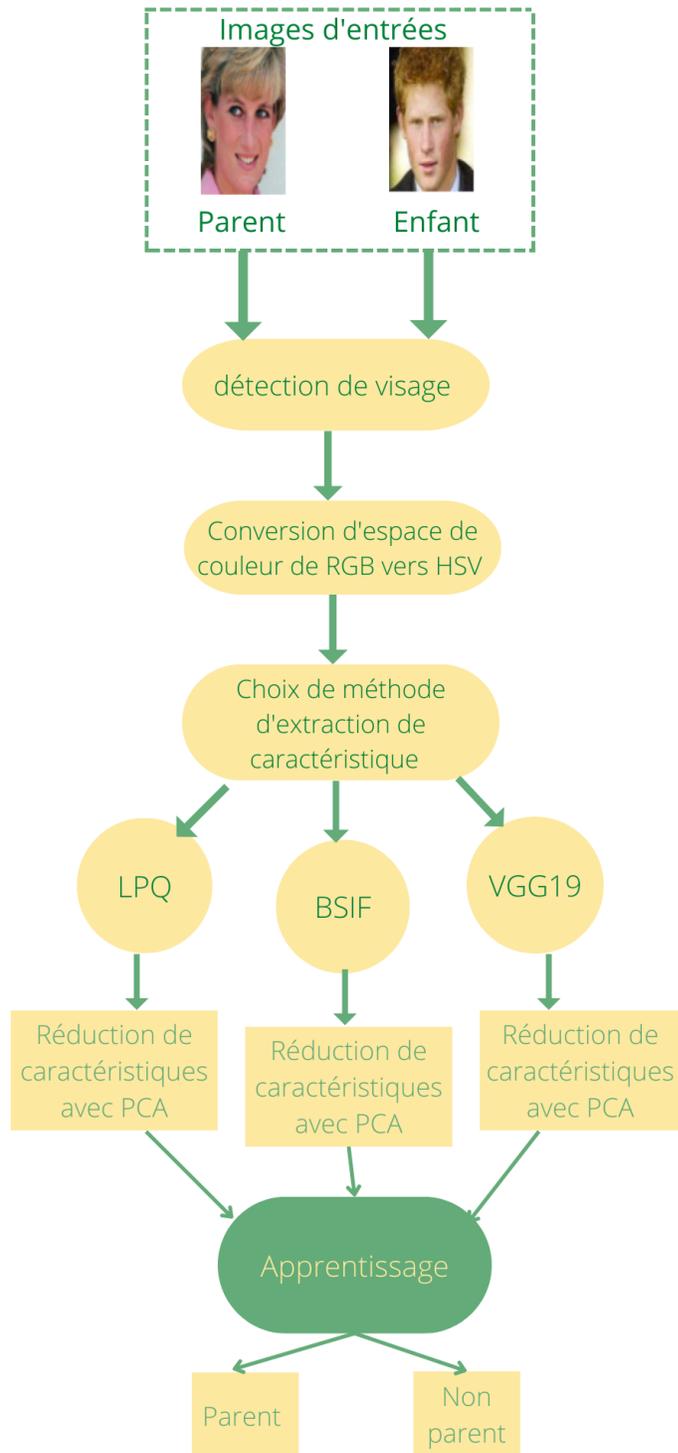


FIGURE 3.1 – Notre système proposé

3.2.1 Détection de visage

La détection de visage est une étape préliminaire dans notre système qui consiste à localiser et d'extraire la région du visage de l'arrière-plan ce qui nous servira dans la suite de notre procédure

cette étape peut se réaliser en utilisant plusieurs méthodes qu'on a citées dans le chapitre précédent vu la vitesse et la précision de détection des cascades de haar nous avons choisi de l'utiliser comme méthode de détection de visage. Séparé la partie visage du reste de l'image d'entrée. Cette étape peut se réaliser en utilisant plusieurs méthodes qu'on a citées dans le chapitre précédent Dans notre travaille nous allons utiliser la détection de visage comme une étape préliminaire afin de localiser et d'extraire la région du visage de l'arrière-plan ce qui nous servira dans la suite de notre procédure , vu la vitesse et la précision de détection des cascades de haar nous avons choisi de l'utiliser comme méthode de détection de visage.

3.2.1.1 L'algorithme de Viola Jones

Cet algorithme comporte quatre étapes principales :

1-Sélection des caractéristiques de type Haar : Tous les visages humains partagent certaines propriétés universelles, comme la région des yeux est plus sombre que les pixels voisins, et la région du nez est plus claire que la région des yeux. Un moyen simple de déterminer quelle région est plus claire ou plus sombre consiste à additionner les valeurs des pixels des deux régions et à les comparer. La somme des valeurs des pixels de la région la plus sombre sera plus petite que la somme des pixels de la région la plus claire. Viola et Jones ont identifié trois types de caractéristiques de type Haar dans leurs recherches : Les caractéristiques de bord, les caractéristiques des lignes et les caractéristiques à quatre côtés . La figure 3.2 nous montre comment s'effectue le balayage des caractéristiques de type Haar sur une image faciale.

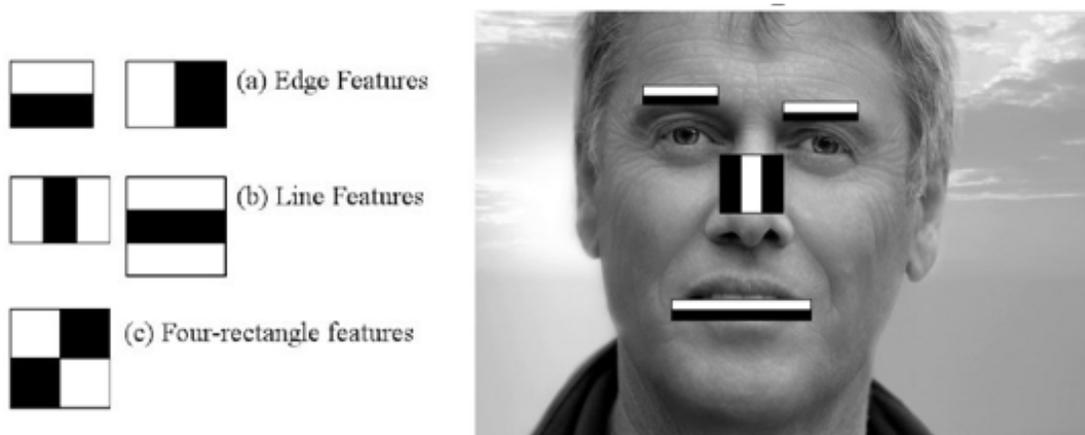


FIGURE 3.2 – Balayage des caractéristiques de type Haar dans l'image

2-Création d'une image intégrale : Dans la section précédente, nous avons vu que pour calculer une valeur pour chaque caractéristique, nous devons effectuer des calculs sur tous les pixels à l'intérieur de cette caractéristique particulière. L'image intégrale joue son rôle en nous permettant d'effectuer rapidement ces calculs intensifs afin de comprendre si une caractéristique ou plusieurs caractéristiques répondent aux critères.

3-Exécution de l'entraînement AdaBoost : Pour décider du type et de la taille d'une caractéristique qui entre dans le classifieur final, AdaBoost vérifie les performances de tous les classifieurs que vous lui fournissez. Pour calculer la performance d'un classificateur, vous l'évaluez sur toutes les sous-régions de toutes les images utilisées pour la formation. Certaines sous-régions produiront une réponse forte dans le classificateur. Elles seront classées comme positives, ce qui signifie que le classificateur pense qu'elles contiennent un visage humain. Le résultat final est un classificateur fort, également appelé classificateur boosté, qui contient les classifieurs faibles les plus performants.

4-Création de cascades de classifieurs : Peut-être que l'AdaBoost sélectionnera finalement les meilleures caractéristiques vers 2500, mais le calcul de ces caractéristiques pour chaque région prend encore beaucoup de temps. Dans la première étape, nous avons un classificateur qui est composé de nos meilleures caractéristiques, en d'autres termes, dans la première étape, la sous-région passe par les meilleures caractéristiques telles que la caractéristique qui identifie l'arête du nez ou celle qui identifie les yeux. Dans les étapes suivantes, nous avons toutes les autres caractéristiques. Lorsqu'une sous-région de l'image entre dans la cascade, elle est évaluée par la première étape. La figure 3.3 montre une image issue de la base de données KINFACEW après avoir effectué le processus de détection de visage



FIGURE 3.3 – Detection de visage

Après l'extraction de la région visage et obtenu une image faciale, l'étape suivante consiste à convertir l'espace colorimétrique de l'image de RGB vers HSV.

3.2.2 Conversion d'espace de couleurs de RGB vers HSV

Après avoir obtenu l'image contenant que le visage on effectue une conversion d'espace de couleur de RGB vers HSV en suivant les formules suivantes :

$$V = \max(R, G, B)$$

$$S = \frac{V - \min(R, G, B)}{V} \quad (3.1)$$

$$H = \begin{cases} \frac{G - B}{V - \min(R, G, B)} & \text{si } V=R \\ 2 + \frac{B - R}{V - \min(R, G, B)} & \text{si } V=G \\ 4 + \frac{R - G}{V - \min(R, G, B)} & \text{si } V=B \\ H + 360 & \text{si } H < 0 \end{cases}$$

Après avoir fait la transformation vers le HSV nous allons séparer indépendamment chaque composante à savoir H, S et V comme le montre la figure 3.4

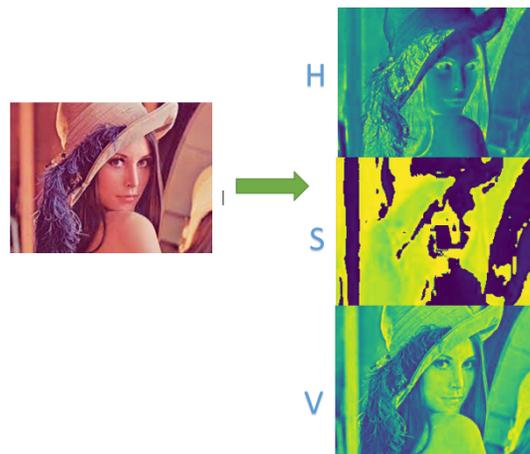


FIGURE 3.4 – Transformations d'espace RGB vers HSV

3.2.3 Extraction de caractéristiques

Après l'obtention des trois images H, S et V l'Etape qui suit consiste à extraire les caractéristiques de chaque image et cela en utilisant trois méthodes qui sont BSIF, LPQ ET VGG19

3.2.3.1 Extraction des caractéristique BSIF

Dans ce descripteur, un code binaire est calculé pour chaque pixel en projetant de façon linéaire des patches locaux sur un sous-espace dont les vecteurs de base sont appris à partir d'images naturelles [2].

La figure 3.5 représente un exemple des filtres utilisés dans l'extraction des caractéristiques BSIF

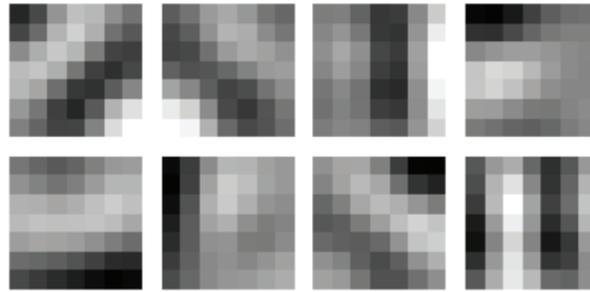


FIGURE 3.5 – différents types de filtres [39]

La réponse du filtre pour un patch est donnée par [39] :

$$s_i = \sum_{u,v} W_i(u,v)X(u,v) = w_i^T x \quad (3.2)$$

$X(u,v)$ est une image et $W_i(u,v)$ est un filtre linéaire. Étant donné les patches aléatoires d'images naturelles, nous déterminons les filtres w_i de manière à ce que les éléments indépendants comme W_{ij} ayant le poids de l'arête reliant les sommets i et j , et 0 s'il n'y a pas d'arête possible. La chaîne de code binaire b , qui correspond à l'image du patch x , est obtenue par la binarisation de chaque élément si les étapes suivantes [39] :

$$B_i = 1 \text{ si } s_i > 0 \text{ et } b_i = 0 \text{ si } s_i < 0 \quad (3.3)$$

De cette façon, les n bits peuvent être calculés en code binaire pour chaque pixel de l'image du visage. La figure 3.6 montre les images résultante de l'application du descripteur Bsif sur les trois images H ,S et V



FIGURE 3.6 – BSIF sur les trois images H ,S et V

Comme deuxième méthode d'extraction des caractéristiques on a la méthode LPQ définie dans le titre suivant

3.2.3.2 Extraction des caractéristiques LPQ :

La LPQ encode l'information de phase locale de quatre fréquences de la transformée de Fourier à court terme. [79]

$$F(u, v) = \sum_y f(x - y) \exp(-j2\pi u^T y) = W_u^T x_f \tag{3.4}$$

Où W_u est le vecteur de base de la transformée de Fourier discrète (DFT) à la fréquence u et x_f est un autre vecteur contenant tous les M_2 fréquence u et x_f est un autre vecteur contenant tous les échantillons d'image M_2 de N_x .

La figure 3.7 illustre les étapes de base de l'opérateur LPQ

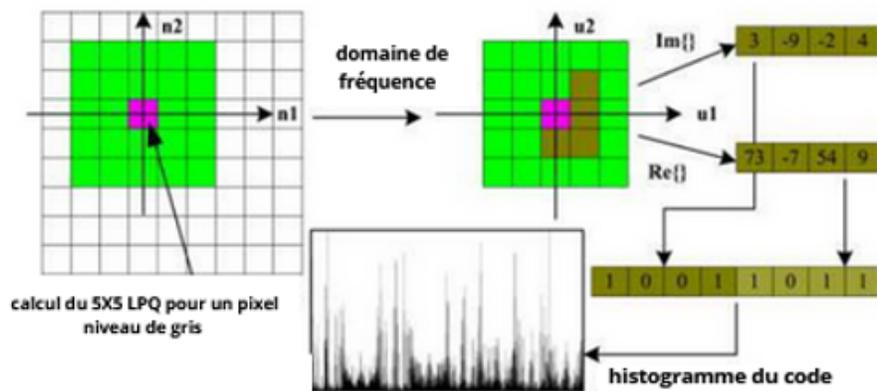


FIGURE 3.7 – les étapes de l'opérateur LPQ de base

L'application du descripteur lpq sur une image nous donne le résultat montré dans la figure 3.8.

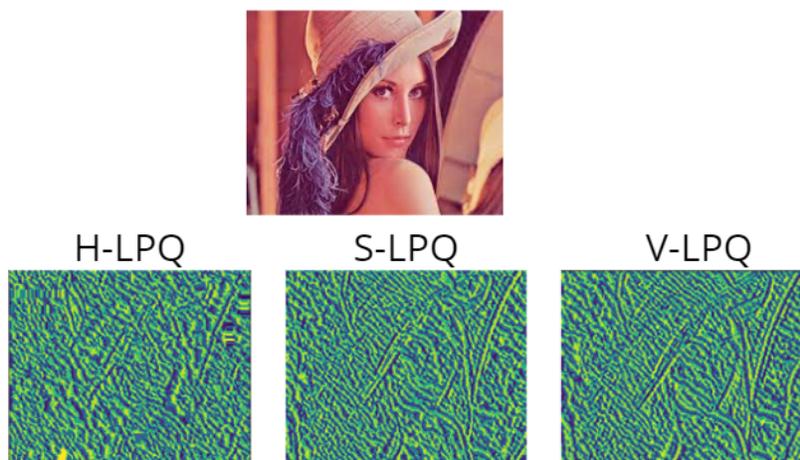


FIGURE 3.8 – LPQ sur les trois images H ,S et V

3.2.3.3 extraction des caractéristique VGG19

VGG [41] est un CNN profond utilisé pour classifier les images. Les couches du modèle VGG19 sont les suivantes :

- Conv3x3 (64)
- Conv3x3 (64)
- MaxPool
- Conv3x3 (128)
- Conv3x3 (128)
- MaxPool
- Conv3x3 (256)
- Conv3x3 (256)
- Conv3x3 (256)
- Conv3x3 (256)
- MaxPool
- Conv3x3 (512)
- Conv3x3 (512)
- Conv3x3 (512)
- Conv3x3 (512)
- MaxPool
- Conv3x3 (512)
- Conv3x3 (512)
- Conv3x3 (512)
- Conv3x3 (512)
- MaxPool
- Fully Connected (4096)
- Fully Connected (4096)
- Fully Connected (1000)
- SoftMax

Son architecture est la suivante :

- Une image RVB de taille fixe ($224 * 224$) a été donnée en entrée de ce réseau, ce qui signifie que la matrice était de forme $(224,224,3)$.
- Le seul prétraitement effectué est la soustraction de la valeur RVB moyenne de chaque pixel, calculée sur l'ensemble de l'apprentissage.
- Ils utilisent des noyaux de taille $(3 * 3)$ avec une taille de pas de 1 pixel, ce qui leur permet de couvrir toute la notion de l'image.
- le rembourrage spatial a été utilisé pour préserver la résolution spatiale de l'image.
- Le max pooling a été effectué sur une fenêtre de $2 * 2$ pixels avec stride 2.
- Il a été suivi d'une unité linéaire rectifiée (ReLU) afin d'introduire une non-linéarité pour que le

modèle classifie mieux et pour améliorer le temps de calcul, car les modèles précédents utilisaient des fonctions tanh ou sigmoïde, qui se sont avérées bien meilleures que celles-ci.

- La dernière étape consiste à mettre mis en œuvre trois couches entièrement connectées, dont les deux premières avaient une taille de 4096, puis une couche de 1000 canaux pour la classification ILSVRC à 1000 voies et la dernière couche est une fonction softmax.

Deux étapes suivent l'obtention des trois images H, S et V de chaque méthode :

1-la concaténation des trois composante H, S et V afin d'obtenir une seule image HSV pour chaque méthode.

2-la concaténation des trois images HSV résultante afin d'avoir une seule matrice contenant les caractéristiques extraite des trois méthodes précédente sur laquelle on applique la réduction des caractéristiques ACP

3.2.4 Réduction de caractéristique avec ACP

L'analyse en composantes principales est une technique de réduction des dimensions bien connue. Elle transforme les variables en un nouvel ensemble de variables appelé composantes principales. Ces composantes principales sont une combinaison linéaire des variables originales et sont orthogonales. La première composante principale représente la plupart des variations possibles des données d'origine et la deuxième composante principale fait de son mieux pour capturer la variance des données. Il ne peut y avoir que deux composantes principales pour un ensemble de données à deux dimensions. Cette méthode est composée de 4 étapes :

Étape 1 : Normalisation

L'objectif de cette étape est de normaliser la gamme des variables initiales continues afin que chacune d'entre elles contribue de manière égale à l'analyse. Plus précisément, la raison pour laquelle il est essentiel d'effectuer la normalisation avant l'ACP est que cette dernière est très sensible aux variances des variables initiales. En d'autres termes, s'il existe de grandes différences entre les plages de variables initiales, les variables dont les plages sont plus grandes domineront celles dont les plages sont plus petites (par exemple, une variable qui se situe entre 0 et 100 dominera une variable qui se situe entre 0 et 1), ce qui entraînera des résultats biaisés. La transformation des données en échelles comparables peut donc éviter ce problème.[38] Mathématiquement, cela peut être fait en soustrayant la moyenne et en divisant par l'écart-type pour chaque valeur de chaque variable.

$$Z = \frac{\text{valeur} - \text{mean}}{\text{stadardeviation}} \quad (3.5)$$

Une fois la standardisation effectuée, toutes les variables seront transformées à la même échelle.

Étape 2 : Calcul de la matrice de covariance L'objectif de cette étape est de comprendre comment les variables de l'ensemble de données d'entrée varient de la moyenne les unes par rapport aux autres, ou en d'autres termes, de voir s'il existe une relation entre elles. Car parfois, les variables sont fortement corrélées de telle sorte qu'elles contiennent des informations redondantes. Ainsi, afin

d'identifier ces corrélations, nous calculons la matrice de covariance.

La matrice de covariance est une matrice symétrique $p \times p$ (où p est le nombre de dimensions) qui a pour entrées les covariances associées à toutes les paires possibles de variables initiales. Par exemple, pour un ensemble de données à 3 dimensions avec 3 variables x , y et z , la matrice de covariance est une matrice 3×3 de ce type : Matrice de covariance pour les données tridimensionnelles

$$\begin{bmatrix} Cov(x, x) & Cov(x, y) & Cov(x, z) \\ Cov(y, x) & Cov(y, y) & Cov(y, z) \\ Cov(z, x) & Cov(z, y) & Cov(z, z) \end{bmatrix} \quad (3.6)$$

Puisque la covariance d'une variable avec elle-même est sa variance ($Cov(a, a) = Var(a)$), dans la diagonale principale (du haut à gauche au bas à droite) nous avons en fait les variances de chaque variable initiale. Et comme la covariance est commutative ($Cov(a, b) = Cov(b, a)$), les entrées de la matrice de covariance sont symétriques par rapport à la diagonale principale, ce qui signifie que les parties triangulaires supérieure et inférieure sont égales.

Étape 3 : Calculer les vecteurs propres et les valeurs propres de la matrice de covariance pour identifier les composantes principales.

Les vecteurs propres et les valeurs propres sont les concepts d'algèbre linéaire que nous devons calculer à partir de la matrice de covariance afin de déterminer les composantes principales des données. Avant de passer à l'explication de ces concepts, comprenons d'abord ce que nous entendons par composantes principales.

Les composantes principales sont de nouvelles variables qui sont construites comme des combinaisons linéaires ou des mélanges des variables initiales. Ces combinaisons sont réalisées de telle sorte que les nouvelles variables (c'est-à-dire les composantes principales) ne sont pas corrélées et que la plupart des informations contenues dans les variables initiales sont comprimées dans les premières composantes. Ainsi, l'idée est que des données à 10 dimensions vous donnent 10 composantes principales, mais l'ACP essaie de mettre le maximum d'informations possibles dans la première composante, puis le maximum d'informations restantes dans la deuxième et ainsi de suite.

Étape 4 : est la sélection des composantes et formation d'un vecteur caractéristique. Prenons l'exemple d'une image de dimension (64,64) après l'application du ACP la taille obtenu est (64,2) et ceci si on choisit 2 comme nombre de composantes principale souhaité. Après l'obtention des images résultante de l'application du ACP sur chaque méthode d'extraction, l'étape qui suit est l'apprentissage.

3.3 L'apprentissage

Avant de commencer l'étape d'apprentissage les étapes précédentes doivent être effectués sur toute les images des bases de donnée utilisé afin d'obtenir les images des caractéristiques avec les quelle on va réaliser l'apprentissage

Les trois sous système de notre architecture utilisent une seule méthode donc une seule matrice de caractéristique pour chaque image de la base de données les figure 3.9 , 3.10 et 3.11 représente les trois sous système de notre approche

La figure 3.9 montre le schéma de notre systeme d'extraction des caractéristiques LPQ.

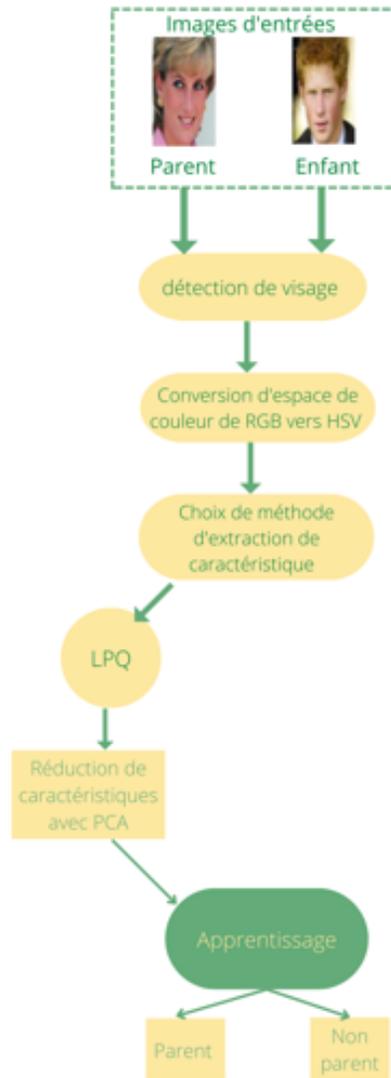


FIGURE 3.9 – Système de verification basé sur la méthode d'extraction de caractéristiques LPQ

La figure 3.10 montre le schéma de notre systeme d'extraction des caractéristiques BSIF.

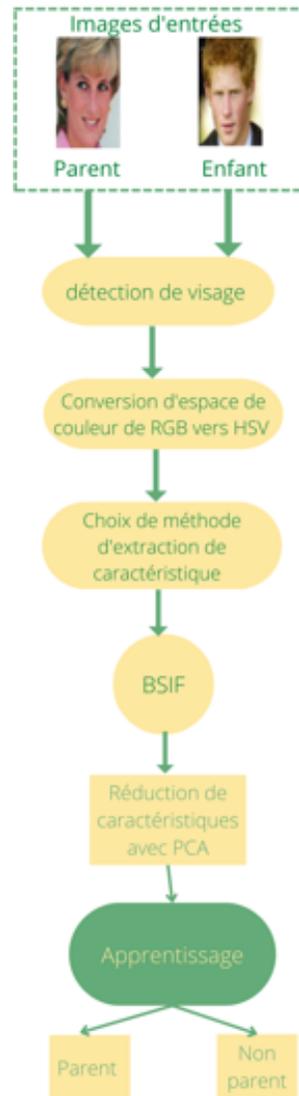


FIGURE 3.10 – Système de vérification basé sur la méthode d'extraction de caractéristiques BSIF

La figure 3.12 montre le schéma de notre système d'extraction des caractéristiques VGG19.

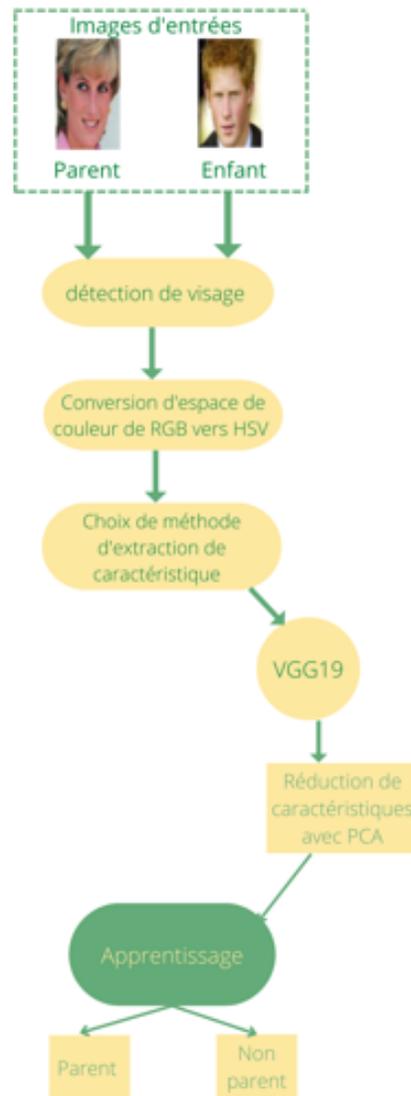


FIGURE 3.11 – Système de vérification basé sur la méthode d'extraction de caractéristiques VGG19

3.3.0.1 Réseaux de neurone convolutionnelle avec keras

Pour la dernière étape de notre système de vérification de parenté on a l'apprentissage avec un réseau de neurones convolutionnelle en utilisant l'API KERAS. Keras est une bibliothèque open source (licence MIT) écrite en Python qui s'appuie principalement sur les travaux réalisés par le développeur de Google François Chollet dans le cadre du projet ONEIRO (Open-ended Neuro-Electronic Intelligent Robot Operating System). La première version de ce logiciel indépendant de la plateforme a été publiée le 28 mars 2015. L'objectif de cette bibliothèque est de permettre le développement rapide de réseaux de neurones. Dans ce cas, Keras n'est pas un framework distinct mais une interface (API) conviviale pour les débutants permettant d'accéder à divers frameworks d'apprentissage automatique et de les programmer. Theano, Microsoft Cognitive Toolkit (anciennement CNTK) et TensorFlow sont quelques-uns des frameworks pris en charge par Keras. Keras

est une bibliothèque qui fonctionne avec des modèles. Elle fournit les blocs de construction pour le développement de modèles d'apprentissage profond complexes. Contrairement aux frameworks indépendants, ce logiciel open source ne traite pas lui-même les opérations simples de bas niveau. Il utilise plutôt les bibliothèques des cadres d'apprentissage automatique associés à cette fin. Celles-ci agissent comme une sorte de moteur dorsal pour Keras. L'idée étant d'être modulaire, les couches souhaitées pour le réseau de neurones en cours de développement sont connectées les unes aux autres sans que l'utilisateur de Keras ait besoin de comprendre ou de contrôler le back-end réel du framework sélectionné.

-les étapes de base de la construction d'un modèle dans Keras : les étapes de base de la construction d'un modèle dans Keras : **-Définir un réseau :** Dans cette étape, vous définissez les différentes couches de notre modèle et les connexions entre elles. Keras dispose de deux principaux types de modèles : Les modèles séquentiels et fonctionnels. Vous choisissez le type de modèle que vous voulez, puis vous définissez le flux de données entre eux. Pour notre part nous avons choisi d'utiliser le modèle séquentiel.

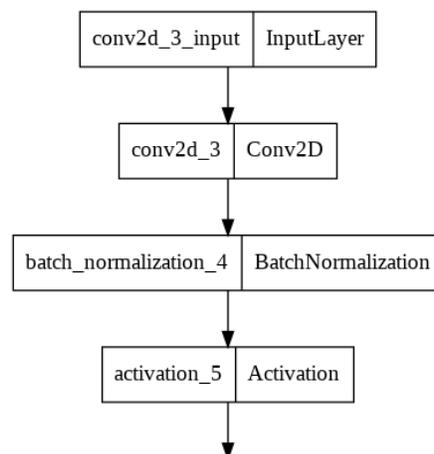
-Compilez un réseau : Compiler du code signifie le convertir en une forme adaptée à la compréhension de la machine. Dans Keras, la méthode `model.compile()` remplit cette fonction. Pour compiler le modèle, Nous définissons la fonction de perte qui calcule les pertes dans notre modèle, l'optimiseur qui réduit les pertes et la métrique qui est utilisée pour déterminer la précision de notre modèle.

-Ajuster le réseau : Cette étape permet d'adapter notre modèle à nos données après la compilation. Ceci est utilisé pour entraîner le modèle sur nos données.

-Évaluer le réseau : Après avoir ajusté notre modèle, nous devons évaluer l'erreur dans notre modèle.

-Faire des prédictions : Nous utilisons `model.predict()` pour faire des prédictions en utilisant notre modèle sur de nouvelles données.

La Figure 3,12 montre l'architecture de notre modèle séquentiel réalisé.



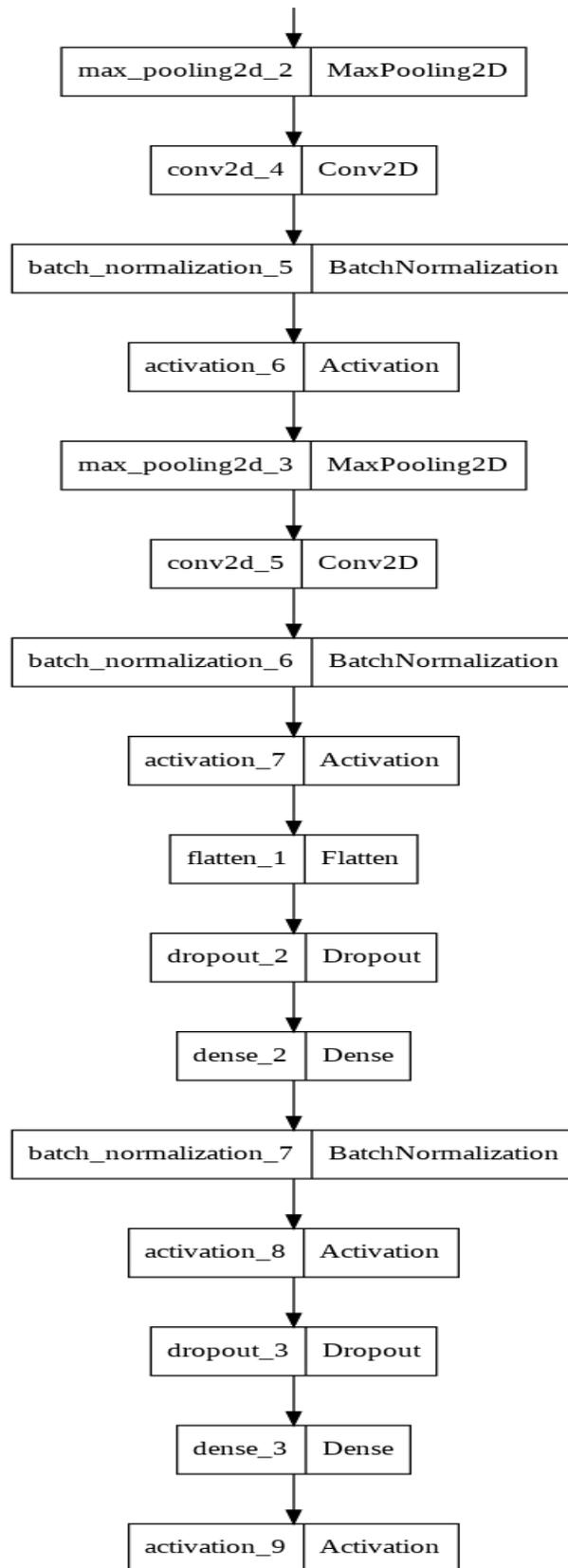


FIGURE 3.12 – Modèle de reseau de neurone utilisé

Il existe trois types de couches dans qui composent le CNN, a savoir les couches de convolution, les couches de mise en commun et les couches entièrement connectées (FC). Lorsque ces couches sont empilées, une architecture CNN est formée. ce qui est montrée dans la figure précédente. Pour ces plusieurs couches, nous pouvons avoir beaucoup de valeurs. Cela dépend principalement du fait qu'un neurone spécifique soit activé ou non. Ainsi, pour normaliser cette gamme de valeurs, nous utilisons des fonctions d'activation afin de rendre l'ensemble du processus statistiquement équilibré. Pour notre part nous avons choisi d'utiliser deux fonction d'activation qui sont sigmoid et softmax

La fonction Sigmoid : La fonction d'activation sigmoïde est une fonction mathématique dont la courbe en forme de "S" est reconnaissable. Elle est utilisée pour la régression logistique et la mise en œuvre de réseaux neuronaux de base. Si nous voulons avoir un classificateur pour résoudre un problème avec plus d'une bonne réponse, la fonction sigmoïde est le bon choix. Nous devons appliquer cette fonction à chaque élément de la sortie brute indépendamment. La valeur de retour de la fonction Sigmoïde est généralement comprise entre 0 et 1 ou entre -1 et 1. Voici à quoi ressemble la fonction sigmoïde :

$$f(x) = \text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (3.7)$$

La fonction Softmax : La fonction d'activation Softmax, également connue sous le nom de SoftArgMax ou de fonction exponentielle normalisée, est une fonction d'activation fascinante qui prend des vecteurs de nombres réels en entrée et les normalise en une distribution de probabilité proportionnelle aux exponentielles des nombres d'entrée. Avant l'application, certaines données d'entrée peuvent être négatives ou supérieures à 1, et leur somme peut ne pas être égale à 1. Après l'application de Softmax, chaque élément sera compris entre 0 et 1, et la somme des éléments sera égale à 1. De cette façon, ils peuvent être interprétés comme une distribution de probabilité. Pour plus de clarté, plus le nombre d'entrée est grand, plus les probabilités seront grandes. Voici à quoi ressemble la fonction softmax :

$$\text{softmax}(z_j) = \frac{e^{z_j}}{\sum_{k=1}^K} \text{ for } j = 1, \dots, K \quad (3.8)$$

Au plus des quatres systemes définis auparavant nous avons réalisé un ciquieme systeme basé sur un autre type de methode de vérification de parenté basé sur le calcul de similarités entre les images parents et enfant qui est la méthode Cosine similarty.

3.3.1 Cosine similarity

La similarité en cosinus est une mesure qui quantifie la similarité entre deux ou plusieurs vecteurs. c'est le cosinus de l'angle entre les vecteurs. Les vecteurs sont généralement non nuls et se situent dans un espace de produit interne. La similarité en cosinus est décrite mathématiquement comme la division entre le produit scalaire des vecteurs et le produit des normes euclidiennes ou de la magnitude de chaque vecteur.

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (3.9)$$

La similarité en cosinus est une valeur qui est limitée par un intervalle contraint de 0 et 1.

ou la mesure de similarité est une mesure du cosinus de l'angle entre les deux vecteurs non nuls A et B.

Supposons que l'angle entre les deux vecteurs soit de 90 degrés. Dans ce cas, la mesure de similarité en cosinus aura une valeur de 0 ; cela signifie que les deux vecteurs sont orthogonaux ou perpendiculaires l'un à l'autre.

Plus la mesure de la similarité en cosinus se rapproche de 1, plus l'angle entre les deux vecteurs A et B est petit. Les images ci-dessous illustrent plus clairement ce phénomène.

La figure 3.13 nous montre le schéma de vérification basé sur le calcul de similarité.

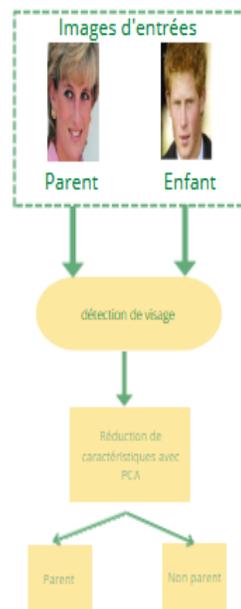


FIGURE 3.13 – Système de vérification basé sur le calcul de similarité en utilisant cosine similarity

3.4 Conclusion

Dans ce chapitre nous avons détaillé le fonctionnement des cinq systèmes de vérification de parenté réalisés : les systèmes basés sur l'extraction des caractéristiques ainsi que ceux basés sur le calcul de similarité ;

Conception et implémentation

4.1 Introduction

On explique dans ce chapitre l'implémentation des quatre système de vérification de parenté qu'on a expliqué auparavant en commençant par représenter l'environnement de développement ensuite on va donner un aperçu sur la base de données utilisé dans la partie apprentissage et enfin des testes et une discussion de résultat est exposé .

4.2 Environnement de développement

Python est un langage de programmation inventé par Guido van Rossum. La première version de python est sortie en 1991.

C'est un langage de programmation interprété, c'est-à-dire qu'il n'est pas nécessaire de le compiler avant de l'exécuter. Si vous avez déjà touché un peu à la programmation, vous verrez que ce langage possède une certaine poésie. Les programmeurs s'amuse souvent à trouver la manière la plus élégante et efficace d'écrire une suite d'instructions.

On a choisi de travailler avec cet environnement en raison de ses plusieurs avantages :

Comme il n'y a pas d'étape de compilation, le cycle édition-test-débugage est incroyablement rapide. Le débogage des programmes Python est facile : un bug ou une mauvaise entrée ne provoquera jamais une erreur de segmentation. Au contraire, lorsque l'interpréteur découvre une erreur, il lève une exception. Si le programme n'attrape pas l'exception, l'interpréteur imprime une trace de la pile. Un débogueur au niveau de la source permet d'inspecter les variables locales et globales, d'évaluer des expressions arbitraires, de définir des points d'arrêt, de parcourir le code ligne par ligne, etc. Le débogueur est écrit en Python même, ce qui témoigne de la puissance introspective de Python. D'un autre côté, le moyen le plus rapide de déboguer un programme est souvent d'ajouter quelques instructions d'impression au code source : le cycle rapide édition-test-débugage rend cette approche simple très efficace. Il existe plusieurs application web ou on peut développer on utilisant le python comme langage de programmation citons Spider ,Jupyter Lab,Pycharm et l'application la plus utiliser dans Le Deep learning qui est Jupyter notebook .

4.2.1 Jupyter notebook

Le Jupyter Notebook est une application web open source que vous pouvez utiliser pour créer et partager des documents contenant du code, des équations, des visualisations et du texte. Jupyter Notebook est maintenu par les personnes du projet Jupyter. Jupyter Notebooks est un projet dérivé du projet IPython, qui avait lui-même un projet IPython Notebook. Le nom, Jupyter, vient des principaux langages de programmation qu'il supporte : Julia, Python et R. Jupyter est livré avec le noyau IPython, qui vous permet d'écrire vos programmes en Python, mais il existe actuellement plus de 100 autres noyaux que vous pouvez également utiliser.

4.2.2 Tkinter

Python offre diverses options pour développer des interfaces graphiques (GUI). Parmi Les plus utilisé on trouve Tkinter qu'on a utilisé pour l'implémentation de notre interface. Tkinter est l'interface Python de la boîte à outils Tk GUI livrée avec Python. Tkinter est la bibliothèque standard d'interface graphique pour Python. Combiné à Tkinter, Python offre un moyen rapide et facile de créer des applications graphiques. Tkinter fournit une interface orientée objet puissante à la boîte à outils Tk GUI. Pour créer une application GUI en utilisant Tkinter tout ce que vous avez à faire, c'est de suivre les étapes suivantes :

- Importer le module Tkinter.
- Créer la fenêtre principale de l'application GUI.
- Ajouter un ou plusieurs des widgets mentionnés ci-dessus à l'application GUI.
- Entrez dans la boucle d'événement principale pour prendre des mesures contre chaque événement déclenché par l'utilisateur.

4.3 Les Bases de Données Utilisées

Nous avons utilisé les deux bases de données de kinfaceW qui sont KinFaceW-I et KinFaceW-II . Les images de visages sont collectées sur Internet, y compris des images de visages de personnalités publiques ainsi que des images de visages de leurs parents ou de leurs enfants. Les images de visages sont capturées dans des environnements non contrôlés dans deux ensembles de données sans aucune restriction en termes de pose, d'éclairage, d'arrière-plan, d'expression, d'âge, d'origine ethnique et d'occlusion partielle. La différence entre KinFaceW-I et KinFaceW-II est que les images de visage avec une relation de parenté ont été acquises à partir de différentes photos dans KinFaceW-I et de la même photo dans KinFaceW-II dans la plupart des cas. Il existe quatre relations de parenté dans les deux ensembles de données : Père-fils (F-S), Père-fille (F-D), Mère-fils (M-S) et Mère-fille (M-D). Dans le jeu de données KinFaceW-I, il y a 156, 134, 116 et 127 paires d'images de parenté pour ces quatre relations. Pour le jeu de données KinFaceW-II, chaque relation contient 250 paires d'images de parenté. Pour faciliter l'utilisation, nous avons étiqueté manuellement les coordonnées de la position des yeux de chaque image de visage, puis nous avons aligné et recadré la région du

visage en 64x64 pour supprimer le fond. Les figures 4.1 suivantes montrent quelques images de visage recadrées à partir des ensembles de données KinFaceW-I et KinFaceW-II, respectivement.[53]



FIGURE 4.1 – Exemples alignés et recadrés des ensembles de données KinFaceW-I et KinFaceW-II, respectivement. De haut en bas, on trouve les relations de parenté F-S, F-D, M-S et M-D, et les deux images voisines de chaque rangée présentent la relation de parenté, respectivement.

4.4 Architecture de l’application

Dans cette section on va présenter les différentes composantes de notre application qu’on a réalisé pour l’exécution de notre système la figure 4.2 représente une structure générale de notre application

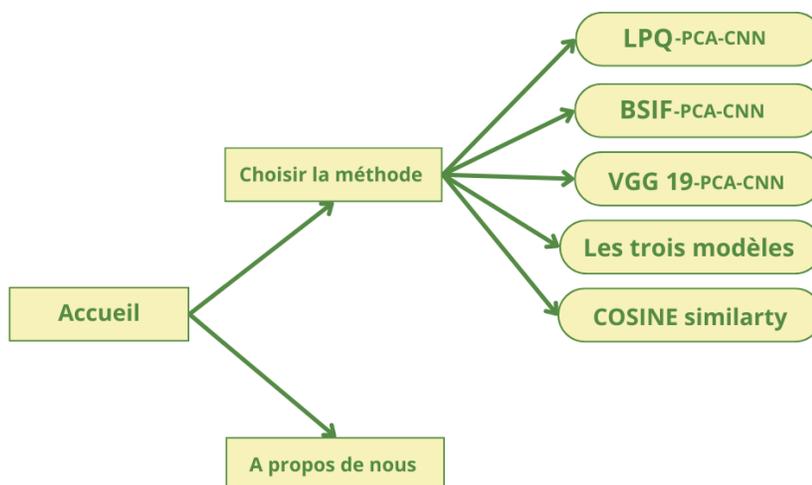


FIGURE 4.2 – Structure générale de notre application

4.4.1 Accueil



FIGURE 4.3 – Interface accueil

(1)- Ce bouton nous permet d'accéder à la fenêtre à-propos de nous

(2)-Ce bouton nous permet d'accéder à un environnement (interface) consacré pour le systeme de verification de parenté basé sur la méthode d'extraction des caracteristiques LPQ

(3)-Ce bouton nous permet d'accéder à un environnement (interface) consacré pour le systeme de verification de parenté basé sur la méthode d'extraction des caracteristiques BSIF .

(4)-Ce bouton nous permet d'accéder à un environnement (interface) consacré pour le systeme de verification de parenté basé sur la méthode d'extraction des caracteristiques VGG19 .

(5)-Ce bouton nous permet d'accéder à un environnement (interface) consacré pour le systeme de verification de parenté basé sur la concaténation des trois methode d'extraction des caracteristique LPQ,BSIF et VGG19.

(6)-Ce bouton nous permet d'accéder à un environnement (interface) consacré pour le systeme de verification de parenté basé sur la méthode Cosine Similarity .

En cliquant sur le bouton a propos de nous la fentrere montré dans la figure 4.4 s'ouvre.

4.4.2 A propos de nous



FIGURE 4.4 – Interface A propos de nous

4.4.3 Choisir la méthode

L'interface d'accueil montré dans la figure 4.3 nous donne le choix entre cinq systèmes de vérification de parenté chacun basé sur une ou des méthodes précises.

4.4.3.1 LPQ-PCA-CNN

En cliquant sur le bouton LPQ-PCA-CNN la fenêtre suivante s'ouvre :

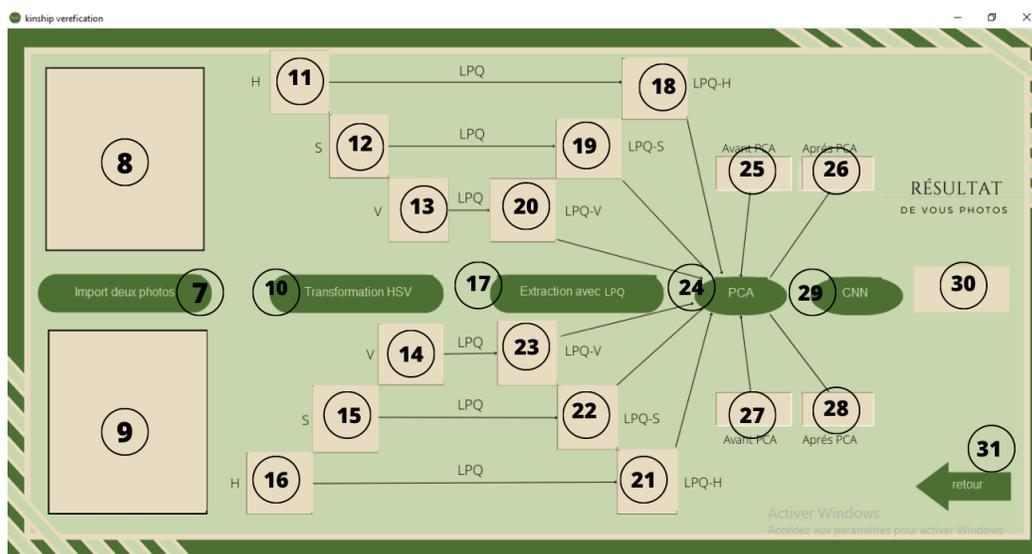


FIGURE 4.5 – Importer deux photos

cette fenetre comporte au total six boutons et :

(7)-Bouton importer deux photos :qui nous permet de charger deux images qui sont susceptible d'avoir une relation de parenté de n'importe quel répertoire existant sur la machine et les afficher dans (8) et (9).

(8),(9)-Affichage des photos chargé auparavant

(10)-Bouton Transformation HSV : qui nous permet de changer l'espace colorimétrique des deux images chargée du RGB vers HSV et afficher leurs trois composantes H,S et V dans les décomposé en H et les afficher dans (11),(12),(13),(14),(15)et (16)respectivement.

(17)-Extraction avec LPQ :qui nous permet d'appliquer la méthode d'extraction LPQ sur les six images résultante du bouton précédent et les afficher dans (18),(19),(20),(21),(22),(23) .

(24)Bouton PCA : qui nous permet d'appliquer la methode de reduction des caracteristique PCA sur les deux images BSIF-HSV resultante apres concatenation de leur trois composante H,S et V et afficher la taille de l'image avant et apres l'application de cette methode dans (25),(26),(27)et(28) respectivement.

(29)Bouton CNN :Nous permet d'utiliser le modele CNN réalisé et entraîné auparavant afin de predire la decision pour les deux images en entré si il existe une relation entre ces deux derniere on aura un 1 afficher dans (30) sinon 0.

(31) Bouton Retour :nous permet de retourner a la fenetre d'accueil

Exemples : 1-Importer deux photos :

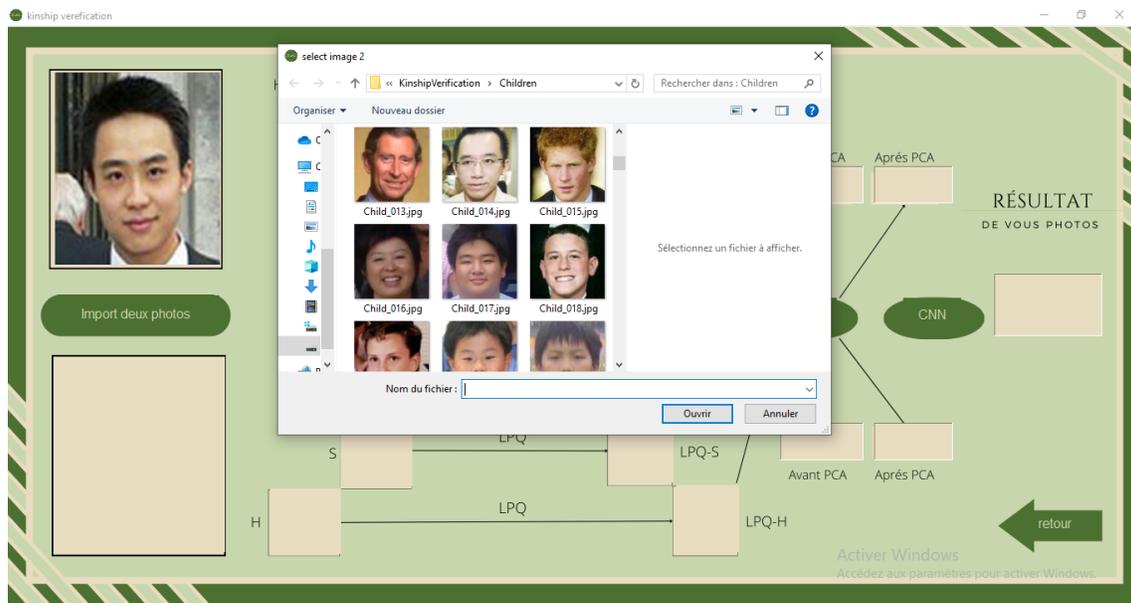


FIGURE 4.6 – Importer deux photos

2-Transformation HSV

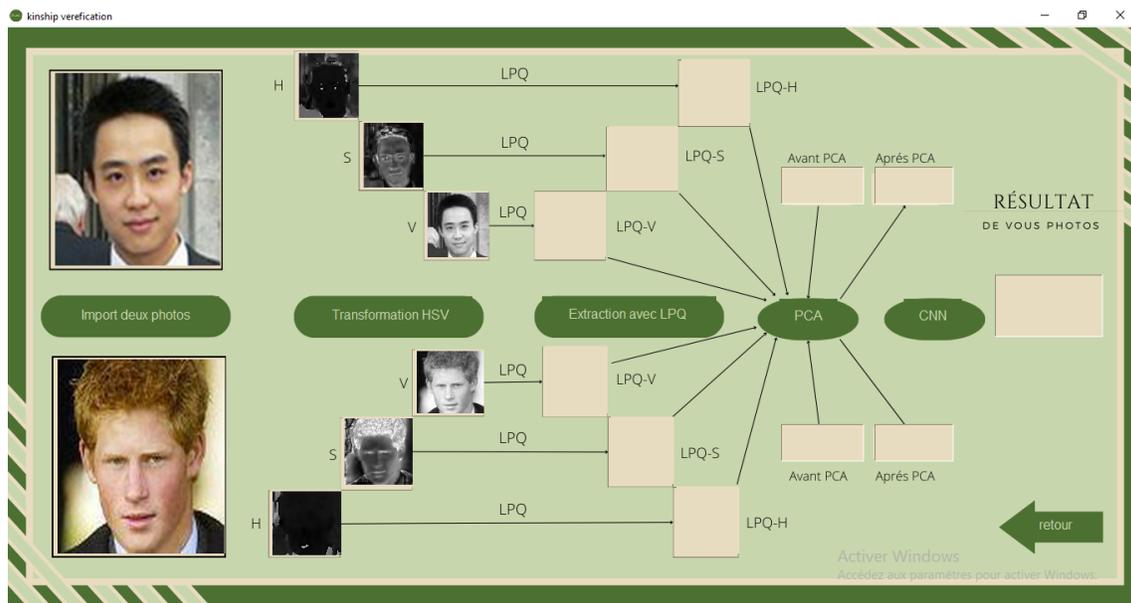


FIGURE 4.7 – Application de la transformation HSV

3-Extraction LPQ :

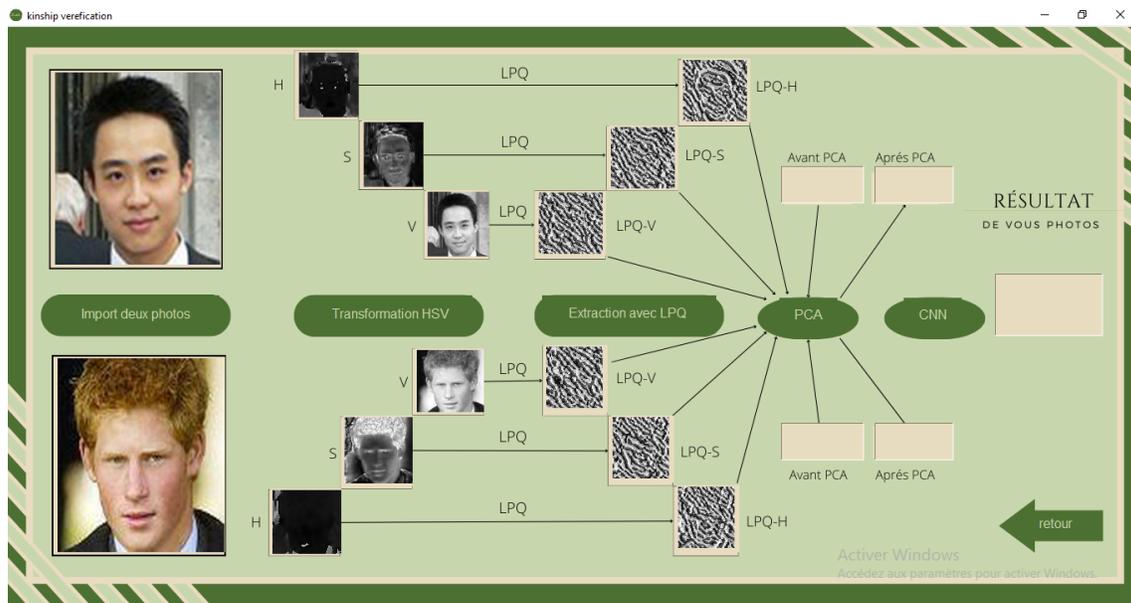


FIGURE 4.8 – Application de l'extraction des caractéristiques LPQ

4-PCA :

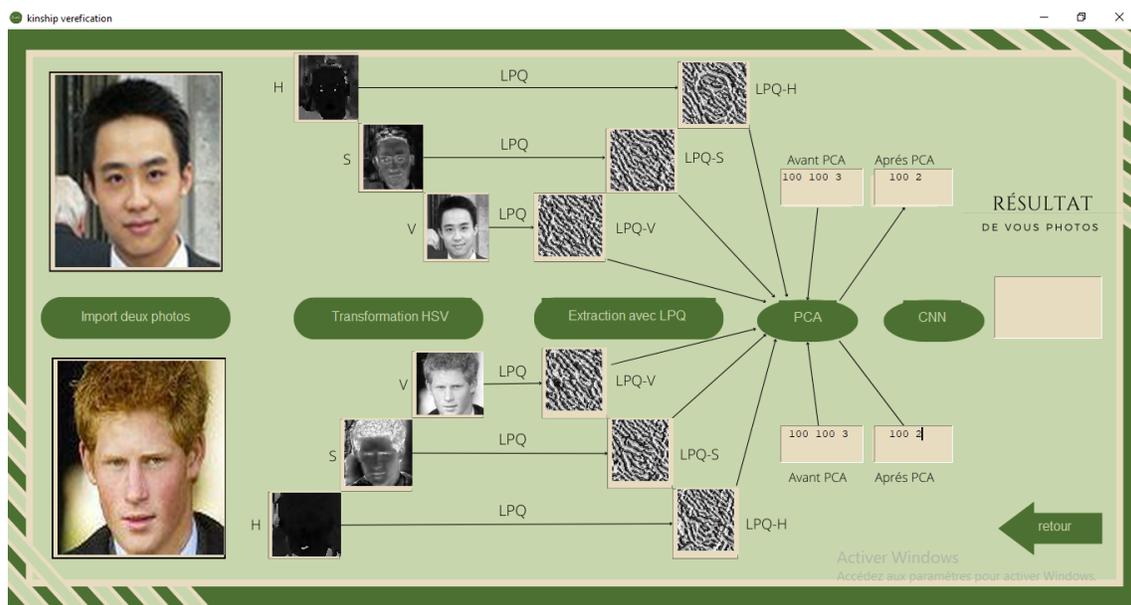


FIGURE 4.9 – Application de la méthode de reduction des caractéristiques PCA

5-CNN :

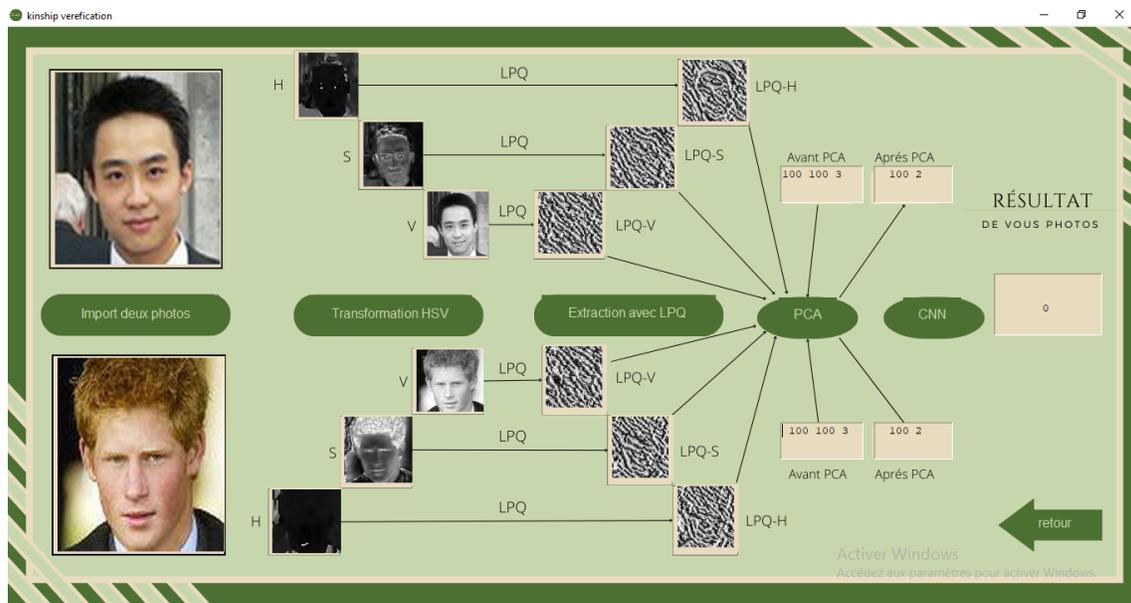


FIGURE 4.10 – Application du modele cnn

4.4.3.2 BSIF-PCA-CNN

En cliquant sur le bouton BSIF-PCA-CNN la fenetre montré dans la figure 4.6 s’ouvre.Elle est composé de six boutons et qui font le meme travail que la fenetre cité auparavant avec une difference de methode d’extraction qui est le BSIF.

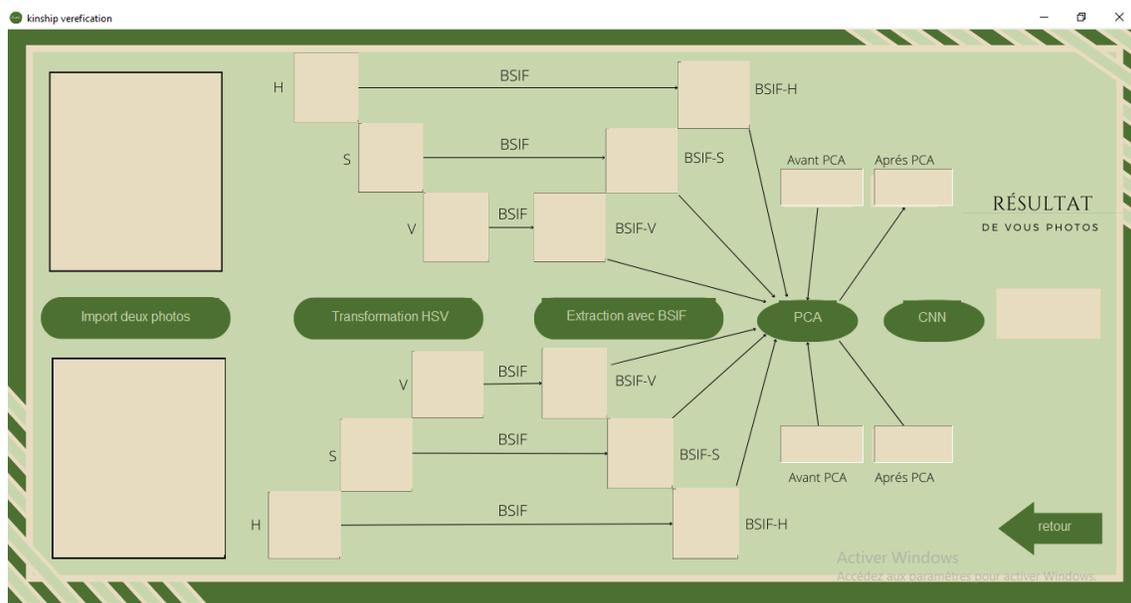


FIGURE 4.11 – Interface de la méthode BSIF

4.4.3.3 VGG19-PCA-CNN

En cliquant sur le bouton VGG19-PCA-CNN la fenetre montré dans la figure 4.7 s’ouvre.Elle est composé de six boutons qui font le meme travaille que les fenetres cité auparavant avec une méthode d’extraction de caractéristique differente qui est VGG19.

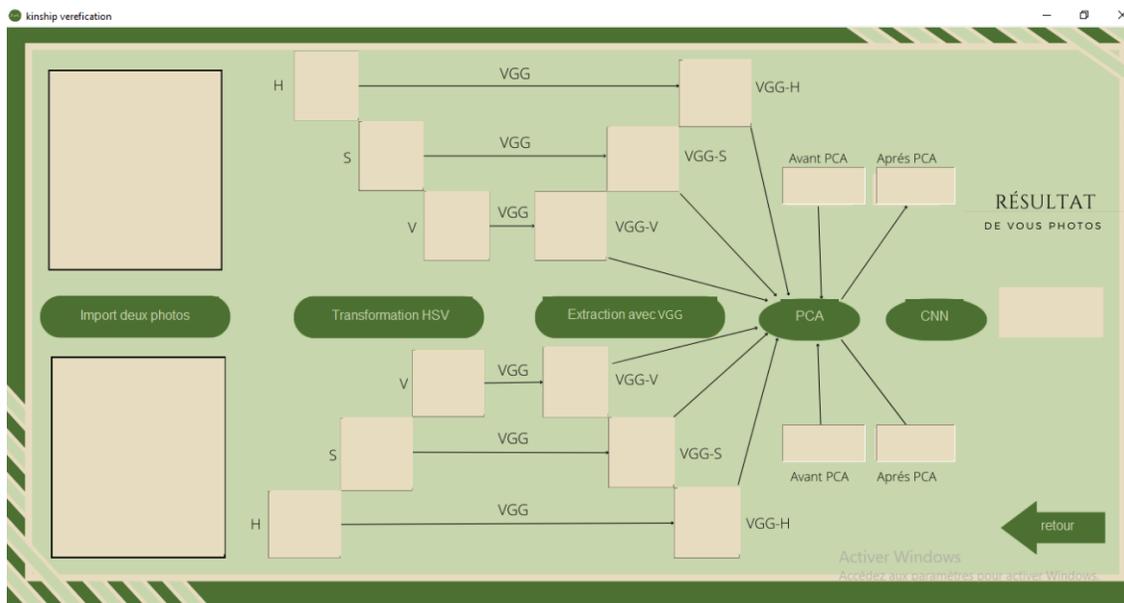


FIGURE 4.12 – Interface de la méthode VGG

4.4.3.4 LPQ-BSIF-VGG19

En cliquant sur le bouton LPQ-BSIF-VGG-PCA-CNN la fenetre montré dans la figure 4.9 s’ouvre.Elle est composé de deux boutons :

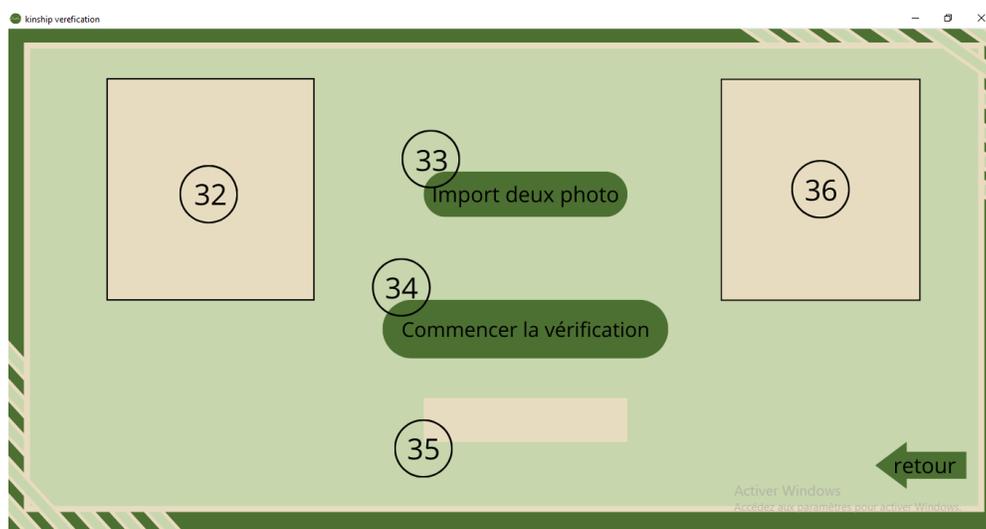


FIGURE 4.13 – Interface de la méthode concatination

(33)-Bouton importer deux photo qui nous permet d’importer deux photo et les afficher dans

(32),(36).

(34)-Bouton Commencer la verification qui nous permet d'appliquer le modele CNN et de predire la decision sur les images de caracteristique resultante de la concatenation des trois methode d'extraction et afficher le resultat dans (35).

4.4.3.5 Cosine similarity

En cliquant sur le bouton Cosine similarity la fenetre montrée dans la figure 4.9 s'ouvre. Elle est composée de trois boutons :

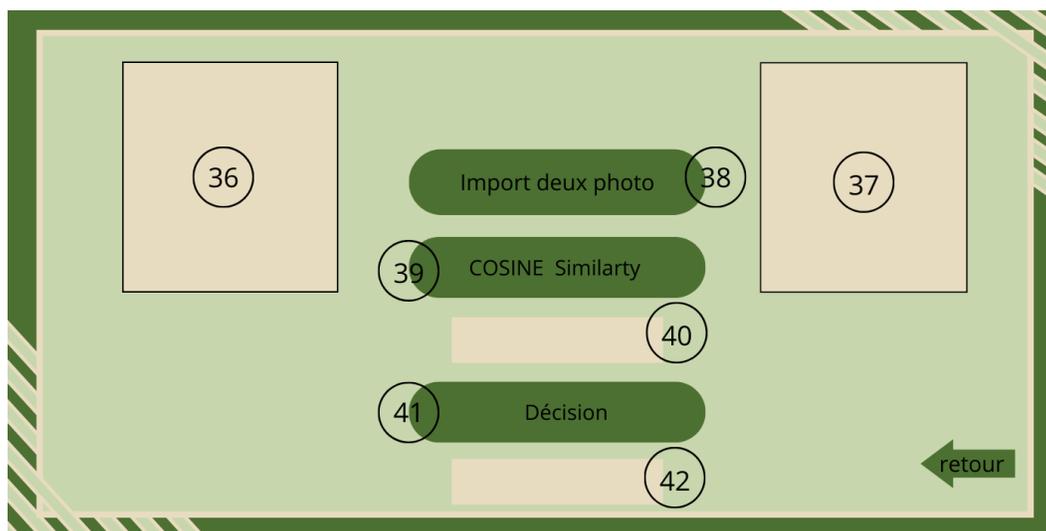


FIGURE 4.14 – Interface de la méthode cosme semilarity

(38)-Bouton importer deux photo qui importe deux images et les affiche dans (3-),(37).

(39)Bouton Cosine Similarity : qui nous permet de calculer la similarity entre les deux images en entrée et l'afficher dans (40).

(41)Bouton Décision qui nous permet d'afficher la decion 1(parent) ou 0(non parent) dans (42) et ceci apres application d'un seuille sur la valeur de la similarity cité auparavant.

4.5 Test et Résultat

Pour vérifier l'efficacité de notre systeme nous avons effectué des tests interne et externe sur les deux bases d'images en utilisant cinq méthodes(LPQ,BSIF,VGG19,la concaténation des trois méthodes et cosme similarity) Ces tests consistent a donner un pourcentage de réussite de classification selon les deux classe parent et non parent.

4.5.1 LPQ test

4.5.1.1 KinFaceW-I

Le tableau 4.1 représente les tests sur la base KinFaceW-I avec la méthode LPQ.

Noyau	Test type	Parent	Faux parent
sigmoid	Interne	52%	30%
	Externe	50%	40%
softmax	Interne	50.30%	33.14%
	Externe	47.4%	60.20%

TABLE 4.1 – test LPQ

Le tableau 4.2 représente les tests sur la base KinFaceW-II avec la méthode LPQ.

4.5.1.2 KinFaceW-II

Noyau	Test type	Parent	Faux parent
sigmoid	Interne	65.13%	35.444%
	Externe	60.084	38.03
softmax	Interne	62.44%	30.8%
	Externe	60.75%	40.07%

TABLE 4.2 – test LPQ

4.5.2 BSIF test

4.5.2.1 KinFaceW-I

Le tableau 4.3 représente les tests sur la base KinFaceW-I avec la méthode BSIF.

Noyau	Test type	Parent	Faux parent
sigmoid	Interne	70.14%	25.2%
	Externe	69.42%	30.54%
softmax	Interne	69.13%	24.44%
	Externe	67.4%	32.97%

TABLE 4.3 – test BSIF

4.5.2.2 KinFaceW-II

Le tableau 4.4 représente les tests sur la base KinFaceW-II avec la méthode BSIF.

Noyau	Test type	Parent	Faux parent
sigmoid	Interne	75.7%	30.4%
	Externe	72.90	20.318
softmax	Interne	70.9%	23.1%
	Externe	70.2%	35%

TABLE 4.4 – test BSIF

4.5.3 VGG19 test

4.5.3.1 KinFaceW-I

Le tableau 4.5 représente les tests sur la base KinFaceW-I avec la méthode VGG19.

Noyau	Test type	Parent	Faux parent
sigmoid	Interne	87.7%	15.3%
	Externe	79.1%	22%
softmax	Interne	85%	3.53%
	Externe	80%	14.17%

TABLE 4.5 – VGG-19

4.5.3.2 KinFaceW-II

Le tableau 4.6 représente les tests sur la base KinFaceW-II avec la méthode VGG19.

Noyau	Test type	Parent	Faux parent
sigmoid	Interne	91.18%	9.74%
	Externe	90.0	15.42
softmax	Interne	90.13%	15.52%
	Externe	87.5%	12.99%

TABLE 4.6 – test VGG-19

4.5.4 Concatenation test

4.5.4.1 KinFaceW-I

Le tableau 4.7 représente les tests sur la base KinFaceW-I avec la concaténation des trois méthodes.

Noyau	Test type	Parent	Faux parent
sigmoid	Interne	95.27%	3.51%
	Externe	98.4%	25.5%
softmax	Interne	93.04%	13.52%
	Externe	90.74%	13.84%

TABLE 4.7 – test Concatenation

4.5.4.2 KinFaceW-II

Le tableau 4.7 représente les tests sur la base KinFaceW-II avec la concaténation des trois méthodes.

Noyau	Test type	Parent	Faux parent
sigmoid	Interne	96.87%	2.024%
	Externe	98.2%	4.02%
softmax	Interne	95.01%	3.24%
	Interne	94.13%	2.574%

TABLE 4.8 – test LPQ

4.5.5 COSINE Similarity test

4.5.5.1 KinFaceW-I

Le tableau 4.8 représente les tests sur la base KinFaceW-I avec la méthode cosine similarity.

article	parent	faux parent
Interne	61.4%	30%
Externe	41.4%	62.6%

TABLE 4.9 – test cosine similarity

4.5.5.2 KinFaceW-II

Le tableau 4.8 représente les tests sur la base KinFaceW-II avec la méthode cosine similarity.

test type	parent	faux parent
Interne	70.25%	20.91%
Externe	68.014%	45.82%

TABLE 4.10 – test cosine similarity

4.5.6 Interprétation et discussion des résultat

En comparant des résultats des cinq systèmes on remarque que la méthode basée sur BSIF a eu de meilleurs résultats par rapport à celle basée sur LPQ et ceci grâce à la précision d'extraction de caractéristique BSIF qui produit des traits plus significatifs par rapport à la méthode LPQ. D'autre part les résultats de VGG19 sont plus satisfaisants par rapport aux deux cités auparavant et ceci grâce à son architecture basée sur un réseau CNN et son préentraînement sur l'extraction des caractéristiques les plus pertinentes. La concaténation de ces trois méthodes nous a fourni le meilleur résultat soit un taux de précision de 98.2%, un résultat jugé meilleur que plusieurs travaux cités dans la section des travaux connexes tel que le travail de Wu et al [83] en 2016 qui ont réalisé trois modèles de vérification : le modèle BSIF avec un taux de réussite de 81.47% dans l'espace HSV, le modèle

LPQ avec un taux de réussite de 80.10% et le modele NRML avec un taux de réussite de 81.27% .Nos résultats se rapproche des résultat de la méthode de Kohli et al [43] en 2017 avec un taux de réussite de 96.2% en se basant sur leur méthode (KVRL-fcDBN).Par conséquent on conclut que la concaténation des methode d'extraction de caractéristiques : LPQ,BIF et VGG19 est un systeme de vérification pertinent alors on juge notre approche satisfaisante.

4.6 conclusion

dans ce chapitre nous avons présenté en détail la conception de notre systeme, ainsi les interface de notre application et les test et les résultat de notre apprentissage .

Conclusion et perspectives

La vérification de la parenté a attiré beaucoup d'attention ces dernières années en raison de sa capacité à améliorer les systèmes biométriques, en tant que biométrie douce pour la vérification du visage (traits de parenté).

Son rôle est important dans de nombreuses applications sociales, parmi ces applications, citons la création d'arbres généalogiques, l'organisation d'albums de famille l'annotation d'images, la recherche d'enfants disparus et la criminalistique.

Bien que le test ADN soit le moyen le plus fiable pour la vérification de la parenté, il ne peut être utilisé dans de nombreuses situations telle que la vérification automatique de la parenté à partir d'images faciales, et cela peut être effectuée, par exemple sur des scènes de vidéosurveillance. L'objectif de notre travail était de mettre en oeuvre un système de vérification de la parenté en utilisant les réseaux de neurones convolutifs CNN.

Notre système comprend 5 étapes importante : La détection de visage, transformation d'espace de couleur, extraction des caractéristiques, la réduction des caractéristiques, et l'apprentissage par les réseaux de neurones convolutifs, pour obtenir un résultat : parent ou non parent pour les images d'entrée.

La première étape consiste a faire une détection de visage avec les filtres de Haar.

La seconde étape consiste a transformé l'espace de couleurs RGB vers le HSV.

La troisième étape Extraction des caractéristiques avec trois méthodes : BSIF, LPQ, VGG19 ainsi que la concatination des trois méthodes.

La quatrième étape réduction des caractéristiques avec la méthodes ACP.

Pour arriver a la dernière étape afin d'avoir un résultat binaire pour la décision parent ou non parent en utilisant le modèle séquentiel.

Les résultats obtenus était d'une précision de 98.2% qui sont jugé excellent.

Bibliographie

- [1] International encyclopedia of the social sciences 2008 kinship.
- [2] Hyvärinen Aapo, Hurri Jarmo, and O. Hoyer Patrik. Natural image statistics. 2009.
- [3] Goyal Aarti and Meenpal Toshani. Eccentricity based kinship verification from facial images in the wild. *Pattern Analysis and Applications*, page 1–26, 2020.
- [4] Hajraoui Abdellatif. *Contribution à la reconnaissance de visages humains*. Université Sultan Moulay Slimane , Mémoire de Fin d’Etudes, 21 Novembre 2015.
- [5] Moujahid Abdelmalik and Dornaika Fadi. A pyramid multi-level face descriptor : application to kinship verification. *Multimedia Tools and Applications*, 2018.
- [6] Y. Adini, Y. Moses, and S. Ullman. Face recognition : the problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7) :721–732, 1997.
- [7] Smith Alvy Ray. Color gamut transform pairs. *ACM SIGGRAPH Computer Graphics*, 12(3) :9–12, 1978.
- [8] Blaustein Andrew R. and O’Hara Richard K. Genetic control for sibling recognition. *Nature*, 290(5803) :246–248, 1981.
- [9] Fatima Bediaf and Nadjat Cherfi. La vérification automatique de parenté basée sur le visage. *Mémoire de Fin d’Etudes, Université Kasdi Merbah Ouargla*, 25 mai 2017.
- [10] Fatima Bediaf and Nadjat Cherfi. *La vérification automatique de parenté basée sur le visage*. Université Kasdi Merbah Ouargla , Mémoire de Fin d’Etudes Master, 25 mai 2017.
- [11] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning : A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8) :1798–1828, 2013.
- [12] Christopher M. Bishop. Pattern recognition and machine learning. *springer*, 2006.
- [13] Jun Bongjin, Kim Taewan, and Kim Daijin. A compact local binary pattern using maximization of mutual information for face analysis. *Pattern Recognition*, 44(3) :532–543, 2011.
- [14] G. Bradski. The opencv library. *Dr. Dobb’s Journal of Software Tools*, 2000.
- [15] Dong Chen, Xudong Cao, Fang Wen, and Jian Sun. Blessing of dimensionality : High-dimensional feature and its efficient compression for face verification. pages 3025–3032, 2013.

- [16] D. Coomans and Massart D.L. Alternative k-nearest neighbour rules in supervised pattern recognition : Part 1. k-nearest neighbour classification by using alternative voting rules. *Analytica Chimica Acta*, 136(3) :15–27, 1982.
- [17] Cortes Corinna and Vapnik Vladimir. *Support-vector networks*, volume 20, page 273–297. Mach Learn, 1995.
- [18] Ashish R. Dandekar and M. S. Nimbarte. Blur insensitive texture classification using local phase quantization. pages 236 – 243, 2008.
- [19] Ashish R. Dandekar and M. S. Nimbarte. A survey : Verification of family relationship from parents and child facial images. pages 1–6, 2014.
- [20] Lieberman Debra, Tooby John, and Cosmides Leda. The architecture of human kin detection. *Nature*, 445(7129) :727–731, 2007.
- [21] Qingyan Duan, Lei Zhang, and Wangmeng Zuo. From face recognition to kinship verification : An adaptation approach. pages 1590–1598, 2017.
- [22] Nour El Houda Bouakal, Messaoud Hettiri, Abdelhakim Chergui, Abdelkrim Ouafi, Azeddine Benlamoudi, and Salah Eddine Bekhouche. Kinship verification system based on the color spaces analysis. pages 1–6, 2022.
- [23] Dornaika Fadi, Arganda-Carreras Ignacio, and O. Serradilla. Transfer learning and feature fusion for kinship verification. *Neural Computing and Applications*, 2020.
- [24] Ruogu Fang, Kevin D. Tang, Noah Snavely, and Tsuhan Chen. Towards computational models of kinship verification. pages 1577–1580, 2010.
- [25] Joblove George H. and Greenberg Donald. Siggraph communications. *Thèse de doctorat, Université Jean Monnet de Saint Etienne*, 1978.
- [26] Yan Haibin. Learning discriminative compact binary face descriptor for kinship verification. *Pattern Recognition Letters*, 117 :146–152, 2019.
- [27] Yan Haibin and Song Chaohui. Multi-scale deep relational reasoning for facial kinship verification. *in Pattern Recognition*, 110 :107541, 2021.
- [28] Yan Haibin and Wang Shiwei. Learning part-aware attention networks for kinship verification. *Pattern Recognition Letters*, 128 :169–175, 2019.
- [29] Ouamane Hanane. Identification de reconnaissance faciale avec des expressions. *Mémoire de Fin d’Etudes, Université Mohamed Khider Biskra*, 07 Juin 2012.
- [30] L. Findlater W. Olive H.Hamilton. E. Gurak. Overview of decision trees. 2001.
- [31] Quinlan J. Ross. C4.5 : Programs for machine learning. *Morgan Kaufmann Publishers*, 1993.
- [32] Wong Jia-Jun and Cho Siu-Yeung. A face emotion tree structure representation with probabilistic recursive neural network modeling. *Neural Computing and Applications*, 19(1) :33–54, 2010.

- [33] Yang Jie, Lu Weier, and Waibel Alex. Skin-color modeling and adaptation. *School of Computer Science Carnegie Mellon University Pittsburgh*, page 687–694, 1997.
- [34] Ye Jieping. Least squares linear discriminant analysis. pages 3025–3032, 2007.
- [35] Wang Jin, Barreto Armando, Wang Lu, Chen Yu, Rishe Naphtali, Andrian Jean, and Adjouadi Malek. Multilinear principal component analysis for face recognition with fewer features. *Neurocomputing*, 73(10) :1550 – 1555, 2010.
- [36] P. Robinson Joseph, Shao Ming, Wu Yue, and Fu Yun. Families in the wild (fiw) : Large-scale kinship image database and benchmarks. *in ACM Multimedia ,2016*, pages 242–246, 2017.
- [37] Robinson Joseph P., Shao Ming, Wu Yue, and Fu Yun. Families in the wild (fiw) : Large-scale kinship image database and benchmarks. *Proceedings of the 24th ACM international conference on Multimedia*, page 242–246, 2016.
- [38] Julie Josse, François Husson, and Jérôme Pagès. Gestion des données manquantes en Analyse en Composantes Principales , journal = Journal de la société française de statistique, year = 2009, number=2, pages=28-5, doi=10.1109/LSP.2016.2603342.
- [39] Kannala Juho and Rahtu Esa. Bsf : binarized statistical image features. *In Proceedings of the 21 st International IEEE Conference on Pattern Recognition (ICPR)*, pages 1363–1366, 2012.
- [40] Zhang Kaihao, Huang Yongzhen, Song Chunfeng, Wu Hong, and Wang Liang. Kinship verification with deep convolutional neural networks. *IEEE Transactions on Image Processing*, 2015.
- [41] Simonyan Karen and Zisserman Andrew. Very deep convolutional networks for large-scale image recognition. 2014.
- [42] Fukunaga Keinosuke. *Introduction to Statistical Pattern Classification*. Academic Press, USA, 1990.
- [43] Naman Kohli, Mayank Vatsa, Richa Singh, Afzel Noore, and Angshul Majumdar. Hierarchical representation learning for kinship verification. *IEEE Transactions on Image Processing*, 26(1) :289–302, 2017.
- [44] Naman Kohli, Mayank Vatsa, Richa Singh, Afzel Noore, and Angshul Majumdar. Hierarchical representation learning for kinship verification. *IEEE Transactions on Image Processing*, 26(1) :289–302, 2017.
- [45] Naman Kohli, Daksha Yadav, Mayank Vatsa, Richa Singh, and Afzel Noore. Supervised mixed norm autoencoder for kinship verification in unconstrained videos. *IEEE Transactions on Image Processing*, 28(3) :1329–1341, 2019.
- [46] Fukushima Kunihiko. Neocognitron : A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(5803) :193–202, 1980.

- [47] Rushi Lan and Yicong Zhou. Quaternion-michelson descriptor for color image classification. *IEEE Transactions on Image Processing*, 25(11) :5281–5292, 2016.
- [48] Rushi Lan, Yicong Zhou, and Yuan Yan Tang. Quaternionic weber local descriptor of color images. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(2) :261–274, 2017.
- [49] Li Lei, Feng Xiaoyi, Wu Xiaoting, Xia Zhaoqiang, and Hadid Abdenour. ” fire recognition using rgb and ycbcr color space ”. *Engineering and Applied Sciences*, 2015.
- [50] Li Lei, Feng Xiaoyi, Wu Xiaoting, Xia Zhaoqiang, and Hadid Abdenour. “kinship verification from faces via similarity metric based convolutional neural network. *in International Conference on Image Analysis and Recognition, 2016*, page 539–548, 2016.
- [51] 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.
- [52] Miguel Bordallo Lopez, Elhocine Boutellaa, and Abdenour Hadid. Comments on the “kinship face in the wild” data sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11) :2342–2344, 2016.
- [53] J. Lu, J. Hu, X. Zhou, Y. Shang, Y.-P. Tan, and G. Wang. Neighborhood repulsed metric learning for kinship verification. In *IEEE International Conference on Computer Vision and Pattern Recognition, 2012*, pages 2594–2601. IEEE, 2012.
- [54] Jiwen Lu, Junlin Hu, and Yap-Peng Tan. Discriminative deep metric learning for face and kinship verification. *IEEE Transactions on Image Processing*, 26(9) :4269–4282, 2017.
- [55] Jiwen Lu, Xiuzhuang Zhou, Yap-Pen Tan, Yuanyuan Shang, and Jie Zhou. Neighborhood repulsed metric learning for kinship verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2) :331–345, 2014.
- [56] Everingham Mark, Gool Luc van, Williams Chris, Winn. John, and Zisserman Andrew. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2) :303–338, June 2010.
- [57] Turk Matthew and Alex Pentland. Face recognition using eigenfaces. pages 586–591, 1991.
- [58] title=Eigenfaces for Recognition year=1991 volume=3 number= pages=72–86 doi=10.1109/CVPR.1991.139758 Matthew,Turk and Pentland, Alex, book-title=Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.
- [59] Bessaoudi Mohcene, Ouamane Abdelmalik, Belahcene Mebarka, Chouchane Ammar, Boutellaa Elhocine, and Bourennane Salah. Multilinear principal component analysis for face recognition with fewer features. *Neurocomputing*, 329(10) :267 – 278, 2019.
- [60] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7) :971–987, 2002.

- [61] Laiadi Oualid, Ouamane Abdelmalik, Boutellaa Elhocine, Benakcha Abdelhamid, Taleb-Ahmed Abdelmalik, and Hadid Abdenour. Kinship verification from face images in discriminative subspaces of color components. *Multimed Tools Appl*, 78 :16465–16487, 2019.
- [62] Hanane Ouamane. *Identification de reconnaissance faciale avec des expressions*. Université Mohamed Khider Biskra , Mémoire de Fin d'Etudes, 07 Juin 2012.
- [63] Phillips P. Jonathon, J. Rauss Patrick, and Z. Der Sandor. Feret recognition algorithm development and test results. *Army Research Laboratory technical report*, 1996.
- [64] P.J. Phillips, Moon Hyeonjoon, S.A. Rizvi, and P.J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10) :1090–1104, 2000.
- [65] P.J. Phillips, Hyeonjoon Moon, P. Rauss, and S.A. Rizvi. The feret evaluation methodology for face-recognition algorithms. pages 137–143, 1997.
- [66] P.J. Phillips, Hyeonjoon Moon, S.A. Rizvi, and P.J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10) :1090–1104, 2000.
- [67] Aliradi Rachid, Belkhir Abdelkader, Ouamane Abdelmalik, and Elmaghraby Adel, Said. Dieda : discriminative information based on exponential discriminant analysis combined with local features representation for face and kinship verification. *Multimedia Tools and Applications*, pages 1–18, 2018.
- [68] N. Ramanathan and Rama Chellappa. Modeling age progression in young faces. 1 :387–394, 2006.
- [69] Joseph Redmon and Ali Farhadi. Yolov3 : An incremental improvement. *arXiv*, 2018.
- [70] O. Duda Richard, E. Hart Peter, and G. Stork David. *Pattern Classification, 2nd Edition*. Wiley, 2000.
- [71] Lior Rokach and Oded Maimon. *Decision Trees*, pages 165–192. Springer US, Boston, MA, 2005.
- [72] Mahpod Shahar and Keller Yosi. Kinship verification using multiview hybrid distance learning. *Computer Vision and Image Understanding*, 167 :28–36, 2018.
- [73] Wang Shiwei and Yan Haibin. Discriminative sampling via deep reinforcement learning for kinship verification. *Pattern Recognition Letters*, 138 :38–43, 2020.
- [74] Dr.E.Ramaraj S.Neelamegam. Classification algorithm in data mining : An overview. *International Journal of P2P Network Trends and Technology (IJPTT)*, V3(5) :1–5, 2013.
- [75] Vieira Tiago, Figueiredo. Identifying kinship cues from facial images. 2013.
- [76] Ojala Timo, Pietikäinen Matti, and Harwood. David. A comparative study of texture measures with classification based on feature distributions. 29, 1996.

- [77] Hastie Trevor, Tibshirani Robert, and Friedman Jerome. The elements of statistical learning : Data mining, inference, and prediction. 2010.
- [78] Lin Tsung-Yi, Maire Michael, Belongie Serge, Bourdev Lubomir, Girshick Ross, Hays James, Perona Pietro, Ramanan Deva, Zitnick C. Lawrence, and Dollár Piotr. Microsoft coco : Common objects in context. *arXiv*, 88, 2014.
- [79] Ojansivu Ville and Heikkila Janne. Blur insensitive texture classification using local phase quantization. page 236 243, 2008.
- [80] Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, pages 34–47, 2001.
- [81] Vezhnevets Vladimir, Sazonov Vassili, and Andreeva Alla. A survey on pixel-based skin color detection techniques. *International Conference on the Computer Graphics and Vision*, page 85 92, 2003.
- [82] Shuyang Wang, Zhengming Ding, and Yun Fu. Cross-generation kinship verification with sparse discriminative metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(11) :2783–2790, 2019.
- [83] Shuyang Wang, Joseph P. Robinson, and Yun Fu. Kinship verification on families in the wild with marginalized denoising metric learning. pages 216–221, 2017.
- [84] Xiaoting Wu, Elhocine Boutellaa, Miguel Bordallo López, Xiaoyi Feng, and Abdenour Hadid. On the usefulness of color for kinship verification from face images. pages 1–6, 2016.
- [85] Siyu Xia, Ming Shao, Jiebo Luo, and Yun Fu. Understanding kin relationships in a photo. *IEEE Transactions on Multimedia*, 14(4) :1046–1056, 2012.
- [86] Qin Xiaoqian, Liu Dakun, and Wang Dong. Heterogeneous similarity learning for more practical kinship verification. *Neural Processing Letters*, 47(3) :1253–1269, 2018.
- [87] Zhou Xiuzhuang, Hu Junlin, Lu Jiwen, Shang Yuanyuan, and Guan Yong. Kinship verification from facial images under uncontrolled conditions. *Proceedings of the 19th ACM international conference on Multimedia, MM '11*, page . 953–956, 2011.
- [88] Zhou Xiuzhuang, Jin Kai, Xu Min, and Guo Guodong. Learning deep compact similarity metric for kinship verification from face images. *Information Fusion*, 48 :84–94, 2019.
- [89] Haibin Yan, Jiwen Lu, and Xiuzhuang Zhou. Prototype-based discriminative feature learning for kinship verification. *IEEE Transactions on Cybernetics*, 45(11) :2535–2545, 2015.
- [90] Li Yong, Zeng Jiabei, Zhang Jie, Dai Anbo, Shan Meina, Kanand Shiguang, and Chen Xinlin. Kinnet : Fine-to-coarse deep metric learning for kinship verification. *Proceedings of the Workshop on Recognizing Families In the Wild*, page 13 – 20, 2017.
- [91] Fu Yun, Cao Liangliang, Guo Guodong, and S. Huang Thomas. Multiple feature fusion by subspace learning. *Proceedings of the 2008 international conference on Content-based image and video retrieval*, page 127–134, 2008.

-
- [92] 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.
- [93] Lei Zhang, Qingyan Duan, David Zhang, Wei Jia, and Xizhao Wang. Advkin : Adversarial convolutional network for kinship verification. *IEEE Transactions on Cybernetics*, 51(12) :5883–5896, 2021.

ReSUMe

L'analyse automatique des visages 'a partir des images a toujours été un sujet d'étude important dans les communautés de la reconnaissance des formes et de l'intelligence artificielle. Les images faciales contiennent de nombreuses informations sur la personne 'a laquelle elles appartiennent, comme l'identité, l'âge, le sexe, l'origine ethnique et l'expression. Pour cette raison, l'analyse des images faciales a de nombreuses applications dans les problèmes du monde reel tels que la reconnaissance des visages, l'estimation démographique, la reconnaissance des expressions faciales et la vérification de la parenté. Nous nous concentrons ici sur la vérification de parenté dont l'objectif de terminer s'il existe une relation de parenté entre deux personnes. Nous proposons un système de vérification de parenté basée sur quatre étape :detection de visage (avec les filtre de haar) ,la conversion d'espace de couleur (HSV) ,l'extraction des caractéristique (LPQ,BSIF,VGG19) ,l'apprentissage en utilisant un réseau convolutif .Notre apprentissage est effectué sur les deux bases de données kinfaceW avec un meilleure résultat de 98,2%.

Mots cles : Detection de visage,extraction; extraction de caractéristique;LPB; BSIF;VGG19;KERAS;CNN.

ABSTRACT

The automatic analysis of faces from images has always been an important topic of study in the pattern recognition and artificial intelligence communities. Facial images contain a lot of information about the person they belong to, such as identity, age, gender, ethnicity and expression. For this reason, facial image analysis has many applications in real-world problems such as face recognition, demographic estimation, facial expression recognition and kinship verification. We focus here on kinship verification whose objective is to determine whether there is a kinship relationship between two people. We propose a kinship verification system based on four steps : face detection (with haar filter), color space conversion (HSV), feature extraction (LPQ, BSIF, VGG19), and learning using a convolutional network. Our learning is performed on the two kinfaceW databases with a best result of 98.2%.

Key words : Face detection;Feature extraction;;LPB;BSIF;VGG19;Haar filters;.