

MA - 004 - 165 - 1

UNIVERSITE DE SAAD DAHLEB BLIDA



Faculté des sciences

Département d'Informatique

Mémoire présenté par : - *Bourada Abdel Amine*
- *Bachene Adel*

En Vue d'obtenir le diplôme de Master

Domaine : Mathématique et Informatique.

Filière : Informatique

Spécialité : Ingénierie Logiciel

Promoteur : Mohamed RédhaSidoumou

Encadreur : Dr Djamel Gaceb

***Reconnaissance des émotions à partir
d'expressions faciales capturées via
webcam***

Président : Chikhi N

Examineur: Aroussi

Examineur : Rezoug

MA-004-165-1

Dédicace

Je dédie ce mémoire :

Pour mes parents

En témoignage de ma reconnaissance envers le soutien, les sacrifices et tous les efforts qu'ils ont fait pour mon éducation ainsi que ma formation.

Pour mes frères et sœurs

Pour leurs affections, compréhension et patience.

Pour mes amis

Bachene Adel

Mohamed Sahed

Ayadi Ramzy

Benaouda Khalil

Billal Alam

Pour le soutien, et pour être là quand j'ai besoin de vous, et pour tout les demeurant qui m'ont soutenu d'une manière ou d'une autre dans la réalisation de ce travail. Je ne vous oublierai jamais.

Pour mon promoteur

Mohamed Redah Sidoumou

Pour l'aide qu'on a reçu de sa part pour, pour sa disponibilité.

Merci

Amine

Dédicace

Je dédie ce mémoire :

Pour mes parents

En témoignage de ma reconnaissance envers le soutien, les sacrifices et tous les efforts qu'ils ont fait pour mon éducation ainsi que ma formation, À celui qui m'a indiqué la bonne voie en me rappelant que la volonté fait toujours les grands hommes

Pour mes frères

Pour leurs affections, compréhension et patience.

Pour mes Amis

Bourada Amine

Mohamed Sahed

Ayadi Ramzy

Mami Mohamed Nadjib

Mazari Oussama

Kortbi Mouna

Pour le soutien, et pour être là quand j'ai besoin de vous, et pour tout les demeurant qui m'ont soutenu d'une manière ou d'une autre dans la réalisation de ce travail. Je ne vous oublierai jamais.

Pour mon promoteur

Mohamed Redah Sidoumou

Pour l'aide qu'on a reçu de sa part pour, pour sa disponibilité.

Merci

Adel

Remerciement :

On remercie tout d'abord notre bon Dieu, le socle et le ressort de toutes nos actions, l'auteur de la réalisation de nos rêves, et de notre mission sur terre. Celui qui est toujours là, lorsque tout semble nous quitter. Dans les douleurs et les pleurs les plus internes, secrètes, sincères et profondes, Toi Allah tu étais toujours présent et plein de ton immense Miséricorde. Aucun remerciement n'est trop grand ni suffisant pour Toi, et merci de nous avoir permis d'arriver à ce niveau-ci dans notre vie. On ne peut que compter sur toi pour continuer le reste de la réalisation de nos rêves, et de notre mission que Tu nous as confié, parce que même si les hommes nous abandonnent, on sait que Toi, tu ne nous abandonneras jamais (الحمد لله). Car notre vie, c'est pour Toi.

La présente étude n'aurait pas été possible sans le bienveillant soutien de certaines personnes. Et on n'est pas non plus capables de dire dans les mots qui conviennent, le rôle qu'elles ont pu jouer à notre côté pour en arriver là. Cependant, on voudrait les prier d'accueillir ici tous nos sentiments de gratitude et de remerciements qui viennent du fond de nos cœurs.

Notre remerciement principal va d'abord à notre promoteur le Professeur Mohamed Redah Sidoumou et notre encadreur Docteur Djamel Gaceb, qui nous ont accompagnés tout au long du projet. Leurs disponibilités et leurs généreux aides au cours de certains de nos moments difficiles ont été d'un très grand apport, et d'un immense réconfort. Alors encore merci Monsieur Mohamed Redah Sidoumou et Monsieur Djamel Gaceb.

Sans oublier de remercier aussi tous nos enseignants qui, à chaque étape de notre cursus de formation, nous ont prodigué d'utiles conseils qui nous ont aidé à bien structurer nos études. Nos remerciements vont aussi à tous nos collègues de la

faculté des Sciences qui, d'une manière ou d'une autre ont joué un rôle non négligeable dans l'accomplissement de cette œuvre.

Et on adresse aussi nos remerciements à monsieur Bachene Mohamed qui nous a aidé en corrigeant notre mémoire et en plus de son support en tant que père et par son accueil chaleureux. Merci infiniment monsieur Bachene Mohamed.

Et on remercie chaque membre de nos familles respectives Bourada et Bachene, pour leur présence, leur soutien, et pour les moments vécus avec eux, merci.

Nos remerciements vont aussi à tous nos amis, Ayadi Ramzy, Mohamed Sahed, Benaouda Khalil, Billal Alam et ainsi tous les membres du Club CSCC...etc. Vous nous avez tous soutenu par votre encouragement, et pour les moments inoubliables vécu avec vous et qu'on espère qu'ils ne s'arrêteront jamais. Au demeurant, que tous ceux dont les noms ne figurent pas sur cette liste et qui nous ont soutenu d'une manière ou d'une autre sachent que leur apport spirituel et matériel n'a pas été vain. Je vous adresse tous nos sentiments de reconnaissance renouvelée. Comment peut-on échouer avec une telle armée derrière nous ?

Tables des Matières :

<i>Tables des matières</i>	5
<i>Intoduction général</i>	10
<i>Problématiques et Objectifs</i>	12
<i>Chapitre 1 : psychologie des émotions</i>	14
1. <i>Introduction</i>	14
2. <i>Qu'ce qu'une émotion ?</i>	14
2.1. <i>Définition consensuelle de l'émotion</i>	15
3. <i>Classification des émotions</i>	16
3.1. <i>Emotions positives et négatives</i>	16
3.2. <i>L'approche dimensionnelle des émotions</i>	16
4. <i>Les émotions de base, émotions universelles</i>	17
5. <i>Qu'est-ce qu'une expression faciale ?</i>	17
6. <i>Conclusion</i>	18
<i>Chapitre 2 : Détection Automatique du visage</i>	19
1. <i>Principe de détection du visage</i>	19
2. <i>Comparaison des algorithmes de détection du visage</i>	20
2.1. <i>Brève description des algorithmes disponibles dans chaque librairie.</i>	20
2.1.1. <i>La librairie SIF (SIFinder)</i>	20
2.1.2. <i>La librairie FDLib (Face Detection Library)</i>	21
2.1.3. <i>FaceOnIt (FOI)</i>	21
2.1.4. <i>FaceSDK (FSDK) et VeriLook (VL)</i>	22
2.1.5. <i>La librairie OpenCV</i>	22
2.1.5.1. <i>La méthode de fonctionnement</i>	22
2.1.5.2. <i>Principe</i>	22
2.1.5.3. <i>Principe de détection par OpenCV</i>	23
2.1.5.4. <i>Comment la détection des composants du visage</i>	

fonctionne dans la méthode utilisée ?	23
3. Discussion	27
4. Conclusion	28
Chapitre 3 : Modèle Actif De Forme	29
1. Introduction	29
2. Forme	29
3. Alignement de forme	29
4. Fonctionnement d'ASM	29
4.1.Modèle de forme (Shape Model)	31
4.1.1. Générer la forme depuis le modèle de forme	32
4.2.Modèle de profile	33
4.2.1. Formé un profile	33
4.2.2. Construire un modèle de profile durant l'apprentissage	34
4.2.3. Rechercher le meilleur profile	34
5. Conclusion	35
Chapitre 4 : Classification	36
1. Introduction	36
2. Dara Mining	37
2.1.Définition	37
2.2.Les taches du Data Mining	37
a) Classification	38
b) Clustering (Segmentation)	38
c) L'estimation	38
d) Prédiction	39
2.3.Les techniques du Data Mining	39
1) K plus proche voisins	39
a) Principe	39
b) Etapes	39
c) L'Algorithme	40
d) Avantage	40
e) Inconvénients	40
2) Les réseaux de neurones	41

a)	<i>La structure</i>	41
b)	<i>Structure d'interconnexion</i>	42
c)	<i>Apprentissage</i>	43
d)	<i>Avantage</i>	44
e)	<i>Inconvénients</i>	44
3)	<i>K-means</i>	44
a)	<i>Algorithme</i>	44
b)	<i>Avantage</i>	45
c)	<i>Inconvénients</i>	45
4)	<i>Machine à vecteur de support</i>	45
a)	<i>Les données sont linéairement séparable</i>	46
b)	<i>Données non linéairement séparable</i>	46
	- <i>Kernel Trick</i>	47
	- <i>La formulation de Kernel en SVM</i>	47
c)	<i>Cas de multi-classes</i>	48
d)	<i>Avantage</i>	48
e)	<i>Inconvénients</i>	48
3.	<i>Etat de l'art</i>	49
4.	<i>Conclusion</i>	50
	Chapitre 5 : les outils de détection des points caractéristiques	51
1.	<i>Introduction</i>	51
2.	<i>Détection du visage</i>	51
1)	<i>OpenCv</i>	51
a)	<i>Charger le détecteur de Haar pour la détection du visage</i>	52
b)	<i>Accéder à la webcam</i>	52
c)	<i>Traitement d'image</i>	52
d)	<i>Détecter le visage</i>	53
3.	<i>Détection des points caractéristiques du visage</i>	54
1)	<i>Le modèle de profile</i>	54
2)	<i>Le modèle de forme</i>	55
	Chapitre 6 : Nos démarches de reconnaissance	57
1.	<i>Introduction</i>	57

2. Les paramètres en entrées	57
2.1. Les positions des points caractéristiques	57
2.1.1. Teste de sélection des points caractéristiques	60
2.1.1.1. Implémentation de l'algorithme de teste	60
2.1.1.2. Les résultats des testes	62
2.2. Les relations spatiales entre les points caractéristiques	63
2.2.1. Méthode de calcul des relations spatiales	64
2.2.2. Teste de sélection des relations spatiales	65
2.2.2.1. Implémentation de l'algorithme de teste	65
2.2.2.2. Les résultats des testes	66
3. Nos démarches de classification	68
3.1. Base d'image pour les apprentissages et les testes	68
3.2. K-means	68
3.2.1. K-means avec les coordonnées des 33 points caractéristiques	69
• Résultats des testes	71
3.2.2. K-means avec les relations spatiales entre les points caractéristiques	72
• Résultats des testes	75
3.2.3. Comparaison des deux méthodes (relations spatiales, coordonnées des points caractéristiques)	76
3.3. Machine à vecteur de support	76
3.3.1. LIBSVM	76
3.3.2. Nos choix de paramètres pour la classification des expressions faciales	77
3.3.3. Etapes d'application du SVM	77
3.3.4. Application du SVM sur les coordonnées des points caractéristiques	78
3.3.4.1. Implémentation du SVM avec les points caractéristiques	79
3.3.4.2. Testes et résultats	80
3.3.5. Application de SVM avec relations spatiales	80

3.3.5.1.	<i>Implémentation de SVM avec les relations spatiales</i>	81
3.3.5.2.	<i>Testes et résultats</i>	82
3.3.6.	<i>Comparaison des résultats des deux méthodes : relations spatiales et coordonnées des points caractéristiques</i>	83
4.	<i>Comparaison des résultats de K-means et de SVM</i>	83
5.	<i>Conclusion</i>	83
Chapitre 7 :	<i>Implémentation</i>	85
1.	<i>Introduction</i>	85
2.	<i>Phase codage</i>	85
2.1.	<i>Langage et outils de développement</i>	85
2.1.1.	<i>C++</i>	85
•	<i>Les caractéristiques de C++</i>	85
2.1.2.	<i>Visual Studio</i>	86
2.1.3.	<i>Librairie OpenCV</i>	86
2.1.4.	<i>STASM</i>	86
3.	<i>Phase D'implémentation de la meilleure méthode avec la Webcam</i>	86
4.	<i>Conclusion</i>	87
	<i>Conclusion général</i>	88
	<i>Références Bibliographiques</i>	89
	<i>Résumé</i>	95
	<i>Summary</i>	96

Introduction générale:

L'évolution des systèmes informatiques a été très rapide ces dernières années. Cette évolution est notamment perceptible à travers l'interaction homme-machine (IHM). Cette dernière a été longtemps basée sur l'usage du triplet clavier-écran-sourie comme moyen de communication dans ce domaine. Aujourd'hui, de nouveaux paradigmes sont considérés offrant à l'utilisateur une évolution continue dans son environnement naturel : les doigts, la main, le visage ou les objets familiers sont envisagés comme des dispositifs d'entrée/sortie. La capture du comportement observable de l'utilisateur en échange avec sa machine et son environnement est devenue un élément essentiel pour ces nouvelles formes d'interaction. Une telle exigence s'appuie fortement sur les techniques les plus avancées en perception artificielle, notamment en vision par ordinateur et reconnaissance de formes.

Les nouvelles générations d'environnement IHM sont devenues multimodales en intégrant plusieurs moyens de communication avec la machine (clavier, souris, Webcam, écran tactile). [A1]

Le visage représente la partie la plus expressive et communicative chez l'homme, c'est la raison pour laquelle il représente un centre d'intérêt majeur dans les recherches en développement de nouvelles IHM et il est considéré comme un moyen très efficace pour révolutionner la communication entre l'homme et la machine. Les expressions faciales traduisent efficacement et très clairement l'état d'esprit de l'utilisateur. Des études en psychologie ont démontré que les expressions faciales jouent un rôle important dans la conversation humaine. L'information essentielle caractérisant une expression faciale peut être perceptible à partir des différentes déformations ou changements sur les principales parties du visage (yeux, bouche, nez, sourcille). Ceci justifie pourquoi l'analyse par ordinateur du visage ou de ses expressions est devenue un domaine en forte émergence. Les applications qui en découlent sont nombreuses :

- Interaction Homme-machine.
- Dans le domaine linguistique, des sciences comportementales, de la psychologie ou de la langue des signes.
- Les animations.

Pour que la machine adopte les modes de communications humains, des avancées ont été faites dans le domaine et il existe des machines qui obéissent aux « doigt, expression faciale, mouvement des mains...etc.).

Notre travail rentre dans ce cadre d'application de reconnaissance des expressions faciales par analyse des séquences vidéo en provenance d'une webcam et image fixe. Nous avons proposé une étude comparative entre deux méthodes de perception des changements faciales : le changement des coordonnées des points caractéristiques du visage et les relations spatiales entre les points caractéristiques qui changent lors d'une expression. Et deux méthodes de classifications ont été testées : une naïve et simple le K-means et l'autre efficace et fiable la machine à vecteur de support SVM.

Nos démarches d'étude comparative vont aider à choisir la meilleure méthode de perception des changements faciaux combinés avec la meilleure méthode de classification dans le but de créer une application de reconnaissance des expressions faciales fiables et efficaces.

Ce mémoire est structuré en sept chapitres. Le premier chapitre présente l'aspect psychologique de l'émotion où on définit les émotions, les deux approches pour classifier les émotions et finalement comment exprimer les émotions et spécialement par expressions faciales. Par la suite, le second chapitre présente les méthodes de détection automatique du visage et les différentes caractéristiques faciales ainsi que le choix des méthodes à utiliser. Dans le chapitre trois on présente le modèle actif de formes sur lequel nous nous sommes basée pour détecter des points caractéristiques du visage. Le quatrième chapitre présente les différentes méthodes de classification. Le cinquième chapitre a été consacré au descriptif des outils utilisés pour la détection du visage et des points caractéristiques. le sixième chapitre présente nos différentes démarches de reconnaissance des expressions faciales et les différents testes appliqués pour présenter les méthodes les plus adaptées pour la reconnaissance des expressions faciales, Finalement, le septième chapitre où on présente les outils utiliser dans l'implémentation et les différents étapes d'implémentations et de testes.

Problématique et Objectif:

L'analyse des visages et des expressions faciales est toujours un domaine de recherche très actif, il concerne plusieurs thématiques de recherche applicative dans l'optique de reconnaître l'état émotionnel de l'utilisateur en Interaction Homme-Machine. Cela est dans le but de pouvoir concevoir des interfaces plus ergonomiques et présentant un meilleur retour d'information. Ces recherches englobent la détection, le suivi, le codage, la reconnaissance et la synthèse des visages en tenant compte les variations possibles de leur apparence (regard, lèvres, expressions, âge, genre...etc.).

Plusieurs domaines reposent sur la reconnaissance des expressions faciales comme : la sécurité (surveillance comme la prédiction des intrusions des voleurs dans des banques avec l'interprétation des expressions faciales des individus captés avec les caméras de surveillance), la robotique (interaction homme-machine), l'handicapé (communication par le visage à travers le langage des signes qui repose sur les expressions faciales pour pouvoir interpréter le sens des signes, les autistes ne peuvent pas reconnaître les émotions des personnes avec qui ils communiquent), les jeux vidéo (les consoles de jeux nouvelle génération : Xbox qui utilise Kinect), l'enseignement à distance (la transmission au professeur de l'état de l'étudiant sous forme d'information de haut niveau), dispositif dans les voitures pour avertir les conducteurs en cas de perte de vigilance, dispositif dans les avions (informations supplémentaires dans la boîte noire).

Les expressions faciales prennent, donc, une place importante dans le processus de communication humain et forment à elles seules un langage Co-verbal. Les expressions faciales accompagnent le langage parlé, en terme d'indicateur émotionnel. Elles expriment une part non-négligeable de son sens dans une communication orale.

Les nouvelles études s'intéressent à donner une description détaillée des expressions en vue de fournir un sens pour la machine. En codant les expressions faciales captées de différentes sources (vidéo, webcam, images).

Notre problématique est de trouver un ensemble de méthodes une fois regroupées vont former un système pour la reconnaissance des expressions faciales afin de pouvoir adapter le mode de communication de la machine aux modes de

communication humaine et l'appliquer ensuite dans un des domaines que nous venons d'expliquer ci-dessus.

Chapitre 1 : Psychologie des émotions

1. Introduction :

Pour pouvoir interpréter les expressions faciales, il faut d'abord comprendre les émotions qui sont communiquées : qu'est-ce qu'une émotion, comment les émotions sont-elles catégorisées, comment sont-elles véhiculées par les expressions du visage ?

2. Qu'Est-Ce Qu'une Emotion :

Les émotions nous signalent quelque chose du monde qui nous entoure ou qui nous habite, mais elles ne sont pas à priori, des pensées. C'est pour cela qu'il est difficile de les décrire, de les expliquer et de les représenter. Le principal problème est de les caractériser à l'aide de mots des manifestations qui associent une expérience mentale, donc subjective, à des manifestations corporelles et physiques, essentiellement ressentis.

Les chercheurs s'accordent à dire que pour parler d'émotion, il faut un Stimulus Inducteur, qui va provoquer une réponse comportementale ou corticale. L'émotion traduit un changement brusque par rapport à un état initial, un trouble qui interrompt le rythme habituel. Elle comprend des phénomènes cognitifs (esprit), physiologiques (organisme) et comportementaux (actions). Les sensations émotionnelles correspondraient à une perception cognitive des changements de l'état physique du corps : accélération du rythme cardiaque, modification hormonale, rougissement, pâleur, expression du visage, mouvements...etc.

Le terme émotion, du latin ex-movere, qui signifie étymologiquement mouvement vers l'extérieur. Sous le terme d'émotions, on englobe habituellement les phénomènes physiques, mentaux et les modifications qui accompagnent la perception. [H1].

Les émotions sont des actions. Certaines se traduisent par des mouvements des muscles du visage, comme les expressions faciales de joie, de colère, etc..., ou du

corps, la fuite ou la posture agressive. D'autre se traduisent par des actions internes, comme celles des hormones, du cœur ou des poumons. Les émotions sont donc d'une certaine façon publiques, on peut les mesurer, les étudier. Les sentiments par contre sont privés, subjectifs. Ils sont ressentis par un individu et par lui seul. Il ne s'agit pas de comportement mais de pensées. [D1]

2.1. Définition Consensuelle de l'émotion :

Il y a plusieurs définitions de l'émotion. Par exemple, pour Joseph LeDoux [L1] qui dit qu'une émotion est une expérience subjective. Ekman et Davidson [E1] définissent quant à eux l'émotion comme une réaction aiguë et transitoire, provoquée par un stimulus spécifique et caractérisée par un ensemble cohérent de réponses cognitives, physiologiques et comportementales. Cette pluralité a amené le besoin de définir l'émotion de manière plus consensuelle, pour travailler sur une base commune. Donc ils ont proposé une définition sur laquelle la majorité des acteurs s'appuient à ce jour :

[H3] *Les émotions sont le résultat de l'interaction de facteurs subjectifs et objectifs, réalisés par des systèmes neuronaux ou endocriniens, qui peuvent :*

1. *Induire des expériences telles que des sentiments d'éveil, de plaisir ou de déplaisir.*
2. *Générer des processus cognitifs tels que des réorientations pertinentes sur le plan perceptif.*
3. *Activer des ajustements physiologiques globaux.*
4. *Induire des comportements qui sont plus souvent, expressifs, dirigés vers un but et adaptatifs.*

Selon cette définition, les émotions sont réduites en trois composantes : une composante physiologique/biologique, une composante comportementale et une composante cognitive.

- *La composante physiologique* : de l'émotion fait référence à l'activation du système nerveux central, du système nerveux autonome et du système endocrinien qui accompagne une expérience émotionnelle et qui prépare l'individu à agir sur son environnement de manière adaptée.

- *La composante comportementale* : de l'émotion diffère d'un individu à un autre. Il s'agit, d'un côté de l'expression communicative, c'est-à-dire des comportements qui visent la communication de l'émotion à autrui et d'un autre côté, des réactions immédiates (réflexes) à une situation, comme par exemple la pétrification face à une situation effrayante. L'expression faciale est certainement la composante comportementale la plus étudiée et la plus significative.
- *La composante subjective* : que l'on désigne également comme composante cognitive, comprend l'ensemble des processus mentaux qui se développent suite à une émotion et qui interviennent dans la perception et la mémorisation d'une situation [P1]. L'expression émotionnelle est donc composée en grande partie de processus cognitifs, qui sont eux-mêmes une des causes principales de déclenchement des réponses émotionnelles.

3. Classification des émotions :

3.1. Émotions positives et négatives :

La distinction fondamentale entre les émotions positives et émotions négatives sont la 1ere façon pour classifier les émotions, qui se base sur les descriptions subjectives des émotions proposées par des sujets de différentes cultures. Cette distinction entre émotions positives et négatives est congruente l'idée de l'existence de deux systèmes de comportement distincts : un système orienté vers le rapprochement et la recherche du plaisir, poussé par une émotion positive et un système de comportement aversif, orienté vers l'évitement, poussé par une émotion négative [L2].

3.2. L'Approche Dimensionnelle des émotions :

L'approche dimensionnelle se propose de décrire les états émotionnels en les décomposant en deux, trois, parfois même quatre facteurs, selon les auteurs. Spécialement pour les expressions faciales, il existe un nuage d'expressions définissables par leur position dans un espace à plusieurs dimensions. La plus utilisée aujourd'hui est la théorie tridimensionnelle de Lang [L3] qui se définit selon trois axes : la valence (plaisante/déplaisante), l'intensité (calme/excitée) et la dominance (contrôlée/incontrôlée).

4. Les émotions de base, émotions universelles :

L'approche catégorielle soutient que l'ensemble des émotions peuvent être catégorisées par un petit ensemble d'émotions basiques, primaires ou fondamentales, qui seraient innées, non réductibles et communes à toute l'espèce humaine. Elles seraient représentées distinctement en mémoire, et seraient associées à des informations, idées et souvenirs également organisés de façon catégorielle [P2].

Cette approche de classification des émotions s'est développée en parallèle de l'étude des expressions faciales. En effet, c'est à partir de l'examen de l'expression faciale qu'Ekman et Friesen ont déduit que certaines émotions étaient indépendantes de la culture puisqu'elles sont exprimées et reconnues sur le visage d'autrui, quelle que soit la culture considérée. [E2] ils ont ainsi identifié six émotions dites universelles plus neutres : la colère, le dégoût, la peur, la joie, la tristesse et la surprise. Ces émotions apparaîtraient très tôt au cours du développement humain.

5. Qu'est-ce qu'une expression faciale ?

Les émotions, sous toutes les formes que l'on vient de décrire, peuvent être exprimées de différentes manières : les mots, les intonations de la voix, la gestuelle et les expressions faciales. Il semble néanmoins qu'il existe une iniquité entre ces différents signaux : c'est du visage, qui est la principale source de communication vers autrui, qu'émane la majorité des signaux émotionnels. C'est pourquoi on peut constater que les expressions faciales représentent le vecteur émotionnel également le plus étudié.

Une expression faciale peut être définie comme un changement dans le visage, perceptible visuellement, dû à l'activation (volontaire ou non) de l'un ou de plusieurs des 44 muscles du visage. Des expressions qui peuvent refléter, l'état émotionnel d'un individu ou un autre aspect de communication sociale [D2][F1].

L'expression des émotions sur le visage est un phénomène inné et universel. Paul Ekman est arrivé à cette conclusion en étudiant différentes cultures et en constatant que toutes exprimaient les émotions sur leurs visages [E2][E3]. De plus, en reprenant trente ans d'étude sur l'identification d'expressions faciales dans différentes cultures,

il a mis en évidence que toutes ces études rapportaient six émotions en commun, ensuite les qualifie de basique.

6. Conclusion :

Le visage représente peut-être le plus important des signaux dans les interactions sociales. Il constitue pour celui qui l'observe un stimulus multidimensionnel qui véhicule simultanément un grand nombre d'informations, chacune ayant une signification complexe. Ainsi, les visages ne fournissent pas uniquement des informations sur l'identité le sexe ou l'âge de la personne, mais ils délivrent aussi des signaux plus subtils liés à l'émotion, l'intention, la fiabilité...etc.

Les expressions faciales émotionnelles sont porteuses de sens, elles véhiculent des informations plus riches et plus diversifiées que l'émotion elle-même.

Chapitre 2 : Détection Automatique du visage :

Avant de procéder à l'analyse de l'expression faciale d'un visage fixe ou en mouvement, il convient de le détecter ou de le suivre afin d'extraire des informations pertinentes pour la reconnaissance de son expression.

1. Principe de la détection de visage:

La détection de visage permet de déterminer la présence et la position des visages disponibles dans une image donnée Ming & al [M1]. Plusieurs contraintes peuvent altérer la détection de visage et doivent être prises en compte:

Poses variables : Les images d'un visage varient en raison de la caméra-face relative poses (frontale, 45 degrés, profil, tête en bas), et certaines caractéristiques du visage comme un œil ou le nez peuvent être partiellement ou complètement obstruée.

La présence ou l'absence d'éléments de structure : Les traits du visage comme les barbes, moustaches et des lunettes peuvent être présents ou non et il y a une grande variabilité parmi ces éléments, y compris la forme, la couleur et la taille.

L'expression du visage : L'apparition de visages est directement touchée par l'expression du visage d'une personne.

Occlusion : Les visages sont partiellement occultés par d'autres objets. Dans une image avec un groupe de personnes, certains visages peuvent obstruer partiellement les autres visages.

L'orientation de l'image : les images du visage varient directement pour différentes rotations autour de l'axe optique des caméras.

Conditions d'imagerie : Lorsque l'image est formée, des facteurs tels que l'éclairage (spectre, la distribution et l'intensité de la source) et les caractéristiques de l'appareil (réponse du capteur, lentilles) affectent l'apparence d'un visage.

2. Comparaison des algorithmes de détection de visage

La détection de visage est une technologie cruciale pour les applications telles que la reconnaissance des visages, la lecture automatique et reconnaissance des expressions faciales. Un aspect qui a ralenti les progrès dans ce domaine est l'absence d'études référentielles dont le but est non seulement le développement de systèmes complets, mais l'analyse de la façon dont les différents éléments d'un système contribuent à son succès.

L'objectif est donc d'effectuer une comparaison grossière des techniques utilisées dans les différents systèmes de détection les plus réussis. En se basant sur deux études de comparaisons entre les algorithmes de détection de visage qui ont été faits au niveau de (Département des Sciences Cognitives, Université de Californie, San Diego) et (Tula State University) [N1].

Voici ci-dessous une liste des bibliothèques disponibles offrant des algorithmes de détection de visage:

- Intel OpenCV (OCV)
- LuxandFaceSDK (FSDK)
- Face Detection Library (FDLib)
- SIFinder (SIF)
- FaceOnIt (FOI)
- NeurotechnologyVeriLook (VL).

2.1. Brève description des algorithmes disponibles dans chaque bibliothèque :

2.1.1. La bibliothèque SIF (SIFinder) :

Cet algorithme [K1] a été développé dans le laboratoire d'analyse des données de l' State Université de Tula State. La méthode repose sur la détection des yeux définies comme zones sombres dans l'image du visage. Au début, l'algorithme trouve des points de luminosité minimale dans l'image. Ces points sont triés ensuite, certains d'entre eux sont éliminés, et le reste est regroupé par paires. Par la suite des transformations affines (rotation, changement d'échelle et décalage) sont appliquées sur chaque paire de points et projetées dans des réseaux de taille fixe. Après ces

transformations, les réseaux sont représentés comme un vecteur de caractéristiques (valeurs de luminosité de nœuds) et sont envoyés au classificateur (SVM) de deux classes, entraînés à l'avance sur un grand nombre de visages et de non-visages. Il faut également définir le bon compromis entre les deux paramètres :

- FAR(*False Acceptance Rate: Rapport d'erreur de type II, qui indique la probabilité d'erreur de classification des images ne contenant pas de visage*),

- FRR(*False Rejection Rate: Rapport d'erreurs de type I, qui indique la probabilité d'erreur de classification des images contenant un visage*), en changeant l'hyperplan de séparation de visage et de non-visage dans l'espace des caractéristiques (valeurs de luminosité des nœuds de treillis).

2.1.2. La librairie FDLib (Face Detection Library):

La bibliothèque de la détection des visages (FDLib) a été développée par **Keinzle et al.** [K2]. Les auteurs ont proposé une méthode de calcul des approximations rapides pour soutenir les fonctions de prise de vecteurs (méthode dite de jeu réduit) dans le domaine de la détection d'objet. Cette méthode crée expansions du noyau rare, qui peuvent être évaluées via des filtres séparables. Cet algorithme a un seul paramètre de réglage qui permet de contrôler la "rigueur" de détection de visage par l'intermédiaire de la modification du nombre de filtres séparables dans lesquelles les vecteurs de support réduits sont décomposés.

2.1.3. FaceOnIt (FOI) :

FaceOnIt (<http://www.faceonit.ch>) est un SDK de détection de visage développé à l'institut de recherche Idiap [S1][M2]. Il est basé sur l'architecture en cascade introduite par Viola-Jones et sur une extension de motifs binaires locaux. PSL ont été proposées par Ojala et al. [O1] pour la classification de la texture. Mais par la suite son invariance de rotation et de la légèreté de calcul ont été utilisés Ahonen et al [A2] pour développer un algorithme efficace et rapide de reconnaissance faciale. Comme un paramètre pour trouver le compromis entre les FAR et FRR, nous avons changé la valeur du seuil de confiance de visage.

2.1.4. FaceSDK (FSDK) et VeriLook (VL) :

FaceSDK (version 2.0) et VeriLook (version 4.0) ont été fournies par Luxand Inc.(<http://www.luxand.com>) et Neurotechnology (<http://www.neurotechnology.com>) respectivement. Ces deux algorithmes sont des produits commerciaux, et donc aucun détail sur le principe de leur fonctionnement n'a été divulgué. Pour trouver le compromis entre les FAR et FRR, ils ont changé la valeur du seuil de confiance du visage pour le paramètre de la fonction « SetFaceDetectionThreshold » FSDK affectant le seuil de FaceSDK VL et modifiés.

2.1.5. La librairie OpenCV (Intel Open Computer Vision library) :

Dans une étude comparative, Viola-Jones [V1] [V2] ont utilisé la réalisation prolongée de l'algorithme de détection objet porteur des fonctions **Haar-like** sur OpenCV 1.0. Les caractéristiques de **Haar-like**, initialement proposées par **Papageorgiou et al.** [P3], valorisent les différences d'intensité moyenne entre deux régions rectangulaires, ce qui les rend capables d'extraire la texture sans dépendre des intensités absolues. Cependant [V1], au cours de leur travail sur des algorithmes de détection d'objets Viola et Jones ont étendu l'ensemble des fonctions et développé une méthode efficace pour évaluer ce qui est appelé une «image intégrale"[V1]. Plus tard **Lienhart et al.** [L4] ont introduit un système efficace pour le calcul de la fonction de rotation de 45 ° et l'ont inclus dans la bibliothèque OpenCV. Il convient de mentionner que face à la plupart des algorithmes existants en utilisant un fort classificateur unique, l'algorithme de **Viola-Jones** utilise un ensemble de classificateurs faibles, construit par seuillage d'une fonction Haar-like. En raison du grand nombre de classificateurs faibles, ils peuvent être classés et organisés en cascade. Dans ce travail, ils ont testé en cascade, pour la détection de visage frontal inclus par défaut dans OpenCV 1.0.

2.1.5.1. La méthode de fonctionnement

Le Détecteur de visage OpenCV utilise une méthode que 'Paul Viola' et 'Michael Jones' ont publiée en 2001. Habituellement appelé tout simplement la méthode Viola-Jones ou même juste Viola-Jones cette approche de la détection d'objets dans les images combinant quatre concepts clés

2.1.5.2. Principe

- Des fonctions simples rectangulaires, appelées fonctions de Haar.

- Une image intégrale pour la détection rapide de fonction.
- La machine « AdaBoost » qui est une méthode d'apprentissage.
- Un classificateur en cascade combinant de nombreuses fonctionnalités efficaces.

2.1.5.3. Principe de détection par OpenCV

OpenCV utilise un type de détecteur de visage appelé un classificateur en cascade de Haar. La figure ci-dessous montre un exemple de détecteur de visage OpenCV dans l'action.



Figure 2.1 : Détection du visage. [C1]

Contenu d'une image, qui peut provenir d'un fichier ou d'une vidéo en direct, le détecteur de visage examine chaque emplacement de l'image et le classe comme "Face" ou "Non Face." La classification suppose une échelle fixe pour le visage, par exemple 50x50 pixels. Cependant les visages dans une image peuvent être plus petits ou plus grandes que cela, le classificateur fonctionne sur l'image à plusieurs reprises, pour rechercher des visages à travers une gamme d'échelles. Cela peut augmenter le coût de traitement, mais grâce à des optimisations algorithmiques, la classification se fera très rapidement, même quand il est appliqué à plusieurs échelles.

L'algorithme utilise les données stockées dans un fichier XML et permet de décider comment classer chaque emplacement de l'image.

2.1.5.4. Comment la détection des composants du visage fonctionne dans la méthode utilisée ?

Les caractéristiques que Viola et Jones utilisent les ondelettes de Haar. Wavelets de Haar sont des longueurs d'onde (wavelength) des ondes carrées. En deux dimensions, une onde carrée est une paire de rectangles adjacents - une lumineuse et une sombre.

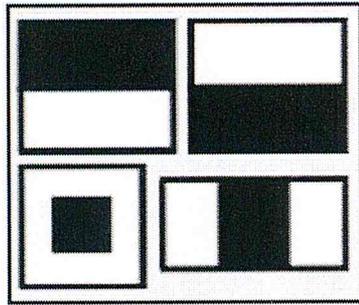


Figure 2.2 : des exemples de caractéristiques de Haar utilisés dans OpenCv. [C1]

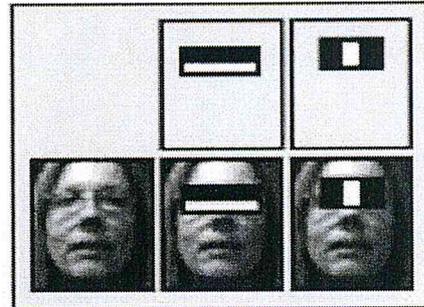


Figure 2.3 : les deux premières caractéristiques de Haar dans l'original Viola-jones cascade. [C1]

Les combinaisons de rectangle réels utilisés pour la détection d'objets visuels ne sont pas de vraies ondelettes de Haar. Au lieu de cela, ils contiennent des combinaisons de rectangle mieux adaptées aux tâches de reconnaissance visuelle. En raison de cette différence, ces caractéristiques sont nommées caractéristiques de Haar, ou des caractéristiques de Haar like, plutôt que ondelettes de Haar. La figure 2.3 montre les caractéristiques qui utilisent OpenCV.

La présence d'une caractéristique de Haar est déterminée par la soustraction de la valeur moyenne de pixel de la région sombre et depuis la valeur moyenne de pixel de la région lumineuse. Si la différence est supérieure à un seuil (fixé lors de l'apprentissage), cette fonctionnalité est dite d'être présent.

Pour déterminer la présence ou l'absence de certaines caractéristiques de Haar à chaque emplacement de l'image et à plusieurs échelles efficacement, Viola et Jones ont utilisé une technique appelée une image intégrale. En général, les «intégrer» signifie l'ajout de petites unités ensembles. Dans ce cas, les petites unités sont des valeurs de pixel. La valeur de l'intégrale pour chaque pixel est la somme de tous les pixels au-dessus et à sa gauche. En partant du haut à gauche et traverser vers la droite et vers le bas, l'image entière peut être intégrée avec quelques opérations entières par pixel.

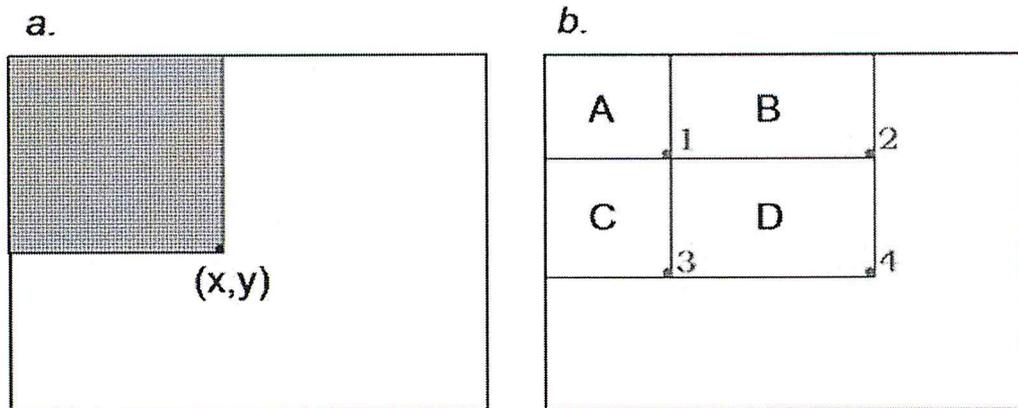


Figure 2.4 : L'astuce d'image intégrée [C1]

- Après l'intégration, le pixel (x,y) contient la somme de toutes les valeurs de pixels dans le rectangle grisé.
- La somme des valeurs de pixels dans le rectangle D est $(x_4,y_4)-(x_2,y_2)-(x_3,y_3)+(x_1,y_1)$

Comme la [figure 2.4] montre, après l'intégration, la valeur à chaque emplacement de pixel (x, y) , contient la somme de toutes les valeurs de pixels dans une zone rectangulaire qui a un coin en haut à gauche de l'image et l'autre à l'emplacement (x, y) . Pour trouver la valeur moyenne de pixels dans ce rectangle, vous n'avez besoin que de diviser la valeur en (x, y) par la superficie du rectangle.

Mais que faire si vous voulez connaître les valeurs additionnées pour un autre rectangle, celui qui n'a pas un coin en haut à gauche de l'image? Figure 2.4 (b) montre la solution à ce problème. Supposons que vous voulez que les valeurs additionnées à D. Vous pouvez considérer sa comme étant la somme des valeurs des pixels dans le rectangle combiné, $A + B + C + D$, moins les sommes en rectangles $A + B$ et $A + C$, plus la somme des valeurs de pixel dans A. En d'autres termes :

$$D = A + B + C + D - (A + B) - (A + C).$$

Idéalement, $A + B + C + D$ est la valeur de l'image intégrale à l'emplacement 4, $A + B$ est la valeur à l'emplacement 2, $A + C$ est la valeur à l'emplacement 3, et A est la valeur à l'emplacement 1. Ainsi, avec une image intégrale, vous pouvez trouver la somme des valeurs de pixel pour un rectangle dans l'image originale avec seulement

trois opérations de entiers: $(x_4, y_4) - (x_2, y_2) - (x_3, y_3) + (x_1, y_1)$.

Pour sélectionner les caractéristique spécifiques de Haar à utiliser, et de fixer des niveaux de seuils, Viola et Jones utilisent une méthode d'apprentissage automatique appelée AdaBoost. Elle combine plusieurs classificateurs "faibles" pour créer un classificateur «fort». «Faible» signifie ici le classificateur ne reçoit que la bonne réponse un peu plus souvent que deviner au hasard. Qui est mauvais, mais si un ensemble de classificateurs faibles sélectionnés par ADaBoost sont combinés et chacun incite la réponse à prendre la bonne direction, qui fait qu'il y aura une force combinée pour arriver à la bonne solution.

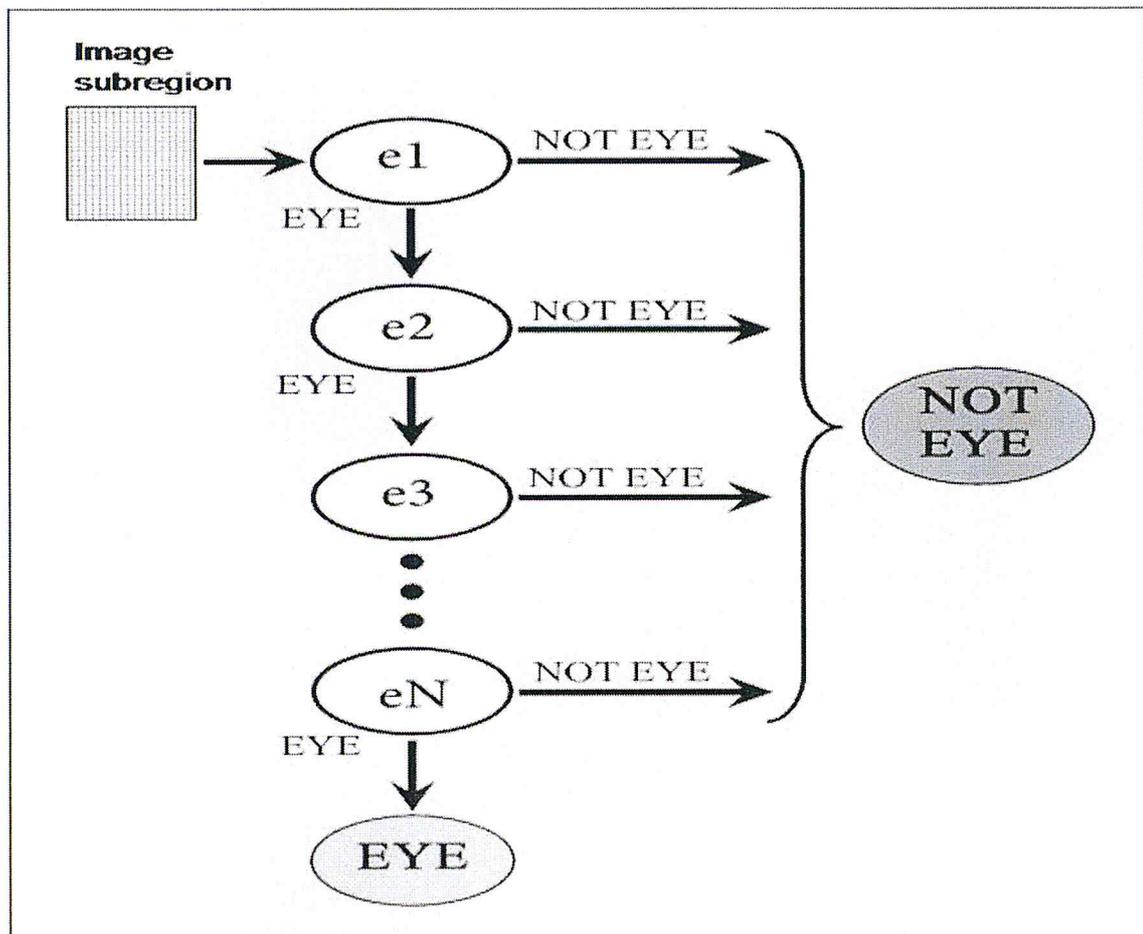


Figure 2.5 : Le classificateur de cascade est un réseau de filtres. Sous-régions de l'image qui en font à travers toute la cascade sont classées comme «eye». Tous les autres sont classés comme « Not Eye ».

Le seuil d'acceptation à chaque niveau définis une valeur suffisamment bas Pour permettre de passer tous les niveaux, ou presque, des exemples d'œil dans l'ensemble de la formation. Les filtres à chaque niveau sont formés aux images de formation classifié qui ont passé toutes les étapes précédentes. (L'ensemble de la formation est une grande base de données des œil, peut-être un millier.) Pendant l'utilisation, si l'un de ces filtres ne parvient pas à passer une zone d'image, cette région est immédiatement classée comme «non Eye ». Quand un filtre passe une zone d'image, il va au filtre suivant dans la chaîne. Les régions de l'image qui passent à travers tous les filtres de la chaîne sont classées comme «Eye». Viola et Jones ont doublés cette chaîne de filtrage d'une cascade.

L'ordre des filtres dans la cascade est basé sur la pondération de l'importance qu'AdaBoost attribue. Les filtres les plus lourdement pondérés viennent en premier, pour éliminer des zones d'image Not-EYE aussi rapidement que possible. La figure 2.3 montre les deux premiers traits de l'original cascade Viola-Jones superposée sur le visage. La première touche au large de la région de la joue étant plus légère que la région des yeux. La seconde utilise le fait que l'arête du nez est plus légère que les yeux.

3. Discussion :

[N1] Dans ce travail, les sept algorithmes de détection de visages ont été testés et le modèle statistique de l'estimation de la position des yeux pour les algorithmes décrivant les visages par rectangle ont été proposé.

Selon le résultat de cette étude **VeriLook** a la meilleure performance dans divers paramètres et a la première place dans l'épreuve de vitesse (18-20 images par seconde). **FDLib** montre de bonnes caractéristiques de vitesse (deuxième place), mais il montre la plus mauvaise performance. **OpenCV**, l'algorithme le plus populaire et libre de détection de visages disponible, a pris la deuxième place dans le teste de performance et a une vitesse suffisante. SIF développé à Tula State University a démontré la performance moyenne. Il est intéressant de noter que **VeriLook** a le plus grand nombre d'images classées uniques, c'est à dire des images qui ont été mal classés par d'autres algorithmes.

4. Conclusion :

Dans ce chapitre nous avons parlé des différentes méthodes de détection du visage existant, et d'une étude comparative effectuée sur ces méthodes [N1] qui a résulté à une classification de ces méthodes selon leurs niveaux de performance et la vitesse d'exécution.

Chapitre 3 : Modèle Actif de Forme

1. Introduction :

La localisation des points caractéristiques (LandMarks) dans le visage est l'une des étapes les plus importantes dans la reconnaissance des émotions à partir d'expressions faciales.

Le modèle actif de forme (Active Shape Model ASM) représente l'une des méthodes utilisées pour la détection des points caractéristiques dans le visage.

2. Forme :

Une forme est un vecteur de coordonnées (x, y) d'une série de points repères. Les points sont reliés les uns aux autres dans un certain sens invariant. Si on bouge la forme, ça reste la même forme. Si on l'étend ou on la pivote, ça reste la même forme.

La distance entre deux points est la distance Euclidienne. La distance entre deux formes est la somme des distances entre leurs points correspondant. La distance entre deux forme x_1 et x_2 est la racine carrée de la distance entre les points de forme $\sqrt{(x_1 - x_2) \cdot (x_1 - x_2)}$ après alignement. [M3]

3. Alignement de forme :

Une forme peut être alignée à une autre forme en appliquant une transformation qui donne la distance minimum entre les formes. Les transformations permises sont : la mise en échelle, la rotation. [M3]

4. Fonctionnement d'ASM :

Le modèle actif de forme est entraîné sur un ensemble d'images où les points caractéristiques sont repérés manuellement (manuellement Landmarké), qui veut dire que quelqu'un doit les marquer manuellement avant le début de l'entraînement (apprentissage). [M3]

Après l'apprentissage on peut utiliser le ASM pour rechercher des caractéristiques sur le visage. L'idée général est d'essayer de localiser chaque point repère (LandMark) indépendamment, après corriger la position si nécessaire en regardant la façon dont les LandMarks sont situés les uns par rapport aux autres. Pour faire cela ASM est construit de deux types de sous-modèles :

- Le modèle de forme.
- Le modèle de profil.

L'algorithme suivant montre les itérations pour une solution en utilisant les deux sous-modèles :

Algorithme ASM

Entrée image du visage

- 1) Générer une forme de départ en localisant la position global du visage.
- 2) *Répéter*
 - 3.a) *pour* chaque point de forme
 - 3.b) *pour* chaque décalage (i.e. regardé dans zone autour du point)
 - 3.c) construire un profil en échantillonnant la zone autour du décalage.
 - 3.d) mesurer l'ajustement du profil par rapport au profil du modèle.
 - 3.e) déplacer le point de décalage de la meilleure adaptation de profil.
- 4) Ajuster la forme suggérée pour se conformer au modèle de forme.
- 5) *Jusqu'à* la convergence (i.e. jusqu'à ce qu'aucune autre amélioration en termes d'ajustement n'est possible).

Sortie : une forme qui donne les coordonnées (x, y) des points de repère pour le visage (LandMarks)

4.1. Modèle de forme (Shape Model)

Le travail du modèle de forme est de convertir la forme suggérer par le modèle de profil à une forme de visage admissible. Avant de construire le modèle de forme, les formes d'apprentissage sont alignées. [M3] Après le modèle de forme se compose d'un visage moyen et la distorsion autorisée pour un visage moyen :

$$\hat{x} = \bar{x} + \Phi b$$

Où :

\hat{x} est le vecteur de forme généré (tous les x suivi par tous les coordonnées y).

\bar{x} est la forme moyenne : la moyenne des formes d'apprentissage alignées x_i , définie

par :

$$\bar{x} = \frac{1}{n_{shapes}} \sum_{i=1}^{n_{shapes}} x_i$$

Φ est la matrice des vecteurs propres (Eigenvectors) de la matrice de covariance S_s des points de forme d'entraînement.

$$S_s = \frac{1}{n_{shapes} - 1} \sum_{i=1}^{n_{shapes}} (x_i - \bar{x})(x_i - \bar{x})^T$$

Équation 3.1 : calcul de l'EigenVector et l'Eigenvalue

EigenVectors et Eigenvalues :

L'EigenVector x , de la matrice A est un vecteur spéciale, avec la propriété suivante :

$$A \cdot x = \lambda \cdot x \text{ où } \lambda \text{ est nommée l'EigenValue}$$

Pour trouver les EigenValues de la matrice A faut d'abord trouver les racines de :

$$\text{Det}(A - \lambda I) = 0$$

Et puis résoudre le système linéaire suivant pour chaque EigenValue pour trouver l'EigenVector :

$$(A - \lambda I) x = 0$$

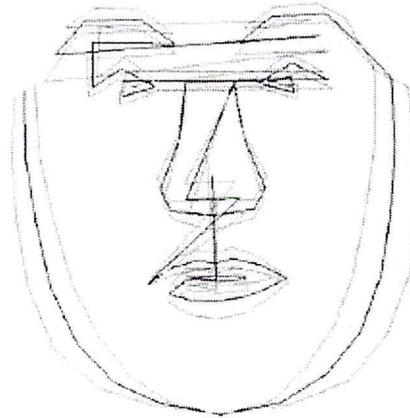


Figure 3.1 : le visage moyen (noir), variation des composants principaux (gris).
[M3]

4.1.1. Générer des formes depuis le modèle de forme :

On peut utiliser l'équation de \hat{x} pour générer différentes formes en variant le paramètre de vecteur b . en gardant les éléments de b dans les limites pour s'assurer de générer des visages plus réalistes. [M3]

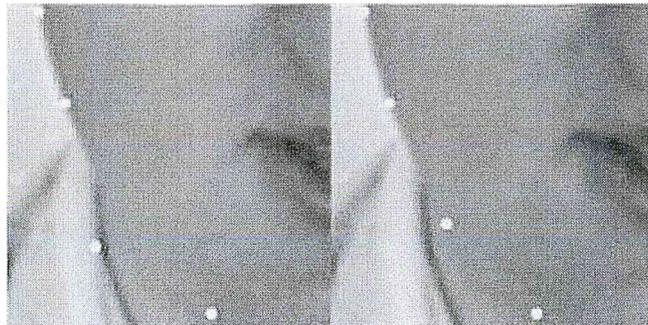


Figure 3.2 : Opération de modèle de forme. [M3]

Gauche : LandMarks localisé en correspondance avec le profil

Droite : Les Mêmes LandMarks conforme au modèle de forme.

4.2. Modèle de profile :

Le travail du modèle de profile est de prendre une forme approximative du visage et produire une meilleur forme suggérée par la comparaison des Templates aux LandMark. On commence la recherche avec le visage moyen depuis le modèle de forme, aligner et positionner avec le détecteur global du visage. [M3]

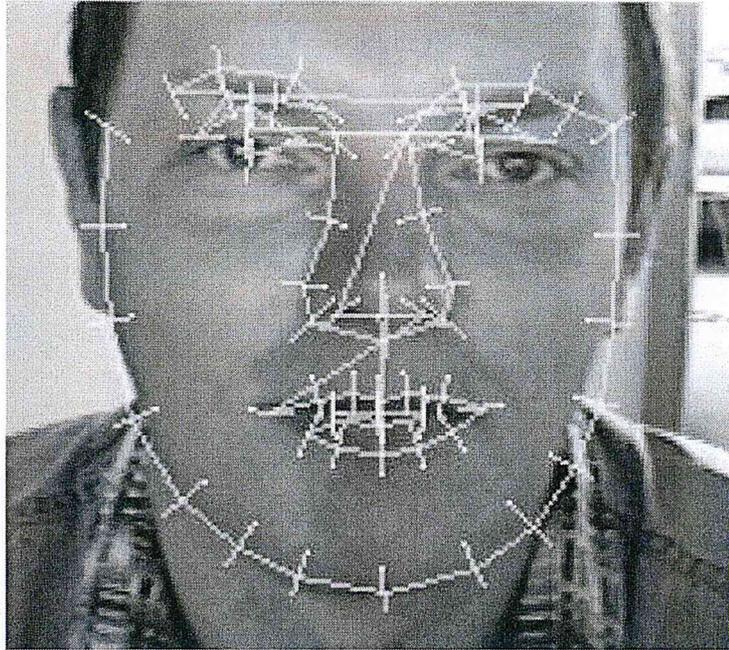


Figure 3.3 : le visage Moyen (jaune) positionné sur un visage de recherche au début de la recherche. Les lignes blanches sont « des moustaches », ainsi que l'intensité de l'image va être échantillonné pour former le profile. [M3]

4.2.1. Formé un profile :

Pour former le vecteur de profile g dans une LandMark, on échantillonne les intensités de l'image au long de la moustache « whisker », une moustache est un vecteur au point repère (Landmark) qui est orthogonal à la bordure de la forme. [M3]

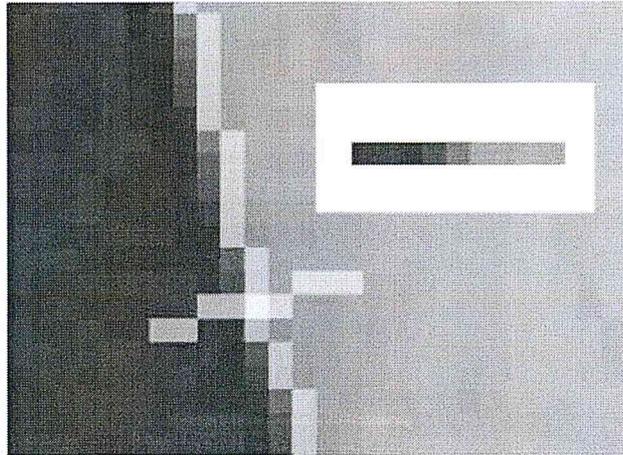


Figure 3.4 : la ligne jaune est la bordure de la forme. La ligne grise est la moustache, orthogonal à la bordure. Le LandMark est à l'intersection de la bordure de la forme et la moustache. [M3]

Le vecteur de profile est formé comme suit :

1. Mettre chaque élément du vecteur de profile au niveau (0...255).
2. Remplacer chaque élément du profile par le gradient d'intensité, ceci est fait par le remplacement de l'élément de profile à chaque position i avec la différence entre elle est l'élément à $i-1$.
3. Diviser chaque élément du vecteur résultant par la somme des valeurs absolue de tous les éléments du vecteur.

4.2.2. Construire un modèle de profile durant l'apprentissage :

Durant l'apprentissage, on construit un modèle pour chaque LandMark en créant un profile moyen \bar{g} et la matrice de covariance S_g de tous les profils de l'apprentissage (un de chaque image) au LandMark. [M3]

4.2.3. Rechercher le meilleur profile :

Durant la recherche, à chaque LandMark on forme plusieurs profils de recherche par la prise d'échantillon de l'image dans le voisinage du LandMark. Chaque LandMark est centré avec un déplacement positif ou négatif au long de la moustache. Typiquement les profils de forme sont décalés jusqu'à environ de ± 3 pixels le long de la moustache. [M3]

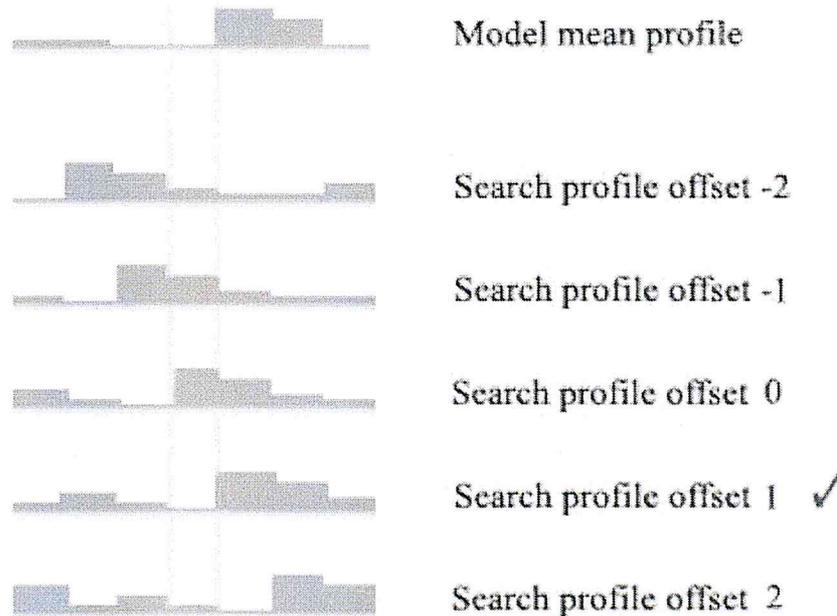


Figure 3.5 : La recherche d'une correspondance de profil. [M3]

Le profil supérieur est le modèle moyen de Profil \bar{g} . Le reste des profils sont des profils générés durant la recherche par la prise d'échantillon de l'image autour la position courante du LandMark. La meilleure correspondance est le offset 1.

La distance entre le profil de recherche g et le modèle moyen de profil \bar{g} est calculée en utilisant la distance de Mahalanobis.

$$Distance = (g - \bar{g})^T S_g^{-1} (g - \bar{g})$$

Ce processus est répété pour chaque LandMark avant de donner la main au modèle de forme.

5. Conclusion :

Dans ce chapitre nous avons parlé de la méthode l'extraction des points repères Landmarks ASM, qui se compose de deux sous-modèles : le modèle de profil qui essaie de localiser chaque LandMark indépendamment, et puis le modèle de forme qui corrige la position si nécessaire en regardant comment les LandMarks sont localisés l'un par rapport à l'autre.

Chapitre 4 : Classification

1. Introduction :

Les émotions sont le résultat de différents facteurs et leur état peut être révélé ou pas via un nombre de canaux comme la voix émotionnelle, la pose, les gestes, la direction du regard, les expressions faciales.

Les expressions faciales expriment les émotions mais les émotions ne sont pas la seule source de ces expressions, dans le contraste de la reconnaissance des expressions faciales, la reconnaissance des émotions est une tentative d'interprétation souvent qui demande la compréhension d'une situation donnée.

L'expression faciale joue un rôle important dans la communication humaine. Mehrabien [M4] indique que dans la communication face-à-face entre les individus, 7% du message de communication est transmis via un langage linguistique, 38% est due au paralangage et 55% est transmis via l'expression faciale. Ceci souligne l'importance de l'expression faciale dans notre vie personnelle et professionnelle.

Ces dernières années l'interaction homme-machine intéresse les chercheurs qui se sont concentrés sur le développement de l'interface et aussi ont amélioré la réaction de l'ordinateur aux actions de l'utilisateur jusqu'ici capté via le clavier et la souris, mais l'avancée technologique permet aux ordinateurs de voir et entendre via la webcam et le microphone et donc avoir un nouveau moyen de capter les actions de l'utilisateur.

C'est la raison pour laquelle on souhaite créer une interface homme machine permettant de décrypter ces expressions d'une manière automatique en exploitant les dernières avancées technologiques en vision par ordinateur. L'objectif est de concevoir une interface plus conviviale et intuitive capable d'analyser l'image (ou vidéo) d'un individu en provenance d'une webcam et identifier ses gestes faciaux, le système doit pouvoir nous fournir des informations qualitatives et quantitatives sur les expressions faciales sur différents niveaux, sur le niveau le plus bas, le système doit capter l'expression faciale de l'individu observé. Dans le niveau suivant, le

système doit classifier les expressions détectées. Et dans le niveau le plus haut le système doit donner une interprétation à l'expression reconnue en indiquant l'émotion exprimée.

2. Data Mining :

Data Mining a attiré beaucoup d'attention dans le domaine de l'information dans les dernières années. En raison de la disponibilité d'énorme quantité de données et le besoin de transformer ces données à des informations et connaissances utiles.

2.1. Définition :

Data Mining [H2] désigne l'extraction des connaissances depuis une grande quantité de données. Donc le Data Mining est un processus où des méthodes intelligentes sont appliquées pour extraire des schémas de données.

Data Mining est un domaine interdisciplinaire il regroupe un ensemble de disciplines: système base de données, statistiques, apprentissage automatique, la visualisation, et la science d'informations.

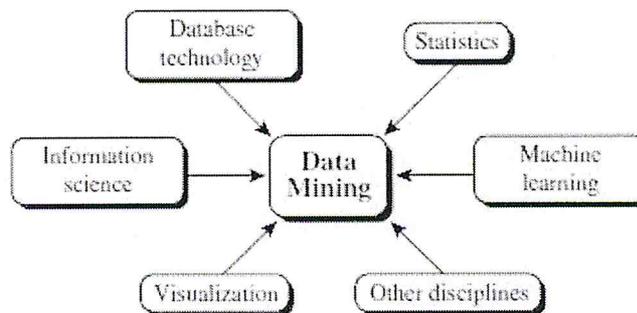


Figure 4.1. Les disciplines du Data Mining

2.2. Les taches du Data Mining :

Plusieurs techniques sont utilisées dans le Data mining pour réaliser plusieurs taches [E4] :

- Classification.
- Clustering (segmentation).
- Estimation.

- Prédiction.

a) Classification :

Est un processus qui permet d'organiser un ensemble de données en classes cohérentes ou homogènes.

Le but de la classification est de former des classes cohérentes et bien isolées.

- Cohérente : les éléments appartenant à la même classe partagent de nombreuses caractéristiques communes et donc se ressemblent fortement.
- Isolé : les éléments appartiennent à deux classes différentes ne se ressemblent pas et ne partagent pas les même caractéristiques.

Les techniques les plus utilisées dans la classification sont :

- Les arbres de décision.
- Les k plus proches voisins.
- Les réseaux de neurones.

b) Clustering (Segmentation) :

Consiste à former des groupes pour construire des segments homogènes à l'intérieur d'une population hétérogène. [E4] Pour cette tâche il s'agit de maximiser l'homogénéité à l'intérieur de chaque groupe et de maximiser l'hétérogénéité entre les groupes.

Les techniques les plus utilisés dans le Clustering sont :

- Algorithme hiérarchique.
- K-means.
- Algorithme génétique.

c) L'estimation :

Contrairement à la classification le résultat de l'estimation permet d'obtenir une variable continue en fonction des caractéristiques de l'objet.

Un des intérêts de l'estimation est de pouvoir ordonner les résultats pour ne retenir si on le désire que les n meilleurs valeurs.

La technique la plus utilisée dans l'estimation est :

- Réseau de neurone.

d) Prédiction :

Consiste à estimer une valeur future à partir des valeurs connues. Elle se base sur le passé et le présent et le résultat situé dans le future. La meilleure méthode pour mesurer la qualité de la prédiction est d'attendre.

Les techniques les plus utilisées pour la prédiction sont :

- Les arbres de décision.
- Les réseaux de neurones.

2.3. Les techniques du DataMining :

1) K plus proche voisins :

L'algorithme de k plus proche voisin est l'un des algorithmes de classification supervisée, qui à partir d'un échantillon fini d'objets déjà classés on classe au mieux de nouveaux objets ne faisant pas déjà partie de l'échantillon initial.

a) Principe :

- Rechercher les k plus proches voisins.
- Déterminer les classes correspondantes.
- Affecter à la classe la plus représentée.

b) Etapas :

- 1) choix d'un entier k.
- 2) calculer les distances.
- 3) Retenir les k observations pour lesquelles ces distances sont les plus petites.
- 4) Compter le nombre de fois ou ces k observations apparaissent dans chacune des classes.
- 5) Choisir la classe la plus représentée.

c) L'Algorithme :

Paramètre: le nombre k de voisin.

Donnée: un échantillon de n exemples et leurs classes (tel que (classe de x est $c(x)$)).

Entrée : un enregistrement y .

Pour chaque exemple x .

Faire calculer la distance (x, y)

Fait.

Pour chaque $x \in kppv(y)$ ($kppv = k$ plus proche voisins)

Faire compter le nombre d'occurrence de chaque classe.

Fait.

Attribuer à y la classe la plus fréquente.

Fin.

- Cas d'égalité :

Dans le cas d'égalité dans le nombre d'occurrence de chaque classe on :

- 1- Augmente la valeur de k de 1 ($k+1$)
- 2- Tirer au hasard une classe parmi les classes candidates.

d) Avantage :

Le plus grand avantage de l'algorithme du k plus proches voisins est qu'il est très simple à mettre en œuvre.

e) Inconvénients :

L'un des inconvénients de cet algorithme est qu'il ne contient pas d'étape d'apprentissage.

Et la classification est coûteuse car à chaque ajout d'un élément on doit calculer avec tous les éléments déjà existants.

2) Les réseaux de neurones :

Plusieurs avancées ont été faites dans le développement de système intelligent, certains inspirés des réseaux de neurones de la biologie, des chercheurs dans différents disciplines sont entrains de concevoir des réseaux de neurones artificiels pour résoudre une variété de problèmes dans la reconnaissance des motifs, la prédiction, l'optimisation, mémoire associative, et le contrôle. [T2]

Un réseau de neurone peut être vu comme une fonction ayant un certain nombre d'entrées et un certain nombre de sorties et pour faire le traitement un certain nombre de couches cachées. Le principe d'apprentissage est de donner en entrée du réseau un certain nombre d'exemples et de fixer la sortie à la valeur désirée.

a) La structure :

Chaque neurone est un processeur élémentaire. Il reçoit un nombre variable d'entrées. A chacune de ces entrées est associé un poids w représentatif de la force de connexion. Chaque processeur élémentaire est doté d'une sortie unique. [T2]

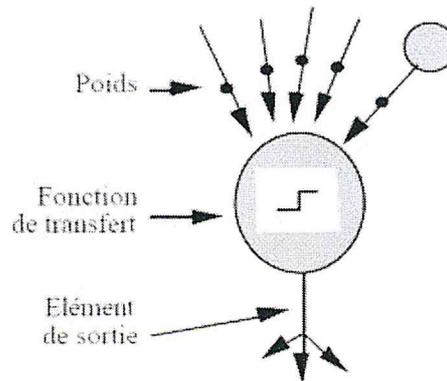


Figure 4.2 : structure du réseau de neurone. [T2]

b) **Structure d'interconnexion :**

- **Réseau multicouches :** [T2] les neurones sont arrangés par couche. Il n'y a pas de connexion entre neurones de la même couche et les connexions ne se font qu'avec les neurones des couches suivantes. Dans la plus part des cas, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante. Donc on appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble des neurones de sortie. Les couches intermédiaires n'ayant aucun contact avec le monde extérieur sont appelés couches cachées.

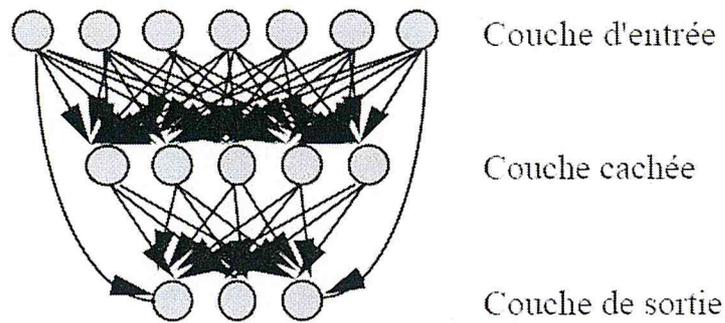


Figure 4.3 : définition des couches d'un réseau multicouches [T2]

- **Réseau à connexion locales :** [T2] Il s'agit d'une structure multicouche, mais où chaque neurone entretient des relations avec un nombre réduit et localisé de neurone de la couche suivante. Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique.

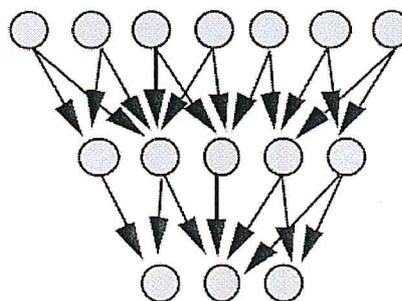


Figure 4.4 : Réseau à connexion locales. [Tou92]

- **Réseau à connexion récurrentes** : [T2] Les connexions récurrentes ramènent l'information en arrière.

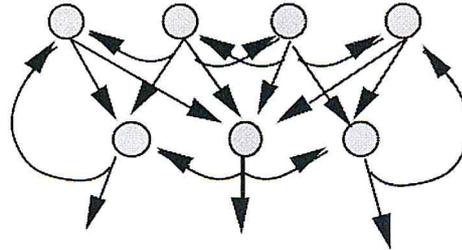


Figure 4.5 : Réseau à connexion récurrentes. [T2]

- **Réseau à connexion complète** : [T2] c'est la structure d'interconnexion la plus générale chaque neurone est connecté avec tous les neurones du réseau et à lui-même.

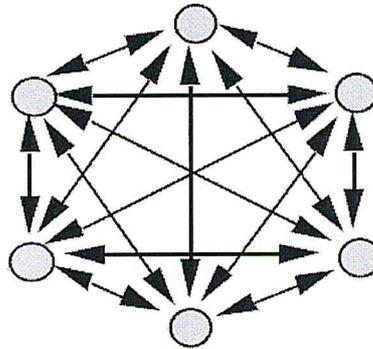


Figure 4.6 : Réseau à connexion complète. [T2]

c) **Apprentissage** :

- **La rétro propagation** : [R1] la rétro propagation est actuellement l'outil le plus utilisé dans le domaine de réseaux de neurones. C'est une technique de calcul des dérivées qui peut être appliquée à n'importe quelle structure de fonctions dérivables. Mathématiquement, cette méthode est basée sur l'algorithme de descente du gradient et utilise les règles de dérivation des fonctions dérivables. dans cette méthode, l'erreur commise en sortie du réseau sera rétro propagée vers les couches d'où le nom de rétro propagation.

- **Kohonen** : [K3] une méthode d'apprentissage non supervisée consistant, pour un réseau, à détecter des *régularités* dans des configurations présentées en entrées et à les classifier. Kohonen est constitué d'une couche d'entrée de N neurones connectés aux M neurones d'une couche de sortie elle-même interconnectée.

d) Avantage :

- Robustesse aux bruits et qui peut donc être utilisé dans la reconnaissance des formes (son, images...etc.).
- La rapidité de classification une fois le réseau construit.

e) Inconvénients :

- Apprentissage très long qui prend en charge plusieurs paramètres.
- Pouvoir explicatif faible vu que les réseaux de neurone sont des boîtes noires.

3) K-means :

k-moyennes ou k-means est une méthode de classification non supervisée, un algorithme de partitionnement de données, son but est de diviser les observations en k partitions (cluster) dans lesquelles chaque observation appartient à la partition avec la moyenne la plus proche.

L'algorithme k-means partitionne un ensemble de n objets à k cluster tel que la similarité intracluster est haute et intercluster est basse. La similarité du cluster est mesurée par la valeur moyenne des objets dans le cluster, qui peut être vue comme le centroid du cluster.

a) Algorithme :

Paramètre : le nombre k des clusters

Entrée :

- Un échantillon de n enregistrements $\{x_1, \dots, x_n\}$.

Répéter

- Choisir k centres initiaux $\{c_1, \dots, c_k\}$.
- Pour chacun des n enregistrements affecter le au cluster i dont le centre c_i le plus proche.

- Calculer le nouveau centre pour tout i tel que c_i est la moyenne des éléments du cluster.

Jusqu'à aucun changement.

Fin.

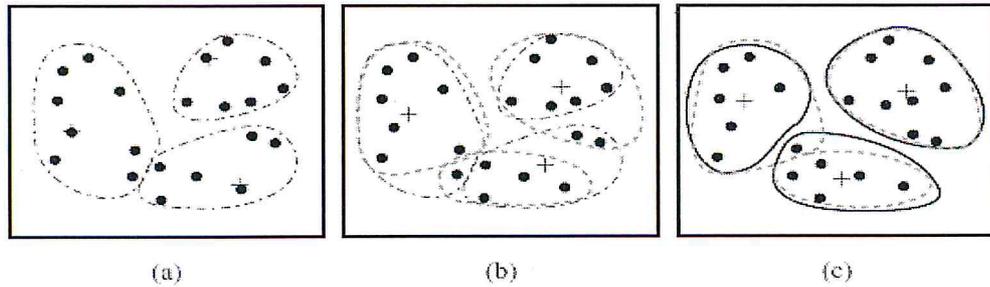


Figure 4.7 : l'algorithme de partitionnement k-means

b) Avantage :

- Relativement extensible dans le traitement d'ensembles de taille importante.
- Simple à appliquer et à comprendre.
- Les objets automatiquement assignés aux clusters.

c) Inconvénients :

- Applicable seulement dans le cas où la moyenne des objets est définie.
- Besoin de spécifier le k nombre de cluster.

4) Machine à vecteur de support :

Machine à vecteur de support (Support Vector Machine SVM) est une méthode de classification supervisée bien adaptée pour les données de grande dimension, introduite par Cortes et Vapnik en 1995, elle est utilisée dans de nombreux problèmes de reconnaissance de forme, catégorisation de texte ou encore diagnostic médical.

Les SVM permet de résoudre des problèmes de discrimination. C'est-à-dire décider à quelle classe appartient un échantillon, ou de régression, c'est-à-dire prédire la valeur numérique d'une variable.

L'algorithme de SVM utilise le mapping non-linéaire pour transformer les données d'apprentissage original à une dimension supérieure. Dans cette nouvelle dimension, il cherche l'hyperplan séparateur optimal linéaire (qui est « la limite de décision », séparant les tuples d'une classe à une autre). Avec le mapping non-linéaire approprié à une assez grande dimension, les données de deux classes peuvent toujours être séparées par un hyperplan. SVM trouve cet hyperplan en utilisant un vecteur de support (les tuples essentielles d'apprentissage) et la marge (définie par le vecteur de support). [H2]

a) Les données sont linéairement séparables :

Un problème de deux classes linéairement séparables, est un problème où un ensemble donné D tel que $(X_1, y_1), (X_2, y_2), \dots, (X_n, y_n)$. Où X_i est un ensemble de tuples d'apprentissage associé au label de classes $y_i \in \{1, -1\}$. Comment trouver un hyperplan séparant les deux classes ? [H2]

SVM cherche un hyperplan avec une marge maximale, chaque hyperplan trouvé peut résoudre le problème et classifier les tuples, mais on s'attend à des hyperplans avec une large marge pour avoir une classification plus précise que les hyperplans avec une petite marge. L'équation de l'hyperplan séparateur avec la marge maximal est :

$$d(X^T) = \sum_{i=1}^N \alpha_i y_i x_i X^T + b$$

Où y_n est le label de classe du vecteur de support x_i , et X^T est le tuple de teste, α_i et b sont des paramètres numériques qui étaient déterminées automatiquement avec l'optimisation. [H2]

b) Données non-linéairement séparable :

Les données non-linéairement séparables sont les données où il n'existe aucune ligne droite qui peut les séparer en différentes classes, mais SVM linéaire peut être étendue pour créer un SVM non-linéaire pour la classification de données non-linéairement séparables. [A4]

Pour étendre SVM on passe par deux étapes principales : dans la 1ere étape, on transforme les données originales en entrées à un espace dimensionnel supérieur en utilisant un mapping non-linéaire. Dans la 2nd étape, on cherche une séparation linéaire avec un hyperplan dans le nouvel espace. Où on va avoir un problème d'optimisation quadratique qu'on peut résoudre en utilisant la formulation linéaire de SVM. La marge maximale trouvée dans le nouvel espace correspond à une séparation non-linéaire avec une hyper-surface dans l'espace original.

- **Kernel trick :**

Le but du kernel est d'éviter le mapping et de faire une transformation à un autre espace dimensionnel supérieur et éviter de faire des opérations de produit $\phi(x_i) \cdot \phi(x_j)$ coûteuses, où $\phi(x)$ est un mapping non-linéaire appliqué pour transformer les tuples d'apprentissage. Au lieu de calculer le produit il est mathématiquement correct de calculer la fonction Kernel. [A4] [H2]

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

Il existe des fonctions Kernel admissibles employées avec le SVM :

Kernel Polynomial de degré h : $K(x_i, x_j) = (x_i \cdot x_j + 1)^h$

Kernel à fonction à base radial : $K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2}$

Kernel Sigmoid: $K(x_i, x_j) = \tanh(kx_i \cdot x_j - \delta)$

- **La formulation de Kernel en SVM:**

Le Kernel est appliqué dans l'équation d'un hyperplan à séparateur maximal :

$$d(X^T) = \sum_{n=1}^N \alpha_n y_n K(x_n, X^T) + b$$

c) **Cas de multi-classes :**

[Wik04] Plusieurs méthodes ont été proposées pour l'étendre sur des cas où plus de deux classes sont séparées. Les deux méthodes les plus connues sont one versus all et one versus one :

- ***One versus all*** : consiste à construire M classificateurs binaires en attribuant le label 1 aux échantillons de l'une des classes et le label -1 à toutes les autres. En phase de teste, le classificateur donnant la valeur de confiance (ex : la marge) la plus élevée remporte le vote.
- ***One versus one*** : consiste à construire $M(M-1)/2$ classificateurs binaires en confrontant chacune des M classes. En phase de teste, l'échantillon à classer est analysé par chaque classificateur et un vote majoritaire permet de déterminer sa classe.

d) **Avantage :**

- Vu que le Kernel implicitement contient une transformation non-linéaire, qui va rendre les données linéairement séparables avec une base théorique robuste et sans le jugement de l'expertise humain.
- Produit des classificateurs très précis.

e) **Inconvénient :**

- Couteuse en calcul.

3. **Etat de l'art :**

Si on veut créer des systèmes informatiques qui peuvent fonctionner d'une façon autonome on a besoin de donner à notre système la capacité à reconnaître les émotions depuis les expressions faciales pour cela plusieurs méthodes de classification sont utilisés dans la littérature.

Nathan C. [N2] a utilisé le réseau de neurone pour la classification des expressions faciales. Le réseau de neurone a en entrée 32X32 pixel carré (qui représente la résolution minimum pour détecter un visage humain selon des recherches précédentes). Ensuite dans les unités cachées où le nombre d'unités variait selon deux conditions. Conditions 1 : 10 unités cachées ont été utilisées. Condition 2 : 5 unités cachées ont été utilisées. Et en sorti les unités de sorties seront pour chaque

expression possible les six catégories des expressions faciales, une unité de sortie est requise, les six unités ont produit des valeurs de 0.0 à 1.0, qui après la mise en échelles sont comparables avec les valeurs des expressions faciales aperçus. Le système final après l'apprentissage et le teste sur 180 images a pu réaliser une moyenne de 73,3%.

Abu sayeed Md. Et al [A3] ont utilisé une approche à multi- détecteur pour localiser les points caractéristiques depuis le contour des composants du visage yeux, bouche, sourcil. Un tableau composé de 11 caractéristiques est ensuite construit en calculant le degré de déplacement de ces 11 points caractéristiques par rapport à l'état neutre. Et finalement les ensembles résultants sont utilisés pour l'apprentissage du classificateur K plus proche voisin pour qu'il puisse classifier les expressions. La méthode a été testée sur une base d'image d'expression faciale et a réaliser un taux de 90.76% de reconnaissance.

Tanvi Sheikh et al [T1] ont proposé une méthode basée sur l'algorithme amélioré de k-means pour la classification des expressions faciales depuis des images frontales du visage. Pour la classification l'algorithme utilise deux caractéristiques : la densité du pixel et le rapport de la hauteur à la largeur des régions recadrées. Le système de reconnaissance comprend l'extraction des caractéristiques et la classification selon les caractéristiques extraites. L'algorithme k-means amélioré va classifier les expressions à l'une des expressions heureux, triste et neutre. La classification des expressions est appliquée sur une base de 200 images (Karolinska Directed Emotional Face Database).

Xue-wen Chen et al. [X1] ont utilisé une méthode d'extraction des caractéristiques basée sur le clustring pour la reconnaissance des expressions faciales et ensuite ils procèdent à la classification comme démontré dans l'organigramme suivant :

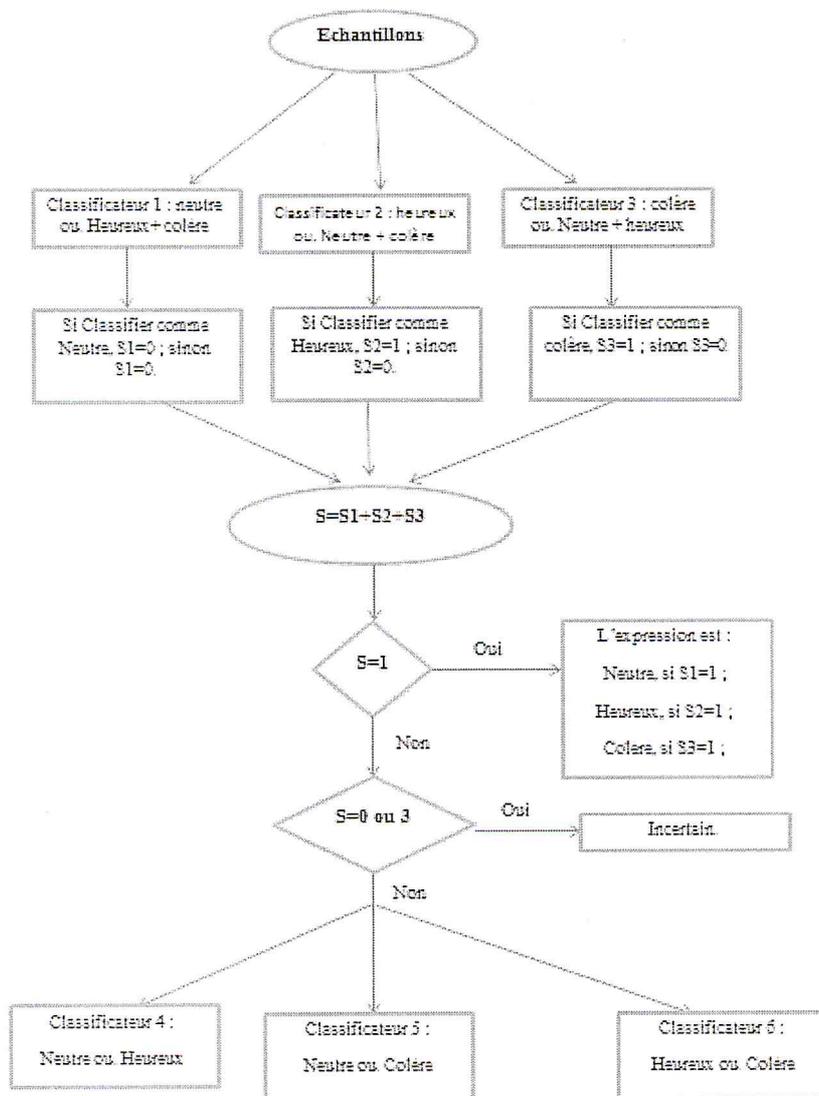


Figure 4.8 : procédure de classification. [X1]

4. Conclusion :

Dans ce chapitre nous avons parlé du DataMining et des différents domaines d'application et les différents tâches et parmi ces tâches la classification avec ces deux types supervisés et non-supervisés et quelques techniques les plus utilisées dans le domaine de la classification.

Chapitre 5 : Les outils de détection du visage et des points caractéristiques

1. Introduction :

Dans le but de détecter les expressions faciales sur le visage et leurs donner une signification il faut d'abord passer par deux étapes très importantes :

- La détection du visage.
- La détection des points caractéristiques.

2. Détection du visage :

Pour la détection du visage plusieurs méthodes ont été décrites dans le chapitre détection du visage parmi ces méthodes on a choisi d'utiliser OpenCv qui est la meilleur méthode open source au niveau de la performance et la vitesse.

2.1. OpenCv :

OpenCv est un détecteur basé sur la méthode Haar de détection de visage développé par Viola et Jones, cette méthode à pour concept que chaque visage frontale à la région des yeux qui est plus sombre que le front et les joues, et la région de bouche est plus sombre que les joues, et ainsi de suite. Elle performe typiquement à peu près 20 étapes de comparaisons comme celle-ci pour décider si c'est un visage ou pas, mais elle doit faire ceci pour tous les positions dans l'image et pour chaque taille possible du visage.

Le détecteur de visage Haar peut être entraîné automatiquement pour trouver des visages depuis un grand nombre d'images, avec les informations enregistrées dans un fichier XML pour qu'il soit utilisé après. Ces classificateurs en cascade [Figure 5.1] sont entraînés typiquement en utilisant au moins 1.000 images de visage unique et 10.000 images qui ne représentent pas des visages (par exemple : images qui représente des arbres, voitures, textes...etc.).

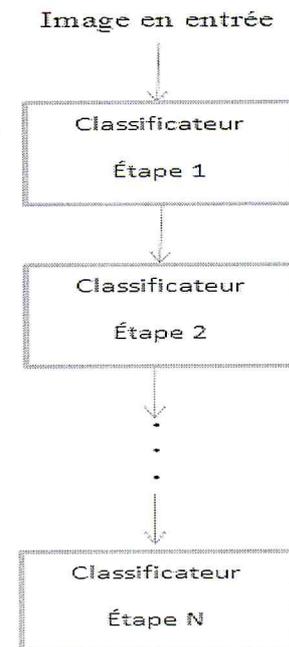


Figure 5.1 : étape de comparaison de Haar

Pour la détection du visage OpenCv utilise les fichiers XML où sont enregistrées les données de l'apprentissage pour distinguer le visage du non-visage dans les images. Il existe plusieurs fichiers XML prédéfinis dans OpenCv, comme :

Tableau 5.1 : fichier XML prédéfini.

Type de classificateur en Cascade	Le nom du fichier XML
Détecteur du visage (par défaut)	haarcascade_frontalface_default.xml
Détecteur du visage (Haar)	haarcascade_frontalface_alt2.xml
Détecteur des yeux (séparer pour droite et gauche)	haarcascade_lefteye_2splits.xml
Détecteur de Bouche	haarcascade_mcs_mouth.xml
Détecteur du nez	haarcascade_mcs_nose.xml
Détecteur d'une personne complète	haarcascade_fullbody.xml

a) Charger le détecteur de Haar pour la détection du visage :

Pour performer la détection du visage il faut charger le fichier XML en utilisant la class d'OpenCv : cascadeClassifier, comme :

```
CascadeClassifier faceDetector;  
faceDetector.load(faceCascadeFilename);
```

Pour utiliser le fichier XML chargé, il faut avoir une image ou un flux vidéo et pour avoir le flux vidéo il faut accéder à la webcam

b) Accéder à la webcam :

Pour récupérer l'image depuis la webcam de l'ordinateur ou un fichier vidéo, on utilise la fonction VideoCapture::open() d'OpenCv, avec le numéro de la webcam ou le nom du fichier vidéo, et puis utiliser le Stream operator de C++ pour récupérer l'image.

c) Traitement d'image :

Avant de détecter le visage avec le classificateur chargé on doit faire quelque traitement initial à l'image de la caméra en appliquant les étapes suivantes :

- **Convertir la couleur aux niveaux de gris (Grayscale)** : la détection du visage marche seulement avec les images en niveaux de gris donc on doit convertir les images en couleur RGB en utilisant la fonction `cvtColor()`.
- **Réduire l'image de la caméra** : la vitesse de détection de visage dépend de la taille de l'image (lent avec de grandes images et rapide avec de petite image), la réduction est faite avec la fonction `resize ()`.
- **Egalisation de l'histogramme** : la détection du visage n'est pas fiable dans les conditions à luminosité faible. Donc on doit effectuer une égalisation de l'histogramme pour améliorer le contraste et la luminosité en utilisant la fonction `equalizeHist()`.

d) Détecter le visage :

Une fois l'image converti aux niveaux de gris, la taille de l'image réduite, et l'histogramme l'égalisé, l'image est prête pour la détection du visage en utilisant la fonction `CascadeClassifier::detectMultiScale()`. Et le résultat sera un visage détecté et encadré avec un carré [Figure 5.2].

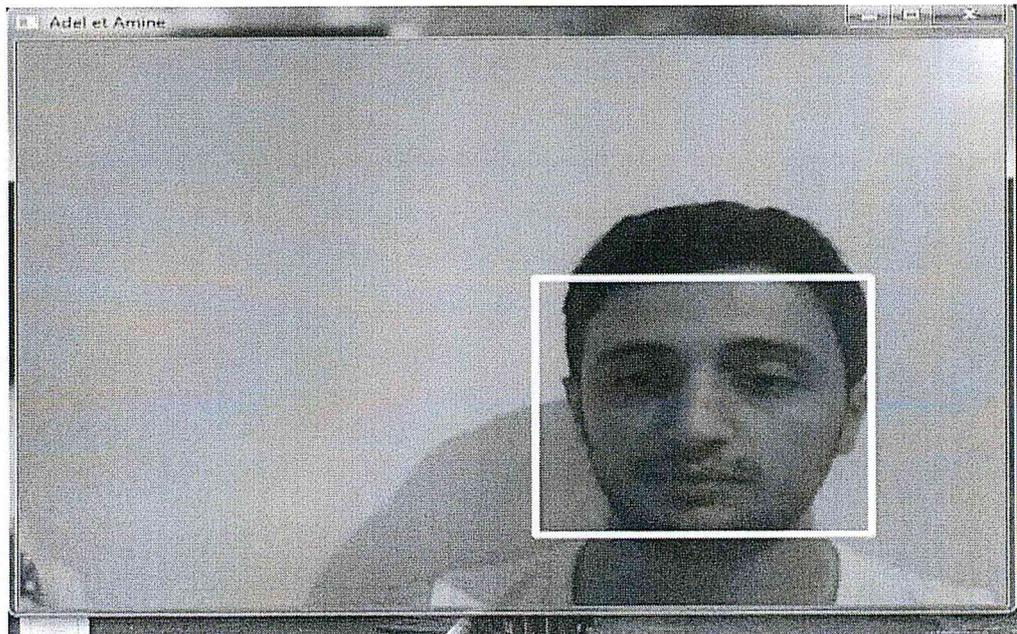


Figure 5.2 : Rectangle de détection du visage.

On passe par les mêmes étapes d'OpenCv pour détecter les composantes du visage (yeux, bouche,...etc.) [Figure 5.3].

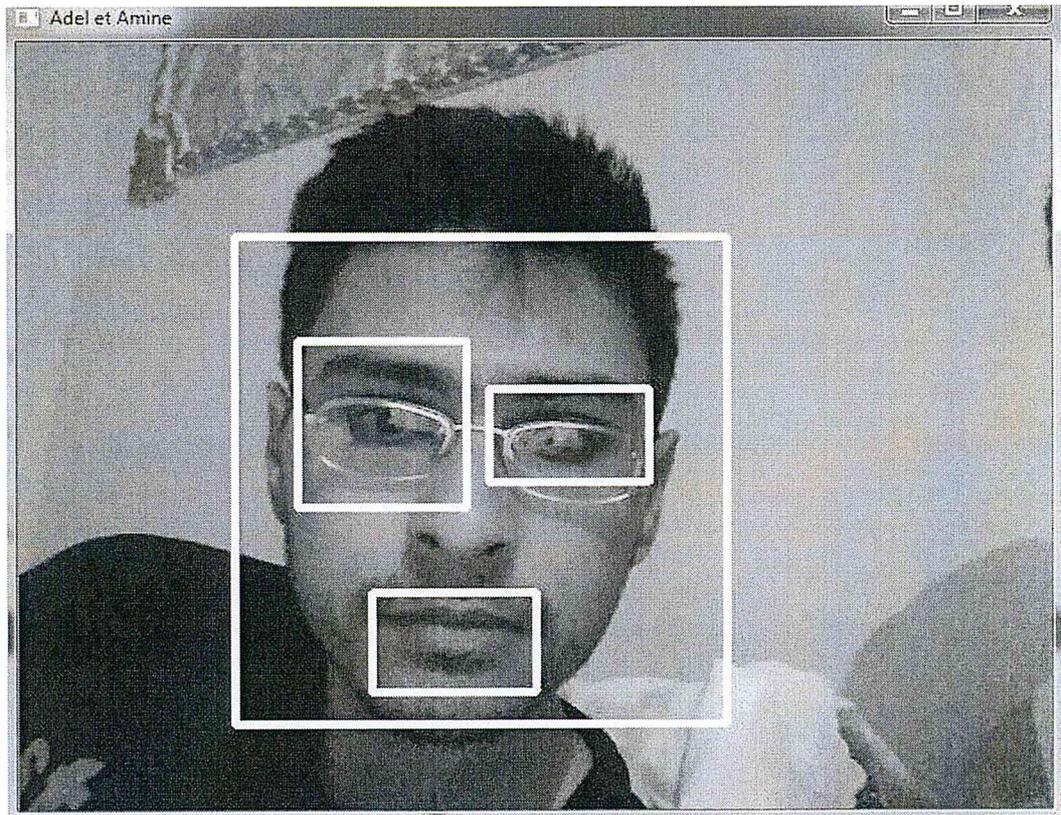


Figure 5.3 : détection des composantes du visage avec OpenCv.

3. Détection des points caractéristiques du visage :

Une fois le visage détecté avec OpenCv, on va procéder à la détection des points caractéristiques du visage les Landmarks. Pour cela on va se baser sur ASM (Active Shape Model) implémenté en utilisant la bibliothèque incluse dans STASM.

Après l'apprentissage du ASM en utilisant un ensemble d'images manuellement Landmarké générant un fichier de type (.Shape) existant dans STASM. On peut utiliser l'ASM pour rechercher les caractéristiques dans un visage [Figure 5.4].

Pour rechercher les caractéristiques dans un visage ASM on utilise deux modèles :

- 1) ***Le modèle de profile :*** pour chaque point caractéristique décrit les caractéristiques de l'image autour du LandMark. Durant l'apprentissage, une prise d'échantillon est effectuée dans la zone autour de chaque LandMark à travers tous les images d'apprentissages pour construire un modèle de profile pour les LandMark. Et durant la recherche, une prise d'échantillon est

effectuée dans la zone de voisinage de chaque tentative de LandMark et on bouge le LandMark qui correspond le plus au LandMark du modèle de Profile, ce qui génère la nouvelle tentative de position du LandMark nommé *la forme suggérée (suggested shape)*.

- 2) *Le modèle de forme* : qui définit les positions relatives permises des LandMarks. Durant la recherche, le modèle de forme ajuste la forme suggérée par le modèle de profil pour qu'elle soit conforme à une forme légitime du visage.

L'algorithme suivant fait des itérations pour une solution en utilisant les deux sous-modèles (le modèle de profile et de forme) pour donner en résultat les coordonnées (x, y) des LandMarks du visage.

Algorithme LandMark

Entré image du visage

- 1) Générer la forme de départ en localisant la position d'ensemble du visage.
 - 2) Répéter
 - a. Pour chaque point de forme
 - b. Pour chaque zone autour du point
 - Construit un profile par l'échantillonnage de la région autour de la zone.
 - Mesurer les profiles qui vont avec le modèle de profile
 - Bouger le point à la zone qui a la meilleure correspondance au profile.
 - Ajuster la forme suggérée pour qu'elle soit conforme avec le modèle de forme.
 - 3) Jusqu'à la convergence (jusqu'à aucune amélioration dans l'ajustement n'est possible)
- Sortie*** les coordonnées (x, y) des LandMarks.



Figure 5.4 : les points caractéristiques détectés

Chapitre 6 : Nos Démarches De Reconnaissance

1. Introduction :

Dans la reconnaissance des expressions faciales il existe une variété de méthodes et de démarches dans la littérature ces méthodes varient que ce soit dans la détection du visage la détection des points caractéristiques et la classification, aussi il existe plusieurs type de données en entrées pour effectuer la classification, dans nos démarches on va faire une étude comparative entre deux types de données en entrée (les coordonnées des points caractéristiques et les relations spatiales) et deux méthodes de classification une méthode naïve et simple qui est le k-means et une autre efficace et largement utilisée et qui donne de très bons résultats qui est la machine à vecteur de support.

2. Les paramètres en entrées :

Pour la reconnaissance des expressions faciales plusieurs paramètres ont été utilisés dans la recherche (par exemple : Flux Optique [Y1], Les relations spatiales [C2], les positions des points caractéristiques [B2]). Pour notre démarche nous avons choisi comme paramètre les positions des points caractéristiques et aussi les relations spatiales entre les points caractéristiques.

2.1. Les positions des points caractéristiques :

Plusieurs publications dans le domaine de la reconnaissance des expressions faciales ont utilisé la position des points caractéristiques, un certain nombre de points caractéristiques ont été sélectionnés pour chaque publication (22 points caractéristiques pour [B2], 34 points caractéristiques pour [Z1]).

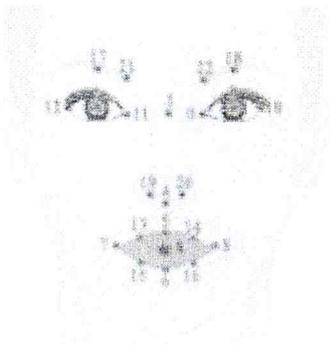


Figure 6.1 : 22 points caractéristiques détecter [B2]



Figure 6.2 : 34 points caractéristiques utiliser [Z1]

Pour notre application une fois le visage et les points caractéristiques détectés avec OpenCV et une méthode basée sur ASM, on a procédé à des testes pour sélectionner les points caractéristiques qui entrent dans chaque expression faciale, les testes ont été appliqués sur la base de Cohn et Kanade [C3]. Pour confirmer la fiabilité des testes effectués on note que la méthode de détection des points caractéristiques affecte le même point à la même position dans le visage en cas de rotation du visage et en cas de changement d'acteur (Figure 6.3).

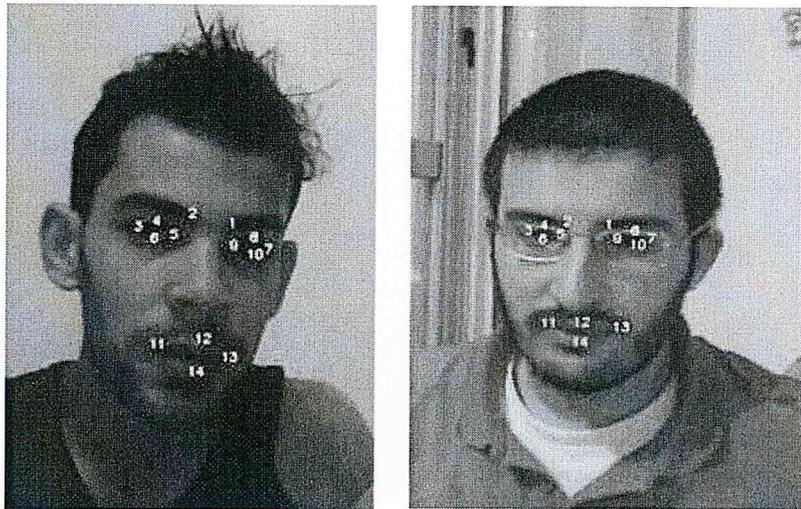


Figure 6.3 : position des points caractéristiques pour différents acteur et différents rotation du visage

Les points caractéristiques sont détectés même dans le cas de l'existence d'un obstacle qui couvre une petite partie du visage (Figure 6.4), mais dans le cas où la couverture du visage ne permet pas la reconnaissance de l'expression tous le visage sera rejeté. La détection des points caractéristiques avec l'existence d'un obstacle est un domaine de recherche très vaste (par exemple : il existe des publications qui utilisent l'historique de la position des points caractéristiques pour les détecter en cas d'obstacle) pour notre cas la méthode basée sur ASM utilise les points caractéristiques d'apprentissage pour pouvoir détecter les points caractéristiques même avec l'existence d'un obstacle en se rapprochant des points les plus proches des points d'apprentissage.



Figure 6.4 : points caractéristique détecté avec l'existence d'un obstacle

Pour s'assurer que les points caractéristiques détectés peuvent être appliqués sur n'importe quelle taille d'image (resp. flux vidéo). Et pour pouvoir ensuite faire une comparaison entre les positions qui changent une normalisation des coordonnées est obligatoire pour négliger la différence de taille des images, pour le faire on va récupérer les coordonnées des points caractéristiques par rapport à la taille du visage et donc ignorer le reste de l'image qui sera réalisée par la sélection d'un point repère $Pr(gX, gY)$ qui sera dans le coin haut gauche du visage et depuis ce point on calcul les coordonnées des points caractéristiques.



Figure 6.5 : point repère

Pour avoir les coordonnées de chaque point caractéristique on prend ces coordonnées par rapport à l'image et on les soustrait aux coordonnées du point repère. Par exemple :

$P1(X, Y)$: point par rapport à l'image totale.

$Pr(gX, gY)$: le point repère.

$P1'(X-gX, Y-gY)$: le point par rapport au visage.

2.1.1. Teste de sélection des points caractéristiques :

Pour pouvoir sélectionner les points caractéristiques qui entrent dans chaque expression faciale on a pris trois images dans chaque expression (Joie, colère, surprise, dégoût, peur, triste, neutre) et on calcule le changement de position de chaque point des 68 points détectés avec la méthode basée sur ASM, par rapport aux autres expressions, et enfin on fait une sélection des points qui atteignent le seuil de changement (tester à 0.007) et qui ont un changement significatif dans toutes les expressions faciales.

2.1.1.1. Implémentation de l'algorithme de teste:

Pour sélectionner les points caractéristiques on implémente l'algorithme suivant :

Algorithme changement PosPoint

Entrer les 68 points de chaque acteur de chaque expression.

1) Pour chaque Point P_i (tq : $i=1 \dots 68$)

a. Pour chaque acteur A_j (tq $j=1 \dots 3$)

i. Pour chaque Expression E_k (tq $k=1 \dots 7$)

1. Si le point $P_i^{[A_j]}[E_k] >$ seuil de changement
(seuil=0.007) ***et*** $P_i^{[A_j]}[E_k] \neq P_i^{[A_j]}[E_{k+1}]$

$Comp[P_i]++$;

Fin Si

Fin Pour

Fin Pour

Fin Pour

Fin



Figure 6.6 : Les points caractéristiques détecter avec la méthode basée sur ASM

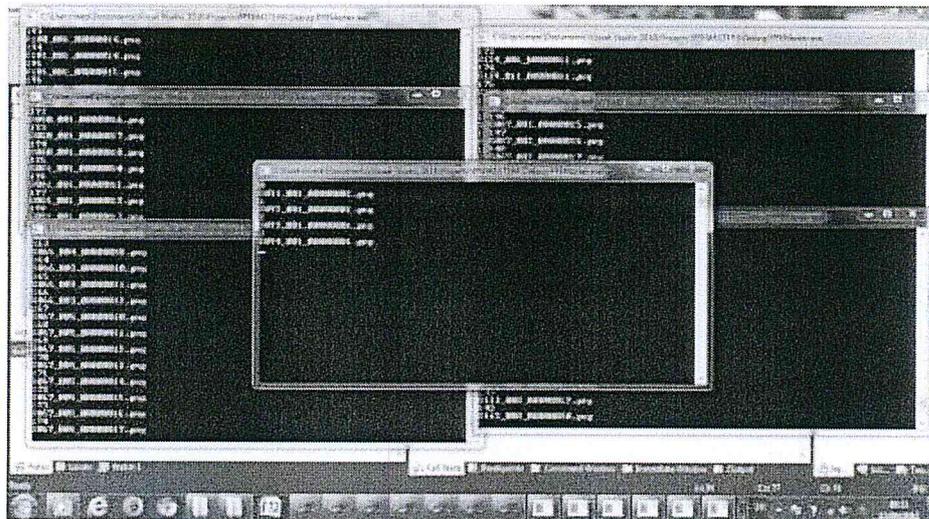


Figure 6.7 : implémentation de l'algorithme de teste

2.1.1.2. Les résultats des testes :

Une fois le teste appliqué sur les 68 points caractéristiques détectés avec la méthode basée sur ASM, un ensemble de points qui ont atteint le seuil de changement établit et ont changé de position pour presque toutes les expressions faciales ont été sélectionnées (Tableau 6.1) (Figure 6.8).

Tableau 6.1 : points caractéristiques

ID POINT	Nombre d'apparition
1	118
2	118
3	120
4	118
5	108
6	114
7	112
8	108
9	108
10	112
11	110
12	114
13	122
14	120
15	116
16	114
17	118
18	118
19	116
20	118
21	122
22	120
23	118
24	104
25	214
26	108
27	120
28	124
29	120
30	118
31	114
32	118
33	118

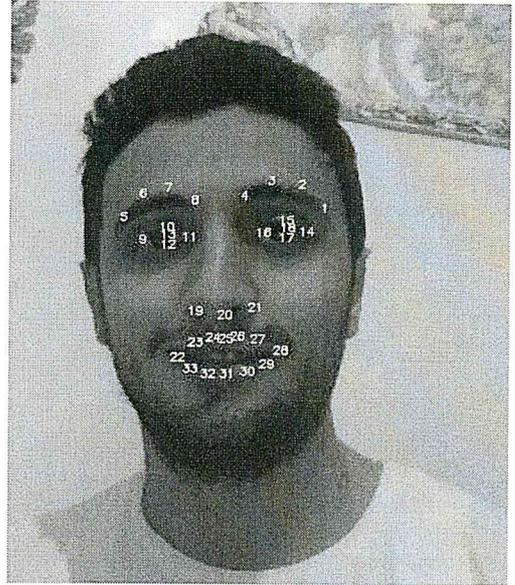


Figure 6.8 : les 33 points caractéristiques sélectionné

Depuis ces résultats ont a conclu que 33 points caractéristiques changent fréquemment de position avec chaque expression faciale.

2.2. Les relations spatiales entre les points caractéristiques :

Les relations spatiales entre les points caractéristiques ont été utilisées dans plusieurs publications, plusieurs combinaisons de relations spatiales ont été réalisées pour pouvoir capter au mieux les expressions projetées par le visage ([C2] ont a utilisé 6 relations spatiales [Figure 6.9], 4 relations spatiales dans [P3] [Figure 6.10]).

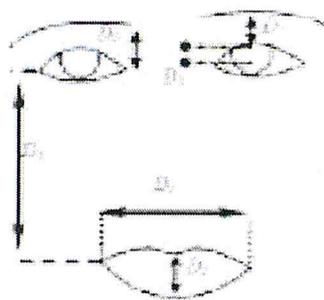


Figure 6.9 : les 6 relations spatiales calculées [C2]

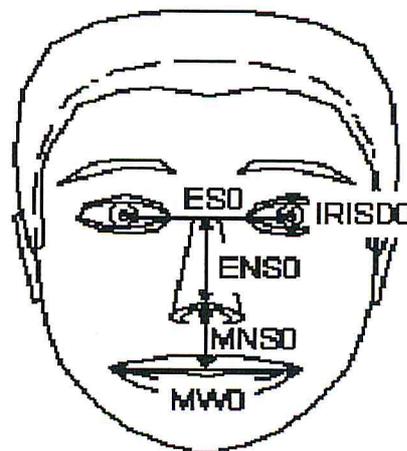


Figure 6.10 : les 4 relations spatiales calculées [P3]

Pour notre application et après que le visage et les points caractéristiques détectés avec OpenCv et la méthode basée sur l'ASM, et nous avons procédé à des testes pour décider des combinaisons des relations spatiales qui entrent dans les expressions faciales et qui aident à déterminer une expression faciale.

Pour confirmer la fiabilité des testes et décider des combinaisons des relations spatiales à calculer, les notes décrites dans la partie positions des points caractéristiques s'appliquent sur cette partie et de plus une normalisation des relations spatiales calculées entre les points caractéristiques pour pouvoir ignorer la taille du visage et diviser soit sur la hauteur gH où la largeur gW du visage (si la

relation spatiale est verticale on divise sur gH sinon si la relation spatiale est horizontale on divise sur gW) pour enfin avoir des relations spatiales normalisées pour toutes les tailles du visage et pour toutes les distances de la Webcam.

2.2.1. Méthode de calcul des relations spatiales :

Comme on l'a déjà mentionné deux types de relations spatiales sont calculés : verticales et horizontales.

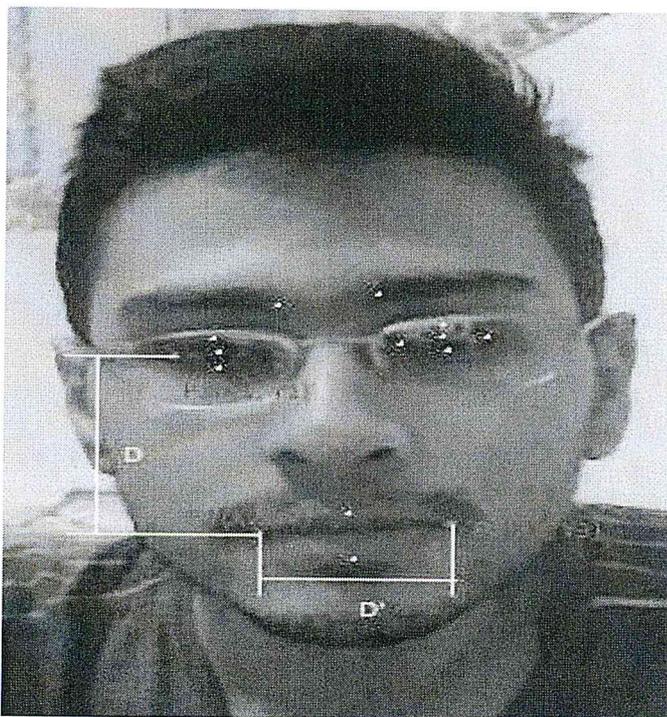


Figure 6.11 : les coordonnées des points caractéristiques pour le calcul des relations spatiales

Par exemple pour calculer les relations spatiales affichées dans la (Figure 6.11).

- Relation spatiale vertical : pour calculer la relation spatiale D on prend l' Y_2 du point P_2 moins l' Y_1 du point P_1 et le résultat on le divise par la hauteur du visage gH pour avoir une relation spatiale normalisée à n'importe quelle taille du visage dans l'image et n'importe quelle distance du visage de la Webcam.

- Relation spatiale Horizontal : pour calculer par exemple la relation spatiale D' on prend le X3 du P3 moins le X2 du P2 et on divise le résultat sur la largeur gW du visage pour avoir une relation spatiale normalisée à n'importe quelle taille du visage dans l'image et n'importe quelle distance du visage de la Webcam.

2.2.2. Teste de sélection des relations spatiales:

Pour sélectionner les relations spatiales qui changent avec l'actionnement d'une expression faciale, on a testé sur 3 images de chaque expression faciale (Joie, colère, surprise, dégoût, peur, triste, neutre) et on a calculé les relations spatiales qui changent entre chaque expression et le reste des expressions pour enfin décider d'une combinaison de relation spatiale qui représente toutes les expressions faciales. Pour le faire on prend les 68 points caractéristiques détectés avec la méthode basée sur ASM et on calcul toutes les relations spatiales possibles entre ces points et on sélectionne qui dépasse le seuil de changement (0,0035) et qui apparait le plus dans le changement entre toutes les expressions.

2.2.2.1. Implémentation de l'algorithme de teste:

Pour sélectionner les relations spatiales qui changent avec le changement des expressions faciales l'algorithme suivant est implémenté :

Algorithme RelationSP

Entrer 68 points caractéristiques de chaque acteur de chaque expression

- 1) Pour*** chaque point P_i (tq $i=1 \dots 68$)
 - a. Pour*** chaque Acteur A_j (tq $j=1 \dots 3$)
 - i. Pour*** chaque Expression E_k (tq $k=1 \dots 7$)
 - 1. Si*** la relation spatiale $D(P_i^{[A_j]}[E_k], P_{i+1}^{[A_j]}[E_k]) >$ seuil de changement (seuil=0,0035) ***et*** $D(P_i^{[A_j]}[E_k], P_{i+1}^{[A_j]}[E_k]) \neq D(P_i^{[A_j]}[E_{k+1}], P_{i+1}^{[A_j]}[E_{k+1}])$
 $Cmp[D(P_i^{[A_j]}[E_k], P_{i+1}^{[A_j]}[E_k])]++;$
Fin si

Fin Pour

Fin Pour

Fin Pour

Fin

Sortie: la liste des relations spatiales qui changent le plus entre les expressions.

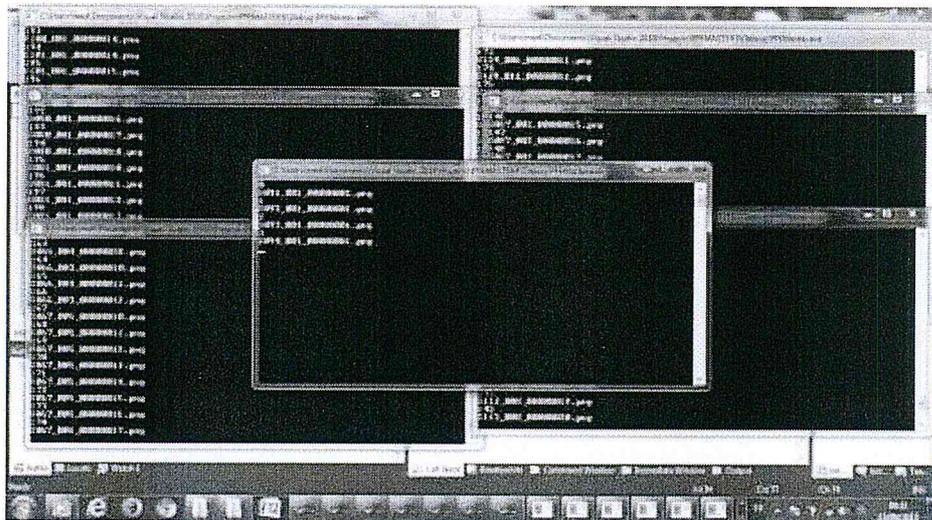


Figure 6.12 : Implémentation de l'algorithme de teste

2.2.2.2. Les résultats des testes :

Une fois les testes appliqués 13 relations spatiales se sont révélées être les relations spatiales qui apparaissent le plus et changent le plus entre les expressions faciales. (Figure 6.13) Parmi ces 13 relations spatiales il existe des relations spatiales qui représentent les mêmes déformations et donc on a procédé à une sélection pour éliminer les redondances (Tableau 6.2). Depuis cette sélection 6 relations spatiales ont été retenues (Figure 6.14).

Tableau 6.2 : relations spatial sélectionné

relations spatiales	nombre d'apparition	sélection
D1	119/126	oui
D2	122/126	non
D3	124/126	oui
D4	122/126	non
D5	115/126	non
D6	122/126	oui
D7	115/126	non
D8	114/126	non
D9	116/126	non
D10	118/126	oui
D11	112/126	non
D12	105/126	oui
D13	115/126	oui

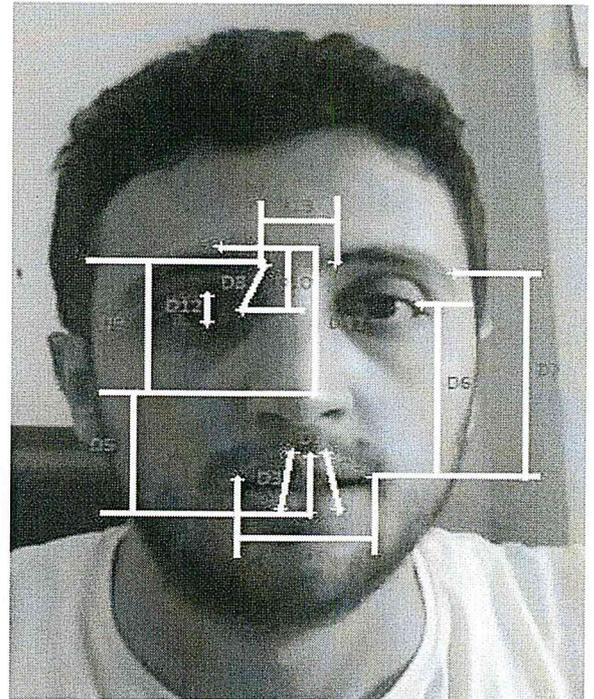


Figure 6.13 : les 13 relations spatiales

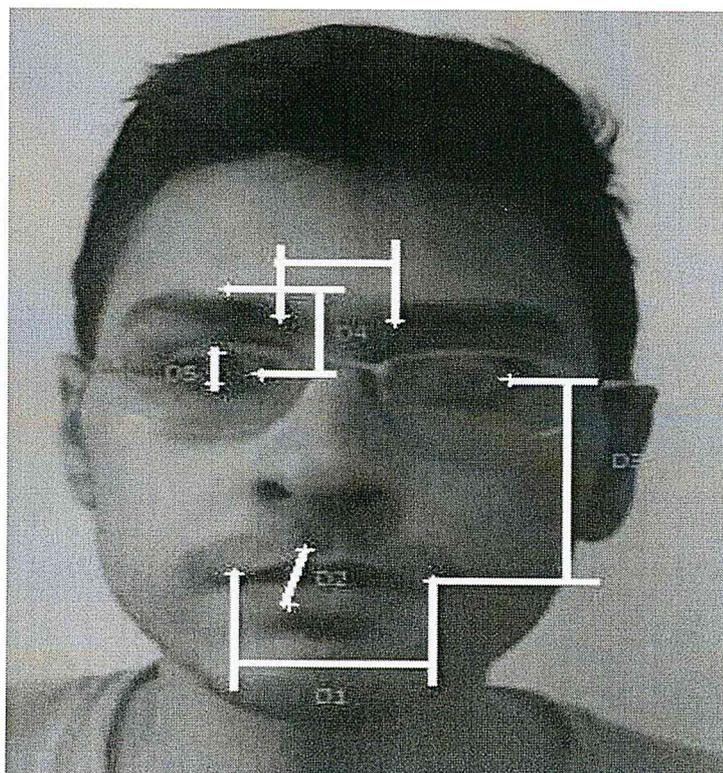


Figure 6.14 : les 6 relations spatiales sélectionnées

3. Nos démarches de classification :

Il existe plusieurs méthodes utilisées dans la classification des données récoltées des expressions faciales qui permet de classifier les expressions faciales (Chapitre Classification).

Pour notre application, on a procédé à une étude comparatif entre deux méthodes, une méthode simple et naïve le K-means, et une méthode efficace et largement utilisée dans la classification machine à vecteur de support (Support Vector Machine SVM). Ces deux méthodes ont été testées avec types de données, les positions des points caractéristiques pour chaque expression, et les relations spatiales entre les points caractéristiques pour chaque expression faciale (calculer dans la partie précédente).

3.1. Base d'image pour les apprentissages et les testes :

Pour faire les apprentissages et les testes on a utilisé la base d'image de Cohn est kanade [C3] qu'on a divisé en deux parties, une partie pour l'apprentissage et l'autre pour les testes. Il existe plusieurs méthodes de division des bases pour l'apprentissage et le teste (cross validation [E5]), mais pour notre cas nous avons procédé à une division en deux pour à la fois superviser les images d'apprentissage et les diviser en dossier séparés par expression faciale.

Pour notre étude nous avons évité de travailler avec la base d'image JAFFE [J1] qui contient des images de femmes japonaises et qui par la morphologie de leurs visages et les expressions qui ne correspondent pas vraiment au visage Nord-Africain et donc les résultats obtenus par les testes sur cette base ne reflètent pas les expressions du visage de cette région du monde.

3.2. K-means :

L'algorithme K-means est le plus populaire et le plus simple des algorithmes de classification non supervisée (clustering). Il regroupe les données en entrée dans un nombre spécifique d'ensemble (cluster).

L'algorithme K-means sera appliqué et testés sur deux types de données : les coordonnées des 33 points caractéristiques sélectionnées, et aussi sur 6 relations spatiales pour chaque expressions et enfin on va comparer les résultats obtenus.

3.2.1. K-means avec les coordonnées des 33 points caractéristiques :

L'algorithme K-means sera appliqué sur la base d'apprentissage pour la création des clusters et ensuite il sera testé avec la base de teste.

Pour débiter avec la création des clusters, une image de chaque expression va être sélectionnée aléatoirement pour ensuite détecter les coordonnées des 33 points caractéristiques [Figure 6.7] dans chaque image et ces coordonnées seront les centroids de départ de chaque point de chaque expression. Ensuite l'algorithme va regrouper le reste des coordonnées des autres images de la base d'apprentissage dans les clusters en calculant la distance Euclidienne avec les centroids et l'image va appartenir au cluster de l'expression qui a le plus grand nombre de points les plus proches au centroid, et une fois tous les coordonnées des images regroupées, les centroids seront recalculés (la moyenne de tous les éléments du cluster) et tous les points vont être testés avec les nouveaux centroid et affectés au cluster qui a le centroid le plus proche jusqu'à ce qu'il n'y aura aucun changement.

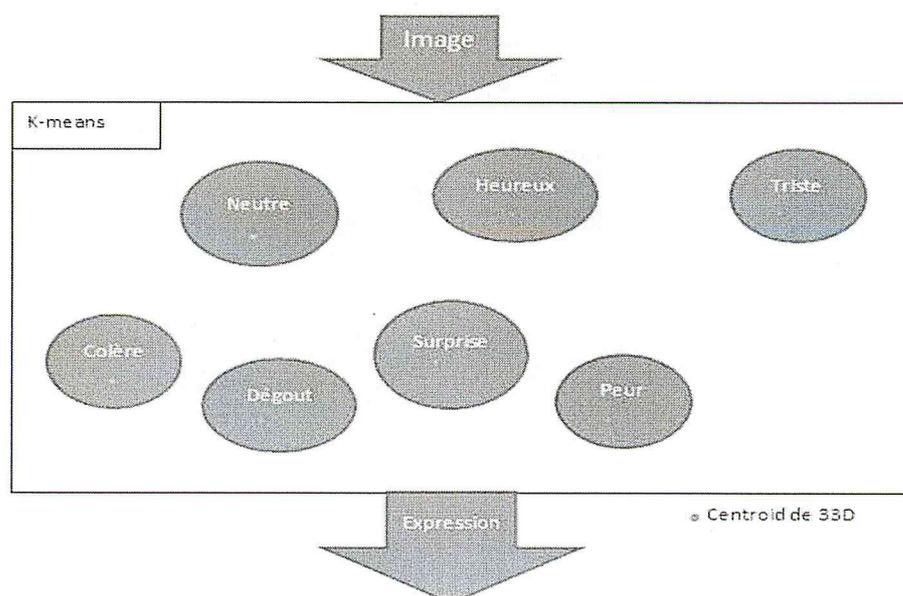


Figure 6.15: Architecture du K-means des points caractéristiques.

Algorithme K-means

C-colère: Centroid de 33 dimensions [33][2]

C-dégout: Centroid de 33 dimensions [33][2]

C-peur: Centroid de 33 dimensions [33][2]

C-joie: Centroid de 33 dimensions [33][2]

C-neutre: Centroid de 33 dimensions [33][2]

C-triste: Centroid de 33 dimensions [33][2]

C-surprise: Centroid de 33 dimensions [33][2]

Début

- 1) Prendre une Image de chaque expression aléatoirement et initialiser les 7 Centroid en affectant les 33 points (X, Y) de l'image.
- 2) Pour chaque expression
 - 2.1) pour chaque Image I_k ($k=1 \dots n$)
 - 2.1.1) pour chaque point P_j ($j=1, \dots, n$)
 - calculer la distance Euclidienne $D_E(P_j, C_n[j])$
 - affecter au cluster qui a le plus grand nombres de points proches au centroid.
 - si une image I_k est ajoutée au cluster
 - recalculer le centroid.
 - refaire 2.1).
- 3) jusqu'à ce qu'il n'y aura aucun changement.

Fin.

- **Résultats des testes :**

Pour comparer les résultats nous l'avons appliqué sur les coordonnées des 33points qu'on a sélectionnés, et les coordonnées des 14points caractéristiques utilisés dans la littérature, pour avoir une idée s'il existe des améliorations apportées aux résultats par rapport aux 14 points [S2].

Tableau 6.3 : résultats des testes avec les 33points caractéristiques.

	neutre	colère	dégout	Peur	Heureux	Triste	Surprise	Ambiguïté	Taux d'erreur	Taux de Bonne Apparition
neutre	8/110	27/110	30/110	21/110	4/110	6/110	1/110	13/110	89/110	8/110
Colère	29/406	25/406	145/406	157/406	11/406	31/406	3/406	5/406	376/406	25/406
Dégout	3/383	34/383	252/383	42/383	12/383	7/383	1/383	32/383	99/383	252/383
Peur	67/409	21/409	116/409	120/409	67/409	7/409	12/409	64/409	225/409	120/409
Heureux	1/402	14/402	156/402	108/402	82/402	9/402	0/402	32/402	288/402	82/402
Triste	22/401	105/401	111/401	70/401	6/401	21/401	18/401	48/401	332/401	21/401
Surprise	4/410	30/410	148/410	52/410	8/410	10/410	113/410	45/410	252/410	113/410
								9,48%	65,88%	24,63%
								Taux total d'erreur	75. 36%	

Tableau 6.4 : résultats des testes avec les 14 points caractéristiques.

	neutre	colère	dégout	Peur	Heureux	Triste	Surprise	Ambiguïté	Taux d'erreur	Taux de Bonne Apparition
neutre	7/110	19/110	24/110	29/110	4/110	7/110	6/110	14/110	89/110	7/110
Colère	42/406	9/406	139/406	89/406	14/406	61/406	14/406	38/406	359/406	9/406
Dégout	21/383	23/383	106/383	52/383	18/383	36/383	26/383	101/383	176/383	106/383
Peur	7/409	13/409	65/409	105/409	50/409	29/409	53/409	87/409	217/409	105/409
Heureux	17/402	33/402	32/402	91/402	69/402	34/402	17/402	109/402	224/402	69/402
Triste	18/401	43/401	89/401	69/401	13/401	47/401	42/401	80/401	274/401	47/401
Surprise	9/410	22/410	133/410	46/410	18/410	20/410	96/410	66/410	248/410	96/410
								19,63%	62,95%	17,41%
								Taux total d'erreur	82,58%	

Depuis les résultats des testes on a conclu qu'il existe deux seuils d'erreurs qu'on prend en considération :

4. Des erreurs de mauvaise classification : (affecter une expression au mauvais cluster (expression)).
5. Ambiguïté : ne pas pouvoir classifier une expression (une expression a deux cluster avec le même taux)

Et un autre type d'erreur qu'on n'a pas pu étudier c'est les expressions étrangères (d'autres expressions que les 7 expressions de base) dû à la limitation de la base d'image utilisée aux 7 expressions universelle.

Depuis la comparaison des deux tableaux [Tableau 6.3] [Tableau 6.4] on remarque que le taux d'erreur dans la méthode avec les 33 points caractéristiques est supérieur au taux d'erreur dans la méthode avec les 14 points caractéristiques, mais le taux d'ambiguïté dans cette dernière est plus élevé que dans la méthode avec les 33 points caractéristiques, d'où on déduit que les 33 points caractéristiques réduisent l'ambiguïté entre les expressions faciales.

3.2.2. *K-means avec les relations spatiales entres les points caractéristiques :*

L'algorithme K-means sera appliqué sur la base d'apprentissage pour la création des clusters et ensuite il sera testé avec la base de teste.

Pour débiter avec la création des clusters, une image de chaque expression va être sélectionnée aléatoirement pour ensuite calculer les 6 relations spatiales dans chaque image [Figure 6.14]. Ces relations spatiales seront les centroids de départ de chaque relation spatiale de chaque expression. Ensuite l'algorithme va regrouper le reste des relations spatiales des autres images de la base d'apprentissage dans les clusters en calculant la distance Euclidienne avec les centroids et l'image va appartenir au cluster de l'expression qui a le plus grand nombre de relations spatiales les plus proches au cluster, et une fois toutes les relations spatiales des images regroupées, les centroids seront recalculés (la moyenne de tous les éléments du cluster) et toutes les relations spatiales vont être testées avec les nouveaux centroid et l'image va être

affectée au cluster a qui le plus grand nombre de relations spatiales sont prêtes et répéter jusqu'à ce qu'il n'y aura aucun changement.

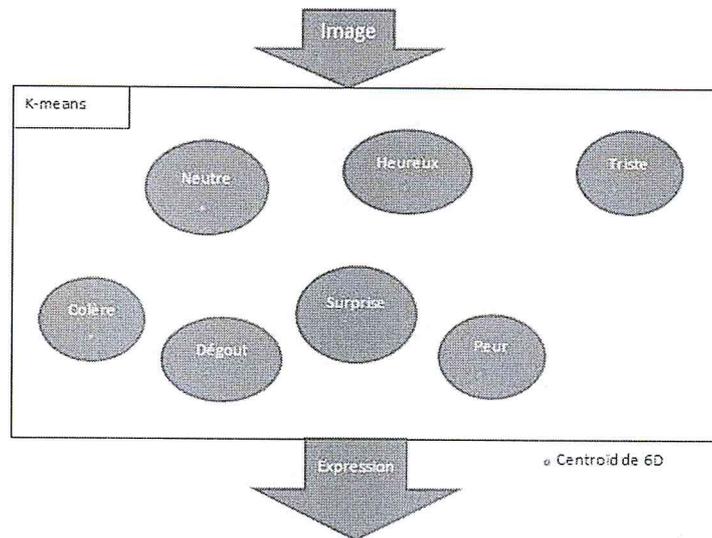


Figure 6.16 : Architecture du K-means pour les relations spatiales.

Algorithme K-means

C-colère: Centroid de 6 dimensions [6][2]

C-dégout: Centroid de 6 dimensions [6][2]

C-peur: Centroid de 6 dimensions [6][2]

C-joie: Centroid de 6 dimensions [6][2]

C-neutre: Centroid de 6 dimensions [6][2]

C-triste: Centroid de 6 dimensions [6][2]

C-surprise: Centroid de 6 dimensions [6][2]

Début

- 1) Prendre une Image de chaque expression aléatoirement et initialiser les 7 Centroid en affectant les 6 relations spatiales de l'image.
- 2) Pour chaque expression
 - 2.1) pour chaque Image I_k ($k=1 \dots n$)
 - 2.1.1) pour chaque relation spatiale R_j ($j=1, \dots, n$)
 - calculer la distance $D_E(R_j, C_n[j])$
 - affecter au cluster qui à le plus grand nombres de relations spatiales proches au centroid.
 - si une image I_k est ajoutée au cluster
 - recalculer le centroid.
 - refaire 2.1).
- 3) jusqu'à ce qu'il n'y aura aucun changement.

Fin.

- **Résultats des testes :**

Pour comparer les résultats nous l'avons appliqué sur les 6 relations spatiales qu'on a sélectionnées, et les relations spatiales utilisées dans la littérature, pour avoir une idée s'il existe des améliorations apportées aux résultats par rapport aux 5 relations spatiales [K4] .

Tableau 6.5 : résultats des testes avec les 6 relations spatiales.

	neutre	colère	dégout	Peur	Heureux	Triste	Surprise	Ambiguïté	Taux d'erreur	Taux de Bonne Apparition
neutre	7/110	14/110	25/110	6/110	3/110	7/110	6/110	42/110	61/110	7/110
Colère	45/406	55/406	136/406	22/406	9/406	53/406	4/406	82/406	269/406	55/406
Dégout	7/383	67/383	85/383	31/383	7/383	6/383	9/383	171/383	127/383	85/383
Peur	2/409	51/409	40/409	128/409	14/409	0/409	12/409	162/409	119/409	128/409
Heureux	5/402	36/402	25/402	125/402	42/402	2/402	6/402	161/402	199/402	42/402
Triste	16/401	67/401	119/401	29/401	8/401	28/401	13/401	121/401	252/401	28/401
Surprise	3/410	26/410	108/410	40/410	27/410	8/410	32/410	166/410	212/410	32/410
								35,89%	49,14%	14,83%
Taux total d'erreur								85,03%		

Tableau 6.6 : résultats des tests avec les 5 relations spatiales.

	neutre	colère	dégout	Peur	Heureux	Triste	Surprise	Ambiguïté	Taux d'erreur	Taux de Bonne Apparition
neutre	8/110	12/110	20/110	6/110	9/110	17/110	1/110	37/110	65/110	8/110
Colère	14/406	14/406	90/406	12/406	39/406	157/406	1/406	79/406	313/406	14/406
Dégout	8/383	32/383	137/383	27/383	16/383	47/383	0/383	116/383	130/383	137/383
Peur	1/409	41/409	72/409	109/409	49/409	10/409	4/409	123/409	177/409	109/409
Heureux	1/402	16/402	128/402	55/402	54/402	8/402	1/402	139/402	209/402	54/402
Triste	12/401	41/401	117/401	27/401	28/401	43/401	6/401	127/401	231/401	43/401
Surprise	4/410	31/410	101/410	64/410	45/410	8/410	26/410	131/410	253/410	26/410
								29,82%	54,66%	15,51%
Taux total d'erreur								84,48%		

Depuis la comparaison des deux tableaux [Tableau 6.5] [Tableau 6.6] on remarque que le taux d'erreur dans la méthode des 6 relations spatiales est inférieur au taux

d'erreur dans la méthode avec les 5 relations spatiales, mais le taux d'ambiguïté dans cette dernière est inférieur dans la méthode avec les 6 relations spatiales.

3.2.3. Comparaison des deux méthodes (relations spatiales, coordonnées des points caractéristiques) :

D'après les résultats des deux méthodes [Tableau 6.3] [Tableau 6.5], on remarque que la méthode de calcul avec les coordonnées spatiales qui a le taux d'erreur le plus bas 75.36% est donc la plus adaptée à travailler avec l'algorithme K-means.

Ces résultats avec des taux d'erreur très élevés sont dû au rapprochement des données en entrées pour chaque expression et donc le système effectue une mauvaise classification et donc de mauvais résultats et pour cela on a pensé à une méthode qui maximise la marge entre les données à classer et donc minimise les taux d'erreurs.

3.3. Machine à vecteur de support :

D'après les résultats des tests sur l'algorithme K-means on a conclu qu'il faut un système pour pouvoir éloigner les valeurs rapprochées qui incitent à l'erreur et la confusion entre les différentes expressions.

Pour cela on a choisi la machine à vecteur de support (SVM) comme méthode de classification pour sa capacité à classer les données non linéairement séparables avec une fiabilité assurée par la séparation entre les classes par des marges qui réduisent les erreurs dû à l'ambiguïté.

Il existe plusieurs méthodes pour implémenter un SVM pour les données non linéairement séparables et faire une classification multi-classes, dans notre approche nous avons choisi d'utiliser la bibliothèque LIBSVM pour l'implémenter.

3.3.1. LIBSVM:

LIBSVM est une bibliothèque utilisée pour implémenter SVM pour résoudre n'importe quel problème de classification, cette bibliothèque contient des fonctions Kernel Paramétrées, les différentes formulations de SVM, et la classification Multi-classes [C4].

3.3.2. Nos choix de paramètres pour la classification des expressions faciales:

Pour notre étude comparative nous avons deux types de données en entrée : les coordonnées des points caractéristiques du visage, et les relations spatiales entre les points caractéristiques.

Puisque ces données vont être classifiées dans 7 classes donc on est dans un cas de Multi-classification de SVM, il existe plusieurs méthodes de multi-classification avec SVM (One-versus-all et One Against One qu'on a déjà expliqué dans le chapitre classification) pour notre application on a choisi d'utiliser la méthode One-versus-All qui d'après [M5] une étude comparative entre les deux méthodes multi-classifications ont conclu que One-Versus-All est mieux adaptée aux problèmes de classifications avec un petit nombre de classes qui est notre cas avec seulement 7 classes. Implémenter à l'aide de SVMlight qui se base sur LIBSVM.

Donc pour une classification non-linéairement séparable multi-classes il faut choisir une fonction Kernel qui donne le meilleur résultat de classification, donc pour notre approche on va tester avec les Kernels définies dans la bibliothèque LIBSVM (*Radial basis function kernel, segmoid kernel, polynomial kernel, Liniar kernal*) en utilisant les paramètres par défaut définies dans LIBSVM pour chaque Kernal.

3.3.3. Etapes d'application du SVM :

Pour appliquer le SVM sur les données en entrée on passe par ces étapes :

- 1) Transformer les données en format SVM package.

SVM requière que chaque instance de donnée soit représentée par un vecteur de valeurs numériques. Par conséquent, il faut une conversion au format numérique pour les données non numériques.

- 2) Effectuer une mise à échelle pour les données.

Le but de la mise à échelle est d'éviter que les attributs numériques largement supérieurs dominent les attributs de valeur inférieure. Exemple : supposant qu'on a procédé à une mise en échelle des données d'apprentissage de [-10, +10] à [-1, +1].

Si l'attribut de test est de [-11, +8] on doit faire une mise à échelle des données de test à [-1.1, +0,8].

3) Procéder à l'apprentissage.

L'apprentissage sera appliqué avec l'un des kernels (*Radial basis function kernel*, *sigmoid kernel*, *polynomial kernel*, *Liniar kernel*).

Tableau 6.7 : Les 4 fonction Kernel.

Kernel	Formule
Linéaire	$X \cdot Z$
Polynomial	$(\gamma X \cdot Z + C)^{\text{degre}}$
Radial Basis Function (RBF)	$\text{Exp}(-\gamma x-z ^2)$
Sigmoid	$\text{Tanh}(\gamma X \cdot Z + C)$

4) Test.

Après une transformation des images de test en format SVM packagent, ils seront testés avec le résultat des Kernels et ensuite affectés aux classes.

3.3.4. Application du SVM sur les coordonnées des points caractéristiques :

Pour appliquer le SVM sur les coordonnées des points caractéristiques, en premier on va récupérer les images par expression, et ensuite prendre les coordonnées (X, Y) de chaque point caractéristique des 33 points et on va les mettre dans un vecteur par image et chaque vecteur aura un Label qui représente la classe de l'expression de l'image. [Figure 6.7]

Exemple :

+2 1:137 2:83 3:105 4:82 5:63 6:100 7:78....

+2 : représente le label de la classe dégoutée.

Le X du 1^{er} point est 1:137

Le Y du 1^{er} point est 2:83

3.3.4.1. Implémentation du SVM avec les points caractéristiques :

Après la transformation des données en format de SVM package, une mise en échelle est appliquée, pour ensuite on procède au training avec la fonction kernel linéaire qui a donner les meilleurs résultats par rapport aux autres Kernel, avec le paramètre de régularisation C=4.555 pour les testes sur les 33 points caractéristiques, et C=13 pour les testes sur les 14 points caractéristiques qui a donner les meilleurs résultats par rapport aux autres paramètres tester [Figure 6.17][Figure 6.18].

```
C:\Users\user\Desktop>svmtrain SVM-33M-TES1 33 PMM>svm_multiclass_classification_33.dat model.d
et 33pts.dat
Reading model... C16 support vectors read! done.
Reading test examples...loading examples...done
Reading examples into memory...100, 200, 300, C200 examples read!
C200 examples? done.
Classifying test examples...100, 200, done
Function callout 10 in case-records: 0.00
Average loss on test set: 37.8225
Case/loss error on test set: 33.02% C200 correct, 67 incorrect, 200 total!

C:\Users\user\Desktop>svmtrain SVM-33M-TES2 33 PMM>svm_multiclass_classification_33.dat model.d
et 33pts.dat
Reading model... C16 support vectors read! done.
Reading test examples...loading examples...done
Reading examples into memory...100, 200, C192 examples read!
C192 examples? done.
Classifying test examples...100, 200, done
Function callout 10 in case-records: 0.00
Average loss on test set: 26.9722
Case/loss error on test set: 26.78% C201 correct, 71 incorrect, 172 total!

C:\Users\user\Desktop>svmtrain SVM-33M-TES3 33 PMM>svm_multiclass_classification_33.dat model.d
et 33pts.dat
Reading model... C16 support vectors read! done.
Reading test examples...loading examples...done
Reading examples into memory...100, 200, C201 examples read!
C201 examples? done.
Classifying test examples...100, 200, done
Function callout 10 in case-records: 0.00
Average loss on test set: 7.0047
Case/loss error on test set: 7.00% C185 correct, 15 incorrect, 201 total!

C:\Users\user\Desktop>svmtrain SVM-33M-TES4 33 PMM>svm_multiclass_classification_33.dat model.d
et 33pts.dat
Reading model... C16 support vectors read! done.
Reading test examples...loading examples...done
Reading examples into memory...100, 200, C201 examples read!
C201 examples? done.
Classifying test examples...100, 200, done
Function callout 10 in case-records: 0.00
Average loss on test set: 14.11% C184 correct, 172 incorrect, 200 total!

C:\Users\user\Desktop>svmtrain SVM-33M-TES5 33 PMM>svm_multiclass_classification_33.dat model.d
et 33pts.dat
Reading model... C16 support vectors read! done.
Reading test examples...loading examples...done
Reading examples into memory...100, 200, C200 examples read!
C200 examples? done.
Classifying test examples...100, 200, done
Function callout 10 in case-records: 0.00
Average loss on test set: 2.84% C201 correct, 5 incorrect, 200 total!

C:\Users\user\Desktop>svmtrain SVM-33M-TES6 33 PMM>
```

Figure 6.17 : implémentation de SVM avec les 33 points caractéristiques

```
C:\Users\user\Desktop>svmtrain SVM-14M-TES1 14 PMM>svm_multiclass_classification_14.dat model.d
et 14pts.dat
Reading model... C16 support vectors read! done.
Reading test examples...loading examples...done
Reading examples into memory...100, 200, C200 examples read!
C200 examples? done.
Classifying test examples...100, 200, done
Function callout 10 in case-records: 0.00
Average loss on test set: 0.4391
Case/loss error on test set: 0.43% C198 correct, 174 incorrect, 200 total!

C:\Users\user\Desktop>svmtrain SVM-14M-TES2 14 PMM>svm_multiclass_classification_14.dat model.d
et 14pts.dat
Reading model... C16 support vectors read! done.
Reading test examples...loading examples...done
Reading examples into memory...100, 200, C201 examples read!
C201 examples? done.
Classifying test examples...100, 200, done
Function callout 10 in case-records: 0.00
Average loss on test set: 7.0047
Case/loss error on test set: 7.00% C185 correct, 15 incorrect, 201 total!

C:\Users\user\Desktop>svmtrain SVM-14M-TES3 14 PMM>svm_multiclass_classification_14.dat model.d
et 14pts.dat
Reading model... C16 support vectors read! done.
Reading test examples...loading examples...done
Reading examples into memory...100, 200, C201 examples read!
C201 examples? done.
Classifying test examples...100, 200, done
Function callout 10 in case-records: 0.00
Average loss on test set: 1.9712
Case/loss error on test set: 1.97% C200 correct, 4 incorrect, 200 total!

C:\Users\user\Desktop>svmtrain SVM-14M-TES4 14 PMM>svm_multiclass_classification_14.dat model.d
et 14pts.dat
Reading model... C16 support vectors read! done.
Reading test examples...loading examples...done
Reading examples into memory...100, 200, C200 examples read!
C200 examples? done.
Classifying test examples...100, 200, done
Function callout 10 in case-records: 0.00
Average loss on test set: 0.9722
Case/loss error on test set: 0.97% C186 correct, 14 incorrect, 190 total!
```

Figure 6.18 : implémentation de SVM avec les 14 points caractéristiques

3.3.4.2. Testes et résultats :

Tableau 6.8 : Résultats test sur 33 points caractéristiques

KERNEL LINEAIRE	TAUX D'APPARITION
NEUTRE	59,82%
HEUREUX	92,04%
COLERE	66,18%
TRISTE	31,84%
PEUR	16,1%
DEGOUT	63,02%
SURPRISE	97,56%
	60,93%

Tableau 6.9 : Résultats test sur 14 points caractéristiques

KERNEL LINEAIRE	TAUX D'APPARITION
NEUTRE	32,73%
HEUREUX	92,54%
COLERE	58,82%
TRISTE	10,85%
PEUR	15,22%
DEGOUT	58,25%
SURPRISE	98,25%
	55,01%

Depuis les résultats des tests [Tableau 6.8] [Tableau 6.9] on remarque que les taux d'apparition varient selon l'expression, et que la méthode avec les 33points caractéristiques donne de meilleurs résultats 60,93% par rapport à la méthode à 14 point caractéristiques.

3.3.5. Application de SVM avec relations spatiales :

Pour appliquer le SVM sur les relations spatiales, en premier on va récupérer les images par expression, et ensuite prendre les relations spatiales entre les points

caractéristiques [Figure 6.14]. Et on va les mettre dans un vecteur par images et chaque vecteur aura un Label qui représente la classe de l'expression de l'image.

Exemple :

+1 1:132 2:79 3:95 4:78 5:59 6:97

+1 : représente le label de la classe colère.

La relation spatiale D1 est 1 :132

La relation spatiale D2 est 2 :79

3.3.5.1. Implémentation de SVM avec les relations spatiales :

Après la transformation des données en format de SVM package, une mise en échelle est appliquée, pour ensuite on procède au training avec la fonction kernel linéaire qui a donner les meilleurs résultats par rapport aux autres Kernel, avec le paramètre de régularisation C=155 qui a donner les meilleurs résultats pour les testes appliquer sur les 6 relations spatiales, et C=150 qui a donner les meilleurs résultats pour les testes appliquer sur les 5 relations spatiales par rapport aux autres paramètres tester pour les deux approches [Figure 6.19] [Figure 6.20].

```
C:\Users\med\Desktop\SUP\SUP\TEST 6D non0svm_multiclass_classify D1G6.dat model
F.dat D1G6Hex.dat
Reading model... (19 support vectors read) done.
Reading test examples, scanning examples... done
Reading examples into memory...100..200..OK (192 examples read)
(192 examples) done.
Classifying test examples..100..200..done
Runtime (without IO) in cpu-seconds: 0.00
Average loss on test set: 61.4563
Zero-one-error on test set: 61.4563 (74 correct, 118 incorrect, 192 total)

C:\Users\med\Desktop\SUP\SUP\TEST 6D non0svm_multiclass_classify FE6D.dat model
F.dat FE6DHex.dat
Reading model... (19 support vectors read) done.
Reading test examples, scanning examples... done
Reading examples into memory...100..200..OK (205 examples read)
(205 examples) done.
Classifying test examples..100..200..done
Runtime (without IO) in cpu-seconds: 0.00
Average loss on test set: 70.536 (44 correct, 161 incorrect, 205 total)
Zero-one-error on test set: 70.536 (44 correct, 161 incorrect, 205 total)

C:\Users\med\Desktop\SUP\SUP\TEST 6D non0svm_multiclass_classify H4G6.dat model
F.dat H4G6Hex.dat
Reading model... (19 support vectors read) done.
Reading test examples, scanning examples... done
Reading examples into memory...100..200..OK (201 examples read)
(201 examples) done.
Classifying test examples..100..200..done
Runtime (without IO) in cpu-seconds: 0.00
Average loss on test set: 12.4378
Zero-one-error on test set: 12.4378 (176 correct, 25 incorrect, 201 total)

C:\Users\med\Desktop\SUP\SUP\TEST 6D non0svm_multiclass_classify H4G6B.dat model
F.dat H4G6BHex.dat
Reading model... (19 support vectors read) done.
Reading test examples, scanning examples... done
Reading examples into memory...100..200..OK (201 examples read)
(201 examples) done.
Classifying test examples..100..200..done
Runtime (without IO) in cpu-seconds: 0.00
Average loss on test set: 60.706 (79 correct, 122 incorrect, 201 total)
Zero-one-error on test set: 60.706 (79 correct, 122 incorrect, 201 total)

C:\Users\med\Desktop\SUP\SUP\TEST 6D non0svm_multiclass_classify H4G6B.dat model
F.dat H4G6BHex.dat
Reading model... (19 support vectors read) done.
Reading test examples, scanning examples... done
Reading examples into memory...100..200..OK (205 examples read)
(205 examples) done.
Classifying test examples..100..200..done
Runtime (without IO) in cpu-seconds: 0.00
Average loss on test set: 1.4634 (202 correct, 3 incorrect, 205 total)
Zero-one-error on test set: 1.4634 (202 correct, 3 incorrect, 205 total)
```

Figure 6.19 : implémentation de SVM avec les 6 relations spatiales

```
C:\Users\med\Desktop\SUP\SUP\TEST 5D non0svm_multiclass_classify D15D.dat model
F.dat D15DHex.dat
Reading model... (17 support vectors read) done.
Reading test examples, scanning examples... done
Reading examples into memory...100..200..OK (192 examples read)
(192 examples) done.
Classifying test examples..100..200..done
Runtime (without IO) in cpu-seconds: 0.00
Average loss on test set: 81.779 (35 correct, 157 incorrect, 192 total)
Zero-one-error on test set: 81.779 (35 correct, 157 incorrect, 192 total)

C:\Users\med\Desktop\SUP\SUP\TEST 5D non0svm_multiclass_classify FE5D.dat model
F.dat FE5DHex.dat
Reading model... (17 support vectors read) done.
Reading test examples, scanning examples... done
Reading examples into memory...100..200..OK (205 examples read)
(205 examples) done.
Classifying test examples..100..200..done
Runtime (without IO) in cpu-seconds: 0.00
Average loss on test set: 79.6585
Zero-one-error on test set: 79.6585 (54 correct, 151 incorrect, 205 total)
Zero-one-error on test set: 79.6585 (54 correct, 151 incorrect, 205 total)

C:\Users\med\Desktop\SUP\SUP\TEST 5D non0svm_multiclass_classify H45D.dat model
F.dat H45DHex.dat
Reading model... (17 support vectors read) done.
Reading test examples, scanning examples... done
Reading examples into memory...100..200..OK (201 examples read)
(201 examples) done.
Classifying test examples..100..200..done
Runtime (without IO) in cpu-seconds: 0.00
Average loss on test set: 12.4378
Zero-one-error on test set: 12.4378 (176 correct, 25 incorrect, 201 total)
Zero-one-error on test set: 12.4378 (176 correct, 25 incorrect, 201 total)

C:\Users\med\Desktop\SUP\SUP\TEST 5D non0svm_multiclass_classify H45DB.dat model
F.dat H45DBHex.dat
Reading model... (17 support vectors read) done.
Reading test examples, scanning examples... done
Reading examples into memory...100..200..OK (201 examples read)
(201 examples) done.
Classifying test examples..100..200..done
Runtime (without IO) in cpu-seconds: 0.00
Average loss on test set: 53.216 (84 correct, 117 incorrect, 201 total)
Zero-one-error on test set: 53.216 (84 correct, 117 incorrect, 201 total)

C:\Users\med\Desktop\SUP\SUP\TEST 5D non0svm_multiclass_classify H45DB.dat model
F.dat H45DBHex.dat
Reading model... (17 support vectors read) done.
Reading test examples, scanning examples... done
Reading examples into memory...100..200..OK (205 examples read)
(205 examples) done.
Classifying test examples..100..200..done
Runtime (without IO) in cpu-seconds: 0.00
Average loss on test set: 0.303 (203 correct, 2 incorrect, 205 total)
Zero-one-error on test set: 0.303 (203 correct, 2 incorrect, 205 total)
```

Figure 6.20 : implémentation de SVM avec les 5 relations spatiales

3.3.5.2. Testes et résultats:

Tableau 6.10 : Taux d'apparition avec les 6 relations spatiales

KERNEL LINEAIRE	TAUX DE BONNE APPARITION
NEUTRE	49,07%
HEUREUX	82,56%
COLERE	56,36%
TRISTE	39,3%
PEUR	21,46%
DEGOUT	38,54%
SURPRISE	98,54%
	55,83%

Tableau 6.11 : tests pour les 5 relations spatiales

KERNEL LINEAIRE	TAUX DE BONNE APPARITION
NEUTRE	39,12%
HEUREUX	87,56%
COLERE	69,61%
TRISTE	41,79%
PEUR	26,34%
DEGOUT	18,23%
SURPRISE	99,02%
	54,52%

Depuis la comparaison entre les deux méthodes de calcul on remarque que la méthode à 5 relations spatiales donne des résultats presque meilleurs 54,52% par rapport à la méthode à 6 relations spatiales proposée 55,83%.

3.3.6. Comparaison des résultats des deux méthodes : relations spatiales et coordonnées des points caractéristiques :

Depuis les résultats des testes [Tableau 6.8] [Tableau 6.10] que la méthode qui utilise les coordonnées des points caractéristiques est plus meilleure avec un résultat de reconnaissance de 60,93% et donc elle est la meilleure méthode parmi les méthodes proposées pour la reconnaissance des expressions faciales, et on note que ces résultats peuvent être améliorés si on utilise une méthode de division de la base d'image en base d'apprentissage et base de teste.

4. Comparaison des résultats de K-means et de SVM:

Les testes de la méthode SVM qui utilise les 33 points caractéristiques donnent les meilleurs résultats que les résultats qu'on a eu dans les testes de K-means et donc elle représente la meilleure combinaison de méthodes de détection de changement dans le visage et de classification et qui vont enfin construire le système de reconnaissance des émotions.

5. Conclusion :

Dans ce chapitre nous avons effectué une étude comparative entre la méthode de détection des changements lors d'une expression faciale avec les coordonnées des points caractéristiques et la méthode de calcul des relations spatiales entre ces points. Une série de test été effectuer pour sélectionner les points caractéristiques qui change significativement lors d'une expression et 33 points caractéristiques ont été sélectionné, et pour sélectionner les relations spatiales qui entre dans les expressions 6 relations spatiales ont été sélectionné.

Deux méthodes de classification une naïve et simple (K-means) et l'autre fiable et efficace (machine à vecteur de support) ont été testé sur l'ensemble des données (les 33 points caractéristiques et les 6 relations spatiales) pour enfin déduire la combinaison de méthodes (coordonnées des points caractéristiques ou relation spatiale) avec les méthodes de classification et une combinaison est avéré la plus efficace entre les 4 avec un taux de reconnaissance total de 60,93%, la combinaison de la méthode SVM avec les coordonnées des 33 points caractéristiques est la

méthode la plus efficace entre les 4 méthodes étudiées, cette méthode a eu de très bons résultats avec 4 expressions avec des taux de reconnaissance élevés (heureux 92,04%, colère 66,18%, dégoût 63,02%, surprise 97,56%).

Chapitre 7 : Implémentation :

1. Introduction :

Nous aborderons dans ce chapitre la phase d'implémentation des méthodes décrites dans le chapitre précédent pour cela on procède à une phase de codage et une phase d'implémentation de la meilleur méthode sélectionnée avec la webcam.

2. Phase Codage :

Dans cette phase on procède au codage des différents algorithmes de teste pour la sélection des points caractéristiques et des relations spatiales et ainsi les algorithmes de classification K-means et SVM.

2.1.Langage et outils de développement :

Dans cette partie nous allons décrire le langage de programmation et les différents outils utilisés.

2.1.1. C++ :

Pour le développement de notre application nous avons choisi d'utiliser le langage de programmation C++, qui est le langage le plus utilisé dans le domaine de traitement d'image, le C++ a été développé par Bjarne Stroustrup au cours des années quatre vingt, il s'agit d'une amélioration du langage C. Les premières améliorations se concrétisèrent par la prise en charge de classes et donc ajouter les propriétés de l'approche orientée objet.

Les caractéristiques de C++ :

C++ est caractérisé par :

- Orienté objet (POO).
- Simple, robuste, dynamique et sécurisé.

- Adapté aux programmes de tailles importantes et donne de l'importance aux performances.
- Un langage de programmation multi paradigmes (il permet plusieurs types de programmations par exemple : programmation objet, programmation procédurale, programmation générique grâce aux Templates).

2.1.2. Visual Studio :

Définition :

Visual studio est un ensemble complet de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C# utilisent tous les mêmes environnements de développement intégré (IDE, Integred Developement Environment), qui leur permet de partager des outils et facilite la création en faisant appel à plusieurs langages.

2.1.3. Librairie OpenCV :

OpenCV est une librairie Open source, développée par Intel, spécialisée dans le traitement d'image à temps réel, utilisée dans notre application pour la détection du visage dans les images et le flux vidéo.

2.1.4. STASM :

STASM est une bibliothèque basée sur le modèle actif de forme ASM, le package est préconfiguré pour la détection des points caractéristiques du visage dans des images stables, elle permet aussi de construire de nouveau ASM.

Le programme est écrit en C++.

3. Phase D'implémentation de la meilleure méthode avec la Webcam :

D'après les résultats des testes des deux méthodes sur les points caractéristiques et les relations spatiales et aussi d'après les résultats obtenus on a remarqué que la meilleure méthode est le SVM qui utilise les 33 points caractéristiques pour classifier les expressions faciales et donc avec un taux de reconnaissance de 60,93% cette

combinaison entre le SVM et les points caractéristiques sélectionnés constitue une solution qui peut être adaptée pour ce genre de projet.

Donc une fois la meilleure méthode sélectionnée on va l'implémenter avec la Webcam qui le but de notre projet est de créer un système qu'on peut utiliser pour reconnaître les émotions à partir d'expressions faciales capturées via Webcam [Figure 7.1].

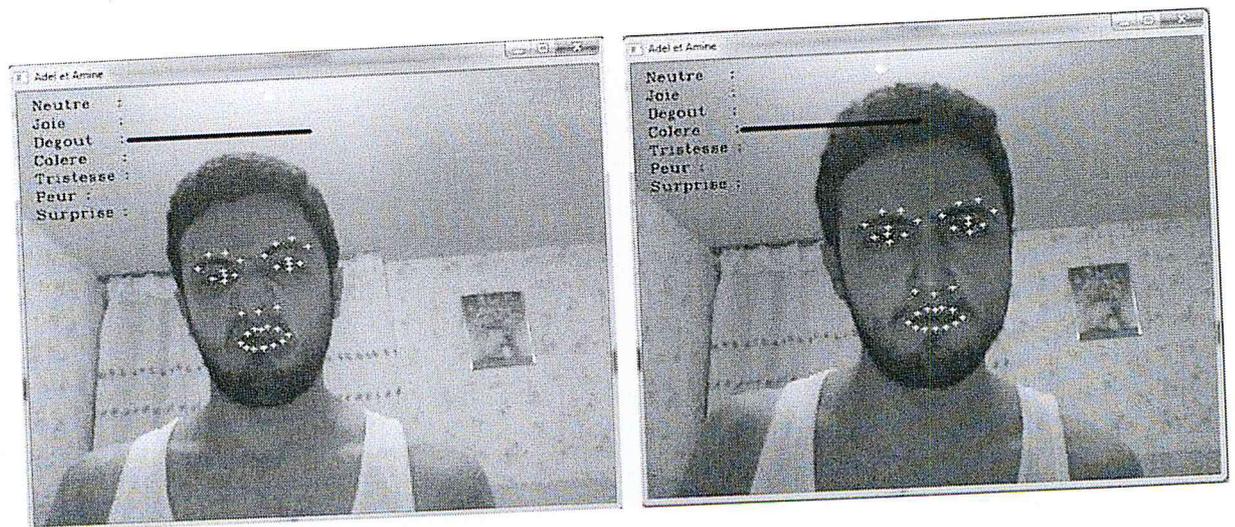


Figure 7.1 : la méthode sélectionnée appliquée sur la webcam

4. Conclusion :

Dans ce chapitre nous avons vu les différents outils utilisés pour le codage des tests effectués et de l'application finale sur la Webcam qu'on a implémentée après avoir sélectionné le meilleur système qui regroupe la détection des changements de la position des 33 points caractéristiques et qu'on classe en utilisant SVM.

Conclusion générale:

Nous venons de présenter dans ce mémoire nos travaux sur la reconnaissance des expressions faciales par vision sur ordinateur. Notre intérêt est d'interpréter les émotions exprimées par l'utilisateur, vu l'importance de telle analyse qui a été soulignée dans plusieurs travaux réalisés dans différents domaines (psychologie, neurophysiologie, psycho comportementaliste, psychologie, informatique...etc.). Notre idée principale d'une part est d'explorer de nouvelles données et d'autre part de tester de nouvelles techniques ou méthodes qui une fois combinées vont donner un système capable de reconnaître les expressions faciales avec fiabilité. Pour le faire nous avons effectué une série de testes sur deux méthodes utilisées pour détecter le changement dans les expressions du visage, la première est de prendre les coordonnées des points caractéristiques pour chaque expression et la tester avec le changement de position, et la seconde est de calculer les relations spatiales entre les points caractéristiques. on a effectué des testes pour sélectionner la meilleure combinaison de points caractéristiques et la meilleure combinaison de relations spatiales qui entrent dans chaque expression, une fois ces combinaisons sélectionnées on a appliqué deux méthodes de classification : une simple et naïve le K-means et l'autre fiable et plus efficace qui est la machine à vecteur de support SVM, et on a comparé les résultats pour en sortir avec un meilleur système qui combine la méthode de détection de changement avec les coordonnées des points caractéristiques et la méthode de classification SVM.

A travers ce travail nous avons découvert le domaine de la reconnaissance automatique de formes et appris beaucoup sur la perception intelligente de l'image.

Comme perspectives de ce travail, nous estimons que nos démarches ont aidé à sélectionner la meilleure méthode parmi toutes les méthodes testées. Mais on souligne que les résultats peuvent être améliorés dans le cas où on utilise des méthodes de division des bases d'images en base d'apprentissage et base de teste (par exemple cross validation [E5]), et encore on met comme perspective l'ajout d'images représentant des expressions étrangères pour apprendre au système à reconnaître les expressions étrangères et qui va l'aider à mieux distinguer les sept expressions de base des autres expressions.

Références Bibliographiques:

- [A1] Face and facial expression analysis based on an active appearance model
Franck Davoine, Bouchra Abboud et Van Mô Dang, HEUDIASYC, UMR 6599
CNRS, Université de Technologie de Compiègne, BP 20529.
- [A2] :Ahonen, T., Hadid, A. and Pietikainen, M.: Face recognition with local binary patterns, Lecture Notes in Computer Science, 469-481 (2004).
- [A3] classification of facial expressions using k-nearest neighbor classifier, Abu Sayeed Md. Soheil and Prabir Bhattacharya, Concordia University Canada, 2007.
- [A4] Kernel Methods, Introducing Machine Learning Course. California Institute of Technology May 22, 2012.
- [B1] Classification supervisée et non supervisée des données de grande dimension, Charles BOUVEYRONI & Stéphane GIRARD. France. 2009
- [B2] Synthesizing Expressions using Facial Feature Point Tracking: How Emotion is conveyed, Tadas Baltrušaitis, Laurel D. Riek, and Peter Robinson Computer Laboratory, University of Cambridge.
- [C1] :http://www.cognotics.com/opencv/servo_2007_series/part_2/index.html, by Robin Hewitt.
- [C2] Catégorisation des expressions faciales par marches aléatoires sur graphe, Youssef CHAHIRI, Youssef ZINBI et Kheir Eddine AZIZI, Université de Caen.
- [C3] Kanade, T., Cohn, J. F., & Tian, Y. "Comprehensive database for facialexpression analysis"; Proceedings of the fourth IEEE International Conference on Automatic Face and Gesture recognition (FG'00), Grenoble, France 46-53.
- [C4] LIBSVM: A Library for Support Vector Machines, Chih-Chung Chang and Chih-Jen Lin, Department of Computer Science, National Taiwan University, Taipei, Taiwan.

[D1] Bechara A, Damasio H, Damasio AR, Lee GP. Different contributions of the human amygdala and ventromedial prefrontal cortex to decision-making. *J Neurosci* 1999; 19: 5473-81.

[D2] Darwin C. *The expression of the emotions in man and animals*. Murray, London 1872.

[E1] Ekman P, Davidson RJ. *The nature of emotion*: Oxford University Press New York, 1994.

[E2] Ekman P, Friesen WV, Ellsworth PC. *Emotion in the human face*: Cambridge University Press Cambridge, 1982.

[E3] Ekman P. The argument and evidence about universals in facial expressions of emotion. *Handbook of social psychophysiology* 1989; 58: 342-353.

[E4] *DataMining : Techniques d'extraction des connaissances*, Georges El Helou et Charbel Abou khalil, l'université de paris 2. 2004.

[E5] *Evaluating Classifiers*, Charles Elkan, January 20, 2012.

[F1] Fridlund AJ, Ekman P, Oster H. Facial expressions of emotion: Review of literature, 1970-1983. *Nonverbal behavior and communication* 1987: 143-224.

[H1] M. *Neurologie des émotions : un panorama des preuves expérimentales et cliniques = Neurology of emotions : An overview of experimental proofs*. *Revue de neuropsychologie* 1998; 8: 471-535.

[H2] *DataMining: concepts and techniques second edition*, Jiawei Han, Micheline Kamber, University of Illinois at Urbana-Champaign, 2006.

[H3] *Profil émotionnel et cognitif au début de la sclérose en plaques : Effets différentiels des émotions sur les performances cognitives*, Hélène Petropoulou le 29 novembre 2006, UNIVERSITE PARIS 8

[J1] JAFFE database: http://www.kasrl.org/jaffe_download.html

[K1] :Krestinin, I.A., Seredin, O.S.: Excluding cascading classifier for face detection. Proc.of the 19th Int. Conf. on Computer Graphics and Vision, 380–381 (2009).

[K2] :Kienzle, W., Bakir, G., Franz, M., Scholkopf, B.: Face detection – efficient and rank deficient, Advan. in neural inform. process. systems 17, 2005. – P. 673–680 (2005).

[K3]http://www-inrev.univ-paris8.fr/extras/Michel_Bret/cours/bret/cours/rn/rn8.htm, par Michel-Bret.

[K4] KHADOUDJA GHANEM, Reconnaissance des Expressions Faciales à Base d'Informations Vidéo ; Estimation de l'Intensité des Expressions Faciales, Université Mentouri de Constantine, Faculté des sciences de l'ingénieur, Département d'informatique, 2010.

[L1] LeDoux J. The emotional brain: The mysterious underpinnings of emotional life: Simon & Schuster, 1998.

[L2] Lang PJ. The emotion probe. Studies of motivation and attention. Am Psychol 1995; 50: 372-85.

[L3] Lang PJ. Behavioral treatment and bio-behavioral assessment: Computer applications. Technology in mental health care delivery systems 1980: 119–137.

[L4] :Lienhart, R. and Maydt, J.: An extended set of Haar-like features for rapid object detection. Proc. Intern. Conf. on Image Processing 2002, 900–903 (2002).

[M1] : Ming-Hsuan Yang, Member, IEEE, David J. Kriegman, Senior Member, IEEE, and Narendra Ahuja, Fellow.: Detecting Faces in Images: A Survey , IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 1, JANUARY 2002.

[M2] :Marcel, S., Keomany, J., et al.: Robust-to-illumination face localisation using Active Shape Models and Local Binary Patterns, IDIAP-RR, V. 47, (2006).

[M3] Locating Facial Features with Active Shape Models, Stephen Milborrow, Supervised by Dr Fred Nicolls Submitted to the Faculty of Engineering, University of Cape Town, in partial fulfillment of the requirements for the Degree of Master of Science, November 16, 2007

[M4] Expert system for automatic analysis of facial expressions, M. Pantic, L.J.M. Rothkrantz, Delft University of Technology, 25 January 2000.

[M5] “One Against One” or “One Against All”: Which One is Better for Handwriting Recognition with SVMs?, *Jonathan Milgram Mohamed Cheriet Robert Sabourin*, École de Technologie Supérieure, Montréal, Canada

[N1] :NikolayDegtyarev and Oleg Seredin .: Comparative Testing of Face Detection Algorithms , Tula State University (2010).

[N2] Rapid Facial Expression Classification Using Artificial Neural Networks, Nathan Cantelmo, Northwestern University, Month 1–2, 2004.

[O1] : Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 24, Issue 7, 971–987 (2002).

[P1] Petropoulou H. Profil émotionnel et cognitif au début de la sclérose en plaques: effets différentiels des émotions sur les performances cognitives. Psychologie. Vol Ph.D. Paris VIII, 2006.

[P2] Petropoulou H. Profil émotionnel et cognitif au début de la sclérose en plaques: effets différentiels des émotions sur les performances cognitives. Psychologie. Vol Ph.D. Paris VIII, 2006.

[P3] :Papageorgiou, C. P., Oren, M., Poggio, T.: A general framework for object detection, Proc. of ICCV, 555–562 (1998).

[R1] *Quelques Types de Réseaux de Neurones La Rétropropagation*, Philippe Leray, INSA Rouen -Département ASI - Laboratoire PSI

[S1]:Sauquet, T., Rodriguez, Y., Marcel, S.: Multiview face detection, IDIAP-RR (2005).

[S2] Shuaishi Liu, Ning Ding, Mao Yang; Keping Liu, Facial Expression Recognition Method Based on the Fusion of Geometry Features and Texture Features, School of Electrical and Electronic Engineering, Changchun University of Technolog, China, Changchun, 2012

[T1] ENHANCED K-MEANS BASED FACIAL EXPRESSIONS RECOGNITION SYSTEM, TANVI SHEIKH, SHIKHA AGRAWAL, CSE Department CSIT Durg.

[T2] LES RESEAUX DE NEURONES ARTIFICIELS INTRODUCTION AU CONNEXIONNISME COURS, EXERCICES ET TRAVAUX PRATIQUES, Claude TOUZET, Juillet 1992.

[V1] Viola, P. and Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple, Proc. IEEE CVPR 2001, (2001)

[V2] Viola, Paul and Jones, Michael J.: Robust Real-Time Face Detection, International Journal of Computer Vision, Volume 57, Issue 2, 137–154 (2004)

[X1] Facial expression recognition: A clustering-based approach, Xue-wen Chen, Thomas Huang, University of Illinois at Urbana-Champaign, September 2002.

[Y1] Recognizing Human Facial Expressions from Long Image Sequences Using Optical Flow, Yaser Yacoob and Larry S. Davis, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 18, NO. 6, JUNE 1996.

[Z1] Feature-Based Facial Expression Recognition: Sensitivity Analysis and Experiments with a Multi-Layer Perceptron, Zhengyou Zhang, nInternational Journal of Pattern Recognition and Artificial Interlligence.

[P3] FACIAL ANIMATION PARAMETERS EXTRACTION AND EXPRESSION RECOGNITION USING HIDDEN MARKOV MODELS, Montse Pardàs, Antonio Bonafonte

Résumé :

L'interprétation des expressions faciales est devenu ces derniers temps un moyen très efficace de communication homme-machine. Au cours de cette thèse, nous avons effectué une série de testes pour pouvoir déterminer les méthodes qui une fois regroupées vont former un système capable de reconnaître les expressions faciales avec fiabilité et efficacité.

Deux testes ont été effectués pour déterminer la méthode de détection de changement dans le visage une par rapport aux changements des positions (coordonnées) des points caractéristiques détectés où 33 points ont été sélectionnés, et l'autre par rapport à la relation spatiale entre les points caractéristiques où 6 relations spatiales ont été sélectionnées.

Une fois les méthodes de détection de changement ont été sélectionnées, des testes de classification ont été effectués sur les deux méthodes en utilisant la méthode simple est naïve le K-means qui est la méthode efficace et largement utilisée la machine à vecteur de support SVM. Qui ont donnés comme résultat la meilleure combinaison de méthodes (détection de changement et classification).

Le système final est composé de la méthode de détection de changement dans le visage par rapport à la position des 33 points caractéristiques et la méthode de classification SVM et qui une fois combinés ont donné un résultat final de 60,93%.

Summary:

The interpretation of facial expression has become a very effective way to man-machine communication lately. In this thesis, we conducted a series of tests on order to determine the methods that when combined will form a system capable of recognizing facial expressions with reliability and efficiency.

Two tests were performed to determine the method of facial changes detection, the first with respect to feature points position (coordinates) changes, where 33 feature points were detected and used, and the other method with respect to the spatial relations between feature points, where 6 spatial relations were selected.

Once these two methods were selected, we conducted classifications tests on the two methods using a simple and naïve classification method K-means and effective and widely used method support vector machine SVM. Who gave as a result the best combination of methods (facial changes detection and classification).

The final system is composed of the 33 feature points changes detection method and SVM classification method and the combination gave 60.93%.