الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

Filière Télécommunication
Spécialité Système de télécommunication

présenté par
MEKLOUT Yanis
&
GHARBI Abdeldjallil

Identification de faces humaines en temps réel par la LBPH

Proposé par:

Dr.NAMANE

Année Universitaire 2019-2020

Remerciements

Au terme de cette étude, nous tenons à présenter nos sincères remerciement au bon dieu de nous avoir accordé la connaissance de la science et nous avoir aidé à réaliser ce projet.

Nous remercions notre directeur de thèse **Dr. NAMANE**, Docteur à l'Université Saad dahleb Blida 1, pour sa grande disponibilité et ses précieux conseils tout au long de la réalisation de ce projet. Nous inspirent une grande admiration et un profond respect, Qu'ALLAH protège votre famille.

Nos remerciements s'adressent aussi à notre enseignant **Monsieur Mamoun** et à tous nos enseignants.

Enfin, on remercie toutes les personnes qui ont contribuées de près ou de loin à la réalisation de ce mémoire

الملخص

الهدف من هذه الدراسة هو عرض تقنيات التعرف المختلفة المستخدمة للكشف عن الوجه والتعرف عليه ، كما يتضمن إنشاء نظام للكشف عن الوجه لشخص أو عدة أشخاص في بيئة حية. لتحقيق ذلك ، قمنا بإعداد مخطط خوارزمية كشف الوجه باستخدام خوارزميات LBPH. قمنا بمقارنة الخوارزميات الأكثر صلة في الأدبيات واخترنا الرسم البياني للنمط الثنائي المحلي لخصائص تصنيفه وتمييزه القوي للوجوه. لبناء مجموعة مصنف قوية. بعد ذلك ، ثم قمنا أيضًا بتطبيق وتتضمن الخوارزمية المقترحة لهذه ، (تعزيز التباين المحلي) CLAHE بعض فلاتر التعزيز لظروف الإضاءة مثل المهمة أربعة مكونات: جمع المعلومات وجمع الوجوه لقواعد البيانات والتعلم والكشف والتنبؤ بالاعتراف. الكلمات ، صورة متكاملة ، مصنفات متتالية ، CLAHE ، LBPH ، الرئيسية: فيولا جونز

كشف الوجه ، Python OpenCV.

Résumé

L'objectif de cette thèse est de présenter les différentes techniques de reconnaissance utilisées pour la détection et la reconnaissance des visages, elle comprend également la création d'un système de détection des visages pour une ou plusieurs personnes dans un environnement vivant.

Pour y parvenir, nous avons préparé un schéma d'algorithme de détection faciale utilisant les algorithmes LBPH. Nous avons comparé les algorithmes les plus pertinents de la science et on a choisi le Local Binary Pattern Histogramme pour ses qualités de classification et sa forte discrimination des visages. Pour construire un ensemble de classificateurs solide. Par la suite, nous avons également appliqué des filtres de boosting pour les conditions d'éclairage comme le CLAHE (Enhance Local Contrast). L'algorithme proposé pour cette tâche comprend quatre composantes:

Collecte d'informations, collecte des visages pour la base de données, apprentissage, détection, reconnaissance et prédiction.

Mots-clés: Viola Jones, LBPH, CLAHE, image intégrale, classificateurs en cascade, détection de visage, Python, OpenCV.

Abstract

The aim of this thesis is to present the different recognition techniques used for face detection and recognition, it also includes the creation of a face detection system for one or several people in a live environment.

To achieve this, we have prepared a facial detection algorithm scheme using the LBPH algorithms. We have compared the most pertinent algorithms in the scientific field and have chosen the Local Binary Pattern Histogramme for its classification qualities and its strong discrimination of the faces. To build a strong classifier set. Subsequently, we also applied some boosting filters for light conditions like the CLAHE (Enhance Local Contrast).The algorithm proposed for this task includes four components:

Information gathering, collecting the faces for the database, learning, detection , recognition and prediction.

Keywords : Viola Jones, LBPH, CLAHE, integral image, cascade classifiers, Face Detection, Python, OpenCV.

Sommaire

Listes des figures

Chapitre 1: Vision artificielle

Chapitre 2: Méthodes de détection faciale.

Chapitre 3: Méthodes de reconnaissance faciale.

Chapitre 4:Réalisation et tests

Introduction générale

La vision artificielle existe depuis plus de 50 ans, mais récemment, nous constatons un regain d'intérêt majeur pour la façon dont les machines " voient " et dont la vision artificielle peut être utilisée pour construire des produits destinés aux consommateurs et aux entreprises. Peu d'exemples de telles applications sont Amazon Go, Google Lens, Autonomous Vehicles, Face Recognition.

Le facteur clé derrière tout cela est la vision artificielle. En termes simples cette dernière est la discipline relevant d'un vaste domaine de l'intelligence artificielle qui enseigne aux machines à voir. Son but est d'extraire le sens des pixels.
Du point de vue de la science biologique, ses objectifs sont de proposer des modèles informatiques du système visuel humain. Du point de vue de l'ingénierie, la vision par ordinateur vise à construire des systèmes autonomes qui pourraient accomplir certaines des tâches que le système visuel humain peut accomplir (et même le dépasser dans de nombreux cas).

Le champ de vision artificielle est assez étendu. Il a des applications de l'industrie à la maison. Cependant, de nombreux processus et techniques sous-jacents sont les mêmes pour tous les domaines d'application. Une vue d'ensemble de ces processus et techniques est donnée. L'acquisition et le traitement d'images numériques sont les premiers sujets puisqu'ils constituent la base d'un traitement de plus haut niveau, comme la reconnaissance de formes, lorsqu'on considère la vision artificielle. L'apprentissage automatique et la reconnaissance des formes sont les deux autres sujets, puisqu'ils sont aussi largement appliqués. En fait, de nombreuses techniques d'apprentissage machinent (Machine Learning) et de reconnaissance de formes,

telles que les réseaux neuronaux et les machines à vecteurs de support (SVM) sont également utilisées dans de nombreux autres domaines que la vision artificielle.

L'augmentation de la capacité informatique et la diminution du rapport prix/performance des systèmes informatiques contribuent au vaste développement des applications informatiques. L'une des priorités est le développement de la capacité de l'ordinateur à imiter l'un des capteurs naturels de l'homme - les yeux. Jusqu'à ce jour, l'homme a déjà réussi à fabriquer un appareil capable d'imiter le fonctionnement des yeux. Et c'est maintenant le moment de faire comprendre à l'ordinateur l'information qu'il contient, dans un domaine d'étude de la vision artificielle. L'un des objectifs de la vision informatique est d'améliorer la capacité du système de surveillance. Le système de surveillance traditionnel dépend fortement de la capacité de l'être humain à percevoir toute information provenant des caméras vidéo. Cependant, grâce aux capacités informatiques avancées d'aujourd'hui, de nombreuses techniques et méthodes ont déjà été déployées pour mettre en place un système de surveillance automatique forcée.

La détection et reconnaissance des visages est l'un des points forts du système de surveillance automatique. Cette focalisation est basée sur la prémisse que le visage d'une personne peut être appliqué dans plusieurs applications pour en extraire des informations utiles. Le système peut rapidement avertir s'il y a des personnes non désirées qui entrent dans une zone (reconnaissance faciale), et beaucoup plus.

Dans ce mémoire, nous présenterons les notions de vision artificielle, de détection faciale et d'identification du visage (reconnaissance). L'objectif global de ce travail est d'approfondir nos recherches sur les algorithmes de détection et de reconnaissance des visages. Nous développerons également un logiciel de reconnaissance des visages en temps réel.

Le mémoire est organisé en quatre chapitres. Le premier chapitre présente une description générale de la vision artificielle, suivi du deuxième chapitre consacré aux méthodes de détection faciale. Le troisième chapitre présente l'étude des méthodes de reconnaissance faciale. Finalement, le quatrième chapitre sera consacré à la partie application et résultats.

Chapitre 1 : Vision artificielle

1.1. Introduction

Dans ce chapitre nous introduirons quelques notions et définitions de base liées à la vision artificielle. Nous donnerons le principe de fonctionnement des systèmes visuels humain ainsi que les systèmes de visualisation de l'ordinateur et la différence des deux.

Actuellement, l'image représente un élément clé dans plusieurs domaines surtout avec la disponibilité des instruments permettant l'acquisition et/ou l'enregistrement des différents types des images avec des qualités très variables dépendant des applications ou de l'utilisation de cette information. L'image n'est plus aujourd'hui un élément dédié seulement aux spécialistes du domaine du traitement du signal et de l'image qui s'intéresse au développement des méthodes et techniques de traitement et analyse de l'image ainsi que les différents outils appliqués à l'image. Par conséquent, on trouve des systèmes de communication (transmission et réception) de l'image, des systèmes de diagnostic basés sur l'image, des systèmes de détection et identification ou de vérification (personnes, maladies, pièces métalliques,…etc.) basés sur les images. Donc, les domaines de télécommunication, l'automatique, l'électronique embarquée, la mécanique, l'électronique biomédicale et l'informatique industrielle nécessitent les connaissances de bases du traitement des images et de la vision pour la bonne compréhension et maîtrise des différents outils utilisés dans la spécialité ainsi que leurs objectifs.

1.2. Définition de la perception visuelle humaine

La vision a suscité l'intérêt de nombreux scientifiques et philosophes depuis déjà très longtemps. Parmi ceux-ci, les neurobiologistes mènent des recherches théoriques et expérimentales afin d'essayer de comprendre l'anatomie et le fonctionnement du cerveau dans son ensemble. Ils ont découvert une structure très complexe qui est loin de leur avoir révèle tous ses secrets. La vision peut être définie comme la fonction par laquelle les images captées par l'œil sont transmises par les voies optiques (cellules rétiniennes et ganglionnaires, nerf optique, chiasma optique) au cerveau [1]. Ces images représentent la quantité de lumière reflétée par les objets qui se trouvent dans la scène.

La lumière réfléchie pénètre dans l'œil par un orifice circulaire situé au centre de l'iris, la pupille. Elle traverse ensuite le cristallin (lentille) qui sert d'objectif et projette sur la rétine une image renversée des objets situés devant l'œil (voir figure 1.1). La lumière est convertie en message bioélectrique par la multitude de cellules photosensibles qui est ensuite véhiculé par le nerf optique vers le cerveau [2].

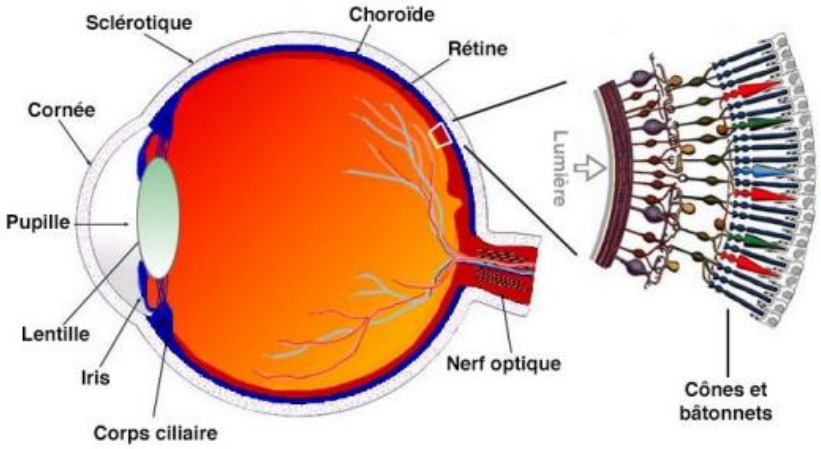


Figure 1.1:Coupe latérale simplifiée de l'œil [1]

1.2.1. Cellules photosensibles

Il existe deux sortes de cellules photosensibles :

- **Les bâtonnets :** sont 25 à 100 fois plus sensibles à la lumière que les cônes. Ils nous permettent de voir dans la pénombre, mais pas de distinguer les couleurs, d'où le dicton "la nuit, tous les chats sont gris" !
- **Les cônes :** moins nombreux et moins sensibles à la lumière, interviennent dans la vision des couleurs et la netteté. Il existe trois types de cônes qui diffèrent par la radiation qu'ils détectent : de courtes, de moyennes et de grandes longueurs d'onde. La gamme de couleurs que nous percevons a pour origine l'ensemble de ces trois réponses fondamentales [2].

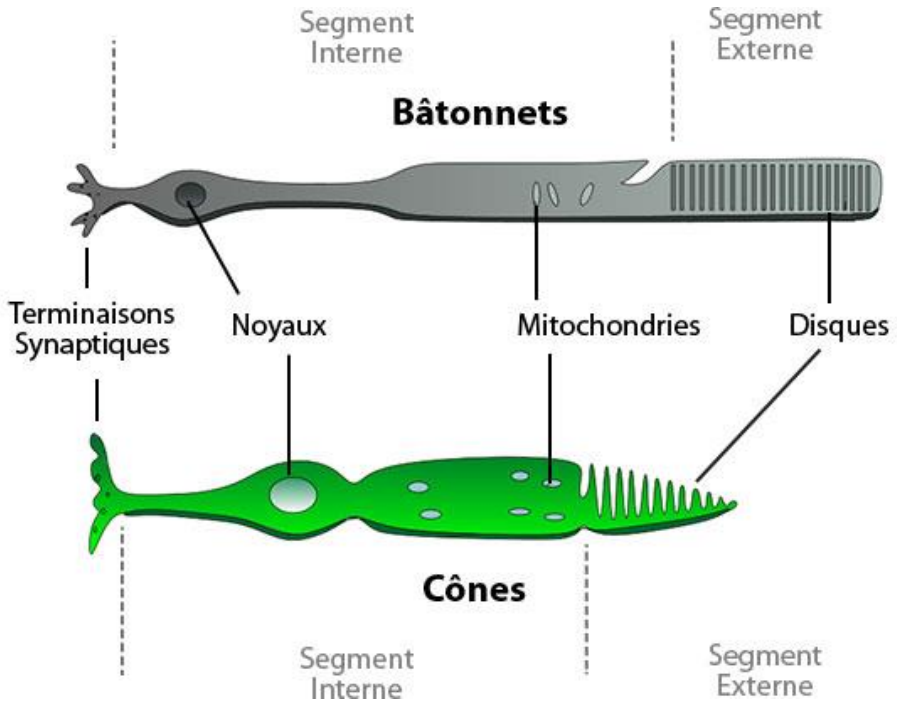


Figure 1.2: les bâtonnets et les cônes [2]

1.2.2. Lumière

La lumière nous est familière. Celle du Soleil nous apporte de la chaleur et contribue de façon essentielle à la perception de notre entourage grâce au sens de

la vue qu'elle nous a amené à développer. Mais quelle est sa nature ? Comment l'interpréter à l'aide de concepts déjà connus ? Est-ce un flot de minuscules particules, chacune transportant un peu d'énergie ? Est-ce une onde qui se propage dans l'espace comme la houle à la surface de l'océan ? Dans l'hypothèse d'une onde, quel est le milieu qui assure sa propagation, jouant ainsi le rôle de l'air pour les ondes sonores ? Lumière, onde, photons, par quelque mot que l'on cherche à la qualifier, elle n'est jamais complète. Explications.

Des siècles d'observations et de développements théoriques, marqués par des expériences célèbres et de grandes controverses ont apporté les réponses à ces questions : la lumière est à la fois une onde et un flux de particules, et sa propagation n'implique aucun milieu spécifique ; elle peut même se propager dans ce que nous appelons couramment 'le vide' – qui n'est alors plus vide, évidemment, puisque la présence du rayonnement lui confère énergie et quantité de mouvement, comme on sait le faire pour les objets macroscopiques [3].

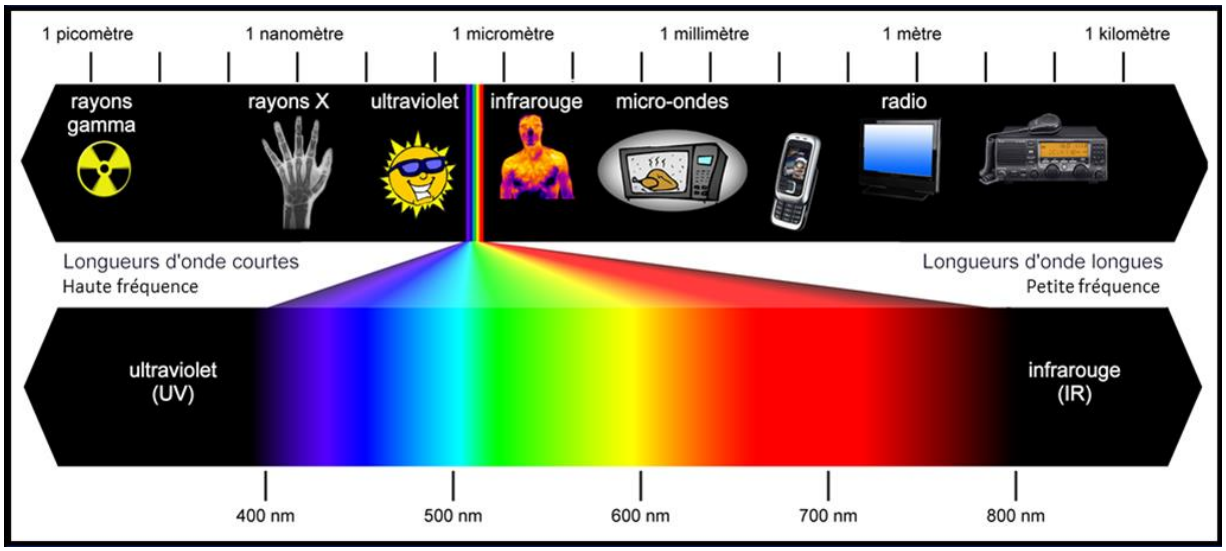


Figure 1.3: Le spectre du visible [3]

La lumière visible correspond aux ondes visibles comprises entre 375 nm et 740 nm (voir la figure 1.3). A chaque couleur perçue par l'œil humain correspond une longueur d'onde (λ en mètre) ou fréquence (f en Hertz). Lorsque les rayons lumineux se réfléchissent sur un objet et pénètrent dans les yeux par la cornée (le revêtement extérieur transparent de l'œil), vous pouvez alors voir cet objet.

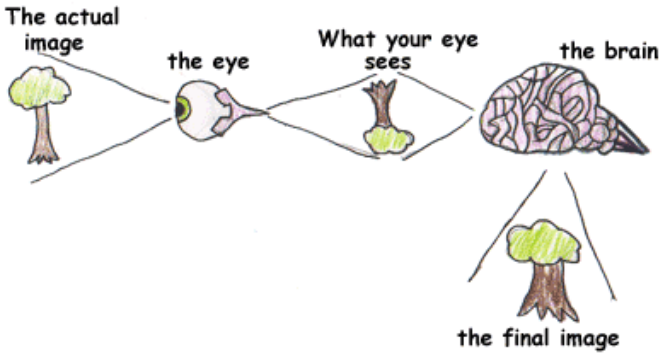


Figure 1.4: Perception humaine [3]

1.3. Vision par ordinateur

La vision par ordinateur est une branche de l'intelligence artificielle dont le principal but est de permettre à une machine d'analyser, traiter et comprendre une ou plusieurs images prises par un système d'acquisition. Il ne s'agit pas de fournir une explication de comment marche la vision biologique mais de créer un modèle qui, vu de l'extérieur, possède des propriétés semblables afin d'automatiser les taches de contrôle, de gestion et d'aide à la décision par le traitement des images [4].

1.3.1. Acquisition d'image

Une image numérique est produite par un ou plusieurs capteurs d'image, avec divers types de caméras photosensibles, notamment des capteurs de distance, une tomographie appareils, radar, caméras ultra-soniques, etc.

Selon le type de capteur, les données d'image résultantes sont une image 2D ordinaire, un volume 3D ou une séquence d'images. Les valeurs des pixels correspondent généralement à la lumière dans une ou plusieurs bandes spectrales (images grises ou images couleur), mais peut également être lié à diverses mesures physiques, telles que la profondeur, l'absorption ou réflectance des ondes sonores ou électromagnétiques, ou résonance magnétique nucléaire.

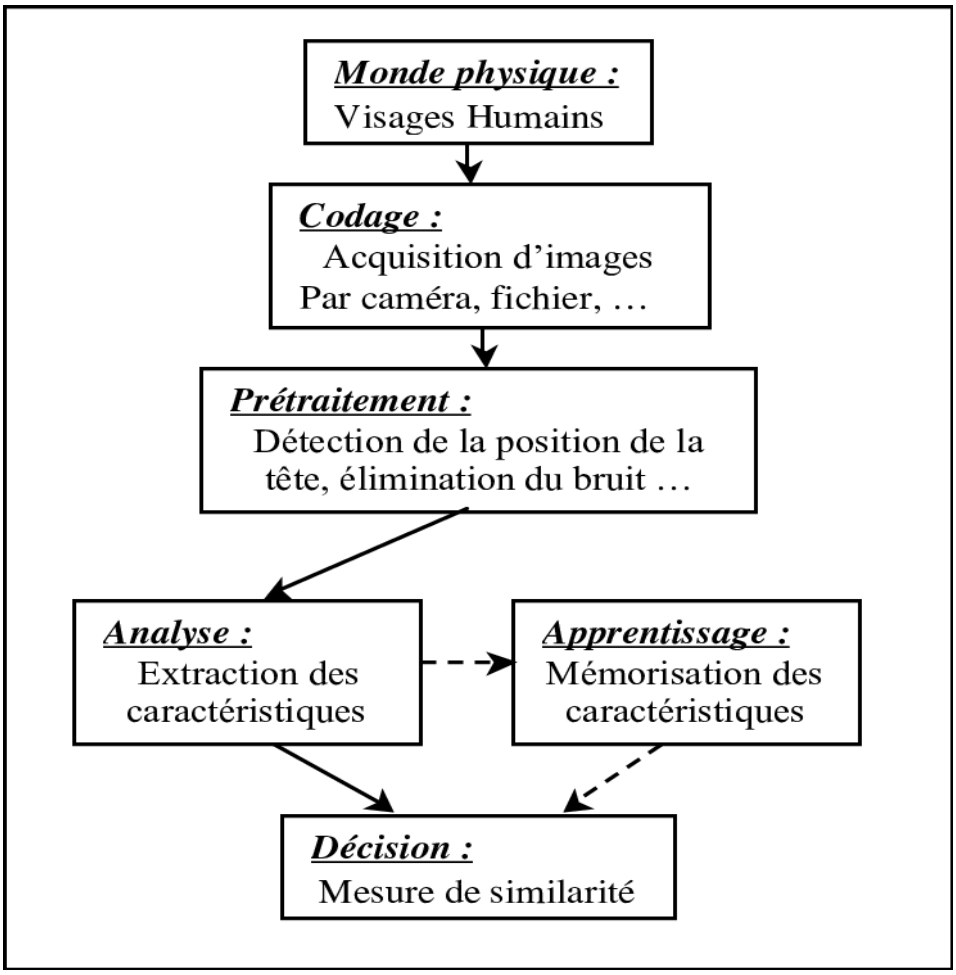


Figure 1.5: Schéma Acquisition d'image [4]

1.3.2. Prétraitement

Échantillonnage afin de s'assurer que le système de coordonnées de l'image est correct.
Réduction du bruit afin de s'assurer que le bruit du capteur n'introduit pas de fausse information.
Amélioration du contraste pour garantir la détection des informations pertinentes.
Représentation à l'échelle de l'espace pour améliorer les structures d'image au niveau local échelles appropriées

1.3.3. Extraction des caractéristiques

Les caractéristiques sont des caractéristiques distincts caractères / mot qui nous permet de le reconnaître. Ce sont de nombreux types de fonctionnalités, mais ils sont

largement classés en deux types: structurel et statistique. Les caractéristiques structurelles représentent la structure ou forme, comme les boucles, les points de branchement, les points d'extrémité, les points, etc.

Les caractéristiques statistiques sont des mesures numériques d'intensités de pixels calculées sur les images. Ils incluent densités de pixels, directions, moments, Fourier descripteurs, etc.

À un certain moment du traitement, une décision est prise sur les points ou régions d'image qui sont pertinents pour un traitement ultérieur. En voici des exemples:

- Sélection d'un ensemble spécifique de points d'intérêt
- Segmentation d'une ou de plusieurs zones d'image contenant un objet d'intérêt spécifique.

1.3.4. Analyse

- Vérification que les données satisfont aux hypothèses basées sur le modèle et spécifiques à l'application.
- Estimation des paramètres spécifiques à l'application, tels que la pose ou la taille de l'objet.
- Reconnaissance d'images: classer un objet détecté en différentes catégories.
- Enregistrement d'images: comparer et combiner deux vues différentes du même objet.

1.3.5. Décision

Cette étape concerne la réussite / échec sur les applications d'inspection automatique, correspondance / non-correspondance dans les applications de reconnaissance, signaler un examen humain supplémentaire dans les domaines médical, militaire, de la sécurité et de la reconnaissance applications.

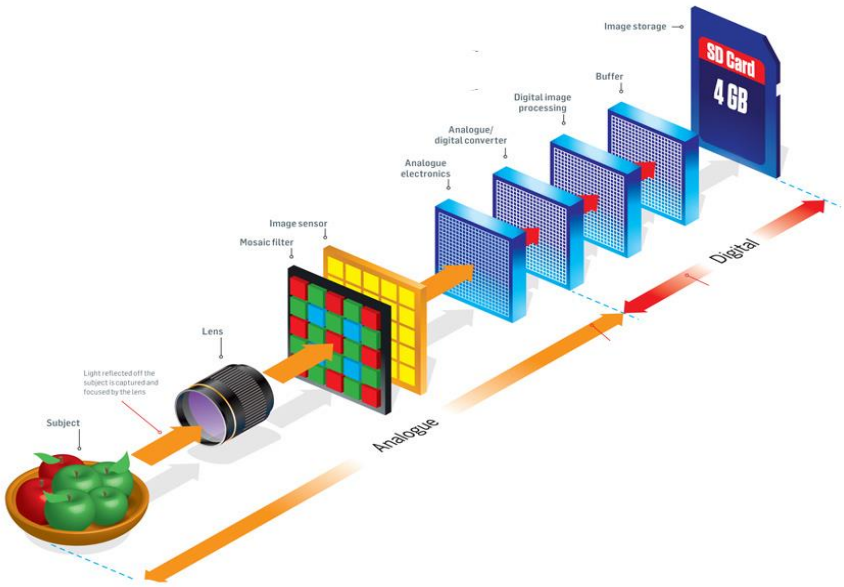


Figure 1.6: Acquisition d'image numérique [4]

1.4. Synthèse des couleurs

La synthèse additive des couleurs est la méthode de création de couleur en mélangeant diverses proportions de deux ou trois couleurs de stimulus distinctes de la lumière. Ces couleurs primaires sont généralement rouges, vertes et bleues, mais elles peuvent avoir n'importe quelle longueur d'onde pour stimuler des récepteurs distincts sur la rétine de l'œil. Les caractéristiques distinctives de la synthèse additive des couleurs sont qu'elle traite des effets de couleur de la lumière plutôt que des pigments, des colorants ou des filtres, et que les stimuli proviennent de sources monochromatiques distinctes. L'écran de télévision couleur (ou moniteur RVB), qui est une mosaïque de points de phosphore rouge, vert et bleu, est l'exemple le plus courant de synthèse additive des couleurs; à des distances d'observation normales, l'œil ne distingue pas les points, mais mélange ou ajoute leurs effets de stimulus pour obtenir un effet de couleur composite. Il s'agit d'un exemple agrandi de synthèse additive des couleurs à partir d'une source de type RVB [5].

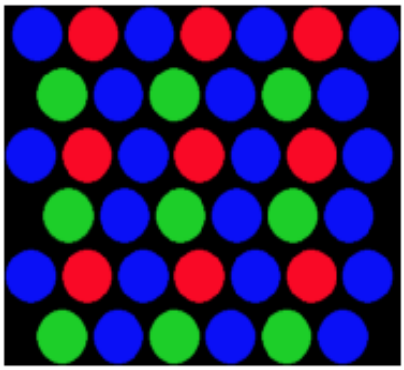


Figure 1.7: synthèse additive [5]

1.4.1. Synthèse additive

La synthèse additive consiste à combiner les lumières de plusieurs sources colorées afin d'obtenir une lumière colorée quelconque dans une gamme déterminée.

La synthèse additive est utilisée par nos écrans et est constituée des trois lumières de base, les couleurs primaires, qui sont le **rouge,** le **vert** et le **bleu**. En mélangeant les couleurs primaires, nous obtenons les couleurs secondaires. Les couleurs secondaires sont plus claires que les primaires, elles absorberont moins de lumière, c'est pour cela que leur synthèse est appelée **additive** [5].

1.4.2. Synthèse soustractive

La synthèse soustractive est utilisée dans l'imprimerie, les couleurs primaires sont le **cyan**, le **magenta**, le **jaune** et le **noir,** la superposition de l'encre cyan et de l'encre jaune donne une nouvelle couleur, le vert. On remarque également dans ce mélange qu'on soustrait au papier les luminosités du cyan et du jaune, ce qui donne une couleur plus foncée que les primaires. La couleur la plus foncée du système est le mélange de toutes les encres primaires. Dans la synthèse soustractive tout est inversé par rapport au système additif. La source lumineuse est le blanc du papier.

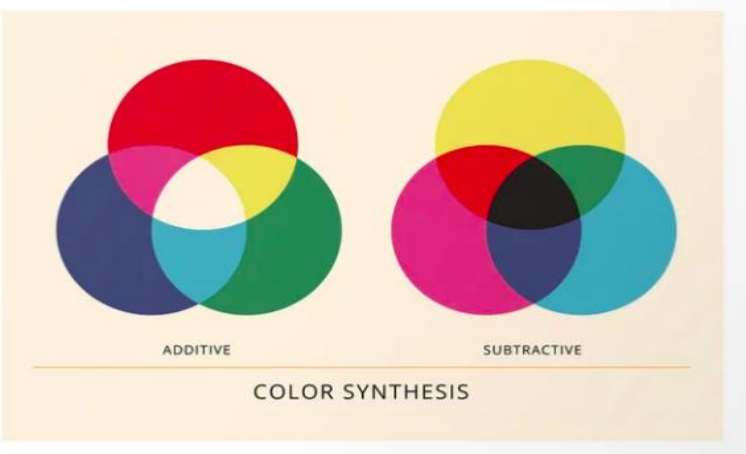


Figure 1.8: Synthèse couleur (Additive, Soustractive) [5]

1.5. Les espaces couleurs

Une gamme de couleurs peut être créée par les couleurs primaires du pigment et ces couleurs définissent alors un espace colorimétrique spécifique. L'espace colorimétrique, également connu sous le nom de modèle de couleur (ou système de couleurs), est un modèle mathématique abstrait qui décrit simplement la gamme de couleurs sous forme de tuples de nombres, généralement sous forme de 3 ou 4 valeurs ou composants de couleur (par exemple RVB). Fondamentalement, l'espace colorimétrique est une élaboration du système de coordonnées et du sous-espace. Chaque couleur du système est représentée par un seul point [5].

1.5.1. Espace RVB

RVB (R = rouge, V = vert, B = bleu) est une sorte d'espace colorimétrique qui utilise le rouge, le vert et le bleu pour élaborer un modèle de couleur. Un espace colorimétrique RVB peut être simplement interprété comme «toutes les couleurs possibles» qui peuvent être constituées de trois couleurs pour le rouge, le vert et le bleu. Dans une telle conception, chaque pixel d'une image se voit attribuer une plage de 0 à 255 valeurs d'intensité des composants RVB. C'est-à-dire qu'en utilisant uniquement ces trois couleurs, il peut y avoir 16 777 216 couleurs à l'écran selon différents rapports de mélange.

De nos jours, la majorité des moniteurs adoptent un espace colorimétrique RVB car il s'agit d'un modèle de couleur pratique pour l'infographie et le système

visuel humain fonctionne de manière similaire. Comme nous le savons, les espaces colorimétriques RVB les plus populaires sont actuellement SRVB et Adobe RVB [5].

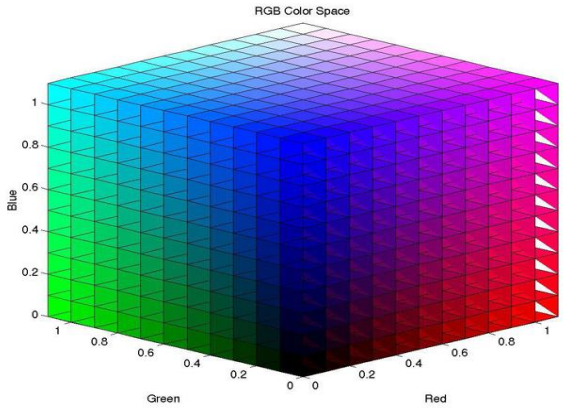


Figure 1.9: L'espace couleur RGB [5]

1.5.2. Espace TSL

Teinte, saturation, luminosité (ou valeur) sont les trois paramètres de description d'une couleur dans une approche psychologique de cette perception. Cette expression désigne des modèles de description des couleurs utilisés en graphisme informatique et en infographie, qui adaptent ces paramètres.

Les sigles TSL ou, en anglais HSL ((en) Hue, Saturation, Lightness) désignent de tels systèmes de description des couleurs. La luminosité étant à peu près ce qu'on appelle la valeur ((en) Value), TSV ou HSV s'utilisent pour différencier des systèmes basés sur les mêmes principes, avec une mise en œuvre différente [6].

Figure 1.10: L'espace TSL [6]

1.5.3. Espace CMJN/CMYN

L'espace CMJN (Cyan Magenta Jaune Noir) / CMYK (Cyan Magenta Yellow Black) est basé sur la synthèse soustractive des couleurs. Cette représentation est principalement utilisée pour l'imprimerie et pour la conception sur ordinateur de textes et illustration devant être imprimés. Pour chaque couleur, on indique la quantité d'encre Cyan, Magenta, Jaune et Noir permettant de la reproduire [6]. On peut simplement passer de l'espace RGB à l'espace CMYK :

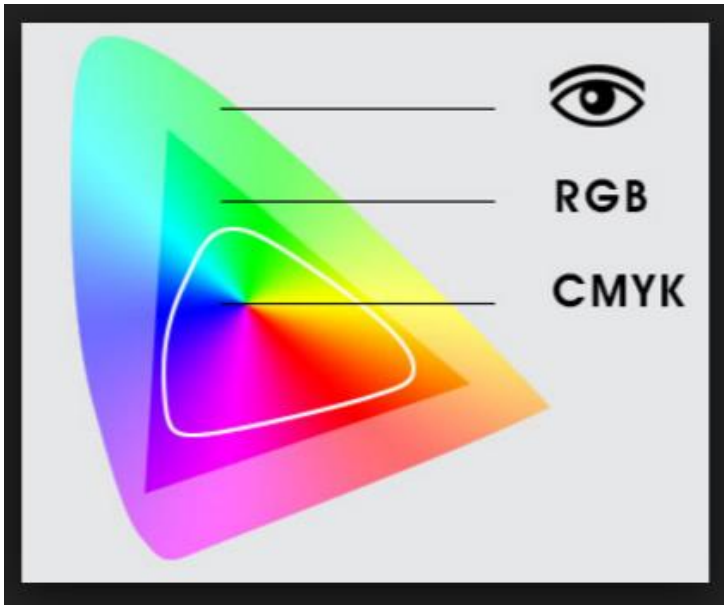


Figure 1.11: L'espace CMYK [6]

1.6. Image

1.6.1. Représentation

Une image numérique est représentée par un tableau ou matrice I de N lignes et M colonnes. Le pixel est un point de l'image désigné par un couple de coordonnées (i,j) où i est l'indice de ligne et j l'indice de colonne. N et M sont respectivement la largeur et la hauteur de l'image I. Par convention, le pixel origine (0, 0) est en général en haut à gauche. Le nombre I (i, j) est la valeur du pixel (i, j) ou

le niveau de gris avec I (i, j) ∈ {Gmin, Gmax}, Gmin et Gmax représentent les valeurs minimale et maximale respectivement. On appelle dynamique de l'image le logarithme en base 2 de Nmax =Gmax-Gmin (nombre de niveaux de gris dans l'image), c.-à-d. le nombre de bits utilisés pour coder l'ensemble des valeurs possibles.

D = log2(Nmax) et Nmax = 2^D

On appelle D dynamique de l'image le logarithme en base 2 de Nmax, c.-à-d. le nombre de bits utilisés pour coder l'ensemble des valeurs possibles.

1.6.2. Échantillonnage

Une image numérique est associée à un pavage de l'espace, en général rectangulaire. Chaque élément du pavage, appelé pixel, est désigné par ses coordonnées entières. Image numérique.

L'échantillonnage est le procédé de discrétisation spatiale d'une image réelle consistant à associer à chaque pixel une unique valeur I(i,j). On parle de sous échantillonnage lorsque l'image est déjà discrétisée et qu'on veut diminuer le nombre de pixels. La quantification désigne la discrétisation tonale correspondant à la limitation du nombre de valeurs différentes que peut prendre chaque pixel. Une image numérique est donc une image échantillonnée et quantifiée [7].

1.6.3. Quantification

La couleur d'un pixel sur un écran est obtenue par une combinaison de trois rayons rouge, vert et bleu d'intensités variables. Ainsi la couleur est codée par 3 nombres correspondant à ces intensités, et généralement ces nombres seront des entiers entre 0 et 255, ce qui permet de coder la couleur avec 3 octets (1 octet pour chacune des composantes rouge, verte, et bleue). Pour une image à niveaux de gris, les rayons rouge, vert et bleu auront une intensité égale, aussi on ne code qu'un

nombre, correspondant à la luminance ; on utilisera donc un octet pour coder celle-ci, puisqu'elle correspondra à un entier entre 0 et 255.

Si pour coder l'intensité (luminance, ou composantes rouge / verte /bleue) on avait utilisé, au lieu d'un octet, un nombre de bits $m < 8$, on aurait 2^m valeurs différentes d'intensité, au lieu de 256. Donc on verrait moins de nuances de couleurs ou de niveaux de gris. Combien de niveaux d'intensité sont nécessaires ? Nous montrons ci-dessous une image à 256 niveaux de gris, affichée successivement avec 256, 128, 64, 32, 16, 8, 4 et 2 niveaux de gris, c.-à-d. 2^m pour m descendant de 8 à 1. En fait, l'image à 2^m valeurs de niveaux de gris a ses niveaux de gris affichés dans l'intervalle de 0 à 255, mais ceux-ci varient par pas de 255 / (2^m - 1), ce qui fait 2^m valeurs allant de 0 à 255 [7].

Figure 1.12: Echantillonnage et quantification [7]

1.6.4. Résolution

La résolution d'une image est le nombre de pixels par pouce qu'elle contient (1 pouce = 2.54 centimètres). Elle est exprimée en "PPP" (points par pouce) ou DPI (dots per inch). Plus il y a de pixels (ou points) par pouce et plus il y aura d'information dans l'image (plus précise). Par exemple, une résolution de 300dpi signifie que l'image comporte 300 pixels par pouce. Grâce à cette formule, il est facile de connaître la dimension maximale d'un tirage.
Il est généralement admis qu'une résolution de 300 ppp pour une image est

largement suffisante avant impression. Cette résolution peut être revue à la baisse dans le cas d'impressions devant être visualisées à une distance plus ou moins éloignée de l'observateur (par conséquent liée au pouvoir séparateur de l'œil humain) [7].

Figure 1.13: Résolution [7]

1.6.5. Formats d'images

Choisir son format d'image est une étape déterminante pour la vision artificielle. Un format se choisit selon ses besoins en fonction de la qualité d'image voulue, des fonctionnalités et de la lisibilité sur un maximum de logiciels, que ce soit des ordinateurs PC, Mac ou des appareils nomades comme les Smartphones [7]

Images vectorielles :

extension	Application	utilisation
WMF	format vectoriel Windows	Cliparts
CDR	Corel Draw	dessin en 2D
DWG	Autocad	professionnel (architecture, conception mécanique …)
…		

Images Bitmap :

extension	nombre de couleurs	Compression	Commentaires
BMP	16 M	Non	format standard Windows
JPG	16M	Oui	format courant sur Internet
Gif	256	Oui	permet les animations (Gif animés) ainsi que le mode transparence. Très utilisé sur Internet modèle de compression déposé ©
ICO	16 ou 256	Non	format des icônes sous Windows
Tiff	16M	Oui	(Tagged image format) utilisé en gestion de document, supporte différents formats.
PCX	16M	Non	ancien format (Paintbrush)
PNG	16M	Oui	concurrent libre du Gif
TGA	16M	Oui / Non	
…			

Figure 1.14:Tableaux des formats d'images [7]

1.6.6. Histogrammes

L'histogramme h(x) d'une image représente la distribution des intensités des pixels. Un histogramme est une fonction qui donne, pour chaque intensité lumineuse, le nombre de pixels ayants cette valeur. L'abscisse d'un histogramme h(x) représente les niveaux d'intensité allant du plus foncé à gauche au plus clair à droite. La dynamique de l'image peut être définie par l'intervalle D = [Nmin, Nmax] ou Nmin et Nmax sont respectivement les niveaux de gris minimal et maximal présents dans l'image. La dynamique maximale est [0,255] [7].

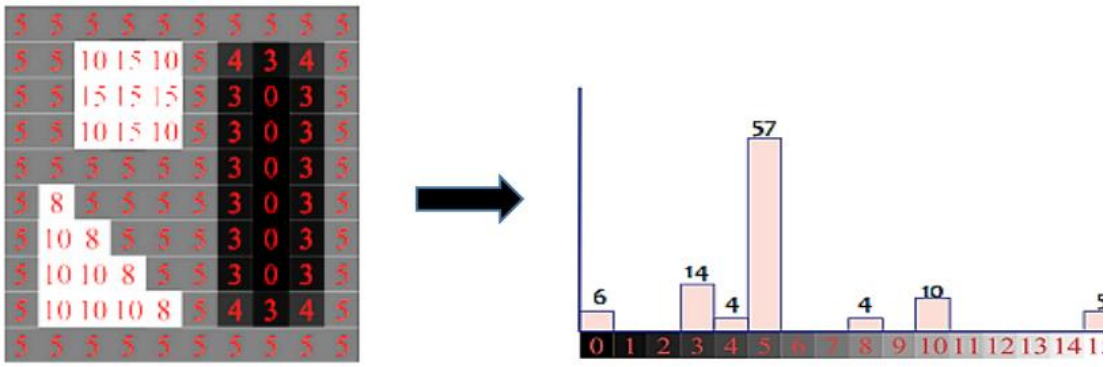


Figure 1.15: L'histogramme d'une image [7]

1.7 Conclusion

Dans ce chapitre, nous avons discuté en profondeur de la signification de la vision artificielle, et quelle est la différence entre la vision humaine et l'ordinateur. Ce domaine de la science est aujourd'hui considéré comme l'un des plus innovants et futuristes, c'est pourquoi nous avons essayé de passer en revue la complexité d'un système de vision par ordinateur pour donner un aperçu des chapitres à venir où nous décortiquerons plus en détail les méthodologies utilisées pour la reconnaissance faciale. Il était également important de discuter des théories des couleurs et des espaces colorimétriques comme RGB et TSL, sans oublier de mentionner comment une image peut être acquise et analysée dans de tels systèmes.

Chapitre 2: Les Méthodes de détection faciale.

2.1. Introduction

Au cours des dernières années, la reconnaissance faciale a occupé une place importante et a été considérée comme l'une des applications les plus prometteuses dans le domaine de l'analyse d'images. La détection des visages peut prendre en compte une partie substantielle des opérations de reconnaissance faciale. Selon sa force à concentrer les ressources informatiques sur la section d'une image tenant un visage. La méthode de détection des visages sur les images est compliquée en raison de la variabilité présente sur les visages humains. Les défis de la détection des visages sont les raisons qui réduisent la précision et le taux de détection de visage. Ces défis sont complexes.

2.2. Défis de la détection des visages

Les défis de la détection de visages peuvent être résumés dans les points suivants :

• Occlusion faciale : l'occlusion faciale cache le visage de tout objet. Il peut s'agir de lunettes, foulard, main, cheveux, chapeaux et tout autre objet. Elle réduit également le taux de détection des visages [8].

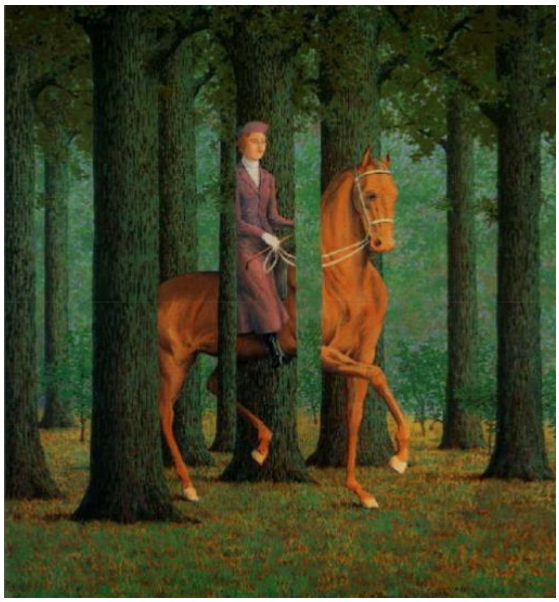


Figure 2.1: Effet occlusion [8]

• Illuminations : les effets d'éclairage peuvent ne pas être uniformes sur l'image. Une partie de l'image peut avoir un éclairage très élevé et d'autres peuvent avoir un éclairage très faible [8].

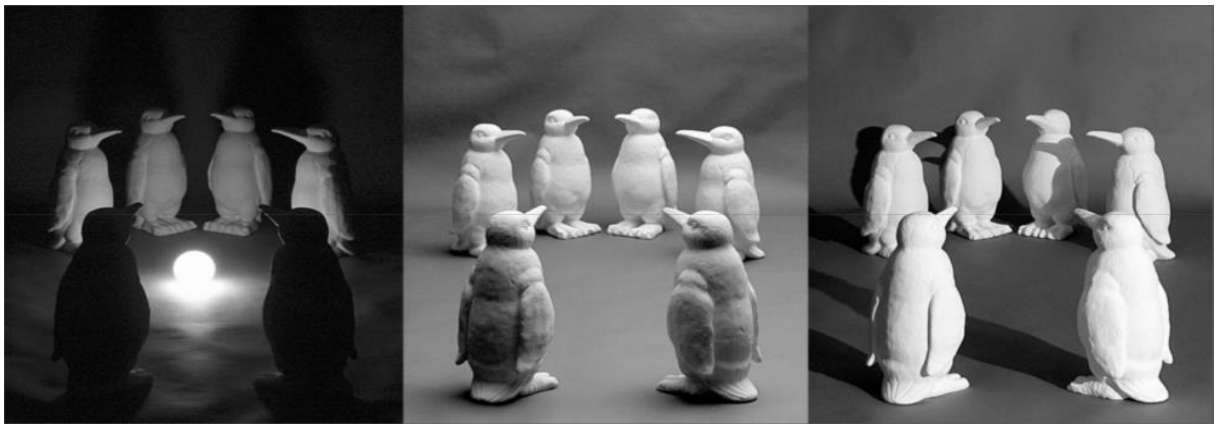


Figure 2.2: Effet de l'illumination [8]

• Arrière-plan complexe : un arrière-plan complexe signifie que beaucoup d'objets sont présents dans l'image.

Figure 2.3: Arrière-plan complexe [8]

• Trop de visages dans l'image : cela signifie que l'image contient trop de visages humains, ce qui représente un défi pour la détection des visages.
• Moins de résolution : la résolution de l'image peut être très mauvaise [8].

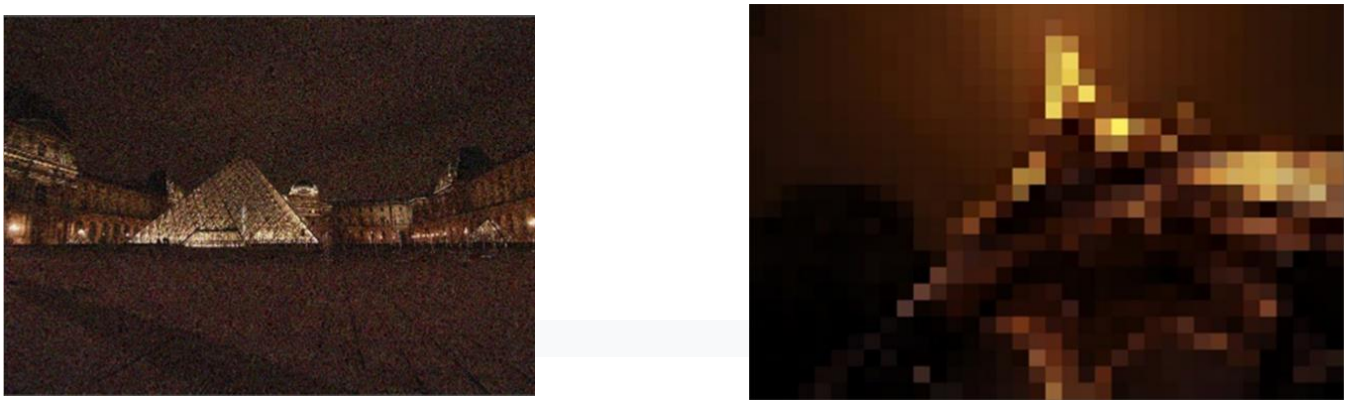


Figure 2.4: Mauvaise résolution [8]

- Couleur de la peau : la couleur de la peau change en fonction des emplacements géographiques. La couleur de peau du chinois est différente de l'africain et la couleur de la peau de l'africain est différente de l'américaine, etc.

David J. Kriegman, NarendraAhuja et Yan Yang [7] ont présenté une classification des méthodes de détection des visages. Ces méthodes sont divisées

en quatre catégories et les algorithmes de détection des visages peuvent appartenir à deux ou plusieurs groupes. Ces catégories sont les suivantes:

1. **Correspondance de modèle (***Template Matching***).**
2. Basé sur les connaissances (*Knowledge-Based)*
3. Basé sur l'apparence (**Appearance-Based**).
4. Basé sur les caractéristiques (*Feature-Based)* [9].

2.3. Correspondance de modèle (*Template Matching*)

Une tâche très courante dans la reconnaissance de modèle est la correspondance de modèle. La détection et la reconnaissance des objets dans les images est un sujet clé de recherche en vision artificielle. Dans ce domaine, la reconnaissance d'image et l'interprétation a attiré de plus en plus attention en raison de la possibilité de dévoiler les mécanismes de perception humaine, et pour le développement de systèmes biométriques pratiques. Diverses méthodes de correspondance de modèles comme NTM, GBM, EBM sont appliqué dans des domaines polyvalent comme le traitement d'image reconnaissance de formes [12]**.**

La correspondance de modèles est une technique de haut niveau qui permet d'identifier les parties d'une image (ou plusieurs images) qui correspondent aux donnée motif d'image. Elle peut être utilisée dans la fabrication comme partie du contrôle qualité, une façon de naviguer sur un robot mobile ou comme un moyen de détecter les bords dans les images [12].

2.3.1. Navie Template Matching(NTM)

Supposons que nous avons une image d'une prise de vue et notre objectif est de rechercher ses broches. Nous avons l'image motif qui représente l'objet de référence que nous recherchons et donc l'image d'entrée à utiliser qui positionne le motif au-dessus de l'image à chaque emplacement réalisable, et à chaque instant, on va donc calculer une mesure numérique de similitude au milieu du motif et donc la

section d'image. Enfin, nous avons tendance à déterminer les emplacements qui donnent les mesures de similitude les plus efficaces que possible [12]**.**

Template image *Input image*

Figure 2.5: Image du motif choisie pour être appariée avec l'image d'entrée [12]

Le problème qui se produit dans la NTM est dans le calcul de la mesure de similitude de l'image du motif aligné et donc la section chevauchée de l'image d'entrée. Ce calcul est égal à une mesure de similitude de 2 figures de mêmes dimensions. Ainsi, la mesure numérique de la similitude de l'image est connue sous le nom de corrélation d'image [12]**.**

2.3.2. **Corrélation croisée normalisée**

La corrélation croisée normalisée est une version améliorée de la méthode de corrélation croisée classique qui introduit deux améliorations par rapport à l'original:

Les résultats sont invariables aux changements globaux de luminosité, c'est-à-dire qu'un éclaircissement ou un assombrissement constant de l'une ou l'autre image n'a aucun effet sur le résultat (ceci est accompli en soustrayant la luminosité moyenne de l'image de chaque valeur de pixel).

La valeur de corrélation finale est mise à l'échelle dans la plage [-1, 1], de sorte que le NCC de deux images identiques est égal à 1, tandis que le NCC d'une image et sa négation est égal a -1. La formule utilisée est la suivante :

$$\mathrm{NCC}(\mathrm{Image1}, \mathrm{Image2}) = \frac{1}{N\sigma_1\sigma_2} \sum_{x,y} (\mathrm{Image1}(x,y) - \overline{\mathrm{Image1}}) \times (\mathrm{Image2}(x,y) - \overline{\mathrm{Image2}})$$

Dans cette formule, nous avons pris deux images Image1 et Image2 et leurs coordonnées en pixels u, v et σ est une constante. La corrélation croisée normalisée peut être généralement appliqué comme une mesure de ressemblance efficace signifiait pour faire correspondre les applications [10].

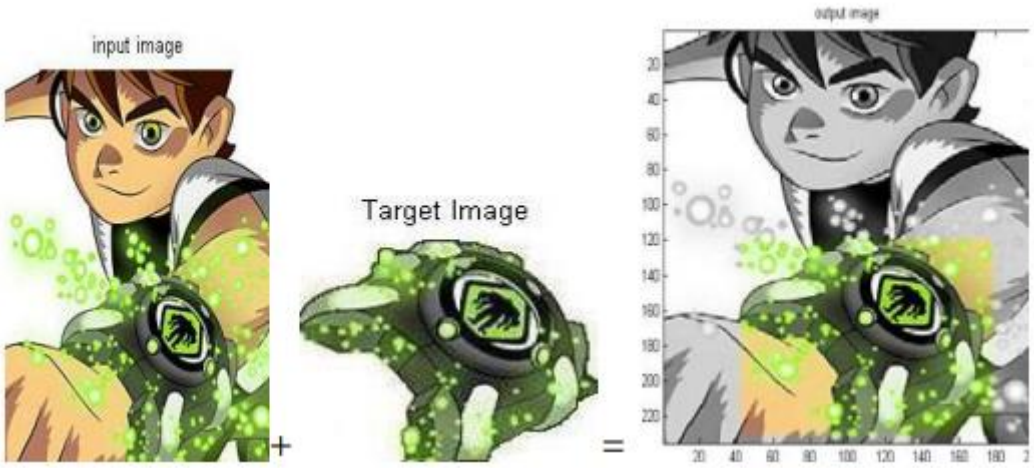


Figure 2.6: Exemple de corrélation croisée normalisée [10]

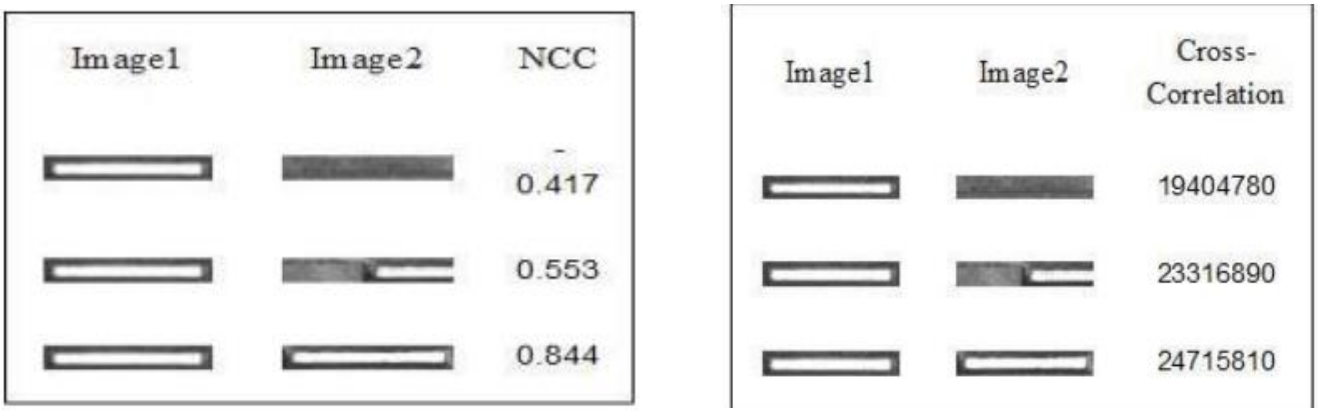


Figure 2.7: Comparaison entre NCC et corrélation croisée [10]

2.3.3. Correspondance basée sur les niveaux de gris (GBM)

Bien que dans certaines applications, l'orientation des objets soit uniforme et fixe (comme nous l'avons vu dans l'exemple de prise), il arrive souvent que les objets à détecter apparaissent en rotation. Dans les algorithmes de correspondance de modèles, la recherche pyramidale classique est adaptée pour permettre la correspondance multi-angles, c'est-à-dire l'identification des instances pivotées du modèle [11].

Ceci est réalisé en calculant non seulement une pyramide d'image de modèle, mais un ensemble de pyramides - une pour chaque rotation possible du modèle. Pendant la recherche pyramidale sur l'image d'entrée, l'algorithme identifie les paires (position du modèle, orientation du modèle) plutôt que les positions uniques du modèle. De manière similaire au schéma d'origine, à chaque niveau de la recherche, l'algorithme ne vérifie que les paires (position, orientation) qui ont obtenu de bons résultats au niveau précédent (c'est-à-dire semblaient correspondre au modèle dans l'image de résolution inférieure) [11].

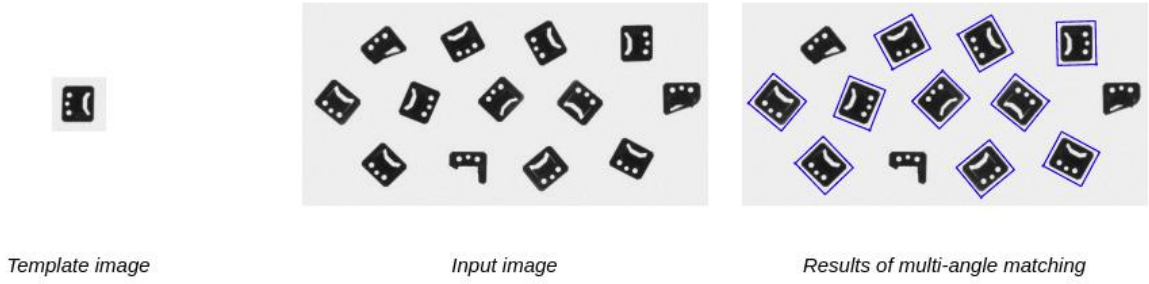


Figure 2.8: Correspondance en niveaux de gris [11]

2.3.4 Pyramide d'image

La pyramide d'images est une série d'images, chaque image étant le résultat d'un sous-échantillonnage (réduction de l'échelle, par le facteur deux dans ce cas) de l'élément précédent.

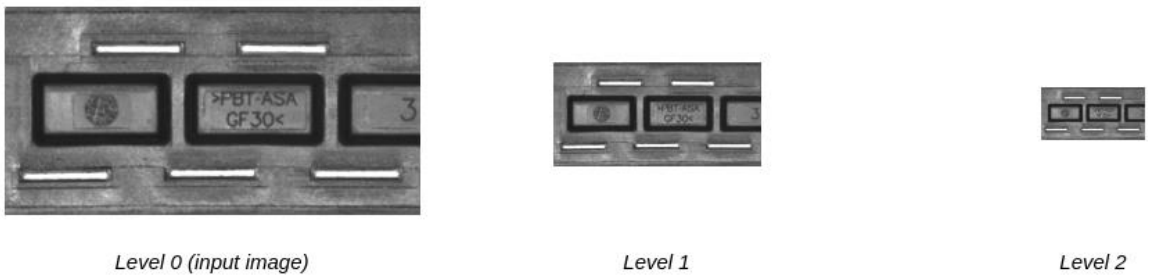


Figure 2.9: Pyramide d'image [11]

2.3.5 Correspondance basée sur les bords (EBM)

La correspondance basée sur les bords améliore encore beaucoup cette méthodologie en décalant le calcul vers les zones de bord de l'objet, elle met à jour la correspondance basée sur les niveaux de gris précédemment expliquée via l'utilisation d'une déclaration vitale - que le formulaire de tout objet est délimitée uniquement par la forme de ses bords. Donc plutôt que de faire correspondre le motif entier, nous avons tendance à retirer son bord et ensuite correspondre uniquement à la proximité par pixels, donc à l'écart de certains calculs superflus. Dans ces applications, l'accélération acquise est généralement important.

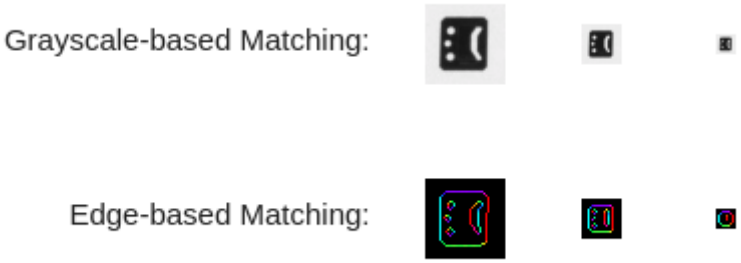


Figure 2.10: Différence entre GBM ET EBM

2.3.6 Conclusion

Pour la grande majorité des applications, la méthode de correspondance basée sur les bords sera à la fois plus robuste et plus efficace que la correspondance basée sur les niveaux de gris. Ce dernier ne doit être pris en compte que si le modèle considéré a des zones de transition de couleur lisses qui ne sont pas

définies par des bords perceptibles, mais doivent toujours être appariées. Limité principalement par la puissance de calcul disponible.

L'estimation est devenue assez bonne avec suffisamment de données. L'appariement est la technique la plus efficace à utiliser dans le motif machines de reconnaissance qui lisent des chiffres et des lettres disponibles dans des contextes normalisés et contraints (moyens scanners qui lit votre numéro de crédit financier machines, chèques qui lisent les codes postaux).

2.4 Basé sur les connaissances (Knowledge-Based)

Ces méthodes se basent sur la connaissance des différents éléments qui constituent un visage et des relations qui existent entre eux. Ainsi, les positions relatives de différents éléments clés tels que la bouche, le nez et les yeux sont mesurées pour servir ensuite à la classification ’visage’ , ’non visage’ par Chiang et al. Le problème dans ce type de méthode est qu’il est difficile de bien définir de manière unique un visage [1].

Dans des approches basées sur la connaissance généralisées, les algorithmes sont développés en basant sur l'heuristique au sujet de visage. Bien qu’il soit simple de créer l'heuristique pour décrire le visage, la difficulté principale est dans la traduction de ces heuristiques dans des règles de classification d'une manière efficace. Si ces règles sont trop détaillées, elles peuvent proposer des détections manquées ; d’autre part, si elles sont plus générales elles peuvent présenter beaucoup de fausses détections. Néanmoins l’heuristique peut être employée à un taux acceptable. Ces méthodes sont conçues principalement pour la localisation de visage [9].

2.4.1 Méthode Kotropoulous et Pitas

Kotropoulous et Pitas [1] utilisent une méthode à base de règles. Tout d’abord, les caractéristiques du visage sont localisées à l’aide de la méthode de projection proposée par Kanade [1] pour détecter les contours d’un visage. Soit I(x,y) l’intensité

de la luminance du pixel (x,y) de l'image de taille m x n, les projections horizontale et verticale de cette image sont définies par :

$$HI(x) = \sum_{y=1}^{n} I(x,y) \text{ et } VI(x) = \sum_{x=1}^{m} I(x,y).$$

Le profil horizontal de l'image originale est calculé en premier. Les deux minimas locaux sont déterminés, ils correspondent aux bords gauche et droit du visage. Ensuite, le profil vertical est à son tour calculé. Les minima locaux de ce profil vertical correspondent aux positions de la bouche, du nez et des yeux. L'inconvénient de cette méthode est qu'elle n'arrive pas à détecter le visage lorsque ce dernier se trouve sur un arrière-plan complexe

.

2.4.2 **Méthode Yang and Huang**

Quant à eux, ont étudié les évolutions des caractéristiques du visage en fonction de la résolution. Quand la résolution de l'image d'un visage est réduite progressivement, par sous-échantillonnage ou par moyenne, les traits macroscopiques du visage disparaissent. Ainsi, pour une résolution faible, la région du visage devient uniforme. Yang et Huang se sont basés sur cette observation pour proposer une méthode hiérarchique de détection de visages. En commençant par les images à faible résolution, un ensemble de candidats de visage est déterminé à l'aide d'un ensemble de règles permettant de rechercher les régions uniformes dans une image. Les candidats de visage sont ensuite vérifiés en cherchant l'existence de traits faciaux proéminents grâce au calcul des minimas locaux à des résolutions supérieures. Une caractéristique intéressante de cette technique « descendante » de recherche de zone d'intérêt (informations globales vers des informations plus détaillées) est de réduire le temps de calcul nécessaire par l'utilisation d'images sous-échantillonnées. Malheureusement, cette technique occasionne de nombreuses fausses détections et un taux faible de détection [15].

2.5 Basé sur l'apparence (Appearance-Based)

Ces approches appliquent généralement des techniques d'apprentissage automatique. Ainsi, les modèles sont appris à partir d'un ensemble d'images représentatives de la variabilité de l'aspect facial. Ces modèles sont alors employées pour la détection. L'idée principale de ces méthodes est de considérer que le problème de la détection de visage est un problème de classification (visage, non-visage) [9]**.**

2.5.1 Face propre (Eigen face)

Elle consiste à projeter l'image dans un espace et à calculer la distance euclidienne entre l'image et sa projection. En effet, en codant l'image dans un espace, on dégrade l'information contenue dans l'image, puis on calcule la perte d'information est grande (évaluée à partir de la distance, que l'on compare à un seuil xé a priori), l'image n'est pas correctement représenté dans l'espace : elle ne contient pas de visage. Cette méthode donne des résultats assez encourageants, mais le temps de calcul est très important.

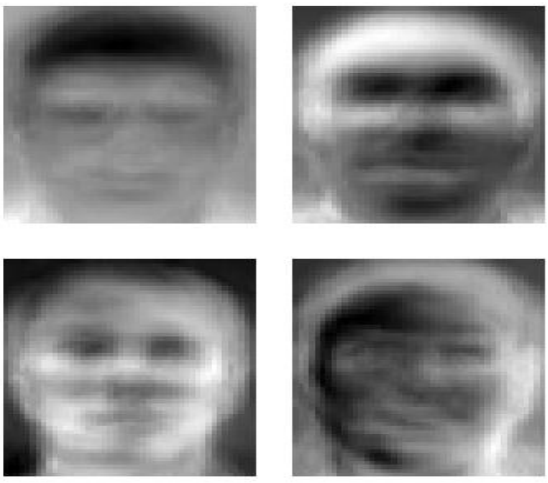


Figure 2.11: Eigen face propres de AT&T Laboratoires Cambridge [9]

2.5.2 Réseau de neurones de Rowley

Cet algorithme est basé sur un réseau de neurones pour détecter des vues frontales droites de visages en images en niveaux de gris. L'algorithme fonctionne en appliquant un ou plusieurs réseaux de neurones directement à parties de l'image d'entrée et arbitrer leurs résultats. Chaque réseau est formé pour produire les présences ou absence d'un visage. Les algorithmes et les méthodes de formation sont conçus pour être généraux, avec peu de personnalisation pour les visages. De nombreux chercheurs sur la détection des visages ont utilisé l'idée que les images faciales peuvent être caractérisées directement en termes d'intensités de pixels.

Ces images peuvent être caractérisées par des modèles probabilistes de l'ensemble des images de visage, La formation d'un réseau de neurones pour la tâche de détection de visage est difficile en raison de la difficulté caractériser des images prototypes «non faciales». Contrairement à la reconnaissance faciale, dans laquelle les classes à discriminer sont des visages différents, les deux classes à distinguer dans la détection sont images contenant des visages et images ne contenant pas de visages. Il est facile d'obtenir un échantillon représentatif des images qui contiennent des visages, mais beaucoup plus difficile d'obtenir un échantillon représentatif de celles qui n'en contiennent pas. Nous évitons le problème d'utiliser un énorme ensemble d'entraînement pour les non-visages en ajoutant sélectivement des images. Cette technique a été inventée par Rowley, Baluja et Kanade: (PAMI, janvier 1998) [9].

Algorithme de Rowley

Première étape

Le premier composant basé sur un réseau neuronal de notre système est un filtre qui reçoit en entrée une région de 20x20 pixels du l'image, et génère une sortie allant de 1 à -1, signifiant la présence ou l'absence d'un visage, respectivement. Pour détecter des visages n'importe où dans l'entrée, le filtre est appliqué à chaque emplacement de l'image. Pour détecter des visages plus grands que la taille de la fenêtre, l'image d'entrée est réduite à plusieurs reprises (par sous-échantillonnage),

et le filtre est appliqué à chaque taille. Ce filtre doit avoir une certaine invariance position et échelle. La quantité d'invariance détermine le nombre d'échelles et de positions à lequel il doit être appliqué. Pour le travail présenté ici, nous appliquons le filtre à chaque pixel.

Deuxième étape

Fusion des détections et de l'arbitrage qui se chevauchent. Les exemples de la Figure 2.12 montrent que la sortie brute d'un réseau unique contiendra un certain nombre de fausses détections. Dans cette section, nous présentons deux stratégies pour améliorer la fiabilité du détecteur fusion de détections qui se chevauchent à partir d'un seul réseau et arbitrage entre plusieurs réseaux [12].

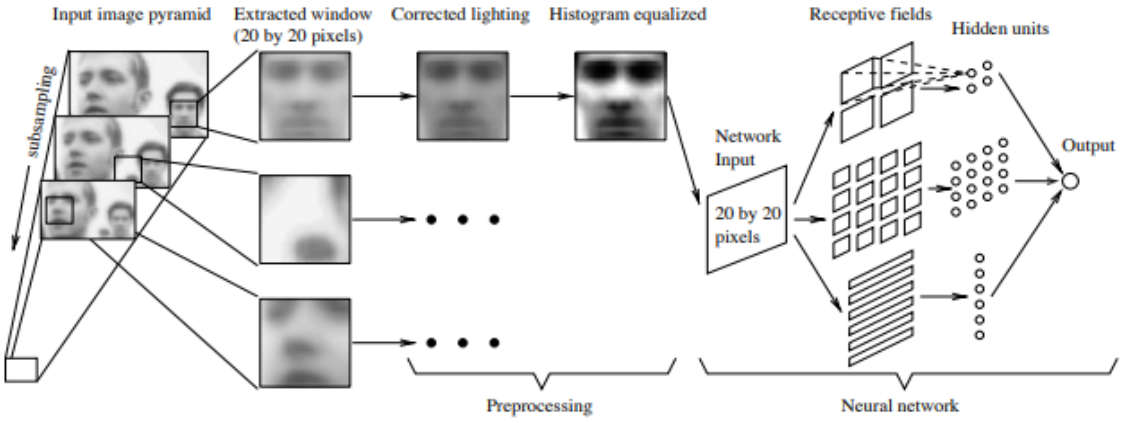


Figure 2.12: Réseau de neurones de Rowley [12]

2.5.3 **Viseur convolutif (Convolutional Face Finder)**

Le viseur à convolution a été présenté dans andrelieson convolutional neural networks (CNN) introduit et utilisé avec succès par "LeCun et al" [20]. Il consiste en un pipeline de convolutions et d'opérations de sous-échantillonnage (voir la figure 2.14). Ce pipeline effectue une extraction automatique des caractéristiques dans des images de taille 32 × 36, et une classification des caractéristiques extraites, dans un seul schéma intégré. La contribution majeure de cet algorithme vient de la conception du système de détection d'un visage rapide et fiable ,pour détecter les modèles de visage de taille et apparence variables, tournées jusqu'à ± 20 degrés

dans le plan de l'image et tourné jusqu'à ± 60 degrés, dans une image complexe du monde réel. Et un taux de fausses alarmes très faible.

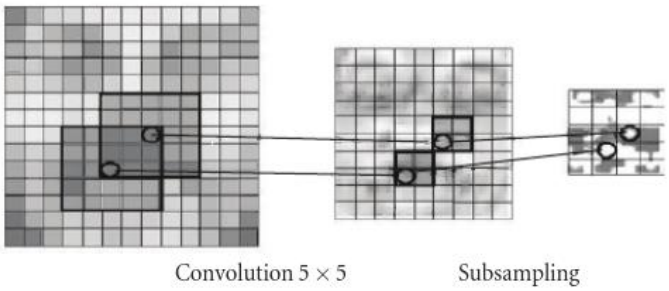


Figure 2.13: Champs réceptifs pour la convolution et
le sous-échantillonnage pour C1-S1 [13]

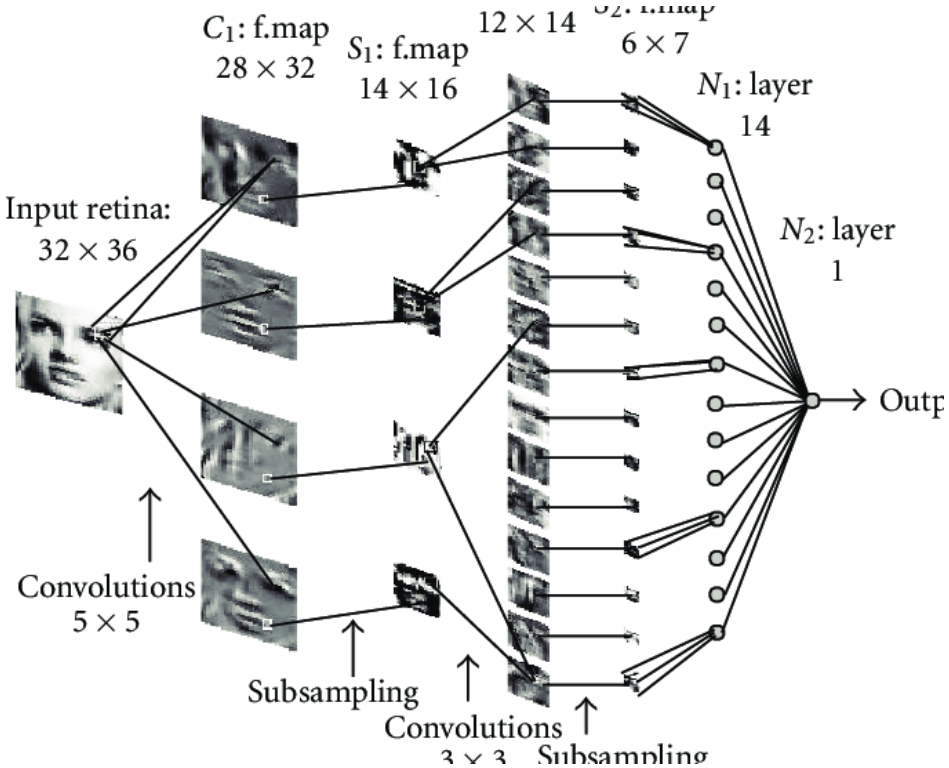


Figure 2.14: Canalisation de recherche de visage par convolution [13]

Le réseau neuronal convolutif, représenté sur la figure 2.14, se compose d'un ensemble de trois types de couches. Les couches Ci appelées couchent convolutionnelles, qui contiennent un certain nombre de plans. La couche C1 est connectée à la rétine, recevant la zone d'image à classer comme visage ou non-visage. Chaque unité dans un plan reçoit des données d'un petit voisinage (champ

de réception local biologique) dans les plans de la couche précédente. Un biais entraînable est ajouté aux résultats de chaque masque de convolution. Plusieurs plans sont utilisés dans chaque couche afin que plusieurs fonctionnalités puissent être détectées. Une fois qu'une entité a été détectée, son emplacement exact est moins important. Par conséquent, chaque couche convolutionnelle Ci est généralement suivie d'une autre couche Si qui effectue des opérations locales de moyenne et de sous-échantillonnage [13].

La couche de sous-échantillonnage gérera les positions exactes des données extraits qui ne sont pas importantes. Seule la position relative d'une entité à une autre caractéristique est pertinente. Elle sera également à réduire la résolution spatiale et Sensibilité au décalage et à la distorsion.

2.5.4 **Classificateurs Haar**

Une méthode bien connue de détection d'objets complexes tels que les visages est l'utilisation de « classificateurs de Haar » montés en cascade (boostés) au moyen d'un algorithme AdaBoost. Cette méthode est implémentée nativement dans la bibliothèque OpenCV et a été présenté initialement dans Viola et Jones .Le principe de cette méthode est d'obtenir un algorithme complexe de classification, composé de classificateurs élémentaires qui éliminent au fur et à mesure les zones de l'image qui ne sont pas compatibles avec l'objet recherché. Ces classificateurs binaires reposent sur des primitives visuelles qui dérivent des fonctions de Haar [10].

La détection d'objets à l'aide de classificateurs en cascade basés sur les fonctionnalités de Haar est une méthode efficace de détection d'objets proposée par Paul Viola et Michael Jones dans leur article, "Rapid Object Detectionusing a Boosted Cascade of Simple Features" en 2001. Il s'agit d'une approche basée sur l'apprentissage automatique où la fonction en cascade est formée à partir d'un grand nombre d'images positives et négatives. Il est ensuite utilisé pour détecter des objets dans d'autres images [10].

Ici, nous allons travailler avec la détection des visages. Initialement, l'algorithme a besoin de beaucoup d'images positives (images de visages) et d'images négatives (images sans visages) pour former le classificateur. Ensuite, nous devons en extraire des fonctionnalités. Pour cela, les fonctionnalités Haar illustrées dans l'image ci-dessous sont utilisées. Ils sont exactement comme notre noyau convolutionnel. Chaque entité est une valeur unique obtenue en soustrayant la somme des pixels sous le rectangle blanc de la somme des pixels sous le rectangle noir [10].

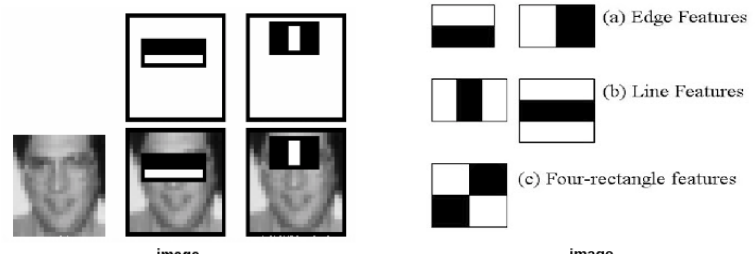


Figure 2.15: Caractéristiques de Haar [10]

Le classificateur final est une somme pondérée de ces classificateurs faibles. Il est appelé faible parce qu'il ne peut à lui seul classer l'image, mais forme avec d'autres un classificateur fort. Le document indique que même 200 fonctionnalités offrent une détection avec une précision de 95%. Leur configuration finale avait environ 6000 fonctionnalités.

2.6 Basé sur les caractéristiques (Feature-Based)

La plupart des approches feature-based ont des étapes consécutives semblables. Habituellement, la première étape consiste à faire éliminer les pixels sans importance en utilisant par exemple le filtrage des couleurs ou bien la détection des bords (edge détection). Les résultats obtenus par la première étape sont ambiguë. Dans la deuxième étape, les dispositifs visuels qui n’ont pas été éliminé dans la première étape, sont organisés dans une connaissance de visage globale ou

géométrie. Puis une analyse de dispositifs est employée pour que les dispositifs ambigus soient réduits, et les endroits du visage et des dispositifs faciaux sont déterminés. L'étape finale peut comporter l'utilisation des Template ou des modèles de forme active [9].

2.6.1 Histogram of Oriented Gradients (HOGs)

Un descripteur de fonctionnalité est une représentation d'une image ou d'un patch d'image qui simplifie l'image en extrayant des informations utiles et en jetant des informations superflues.

Typiquement, un descripteur d'entité convertit une image de taille largeur x hauteurs x 3 (canaux) en un vecteur d'entité / tableau de longueur n. Dans le cas du descripteur de caractéristique HOG, l'image d'entrée est de taille 64 x 128 x 3 et le vecteur de caractéristique de sortie est de longueur 3 780.

L'histogramme des gradients orientés (HOG) est un descripteur de caractéristiques utilisé pour détecter des objets en vision artificielle et en traitement d'image. La technique du descripteur HOG compte les occurrences de l'orientation du gradient dans des parties localisées d'une fenêtre de détection d'image ou région d'intérêt (ROI).

L'implémentation de l'algorithme de descripteur HOG est la suivante:

1. Divisez l'image en petites régions connectées appelées cellules et calculez pour chaque cellule un histogramme des directions de gradient ou des orientations de bord pour les pixels de la cellule.

2. Discrétisez chaque cellule en cases angulaires en fonction de l'orientation du gradient.

3. Le pixel de chaque cellule contribue au gradient pondéré de sa corbeille angulaire correspondante.

4. Les groupes de cellules adjacentes sont considérés comme des régions spatiales appelées blocs. Le regroupement des cellules en un bloc est la base du regroupement et de la normalisation des histogrammes.

5. Le groupe d'histogrammes normalisé représente l'histogramme de bloc. L'ensemble de ces histogrammes de blocs représente le descripteur.

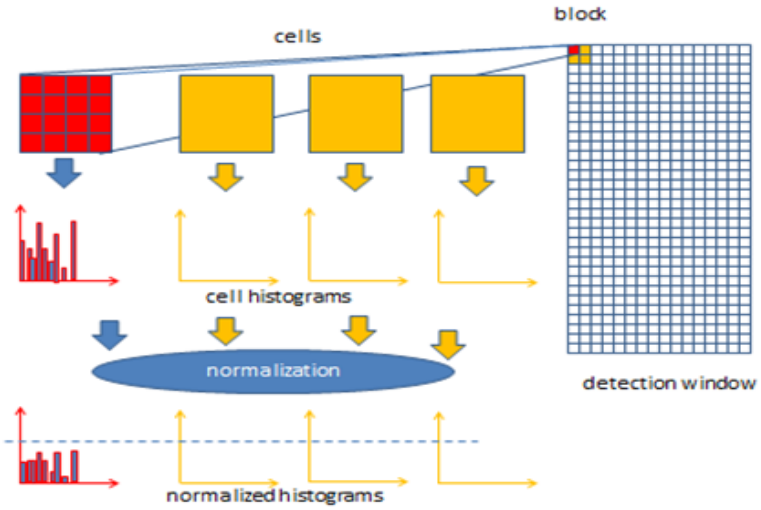


Figure 2.16: schéma de mise en œuvre de l'algorithme

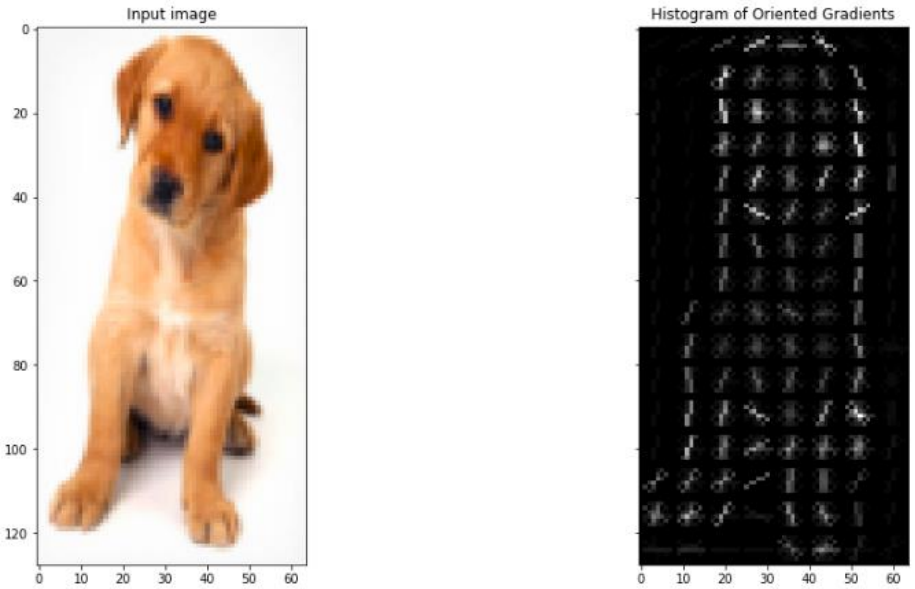


Figure 2.17: Résultat de l'application de HOG

2.6.2 SIFT (Scale Invariant FeatureTransform)

La scale-invariant feature transform (SIFT), que l'on peut traduire par « transformation de caractéristiques visuelles invariante à l'échelle », est un algorithme utilisé dans le domaine de la vision par ordinateur pour détecter et identifier les éléments similaires entre différentes images numériques (éléments de paysages, objets, personnes, etc.).

Ainsi, en 2004, D.Lowe, Université de la Colombie-Britannique, a proposé un nouvel algorithme, Scale Invariant FeatureTransform (SIFT) dans son article, Distinctive Image Features from Scale-Invariant Keypoints, qui extrait les points clés et calcule ses descripteurs [21].

L'algorithme SIFT comprend principalement quatre étapes :

- **Détection d'extrema d'espace-échelle**

Il est évident que nous ne pouvons pas utiliser la même fenêtre pour détecter des points clés avec une échelle différente. Ça marche avec un petit coin. Mais pour détecter des coins plus grands, nous avons besoin de fenêtres plus grandes. Pour cela, le filtrage d'espace d'échelle est utilisé. On y trouve du laplacien de gaussien pour l'image avec différentes valeurs sigma, LoG agit comme un détecteur de blob qui détecte les blobs de différentes tailles en raison du changement de sigma. En bref, sigma agit comme un paramètre d'échelle. Par exemple, dans l'image ci-dessous, le noyau gaussien avec un sigma bas donne une valeur élevée pour un petit coin tandis que le noyau gaussien avec un sigma élevé convient bien pour un coin plus grand. Ainsi, nous pouvons trouver les maxima locaux à travers l'échelle et l'espace qui nous donne une liste de valeurs (x, y, sigma) ce qui signifie qu'il y a un point clé potentiel à (x, y) à l'échelle sigma [21].

Mais ce LoG est un peu coûteux, parce que l'algorithme SIFT utilise la différence des Gaussiens qui est une approximation du LoG. La différence de gaussienne est obtenue comme la différence de flou gaussien d'une image avec deux sigmas différents, que ce soit sigma et k.sigma. Ce processus est effectué pour différentes octaves de l'image dans la pyramide gaussienne. Il est représenté dans l'image ci-dessous:

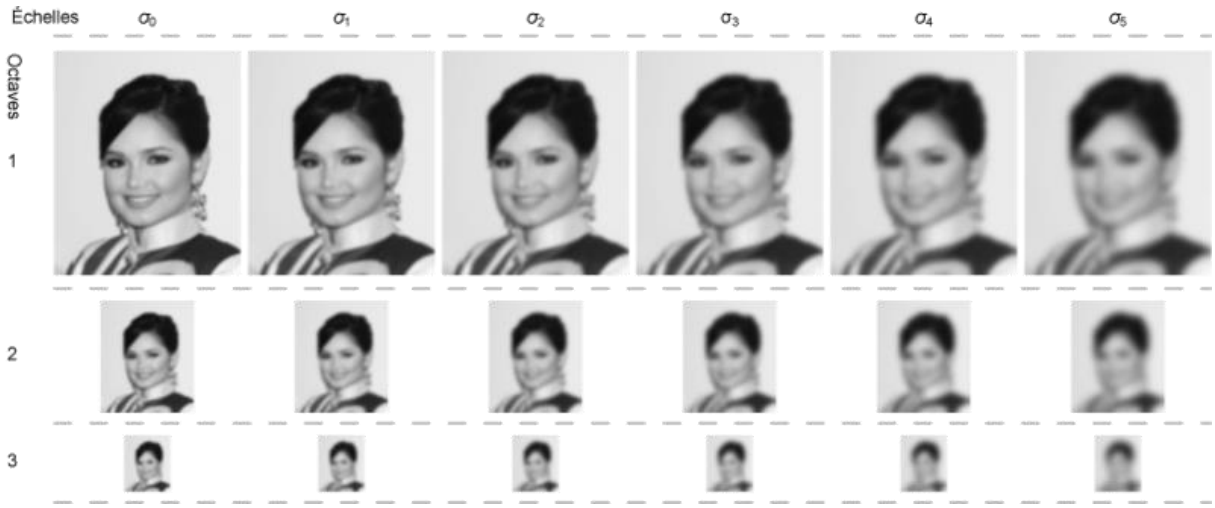


Figure 2.18: Image avec un sigma différent [21]

- **Localisation des points clés**

Une fois que les emplacements de points clés potentiels ont été trouvés, ils doivent être affinés pour obtenir des résultats plus précis. Ils ont utilisé l'expansion de l'échelle de l'échelle de la série Taylor pour obtenir un emplacement plus précis des extrema, et si l'intensité à cet extrema est inférieure à une valeur seuil (0,03 selon le document), elle est rejetée. Ce seuil est appelé « contrast threshold »**.** le DoG(difference of gaussians) a une réponse plus élevée pour les bords, donc les bords doivent également être supprimés [21].

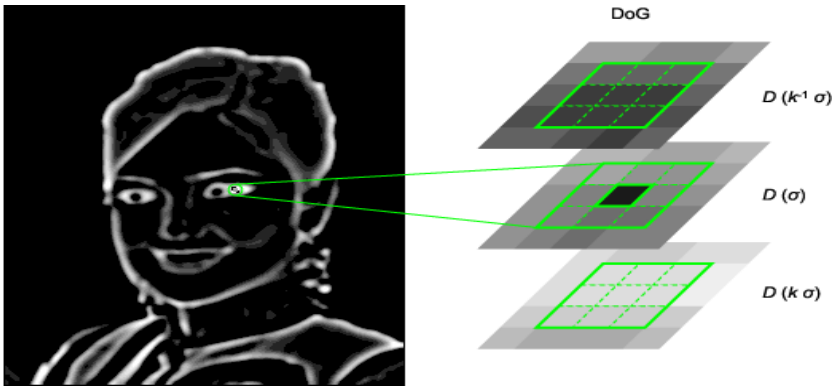


Figure 2.19: Détection de points clé [21]

- Affectation d'orientation

Une orientation est désormais attribuée à chaque point clé pour obtenir une invariance à la rotation de l'image. Un voisinage est pris autour de l'emplacement du point clé en fonction de l'échelle, et l'amplitude et la direction du gradient sont calculées dans cette région. Un histogramme d'orientation avec 36 cases couvrant 360 degrés est créé (il est pondéré par la magnitude du gradient et la fenêtre circulaire pondérée gaussienne avec \ sigma égal à 1,5 fois l'échelle du point clé). Le pic le plus élevé de l'histogramme est pris et tout pic supérieur à 80% est également pris en compte pour calculer l'orientation. Il crée des points clés avec le même emplacement et la même échelle, mais des directions différentes. Il contribue à la stabilité de la correspondance [21].

- Descripteur du point clé

Le descripteur de point clé est maintenant créé. Un quartier 16x16 autour du point clé est pris. Il est divisé en 16 sous-blocs de taille 4x4. Pour chaque sous-bloc, un histogramme d'orientation à 8 cases est créé. Ainsi, un total de 128 valeurs de bac est disponible. Il est représenté comme un vecteur pour former un descripteur de point clé. En plus de cela, plusieurs mesures sont prises pour atteindre la robustesse contre les changements d'éclairage, la rotation, etc [21].

- Correspondance de points clés

Les points clés entre deux images sont appariés en identifiant leurs voisins les plus proches. Mais dans certains cas, le deuxième match le plus proche peut être très proche du premier. Cela peut se produire en raison du bruit ou d'autres raisons. Dans ce cas, le rapport de la distance la plus proche à la deuxième distance la plus proche est prise. S'il est supérieur à 0.8, ils sont rejetés. Il élimine environ 90% des fausses correspondances et ne rejette que 5% des correspondances correctes [21].

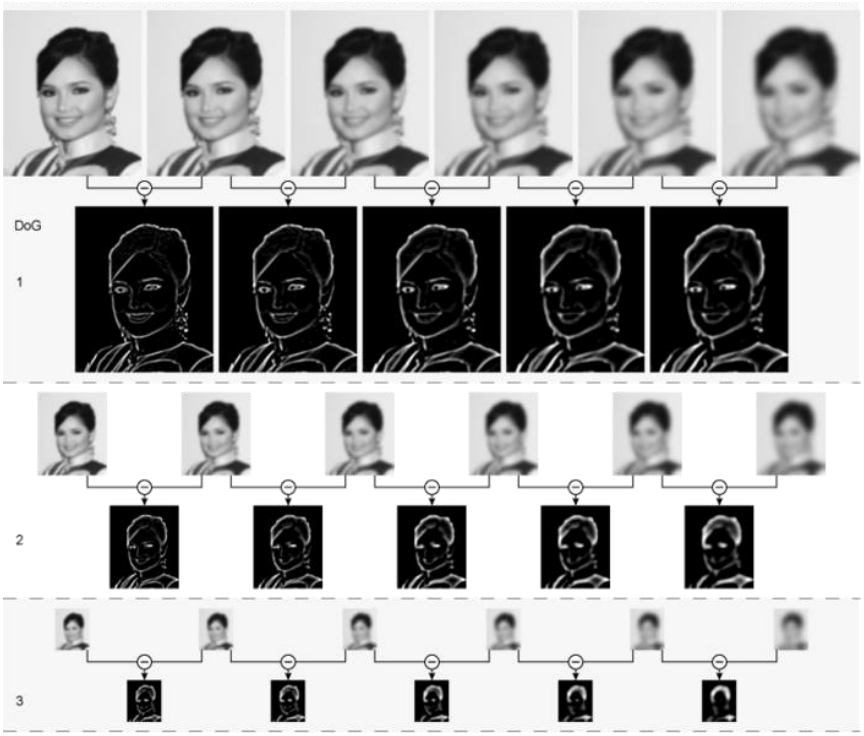


Figure 2.20: Résultat final de l’algorithme [21]

2.6.3 Speeded Up Robust Features (SURF*)*

Dans la dernière méthode, nous avons vu SIFT pour la détection et la description des points clés. Mais c'était relativement lent et les gens avaient besoin d'une version plus rapide. En 2006, trois personnes, Bay, H., Tuytelaars, T. et Van Gool, L, ont publié un autre article, «SURF: Speeded Up RobustFeatures», qui a introduit un nouvel algorithme appelé SURF. Comme son nom l'indique, il s'agit d'une version accélérée de SIFT [22].

Dans SIFT, Lowe a approché le laplacien de gaussien avec la différence de gaussien pour trouver l'échelle-espace. SURF va un peu plus loin et se rapproche de LoG avec Box Filter. L'image ci-dessous montre une démonstration d'une telle approximation. Un gros avantage de cette approximation est que la convolution avec un filtre en boîte peut être facilement calculée à l'aide d'images intégrales. Et cela peut être fait en parallèle pour différentes échelles. Les SURF s'appuient également sur le déterminant de la matrice de Hesse pour l'échelle et l'emplacement [22].

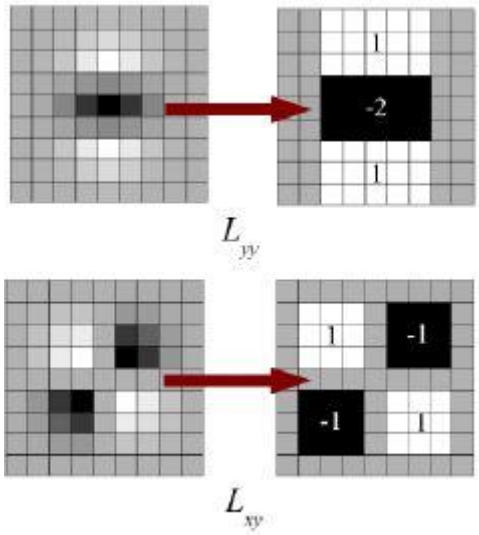


Figure 2.21:Filtre approximatif du laplacien [22]

Pour l'attribution de l'orientation, SURF utilise des réponses en ondelettes dans les directions horizontale et verticale pour un voisinage de taille 6s. Des poids gaussiens adéquats lui sont également appliqués. Ensuite, ils sont tracés dans un espace comme indiqué dans l'image ci-dessous. L'orientation dominante est estimée en calculant la somme de toutes les réponses dans une fenêtre d'orientation coulissante d'angle 60 degrés. Ce qui est intéressant, c'est que la réponse en ondelettes peut être trouvée en utilisant des images intégrales très facilement à n'importe quelle échelle. Pour de nombreuses applications, l'invariance de rotation n'est pas requise, donc pas besoin de trouver cette orientation, ce qui accélère le processus [22].

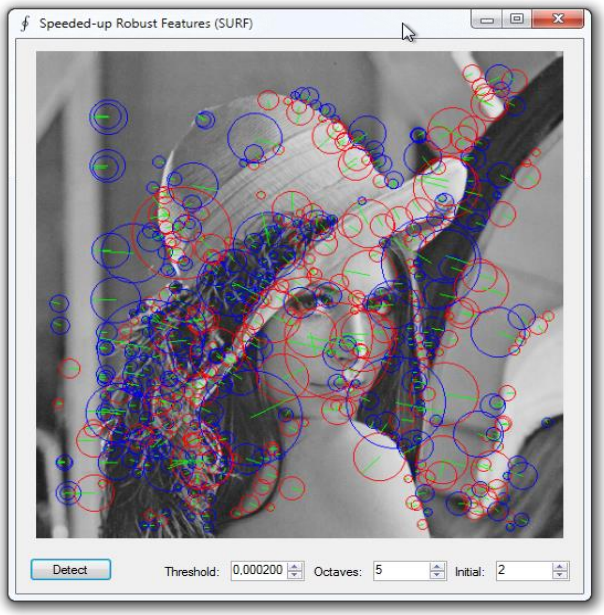


Figure 2.22: Algorithme SURF sur les visages humains [22]

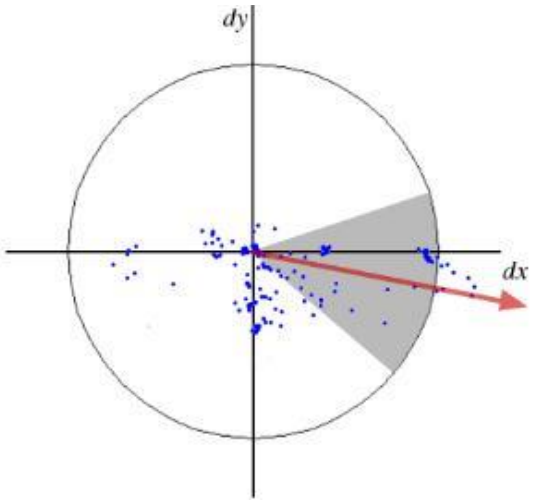


Figure 2.23: Réponses en ondelettes dans le sens horizontal et vertical [22]

2.6.4 Binary Robust Independent Elementary Features (BRIEF)

BRIEF est un descripteur de point caractéristique à usage général qui peut être combiné avec des détecteurs arbitraires. Il est robuste aux classes typiques de

transformations d'images photométriques et géométriques. BRIEF cible les applications en temps réel en leur laissant une grande partie de la puissance CPU disponible pour les tâches suivantes, mais permet également d'exécuter des algorithmes de correspondance de points de fonctionnalité sur des appareils peu performants comme les téléphones mobiles [22].

BRIEF est extrêmement efficace à calculer, ce qui se traduit par une accélération de près de deux ordres de grandeur par rapport à U-SURF-64. Comparé à une récente implémentation GPU de SURF, BRIEF-32P est toujours environ 4 fois plus rapide à calculer tout en utilisant un seul processeur 2,53 GHz. Les timings sont les valeurs médianes obtenues à partir de 10 exécutions d'un cycle standard de détection-description-correspondance [22].

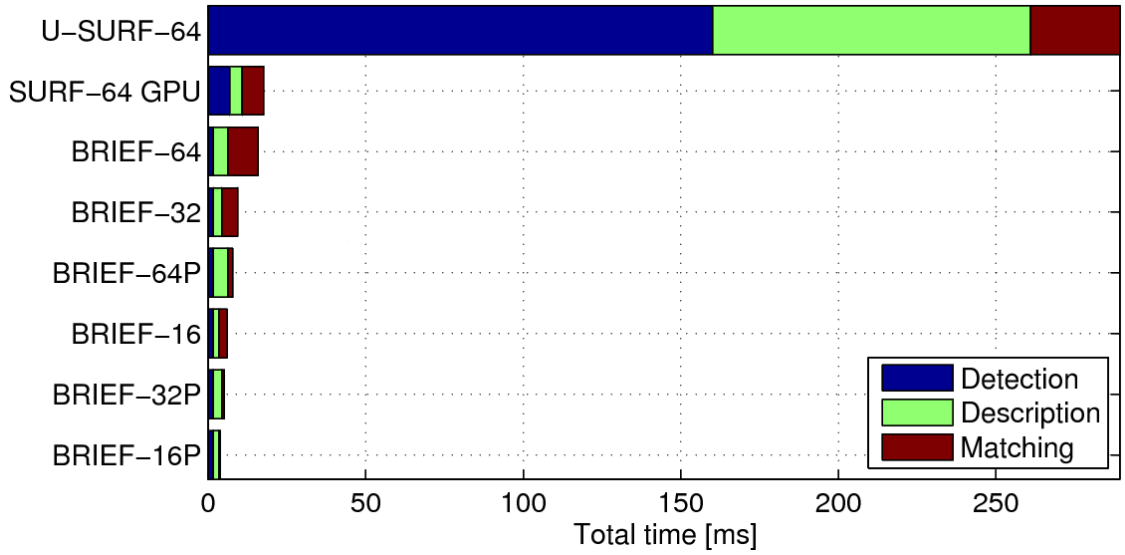


Figure 2.24: Comparaison entre BRIEF et SURF [22]

Après avoir détecté le point clé, nous continuons à calculer un descripteur pour chacun d'eux. Les descripteurs d'entités codent des informations intéressantes en une série de nombres et agissent comme une sorte d'empreinte digitale qui peut être utilisée pour différencier une entité d'une autre. Le voisinage défini autour du pixel (point-clé) est connu sous le nom de patch, qui est un carré d'une certaine largeur et hauteur de pixel. Les patchs d'image pourraient être efficacement classés sur la base d'un nombre relativement faible de comparaisons d'intensité par paire.

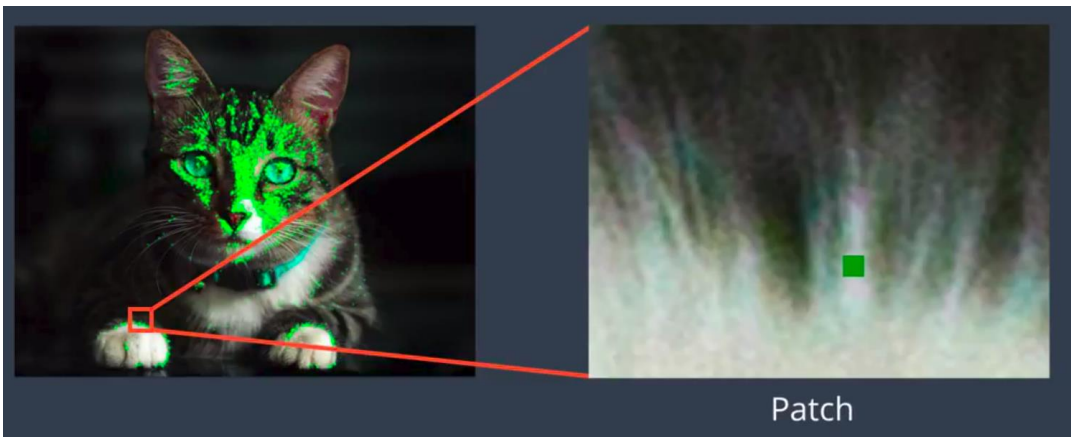


Figure 2.25: Extraction de patch d'une image [22]

Convertissez brièvement les correctifs d'image en un vecteur d'entités binaires afin qu'ils puissent ensemble représenter un objet. Le vecteur d'entités binaires, également connu sous le nom de descripteur d'entités binaires, est un vecteur d'entités qui ne contient que 1 et 0. En bref, chaque point clé est décrit par un vecteur d'entités composé d'une chaîne de 128 à 512 bits [24].

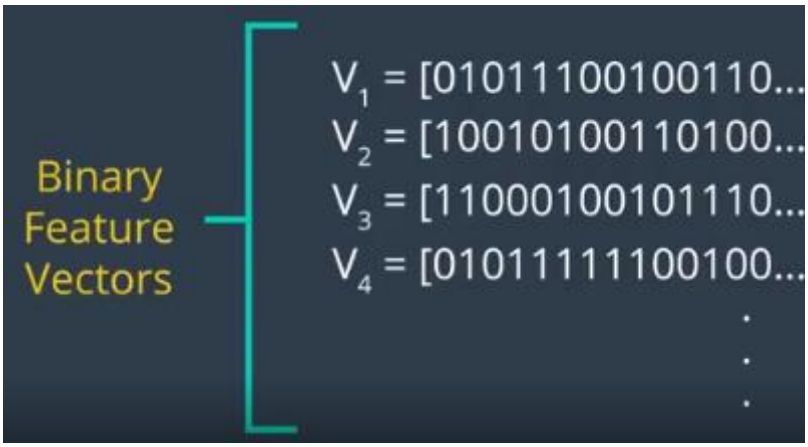


Figure 2.26: Vecteur d'entités binaires [24]

- **Noyaux de lissage**

BRIEF traite de l'image au niveau des pixels et est donc très sensible au bruit. En pré-lissant le patch, cette sensibilité peut être réduite, augmentant ainsi la stabilité et la répétabilité des descripteurs. C'est pour la même raison que les images doivent être lissées avant de pouvoir être significativement différencié lors de la recherche de bords [24].

Figure 2.27: Lissage de l'image [24]

- **Avantages de BRIEF**

BRIEF s'appuie sur un nombre relativement faible de tests de déférence d'intensité pour représenter un patch d'image sous la forme d'une chaîne binaire. Non seulement la construction et l'appariement de ce descripteur sont beaucoup plus rapides que pour d'autres descripteurs de pointe, mais il a également tendance à produire des taux de reconnaissance plus élevés, tant que l'invariance aux grandes rotations dans le plan n'est pas une exigence [24].

2.6.5 Oriented FAST and Rotated BRIEF (ORB)

L'ORB est essentiellement une fusion du détecteur de points clés FAST et du descripteur BRIEF avec de nombreuses modifications pour améliorer les performances. Tout d'abord, utilisez FAST pour trouver les points clés, puis appliquez la mesure de coin Harris pour trouver les N points supérieurs parmi eux. Il utilise également la pyramide pour produire des fonctionnalités multi-échelles. Mais un problème est que, FAST ne calcule pas l'orientation. Qu'en est-il de l'invariance de rotation? Les auteurs ont proposé la modification suivante [24].

Il calcule le centre de gravité pondéré en intensité du patch avec le coin situé au centre. La direction du vecteur de ce point d'angle au centre de gravité donne l'orientation. Pour améliorer l'invariance de rotation, les moments sont calculés avec x et y qui doivent être dans une région circulaire de rayon r, où r est la taille du patch.

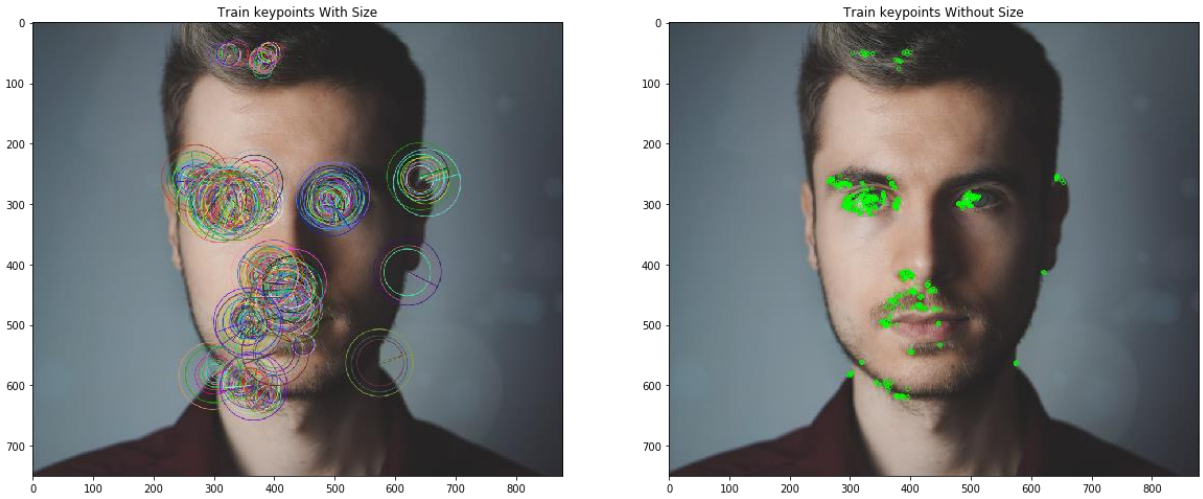


Figure 2.28: Détection des points clé avec ORB [24]

Maintenant, pour les descripteurs, ORB utilise de brefs descripteurs. Mais nous avons déjà vu que BRIEF fonctionne mal avec la rotation. Donc, ce que fait l'ORB, c'est de "piloter" BRIEF en fonction de l'orientation des points clés. Pour tout ensemble de fonctionnalités de n tests binaires à l'emplacement (xi, yi), définissez une matrice 2 fois n, S contient les coordonnées de ces pixels. Ensuite, en utilisant l'orientation du patch, theta, sa matrice de rotation est trouvée et fait tourner le S pour obtenir la version pilotée (tournée) S_t heta [24].

L'ORB discrétise l'angle par incréments de 2 x Pi /30 (12 degrés) et construit une table de recherche de modèles BRIEF pré calculés. Tant que l'orientation des points clés \ thêta est cohérente dans toutes les vues, l'ensemble correct de points S_ thêta sera utilisé pour calculer son descripteur [24].

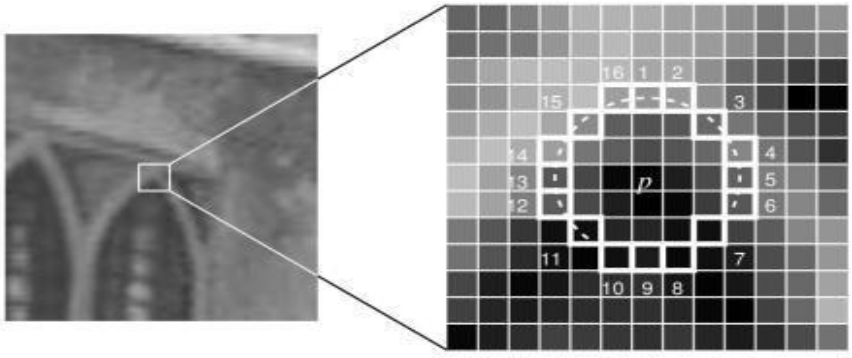


Figure 2.29: Point clé d'une image [24]

BRIEF a la propriété importante que chaque caractéristique de bit a une grande variance et une moyenne proche de 0.5. Mais une fois qu'il est orienté selon la direction des points clés, il perd cette propriété et devient plus distribué. Une variance élevée rend une caractéristique plus discriminante, car elle répond différemment aux entrées. Une autre propriété souhaitable est que les tests ne soient pas corrélés, car chaque test contribuera au résultat. Pour résoudre tout cela, ORB lance une recherche gourmande parmi tous les tests binaires possibles pour trouver ceux qui ont à la fois une variance élevée et des moyennes proches de 0.5, ainsi que non corrélés. Le résultat est appelé rBRIEF.

Pour l'appariement des descripteurs, le LSH à sondes multiples qui améliore le LSH traditionnel est utilisé. Le document indique que ORB est beaucoup plus rapide que SURF et SIFT et que le descripteur ORB fonctionne mieux que SURF. ORB est un bon choix dans les appareils à faible consommation pour l'assemblage panoramique [24].

2.7 **Conclusion**

Dans ce chapitre, nous avons présenté un historique des travaux effectués dans le domaine de la détection de visage, ensuite, en revue les étapes d'évolution de ce domaine, nous avons fourni une catégorisation des techniques utilisées pour la détection. Nous avons vu des méthodes de détection de visages utilisant des études locales et qui nécessitent en général une segmentation. L'image ainsi traitée peut alors alimenter un réseau de neurones par exemple, ou être utilisée par des méthodes basées sur des modèles de connaissance. Dans le but d’améliorer le traitement en apportant une contribution du point de vue calcul, nous avons opté pour la méthode HOG et classification par la méthode Haar.

Chapitre 3: Méthodes de reconnaissance faciale.

3.1. Introduction

Les systèmes de reconnaissance faciale existent depuis près de 50 ans. En raison de ses nombreuses applications pratiques dans le domaine de la biométrie, de la sécurité de l'information, du contrôle d'accès, de l'application des lois, des cartes à puce et du système de surveillance. La première application à grande échelle de la reconnaissance faciale a été réalisée en Floride. Ces dernières années, les techniques basées sur la biométrie sont devenues l'option la plus prometteuse pour reconnaître les individus, au lieu de certifier les personnes et de leur permettre d'accéder à des domaines physiques et virtuels basés sur des mots de passe, des codes PIN, des cartes à puce, des cartes en plastique, des jetons, des clés, etc.

Ces méthodes examinent les caractéristiques physiologiques ou comportementales d'un individu afin de déterminer ou de vérifier son identité. Les mots de passe et les NIP sont difficiles à retenir et peuvent être volés ou devinés; les cartes, jetons, clés et similaires peuvent être égarés, oubliés ou dupliqués; les cartes magnétiques peuvent être corrompues et peu claires. Cependant, les traits biologiques d'un individu ne peuvent être égarés, oubliés, volés ou falsifiés. Afin de développer un système de reconnaissance faciale utile et applicable, plusieurs facteurs doivent être pris en compte.

1. La vitesse globale du système, de la détection à la reconnaissance, devrait être acceptable.
2. La précision doit être élevée
3. Le système doit être facilement mis à jour et agrandi, ce qui permet d'augmenter facilement le nombre de sujets pouvant être reconnus.

3.2. Système de reconnaissance

Un énoncé général du problème de la reconnaissance de visages peut être formulé comme suit : étant donné une image ou une vidéo d'une scène, identifier ou vérifier une ou plusieurs personnes dans la scène en utilisant une base de données de visages stockés. L'information collatérale disponible telle que la race, l'âge, le genre, le sexe, l'expression faciale ou la parole peut être utilisée pour rétrécir l'étendu de la recherche (amélioration de la reconnaissance). La solution à ce problème implique la segmentation des visages (détection de visage) à partir de scènes complexes, l'extraction des traits des régions du visage, la reconnaissance ou la vérification. Dans les problèmes de reconnaissance, l'entrée du système est un visage inconnu, et le système retourne son identité déterminée à partir d'une base de données d'individus connus, tandis que dans les problèmes de vérification, le système doit confirmer ou rejeter l'identité réclamée du visage d'entrée.

Les systèmes technologiques peuvent parfois varier en matière de reconnaissance faciale, mais le fonctionnement général est le suivant.

3.2.1. Étape 1: Détection de visage

Pour commencer, la caméra détectera et reconnaîtra un visage, seul ou dans une foule. Le visage est mieux détecté lorsque la personne regarde directement la caméra. Les progrès technologiques ont également permis de légères variations de cela.

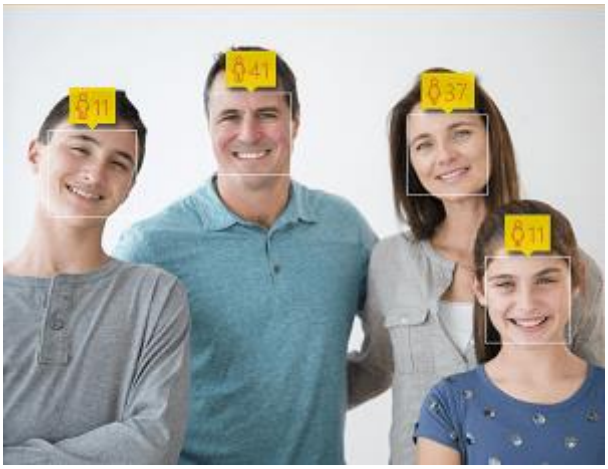


Figure 3.1: Détection des visages.

3.2.2. Étape 2: Analyse du visage

Ensuite, une photo du visage est capturée et analysée. La plupart des fonctions de reconnaissance faciale reposent sur des images 2D plutôt que sur 3D car elles permettent de faire correspondre plus facilement une photo 2D avec des photos publiques ou celles d'une base de données. Des repères ou des points nodaux distinctifs composent chaque visage. Chaque visage humain a 80 points nodaux. Un logiciel de reconnaissance faciale analysera les points nodaux tels que la distance entre vos yeux ou la forme de vos pommettes.

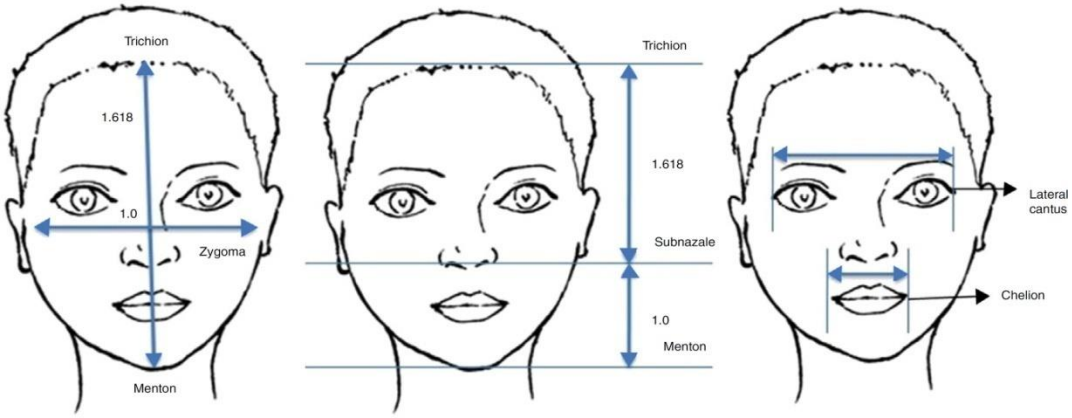


Figure 3.2: Analyse du visage.

3.2.3. Étape 3: Conversion d'une image en données

L'analyse du visage est ensuite transformée en une formule mathématique. Ces traits du visage deviennent des nombres dans un code. Ce code numérique est appelé une empreinte faciale. Semblable à la structure unique d'une empreinte digitale, chaque personne a sa propre empreinte faciale.

3.2.4. Étape 4: Trouver une correspondance

Ce code est ensuite comparé à une base de données d'autres empreintes faciales. Cette base de données contient des photos avec identification qui peuvent être comparées.

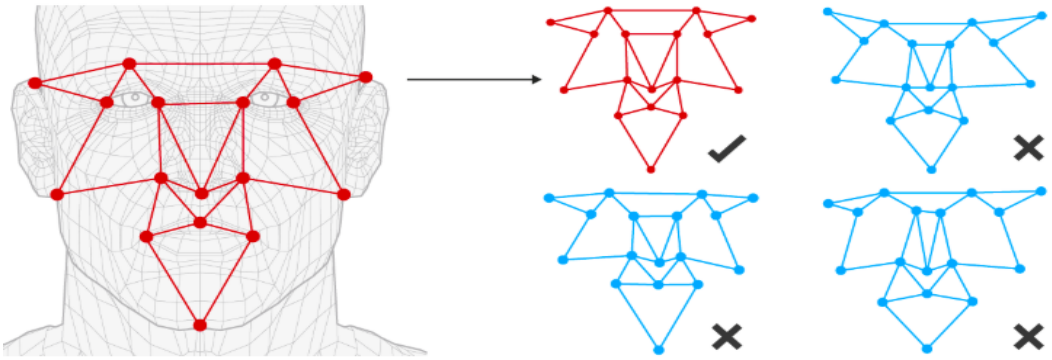


Figure 3.3: Trouver une correspondance.

FBI a accès à plus de 641 millions de photos, dont 21 bases de données d'État telles que les DMV. Un autre exemple de base de données à laquelle beaucoup ont accès est les photos de Facebook. Toutes les photos marquées du nom d'une personne font partie de la base de données Facebook.

La technologie identifie ensuite une correspondance pour les fonctionnalités exactes dans la base de données fournie. Il revient avec la correspondance et les informations jointes telles que le nom et l'adresse.

3.3. Méthodes de reconnaissance

De nombreuses approches de reconnaissance faciale ont été proposées au cours des deux dernières décennies. Basé sur l'étude psychologique de la façon dont les humains utilisent les caractéristiques holistiques et locales, ces approches peuvent être globalement classées comme approche holistique et approche basée sur les caractéristiques.

3.3.1 Approche holistique

Les approches basées sur des modèles holistiques utilisent des méthodes statistiques pour représenter l'image dans son ensemble. Ces méthodes utilisent toute la région du visage comme entrée brute d'un système de reconnaissance plutôt que les caractéristiques locales du visage. Une image est considérée comme un vecteur à haute dimension, c'est-à-dire un point dans un espace vectoriel de grande dimension. Basé sur l'apparence ou basé sur la vue les approches utilisent des

techniques statistiques pour analyser la distribution de l'image vectrice dans l'espace vectoriel, et dériver une représentation efficace et effective dans l'espace de fonctionnalité. Donné une image de test, la similitude entre les prototypes stockés et la vue de test est ensuite effectuée dans l'espace de fonctionnalité. Ces schémas peuvent être subdivisés en deux groupes: statistiques et IA (Intelligence artificielle). Méthode d'analyse des composants principaux (ACP) et la méthode d'analyse discriminante linéaire (LDI) sont des exemples de statistiques holistiques approches de reconnaissance faciale. Les modèles de Markov cachés sont des exemples d'approches qui utilisent l'intelligence artificielle.

PCA

La méthode d'analyse en composantes principales (ACP) est une méthode et technique statistique courante pour trouver les motifs dans les données de haute dimension. Ceci est réalisé en simplifiant un ensemble de données dans une dimension inférieure tout en conservant les caractéristiques de l'ensemble de données. La réduction de la dimensionnalité est effectuée par ACP pour réduire la dimension des données à des limites plus faciles à gérer, pour saisir les caractéristiques saillantes propres aux classes et pour éliminer la redondance. Pour ce faire, une image de visage est projetée sur plusieurs modèles de visage appelés faces propres qui peuvent être considérés comme un ensemble de caractéristiques qui caractérisent la variation entre les images du visage.

Les faces propres définissent un espace caractéristique qui réduit considérablement la dimensionnalité de l'espace d'origine, l'identification s'effectue dans cet espace réduit. Une fois qu'un ensemble de faces propres est calculée, une image de visage peut être approximativement reconstruite à l'aide d'une combinaison des faces propres. Les poids de projection forment un vecteur caractéristique pour la représentation et la reconnaissance du visage. Lorsqu'une nouvelle image de test est donnée, les poids sont calculés en projetant l'image sur les vecteurs à face propre. La classification est ensuite effectuée en comparant les distances entre les poids. En utilisant toutes les faces propres extraites des images originales, on peut reconstituer l'image d'origine à partir des faces propres, de sorte qu'elle corresponde exactement à l'image d'origine. L'ACP semble bien fonctionner lorsqu'une seule image de chaque

individu est disponible, mais lorsque plusieurs images par personne sont présentes, PCA choisira alors la projection qui maximise la diffusion totale et conserve les variations indésirables dues à l'éclairage et à l'expression faciale. Selon les études menées par Moses et. Al. (1994), les variations entre les images d'un même visage dues à l'éclairage et la direction de l'éclairage sont presque toujours plus grandes que les variations d'image en raison d'un changement d'identité faciale. La figure 3.4 illustre clairement cela.

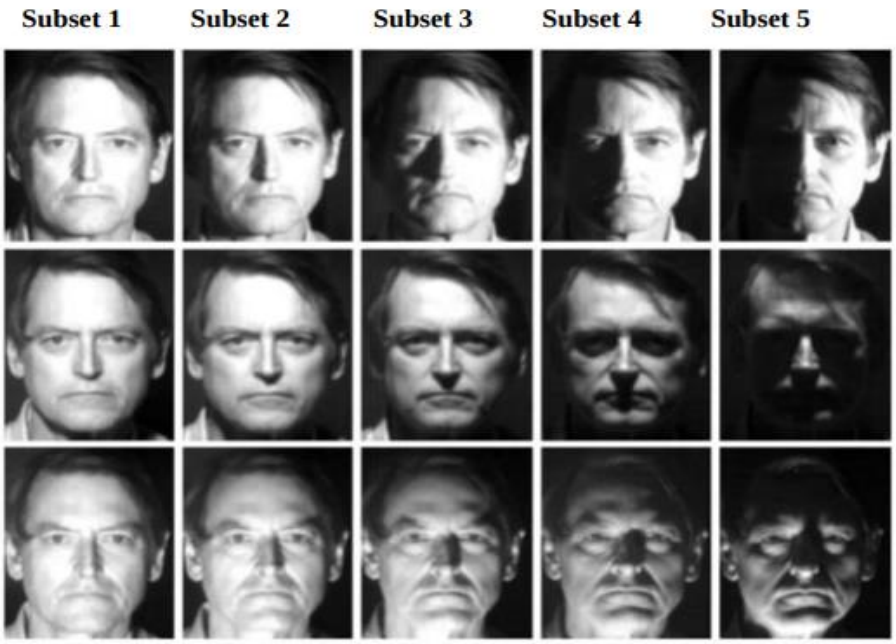


Figure 3.4: La même personne vue dans différentes conditions d'éclairage peut apparaître de façons différentes.

Fisher eyes

Belhumeur et. Al. En 1994 a proposé la méthode de Fisher faces en utilisant le PCA et L'analyse discriminante linéaire de Fisher pour produire une matrice de projection sous-spatiale qui est très similaire à celle de la méthode faces propres. C'est l'un des plus réussis méthodes de reconnaissance faciale largement utilisées. L'approche Fisher faces profite des informations intra-classe pour minimiser les

variations au sein de chaque classe, mais maximiser la séparation des classes, le problème des variations dans les mêmes images car différentes conditions d'éclairage peuvent être surmontées. La méthode de Fisher faces capte les informations discriminatoires mieux que les faces propres. Cependant, Fisher face nécessite plusieurs images d'entraînement pour chaque visage, de sorte qu'il ne peut pas être appliqué dans la reconnaissance où un seul exemple d'image par personne est disponible pour formation. La figure 3.5 montre quelques exemples de faces propres et de faces de Fisher et comment les Fisher face capturent mieux les informations discriminatoires que la méthode des faces propres.

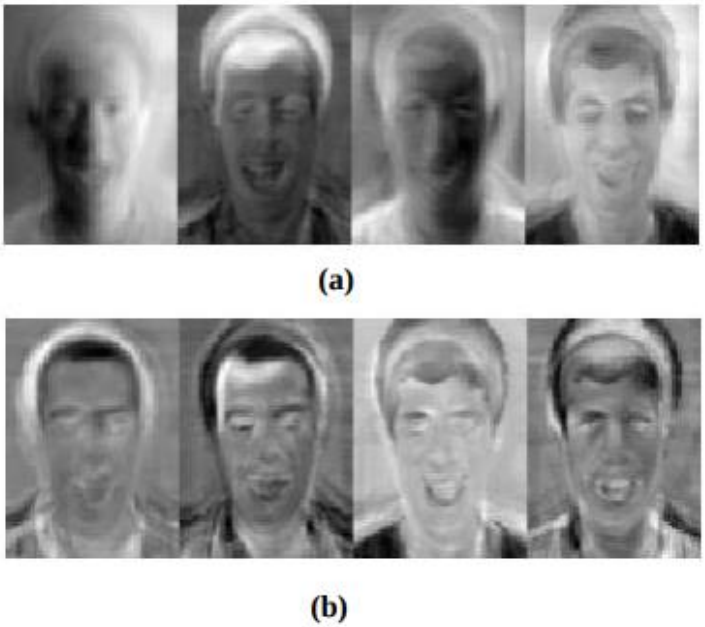


Figure 3.5: Un échantillon de faces propres qui montre différents modèles d'éclairage

(a) Un échantillon de faces propres montre la tendance des principaux composants à capturer des variations importantes de l'ensemble de formation telles que la direction de l'éclairage,

(b) L'échantillon correspondant de Fisher faces montre la capacité de ne pas tenir compte de ces facteurs sans rapport avec la classification.

Les modèles cachés de Markov

Avant de passer à la description d'un modèle de Markov caché, voyons d'abord ce qu'est un modèle de Markov. Le modèle de Markov, aussi appelé Chaîne de Markov, est un modèle statistique composé d'états et de transitions. Une transition matérialise la possibilité de passer d'un état à un autre.

Dans le modèle de Markov, les transitions sont unidirectionnelles une transition de l'état A vers état B ne permet pas d'aller de l'état B vers l'état A. Tous les états ont des transitions vers tous les autres états, y compris vers eux-mêmes. Chaque transition est associée à sa probabilité d'être empruntée et cette probabilité peut éventuellement être nulle.

Voici la représentation d'un modèle de Markov simple avec 2 états :

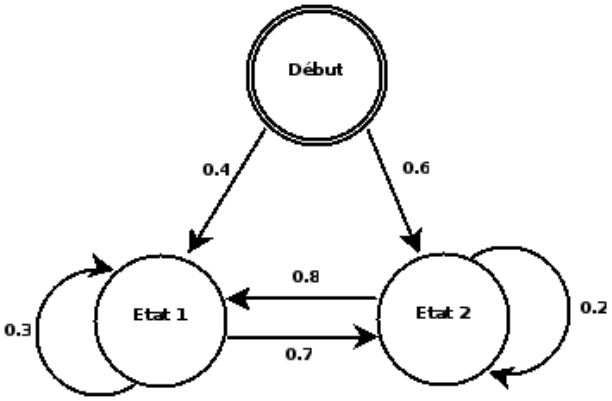


Figure 3.6: Modèle de Markov à 2 états.

Un modèle de Markov caché est basé sur un modèle de Markov, sauf qu'on ne peut pas observer directement la séquence d'états : les états sont cachés. Chaque état émet des "observations" qui, elles, sont observables. On ne travaille donc pas sur la séquence d'états, mais sur la séquence d'observations générées par les états.

Pour comprendre, reprenons le modèle de Markov représenté plus haut et transformons le en modèle de Markov caché :

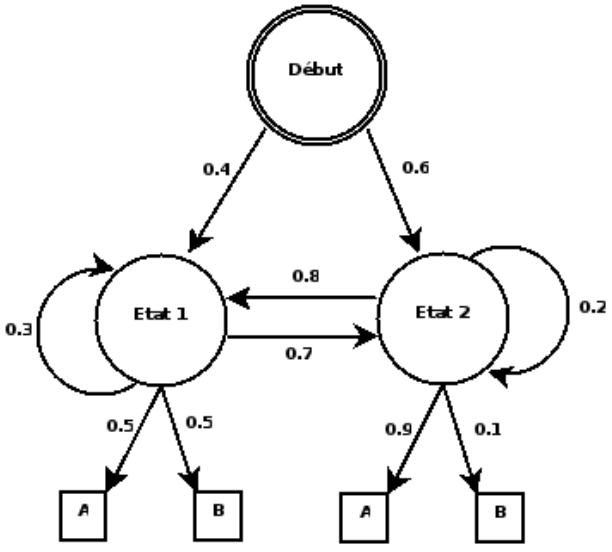


Figure 3.7: Modèle de Markov caché.

Il existe plusieurs algorithmes pour réaliser l'apprentissage des paramètres d'un HMM. On peut citer notamment : L'algorithme de Baum-Welch est un cas particulier de l'algorithme espérance maximisation et utilise l'algorithme Forward-Backward. Il permet de ré-estimer les paramètres de manière itérative.

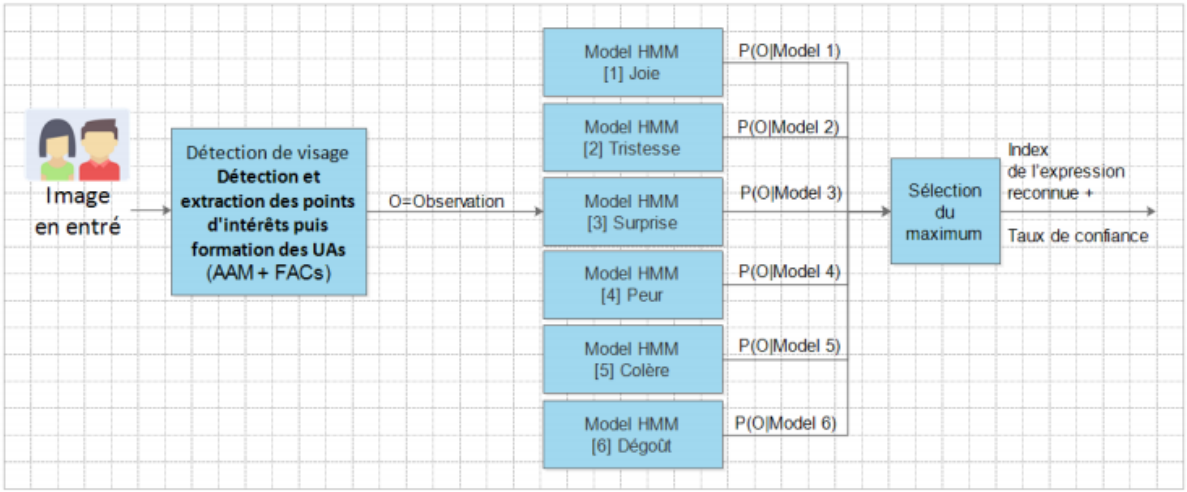


Figure 3.8: Descriptif du fonctionnement du classificateur HMM.

Une approche alternative au HMM 2D a été proposée par [Othman & Abounasr, 2000]. Ces auteurs proposent un système HMM 2D à faible complexité (LC2D HMM) pour la reconnaissance faciale. Le but de cette recherche est de construire un HMM 2D complet mais avec une complexité réduite. Le défi est de tirer parti d'une structure HMM 2D complète, mais sans la complexité totale qu'implique un modèle 2D sans contrainte. Leur modèle est implémenté dans le domaine compressé 2D DCT avec des blocs de 8 × 8 pixels sans chevauchement pour maintenir la compatibilité avec les images JPEG standard. Le LC2D HMM est basé sur une hypothèse clés:

L'état actif au bloc d'observation Bk, l ne dépend que des voisins immédiats verticaux et horizontaux, Bk − 1, l et Bk, l − 1.
D'un point de vue mathématique, cette hypothèse est équivalente à un modèle de Markov de second ordre, nécessitant une matrice de transition 3D.Les états actifs

aux 2 blocs d'observation situés dans des emplacements de voisinage anti-diagonaux, Bk − 1, l et Bk, l − 1 sont statistiquement indépendant étant donné l'état

Actuel. Cette hypothèse permet de séparer la matrice de transition d'état 3D en deux matrices de transition 2D distinctes, pour les transitions horizontales et verticales. Cela diminue considérablement la complexité du modèle. Cette topologie de modèle à faible complexité et la numérisation d'images sont illustrées dans la figure 3.9.

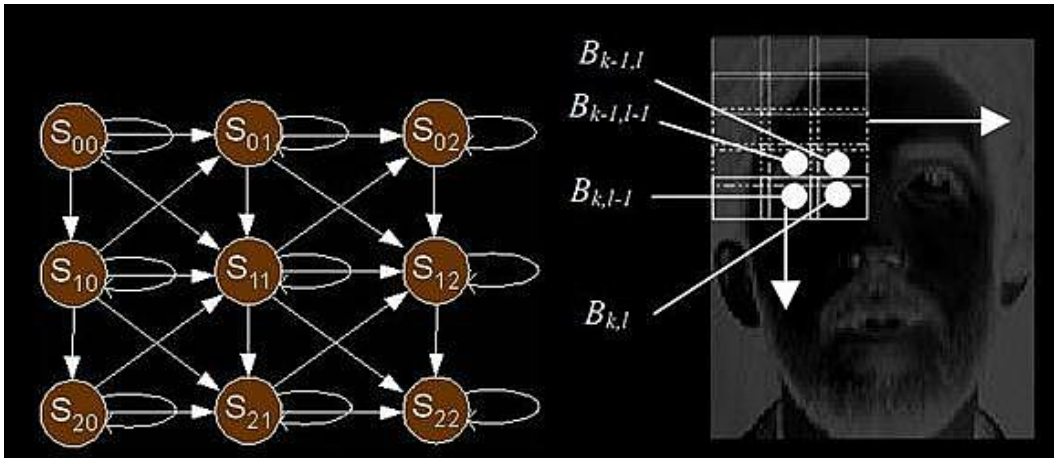


Figure 3.9: Balayage de l’image avec le Model de Markov.

3.3.2 Approche basée sur les caractéristiques

L'approche basée sur les caractéristiques de la reconnaissance faciale est née du fait que l'identification du visage est possible même avec un peu de détails faciaux. Il est présumé que la configuration géométrique globale du visage est suffisante pour la reconnaissance. Selon la façon dont les détails du visage sont définis (c'est souvent aussi simple que les yeux, le nez, et la bouche), un vecteur caractéristique est lié à ce détail.

Ce vecteur caractéristique, dont les composants définissent un espace dans lequel toutes les images de visages peuvent être placées, définit de façon unique le visage. Le but de toute reconnaissance basée sur les fonctionnalités est d'établir la description optimale de cet espace défini par un vecteur caractéristique. Les premiers travaux effectués sur la reconnaissance automatique du visage étaient principalement basés sur techniques basées sur les fonctionnalités. Dans ces approches, un visage est représenté comme une collection de certaines

caractéristiques abstraites. En règle générale, dans ces méthodes, les fonctionnalités locales telles que les yeux, le nez et la bouche sont d'abord extraits et leurs emplacements et les statistiques locales sont introduit dans un classificateur structurel. Les mesures structurelles comprennent des paramètres tels que rapports des distances, des angles et des zones entre les éléments élémentaires tels que les yeux, modèles de nez, de bouche ou de visage tels que la largeur et la longueur du nez. Les modèles statistiques standards des techniques de reconnaissance sont ensuite utilisés pour faire correspondre les visages en utilisant ces mesures. Les approches basées sur les fonctionnalités sont principalement classées en approches général et Elastic Bunch Graph Matching et une description détaillée des deux approches sont données ci-dessous.

3.3.3 Approches générales

Les approches générales de la reconnaissance faciale basée sur les fonctionnalités sont concernées en utilisant les informations préalables du visage pour trouver les caractéristiques du visage local. Alternativement, une autre approche générale consiste à trouver des géométries locales importantes du visage qui correspondent aux caractéristiques locales des visages. Il y a beaucoup de caractéristiques géométriques sur la base des points. Les entités géométriques peuvent être générées par des segments, des périmètres et les zones de certaines figures formées par les points. La figure 3.10 illustre comment dans le visage les points et la distance entre eux sont utilisés dans la reconnaissance faciale basée sur les entités [23].

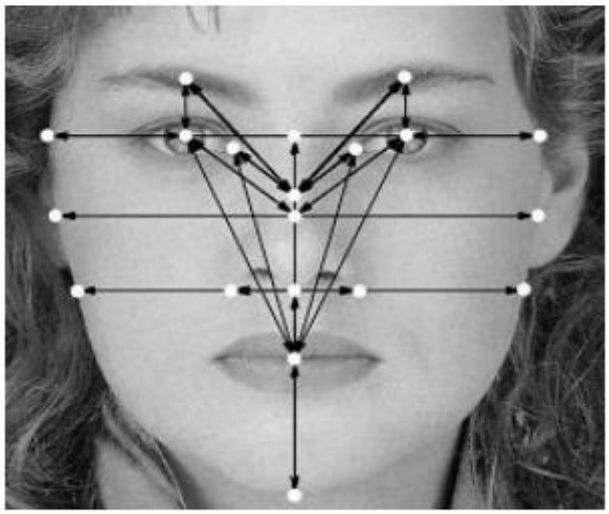


Figure 3.10: Certains points faciaux et distances entre eux utilisés dans la reconnaissance faciale [23]

L'une des premières tentatives de ce type a été réalisée par le Dr. Takeo Kanade, qui a utilisé des méthodes simples de traitement d'image pour extraire un vecteur de 16 paramètres faciaux qui étaient des rapports de distances euclidienes, de zones et d'angles, et utilisaient une simple mesure de distance pour l'appariement pour atteindre une performance de pointe de 75% sur une base de données de 20 personnes différentes utilisant 2 images par personne (une pour référence et une pour les tests). [23],

Elastic Bunch Graph Matching

Un autre système de reconnaissance faciale basé sur les fonctionnalités est le Système Elastic de graphe EBGM (Elastic Bunch Graph Matching), basé sur les Architectures de lien (DLA). La DLA commence par extraire d'abord les fonctionnalités locales, puis effectue la correspondance à l'aide des informations globales décrivant la géométrie de la connexion d'importantes caractéristiques locales. L'EBGM reconnaît les visages en faisant correspondre l'ensemble de sondes représenté par le saisissez des graphes de face, dans l'ensemble de la galerie qui est représenté comme le graphe de face du modèle. Le concept de nœuds étant fondamental pour la correspondance de graphe élastique utilisé. Essentiellement, chaque nœud du graphe de face d'entrée est représenté par un point caractéristique du visage. Par exemple, les nœuds représentent un œil, un nez et le concept continu

de représenter les autres traits du visage. Par conséquent, les nœuds du graphique de face d'entrée sont interconnectés pour former un graphique comme des données structuré adaptée à la forme du visage comme illustré à la figure 3.11. En revanche, le graphique du visage du modèle représente l'ensemble de galeries utilisant uniquement un modèle face graphique pour représenter l'ensemble de la galerie. Cependant, le visage du modèle conceptuel peut être considéré comme un certain nombre de graphes de face d'entrée empilés sur les uns au-dessus des autres et concaténés pour former un graphique de face de modèle, sauf que cela est appliqué à l'ensemble de galeries au lieu de l'ensemble de sondes [23].

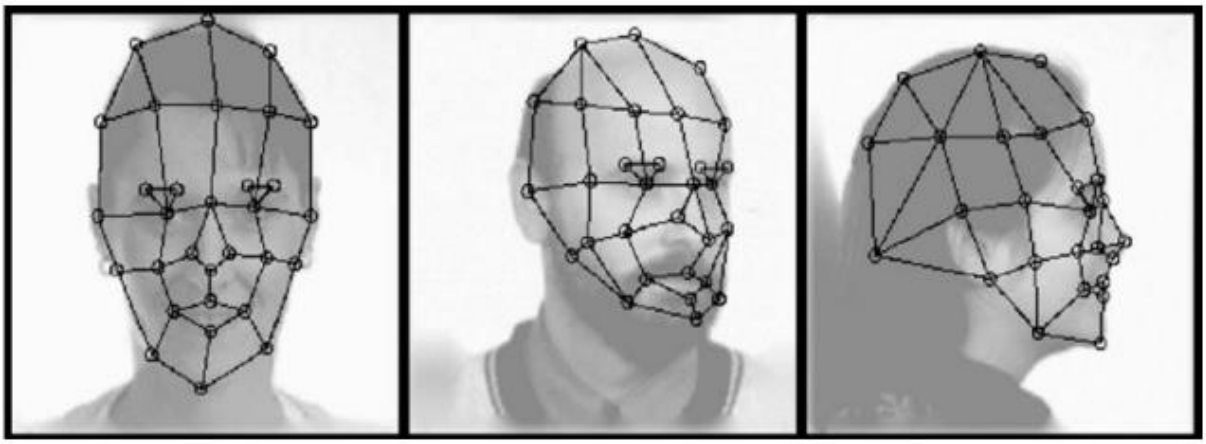


Figure 3.11: Graphe du visage [23]

Par conséquent, cela permettrait le regroupement des mêmes types des traits du visage de différents individus. Par exemple, les yeux de différents individus pourraient être regroupés pour former le point caractéristique oculaire du modèle graphique du visage et le nez de différents individus peuvent être regroupé pour former le point caractéristique du nez pour le graphique de face du modèle. Une illustration du visage du modèle graphique est montrée dans la figure 3.12 Étant donné la définition du graphique de face d'entrée et du graphique de face du modèle, déterminer l'identité pour le graphique de face d'entrée est d'atteindre la plus petite distance par rapport au graphique de face du modèle pour une face de galerie particulière. La distance est déterminée par la mesure de similitude des nœuds pour les graphiques de face d'entrée avec le modèle graphique du visage [23].

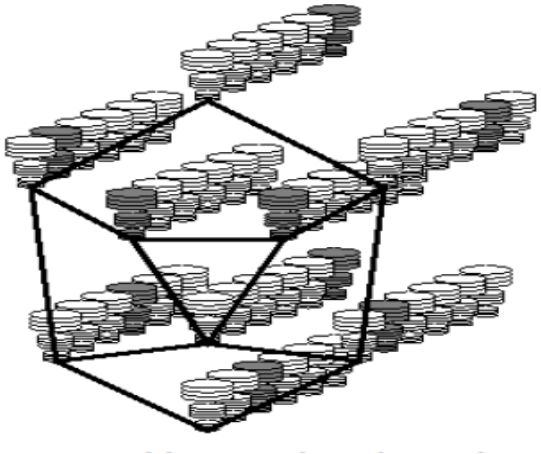


Figure 3.12: Modélisation de Face Graphe (Wiskott et. al. 1999) [23]

Méthodes hybrides

Le système de perception humaine reconnaît les visages en utilisant les caractéristiques faciales locales et toutes les informations sur la région du visage. La méthode hybride est plus similaire aux systèmes de perception de l'homme car il est influencé à la fois par des méthodes basées sur les caractéristiques et holistiques. Il utilise à la fois les caractéristiques locales et toute la région du visage pour reconnaître un visage, il doit également utiliser un système de reconnaissance de machine en combinant des Eigenface et d'autres modules propres comme Eigeneyes, Eiganmouth et Eigennose. De plus, une légère amélioration par rapport aux Eigenface holistiques ou des modules propres basés sur une correspondance structurelle peuvent être obtenus. L'approche hybride est plus efficace et efficiente en présence de non pertinents des données. Les facteurs clés affectant les performances dépendent des fonctionnalités sélectionnées et les techniques utilisées pour les combiner. Les méthodes basées sur les fonctionnalités et holistiques ne

manque pas d'inconvénients. La méthode basée sur les fonctionnalités est parfois gravement affectée par le problème de précision, car la localisation précise des caractéristiques est une étape très importante.

Sur d'autre part, l'approche holistique utilise des algorithmes plus complexes exigeant plus de temps de formation et exigences de stockage. Approche hybride pour la reconnaissance faciale qui exploite à la fois les caractéristiques locales discriminantes et globales par une fusion pour la représentation et la reconnaissance des visages est développé par Jamuna, Kanta Sing [24]. Les images du visage entier sont divisées dans un certain nombre de sous-régions qui ne se chevauchent pas pour extraire le discriminant local caractéristiques. Les traits discriminants globaux sont extraits de tout le visage image. Les méthodes PCA et Fisher's Linear Discriminant sont appliquées pour un vecteur d'entités fusionné pour extraire des entités discriminantes de dimension inférieure.

3.4 Algorithmes de reconnaissance

3.4.1 Analyse en composantes principale

L'analyse en composantes principale (ACP) est une méthode de la famille de l'analyse des données, qui consiste à transformer des variables liées entre elles en nouvelles variables sont nommées «composante principales». Elle permet au praticien de réduire l'information en un nombre de composantes plus limité que le nombre initial de variable. Il s'agit d'une approche à la fois géométrique et statistique. Lorsqu'on veut alors compresser un ensemble de N variables aléatoires, les n premiers axes de l'ACP sont un meilleur choix, du point de vue de l'inertie ou la variance. L'ACP a été de nouveau développé et formalisée dans les années 30 par Harold hotline la puissance mathématique de l'économiste et statisticien américain le conduisait aussi à développer l'analyse canonique, Les champs d'application sont aujourd'hui multiples, allant de la biologie à la recherche économique et sociale, et plus récemment le traitement d'images [25]. L'ACP est majoritairement utilisée pour :

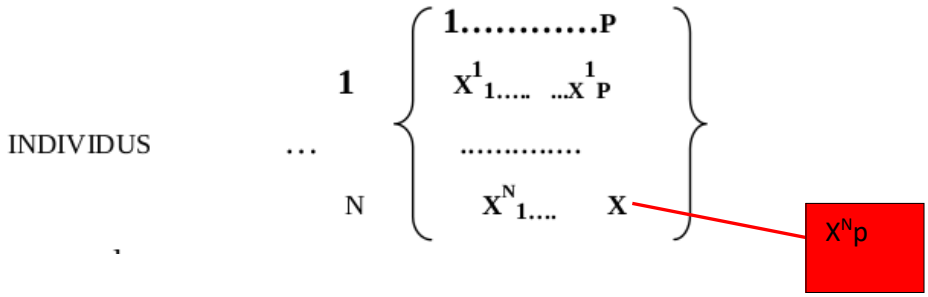
1. Décrire et visualiser des données.

2. Les décorreler dans la nouvelle base, constituée des nouveaux axes, les variables ont une corrélation nulle [25].

Formulation de l'ACP :

La mise en œuvre mathématique de l'ACP peut être divisée en 6 étapes principales.

Etape1 : préparation des données : les données que l'on doit traiter par l'ACP sont stockées dans un tableau X de type Individus/variables de la forme :



Nous avons alors :

• p variables, représentées en colonnes.

• N individus, représentées les lignes.

• Des valeurs prises par chaque variable, pour chaque individu notée :

$$(X_j^i)\ (1 \leq i \leq N,\ 1 \leq j \leq P) \text{ avec } \forall (i,j),\ X_j^i \in \mathbb{R}$$

Etape 2 : Calcul de la matrice des coefficients de corrélations :

Dans cette étape, nous calculons la matrice de corrélation des données contenues dans la table à X_{cr} notée «Corr ».

Etape 3 : Calcul des valeurs et vecteurs propres de la matrice de corrélations.

Les valeurs et vecteurs propres de la matrice « Corr » sont les facteurs utilisés pour construire les composantes principales [25].

Etape 4 : Classement des vecteurs propres dans l'ordre décroissants des valeurs propres associées, On dispose alors des facteurs dans l'ordre décroissant de la quantité d'information qu'ils expliquent. Il est également possible d'exprimer en pourcentage l'importance de chacun, afin de visualiser l'importance relative des composantes principales <<U>> la matrice dont les colonnes sont les valeurs propres de « Corr » classées par ordre décroissant de leurs *valeurs propres* associées [25].

Etape 5 : Calcul de la matrice des composantes principales : la matrice appelée matrice des composantes principales, notée CP, est celle qui contient les coordonnées des individus dans l'espace formé par celles-ci. Elle est calculée de la façon suivante : $CP = X_{cr}. U$

Etape 6 : Représentations graphiques : le but de l'ACP étant de résumer une situation donnée, la représentation graphique est la phase finale et la plus importante de ce processus, car elle permet d'avoir rapidement un aperçu de ce que le calcul numérique ne peut pas fournir [25].

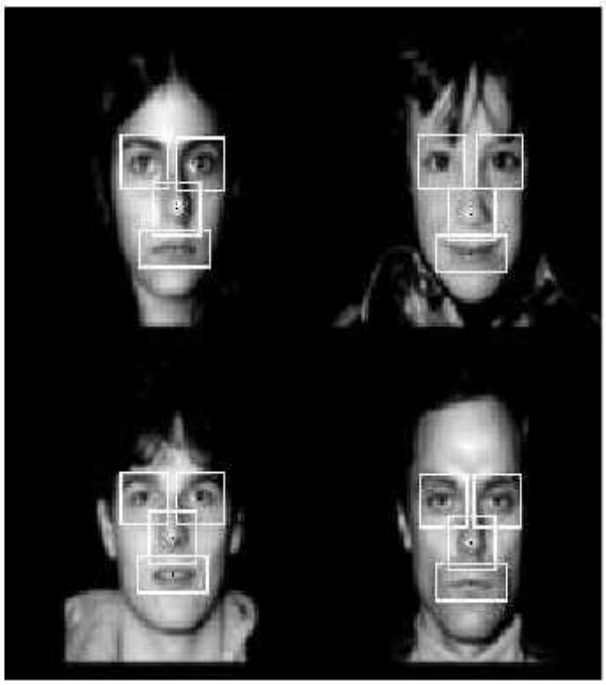


Figure 3.13: Caractéristiques autour desquelles est réalisée une ACP [25]

3.4.2 **Analyse des composants indépendante (ICA)**

L'ICA est une technique permettant d'extraire des variables statistiquement indépendantes d'un mélange d'entre elles (Bell et Sejnowski, 1995). L'ICA a été appliqué avec succès à de nombreux problèmes différents tels que l'analyse des données MEG et EEG, trouvant des facteurs cachés dans les données financières et la reconnaissance faciale [25].

La technique ICA vise à trouver une transformation linéaire pour les données d'entrée en utilisant une base aussi statistiquement indépendante que possible. Ainsi, l'ICA peut être considéré comme une généralisation de l'analyse en composantes principales (ACP).Ce qui suggère qu'elles peuvent conduire à des représentations plus précises. La figure 3.14 montre la différence entre les images de base PCA et ICA [25].

Figure 3.14: Quelques images de base originales (à gauche), PCA (au centre) et ICA (à droite) pour la base de données de Yale Face [25]

Dans le contexte de la reconnaissance faciale, l'ICA s'est avéré produire de meilleurs résultats que ceux obtenus avec l'ACP.

3.4.3 Analyse discriminante linéaire

La LDA (Linear Discriminant Analysis) est une amélioration de PCA (Principal Component Analysis). PCA n'utilise pas le concept de classe, contrairement à LDA. Les images de visages d'une même personne sont traitées ici comme de la même classe. PCA et LDA réduisent toutes les dimensions. Ils transforment les images en

tant que vecteur vers un nouvel espace avec de nouveaux axes. Les axes de projection choisis par PCA peuvent ne pas fournir un bon pouvoir de discrimination. LDA essaie de trouver des axes de projection, tels que les classes sont mieux séparées [25].

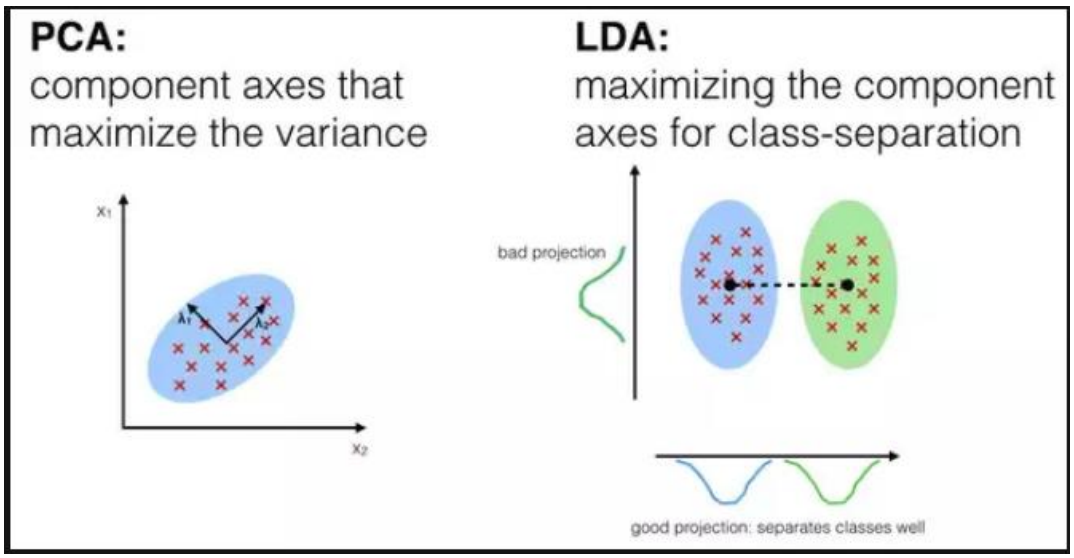


Figure 3.15: Comparaison des vecteurs PCA et LDA [25]

L'analyse discriminante linéaire ou la méthode Fisher faces surmonte les limites de la méthode des faces propres en appliquant le critère discriminant linéaire de Fisher. Ce critère essaie de maximiser le rapport du déterminant de la matrice de diffusion inter-classe des échantillons projetés. Les discriminants de Fisher regroupent les images de la même classe et séparent les images de différentes classes. Les images sont projetées de l'espace dimensionnel (où N2 est le nombre de pixels de l'image) vers l'espace dimensionnel C (où C est le nombre d'images classées) [25].

Par exemple, considérons deux ensembles de points dans un espace à deux dimensions qui sont projetés sur une seule ligne. Selon la direction de la ligne, les points peuvent être mélangés ou séparés. Les discriminants de Fisher trouvent la ligne qui sépare le mieux les points. Contrairement à la méthode PCA qui extrait les fonctionnalités pour représenter au mieux les images de visage. La méthode LDA essaie de trouver le sous-espace qui discrimine le mieux les différentes classes de visages [25].

Algorithme

- Soit l’Image carré avec Largeur = Hauteur = N.
- M est le nombre d'images dans la base de données.
- P est le nombre de personnes dans la base de données.

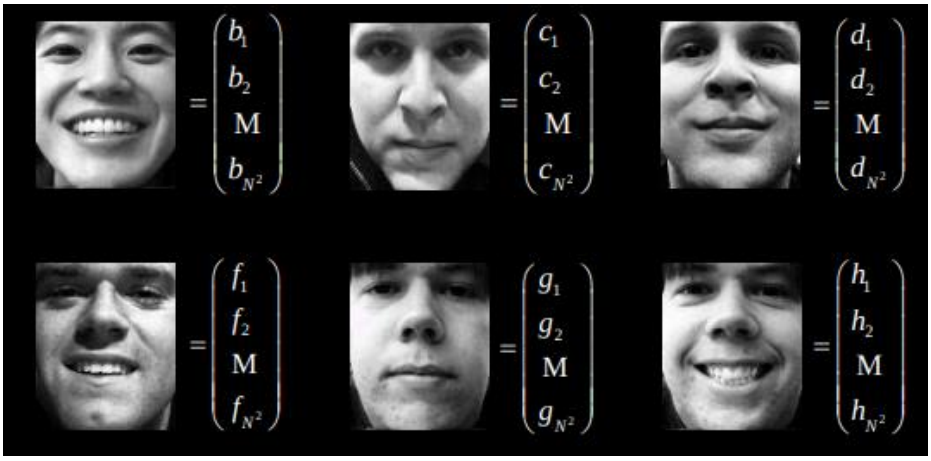


Figure 3.16: Base de données. [25]

Nous calculons la moyenne de tous les visages, on calcule le visage moyen de chaque personne, on soustraire des visages d'entraînement .Après nous construisons des matrices de dispersion (W).

Les colonnes de W sont des vecteurs propres de Sw.Sb. Nous devons calculer l'inverse de SW, multiplier les matrices. Nous devons calculer les vecteurs propres, pour classer le visage on doit le projeter sur l'espace LDA et exécuter un classificateur de voisin le plus proche.

3.4.4 Réseaux de neurones convolutifs (CNN)

La reconnaissance faciale est le processus de reconnaissance du visage. Cela a été un outil d'interaction homme-machine en raison de son utilisation dans les systèmes de sécurité, contrôle d'accès, vidéo surveillance et même il est également utilisé dans les réseaux sociaux comme Facebook. Après un développement rapide de l'intelligence artificielle, le visage la reconnaissance a une fois de plus attiré l'attention en raison de sa nature non intrusive et comme il s'agit de la principale

méthode d'identification pour l'homme lorsqu'elle est comparée à d'autres types des techniques biométriques [20].

Les méthodes basées sur l'apprentissage profond (Deep learning) peuvent extraire les traits de visage plus compliqués. Ce dernier s'est avéré excellent révélant des structures complexes dans des données de haute dimension, Il aborde le problème de l'apprentissage des représentations hiérarchiques avec un seul algorithme ou quelques algorithmes et a principalement battu des records en reconnaissance image, traitement du langage naturel, sémantique, segmentation et de nombreux autres scénarios du monde réel. Il existe différentes approches de l'apprentissage profond comme le Réseau Neuronal Convolutif (CNN), auto-encodeur empilé, et Deep Belief Network (DBN) [20].

Les réseaux composés de plusieurs couches. Les CNN sont constitués de filtres ou noyaux ou neurones qui ont des poids ou des paramètres d'apprentissage. Chaque filtre prend des entrées, effectue une convolution de l'architecture CNN typique peut être vue comme le montre la Figure 3.17.

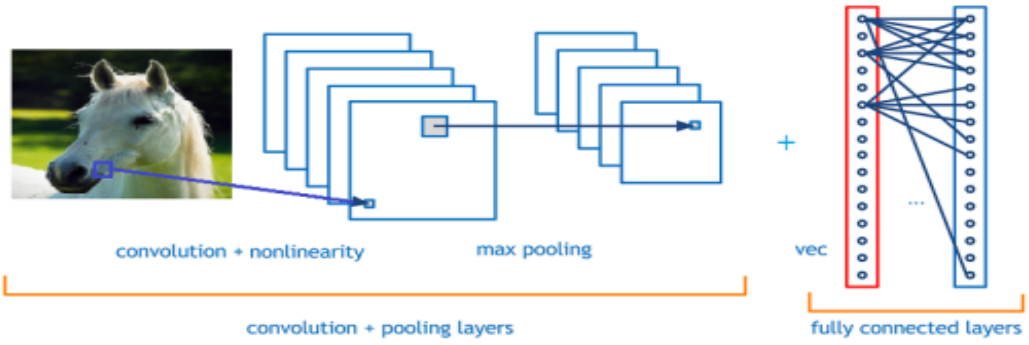


Figure 3.17: Conception traditionnelle de réseaux de neurones convolutifs [20]

- Couche Convolution

La couche Convolution constitue la pierre angulaire d'un réseau de convolution qui effectue la plupart des calculs. Le but principal de la couche de convolution est d'extraire des caractéristiques des données d'entrée qui sont une image. La convolution préserve la relation spatiale entre les pixels en apprenant les caractéristiques de l'image à l'aide de petits carrés d'image d'entrée. L'image d'entrée est alambiquée en utilisant un ensemble de neurones. Cela produit une

carte d'entités ou une carte d'activation dans l'image de sortie et après que les cartes d'entités sont alimentées en entrée elles sont transmises à la prochaine couche convolutionnelle [20].

- **Couche de mise en commun**

La couche de mise en commun réduit la dimensionnalité de chaque activation carte, mais continue d'avoir les informations les plus importantes. Les images d'entrée sont divisées en un ensemble de non-chevauchement rectangle. Chaque région est sous-échantillonnée par un non-linéaire fonctionnement tel que la moyenne ou le maximum. Cette couche atteint une meilleure généralisation, convergence plus rapide, robuste à la traduction et la distorsion et est généralement placée entre convolution couches [20].

- **Couche ReLU**

Le ReLU (abréviation de Unité Linéaire Rectifiée). Cette fonction, appelée aussi fonction d'activation non saturante, augmente les propriétés non linéaires de la fonction de décision et de l'ensemble du réseau sans affecter les champs récepteurs de la couche de convolution. Elle est une opération non linéaire qui comprend des unités utilisant le redresseur. C'est une opération élémentaire qui signifie qu'il est appliqué par pixel et reconstitue tous les négatifs les valeurs dans la carte d'entités par zéro. Afin de comprendre comment le ReLU fonctionne, nous supposons qu'il y a une entrée de neurone étant donné x et à partir de ce que le redresseur est défini comme f (x) = max (0, x) dans la littérature pour les réseaux de neurones [20].

- **Couche entièrement connectée**

Le terme de couche entièrement connectée (Full Connected Layer) fait référence à ce que chaque filtre dans la couche précédente est connecté à chaque filtre dans la couche suivante. La sortie de la convolution, du regroupement et de ReLU constituent des modes de réalisation de caractéristiques de haut niveau de l'image d'entrée.

Le traitement d'image basée sur des réseaux de neurones convolutifs doit collecter un grand nombre d'images pour que l'ordinateur apprenne. Ce sujet prendra beaucoup d'images, il recadrées l'image des parties non pertinentes du visage. À

ce stade, les images collectées ont été rognées et redimensionnées. Ensuite, toutes les images sont assemblées et cousues dans le dataset, Chaque ligne de l'image finale du dataset présente les images redimensionne et en niveaux de gris de deux personnes La figure ci-dessous fait l'objet de l'ensemble de données de visage à former [20].

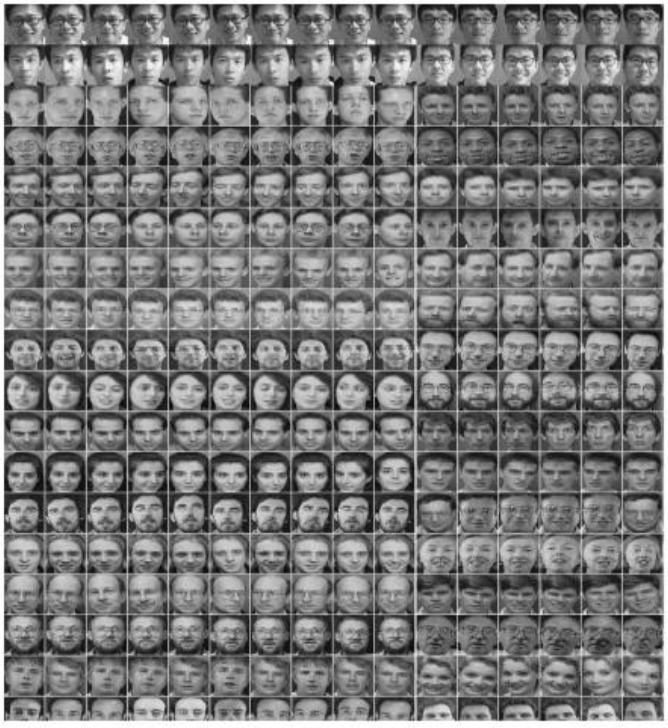


Figure 3.18: Exemple de base de données d’entraînement [20]

Les données d'entraînement sont ensuite transmises au CNN de cette manière:

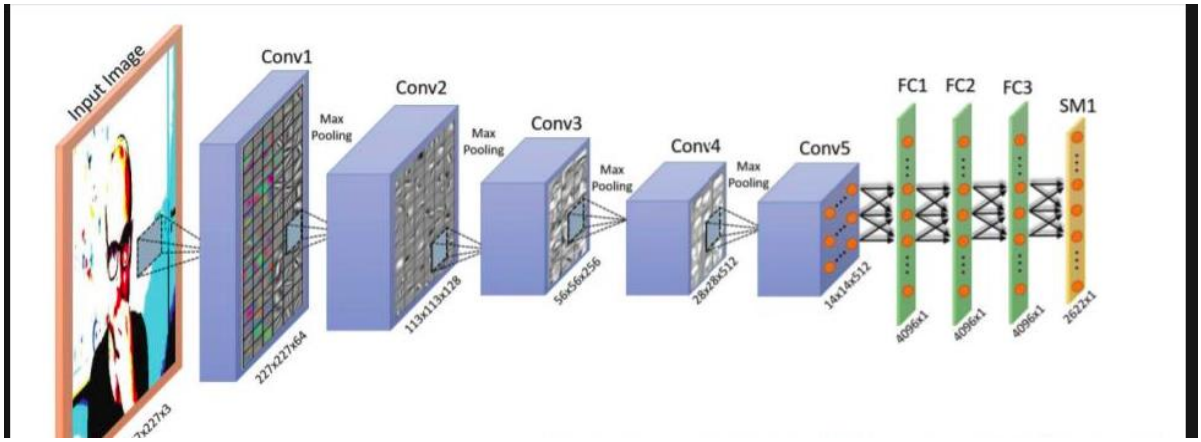


Figure 3.19: Schéma de base d'un réseau de neurones convolutifs dans la reconnaissance faciale [20]

3.4.5 Local Binary Patterns Histogramme (LBPH)

Introduction

Les modèles binaires locaux (LBP) sont des descripteurs non paramétriques dont le but est de résumer efficacement la structure locale des images. Ces dernières années, il a suscité un intérêt croissant dans de nombreux domaines du traitement d'image et de la vision artificielle, et a montré son efficacité dans un certain nombre d'applications, en particulier pour l'analyse de l'image du visage, y compris des tâches aussi diverses que la détection du visage, reconnaissance faciale, analyse des expressions faciales, classification,…etc.

L'opérateur LBP d'origine étiquette les pixels d'une image avec des nombres décimaux, appelés modèles binaires locaux ou codes LBP, qui codent la structure locale autour de chaque pixel. Il procède ainsi, comme illustré sur la figure 3.20. Chaque pixel est comparé à huit voisins dans un carré de 3x3 en soustrayant la valeur du pixel central. Les valeurs strictement négatives résultantes sont codé avec 0 et les autres avec 1. Un nombre binaire est maintenu en concaténant tous ces codes binaires dans le sens horaire direction à partir de celle en haut à gauche qui correspondant à la valeur décimale utilisée pour l'étiquetage. Les nombres binaires dérivés sont appelés modèles binaires locaux ou codes LBP [26].

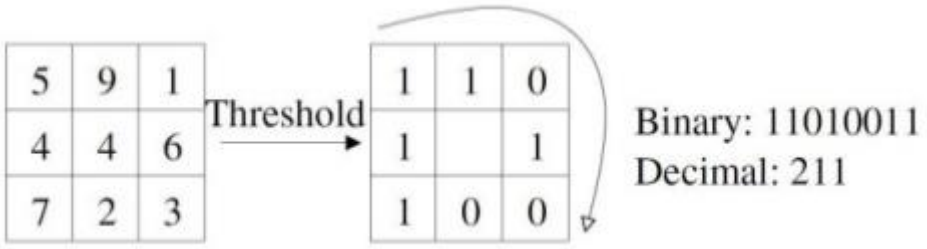


Figure 3.20: Exemple de l'opérateur LBP de base [26]

Une limitation de l'opérateur LBP de base est que son petit carré de 3x3 ne peut pas capturer les caractéristiques de structures dominantes à grande échelle. Pour gérer la texture à différentes échelles, l'opérateur est ensuite été généralisé pour utiliser les quartiers de différentes tailles. Un quartier local est défini comme un ensemble de points d'échantillonnage régulièrement espacés sur un cercle centré aux pixels étiquetés et les points d'échantillonnage qui ne tombent pas dans les pixels sont interpolés en utilisant une interpolation bilinéaire, permettant ainsi à tout rayon et tout nombre de points d'échantillonnage de figurer sur le quartier. La figure 3.21 montre quelques exemples d'opérateur LBP étendu, où la notation (P, R) désigne un voisinage de P points d'échantillonnage sur un cercle de rayon de R [26].

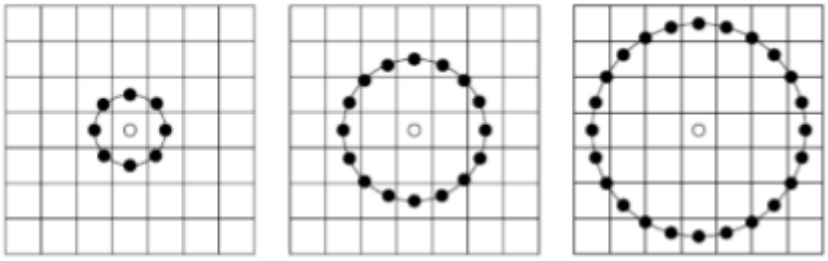


Figure 3.21: Exemples de l'opérateur LBP étendu: la circulaire (8, 1), (16, 2),et (24, 3) quartiers [26]

En utilisant le LBP combiné avec des histogrammes, nous pouvons représenter les images de visage avec un simple vecteur de données. Comme le LBP est un descripteur visuel, il peut également être utilisé pour des tâches de reconnaissance faciale, comme on peut le voir dans l'explication étape par étape suivante. Le LBPH utilise 4 paramètres:

Rayon: le rayon est utilisé pour construire le motif binaire local circulaire et représente le rayon autour du pixel central. Il est généralement défini sur 1 pixel [26].

Voisins: le nombre de points d'échantillonnage pour construire le motif binaire local circulaire. Plus vous incluez d'échantillons, plus le coût de calcul est élevé. Il est généralement défini sur 8 [26].

Grille-X: le nombre de cellules dans le sens horizontal. Plus il y a de cellules, plus la grille est fine, plus la dimensionnalité du vecteur d'entités résultant est élevée. Il est généralement défini sur 8 [26].

Grilles-Y: le nombre de cellules dans le sens vertical. Plus il y a de cellules, plus la grille est fine, plus la dimensionnalité du vecteur d'entités résultant est élevée. Il est généralement défini sur 8 [26].

Formation de l'algorithme:

Tout d'abord, nous devons former l'algorithme. Pour ce faire, nous devons utiliser un ensemble de données avec les images faciales des personnes que nous voulons reconnaître. Nous devons également définir un ID (il peut s'agir d'un numéro ou du nom de la personne) pour chaque image, de sorte que l'algorithme utilisera ces informations pour reconnaître une image d'entrée et vous donner une sortie. Les images d'une même personne doivent avoir le même ID. Avec l'ensemble de formation déjà construit, voyons les étapes de calcul du LBPH [26].

- Première étape

La première étape de calcul du LBPH est de créer une image intermédiaire qui décrit l'image originale d'une meilleure manière, en mettant en évidence les caractéristiques faciales. Pour ce faire, l'algorithme utilise un concept de fenêtre coulissante, basé sur les paramètres rayon et voisins. L'image ci-dessous montre cette procédure:

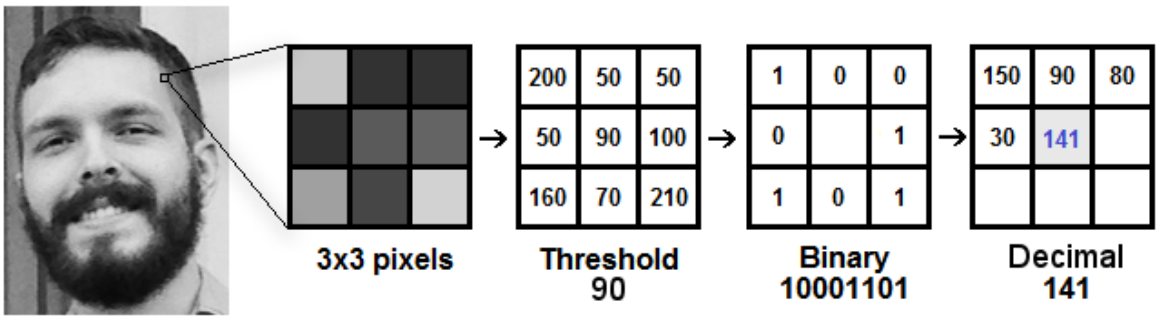


Figure 3.22: Conversion de l'image à niveau de gris en point LBP [26]

Supposons que nous avons une image faciale en niveaux de gris. On peut obtenir une partie de cette image sous forme d'une fenêtre de 3x3 pixels. Elle peut également être représenté comme une matrice 3x3 contenant l'intensité de chaque pixel (0 ~ 255). Ensuite, nous devons prendre la valeur centrale de la matrice à utiliser comme seuil. Cette valeur sera utilisée pour définir les nouvelles valeurs des 8 voisins Pour chaque voisin de la valeur centrale (seuil), nous fixons une nouvelle valeur binaire. Nous définissons 1 pour des valeurs égales ou supérieures au seuil et 0 pour des valeurs inférieures au seuil [26].

Maintenant, la matrice ne contiendra que des valeurs binaires (en ignorant la valeur centrale). Nous devons concaténer chaque valeur binaire de chaque position de la matrice ligne par ligne en une nouvelle valeur binaire (par exemple 10001101).

Ensuite, nous convertissons cette valeur binaire en une valeur décimale et la définissons sur la valeur centrale de la matrice, qui est en fait un pixel de l'image d'origine. A la fin de cette procédure (procédure LBP), nous avons une nouvelle image qui représente mieux les caractéristiques de l'image originale [26].

- **Deuxième étape**

Elle consiste à extraire les histogrammes. Maintenant, en utilisant l'image générée à la dernière étape, nous pouvons utiliser les paramètres Grille X et Grille Y pour diviser l'image en plusieurs grilles, comme on peut le voir dans l'image suivante:

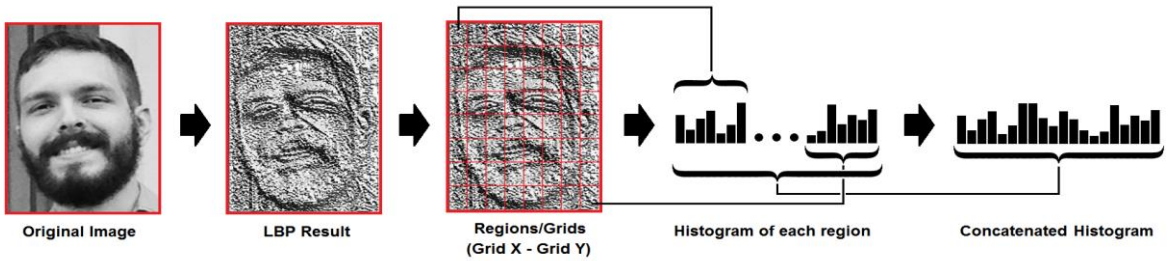


Figure 3.23: Histogramme à partir de l'image LBP [26]

Comme nous avons une image en niveaux de gris, chaque histogramme (de chaque grille) ne contiendra que 256 positions (0 ~ 255) représentant les occurrences de chaque intensité de pixel. Ensuite, nous devons concaténer chaque histogramme pour créer un nouvel histogramme plus grand. En supposant que nous ayons des grilles 8x8, nous aurons 8x8x256 = 16,384 positions dans l'histogramme final. L'histogramme final représente les caractéristiques de l'image originale de l'image [26].

- **Dernière étape**

Est la réalisation de la reconnaissance faciale: dans cette étape, l'algorithme est déjà formé. Chaque histogramme créé est utilisé pour représenter chaque image du jeu de données d'apprentissage. Donc, étant donné une image d'entrée, nous effectuons à nouveau les étapes pour cette nouvelle image et crée un histogramme qui représente l'image.

Donc, pour trouver l'image qui correspond à l'image d'entrée, il suffit de comparer deux histogrammes et de renvoyer l'image avec l'histogramme le plus proche. Nous pouvons utiliser différentes approches pour comparer les histogrammes (calculer la distance entre deux histogrammes), par exemple: distance euclidienne, chi carré, valeur absolue, etc. Dans cet exemple, nous pouvons utiliser la distance euclidienne (qui est bien connue) basée sur la formule suivante:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

Ainsi, la sortie de l'algorithme est l'ID de l'image avec l'histogramme le plus proche. L'algorithme doit également renvoyer la distance calculée, qui peut être utilisée comme mesure de «confiance» [26].

3.5 Conclusion

Dans ce chapitre nous avons effectué une étude approfondie sur différentes méthodes de reconnaissance faciale. Après une analyse détaillée, il a été révélé que l'ACP est la technique la mieux adaptée lorsque la dimension des caractéristiques est plus élevée pour les images de visage originales, tandis que la méthode des caractéristiques d'image aux faces propres fonctionne bien pour la reconnaissance frontale du visage. Parmi les méthodes de reconnaissance faciale, les plus populaires sont les réseaux de neurones convolutifs, le LBPH et le LDA. Ces méthodes fournissent de meilleurs résultats lorsque la dimension de l'image est inférieure à 150*150 pixel ou plus. En outre, il est suggéré que les méthodes PCA, CNN et LDA doivent encore être améliorées afin de trouver des résultats plus satisfaisants.

Chapitre 4 : Réalisation et tests

4.1. Introduction

Dans ce chapitre, nous présentons la partie réalisation de notre projet. Il s'agit d'un système d'identification biométrique de personnes basé sur la méthode LBPH. Nous présentons aussi l'environnement du travail, matériel et logiciel ainsi que les résultats obtenus sur notre propre base de données sous différentes conditions.

4.2. Environnement matériel

Nous avons utilisé :

Un ordinateur portable Dell inspiron 15 3000series avec les caractéristiques suivantes :
Kali linux 2016.1
Processeur Intel(R) Pentium(R) 3558U @ 1.70GHz
Mémoire installée (RAM) 4,00Go
Type du système : Système d'exploitation 64 bits, processeur x64

Figure 4.1 ordinateur portable Dell inspiron 15 3000series

- **Camera speed Link SNAPPY SMART WEBCAM**

L'appareil photo utilisé est une webcam Speed Link avec les détails techniques suivants :

Référence produit: SL-6825-BK-01

Webcam USB

Résolution VGA 640×480 pixels

Résolution photo de 8 mégapixels au plus (interpolée)

Zoom numérique 4×

Fonction Face Tracking

Enregistrement vidéo jusqu'à 30 fps

Support pour bureau et fixation pour écran dépliable

Entièrement pivotable sur 360°

Mise au point manuelle de 30 mm à l’infini

Câble de 1,2m

Figure 4.2: Webcam Speed link

4.3. Environnement logiciel

4.3.1. PYTHON (version 3.7.3)

Le système a été réalisé avec le langage de programmation Python, qui est un langage de programmation interprété, multi-paradigme et multi plate formes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par

ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

Le langage Python est placé sous une licence libre et fonctionne sur la plupart des plates-formes informatiques, des smartphones aux ordinateurs centraux, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

Figure 4.3: Logo de Python.

4.3.2. Qu'est-ce que Python ?

Python est un langage de programmation interprété, orienté objet et de haut niveau avec une sémantique dynamique. Ses structures de données intégrées de hauts niveaux, combinés à une frappe dynamique et à une liaison dynamique, le rendant très attrayant pour le développement rapide d'applications, ainsi que pour une utilisation en tant que langage de script ou de collage pour connecter les composants existants entre eux. La syntaxe simple et facile à apprendre de Python met l'accent sur la lisibilité et réduit donc le coût de maintenance du programme. Python prend en charge les modules et les packages, ce qui encourage la modularité du programme et la réutilisation du code. L'interpréteur Python et la bibliothèque standard étendue sont disponibles gratuitement sous forme source ou binaire pour toutes les principales plates-formes et peuvent être librement distribués.

4.3.3 Open cv (version 4.1.1)

OpenCV (Open Source Computer Vision Library) est une bibliothèque de logiciels de vision par ordinateur et d'apprentissage automatique open source. OpenCV a été construit pour fournir une infrastructure commune pour les applications de vision par ordinateur et pour accélérer l'utilisation de la perception des machines dans les produits commerciaux. Étant un produit sous licence BSD, OpenCV facilite l'utilisation et la modification du code par les entreprises.

La bibliothèque possède plus de 2500 algorithmes optimisés, qui comprennent un ensemble complet d'algorithmes de vision par ordinateur et d'apprentissage machine classiques et à la pointe de la technologie. Ces algorithmes peuvent être utilisés pour détecter et reconnaître des visages, identifier des objets, classer les actions humaines dans des vidéos, suivre les mouvements de caméra, suivre des objets en mouvement, extraire des modèles 3D d'objets, produire des nuages de points 3D à partir de caméras stéréo. Ils peuvent aussi assembler des images pour produire une haute résolution de l'image d'une scène entière, trouver des images similaires dans une base de données d'images, supprimer les yeux rouges des images prises au flash, suivre les mouvements des yeux, reconnaître les paysages et établir des marqueurs pour les recouvrir de réalité augmentée, etc. OpenCV compte plus de 47 000 utilisateurs communauté et nombre de téléchargements estimé à plus de 18 millions. La bibliothèque est largement utilisée dans les entreprises, les groupes de recherche et les organismes gouvernementaux.

Figure 4.4: Logo d'OpenCV

4.4. PyQt

PyQt est un ensemble de liaisons Python v.2 et v.3 pour le cadre d'application Qt de qui fonctionne sur toutes les plates-formes prises en charge par Qt, y compris Windows, OS X, Linux, iOS et Android. Les liaisons sont implémentées comme un ensemble de modules Python et contiennent plus de 1 000 classes.

Qt est plus qu'une boîte à outils GUI. Il comprend des abstractions de sockets réseau, des threads, Unicode, des expressions régulières, des bases de données SQL, SVG, OpenGL, XML, un navigateur Web entièrement fonctionnel, un système d'aide, un cadre multimédia, ainsi qu'une riche collection de widgets GUI.

4.5. SQLite (version 3)

SQLite est une bibliothèque en langage C qui implémente un petit moteur de base de données SQL complet, rapide, autonome, haute fiabilité et complet. il est le moteur de base de données le plus utilisé au monde. SQLite est intégré à tous les téléphones mobiles et à la plupart des ordinateurs et est intégré à d'innombrables autres applications que les gens utilisent tous les jours.

Figure 4.5: Logo de SQLite

4.6. Étapes de réalisation

Pour réaliser ce projet, nous étions obligé de passer par quatre principales étapes qui sont la création de bases de données, détection de visage, entrainement des données et reconnaissance de visage. Ces étapes sont expliquées dans les sous sections suivantes

4.6.1 Étape 1: Base de données

La première chose à faire est de créer la base de données pour stocker les visages. Il est très important de créer d'abord la base de données car elle contiendra tous les identifiants et noms d'utilisateurs. Cette base de données SQL hébergera toutes les informations utilisateur. Sqlite3 a été utilisé et le fichier de base de données est automatiquement généré dans le dossier du répertoire de travail.

```
In [5]: import sqlite3
        conn = sqlite3.connect('database.db')
        c = conn.cursor()
        sql = """
        DROP TABLE IF EXISTS users;
        CREATE TABLE users (
                  id integer unique primary key autoincrement,
                  name text
        );
        """
        c.executescript(sql)
        conn.commit()
        conn.close()

In [ ]:
```

4.6.2 Étape 2: Détection et acquisition des visages

Dans la deuxième étape, l'algorithme de détection des visages sera utilisé pour détecter tous les visages humains dans l'image que la caméra fournira au système. Une fois les visages détectés avec l'algorithme des cascades de Haar, ils seront recadrés pour s'adapter uniquement aux caractéristiques les plus pertinentes du visage. Ces caractéristiques comme les yeux, la bouche, la forme du visage seront convertis en une image à niveaux de gris et numérisées à un format binaire utilisant les valeurs de pixels des images capturées (vecteur d'histogramme). L'algorithme organisera ensuite ces valeurs binaires en des vecteurs et chaque personne aura son propre vecteur associé à son identifiant. Une fois cette étape terminée, toutes les

images capturées de la personne seront stockées dans un ensemble de données (Dataset) avec un identifiant spécifique qui sera également stocké dans la base de données SQL. Chaque personne aura 50 images de taille 226x226 pixels dans un dossier appelé Dataset qui sera généré automatiquement.

Figure 4.6: Exemple d'image du dataset 226*226 pixel

Pour assurer un meilleur taux de reconnaissance, les images de la base de données subissent un prétraitement pour améliorer leurs qualités. Le rôle principal de ce prétraitement est de normaliser les conditions d'éclairage pour toutes les images. Pour améliorer le contraste local de l'image nous avons utilisé le CLAHE. Après le prétraitement nous obtenons 50 nouvelles images. A la fin de cette étape, chaque personne aura 100 images dans la base de données (50 en niveaux de gris et 50 prétraitées avec le CLAHE).

4.6.3 Étape 3: Entraînement des données générées

À cette fin, des histogrammes de motifs binaires locaux (LBPH) ont été envisagés et cette méthode effectue la reconnaissance en comparant les visages à reconnaître avec un ensemble d'apprentissage de visages connus. Dans cette étape, l'algorithme LPBH prendra toutes les images de la base de donnée et formera un modèle de reconnaissance, ce processus est expliqué en détail dans le chapitre 3. Une fois l'entraînement est terminé, un fichier yaml sera créé et chaque personne de la base de données aura son propre vecteur de reconnaissance. Ce fichier sera utilisé ultérieurement pour la reconnaissance.

YAML signifie (Yaml Ain't Markup Language), et cette technologie de format de fichier est utilisée dans les documents. Ces documents sont enregistrés au format texte brut et sont ajoutés avec l'extension .yml. La sérialisation efficace des données était l'objectif principal des développeurs du format .yml, car il permet aux utilisateurs de créer des fichiers .yml avec un contenu indépendant de tout langage de balisage particulier. Ces fichiers .yml peuvent également être lus par n'importe quel éditeur de texte développé pour créer, ouvrir et modifier des fichiers en texte brut, que ce soit un logiciel d'édition de texte pour les systèmes basés sur Microsoft Windows comme Microsoft Notepad et Microsoft WordPad, ou pour les plates-formes Mac comme le logiciel Apple TextEdit. Les bibliothèques YAML peuvent également être utilisées pour intégrer le format YML dans plusieurs langages de programmation. Ces langages de programmation peuvent être Ruby, C / C ++, Python, Perl, PHP, Java, Javascript, AJAX, C # et ainsi de suite.

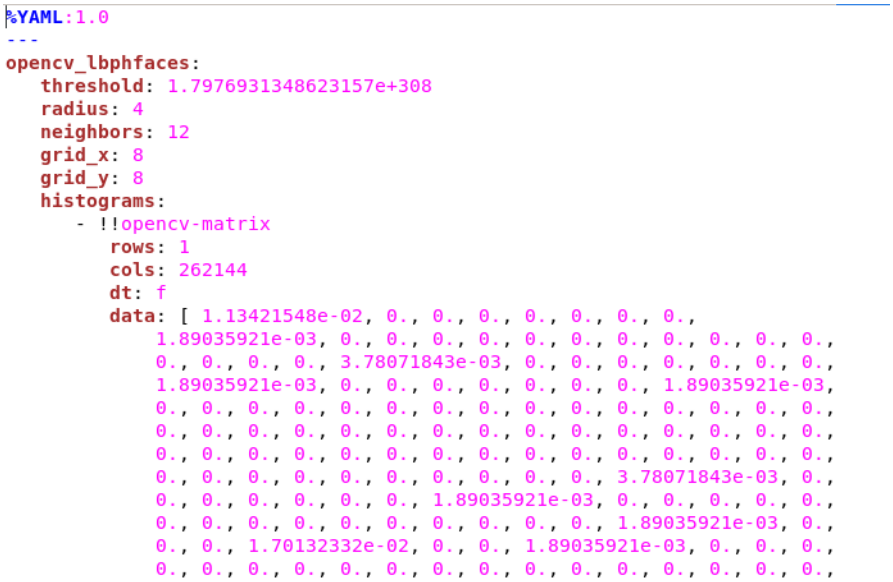


Figure 4.7: Image du contenue du fichier YAML

4.6.4 Étape 4: Reconnaissance faciale

Dans cette étape, une personne se présente devant la caméra, la webcam fait l'acquisition et envois l'image au système qui à son tour normalise l'image, détecte le visage, applique la LBPH et fait la classification. Si cette personne est dans la base

de données, notre système de reconnaissance fera une "prédiction" en renvoyant son identifiant et un index, montrant à quel point le système est confiant de la correspondance de cette personne. Dans cette étape, nous avons présenté à notre système différentes personnes pour faire des tests, certaines parmi lesquelles sont dans la base et d'autres ne le sont pas. Notre caméra a été utilisée pour capturer un flux en direct d'un emplacement spécifique, si une personne se trouve dans ce cadre, son visage sera détecté et un processus de reconnaissance sera déclenché pour voir si cette personne est dans la base de données. L'algorithme LBPH utilisera le fichier YML pour faire la prédiction. Si la confiance est supérieure à 65%, cette personne sera étiquetée avec l'identifiant approprié. Sinon l'algorithme retournera un message disant que la personne n'est pas dans la base de données.

Après avoir testé de nombreux paramètres pour le LBPH, nous avons décidé de choisir:

Rayon = 4

Le rayon est utilisé pour construire le motif binaire local circulaire et représente le rayon autour du pixel central. Il est généralement défini sur 1.

MinNeighbohrs(voisins minimum) = 16 pixels

Le MinNeighbohrs est le nombre de points d'échantillonnage pour construire le motif binaire local circulaire. Gardez à l'esprit que plus vous incluez d'échantillons, plus le coût de calcul est élevé. Il est généralement défini sur 8.

Taille de la Cellule = 16 * 16 pixels

Grille X: le nombre de cellules dans le sens horizontal. Plus il y a de cellules, plus la grille est fine, plus la dimensionnalité du vecteur d'entités résultant est élevée. Il est généralement défini sur 8.

Grilles-Y: le nombre de cellules dans le sens vertical. Plus il y a de cellules, plus la grille est fine, plus la dimensionnalité du vecteur d'entités résultant est élevée. Il est généralement défini sur 8.

Threshold (seuil) = 1.8e+308

Le seuil appliqué dans la prédiction. Si la distance au voisin le plus proche est supérieur au seuil, cette méthode renvoie -1.

4.7 Amélioration

Pour des raisons de conditions d'éclairage, nous avons ajouté une étape supplémentaire dans l'algorithme qui consiste à ajouter un filtre d'égalisation d'histogramme adaptatif à contraste limité (CLAHE). Juste après avoir extrait les vecteurs d'histogramme des images. La méthode a trois paramètres:

4.7.1 Taille du bloc :

La taille de la région locale autour d'un pixel pour lequel l'histogramme est égalisé. Cette taille doit être supérieure à la taille des entités à conserver.

4.7.2 Bacs d'histogramme :

Le nombre de cases d'histogramme utilisées pour l'égalisation de l'histogramme. L'implémentation fonctionne en interne avec une résolution d'octets, donc les valeurs supérieures à 256 ne sont pas significatives. Cette valeur limite également la quantification de la sortie lors du traitement d'images 8 bits gris ou 24 bits RVB. Le nombre de cases d'histogramme doit être inférieur au nombre de pixels d'un bloc.

4.7.3 Pente maximale

Limite l'étirement du contraste dans la fonction de transfert d'intensité. Des valeurs très grandes permettront à l'égalisation de l'histogramme de faire ce qu'elle veut, c'est-à-dire un contraste local maximal. La valeur 1 donnera l'image d'origine.

De plus en addition on a ajouté deux autres filtres pour une luminosité extrême et une obscurité extrême.

Image après histogramme

CLAHE

Luminosité extrême

Obscurité extrême

Figure 4.8: Application des filtres.

4.8 Interface de l’application

L'interface suivante a été réalisée en utilisant PyQt et python, il s'agit d'une interface utilisateur pour une application de reconnaissance faciale. Nous pouvons ajouter des utilisateurs à la base de données, former des modèles et bien sûr reconnaître les visages.

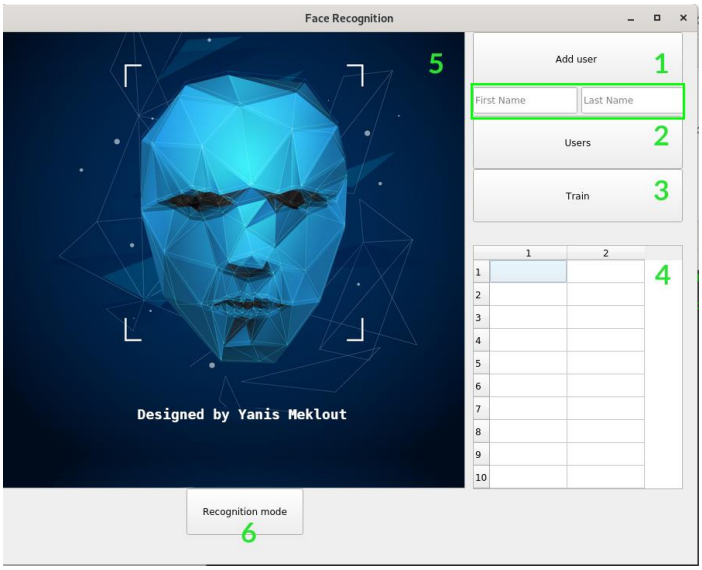


Figure 4.9: Interface utilisateur graphique.

Section 1: bouton pour ajouter des utilisateurs.

Section 2: bouton pour voir la liste des utilisateurs de la base de données.

Section 3: bouton pour former les modèles de base de données.

Section 4: liste de tous les utilisateurs.

Section 5: section d'affichage.

Section 6: bouton pour démarrer la reconnaissance.

Ici, nous allons ajouter un utilisateur à la base de données :

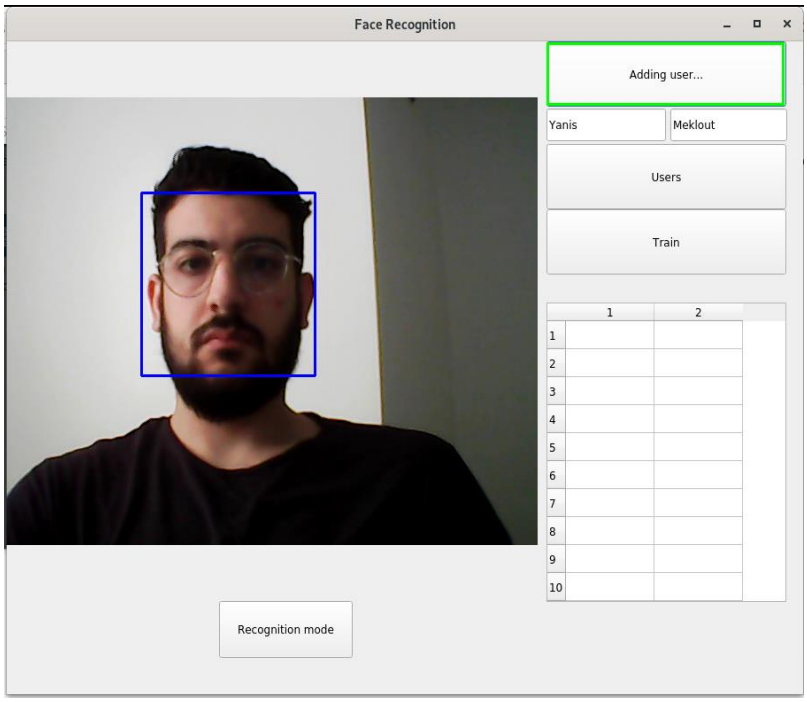


Figure 4.10: Ajout d’un utilisateur.

Pour vérifier la liste des utilisateurs et si l'utilisateur actuel a été ajouté avec succès, nous cliquons sur le bouton « Users » :

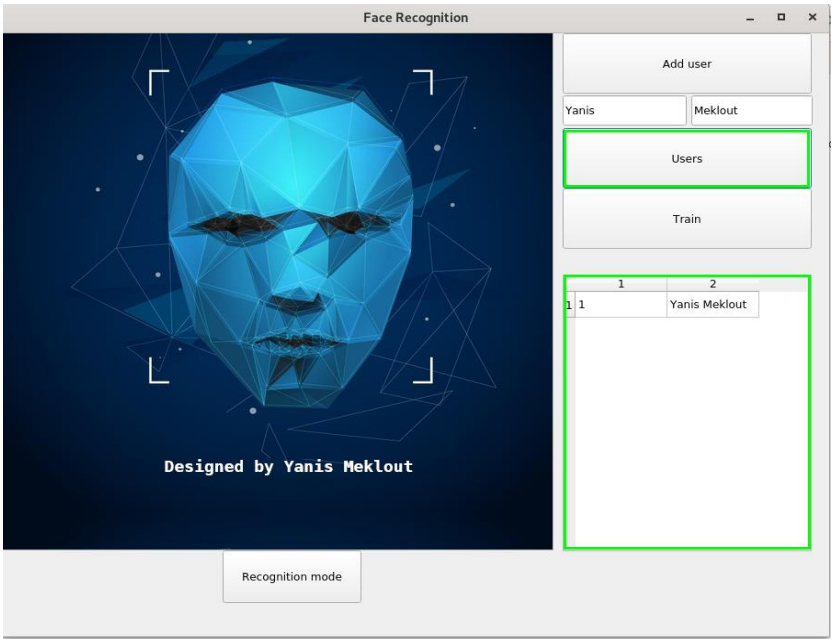


Figure 4.11: Liste des utilisateurs.

Après avoir collecté les informations utilisateur, nous formons le modèle de prédiction en cliquant sur le bouton « Train » :

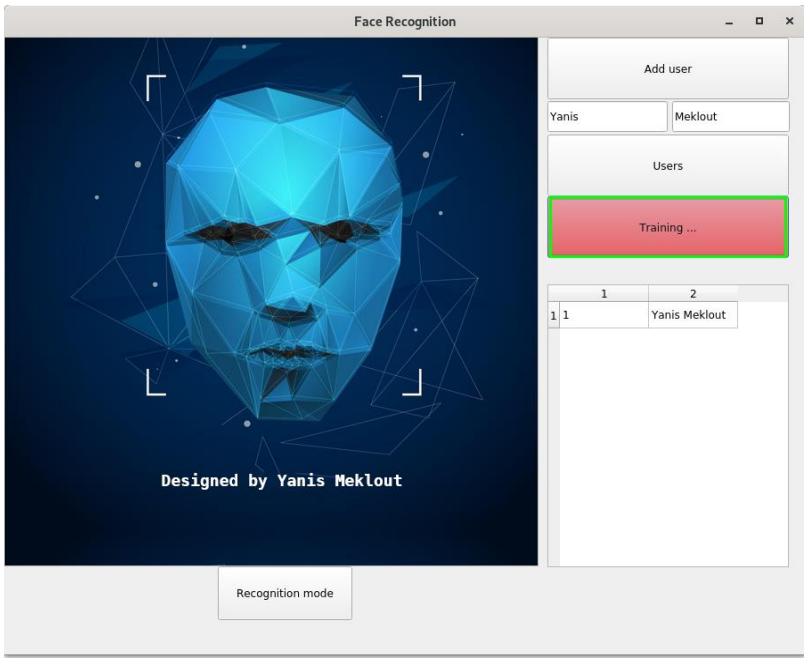


Figure 4.12: Entraînement de la base de données.

Reconnaissance faciale de la personne :

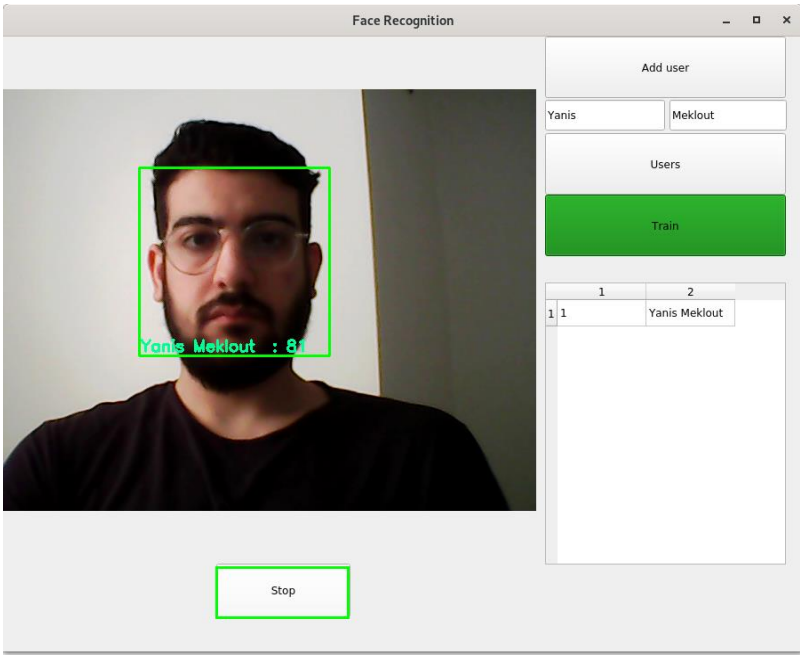


Figure 4.13: Résultat de la reconnaissance.

4.9 Tests et résultats

Pour les tests, nous avons utilisé 20 personnes et testé le logiciel sur 3 emplacements différents.

L'ensemble de données se compose de 15 personnes enregistrées dans la base de données, 10 enregistre avec camera live et 5 à l'aide de photos. Chaque personne dans la base de données a 200 images (50 en niveaux de gris, 50 en niveaux de gris CLAHE, 50 niveaux de gris avec luminosité extrême, 50 niveaux de gris avec obscurité extrême). De plus, 5 personnes ne sont pas enregistrées dans la base de données, 3 personnes avec camera live et 2 utilisant des images seront ajoutées dans la phase de test.

4.9.1. Test:

Le test a été effectué manuellement. si la personne testée positive avec le nom correct et qu'il est dans la base de données, il sera étiqueté comme un vrai positif, si la personne teste négatif et n'est pas réellement dans la base de données, elle sera étiquetée comme un vrai négatif, si la personne est reconnue par l'algorithme comme quelqu'un d'autre, elle sera étiquetée comme un faux positif, et enfin si la personne n'est pas reconnu mais est en fait dans la base de données, il sera étiqueté comme un faux négatif.

Nous avons fait ce test particulier pour différents paramètres de l'algorithme dans les 3 endroits différents:

Paramètre:

Taille des images = 226 x226

Paramètres LBPH:

Rayon = 1,4,8.

MinNeighbohrs = 8,12,16 pixels

Seuil = la destination d'entraînement

Taille des cellules = 8 x 8,16 x 16

4.9.2. Résultats

Emplacement	Vrai positif	Vrai négatif	Faux positif	Faux négatif
Emplacement 1	100% (15/15)	0 % (0/15)	0 % (0/15)	0%(0/15)
Emplacement 2	93.33% (14/15)	0 %(0/15)	6.66% (1/15)	0% (0/15)
Emplacement 3	86.66% (13/15)	0 %(0/15)	6.66 % (1/15)	6.66% (1/15)

Du tableau ci-dessus, nous pouvons voir que dans le premier emplacement, le taux de vrai positif était de 100%, donc l'ensemble des données de test a été identifié avec succès et sans erreur.

Pour le deuxième emplacement, nous pouvons voir que 14/15 personnes de l'ensemble de données de test ont été identifiées avec succès mais une personne a était identifiée comme faux positif (identifié comme quelqu'un d'autre).Et pour le 3ème emplacement 86,66% de l'ensemble de données a été identifié avec succès avec une personne étant un faux positif et un faux négatif signifiant qu'il était dans la base de données mais qu'il n'était pas reconnu. A cause de la variation de lumière extrême dans le 3éme emplacement

4.10 Conclusion

Dans ce chapitre, nous avons présenté la mise en œuvre en temps réel du LBPH en utilisant Python, OpenCV, Pyqt. Les résultats obtenus étaient assez bons malgré les problèmes d'éclairage que nous avons essayé de minimiser le plus possible. L'efficacité et la précision du système étaient acceptables pour un taux d'identification correct de 93.33% avec un minimum de confiance de 72%.

Conclusion général

L’étude des systèmes biométriques, en particulier la détection et la reconnaissance du visage, nous conduit à conclure qu’ils sont très puissants en termes d’identification mais ils rencontrent certains problèmes, la difficulté des traitements et la complexité des algorithmes utilisés et le temps de calcul rendent le processus d’identification moins fiable relativement aux attentes des utilisateurs dans les différents domaines. Ceci représente, pour les chercheurs un défi à conquérir.

Le but de ce projet était de développer un système de reconnaissance faciale. Tous les objectifs ont été atteints, déterminant ainsi l'efficacité et la précision du système et l’algorithme LPBH. La précision du système était basée sur le taux de reconnaissance faciale et l'efficacité du système était déterminée par le temps de calcul.

Même si les systèmes de reconnaissance de machines actuels ont atteint un certain niveau de perfection, il existe néanmoins de nombreuses conditions d'application réelles qui limitent leurs bonnes performances. Nous tenons à signaler les principaux obstacles rencontrés lors de nos études qui se résume à ce qui suit : Luminosité et éclairage, déformations non-rigides, variations inter-personnes, micro-expression expression subtile (début d’une émotion). Sans oublié le « problème universel »

Qui est le temps de calcul ce qui nous incites à utiliser l’algorithme LBPH et python afin de projeter une éventuelle « parallélisassions » des traitements

Dans le futur, nous espérons améliorer nos travaux en se penchant sur les

Problèmes suivants:

1. La détection de visages humains de différentes vues (frontale, de profil,...)
2. Reconnaissance faciale en 3d.
3. Reconnaissance des émotions humaines.

Bibliographie

[1] : Nassim HAMITOUCHE, Zakaria SALMI, « Système d'identification biométrique de personnes par reconnaissance de l'iris », Ecole national Supérieure d'Informatique (ESI) Oued-Smar, Alger, Algérie ,2009

[2] : A.Belaid et Y.Belaid «reconnaissance des formes : méthodes et applications » .inter éditions, 1992

[3] : P.Faber «Exercices de reconnaissance des formes par ordinateur avec solutions commentées, programmes d'applications et rappels de cours ». Masson, Paris, 1989.

[4] : Commission d'accès à l'information du Québec, « La biométrie au Québec : Les enjeux », Juillet 2002

[5] : T.Shtonham « Pratical face recognition and verification with WISCAD . DANS H.D.Ellis,M.Jeeves, F.Newcombe et A.Young :Aspects of face processing », Martinus Nijhoff Publishers,pp 426-441, Dordrecht, 1986.

[6] : E.Diday, J.LEMAIRE, j.Pouget et F.Testu «Element d'analyse de données» Dunod, Paris, 1982.

[7] : J.Yang, D.Zhang, A.F.Frangi and J-Y.Yang, «Two Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition», IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.26, No.1, January 2004.

[8] : S. Z. Li, A. K. Jain «Handbook of Face Recognition», Springer, 2004

[9] : L. Bourdev et J. Brandt, « Robust Object Detection via Soft Cascade », CVPR, p. 236-243 , 2005.

[10] : Paul Viola and Michael Jones. «Robust real-time object detection», In Second international work shop on statistical and computation atheories of vision, Vancouver, Canada, July 13 2001. [Perlibakas 2004] V. Perlibakas, 2004. Distance measures for PCA-based face recognition, In Pattern Recognition Letters 25, 711-724.

[11] : M.Paisar, Y.Suchart « forensic science international », 2013.

[12] : Nazil Perveen, Darshan Kumar, Ishan Bhardwaj « An Overview on Template Matching Methodologies and its Applications», ID: 54957226, 2013 .

[13] : Armeni, I. Sener, O. Zamir, A. R. Jiang, H. Brilakis, I. Fischer «3D Semantic Parsing of Large-Scale Indoor Spaces», In Proceedings of the IEEE Conference ,2016.

[14] : Alain Treméau, Frank Boochs, Jean-Jacques Ponciano «Knowledge-based object recognition in Point Clouds and image data sets», 2017.

[15] : Ioan Buciu «Overview of face recognition techniques», 2008.

[16] : P. Phillips, P. Grother, R. Michaels, D. Blackburn, E.Tabassi, and M. Bone. «Face recognition vendor test», 2002.

[17] : Ivyarajsinh, N. Parmar Brijesh, B. Mehta January «Face Recognition Methods & Applications D», 2014.

[18] : O.DénizM, Castrillón, M.Hernández «Face recognition using independent component analysis and support vector machines, open overlay panel», 2003.

[19] : Divyansh Dwivedi « Face Recognition for Beginners», 2018.

[20] : Musab Coşkun, Ayşegül Uçar «Face Recognition Based on Convolutional Neural Network», DOI: 10.1109/MEES.2017.8248937 , 2017.

[21] : David Lowe "Newer journal paper IJCV 2004", University of British Columbia Initial paper ICCV 1999

[22] : Bay, H., Tuytelaars, T. and Van Gool « SURF: Speeded Up Robust Features», 2006.

[23] : Takeo Kanade, https://fr.wikipedia.org/wiki/Takeo_Kanade

[24] :Anindya Jana, N Basanta Singh, Jamuna Kanta Sing, Subir Kumar Sarkar « Face recognition using principal component analysis and RBF neural networks», 2008.

[25]: Harold Hotelling https://fr.wikipedia.org/wiki/Harold_Hotelling

[26]: T.AHONEN « Face recognition with local binary pattern». Computer Vision, ECCV, 2004.