

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDELHAMID BEN BADIS DE MOSTAGANEM
FACULTÉ DES SCIENCES EXACTES ET DE L'INFORMATIQUE
DÉPARTEMENT DE MATHÉMATIQUES ET INFORMATIQUE



MÉMOIRE

Master Académique

pour obtenir le diplôme de Master délivré par

Université de Mostaganem

Spécialité “Modélisation contrôle et optimisation”

présenté et soutenu publiquement par

Halima FARAH

LE 27/JUIN/2021

**EXTRACTION DE CARACTÉRISTIQUES AVEC LES ONDELETTES
DE GABOR POUR LA DÉTECTION FACIALE**

**ENCADREUR : M. AMIR ABDESSAMMAD, PROFESSEUR (UNIVERSITÉ DE MOSTAGANEM,
ALGÉRIE)**

**JURY : MME. HAMOU MAAMAR MAGHNA, MCB PRÉSIDENTE (UNIVERSITÉ DE
MOSTAGANEM, ALGÉRIE)**

**JURY : MME. ABBASSA NADIRA, MCB EXAMINATRICE (UNIVERSITÉ DE MOSTAGANEM,
ALGÉRIE)**

**LABORATOIRE DE MATHÉMATIQUES PURES ET APPLIQUÉES
FACULTÉ DES SCIENCES EXACTES ET DE L'INFORMATIQUE (FSEI)
CHEMIN DES CRÊTES (EX-INES), 27000 MOSTAGANEM, ALGÉRIE**

Table des matières

| | | |
|----------|--|-----------|
| 0.1 | Introduction | 7 |
| 1 | Traitement d'image sur Matlab | 8 |
| 1.1 | Introduction | 8 |
| 1.2 | Traitement d'image de base avec MATLAB | 8 |
| 1.2.1 | Définition d'une image numérique | 8 |
| 1.2.2 | Lire une image | 9 |
| 1.2.3 | Afficher une image | 9 |
| 1.3 | Prétraitement d'une image | 9 |
| 1.3.1 | Histogramme d'une image | 9 |
| 1.3.2 | Egalisation d'histogramme | 11 |
| 1.3.3 | La convolution | 11 |
| 1.3.4 | Transformée de Fourier | 13 |
| 1.3.5 | Théorème de Plancherel | 14 |
| 1.3.6 | La convolution de deux images | 15 |
| 1.4 | Filtrage d'une image | 16 |
| 1.4.1 | Filtre Linéaire | 16 |
| 1.4.2 | Filtres gaussiens | 17 |
| 1.4.3 | Transformée de Fourier d'une gaussienne | 18 |
| 1.5 | Conclusion | 18 |
| 2 | Filtre de Gabor | 19 |
| 2.1 | Filtre de Gabor de dimension un | 19 |
| 2.1.1 | Transformée de Fourier | 19 |
| 2.2 | Filtre de Gabor 2D | 20 |
| 2.2.1 | L'enveloppe gaussienne | 20 |
| 2.2.2 | Sinusoïde complexe | 20 |
| 2.3 | La fonction de Gabor | 22 |
| 2.3.1 | Transformée de Fourier de fonction Gabor 2D | 23 |
| 2.3.2 | Transformée de Fourier d'un filtre de Gabor 2D | 24 |
| 2.4 | Filtre Gabor 2D pour la détection de visage | 24 |
| 2.4.1 | Filtre de Gabor 2D | 24 |
| 2.4.2 | Filtres Gabor pour l'extraction des traits du visage | 25 |
| 2.4.3 | La représentation des visages | 26 |
| 2.5 | Conclusion | 27 |
| 3 | La détection faciale | 28 |
| 3.1 | Classification | 28 |
| 3.2 | Extraction de caractéristique | 28 |
| 3.3 | Procédure scanning | 31 |
| 3.4 | Résultats d'exécution | 31 |

3.5 Conclusion générale 33

Table des figures

| | |
|---|----|
| 1.1 Lire Image dans Matlab | 9 |
| 1.2 Afficher Image | 9 |
| 1.3 Histogramme assossier au tableaux (1.2) | 10 |
| 1.4 Histogramme d'une Image $n * n$ | 11 |
| 1.5 Egalisation d'histogramme d'une Image | 11 |
| 1.6 Exemple de convolution | 16 |
| 1.7 le Filtre de Gausse | 17 |
| | |
| 2.1 La partie réelle d'une sinusoide | 21 |
| 2.2 La partie imaginaire d'une sinusoide | 21 |
| 2.3 Filtre d'une sinusoide partier réelle | 22 |
| 2.4 Filtre d'une sinusoide partier Imaginaire | 22 |
| 2.5 La fonction de Gabor 2D | 23 |
| 2.6 Les parties réelles des filtre de Gabor (avec 4 écheles et 8 orientations | 26 |
| 2.7 Convolution d'une image avec deux filtre de Gabor [10] | 26 |
| | |
| 3.1 Figure represente les visages | 29 |
| 3.2 Figure represente les non-visages | 29 |
| 3.3 Exemple1, image test | 29 |
| 3.4 Exemple2, image test | 29 |
| 3.5 Le schéma de code Matlab | 31 |
| 3.6 Résultat d'exemple 1 | 31 |
| 3.7 Résultat d'exemple 2 | 32 |

Remerciment

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon études et qui m'ont aidée lors de la rédaction de ce mémoire.

Je voudrais dans un premier temps remercier, mon directeur de mémoire M.Amir.Abdassamed, professeur de mathématique à l'université de Mostaganem , pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.

je voudrais remercier les membres du jury ,Melle ,Hamou Maamar, d'avoir bien voulu présider mon travail, et Melle Abassa d'avoir fait partis de ses membres .

Mes sincères gratitudes à M.Bahri professeur de mathématique à l'université de Mostaganem, pour ses conseils .

Nous tenons également à remercier toute l'équipe pédagogique de l'université de Mostaganem .

Dans l'impossibilité de citer tous les noms, nos sincères remerciements vont à tous ceux et celles, qui de près ou de loin, ont permis par leurs conseils et leurs compétences la réalisation de ce mémoire

Dédicace

À ma chère maman pour tous ses sacrifices, son amour, sa tendresse, son soutien et ses prières tout au long de mes études,
À mes chères sœurs Djamila et Feriel pour leurs encouragements constants et leur soutien moral,
À mon cher frère Brahim pour son soutien et ses encouragements,
À la femme de mon frère ,et ma soeur Théea pour son soutien moral,
À la petite fille de notre famille Nazeeha pour son amour et son soutien avec votre sourire,
À toute ma famille pour leur soutien tout au long de mon parcours universitaire,
À mes amis H.Soumia, L.Soumia, H.Amina de toujours aidé et encouragé ,
Que ce travail soit l'accomplissement de vos soi-disant désirs, et une évocation de votre soutien indéfectible,
Merci d'être toujours là pour moi.

0.1 Introduction

La détection d'un visage consiste à détecter la localisation et la présence d'un visage dans une image numérique. La reconnaissance faciale, s'est rapidement développée au fil des ans et est maintenant un domaine de recherche très actif, avec d'importantes applications dans la sécurité, le contrôle d'accès, l'interaction homme-machine et d'autres domaines.

Les algorithmes de la détection faciale sont basés sur deux principes, le premier est l'extraction des caractéristiques, cette tâche nous permet de distinguer les visages d'un autre objets, le second est la classification supervisée, cette tâche permet la reconnaissance d'un visage à un non-visage. Les machines à vecteurs de support (SVMs) [15] est une méthode de classification supervisée. Il s'agit de la recherche d'un hyperplan unique, donnant la marge maximale de séparation entre deux classes, grâce à la résolution d'un programme quadratique avec matrice Hessienne définie positive [2] [18]

Le Filtre de Gabor est très utilisé pour l'extraction des caractéristiques d'une image. Les représentations de fréquence et d'orientation du Filtre de Gabor, sont similaires à celles du système visuel humain, et elles se sont avérées particulièrement appropriées, pour la représentation des visages. Dans le domaine spatial, une fonction de Gabor 2D est une fonction de noyau gaussienne modulée par une onde plane sinusoïde. Vue le grand spectre d'applications des filtre de Gabor en traitement d'images numériques, nous consacrons dans ce modeste travail une introduction au filtre de Gabor et nous présentons une application dans le domaine de détection facial [17][19][1].

Le mémoire est organisé comme suit, le premier chapitre est une petite introduction au traitement d'image numérique, nous avons utilisé le logiciel Matlab [7] comme environnement de calcul, essentiellement nous avons abordé le problème de pré-traitement ainsi que les filtres linéaires. Au second chapitre nous présentons le Filtre de Gabor et les techniques utilisées pour extraire les caractéristiques discriminantes à partir d'une image de visage. Le troisième chapitre, se dédie au simulations numériques, nous avons d'abord étudié est compris le code de O. Sakhi [13] qui utilise justement le filtre de Gabor pour la détection de visage, quelques tests numériques sont présentés en utilisant ce code. Finalement nous concluons le manuscrit au dernier chapitre.

Chapitre 1

Traitement d'image sur Matlab

1.1 Introduction

Le traitement d'image est une méthode pour effectuer certaines opérations sur une image, afin d'obtenir une image améliorée ou d'en extraire des informations utiles.

Il s'agit d'un type de traitement de signal dans lequel l'entrée est une image et la sortie peuvent être une image ou des caractéristiques associées à cette image.

De nos jours, le traitement d'image fait partie des technologies à croissance rapide. Il constitue également un domaine de recherche central dans les disciplines de l'ingénierie et de l'informatique.[17]

Les images peuvent être en couleur, en niveaux de gris ou en noir et blanc. Une image en niveaux de gris (d'intensité) peut être considérée simplement comme un tableau (ou matrice) 2D, où chaque entrée contient la valeur d'intensité du pixel correspondant au niveaux de gris typiques pour l'intensité, les images peuvent avoir des valeurs entières de [0, 255] où 0 est noire et 255 et blanc .[7]

Il existe de nombreux types de formats de fichiers image utilisés pour stocker des images[7]. Nous on vas utiliser le format PNG .

1.2 Traitement d'image de base avec MATLAB

MATLAB est un logiciel de codage très simple. Toutes les variables de données dans MATLAB sont considérées comme une matrice et des opérations matricielles sont utilisées pour les analyser. MATLAB dispose des différentes boîtes à outils en fonction des domaines d'application. Dans cette section on va présenter est expliquer l'utilisation de ses fonctions de base pour l'image numérique, qui se trouve dans la boite à outil " Numerical Image procesing ".[7]

1.2.1 Définition d'une image numérique

Une image numérique est une image à des composante également appelées pixels, chacun avec des quantités finies et discrètes de représentation numérique pour son intensité ou son niveau de gris qui est une sortie de ses fonctions bidimensionnelles alimentées en entrée par ses coordonnées spatiales notées par x , y sur l'axe des x et l'axe des y , elle peut être de type vectoriel ou matriciel .[3]

1.2.2 Lire une image

La commande pour lire les images dans Matlab est : "imread". Si nous exécutons cette fonction avec des données existantes, l'image est convertie en une matrice bidimensionnelle avec des lignes et des colonnes comprenant une valeur de gris(d'intensité) dans chaque cellule [7]

```
>>imread(Im)
```

```
>> whos img
Name      Size      Bytes  Class  Attributes
img      136x179x3    73032  uint8
```

FIGURE 1.1 – Lire Image dans Matlab

1.2.3 Afficher une image

Il existe deux commandes de base pour afficher les images : "imshow "et "imagesc". En général, "imshow" est la commande préférée, car elle rend l'image plus précise, en particulier en termes de taille et pour les images couleur. [7]

```
>>imshow(Im)
```



FIGURE 1.2 – Afficher Image

1.3 Prétraitement d'une image

Les prétraitements utilisent des opérations sur les images telles que l'égalisation d'histogramme, filtrages (passe-bas, passe-haut) pour corriger la luminosité, réduire les bruits et rehausser les contrastes [3] .

1.3.1 Histogramme d'une image

L'histogramme d'une image est une fonction discrète représente le nombre de pixels en fonction du niveau de gris de l'images [7].

En pratique l'histogramme d'une image à valeurs entières est calculé de la manière suivante :

Pour chaque niveau de gris $\nu \in [0, \dots, 255]$, on compte le nombre de pixels x .

Par exemple, soit l'image de 5 pixels par 5 pixels avec des valeurs comprises entre 0 et 5 :

| | | | | |
|---|---|---|---|---|
| 0 | 2 | 3 | 1 | 5 |
| 0 | 2 | 3 | 0 | 5 |
| 0 | 1 | 4 | 4 | 5 |
| 1 | 2 | 4 | 5 | 4 |
| 1 | 3 | 3 | 5 | 5 |

TABLEAU 1.1 – Représentation matricielle d'une image 5*5

Son histogramme est une fonction qui, à chaque valeur de niveau de gris compris entre 0 et 5, associe le nombre de pixels ayant cette valeur.

Pour une image de dimension $n * n$ (voir Figure (1.4))

| | | | | | | |
|---------------------------|---|---|---|---|---|---|
| valeur de niveaux de gris | 0 | 1 | 2 | 3 | 4 | 5 |
| Nombre de pixel | 5 | 4 | 3 | 4 | 2 | 5 |

TABLEAU 1.2 – Tableau représente l'histogramme au tableaux(1.1)

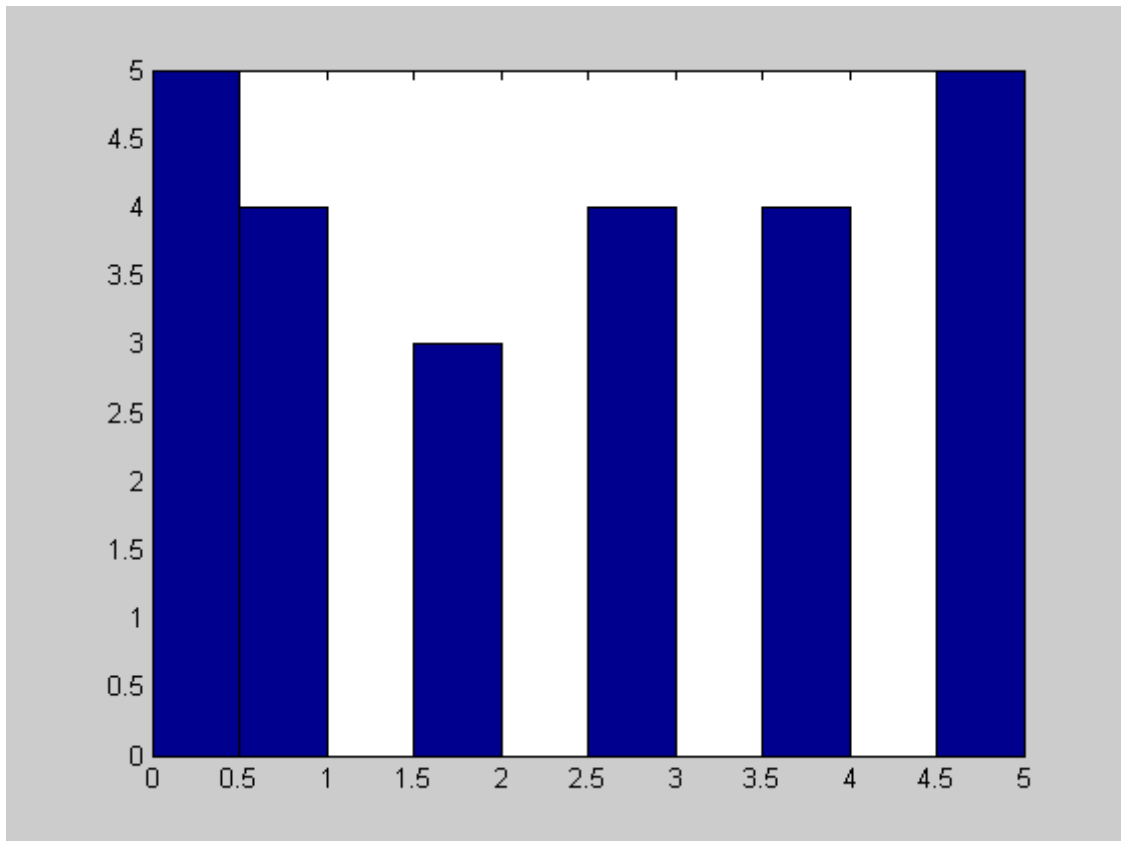


FIGURE 1.3 – Histogramme associée au tableau (1.2)

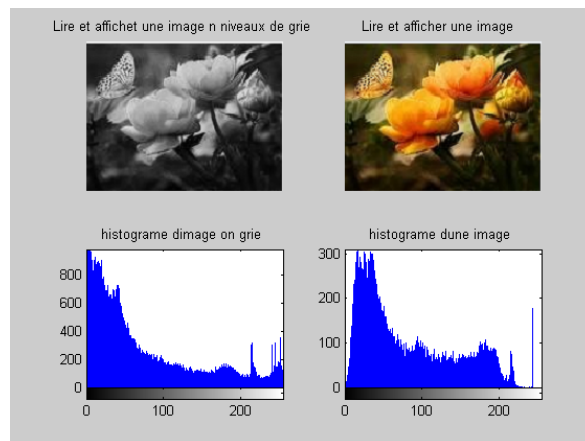


FIGURE 1.4 – Histogramme d'une Image $n * n$

1.3.2 Egalisation d'histogramme

L'égalisateur d'histogramme est une méthode pour améliorer l'image. De sorte que tout les pixels ont la même valeur de niveaux de gris (même fréquence).[7].

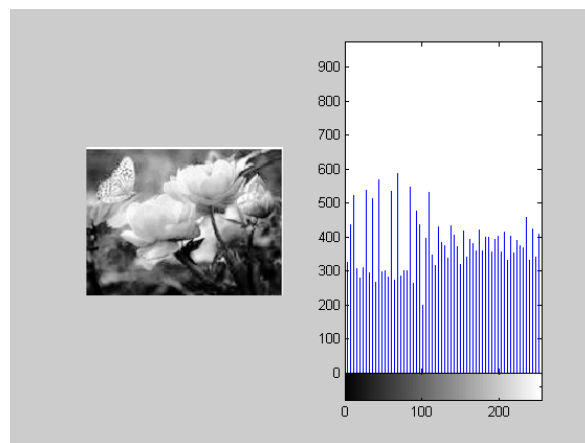


FIGURE 1.5 – Egalisation d'histogramme d'une Image

1.3.3 La convolution

La convolution [6]est liéé aux traitement de signale. La convolution entre deux signal $f(t)$ et $g(t)$ est un opérateur notée $*$ donnée par l'expression suivante :

$$f(t) * g(t) = \int_{-\alpha}^{+\alpha} f(\tau)g(t - \tau)d\tau. \quad (1.1)$$

Exemple :

Soit la fonction de port définie par :

$$f(t) = \begin{cases} 1 & \text{si } t \in [-1, 1] \\ 0 & \text{sinon} \end{cases} \quad (1.2)$$

on vas calculer le produit de convolution de f par lui même c'est à dire $f * f$

$$\begin{aligned} (f * f)(t) &= \int_{-\infty}^{+\infty} f(t) f(t - \tau) d\tau \\ &= \int_{-\infty}^{-1} 0 f(t - \tau) d\tau \int_{-1}^1 f(t - \tau) d\tau \int_1^{+\infty} f(t - \tau) d\tau \\ &= \int_{-1}^1 f(t - \tau) d\tau \end{aligned} \quad (1.3)$$

d'après (1.1) on a :

$$f(t - \tau) = \begin{cases} 1 & \text{si } -1 \leq t - \tau \leq 1 \\ 0 & \text{sinon} \end{cases} \quad (1.4)$$

on a $-1 \leq t - \tau \leq 1 \Rightarrow t - 1 \leq \tau \leq t + 1$.

Pour calculer l'intégrale (1.3) on distingue 4 cas :

- Premier cas :
si $t + 1 < -1 \Rightarrow t < -2$, et donc (1.3) devient :

$$(f * f)(t) = \int_{-1}^1 0 dt = 0.$$

- Deuxième cas :
si $t - 1 > 1 \Rightarrow t > 2$, donc on a

$$(f * f)(t) = \int_{-1}^1 f(t - \tau) d\tau = \int_{-1}^1 0 dt = 0.$$

- Troisième cas :
si

$$\begin{cases} -1 < t + 1 < 1 \\ \text{et} \\ t - 1 < -1 \end{cases}$$

ce qui donne, $-2 \leq t \leq 0$
donc

$$\begin{aligned} (f * f)(t) &= \int_{-1}^1 f(t - \tau) d\tau = \int_{-1}^{t+1} dt \\ (f * f)(t) &= t + 1 - (-1) = t + 2, \end{aligned} \quad (1.5)$$

- Quatrième cas :
si

$$\begin{cases} -1 < t-1 < 1 \\ et \\ t+1 > 1, \end{cases}$$

$0 \leq t \leq 2$
donc,

$$(f * f)(t) = \int_{-1}^1 f(t-\tau) d\tau = \int_{t-1}^1 dt = 1 - (t-1) = 2 - t, \quad (1.6)$$

d'où le produit de convolution de $(f * f)(t)$ est :

$$(f * f)(t) = \begin{cases} 0 & si, t < -2 \\ t+2 & si, -2 \leq t \leq 0 \\ 2-t & si, 0 \leq t \leq 2 \\ 0 & si, t \geq 2. \end{cases} \quad (1.7)$$

En pratique, il est toujours difficile de calculer le produit de convolution de deux fonctions. Le résultat suivant nous permet de faire se calcule avec le produit usuel, en utilisant les transformées de Fourier [16].

1.3.4 Transformée de Fourier

Soit une fonction f définie sur \mathbf{R} , continue par morceaux sur tout segment, à valeur réelles ou complexes. On appelle la transformée de Fourier [16][6] de f , la fonction notée TF, définie sur \mathbf{R} par :

$$TF(f(t)) = \int_{-\infty}^{+\infty} f(t) \exp -2\pi i \lambda t dt. \quad (1.8)$$

Exemple

Soit la fonction définie par :

$$\Psi(t) = \begin{cases} 1 & si, t \in [-1/2, 1/2] \\ 0 & sinon, \end{cases} \quad (1.9)$$

et donc

$$TF(\Psi(\lambda)) = \int_{-\infty}^{+\infty} \Psi(t) \exp(-2\pi i \lambda t) dt$$

$$TF(\Psi(\lambda)) = \int_{-\infty}^{-\frac{1}{2}} \Psi(t) \exp(-2\pi i \lambda t) dt + \int_{-\frac{1}{2}}^{+\frac{1}{2}} \Psi(t) \exp(-2\pi i \lambda t) dt + \int_{\frac{1}{2}}^{+\infty} \Psi(t) \exp(-2\pi i \lambda t) dt,$$

or, la fonction Ψ est nulle sur $[-\infty, -\frac{1}{2}[$ et $]\frac{1}{2}, +\infty [$ par définition, et donc

$$TF(\Psi(\lambda)) = \int_{-\frac{1}{2}}^{+\frac{1}{2}} \Psi(t) \exp -2\pi i \lambda t dt,$$

comme $\Psi(t) = 1$ sur $[-\frac{1}{2}, \frac{1}{2}]$,

$$\text{TF}(\Psi(\lambda)) = \int_{-\frac{1}{2}}^{+\frac{1}{2}} \exp(-2\pi i \lambda t) dt, \quad (1.10)$$

on a si $\lambda \neq 0$, l'équation (1.8) devient,

$$\begin{aligned} \text{TF}(\Psi(\lambda)) &= \left. \frac{\exp(-2\pi i \lambda t)}{-2\pi i \lambda} \right]_{-\frac{1}{2}}^{+\frac{1}{2}} \\ &= \frac{-1}{2\pi i \lambda} [\exp(-\pi i t) - \exp(\pi i t)] \\ &= \frac{-1}{2\pi i \lambda} [(\cos \pi \lambda - i \sin \pi \lambda) - (\cos \pi \lambda + i \sin \pi \lambda)] \\ &= \frac{-1}{2\pi i \lambda} (-2i \sin \pi \lambda) \\ \text{TF}(\Psi(\lambda)) &= \frac{1}{\pi \lambda} \sin \pi \lambda. \end{aligned} \quad (1.11)$$

Si $\lambda = 0$

$$\begin{aligned} \text{TF}(\Psi(0)) &= \int_{-\frac{1}{2}}^{+\frac{1}{2}} \Psi(0) \exp(-2\pi i 0 t) dt \\ &= \int_{-\frac{1}{2}}^{+\frac{1}{2}} dt \\ \text{TF}(\Psi(0)) &= t \Big|_{-\frac{1}{2}}^{+\frac{1}{2}} = 1. \end{aligned} \quad (1.12)$$

D'où la transformée de Fourier de Ψ est :

$$\text{TF}(\Psi(\lambda)) = \begin{cases} \frac{\sin \pi \lambda}{\pi \lambda} & \text{si } \lambda \neq 0 \\ 1 & \text{si } \lambda = 0 \end{cases} \quad (1.13)$$

Parmi les propriétés de la transformée de Fourier l'inversibilité .

1.3.5 Théorème de Plancherel

La transformée de Fourier du produit de convolution de deux signaux $s(t)$ et $r(t)$ est égale au produit des transformées de Fourier de ces deux signaux[6] :

$$\text{T.F.}(s(t) * r(t)) = \text{T.F.}(s(t))\text{T.F.}(r(t)). \quad (1.14)$$

Il est possible de calculer le produit de convolution de deux signaux, en calculant la transformée de Fourier inverse .

$$\text{TF}^{-1}[\text{TF}(s(t))\text{TF}(r(t))] = s(t) * r(t). \quad (1.15)$$

Preuve

Soit deux signaux $s(t)$ et $r(t)$, son transformé de Fourier $TF(s(t))$ et $TF(r(t))$ respectivement. La transformée de fourier du produit de convolution de s par r est calculé comme suit [6] :

$$\begin{aligned} TF(s(t) * r(t)) &= \int_t \left[\int_{\tau} (s(t) * r(t - \tau)) \exp(\pi i f t) d\tau \right] dt. \\ &= \int_t \left(\int_{\tau} (s(\tau) r(t - \tau) \exp(-\pi i f t) dt d\tau) \right). \end{aligned}$$

Remarque

Le produit de convolution dépend de t , mais t et un paramètre, donc l'intégration est fait par à port à τ .

On pose $\mu = t - \tau \Rightarrow d\mu = dt$ (car τ est résolue par le produit de convolution)

$$\begin{aligned} &= \int_t (s(\tau) \left[\int_{\tau} r(\mu) \exp(-2\pi i f \tau) \exp(-2\pi i f \mu) d\tau \right] d\mu) \\ &= \int_t \int_{\tau} (s(\tau) \exp(-2\pi i f \tau) d\tau \int r(\mu) \exp(-2\pi i f \mu) d\mu) \\ &= \int_{\tau} (s(\tau) \exp(-2\pi i f \tau) d\tau \int r(\mu) \exp(-2\pi i f \mu) d\mu). \end{aligned} \tag{1.16}$$

Remarque

μ et τ sont des variables indépendentes, l'intégrale (1.12) devient un produit de deux intégrales simple .

D'où

$$TF(s(t) * r(t)) = TF(s(t)).TF(r(t)). \tag{1.17}$$

1.3.6 La convolution de deux images

La convolution est courante en traitement d'images[11]. Elle consiste en une opération de multiplication de deux matrices de tailles différentes (généralement une petite et une grande), de même dimension, produisant une nouvelle matrice de même dimension. La convolution est donc le traitement d'une matrice (p.ex. une image) par une autre petite Matrice appelée noyau (kernel).

En mathématique, le produit de convolution entre deux images f et g en niveaux de gris, notée $f * g$ est définie par [17] :

$$(x, y) \in Z^2; (f * g)(x, y) = \sum_i \sum_j f(i, j), g(x - i, y - j), \tag{1.18}$$

avec

$g(x-i,y-j)$: joue le rôle de fenêtre glissante .

L'opération de pondération est la multiplication de f par la translaté de g .

La somation et l'opération de moyennage.

Par exemple :

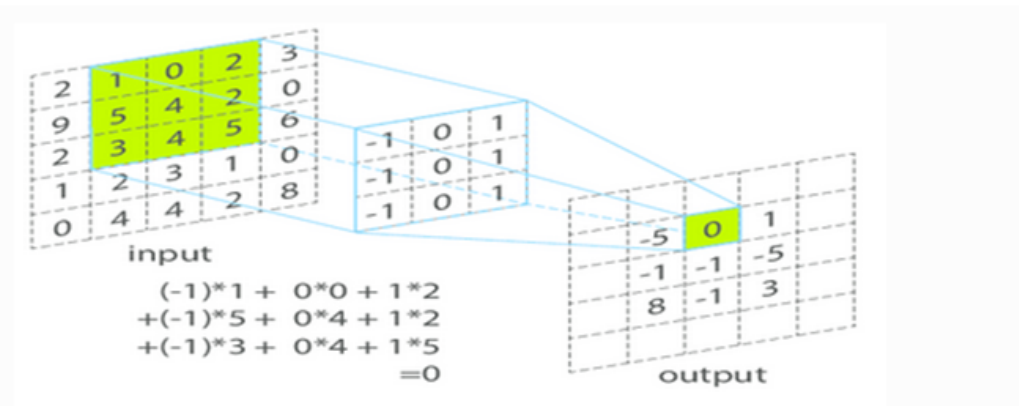


FIGURE 1.6 – Exemple de convolution

1.4 Filtrage d'une image

Le filtrage est une technique permettant de modifier ou d'améliorer une image[17]. En traitement de signal classique, on effectue souvent le filtrage d'un signal bruité dans le domaine fréquentiel. Selon cette procédure, on multiplie la transformée de Fourier TFS(ω) du signal temporel $s(t)$ par le Filtre F(ω) :

$$S_{\text{filtre}}(\omega) = F(\omega) \times \text{TFS}(\omega). \quad (1.19)$$

En vision artificielle, l'utilisation de la transformée de Fourier est peu répandue car il est plus naturel de travailler directement dans l'image. Or, d'après le théorème de Plancherel, on sait qu'une multiplication dans le domaine des fréquences est équivalente à une convolution dans le domaine spatial.

Les opérations de traitement d'image misent en œuvre avec filtration incluent le lissage, la netteté et l'amélioration des bords, réduire le bruit.[12]

Pour filtrer une image; il existe des filtres linéaires et non linéaires, dans ce travail, on s'intéresse uniquement aux filtres linéaires.

1.4.1 Filtre Linéaire

Un filtre F est une application qui consiste à transformer mathématiquement un signal μ de dimension n_μ en un signal y de taille n_y , à chaque entrée $\mu(k)$ correspond une sortie $y(k)$.

Le Filtrage Linéaire se fera par la convolution d'un filtre exprimé dans le domaine spatial avec l'image. L'équation de convolution pour chaque pixel d'illuminance $I(i, j)$ est donnée par :

$$I_{\text{filtre}}(i, j) = A(h, k) \times I(i, j) = \left(\frac{1}{m^2}\right) \sum_{h=-\frac{m}{2}}^{\frac{m}{2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} A(h, k) \times I(i-h, j-k). \quad (1.20)$$

Pour chaque pixel (i, j) , les valeurs d'illuminance $I(i, j)$ des pixels couverts par le masque $A(h, k)$ sont multipliées par les valeurs du masque et additionnées pour produire la moyenne en multipliant par la taille du masque $\frac{1}{m^2}$. Parmi les filtres Linéaires on a :

1.4.2 Filtrés gaussiens

Les filtres Gaussiennes sont une classe de filtres passe-bas, possède un noyau de convolution de forme Gaussienne. Le Filtre Gaussienne est donnée par la formule suivante [12] :

$$g(x, y) = \left(\frac{1}{2\pi\sigma^2}\right) \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right) \quad (1.21)$$

où :

x : la distance entre l'origine et l'axe xx' .

y : la distance entre l'origine et l'axe yy'

σ : la déviation de la distribution Gaussiennee .

En pratique si par exemple en prend $\sigma = 0.8$, on a le filtre 3×3 suivant :

| | | |
|----------|--------|---------|
| G(-1,-1) | G(0,1) | G(1,-1) |
| G(-1,0) | G(0,0) | G(1,0) |
| G(-1,1) | G(0,1) | G(1,1) |

| | | |
|------|-----|------|
| 1/16 | 1/8 | 1/16 |
| 1/8 | 1/4 | 1/8 |
| 1/16 | 1/8 | 1/16 |

TABLEAU 1.3 – Tableau représente le filtre de Gauss 3×3 pour $\sigma = 0.8$

Le filtre gaussien généralement avec $\sigma < 1$ est utilisé pour réduire le bruit, et si $\sigma > 1$ c'est dans le but de fabriquer une image qu'on va utiliser pour faire un " masque flou " personnalisé. Il faut noter que plus σ est grand, plus le flou appliqué à l'image sera marqué. La Figure(1.8) représente le Filtre de Gausse.

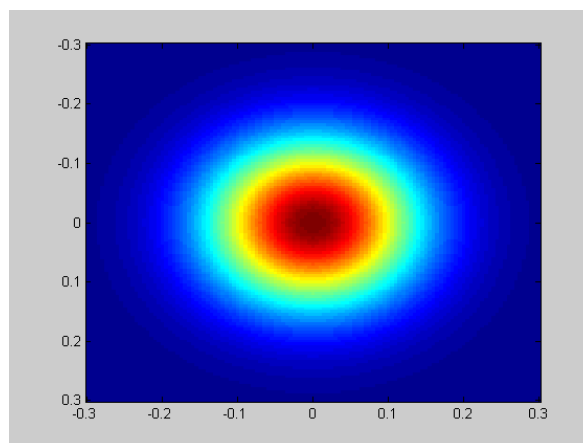


FIGURE 1.7 – le Filtre de Gausse

1.4.3 Transformée de Fourier d'une gaussienne

La transformé de Fourier d'une Gaussienne $g(x, y)$ [14] :

$$\begin{aligned}
 g(x, y) &= \exp(-\pi(x^2 + y^2)), \\
 \text{TF}(g(x, y)) &= \int_{-\infty}^{+\infty} \left[\int_{-\infty}^{+\infty} \exp(-\pi(x^2 + y^2)) \exp(-2\pi i \mu x) \exp(-2\pi i \nu y) dx \right] dy \\
 &= \int_{-\infty}^{+\infty} \exp(-\pi(x)^2) \exp(-2\pi i \mu x) dx \int_{-\infty}^{+\infty} \exp(-\pi y^2) \exp(-2\pi i \nu y) dy \\
 &= \exp(-\pi \mu^2) \exp(-\pi \nu^2) \\
 &= \exp(-\pi(\mu^2 + \nu^2)) \\
 &= \exp(-\pi(\mu^2 + \nu^2)).
 \end{aligned}$$

Nous concluons que la transformée de Fourier d'une Gaussienne est une Gaussienne.

1.5 Conclusion

Les méthodes de traitement d'images convergent vers deux objectifs essentielles : l'extraction d'un maximum d'informations, et avoir un meilleur aspect de l'image. Le filtrage linéaire représente une des opérations les plus élémentaires dans le cadre du traitement des images, pour effectuer de tels traitements dans un ordinateur, les images doivent auparavant être numérisées. Nous avons, présentés dans cette partie quelques opérations sur les images, telles que, l'histogramme, l'égalisation d'histogramme, des exemples de filtre linéaire. Dans ce travail on s'intéresse à un filtre très utilisé dans le domaine de traitement d'image numérique le filtre de Gabor [10],[12],[1].

Chapitre 2

Filtre de Gabor

En 1946, Gabor a proposé un filtre 1D qui optimise la relation entre le temps et la fréquence. Daugman a ensuite développé cette idée pour construire des filtres 2D optimisant la relation entre la localisation spatiale et la fréquence spatiale.

Dans le domaine spatial, un filtre de Gabor 2D est une fonction à noyau gaussien modulé par une onde sinusoïde.

2.1 Filtre de Gabor de dimension un

Les filtres de Gabor de dimensions 'un' (1D) peuvent être utilisés comme filtres passe-bande pour les signaux unidimensionnels (par exemple, la parole). Un filtre de Gabor (1D) complexe est défini comme le produit d'un noyau gaussien modulé par une sinusoïde complexe [11] :

$$G_1(t) = K \exp(i\theta) \omega(at) s(t). \quad (2.1)$$

Où

$$\omega(t) = \exp(-\pi t^2), \quad (2.2)$$

et

$$s(t) = \exp(i(2\pi f_0 t)). \quad (2.3)$$

Avec

$$\exp(i\theta) s(t) \exp(i(2\pi f_0 t + \theta)) = (\cos(2\pi f_0 t + \theta) + i \sin(2\pi f_0 t + \theta)) \quad (2.4)$$

K , θ et f_0 sont les paramètres de filtre.

2.1.1 Transformée de Fourier

La transformée de Fourier d'un filtre de Gabor 1D, notée TFG1 est donnée par [11] :

$$\text{TF}(G_1(f)) = K \exp(i\theta) \int_{-\infty}^{+\infty} \exp(-i2\pi f t) \omega(at) s(t) dt \quad (2.5)$$

$$= k \exp(i\theta) \int_{-\infty}^{+\infty} \exp(-i2\pi(f - f_0)t) \omega(at) dt = \frac{K}{a} \exp(i\theta) \omega\left(\frac{f - f_0}{a}\right) \quad (2.6)$$

2.2 Filtre de Gabor 2D

Une fonction de Gabor (2D) complex dans le domaine spatial est donnée comme suit [11] :

$$G_2(x, y) = s(x, y)\omega_r(x, y). \quad (2.7)$$

Où

$s(x, y)$ est la sinusoïde complex, et $\omega_r(x, y)$ une fonction gaussienne 2D connue sous le nom enveloppe.

2.2.1 L'enveloppe gaussienne

L'enveloppe gaussienne est définie par :

$$\omega_r(x, y) = K \exp(-\pi(a^2((x - x_0)_r)^2 + b^2((y - y_0)_r)^2)). \quad (2.8)$$

Où (x_0, y_0) les sommets de la fonction, a et b les échelles de la gaussienne. L'indice r représente une opération de rotation telle que :

$$\begin{aligned} (x - x_0)_r &= (x - x_0) \cos \theta + (y - y_0) \sin \theta \\ \text{et } (y - y_0)_r &= -(x - x_0) \sin \theta + (y - y_0) \cos \theta. \end{aligned}$$

Remarque

Dans la formule (1.8) $(x, y) = (0, 0)$, $K = \frac{1}{2\pi\sigma^2}$, $a = b = 1$.

2.2.2 Sinusoïde complex

Une sinusoïde complex et définie par [11] :

$$s(x, y) = \exp(i(2\pi(\mu_0 x + \nu_0 y) + P)), \quad (2.9)$$

Où (μ_0, ν_0) la fréquence spatiale et P la phase de la sinusoïde. Nous pouvons considérer la sinusoïde comme deux fonctions réelles distinctes, distribuée dans la partie réelle et imaginaire d'une fonction complex (voir Figure (2.1) et (2.2)).

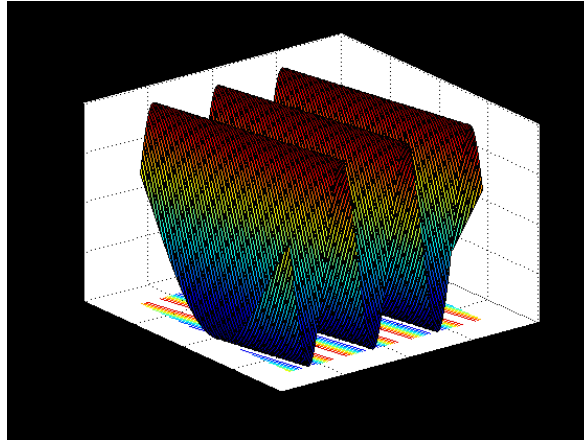


FIGURE 2.1 – La partie réelle d'une sinusoïde

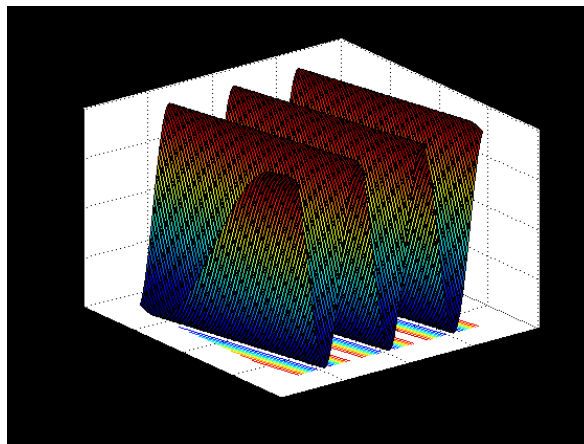


FIGURE 2.2 – La partie imaginaire d'une sinusoïde

Les équations de la partie réelle et imaginaire d'une sinusoïde sont données par les deux formules suivantes :

$$\text{Re}(s(x, y)) = \cos(2\pi(\mu_0 x + \nu_0 y) + P), \quad (2.10)$$

$$\text{Im}(s(x, y)) = \sin(2\pi(\mu_0 x + \nu_0 y) + P), \quad (2.11)$$

avec μ_0 et ν_0 définies la fréquence spatiale de la sinusoïde en coordonnées cartésiennes respectivement. On peut exprimer la fréquence spatiale en coordonnées polaires sous forme de magnitude F_0 et de direction ω_0 .

$$F_0 = \sqrt{\mu_0^2 + \nu_0^2}, \omega_0 = \tan^{-1}\left(\frac{\mu_0}{\nu_0}\right). \quad (2.12)$$

i.e

$$\mu_0 = F_0 \cos \omega_0, \nu_0 = F_0 \sin \omega_0. \quad (2.13)$$

En utilisant cette représentation, la sinusoïde complexe devient :

$$s(x, y) = \exp(i(2\pi F_0(x \cos \omega_0 + y \sin \omega_0) + P)) \quad (2.14)$$

La figure (2.3) représente les filtres d'une sinusoïde " partie réelle et partie imaginaire ".

Remarque

Dans les Figures (2.3), (2.4), $P = 0$.

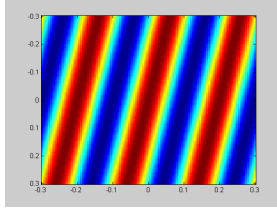


FIGURE 2.3 – Filtre d’une sinusoïde partie réelle

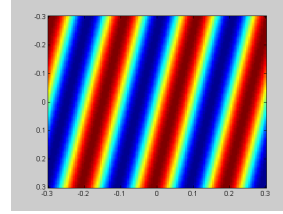


FIGURE 2.4 – Filtre d’une sinusoïde partie Imaginaire

2.3 La fonction de Gabor

La fonction complexe de Gabor est définie par neuf paramètres :

- 1) K , le facteur d’échelle de l’enveloppe gaussienne .
- 2) (a, b) , les paramètres de l’enveloppe gaussienne par rapport à les deux axes .
- 3) θ , l’angle de rotation.
- 4) (x_0, y_0) , l’emplacement du pic de l’enveloppe gaussienne.
- 5) μ_0, ν_0 , les fréquences spatiales de la sinusoïde par rapport aux les coordonnées cartésiennes.
- 6) P , la phase relative de sinusoïde .

Dans le domaine spatial la fonction de Gabor 2D [14] est :

$$g(x, y) = K \exp(-\pi(a^2(x - x_0)_r)^2 + b^2((y - y_0)_r)^2)) \exp(i(2\pi(\mu_0 x + \nu_0 y) + P)). \quad (2.15)$$

Or, avec les coordonnées polaires :

$$g(x, y) = K \exp(-\pi(a^2(x - x_0)_r)^2 + b^2((y - y_0)_r)^2)) \exp(i(2\pi). \quad (2.16)$$

$$F_0(x \cos \omega_0 + y \sin \omega_0) + P)). \quad (2.17)$$

La figure (11) représente la Fonction de Gabor 2D.

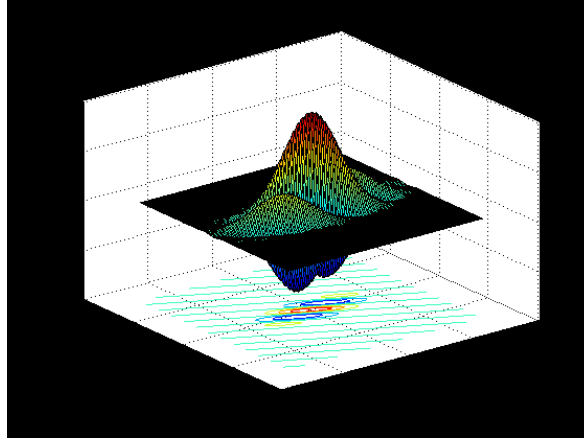


FIGURE 2.5 – La fonction de Gabor 2D

2.3.1 Transformée de Fourier de fonction Gabor 2D

Soit l'enveloppe gaussienne $\omega(x)$ et la sinusoïde $s(x)$:

$$\omega(x) = \exp(-\pi x^T x), \quad (2.18)$$

$$ets(x) = \exp(i2\pi\mu_0^T x) \quad (2.19)$$

Soit la fonction de Gabor définie par :

$$G(x) = K \exp(i\pi) \omega(A(x - x_0)) s(x) \quad (2.20)$$

Ou :

$K, A, P, \mu_0,$ et x_0 sont les paramètres de la fonction dans l'équation (2.17), la transformée de Fourier de cette fonction est [?] :

$$Tf(G(\mu)) = \int_{-\infty}^{+\infty} g(x) \exp(-2\pi i \mu^T x) dx \quad (2.21)$$

$$= K \exp(iP) \int_{-\infty}^{+\infty} \omega(A(x_0)) \exp(-2\pi i (\mu - \mu_0)^T x) dx, \quad (2.22)$$

soit

$$x = A(x - x_0),$$

on a

$$x = A^{-1}x + x_0 \Rightarrow dx = Adx,$$

et donc on a :

$$TF(G(\mu)) = \frac{K}{\|A\|} \int_{-\infty}^{+\infty} \exp(iP) \omega(x) \exp(-2i(\mu - \mu_0)^T (A^{-1}(\mu - \mu_0^T (A^{-1}x - x_0))) dx \quad (2.23)$$

$$= \frac{K}{\|A\|} \exp(iP) \exp(-\mu_0^T x_0) \int_{-\infty}^{+\infty} \omega(x) \exp(-2\pi i (A^{-T}(\mu - \mu_0)^T x)) dx. \quad (2.24)$$

Donc,

$$\text{TF}(G(\mu)) = \frac{K}{\|A\|} \exp(iP) \exp(-2\pi(A^{-T}(\mu - \mu_0))^T x_0) \omega(A^{-T}(\mu - \mu_0)), \quad (2.25)$$

où, nous avons utilisé le fait que :

$$\text{TF}(\omega(\cdot)) = \omega(\cdot). \quad (2.26)$$

2.3.2 Transformée de Fourier d'un filtre de Gabor 2D

La transformée de Fourier pour un filtre Gabor 2D est donnée par la fomulle suivante [11][14] :

$$\text{TF}(G2(\mu, \nu)) = \frac{K}{ab} \exp(i(-2\pi(x_0(\mu - \mu_0) + y_0(\nu - \nu_0)) + P)) \exp(-\pi(\frac{((\mu - \mu_0)_r)^2}{a^2} + \frac{((\nu - \nu_0)_r)^2}{b^2})). \quad (2.27)$$

Or, dans les cordonées polaire :

La magnitude :

$$\text{TF}(G2(\mu, \nu)) = \frac{K}{ab} \exp(-\pi(\frac{((\mu - \mu_0)_r)^2}{a^2} + \frac{((\nu - \nu_0)_r)^2}{b^2})). \quad (2.28)$$

La phase

$$\text{TF}(G2(\mu, \nu)) = -2\pi(x_0(\mu - \mu_0) + y_0(\nu - \nu_0)) + P. \quad (2.29)$$

2.4 Filtre Gabor 2D pour la détection de visage

De nos jours, les fonctions de Gabor sont très utilisées pour l'extraction des caractéristiques, dans l'analyse des images (par exemple, la classification, la segmentation ou la détection de contours), plus pratiquement dans la reconnaissance faciale[4]. Les caractéristiques de Gabor extraient les informations locales qui sont ensuite combinées pour reconnaître un objet (par exemple, un visage humains) ou une région d'intérêt.

2.4.1 Filtre de Gabor 2D

Les caractéristiques des ondelettes de Gabor (filtres)[12], pour les représentations de fréquence et d'orientation, sont similaires à celles du système visuel humain. Les fonctionnalités basées sur les filtres de Gabor, directement extraites l'images d'objets en niveaux de gris, ont été largement appliquées avec succès à la segmentation de texture[1], à la reconnaissance de chiffres manuscrits[19], la reconnaissance d'empreintes digitales [5] et la reconnaissance facial[10].

Un filtre de Gabor 2D est une fonction à noyau gaussienne modulée par une onde plane sinusoïde [10] :

$$\Phi_{\pi(f,\theta,\gamma,\eta)}(x, y) = \frac{f^2}{\pi\gamma\eta} \exp(-(\alpha^2 x'^2 + \beta^2 y'^2)) \exp(2\pi f x'), \quad (2.30)$$

$$x' = x \cos \theta + y \sin \theta,$$

$$et y' = -x \sin \theta + y \cos \theta$$

f la fréquence de l'onde plane sinusoidale.

θ l'angle de rotation de la Gaussienne et l'onde plane .

α la netteté, le long du grande axe gaussienne (parallèle à l'onde) .

β la netteté, le long du petite axe (perpendiculaire à l'onde) .

$\gamma = \frac{f}{\alpha}$ et $\eta = \frac{f}{\beta}$ sont définis pour conserver le rapport entre la fréquence et la netteté constante .

Chaque Filtre se présente sous la forme d'ondes planes de fréquence f restreintes par une fonction enveloppe gaussienne de largeur relative à α et β .

$$\Phi_{\mu, \nu} = \Phi_{\pi(f_{\mu}, \theta_{\nu}, \gamma, \eta)}, \quad (2.31)$$

$$f_{\mu} = \frac{f_{max}}{\sqrt{2}^{\mu}}$$

$$\theta_{\nu} = \frac{\nu}{8} \pi$$

$$\mu = 0, \dots, U - 1; \nu = 0, \dots, V - 1.$$

Remarque

$\Phi_{\pi(f_{\mu}, \theta_{\nu}, \gamma, \eta)}$ est donnée par la formule (3.1), on remplace f par f_{μ} et θ par θ_{ν} .

2.4.2 Filtres Gabor pour l'extraction des traits du visage

Dans la section précédente, nous avons vu que les filtre de Gabor depend de plusieurs parametres . Dans le cadre de la detection faciale, on peut pas reconstruire des signaux contenant des fréquences supérieure à la moitié de fréquence d'échantillonnage .Par conséquent, la fréquence limite d'une image 2D est $\frac{1}{2}$ (cycle/pixel), alors que la fréquence inférieure et de 0 .

Nous choisissons $f_{max} = 0.25$ pour la reconnaissance faciale. Les paramètres γ et η déterminent le rapport entre la fréquence centrale et la taille de l'enveloppe gaussienne. Une fois le rapport fixé, la taille de l'enveloppe gaussienne décroît de façon monotone avec la valeur de la fréquence centrale. On suppose que $\alpha = \beta$ et $\gamma = \eta = \sqrt{2}$, nous choisissons 5 échelle et 8 orientation [10].

Pour extraire des caractéristiques utiles d'une image, par ex. face, on construisons un ensemble de filtres de Gabor avec différentes fréquences et orientations, qui former une banc de filtre de Gabor. Dans la plupart des cas, le banc de filtres Gabor est utilisé avec cinq fréquences (échelles) et huit orientations [10] (voir Figure (3.1)).

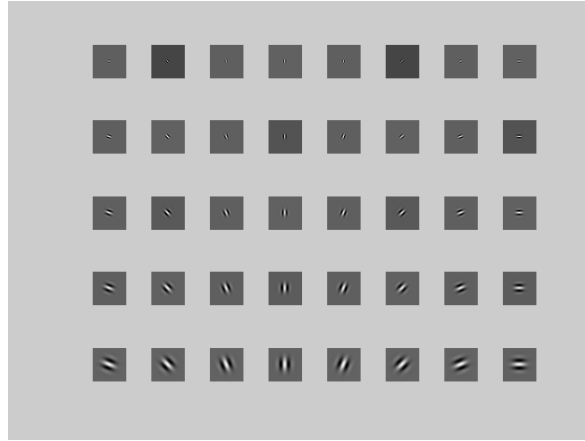


FIGURE 2.6 – Les parties réelles des filtre de Gabor (avec 4 écheles et 8 orientations)

2.4.3 La représentation des visage

La représentation de Gabor d'une image (visage) peuvent être obtenue en convoluant l'image avec la famille des filtres de Gabor [10] /

$$G_{\mu,\nu} = I(x, y) * \phi_{\mu,\nu}. \quad (2.32)$$

Où $G_{\mu,\nu}$ définie le résultat de convolution bidimensionnel correspondant au filtre de Gabor à l'échelle μ et à l'orientation ν .

La figure (3.2) montre les résultats de convolution d'une image (visage) avec deux filtres de Gabor .

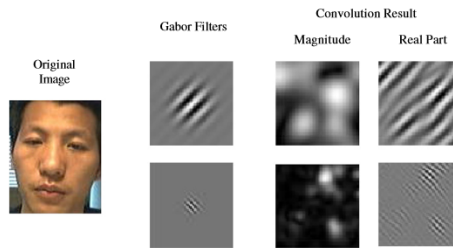


FIGURE 2.7 – Convolution d'une image avec deux filtre de Gabor [10]

L'image du visage est convoluée avec deux filtres de Gabor à des orientations et des échelles différentes, l'amplitude et la partie réelle des résultats de convolution sont affichées dans la Figure(3.2). Ainsi, chaque image (visage) peuvent être représenter par un ensemble des coefficients d'ondellette de Gabor $G_{\mu,\nu}(x^{\rightarrow}, \mu = 0, \dots, 4, \nu = 0, \dots, 7)$, l'amplitude de chacun $G_{\mu,\nu}(x^{\rightarrow})$ est sous-échantillonné par un facteur r , en normalisé à une moyenne nulle et une variance libre, et transformée en un vecteur $x_{\mu,\nu}^r$, en concaténant les lignes. Un vecteur x^r de caractéristique discriminant peuvent être dérivé pour représenter l'image I en concaténant ces vecteurs $x_{\mu,\nu}^r$:

$$x^r = ((x_{0,0}^r)^t (x_{0,1}^r)^t \dots (x_{4,7}^r)^t)^t \quad (2.33)$$

Le vecteur caractéristique x^r inclue tous les éléments de l'ensemble représentatif d'ondellettes de Gabor $G_{\mu,\nu}(x^{\rightarrow}, \mu = 0, \dots, 4, \nu = 0, \dots, 7)$. cependant, la dimension du vecteur x^r est assez élevée .

2.5 Conclusion

L'utilisation des filtres Gabor dans l'extraction des caractéristiques peuvent être justifiée par des découvertes biologiques dans les systèmes de vision, des statistiques d'images naturelles et le succès des applications existantes. Il est bien utilisé pour la reconnaissance faciale, nous permet d'extraire l'essentiel des caractéristiques globales du visage et les classifier par les SVMs .

Chapitre 3

La détection faciale

Dans les méthodes basées sur la classification, la détection de visage est effectuée en déplaçant une fenêtre de recherche sur une image d'entrée et en catégorisant l'objet avec un classificateur. Pour une bonne détection on effectue deux étapes fondamentales : L'extraction des caractéristiques (utilisant les filtres de Gabor). La classification (utilisant les SVM) .

3.1 Classification

Etant données un ensemble de données $x_{i,j}$ ou $x_{i,j} \in \mathbf{R}$, et un vecteur y dans \mathbf{R} telle que $y_j = \pm 1$ ou $y_j \in \{0, 1\}$. Nous disposons donc d'un ensemble de données étiquetées. Le problème de classifier consiste à déterminer une fonction $h : \mathbf{R}^d \rightarrow \{0, 1\}$ qui permettra de prédire la classe de chaque donnée $x_{i,j}$. Généralement, l'ensemble des données est représenté en deux ensembles, un ensemble d'apprentissage sur lequel on va engendrer le classifieur h et un ensemble de test sur lequel on va déterminer l'efficacité de ce classifieur h . Il existe plusieurs algorithmes de classification on peut citer réseaux de neurones [8], analyse de discrimination linéaire et quadratique et régression linéaire .

Dans le code élaboré par O.Sakhi [13], l'auteur a utilisé l'algorithme de classification par les machines à vecteurs de supports (SVMs). Le lecteur intéressé par cette technique peut consulter [18],[2] .

3.2 Extraction de caractéristique

Nous suivons dans ce travail, l'approche utilisée par O.Sakhi [13], pour la détection des visages dans une image. À fin de construire le classifieur il est nécessaire d'avoir un ensemble de données constitué de deux classes (visage et non visage) et des images testées (voir les figures (3.1),(3.2),(3.3)),



FIGURE 3.1 – Figure represente les visages

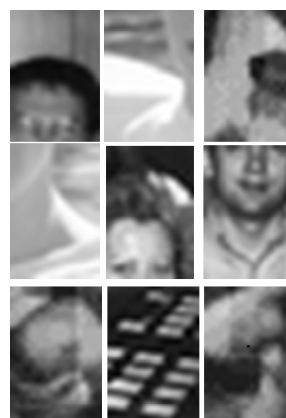


FIGURE 3.2 – Figure represente les non-visages

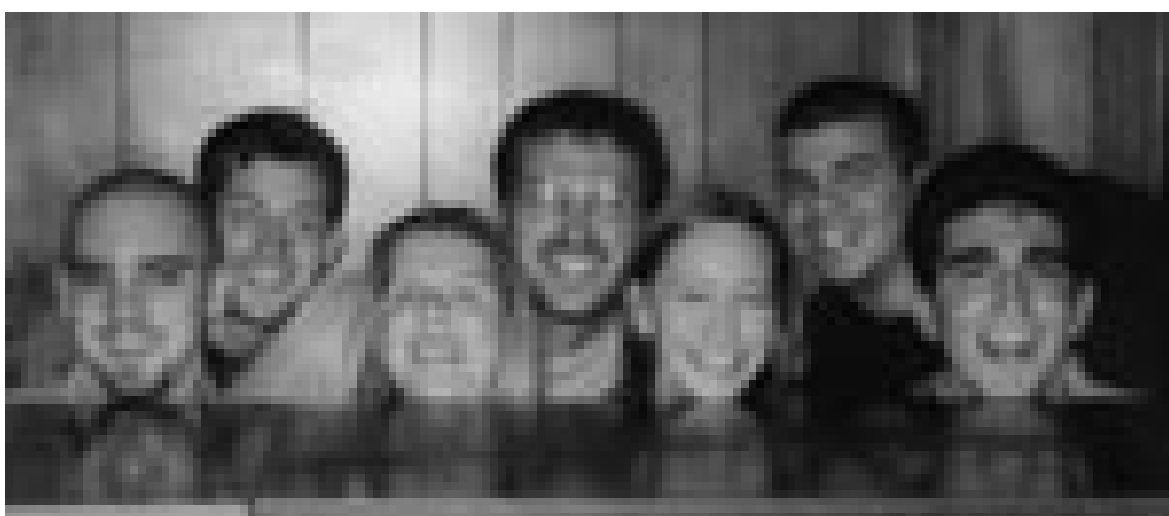


FIGURE 3.3 – Exemple1, image test

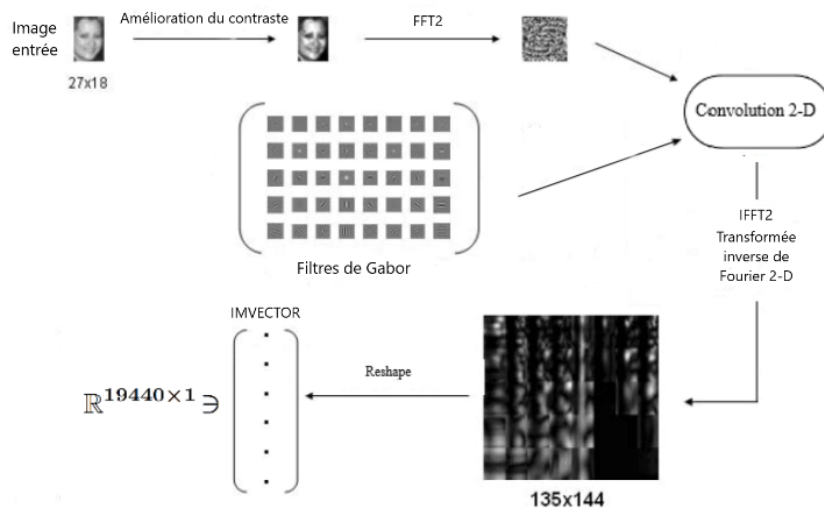


FIGURE 3.4 – Exemple2, image test

Extraire les caractéristique dans une image, dans ce cadre, consiste à transformée une image (27×18) à un vecteur unicolone de $\mathbf{R}^{19440 \times 1}$, on utilise le filtre de Gabor (3.1).

Dans un premier temps il est nécessaire de traiter l'image par la technique présentée au chapitre 1. En effet la fonction matlab 'adaphisteq' est utilisée pour améliorer l'image .Le produit des transformées des fourier entre l'image (visage) et le banc de filtre de Gabor (3.2) est utilisé afin d'écrire le produit de convolution (1.5) qui est trop couteux .

L'inverse de la transformée de Fourier est apliquée à ce produit pour rendre l'image dans ses données spaciales. La taille de l'image résultant est (135×144) , une vectorisation de ce vecteur est utilisée par la fonction 'reshepe' de Matlab, à fin d'avoir une donnée de dimention $\mathbf{R}^{19440 \times 1}$.



3.3 Procédure scanning

Cette procédure est l'étape sur laquelle le code traite le classifieur SVMs, sur une image test (Figure (3.2), Figure(3.1)). Le code 'imscan' dans [13] permet dans un premier temps de confirmer les points pixels qu'une image (27 × 18) est appliquée le classifieur SVMs afin de déduire la classe de cette image. Les visages seront cadrés par un rectangle dans l'image finale.

3.4 Résultats d'exécution

On exécute le code Matlab associé à l'approche de O. Sakhi [13], schématiser dans la figure (4.1), on obtient les images suivantes (voir les figures).

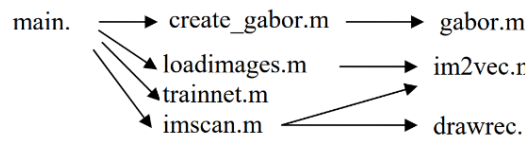


FIGURE 3.5 – Le schéma de code Matlab

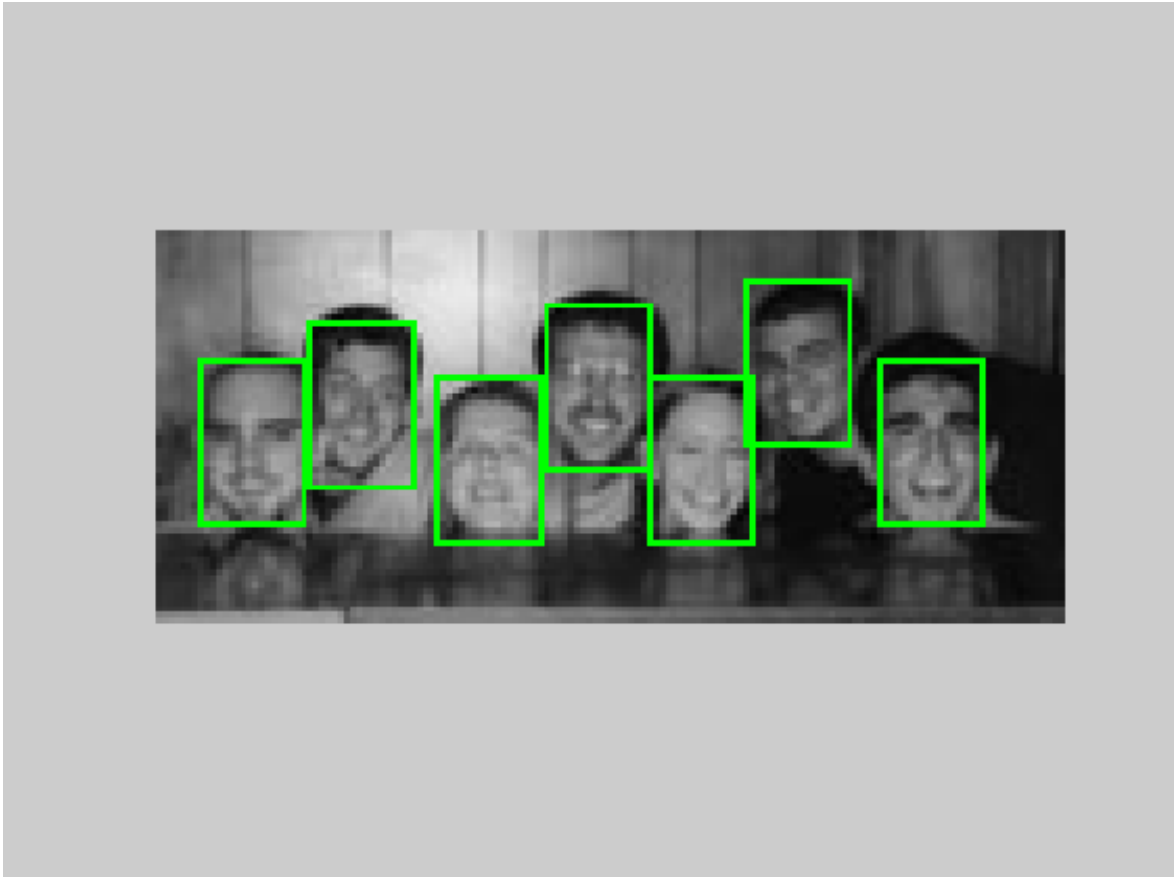


FIGURE 3.6 – Résultat d'exemple 1

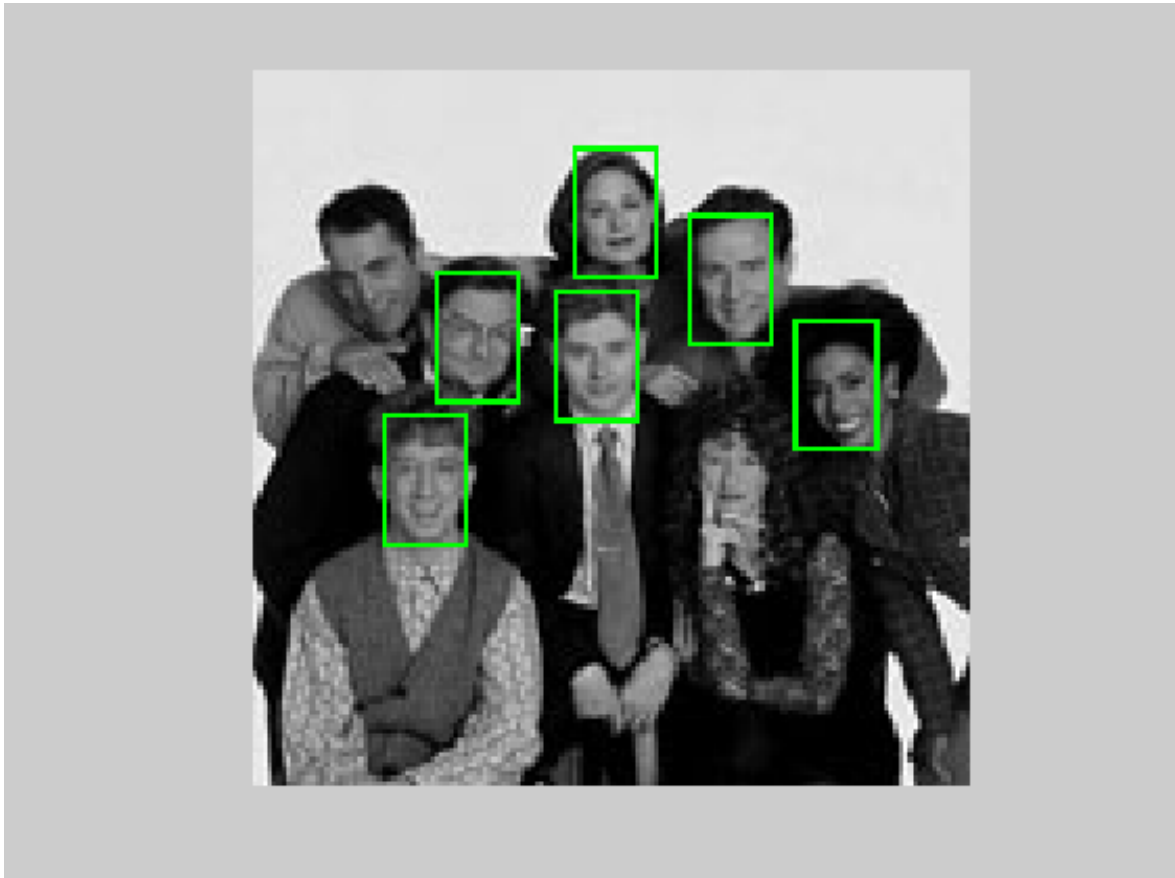


FIGURE 3.7 – Résultat d'exemple 2

3.5 Conclusion générale

La détection faciale (DF) est un domaine pionnier dans le domaine de vision par ordinateur, ceci représente une grande branche de l'intelligence artificielle. Cependant, il s'avère que la détection faciale s'appuie sur plusieurs concepts mathématiques, à la fois pour l'extraction des caractéristiques et la classification. Dans ce travail, nous avons étudié une technique très utilisée dans l'extraction des caractéristiques d'une image à savoir le Filtre de Gabor. Bien que notre objectif, au départ était d'étudier les ondelettes de Gabor¹ [9] et leurs applications dans le problème (DF), les diverses lectures sur le sujet, nous ont permis de constater que le filtre de Gabor était largement suffisant pour effectuer l'étude. Cependant, comme perspective nous envisagerons dans l'avenir, de voir le comportement des ondelettes de Gabor dans ce contexte.

1. Une ondelette définie à partir d'une ondelette mère avec des échelles et des orientations différentes

Bibliographie

- [1] F. Farrokhnia A.K. Jain. Unsupervised texture segmentation using gabor filters, pattern recognition. 1991. [7](#), [18](#), [24](#)
- [2] Andreas Antoniou and Wu-Sheng Lu. *Practical Optimization : Algorithms and Engineering Application, Department of Electrical and Computer Engineering University of Victoria, Canada, Site :aantoniou@shaw.ca*. 2007. [7](#), [28](#)
- [3] Maïtine Bergounioux. Quelques méthodes de filtrage en traitement d'image, site : <https://hal.archives-ouvertes.fr/hal-00512280v1>, 2010. [8](#), [9](#)
- [4] Moufida Berkani chrifa, Gatrani. Detection et reconnaissance de visage,. [24](#)
- [5] S.D. Wang C.J. Lee. Fingerprint feature extraction using gabor filters, electronics letters. 34(4), 1999. [24](#)
- [6] Chapitre5 Cour, Analyse des signaux. Convolution et corelation, site : perso.univ-lemans.fr. [11](#), [13](#), [14](#), [15](#)
- [7] User's Cuide. *Image Processing Toolbox, Mathworks, Site : www.Mathworks.com* ., 2020. [7](#), [8](#), [9](#), [11](#)
- [8] Martinez J.M Samuelide M Gordon M.B Badran F Thira SandHéault L Dreyfus, G. Réseaux de neurones.paris.eyrolles. 39, 2002. [28](#)
- [9] Sonia Hamnane. De l'analyse de fourier traditionnelle aux ondelettes : une approche par le signal musical, université paris vi, pierre et marie curie. 2004-2005. [33](#)
- [10] Michael Fairhurs LinLin Shen, Li Bai. Gabor wavelets and general discriminant analysis for face identification and verification. 16 May 2006. [4](#), [18](#), [24](#), [25](#), [26](#)
- [11] Javier R. Movellan. A comparison of gabor filter methods for automatic detection of facial landmarks. 2002. [15](#), [19](#), [20](#), [24](#)
- [12] Javier R. Movellan. Gabor features in image analysis. 2012. [16](#), [17](#), [18](#), [24](#)
- [13] O.Sakhi. Face detection system, matlab central file exchange, site : https://www.mathworks.com/matlab_central/leexchange/11073-face-detection-system). 2020. [7](#), [28](#), [31](#)
- [14] Javier R.Movellan. Tutorial on gabor filters. 1996. [18](#), [22](#), [24](#)
- [15] Abe S. Support vector machines for pattern classification. second ed.london : Springer. 2010. [7](#)
- [16] Lan N. Sneddon. *FOURIER TRANSFORME, Dover Publication, INC ,New york*. 1991. [13](#)
- [17] Ouiame Talbi. Filtres de gabor pour cnns. 25/may/2018. [7](#), [8](#), [15](#), [16](#)
- [18] V.Vapnik. The nature of statistical learning theory, springer-verlag, new york. 1995,. [7](#), [28](#)
- [19] M. Watanabe T. Yasuda Y. Mitani S. Tomita Y. Hamamoto, S. Uchimura. A gabor filter-based method for recognizing handwritten numerals, pattern recognition. 35(4), 1998. [7](#), [24](#)

Résumé

Dans ce mémoire nous avons utilisé le logiciel Matlab comme environnement de calcul, nous présentons le Filtre de Gabor et les techniques utilisées pour extraire les caractéristiques discriminantes à partir d'une image de visage. Une simulations numériques, nous avons étudié est compris le code de O. Sakhi qui utilise justement le filtre de Gabor pour la détection de visage, quelques tests numériques sont présentés en utilisant ce code.

abstract In this thesis we used the Matlab software as a computational environment, we present the Gabor filter and the techniques used to extract the discriminating characteristics from a face image. A numerical simulations we have studied is understood the code of O. Sakhi which precisely uses the Gabor filter for face detection, some numerical tests are presented using this code