

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de Fin d'études Master

Filière : Informatique

Option : Systèmes Informatiques

Thème :

**Automatisation de la sélection des paramètres optimisés de
filtre Gabor pour la détection des expressions faciales**

Encadré par :

DR. KOUAHLA MOHAMED NADJIB

Présenté par :

BELFARHI aymen

Octobre 2020

Résumé

Pour l'être humain, une expression faciale présente un mouvement ou un geste du visage. Pour déterminer ces expressions faciales il existe un mécanisme général, il se compose de trois étapes : détection de visage, extraction des caractéristiques, classification et décision. Notre projet consiste à détecter les expressions faciales à l'aide des filtres de Gabor dans la partie d'extraction des caractéristiques avec l'utilisation de l'algorithme génétique pour l'optimisation, avec un classifieur SVM pour les machines Learning. Un test expérimental a été fait sur des bases de données JAFFE et CK et la combinaison de ces deux bases de données, le taux de reconnaissance obtenu est 100% pour la base de données JAFFE, 80% pour la base CK, et 83% pour la base combinée.

REMERCIEMENT

Au terme de ce mémoire, je tiens à exprimer mes remerciements et ma profonde gratitude avant tout au bon DIEU, Dieu est grand.

Je saisis de cette occasion pour exprimer mon profond remerciement à mon enseignant encadreur Monsieur **KOUAHLA Mohamed Nadjib** pour son soutien et son louable effort depuis ma troisième année licence vraiment merci pour tout monsieur, mes remerciements y vont aussi à l'endroit de tous les professeurs du département informatique de l'université 08 Mai 1945 de Guelma que j'aime monsieur KOUAHLA ZINEDINE, CHOHLA CHAMSEDINE...etc. Plus particulièrement à monsieur Adel Boughida qui m'a été d'une très grande aide. Pour terminer, je remercie tous ceux qui nous ont aidés à élaborer ce travail.

Dédicace

À mes chers parents, pour les sacrifices déployés pour moi, pour leur amour et leur confiance en moi. Ils ont tout fait pour moi, bonheur et ma réussite. Pour que je serai fort et le meilleur. Merci beaucoup ! une longue vie inchalah.

À ma petite sœur, À mes meilleurs amis : Jouhaina mounder mourad akrem ala djams et les autres amis de près et de loin.

Merci À toute personne qui m'a aidée et soutenue de près et de loin.

Table des matières

1	Etat de l'art	6
1	Introduction	7
2	Expressions faciales	7
2.1	Définition	7
2.2	Pourquoi les expressions faciales existent-elles ?	7
2.3	Descriptions des expressions faciales	7
2.4	Méthodologie de la reconnaissance d'expression faciale	8
3	Filtre de Gabor	9
3.1	Définition	9
3.2	Différents paramètres qui contrôlent la forme et la taille du filtre 2D Gabor	12
3.3	Synthèse des travaux développés pour la reconnaissance d'expres- sions faciales à l'aide du filtre de Gabor	15
4	Algorithmes génétiques	20
4.1	Introduction	20
4.2	Définition	20
4.3	Notions théoriques	20
4.4	Opérateur d'un algorithme génétique	22
4.5	Domaines d'application	23
5	Conclusion	24
2	Conception	25
1	Introduction :	26
2	Architecture du système :	26
3	Les différentes étapes de notre système :	26
3.1	détection de visage :	26
3.2	extraction des caractéristiques avec Filtre de Gabor :	27
4	La classification :	30
4.1	principe :	30
4.2	validation des paramètres du modèle :	30
4.3	sélection des paramètres optimale de SVM :	30
4.4	test du modèle :	30
4.5	classification :	31
5	Conclusion :	31
3	Implémentation et résultats expérimentaux	32
1	Introduction :	33
2	Implémentation du système :	33

2.1	Environnement et les Ressources :	33
2.2	Bibliothèques utilisées :	35
2.3	Les bases de Données :	35
2.4	Implémentation et résultats :	36
3	Résultats expérimentaux :	37
3.1	Discussion :	44
3.2	Conclusion :	46

Table des figures

1.1	Les différents expressions faciales d'Ekman [1]	8
1.2	Méthodologie de la reconnaissance d'expression faciale	9
1.3	Un filtre de Gabor 2-D obtenu en modulant l'onde sinusoïdale avec un gaussien	10
1.4	Une banque de 16 filtres Gabor orientés selon un angle de 11.250 (c'est-à-dire si le premier filtre est à 00, alors le second sera à 11.250, le troisième sera à 22.50, et ainsi de suite.)	11
1.5	: (a) L'image d'entrée d'un éléphant et (b) l'image de sortie après l'avoir traversée la banque de filtres de Gabor	11
1.6	Les caractéristiques de bord orienté correspondantes étant détectées lors de leur passage dans des filtres Gabor orientés individuellement	12
1.7	gardant les autres paramètres inchangés ($\theta = 00$, $\gamma = 0,25$, $\sigma = 10$, $\psi = 0$), et en changeant le lambda de 30 à 60 et 100, la fonction de Gabor devient plus épaisse	13
1.8	gardant les autres paramètres inchangés ($\lambda = 30$, $\gamma = 0,25$, $\sigma = 10$, $\psi = 0$), et en changeant le thêta de 0 à 45 et 90, la fonction de Gabor tourne.	14
1.9	gardant les autres paramètres inchangés ($\lambda = 30$, $\theta = 00$, $\sigma = 10$, $\psi = 0$), et en changeant le gamma de 0,25 à 0,5 et 0,75, la fonction de Gabor se raccourcit.	14
1.10	gardant les autres paramètres inchangés ($\lambda = 30$, $\theta = 00$, $\gamma = 0,25$, $\psi = 0$), et en changeant le sigma de 10 à 30 et 45, le nombre de bandes dans la fonction Gabor augmente.	15
1.11	Bio-inspiration de l'algorithme génétique.	20
1.12	clarifier les éléments de base pour une population	21
1.13	Croisement de deux brins de chromosome	22
1.14	Une mutation	23
1.15	Architecture d'un algorithme génétique	24
2.1	schéma de notre système proposé.	26
2.2	Exemple sur la détection de visage.	27
2.3	Exemple des régions détecté dans un visage.	28
2.4	Exemple d'application d'un filtre de Gabor sur un visage.	29
2.5	Mécanisme de transformation d'un visage a un vecteur de caractéristique	30
3.1	Environnement de google cloud.	34
3.2	Logo de python et son environnement.	34
3.3	Logo de Pycharm.	35
3.4	Quelque images sur les bases de données JAFFE et CK+.	36
3.5	Sélection d'une image d'un visage.	37

3.6	Sélection de la base de données.	38
3.7	L'interface initiale.	38
3.8	L'importation des paramètres de Gabor.	39
3.9	L'extraction des caractéristique.	39
3.10	Capture sur l'apprentissage.	40
3.11	Capture sur les meilleurs paramètres de SVM.	40
3.12	La détection de visage.	41
3.13	Détection des points de repères.	41
3.14	Détection des régions d'intérêt.	42
3.15	L'application du filtre de Gabor sur les trois régions.	42
3.16	Capture sur le vecteur de caractéristique.	43
3.17	Capture sur l'émotion du visage entée.	43
3.18	Comparaison entre le développement des trois bases de données.	44
3.19	Matrice de confusion avec et sans normalisation de la base de données JAFFE.	45
3.20	Matrice de confusion avec et sans normalisation de la base de données CK+.	45
3.21	Matrice de confusion avec et sans normalisation de la base de données JAFFE+CKplus.	46

Liste des tableaux

1.1	Quelques techniques développées pour la reconnaissance des expressions faciales à l'aide du filtre de Gabor.	18
1.2	Quelques techniques développées pour la reconnaissance des expressions faciales à l'aide du filtre de Gabor.	19
3.1	Tableaux comparatif entre notre travail et d'autres travaux.	44

Chapitre 1

Etat de l'art

1 Introduction

Dans ces dernières années, la reconnaissance des expressions faciales est considérée comme un sujet de recherche active vu les diverses applications dans le monde réel, notamment l'interaction homme-machine, la sécurité, les jeux, l'animation, le domaine médical, etc.

Le visage est une partie très importante du corps humain qui est visuellement utilisé à des fins de reconnaissance. La reconnaissance de l'expression faciale utilise généralement un processus de trois étapes comprenant l'acquisition du visage, l'extraction des traits du visage et la construction d'un classificateur pour atteindre l'objectif.

Dans ce chapitre, nous présentons quelques notions sur la reconnaissance des expressions faciales et le rôle de filtre de Gabor pour la reconnaissance faciale et l'importance des algorithmes génétiques pour optimiser les résultats.

2 Expressions faciales

2.1 Définition

L'expression faciale est définie par un ensemble des signes du visage qui traduisent un sentiment, une émotion. L'expression faciale est un changement dans le visage, perceptible visuellement, dû à l'activation (volontaire ou non) de l'un ou de plusieurs des 44 muscles composant le visage [1]. Ainsi, les expressions faciales représentent l'un des éléments les plus importants dans le processus de communication. Les expressions faciales peuvent permettre aussi de distinguer entre certaines émotions spécifiques et procurer de l'information à propos de l'intensité des émotions ressenties (Tremblay, Deschênes, Poulin, Roy, Kirouac, & Kappas, 1993) [2].

2.2 Pourquoi les expressions faciales existent-elles ?

Il ya trois points de vue différents a proposé [3] :

- Les expressions faciales sont des signaux de conversation qui dépendent des intentions de l'émetteur, du comportement du récepteur et du contexte de l'interaction : hausser les sourcils = hésitation, froncer les sourcils = incompréhension, hausser le menton = ignorance.
- Les expressions faciales sont des activateurs et des régulateurs d'émotion dus à la plasticité du cerveau : changent la température de l'hypothalamus pour faciliter ou inhiber la production de neurotransmetteurs reliés aux émotions.
- Les émotions sont au centre de l'explication des expressions faciales : sélection naturelle : 7 émotions de base ; 7 expressions faciales universelles (colère, dégoût, joie, mépris, peur, surprise, tristesse) [1].

2.3 Descriptions des expressions faciales

La recherche la plus notable sur le sujet est venue du psychologue Paul Ekman, qui a été le pionnier de la recherche sur la reconnaissance des émotions dans les années 1960, À partir de cette étude, les six émotions de base ont été proposées [4].



FIGURE 1.1 – Les différents expressions faciales d'Ekman [1]

Joie : est une émotion qui est souvent associée à un état d'esprit qui reflète le contentement, la satisfaction, le plaisir ou la joie, Le bonheur se caractérise par une expression faciale qui fait que quelqu'un soulève les coins de sa bouche vers le haut [5].

Dégoût : émotion associée à des choses insalubres, non comestibles, infectieuses ou offensantes .Le dégoût est caractérisé par une expression faciale qui fait que quelqu'un soulève sa lèvre supérieure, plisse son pont de nez et soulève ses joues [5].

Colère : c'est l'état d'une irritation mineure ou à une rage intense, une émotion souvent associée à une irritation mineure ou à une rage intense [5].

Peur : une émotion associée à des stimuli menaçants ou dangereux, En cas de peur, il est dit qu'une personne soulève les sourcils, ouvre légèrement la bouche et ouvre les yeux d'une manière plus large que la normale [5].

Surprise : c'est une émotion souvent associée à un bref état d'être, La surprise est caractérisée par une expression faciale qui amène quelqu'un à cambrer les sourcils, à ouvrir largement les yeux et à laisser tomber sa mâchoire [5].

Tristesse : une émotion souvent associée aux sentiments de désavantage, de perte et d'impuissance, La tristesse est caractérisée par une expression faciale qui amène quelqu'un à abaisser les coins de sa bouche et à relever la partie intérieure de ses sourcils [5].

2.4 Méthodologie de la reconnaissance d'expression faciale

Le système de reconnaissance des expressions faciales s'applique en trois étapes comme suit [6] :

- Détection du visage [7] .
- Extraction de caractéristiques (partie du repère facial comme les yeux, bouche ou tout le visage) [8].
- Classification des expressions [9].

Dans le système de reconnaissance des expressions faciales, la partie la plus importante est le processus d'extraction des fonctionnalités approprié produira un meilleur système de reconnaissance avec un résultat plus précis.

Les algorithmes utilisés pour l'extraction des traits du visage peuvent être classés en deux groupes principaux, à savoir : méthodes à base **géométrique** [10] qui collectent des points

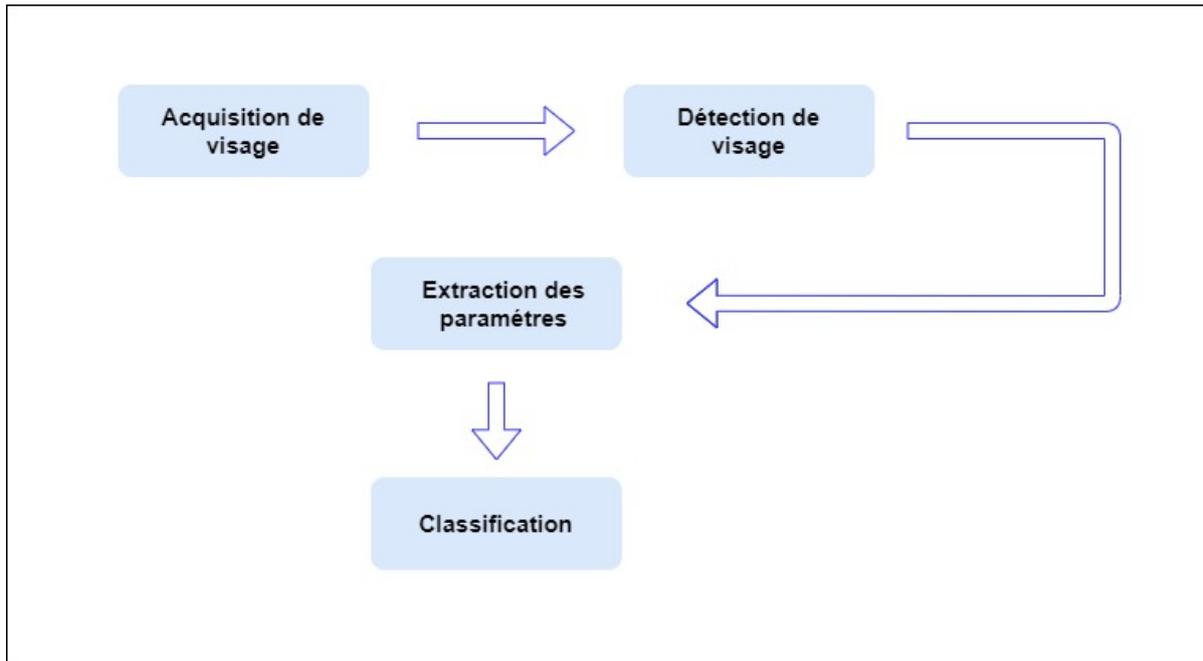


FIGURE 1.2 – Méthodologie de la reconnaissance d'expression faciale

caractéristiques ou le mouvement des points en les retraçant des images du visage et en classant les expressions des entités suivies ; et les méthodes basées sur **l'apparence** [10] qui collectent tout ou partie du visage repères et les organiser comme une longue fonctionnalité de tableau vecteur et les appliquer dans le processus de classification. La méthode basée sur l'apparence a plus d'avantages qu'à la méthode basée sur la géométrie [11].

3 Filtre de Gabor

3.1 Définition

Filtre de Gabor, nommé d'après Dennis Gabor, est un filtre linéaire utilisé dans des milliers d'applications de traitement d'image pour la détection des bords, l'analyse de texture, l'extraction de caractéristiques, etc. Les caractéristiques de certaines cellules du cortex visuel de certains mammifères peuvent être approximées par ces filtres [12]. Il a été démontré que ces filtres possèdent des propriétés de localisation optimales dans le domaine spatial et fréquentiel et sont donc bien adaptés aux problèmes de segmentation de texture. Les filtres de Gabor sont des classes spéciales de filtres passe-bande, c'est-à-dire qu'ils autorisent une certaine « bande » de fréquences et rejettent les autres. Un filtre de Gabor peut être considéré comme un signal sinusoïdal de fréquence et d'orientations particulières, modulé par une onde gaussienne. Un tel filtre Gabor 2D est illustré sur la figure ci-dessus.

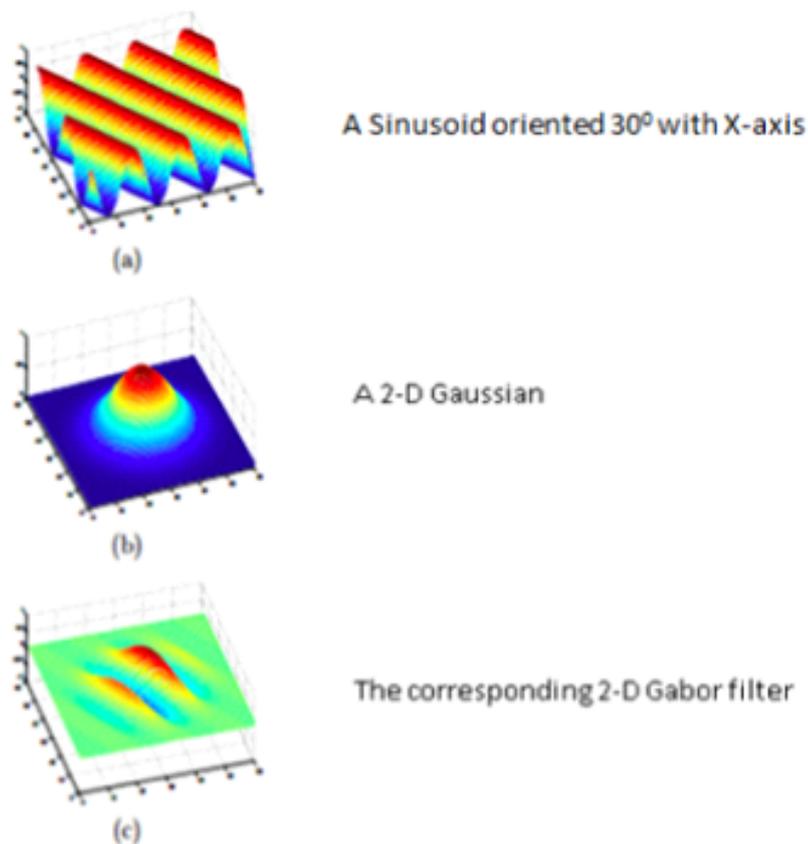


FIGURE 1.3 – Un filtre de Gabor 2-D obtenu en modulant l'onde sinusoïdale avec un gaussien

De la figure ci-dessus, la sinusoïde a été localisée spatialement. Pratiquement, pour faire une analyse à la texture ou obtenir une caractéristique de l'image, avec l'utilisation d'un nombre d'orientations différentes et une banque de filtres de Gabor [12].

Lorsque l'image d'entrée est convoluée (fusionné) avec tous les filtres Gabor, les motifs sont facilement mis en évidence comme illustré à la figure 1.4. Lorsqu'un filtre Gabor est appliqué à une image, il donne la réponse la plus élevée aux bords et aux points où la texture change. Lorsque nous disons qu'un filtre répond à une caractéristique particulière, le filtre a une valeur distinctive à l'emplacement spatial de cette caractéristique [13].

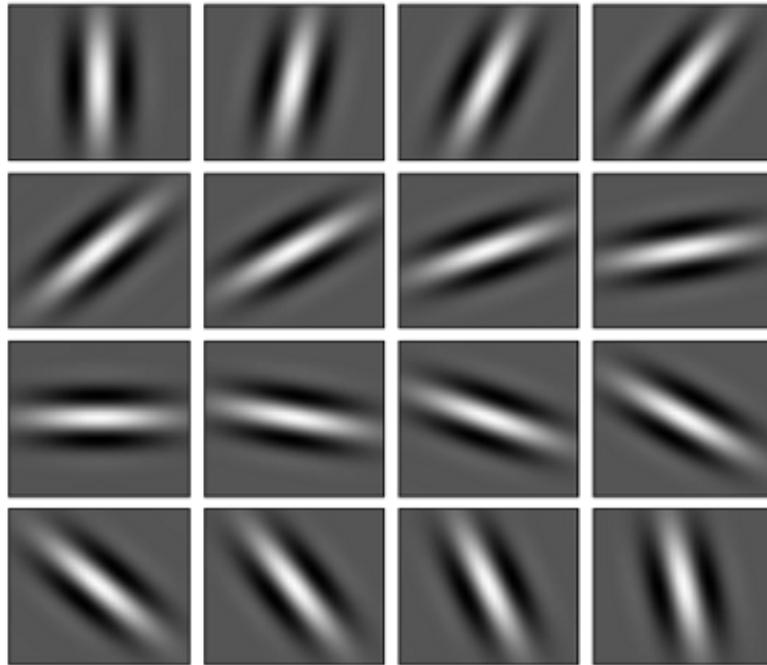


FIGURE 1.4 – Une banque de 16 filtres Gabor orientés selon un angle de 11.250 (c'est-à-dire si le premier filtre est à 00 , alors le second sera à 11.250 , le troisième sera à 22.50 , et ainsi de suite.)



FIGURE 1.5 – : (a) L'image d'entrée d'un éléphant et (b) l'image de sortie après l'avoir traversée la banque de filtres de Gabor

Pour mieux comprendre ce que chaque filtre détecte dans l'image d'entrée, considérez un simple cercle blanc sur fond noir. Lorsque cette image passe à travers chaque filtre de la banque de filtres, le bord du cercle qui est détecté est le bord orienté selon un angle auquel le filtre de Gabor est orienté. La figure 1.6 le montre clairement.

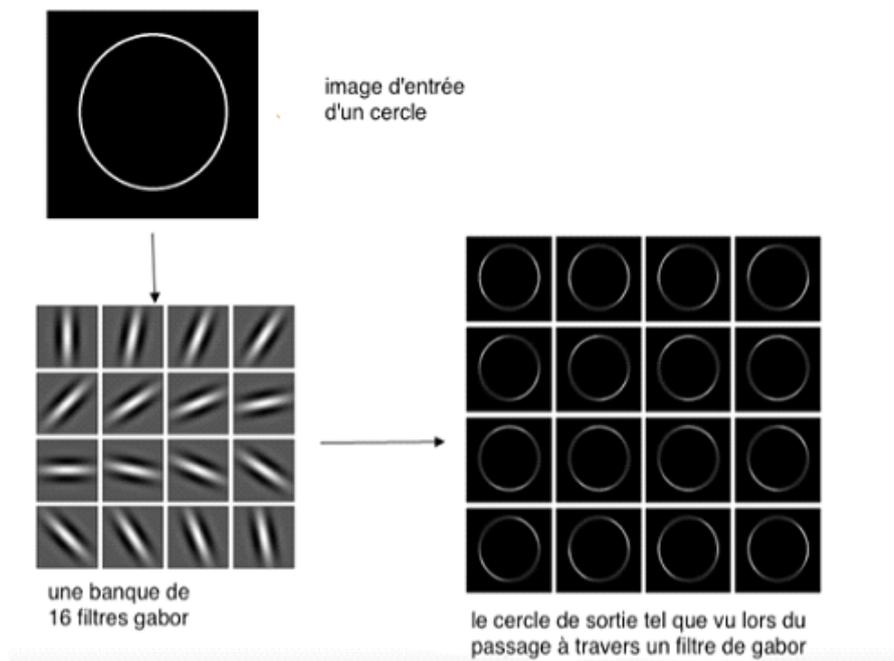


FIGURE 1.6 – Les caractéristiques de bord orienté correspondantes étant détectées lors de leur passage dans des filtres Gabor orientés individuellement

3.2 Différents paramètres qui contrôlent la forme et la taille du filtre 2D Gabor

Il existe certains paramètres qui contrôlent la façon dont le filtre Gabor sera et à quelles fonctionnalités il répondra. Un filtre de Gabor 2D peut être considéré comme un signal sinusoïdal de fréquence et d'orientations particulières, modulé par une onde gaussienne [12]. Le filtre a une composante réelle et imaginaire représentant des directions orthogonales. Les deux composants peuvent être formés en un nombre complexe ou utilisés individuellement. Les équations sont présentées ci-dessous :

Complexe

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(\frac{-(x'^2 + \gamma^2 y'^2)}{2\sigma^2}\right) \exp(i(2\pi \frac{x'}{\lambda} + \psi)) \quad (1.1)$$

Réel

$$g(x, y : \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(\frac{-(x'^2 + \gamma^2 y'^2)}{2\sigma^2}\right) \cos(2\pi \frac{x'}{\lambda} + \psi) \quad (1.2)$$

Imaginaire

$$g(x, y : \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(\frac{-(x'^2 + \gamma^2 y'^2)}{2\sigma^2}\right) \sin(2\pi \frac{x'}{\lambda} + \psi) \quad (1.3)$$

Où

$$x' = x \cos \theta + y \sin \theta \quad (1.4)$$

Et

$$y' = x \sin \theta + y \cos \theta \quad (1.5)$$

Dans l'équation ci-dessus,

λ - Longueur d'onde de la composante sinusoïdale.

Θ - L'orientation de la normale aux bandes parallèles de la fonction de Gabor.

Ψ - Le décalage de phase de la fonction sinusoïdale.

σ - Le sigma / écart type de l'enveloppe gaussienne

γ - Le rapport d'aspect spatial et spécifie l'ellipticité du support de la fonction de Gabor. Les cinq paramètres mentionnés ci-dessus contrôlent la forme et la taille de la fonction Gabor. Le rôle de chaque paramètre est expliqué en détail ci-dessous. Pour illustrer les effets des paramètres, les valeurs suivantes sont choisies comme point de départ :

$$\text{Lambda } (\lambda) = 30$$

$$\text{Thêta } (\theta) = 00$$

$$\text{Gamma } (\gamma) = 0.25$$

$$\text{Sigma } (\sigma) = 10$$

$$\text{Psi } (\psi) = 0$$

Lambda (λ)

La longueur d'onde régit la largeur des bandes de fonction de Gabor. L'augmentation de la longueur d'onde produit des bandes plus épaisses et la diminution de la longueur d'onde produit des bandes plus fines. En gardant les autres paramètres inchangés et en changeant le lambda à 60 et 100, les rayures deviennent plus épaisses [w1].

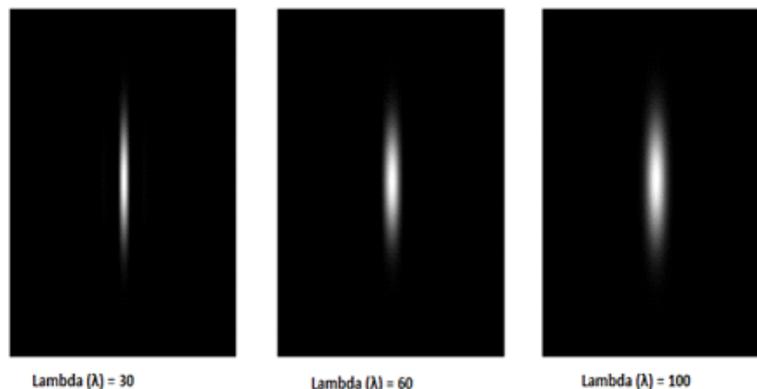


FIGURE 1.7 – gardant les autres paramètres inchangés ($\theta = 00$, $\gamma = 0,25$, $\sigma = 10$, $\psi = 0$), et en changeant le lambda de 30 à 60 et 100, la fonction de Gabor devient plus épaisse

Thêta (θ)

Le thêta contrôle l'orientation de la fonction de Gabor. Le thêta à zéro degré correspond à la position verticale de la fonction de Gabor [w1].

Gamma (γ)

Le rapport d'aspect ou gamma contrôle la hauteur de la fonction Gabor. Pour un rapport d'aspect très élevé, la hauteur devient très petite et pour une valeur gamma très petite, la hauteur devient assez grande. En augmentant la valeur du gamma à 0,5 et 0,75, en gardant les autres paramètres inchangés, la hauteur de la fonction Gabor diminue [w1].

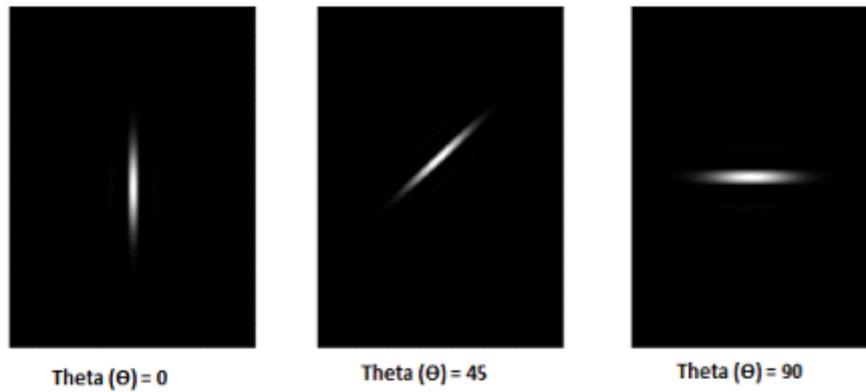


FIGURE 1.8 – gardant les autres paramètres inchangés ($\lambda = 30, \gamma = 0,25, \sigma = 10, \psi = 0$), et en changeant le θ de 0 à 45 et 90, la fonction de Gabor tourne.

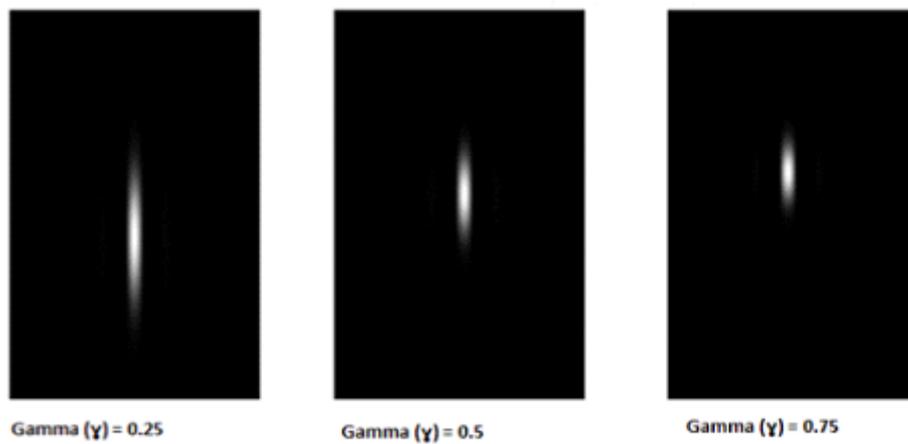


FIGURE 1.9 – gardant les autres paramètres inchangés ($\lambda = 30, \theta = 00, \sigma = 10, \psi = 0$), et en changeant le γ de 0,25 à 0,5 et 0,75, la fonction de Gabor se raccourcit.

Sigma (σ)

La bande passante ou sigma contrôle la taille globale de l'enveloppe Gabor. Pour une plus grande bande passante, l'enveloppe augmente permettant plus de bandes et avec une petite bande passante, l'enveloppe se resserre. En augmentant le sigma à 30 et 45, le nombre de rayures dans la fonction Gabor augmente [w1].

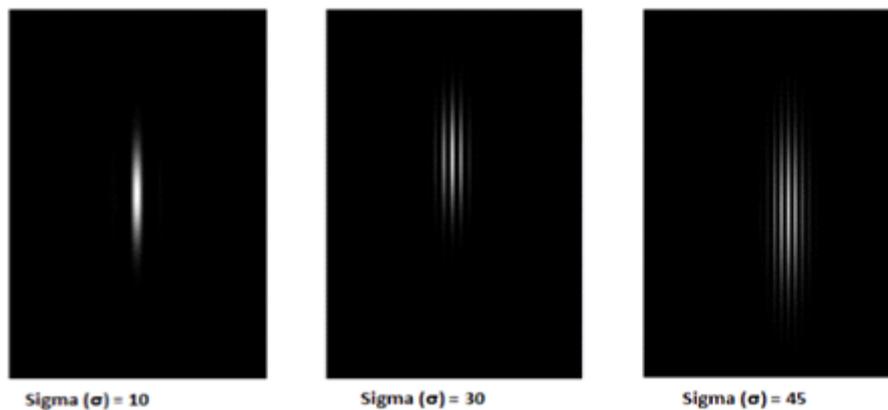


FIGURE 1.10 – gardant les autres paramètres inchangés $\lambda = 30$, $\theta = 00$ $\gamma = 0,25$, $\psi = 0$), et en changeant le sigma de 10 à 30 et 45, le nombre de bandes dans la fonction Gabor augmente.

3.3 Synthèse des travaux développés pour la reconnaissance d'expressions faciales à l'aide du filtre de Gabor

Le tableau suivant (tableau1) présente quelques techniques récentes développées pour la reconnaissance des expressions faciales à l'aide du filtre de Gabor :

Ramachandran Vedantham et ses collègues ont proposé un système de reconnaissance des expressions faciales en temps réel dans une vidéo en utilisant le classificateur du plus proche voisin. La méthode proposée comporte cinq étapes, La première étape est d'acquérir de la vidéo en temps réel à partir d'une webcam en utilisant les Progiciels d'acquisition d'images existants de MATLAB. Ensuite détecter le visage dans les images en utilisant la méthode Viola Jones avec les caractéristiques du Haar. La troisième étape consiste à trouver les caractéristiques importantes dans un visage en utilisant le filtre du Gabor (). Finalement, ils ont utilisé la méthode des K plus proche voisin (KNN) pour la classification. La méthode a été testée sur la base de données Cohn Kanade qui contient six expressions de base telles que la colère, la joie, le dégoût, la peur, la tristesse et la surprise. Cette méthode a enregistré un taux de reconnaissance moyen de 96,67% [14].

Jun Zhou et Al ont présenté une méthode de reconnaissance des expressions faciales basée sur la transformation en ondelettes de Gabor comme caractéristiques. Ils ont utilisé la méthode factorisation d'une matrice non négative pour réduire la dimension des caractéristiques de l'expression. A la fin, un classifieur à deux couches (TLC) est conçu pour la reconnaissance des expressions. Les expériences sont effectuées sur la base de données d'expressions faciales JAFFE, les résultats montrent que la méthode proposée est plus performante : si on fait une comparaison avec d'autres méthodes on trouve que le taux de reconnaissance de la méthode (image + NMF + KNN) est estimé : 83,7% et la méthode (Gabor + NMF + KNN) est : 93,7% par contre la méthode (Gabor + PCA + TLC) environ : 96,1% et la méthode proposée dans cet article avec un taux de 98,1% [15].

Archana Kumari Sharma et Al ont présenté une analyse comparative des différentes techniques d'extraction de caractéristiques dans les systèmes de reconnaissance des expressions faciale (FER). Ils ont groupé les caractéristiques comme des caractéristiques basées sur l'apparence (caractéristiques non géométriques / non structurelles) et les caractéris-

tiques géométriques / structurelles représentent le contour et la position de la partie du visage comme le front, les yeux, le nez, les lèvres et le menton. Autres chercheurs ont travaillé sur la combinaison de la caractéristique géométrique et de la caractéristique basée sur l'apparence pour la reconnaissance efficace de l'expression faciale. Une autre classification de caractéristiques basée sur le domaine spatial et le domaine fréquentiel. Les approches dans le domaine spatial sont LBP (modèle binaire local), PCA (analyse en composantes principales) et filtre de Gabor. L'approche dépendante du domaine fréquentiel comprend les techniques d'extraction de caractéristiques basées sur la technologie DWT (Discrete Wavelet Transform), DCT (Discrete Cosine Transform) et Fourier Transform [16].

Hung-Hsu Tsai et Yi-Cheng Chang ont présenté une nouvelle technique de reconnaissance d'expression faciale (FER) basée sur plusieurs contributions comme : La combinaison des caractéristiques de Haar avec le filtre de selfquotient image (SQI) pour la détection de visage. Ensuite, les caractéristiques angular radial transform (ART), discrète cosin transform (DCT) et le filtre de Gabor (GF) sont simultanément utilisés pour former un SVM qui joue le rôle d'un classifieurs multi-classes. La méthode a été testé sur trois base de données différents Cohn-Kanade face image (CKFI), FG-NET et la base (JAFFE) [11].

Kunika Verma et Ajay Khunteta ont mentionné une méthode de reconnaissance d'expressions faciales à l'aide du filtre de Gabor et un réseau de neurones artificiel multicouche. Dans un premier temps l'utilisation de l'algorithme de viola Jones pour la détection visage, après la détection le filtre de Gabor est utilisé pour l'extraction des traits du visage et les caractéristiques faciales significatives (utilisation d'une banque de filtres de Gabor 5x8). Enfin la classification de l'expression faciale se fait via un algorithme de rétro propagation multicouche à action anticipée (NN). La technique proposée par ces auteurs se fait à l'aide de la base de données JAFFE atteinte un taux de reconnaissance 85,7% [12].

MAHESH M. GOYANI, a essayé d'accroître la précision et la robustesse globales du système de reconnaissance d'expressions faciales. D'abord la première étape consiste à détecter le visage par la méthode de Haar Cascade. Puis l'extraction de deux types de caractéristiques en utilisant le filtre de Gabor pour les détails structurels et le Modèle directionnel local (LDP) pour les détails texturaux qui se sont fusionnées à l'aide de l'analyse de corrélation canonique (CCA). Enfin, l'utilisation de l'analyse en composantes principales (ACP) et de l'analyse discriminante linéaire (LDA) pour obtenir un vecteur de caractéristique optimal. La technique est testée sur un ensemble de données de référence comme CK, JAFFE, TFEID et CASIA-VIZ et sur deux ensembles de données d'expression spontanée appelés SFEW (standard) et WESFED (interne), dans des conditions différentes. Les résultats sont comme suit : dans la base de données CK la précision égale à 99.1% et dans la base de données JAFFE la précision égale à 100% par contre avec la base de CASIA-VIZ la précision égale à 92% et à la base SFEW la précision égale à 72.1%. [17].

Hamid Sadeghi et Abolghasem-A. Raie ont présenté un système de reconnaissance des expressions faciales qui comprend trois étapes essentielles, la première étape consiste à détecter le visage avec la méthode LANDMARK. Ensuite l'extraction des caractéristiques est la deuxième étape, dans cette étape l'image du visage détecté est convoluée avec le filtre de Gabor puis les matrices de convolution obtenues sont correctement codées sur la base des réponses maximales et minimales. A la fin de cette étape d'extraction, le vecteur de caractéristique est obtenu en calculant l'histogramme de ces codes. Enfin pour la clas-

sification des caractéristiques les auteurs ont utilisé le classificateur bien connu SVM et la boîte à outils LIBSVM est utilisée. Les auteurs ont testé la méthode sur ces ensembles de données CK+, SFEW et MMI et les résultats sont : dans la base CK+ le taux de reconnaissance égal à 95.72% et dans la base SFEW le taux de reconnaissance égal à 36.92% et dans la base MMI égal à 71.88% [18].

Réf	detection de visage	Caractéristiques	BDD images/vidéos	Algorithme pour optimisation	Algorithme pour classification	Résultats	Limites de l'approche
[14]	Haar	Gabor	Cohn Kanade		KNN	96,67%	
[15]		Ondelettes de Gabor	JAFFE	G-NMF	TLC	98,1%	Des limites pour suivre et identifier les dynamiques Expressions
[16]		Non géométriques / non structurales ; géométriques / structurales ; Combinaison des deux méthodes , Gabor		LBP ; PCA	DWT, DCT, Fourier Transforme [].		Comparaison entre plusieurs approches et chacune a ses propres limites
[11]	Haar, SQI	Gabor, DCT, ART	CKFI,FG-NET, JAFFE		SVM	98.59%	

TABLE 1.1 – Quelques techniques développées pour la reconnaissance des expressions faciales à l'aide du filtre de Gabor.

Réf	detection de visage	Caractéristiques	BDD images/vidéos	Algorithme pour optimisation	Algorithme pour classification	Résultats	Limites de l'approche
[12]	Viola jones	Réseau neural, Gabor	JAFFE		ANN	85,7%	Pas robuste avec tels changements comme l'âge et les cultures (par exemple les Européens et les Asiatiques)
[17]	Haar	Gabor, LDP, CCA	CK, JAFFE, TFEID, CASIA-VIZ, SFEW	ACP, LDA		99.1%, 100%, 92%, 72.1%	
[18]	LANDMARK	Gabor	CK+, SFEW, MMI	SVM, LIBSM		95.72%, 36.92%, 71.88%	

TABLE 1.2 – Quelques techniques développées pour la reconnaissance des expressions faciales à l'aide du filtre de Gabor.

4 Algorithmes génétiques

4.1 Introduction

GA est un algorithme de recherche heuristique adaptatif basé sur les idées évolutives de la sélection naturelle et de la génétique. Les techniques de base de l'AG sont conçues pour simuler les processus dans les systèmes naturels nécessaires à l'évolution, y compris l'héritage, la mutation, la sélection naturelle, et l'hybridation, en particulier ceux qui suivent les principes initialement posés par Charles Darwin de «survie du plus apte». Étant donné que dans la nature, la concurrence entre les individus pour des ressources limitées fait que les individus les plus aptes dominent sur les plus faibles [19].

4.2 Définition

L'algorithme génétique (AG) [20] est une technique d'optimisation populaire, bio-inspirée et basée sur les concepts de génétique naturelle et les théories de sélection naturelle proposées par Charles Darwin. L'algorithme fonctionne sur trois opérateurs génétiques de base de sélection, de croisement et de mutation [21].



FIGURE 1.11 – Bio-inspiration de l'algorithme génétique.

4.3 Notions théoriques

Population

La population pour un AG est analogue à la population des êtres humains sauf qu'à la place des êtres humains, nous avons des solutions candidates représentant des êtres humains[21].

Chromosome

Un chromosome est l'une de ces solutions au problème donné [21].

Gène

Un gène est une position d'élément d'un chromosome [21].

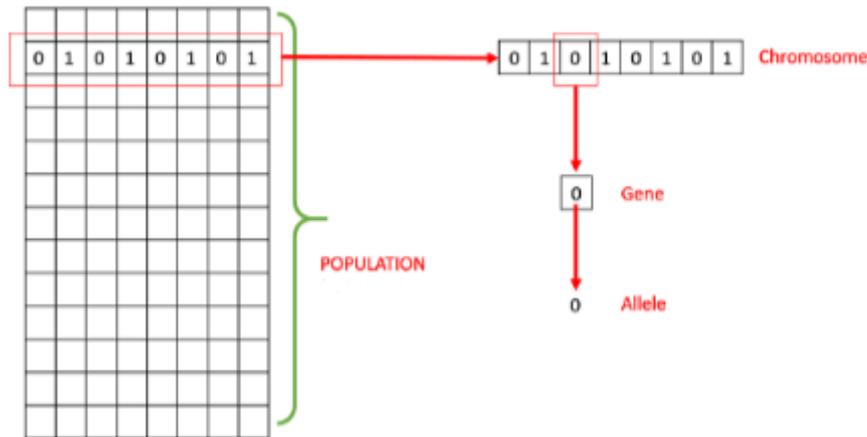


FIGURE 1.12 – clarifier les éléments de base pour une population

Allèle

C'est la valeur attribué à un gène pour un chromosome particulier [21].

Initialisation de la population

Pour l'initialisation de la population, deux questions se posent : Quelle est la taille initiale de la population ? et Quelle est la procédure pour initialiser la population ?

Concernant la taille de la population, l'objectif est de trouver un compromis entre la complexité globale et la performance de la solution (Holland, 1992). Une petite population peut ne pas converger et une grande population peut exiger une quantité excessive de mémoire et de temps de calcul. Il s'avère que la taille de la population initiale doit être soigneusement choisie, adaptée au problème.

Deuxièmement, pour initialiser une population, il existe généralement deux manières : l'initialisation heuristique ou l'initialisation aléatoire. L'initialisation heuristique, même si elle permet de converger plus rapidement vers une solution, a le problème de contraindre l'AG à rechercher des solutions dans un domaine spécifique et elle peut ne pas converger vers un optimum global (Hue, 1997). Une initialisation aléatoire est un facteur facilitant pour empêcher GA de rester coincé dans un optimum local [W2].

Fonction de fitness

La fonction de Fitness permet à l'AG d'analyser les performances de chaque chromosome de la population. La fonction de fitness est la seule relation entre l'AG et l'application elle-même, elle doit être choisie avec soin. La fonction de fitness doit refléter correctement l'application en ce qui concerne la façon dont les paramètres doivent être minimisés [W2].

Élitisme

L'opérateur élitiste assure que l'AG ne s'aggravera pas au fur et à mesure de sa progression. L'opérateur élitiste copie le meilleur chromosome vers la génération suivante en contournant les opérateurs de croisement et de mutation. Cela garantit que le meilleur chromosome ne diminuera jamais en forme [21].

Convergence (Arrêt du processus)

Il est transmis de génération en génération que la fonction adaptative du meilleur chromosome et la valeur moyenne de l'ensemble de la population se développeront progressivement et tendront vers l'optimum (global) [W2]. La convergence est une amélioration progressive vers l'uniformité. Lorsque 95% de la population a la même valeur génétique, le gène a déjà convergé. Lorsque tous les gènes sont fusionnés, les populations sont fusionnées.

4.4 Opérateur d'un algorithme génétique

Sélection

L'opérateur de sélection sélectionne les chromosomes de la génération actuelle comme parents de la génération suivante [W4].

CROSSOVER

Crossover ou croisement est la principale routine de recherche locale de l'AG. L'opérateur de croisement calcule deux descendants pour chaque paire parent donnée par l'opérateur de sélection. Ces descendants, après mutation, constituent la nouvelle génération. La reproduction a pour résultat que la paire de descendants est exactement égale à la paire parentale [W4].

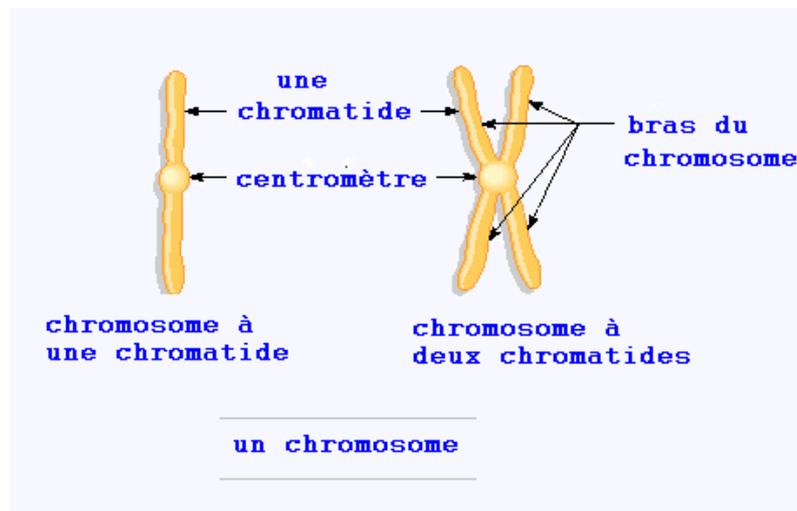


FIGURE 1.13 – Croisement de deux brins de chromosome

Mutation

Cet opérateur consiste à changer la valeur allélique d'un gène avec une probabilité p_m très faible, Une mutation consiste simplement en l'inversion d'un bit (ou de plusieurs bits) [W4]. On peut donc résumer la mutation de la façon suivante :

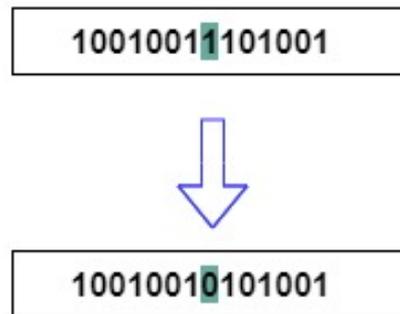


FIGURE 1.14 – Une mutation

Architecture d'un Algorithme Génétique

Le processus général pour un AG est décrit comme suit (Eberhart et al., 1996), (Kim et Cho, 2000) :

Étape 1 : Initialisez une population.

Étape 2 : calculez la fonction de remise en forme pour chaque chromosome dans la population.

Étape 3 : Reproduisez les chromosomes, en fonction de leur aptitude.

Étape 4 : Effectuez le croisement et la mutation.

Étape 5 : Revenir à l'étape 2 ou arrêter selon un état donné les critères d'arrêt.

4.5 Domaines d'application

Les Algorithmes Génétiques peuvent être utilisés pour optimiser des fonctions numériques difficiles, d'emplois du temps, traitement d'image, contrôle de systèmes industriels...Aussi bien que pour contrôler un processus évoluant dans le temps : Acclimatation et l'adaptation des communautés à des conditions changeantes [W3].

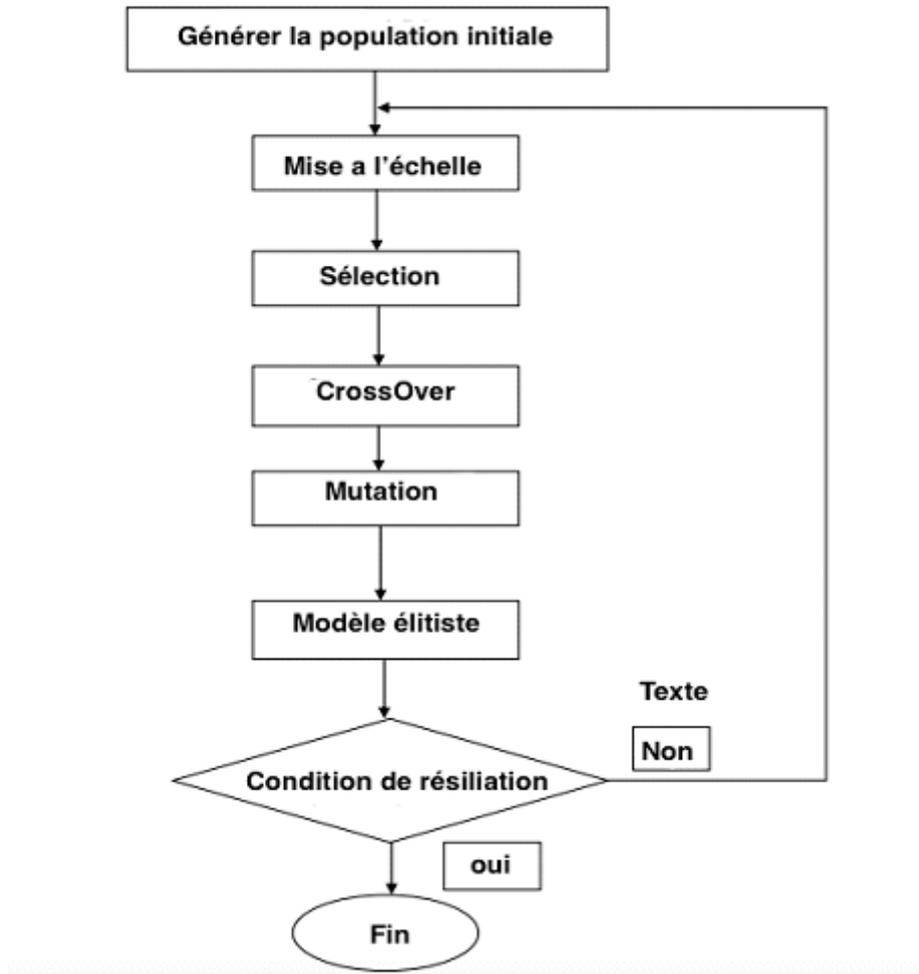


FIGURE 1.15 – Architecture d'un algorithme génétique

5 Conclusion

Dans ce premier chapitre, on a présenté quelques notions et des concepts pour la reconnaissance des expressions faciales. Nous avons eu à définir les expressions faciales, et qu'est-ce que le filtre de Gabor et ses paramètres et l'utilité de chaque paramètre avec des techniques récentes développées pour la reconnaissance des expressions faciales à l'aide du filtre de Gabor. A la fin on a présenté le principe des algorithmes génétiques et sa mode de fonctionnement. Nous avons également présenté trois familles pour la reconnaissance des expressions faciales, l'applications de ces axes pour atteindre l'objectif sera diffuser dans le deuxième chapitre et nous rentrerons plus en détail dans la technique.

Chapitre 2

Conception

1 Introduction :

Les expressions faciales sont utilisées comme des paramètres par les humains pour la communication non verbale, ces paramètres sont considérés comme des indicateurs de la situation émotionnelle de la personne. En général le système FER (abréviation) constitué de trois étapes, pour cela on va utiliser 1) la détection visage dans une image, après 2) L'extraction des caractéristiques de l'émotion, finalement 3) la reconnaissance de l'expression faciale en utilisant la classification.

2 Architecture du système :

Pour notre système, nous proposons une méthode pour la reconnaissance des expressions faciales, basée sur les filtres de Gabor pour l'extraction des caractéristiques, SVM [22] pour la classification et un algorithme génétique comme un outil d'optimisation. D'abord on a utilisé pour la première étape (Détection de visage) l'algorithme Viola-Jones, après ça on extrait les régions d'intérêt en utilisant le filtre de Gabor, après on utilise l'algorithme génétique pour optimiser les paramètres de Gabor et les hyper paramètres du classificateur SVM [22]. Finalement le modèle SVM entraîne et détecte les émotions. Notre système proposé est modélisé par une figure ci-dessus :

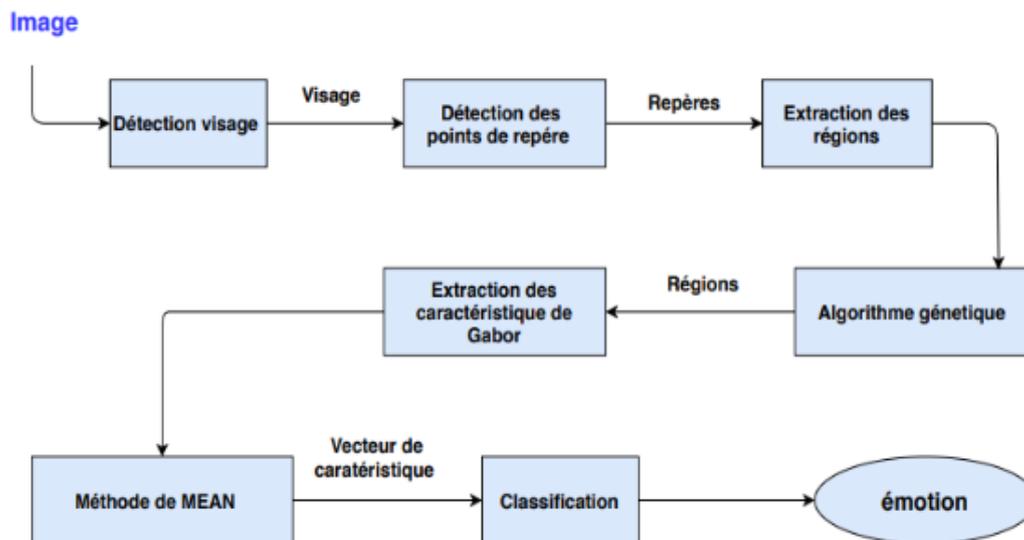


FIGURE 2.1 – schéma de notre système proposé.

3 Les différentes étapes de notre système :

3.1 détection de visage :

Au début avant la détection on fait deux prétraitements pour chaque image de la base de données, la première c'est l'alignement de visages [] consiste à centré les visages sur l'image et pivoter les yeux sur une ligne horizontale (c'est-à-dire que le visage est tourné de telle sorte que les yeux se trouvent le long des mêmes coordonnées y) plus

l'approximation de la taille des visages dans les images. Le deuxième prétraitement est l'égalisation d'histogramme [] utilisé pour améliorer l'apparence de l'image. Dans cette partie très importante dans notre système, consiste à détecter le visage dans une image en utilisant l'algorithme de Viola Jones [23].

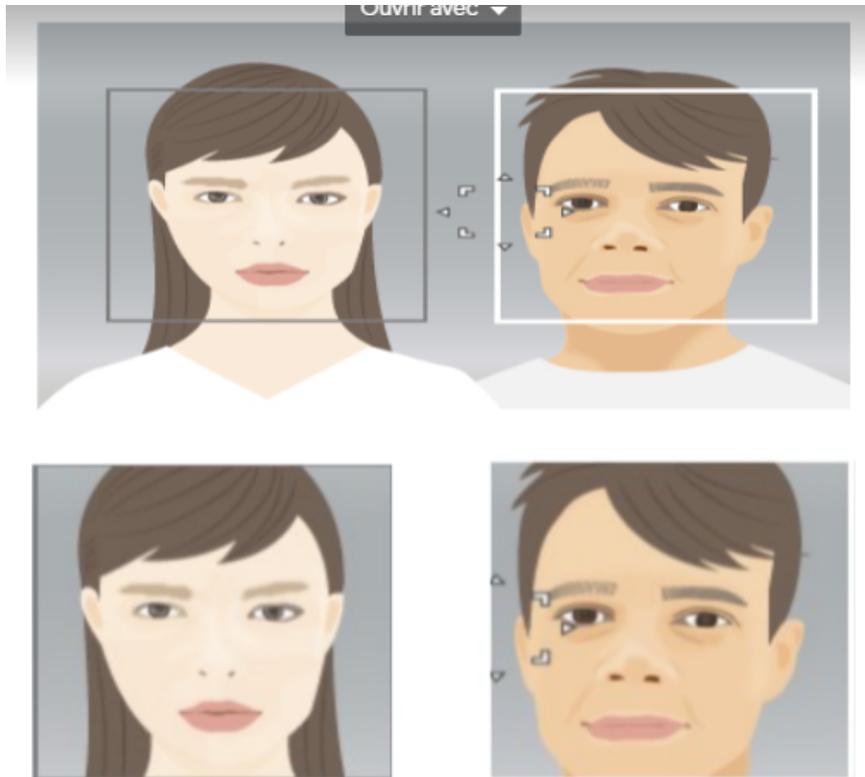


FIGURE 2.2 – Exemple sur la détection de visage.

3.2 extraction des caractéristiques avec Filtre de Gabor :

Cette étape combine deux étapes importantes pour l'extraction des caractéristiques.

extractions des régions d'intérêts :

Notre technique propose d'extraire trois régions qui sont les régions d'intérêts pour la détection des expressions faciales, et ces régions sont : 1- œil droit avec son sourcil. 2- œil gauche avec son sourcil. 3- la bouche. Nous allons extraire les caractéristiques qui influencent la détection des émotions à partir ces régions, pour effectuer cette étape on a besoin de détecter les repères depuis le visage.

1. **Détection des repères [24] :** L'idée d'utiliser des points de repère pour détecter les régions est inspirée par plusieurs chercheurs. Pour cela on utilise l'algorithme de (Face Landmark Détection) qui est basé sur l'ensemble d'arbres de régression [24]. Cette technique utilise une fonction (différences d'intensité des pixels) pour estimer directement les positions des points de repère. L'algorithme est rapide pour détecter un ensemble de 68 points de repère sur un visage donné []. Après la détection des points, on utilise certains de ces points pour définir les trois régions (œil droit avec son sourcil, œil gauche avec son sourcil, la bouche), qui se présentent sous la forme d'un rectangle.

Un facteur qui est très important lors la détection des régions, la dimension moyenne de rectangle, donc on a calculé la dimension moyenne de chaque rectangle pour chaque type de région, après nous redimensionnons toutes les régions avec leur dimension moyenne. Cette opération pour que la taille du vecteur d'entités soit le même pour chaque visage de la base de données. Par exemple, la taille de chaque région pour la base de données JAFFE : 1) Région 1 : 55px 60px. 2) Région 2 : 55 px 60 px. 3) Région 3 : 47 px 23 px.

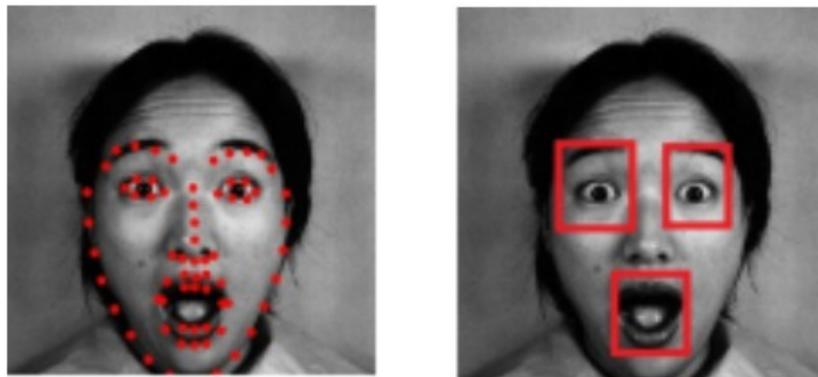


FIGURE 2.3 – Exemple des régions détecté dans un visage.

2. **Extraction des caractéristiques avec le filtre de Gabor :** Dans cette partie, nous parlerons sur le filtre de Gabor et comment ce filtre fait l'extraction des caractéristiques, et pour obtenir des fonctionnalités à partir des régions on utilise une banque de filtres de Gabor avec un nombre certain d'orientations et de fréquences. Dans l'étape suivante, nous parlerons de la façon de déterminer la valeur de nombre d'orientations et de fréquences en fonction de l'algorithme génétique. Par exemple avec un nombre d'orientations égal à cinq et de fréquences égales à deux. Les paramètres détaillés des filtres Gabor sont : 1) Taille du filtre = $3\text{px} * 3\text{px}$. 2) Longueurs d'onde = 0,4, 0,8. 3) Orientations = 5, 2 5, 5, 5, 4) Écart type = 3. 5) Rapport d'aspect spatial = 1. 6) Décalage de phase = $\pi/2$. L'exemple de la figure 2.4 montre l'application de l'un de ces filtres Gabor sur un visage.

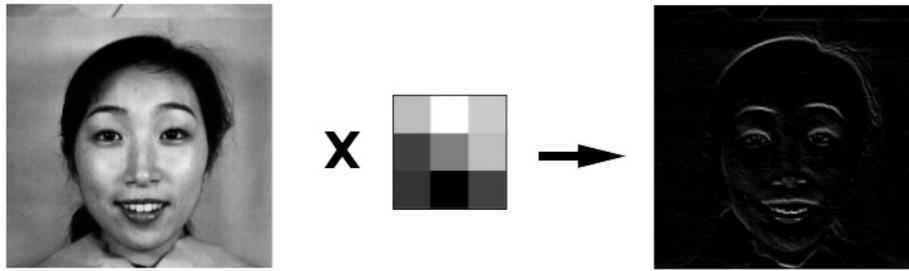


FIGURE 2.4 – Exemple d'application d'un filtre de Gabor sur un visage.

3. **Réduction de caractéristiques** : Après l'application de filtres de Gabor, nous utilisons une méthode Average absolute deviation [25] pour construire un vecteur de caractéristiques à partir des images appliquées par Gabor. On note que la méthode Average absolute deviation avec sa formule :

$$\frac{1}{n} \sum_{i=1}^n = \frac{|x_1 - m| + |x_2 - m| + \dots + |x_n - m|}{n} \quad (2.1)$$

est la distance entre chaque valeur de l'ensemble de données et la moyenne ou la médiane de cet ensemble de données. Le calcul de cet écart doit être en trois étapes : 1. calculez la moyenne de toutes les valeurs. 2. Trouvez la distance de chaque valeur de cette moyenne (soustrayez la moyenne de chaque valeur, ignorez les signes moins). 3. calculez ensuite la moyenne de ces distances.

Par exemple : comme l'ancien exemple en haut d'un nombre d'orientations égal à cinq et de fréquences égales à deux, nous obtenons quarante images et en ajoutant les trois régions nous obtenons 120 images. Pour chaque image, on applique la méthode de Mean [25] on obtient une valeur et on fait la même méthode pour le reste des images donc on aura 120 valeurs à la fin. Ces valeurs font ce qu'on appelle vecteur de caractéristique.

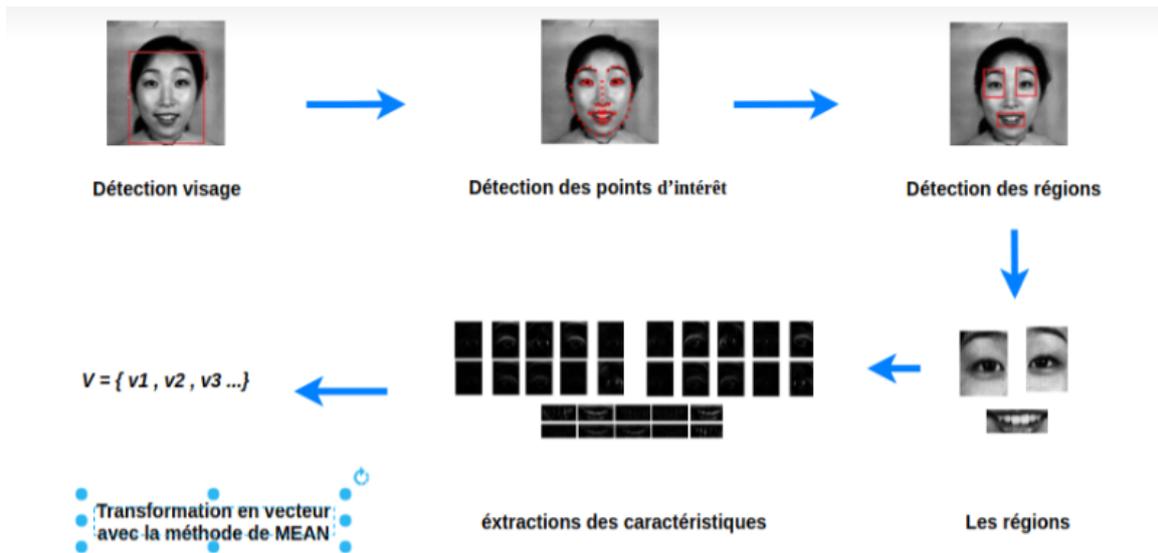


FIGURE 2.5 – Mécanisme de transformation d'un visage a un vecteur de caractéristique

4 La classification :

4.1 principe :

dans l'approche proposée, nous utilisons l'algorithme de SVM (support vector machine) pour l'apprentissage du modèle sur la base de données JAFFE [26] , pour cela, nous considérons 80% pour l'apprentissage et la validation, et 20% de la base pour le test.

4.2 validation des paramètres du modèle :

Premièrement, on utilise la base d'apprentissage pour la validation des paramètres de SMV (c, gamma, kernel) avec 10-fold cross validation, on découpe la base d'apprentissage sur 10 folds. Chaque fois, on change le fold du test et on utilise le reste du fold (9 folds) pour l'apprentissage. On obtient 10 taux de reconnaissance, et on calcule le taux moyen à la fin.

4.3 sélection des paramètres optimale de SVM :

Pour cela, on travaille avec GridSearch [27]. Cette technique consiste à trouver en essayant simplement toutes les combinaisons et voir quels paramètres fonctionnent le mieux. L'idée principale derrière cela est de créer une grille d'hyperparamètres et d'essayer simplement toutes leurs combinaisons. Le meilleur triple de paramètres (C, Gamma, Kernel) on l'utilise dans notre modèle.

4.4 test du modèle :

On teste la performance du modèle appris sur la base test (20%).

4.5 classification :

Le classifieur optimiser va prédire l'émotion du vecteur entrée (Vecteur de 120 paramètres) à une des émotions possibles.

5 Conclusion :

Dans ce chapitre, nous avons défini l'architecture générale de notre système et les fonctionnalités de chaque partie de l'architecture telles que détection visage, extraction des caractéristiques en utilisant filtre de Gabor, classification. Avec cette conception, on peut implémenter notre système qu'on va le détailler dans le chapitre suivant.

Chapitre 3

Implémentation et résultats expérimentaux

1 Introduction :

L'objectif de ce chapitre est de présenter les étapes de l'implémentation de l'approche proposée dans le cadre d'un système de reconnaissance des expressions faciales à l'aide de filtre de Gabor. On commence tout d'abord par la présentation des ressources, du langage et de l'environnement de développement utilisé dans notre travail. Puis les étapes de la réalisation du modèle et on termine par les tests effectués. Ce chapitre est composé de deux parties, l'implémentation du système et les résultats expérimentaux des tests.

2 Implémentation du système :

2.1 Environnement et les Ressources :

1. **Google colab** : Colab est un environnement de notebook Jupyter gratuit qui s'exécute entièrement dans le cloud. Plus important encore, il ne nécessite pas de configuration et les blocs-notes que vous créez peuvent être modifiés simultanément par les membres de votre équipe - tout comme vous modifiez des documents dans Google Docs. Colab prend en charge de nombreuses bibliothèques d'apprentissage automatique populaires qui peuvent être facilement chargées dans votre ordinateur portable [28].

Pour quoi Google Colab?. En tant que programmeur, vous pouvez effectuer les opérations suivantes à l'aide de Google Colab :

- ✓ Ecrire et exécuter du code en Python.
- ✓ Documentez votre code qui prend en charge les équations mathématiques.
- ✓ Créer / télécharger / partager des blocs-notes.
- ✓ Importer / enregistrer des blocs-notes depuis / vers Google Drive.
- ✓ Importer / publier des blocs-notes depuis GitHub.
- ✓ Importez des ensembles de données externes, par ex. depuis Kaggle.
- ✓ Intégrer PyTorch, TensorFlow, Keras, OpenCV.
- ✓ Service Cloud gratuit avec GPU gratuit.

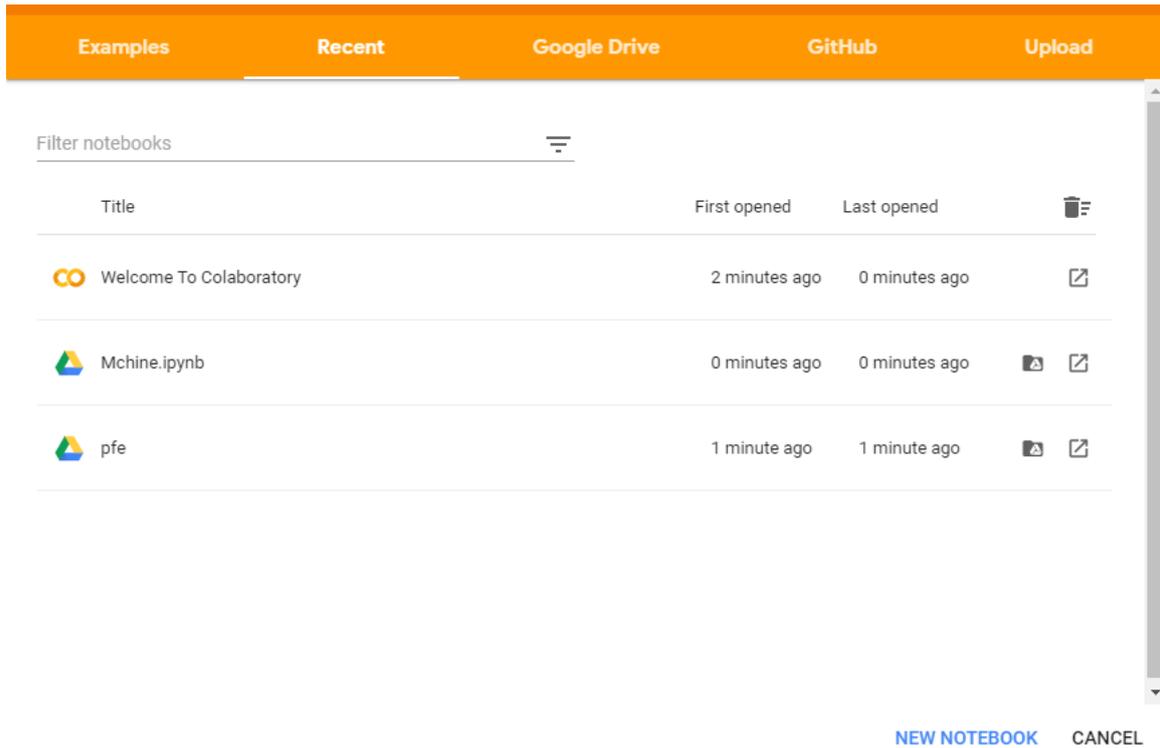


FIGURE 3.1 – Environnement de google cloud.

2. **Python** : Python est un langage de programmation de haut niveau conçu pour être facile à lire et simple à implémenter. Il est open source, ce qui signifie qu'il est gratuit, même pour des applications commerciales. Python peut fonctionner sur les systèmes Mac, Windows et Unix et a également été porté sur des machines virtuelles Java et .NET [29].

Python est considéré comme un langage de script, comme Ruby ou Perl et il est souvent utilisé pour créer des applications Web et du contenu Web dynamique. Des exemples d'applications prenant en charge une API Python incluent GIMP, Inkscape, Blender et Autodesk Maya.



FIGURE 3.2 – Logo de python et son environnement.

3. **Pycharm** : De nos jours, de nombreux programmeurs optent Python pour créer

des applications avec une base de code concise, propre et lisible. Ils peuvent même accélérer le développement d'applications logicielles personnalisées en tirant parti d'un certain nombre d'environnements de développement intégrés (IDE) pour Python [30].

Pour notre cas on a utilisé Pycharm pour créer une interface de notre système.



FIGURE 3.3 – Logo de Pycharm.

2.2 Bibliothèques utilisées :

1. **Scikit-learn** : Scikit-learn fournit une gamme d'algorithmes d'apprentissage supervisés et non supervisés via une interface cohérente en Python. Il est licencié sous une licence BSD simplifiée permissive et il est distribué sous de nombreuses distributions Linux, encourageant une utilisation académique et commerciale. La bibliothèque est basée sur SciPy (Scientific Python) qui doit être installé avant de pouvoir utiliser scikit-learn [31].
2. **Dlib** : dlib est une bibliothèque de logiciels multi-plateformes à usage général écrite dans le langage de programmation C ++. Nous allons apprendre dlib et comprendre comment trouver et utiliser les traits du visage humain. Selon son propre site Web, dlib.net, dlib est un outil C ++ moderne contient des algorithmes d'apprentissage automatique et des outils pour créer des logiciels complexes en C ++ pour résoudre des problèmes du monde réel [32].
3. **OpenCV** : OpenCV est une bibliothèque open source pour l'analyse des images et des vidéos, introduite à l'origine il y a plus de dix ans par Intel. Depuis lors, un certain nombre de programmeurs ont contribué aux développements les plus récents de la bibliothèque. Le dernier changement majeur a eu lieu en 2009 (OpenCV 2) qui comprend les principaux changements apportés à l'interface C++ [33].

2.3 Les bases de Données :

Les bases de données des images des expressions faciales que nous avons utilisé pour les tests sont : JAFFE et CK+ et la combinaison de ces deux bases JAFFE+CK.

1. **JAFFE** : La base de données des expressions faciales féminines japonaises (JAFFE) contient 213 échantillons d'expressions posées de 10 femmes japonaises. Chaque personne a 3 à 4 images avec chacune des six expressions faciales de base (colère, dégoût, peur, bonheur, tristesse et surprise) et une image avec une expression neutre.
2. **CK+** : C'est la base de données la plus largement utilisée pour évaluer les systèmes FER. Elle contient 593 séquences vidéo de 123 sujets. Les séquences varient en durée de 10 à 60 images et montrent un passage d'une expression faciale neutre à l'expression maximale. Les séquences varient en durée de 10 à 60 images et montrent un passage d'une expression faciale neutre à l'expression maximale. Parmi ces vidéos, 327 séquences de 118 sujets sont étiquetées avec sept étiquettes d'expression de base (colère, mépris, dégoût, peur, bonheur, tristesse et surprise) basées sur le Facial Action Coding System (FACS) [33].

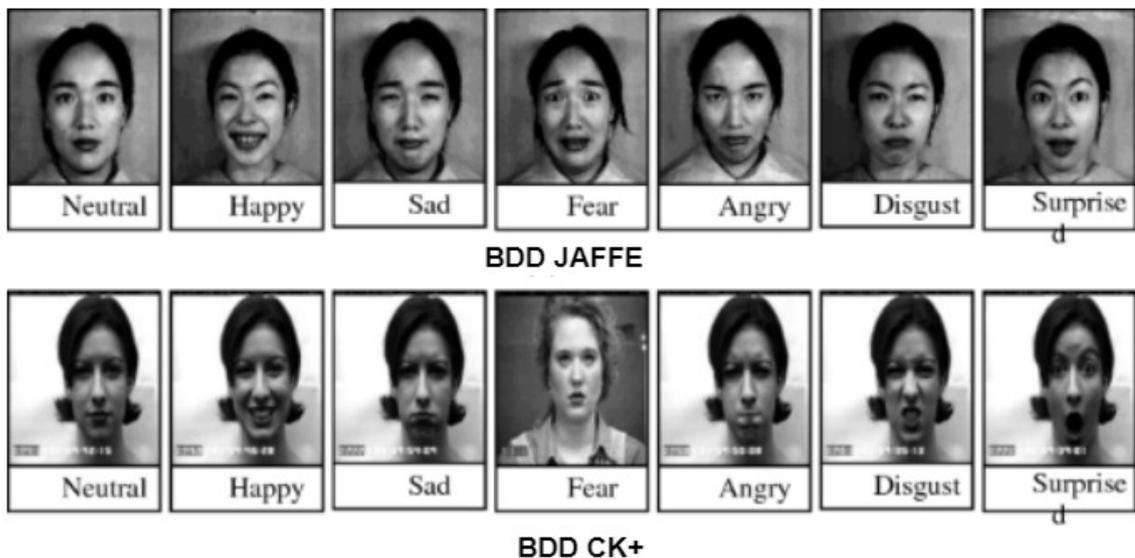


FIGURE 3.4 – Quelques images sur les bases de données JAFFE et CK+.

2.4 Implémentation et résultats :

Cette section décrit les expériences réalisées et les résultats expérimentaux de notre système sur la reconnaissance des expressions faciales. Elle contient deux parties pour examiner la performance de l'information.

- ✓ La première partie : le système est testé en utilisant une base de données JAFFE.
- ✓ La deuxième partie : le système est testé en utilisant une base de données CK+.
- ✓ La troisième partie : le système est testé en utilisant une base de données combinée de deux bases JAFFE et CK+.
- ✓ La quatrième partie : notre système est testé en utilisant des images externes afin de vérifier l'efficacité de notre système.

Le système proposé suit principalement les étapes suivantes :

- ✓ Faire une détection de visage en utilisant l'algorithme de Viola Jones.
- ✓ Détection des points de repères.
- ✓ Détection des régions.
- ✓ Appliquer une banque de filtres de Gabor sur chaque région.
- ✓ Optimiser les paramètres de Gabor par un algorithme génétique.
- ✓ Extraire le vecteur de caractéristiques avec en utilisant la méthode de Mean.
- ✓ En utilisant un classifieur SVM pour classifier l'émotion selon le vecteur de caractéristique.

3 Résultats expérimentaux :

Nous arrivons à la phase la plus importante du projet celle des tests dont l'objectif est de valider la conception proposée, sur notre interface on va tester des images extérieures. Et voilà le processus de notre projet :

On choisit la base de données avec laquelle on veut travailler (j'ai choisi : Jaffe) et on lance notre apprentissage et l'algorithme génétique commence ces tâches. Du coup le résultat s'affichera pour chacune des générations. On choisit le chromosome qui nous donne le meilleur résultat et on prend ses paramètres et on les met Dans un fichier XML.

Sur notre interface du système on importe une photo extérieure après on choisit la base de données avec laquelle on veut travailler (Jaffe).

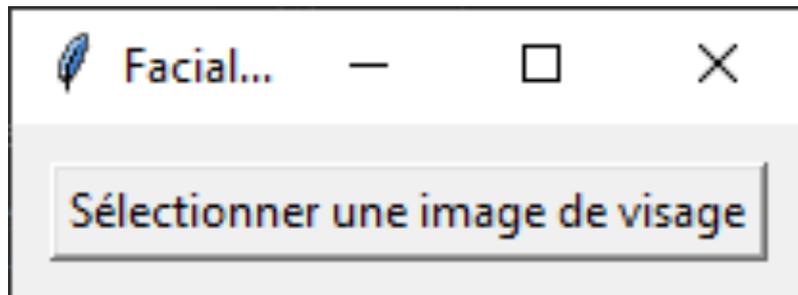


FIGURE 3.5 – Sélection d'une image d'un visage.

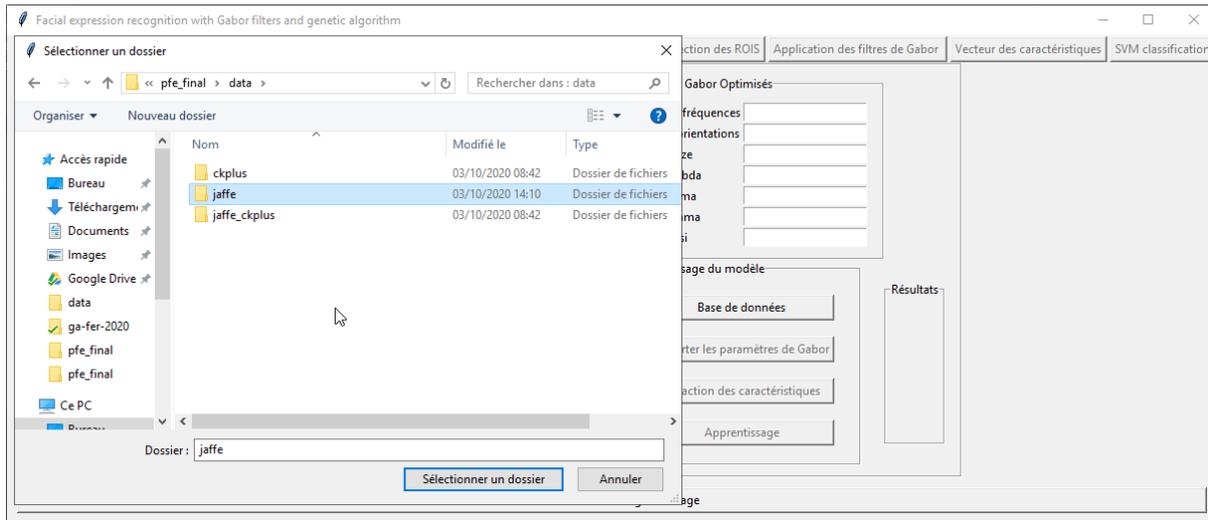


FIGURE 3.6 – Sélection de la base de données.

On importe le fichier xml qui inclut les meilleurs paramètres de Gabor.

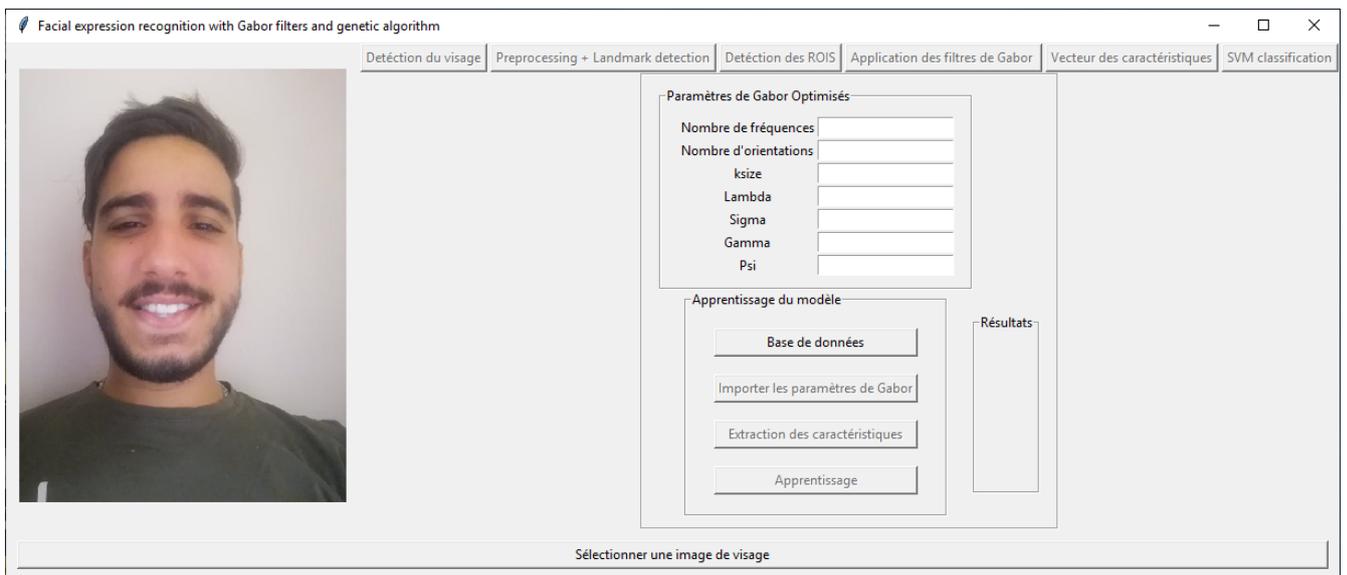


FIGURE 3.7 – L'interface initiale.

Après, on fait une extraction des caractéristique, où on exécute toutes les étapes comme ceci : détection de visage de toute la base de données, détection de tous les régions de chaque visage, application du filtre de Gabor pour chaque région avec les paramètres de Gabor sélectionnées dans le fichier xml, application de la méthode Mean pour chaque région pour avoir un vecteur de caractéristique, et on divise la base sur 80 pour cent pour l'apprentissage et 20 pour cent pour les tests.

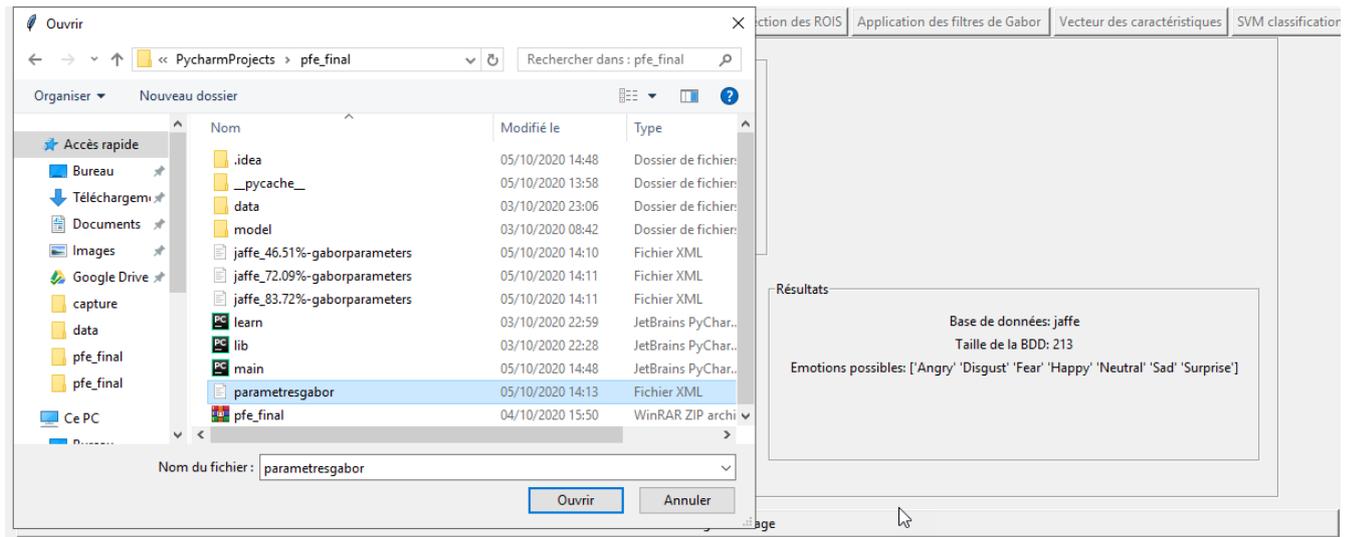


FIGURE 3.8 – L’importation des paramètres de Gabor.

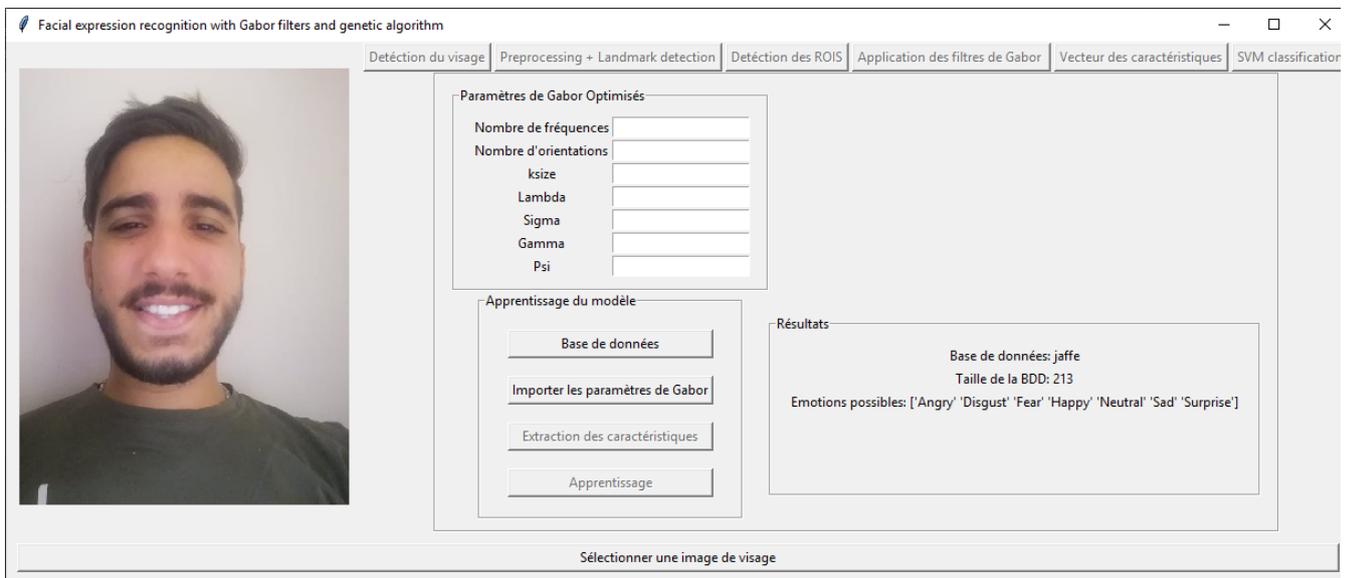


FIGURE 3.9 – L’extraction des caractéristique.

En fonction de la méthode de la GRID SEARCH on fait l’apprentissage pour avoir les meilleurs paramètres du classifieur SVM (C, Gamma, Kernel).

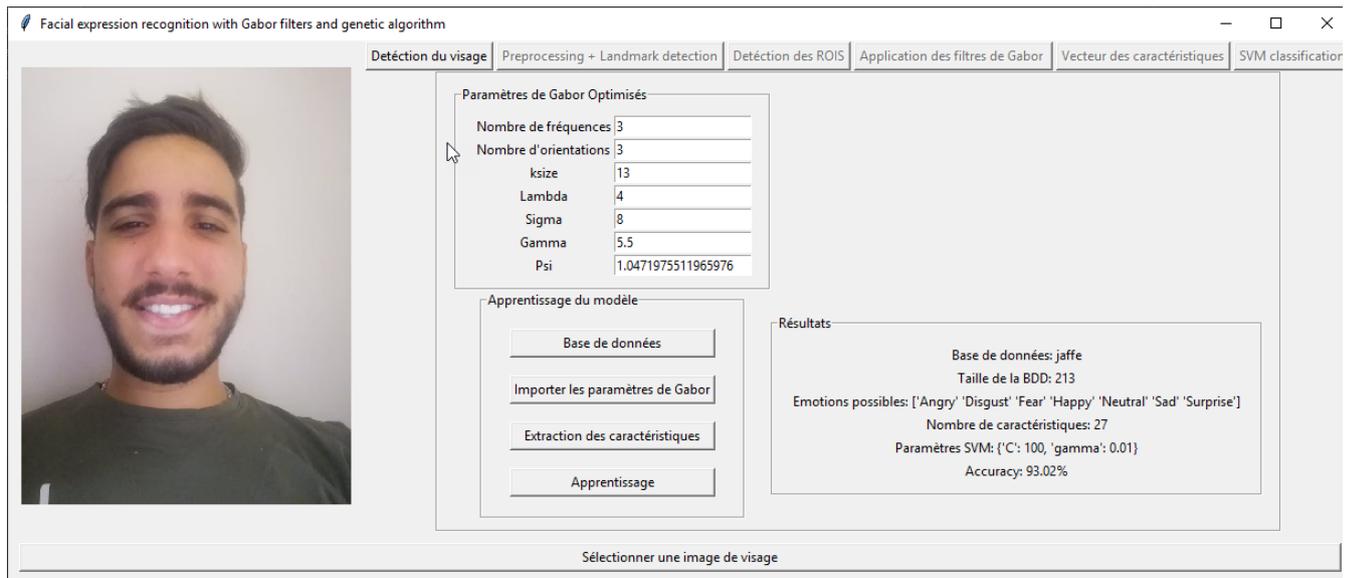


FIGURE 3.10 – Capture sur l’apprentissage.

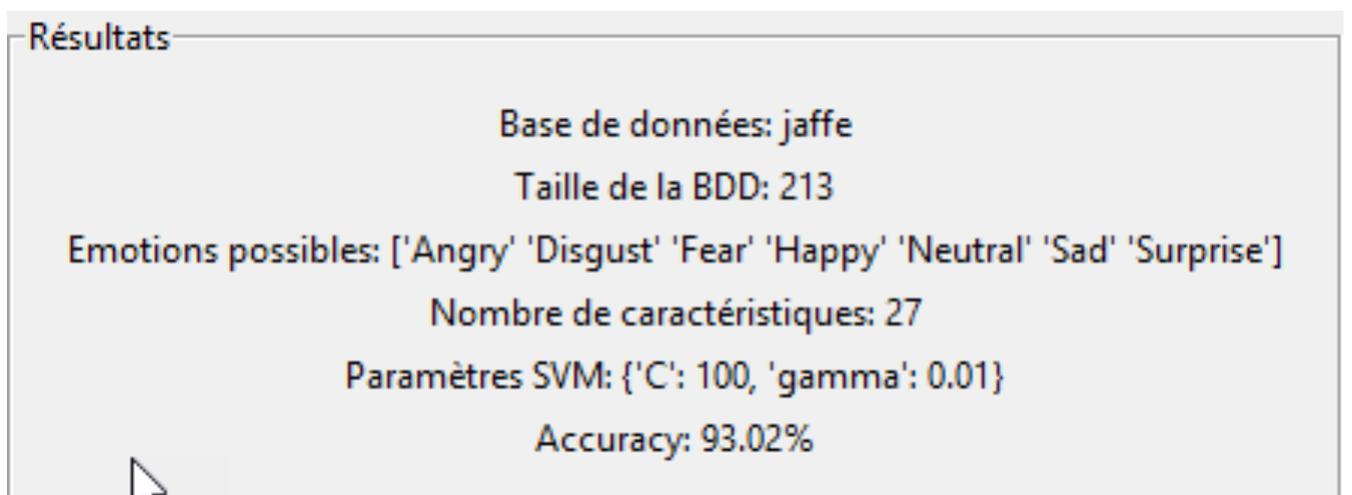


FIGURE 3.11 – Capture sur les meilleurs paramètres de SVM.

Puis, on fait la détection de visage sur l’image entrée.

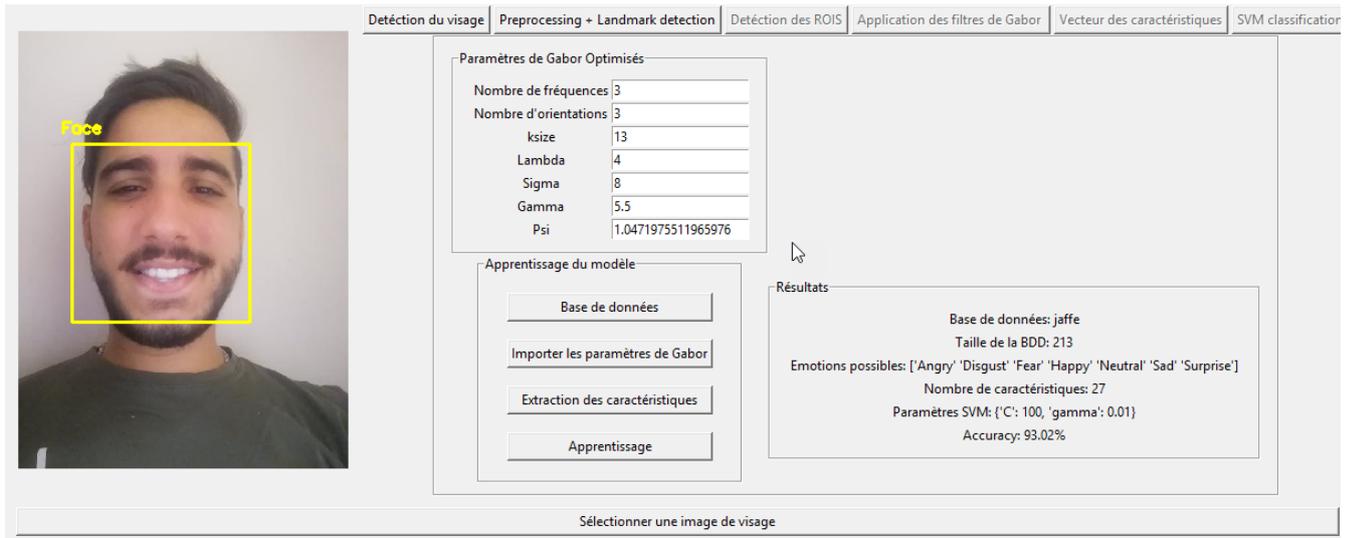


FIGURE 3.12 – La détection de visage.

Ensuite on fait une détection sur les points de repères.

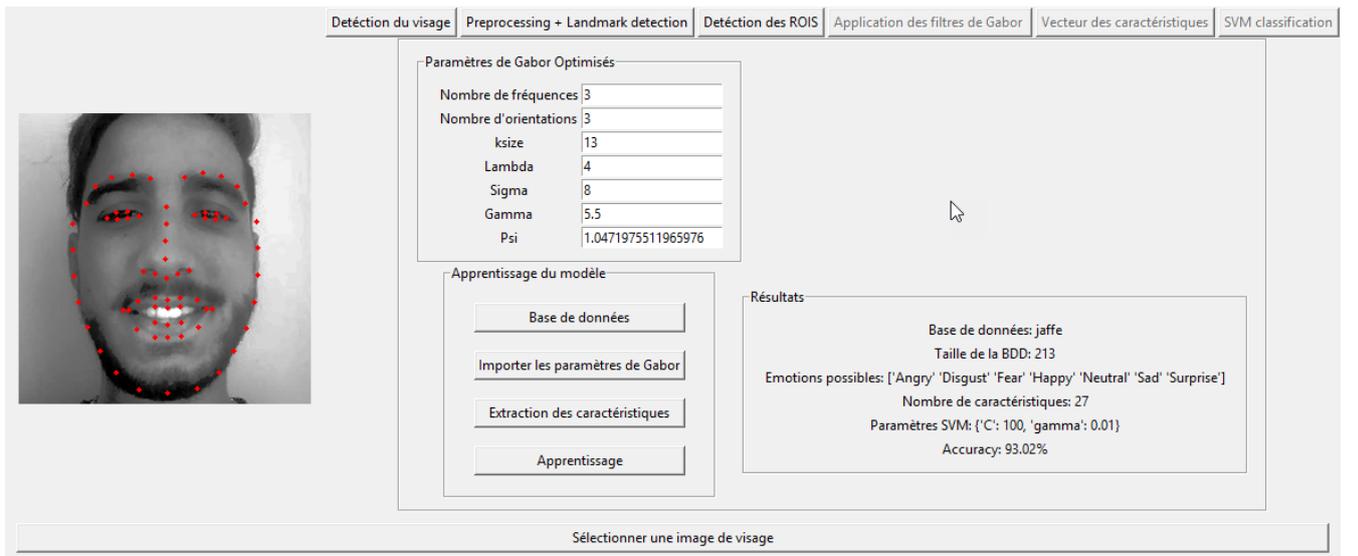


FIGURE 3.13 – Détection des points de repères.

Après on détecte les régions d'intérêt.
 Puis, on applique le filtre de Gabor avec les paramètres choisis dans le fichier xml sur les trois régions.

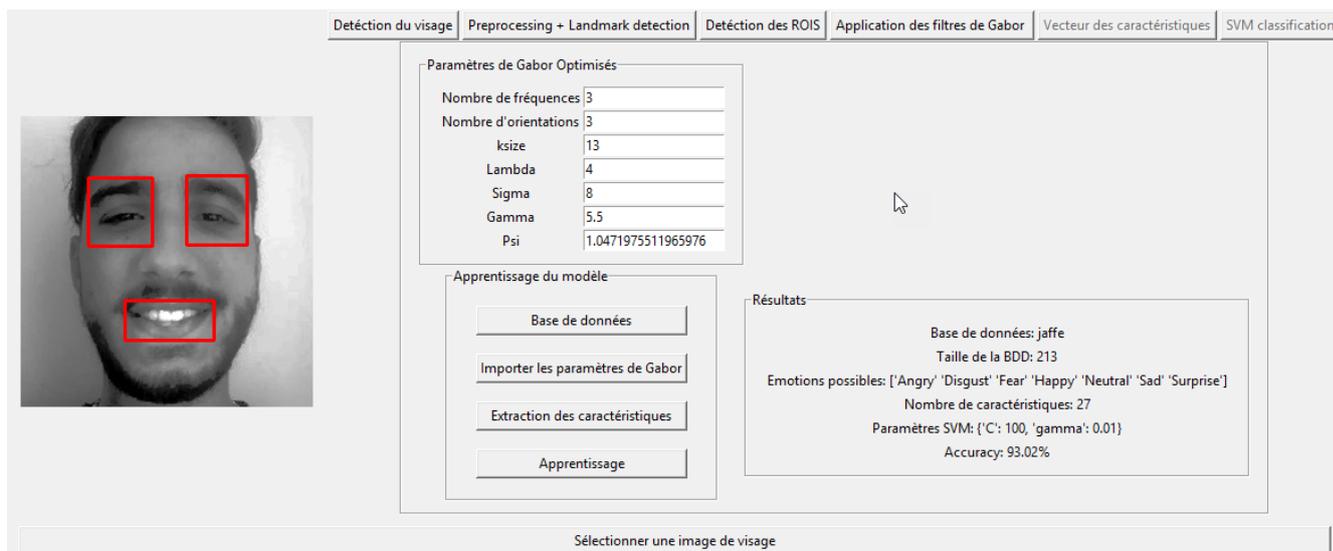


FIGURE 3.14 – Détection des régions d’intérêt.

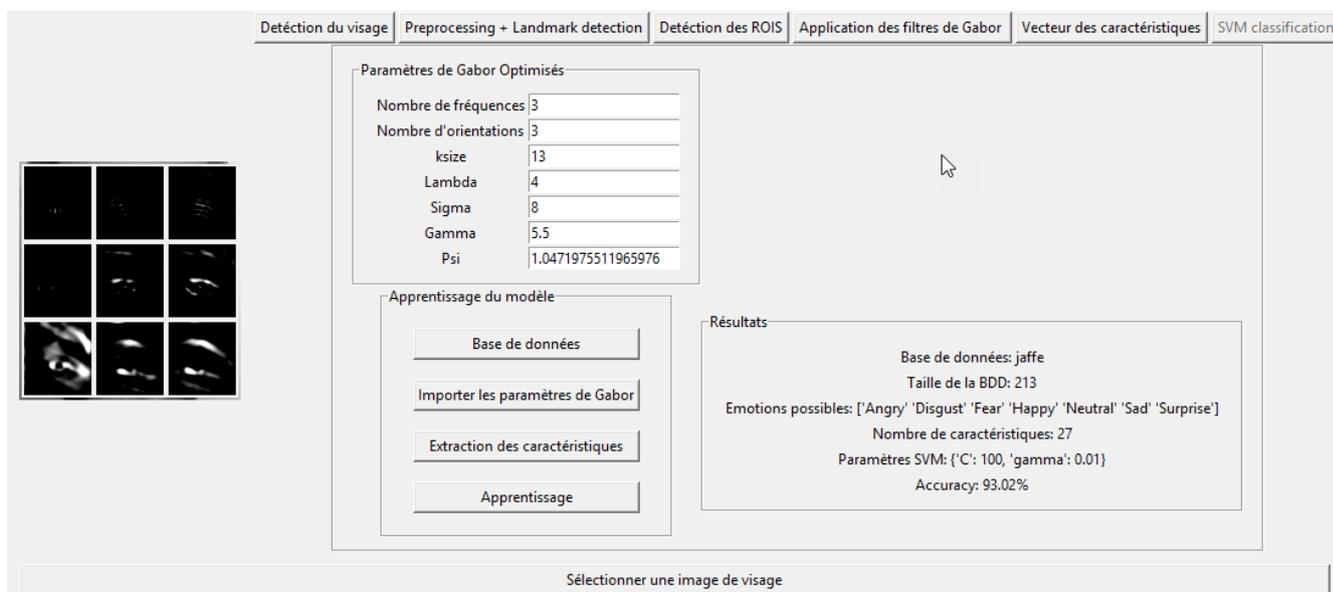


FIGURE 3.15 – L’application du filtre de Gabor sur les trois régions.

On obtient un vecteur de caractéristiques..

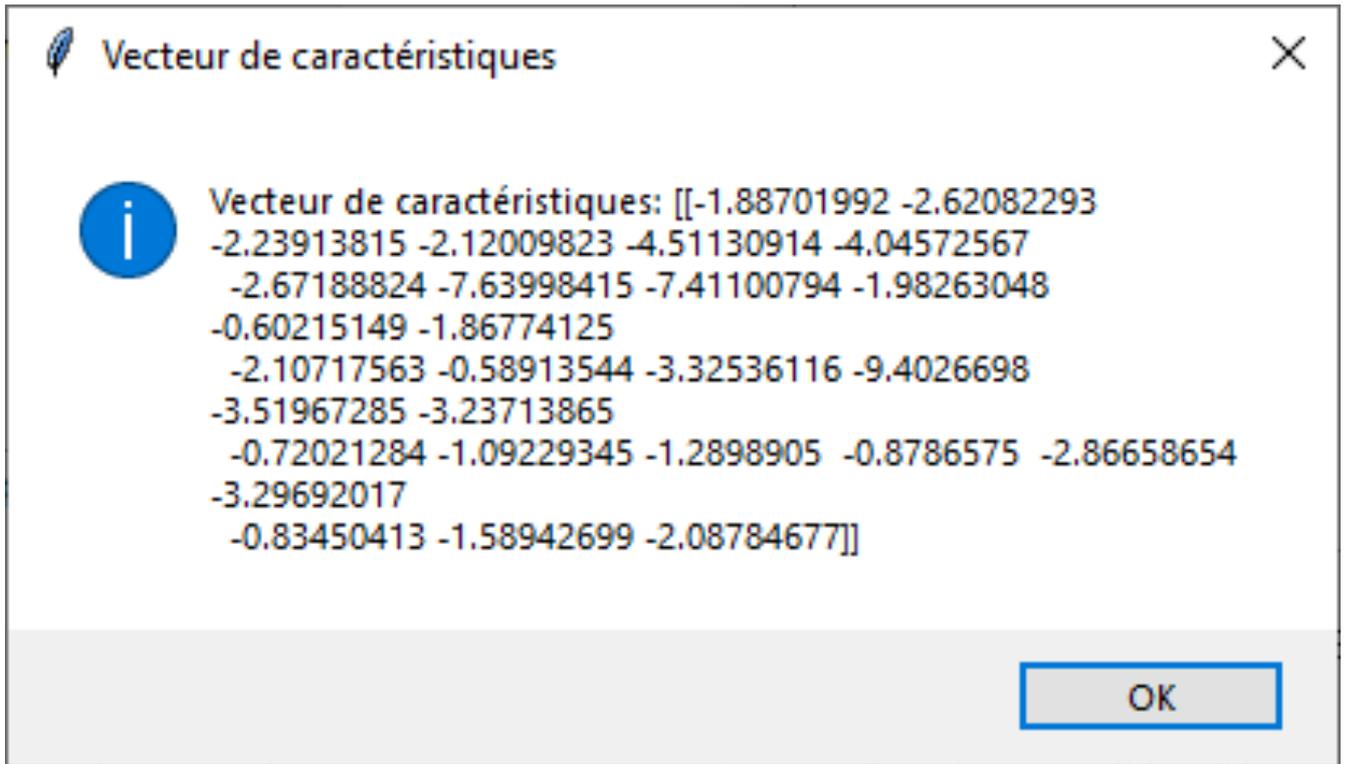


FIGURE 3.16 – Capture sur le vecteur de caractéristique.

Finalement on clique sur classification pour avoir le résultat qui est l’émotion du visage entrée.

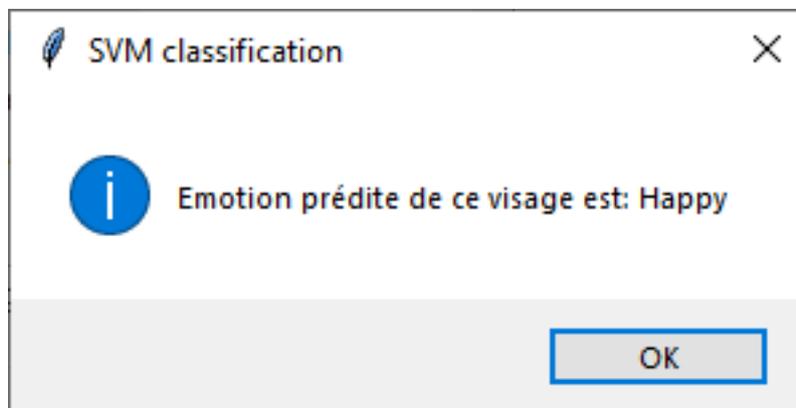


FIGURE 3.17 – Capture sur l’émotion du visage entrée.

Nous arrivons à la phase de quantification et valorisation des résultats de l’approche développée, nous se basant sur les trois bases de données et leur développement Au fur et à mesure que les générations progressent, nous se basant sur : le taux de reconnaissance (Accuracy).

Dans le tableau ci-dessus, une comparaison entre notre travail avec des travaux déjà existe.

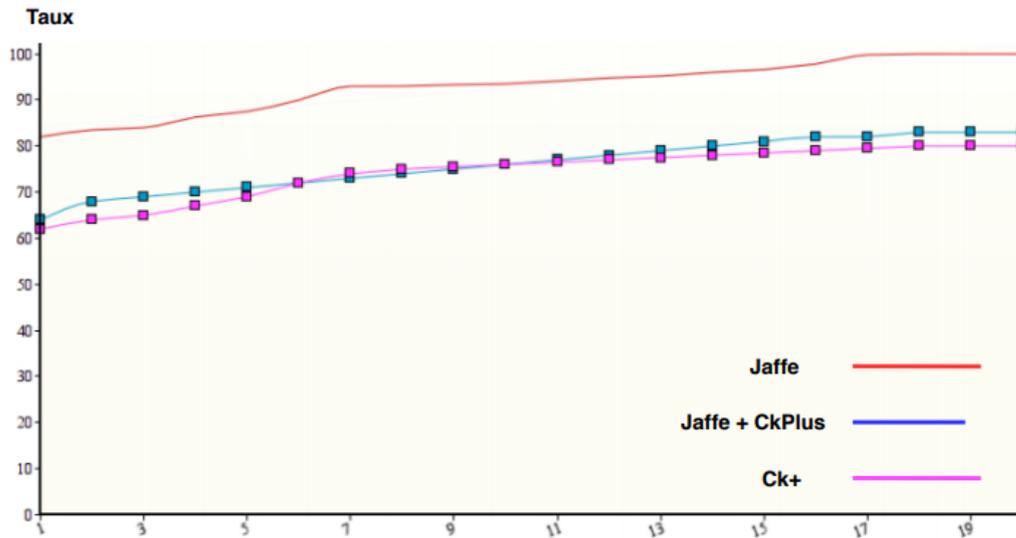


FIGURE 3.18 – Comparaison entre le développement des trois bases de données.

Réf	detection de visage	Caractéristiques	BDD images/vidéos	Algorithme pour optimisation	Algorithme pour classification	Résultats
[14]	Haar	Gabor	Cohn Kanade		KNN	96,67%
[12]	Viola jones	Réseau neuronal, Gabor	JAFFE		ANN	85,7%
[18]	LANDMAR	Gabor	CK+, SFEW, MMI	SVM, LIBSM		95.72%, 36.92%, 71.88%
Notre méthode	Viola Jones	Gabor	JAFFE, CK+, JAFFE+CK+	A.G	SVM	100%, 80%, 83%

TABLE 3.1 – Tableaux comparatif entre notre travail et d’autres travaux.

3.1 Discussion :

Les performances obtenues sont acceptables compte tenu la taille réduite de l’échantillon d’images de la BDD. Ceci s’explique par le fait que notre approche sera plus efficace est plus performante dans le cas de BDD de grandes tailles. Toute fois vu la taille de la base de données ceci est tout de même acceptable même si on aurait voulu avoir beaucoup plus.

Pour permettre la visualisation des performances du SVM le mieux appris (avec les bases de données JAFFE, CK+, JAFFE+CKPLUS), les figures ci-dessus montrent la matrice de confusion avec et sans normalisation par le nombre d’éléments dans chaque classe. L’utilisation de la matrice de confusion peut être intéressante en cas de déséquilibre de classe pour avoir une bonne interprétation visuelle de la classe mal classée. On observe

aucun visage JAFFE mal classé dans l'ensemble de test, par contre plusieurs visages mal classés dans les bases de données CK+ et JAFFE+CKplus.

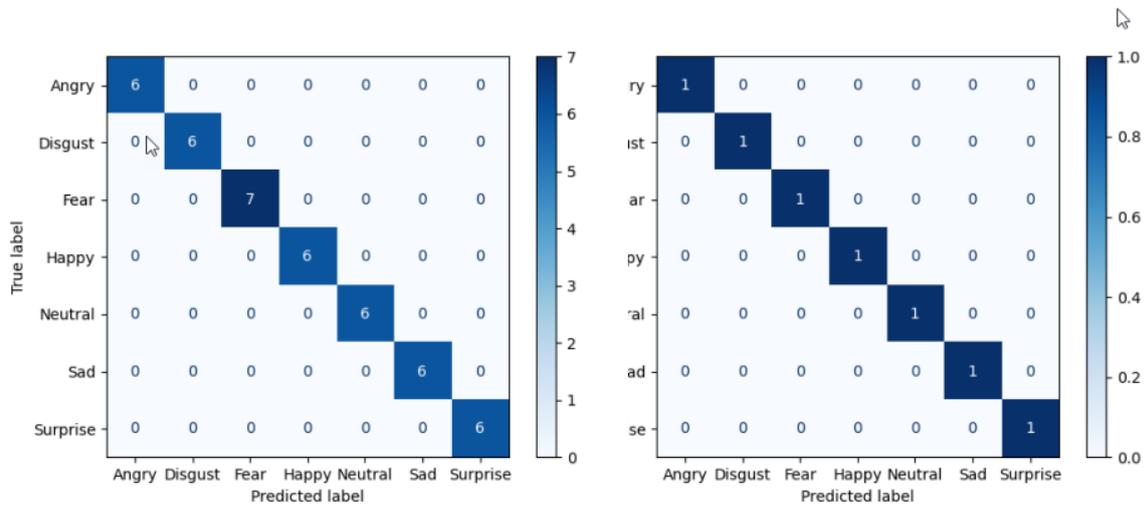


FIGURE 3.19 – Matrice de confusion avec et sans normalisation de la base de données JAFFE.

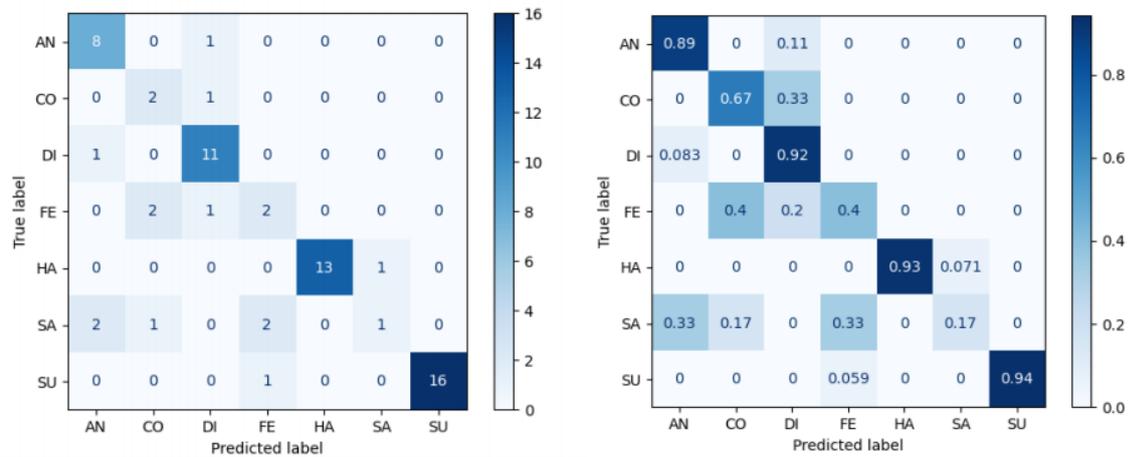


FIGURE 3.20 – Matrice de confusion avec et sans normalisation de la base de données CK+.

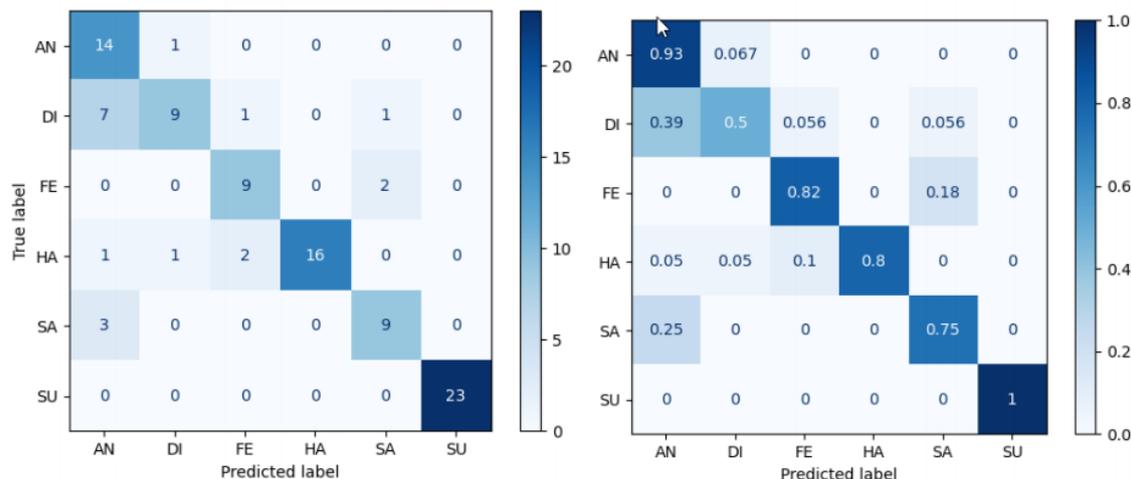


FIGURE 3.21 – Matrice de confusion avec et sans normalisation de la base de données JAFFE+CKplus.

3.2 Conclusion :

Nous avons présenté dans ce chapitre l’implémentation de l’approche de reconnaissance des expressions faciales basée sur les filtres de Gabor. Notre but majeur est d’optimiser l’efficacité de notre système en utilisant un algorithme génétique. Les résultats présentés sont très encourageants par rapport aux expériences dans le domaine.

Conclusion générale :

Dans notre mémoire, nous nous sommes intéressés au domaine de la machine Learning pour faire des études sur la détection des expressions faciales, et proposé un système pour détecter l'expression avec un bon résultat.

La méthode que nous avons eu a utilisé est l'utilisation de filtre de Gabor et un algorithme génétique pour optimiser les paramètres de ce filtre, le modèle obtenu a été tester sur deux bases de données qui sont Jaffe et Ck, qui ont nous a donné de très bons résultats. Et cette technologie ne manque que d'un matériel puissant (Supercalculateur), d'une très grande base de données.

A la fin, avant de passer aux perspectives, ce travail nous a permet d'apprendre beaucoup de choses et à explorer de nouvelles méthodes dans ce domaine de recherche et à lire des bons articles de recherche dans ce domaine .

Perspective :

Comme perspectives de recherches futures, nous envisageons de :

1. De tester sur notre modèle des grandes bases de données pour éviter le problème de overfitting.
2. essayé de mettre même les paramètres de SVM dans l'algorithme génétique.
3. tester notre modèle sur des super ordinateurs pour le réajuster d'une meilleure manière, car pour tester notre système et attendre ces résultats prend beaucoup de temps. Et ces supers ordinateurs vont accélère les traitement et l'apprentissage.

Webographie

[W1] : <https://www.google.com/search?q=texoq=texaqs=chrome..69i57j0l3j46j0j69i60l2.1399j1j7s8>

[W2] : <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3>

[W3] : https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_introduction.htm

[W4] : <https://blog.floydhub.com/introduction-to-genetic-algorithms/>

Bibliographie

- [1] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1) :71–86, 1991.
- [2] Pawel Tarnowski, Marcin Kolodziej, Andrzej Majkowski, and Remigiusz J Rak. Emotion recognition using facial expressions. In *ICCS*, pages 1175–1184, 2017.
- [3] Nene Adama Dian DIALLO. La reconnaissance des expressions faciales. 2019.
- [4] Andrew J Calder, Jill Keane, Tom Manly, Reiner Sprengelmeyer, Sophie Scott, Ian Nimmo-Smith, and Andrew W Young. Facial expression recognition across the adult life span. *Neuropsychologia*, 41(2) :195–202, 2003.
- [5] Curtis Padgett and Garrison W Cottrell. Representing face images for emotion classification. In *Advances in neural information processing systems*, pages 894–900, 1997.
- [6] WeiFeng Liu and ZengFu Wang. Facial expression recognition based on fusion of multiple gabor features. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 536–539. IEEE, 2006.
- [7] G Hemalatha and CP Sumathi. A study of techniques for facial detection and expression classification. *International Journal of Computer Science and Engineering Survey*, 5(2) :27, 2014.
- [8] Siti Khairuni Amalina Kamarol, Mohamed Hisham Jaward, Jussi Parkkinen, and Rajendran Parthiban. Spatiotemporal feature extraction for facial expression recognition. *IET Image Processing*, 10(7) :534–541, 2016.
- [9] Jamal Hussain Shah, Muhammad Sharif, Mussarat Yasmin, and Steven Lawrence Fernandes. Facial expressions classification and false label reduction using lda and threefold svm. *Pattern Recognition Letters*, 2017.
- [10] Dhikra Bouzit. Reconnaissance de visage basée sur une approche triangulaire. 2019.
- [11] Hung-Hsu Tsai and Yi-Cheng Chang. Facial expression recognition using a combination of multiple facial features and support vector machine. *Soft Computing*, 22(13) :4389–4405, 2018.
- [12] Kunika Verma and Ajay Khunteta. Facial expression recognition using gabor filter and multi-layer artificial neural network. In *2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC)*, pages 1–5. IEEE, 2017.
- [13] Taye Oladele Aro, Bamidele A Oluwade, Oluwakemi Christiana Abikoye, and Amos Orenyi Bajeh. A 2-dimensional gabor-filters for face recognition system : A survey. 2017.
- [14] Ramachandran Vedantham, Lavanya Settipalli, and E Sreenivasa Reddy. Real time facial expression recognition in video using nearest neighbor classifier. *International Journal of Pure and Applied Mathematics*, 118(9) :849–854, 2018.

- [15] Jun Zhou, Sue Zhang, Hongyan Mei, and Dawei Wang. A method of facial expression recognition based on gabor and nmf. *Pattern Recognition and Image Analysis*, 26(1) :119–124, 2016.
- [16] Archana Kumari Sharma, Umesh Kumar, Sandeep K Gupta, Uma Sharma, and Shubh LakshmiAgrwal. A survey on feature extraction technique for facial expression recognition system. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pages 1–6. IEEE, 2018.
- [17] I Michael Revina and WR Sam Emmanuel. Face expression recognition with the optimization based multi-svnn classifier and the modified ldp features. *Journal of Visual Communication and Image Representation*, 62 :43–55, 2019.
- [18] Hamid Sadeghi and Abolghasem-A Raie. Human vision inspired feature extraction for facial expression recognition. *Multimedia Tools and Applications*, 78(21) :30335–30353, 2019.
- [19] SN Sivanandam and SN Deepa. Genetic algorithm optimization problems. In *Introduction to genetic algorithms*, pages 165–209. Springer, 2008.
- [20] Lawrence Davis. *Handbook of genetic algorithms*. 1991.
- [21] AJ Sobey and PA Grudniewski. Re-inspiring the genetic algorithm with multi-level selection theory : multi-level selection genetic algorithm. *Bioinspiration & biomimetics*, 13(5) :056007, 2018.
- [22] Michael E Mavroforakis and Sergios Theodoridis. A geometric approach to support vector machine (svm) classification. *IEEE transactions on neural networks*, 17(3) :671–682, 2006.
- [23] Yi-Qing Wang. An analysis of the viola-jones face detection algorithm. *Image Processing On Line*, 4 :128–148, 2014.
- [24] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *European conference on computer vision*, pages 94–108. Springer, 2014.
- [25] David C Howell. Median absolute deviation. *Encyclopedia of statistics in behavioral science*, 2005.
- [26] Frank Y Shih, Chao-Fa Chuang, and Patrick SP Wang. Performance comparisons of facial expression recognition in jaffe database. *International Journal of Pattern Recognition and Artificial Intelligence*, 22(03) :445–459, 2008.
- [27] QiuJun Huang, Jingli Mao, and Yong Liu. An improved grid search algorithm of svr parameters optimization. In *2012 IEEE 14th International Conference on Communication Technology*, pages 1022–1026. IEEE, 2012.
- [28] Umberto Michelucci. Introduction and development environment setup. In *Advanced Applied Deep Learning*, pages 1–26. Springer, 2019.
- [29] Andreas C Müller, Sarah Guido, et al. *Introduction to machine learning with Python : a guide for data scientists*. " O'Reilly Media, Inc. ", 2016.
- [30] YASH A AKBARI. Python programming.
- [31] Ekaba Bisong. Introduction to scikit-learn. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pages 215–229. Springer, 2019.
- [32] Joeran Beel, Bela Gipp, Stefan Langer, Marcel Genzmehr, Erik Wilde, Andreas Nürnberger, and Jim Pitman. Introducing mr. dlib, a machine-readable digital library. In

Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries, pages 463–464, 2011.

- [33] Ivan Culjak, David Abram, Tomislav Pribanic, Hrvoje Dzapov, and Mario Cifrek. A brief introduction to opencv. In *2012 proceedings of the 35th international convention MIPRO*, pages 1725–1730, 2012.