REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

Filière : Informatique
Option : Intelligence Artificielle

Mémoire

présenté pour obtenir le diplôme de master académique en

Montage d'un Système de Reconnaissance des Expressions Faciales avec le Deep Learning

---

Présenté par:
CHETTOUH HADJER

Encadré par :
Mme Hattab Dalila

Septembre 2020

Remerciements

Tout d'abord je tiens à remercier ALLAH de m' avoir donné la patience, la santé et le courage pour arriver jusqu'à là.

Je remercier énormément ma mére et ma fille Imen qui par leurs soutien, leurs prières et leurs encouragements, j'ai a pu surmonter tous les obstacles.

Je tiens à remercier Mde Hattab pour son encadrement,son suivi, ses conseils et ses critiques constructives.

Je tiens aussi à remercier les membres du jury pour leur précieux temps accordé à l'étude de notre mémoire.

Je remercie et mon gratitude vont aux professeurs et enseignants de département

d'informatique ainsi que ses étudiants au long de cette année spéciale .

Que toute personne ayant œuvré de près ou de loin à la réalisation de ce proje,tu trouve ici le témoignage de mon plus profonde reconnaissance.

Dédicace

Je dédie ce travail

Aux plus merveilleux et adorables couples, à cette source de tendresse et de générosité qu'ils m'ont apportée . À mes chers parents qui m'ont setenu et encouragé avec leurs priéres.que dieu vous benisses et vous protéges nchallah.

À mes chers enfants « Monther, Imen, Takwa, Yakin,Aymen » qui ont toujours été à mes côtés sur tous ma fille aînée Imen que dieu la protége.

À mon mari pour son soutien.

À mes soeurs pour leur tendresse et leur présence malgré la distance qui nous sépare.

À toute ma famille sans aucune exception.

À tous mes professeurs et enseignants que j'ai eu durant tout mon cursus scolaire et qui m'ont permis de réussir dans mes études.

A tous mes amis (es).

A toute personne ayant contribué à ce travail de près ou de loin. le courage, la force et la volonté pour réussir et de nous avoir éclairci le chemin tout au long de notre vie.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

Liste des abréviations

CNN	Le Convolutional Neural Networks
FACS	Facial Action Coding System
AUs	Actions Units
FFP	Facial Feature Points
FAPU	Facial Animation Parameter Units
LDA	Linear Discriminant Analysis
LBP	Local Binary Patterns
PCA	Principal Component Analysis
SVM	Support Vector Machines
AAM	Active Appearance Models
LBP	Local Binary Pattern
RNN	Recurrent Neural Network
MLP	Multi Layer Perceptron
ReLu	Rectified Linear Units

RÉSUMÉ

L'expression faciale est l'un des moyens non verbaux les plus couramment utilisés par les humains pour transmettre les états émotionnels internes et, par conséquent, joue un rôle fondamental dans les interactions interpersonnelles. Bien qu'il existe un large éventail d'expressions faciales possibles, les psychologues ont identifié six expressions fondamentales (la joie, la tristesse, la surprise, la colère la peur et le dégoût) universellement reconnues.

La reconnaissance des émotions est l'un des domaines scientifiques les plus complexes. Ces dernières années, de plus en plus d'applications tentent de l'automatiser. Ces applications innovantes concernent plusieurs domaines comme l'aide aux enfants autistes, les jeux vidéo, l'interaction homme-machine.

Nous proposons dans ce travail un système capable de détecter et d'identifier l'utilisateur à travers ses expressions faciales afin de reconnaître son état émotionnel. Le système utilise un classifieur d'expressions faciales basé sur l'apprentissage profond (Deep learning) et qui applique un algorithme de réseaux de neurones convolutifs (Xception).

Les expériences ont été menées afin de vérifier la faisabilité du système proposé. Son objectif est la validation de la détection des visages et la reconnaissance de l'utilisateur et de ses émotions à travers ses expressions faciales avec la base de données fer2013.

Les résultats expérimentaux montrent la fiabilité du CNN Xception avec sa spécificité conventionnelle qui inclut des couches de convolution séparable. La précision augmente même lorsque le modèle ne traite pas la totalité de paramètres, ce qui génère des résultats remarquables sur la base de données FER-2013.

Mots-clés : Expression faciale, émotion, Deep Learning, Réseaux de neurones convolutifs.

Abstract

Facial expression is one of the most common non-verbal means used by humans to convey internal emotional states and, for example, therefore, plays a fundamental role in interpersonal interactions. Although there is a wide range of possible facial expressions, psychologists have identified six basic expressions (joy, sadness, surprise, universally recognized anger fear and disgust.

Recognizing emotions is one of the most complex fields of science. In recent years, more and more applications are trying to automate it. These innovative applications concern several areas such as helping autistic children, video games, human-machine interaction.

We propose in this work a system capable of detecting and identifying the user through his facial expressions in order to recognize his emotional condition. The system uses a facial expression classifier based on Deep Learning and which applies a convolutional neural network algorithm (Xception).

The experiments were carried out in order to verify the feasibility of the proposed system. Its objective is the validation of face detection and recognition of the user and his emotions through his facial expressions with the fer2013 database.

The experimental results show the reliability of CNN Xception with its conventional specificity which includes separable convolution layers. The precision increases even when the model does not process all of the parameters, which generates remarkable results on the FER-2013 database.

Keywords : Facial expression, emotion, Deep Learning, Convolutional neural networks.

INTRODUCTION GÉNÉRALE

Le visage humain peut révéler beaucoup d'informations, par exemple, un médecin peut diagnostiquer un patient juste en regardant son visage, un psychologue peut faire un rapport de diagnostic aussi, et un policier peut juger quelqu'un de l'apparence de son visage. Par conséquent, en regardant simplement la face de quelqu'un, nous pourrions savoir beaucoup de choses sur lui, s'il est heureux ou en colère ou malade, ou s'il est digne de confiance ou non, s'il dit la vérité ou ment ... etc.

Les expressions faciales jouent un rôle irremplaçable dans la communication non verbale. Elles communiquent l'émotion et signalent les intentions, la vigilance, la douleur et les traits de personnalité. Les émotions peuvent être exprimées à la fois verbalement et non verbalement. Il existe de nombreux canaux tels que la voix, le visage et les gestes corporels à travers lesquels l'information non verbale est transmise aux observateurs. En outre, Mehrabian et Ferris [43] ont indiqué que l'expression faciale du locuteur contribue à 55% à l'effet du message parlé, alors que la partie verbale et la partie vocale qu'avec 7% et 38% respectivement. Ainsi, le visage a tendance à être la forme la plus visible de la communication de l'émotion. Il fait de la reconnaissance d'expression faciale un moyen largement utilisé pour mesurer l'état émotionnel des êtres humains. A cet égard, les expressions faciales fournissent des informations aux observateurs sur l'expérience émotionnelle d'une personne.

Les expressions faciales peuvent non seulement changer le flux de la conversation mais aussi fournir aux auditeurs un moyen de communiquer une grande quantité d'informations au locuteur sans même prononcer un seul mot. Lorsque l'expression faciale ne coïncide pas avec les mots parlés, alors l'information véhiculée par le visage prend plus de poids dans le décodage des informations.

L'analyse automatique de l'expression du visage est un problème qui affecte d'importantes

applications dans de nombreux domaines tels que l'interaction homme-machine. En fait, bien que les nouvelles technologies soient présentes dans notre vie quotidienne, ils ne fournissent pas une interface adéquate qui les rend plus abordables pour les utilisateurs. Par conséquent, l'informatique affective en améliorant l'interaction homme-ordinateur, permet aux ordinateurs d'être plus adaptés à l'homme et non pas l'inverse. L'intérêt de la recherche est de permettre aux systèmes informatiques de reconnaître les expressions et d'utiliser les informations émotives intégrées dans les interfaces homme-machine. Divers algorithmes d'extraction de caractéristiques et d'apprentissage automatique ont été développés. La plupart de ces méthodes ont déployé des fonctionnalités manuelles suivies d'un classifieur, tel que la méthode SVM [50], Adaboost [61], ou encore la méthode Forêts aléatoires [9]. Le succès actuel des réseaux de neurones convolutifs (CNN) dans la classification d'images s'est étendu au problème de la reconnaissance de l'expression faciale. Le deep Learning et plus particulièrement les réseaux de neurones convolutionnels (CNN) ont apparu spécialement pour résoudre les problèmes rencontrés du machine Learning. L'un des ingrédients les plus importants pour le succès de ces méthodes est la disponibilité de grandes quantités de données d'entraînement. Le Convolutional Neural Networks (CNN) est l'une des structures réseau les plus représentatives de la technologie d'apprentissage en profondeur et a connu un grand succès dans le domaine du traitement et de la reconnaissance d'images.

L'objectif de ce mémoire consiste à proposer une approche de reconnaissance des expressions faciales en se basant sur la méthode des réseaux de neurones convolutifs.

Ce mémoire est constitué en quatre chapitres, et organisé comme suit :

- Dans le **premier chapitre**, nous présentons l'expression faciale, les systèmes de codifications et les systèmes de reconnaissances.

- Ensuite, **le second chapitre**, nous le consacrons à la présentation de l'apprentissage profond, où nous donnerons plus de détails sur les réseaux de neurones convolutifs.

- La conception de notre approche de reconnaissance des expressions faciales basée sur les CNNs est présentée dans **le troisième chapitre**.

- **Le dernier chapitre** est consacré à la description des différents outils utilisés dans le développement de notre application, ainsi que les différents résultats obtenus.

- Et enfin, nous terminerons ce mémoire par une conclusion générale et quelques perspectives.

CHAPITRE 1

LES EXPRESSIONS FACIALES

1.1 Introduction

Les émotions sont indispensables à notre vie. Elles permettent d'améliorer la communication entre les individus, d'assurer une meilleure compréhension du message véhiculé et de s'adapter à une situation donnée. Les émotions jouent un rôle primordial pour la prise de décision. Elles influencent également les comportements et façonnent la personnalité.Ces différents rôles intéressent beaucoup d'applications qui tentent d'automatiser la reconnaissance des émotions.

Il existe de nombreuses applications pouvant profiter d'un tel outil et ceci afin de modifier dynamiquement le comportement de l'application ou de récolter des informations sur l'état de l'utilisateur. Par exemple, dans le domaine des jeux vidéo la détection des émotions commence à atteindre le grand public.Microsoft propose au travers de la Xbox One une analyse du rythme cardiaque pour l'identification d'émotions. Ce type d'application peut permettre de modifier l'environnement virtuel du jeu en fonction des émotions (modification de la météo, du comportement des avatars, difficulté,...) mais aussi pouvoir identifier la validité d'un scénario en déterminant les émotions ressenties chez l'utilisateur. D'autres contextes applicatifs peuvent être imaginés comme dans le domaine médical pour identifier la dépression ou des troubles liés au stress. Dans le domaine de l'apprentissage également, nous pouvons imaginer l'emploi d'un tel détecteur afin d'améliorer la productivité de l'apprentissage et obtenir des informations sur l'état d'écoute de l'auditoire.

Dans ce premier chapitre, nous présentons quelques notions concernant les émotions, les expressions faciales telles que leurs définitions, leurs différentes théories, et leurs composantes, et nous terminons le chapitre par les différentes techniques d'apprentissage automatique.

1.2 Problématique

L'interaction homme-machine a longtemps se limiter ses recherches au développement de techniques fondées sur l'usage du triplet écran-clavier-souris. Aujourd'hui, elle se dirige vers de nouveaux paradigmes : l'utilisateur doit pouvoir évoluer sans obstacles dans son milieu naturel ; les doigts, la main, le visage ou les objets familiers sont envisagés comme autant de dispositifs d'entrée/sortie, la frontière entre les mondes électronique et physique tend à devenir floue.

Ces nouvelles formes d'interaction ont besoin généralement de capturer du comportement observable d'un utilisateur et de son environnement. Elles se basent pour cela sur des techniques de vision par ordinateur. Les générations futures d'environnement Homme-Machine deviendront multimodales en intégrant de nouvelles informations, tire son origine de la prise en compte de la parole et/ou des expressions faciales, pour faire passer l'utilisation des machines en une manière directe et naturelle.

L'état émotionnel de l'être humain affecte d'une manière directe son comportement et son rendement dans leurs tâches quotidiennes. Pour cela la détection de son expression faciale qui va préciser son émotion devient une tâche indispensable pour préciser son émotion avant d'effectuer son travail, tel que la conduite, la robotique sociale et le traitement médical.

Ainsi, plusieurs questions se posent et ouvrent sur les problématiques suivantes Comment modéliser les émotions en tenant compte de leur complexité ? Comment effectuer l'échange émotionnel lors d'une interaction homme-machine ou machine-machine ? Comment modéliser l'aspect psychologique et émotionnel humain en informatique en se basant sur les différentes théories, théorèmes ?

1.3 Expression faciale et émotion

Expressions et émotions sont très liées et parfois confondues, l'émotion est un des générateurs des expressions faciales.Une émotion implique généralement une expression faciale correspondante (dont l'intensité peut être plus ou moins contrôlée selon les individus), mais l'inverse n'est pas vrai.

1.3.1 Définition des émotions

Les émotions, de façon générale, sont des états motivationnels. Elles sont constituées d'impulsions, de désirs ou d'aversions ou, plus généralement, elles comportent des changements de motivation. Elles poussent l'individu à modifier sa relation avec un objet, un état du monde,

un état de soi, ou à maintenir une relation existante malgré des obstacles ou des interférences. Notons une caractéristique essentielle de ces motivations : les émotions sont relationnelles. Elles se jouent entre le sujet et le monde. Les émotions ne sont pas des états subjectifs, intérieurs à une personne, ou du moins pas en première instance. Évidemment, une émotion peut rester intérieure à une personne et rester limitée à son expérience intime. Mais, même dans ce cas, la tendance à l'action est présente, se manifeste dans le ressenti et à travers l'imagination .En colère, on pense à ce qu'on voudrait faire à l'adversaire ou, de façon plus discrète encore, à ce qu'on aimerait qu'il lui arrive. Dans l'inquiétude, les pensées vont de-ci de là sans repos, raidissant le dos pour faire face à ce qui pourrait arriver [13].

L'expérience subjective, le ressenti des émotions, est largement le reflet des tendances à l'action, comme le montrent les recherches portant sur la description des expériences émotionnelles. Les émotions dites « de base » sont caractérisées par des mondes de préparation distincts et spécifiques : la peur par la tendance à s'éloigner ou à se protéger, la colère par l'opposition et l'hostilité, la honte et la culpabilité par la soumission, et les émotions de joie et de tristesse par des tendances plus diffuses d'augmentation et de diminution de l'activation générale [24].

1.3.2 Les expressions faciales

Tout d'abord, il est important de faire la distinction entre la reconnaissance des expressions faciales et la reconnaissance d'émotions. Les émotions résultent de plusieurs facteurs et peuvent être révélées par la voix, la posture, les gestes, la direction de regard et les expressions faciales. Par contre, les émotions ne sont pas la seule origine des expressions faciales. En effet, celles-ci peuvent provenir de l'état d'esprit (ex : la réflexion), de l'activité physiologique (la douleur ou la fatigue) et de la communication non verbale (émotion simulée, clignotement de l'œil, froncement des sourcils).

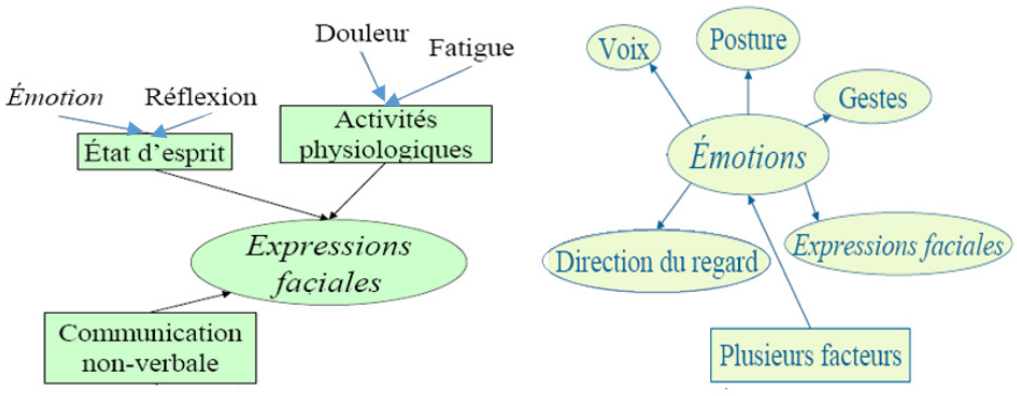


FIGURE 1.1 – Générateurs de l'expression faciale et de l'émotion.

1. Définition des expressions faciale

L'expression faciale est une mimique faciale chargée de sens. Le sens peut être l'expression d'une émotion, un indice sémantique ou une intonation dans la Langue des Signes. Les expressions du visage sont principalement générées par la contraction des muscles qui induisent des modifications temporaires des caractéristiques de forme du visage telles que le clignement des paupières, le haussement des sourcils, la forme de la bouche ou encore des modifications de la texture de la peau telle que l'apparition de rides ou de fossettes. Les changements sont souvent brefs, de l'ordre de quelques secondes (entre 250 ms et 5 secondes).
L'expression faciale est un aspect important du comportement et de la communication non verbale [18] où le changement dans le visage , perceptible visuellement ,dû à l'activation (volontaire ou non) de l'un ou de plusieurs des 44 muscles composant le visage (250000 expressions possibles). Les expressions faciales sont très importantes pour pouvoir connaître l'état de la personne. Grâce à ces expressions il est possible de faire plusieurs déductions et de récupérer plusieurs informations comme :

- L'état affectif que ce soit les émotions (peur, colère, joie, surprise, tristesse, dégoût) ou bien certaines humeurs.
- L'activité cognitive comme la concentration, l'ennui ou la perplexité.
- Le tempérament et la personnalité.

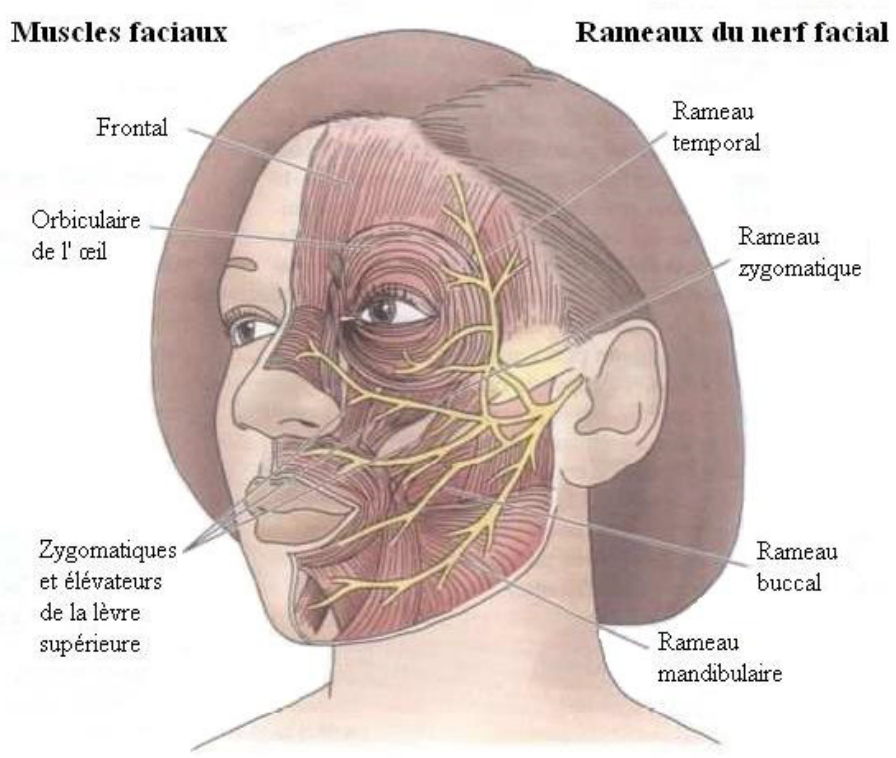


FIGURE 1.2 – Muscles faciaux et leur contrôle nerveux [1].

2. Différents types des expressions faciales :

Le tableau représenté ci-dessous (1.1) présente une classification des expressions faciales selon certains chercheurs à base d'émotion et d'inclusion.

Théoriciens	Émotions de base	Base d'inclusion
Plutchik	Acceptation, colère, anticipation, dégoût, joie, peur, tristesse, surprise	Relation aux processus biologiques adaptatifs.
Arnold	Colère, aversion, courage, abattement, désir, désespoir, peur, haine, espoir, amour, tristesse	Relation aux tendances d'action
Frijda	Désir, bonheur, intérêt, surprise, émerveillement, chagrin	Désir, bonheur, intérêt, surprise, émerveillement, chagrin
McDougall	Colère, dégoût, exaltation, peur, soumission, émotion tendre, émerveillement	Relation à l'instinct
Ekman , Friesen et Ellsworth	Colère, dégoût, peur, joie, tristesse, surprise	Expressions faciales universelles

TABLE 1.1 – Différentes classifications d'expressions faciales [57].

3. Description des six expressions faciales

Lors de la production d'une expression faciale, il apparait sur le visage un ensemble déformation au niveau des traits permanents du visage. Les émotions les plus fréquemment étudiées et utilisées sont les six émotions d'Ekman : la peur, la colère, la joie, la tristesse, le dégout, la surprise [68].

- Joie : Elle se caractérise par l'état d'une personne dans une condition de satisfaction intense [52]. Elle est due par rapport au désir, à la réussite, au bien-être, et l'accomplissement [37], mais aussi l'approche.
- Tristesse : c'est l'état d'une personne qui souffre moralement suite à une insatisfaction et des soucis [52]. Elle est souvent due soit à une perte, ou un deuil, ou un obstacle [37], la personne se replie souvent sur soi.

- Colère : c'est l'état d'une personne dans une réaction violente et agressive lors d'une contrariété [52]. Elle est souvent due soit à une injustice, ou un dommage, atteinte au système de valeurs [37]. La plupart des personnes en ce moment attaquent.

– Dégout : c'est l'état d'une personne qui a une répugnance pour certains aliments ou à un manque d'appétit [52]. Elle est souvent due soit à un rejet, ou contre quelqu'un, ou à une aversion [37]. La plupart des personnes en ce moment préfèrent se retirer.

– Peur : c'est l'état d'une personne menacée par un danger réel ou imaginaire. Elle est souvent due soit à une menace, ou à un danger, ou à des inconnus . La plupart des personnes en ce moment préfèrent prendre la fuite [37].

– Surprise : c'est l'état d'une personne étonnée par quelque chose d'inattendu. Elle est souvent due soit à un danger immédiat, ou à un imprévu, ou à des inconnus. La plupart des personnes en ce moment préfèrent prendre la fuite ou elles sursautent [52].

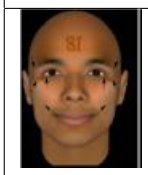
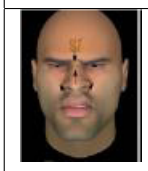
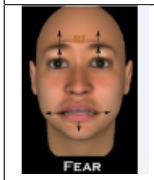
Expressions	Distance entre paupières	Distance entre œil et sourcil	Distance entre les coins de la bouche	Distance entre lèvre supérieure et lèvre inférieur	Distance entre les coins de l'œil et de la bouche
 Joie	Accroît ou décroît	Accroît ne change pas ou décroît	Accroît	Ne change pas ou accroît	Décroît
 Surprise	Accroît	Accroît	Ne change pas ou décroît	Accroît	Ne change pas ou accroît
 Colère	Décroît où accroît	Décroît	Ne change pas ou décroît	Accroît ne change pas ou décroît	Ne change pas ou accroît
 Dégout	Décroît	Décroît	Accroît ne change pas ou décroît	Accroît	Accroît ne change pas ou décroît
 Trisse	Décroît	Accroît	Ne change pas ou accroît	Ne change pas ou accroît	Ne change pas ou décroît
 Peur	Ne change pas ou accroît	Ne change pas ou accroît	Ne change pas ou décroît	Ne change pas ou accroît	Ne change pas ou décroît

TABLE 1.2 – Descriptions des six expressions faciales [52].

4. Les Systèmes de codification

Du point de vue physiologique, l'expression faciale est une conséquence de l'activité musculaire faciale. Ces muscles sont également appelés muscles mimétiques ou muscles des expressions faciales. L'étude de l'expression faciale ne peut se faire sans l'étude de l'anatomie du visage et de la structure sous-jacente des muscles faciaux.

Les chercheurs ont concentré leur attention sur un système de codage pour les expressions faciales. Plusieurs systèmes ont été proposés parmi eux l'outil d'Ekman. En 1978 Ekman a développé un outil de codification des expressions du visage largement utilisé aujourd'hui. Il s'intéresse désormais à l'analyse des expressions de manière informatique.[25] Ce qui suit nous définissant les principes de quelque systèmes :

1. **Système de Codification des Actions Faciales (Facial Action Coding System)FACS :**
   Le système a été spécialisé pour l'analyse des séquences vidéo d'une gamme d'individus et en associant les changements d'apparence faciale avec les contractions des muscles sous-jacents. Cette étude a permis le codage de 44 unités d'action distinctes (AUs) c'est-à-dire anatomiquement liées à la contraction de muscles faciaux spécifiques, chacune étant intrinsèquement liée à un petit ensemble d'activations musculaires localisées.
   Bien que FACS soit un système de description bénéficiant d'une grande maturité (environ vingt années de développement), il souffre cependant de quelques inconvénients [64] :
   Complexité : On estime qu'il faut 100 heures d'apprentissage pour en maîtriser les principaux concepts[25].
   Difficulté de manipulation par une machine : FACS a d'abord été créé pour des psychologues, Certaines mesures restent floues et difficilement évaluables par une machine.
   Manque de précision : les transitions entre deux états d'un muscle sont représentées de manière linéaire, ce qui est une approximation de la réalité. En particulier les mesures temporelles de l'activation des muscles faciaux (onset, apex et offset) ne sont pas mises en évidence.

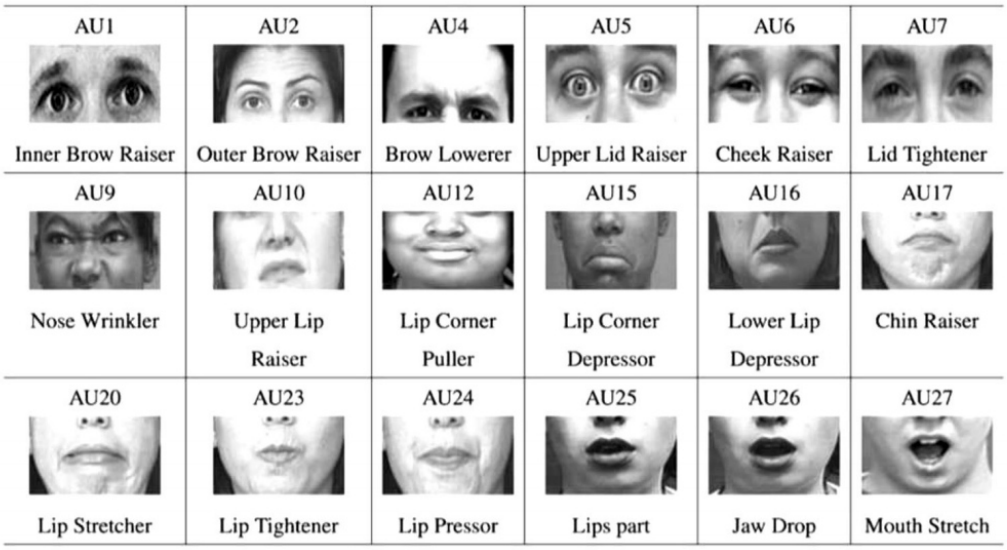


FIGURE 1.3 – Liste des Actions Units relatives aux 6 expressions faciales [64].

2. **MPEG4 :**
   La norme de codage vidéo MPEG-4 dispose d'un modèle du visage humain développé par le groupe d'intérêt Face and Body AdHoc Group . C'est un modèle 3D articulé. Ce modèle est construit sur un ensemble d'attributs faciaux, appelés Facial Feature Points (FFP). Des mesures sur ces FFP sont effectuées pour former des unités de mesure (Facial Animation Parameter Units) qui servent à la description des mouvements musculaires (Facial Animation Parameters - équivalents des Actions Unitaires d'Ekman).
   Les Facial Animation Parameter Units (FAPU) permettent de définir des mouvements élémentaires du visage ayant un aspect naturel. En effet, il est difficile de définir les mouvements élémentaires des muscles de manière absolue : le déplacement absolu des muscles d'une personne à l'autre change, mais leur déplacement relatifs à certaines mesures pertinentes sont constantes. C'est ce qui permet d'animer des visages de manière réaliste et peut permettre de donner des expressions humaines à des personnages non-humains. Comme exemples de FAPU, on peut citer la largeur de la bouche, la distance de séparation entre la bouche et le nez, la distance de séparation entre les yeux et le nez, etc. Par exemple, l'étirement du coin de la lèvre gauche (Facial Animation Parameter 6 stretch-l-cornerlip) est défini comme le déplacement vers la droite du coin de la lèvre gauche d'une distance égale à la longueur de la bouche. Les FAPUs sont donc des mesures qui permettent de décrire des mouvements élémentaires et donc des animations [25].

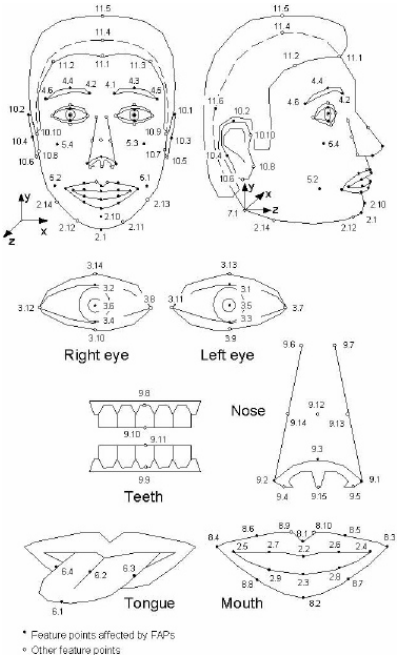


FIGURE 1.4 – Modèle MPEG4 l'ensemble d'attributs faciaux [25] .

3. **Candide** Candide est un modèle du visage, contenait 75 sommets et 100 triangles. Il est composé d'un modèle en fil de fer présentant un visage générique et d'un ensemble de paramètres [3] :

   (a) ***Paramètres de forme (Shape Units)***

   Ces paramètres permettent d'adapter le modèle générique à un individu particulier. Ils représentent les différences inter-individus et sont au nombre de 12 [3] :

   - Hauteur de la tête,
   - Position verticale des sourcils,
   - Position verticale des yeux,
   - Largeur des yeux,
   - Hauteur des yeux,
   - Distance de séparation des yeux,
   - Profondeur des joues,
   - Profondeur du nez, ;
   - Position verticale du nez,
   - Degré de courbure du nez (s'il pointe vers le haut ou non),
   - Position verticale de la bouche,
   - Largeur de la bouche.

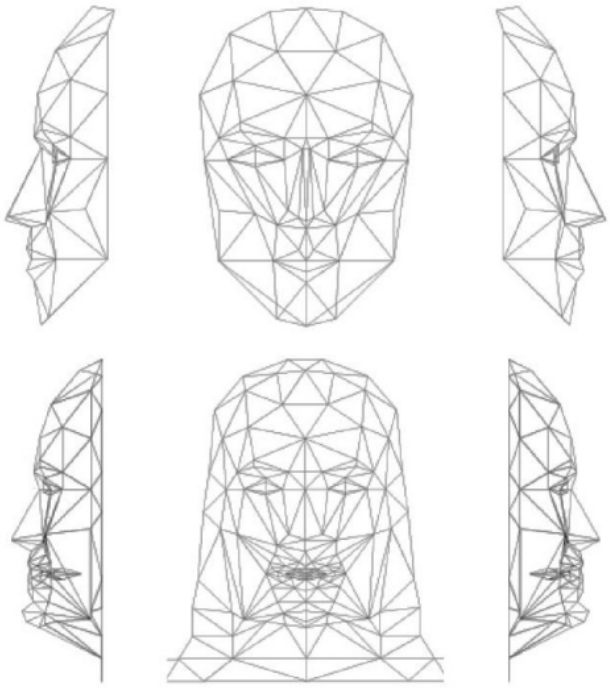


FIGURE 1.5 – En haut : CANDIDE-1 avec 79 sommets et 108 surfaces.En bas : CANDIDE-2 avec 160 sommets et 238 surfaces [3]

(b) ***Paramètres d'animation (Animation Units)*** Ces paramètres représentent les différences intra-individus, c'est-à-dire les différentes actions faciales. Ils sont composés d'un sous-ensemble de FACS et d'un sous-ensemble des FAPs de MPEG4. Les FAPs sont définis par rapport à leur FAPU correspondant. Ces paramètres, qu'ils soient d'animation ou de forme, sont représentés sous forme d'une liste de points du modèle de fil de fer à mettre à jour. Candide permet de voir clairement la différence entre les AUs de FACS et les FAPs de MPEG4 : les AUs de FACS sont exprimées de manière absolue, à la différence des FAPs qui sont exprimés par rapport à des mesures du visage (les FAPUs) [25].

1.4 Un système d'analyse des expressions faciales

Un système qui effectue une reconnaissance automatique des expressions faciales est généralement compose de trois modules principaux.La première étape consiste à détecter et enregistrer la région du visage dans les images. Par la suite, l'extraction des informations nécessaires qui décrivent au mieux l'expression. A la fin, en se basant sur ces informations, l'image sera affectée à une catégorie d'expressions à l'aide d'un classifieur. La figure 1.6 schématise les différentes étapes d'un système de reconnaissance des expressions faciales. D'autres filtres ou modules de pré-traitement de données peuvent être utilisés entre ces modules principaux pour améliorer les

résultats de détection, d'extraction de caractéristiques ou de classification.

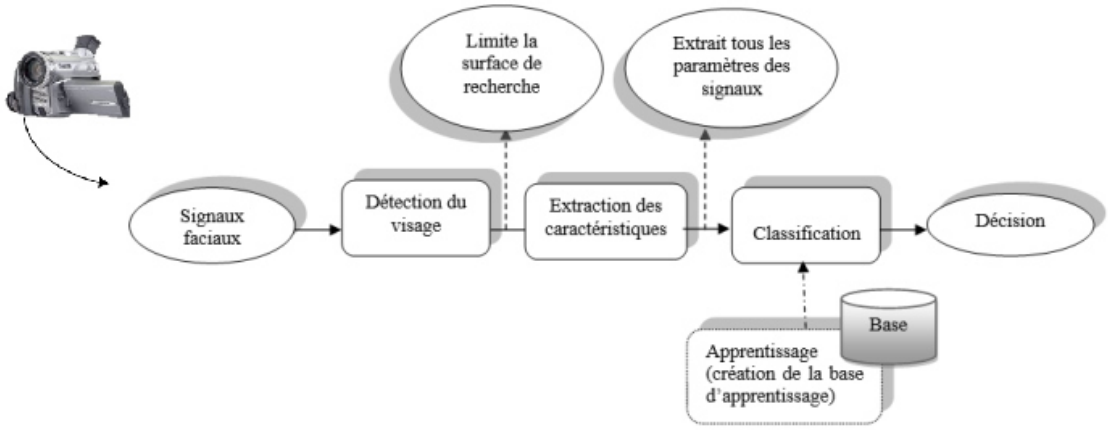


FIGURE 1.6 – Architecture d'un système de reconnaissance des expressions faciales

1.4.1 Détection du visage

La détection de visage consiste à déterminer la présence ou l'absence de visages dans une image et en cas de présence à déterminer sa localisation. C'est une tâche préliminaire nécessaire et fondamentale à la plupart des techniques d'analyse du visage. Les techniques utilisées sont généralement issues du domaine de la reconnaissance des formes. En effet, le problème peut être vu comme la détection de caractéristiques communes à l'ensemble des visages humains : il s'agit de comparer une image à un modèle générique de visage et d'indiquer s'il y a ou non ressemblance. La sortie d'un détecteur de visage indique le nombre de visages présents dans l'image. De plus, la plupart des détecteurs de visage actuels sont aussi des localisateurs de visages : ils renvoient une localisation des visages détectés (une boîte englobante par exemple) [17]
Les principales difficultés sont la robustesse aux différentes identités, poses du visage, expressions faciales et aux variations d'illumination [63][45].

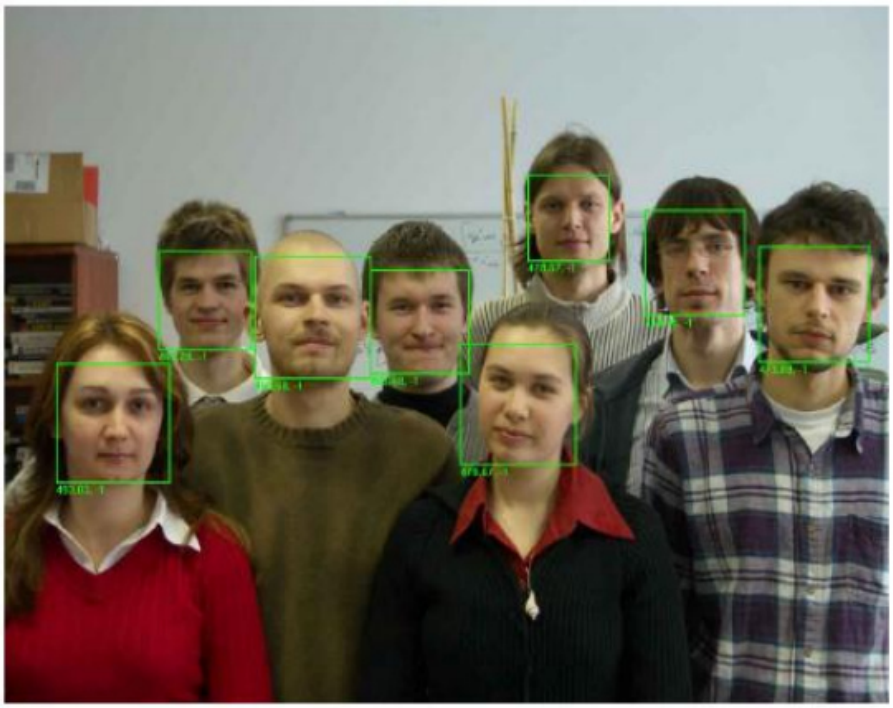


FIGURE 1.7 – Détection de visage

Caractéristiques d'un détecteur de visage

Un détecteur de visage idéal doit posséder les caractéristiques suivantes :

- Robustesse : il doit être capable de gérer les variations d'apparence selon la pose, la taille, l'éclairage, l'occlusion, les milieux complexes, les expressions faciales et la résolution.
- Rapidité : il doit être assez rapide pour effectuer le traitement en temps réel.
- Simplicité : il doit être simple. Par exemple, le temps de formation doit être court, le nombre de paramètres doit être petit, et les échantillons d'apprentissage doivent pouvoir être recueillis à moindre coût[44].

Les techniques de détection de visages

Il existe plusieurs techniques de détection du visage, nous citons les plus utilisées :

1. **Le traitement automatique du visage :** C'est une méthode qui spécifié les visages par des distances et des proportions entre des points particuliers comme les points autour des yeux, le nez, les coins de la bouche. Mais cette technique n'est pas efficace dans des situations de capture d'image avec peu d'éclairage [56].
2. **Eigen Face :** Les « Eigen Faces » est une méthode de caractérisation efficace dans des traitements faciaux tels que la détection et la reconnaissance du visage est basé sur la représentation des caractéristiques de visage à partir d'images modèles en niveau de gris [68].

3. **L'analyse des points particuliers** C'est une technique d'identification faciale la plus utilisée, elle est capable de s'adapter à des changements d'aspect facial comme le sourire, froncement des sourcils, et les grimasse les plus compliquées[68].
4. **LDA (Linear discriminant analysis) fisher :** Elle est basée sur l'analyse discriminante prédictive. Il s'agit d'expliquer et de prédire l'appartenance d'un individu à une classe prédéfinie à partir de ses caractéristiques mesurées à l'aide de variables prédictives [55].
5. **Méthode LBP (Local Binary Patterns) :** La technique des Modèles binaires locaux divisent le visage en sous–régions carrées de taille égale là où on calcule les caractéristiques LBP. Les vecteurs obtenus sont ensuite concaténés pour obtenir le vecteur de caractéristiques final[68].
6. **Filtre de Haar :** Cette méthode de détection du visage utilise un filtre multi-échelles de Haar. Les caractéristiques d'un visage sont décrites dans un fichier XML. Elles ne sont pas choisies au hasard et reposent sur un échantillon de quelques centaines d'images tests.Cette méthode proposée par Paul Viola et Michael Jones.

1.4.2 Extraction des caractéristiques

Une fois que le visage est détecté dans une image, la prochaine étape est l'extraction de caractéristiques du visage montré, qu'on appelle aussi les points caractéristiques ou ("Landmarks"). Ces points permettent d'encadrer les régions telles que les yeux, la bouche, le nez, les sourcils, etc.

La détection des points caractéristiques du visage commence habituellement à partir d'une boite englobante rectangulaire renvoyée par un détecteur de visage qui localise ce dernier. L'extraction de caractéristiques géométriques telles que les contours des composants faciaux, les distances faciales, etc. fournit les emplacements ou les caractéristiques d'apparence. peuvent être calculées. En raison de la grande variabilité dans les types de visages, il est très difficile pour la machine d'extraire les traits faciaux. De ce fait, les méthodes d'extraction des caractéristiques pour l'analyse d'expression peuvent êtres séparées en deux types d'approches : les méthodes basées sur les caractéristiques géométriques et les méthodes basées sur l'apparence [50] :

Les caractéristiques géométriques

Représentent la forme et l'emplacement des composants du visage (y compris la bouche, les yeux, les sourcils et le nez). Les composants faciaux ou les traits faciaux sont extraits pour former un vecteur de caractéristiques représentant la géométrie du visage.

Les caractéristiques d'apparence

Représentent les changements d'apparence (texture de la peau) du visage, tels que les rides et les sillons. Ces caractéristiques d'apparence peuvent être extraites sur tout le visage ou sur des régions spécifiques du visage. Selon les différentes méthodes d'extraction des caractéristiques, les effets de la rotation de la tête dans le plan et les différentes échelles de prise de vue du visage peuvent être éliminés par une normalisation de ce dernier avant l'extraction des caractéristiques ou par une représentation des caractéristiques avant l'étape de reconnaissance d'expression.

1.4.3 Classification des expressions

C'est la dernière étape d'un système de reconnaissance des expressions faciales. Cette étape consiste à reconnaitre l'ensemble de six expressions prototypes .Les travaux de recherche concernés par ce créneau sont divisés en trois parties, à savoir, les approches globales, les approches locales, et enfin les approches hybrides. Chaque famille d'approches présente ses avantages et ses inconvénients vis-à-vis des problèmes liés aux conditions environnementales, le changement de l'échelle, les orientations des images, les positions de la tête... etc.

Méthode globale

Elles sont basées sur des techniques d'analyse statistique bien connues. Il n'est pas nécessaire de repérer certains points caractéristiques du visage (comme les centres des yeux, les narines, le centre de la bouche, etc.) à part pour normaliser les images. Dans ces méthodes, les images de visage (qui peuvent être vues comme des matrices de valeurs de pixels) sont traitées de manière globale et sont généralement transformées en vecteurs, plus faciles à manipuler. L'avantage principal des méthodes globales est qu'elles sont relativement rapides à mettre en œuvre et que les calculs de base sont d'une complexité moyenne. En revanche, elles sont très sensibles aux variations d'éclairement, de pose et d'expression faciale.
Ceci se comprend aisément puisque la moindre variation des conditions de l'environnement ambiant entraîne des changements inéluctables dans les valeurs des pixels qui sont traités directement. Ces méthodes utilisent principalement une analyse de sous-espaces de visages. L'utilisation de techniques de modélisation de sous-espace a fait avancer la technologie de reconnaissance faciale de manière significative.
Nous pouvons distinguer deux types de techniques parmi les méthodes globales : les techniques linéaires et les techniques non linéaires.
Les techniques linéaires projettent linéairement les données d'un espace de grande dimension (par exemple, l'espace de l'image originale) sur un sous-espace de dimension inférieure. Malheureusement, ces techniques sont incapables de préserver les variations non convexes des variétés (géométriques donc au sens mathématique du terme) de visages afin de différencier des indivi-

dus. Dans un sous-espace linéaire, les distances euclidiennes et plus généralement les distances de Mahalanobis, qui sont normalement utilisées pour faire comparer des vecteurs de donnés, ne permettent pas une bonne classification entre les classes de formes et entre les individus eux-mêmes. La technique linéaire la plus connue et sans aucun doute l'analyse en composantes principales (PCA), également appelée transformée de Karhunen-Loeve. Le PCA fut d'abord utilisé afin de représenter efficacement des images de visages humains. [10] [59] Bien que ces méthodes globales linéaires basées sur l'apparence évitent l'instabilité des toutes premières méthodes géométriques qui ont été mises au point, elles ne sont pas assez précises pour décrire les subtilités des variétés (géométriques) présentes dans l'espace de l'image originale. Ceci est dû à leurs limitations à gérer la non-linéarité en reconnaissance faciale : les déformations de variétés non linéaires peuvent être lissées et les concavités peuvent être remplies, causant des conséquences défavorables.
Afin de pouvoir traiter ce problème de non-linéarité sur la reconnaissance des expressions faciales, de telles méthodes linéaires ont été étendues à des techniques non linéaires basées sur a notion mathématique de noyau ("kernel") comme le Kernel PCA et le Kernel LDA [34].

Méthodes locales

Elles sont basées sur des modèles, utilisant des connaissances a priori que l'on possède sur la morphologie du visage et s'appuie en général sur des points caractéristiques de celui-ci.
Kanade a présenté un des premiers algorithmes de ce type [29] en détectant certains points ou traits caractéristiques d'un visage puis en les comparants avec des paramètres extraits d'autres visages. Ces méthodes constituent une autre approche pour prendre en compte la non-linéarité en construisant un espace de caractéristiques local et en utilisant des filtres d'images appropriés, de manière à ce que les distributions des visages soient moins affectées par divers changements.
Les approches bayésiennes, les machines à vecteurs de support (SVM) [15], la méthode des modèles actifs d'apparence (AAM) [4] ou encore la méthode "local binary pattern"(LBP) ont été utilisées dans ce but.
Toutes ces méthodes ont l'avantage de pouvoir modéliser plus facilement les variations de pose, d'éclairage et d'expression par rapport aux méthodes globales. Toutefois, elles sont plus lourdes à utiliser puisqu'il faut souvent placer manuellement un assez grand nombre de points sur le visage alors que les méthodes globales ne nécessitent de connaître que la position des yeux afin de normaliser les images, ce qui peut être fait automatiquement et de manière assez fiable par un algorithme de détection [6]

Méthodes hybrides

Ce sont une combinaison des méthodes globales et locales en combinant la détection de caractéristiques géométriques (ou structurales) avec l'extraction de caractéristiques d'apparence

locales. Elles permettent d'augmenter la stabilité de la performance de reconnaissance lors de changements de pose, d'éclairement et d'expressions faciales. [54]

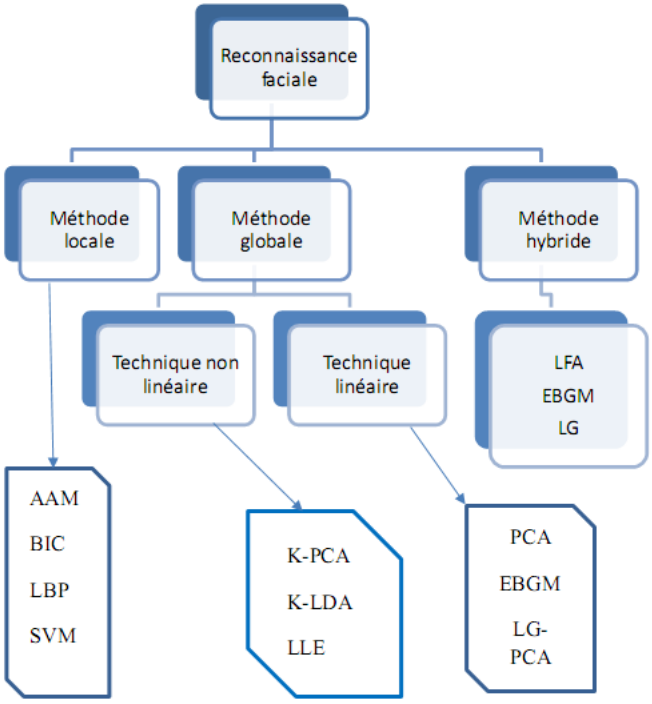


FIGURE 1.8 – Classification des algorithmes principaux utilisés en reconnaissance faciale[4][54] .

Synthèse sur la classification des expressions faciales

Le tableau suivant tableau (1.3) présente quelques techniques récentes développées pour la reconnaissance des expressions faciales :

Références	Techniques	Expressions faciales	Bases de données	Types	Sujets	Performance
[58]	LPB, SVM	Six	CK+	images	50	97%
[26]	EBGM,ELM	Six	CK+	Images	40	96%
[40],[39]	PCA, LDA	Six	CK+	193 images	9	92% 75%
[69]	SVM	Sourire spontané et sourire sournois	MMI	vidéos	52	94%
[46]	HOG, SVM	Six	CK+	images		90%
[12]	HMMS	Sept	Propre base Cohn-kanade	Vidéos	5 sujets 53 sujets	84,46% 58,63%
[17]	MAA	Sept	CMU	166 images	30 sujets	84, 34%
[53]	Codification Des règles	Six		images	8 sujets	92%
[41]	Multiboots, SVM linéaire	Six	BU-3DDFE	Scans 3D	60 sujets	97,75% 98,81%
[23]	CNN	Sept	Fer2013	images	35 000 sujets	66,67%
[21]	CNN	Six	CK+	images	123 sujets	91%

TABLE 1.3 – Synthèse des travaux développés.

Bases des données d'expressions faciales

L'un des aspects les plus importants du développement de tout nouveau système de reconnaissance ou de détection d'expression faciale est le choix de la base de données qui sera utilisée pour tester ce nouveau système. De plus, des bases de données communes sont nécessaires pour évaluer les algorithmes de manière comparative. Les bases de données disponibles peuvent être classées en deux catégories : les bases d'expressions faciales spontanées et les bases d'expressions faciales posées[35]

Exemple de base des données

La base de données	Description	Lien
Ck+(Cohn-Kanade)	593 séquences vidéo à la fois posées et non posées émotions (spontanées) 123 sujets âgés de 18 à 30 ans Fournit des protocoles et des résultats de base pour le visage suivi des fonctionnalités, unités d'action et reconnaissance des émotions Résolutions d'image de 640 X480 et 640 X490	http ://www.consortium.ri.cmu.edu /ckagree/
CE(Compound Emotion)	5060 images correspondant à 22 catégories de base et émotions composées 230 sujets humains (130 femmes et 100 hommes, l'âge moyenne 23 ans) Comprend la plupart des races Résolution d'image de 3000X 4000	http ://cbcsl.ece.ohio-state.edu/ dbform-compound.html
DISFA	130 000 images vidéo stéréo en haute résolution 27 sujets adultes (12 femmes et 15 hommes) 66 points de repère du visage pour chaque image Résolution d'image de 1024X 768	http : //www.engr.du.edu/mmahoor /DISFA.htm
BU-3DFE	Visages humains 3D et émotions faciales 100 sujets dans la base de données, 56 femmes et 44 hommes, avec environ six émotions 25 modèles d'émotion faciale 3D par sujet Résolution d'image de 1040 1329	http ://www.cs.binghamton.edu/ lijun/Research/3DFE/3DFE- Analysis.html
JAFFE	213 images de sept émotions faciales Dix modèles féminins japonais Six adjectifs d'émotion de 60 sujets japonais Résolution d'image 256 256	http ://www.kasrl.org/jaffe-info.html
B+	16 128 images faciales 28 sujets distincts pour 576 conditions d'observation Résolution d'image de 320 243	http ://vision.ucsd.edu/content/ extended-yale-face-database-b-b
MMI	Plus de 2900 séquences vidéo et images haute résolution images de 75 sujets-238 séquences vidéo sur 28 sujets, hommes et femmes-Résolution d'image de 720 - 576	https ://mmifacedb.eu/

BP4D-Spontanous	La base de données vidéo 3D comprend 41 participantes (23 femmes, 18 hommes), avec des émotions faciales spontanées 11 Asiatiques, six Afro-Américains, quatre Hispaniques, et 20 euro-américains Résolution d'image de 1040 1329	http ://www.cs.binghamton.edu/ lijun/Research/3DFE/3DFE-Analysis.html
KDEF	4900 images d'expressions faciales humaines d'émotion 70 individus, sept expressions émotionnelles différentes avec 5 angles différents Résolution de l'image 562 762	http ://www.emotionlab.se/ resources/kdef
FER2013	compte environ 37 000 personnes bien structurées En Images de gris à 48X48 pixels les visages regroupés automatiquement par l'API Google de recherche d'image	https ://www.kaggle.com/datasets

TABLE 1.4 – Exemple de bases des données [35]

Dans ce projet, nous avons utilisé une base des données FER2013 fourni par Kaggle.

1.5 Conclusion

Dans ce chapitre nous avons expliqué dans un premier temps les problématiques qui stimule la recherche dans le domaine de reconnaissance des expressions faciales et la définition des émotions. Dans un second temps,nous avons discuté sur les théories et les représentations les plus connues dans la reconnaissance d'expressions faciales : les techniques de codifications, les approches de détections de visages dans des images ainsi que l'extraction des caractéristiques. Finalement, les bases de données fréquemment utilisées ont été décrites.

Dans le chapitre suivant nous présenterons l'apprentissage profond en citant les différents types de réseaux de neurones et en détaillant les réseaux de neurones convolutifs .

CHAPITRE 2

DEEP LEARNING

2.1 Introduction

Le Deep Learning est un nouveau domaine de recherche du Machine Learning, qui a été introduit dans le but de rapprocher le ML de son objectif principal : L'intelligence artificielle. Il concerne les algorithmes inspirés par la structure et le fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données.

Dans ce deuxième chapitre, nous présentons quelques notions concernant réseaux de neurones

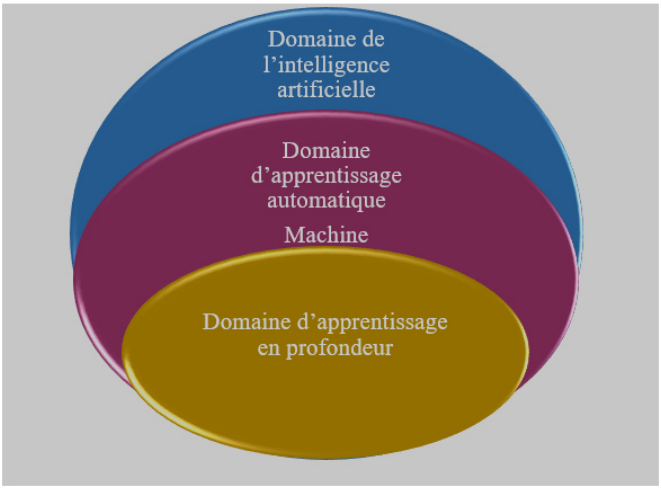


FIGURE 2.1 – La relation entre l'intelligence artificielle, le ML et le deep Learning

telles que leurs définitions, historiques et leurs différentes topologies. Ensuite, nous décrivons le deep learning et ses différentes architectures existantes. Enfin, nous intéressons durant cette uniquement sur Les réseaux de neurones à convolution CNN.

2.2 L'Apprentissage En Profondeur (Le Deep Learning)

Le terme "Deep Learning" ou Apprentissage profond, a été introduit pour la première fois au ML par Dechter (1986), et aux réseaux neuronaux artificiels par Aizenberg et al (2000) [5].

2.2.1 Définition

L'apprentissage en profondeur est un ensemble d'algorithmes d'apprentissage automatique qui tentent d'apprendre à plusieurs niveaux, correspondant à différents niveaux d'abstraction. Il a la capacité d'extraire des caractéristiques à partir des données brutes grâce aux multiples couches de traitement composé de multiples transformations linéaires et non linéaires et apprendre sur ces caractéristiques petites à petit à travers chaque couche avec une intervention humaine minime [48]

FIGURE 2.2 – Un processus de Deep Learning : les images sont transmises à un réseau, qui apprend automatiquement les caractéristiques et classe les objets.[49]

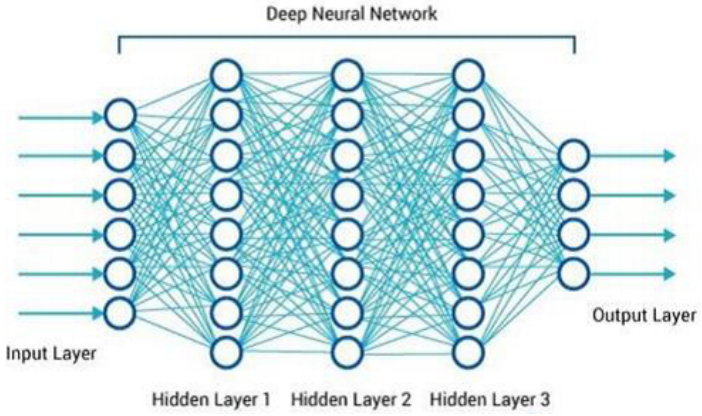


FIGURE 2.3 – Schéma illustratif de DL avec plusieurs couches [20].

2.2.2 Histoire Du Deep Learning

Le tableau si dessous resume l'historique de deep learning :

Année	Contributeur	Contribution
300 AC	Aristotle	introduction de l'associationnisme, début de l'histoire des humains qui essayent de comprendre le cerveau
1873	Alexander Bain	introduction du Neural Groupings comme les premiers modèles de réseaux de neurones
1943	McCulloch and Pitts	introduction du McCulloch–Pitts (MCP) modèle considérer comme L'ancêtre des réseaux de neurones artificielles
1949	Donald Hebb	considérer comme le père des réseaux de neurones, il introduit la règle d'apprentissage de Hebb qui servira de fondation pour les réseaux de neurones modernes
1958	Frank Rosenblatt	introduction du premier perceptron
1974	Paul Werbos	introduction de la rétro-propagation
1980	Teuvo Kohonen	introduction des cartes auto organisatrices
1980	Kunihiko Fukushima	introduction du Neocognitron, qui a inspiré les réseaux de neurones convolutif
1982	John Hopfield	introduction des réseaux de Hopfield
1985	Hilton and Sejnowski	introduction des machines de Boltzmann
1986	Paul Smolensky	introduction de Harmonium, qui sera connu plus tard comme machines de Boltzmann restreintes
1986	Michael I. Jordan	définition et introduction des réseaux de neurones récurrent
1990	Yann LeCun	introduction de LeNet et montra la capacités des réseaux de neurones profond
1997	Schuster and Paliwal	introduction des réseaux de neurones récurrent bidirectionnelles
1997	Hochreiter and Schmidhuber	introduction de LSTM, qui ont résolu le problème du vanishing gradient dans les réseaux de neurones récurrent
2006	Geoffrey Hinton	introduction des Deep belief Network
2009	Salakhutdinov and Hinton	introduction des Deep Boltzmann Machines
2012	Alex Krizhevsky	introduction de AlexNet qui remporta le challenge ImageNet

TABLE 2.1 – Les étapes majeurs du Deep Learning [71]

2.2.3 Pour quoi le choix Deep Learning

Les algorithmes de ML fonctionnent bien pour une grande variété de problèmes. Cependant ils ont échoues à résoudre quelques problèmes majeurs de l'IA telle que la reconnaissance vocale et la reconnaissance d'objets.[48] Tout d'abord les différents algorithmes du deep Learning ne sont apparus qu'à l'échec de l'apprentissage automatique tentant de résoudre une grande variété de problèmes de l'intelligence artificielle (l'IA) :

- Afin d'améliorer le développement des algorithmes traditionnels dans de telles tâches de l'IA.
- De développer une grande quantité de données telle que les big data.
- De s'adapter à n'importe quel type de problème.
- D'extraire les caractéristiques de façon automatique [48].

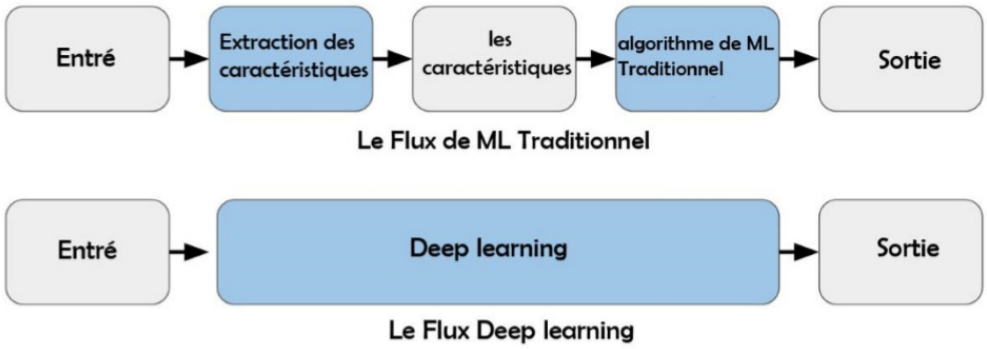


FIGURE 2.4 – Comparaison entre la machine Learning et le Deep Learning.[49]

Donc l'apprentissage en profondeur utilise des réseaux de neurones pour apprendre des représentations utiles de caractéristiques directement à partir de données.

2.3 Réseaux De Neurones

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau.

2.3.1 Définition

Un réseau de neurones artificiels est un système dont la conception est à l'origine inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques[49]. Les réseaux de neurones artificiels sont des réseaux fortement connectés par des

processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire (neurone artificiel) calcule une sortie unique sur la base des informations qu'ils reçoivent.
Les points essentiels qu'on peut retenir sur les réseaux de neurones sont les suivants :

- Ce sont des systèmes composés de neurones répartis en plusieurs couches connectées entre elles
- Ces systèmes peuvent résoudre de divers problèmes statistiques en général, et spécialement des problèmes de classification, en calculant à partir de l'entrée du réseau le score (ou la probabilité) de chaque classe. La classe attribuée à l'objet c'est celle de la probabilité la plus élevé
- L'entrée de chaque couche les données sont traitées et transformée en calculant une combinaison linéaire puis en appliquant une fonction non-linéaire, appelée fonction d'activation. Les coefficients de la combinaison linéaire définissent les paramètres (ou poids) de la couche
- La dernière couche calcule les probabilités finales à partir d'une fonction d'activation (classification binaire) ou la fonction softmax(classification multi-classes)
- On associée aussi une fonction de perte loss-function à la couche finale pour calculer l'erreur de classification. Il s'agit en général de l'entropie croisée (accurancy)
- On calcule les poids des couches par rétro-propagation du gradient : Calcule les paramètres qui minimisent la fonction de perte régularisée progressivement on partant de la dernière couche à la première couche, L'optimisation se fait avec une descente du gradient stochastique. [49]

2.3.2 Historique

Les réseaux neuronaux ont vu le jour qu'en 1943 par W.MCCulloch et W. Pitts du neurone formel qui est une abstraction du neurone physiologique. Par cette présentation, ils ont pu démontrer que le cerveau est équivalent à une machine de Turing, la pensée devient alors purement des mécanismes matériels et logiques. Ils déclarèrent en 1955 "Plus nous apprenons de choses au sujet des organismes, plus nous sommes amenés à conclure qu'ils ne sont pas simplement analogues aux machines, mais qu'en est-il de cette machine. La démonstration de McCulloch et Pitts a été l'un des acteurs importants de la création de la cybernétique.
En 1949, D. Hebb présenta dans son ouvrage "The Organization of Behavior" une règle d'apprentissage. De nombreux modèles de réseaux aujourd'hui s'inspirent encore de la règle de Hebb. En 1958, F. Rosenblatt développe le modèle du Perceptron. C'est un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones : une couche de perception et une couche liée à la prise de décision. C'est le premier système artificiel capable d'apprendre par expérience. Dans la même période, le modèle de L'Adaline (ADAptive LINar Element) a été présenté par B. Widrow, chercheur américain à Stanford. Ce modèle sera par la suite le modèle de base des réseaux multicouches. En 1969, M. Minsky et S. Papert publient une critique des propriétés du

Perceptron. Cela va avoir une grande incidence sur la recherche dans ce domaine. Elle va fortement diminuer jusqu'en 1972, où T. Kohonen présente ses travaux sur les mémoires associatives et propose des applications à la reconnaissance de formes. C'est en 1982 que J. Hopfield présente son étude d'un réseau complètement rebouclé, dont il analyse la dynamique.
Aujourd'hui, les réseaux neuronaux sont utilisés dans de nombreux domaines (entre autres, vie artificielle et intelligence artificielle) à cause de leur propriété en particulier, leur capacité d'apprentissage, et qu'ils soient des systèmes dynamiques [20].

2.3.3 Topologie

Chaque réseau de neurones est connecté entre eux de diverses manières. Dans la figure suivante. Nous pouvons distinguer deux familles de réseaux de neurones : non bouclés ou statiques (a) et (b) et bouclés (dynamiques) (c) et (d).

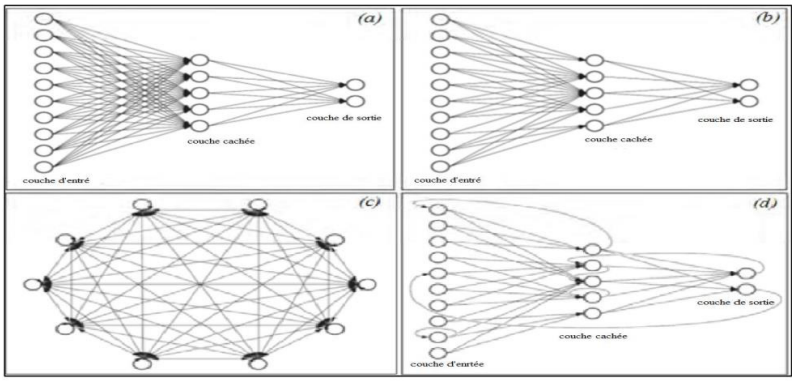


FIGURE 2.5 – Topologie des Réseaux de neurones artificiels.

2.3.4 Les différentes Architectures du Deep Learning

Bien qu'il existe un grand nombre de variantes d'architectures profondes. Il n'est pas toujours possible de comparer les performances de toutes les architectures, car elles ne sont pas toutes évaluées sur les mêmes ensembles de données. Le Deep Learning est un domaine à croissance rapide, et de nouvelles architectures, variantes ou algorithmes apparaissent toutes les semaines.

1–Les Réseaux de Neurones Convolutifs

Convolutional Neural Network (CNN) (réseaux de neurones convolutifs) sont un type de réseau de neurones spécialisés pour le traitement de données ayant une topologie semblable à une grille. Qui se sont avérés très efficaces dans des domaines tels que la reconnaissance et la classification d'images et vidéos. CNN a réussi à identifier les visages, les objets, panneaux de circulation et

auto-conduite des voitures [8]. Récemment, les CNN ont été efficaces dans plusieurs tâches de traitement du langage naturel (telles que la classification des phrases) [32]. [70] [14].
Dans le ML, un réseau convolutif est un type de réseau de neurones feed-forward, il a été inspiré par des processus biologiques [42]. Il existe quatre (4) principales opérations illustrées dans le CNN à savoir :

- La couche convolution
- La couhe Rectified Linear Unit
- La couche Pooling
- La couche entièrement connectée

2–Réseau de Neurones Récurrents

L'idée derrière les RNN est d'utiliser des informations séquentielles. Dans un réseau neuronal traditionnel, nous supposons que toutes les entrées (et les sorties) sont indépendantes les unes des autres. Mais pour de nombreuses tâches, c'est une très mauvaise idée. Si on veut prédire le prochain mot dans une phrase, il faut connaître les mots qui sont venus avant. Les RNN sont appelés récurrents, car ils exécutent la même tâche pour chaque élément d'une séquence, la sortie étant dépendante des calculs précédents.
Une autre façon de penser les RNN est qu'ils ont une « mémoire » qui capture l'information sur ce qui a été calculé jusqu'ici. En théorie, les RNN peuvent utiliser des informations dans des séquences arbitrairement longues, mais dans la pratique, on les limite à regarder seulement quelques étapes en arrière.[20].
Il est utilisé pour :

- La modélisation du langage et génération de texte
- La traduction automatique
- La reconnaissance vocale
- Et la description des images

3–Modèle Génératif

Si les modèles discriminatifs comme (CNN, RNN) sont utilisés pour prédire les données du label et de l'entrée, tant disque le modèle génératif décrit comment générer les données, il apprend et fait des prédictions en utilisant la loi de Bayes [51].
Cependant les modèles génératifs sont capables de bien plus que la simple classification comme par exemple générer de nouvelles observations.
Voici quelques exemples de modèle génératif :

- Boltzmann Machines [2]
- Restricted Boltzmann Machines [60]
- Deep Belief Networks [31]

– Deep Boltzmann Machines [65]
– Generative Adversarial Networks
– Generative Stochastic Networks [28]
– Adversarial auto encoders [19]

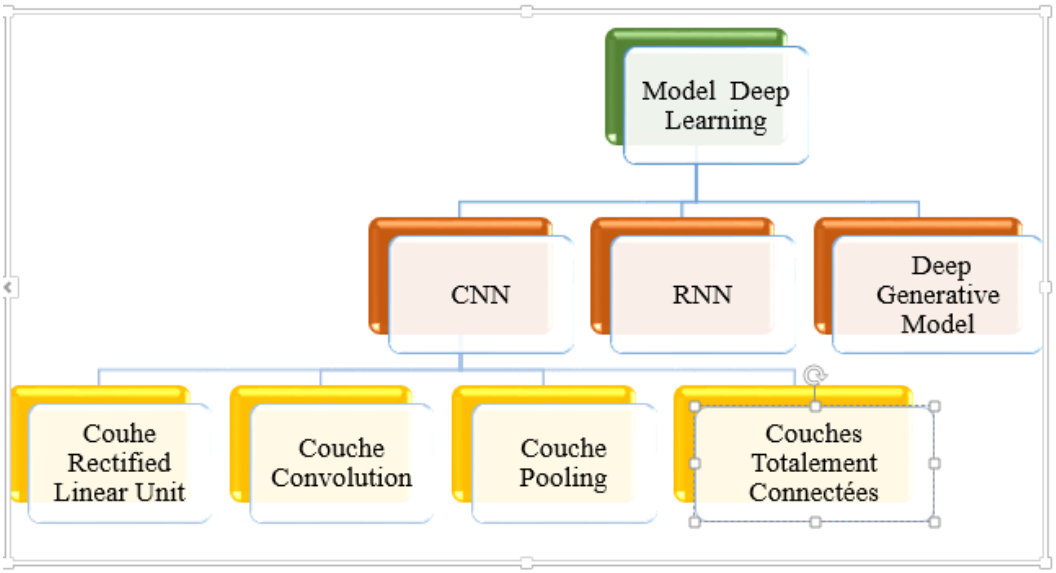


FIGURE 2.6 – Différents modèles du Deep Learning

2.3.5 Exemples d'Application de Deep Learning

Les applications du Deep Learning sont utilisées dans divers secteurs, de la conduite automatisée aux dispositifs médicaux [20].Grâce au deep Learning nous pouvons maintenant

– Faire une colorisation des images en noir et blanc.
– Ajouter des sons à des films silencieux.
– Faire de la traduction automatique.
– Faire de classification des objets en photographies.
– Générer d'écriture automatique.
– Génération de légende d'image.
– Jeu automatique

2.4 Réseaux de Neurones Convolutifs CNN

Durant cette phase nous baserons uniquement sur le CNN.

2.4.1 Présentation

Les réseaux de neurones convolutifs CNN (Convolutional Neural Network) sont les structures les plus performantes dans des domaines tels que la reconnaissance et la classification d'images. CNN ont réussi à identifier les visages, les objets, panneaux de circulation et auto-conduite des voitures, durant cette phase de formation, nous nous baserons uniquement sur le CNN.
Le nom « réseau de neurones convolutif » indique que le réseau emploie une opération mathématique appelée convolution. La convolution est une opération linéaire spéciale. Les réseaux convolutifs sont simplement des réseaux de neurones qui utilisent la convolution à la place de la multiplication matricielle dans au moins une de leurs couches.
Ils comportent deux parties principales. S'il y on a en entrée, une image qu'elle doit être sous la forme d'une matrice de pixels, de 2 dimensions pour une image en niveaux de gris et en 3 dimensions si elle est en couleur, pour représenter les couleurs fondamentales [Rouge, Vert, Bleu]. Cette image passe par la première partie d'un CNN qui est la partie convolutive Elle fonctionne comme un extracteur de caractéristiques des images. L'image passe à travers une succession de filtres, ou noyaux de convolution, pour la transformé en nouvelles images appelées cartes de convolutions « feature maps ». Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN. Le résultat en sortie de la partie convolutive est branché en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron multicouche) qui consiste à combiner les caractéristiques de tous le réseau pour classer l'image et à la sortie qui est une couche comportant un neurone par classe, on obtient des valeurs numériques généralement normalisées entre 0 et 1, pour présenter la distribution de probabilité sur les classes [49]

2.4.2 Architecture de Réseaux de Neurone Convolutifs

Comme nous l'avons mentionnée précédemment, les réseaux de neurones convolutifs sont basés sur le perceptron multicouche(MLP).
L'architecture CNN est formée par un empilement de couches de traitement indépendantes :

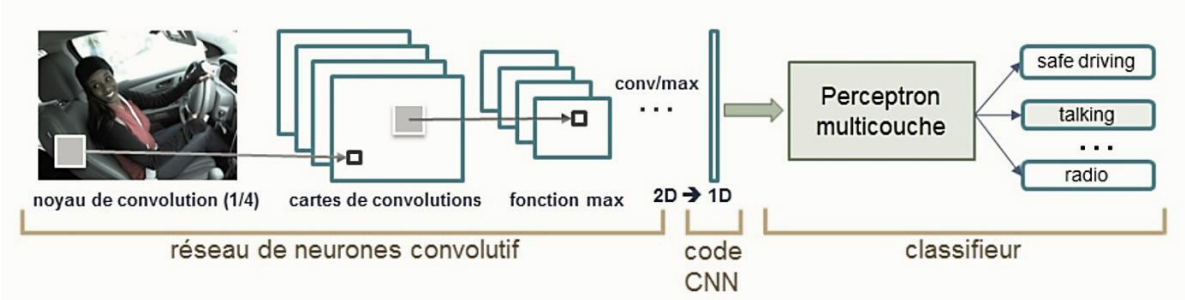


FIGURE 2.7 – Architecture standard d'un réseau de neurone convolutionnel [47]

.

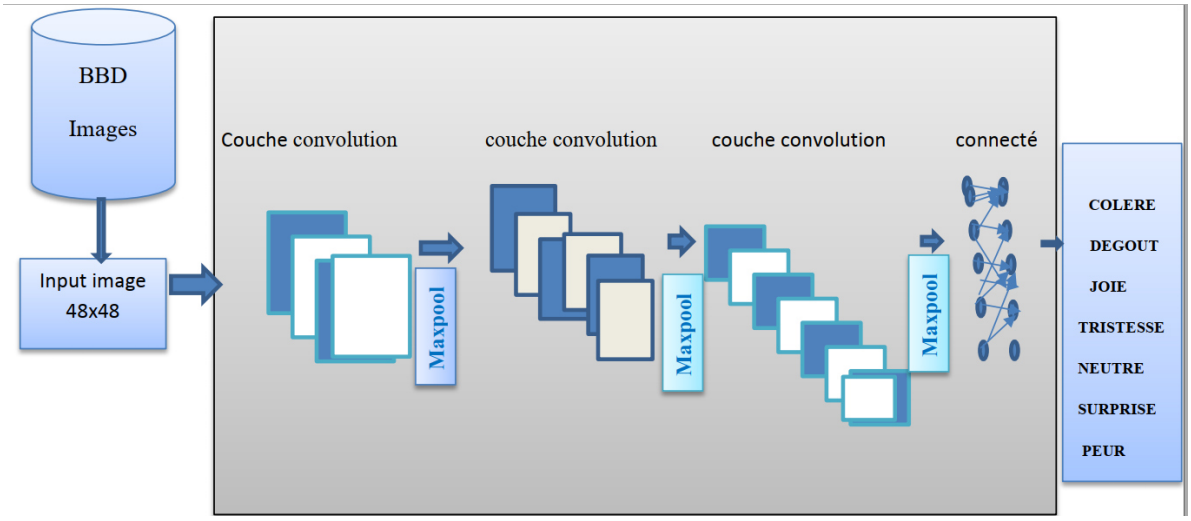


FIGURE 2.8 – Architecture Proposée

2.4.3 Les Différentes Couches de CNN

Il existe trois principales couches dans le CNN, tout en ayant un rôle bien défini.

1–La Couche De Convolution (CONV)

Trois hyper paramètres permettent de dimensionner le volume de la couche de convolution (aussi appelé volume de sortie) : La profondeur, le pas et la marge.

- Profondeur de la couche : nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
- Le pas : contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.
- La marge (à 0)' ou 'zero padding ' : parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. La taille de ce 'zero-padding' est le troisième hyper-paramètre. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée [47]

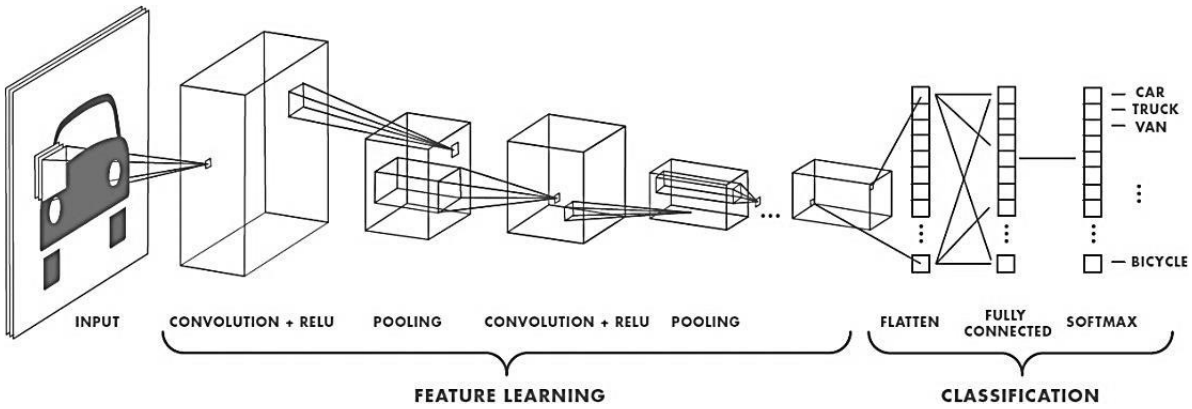


FIGURE 2.9 – Exemple de réseau composé de nombreuses couches à convolution. Des filtres sont appliqués à chaque image utilisée pour l'apprentissage à différentes résolutions, et la sortie de chaque image convoluée est utilisée comme entrée de la couche suivante[49]

.

Dans la terminologie du réseau convolutif, le premier argument de la convolution est souvent appelé l'entrée (input) et le second argument comme noyau (kernel). La sortie est parfois appelée la carte des caractéristiques (feature map).

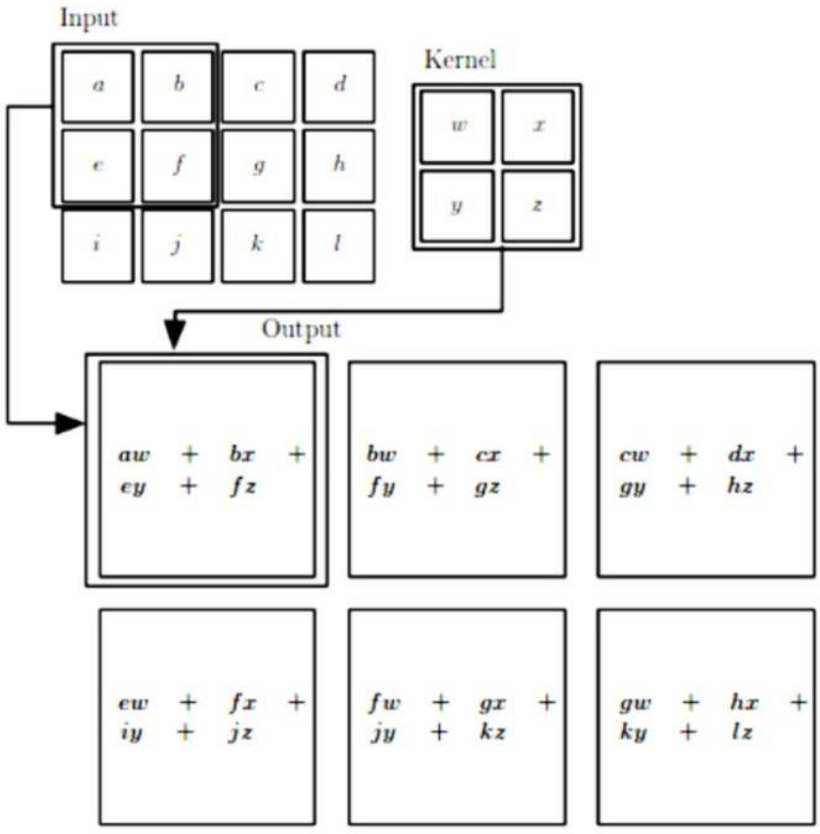


FIGURE 2.10 – Exemple d'une convolution 2D.[27]

Les Différentes Convolutions Il existe plusieurs types de convolutions, même si en général on utilise celle de base, il peut s'avérer utile de connaître les outils à notre disposition.

- La convolution classique qui représente le décalage du noyau entre chaque calcul, et le padding qui est la manière dont on peut « dépasser » de l'image pour appliquer la convolution.
- La dilated convolution, identique à la convolution à ceci près que le kernel est éclaté (on prend, par exemple, un pixel sur deux pour calculer la convolution). Il y a un paramètre supplémentaire : la dilation rate, qui est le nombre de pixels à ignorer.
- La transposed convolution qui construit la sortie comme si on inversait une convolution sur l'image
- La séparable convolution, qui est une convolution décomposable en convolutions plus simples.

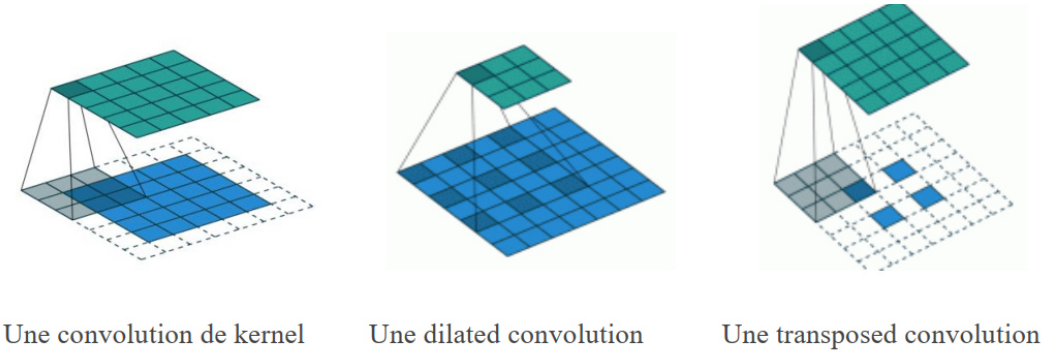


FIGURE 2.11 – Différents types de convolutions.

2–La Couche De Pooling

Le pooling est une forme de sous-échantillonnage de l'image, il permet de réduire progressivement la taille des représentations afin de réduire la quantité de paramètres et de calcul dans le réseau ainsi que l'invariance aux petites translations, il est donc fréquent d'insérer périodiquement une couche de pooling entre deux couches convolutives successives d'une architecture CNN pour contrôler le sur-apprentissage. L'opération de pooling créait aussi une forme d'invariance par translation. La couche de pooling fonctionne indépendamment sur chaque tranche de profondeur de l'entrée et la redimensionne uniquement au niveau de la surface. La forme la plus courante est une couche de mise en commun avec des filtres de taille 2x2 (largeur/hauteur) et comme valeur de sortie la valeur maximale en entrée. On parle dans ce cas de « Max-Pool 2x2 ». Le pooling permet de gros gains en puissance de calcul. Cependant, en raison de la réduction agressive de la taille de la représentation et donc de la perte d'information associée, la tendance actuelle est d'utiliser de petits filtres (type 2x2). Il est aussi possible d'éviter la couche de pooling mais cela implique un risque sur-apprentissage plus important

Il existe plusieurs types de pooling :

- Le « max pooling », qui revient à prendre la valeur maximale de la sélection. C'est le type

le plus utilisé, car il est rapide à calculer (immédiat), et permet de simplifier efficacement l'image .

– Le « mean pooling » (ou average pooling), soit la moyenne des pixels de la sélection : on calcule la somme de toutes les valeurs et on divise par le nombre de valeurs. On obtient ainsi une valeur intermédiaire pour représenter ce lot de pixels
– Le « sum pooling », c'est la moyenne sans avoir divisé par le nombre de valeurs (on ne calcule que leur somme)
– La séparable convolution, qui est une convolution décomposable en convolutions plus simples.

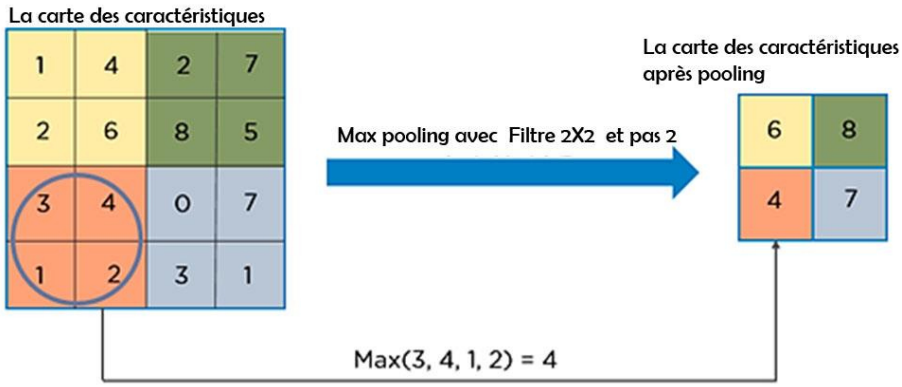


FIGURE 2.12 – Pooling avec un filtre 2x2 et un pas de 2

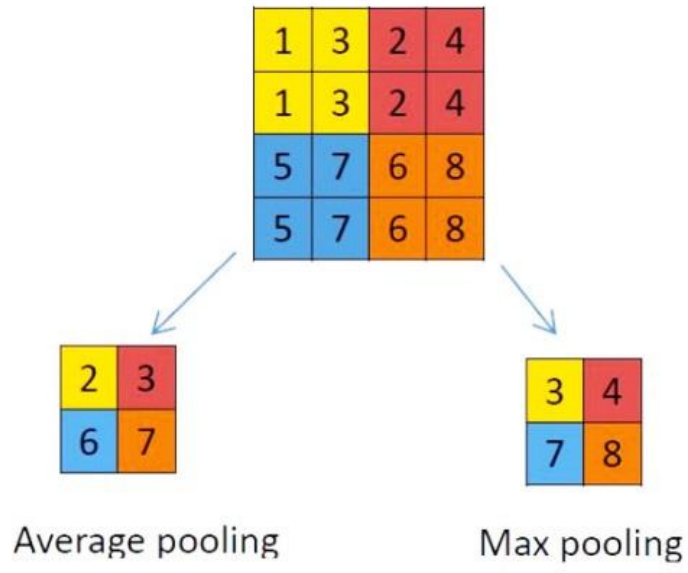


FIGURE 2.13 – (à gauche) Average pooling : chaque case correspond à la moyenne du carré d'entrée de la même couleur, ex de la case jaune : (1+ 3+ 1+ 3)/4 = 2. (à droite) Max pooling : chaque case correspond à la valeur maximum du carré d'entrée de la même couleur, ex de la case bleu : max(5, 7, 5, 7) = 7 [48]

3–Couches Entièrement Connectées

Après l'extraction des caractéristiques des entrées, on attache à la fin du réseau un perceptron ou bien un MLP (multi layer perceptron). Le perceptron prend comme entrée les caractéristiques extraites et produit un vecteur de N dimensions ou N est le nombre de classe ou chaque élément est la probabilité d'appartenance à une classe. Chaque probabilité est calculée à l'aide de la fonction softmax [48] dans le cas où les classes sont exclusivement mutuelles. Le terme «entièrement connecté » implique que chaque neurone dans la couche précédente est connecté à chaque neurone sur la couche suivante.

2.4.4 Les Fonctions d'Activation

La fonction d'activation est une fonction mathématique appliquée à un signal en sortie d'un neurone artificiel. Le terme de "fonction d'activation" vient de l'équivalent biologique "potentiel d'activation", seuil de stimulation qui, une fois atteint entraîne une réponse du neurone. La fonction d'activation est souvent une fonction non-linéaire. Leur but est de permettre aux réseaux de neurones d'apprendre des fonctions plus complexes qu'une simple régression linéaire car le fait de multiplier les poids d'une couche cachée est juste une transformation linéaire. Exemple de fonction d'activation Le ReLu (Rectified Linear Units) : Elle est utilisée après chaque opération de convolution, ou toutes les valeurs de pixels négatifs sont mises à zéro.

2.5 Optimisation pour l'Apprentissage en Deep Learning

La descente du gradient est un algorithme d'optimisation souvent utilisé pour trouver les poids ou les coefficients des algorithmes d'apprentissage automatique, tels que les réseaux de neurones artificiels et la régression logistique. Cela fonctionne en permettant au modèle de faire des prédictions sur les données d'apprentissage et en utilisant l'erreur sur les prédictions pour mettre à jour le modèle de manière à réduire l'erreur. Le but de l'algorithme est de trouver des paramètres de modèle (par exemple, des coefficients ou des poids) qui minimisent l'erreur du modèle sur le jeu de données d'apprentissage. Pour ce faire, il modifie le modèle en le déplaçant le long d'une pente d'erreur vers une valeur d'erreur minimale. Cela donne à l'algorithme le nom de « descente de gradient » [16].Il existe trois variantes de cette méthode :

- Batch gradient descent
- Descente de gradient stochastique
- Mini-batch gradient descent

2.6 Quelques Réseaux Convolutifs Célèbres

- LeNet [38] : Les premières applications réussies des réseaux convolutifs ont été développées par Yann LeCun dans les années 1990. Parmi ceux-ci, le plus connu est l'architecture LeNet utilisée pour lire les codes postaux, les chiffres, etc.
- AlexNet [36] : Le premier travail qui a popularisé les réseaux convolutifs dans la vision par ordinateur était AlexNet, développé par Alex Krizhevsky, Ilya Sutskever et Geoff Hinton. Ce CNN été soumis au défi de la base ImageNet en 2012 et a nettement surpassé ses concurrents. Le réseau avait une architecture très similaire à LeNet, mais était plus profond, plus grand et comportait des couches convolutives empilées les unes sur les autres (auparavant, il était commun de ne disposer que d'une seule couche convolutifs toujours immédiatement suivie d'une couche de pooling).
- ZFnet [72] : C'était une amélioration de AlexNet en ajustant les hyper-paramètres de l'architecture, en particulier en élargissant la taille des couches convolutifs et en réduisant la taille du noyau sur la première couche.
- GoogLeNet[67] : C'est un modèle de Google. Sa principale contribution a été le développement d'un module inception qui a considérablement réduit le nombre de paramètres dans le réseau (4M, par rapport à AlexNet avec 60M). En outre, ce module utilise le global Average pooling ce qui élimine une grande quantité de paramètres. Il existe également plusieurs versions de GoogLeNet, parmi elles, Inception-v4 [66] et Xception [7] ce dernier est l'un des modèles lesquels notre système s'inspire, plus de détails dans le chapitre de conception.
- ResNet [30] : Residual network développé par Kaiming He et al. Été le vainqueur de ILSVRC 2015 . Il présente des sauts de connexion et une forte utilisation de la batch normalisation. Il utilise aussi le global AVG pooling au lieu du PMC à la fin.
- VGG Net[62] : Il s'agit d'une structure du Visual Geometry Group d'Oxford réalisée par Andrea Vedaldi et Andrew Zisserman (en 2017).

2.7 Conclusion

Dans ce chapitre, nous avons vu qu'est-ce que les réseaux de neurones et leurs différentes types en suite on a essayé d'expliquer les réseaux de neurones convolutifs CNN et leur structures, et ses différentes couches. Le CNN à quatre principales opérations : convolution, la fonction non-linéarité (ReLU), Pooling et couche entièrement connectée. Première opération est la convolution pour l'extraction de caractéristiques de l'image d'entrée.

La deuxième opération est la fonction nonlinéarité (ReLU) pour remplacer toutes les valeurs de pixels négatives par zéro. Troisième opération est la Pooling pour réduire progressivement la taille de la carte de caractéristiques rectifiée. Enfin une couche entièrement connectée pour la classification. Et à la fin nous avons présenté quelques exemple d'architectures, parmi eux Xception et VGGnet.

CHAPITRE 3

CONCEPTION

3.1 Introduction

La reconnaissance des expressions faciales est un problème important, qui trouve des applications dans différents domaines. Plusieurs méthodes traditionnelles ont été utilisées dans la reconnaissance d'expression telle que les SVM [50] ,Adaboost [61],Forêts aléatoires[9], l'apprentissage profond (et principalement les CNN) permet de supprimer ou réduire fortement la dépendance des modèles physiques .

Dans ce chapitre, nous présentons notre conception en suivant les éléments suivant : Tout d'abord on commence par présenter comment le système est sensé fonctionner. Il s'agit du modèle conceptuel présentant les grandes fonctionnalités du système ainsi que le flux de données entre ces différentes fonctionnalités. Ensuite, nous passons à une description plus détaillée, nous décrivons en détaille la conception du modèle propose en donnant les détails de chaque module de la conception, enfin nous définissons par la suite les paramètres et les détails techniques relatifs à l'analyse des expressions ainsi que l'architecture utilisée.

3.2 Présentation du système REFCNN

Nous proposons une conception DE REFCNN un système de reconnaissance d'expression faciale à partir d'un visage en temps réel à base de réseaux CNNs. Ce système consiste à détecter un visage d'une personne à partir d'une image, séquence vidéo ou via une caméra pour connaître l'expression avec un taux de précision associé au six expression universel à savoir la joie, Le dégoût la peur, la colère, La tristesse, La surprise plus l'état neutre.

3.3 Architecture globale du système REFCNN

Toutes les solutions apportées à la reconnaissance d'émotions sont structurées selon la même architecture globale, en trois modules principaux fonctionnant indépendamment :

- La détection de visage,
- L'extraction de caractéristiques
- La classification.

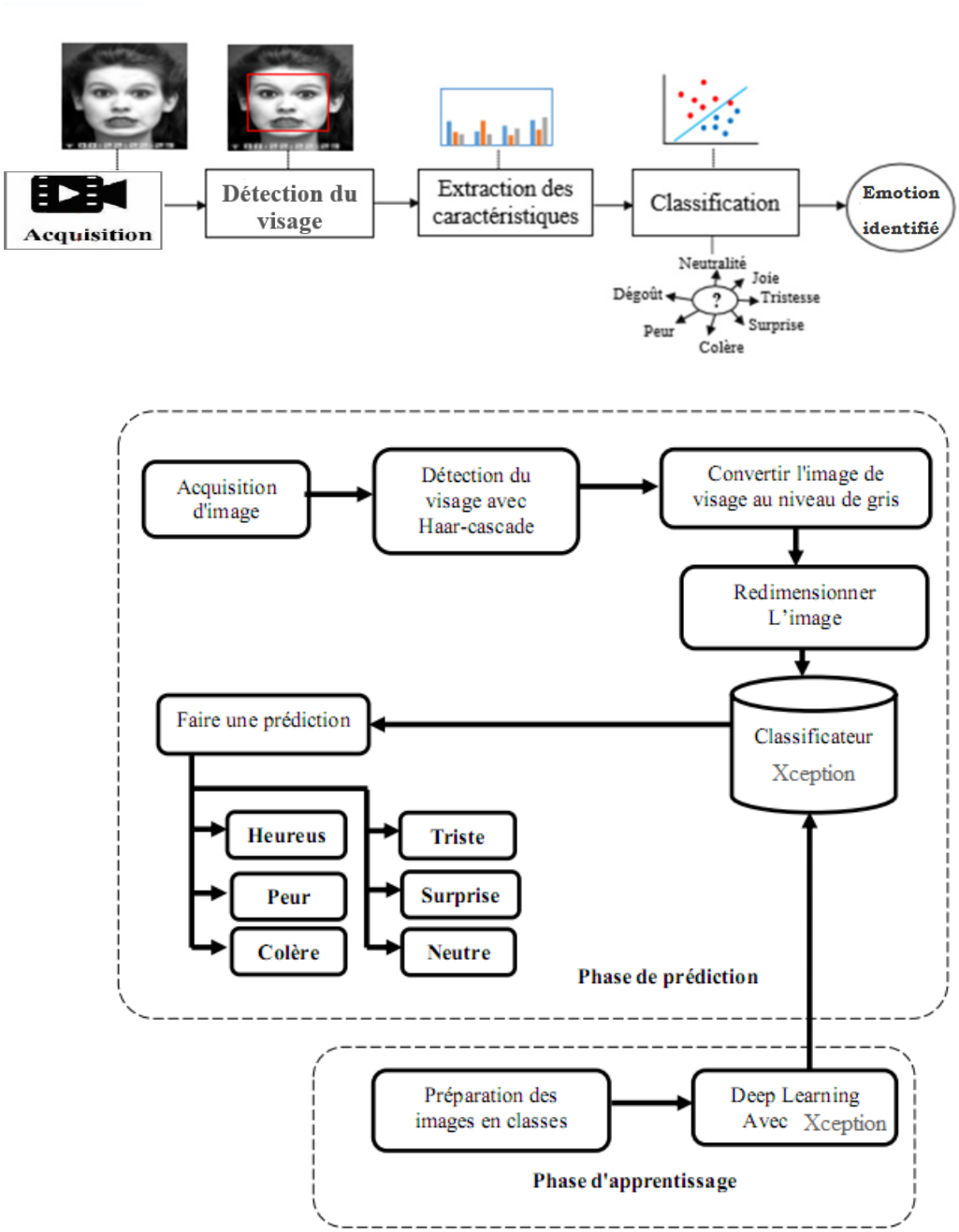


FIGURE 3.1 – Schéma globale du système

3.4 Conception détaillé de système

Dans ce qui suit, nous allons détailler chacune des étapes décrites ci-dessus.

3.4.1 Détection du visage

Le problème de détection des visages implique l'identification de la présence de visages dans une image et la détermination des emplacements et des échelles des visages.
L'efficacité des systèmes FER dépend essentiellement de la méthode utilisée pour localiser le visage dans l'image. Dans notre méthode, nous utilisons l'algorithme Viola-Jones [33] pour détecter diverses parties du visage humain telles que la bouche, les yeux, le nez, les narines, les sourcils, les lèvres et oreilles.

Principe de la méthode de Viola et Jones : La méthode de Viola et Jones consiste à balayer une image à l'aide d'une fenêtre de détection de taille initiale 24px par 24px (dans l'algorithme original) et de déterminer si un visage y est présente. Lorsque l'image a été parcourue entièrement, la taille de la fenêtre est augmentée et le balayage recommence, jusqu'à ce que la fenêtre fasse la taille de l'image. L'augmentation de la taille de la fenêtre se fait par un facteur multiplicatif de 1.25. Le balayage, quant à lui, consiste simplement à décaler la fenêtre d'un pixel. Ce décalage peut être changé afin d'accélérer le processus, mais un décalage d'un pixel assure une précision maximale. Cette méthode est une approche basée sur l'apparence, qui consiste à parcourir l'ensemble de l'image en calculant un certain nombre de caractéristiques dans des zones rectangulaires qui se chevauchent. Elle a la particularité d'utiliser des caractéristiques très simples, mais très nombreuses.

La bibliothèque OpenCV présente une implémentation de cet algorithme sous le nom « détecteur en cascades de Haar ». Le point fort de cette méthode est la rapidité de détection ce qui la rend capable de s'exécuter en temps réel et de répondre aux exigences du traitement vidéo. Toutefois, elle présente quelques limites telles que la difficulté de détection simultanée du plusieurs vues de même objet, la durée nécessaire à la phase d'apprentissage des cascades est relativement assez grande et le nombre d'échantillons d'apprentissage est important.
Étant donné que nous traitons des séquences vidéo, la phase de détection implique implicitement la phase de suivi du visage dans la scène, puisque nous traitons la séquence vidéo image par image. Cette étape se décomposée de trois tâches à savoir, la détection du visage à l'aide de la méthode, l'extraction des traits faciaux ou-bien les points caractéristiques et le suivi des déplacements du visage dans la scène.

FIGURE 3.2 – Processus de détection du visage.

3.4.2 Extraction de caractéristiques faciales.

Une fois le visage détecté dans l'image, le système lance le processus d'extraction des caractéristiques qui va convertir les données des pixels à des représentations et configuration plus réduite et optimal pour que la représentation extraite soit utilisé dans le processus de la classification , cette étape réduit les dimensions de l'image en entrée en gardant les données les plus utiles pour la classification.

L'étape d'extraction représente le cœur du système de reconnaissance, on extrait de l'image les informations qui seront sauvegardées en mémoire pour être utilisées plus tard dans la phase de décision (classification).

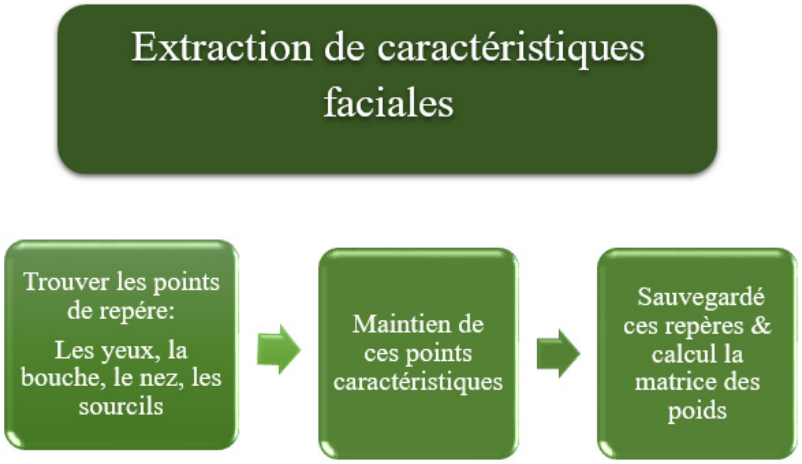


FIGURE 3.3 – Processus de l'étape d'extraction de caractéristiques faciales.

La détection de ces points est implémentée dans la librairie dlib [22] utilisé sous le langage Python. Elle permet de produit 68 point 2D de coordonnées (x, y) qui cartographient des structures faciales spécifiques. Ces points sont stockés dans un tableau indexé. Voici donc les indices de chaque point parmi les 68 points (figure 3.4).

FIGURE 3.4 – Le résultat de détection des points caractéristiques à partir du visage en utilisant dlib[22].

Ce processus est basé sur la technologie de réseau de neurones convolutionel (CNN), Nous avons utilisé l'architectures de CNN, Xception [7], dans ce qui suit nous représentant cette architecture et son adaptation à notre problème.

3.4.3 Présentation l'architecture Xception

Xception est une structure moderne proposé par François Chollet lui-même, créateur et responsable de la maintenance de la bibliothèque Keras de Google. Xception est une extension de l'architecture Inception [11] qui remplace les modules Inception standard par des convolutions séparables en profondeur ce qui permet de diminuer la taille de l'architecture jusqu'au 91Mo (voir la figure 3.5).

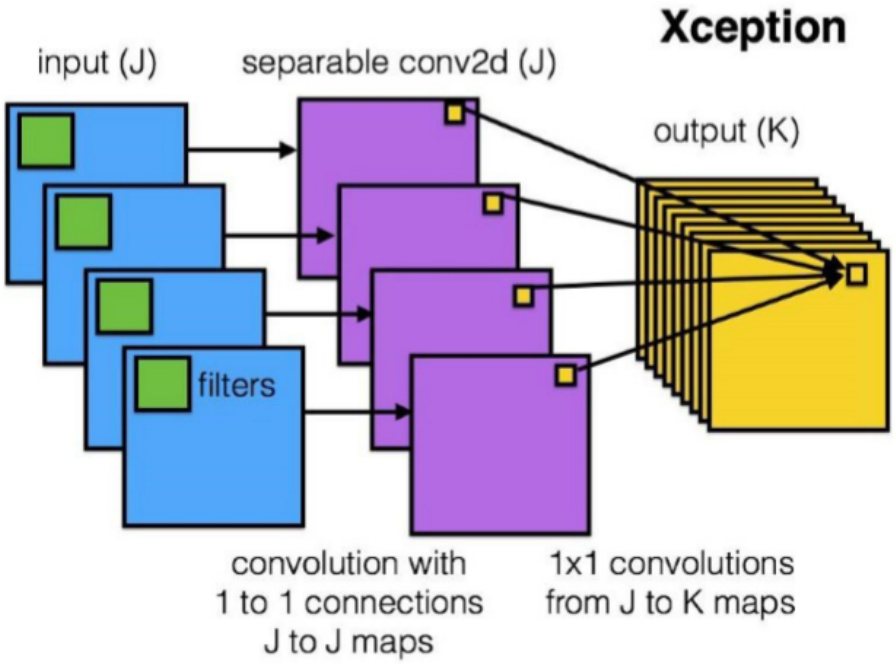


FIGURE 3.5 – Architecture de Xception [49] .

La spécification majeure de la structure Xception c'est le concept de Convolution spatiale en profondeur [11]. Mais qu'est-ce que c'est et en quoi est-ce différent d'une convolution normale ?

Définition de la Convolution séparable

C'est une solution au problème de la complexité des calculs au niveau de convolution qui rend le fonctionnement de réseau un peu long, avec moins de calculs, le réseau est en mesure de traiter davantage de données en un temps réduit. Il existe deux types de convolutions séparables [11] :

- La convolution séparable spatiale :
  C'est la solution la plus facile parce qu'elle consiste à diviser la couche de convolution en deux, elle traite principalement des dimensions spatiales d'une image et du noyau : la largeur et la hauteur. Malheureusement elle est très limité par conséquent elle n'est pas très utilisé dans l'apprentissage en profondeur
- La convolution séparable en profondeur :
  C'est la solution qui nous intéresse dans notre étude car elle traite non seulement des dimensions spatiales, mais également de la dimension en profondeur c'est-à-dire le nombre de canaux. Une image d'entrée peut avoir 3 canaux : RVB. Après quelques convolutions, une image peut avoir plusieurs canaux. Semblable à la convolution spatiale séparable, une convolution séparable en profondeur divise un noyau en 2 noyaux distincts qui effectuent

deux convolutions : La convolution en profondeur et la convolution ponctuelle (taille de 1x1).

La convolution séparable en profondeur dans Xception

La convolution modifiée dans la structure Xception est la convolution ponctuelle suivie d'une convolution en profondeur. Selon lequel, une convolution 1 × 1 est effectuée avant toutes les convolutions spatiales de taille n × n. Ainsi, c'est un peu différent de l'original. Le modèle Xception atteint une précision de 94,5% sur la base de données ImageNet [11] .

Quelques modèles ont réduit l'ensemble des paramètres dans leurs dernières couches en incluant une fonction Globale de pooling moyenne « Opération Global de pooling » comme dans Xception , ce qui réduit l'ensemble des cartes de caractéristiques en une valeur scalaire en prenant la moyenne de tous les éléments de la carte. Nous proposons d'éliminer les couches entièrement connectées dans tous les CNN de notre système et nous incluons la fonction Globale de pooling moyen de tel sorte, nous obtenons à la sortie des réseaux un nombre des cartes de caractéristiques égale au nombre de classe (07 ; les six expressions et l'état neutre). Ensuite, nous appliquons la fonction d'activation softmax, qui est le même principe du modèle Xception.

3.5 Conclusion

Dans ce chapitre, nous avons introduit le modèle Xception utilisé pour la classification des images, et nous avons remarqué que cette architecture a une caractéristique spécifique.

Dans le prochain chapitre, nous présentons l'utilisation de ce modèle pour le tester sur la reconnaissance d'expression faciale d'un visage réel, afin de révéler les performances d'un système FER par apprentissage profond.

CHAPITRE 4

IMPLÉMENTATION ET RÉSULTATS

4.1 Introduction

L'objectif de ce chapitre est de présenter les étapes de l'implémentation de l'approche proposée dans le cadre d'un système de reconnaissance des expressions faciales et les différentes étapes de réalisation. Nous nous sommes intéressés à l'utilisation de réseau neuronal convolutif.
Nous commençons tout d'abord par la présentation des ressources, du langage et de l'environnement de développement que nous avons utilisés. Puis les étapes de la réalisation du modèle.

Nous poursuivons ce chapitre par la présentation des différents résultats expérimentaux obtenus, quelques captures d'écrans de notre application et une petite discussion est donnée à la fin de ce chapitre.

4.2 Environnement de travail

Dans cette section, nous présenterons les environnements matériel et logiciel de notre travail.

4.2.1 Environnement matériel

Afin de mener à bien ce projet, il a été mis à notre disposition un ordinateur AZUZ avec les caractéristiques suivantes :

- Processeur : Intel ® Core ™ i3-5005U CPU @ 2.00 GHz 2.00 GHz
- RAM : 4,00 Go
- Système d'exploitation : Windows 10, 64 bits
- Carte graphique : Intel(R) HD Graphics 5500

4.2.2 Environnement de développement

En utilisant le langage généraliste python, la bibliothèque de vision par ordinateur Opencv, Tensarflow de google et le Framework Keras pour la mise en œuvre de nos réseaux de neurones, ainsi que d'autre modules et bibliothèques (numpy, marplotlib, etc...).

python

Python est un langage de programmation de haut niveau utilisé pour la programmation générale. Créé par Guido van Rossum et sorti en 1991, Python a une philosophie de conception qui met l'accent sur la lisibilité du code, notamment en utilisant des espaces importants.
Ce langage de programmation présente de nombreuses caractéristiques intéressantes :

- Il est gratuit et multiplateforme. : Windows, Mac OS X, Linux, Android, iOS, depuis les mini-ordinateurs Raspberry Pi jusqu'aux supercalculateurs.
- C'est un langage de haut niveau. Il demande relativement peu de connaissance sur le fonctionnement d'un ordinateur pour être utilisé.
- C'est un langage interprété. Un script Python n'a pas besoin d'être compilé pour être exécuté, contrairement à des langages comme le C ou le C++.
- C'est un langage dynamique ,extensible Il favorise la programmation structurée fonctionnelle et orientée objet. C'est-à-dire qu'il est possible de concevoir en Python des entités qui miment celles du monde réel (une cellule, une protéine, un atome, etc.) avec un certain nombre de règles de fonctionnement et d'interactions.
- Il est relativement simple à prendre en main[49].
- La syntaxe de Python est très simple et, combinée à des types de données évolués (listes, dictionnaires,...), conduit à des programmes à la fois très compacts et très lisibles.
- Il gère ses ressources (mémoire, descripteurs de fichiers...) sans intervention du programmeur [49].
- Enfin, il est très utilisé en bio-informatique et l'intelligence artificielle et plus généralement en analyse de données. Toutes ces caractéristiques font que Python est un outils idéal pour implémenter notre application

Bibliothèques utilisées

1. **OpenCV (Open Source Computer Vision Library)**

   Est une bibliothèque proposant un ensemble de plus de 2500 algorithmes de vision par ordinateur spécialisé dans le traitement d'images, accessible au travers d'API pour les langages C, C++, et Python. Elle est distribuée sous une licence BSD (libre) pour les plateformes Windows, GNU/Linux, Android et MacOS [20]. OpenCv est spécialisé en :

   TensorFlow est un framework de programmation pour le calcul numérique de Google initié

et développé par l'équipe Google Brain spécialisé dans l'intelligence artificielle, et rendu Open Source en Novembre 2015, et devenir très rapidement l'un des frameworks les plus utilisés pour le Deep Learning, Les caractérestiques principale de tensorflow sont :

- Manipulation des données d'images et vidéo, les matrices et les vecteurs.
- Différentes structures de données dynamiques (listes, files d'attente, ensembles, arbres, graphiques).
- Analyse du mouvement (flux optique, segmentation du mouvement, suivi), Reconnaissance d'objets.
- Interface graphique de base (image / vidéo à afficher, gestion du clavier et de la souris, barres de défilement. . .)

2. **(Numpy)**

   Est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion facilitée des tableaux de nombres, des fonctions sophistiquées (diffusion), on peut aussi l'intégrer le code C / C ++ et Fortran.

3. **(Matplotlib)**

   Est une bibliothèque de traçage pour le langage de programmation Python et son extension mathématique numérique NumPy . Il fournit une API orientée objet permettant d'incorporer des graphiques dans des applications à l'aide de kits d'outils d'interface graphique à usage général tels que Tkinter , wxPython , Qt ou GTK + .

4. **(Tensorflow)**

   TensorFlow est un framework de programmation pour le calcul numérique de Google initié et développé par l'équipe Google Brain spécialisé dans l'intelligence artificielle, et rendu Open Source en Novembre 2015, et devenir très rapidement l'un des frameworks les plus utilisés pour le Deep Learning, Les caractérestiques principale de tensorflow sont :

   - Multi-plateformes (Windows , Linux, Mac OS, et même Android et iOS)
   - APIs en Python, C++, Java et Go (l'API Python est plus complète cependant, c'est sur celle-ci que nous allons travailler)
   - Temps de compilation très courts dû au backend en C/C++
   - Supporte les calculs sur CPU, GPU et même le calcul distribué sur cluster
   - Une documentation extrêmement bien fournie avec de nombreux exemples et tutoriels
   - Last but not least : Le fait que le framework vienne de Google et que ce dernier ait annoncé avoir migré la quasi-totalité de ses projets liés au Deep Learning en TensorFlow est quelque peu rassurant .Cependant dans FACECNN nous utilisant Cette plateforme via la bibliothèque Keras ce qui est sur-couche à TensorFlow donc un niveau plus haut que Tensorflow

5. **(Keras)**

Keras est une bibliothèque open source écrite en python et permettant d'interagir avec les algorithmes de réseaux de neurones profonds et de machine Learning, notamment Tensorflow et Theano. Elle a été initialement écrite par François Chollet . Keras nous permet de créer de nouvelles couches, des fonctions et développer des modèles à la pointe de la technologie avec peu de restrictions en quelques lignes de code.

4.3 La Base De Données BDD

Pour augmenter la précision et améliorer la performance des modèles , il est préférable d'entraînés le réseau avec beaucoup d'échantillons d'image.
La base de données d'images des expressions faciales que nous avons utilisé est celle de la base de données de Fer2013, elle comprend un total de 35887 images en niveaux de gris, pré-recadrées de 48x48 pixels de visages chacun étiquetés avec l'une des 7 classes d'émotion suivantes :
0 : 4593 $images \rightarrow col\grave{e}re$
1 : 547 $images \rightarrow d\acute{e}go\hat{u}t$
2 : 5121 $images \rightarrow peur$
3 : 8989 $images \rightarrow heureuse$
4 : 6077 $images \rightarrow triste$
5 : 4002 $images \rightarrow surprise$
6 : 6198 $images \rightarrow neutre$

Qui sont reparties en deux parties à savoir :
– 3589 images représentant les images de test
– 28709 représentent les images d'apprentissage.

Toutes ces images ont été déjà résolues par kaggle et contiennent chacune une taille de (48x48 pixels).

FIGURE 4.1 – Échantillon d'images BDD de Fer2013.

4.4 Implémentation et Réalisation

Cette section décrit les différents modules du notre système de reconnaissance.

Module de détection

Pour détecter des visages nous avons utilisé une méthode populaire, proposée par Paul Viola et Michael Jones dans leur article "Détection rapide d'objets utilisant une cascade de fonctions simples" en 2001 [49]. Dans notre système nous avons utilisé haarcascade-eye.xml de la bibliothèque OpenCV qui fournit la méthode Haar Cascade.

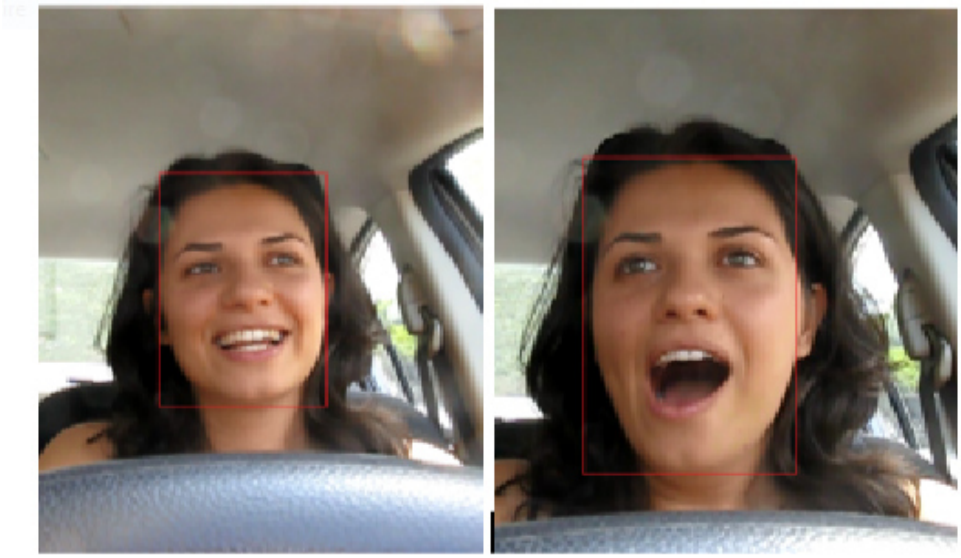
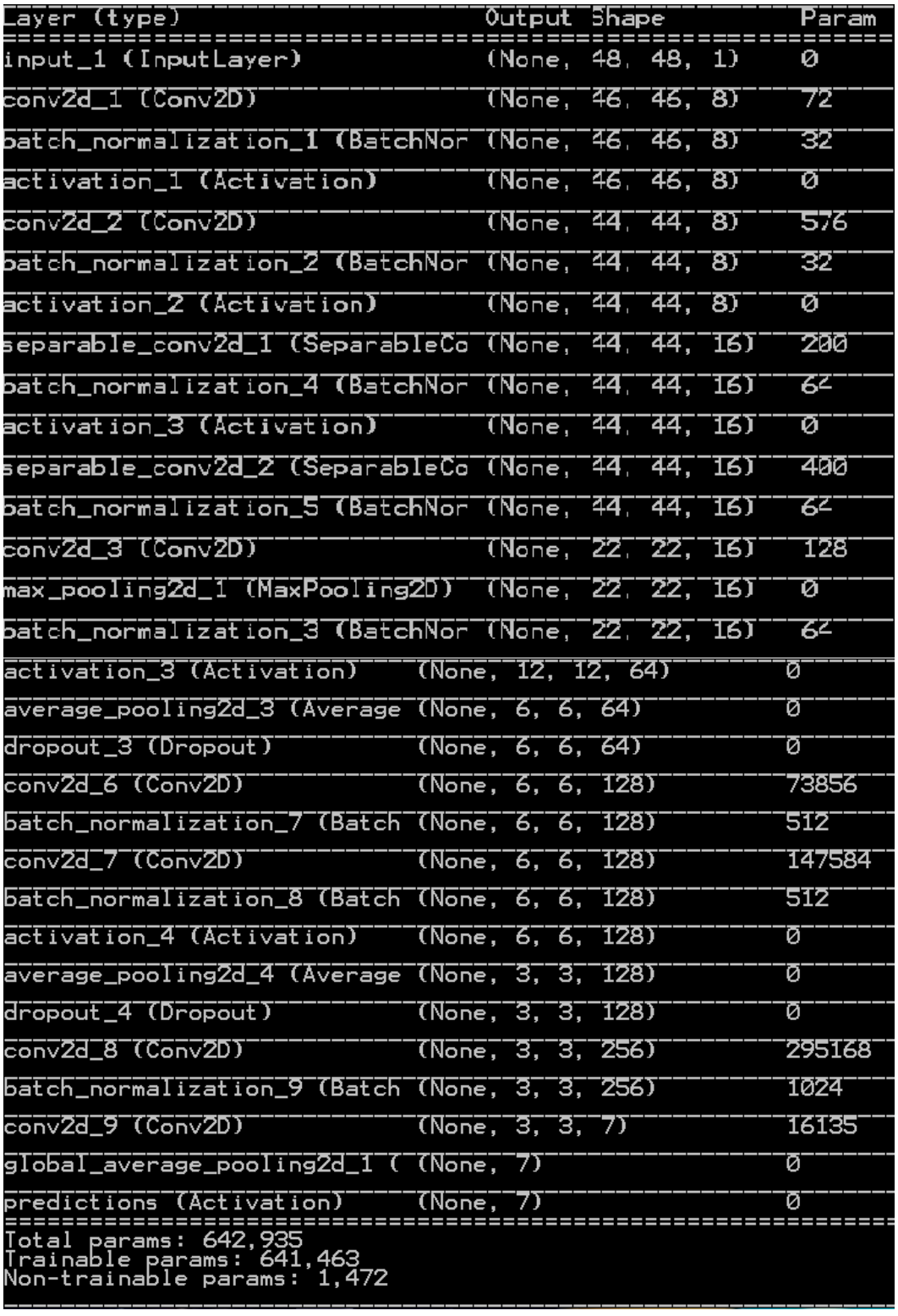


FIGURE 4.2 – Détection de visage et dessin de rectangle englobant dans chaque frame.

4.4.1 Module de reconnaissance

Après la détection du visage, notre système extrait les caractéristiques depuis les données acquises en utilisant les réseaux de neurones convolutionel. Nous avons choisi l'architecture Xception De notre système REFCNN . Il est constitué de deux couches de convolutions normale en premier lieu, ensuite il y a deux couches de convolutions, chacune d'elle est suivie par une convolution séparable en profondeur, Normalisation par lot (BatchNormalization) et Activation ReLu et une autre convolution séparable suivis par la normalisation et fonction Maxpooling.
Et à la fin de notre réseau, on a ajouté une couche de convolution suivie par un pooling moyen et la fonction Softmax qui nous renvoi la classification. La figures 4.3 illustre les différentes couches de cette architecture.



Layer (type)	Output Shape	Param
input_1 (InputLayer)	(None, 48, 48, 1)	0
conv2d_1 (Conv2D)	(None, 46, 46, 8)	72
batch_normalization_1 (BatchNor	(None, 46, 46, 8)	32
activation_1 (Activation)	(None, 46, 46, 8)	0
conv2d_2 (Conv2D)	(None, 44, 44, 8)	576
batch_normalization_2 (BatchNor	(None, 44, 44, 8)	32
activation_2 (Activation)	(None, 44, 44, 8)	0
separable_conv2d_1 (SeparableCo	(None, 44, 44, 16)	200
batch_normalization_4 (BatchNor	(None, 44, 44, 16)	64
activation_3 (Activation)	(None, 44, 44, 16)	0
separable_conv2d_2 (SeparableCo	(None, 44, 44, 16)	400
batch_normalization_5 (BatchNor	(None, 44, 44, 16)	64
conv2d_3 (Conv2D)	(None, 22, 22, 16)	128
max_pooling2d_1 (MaxPooling2D)	(None, 22, 22, 16)	0
batch_normalization_3 (BatchNor	(None, 22, 22, 16)	64
activation_3 (Activation)	(None, 12, 12, 64)	0
average_pooling2d_3 (Average	(None, 6, 6, 64)	0
dropout_3 (Dropout)	(None, 6, 6, 64)	0
conv2d_6 (Conv2D)	(None, 6, 6, 128)	73856
batch_normalization_7 (Batch	(None, 6, 6, 128)	512
conv2d_7 (Conv2D)	(None, 6, 6, 128)	147584
batch_normalization_8 (Batch	(None, 6, 6, 128)	512
activation_4 (Activation)	(None, 6, 6, 128)	0
average_pooling2d_4 (Average	(None, 3, 3, 128)	0
dropout_4 (Dropout)	(None, 3, 3, 128)	0
conv2d_8 (Conv2D)	(None, 3, 3, 256)	295168
batch_normalization_9 (Batch	(None, 3, 3, 256)	1024
conv2d_9 (Conv2D)	(None, 3, 3, 7)	16135
global_average_pooling2d_1 (	(None, 7)	0
predictions (Activation)	(None, 7)	0

Total params: 642,935
Trainable params: 641,463
Non-trainable params: 1,472

FIGURE 4.3 – CNN du système REFCNN.

Module qui charge la base des données Les images dans la base de données fer2013 sont traitées de manière que les visages soient presque centrés et chaque visage occupe environ la même quantité d'espace dans chaque image.

Chaque image est donnée sous forme de chaîne images de taille (48 × 48) stockée en tant que vecteur de ligne sous le format (.csv). Nous commençons le processus du chargement des données en appelant la classe load-and-process.py qui charge la base de données et faire un traitement des images et les préparé pour les servir au modèle CNN choisi par la suite. Deux fonctions qui composent cette classe :

La fonction *(def load-fer2013)* Le rôle de cette fonction est de lire le fichier de la base fer2013 de type .csv, convertit la séquence de pixels de chaque ligne en image de dimension (48 * 48) retourne des image contiens des visages et des étiquettes d'émotion Return(faces, emotion).

```
def load_fer2013():
        data = pd.read_csv(dataset_path)
        pixels = data['pixels'].tolist()
        width, height = 48, 48
        faces = []
        for pixel_sequence in pixels:
            face = [int(pixel) for pixel in
pixel_sequence.split(' ')]
            face = np.asarray(face).reshape(width, height)
            face = cv2.resize(face.astype('uint8'),image_size)
            faces.append(face.astype('float32'))
        faces = np.asarray(faces)
        faces = np.expand_dims(faces, -1)
        emotions = pd.get_dummies(data['emotion']).as_matrix()
        return faces, emotions
```

FIGURE 4.4 – La fonction def load-fer2013()

La fonction *def preprocess-input* Cette fonction est la meilleur méthode pour le le modèle de réseaux de neurones dans les problèmes de vision par ordinateur, c'est un procédure standard pour l'apprentissage,

```
def preprocess_input(x, v2=True):
    x = x.astype('float32')
    x = x / 255.0
    if v2:
        x = x - 0.5
        x = x * 2.0
    return x
```

FIGURE 4.5 – La fonction def preprocess-input()

Pour les redimensionnant entre -1 et 1, les images sont redimensionnées à [0,1] en le divisant par 255. De plus, la soustraction par 0,5 et la multiplication par 2 modifient le va jusqu'à [-1,1].

L'apprentissage de classificateur d'images de réseau neuronal convolutionnel avec Keras

Après la préparation des données, on a procéder à l'apprentissage en utilisant l'architecture CNN simple de xception, et on a utilisé une classe avec le nom cnn.py qui implémente l'architecture xcetion du CNN. Nous commençons à former le classifieur d'images en utilisant l'apprentissage profond, avec Keras et Tensorflow.
Nous créons un modèle Séquentiel en transmettant une liste d'instances de couche au constructeur. Nous effectuons ensuite un partage d'apprentissage et tests sur les données en utilisant 75% des images pour l'apprentissage et 25% pour les tests.

Formation du modèle avec keras

1. Importer les bibliothèques installées
2. Importer le type de modèle séquentiel de Keras qui il s'agit d'un empilement linéaire de couches de réseaux de neurones à travers la commande : from keras. models import Sequential
3. Importer les couches "principales" de Keras qui sont utilisées dans presque tous les réseaux de neurones : From keras. Layers import Dense, Dropout, Activation, Flatten
4. Importer les couches CNN de Keras. Ce sont les couches convolutives qui nous aideront à former notre modèle : from keras. layers import Convolution2D, MaxPooling2D.

```
4
5   from keras.callbacks import CSVLogger, ModelCheckpoint, EarlyStopping
6   from keras.callbacks import ReduceLROnPlateau
7   from keras.preprocessing.image import ImageDataGenerator
8   from load_and_process import load_fer2013
9   from load_and_process import preprocess_input
10  from models.cnn import mini_XCEPTION
11  from sklearn.model_selection import train_test_split
12
13  # parameters
14  batch_size = 32
15  num_epochs = 10000
16  input_shape = (48, 48, 1)
17  validation_split = .2
18  verbose = 1
19  num_classes = 7
20  patience = 50
21  base_path = 'models/'
22
23  # data generator
24  data_generator = ImageDataGenerator(
25                          featurewise_center=False,
26                          featurewise_std_normalization=False,
27                          rotation_range=10,
28                          width_shift_range=0.1,
29                          height_shift_range=0.1,
30                          zoom_range=.1,
31                          horizontal_flip=True)
32
33  # model parameters/compilation
```

```
32
33  # model parameters/compilation
34  model = mini_XCEPTION(input_shape, num_classes)
35  model.compile(optimizer='adam', loss='categorical_crossentropy',
36                metrics=['accuracy'])
37  model.summary()
38
39
40
41
42
43      # callbacks
44  log_file_path = base_path + '_emotion_training.log'
45  csv_logger = CSVLogger(log_file_path, append=False)
46  early_stop = EarlyStopping('val_loss', patience=patience)
47  reduce_lr = ReduceLROnPlateau('val_loss', factor=0.1,
48                                patience=int(patience/4), verbose=1)
49  trained_models_path = base_path + '_mini_XCEPTION'
50  model_names = trained_models_path + '.{epoch:02d}-{val_acc:.2f}.hdf5'
51  model_checkpoint = ModelCheckpoint(model_names, 'val_loss', verbose=1,
52                                                    save_best_only=True)
53  callbacks = [model_checkpoint, csv_logger, early_stop, reduce_lr]
54
55  # loading dataset
56  faces, emotions = load_fer2013()
57  faces = preprocess_input(faces)
58  num_samples, num_classes = emotions.shape
59  xtrain, xtest,ytrain,ytest = train_test_split(faces, emotions,test_size=0.2,
60                                                shuffle=True)
61  model.fit_generator(data_generator.flow(xtrain, ytrain,
62                                          batch_size),
63                      steps_per_epoch=len(xtrain) / batch_size,
64                      epochs=num_epochs, verbose=1, callbacks=callbacks,
65                      validation_data=(xtest,ytest))
66
```

FIGURE 4.6 – Le code source de l'architecture Xception

La fonction principale qui est responsable de l'apprentissage c'est la fonction «fit ()» avec les paramètres suivants : Données d'entraînement (Xtrain), données cibles (ytrain), données de validation et nombre d'époques.
Pour nos données de validation, nous utilisons l'ensemble de test fourni dans notre ensemble de données, que nous avons divisé en Xtest et Ytest. TrainX et testX constituent les données d'image elles-mêmes, tandis que trainY et testY constituent les étiquettes.

```
model.fit_generator(data_generator.flow(xtrain, ytrain,
                                        batch_size),
                    steps_per_epoch=len(xtrain) / batch_size,
                    epochs=num_epochs, verbose=1, callbacks=callbacks,
                    validation_data=(xtest,ytest))
```

FIGURE 4.7 – La fonction d'apprentissage

Résultat de l'apprentissage

Sachant que nous n'avons pas utilisé une unité de traitement graphique, le temps de l'apprentissage a été si longtemps et il peut prendre des jours pour terminer le processus de l'apprentissage 4.8.

```
Anaconda Prompt (anaconda3) - python train_emotion_classifier.py
Non-trainable params: 1,472
_________________________________________________________________
C:\Users\Admin\Desktop\kiki\load_and_process.py:21: FutureWarning: Method .as_matrix will be removed in a future
 version. Use .values instead.
  emotions = pd.get_dummies(data['emotion']).as_matrix()
Epoch 1/10000
898/897 [==============================] - 1216s 1s/step - loss: 1.7888 - acc: 0.3188 - val_loss: 1.6610 - val_a
cc: 0.3788

Epoch 00001: val_loss improved from inf to 1.66097, saving model to models/_mini_XCEPTION.01-0.38.hdf5
Epoch 2/10000
898/897 [==============================] - 1219s 1s/step - loss: 1.5298 - acc: 0.4233 - val_loss: 1.5510 - val_a
cc: 0.4434

Epoch 00002: val_loss improved from 1.66097 to 1.55096, saving model to models/_mini_XCEPTION.02-0.44.hdf5
Epoch 3/10000
898/897 [==============================] - 1210s 1s/step - loss: 1.4179 - acc: 0.4684 - val_loss: 1.5634 - val_a
cc: 0.4356

Epoch 00003: val_loss did not improve from 1.55096
Epoch 4/10000
898/897 [==============================] - 1212s 1s/step - loss: 1.3432 - acc: 0.4940 - val_loss: 1.3441 - val_a
cc: 0.4905

Epoch 00004: val_loss improved from 1.55096 to 1.34413, saving model to models/_mini_XCEPTION.04-0.49.hdf5
Epoch 5/10000
898/897 [==============================] - 1167s 1s/step - loss: 1.2945 - acc: 0.5110 - val_loss: 1.4013 - val_a
cc: 0.4805

Epoch 00005: val_loss did not improve from 1.34413
Epoch 6/10000
898/897 [==============================] - 27507s 31s/step - loss: 1.2544 - acc: 0.5294 - val_loss: 1.2589 - val
_acc: 0.5373

Epoch 00006: val_loss improved from 1.34413 to 1.25894, saving model to models/_mini_XCEPTION.06-0.54.hdf5
Epoch 7/10000
898/897 [==============================] - 1292s 1s/step - loss: 1.2234 - acc: 0.5391 - val_loss: 1.3189 - val_a
cc: 0.5167

Epoch 00007: val_loss did not improve from 1.25894
Epoch 8/10000
898/897 [==============================] - 1232s 1s/step - loss: 1.1995 - acc: 0.5513 - val_loss: 1.1980 - val_a
cc: 0.5575
```

FIGURE 4.8 – Le processus de l'apprentissage

Une fois que le modèle est formé, nous allons essayer de lui attribuée certains paramètres :
Epoch : désigne Le nombre d'époques(le nombre de fois où le modèle parcourt les données) .
Loss : désigne le taux d'erreur.
Accuracy : désigne le taux de précision.
val_loss : désigne valeur perdu.

val_acc : désigne la valeur de précision.

On peut voir que le réseau s'est entraîné pendant 10 époques et nous avons atteint une grande précision (86,69%) et une faible perte qui suit la perte d'entraînement, comme le montre le tableau ci-après.

Epoch	perte d'apprentissage	précision d'apprentissage	valeur perdu	valeur de précision
1	1.7888	0.3188	1.6610	0.3788
2	1.5298	0.4233	1.5510	0.4434
3	1.4179	0.4684	1.5634	0.4356
4	1.3432	0.4940	1.3441	0.4905
5	1.2945	0.5110	1.4013	0.4805
6	1.2544	0.5294	1.2589	0.5373
7	1.2234	0.5391	1.3189	0.5167
8	1.1995	0.5513	1.1980	0.5575
9	1.1845	0.5544	1.2149	0.5534
10	1.1599	0.5629	1.2362	0.5361

TABLE 4.1 – Résultats de l'apprentissage

Après l'analyse des résultats obtenus, On constate les remarques suivantes :
La précision de l'apprentissage et de test augmente avec le nombre d'époque, ceci reflète qu'à chaque époque le modèle apprenne plus d'informations. Si la précision est diminuée alors on aura besoin de plus d'information pour faire apprendre notre modèle et par conséquent on doit augmenter le nombre d'époque et vice versa.
De même, l'erreur d'apprentissage et de la validation diminue avec le nombre d'époque.

4.4.2 Présentation de l'application

C'est l'interface de notre application où nous testant notre modèle , le système présent les visages acquit via une webcam détectés et l'état émotionnels de l'individu avec une fenêtre qui représente la probabilité du résultats obtenus.

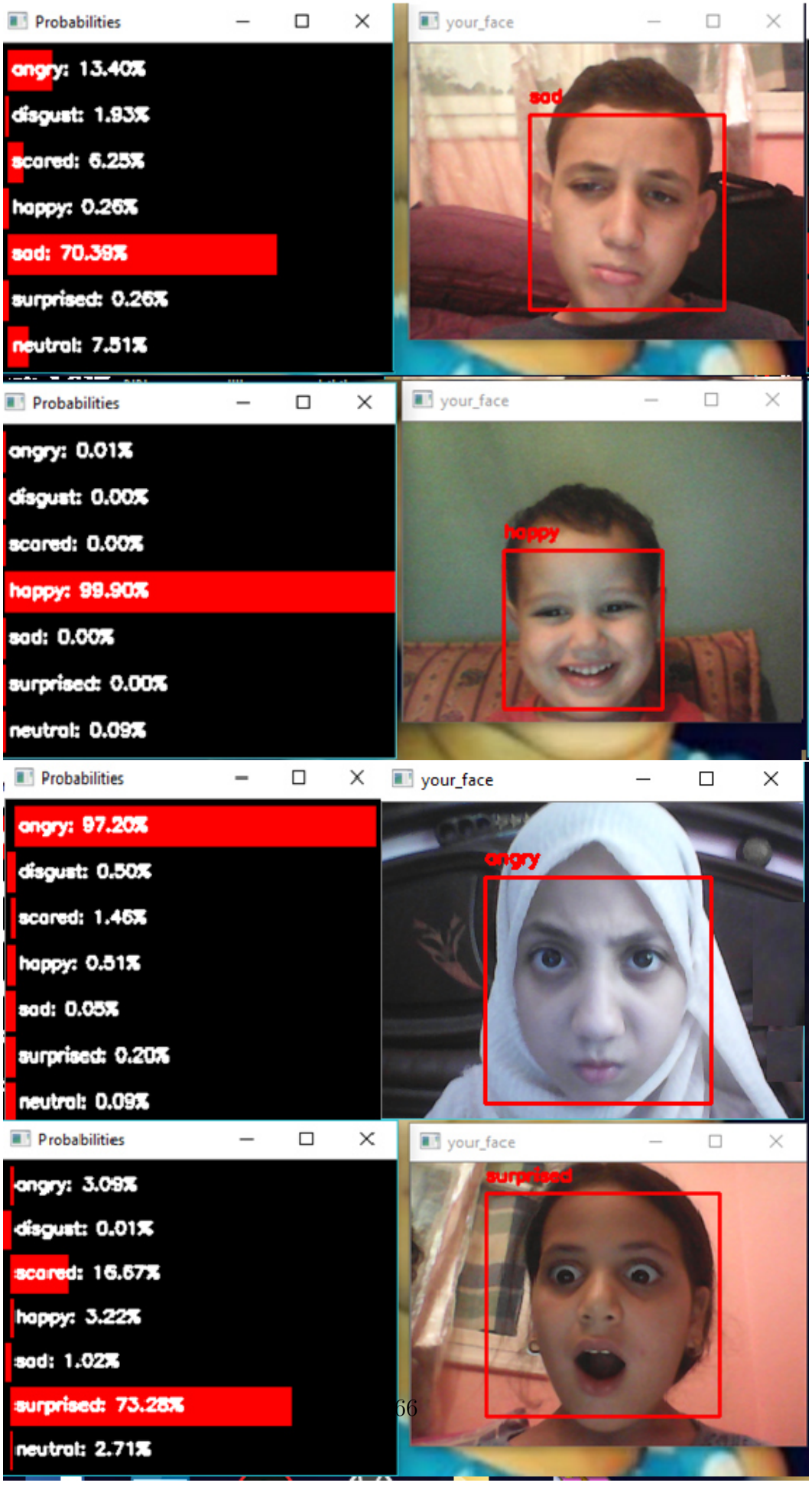


FIGURE 4.9 – Résultats

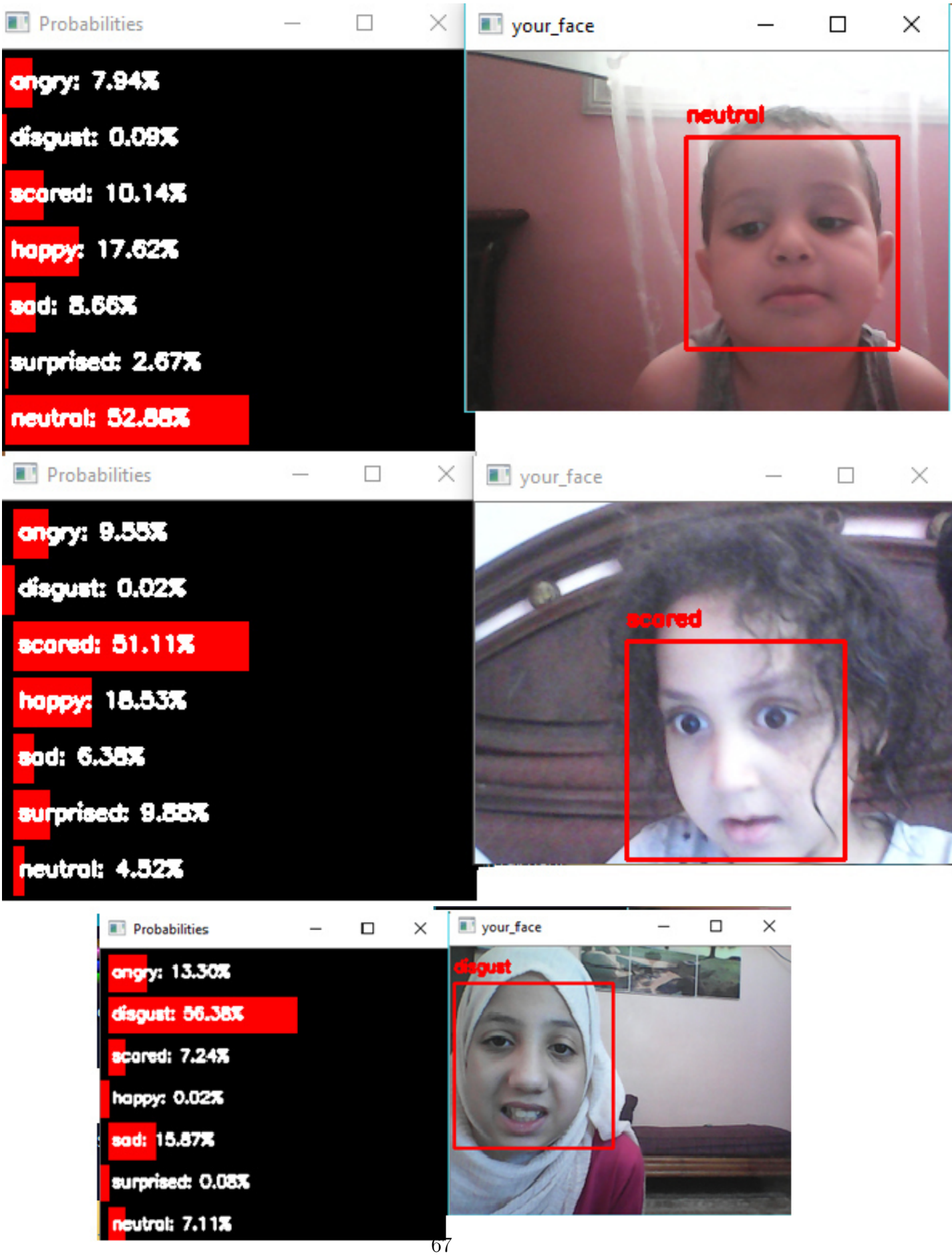
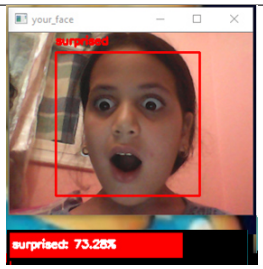
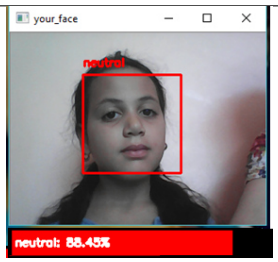
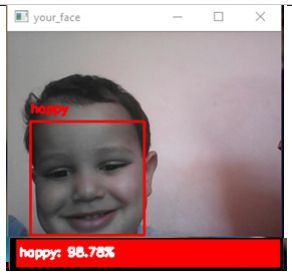
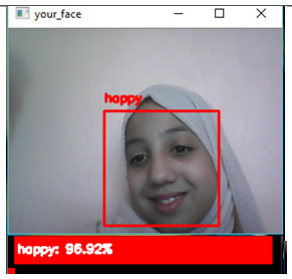
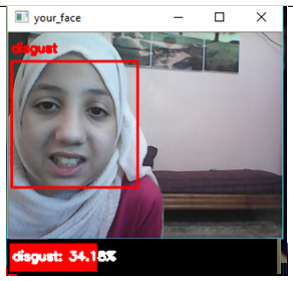


FIGURE 4.10 – Résultats.

4.4.3 Évaluation de notre classificateur d'images

Cette étape consiste à évaluer notre classifieur, s'il fonctionne bien ou non. On va analyser les résultats obtenus après les tests.
Le tableau ci-dessous, montre quelques tests et leurs résultats en utilisant notre système.

Tests	Résultats	Observations
Surpris		Positif
Neutre		Positif
Heureux		Positif
Heureux		Positif
Dégoût		Positif

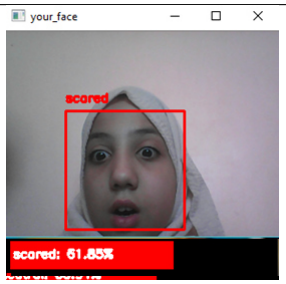
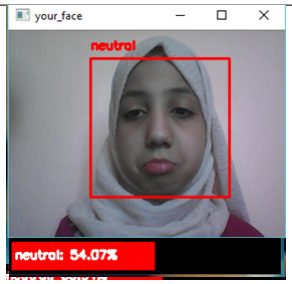
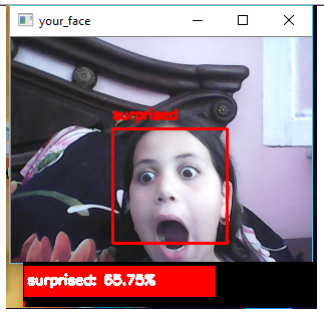
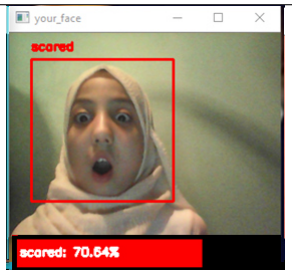
Surpris		Négatif
Triste		Négatif
Triste		Positif
Peur		Négatif
Peur		Positif

TABLE 4.2 – Tests et résultats avec webcam

Le tableau 4.2 montre les résultats obtenu du système REFCNN, sur l'ensemble de test en direct sur la webcam, le tableau donne le nombre de prédictions d'émotion et quelques indications.

Le modèle fonctionne très bien sur la classification des émotions dont les 10 essais qu'on a fait 07 positives et 03 négatifs, ce qui se traduit par des scores de précision relativement élevés pour l'heureuse, la neutre, la peur la colère et la surprise puisque l'utilisateur ne peut pas rester devant le webcam une durée très longue. Mais après plusieurs tests,on peut voir que notre classificateur fonctionne très bien sur la classification des expressions : heureux, peur, neutre, et surpris avec des bonnes précisions entre 98% et 70%.

Mais par contre, le classificateur semble faible pour les expressions : colère,dégoût et triste, avec des faibles précisions 34% et 53%.

Mais pour l'expression de surpris, la plupart des prédictions ont été mal prédit comme peur, à cause de la ressemblance de ces expressions faciales. La même chose pour l'expression de dégoût, les prédictions ont été mal prédit à cause de la ressemblance avec l'expression du tristesse.

4.5 Conclusion

Dans ce dernier chapitre on a projeté la lumière sur l'environnement de travail, coté matériel et coté soft pour réaliser notre système.

On a créé un classificateur d'expressions faciales à l'aide de Deep learning avec une architecture CNN simple "Xception",les expérimentations ont montré que le modèle de Xception est efficace en cas général,mais il souffre de lacunes pour certaines expressions.

CONCLUSION ET PERSPECTIVES

L'approche globale de l'analyse automatique des expressions faciales comprend généralement trois étapes. Étant donné une image d'entrée ou une séquence d'images, la première étape consiste à localiser le visage, détecter un ensemble de points faciaux. Une fois le visage détecté, l'étape suivante concerne l'extraction des caractéristiques du visage. L'étape finale prend comme entrée le vecteur de caractéristiques extrait précédemment pour effectuer la tâche de classification en utilisant une technique d'apprentissage automatique.

A travers ce mémoire, nous avons fait une généralité sur la reconnaissance des expressions faciales et l'utilisation d'un réseau de neurones convolutif qui exploite les convolutions séparables en profondeur. En parcourant les différents chapitres, nous avons décrit et clarifié la définition de la reconnaissance des expressions faciales, l'architecture du système, les objectifs atteints,l'exposition des résultats obtenus et leur discussion lors de différents tests réalisés. Notre système utilise un classifieur d'expressions faciales créé à l'aide de Deep learning avec une architecture CNN simple "Xception". On a choisi cette architecture parce qu'elle est plus simple et n'a pas besoin d'un matériel très puissant pour se servir.

Perspectives

Nos expériences démontrent une grande fiabilité du modèle Xception, en appliquant l'apprentissage sur l'ensemble de données FER-2013. Comme perspectives, nous souhaitons :

- Tester notre modèle sur d'autres bases des données plus volumineuse tel que CK+.

- Appliquer notre modèle sur des images 3D acquises par des caméras de profondeur.

- Inclure plus de classes, donc étendre le FER pour reconnaitre les micro-expressions, ce qui permettra plus de précision pour indiquer l'état émotionnel.

- Utiliser l'architecture Xception avec d'autres architectures en appliquant une hybridation comme l'architecture VGG ou LeNet.

En conclusion, ce projet nous a permis d'acquérir de nouvelles connaissances. Nous avons pu découvrir au cours de ce travail, de nouvelles notions tel que la notion de l'émotion et la manipulation de l'expression faciale et de faire connaissance des difficultés de confondre entre l'émotion et l'expression faciale et particulièrement la découverte de l'interaction Homme-Machine durant la reconnaissance des expressions faciales.

BIBLIOGRAPHIE

[1] Faiza Abdat. *Reconnaissance automatique des émotions par données multimodales : expressions faciales et des signaux physiologiques*. PhD thesis, Metz, 2010.

[2] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1) :147–169, 1985.

[3] Jörgen Ahlberg. Candide-3-an updated parameterised face. 2001.

[4] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *European conference on computer vision*, pages 469–481. Springer, 2004.

[5] Igor Aizenberg, Naum N Aizenberg, and Joos PL Vandewalle. *Multi-valued and universal binary neurons : theory, learning and applications*. Springer Science & Business Media, 2013.

[6] Stefano Arca, Paola Campadelli, and Raffaella Lanzarotti. A face recognition system based on automatically determined facial fiducial points. *Pattern recognition*, 39(3) :432–443, 2006.

[7] Octavio Arriaga, Matias Valdenegro-Toro, and Paul Plöger. Real-time convolutional neural networks for emotion and gender classification. *arXiv preprint arXiv :1710.07557*, 2017.

[8] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning : A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8) :1798–1828, 2013.

[9] Simon Bernard. *Forêts Aléatoires : De l'Analyse des Mécanismes de Fonctionnement à la Construction Dynamique*. PhD thesis, 2009.

[10] Glen D Brown, Satoshi Yamada, and Terrence J Sejnowski. Independent component analysis at the neural cocktail party. *Trends in neurosciences*, 24(1) :54–63, 2001.

[11] François Chollet. Xception : Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[12] Ira Cohen, Nicu Sebe, Ashutosh Garg, Lawrence S Chen, and Thomas S Huang. Facial expression recognition from video sequences : temporal and static modeling. *Computer Vision and image understanding*, 91(1-2) :160–187, 2003.

[13] Jean-Marc Colletta and Anna Tcherkassof. Les émotions : cognition, langage et développement, sprimont. *Belgique) : Mardaga*, 2003.

[14] Ronan Collobert and Jason Weston. A unified architecture for natural language processing : Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

[15] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6) :681–685, 2001.

[16] Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pages 2933–2941, 2014.

[17] Franck Davoine, Bouchra Abboud, and Mo Dang. Analyse de visages et d'expressions faciales par modèle actif d'apparence. 2004.

[18] Fernando De la Torre and Jeffrey F Cohn. Facial expression analysis. In *Visual analysis of humans*, pages 377–409. Springer, 2011.

[19] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015.

[20] Nene Adama Dian DIALLO. La reconnaissance des expressions faciales. 2019.

[21] Hui Ding, Shaohua Kevin Zhou, and Rama Chellappa. Facenet2expnet : Regularizing a deep face recognition net for expression recognition. In *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)*, pages 118–126. IEEE, 2017.

[22] IKHADRAOUI DJIHENE. La détection des points caractéristiques dans un visage. In *Expressions faciales pour la détection de fatigue*, 2019.

[23] Dennis Núñez Fernández. Multi-subject continuous emotional states monitoring by using convolutional neural networks. In *2019 International Conference on Control of Dynamical and Aerospace Systems (XPOTRON)*, pages 1–4. IEEE, 2019.

[24] Nico H Frijda et al. *The emotions*. Cambridge University Press, 1986.

[25] Khadoudja Ghanem. Reconnaissance des expressions faciales à base d'informations video ; estimation de l'intensité des expressions faciales. 2010.

[26] Deepak Ghimire, Joonwhoan Lee, Ze-Nian Li, and Sunghwan Jeong. Recognition of facial expressions based on salient geometric features and support vector machines. *Multimedia Tools and Applications*, 76(6) :7921–7946, 2017.

[27] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[28] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[29] Guodong Guo, Stan Z Li, and Kapluk Chan. Face recognition by support vector machines. In *Proceedings fourth IEEE international conference on automatic face and gesture recognition (cat. no. PR00580)*, pages 196–201. IEEE, 2000.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[31] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. In *Neural networks : Tricks of the trade*, pages 599–619. Springer, 2012.

[32] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks : Training neural networks with low precision weights and activations. *The Journal of Machine Learning Research*, 18(1) :6869–6898, 2017.

[33] Bendjillali Ridha Ilyas, Beladgham Mohammed, Merit Khaled, Abdelmalik Taleb Ahmed, and Alouani Ihsen. Facial expression recognition based on dwt feature for deep cnn. In *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 344–348. IEEE, 2019.

[34] Takeo Kanade. Picture processing system by computer complex and recognition of human faces. 1974.

[35] Byoung Chul Ko. A brief review of facial emotion recognition based on visual information. *sensors*, 18(2) :401, 2018.

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[37] Andreas Lanitis, Christopher J. Taylor, and Timothy F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and machine intelligence*, 19(7) :743–756, 1997.

[38] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998.

[39] Michael Lyons, Shigeru Akamatsu, Miyuki Kamachi, and Jiro Gyoba. Coding facial expressions with gabor wavelets. In *Proceedings Third IEEE international conference on automatic face and gesture recognition*, pages 200–205. IEEE, 1998.

[40] Michael J Lyons, Julien Budynek, and Shigeru Akamatsu. Automatic classification of single facial images. *IEEE transactions on pattern analysis and machine intelligence*, 21(12) :1357–1362, 1999.

[41] Ahmed Maalej, Boulbaba Ben Amor, and Mohamed Daoudi. Analyse locale de la forme 3d pour la reconnaissance d'expressions faciales. 2011.

[42] Masakazu Matsugu, Katsuhiko Mori, Yusuke Mitari, and Yuji Kaneda. Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, 16(5-6) :555–559, 2003.

[43] Albert Mehrabian and Susan R Ferris. Inference of attitudes from nonverbal communication in two channels. *Journal of consulting psychology*, 31(3) :248, 1967.

[44] Hugo Mercier. Analyse automatique des expressions du visage. *Application à la langue des signes. Master report, UPS Toulouse*, 19, 2003.

[45] Hugo Mercier. Outils informatiques d'analyse des expressions faciales en langue des signes. *Paul Sabatier Toulouze III*, 2007.

[46] Uroš Mlakar, Iztok Fister, Janez Brest, and Božidar Potočnik. Multi-objective differential evolution for feature selection in facial expression recognition systems. *Expert Systems with Applications*, 89 :129–137, 2017.

[47] Mohammed Zakaria Mokri. *Classification des images avec les réseaux de neurones convolutionnels.* PhD thesis, 09-01-2018, 2017.

[48] Djaloul Youcef Moualek. *Deep Learning pour la classification des images.* PhD thesis, 07-03-2017, 2017.

[49] Foued NACER. Reconnaissance d'expression faciale à partir d'un visage réel. 2019.

[50] Mina Navraan, Nasrollah Moghadam Charkari, and Muharram Mansoorizadeh. Automatic facial emotion recognition method based on eye region changes. *Information Systems & Telecommunication*, page 221, 2016.

[51] Andrew Y Ng and Michael I Jordan. On discriminative vs. generative classifiers : A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848, 2002.

[52] Curtis Padgett and Garrison W Cottrell. Representing face images for emotion classification. In *Advances in neural information processing systems*, pages 894–900, 1997.

[53] Maja Pantic and Leon J. M. Rothkrantz. Automatic analysis of facial expressions : The state of the art. *IEEE Transactions on pattern analysis and machine intelligence*, 22(12) :1424–1445, 2000.

[54] Vytautas Perlibakas. Face recognition using principal component analysis and log-gabor filters. *arXiv preprint cs/0605025*, 2006.

[55] P Jonathon Phillips, Sandor Z Der, Patrick J Rauss, and Or Z Der. *FERET (face recognition technology) recognition algorithm development and test results.* Army Research Laboratory Adelphi, MD, 1996.

[56] P Jonathon Phillips, Hyeonjoon Moon, Syed A Rizvi, and Patrick J Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 22(10) :1090–1104, 2000.

[57] Robert Plutchik. A general psychoevolutionary theory of emotion. In *Theories of emotion*, pages 3–33. Elsevier, 1980.

[58] Chao Qi, Min Li, Qiushi Wang, Huiquan Zhang, Jinling Xing, Zhifan Gao, and Huailing Zhang. Facial expressions recognition based on cognition and mapped binary patterns. *IEEE Access*, 6 :18795–18803, 2018.

[59] Brian Sagi, Syrus C Nemat-Nasser, Rex Kerr, Raja Hayek, Christopher Downing, and Robert Hecht-Nielsen. A biologically motivated solution to the cocktail party problem. *Neural Computation*, 13(7) :1575–1602, 2001.

[60] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798, 2007.

[61] Shiguang Shan, Peng Yang, Xilin Chen, and Wen Gao. Adaboost gabor fisher classifier for face recognition. In *International Workshop on Analysis and Modeling of Faces and Gestures*, pages 279–292. Springer, 2005.

[62] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*, 2014.

[63] Catherine Soladie. *Représentation invariante des expressions faciales. : Application en analyse multimodale des émotions.* PhD thesis, Supélec, 2013.

[64] Daniel Llatas Spiers. Facial emotion detection using deep learning, 2016.

[65] Nitish Srivastava and Russ R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230, 2012.

[66] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv :1602.07261*, 2016.

[67] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[68] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1) :71–86, 1991.

[69] Michel F Valstar, Hatice Gunes, and Maja Pantic. How to distinguish posed from spontaneous smiles using geometric features. In *Proceedings of the 9th international conference on Multimodal interfaces*, pages 38–45, 2007.

[70] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in neural information processing systems*, pages 2643–2651, 2013.

[71] Haohan Wang and Bhiksha Raj. On the origin of deep learning. *arXiv preprint arXiv :1702.07800*, 2017.

[72] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.