

RÉPUBLIQUE ALGERIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de JIJEL



Faculté des Sciences Exactes et d'Informatique

Département d'informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de master

Option : *Intelligence Artificielle*

Thème

Etude et Construction d'un Modèle d'Apprentissage
de l'Apparence Pour la Reconnaissance d'Objets

Encadré par :

M^r. Mokhtar TAFFAR

Présenté par :

M^{elle}. Radja HANNOUF

M^{elle}. Imene BOURIDANE

Promotion 2015

Nous tenons avant tout à remercier *Allah* le tout puissant de nous avoir donné la force et la volonté pour achever ce modeste travail.

Nous tenons à remercier particulièrement nos parents; notre succès demeure de loin le fruit de leurs longues années de sacrifices et d'éducation.

Nos vifs remerciements, accompagnés de toute notre gratitude, vont tout d'abord à notre encadreur **M. TAFFAR Mokhtar**, pour nous avoir proposé ce sujet et dirigé dans notre travail ainsi pour sa disponibilité.

Nous remercions aussi **M. ZENNIR Mohammed Nadjib**, pour son aide précieuse, sa disponibilité et avis éclairés.

Nous remercions également les enseignants du département d'informatique pour tout le savoir qu'ils ont su nous transmettre durant ces dernières années.

Nous ne pouvons nous empêcher d'avoir une pensée pour ceux et celles qui ont répondu présents et nous ont offert leur soutien moral dans les moments difficiles et qui étaient à nos cotés pour partager avec nous les moments de joie.

Ce travail modeste est dédié :

À nos très chers parents qui n'ont pas cessé de nous encourager durant nos études, qu'Allah les bénisse;

À nos frères & sœurs;

À nos cousins et cousines et tous ceux que nous aimons;

À nos amis et nos camarades de l'Université de JIJEL, ainsi qu'à toutes les personnes qui nous ont aidé à la réalisation de notre travail ;

Et à tous ceux qui nous ont enseigné durant notre parcours scolaire.

Sommaire

Introduction générale.....	1
-----------------------------------	----------

CHAPITRE1 : TRAITEMENT D'IMAGE

1.1 Introduction.....	3
1.2 Notion d'image	3
1.3 Histogramme	4
1.3.1 Introduction	4
1.3.2 Histogramme cumulé.....	6
1.3.3 Information extraite d'un histogramme	7
1.3.4 Opérations Sur Les Histogrammes	7
1.3.4.1 Egalisation d'histogramme	7
1.3.4.2 Etirement de l'histogramme	8
1.4 Filtrage des images.....	10
1.4.1 Fonctionnement d'un filtre	11
1.4.2 Filtre Linéaire	11
1.4.2.1 Filtre Passe-Haut	12
1.4.2.1.1 Filtre de Sobel	12
1.4.2.2 Filtre Passe-Bas	13
1.4.2.2.1 Filtre Moyenneur	14
1.4.2.2.2 Filtre gaussien	15
1.4.3 Filtre Non-Linéaire	16
1.4.3.1 Filtre médian	16
Conclusion.....	18

CHAPITRE 2 : RECONNAISSANCE DES OBJETS

2.1 Introduction	19
2.2 processus de reconnaissance des objets	19
2.2.1 Acquisition	20
2.2.2 Prétraitement	20
2.2.3 Extraction des caractéristiques	20
2.2.4 Apprentissage	20
2.2.5 Classification	20
2.2.5.1 Le classifieur SVM	20

2.2.5.2 le classifieur K- plus proches voisins	21
a. principe de K-ppv.....	21
2.3 Reconnaissance de visage	21
2.4 les méthode de reconnaissance de visage.....	22
2.4.1 Méthodes globales.....	22
2.4.1.1 Analyse en composantes principales (ACP)	22
2.4.1.2 Analyse Discriminante Linéaire (ADL)	23
2.4.2 Méthodes locales	23
2.4.2.1 méthode par le modèle actif d'apparence (AAM)	23
2.4.2.2 la méthode local binary pattern	24
2.4.2.3 Le descripteur SIFT	24
a- Principe	24
b- Propriétés des SIFT	25
2.5 Principales difficultés de la reconnaissance de visages	25
2.5.1 Changement d'illumination	25
2.5.2 Variation de pose	26
2.5.3 Expressions faciales	26
2.5.4 Présence ou absence des composants structurels	27
2.5.5 Occultations partielles	27
2.6 Conclusion	28

CHAPITRE 3 : LES MOTIFS BINAIRES LOCAUX

3.1 Introduction	29
3.2 Méthode des motifs binaire locaux	29
3.3 Etapes d'extraction des motifs binaires locaux	30
3.4 LBP pour la reconnaissance de visages	32
3.5 Derivation	32
3.6 Lbp Invariant à la rotation	33
3.7 Motif binaire uniforme et non-uniforme.....	34
3.8 Histogramme LBP	36
3.8.1 distance entre histogrammes LBP	37
3.9 Avantages de motifs binaires locaux.....	39
3.10 Inconvénients de motifs binaires locaux	39

3.11 Les Motifs ternaires locaux.....	39
3.11.1 extraction des motifs ternaires locaux	40
3.12 Conclusion	41

CHAPITRE 4 : Conception et Réalisation .

4.1. Introduction	42
4.2. Conception du système	42
4.2.1 Processus de reconnaissances des objets	42
4.2.1.1. La Base de données ORL	43
4.2.2. Phase d'apprentissage	43
4.2.2.1. Extraction des points d'intérêts Sift	43
4.2.3. Choix des points d'intérêts qui font partie du visage	44
4.2.4. Extraction des points caractéristiques LBPs	45
4.2.5. Détection et localisation	45
4.2.6. Classification	45
4.2.2. Mesure de Similarité	45
4.4. Environnement et outils de travail	48
4.4.1. Environnement de programmation.....	48
4.4.1.1. Langage de programmation Python	48
4.4.1.2. OpenCV	49
4.4.2. Présentation de l'application	48
4.5 Expérimentation et résultats	54
4.5.1 Mesure d'évaluation	54
4.6 Conclusion	58
Conclusion Générale.....	59

Liste de figures

CHAPITRE1 : TRAITEMENT D'IMAGE

Figure1 .1 : Représentation de l'image monochrome (Niveau de Gris)	3
Figure1 .2 : Représentation de l'image couleur	4
Figure1 .3 : Nombre de Pixel ayant ce niveau de gris	5
Figure1 .4 : Image et sont histogramme niveaux de gris	5
Figure1 .5 : Image et sont histogramme en couleur	6
Figure1 .6 : Histogrammes cumulé niveaux de gris et couleur	6
Figure1.7 : Egalisation d'histogramme	8
Figure1.8 : Etirement d'histogramme	10
Figure1.9 : Fonctionnement d'un filtre.....	11
Figure1.10 : Filtre passe-haut	12
Figure1.11 : Filtre vertical et horizontal.....	12
Figure1.12 : Additionnement du filtre vertical et horizontal.....	13
Figure1.13 : Application du filtre Sobel	13
Figure1.14 : Filtre passe-haut	14
Figure1.16 : Application du Filtre Moyeneur	15
Figure1.17 : Application du Filtre Gaussien	16
Figure1.18 : Application du Filtre Médian	17

CHAPITRE 2 : RECONNAISSANCE DES OBJETS

Figure 2.1 : processus de reconnaissance des objets	19
--	----

Figure 2.2 : Operateur former le descripteur SIFT de 128 éléments.....	25
Figure 2.2 : Exemple de variation d'éclairage.	26
Figure 2.3 : Exemples de variation de poses.	26
Figure 2.4 : Exemples de variation d'expressions.	27
Figure 2.5 : Exemple de port de lunettes	27

CHAPITRE 3 : LES MOTIFS BINAIRES LOCAUX

Figure3.1 : construction d'un motif binaire et du code LBP	29
Figure 3.2: Exemple des LBP avec différent voisinages et rayons	32
Figure 3.3 : les 36 uniques motifs binaires locaux invariants en rotation possibles dans $P = 8$	34
Figure 3.4 : des motifs binaire locaux uniforme et non uniforme	35
Figure 3.5 : Motifs binaire locaux uniforme	36
Figure 3.6 : L'histogramme LBP	37
Figure 3.7 : L'opérateur LTP	40
Figure 3.8 : Operateur divisé le code ternaire en deux codes binaires.....	41

CHAPITRE 4 : Conception et réalisation

Figure 4.1 : Precessus de reconnaissance d'objet avec la méthode LBP	42
Figure 4.2 : Exemple des images de la base TA&T	43
Figure 4.2 : Exemple des points d'intérêt Sift	44
Figure 4.3 : Exemple de l'application qui permet de faire le choix des points	44
Figure 4.4 : L'interface de l'application	50
Figure 4.5 : La partie chargement de l'image avant le clique sur le bouton Traitement.....	51
Figure 4.6 : La partie traitement de l'image après le clique sur le bouton Traitement	51
Figure 4.7 : Histogramme LBP de l'image requête	51
Figure 4.8 : Détection ou non d'une image similaire	52
Figure 4.9 : Classifications après détection et classement par le plus proche voisin	53
Figure 4.10 Résultats obtenus par la métrique d'intersection avec LBP _{4.1}	55
Figure 4.11 Résultats obtenus par la métrique Log-likelihood avec LBP _{8.2}	56

Figure 4.12 Résultats obtenus par la métrique Corrélation avec LBP $_{8,2}^U$	56
Figure 4.13 Résultats obtenus par la métrique Chi square statistic avec LBP $_{4,1}$	57

Liste de des tableaux

CHAPITRE 3 : LES MOTIFS BINAIRES LOCAUX

Table 3.1 : Exemple des motifs binaire locaux uniforme et non uniforme.....	35
--	----

CHAPITRE 4 : Conception et réalisation

Tableau 4.1: Matrice de confusion	54
Tableau 4.2 Résultats obtenus de la détection avec différents LBP	57

Résumé

La reconnaissance est une tâche fondamentale dans le processus d'analyse des objets dans les images. La reconnaissance faciale est actuellement un domaine en plein essor. Elle rentre petit à petit dans nos vies au travers de nos téléphones mobiles ou de nos ordinateurs portables. Malgré l'amélioration des performances et le taux élevé de détection atteint, les systèmes en quête de plus performances sur divers l'objet à détecter reste un domaine d'études ouvert. L'objectif de notre projet est de mettre en œuvre un système d'identification de visage. Ce système nécessite la construction d'une base de données comme modèle de référence de la texture des objets.

Nous présentons la technique permettant de reconnaître une personne parmi l'ensemble. Le système de reconnaissance faciale est basé sur le principe de motifs binaire locaux nous utilisons la bibliothèque OpenCV et son implémentation de l'algorithme LBP (motifs binaire locaux) et descripteur SIFT pour la détection et la classification de visage.

Ainsi, il s'agit, dans un premier temps, d'étudier les éléments et mécanismes permettant de décrire les objets d'une certaine classe par leur texture LBP. Ensuite, dans un second temps, de construire par apprentissage un modèle pour discerner et reconnaître les visages décrits des autres visages ou mêmes des objets d'autres classes.

Mots-clés: Détection de visage, reconnaissance faciale, LBP, SIFT, Analyse d'image, Classifieurs, Mesures de Similarité.

Introduction Générale

La vision par ordinateur est une branche de l'intelligence artificielle qui se concentre sur l'extraction d'information utile contenue dans une image. Durant les dernières décennies, un grand nombre d'algorithmes innovants ont été développés dans ce but. Ces recherches ont permis d'atteindre des taux de reconnaissance appréciables sur de nombreuses bases de données de test. Ces méthodes extraient une représentation de l'image sous forme de descripteurs. Elles utilisent ensuite l'apprentissage automatique pour traiter ces descripteurs et obtenir le résultat recherché comme, par exemple, la reconnaissance des objets présents dans l'image (visage) ou la classification de cette image dans une catégorie déterminée.

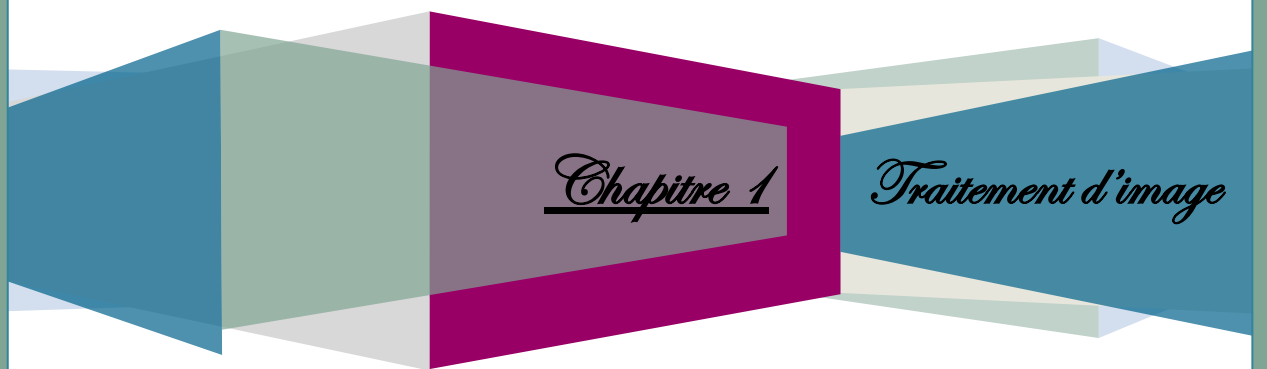
La reconnaissance de visage présent dans l'image est plus facile qu'il utilise les points caractéristique de l'image. Les caractéristiques essentielles pour la reconnaissance du visage sont: les yeux, la bouche, le tour du visage, le bout du nez,... etc. Selon le système utilisé, l'individu doit être positionné devant la caméra où peut être en mouvement à une certaine distance. Les données qui sont obtenues sont comparées au fichier référence. Le logiciel doit être capable d'identifier un individu malgré différents artifices physiques (moustache, barbe, lunettes, etc..).

Le visage est une biométrie relativement peu sûre. En effet, le signal acquis est un sujet à des variations beaucoup plus élevées que d'autres caractéristiques. Celles-ci peuvent être causées, entre autres, par le maquillage, la présence ou l'absence de lunettes, le vieillissement et l'expression d'une émotion. La méthode de la reconnaissance du visage est sensible à la variation de l'éclairage et le changement de la position du visage lors de l'acquisition de l'image.

Ce mémoire traite un sujet portant sur l'authentification du visage. Un système d'authentification a pour but de vérifier l'identité d'un individu après que celui-ci se soit identifié. Il ne s'agit donc pas d'un système d'identification qui lui se charge de découvrir l'identité a priori inconnue d'un individu. Plusieurs méthodes ont été développées dans la littérature pour la reconnaissance de visage. Dans notre travail nous avons étudié la techniques d'extraction des

caractéristiques de visage, La méthode LBP (local Binary Pattern), c'est une méthode mathématique dont son principe consiste à caractériser la texture d'une image par calcul du code LBP pour tous les pixels des points caractéristiques d'image ensuite calcule l'histogramme de cette valeur LBP pour former un vecteur de caractéristiques représentant l'image faciale. Nous avons choisi d'articuler notre étude autour de quatre chapitres principaux.

- Premier chapitre : traitement d'image, les histogrammes et les filtres.
- Dans le deuxième chapitre, nous évoquerons le processus de reconnaissance d'objet. nous allons définir les différentes étapes de processus, aussi voir les différentes méthodes de reconnaissance de visage et les difficultés de cette reconnaissance.
- Le troisième chapitre nous présenterons l'approche LBP (Local Binary Pattern). C'est une méthode mathématique qui est basée sur la caractérisation de la texture d'image par le calcul des valeurs LBP pour chaque pixel d'image.
- Dans le quatrième chapitre, Nous présentons la démarche que nous avons suivie étape par étape jusqu'à de faire la classification des images basée sur l'apprentissage des motifs LBP et le descripteur SIFT.



Chapitre 1

Traitement d'image

1.1 Introduction

Le **traitement d'images** est une discipline qui étudie les images numériques et leurs transformations, c'est l'ensemble des techniques permettant de modifier une image dans le but de l'améliorer ou d'en extraire des informations.

Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et /ou de numérisation, assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images traitées.

1.2 Notion d'image

Une image **numérique** (« **digital** ») est une image $f(x,y)$ dont à la fois le domaine spatial et la luminosité ont été discrétisés. Elle est représentée par une matrice d'entiers pour l'image monochrome :

Une image **monochrome** est une fonction à **2 dimensions** de l'intensité lumineuse, $f(x,y)$, où x et y sont des coordonnées spatiales et la valeur de f en (x,y) est proportionnelle à la luminosité (« **brighness** ») de l'image en ce point. Elle est représentée par une matrice d'entiers



Figure1 .1 : Représentation de l'image monochrome (Niveau de Gris)

☞ pour une image en **couleur**, f est un vecteur dont chaque composante donne la luminosité de l'image au point (x,y) relativement à l'espace des couleurs choisi (ex : RVB, XYZ, Lab, ...). Elle est représentée par **trois matrices** d'entiers. (Chacune représentant un sous-espace de l'espace des couleurs. ex : la couleur rouge dans le système RVB) :

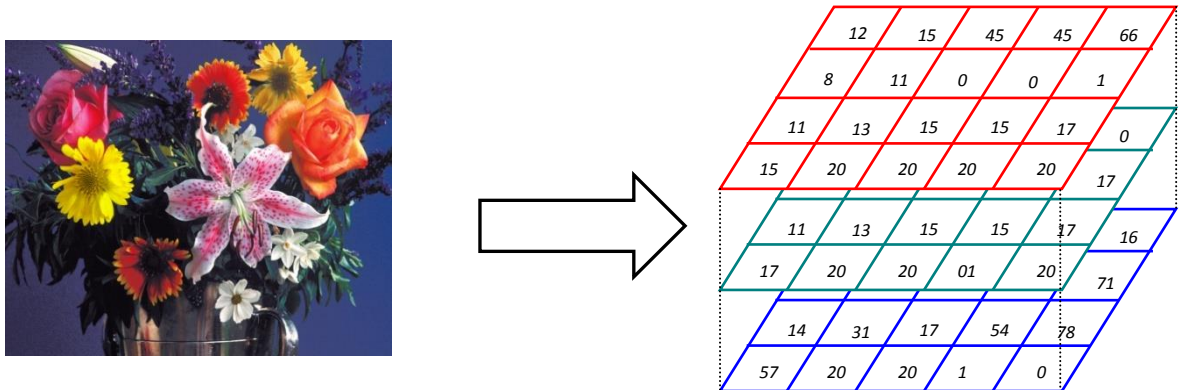


Figure1 .2 : Représentation de l'image couleur

Il existe différentes méthodes afin d'améliorer la qualité d'une image. Il y a des méthodes qui passent par le calcul d'histogramme, d'autre par le filtrage ou encore par l'amélioration du rapport image/bruit. Ces méthodes doivent être utilisées en fonction de ce que l'on veut obtenir.

En effet chacune de ces méthodes aurait un effet bien spécifique sur l'image que se soit pour augmenter les contrastes ou bien mettre en avant des petits détails de l'image.

Nous allons maintenant voir le fonctionnement des ces diverses méthodes ainsi que les résultats que l'on obtient pour chacune.

1.3 Histogramme

1.3.1 Introduction

L'histogramme est un outil très utile pour étudier la répartition des composantes d'une image mais il permet également de corriger le contraste et l'échelle des couleurs pour des images sur-exposées ou sous-exposées. En outre sa modification n'altère pas les informations contenues dans l'image mais les rend plus ou moins visibles.

Un histogramme est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse. Par convention un histogramme représente le niveau d'intensité en abscisse en allant du plus foncé (à gauche) au plus clair (à droite).

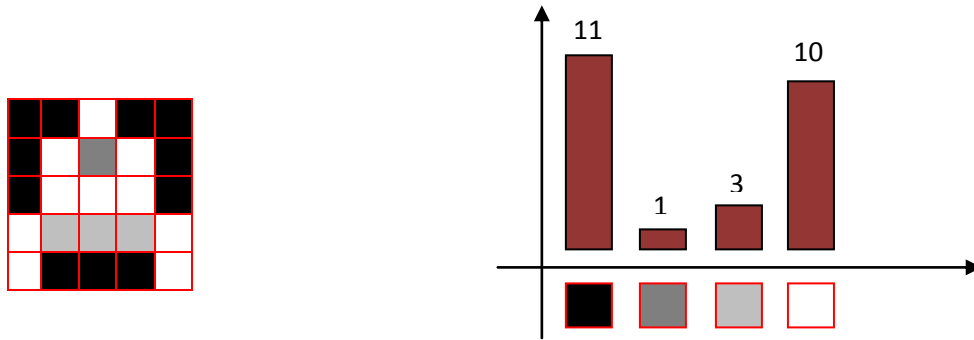


Figure1 .3 : Nombre de Pixel ayant ce niveau de gris

Ainsi, l'histogramme d'une image en 256 niveaux de gris signifie que nous allons *compter*, pour chaque niveau de gris i compris entre 0 et 255, le nombre de pixels qui portent la valeur i . On notera ce nombre hi . On pourra ensuite représenter cet histogramme par un diagramme qui prend en abscisse les valeurs de i (de 0 à 255, donc), et en ordonnée les hi correspondants.

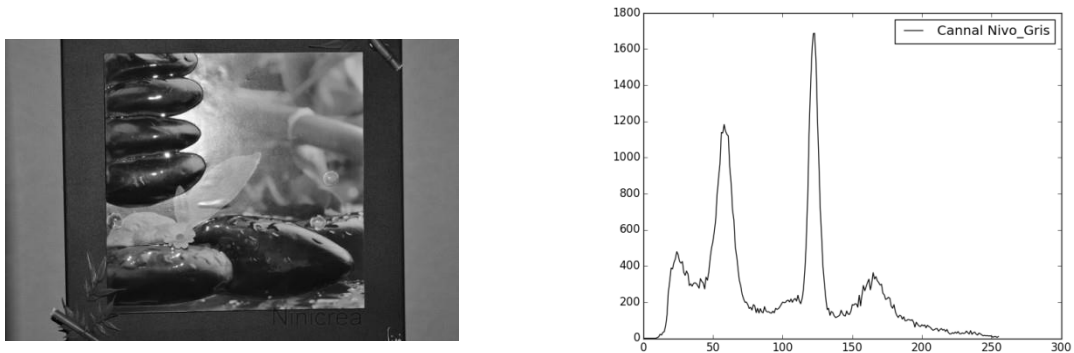


Figure1 .4 : Image et sont histogramme niveaux de gris

Pour les images en couleur plusieurs histogrammes sont nécessaires. Par exemple pour une image codée en RGB :

- trois histogrammes représentant respectivement la distribution des valeurs Respectives des composantes rouges, bleues et vertes

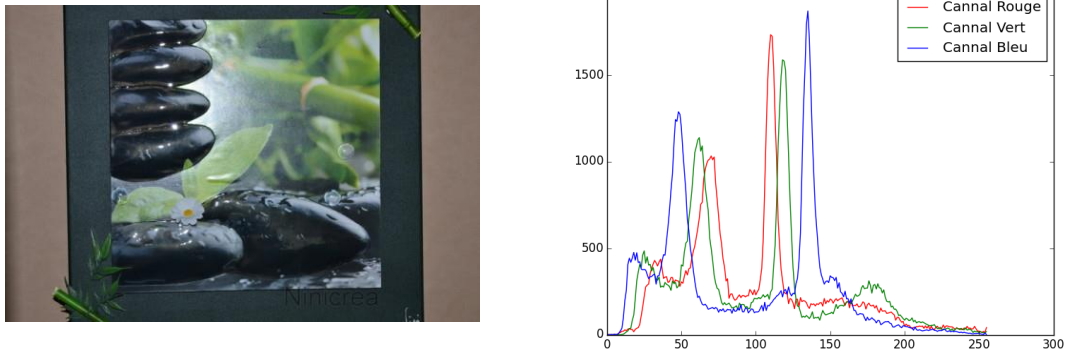


Figure1 .5 : Image et sont histogramme en couleur

1.3.2 Histogramme cumulé

En plus de l'histogramme classique, il peut être intéressant dans certains cas de travailler sur l'histogramme *cumulé*. Ce dernier s'obtient en associant à chaque niveau de gris i le nombre h^c_i de pixels de l'image qui ont une valeur inférieure ou égale à i . En quelque sorte, cela revient à calculer l'histogramme de l'image et d'associer à chaque niveau i la somme des h_j pour $j \leq i$. En formule, cela donne :

$$h_j = h_0 + h_1 + \dots + h_i$$

Voici les deux histogrammes cumulés en niveau de gris et en couleur de l'image précédente

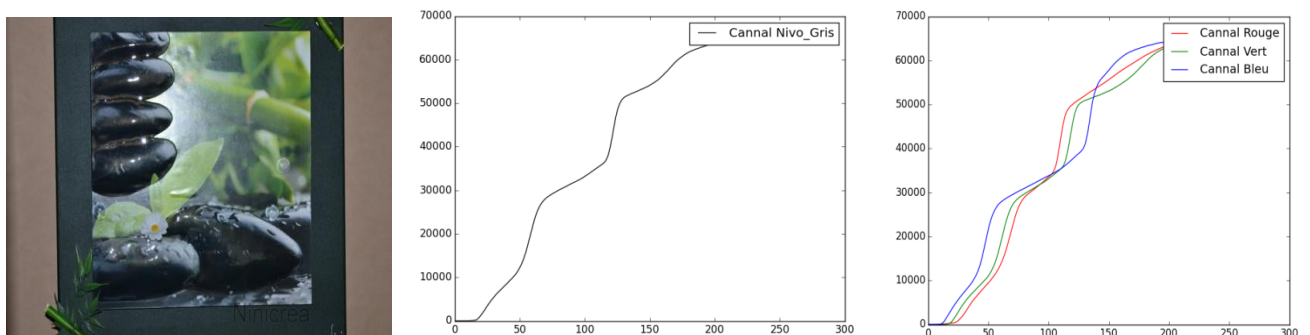


Figure1 .6 : Histogrammes cumulé niveaux de gris et couleur

1.3.3 Information extraite d'un histogramme

L'histogramme nous donne une information sur la répartition des valeurs des pixels dans l'image. Un histogramme avec beaucoup de valeurs proche de zéro traduira une image sous-exposée (image très sombre) alors qu'un histogramme dont les valeurs sont élevées traduira une image sur-exposée (image très claire)

Le but de traiter l'histogramme d'une image consiste à augmenter le contraste sans faire de saturation. Elle est très utile dans le domaine médical où elle trouve un automatisme afin de traiter scanner et autres radio le plus rapidement possible afin de bien voir les détails. Un histogramme permet d'avoir deux informations sur une image

- ✓ Savoir si l'image est sombre ou non. Une image sera sombre lorsque les pics sont sur plus vers la gauche de l'histogramme. S'ils sont vers la droite, l'image va être claire.
- ✓ Avoir des informations sur le contraste de l'image : si les pics sont au milieu de l'histogramme, l'image ne sera pas très contrastée. Si ceux-ci sont sur les bords, l'image sera fortement contrastée

1.3.4 Opérations Sur Les Histogrammes

La méthode pour faire du rehaussement d'image consiste en le calcul de l'histogramme de cette image. Il existe ensuite différentes méthodes :

1.3.4.1 Egalisation d'histogramme

L'égalisation d'histogramme a pour but d'harmoniser la répartition des niveaux de luminosité de l'image, de telle manière à tendre vers un même nombre de pixel pour chacun des niveaux de l'histogramme. Cette opération vise à augmenter les nuances dans l'image [2].

$$f_{new}(x, y) = \frac{(2^D - 1) \times HC(f[x, y])}{wh}$$

Avec :

D : Dynamique

HC (f[x,y]) : histogramme cumulé

(w,h) : dimension de l'image

dans cette exemple : $D=8, w= 183$ et $h= 275$

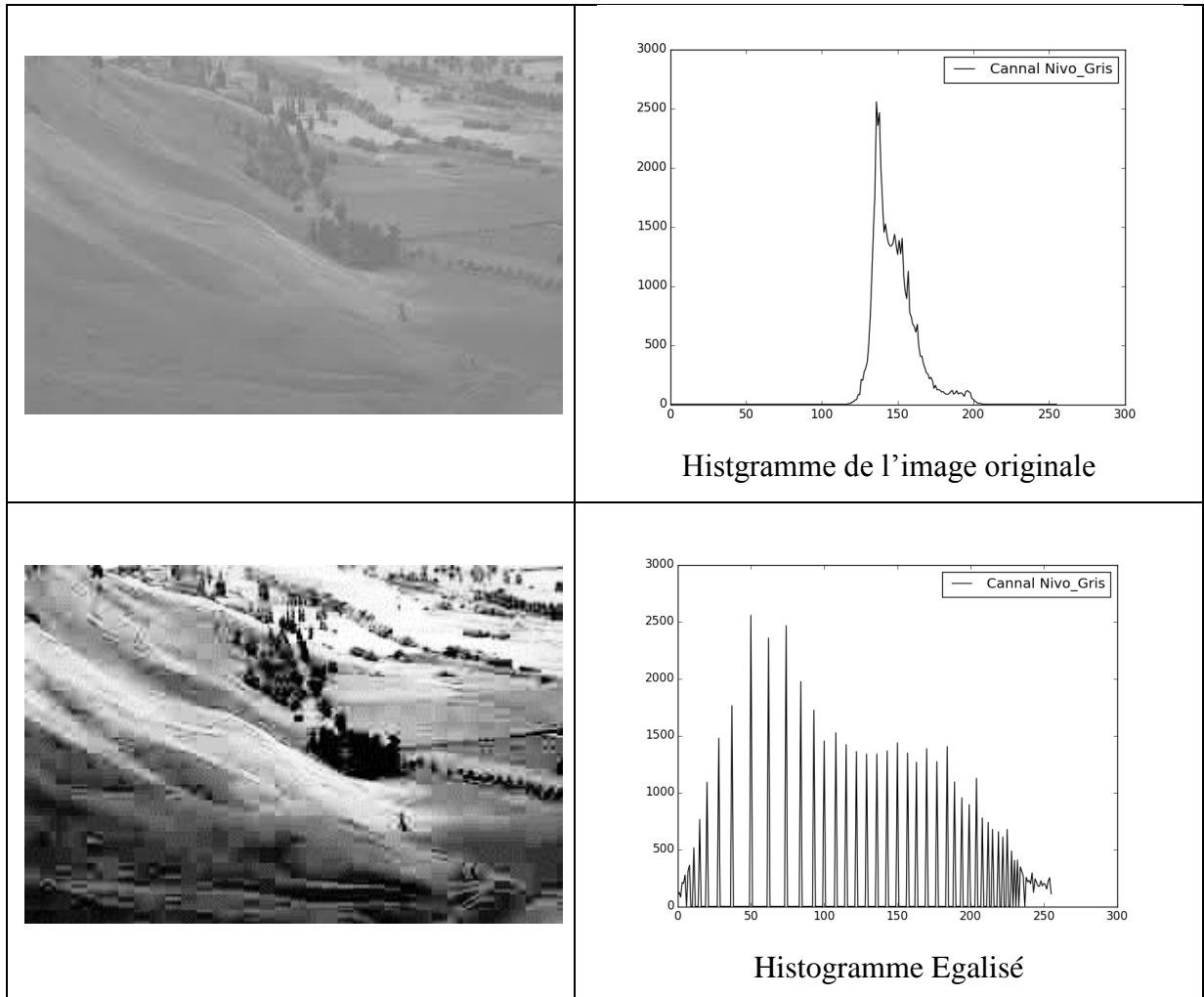


Figure1.7 : Egalisation d'histogramme

1.3.4.2 Etirement de l'histogramme

L'étirement d'histogramme (aussi appelé "linéarisation d'histogramme" ou "expansion de la Dynamique") consiste à répartir les fréquences d'apparition des pixels sur la largeur de l'histogramme. Ainsi il s'agit d'une opération consistant à modifier l'histogramme de telle manière à répartir au mieux les intensités sur

L'échelle des valeurs disponibles. Ceci revient à étendre l'histogramme afin que la valeur d'intensité la plus faible soit à zéro et que la plus haute soit à la valeur maximale.

De cette façon, si les valeurs de l'histogramme sont très proches les unes des autres, l'étirement va permettre de fournir une meilleure répartition afin de rendre les pixels clairs encore plus clairs et les pixels foncés proches du noir.

$$f_{new}(x,y) = \left[\frac{f(x,y) - f(x,y)_{min}}{f(x,y)_{max} - f(x,y)_{min}} \right] \times 255$$

Avec :

$f(x,y)$: la valeur de gris du pixel courant

$f_{new}(x,y)$: la nouvelle valeur de gris du pixel courant

$f(x,y)_{max}$: plus grand ton de gris

$f(x,y)_{min}$: plus petit ton de gris

➤ **Remarque** : pas d'effet si $f(x,y)_{min} = 0$ et $f(x,y)_{max} = 255$

dans cet exemple : $f(x,y)_{min} = 117$, $f(x,y)_{max} = 209$ et $D=8$

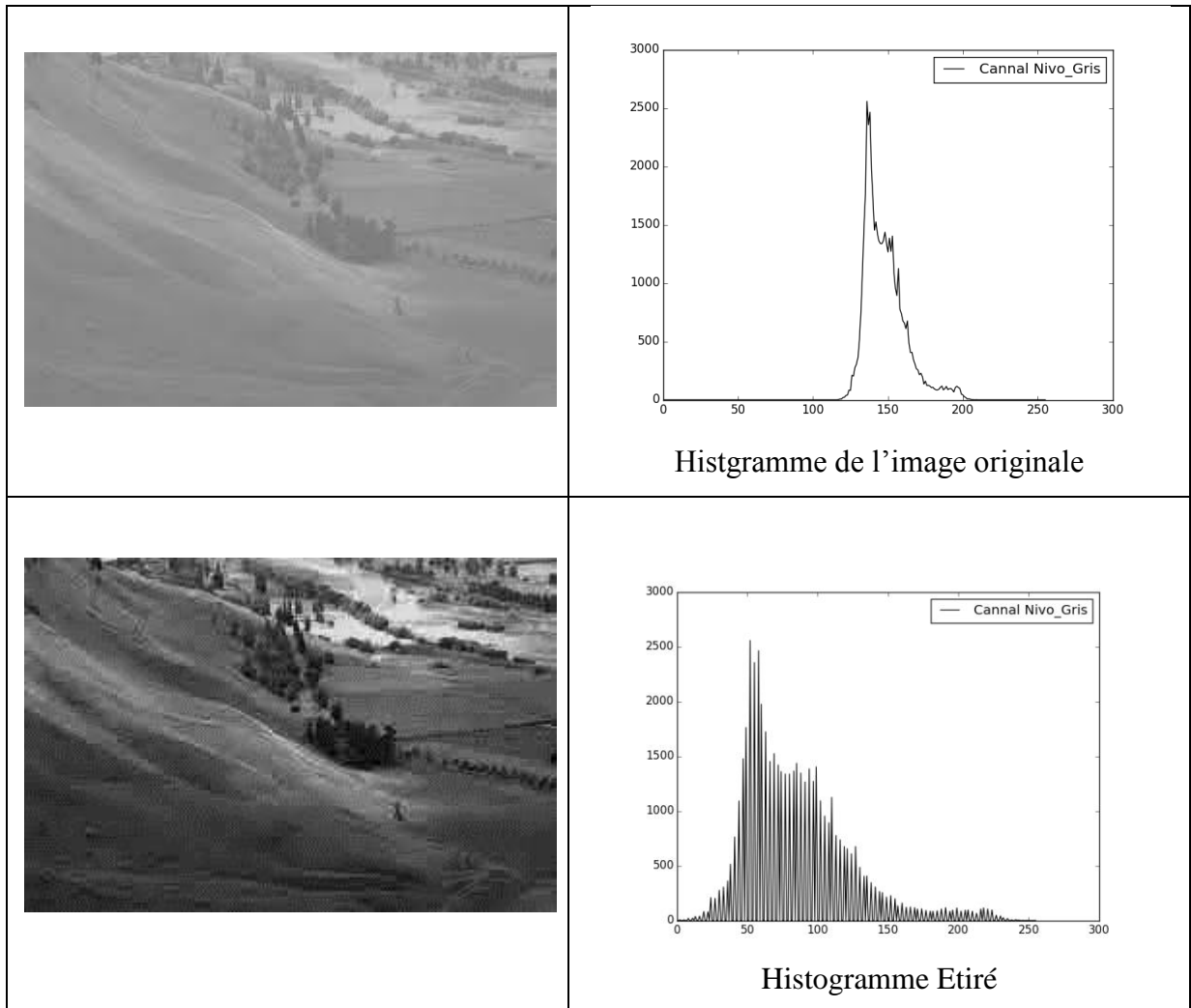


Figure1.8 : Etirement d'histogramme

☞ Contrairement aux manipulations d'histogrammes, qui sont des opérations ponctuelles sur tous les pixels de l'image, les filtrages sont des produits de convolution qui mettent en jeu l'environnement (voisinage) de chaque pixel.

1.4. Filtrage des images

Filtrer une image c'est lui appliquer une transformation mathématique qui modifie les valeurs de gris de tout ou partie des pixels. Si cette fonction est linéaire au sens algébrique du terme, on parle de filtrage linéaire. Si la fonction prend en compte les valeurs de gris au voisinage de chaque pixel

transformé, on parle de fonction de convolution, ils servent à éliminer le bruit sur les images (donc à l'homogénéiser) ou au contraire à en rehausser les contrastes .

Il existe deux grands types de filtres :

☞ les filtres linéaires (ou filtres de convolution).

☞ les filtres non linéaires.

1.4.1 Fonctionnement d'un filtre

La valeur de gris d'un pixel est modifiée en tenant compte de la valeur de gris des Pixels voisins. On applique une matrice de calcul que l'on déplace pixel après pixel par glissement dans l'image. Après avoir fait une ligne on descend d'un cran et on fait une autre ligne. Une nouvelle image est ainsi générée (un facteur de normalisation permet de conserver un intervalle de valeurs identique en entrée et en sortie).

Le noyau change ainsi tous les pixels sauf ceux situés sur le bord de l'image.

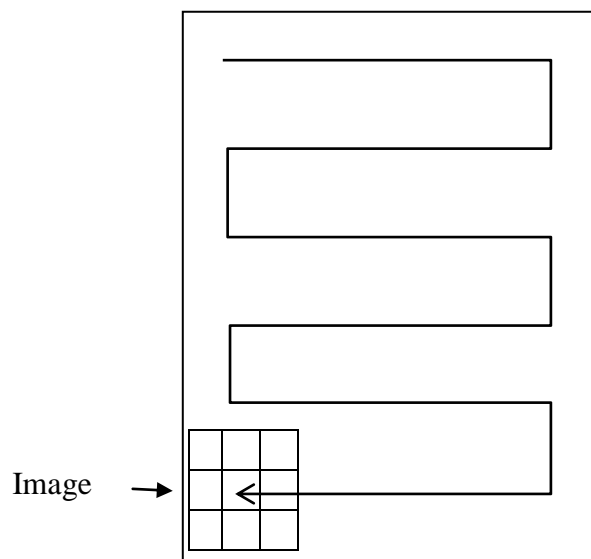


Figure1.9 : Fonctionnement d'un filtre

1.4.2 Filtre Linéaire :

Un filtre linéaire est réalisé grâce à la convolution de l'image avec un « *noyau* » représentant la réponse impulsionnelle du filtre. Le noyau est aussi appelé *masque* pour rappeler l'opération de multiplication effectuée par la convolution.

1.4.2.1 Filtre passe-haut :

Un filtre « **passe haut** » favorise les hautes fréquences spatiales, comme les détails, et de ce fait, il améliore le contraste. Un filtre « **passe haut** » est caractérisé par un noyau comportant des valeurs négatives autour du pixel central, comme dans l'exemple ci-dessous:

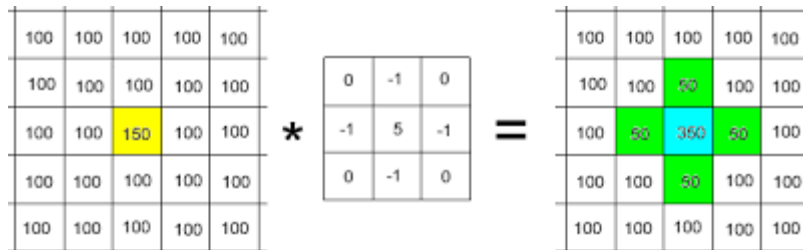


Figure1.10 : Filtre passe-haut

Les filtres passe-haut permettent d'accentuer les détails et le contraste, c'est la raison pour laquelle le terme de filtre "*d'accentuation*" est parfois utilisé.

1.4.2.1.1. Filtre de Sobel :

Le **filtre de Sobel** est un opérateur utilisé en traitement d'image pour la détection de contours. Il s'agit d'un des opérateurs les plus simples qui donne toutefois des résultats corrects.

Pour faire simple, l'opérateur calcule le gradient de l'intensité de chaque pixel. Ceci indique la direction de la plus forte variation du clair au sombre, ainsi que le taux de changement dans cette direction. On connaît alors les points de changement soudain de luminosité, correspondant probablement à des bords, ainsi que l'orientation de ces bords.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A \quad \text{et} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * A$$

Figure1.11 : Filtre vertical et horizontal

En chaque point, les approximations des gradients horizontaux et verticaux peuvent être combinées comme suit pour obtenir une approximation de la norme du gradient:

$$G = \sqrt{G_x^2 + G_y^2}$$

Figure1.12 : Additionnement du filtre vertical et horizontal

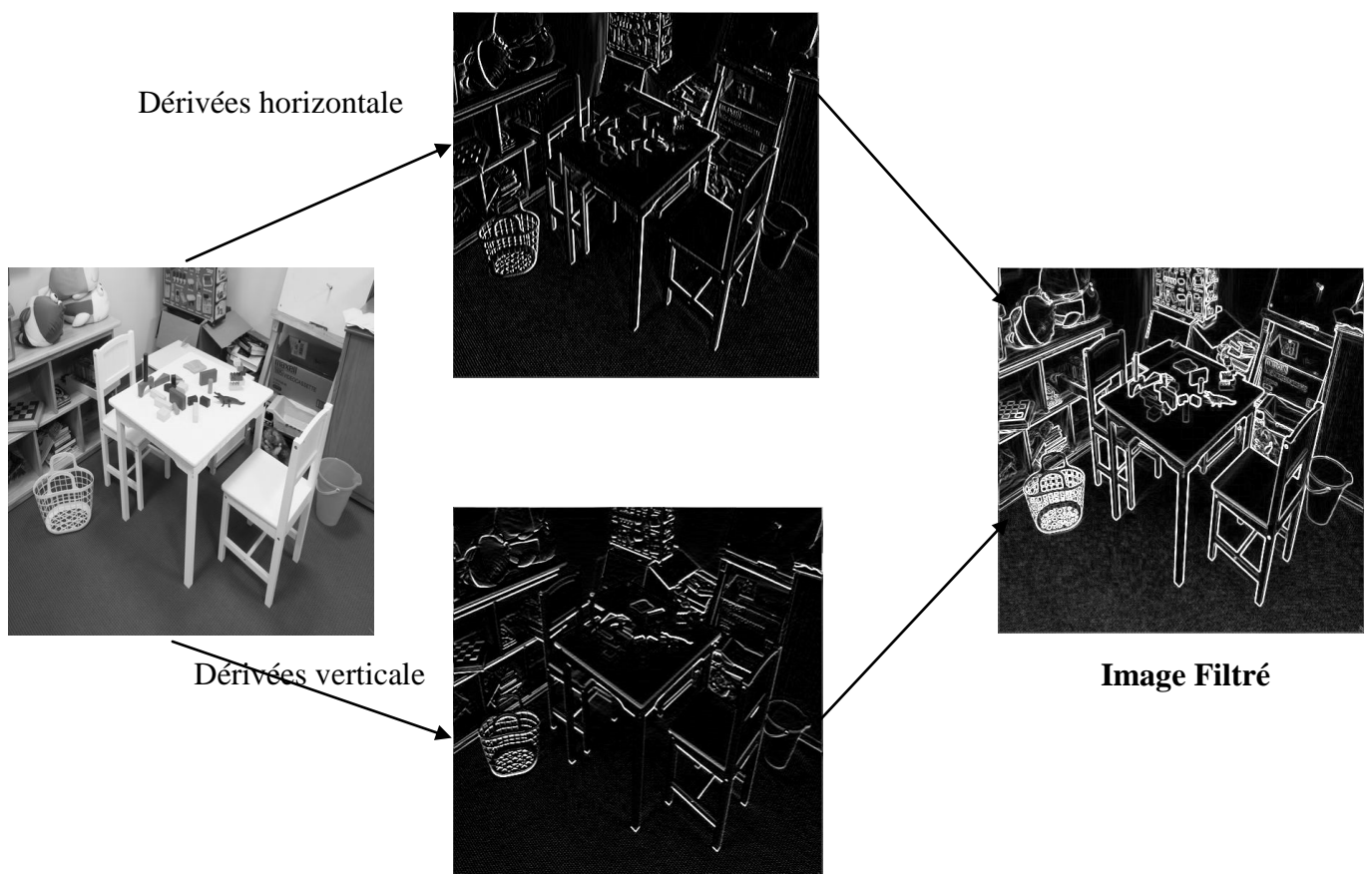


Figure1.13 : Application du filtre Sobel

1.4.2.2 Filtre passe-bas :

Les filtres « **passe bas** » agissent en sens inverse des filtres passe haut et le résultat est, un adoucissement des détails, ainsi qu'une réduction du bruit granuleux.

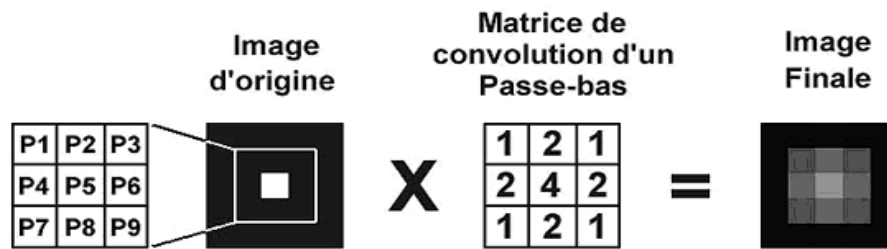


Figure1.14 : Filtre passe-haut

1.4.2.2.1 Filtre Moyenneur

Ce filtre lisseur part du principe que la valeur d'un pixel est relativement similaire à son voisinage. Il fait donc en sorte que chaque pixel est remplacé par la moyenne pondérée de ses voisins. Si on applique un filtre moyenneur [1] de taille $\lambda=3$, cela signifie qu'on additionne la valeur de tous les pixels du voisinage du pixel traité. On obtient ainsi la matrice de convolution suivante :

$$h=1/9 \quad X \quad \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Figure1.15 : matrice de convolution filtre moyenn

h s'appelle le masque de convolution. La somme des coefficients du masque valant 1, le lissage préservera toute zone de l'image où le niveau de gris est constant.

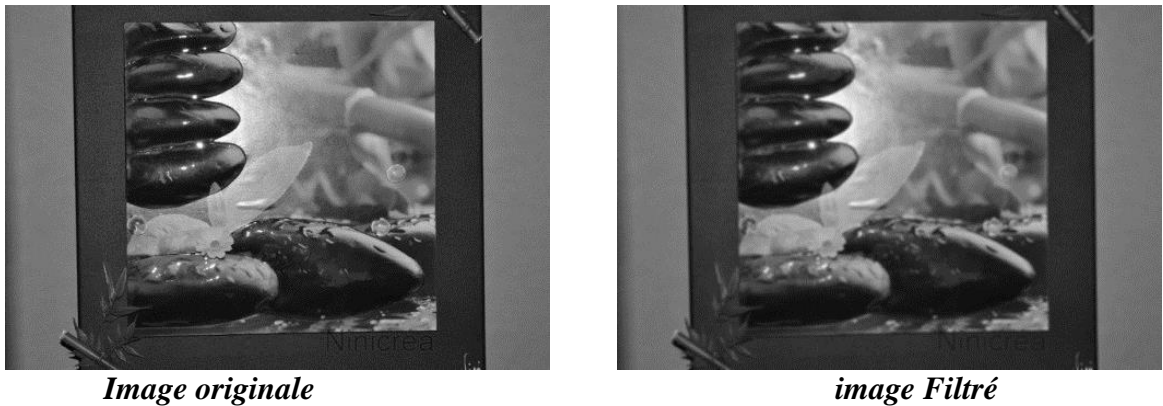


Figure1.16 : Application du Filtre Myenneur

1.4.2.2.2 Filtre gaussien :

Le filtre Gaussien [1] est un filtre de traitement d'image appliqué par convolution (utilise un masque (matrice) appliqué à chaque pixel) .Ce type de filtre est utilisé pour diminuer le bruit ou ou appliquer un flou sur une image.

Principe : calculer la moyenne pondérée des valeurs du voisinage. Les pixels du voisinage qui sont proches du pixel central ont un poids plus fort (= plus d'influence) que ceux qui sont plus éloignés.

$$G(x,y) = \left[\frac{1}{2*\sigma*\pi} \right] * e^{-\frac{(x^2+y^2)}{2*\sigma^2}}$$

❖ En fixant $\sigma = 0.8$ et $coeff = 16$, on obtient le noyau de convolution suivant :

$$h=1/16 \times \begin{matrix} \begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix} \end{matrix}$$

*Image originale**image Filtré***Figure1.17** : Application du Filtre Gaussien

1.4.2 Filtre Non-Linéaire :

Certains filtres ne sont pas linéaires, ce qui signifie que la valeur centrale (le pixel en cours de traitement) est remplacée par une valeur qui existe déjà dans l'image.

1.4.3.1 Filtre Médian :

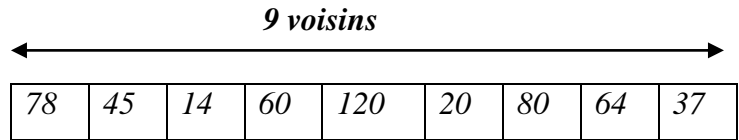
La médiane d'un ensemble est une valeur telle que la moitié des éléments sont plus petit, et la moitié sont plus grands, Le filtre médian [1] est utilisé en traitement d'images pour "nettoyer" les images en éliminant certains défauts parasites. L'algorithme du filtrage médian est très simple : chaque pixel de l'image à filtrer est remplacé par la valeur médiane des pixels voisins. L'image est ainsi transformée par le filtre médian en une autre image de même taille. Pour chaque pixel P de l'image d'origine on crée une liste des pixels situés dans un voisinage 3x3 autour de P. Les 9 pixels de la liste sont ensuite triés dans l'ordre de leurs valeurs numériques. La valeur médiane est celle du pixel situé au milieu de la liste triée. Le pixel P de l'image filtrée prend alors pour valeur la valeur médiane. Dans notre cas nous supposons que les images sont représentées en 256 niveaux de gris. Les valeurs des pixels sont donc comprises entre 0 (noir) et 255 (blanc).

Voisinage de $P = 120$

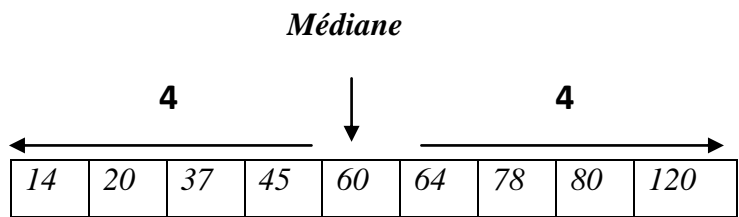
78	45	14
60	120	20
80	64	37



78	45	14
60	60	20
80	64	37



Liste non triée



Liste triée



Image originale



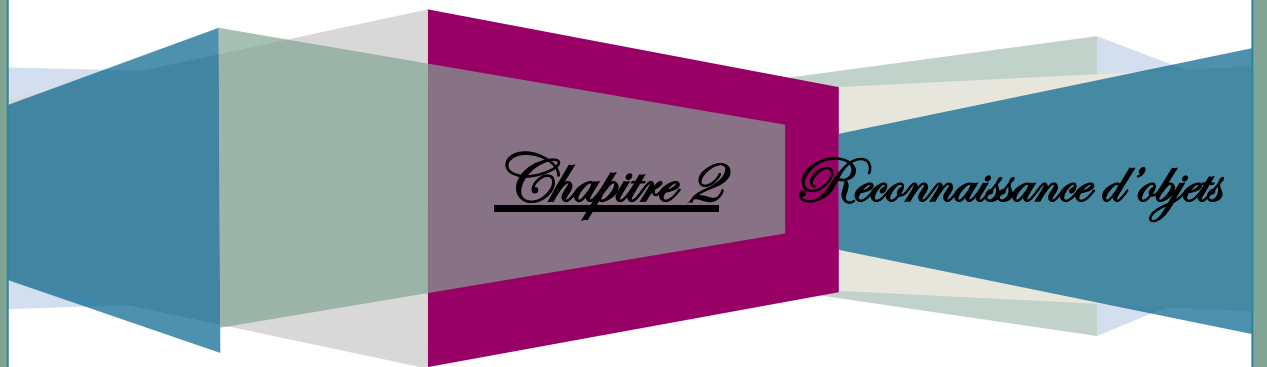
image Filtré

Figure1.18 : Application du Filtre Médian

Conclusion

Ce chapitre n'a fait qu'effleurer l'immense liste des traitements que l'on peut faire subir à une image. Le traitement des images est un domaine très actif, où les avancées théoriques se concrétisent sous la forme d'algorithmes rapides de calcul qui ont des applications importantes pour la manipulation des contenus numériques.

La compréhension du traitement d'images commence par la compréhension de ce qu'est une image. Le mode et les conditions d'acquisition et de numérisation des images traitées conditionnent largement les opérations qu'il faudra réaliser pour extraire de l'information.



Chapitre 2

Reconnaissance d'objets

2.1 Introduction

La reconnaissance d'objets [3] est un domaine très actif en vision par ordinateur. L'approche générale consiste à considérer que l'on dispose d'une banque de données où sont stockés des modèles d'objets et une vue du monde réel. On doit répondre à une question essentielle: L'objet extrait existe-t-il dans la banque de données. Le problème dépend de la modélisation d'objets adoptée.

Dans la majorité des cas cette modélisation est purement géométrique: un ensemble de caractéristiques appelées primitives. L'opération de reconnaissance consiste en l'appariement des ces caractéristiques et celles décelées dans l'image.

2.2 processus de reconnaissance des objets

Le schéma classique d'un processus de reconnaissance des objets [4], présenté dans la **figure 2.1** permet de décrire les principaux traitements que l'on peut être amené à effectuer et leurs objectifs.

On peut distinguer deux étapes pour la reconnaissance d'objets, la première consiste à apprendre la description d'objets à partir d'une base d'exemples (Apprentissage), et la seconde va reconnaître un objet à partir de sa description extraite de l'image (classification).

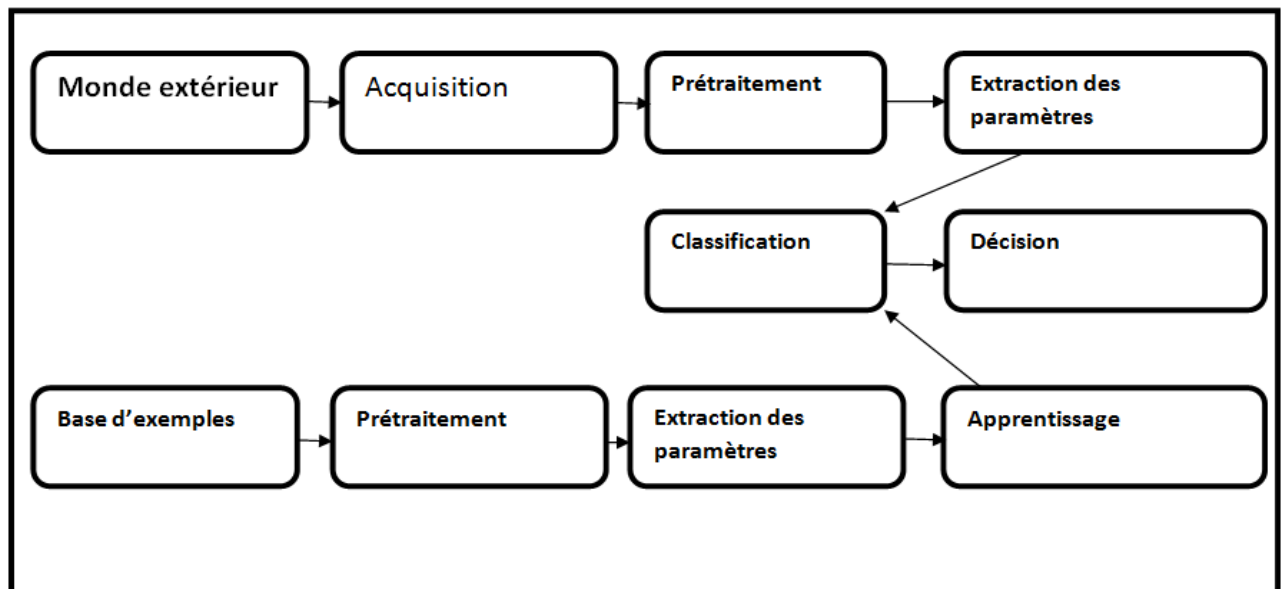


Figure 2.1 : processus de reconnaissance des objets

2.2.1 Acquisition :

La première étape d'une chaîne de traitement et d'analyse des numériques est celle de l'acquisition de l'image. On doit s'assurer d'être entouré meilleures conditions pour de bonnes prises (caméra, support de caméra).

2.2.2 Prétraitement :

L'image brute résultat de l'acquisition n'est pas prête pour subir l'opération de classification. En effet le bruit qui accompagne l'opération d'acquisition, l'orientation et le positionnement quelconque des objets font que certains prétraitements sont nécessaires pour se doter de plus de chance de réussir cette opération de classification.

2.2.3 Extraction des caractéristiques :

Le prétraitement a fini par nous donné une image dotée de meilleurs chance pour un classement correct. L'étape suivante est d'extraire de cette image un ensemble de paramètres (caractéristique) qui serviront à l'identifier.

2.2.4 Apprentissage :

C'est l'étape où on fait apprendre les individus au système, elle consiste à mémoriser les paramètres, après extraction, dans une base de données bien ordonnées pour faciliter la phase de reconnaissance et la prise d'une décision, elle est en quelque sorte la mémoire du système.

2.2.5 Classification :

En reconnaissance de formes et plus particulièrement de l'écrit, les phases d'apprentissage et de classification constituent l'une des étapes les plus fondamentales qui conditionne en grande partie les performances du système. Ces techniques d'apprentissage artificiel ont connu ces dernières décennies des avancées fondamentales à travers des modèles bien connus tels que les réseaux de neurones et les Séparateurs à Vaste Marge (SVM) et qui ont montré globalement de bonnes performances KPPV [5].

2.2.5.1 Le classifieur SVM

le classifieur séparateur à veste marge (SVM) est une méthode de classification de données, été proposée par *Vapnik* en 1995 . Initialement, elle a été définie pour un problème de discrimination à deux classes (cas binaire). De manière simple, il s'agit de construire, dans l'espace des attributs un hyperplan séparant les points représentatifs des prototypes de chacune des deux classes. Cette approche, également appelée méthode des séparateurs à vastes marges, repose sur deux notions clés : **la notion de marge maximale** et **la notion de noyau** La marge est la distance séparant la frontière de séparation (l'hyperplan) et les

prototypes les plus proches de cette frontière. Ces prototypes sont appelés vecteurs supports. Dans les SVM, la frontière de séparation est choisie comme étant celle qui maximise la marge.

2.2.5.2 le classifieur K- plus proches voisins

K plus proches voisins **K-PPV [5]** est l'une des méthodes les plus populaires et les plus simples d'apprentissage supervisé utilisés pour le classement dans les différents domaines de la reconnaissance des objets et des problèmes d'exploration de données, elle prédit la classe d'un nouveau cas, en fonction de la majorité des k-plus proche voisins.

La méthode utilise donc deux paramètres : le nombre K et la fonction de similarité pour comparer le nouveau cas aux cas déjà classés.

La méthode K-PPV a prouvé son efficacité et les expériences montrent qu'il résiste bien aux données bruitées. Par contre, ils requièrent de nombreux exemples. Il a l'avantage d'être extrêmement simple et de donner en général de bons résultats. Mais l'amélioration de la performance d'un classifieur reste toujours attractive pour les chercheurs.

a. principe de K-ppv

C'est une approche très simple et directe. Elle ne nécessite pas d'apprentissage mais simplement le stockage des données d'apprentissage.

Son principe est le suivant. Une donnée de classe inconnue est comparée à toutes les données stockées. On choisit pour la nouvelle donnée, la classe majoritaire parmi ses K plus proches voisins (Elle peut donc être lourde pour des grandes bases de données) au sens d'une distance choisie (**métrique de similarité**).

2.3 Reconnaissance de visage

La reconnaissance de visages est la technique la plus commune et populaire. Elle reste la plus acceptable puisqu'elle correspond à ce que les humains utilisent dans l'interaction visuelle; et par rapport aux autres méthodes, la reconnaissance du visage s'avère plus avantageuse, d'une part c'est une méthode non intrusive, c'est-à-dire elle n'exige pas la coopération du sujet (En observant les individus à distance), et d'une autre part les capteurs utilisés sont peu coûteux (une simple caméra) contrairement à l'empreinte digitale et l'iris où le sujet devra être très proche du capteur et devra coopérer pour l'acquisition de l'image sans oublier le coût de l'équipement nécessaire pour l'acquisition (équipement spécial coûteux). Malgré que certains disent que la reconnaissance de visages est une biométrie relativement peu sûre, sur le fait que le signal acquis est sujet à des variations beaucoup plus élevées que d'autres

caractéristiques, comme la variation de l'éclairage, le changement de la position du visage, la présence ou l'absence de lunettes et autres, mais au cours de ces dernières années plusieurs techniques de traitements d'images sont apparues, telle que la détection du visage, la normalisation de l'éclairage, etc. Sans oublier le développement considérable des technologies des caméras numériques, ce qui néglige l'effet de ces problèmes.

2.4 les méthodes de reconnaissance de visage

2.4.1 Méthodes globales

Le principe de ces méthodes est de représenter une image faciale par un seul vecteur de grande dimension en concaténant les niveaux de gris de tous les pixels du visage. Cette représentation, appelée description basée sur l'apparence globale, a deux avantages.

Premièrement, elle conserve implicitement toutes les informations de texture et de forme utiles pour différencier des visages.

Deuxièmement, elle peut tenir compte des aspects d'organisation structurelle globaux du visage. Toutefois, son inconvénient majeur réside dans la dimension très grande de l'espace image qu'elle nécessite ce qui rend très difficile la classification. Pour traiter le problème des données de grande dimension, des techniques de réduction de la dimensionnalité peuvent être utilisées. L'une des techniques les plus courantes pour la reconnaissance de visages est la description par visages propres, qui est basée sur (ACP) [6].

2.4.1.1 Analyse en composantes principales (ACP)

Une méthode très populaire, basée sur la technique ACP, est la méthode Eigen-face. Son principe est le suivant : étant donné un ensemble d'images de visages exemples, il s'agit tout d'abord de trouver les composantes principales de ces visages. Ceci revient à déterminer les vecteurs propres de la matrice de covariance formée par l'ensemble des images exemples. Chaque visage exemple peut alors être décrit par une combinaison linéaire de ces vecteurs propres. Pour construire la matrice de covariance, chaque image de visage est transformée en vecteur. Chaque élément du vecteur correspond à l'intensité lumineuse d'un pixel. L'avantage de l'ACP est une technique rapide, simple et populaire dans l'identification de modèle, c'est l'une des meilleures techniques. Les projections de l'ACP sont optimales pour la reconstruction d'une base de dimension réduite. Cependant, l'ACP n'est pas optimisée pour la séparabilité (discrimination) de classe. Une alternative qui est l'analyse discriminante linéaire (LDA) tient compte de ceci.

2.4.1.2 Analyse Discriminante Linéaire (ADL)

Une autre méthode très connue est celle basée sur l'ADL (Analyse discriminante linéaire) [6]. L'objectif de la plupart des algorithmes basés sur l'ADL, est de trouver les directions de projection les plus discriminantes dans l'espace propre, en maximisant le ratio entre les variations interpersonnelles et les variations intra-personnelles. Comme les variations intra-personnelles peuvent être petites (notamment quand il n'y a pas beaucoup d'images par individu), ce ratio est difficile à maximiser puisque il est déjà grand. ce problème est encore appelé Small Sample Size.

Pour l'éviter, on peut utiliser tout d'abord l'ACP et ensuite l'ADL, et cette méthode est appelée Fisher-faces. Voilà pourquoi les méthodes basées sur l'ADL ne fonctionnent bien que lorsque beaucoup d'images par personne sont disponibles dans la base d'apprentissage. En revanche, quand il n'y a pas beaucoup d'images par personne, les méthodes basées sur l'ADL marchent moins bien que celles basées sur l'ACP.

2.4.2 Méthodes locales

Les méthodes locales, basées sur des modèles, utilisent des connaissances a priori que l'on possède sur la morphologie du visage et s'appuient en général sur des points caractéristiques, en détectant certains points ou traits caractéristiques d'un visage puis en les comparant avec des paramètres extraits d'autres visages.

Ces méthodes constituent une autre approche pour prendre en compte la non-linéarité en construisant un espace de caractéristiques local et en utilisant des filtres d'images appropriés, de manière à ce que les distributions des visages soient moins affectées par divers changements.

2.4.2.1 Méthode par le modèle actif d'apparence (AAM)

Franck et al ont travaillé sur reconnaissance et synthèse d'expressions faciales par modèle d'apparence [3]. Et dans ce travail ils traitent une application nouvelle des modèles actifs d'apparence pour l'analyse et la synthèse de visages expressifs, ainsi que pour la reconnaissance d'expressions faciales. Ils considèrent les six expressions universelles que sont les expressions de joie, colère, peur, dégoût, tristesse et surprise. Après une description de ce type de modèle (calculé à l'aide de trois ACP ou d'une seule), ils proposent une méthode originale d'analyse et de synthèse permettant, à partir d'une seule photo, d'annuler l'expression d'un visage quelconque, puis de synthétiser une expression faciale artificielle sur

ce même visage. Ils proposent pour cela deux approches permettant la modélisation des expressions faciales. Enfin, l'intérêt du modèle actif d'apparence pour la reconnaissance automatique d'expressions faciales est montré.

2.4.2.2 La méthode local binary pattern

Les motifs binaires locaux [7] ont initialement été proposés par *Ojala* en 1996 afin de caractériser les textures présentes dans des images en niveaux de gris. Ils consistent à attribuer à chaque pixel P de l'image $I(i,j)$ à analyser, une valeur caractérisant le motif local autour de ce pixel. Ces valeurs sont calculées en comparant le niveau de gris du pixel central P aux valeurs des niveaux de gris des pixels voisins. Cette méthode est plus détaillée dans le (chapitre3).

2.4.2.3 Le descripteur SIFT

Le but d'un descripteur est de décrire une région d'une image. il doit être robuste à de petites transformations tout en étant suffisamment spécifique, par exemple la distance entre deux roues différentes doit être plus grande que la distance entre deux roues identiques, vues sous des angles légèrement différents. Le descripteur d'image SIFT (Scale Invariant Feature Transform) [8] a été introduit dans [Lowe, 2004]. il à été conçu pour être invariant à la fois aux changements d'échelles et aux rotations. De plus il est robuste aux transformations affines, au bruit et aux variations d'illuminations.

a- Principe

Le descripteur SIFT construit un histogramme des orientations des gradients présents au voisinage de la clé détectée. En pratique, le voisinage 16×16 de la clé courante est divisé en 4×4 quadrants au sein desquels les orientations sont quantifiées selon huit directions. Un tel procédé conduit à 16 histogrammes de 8 cases qui sont concaténés pour former le descripteur SIFT de 128 éléments. Chacune des cases représente la somme des amplitudes des gradients dans une direction donnée, pour un quadrant donné. Pour garantir une invariance du descripteur à l'orientation, une orientation de référence est attribuée préalablement à chaque clé et est utilisée comme angle de référence pour le calcul de toutes les orientations du descripteur. L'orientation de référence est donnée par le maximum de l'histogramme des orientations des gradients autour du point. L'histogramme quantifie les 360 degrés sur 36 cases. Chaque nouvelle orientation est pondérée par l'amplitude du gradient correspondant. Il est à noter que dans la version originale, une clé supplémentaire est créée pour toute orientation ayant obtenu un score de plus de 80% du maximum, ce qui conduit

potentiellement à l'extraction de plusieurs points de même localisation mais d'orientations différentes.

Enfin, une étape finale de normalisation du descripteur a pour but d'assurer une robustesse aux variations affines d'illumination [8].

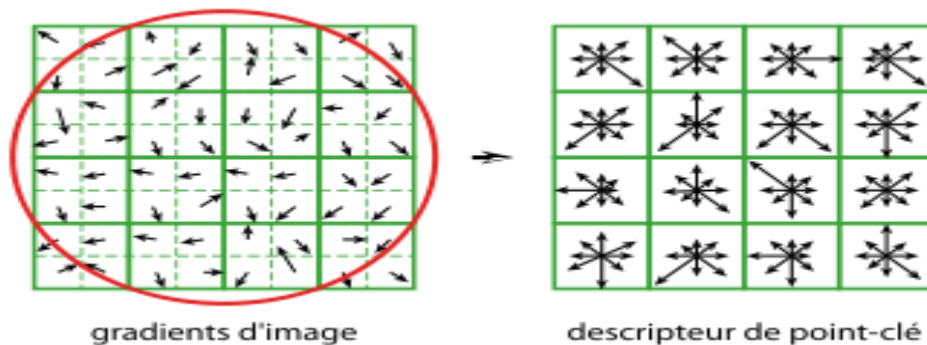


Figure 2.2 : Operateur former le descripteur SIFT de 128 éléments.

b- Propriétés des SIFT

- Une caractéristique unique peut être correctement associée à la haute probabilité contre une grande base de données de caractéristiques à partir de nombreuses images.
- hautement distinctif
- Échelle et la rotation invariante.
- Partiellement invariant à la caméra 3D point de vue
- Peut tolérer jusqu'à environ 60 degrés hors du plan de rotation
- Partiellement invariant aux changements de l'éclairage
- Peut être calculée rapidement et efficacement.

2.5 Principales difficultés de la reconnaissance de visages

La reconnaissance de visage marque les nombreuses difficultés, parmi eux [9]:

2.5.1 Changement d'illumination

L'apparence d'un visage dans une image varie énormément en fonction de l'illumination de la scène lors de la prise de vue (voir figure 2.2). Les variations d'éclairage rendent la tâche de reconnaissance de visage très difficile. En effet, le changement d'apparence d'un visage du à l'illumination, se révèle parfois plus critique que la différence physique entre les individus, et peut entraîner une mauvaise classification des images d'entrée.

L'identification de visage dans un environnement non contrôlé reste donc un domaine de recherche ouvert. Les évaluations FRVT ont révélé que le problème de variation d'illumination constitue un défi majeur pour la reconnaissance de visage.



Figure 2.2 : Exemple de variation d'éclairage.

2.5.2 Variation de pose

Le taux de reconnaissance de visage baisse considérablement quand des variations de pose sont présentes dans les images.

La variation de pose est considérée comme un problème majeur pour les systèmes de reconnaissance faciale. Si la rotation est supérieure à 30° , la normalisation géométrique n'est plus possible (voir figure 2.3).



Figure 2.3 : Exemples de variation de poses.

2.5.3 Expressions faciales

Un autre facteur qui affecte l'apparence du visage est l'expression faciale (Voir figure 2.4). La déformation du visage qui est due aux expressions faciales est localisée principalement sur la partie inférieure du visage. L'information faciale se situant dans la partie supérieure du visage reste quasi invariable. Elle est généralement suffisante pour effectuer une identification. Toutefois, étant donné que l'expression faciale modifie l'aspect du visage, elle entraîne forcément une diminution du taux de reconnaissance. L'identification de visage avec expression faciale est un problème difficile qui est toujours d'actualité et qui reste non résolu.



Figure 2.4 : Exemples de variation d'expressions.

2.5.4 Présence ou absence des composants structurels

La présence des composants structurels telle que la barbe, la moustache, ou bien les lunettes (figure 2.5) peut modifier énormément les caractéristiques faciales telles que la forme, la couleur, ou la taille du visage. De plus, ces composants peuvent cacher les caractéristiques faciales de base causant ainsi une défaillance du système de reconnaissance. Par exemple, des lunettes opaques ne permettent pas de bien distinguer la forme et la couleur des yeux, et une moustache ou une barbe modifie la forme du visage.



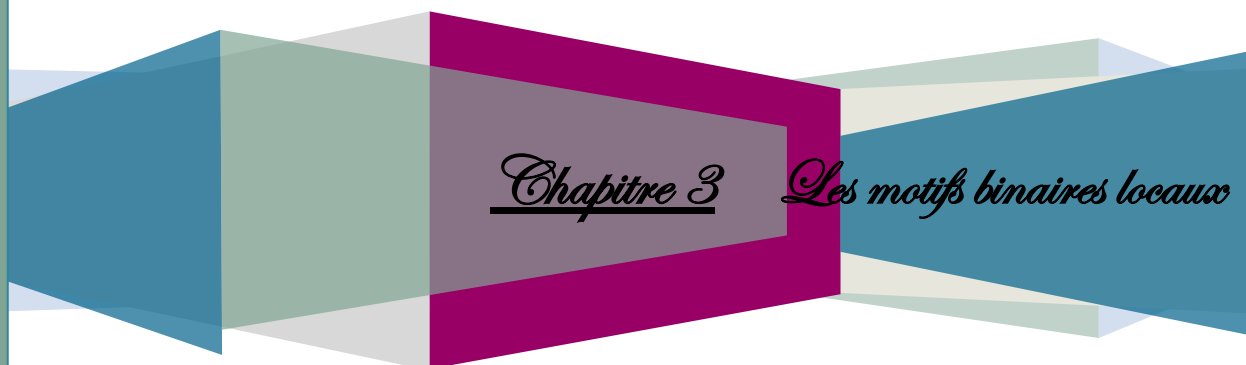
Figure 2.5 : Exemple de port de lunettes

2.5.4 Occultations partielles

Le visage peut être partiellement masqué par des objets dans la scène, ou par le port d'accessoire tels que lunettes, écharpe... Dans le contexte de la biométrie, les systèmes proposés doivent être non intrusifs c'est-à-dire qu'on ne doit pas compter sur une coopération active du sujet. Par conséquent, il est important de savoir reconnaître des visages partiellement occultés.

2.6 Conclusion

Dans ce chapitre, nous avons présenté le processus de reconnaissance des objets et ces différentes étapes .Cette étude nous a permis de constater que la reconnaissance de visage plus en plus l'intérêt de la communauté scientifique Enfin, nous avons mis en évidence les différentes difficultés de la reconnaissance de visages.



Chapitre 3

Les motifs binaires locaux

3.1 Introduction

Le motif binaire local (LBP) a été initialement conçu pour la description de la texture. Il est invariant aux transformations monotones en niveaux de gris qui est très important pour l'analyse de la texture. Aussi en raison du traitement de simplicité de calcul de l'image en temps réel .avec LBP il est possible de décrire la texture et la forme d'une image numérique. Cela se fait en divisant une image en plusieurs petites régions dont les caractéristiques sont extraites.

Ces caractéristiques consistent en des configurations binaires qui décrivent les environs de pixels dans les régions. Les caractéristiques obtenues à partir des régions sont concaténées en un seul histogramme de caractéristique, qui forme une représentation de l'image. Les images peuvent ensuite être comparées par mesure de la similitude (distance) entre leurs histogrammes.

Selon plusieurs études la reconnaissance de visage utilisant la méthode LBP fournit de très bons résultats, en termes de vitesse et les performances de discrimination. En raison de la façon dont la texture et la forme d'images sont décrites, la méthode semble être assez robuste contre les images de visage avec différentes expressions faciales, différentes conditions éclaircissantes, rotation d'image et le vieillissement des personnes.

3.2 Méthode des motifs binaire locaux

Le concept du LBP (**Local Binary Pattern**) est simple, il propose d'assigner un code binaire à un pixel en fonction de son voisinage. Ce code décrivant la texture locale d'une région est calculé par seuillage d'un voisinage avec le niveau de gris du pixel central. Afin de générer un motif binaire, tous les voisins prendront alors une valeur "1" si leur valeur est supérieure ou égale au pixel courant et "0" autrement (figure 1) [7].

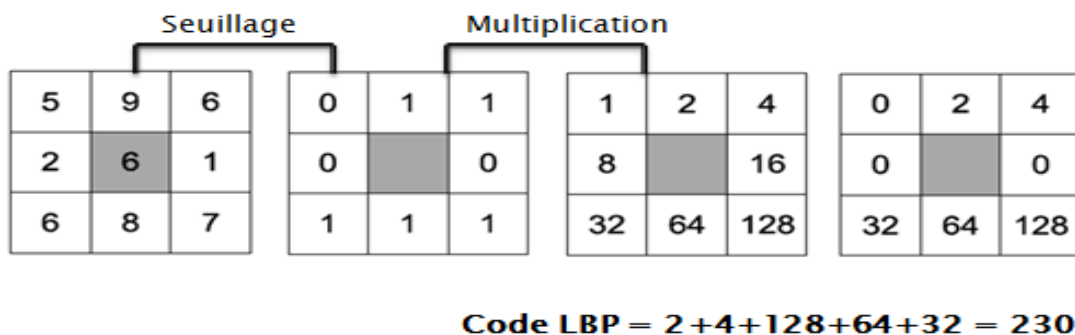


Figure 1 : Construction d'un motif binaire et calcul du code LBP.

Les pixels de ce motif binaire sont alors multipliés par des poids et sommés afin d'obtenir un code LBP du pixel courant. On obtient donc pour toute l'image, des pixels dont l'intensité se situe entre 0 et 255 comme dans une image à 8 bits ordinaire. Plutôt que de décrire l'image par la séquence des motifs LBP, on peut choisir comme descripteur de texture un histogramme de dimension 255[7].

Etant donné une position de pixel (x_y, y_c) LBP est définie comme un ensemble ordonné de comparaisons binaires des intensités de pixel entre le pixel central et ses pixels voisins. il s'exprime comme suit:

$$LBP(x_c, y_c) = \sum_{n=0}^7 s(l_n - l_c) 2^n \tag{3.1}$$

ou I_c correspond a la valeur de gris du pixel courant (de coordonnées x_c, y_c)
 I_n sont les valeurs de gris des 8 voisins entourant le pixel courant (I_c). et $S(k)$ est la fonction de seuillage défini comme suit :

$$S(k) = \begin{cases} 1, & \text{si } k \geq 0 \\ 0, & \text{si } k < 0 \end{cases}$$

3.3 Etapes d'extraction des motifs binaires locaux

Étapes d'extraction des motifs binaires locaux sont comme suit :

- **Etape 1** : les huit valeurs du voisinage du pixel à traiter sont extraites pour former une matrice 3x3 pixels.

9	1	4	2	6
7	8	9	2	7
6	6	5	3	3
8	1	4	7	1
4	6	2	1	3

- **Etape 2 :** Un seuillage est effectué sur la valeur des pixels du voisinage.

1	1	0
1		0
0	0	1

- **Etape 3:** La matrice des poids est obtenue à partir des valeurs de puissance de deux (2) appeler (**la matrice multiplicatrice**).

1	2	4
8		16
32	64	128

- **Etape 4:** une multiplication élément par élément est effectuée entre la matrice des poids et la matrice multiplicatrice.

1	2	0
8		0
0	0	128

- **Etape 5:** la sommation des valeurs de la matrice résultante de l'étape 4.

$$\text{LBP} = 1+2+8+128 = 139.$$

3.4 LBP pour la reconnaissance de visages

Une fois le code LBP calculé pour tous les pixels de l'image, on calcule l'histogramme de cette image LBP pour former un vecteur de caractéristiques représentant l'image faciale. En réalité, afin d'incorporer plus informations spatiales au vecteur représentant le visage, on Divise tout d'abord l'image codée par l'opérateur LBP en petites régions et l'histogramme est Construit pour chaque région. Finalement, on concatène tous les histogrammes des régions afin de former un grand histogramme représentant l'image des caractéristiques faciales.

3.5 Derivation

Le motif binaire local avec 8 pixels dans un bloc de 3 x 3 pixels, cette formulation générique de l'opérateur ne met pas de limites à la taille du voisinage au nombre d'échantillonnage des points [10].

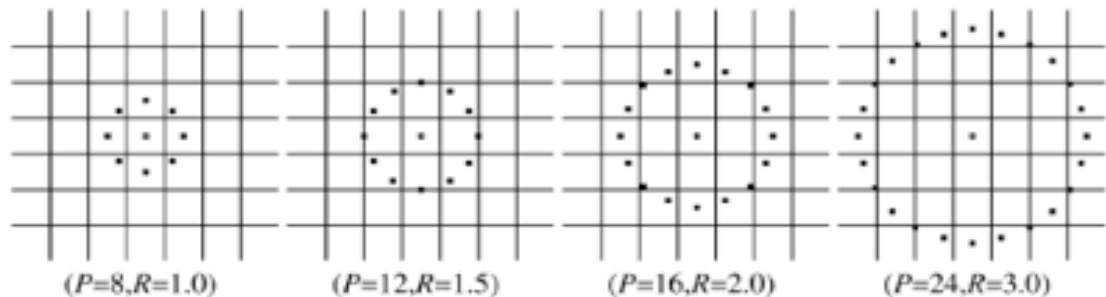


Figure 3.2: Exemple des LBP avec différent voisinages et rayons

Considérons une image $I(x, y)$ en niveaux de gris et soit g_c désignent le niveau de gris d'un pixel (x, y) , $g_c = I(x, y)$, et g_p désignent la valeur de gris d'un point de prélèvement (d'un point d'échantillon) dans un voisinage circulaire régulièrement espacé de points d'échantillonnage P et le rayon R autour du point (x, y) [2]:

$$g_p = I(x_p, y_p), p = 0, \dots, P - 1 \quad (3.2).$$

$$x_p = x + R \cos(2\pi p / P) \quad (3.3).$$

$$y_p = y - R \sin(2\pi p / P) \quad (3.4).$$

En supposant que la texture locale de l'image $I(x, y)$ est caractérisée par l'articulation la distribution de valeurs de gris de $P + 1$ ($P > 0$) pixels :

$$T = t(g_c, g_0, g_1, \dots, g_{p-1}). \quad (3.5).$$

Sans perte d'information, la valeur du pixel central peut être soustraite du voisinage :

$$T = t(g_c, g_0 - g_c, g_1 - g_c, \dots, g_{p-1} - g_c). \quad (3.6).$$

Dans l'étape suivante la distribution conjointe est évalué en considérant le pixel central pour

être statistiquement indépendante des différences, ce qui permet de factorisation de la distribution:

$$T \approx t(g_c) t(g_c, g_0 - g_c, g_1 - g_c, \dots, g_{p-1} - g_c). \quad (3.7).$$

Maintenant, le premier facteur $t(g_c)$ est la distribution d'intensité sur $I(x, y)$. Du point de vue de l'analyse des modèles de texture locales, il ne contient aucune information utile. Plutôt la distribution conjointe des différences

$$t(g_0 - g_c, g_1 - g_c, \dots, g_{p-1} - g_c). \quad (3.8).$$

Peut être utilisé pour modéliser la texture locale. Les différences $g_p - g_c$ sont invariantes des changements de la valeur de gris moyen de l'image, mais pas à d'autres changements dans les niveaux de gris.

Deuxièmement, afin de l'utiliser pour la classification de la texture du dictionnaire (**du table de codage**) doit être formé similaire aux autres méthodes. Afin de remédier à ces difficultés, seuls les signes des différences sont considérés:

$$t(s(g_0 - g_c), s(g_1 - g_c), s(g_{p-1} - g_c)). \quad (3.9).$$

Où $s(z)$ est la fonction de seuillage :

$$S(k) = \begin{cases} 1, & \text{si } k \geq 0 \\ 0, & \text{si } k < 0 \end{cases}$$

Puis, des poids binomiaux sont assignés pour chaque signe $s(z)$, et cette distribution des différences est transformée à un code LBP unique. Le code caractérise la texture locale autour d'un pixel (x_c, y_c) de l'image est :

$$LBP_{P,r}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p. \quad (3.10).$$

3.6 Lbp Invariant à la rotation

La définition de ce motif ordonné au fait que le voisinage construit soit circulairement symétrique a permis de définir un second motif, lui aussi invariant à toute transformation monotone de l'échelle des niveaux de gris mais également invariant à toute rotation de l'image, Pour tout pixel (x, y) , le calcul de ce motif est donné par :

$$LBP = \min \{ROR((x, y), i) | i [0, p - 1]\} \quad (3.11).$$

où $ROR(a, i)$ correspond au résultat de i décalages circulaires successifs vers la droite des bits du nombre a (codé sur P bits). LBP, quantifie ainsi les statistiques d'occurrence de modèles individuels invariants en rotation correspondant à certaines micro-caractéristiques de l'image [11].

La **Figure 3.3** , présente les 36 uniques motifs binaires locaux invariants en rotation possibles dans le cas de $P = 8$. A titre d'exemple, le motif "0" permet de détecter les tâches claires, le motif "8" les tâches sombres et les surfaces monotones, le motif "3" les coins et le motif "4" les contours.

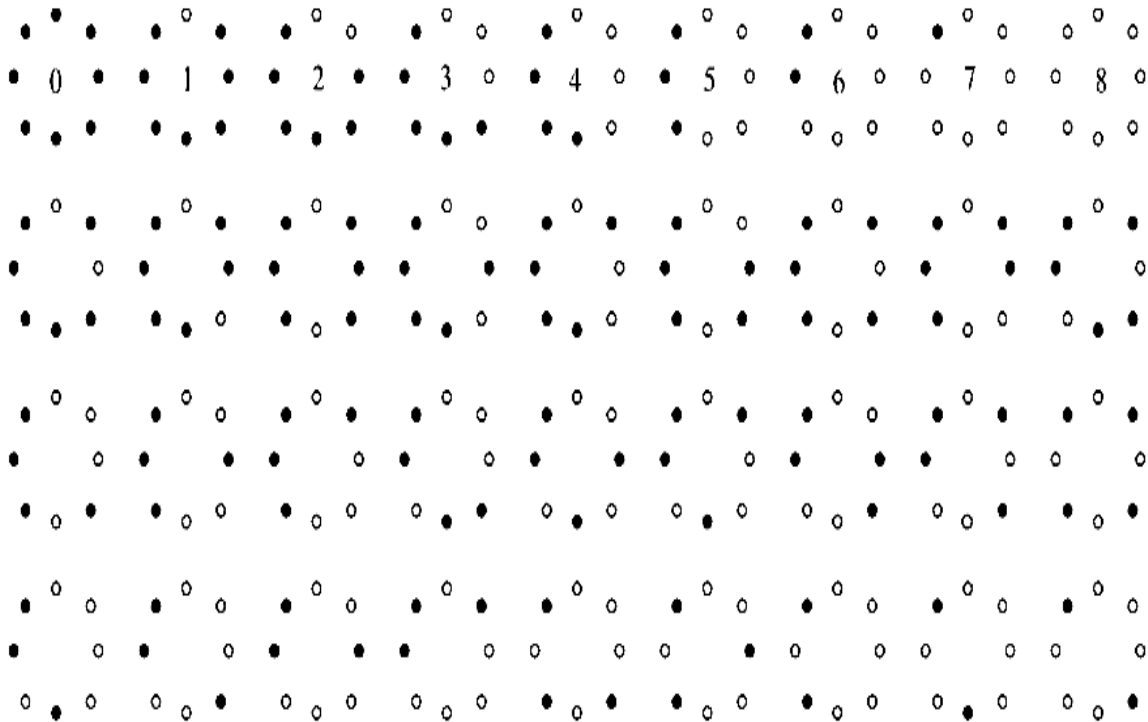


Figure 3.3 : les 36 uniques motifs binaires locaux invariants en rotation possibles dans $P = 8$

3.7 Motif binaire uniforme et non-uniforme

Un code LBP est uniforme s'il contient au plus deux transitions de bits de 0 a 1 ou vice-versa lorsque la chaine binaire est considéré circulaire. Le tableau suivant représente quelque code LBP uniforme et non-uniforme [12] :

Code LBP	Motif Uniforme
1111111	Uniforme
00001110	Uniforme
00000000	Uniforme
11001001	Non uniforme
01010011	Non Uniforme
01010101	Non Uniforme

Table 3.1 : Exemple des motifs binaire locaux uniforme et non uniforme

Dans l'exemple précédant : **00001110** ou **1111111** est un motif uniforme lorsque il possédant exactement 0ou2 transitions (01ou10) dans un parcours circulaire. Par contre **11001001** et **01010011** n'est pas uniforme lorsque il possédant un nombre de transitions supérieur a deux .

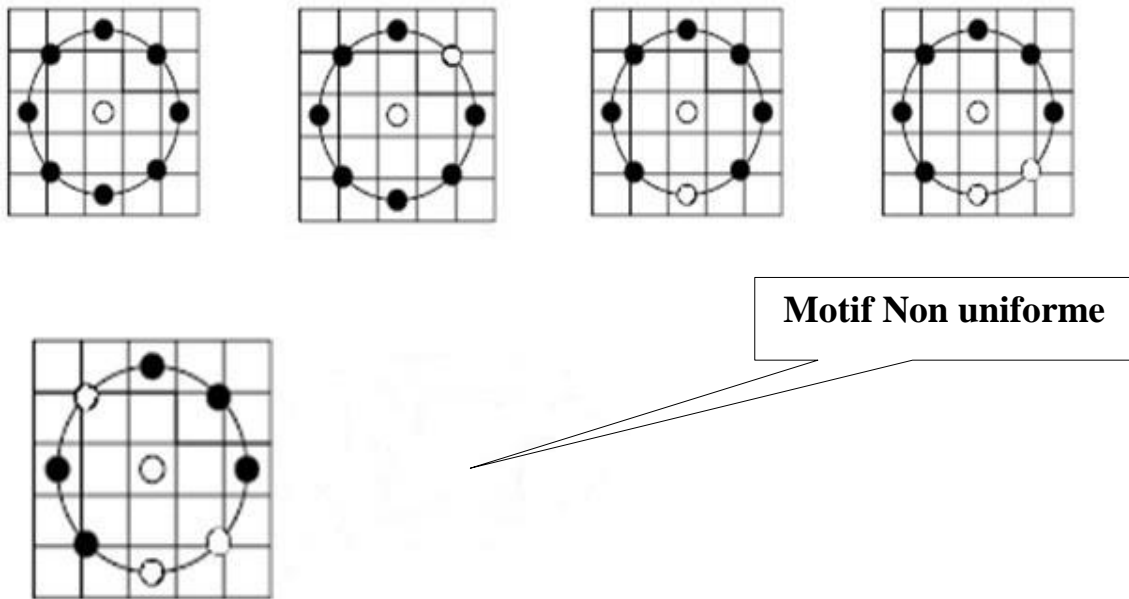


Figure 3.4 : des motifs binaire locaux uniforme et non uniforme

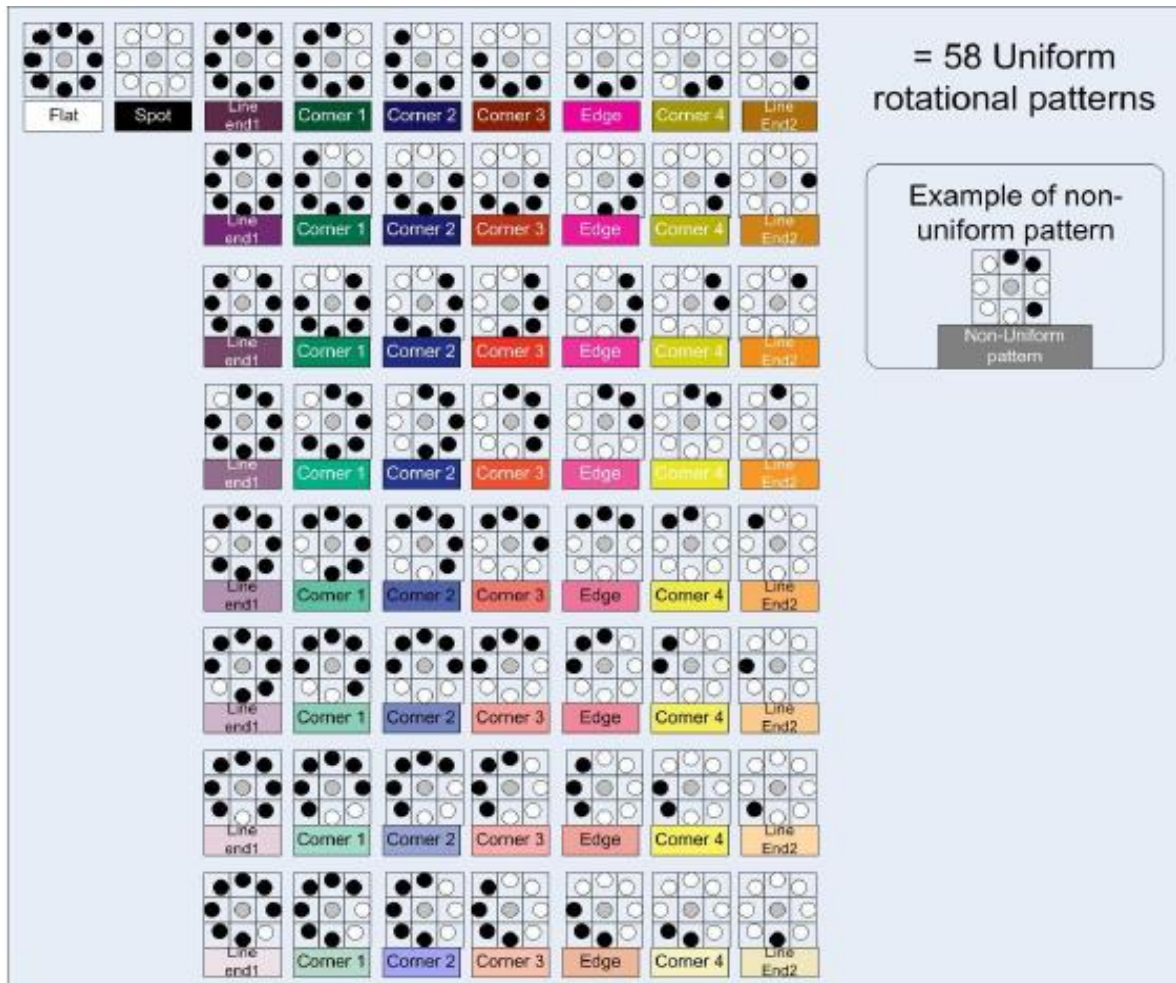


Figure 3.5 : Motifs binaire locaux uniforme

3.8 Histogramme LBP

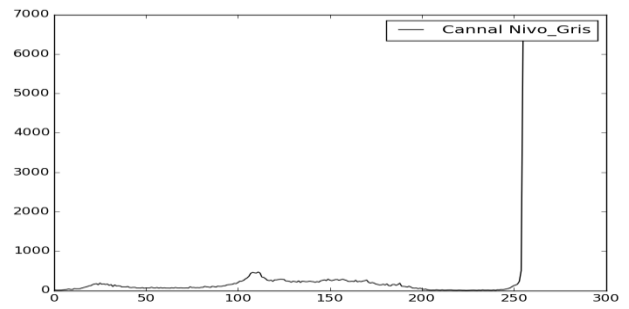
Apprêt le calcule LBP de chacun des pixels. Plutôt que de décrire l'image par la suite des motifs LBP, on va réduire la quantité de données en construisant un histogramme de dimension 59. Chacune des 58 premières dimensions contiendra le nombre d'occurrences de l'un des motifs uniforme. La dernière contiendra le nombre d'occurrence de tous les motifs non uniformes, ce regroupement permettant de réduire la dimension sans perdre trop d'information utile. Chaque image est découpée selon une grille régulière.

L'histogramme est calcule sur chacune des cellules. L'image est décrite par la concaténation de tous les histogrammes.

Alors que chacun des codes uniformes est regroupé dans un histogramme unique Par exemple, quand $P = 8$, nous avons 58 codes uniformes mais l'histogramme est de dimension 59. De même manière $P = 6$ produit un histogramme de dimension 33[13].



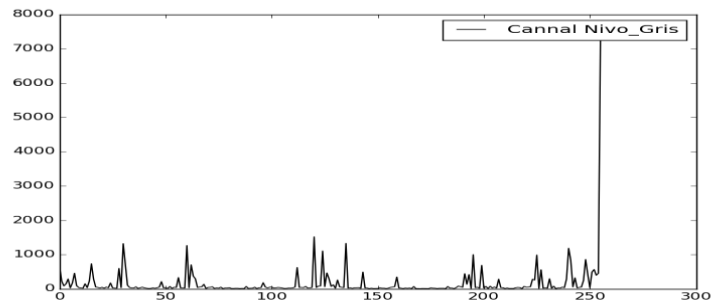
image



L'histogramme



Image LBP



L'histogramme LBP

Figure 3.6 : L'histogramme LBP .

L'histogramme est défini par :

$$H_{i,j} = \sum_{x,y} I\{f_i(x,y) = i\} I\{(x,y) \in R_j\}$$

$$i = 0, \dots, n - 1, j = 0, \dots, m - 1 \tag{3.12}.$$

Telle que :

- n la dimension de l'histogramme local.
- m le nombre de régions sur l'image.

3.8.1 distance entre histogrammes LBP

Etant donnés deux histogrammes de LBP h_1, h_2 de deux visages, l'étape suivante est d'utiliser une métrique pour calculer la similarité entre ces deux histogrammes. En testant les trois métriques [7]:

- **Chi carré statistique (X2):**

ont observé que la première métrique permet d'obtenir les meilleurs résultats :

$$X^2_{h^1, h^2} = \sum_{i=0}^n \left(\frac{(h_i^1 - h_i^2)^2}{h_i^1 + h_i^2} \right) \quad (3.13).$$

- **Intersection histogramme :**

$$D(S, M) = \sum_{i=0}^n \min(S_i, M_i) \quad (3.14).$$

- **Log-varaisemblance Statistique (log-likelihoodstatistic) :**

$$L(S, M) = -\sum_{i=0}^n S_i \log M_i \quad (3.15).$$

Il existe plusieurs métriques de similarité:

- **Distance de Mahalanobis**

Cette distance prend en compte la corrélation entre les distributions des classes. Elle est ainsi définie par :

$$D_m = \sqrt{(f_1 - f_2)^T C^{-1} (f_1 - f_2)} \quad (3.16).$$

Où C est la matrice de covariance. Dans les cas où les dimensions des caractéristiques sont indépendantes, C ne comporte que les variances et la distance de Mahalanobis se simplifie sous la forme :

$$D_m = \frac{\sum (f_1(i) - f_2(i))^2}{c_i} \quad (3.17).$$

Si C est la matrice identité, est la distance euclidienne.

- **Intersection d'histogrammes :**

Cette mesure est l'une des premières distances utilisée dans la recherche d'image Par le contenu. Elle a été proposée par Swain et Ballard mesurant la partie commune entre deux histogrammes .

Etant donné deux histogrammes h_1 et h_2 :

$$D_{Intersec} = \frac{\sum_{i=0}^n \min(h_1(i), h_2(i))}{\sum_{i=0}^n h_2(i)} \quad (3.18).$$

- **Earth Mover Distance :**

Earth Mover Distance consiste à minimiser le coût de transformation d'une distribution en une autre sous certaines contraintes de déplacement des classes de descripteurs.

$$D_{EMD} = \frac{\sum_{ij} g_{ij} d_{ij}}{\sum_{ij} g_{ij}} \quad (3.19).$$

- d_{ij} représente la disimilarité entre deux indices (i, j)
- g_{ij} est le flot optimal entre deux distributions dont le coût total est :

$$D_{EMD} = \sum_{ij} g_{ij} \quad (3.20).$$

Le coût est minimisé sous les contraintes suivantes :

- $\forall_{ij}, g_{ij} \geq 0$.
- $\forall_j, f_2(j) \geq \sum_i g_{ij}$.
- $\forall_i, f_1(i) \geq \sum_j g_{ij}$.
- $\min(f_1(i), f_2(i)) = \sum_j g_{ij}$.

- **Distance de Minkowski :**

La distance de Minkowski est une famille de distances vectorielles. Soit deux vecteurs (f_1, f_2) de caractéristiques, elle s'exprime par :

$$d^p(f_1, f_2) = (\sum_{i=1}^n |f_1(i) - f_2(i)|^p)^{1/p} \quad (3.21).$$

- p est le facteur de Minkowski
- n la dimension de l'espace caractéristique.

La distance Euclidienne est un cas particulier de cette distance où $p=2$, de même que la distance de Manhattan ($p=1$).

3.9 Avantages de motifs binaires locaux

La méthode de motifs binaires locaux offre de bonnes performances et contient des informations structurelles et statistiques. L'opérateur LBP peut être invariant aux rotations et changements d'échelles et offre une tolérance face aux changements de luminosité. De plus, c'est une méthode de faible complexité qui rend possible l'analyse d'image 2D en temps réel.

3.10 Inconvénients de motifs binaires locaux

L'inconvénient de descripteur LBP est le nombre de paramètres à fixer. Effectivement, les versions de ce descripteur sont nombreuses, suivant la valeur fixée N correspondant au nombre de pixels voisins et suivant le rayon R choisi.

3.11 Les Motifs ternaires locaux

LBP de sont résistants aux effets d'éclairage dans le sens où ils sont insensibles aux transformations monotones de niveau de gris, et ils ont été montrés pour avoir un pouvoir discriminant élevé pour la classification de texture.

Il s'agit de la généralisation des local binary patterns au système ternaire. Elle a été proposée par Tan et Triggs [14] comme solution au problème de sensibilité qu'éprouve le LBP face au bruit aléatoire et celui de quantification. Le principe est le suivant : alors que les LBP appliquaient un seuil égal à la valeur du pixel central, la conversion en Local Ternary Patterns

(LTP) attribue à la valeur 0 aux pixels dont la valeur se trouve dans un voisinage de la valeur du pixel central, 1 à ceux dont la valeur est au-delà de ce voisinage et -1 à ceux dont la valeur est en dessous.

La formulation mathématique est la suivante pour u un pixel périphérique d'un entourage à convertir, i_c la valeur du pixel central et t le voisinage [14] :

$$LTP_{p,r} = \sum_{i=0}^{P-1} s'(p_i - p_c) \times 3^i \quad (3.22).$$

$$s'(u, i_c, t) = \begin{cases} 1, & u \geq i_c + t \\ 0, & |u - i_c| < t \\ -1, & u \leq i_c - t \end{cases}$$

3.11.1 extraction des motifs ternaires locaux

Comme fait pour l'opérateur LBP basique, une illustration de l'opérateur LTP basique est faite comme suit [15] :

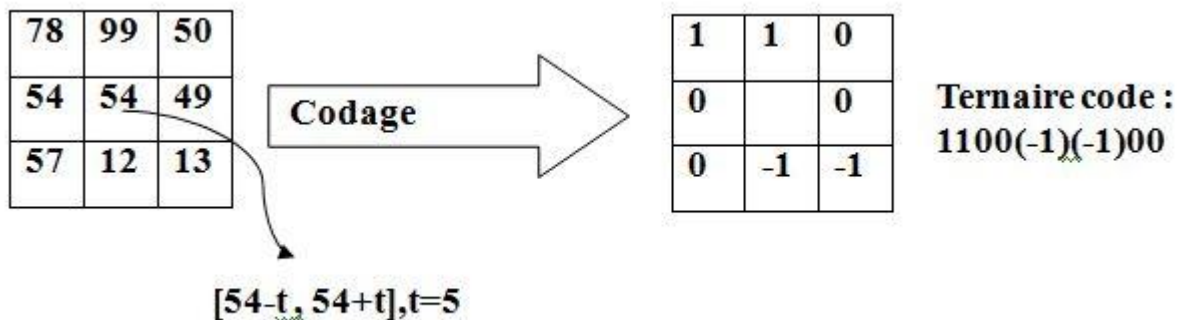


Figure 3.7 : L'opérateur LTP .

Ensuite, ce code ternaire peut être transformé soit en un nombre réel soit en code binaire pour passer à la phase suivante. Dans le premier cas on pourrait utiliser un code de valeurs 3^n similaire au code binaire 2^n . Tan et Triggs ont quant à eux divisé le code ternaire en deux codes binaires traités séparément et rassemblés ensuite lors de la phase de comparaison.

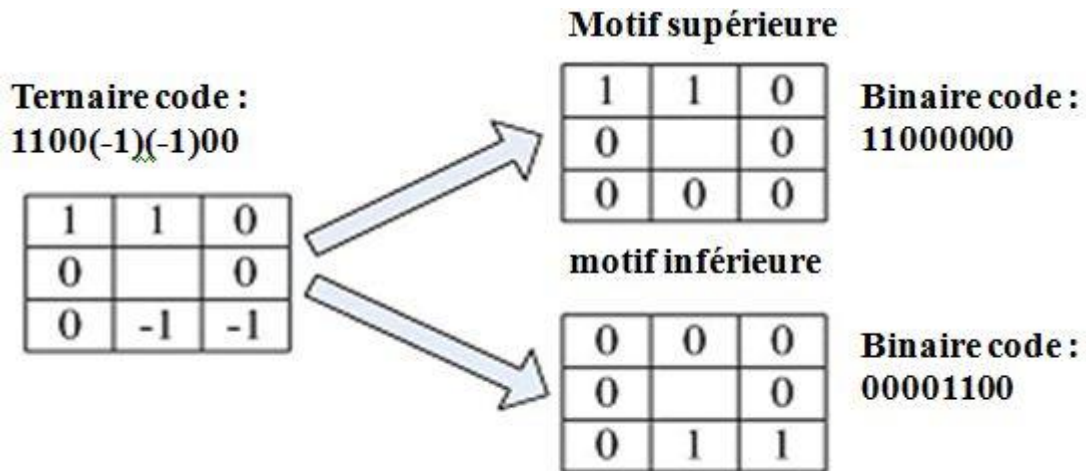


Figure 3.8 : Operateur divisé le code ternaire en deux codes binaires.

Cette méthode à l'avantage de garder le système simple d'élimination des patterns non-niformes.

3.12 Conclusion

Ce chapitre a été consacré à la présentation des deux descripteurs LBP et LTP ces technique permettent de diminuer le nombre de composantes toutes en gardant l'information caractérisant l'objet à analyser sans perte d'informations significatives. Le descripteur LBP est le plus important pour la reconnaissance de visage.



Chapitre 4

Conception et réalisation

4.1. Introduction :

L'objectif de ce travail est de construire un modèle et d'étudier un système de reconnaissance d'objets (appliquer sur les visages). Dans ce chapitre on va présenter notre application qui permet d'extraire les points d'intérêt d'une image de test en utilisant le descripteur Sift qui calcule l'histogramme des descripteurs LBP d'une image, de détecter le visage le plus similaire à l'image test en utilisant différents types de métriques pour mesurer la similarité entre les histogramme de notre base d'images et l'image test.

4.2. Conception du système

4.2.1. Processus de reconnaissances des objets

Le schéma suivant illustre le fonctionnement général de notre système de détection.

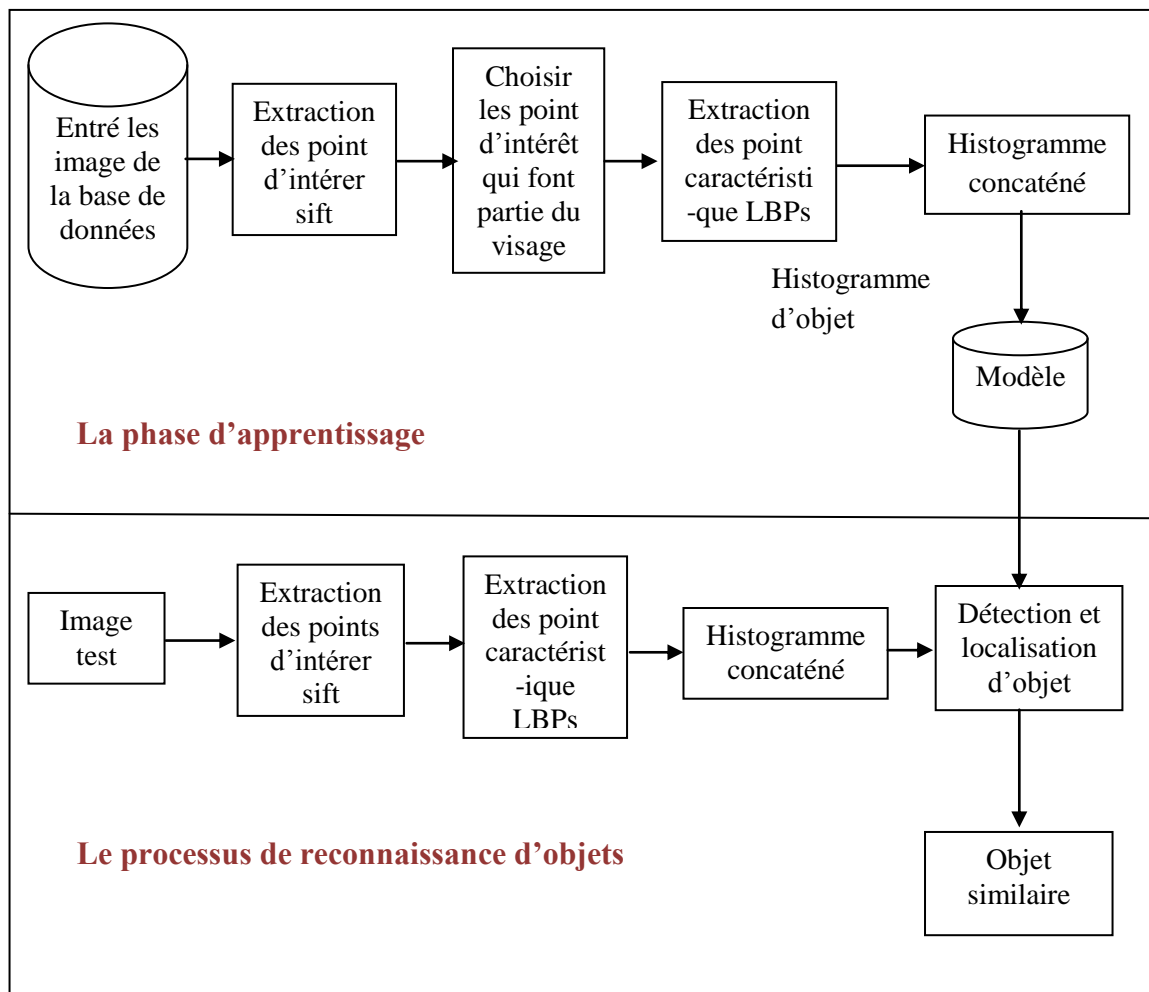


Figure 4.1 : Processus de reconnaissance d'objet avec la méthode LBP

4.2.1.1. La Base de données ORL

Conçu par AT&T laboratoires de l'université de Cambridge en Angleterre, la base de données ORL (Olivetti Research Laboratory) est une base de données de référence pour les systèmes de reconnaissance automatique des visages. En effet tous les systèmes de reconnaissance de visages trouvés dans la littérature ont été testés par rapport à l'ORL.

La base de données ORL est constituée de 40 individus, chaque individu possède 10 Poses, donc la base contient 400 images. Les poses ont été prises sur des intervalles de temps différents pouvant aller jusqu'à trois mois. L'extraction des visages à partir des images a été faite manuellement. Nous présenterons dans ce qui suit les figures montrant les spécificités de la base de données de référence ORL.

Taille d'image : 92×112 pixels

Format de l'image : JPG



4.2.2. Phase d'apprentissage

4.2.2.1. Extraction des points d'intérêts Sift

Dans cette étape d'extraction des points d'intérêts on a utilisé la fonction prédéfinie de l'opencv qui retourne une liste de coordonnées (x,y)

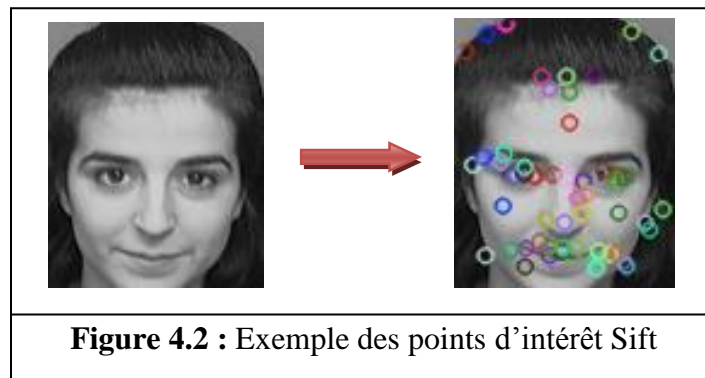


Figure 4.2 : Exemple des points d'intérêt Sift

4.2.3. Choix des points d'intérêts qui font partie du visage

Pour cette étape de choix des points d'intérêts résultantes dans l'étape précédente on a construit la partie de l'application qui permet de consulter les point d'intérêts un par un puis choisir les points qui font partie du visage en cliquant sur bouton sauvegarder

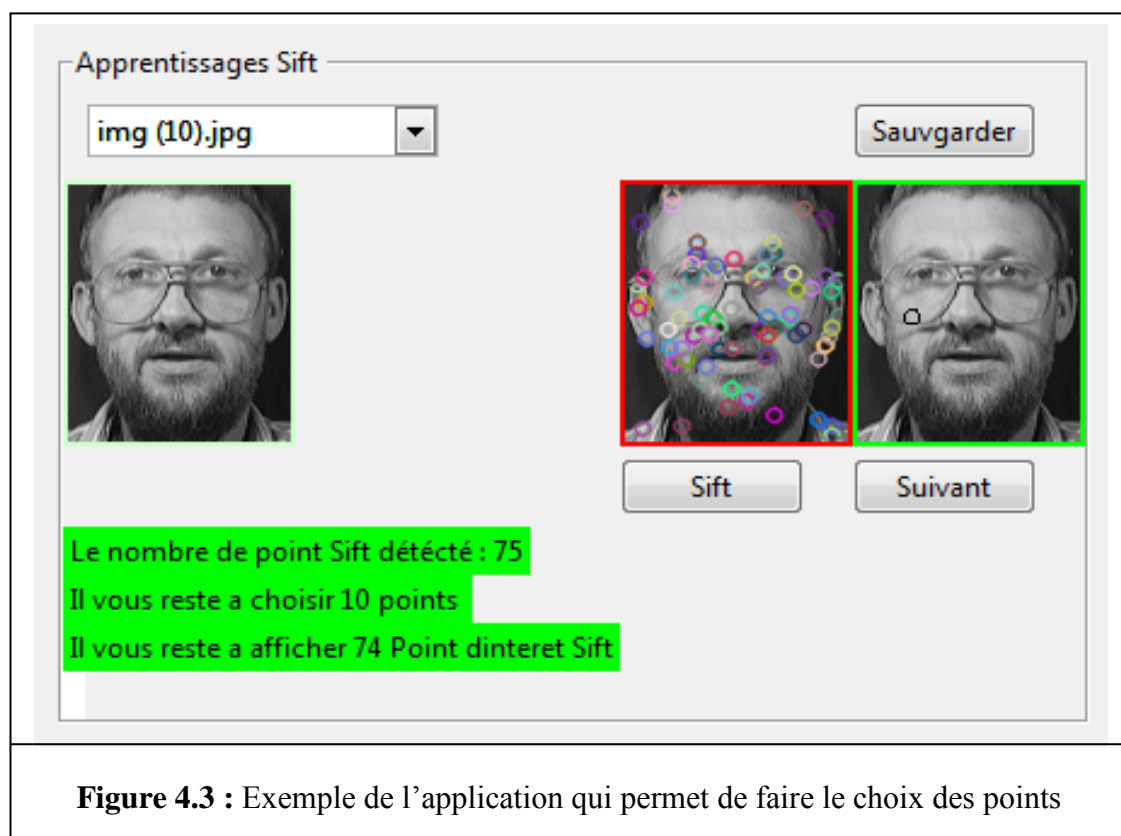


Figure 4.3 : Exemple de l'application qui permet de faire le choix des points

4.2.4. Extraction des points caractéristiques LBPs

Dans cette étape nous avons fait l'extraction des caractéristiques LBP comme suit :

Pour chaque pixel choisi dans l'étape précédente on a récupéré les 15 voisins du pixel courant (région 15×15) puis calculé les LBPs de chaque pixel qui font partie de cette région.

Puis construire les histogrammes concaténés qui représentent ces régions de l'image

- Calcul des LBPs

Nous avons traité plusieurs variantes des LBPs, dans notre application on a programmé trois types de LBPs : LBP 4.1, LBP 8.2 et LBP 8.2^(u2)

4.2.5. Détection et localisation

Dans notre système la détection et la reconnaissance de l'image nécessite un modèle et un classifieur d'apprentissage, nous avons choisie dans notre travail le classifieur K plus proche voisin.

Premièrement nous avons calculé la distance entre tous les histogrammes LBPs des différentes images de la base d'apprentissage et l'histogramme de l'image de teste, puis on cherche la distance minimale qui représente l'image la plus similaire à l'image de teste.

Après calcul de la distance on va maintenant vérifier si il existe une image similaire à l'image de teste pour cela on a un seuil de détection il y'a une distance inférieure ou égale au seuil cela veut dire qu'il existe au moins une image similaire à l'image de teste sinon il n'y a pas d'image similaire à l'image de teste.

4.2.6. Classification

Après avoir calculé la distance entre les histogrammes de la base d'apprentissage et l'image de teste on utilise une métrique de similarité. Dans cette phase nous avons classé les distances par ordre croissant puis gardé (affiché) les dix premières images de la base d'apprentissage.

4.2.2. Mesure de Similarité

Dans de nombreuses applications, il est utile de calculer une distance (ou plus généralement une mesure de similarité/dissimilarité) entre histogrammes. On ce qui nous concerne c'est

pour trouver des images similaires à une image requête. On définit pour cela trois types de distances pour mesurer si deux histogrammes sont « proches » l'un de l'autre.

Soit h_1 et h_2 deux histogrammes de même taille N (avec le même nombre de classes)

- **Corrélation**

$$d(h_1, h_2) = \frac{\sum_{i=1}^N \bar{h}_1(i) \bar{h}_2(i)}{\sqrt{\sum_{i=1}^N \bar{h}_1(i)^2 \sum_{i=1}^N \bar{h}_2(i)^2}} \quad \text{où} \quad \bar{h}(i) = h(i) - \frac{1}{N} \sum_{i=1}^N h(i)$$

Algorithme 4.2 : Algorithme de la fonction de Corrélation

Déclaration :
 Soit h_1 et h_2 deux histogrammes de même taille N

Fonction Corrélation (h_1, h_2)

Début

Distance $\leftarrow 0$

Som1, Som2, Som3, Som4, Som5 $\leftarrow 0,0,0,0,0$

Pour $i \leftarrow 1$ **jusqu'à** N **faire :**

Som1 \leftarrow Som1 + ($h_1[i]$)

Som2 \leftarrow Som2 + ($h_2[i]$)

Fin pour

Pour $j \leftarrow 1$ **jusqu'à** N **faire :**

Som3 \leftarrow Som3 + ($h_1[j] - 1/N(\text{Som1}) \times (h_2[j] - 1/N(\text{Som2}))$)

Som4 \leftarrow Som4 + ($h_1[j] - 1/N(\text{Som1})$)²

Som5 \leftarrow Som5 + ($h_2[j] - 1/N(\text{Som2})$)²

Fin pour

Retourner Som3 / $\sqrt{\text{Som4} \times \text{Som5}}$

Fin

Fin fonction

- **Intersection entre histogrammes (histogramme intersection)**

$$d(h_1, h_2) = \sum_{i=1}^N \min [h_1(i), h_2(i)]$$

Algorithme 4.1 : Algorithme de la fonction de l'Intersection entre histogrammes

Déclaration :

Soit h_1 et h_2 deux histogrammes de même taille N

Fonction Intersection histogrammes (h_1, h_2)

Début

Distance $\leftarrow 0$

Pour $i \leftarrow 1$ **jusqu'à** N **faire :**

Distance \leftarrow Distance + min($h_1[i]$, $h_2[i]$)

Fin pour

Retourner Distance

Fin

Fin fonction

- **Chi carré statistique (X^2) (Chi square statistic)**

$$d(h_1, h_2) = \sum_{i=1}^N \frac{[h_1(i) - h_2(i)]^2}{h_1(i) + h_2(i)}$$

Algorithme 4.2 : Algorithme de la fonction de Chi carré statistique(X^2)

Déclaration :

Soit h_1 et h_2 deux histogrammes de même taille N

Fonction Chi carré statistique (h_1, h_2)

Début

Distance $\leftarrow 0$

Pour $i \leftarrow 1$ **jusqu'à** N **faire :**

Si $h_1[i] + h_2[i] \neq 0$ **alors :**

Distance \leftarrow Distance + (($h_1[i] - h_2[i]$)² / ($h_1[i] + h_2[i]$))

Fin Si

Fin pour

Retourner Distance

Fin

Fin fonction

- **Log-vraisemblance Statistique (Log-likelihood statistic)**

$$d(h_1, h_2) = - \sum_{i=1}^N (h_1(i) \times \log(h_2(i)))$$

Algorithme 4.3 : Algorithme de la fonction de Log-vraisemblance Statistique

Déclaration :

Soit h_1 et h_2 deux histogrammes de même taille N

Fonction Log-vraisemblance Statistique (h_1, h_2)

Début

Distance \leftarrow 0

Pour $i \leftarrow 1$ **jusqu'a** N **faire :**

Si $h_2[i] \neq 0$ **alors :**

Distance \leftarrow Distance + ($h_1[i] \times \log h_2[i]$)

Fin Si

Fin pour

Retourner - Distance

Fin

Fin fonction

De nombreuses autres mesures sont possibles, notamment la divergence de **Kullback-Leibler** et la distance de **Bhattacharyya**, puisque la comparaison d'histogrammes se ramène à une mesure de dissimilarité entre deux distributions de probabilités.

4.4. Environnement et outils de travail

4.4.1. Environnement de programmation

Pour nos besoins de programmation on a choisi Windows7 puisque notre outil de travail est compatible avec la version du Windows qu'on utilise.

4.4.1.1. Langage de programmation Python

Python est un langage de programmation interprété et libre (comme dans "logiciel libre") créé par Guido van Rossum en 1990.

Il s'agit d'un langage multi-paradigme, ce qui signifie qu'au lieu d'imposer aux développeurs un type de programmation précis, il leur laisse adopter celui de leur choix. Il permet ainsi la programmation orientée objet, impérative structurée ou encore fonctionnelle. Python gagne partout du terrain car il facilite le développement d'applications, de scripts et de prototypes.

4.4.1.2. OpenCV

OpenCV (Open Source Computer vision) est une bibliothèque graphique libre, initialement développée par Intel et maintenant soutenu par la société de robotique Willow Garage, spécialisée dans le traitement d'images en temps réel. Elle est livrée avec une Interface de programmation en C, C++, Python et Androïde.

La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées, elle propose la plupart des opérations classiques en traitement d'images :

- lecture, écriture et affichage d'image;
- calcul de l'histogramme des niveaux de gris ou d'histogramme de couleur;
- lissage, filtrage.

4.4.1. Présentation de l'application

Cette partie du chapitre est consacré à la conception de l'application qui permettra d'identifier des personnes par reconnaissance faciale. Le programme de reconnaissance faciale sera codé en Python. Il utilisera les bibliothèques OpenCV pour pouvoir utilisé le descripteur Sift

Le rôle de l'application sera d'identifier, parmi une base de visages connus, lequel se rapproche le plus d'un visage passé en paramètre. Il utilisera la technique dite LBP (local binary pattern). Il devra également indiquer à quel point il s'en similaire en s'appuyant sur un seuil. Puis elle affichera dans une fenêtre le visage reconnu s'il existe.

L'interface de notre application est fractionnée sur quatre parties, chaque partie est caractérisée par des fonctionnalités différentes aux autres.

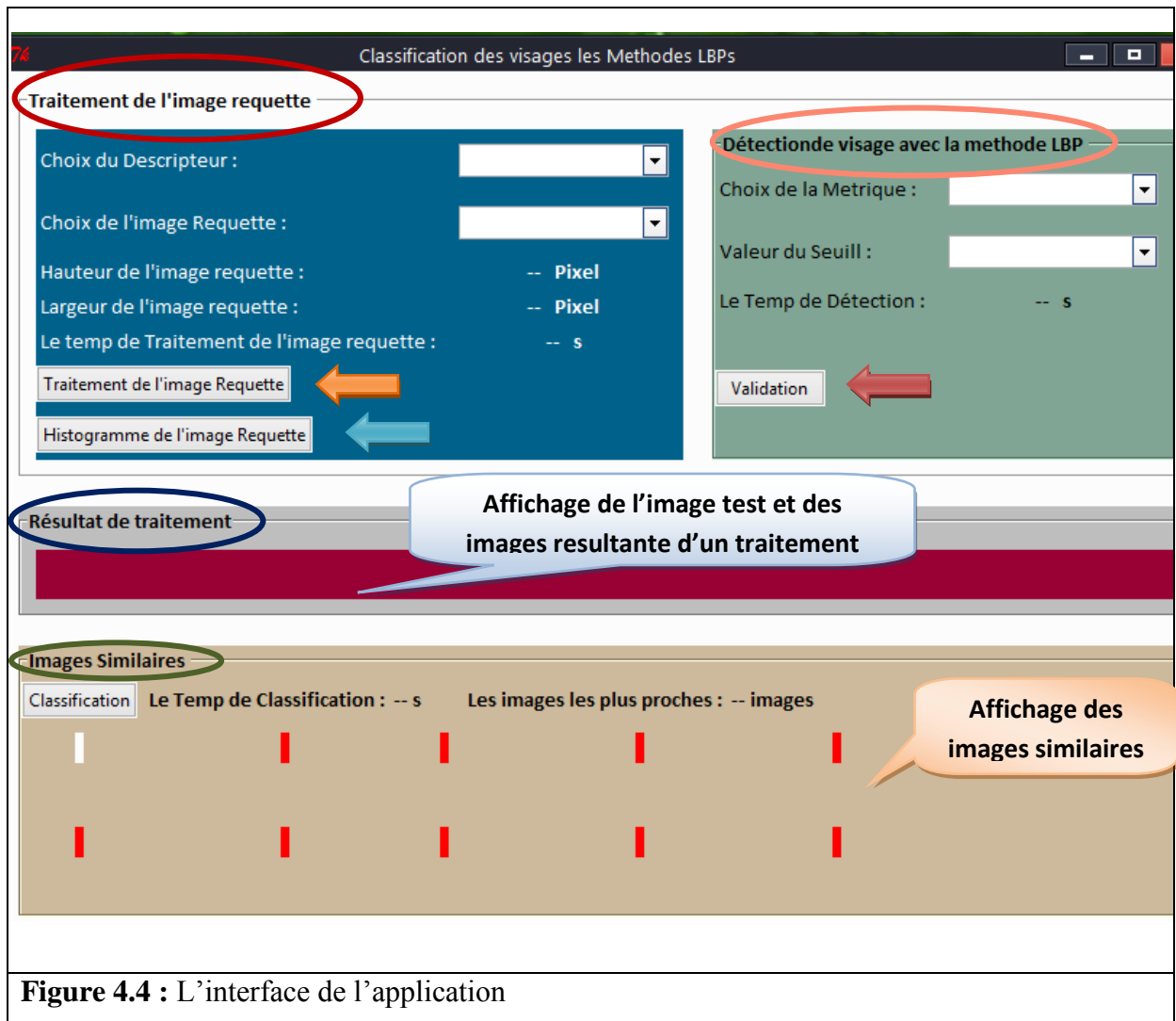


Figure 4.4 : L'interface de l'application

- **La partie de chargement et traitement de l'image** (cercle de couleur grenat)

C'est pour charger le type du motif LBP et l'image de teste en niveau de gris, un affichage de l'image ainsi que sa hauteur et sa largeur.

On cliquant sur le bouton «**Traitement de l'image Requette**» (fleche de couleur orange) cela va actionner le calcul de LBP de l'image de teste puis afficher le temp de traitement et l'image LBP qui correspond a l'image teste.

Elle comporte aussi un bouton «**Histogramme de l'image Requette**» (fleche de couleur bleu) pour la création et l'affichage de l'histogramme LBP de l'image requête.

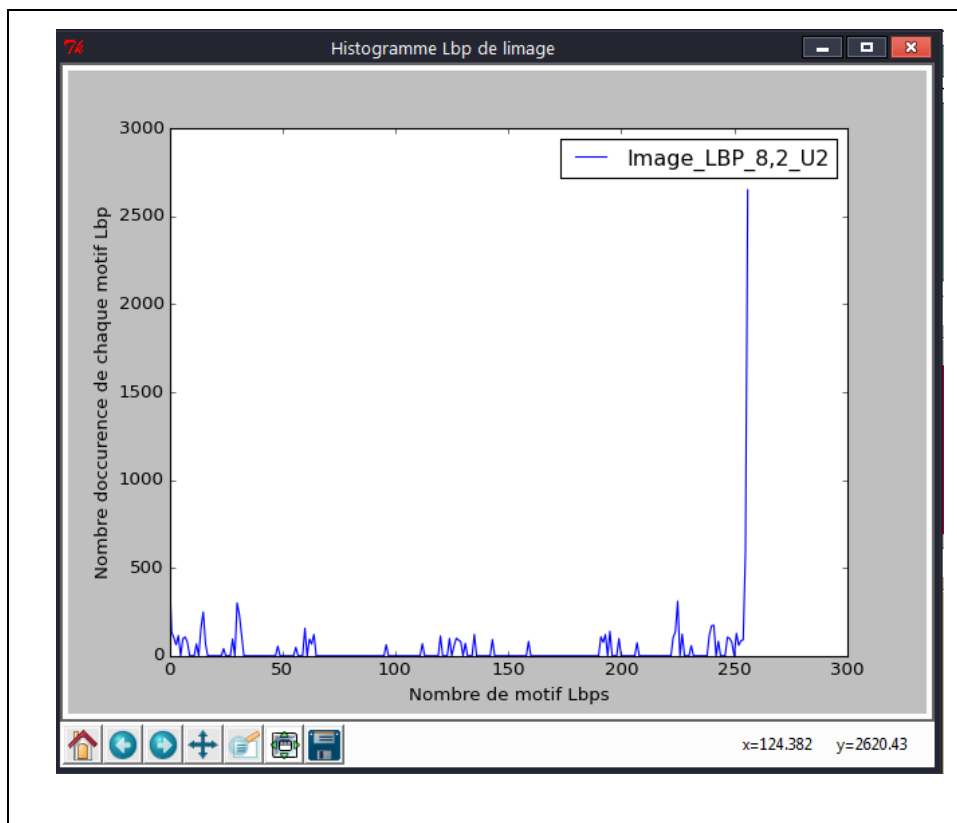
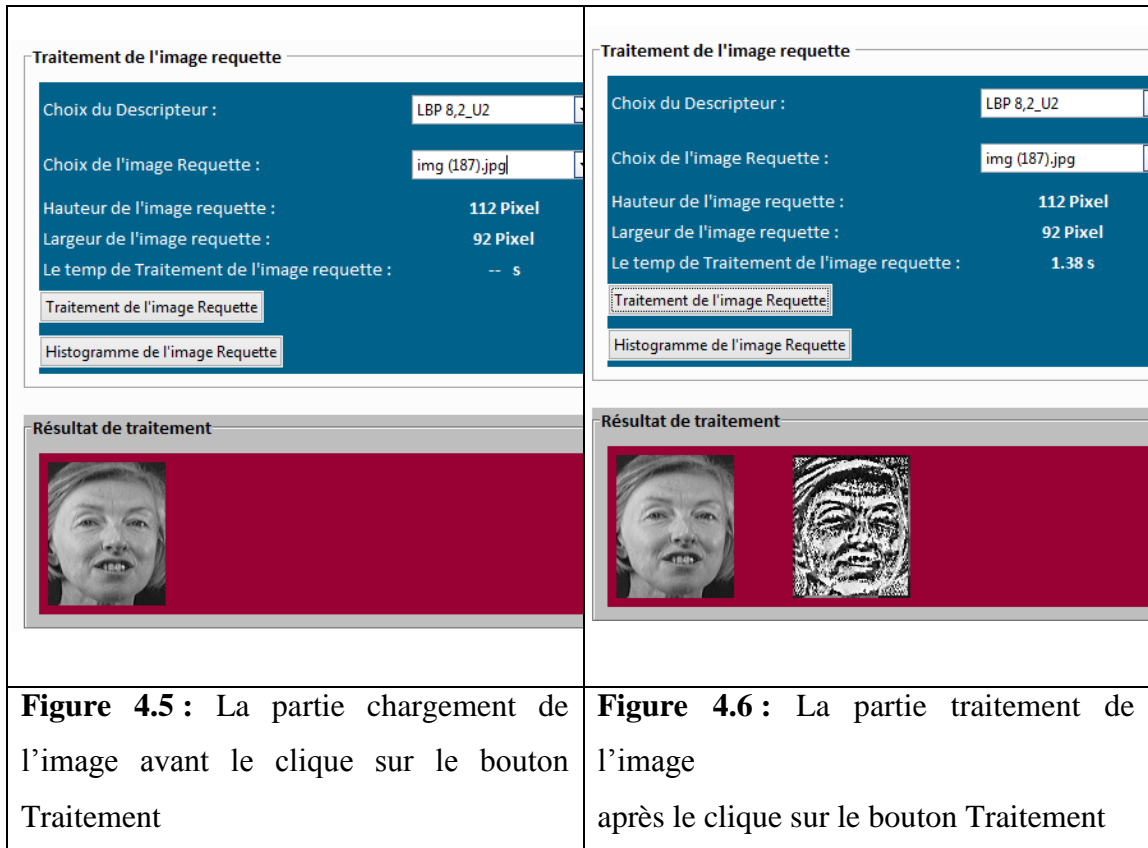


Figure 4.7 : Histogramme LBP de l'image requête

- La partie Détection** (cercle de couleur orange)

cette partie de l'application contient un choix de métrique, un **seuil** qui permettra de détecté ou non une image similaire à l'image requête puis afficher l'image similaire et sa distance avec l'image requête (si l'image similaire existe sinon pas de détection) et le nombre d'image similaire a l'image requête (qui ont une distance inférieur ou égale au seuil), un bouton «**Validation**» (fleche de couleur rouge) qui permet de calculé la distance entre l'histogramme de l'image requête et tous les histogrammes des images de la base d'apprentissage puis affiché le temps de détection.

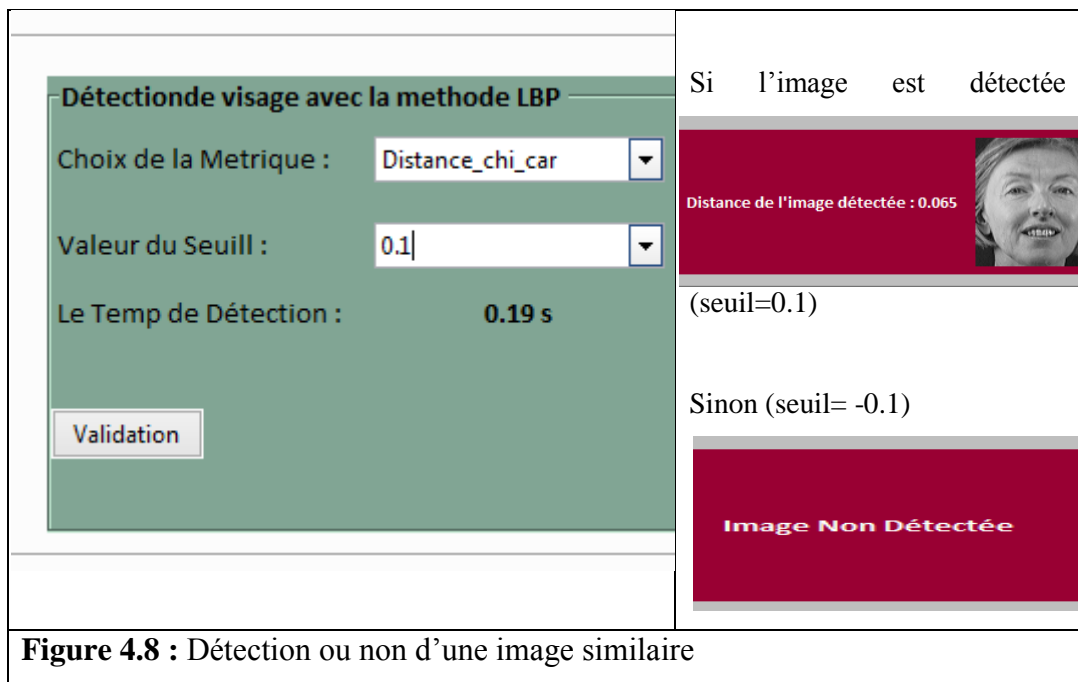


Figure 4.8 : Détection ou non d'une image similaire

- La partie Classification** (cercle de couleur vert)

Elle est conçue pour classifié les images de la base d'apprentissage. Cette partie de l'application est composé d'un bouton «**Classification**» pour classer et afficher les dix première images de la base d'apprentissage qui sont proche de l'image requête en termes de distance. Ce bouton affiche aussi le temps de classification

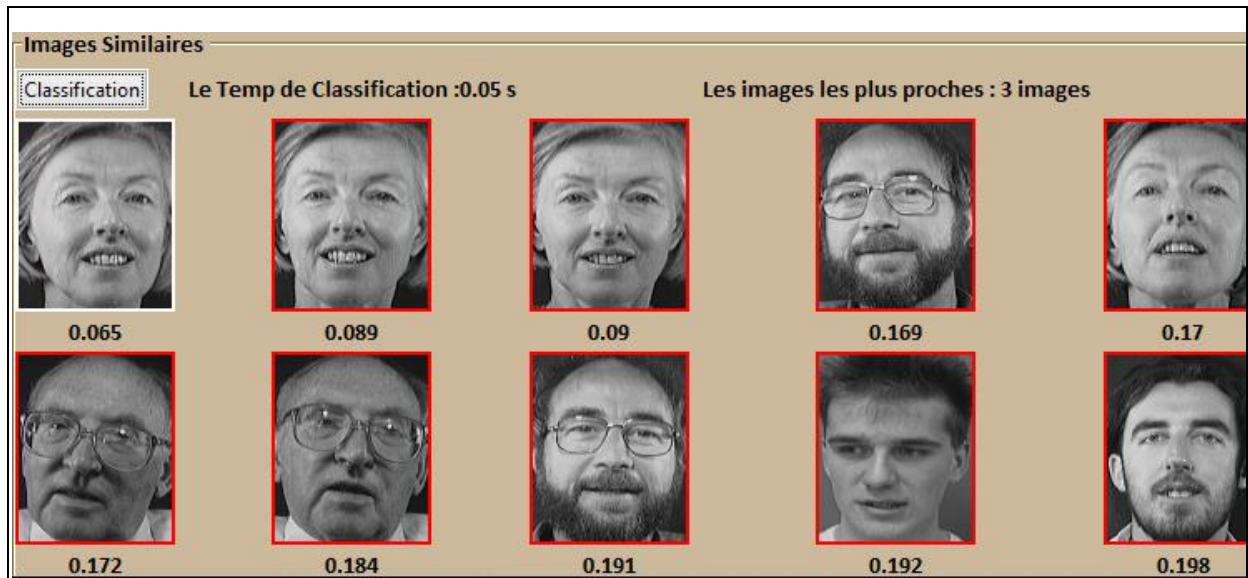


Figure 4.9 : Classifications après détection et classement par le plus proche voisin

4.5 Expérimentation et résultats

Dans cette partie nous avons évalué notre système de reconnaissance/détection des visages, pour faire les tests on a utilisé 40 images et on a aussi une base d'apprentissage qui contient 80 images (on a 20 personnes et 4 images avec différentes positions pour chaque personne)

4.5.1 Mesure d'évaluation

Évaluer les performances d'un système de reconnaissance/détection des visages est un enjeu de grande importance, les mesures les plus courantes pour évaluer un système sont le temps de réponse et l'espace utilisé. Plus le temps de réponse est court, plus l'espace utilisé est petit, et plus le système est considéré bon. Mais pendant longtemps, le critère retenu pour évaluer ces performances a été le taux de bonne classification, c'est-à-dire le nombre d'éléments d'une base de test correctement classés.

Les mesures que nous allons évoquer utilisent la matrice de confusion, **tableau 4.1**, qui permet la différenciation des erreurs selon chaque classe en vue d'évaluer un classifieur.

❖ Matrice de confusion

	Décision Positifs	Décision Négatifs
Étiquette Positifs	Vrai Positifs, VP	Faux Négatifs, FN
Étiquette Négatifs	Faux positifs, FP	Vrai Négatifs, VN

Tableau 4.1: Matrice de confusion

Dans notre cas on a un :

- ❖ **Vrai Positifs** : c'est une image jugée similaire à l'image test et qui correspond à la même personne de l'image test
- ❖ **Faux Négatifs** : c'est une image jugée non similaire à l'image test mais qui est pourtant la même personne de l'image test
- ❖ **Faux Positifs** : c'est une image jugée similaire à l'image test mais qui n'est pas la même personne de l'image test
- ❖ **Vrais Négatifs** : c'est une image jugée similaire à l'image test mais qui n'est pas la même personne de l'image test

❖ Rappel

$$\text{Rappel} = \frac{VP}{VP+FN}$$

❖ Précision

$$\text{Précision} = \frac{VP}{VP+FP}$$

❖ Accuracy

$$\text{Accuracy} = \frac{VP+FN}{VP+FP+FN+VN}$$

Maintenant que nous avons caractérisé notre problème (estimation des taux de bonne et mauvaise classification) via la matrice de confusion, nous allons évaluer les motifs LBPs dans notre application ($LBP_{(4,1)}$, $LBP_{(8,2)}$, $LBP_{(8,2)^{u2}}$, $SIFT+LBP_{(4,1+8,2)^{u2}}$) pour chaque descripteurs et métriques dans notre application on à utilisé 4 type de métrique (X^2 , Intersection d'histogramme, Corrélation, Log-likelihood) le rappel et la précision puis représenter les résultats à l'aide de la courbe Précision / Rappel après les testes on a remarqué que :

- ✓ **La métrique d'intersection** d'histogramme est une métrique non optimale par rapport au autre métrique parce qu'elle donne une mauvaise détection et classification et cela avec tout les type de motifs LBP que on a utilisé la figure suivante représente le résultat obtenus par cette métrique

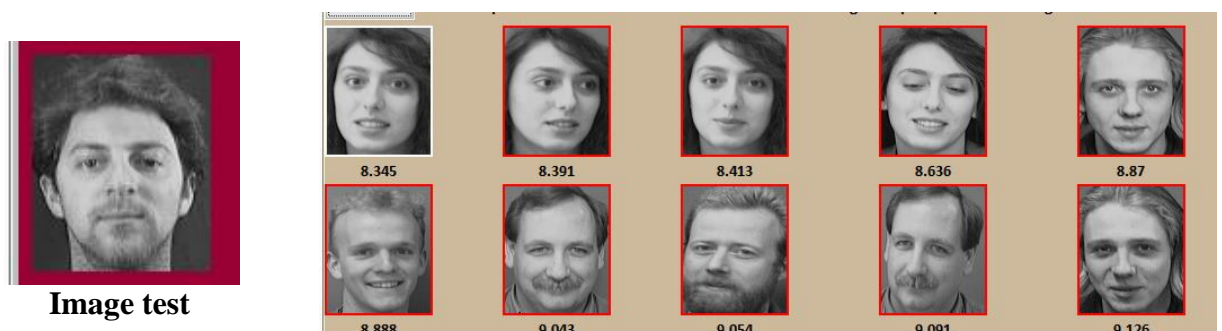


Figure 4.10 Résultats obtenus par la métrique d'intersection avec LBP 4.1

- ✓ **La métrique Log-likelihood** après l'utilisation de cette métrique, nous avons rencontré que malgré elle donne une classification assez bonne mais nous avons rencontré des difficultés dans le calcul du seuil puisque les valeurs obtenus sont des valeurs négatives la figure suivante représente le résultat obtenus par cette métrique

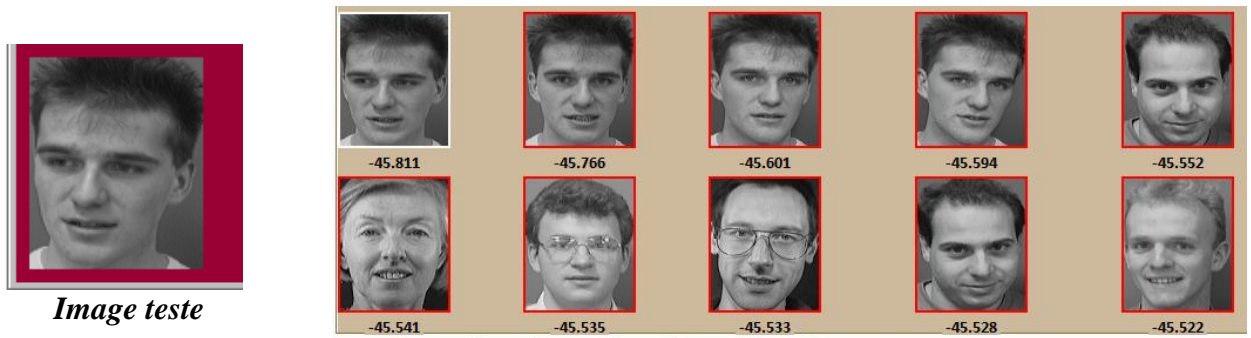


Figure 4.11 Résultats obtenus par la métrique Log-likelihood avec LBP _{8,2}

- ✓ **La métrique Corrélation** est une métrique non optimale parce qu'elle donne une mauvaise détection et classification la figure suivante représente le résultat obtenus par cette métrique



Figure 4.12 Résultats obtenus par la métrique Corrélation avec LBP _{8,2}^{U2}

- ✓ **La métrique Chi square statistic**, c'est la métrique la plus simple et la plus optimale car elle donne de bonne résultats de détection et de classification, c'est pour cela que nous avons choisi cette métrique pour la suite des expérimentations selon les différents motif LBP

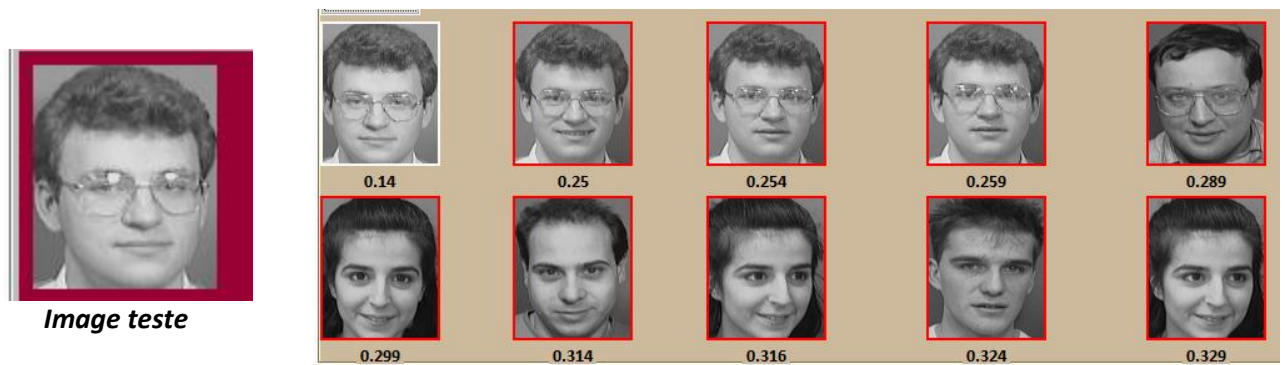


Figure 4.13 Résultats obtenus par la métrique Chi square statistic avec LBP_{4,1}

		<i>Chi carré statistique</i>							
		<i>Seuil</i>	<i>VP</i>	<i>FP</i>	<i>FN</i>	<i>VN</i>	<i>Précision</i>	<i>Rappel</i>	<i>Accuracy</i>
L.B.P.4,1		0,2	1	0	3	76	1	0.33	0.96
		0,3	6	2	0	72	0.75	1	0.97
		0,4	32	28	0	20	0.53	1	0.65
		0,5	71	67	0	9	0.51	1	0.54
		0,6	77	73	0	3	0.51	1	0.52
		0,7	77	73	0	3	0.51	1	0.52
		0,8	79	75	0	1	0.51	1	0.51
		0,9	80	76	0	0	0.51	1	0.51
L.B.P.8,2		0,2							
		0,3	2	0	2	78	1	0.5	0.79
		0,4	3	0	1	77	1	0.75	0.98
		0,5	4	0	0	76	1	1	1
		0,6	15	11	0	65	0.57	1	0.87
		0,7	40	36	0	40	0.52	1	0.87
		0,8	51	47	0	29	0.52	1	0.62
		0,9	57	53	0	23	0.51	1	0.60
L.B.P.8,2 ^{U2}		0,2	3	0	1	77	1	0.75	0.98
		0,3	4	0	0	76	1	1	1
		0,4	9	5	0	71	0.64	1	0.94
		0,5	25	21	0	55	0.54	1	0.79
		0,6	52	46	0	28	0.53	1	0.63
		0,7	67	63	0	13	0.51	1	0.55
		0,8	73	69	0	7	0.51	1	0.53
		0,9	77	73	0	3	0.51	1	0.52

Tableau 4.2 Résultats obtenus de la détection avec différents LBP

- ✚ Pour le descripteur (SIFT+LBP_(4,1+8,2^{u2})) on a remarqué que le temps de traitement est très long alors qu'on n'utilise pas toute la base de visage et les résultats ne sont pas très bonne et on a trouver des difficulté a fixé un intervalle pour le seuil
- ✚ Après ces résultats nous avons remarqué que chaque motif LBP a un seuil optimal puis que les quatre motif LBP (LBP_(4,1), LBP_(8,2), LBP_(8,2^{u2})) donne de bonne résultats

Conclusion

Nous avons développé un système de détection de l'apparence du visage et reconnaissance basé sur les caractéristiques locales LBP. Ce système utilise une base d'apprentissage construit sur un ensemble d'images permettant de recouvrir la variabilité et la diversité de l'apparence des visages humaines (traitant les visages de la base ORL).

Nous avons testé notre système sur plusieurs variantes LBP. Les expérimentations nous ont permis de conclure que la combinaison des caractéristiques LBP_{4,1} et LBP_{8,2} calculées autour des points SIFT procure une mauvaise représentativité de l'apparence ainsi que les de mauvais résultats de détection/reconnaissance alors que l'utilisation des caractéristiques LBP_{4,1} et LBP_{8,2} a eux seul en parcourant l'image entière donne une meilleur détection/reconnaissance.

Conclusion Générale

Dans le cadre de ce projet de fin d'étude nous avons étudié et implémenté une méthode basée motif binaire locaux(LBP) pour la reconnaissances/détection des visages. Notre technique consiste d'abords a détecté les caractéristique SIFT (reconnue par leur invariances) puis extraire la région local 15X15 au tour de point d'intérêt SIFT afin de calculé l'histogramme LBP de cette région (celui de LBP 4,1 et LBP 8,2)

Et en fin faire la concaténation des histogrammes pour obtenir la description de l'apparence basé LBP de la région du visage.

Notre algorithme de reconnaissance ce base sur la comparaison des histogrammes LBP de l'image test et des modèles d'histogrammes LBP. Plusieurs distance de similarité en était utilisée (tel que khi-deux, likelihood, intersection d'histogramme et corrélation).

Les résultats obtenue par notre algorithme sur la base de données ORL ont montré que la description LBP 4,1 et LBP 8,2 utilisé séparément ce comporte bien durant le processus détection/reconnaissance et donne un taux de reconnaissance acceptable et de meilleurs résultats que la description de texture LBP4,1 + LBP 8,2.

Enfin, nous dirons que ce travail appartient a un large domaine de recherche et en plein engouement. D'autres travaux restent à réaliser en perspective dans le cadre de ce projet, nous citons particulièrement, étendre le modèle LBP pour le rendre invariant a l'éclairage. L'appliqué sur d'autres classes d'objets (voiture,...), et le combinais a d'autre types de classifieur.

Bibliographies

- [1] Lingrand Diane cours de traitement D'image.
- [2] Mohamed Sahbi Bahroun, Traitements de base sur les images.
- [3] Fannie Puech, <vers plus de robustesse en reconnaissance d'objets et de visages pour l'analyse d'images issues de vidéos de concert> 2012.
- [4] Bellal Salaheddine < Reconnaissance d'objets dans une image application AU TR / contrôle de qualité >, université Hadj Lakhdar Batna.
- [5] [Nadif, 1991] : M. Nadif, <Classification automatique et données manquantes >. Thèse de Doctorat, Université de Metz, 1991.
- [6] Mathieu Van Wambeke < Reconnaissance et suivi de visages et implémentation en robotique temps-réel >.
- [7] A. Hadid , G. Zhao, T. Ahonen, and M. Pietikainen < Face Analysis Using Local Binary Patterns > ,
- [8] Lowe, D.G.2004. Distinctive Image Features from Scale-Invariant Keypoints. January 5, 2004.
- [9] Boudjellal Sofiane < Détection et identification de personne par méthode biométrique >.
- [10] TOPI MÄENPÄÄ , < The local binary pattern approach To texture analysis – Extensions and applications >, University of Oulu 2003.
- [11] T. Ojala, M. Pietikainen, and T. Maenpaa, < Multiresolution gray-scale and rotation invariant texture classification with local binary patterns> Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.24, pp.971-987, 2002.
- [12] Timo Ojala, Matti Pietikäinen and Topi Mäenpää , < Multiresolution Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns>.
- [13] Laura Sinchez Lopez , <Local Binary Patterns applied to Face Detection and Recognition >, November 2010.

[14] Xiaoyang Tan and Bill Triggs , < Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions >.

[15] Fabrizio Lamberti , < Improved Local Ternary Patterns for Automatic Target Recognition in Infrared Imagery >.