

Remerciements

C'est avec un grand plaisir qu'on vient remercier par ces petits mots tous ceux qui nous ont aidé à la concrétisation de ce modeste travail.

Nous adressons tout d'abord nos vifs remerciements à M. HENNI Sid Ahmed pour avoir accepté de nous encadrer et d'avoir enrichi ce mémoire par ses précieux conseils et ses nombreuses orientations.

Nous remercions également M. Mohamed BENAOUALI et M. Mohamed BENTOUMI d'en être les examinateurs.

Nous remercions également au Dr. LARBI BEKLAOUZ Hadj, qui nous à encourager à reprendre et poursuivre notre cursus universitaire et académique.

Nous tenons à exprimer notre profonde reconnaissance à M. Mohamed BENGOUDIFA, Directeur d'agence de la Caisse Nationale du Logement de Mostaganem pour son soutien moral, son encouragement continu et son appui sans faille. On aimerait lui dire à travers ces quelques mots qu'il est une personne dotée de grandes valeurs humaines et d'un dynamisme et un savoir-faire exceptionnelles, on a beaucoup appris en le côtoyant et on est fière et ravi d'avoir été un jour dans son équipe de travail.

Dédicaces

A mes parents ... qui ont su inculquer en moi la soif d'apprendre.

A mon épouse ... pour son soutien et ses encouragements.

A mes adorables fillettes : Sabrina, Aya et Lina ... pour qui je souhaite plein de joie, de succès et de réussite dans la vie en général et dans leurs parcours pédagogiques en particulier.

A mon binôme Abed SELMA ...

AIT AOUDIA Amar

A ma grande famille ...

A ma petite famille ...

A tous mes amis ...

SELMA Abed

Table des matières

1	Chapitre 1 : Les Expressions Faciales : Etat de l'art.....	11
1.1	Introduction :.....	11
1.2	Définition.....	12
1.3	Domaines d'application.....	13
1.4	Processus d'analyse automatique des expressions faciales	14
1.4.1	Détection de visage	14
1.4.2	Extraction des caractéristiques principales.....	16
1.4.3	Classification des expressions faciales	17
1.5	Approches d'analyse automatique des expressions faciales	17
1.5.1	Les approches globales	17
1.5.2	Les approches locales.....	18
1.5.3	Les approches hybrides.....	18
1.6	Système de codage d'actions faciales	20
1.6.1	Le système de codage d'actions faciales (FACS)	20
1.6.2	Codage MPEG4.....	25
1.6.3	Candide.....	26
1.7	Bases des données d'expressions faciales	29
1.8	Conclusion	32
2	– Chapitre 2 : Réseaux Neuronaux et Apprentissage Profond.....	34
2.1	- Introduction :	34
2.2	Notions de base [17]:	34
2.2.1	L'intelligence artificielle (AI).....	34
2.2.2	Apprentissage automatique (ML)	35
2.2.3	Apprentissage profond (DL)	35
2.3	Concept du Deep Learning.....	36
2.4	Réseaux de neurones convolutifs CNN	37
2.4.1	Partie convolutive	37
2.4.2	Partie classification.....	38
2.5	Architecture d'un Convolutional Neural Network	38
2.5.1	Couche de convolution (CONV).....	39
2.5.2	Couche de Pooling (POOL)	40
2.5.3	La couche d'activation ReLU (Rectified Linear Units).....	41

2.5.4	Couche Fully Connected (FC).....	42
2.6	Réseaux Convolutifs Célèbres	42
2.7	Conclusion	43
3	Chapitre 3 : Implémentation et Réalisation	44
3.1	Introduction	44
3.2	Environnement de travail.....	44
3.2.1	Environnement matériel	44
3.2.2	Environnement de développement	45
3.2.3	Bibliothèques utilisées	47
3.3	La Base De Données	48
3.4	Implémentation et Réalisation.....	52
3.5	Présentation de l'application	Erreur ! Signet non défini.
3.6	Évaluation de notre modèle.....	60
3.7	Conclusion	61
4	Conclusion Générale :	62

Liste d Figures

Figure 1-1: Les muscles mimiques de la tête et leurs actions [19]	11
Figure 1-2: Joie, Colère, Surprise correspondent chacune à une expression faciale unique [18]	13
Figure 1-3: Etapes de base pour l'analyse des expressions faciales	14
Figure 1-4: Organigramme des différentes méthodes d'analyse automatique des expressions faciales ...	19
Figure 1-5: Quelques images des unités d'actions faciales [23]	21
Figure 1-6: Modèle MPEG4 l'ensemble d'attributs faciaux [14].....	26
Figure 1-7: En haut : CANDIDE-1 avec 79 sommets et 108 surfaces. En bas : CANDIDE-2 avec 160 sommets et 238 surfaces [15]	28
Figure 2-1: Représentation des techniques de science des données	34
Figure 2-2: Comparaison entre un neurone biologique et un perceptron [24]	36
Figure 2-3: Architecture générale du réseau de neurones artificiels ANN [25].....	37
Figure 2-4: Architecture de base du réseau CNN [26]	38
Figure 2-5: Schéma du parcours de la fenêtre de filtre sur l'image [27]	39
Figure 2-6: Processus de Max-Pooling [27]	41
Figure 2-7: Fonction d'activation ReLU [27].....	41
Figure 2-8: Fonctionnement d'un réseau neuronal à 2 couches cachées [27]	42
Figure 3-1: Nombre d'images disponibles pour l'apprentissage	49
Figure 3-2 : Nombre d'images disponibles pour le test	50
Figure 3-3 : Echantillon des images de la base de données.....	51
Figure 3-4: La courbe de perte générée pour notre modèle	59
Figure 3-5: La courbe de précision générée pour notre modèle	59
Figure 3-6: Echantillon de personnes testées par sexe et âge.....	60
Figure 3-7: Quelques résultats de test obtenus sur des cas réels.....	60

Liste des Tableaux

Table 1-1: Classification des différentes approches de détection de visages et leurs méthodes [10]	16
Table 1-2: Liste des Action Units et Action Descriptors (avec le détail des muscles actionnés) [13]	22
Table 1-3: Les actions faciales reliés aux émotions universelles [13]	24
Table 1-4: Exemples de Bases de données [16]	29
Table 3-1: Nombre d'images par catégorie.....	49

Liste Des Abréviations

RAEF	Reconnaissance Automatique des Expressions Faciales
FER	Facial Expression Recognition
ROI	Regions Of Interest
CNN	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

ملخص:

تأثرت العديد من جوانب حياتنا اليومية مثل: السلامة، الصحة، التسويق وواجهات الإنسان والآلة ... إلخ ، بمجال التعرف التلقائي على تعابير الوجه ، الذي يعتبر ميدانا خصبا للبحوث. لكن النجاح الواسع الذي عرفه "التعلم العميق" دفع الباحثين لاستغلال الأنواع المختلفة من بنى هذه التقنية للحصول على أداء أفضل. في هذه المذكرة ، نقترح مساهمتنا في التعرف على تعابير الوجه باستخدام نموذج التعلم الخاص بنا. أولاً ، يتم تحديد الوجه من فيديو في الوقت الفعلي ، ومن ثم استخراج الملامح العميقة باستخدام شبكة عصبية تلافيفية ، ثم يتم نقلها إلى المصنف للتعرف بأحد تعابير الوجه السبعة التي حددها دوشين ، وهي: الفرح ، الحزن ، الغضب ، الاشمئزاز ، الخوف ، المفاجأة والحياد.

الكلمات المفتاحية: تعبيرات الوجه ، التعلم العميق ، الشبكة العصبية التلافيفية

ABSTRACT:

Several aspects of our daily life such as: security, health, marketing, and human-computer interfaces ...etc, are affected by the automatic recognition of facial expressions (ARFE), which is a vast field of research. But the success of Deep Learning, has pushed researchers to exploit the different types of architectures of this technique, to obtain better performance. In this thesis, we propose our contribution to facial expression recognition using our learning model. First, the face is detected from real time videos, a deep feature extraction performed using convolutional neural network, and then passed to a classifier to predict one of the seven facial expressions defined by Duchenne, which are: joy, sadness, anger, disgust, fear, surprise, neutrality.

Keywords: Facial expression, Deep Learning, Convolutional neural network

RÉSUMÉ :

Plusieurs aspects de notre vie quotidienne tels que : la sécurité, la santé, le marketing et les interfaces homme-machine ...etc, sont affectés par la reconnaissance automatique des expressions faciales (RAEF), qui est un vaste champ de recherche. Mais le succès de l'apprentissage profond (Deep Learning), a poussé les chercheurs à exploiter les différents types d'architectures de cette technique, pour obtenir de meilleures performances. Dans ce mémoire, Nous proposons notre contribution à la reconnaissance des expressions faciales en utilisant notre modèle d'apprentissage. Tout d'abord, le visage est détecté à partir de vidéos en temps réel, une extraction des caractéristiques profondes réalisé à l'aide de réseau neuronal convolutif, puis transmises à un classificateur pour prédire l'une des sept expressions faciales définies par Duchenne, et qui sont : la joie, la tristesse, la colère, le dégoût, la peur, la surprise, la neutralité.

Mots-clés : Expression faciale, Deep Learning, Réseau neuronal convolutif

INTRODUCTION GENERALE

Notre visage, nous permet de transmettre aux autres, une pensée, un ressenti particulier, un état émotionnel ou véhiculer un message. Il est le plus complexe tous les canaux de communication. Les expressions faciales fournissent des informations sociales à notre entourage. On peut sourire pour montrer notre bonheur ou notre accord, froncer les sourcils pour montrer à quelqu'un qu'on est agacé ou en colère, comme on peut soulever les sourcils pour poser une question ou appuyez un mot.

Avec 43 groupes musculaires au niveau du visage situés juste sous la surface de la peau, on peut former plusieurs milliers de combinaisons différentes. Parmi toutes ces combinaisons, sept d'entre elles reflètent ce que l'on appelle communément les expressions faciales universelles, relatives aux émotions et qui sont :

1. La joie
2. La tristesse
3. La colère
4. La peur
5. La surprise
6. Le mépris
7. Le dégoût

Ces expressions faciales communes à tous les êtres humains ont été citées pour la première fois par le neurologue français Guillaume Duchenne de Boulogne [1] et étudiées plus avant par le naturaliste Charles Darwin [2]. Au cours de ces dernières années, la recherche et le développement de technologies dans le champ de l'analyse des expressions faciales ont fortement augmenté pour cibler divers centres d'intérêt. En marketing analyser les émotions des utilisateurs face à des publicités en ligne aide à cibler leurs besoins. Dans le domaine du divertissement, Les grands studios d'animation ont étudié les méthodes liées aux expressions faciales détaillées par le célèbre psychologue américain Paul Ekman [3] afin de rendre leurs personnages aussi réalistes que possible, de même pour les jeux vidéo les concepteurs ont appliqué ces méthodes à des avatars en 3D et à certains robots grâce au machine Learning. Dans le domaine de la santé et de l'assistance sociale des nouvelles technologies ouvrent de nouveaux horizons en matière

de prise en charge des personnes les plus vulnérables notamment les personnes âgées. Divers algorithmes d'extraction de caractéristiques et d'apprentissage automatique ont été développés, tels que la méthode SVM [4], Adaboost [5], ou encore la méthode Forêts aléatoires [6]. Le succès des réseaux de neurones convolutifs (CNN) dans le traitement d'images s'est étendu au problème de la reconnaissance de l'expression faciale. Le Deep Learning et plus particulièrement les réseaux de neurones Convolutifs (CNN) ont apparus 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 de trois chapitres, et il est organisé comme suit : – Dans le premier chapitre, nous présentons des notions importantes sur les expressions faciales. – Ensuite, le second chapitre, nous le consacrons à la présentation des différentes méthodes et outils utilisés pour l'analyse et la détection des expressions faciales. – Enfin dans le troisième chapitre nous présentons notre contribution à la conception et implémentation de notre système de reconnaissance des expressions faciales basé sur les CNNs. – Nous clorons ce mémoire par une conclusion générale et quelques perspectives.

CHAPITRE 1

LES EXPRESSIONS FACIALES : ÉTAT DE L'ART

1.1 Introduction :

Il est difficile voire impossible, de comprendre l'état d'esprit interne d'une personne sans jeter un œil sur son visage, c'est à travers ses multiples expressions qu'on se fait une idée sur l'état affectif et cognitif des gens qui nous entourent, leur humeur, leur tempérament, leurs intentions, leur degré de concentration, d'ennui ou de perplexité, leur degré de vigilance, ou de douleur et leurs traits de personnalité.

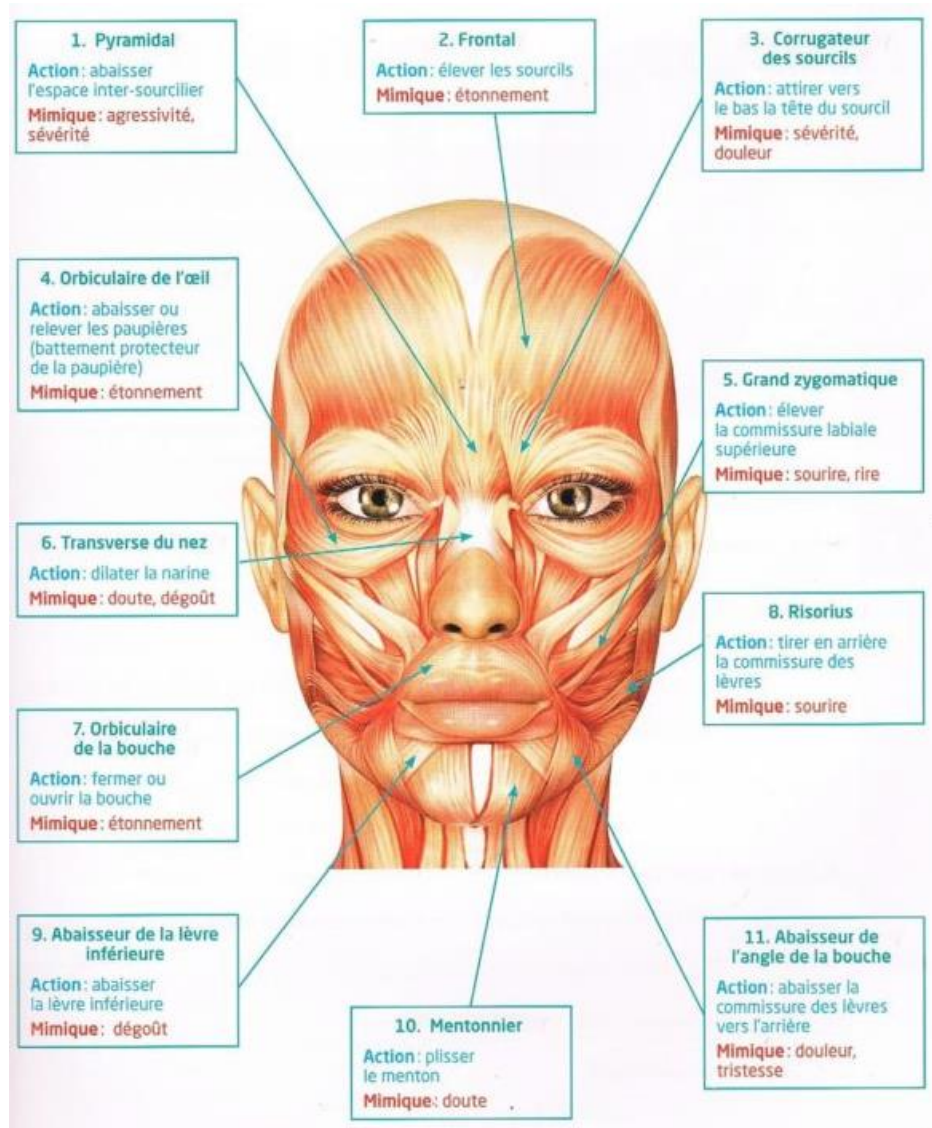


Figure 1-1: Les muscles mimiques de la tête et leurs actions [21]

Les expressions faciales sont la méthode de communication la plus naturelle des humains, avant même de pouvoir prononcer un mot, un nouveau-né sait "parler" avec son visage. En fait, tout être humain en bonne santé a la capacité de communiquer dans ce langage non verbal.

En effet, c'est le nerf facial qui stimule les muscles du visage, appelés aussi les muscles mimiques, qui se contractent à leur tour et font plier notre peau pour produire une variété étonnante d'expressions faciales (Figure 1-1).

Le célèbre psychologue américain Paul Ekman a pu déterminer qu'on combinant les contractions de 43 muscles du visage on pouvait produire une dizaine de milliers d'expressions, 3000 d'entre elles sont significatives [7].

Dans ce chapitre, on présentera une revue de littérature qui décrit les principales approches de détection des expressions faciales.

1.2 Définition

Tout d'abord, il est primordial de connaître la différence entre la reconnaissance d'expressions faciales et la reconnaissance d'émotions. Les émotions sont la combinaison de plusieurs facteurs observés comme : un ton élevé ou étouffé, les gestes des mains ou mouvement de la tête, une posture droite ou relâchée, un regard évasif ou appuyé en plus des expressions du visage (froncement du nez, sourcils abaissés, bouche ouverte).

En revanche, les émotions ne sont pas la seule origine des expressions faciales. En effet, celles-ci peuvent provenir de l'état d'esprit (réflexion, ennui), de l'activité physiologique (la douleur ou la fatigue) et de la communication non verbale (émotion simulée, clignotement de l'œil, froncement des sourcils). Néanmoins, sept émotions de base correspondent chacune à une expression faciale unique, et ce, quelles que soient l'ethnicité et la culture de la personne observée, ces émotions sont : la colère, le dégoût, l'étonnement, la joie, le mépris, la peur, et la tristesse.

La reconnaissance des expressions faciales consiste à classer les déformations des structures faciales et les mouvements faciaux uniquement à partir des informations visuelles. La reconnaissance des émotions, quant à elle, nécessite une interprétation qui exige une information contextuelle plus complète.



Figure 1-2: Joie, Colère, Surprise correspondent chacune à une expression faciale unique [20]

1.3 Domaines d'application

L'analyse de l'impact de toute situation, contenu, produit ou service, censé susciter des réponses faciales volontaires ou involontaires, est d'un intérêt majeur, car cette analyse permet de détecter l'état cognitif et affectif de la personne impactée et par conséquent comprendre et même prédire ses intentions, ses actions et ses réactions. De ce fait, l'analyse des expressions faciales trouve de nombreuses applications dans des domaines divers tels que [8]:

1. Marketing : des applications pour mesurer la satisfaction des clients, prévoir les produits qui les intéressent.
2. Médecine : aide à la détection de certaines maladies psychologiques, aide à l'apprentissage des émotions pour les enfants autistes.
3. Sécurité : détection du stress et de comportements suspects.
4. Interaction Homme-Machine : robot d'accompagnement, voiture intelligente.
5. Éducation : apprentissage à distance et le développement de jeux interactifs.

1.4 Processus d'analyse automatique des expressions faciales

Depuis le milieu des années 70, différentes approches ont été proposées pour l'analyse des expressions faciales des images faciales statiques ou des séquences d'images, elles s'accordent tous sur trois étapes de base qui sont la détection du visage, l'extraction des expressions faciales, et la classification de ces dernières.

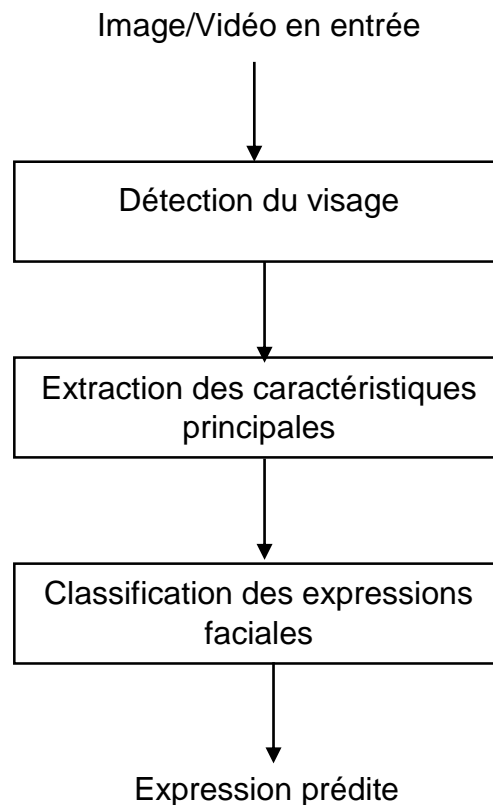


Figure 1-3: Etapes de base pour l'analyse des expressions faciales

1.4.1 Détection de visage

La détection de visage est l'un des champs d'études le plus active de la vision par ordinateur, c'est un cas particulier de la détection des formes, où l'on cherche à détecter la présence et la localisation exacte du visage dans une image.

Elle joue actuellement un rôle primordial en tant que première étape dans de nombreuses applications clés, notamment la reconnaissance de visage, vérification de personnes, télésurveillance, détection des expressions faciales, l'interaction homme-machine. La détection des visages a un effet très significatif car c'est sur sa fiabilité et sa précision que repose les performances de tels systèmes.

Yan, Kriegman et Ahuja [9] ont présenté une classification de ces méthodes en quatre catégories, et les algorithmes de détection des visages peuvent appartenir à deux ou plusieurs groupes. Ces catégories sont les suivantes :

- ✓ Méthodes basées sur les connaissances:(locale, top-down)
 - encode les connaissances de ce qui constitue un visage
 - décrit les relations entre traits de visages
- ✓ Méthodes basées sur la Corrélacion de modèles de visage : (globale, template matching)
 - compare avec des prototypes de visages au complet ou de parties de visages
- ✓ Méthodes basées sur les traits invariants:(locale, bottom-up)
 - tente d'isoler les traits structurels invariants d'un visage, même si la pose, la vue et la lumière varient
- ✓ Méthodes basées sur l'apparence par des méthodes de classification : (globale, ML-PR)
 - compare avec des modèles appris implicitement, à partir d'un ensemble représentatif d'images

Table 1-1: Classification des différentes approches de détection de visages et leurs méthodes [10]

Approche	Méthode utilisée
1- Méthodes descendantes basée sur la connaissance (Knowledge based method)	Méthode Multi résolution à base de règles
2- Méthodes ascendantes basées sur les traits invariants	
- Caractéristiques du visage	Grouping of edges
- Texture	Space Gray-level Dependence matrix of face pattern (SGLD)
- Couleur de la peau	Mixture of Gaussian (GMM)
- Caractéristiques multiples	Integration of skin color, size and shape
3- Méthodes basées sur les prototypes (Template matching)	
- Modèles prédéfinis de visage	Shape template
- Modèles déformables	Active Shape Model (ASM)
4- Méthodes basées sur l'apparence (Appearance-based method)	
- EigenFace	Eigenvector decomposition and clustering
- Distribution-based	Gaussian distribution and multilayer perceptron
- Réseau de neurones	Ensemble of neural networks and arbitration schemes
- Support Vector Machine (SVM)	SVM with polynomial of kernel
- Classificateur Naïve Bayes	Joint statistical of local appearance and position
- Modèle de Markov caché (HMM)	Higher order statistics with HMM
- Théorie de l'information	Kullback relative information

1.4.2 Extraction des caractéristiques principales

Une fois le visage détecté et localisé, il s'agit dans cette étape d'extraire les informations sur l'expression faciale produite en réduisant la dimensionnalité des régions d'intérêt (ROI) de telle sorte à conserver les caractéristiques invariantes et distinctives.

L'extraction de caractéristiques consiste à capturer les informations importantes de l'image sous forme de vecteurs, symboles et valeurs. La technique d'extraction des caractéristiques est décidée en fonction de l'environnement de l'application et de la méthode de classification : Modèle binaire local (LBP), Gabor, Haar, . . . etc. L'applicabilité et la faisabilité de l'application d'une méthode particulière devaient être prises en compte lors du choix d'une méthode d'extraction de caractéristiques appropriée, car cela peut avoir un contrôle direct sur les performances des algorithmes qui est un goulot d'étranglement des applications FER.

1.4.3 Classification des expressions faciales

Une classification est effectuée pour identifier l'expression faciale de l'image. La sélection d'un classeur approprié exerce une influence significative sur la précision de l'expression capturée dans les systèmes FER : Support Vector Machine (SVM), Bayesian, k-Nearest Neighbors (kNN), Probabilistic Neural Network (PNN) et Sparse Representation based Classifier (SRC) sont quelques-uns des classificateurs les plus utilisés dans les applications FER [11].

1.5 Approches d'analyse automatique des expressions faciales

Les méthodes de reconnaissance faciales peuvent être séparées en deux grandes familles, les méthodes globales (holistiques) et les méthodes locales (analytique), basées sur des modèles. La technique des réseaux neuronaux sera présentée plus détaillée dans le prochain chapitre.

1.5.1 Les approches globales

Les approches globales considèrent le visage dans sa totalité et lui appliquent des techniques d'analyse statistique bien connues. Leur principe repose sur la projection de l'image acquise en entrée du visage sous une forme vectorielle, dans un espace de plus faible dimension pour ne garder que les caractéristiques importantes et suffisamment discriminantes qui rendront la reconnaissance plus facile.

L'avantage de ces méthodes est la rapidité de leur mise en oeuvre, les calculs reposant sur des opérations matricielles relativement simples, cependant, elles sont sensibles aux contraintes telles que l'éclairage, la pose ou encore l'expression faciale.

Les techniques utilisées dans l'approche globale peuvent être regroupées en deux types : les techniques linéaires et les techniques non linéaires.

Nous pouvons distinguer deux types de techniques parmi les méthodes globales : les techniques linéaires et les techniques non linéaires.

1.5.1.1 *Les techniques linéaires*

Les techniques linéaires projettent linéairement les données d'un espace de grande dimension sur un sous-espace de dimension inférieure. L'inconvénient de ces techniques est le manque de précision 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.

La technique linéaire la plus connue est l'analyse en composantes principales (PCA).

1.5.1.2 *Les techniques non linéaires*

Afin de remédier au problème de non-linéarité les méthodes linéaires ont été étendues à des techniques non linéaires basées sur la notion mathématique de noyau "kernel" comme le Kernel PCA et le Kernel LDA [12].

1.5.2 *Les approches locales*

Les approches locales, basées sur des modèles, exploitent les connaissances qu'on possède sur la morphologie du visage pour extraire les points caractéristiques de celui-ci. 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 que les distributions des visages soient moins affectées par divers changements.

Les approches Bayésiennes, les machines à vecteurs de support, la méthode des modèles actifs d'apparence ou encore la méthode « local binary pattern » ont été utilisées dans ce but.

L'avantage de ces méthodes est 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 mettre en oeuvre.

1.5.3 *Les approches hybrides*

Les approches hybrides permettent de prendre profit des avantages 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'éclairage et d'expressions faciales.

L'analyse de caractéristiques locales et les caractéristiques extraites par ondelettes de Gabor (comme l'Elastic Bunch Graph Matching, sont des algorithmes hybrides typiques.

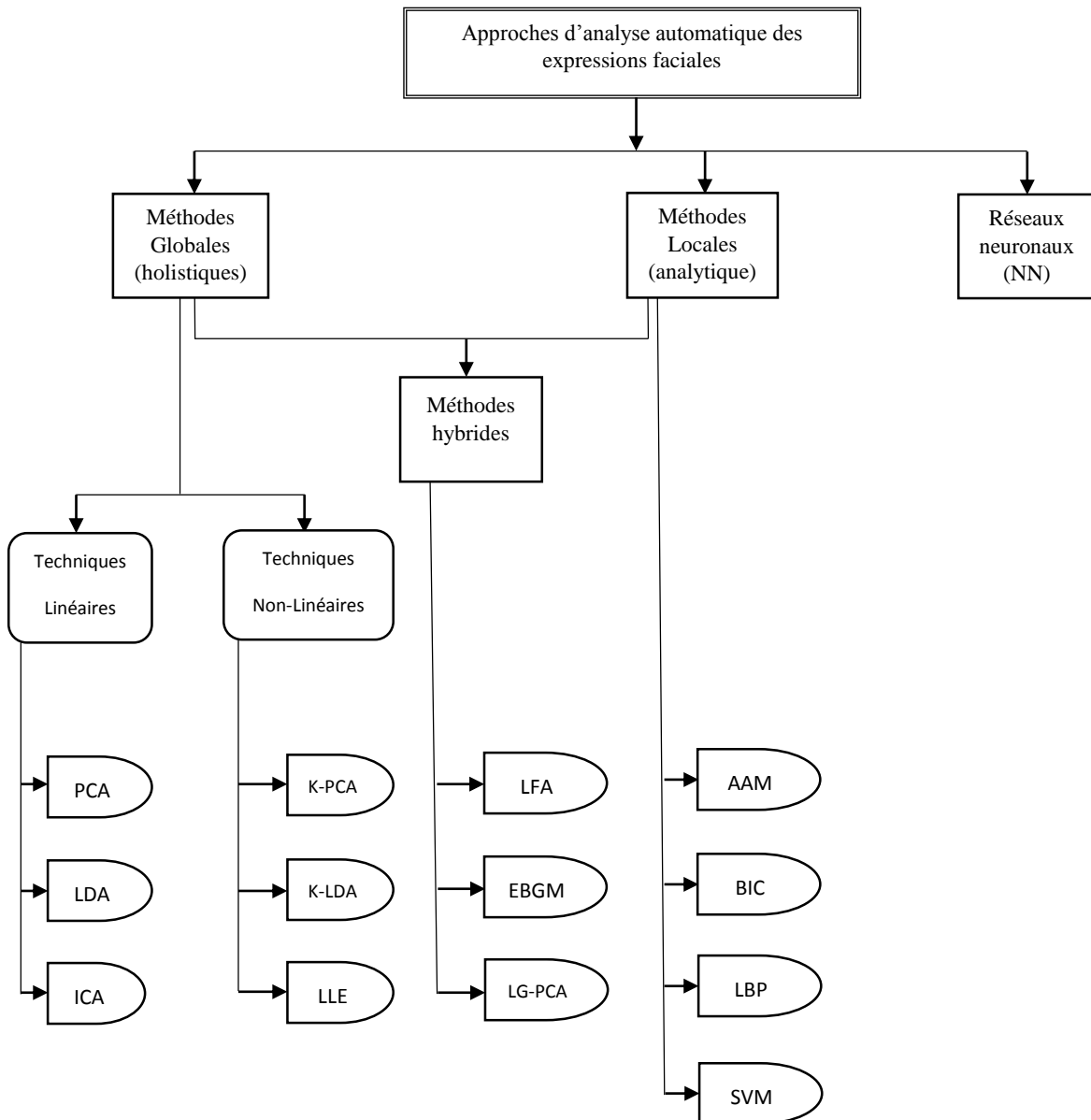


Figure 1-4: Organigramme des différentes méthodes d'analyse automatique des expressions faciales

1.6 Système de codage d'actions faciales

L'expression faciale physiologiquement parlant est le résultat de l'activité musculaire faciale. L'étude de l'expression faciale passe nécessairement par l'étude de l'anatomie du visage et des différents muscles faciaux.

Les chercheurs ont porté une attention particulière à l'étude et au codage des expressions faciales et ils ont proposé plusieurs systèmes et outils de codification. Nous allons de ce qui suit décrire les principes des systèmes les plus connus :

1.6.1 Le système de codage d'actions faciales (FACS)

Le Facial Action Coding System (FACS) est une méthode permettant de décrire les mouvements faciaux. Cette technique est le fruit des recherches des psychologues: Ekman, Wallace, Hager et Friesen dans les années 70. Une première édition est parue en 1978 et une importante mise à jour a été effectuée en 2002 [13]. Le FACS est à ce jour la méthode la plus largement reconnue et utilisée pour coder les mouvements du visage humain.

Le Facial Action Coding System permet de décomposer les mouvements faciaux en unité d'action. Il dissèque les expressions observées et les classe en actions isolées d'un ou plusieurs muscles peauciers du visage. Ces actions sont appelées AU (Action Unit). Au niveau du visage, il existe une trentaine d'AUs principales dont les combinaisons permettent de décrire environ 10 000 expressions faciales. De plus, le Facial Action Coding System permet aussi de décrire l'intensité et l'asymétrie d'une expression observée.



Upper Face Action Units					
AU 1	AU 2	AU 4	AU 5	AU 6	AU 7
					
Inner Brow Raiser	Outer Brow Raiser	Brow Lowerer	Upper Lid Raiser	Cheek Raiser	Lid Tightener
*AU 41	*AU 42	*AU 43	AU 44	AU 45	AU 46
					
Lid Droop	Slit	Eyes Closed	Squint	Blink	Wink
Lower Face Action Units					
AU 9	AU 10	AU 11	AU 12	AU 13	AU 14
					
Nose Wrinkler	Upper Lip Raiser	Nasolabial Deepener	Lip Corner Puller	Cheek Puffer	Dimpler
AU 15	AU 16	AU 17	AU 18	AU 20	AU 22
					
Lip Corner Depressor	Lower Lip Depressor	Chin Raiser	Lip Puckerer	Lip Stretcher	Lip Funneler
AU 23	AU 24	*AU 25	*AU 26	*AU 27	AU 28
					
Lip Tightener	Lip Pressor	Lips Part	Jaw Drop	Mouth Stretch	Lip Suck

Figure 1-5: Quelques images des unités d'actions faciales [25]

Parmi les AUs, certaines sont dites « fiables », ce qui signifie que leur mise en œuvre volontaire est difficilement réalisable. Pour qu'une AU soit fiable, il faut que moins de 25% de la population soit capable de réaliser volontairement et correctement l'AU au premier essai, et sans mobiliser d'autres AUs.

CHAPITRE 1 : LES EXPRESSIONS FACIALES : ETAT DE L'ART

Table 1-2: Liste des Action Units et Action Descriptors (avec le détail des muscles actionnés) [14]

N° AU	Nom de la FACS	Base musculaire
0	Visage neutre	
1	Remontée de la partie interne des sourcils	frontalis (pars medialis)
2	Remontée de la partie externe des sourcils	frontalis (pars lateralis)
4	Abaissement et rapprochement des sourcils	muscle pyramidal du nez, muscle procerus depressor supercilii, corrugator supercilii
5	Ouverture entre la paupière supérieure et les sourcils	Muscle orbiculaire de l'œil
6	Remontée des joues	Muscle orbiculaire de l'œil (pars orbitalis)
7	Tension de la paupière	Muscle orbiculaire de l'œil (pars palpebralis)
8	Lèvres collées	Muscle orbiculaire de la bouche
9	Plissement de la peau du nez vers le haut	Muscle releveur naso-labial
10	Remontée de la partie supérieure de la lèvre	Muscle releveur de la lèvre supérieure
11	Ouverture du nasolabial	Muscle petit zygomatique
12	Étirement du coin des lèvres	Muscle grand zygomatique
13	Étirement et rentrée des lèvres	Muscle élévateur de l'angle de la bouche (également appelé caninus)
14	Plissement externe des lèvres (fossettes)	Muscle buccinateur
15	Abaissement des coins externes des lèvres	Muscle abaisseur de l'angle de la bouche
16	Ouverture de la lèvre inférieure	Muscle abaisseur de la lèvre inférieure
17	Élévation du menton	Muscle mentonnier
18	Froncement central des lèvres	Muscle orbiculaire de la bouche
19	Sortie de la langue	
20	Étirement externe des lèvres	Muscle risorius et muscle peaucier du cou
21	Tension du cou	muscle peaucier du cou
22	Lèvres en "O"	Muscle orbiculaire de la bouche
23	Tension refermante des lèvres	Muscle orbiculaire de la bouche
24	Lèvres pressées (pincement des lèvres)	Muscle orbiculaire de la bouche
25	Ouverture de la bouche et séparation légère des lèvres	Muscle abaisseur de la lèvre inférieure, ou relaxation du muscle orbiculaire de la bouche

N° AU	Nom de la FACS	Base musculaire
26	Ouverture de la mâchoire	Muscle masséter ; muscle temporal relâché et Muscle ptérygoïdien médial
27	Bâillement	Pterygoïdes, muscle digastrique
28	Succion interne des lèvres	Muscle orbiculaire de la bouche
29	Poussée de la mâchoire	
30	Déplacement de côté de la mâchoire	
31	Serrement de la mâchoire	Muscle masséter
32	Morsure des lèvres	
33	Gonflement des joues	
34	Bouffée des joues	
35	Aspiration des joues	
36	Bombement de la langue	
37	Essuyage des lèvres	
38	Dilatation des nasaux	nasalis (pars alaris)
39	Compression des nasaux	nasalis (pars transversa) et depressor septi nasi
41	Abaissement de la glabelle	Mouvement distinct de AU 4 : Muscle pyramidal du nez
42	Abaissement interne des sourcils	Mouvement distinct de AU 4: Depressor supercilii
43	Yeux fermés	Relaxation du levator palpebrae superioris
44	Rapprochement des sourcils	Mouvement distinct de AU 4 : muscle corrugateur du sourcil
45	Clignotement	Relaxation du levator palpebrae superioris ; contraction du muscle orbiculaire de l'œil (pars palpebralis)
46	Clignement de l'œil	Muscle orbiculaire de l'œil

Les AU n°1, 6, 10, 15 et 23 sont dites « fiables » car difficiles à simuler volontairement par plus de 25% de la population.

Table 1-3: Les actions faciales reliés aux émotions universelles [14]

Émotion	Action Units
Joie	6+12
Tristesse	1+4+15
Surprise	1+2+5B+26
Peur	1+2+4+5+20+26
Colère	4+5+7+23
Dégoût	9+15+16
Mépris	R12A+R14A

Le degré d'activation

L'intensité des FACS est notée par les lettres A à E (d'une intensité minimale à maximale) pour chaque nombre correspondant à une Action Unit (par exemple AU 1A correspond à un faible niveau d'AU 1 tandis qu'AU 1E correspond à l'intensité maximale).

A Trace

B Léger

C Marqué ou prononcé

D Sévère ou extrême

E Maximum

1.6.2 Codage 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 Paramètres - 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 relatif 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-corner lip) 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 [15].

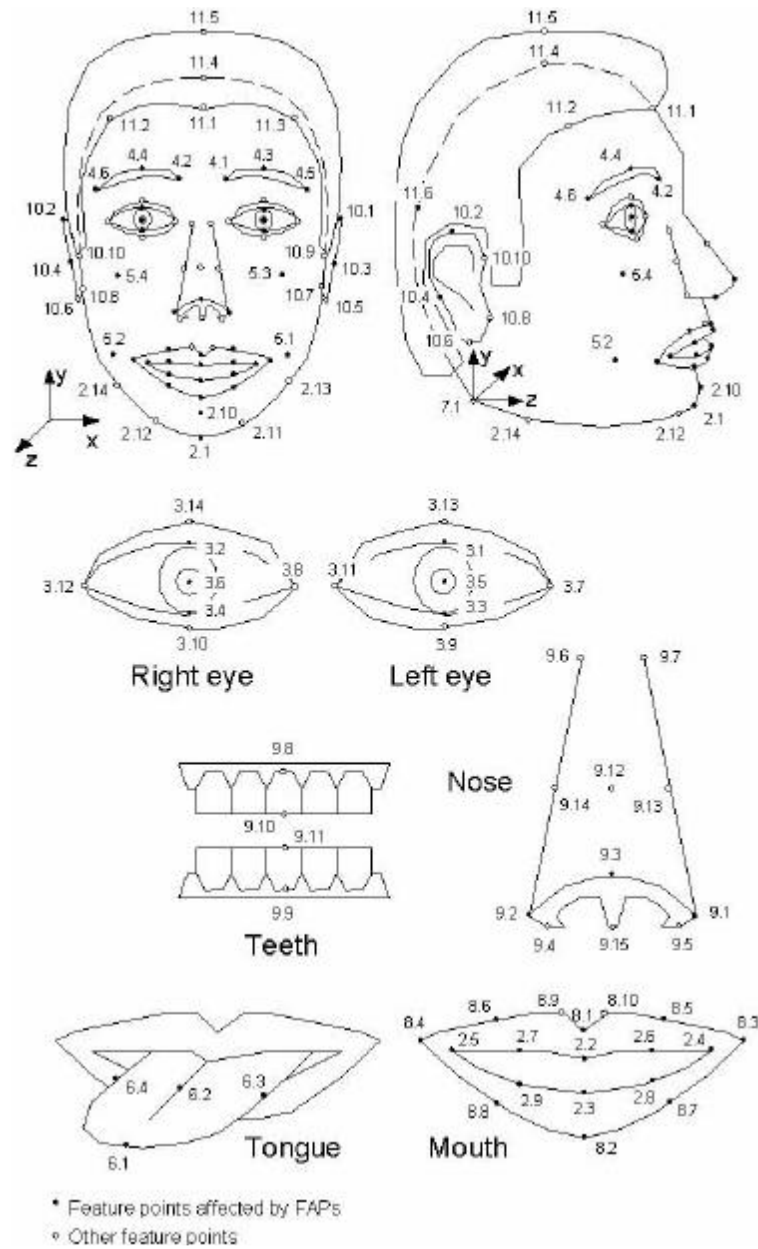


Figure 1-6: Modèle MPEG4 l'ensemble d'attributs faciaux [15]

1.6.3 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 [16]:

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 [16]:

- 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.

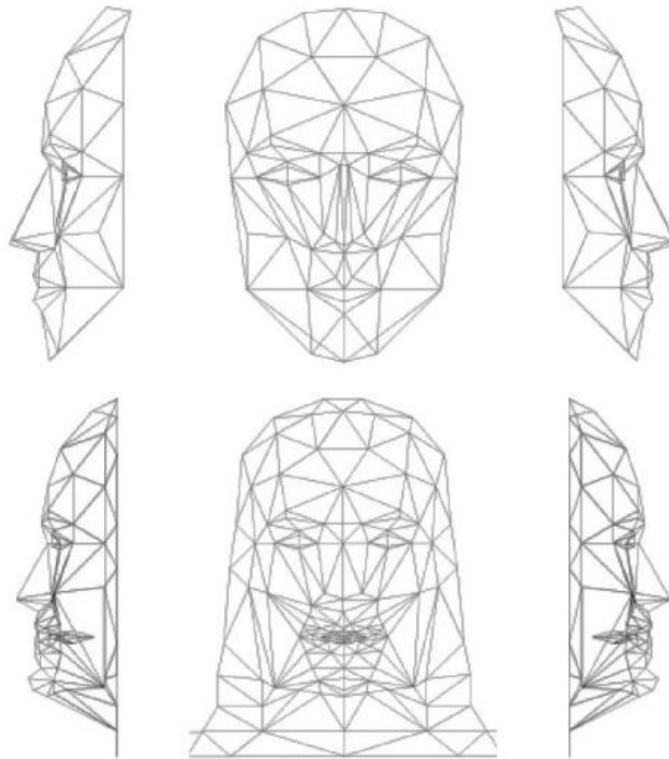


Figure 1-7: En haut : CANDIDE-1 avec 79 sommets et 108 surfaces.
En bas : CANDIDE-2 avec 160 sommets et 238 surfaces [16]

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, qui 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) [15].

1.7 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'expressions faciales 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 [17]

Table 1-4: Exemples de Bases de données [17]

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 Resolution d'image de 3000X 4000	http://cbcsl.ece.ohio-state.edu/dbform-compound.html
DISFA	130 000 images vidéo stéréo en haute resolution 27 sujets adultes (12 femmes et 15 hommes) 66 points	http://www.engr.du.edu/mmahoor/DISFA.htm

CHAPITRE 1 : LES EXPRESSIONS FACIALES : ETAT DE L'ART

	de repère du visage pour chaque image Resolution d'image de 1024X 768	
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 Resolution 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 Resolution d'image 256 256	http://www.kasrl.org/jaffeinfo.html
B+	16 128 images faciales 28 sujets distincts pour 576 conditions d'observation Resolution 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 resolution images de 75 sujets- 238 séquences vidéo sur 28 sujets, hommes et femmes-Resolution d'image de 720 - 576	https://mmifacedb.eu/
BP4DSpontaneous	La base de données vidéo 3D comprend 41 participantes (23 femmes, 18 hommes),	http://www.cs.binghamton.edu/lijun/Research/3DFE/3DFEAnalysis.html

CHAPITRE 1 : LES EXPRESSIONS FACIALES : ETAT DE L'ART

	avec des émotions faciales spontanées 11 Asiatiques, six Afro-Américains, quatre Hispaniques, et 20 euro-américains Resolution d'image de 1040 1329	
KDEF	4900 images d'expressions faciales humaines d'émotion 70 individus, sept expressions émotionnelles différentes avec 5 angles différents Resolution de l'image 562 762	http://www.emotionlab.se/resources/kdef
FER2013	Compte environ 37 000 personnes bien structurées En Images de gris a 48X48 pixels les visages regroupes automatiquement par l'API Google de recherche d'image	https://www.kaggle.com/datasets

1.8 Conclusion

Nous avons présenté dans ce chapitre une revue de littérature sur les méthodes de reconnaissance d'expressions faciales, ainsi que les travaux relatifs à ces approches. Nous avons discuté des théories et des représentations les plus répandues : les techniques de codifications, les approches de détection de visages ainsi que l'extraction des caractéristiques. Finalement, on a présenté un recueil des bases de données les plus fréquemment utilisées dans le domaine.

Dans le chapitre suivant on s'intéressera de plus près au domaine de l'apprentissage profond et des réseaux de neurones plus particulièrement au réseau de neurones convolutif.

CHAPITRE 2

RESEAUX NEURONAUX ET APPRENTISSAGE PROFOND

2.1 - Introduction :

Au cours de la dernière décennie, l'intelligence artificielle (IA) est devenue un sujet populaire chez la communauté scientifique et non scientifique, une abondance d'articles ont couvert les sujets de l'apprentissage automatique (AO) « Machine Learning », de l'apprentissage profond (AP) « Deep Learning » et de l'IA, Pourtant une certaine confusion entoure encore les notions l'IA, l'AO et l'AP.

En 1956, un groupe d'informaticiens ont proposé que les ordinateurs peuvent être programmés pour penser et raisonner, et "que chaque aspect de l'apprentissage ou toute autre caractéristique de l'intelligence peut, en principe, être décrit avec précision pour permettre à une machine de le simuler". Ils ont nommé ce principe "l'intelligence artificielle".

L'IA est un domaine axé sur l'automatisation des tâches intellectuelles normalement effectuées par les humains, et la ML et la DL sont des méthodes spécifiques pour atteindre cet objectif. En d'autres termes, elles font partie du domaine de l'IA.

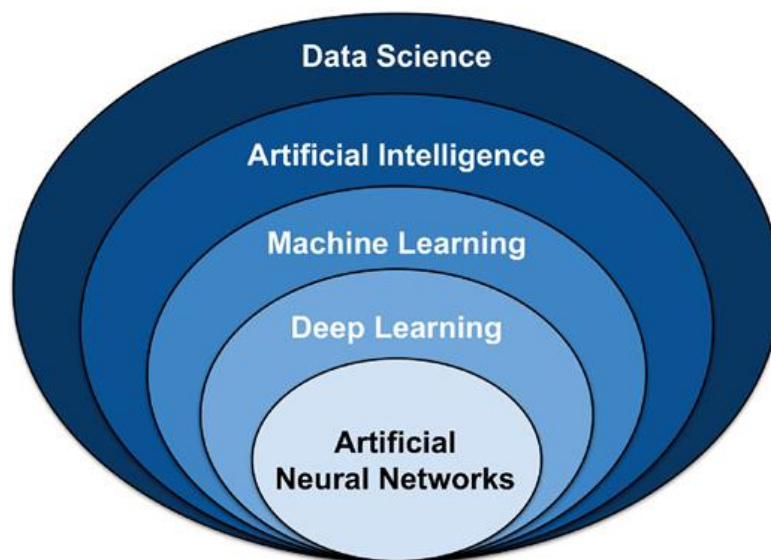


Figure 2-1: Représentation des techniques de science des données

2.2 Notions de base :

2.2.1 L'intelligence artificielle (AI)

L'IA est généralement définie comme la capacité d'une machine à exécuter des fonctions cognitives que nous associons à l'esprit humain, telles que la perception, le

raisonnement, l'apprentissage, l'interaction avec l'environnement, la résolution de problèmes et même l'exercice de la créativité. La robotique et les véhicules autonomes, la vision par ordinateur, le langage, les agents virtuels et l'apprentissage automatique sont des exemples de technologies qui permettent à l'IA de résoudre des problèmes commerciaux [18].

2.2.2 Apprentissage automatique (ML)

La plupart des progrès récents de l'IA ont été réalisés en appliquant l'apprentissage automatique (Machine Learning) à de très grands ensembles de données. Les algorithmes d'apprentissage automatique détectent des modèles et apprennent à faire des prédictions et des recommandations en traitant des données et des expériences, plutôt qu'en recevant des instructions de programmation explicites. Les algorithmes s'adaptent également en fonction des nouvelles données et expériences pour améliorer leur efficacité au fil du temps [18].

2.2.3 Apprentissage profond (DL)

L'apprentissage profond (Deep Learning) est un type d'apprentissage automatique qui peut traiter un plus large éventail de ressources de données, nécessite moins de prétraitement des données par les humains et peut souvent produire des résultats plus précis que les approches traditionnelles d'apprentissage automatique (bien qu'il nécessite une plus grande quantité de données pour y parvenir). Dans l'apprentissage profond, des couches interconnectées de calculateurs logiciels appelés "neurones" forment un réseau neuronal. Le réseau peut ingérer de grandes quantités de données d'entrée et les traiter à travers plusieurs couches qui apprennent des caractéristiques de plus en plus complexes des données à chaque couche. Le réseau peut ensuite déterminer les données, vérifier si sa détermination est correcte et utiliser ce qu'il a appris pour déterminer de nouvelles données. Par exemple, une fois qu'il a appris à quoi ressemble un objet, il peut reconnaître cet objet dans une nouvelle image [18].

2.3 Concept du Deep Learning

Le Deep Learning essaye de reproduire le fonctionnement du cerveau humain en calquant l'architecture du système nerveux qui est composé de milliards de cellules neurones biologiques liées entre elles et formant des réseaux plus ou moins complexes. Le Deep Learning s'appuie sur un réseau de neurones artificiels composés de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente (Figure 2-3).

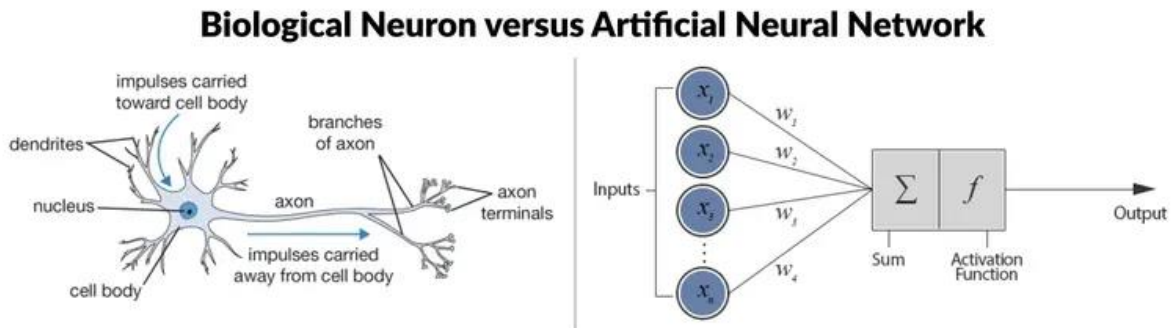


Figure 2-2: Comparaison entre un neurone biologique et un perceptron [26]

Le vecteur w porte le nom de vecteur de poids (qui s'ajuste lors de l'entraînement).

Le vecteur x porte le nom de vecteur d'entrée.

f porte le nom de fonction d'activation.

Un réseau de neurones artificiels est organisé comme suit :

- Un certain nombre de neurones d'entrée
- Un certain nombre de couches cachées
- Un certain nombre de neurones de sortie, qui correspondent aux prédictions que l'on veut.

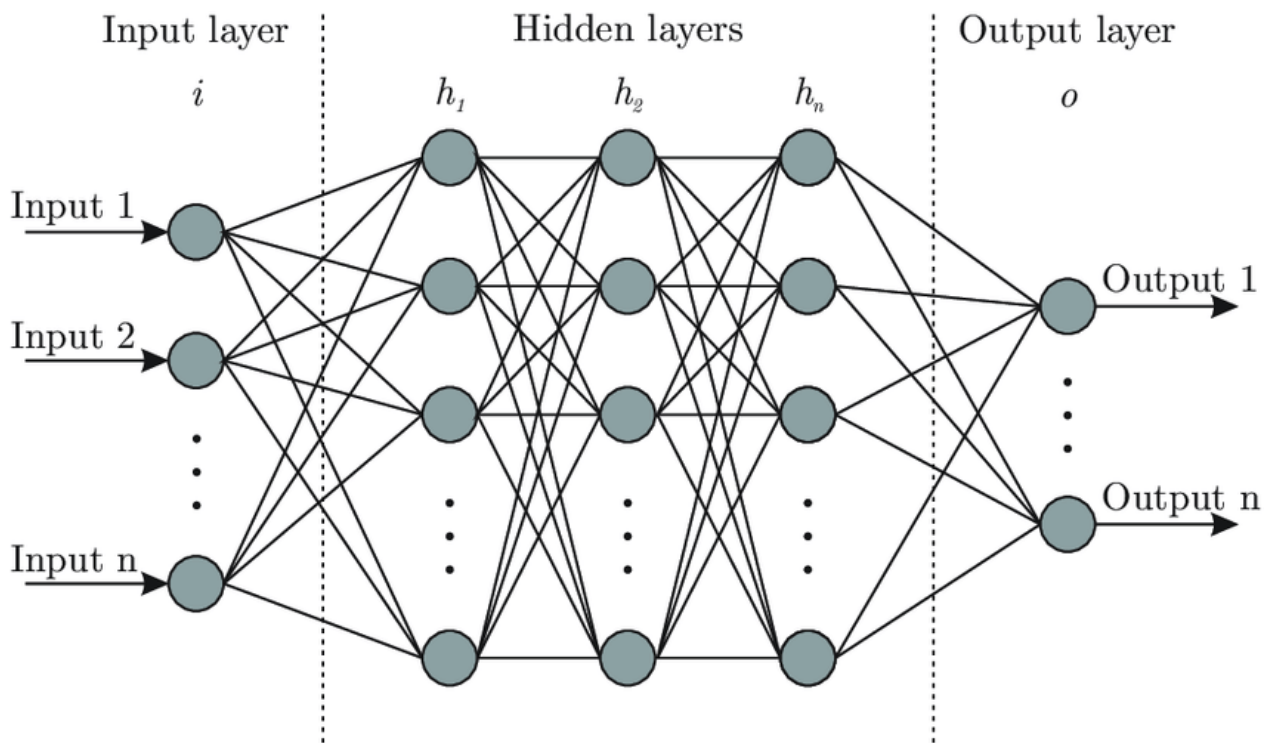


Figure 2-3: Architecture générale du réseau de neurones artificiels ANN [27]

Il existe des architectures plus performantes que d'autres, mais il n'y a pas vraiment de règle mathématique derrière. C'est l'expérience qui prime sur le choix des structures de modèle.

2.4 Réseaux de neurones convolutifs CNN

Les CNN désignent une sous-catégorie de réseaux de neurones artificiels qui désignent des modèles de classification d'images réputés être les plus performant, ils permettant notamment la reconnaissance d'images en attribuant automatiquement à chaque image fournie en entrée, une étiquette correspondant à sa classe d'appartenance.

Contrairement à un modèle MLP (Multi Layers Perceptron) classique qui ne contient qu'une partie classification, l'architecture du Convolutional Neural Network dispose en amont d'une partie convolutive et comporte par conséquent deux parties bien distinctes : partie convolutive et partie classification

2.4.1 Partie convolutive

Son objectif final est d'extraire des caractéristiques propres à chaque image en les compressant de façon à réduire leur taille initiale. En résumé, l'image fournie en entrée passe à travers une succession de filtres, créant par la même occasion de

nouvelles images appelées cartes de convolutions. Enfin, les cartes de convolutions obtenues sont concaténées dans un vecteur de caractéristiques appelé code CNN. Par exemple, il existe des filtres de convolution fréquemment utilisés et permettant d'extraire des caractéristiques plus pertinentes que des pixels comme la détection des bords (filtre dérivateur) ou des formes géométriques. Le choix et l'application des filtres se fait automatiquement par le modèle.

Parmi les filtres les plus connus, on retrouve notamment le filtre moyenneur (calcule pour chaque pixel la moyenne du pixel avec ses 8 proches voisins) ou encore le filtre gaussien permettant de réduire le bruit d'une image fournie en entrée.

2.4.2 Partie classification

Le code CNN obtenu en sortie de la partie convolutive est fourni en entrée dans une deuxième partie, constituée de couches entièrement connectées appelées perceptron multicouche (MLP pour Multi Layers Perceptron). Le rôle de cette partie est de combiner les caractéristiques du code CNN afin de classer l'image.

L'algorithme que les perceptrons utilisent pour mettre à jour leurs poids (ou coefficients de réseaux) s'appelle la rétropropagation du gradient de l'erreur, célèbre algorithme de descente de gradient.

2.5 Architecture d'un Convolutional Neural Network

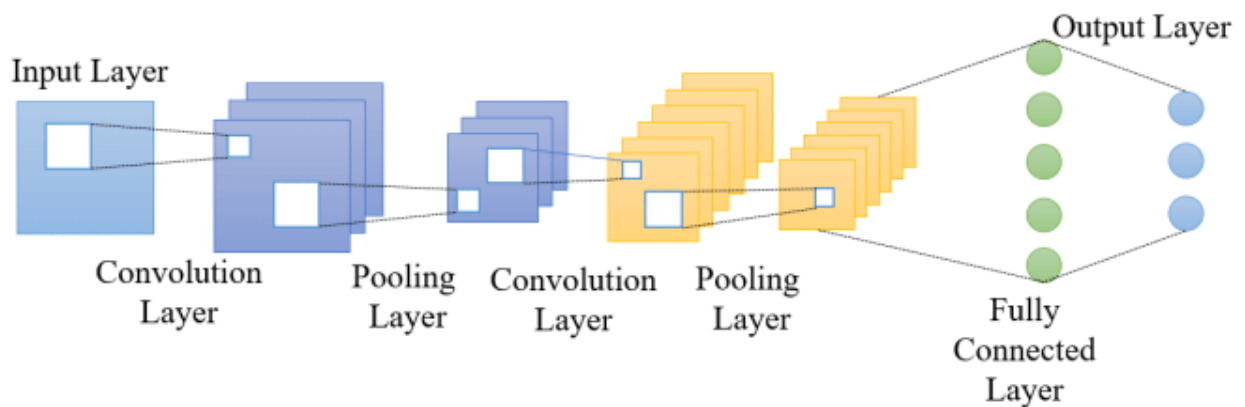


Figure 2-4: Architecture de base du réseau CNN [28]

Généralement, l'architecture d'un Convolutional Neural Network est sensiblement la même :

2.5.1 Couche de convolution (CONV)

Le rôle de cette première couche est d'analyser les images fournies en entrée et de détecter la présence d'un ensemble de features. On obtient en sortie de cette couche un ensemble de features maps.

La convolution est une opération mathématique simple généralement utilisée pour le traitement et la reconnaissance d'images. Sur une image, son effet s'assimile à un filtrage dont voici le fonctionnement :

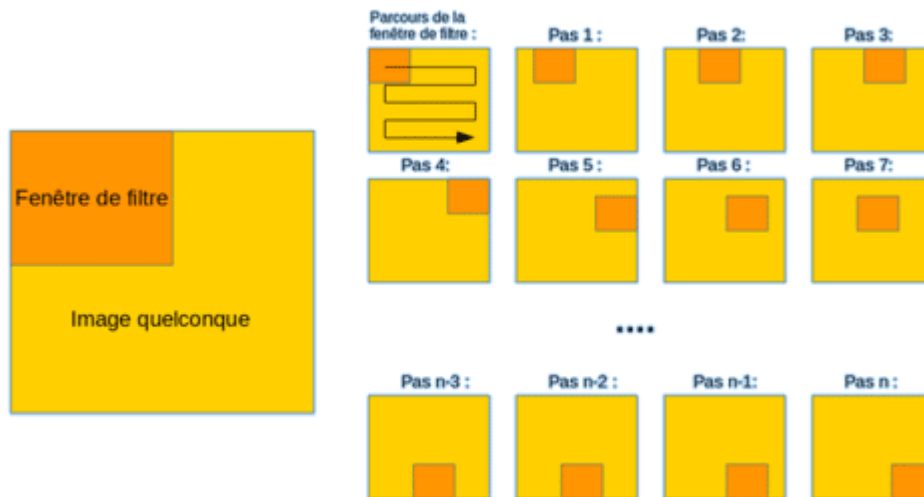


Figure 2-5: Schéma du parcours de la fenêtre de filtre sur l'image [29]

1. Dans un premier temps, on définit la taille de la fenêtre de filtre située en haut à gauche.
2. La fenêtre de filtre, représentant la feature, se déplace progressivement de la gauche vers la droite d'un certain nombre de cases défini au préalable (le pas) jusqu'à arriver au bout de l'image.
3. À chaque portion d'image rencontrée, un calcul de convolution s'effectue permettant d'obtenir en sortie une carte d'activation ou feature map qui indique où est localisées les features dans l'image : plus la feature map est élevée, plus la portion de l'image balayée ressemble à la feature.

CHAPITRE 2 : RESEAUX NEURONAUX ET APPRENTISSAGE PROFOND

Outre sa fonction de filtrage, l'intérêt de la partie convolutive d'un CNN est qu'elle permet d'extraire des caractéristiques propres à chaque image en les compressant de façon à réduire leur taille initiale, via des méthodes de sous-échantillonnage tel que le Max-Pooling.

2.5.2 Couche de Pooling (POOL)

La couche de Pooling est une opération généralement appliquée entre deux couches de convolution. Celle-ci reçoit en entrée les features maps formées en sortie de la couche de convolution et son rôle est de réduire la taille des images, tout en préservant leurs caractéristiques les plus essentielles. Parmi les plus utilisés, on retrouve le max-pooling ou encore l'average pooling dont l'opération consiste à conserver à chaque pas, la valeur moyenne de la fenêtre de filtre.

Finalement, on obtient en sortie de cette couche de Pooling, le même nombre de feature maps qu'en entrée mais considérablement compressées.

Le Max-Pooling est un processus de discrétisation basé sur des échantillons. Son objectif est de sous-échantillonner une représentation d'entrée (image, matrice de sortie de couche cachée, etc.) en réduisant sa dimension. De plus, son intérêt est qu'il réduit le coût de calcul en réduisant le nombre de paramètres à apprendre et fournit une invariance par petites translations (si une petite translation ne modifie pas le maximum de la région balayée, le maximum de chaque région restera le même et donc la nouvelle matrice créée restera identique).

Exemple :

Imaginons que nous avons une matrice 4x4 représentant notre entrée initiale et un filtre d'une fenêtre de taille 2x2 que nous appliquerons sur notre entrée. Pour chacune des régions balayées par le filtre, le max-pooling prendra le maximum, créant ainsi par la même occasion une nouvelle matrice de sortie où chaque élément correspondra aux maximums de chaque région rencontrée.

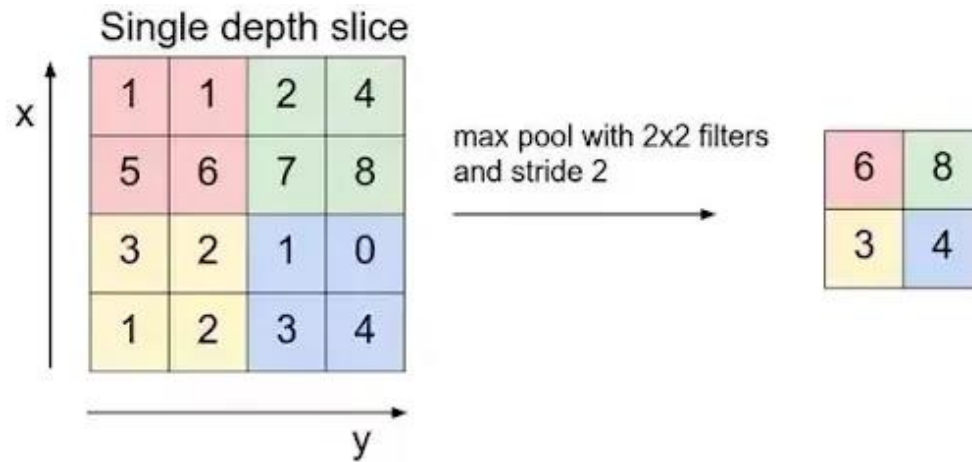


Figure 2-6: Processus de Max-Pooling [29]

La fenêtre de filtre se déplace de deux pixels vers la droite (stride/pas = 2) et récupère à chaque pas "l'argmax" correspondant à la valeur la plus grande parmi les 4 valeurs de pixels.

2.5.3 La couche d'activation ReLU (Rectified Linear Units)

Cette couche remplace toutes les valeurs négatives reçues en entrées par des zéros. L'intérêt de ces couches d'activation est de rendre le modèle non linéaire et de ce fait plus complexe.

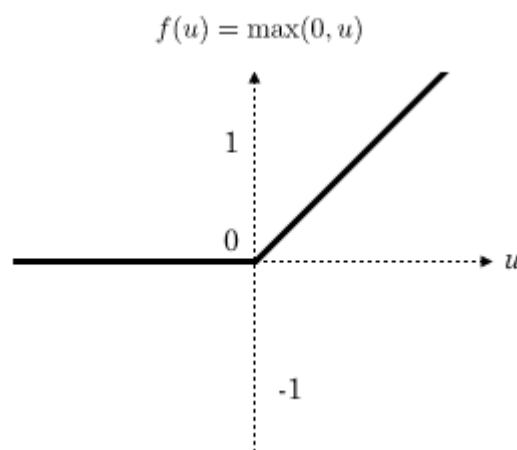


Figure 2-7: Fonction d'activation ReLU [29]

2.5.4 Couche Fully Connected (FC)

Ces couches sont placées en fin d'architecture de CNN et sont entièrement connectées à tous les neurones de sorties (d'où le terme fully-connected). Après avoir reçu un vecteur en entrée, la couche FC applique successivement une combinaison linéaire puis une fonction d'activation dans le but de classifier l'input image (Figure 2-8). Elle renvoie enfin en sortie un vecteur de taille d correspondant au nombre de classes dans lequel chaque composante représente la probabilité pour l'input image d'appartenir à une classe.

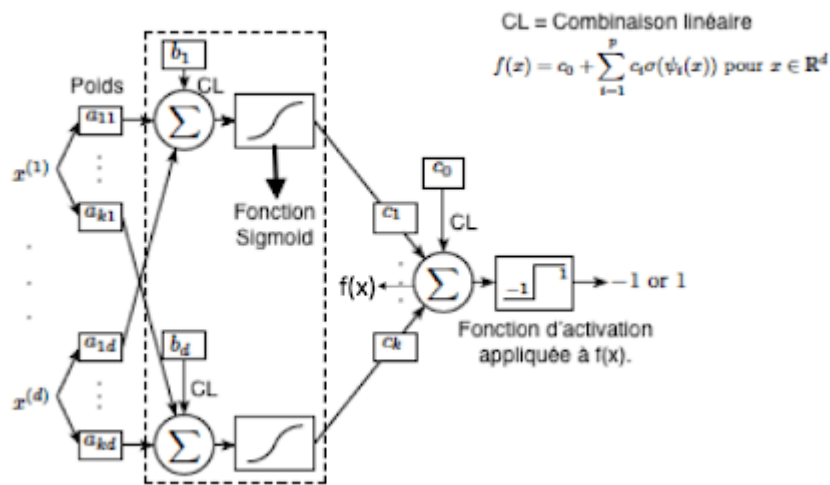


Figure 2-8: Fonctionnement d'un réseau neuronal à 2 couches cachées [29]

2.6 Réseaux Convolutifs Célèbres

- LeNet: 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: 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 a 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: C'était une amélioration de AlexNet en ajustant les hyperparamètres de l'architecture, en particulier en élargissant la taille des couches convolutifs et en réduisant la taille du noyau sur la première couche.

- GoogLeNet: 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 et Xception.

- ResNet: Residual network développé par Kaiming He et al. Ete 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: 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 présenté les notions phares du Deep Learning tout en essayant d'expliquer le principe de fonctionnement. On s'est ensuite attardé plus particulièrement sur les réseaux de neurones convolutifs CNN, on a présenté leur structure, et leurs différentes couches.

Le CNN possède quatre principales opérations : la convolution, la fonction non-linéarité (ReLU), Pooling et couche entièrement connectée. La première permet l'extraction des caractéristiques de l'image en entrée. La seconde permet de remplacer toutes les valeurs de pixels négatives par zéro. La troisième réduit progressivement la taille de la carte des caractéristiques rectifiée. Enfin la dernière permet la classification. Dans le chapitre suivant on présentera notre modèle pour la détection des expressions faciales en se basant sur l'architecture des réseaux de neurones convolutifs CNN.

CHAPITRE 3

IMPLEMENTATION ET REALISATION

3.1 Introduction

L'objectif de ce chapitre est de créer un modèle de Deep Learning capable de reconnaître sur un visage humain l'expression qui s'affiche. Le modèle aura sept types de réponse : 'colère', 'dégout', 'peur', 'joie', 'neutre', 'tristesse', 'surprise', c'est donc une classification catégorielle.

Nous allons prendre un jeu de données de 28 709 images pour l'entraînement et 7178 images pour le test. Notre base de données (déjà divisé en deux dossiers l'un d'entraînement et l'autre de test) est constituée d'images de différentes expressions faciales téléchargée à partir du dépôt Kaggle [19].

Nous allons dans ce qui suit détailler les étapes de l'implémentation de notre approche et les différentes étapes de réalisation.

Nous commençons tout d'abord par la présentation des ressources, du langage et de l'environnement de développement que nous avons utilisé, puis les étapes de la réalisation de notre modèle.

Nous terminerons ce chapitre par la présentation des différents résultats expérimentaux obtenus, appuyés par des captures d'écrans et une brève discussion.

3.2 Environnement de travail

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

3.2.1 Environnement matériel

Afin de mener à bien ce projet, on s'est mené d'un ordinateur portable HP Folio 9480m avec les caractéristiques suivantes :

- Processeur: Intel(R) Core(TM) i5-4310U CPU @ 2.00GHz 2.60 GHz
- RAM: 8,00 Go
- System d'exploitation: Windows 10 Professionnel, 64 bits
- Carte graphique: Intel(R) HD Graphics Family

3.2.2 Environnement de développement

Notre choix s'est porté sur le langage généraliste Python, la bibliothèque de vision par ordinateur Opencv, Tensorflow de google et le Framework Keras pour la mise en oeuvre de notre réseau de neurones, ainsi que d'autres modules et bibliothèques (Numpy, Matplotlib, etc...), en plus du notebook Jupyter pour l'édition et test de notre code



3.2.2.1 Python

Python est un langage portable, dynamique, extensible, gratuit, qui permet une approche modulaire et orientée objet de la programmation. Python est développé depuis 1989 par Guido van Rossum et de nombreux contributeurs bénévoles. Ce langage de programmation présente de nombreuses caractéristiques intéressantes :

- Gratuit et Multiplateforme (Portabilité): Windows, Mac OS X, Linux, Android, iOS, depuis les mini-ordinateurs Raspberry Pi jusqu'aux supercalculateurs.
- Langage de haut niveau: Il demande relativement peu de connaissance sur le fonctionnement d'un ordinateur pour être utiliser.
- Langage interprété : Un script Python n'a pas besoin d'être compiler pour être exécuté, contrairement à d'autres langages
- Dynamique, Extensible: Il favorise la programmation structurée fonctionnelle et orientée objet.
- Syntaxe simple combinée à des types de données évolués (listes, dictionnaires,) qui conduit à des programmes à la fois très compacts et très lisibles.

- Gestion automatique ses ressources sans intervention humaine (mémoire, descripteurs de fichiers...).

- Très populaire en bio-informatique et l'intelligence artificielle et plus généralement en analyse de données.

Toutes ces caractéristiques font que Python est un outil idéal pour implémenter notre application

3.2.2.2 *Jupyter*

Le notebook Jupyter s'est vite imposé comme un outil incontournable dans les milieux académique et scientifique, il permet facilement de travailler sur du code au stade du développement.

L'interface de Jupyter est basée sur une application web, à laquelle on accède via à un navigateur web. Une application web offre beaucoup de flexibilité en termes de déploiement, vous pouvez l'exécuter en local ou bien la déployer sur un serveur distant.

Les points forts de Jupyter sont :

- Jupyter supporte actuellement 40 langages de programmation dont Python, il suffit d'installer le kernel d'exécution correspondant à ce langage et le tour est joué.

- Design et ergonomie irréprochable se présentant comme un document composé d'un enchaînement de blocs, chaque bloc étant constitué de deux parties :

Une case modifiable dans laquelle vous écrivez votre code ;

Sous cette case se trouve une zone de résultat où s'affiche l'output généré par l'exécution du code.

Plus besoin de jongler entre l'éditeur de texte et la console d'exécution, le code et les résultats sont au même endroit.

- La facilité de fragmenter son code pour n'en exécuter qu'une partie précise sur laquelle on souhaite faire des ajustements ou des essais.

3.2.3 Bibliothèques utilisées

3.2.3.1 *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].

- 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. . .)

3.2.3.2 *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.

3.2.3.3 *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 a l'aide de kits d'outils d'interface graphique a usage général tels que Tkinter , wxPython , Qt ou GTK+ .

3.2.3.4 *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 caractéristiques 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 du au backend en C/C++
- Supporte les calculs sur CPU, GPU et même le calcul distribue sur cluster
- Une documentation extrêmement bien fournie avec de nombreux exemples et tutoriels

3.3 La Base De Données

Afin d'augmenter la précision et améliorer la performance de notre modèle, on a choisi d'entraîner le réseau avec un large choix d'images.

Pour l'entraînement et le test, la base de données qu'on a utilisée est disponible ici [19]:

La base de données d'images des expressions faciales que nous avons utilisée comprend un total de 28709 images pour l'entraînement et 7178 images pour le test prétraitées en niveaux de gris et redimensionnées en 48x48 pixels et classer dans 7 dossiers correspondant aux 7 classes d'émotions :

CHAPITRE 3 : IMPLEMENTATION ET REALISATION

Table 3-1: Nombre d'images par catégorie

Catégorie	Entrainement	Test
Colère	3 995	958
Dégout	436	111
Peur	4 097	1 024
Joie	7 215	1 774
Neutre	4 965	1 233
Tristesse	4 830	1 247
Surprise	3 171	831
Total	28 709	7 178

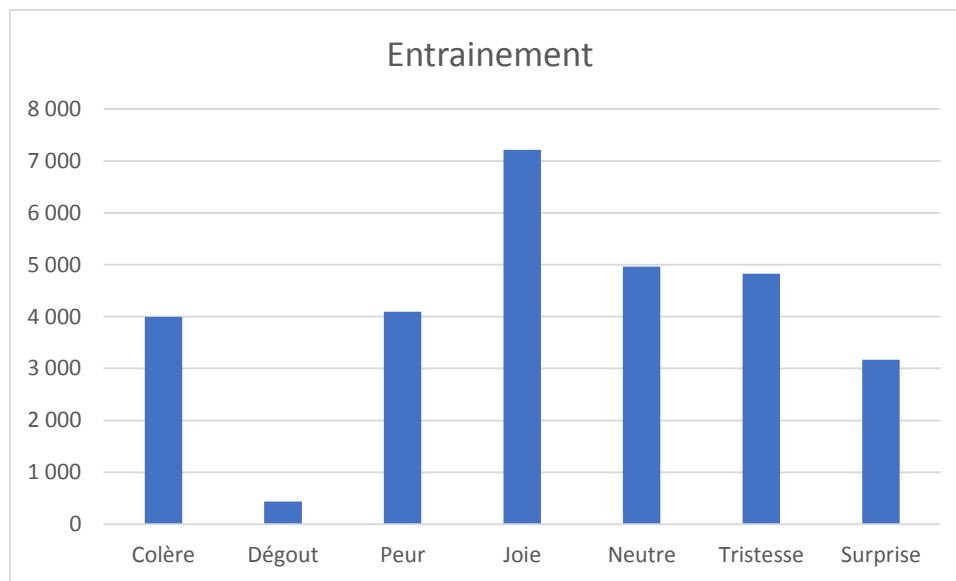


Figure 3-1: Nombre d'images disponibles pour l'apprentissage

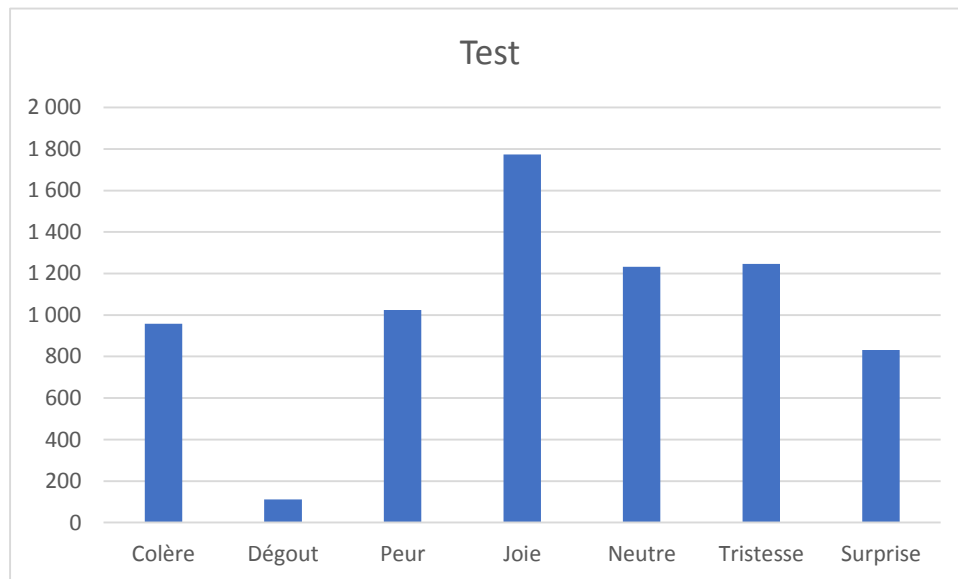


Figure 3-2 : Nombre d'images disponibles pour le test



Figure 3-3 : Echantillon des images de la base de données [19]

3.4 Implémentation et Réalisation

Etape 1 :

Importation des bibliothèques python requises comme numpy, seaborn, matplotlib, tensorflow.

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import utils
import os
%matplotlib inline

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Dense, Input, Dropout, Flatten, Conv2D
from tensorflow.keras.layers import BatchNormalization, Activation, MaxPooling2D
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
from tensorflow.keras.utils import plot_model

import plotly.graph_objects as go
from plotly.subplots import make_subplots

from IPython.display import SVG, Image
from livelossplot.tf_keras import PlotLossesCallback
import tensorflow as tf
print("Tensorflow version:", tf.__version__)
```

Dans le code ci-dessus, nous avons importé différents modules de keras qui est une enveloppe de tensorflow. Nous avons importé en particulier les modules de keras qui nous aideront à construire notre modèle CNN. A la dernière ligne du code ci-dessus, nous avons imprimé la version actuelle de tensorflow dans le système.

Etape 2 :

Nous allons obtenir quelques informations sur le dossier train et test de la base de données sur laquelle notre modèle sera entraîné et validé.

```
for expression in os.listdir("train/"):
    print(str(len(os.listdir("train/" + expression))) + " " + expression + " images")
```

```
3995 angry images
436 disgust images
4097 fear images
7215 happy images
4965 neutral images
4830 sad images
3171 surprise images
```

Le code ci-dessus utilise la bibliothèque `os` pour obtenir le chemin du dossier du train et après cela, nous imprimons le nombre total et la catégorie de chaque dossier d'expression.

Etape 3 :

Nous allons maintenant générer des groupes de formation (train) et de test (validation) afin que notre modèle puisse être formé et validé sur les données de test. Il s'agit d'une étape très importante, car sans cela, nous ne pouvons pas avoir un modèle précis et, sans entraînement, le modèle ne sait pas ce qu'il doit rechercher et prédire.

```
img_size = 48
batch_size = 64

datagen_train = ImageDataGenerator(horizontal_flip=True)

train_generator = datagen_train.flow_from_directory("train/",
                                                    target_size=(img_size,img_size),
                                                    color_mode="grayscale",
                                                    batch_size=batch_size,
                                                    class_mode='categorical',
                                                    shuffle=True)

datagen_validation = ImageDataGenerator(horizontal_flip=True)
validation_generator = datagen_validation.flow_from_directory("test/",
                                                             target_size=(img_size,img_size),
                                                             color_mode="grayscale",
                                                             batch_size=batch_size,
                                                             class_mode='categorical',
                                                             shuffle=False)
```

```
Found 28709 images belonging to 7 classes.
Found 7178 images belonging to 7 classes.
```

Dans le code ci-dessus, nous avons défini la taille de l'image à 48, donc chaque image sera réduite à une taille de 48x48. Ensuite, nous avons défini la taille du lot égale à 64, ce qui signifie qu'à chaque époque, c'est-à-dire lorsque le modèle est exécuté à travers le jeu de données d'entraînement à chaque cycle, le nombre d'images qui seront passées

sera de 64. Cela signifie que le modèle prendra les 64 premières images pour l'entraînement lors de la première époque et continuera ainsi jusqu'à ce que toutes les époques soient terminées.

Après cela, nous avons utilisé ImageDataGenerator du module keras qui génère des lots de données d'images tensorielles avec une augmentation des données en temps réel. Dans ce cas, nous avons gardé l'inversion horizontale vraie, ce qui signifie qu'il va inverser les images d'entrée de manière aléatoire et horizontale.

Nous allons maintenant effectuer l'étape principale et la plus importante, c'est-à-dire générer des images de données d'entraînement et de test.

Tout d'abord, nous allons choisir les données de formation. Ici, nous avons utilisé une fonction de `datagen_train` qui provient de ImageDataGenerator, c'est-à-dire, `flow_from_directory` qui prend quelques paramètres comme le chemin du jeu de données, `target_size` (taille de l'image de sortie), `color_mode` (couleur des images de sortie, nous avons défini `ayscale` qui donne des images grises), `batch_size`, `class_mode` (détermine le type de tableaux d'étiquettes qui sont retournés, nous avons spécifié `categorical`) et nous avons `shuffle False`.

Maintenant, nous allons faire la même chose pour générer les images de l'ensemble de données de test. Le seul changement sera le chemin du dossier de test.

Etape 4 :

Après avoir importé les bibliothèques, obtenu le jeu de données, et créé les images de formation et de test. Nous allons à présent passer à une autre étape importante qui est la construction de notre modèle CNN.

Notre modèle contient quatre couches de convolution, une couche `flatten` et deux couches entièrement connectées

On utilise la fonction `softmax` dans la dernière couche de notre modèle pour prédire la probabilité d'appartenance à l'une des sept classes

```

# Initialising the CNN
model = Sequential()

# 1 - Convolution
model.add(Conv2D(64,(3,3), padding='same', input_shape=(48, 48,1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

# 2nd Convolution Layer
model.add(Conv2D(128,(5,5), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

# 3rd Convolution Layer
model.add(Conv2D(512,(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

# 4th Convolution Layer
model.add(Conv2D(512,(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

# Flattening
model.add(Flatten())

# Fully connected Layer 1st Layer
model.add(Dense(256))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

# Fully connected Layer 2nd Layer
model.add(Dense(512))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Dense(7, activation='softmax'))

opt = Adam(lr=0.0005)
model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

```

neuronal.

Tout d'abord, nous avons initialisé le modèle CNN en utilisant la fonction `sequential()`. Après cela, nous avons créé les quatre premières couches du réseau de neurones qui sont des couches de convolution.

La couche de convolution est ajoutée à notre modèle en utilisant `model.add()`. Cette méthode prend quelques paramètres comme `Conv2D` qui spécifie que la couche à ajouter

est une couche de convolution. Cette méthode prend également deux paramètres, le premier étant le nombre de filtres et le pooling. Après cela, nous appliquons la normalisation de lot qui applique une transformation qui maintient la moyenne de sortie proche de 0 et la déviation standard de sortie proche de 1. Après cela, on ajoute une fonction d'activation qui effectue une transformation non linéaire sur le signal d'entrée. Cette sortie transformée est ensuite envoyée à la couche suivante de neurones comme entrée. Après cela, nous avons appliqué MaxPolling2D qui ne retient que le maximum d'un pool. Et enfin, nous ajoutons le dropout qui empêche le modèle de s'adapter excessivement. Et ici nous terminons la construction d'une couche de convolution.

Après cela, nous avons ajouté une couche d'aplatissement qui convertit les données en un tableau unidimensionnel pour l'entrée dans la couche suivante. Nous aplatissons la sortie des couches convolutionnelles pour créer un seul vecteur de caractéristiques. Et il est connecté au modèle de classification final, qui est appelé une couche entièrement connectée.

Enfin, nous avons ajouté deux couches entièrement connectées au modèle.

Pour toutes les couches jusqu'à présent, nous avons conservé la fonction d'activation ReLU car ReLU est à moitié rectifié (à partir du bas). $f(z)$ est égal à zéro lorsque z est inférieur à zéro et $f(z)$ est égal à z lorsque z est supérieur ou égal à zéro.

Mais après toutes les couches ci-dessus, nous allons finalement ajouter une couche dense dont l'activation est définie comme Softmax, qui transforme les nombres, en probabilités dont la somme est égale à un. La fonction Softmax produit un vecteur qui représente les distributions de probabilité d'une liste de résultats potentiels.

Ainsi, dans la sortie du modèle ci-dessus, nous obtenons finalement des probabilités comprises entre 0 et 1, ce qui nous permet de classer facilement l'expression.

Ensuite, nous avons utilisé `model.compile` pour compiler le modèle.

Étape 5 :

A partir de maintenant, nous avons compilé notre modèle. Maintenant, nous allons visualiser l'architecture du modèle en utilisant le code donné ci-dessous.

```
plot_model(model, to_file='model.png', show_shapes=True, show_layer_names=True)
Image('model.png',width=400, height=200)
```

Dans le code ci-dessus, nous avons utilisé `plot_model` qui est également une fonction de la bibliothèque `keras`. Nous lui avons donné les mêmes paramètres comme le nom du modèle, le nom du fichier, c'est-à-dire, `to_file`, `show_shapes` qui affiche les informations de forme et `show_layer_names` : pour afficher les noms des couches.

Ensuite, nous avons utilisé la fonction `Image` de la bibliothèque `lpython` pour afficher l'architecture du modèle dans la sortie. Elle prend quelques arguments comme le nom du fichier, la largeur et la hauteur de l'image à afficher. La sortie de cette fonction (l'image) est montrée ci-dessous.

Etape 6 :

Maintenant que nous avons construit avec succès l'architecture du modèle, il est temps d'entraîner le modèle et d'évaluer les résultats.

```
%%time

epochs = 15
steps_per_epoch = train_generator.n//train_generator.batch_size
validation_steps = validation_generator.n//validation_generator.batch_size

reduce_lr = ReduceLRonPlateau(monitor='val_loss', factor=0.1,
                              patience=2, min_lr=0.00001, mode='auto')
checkpoint = ModelCheckpoint("model_weights.h5", monitor='val_accuracy',
                             save_weights_only=True, mode='max', verbose=1)
callbacks = [PlotLossesCallback(), checkpoint, reduce_lr]

history = model.fit(
    x=train_generator,
    steps_per_epoch=steps_per_epoch,
    epochs=epochs,
    validation_data = validation_generator,
    validation_steps = validation_steps,
    callbacks=callbacks
)
```

On entraîne le modèle avec la fonction `fit()` .

On choisit ces paramètres pour l'entraînement :

- Le chemin du fichier contenant les données d'entraînement
- Le nombre d'epoch, le nombre de fois que le modèle repasse sur l'ensemble de données
- `steps_per_epoch`, est en fait similaire a `batch_size`, ici 448 `steps_per_epoch` correspond à un `batch_size` de 64 (nombre de données d'entraînement divisé par nombre de `steps_per_epoch`)
- Le chemin du fichier contenant les données de validation (pour évaluer le Surapprentissage (overfitting) du modèle)
- `validation_steps`, c'est le nombre de `steps_per_epoch` après lequel le modèle se test sur les données de validation

On peut résumer ces paramètres ainsi : le modèle s'entraîne 15 fois sur l'ensemble de données. Chaque fois qu'il s'entraîne, les données sont divisées en 64 lots (batch) de 448 images sur lesquels il se focalise un à un et il teste son apprentissage sur l'ensemble des données de validation.

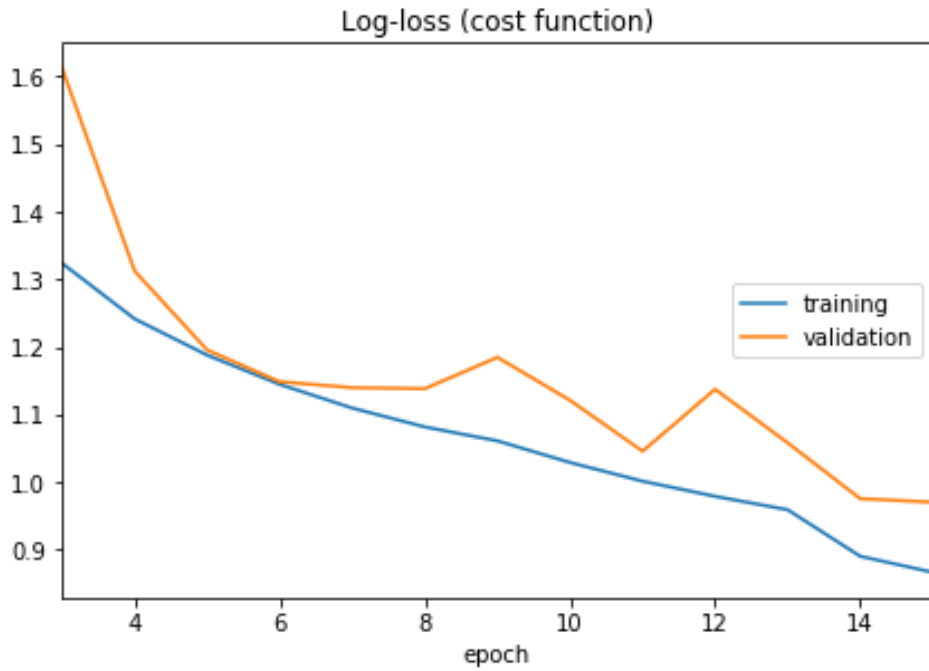


Figure 3-4: La courbe de perte générée pour notre modèle

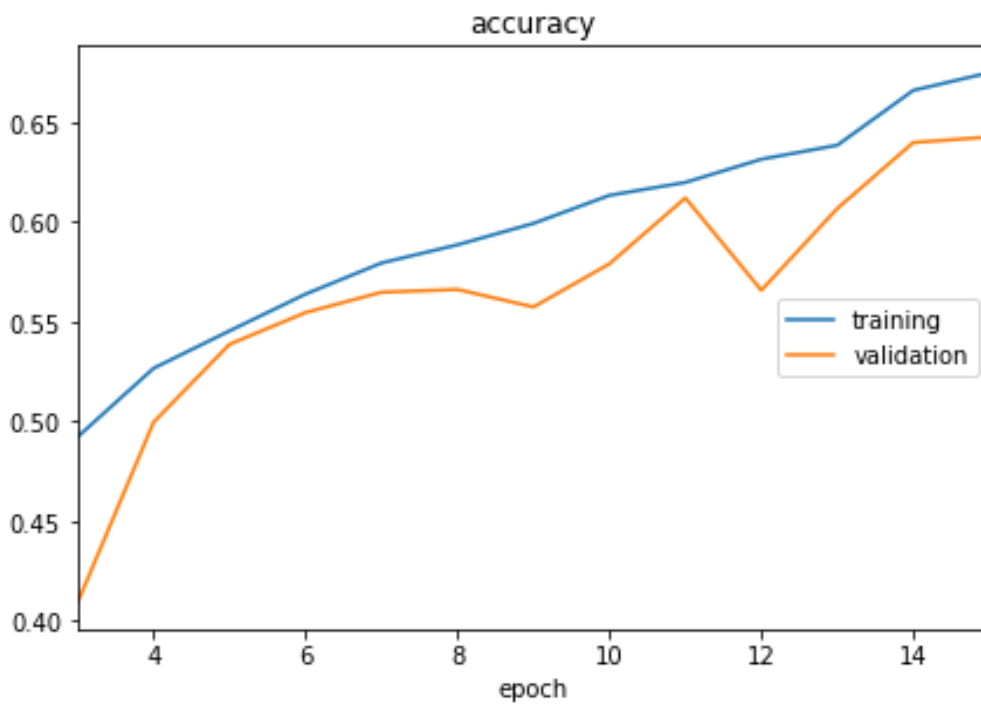


Figure 3-5: La courbe de précision générée pour notre modèle

training (min: 0.866, max: 1.786, cur: 0.866)

validation (min: 0.970, max: 1.705, cur: 0.970)

accuracy:
training (min: 0.313, max: 0.675, cur: 0.675)
validation (min: 0.381, max: 0.643, cur: 0.643)

Epoch 00015: saving model to model_weights.h5
448/448 [=====] - 27s 60ms/step - loss: 0.8659 - accuracy: 0.6748 - val_loss:
0.9700 - val_accuracy: 0.6426
CPU times: user 6min 50s, sys: 57.4 s, total: 7min 47s
Wall time: 6min 46s

Nous avons obtenu une précision de près de 68% sur le jeu de données en exécutant seulement 15 époques.

3.5 Évaluation de notre modèle

Nous allons évaluer dans cette partie la robustesse de notre modèle en analysant les résultats obtenus sur un échantillon de 35 personnes de sexe, âge et couleur de peau différentes.

Age	Sexe	
	Masculin	Féminin
Plus de 65 ans	2	3
Entre 20 et 50 ans	15	8
Moins de 10 ans	2	5

Figure 3-6: Echantillon de personnes testées par sexe et âge

Figure 3-7: Quelques résultats de test obtenus sur des cas réels

Le modèle a montré d'excellentes performances pour la prédiction des expressions de : Joie, Surprise, Neutralité, Tristesse allant de 83% à 98%.

En revanche, le modèle semble moins performant pour les expressions de colère, dégoût et peur avec des faibles taux de précisions entre 38% et 53%.

3.6 Conclusion

Dans ce dernier chapitre nous avons présenté un modèle pour la classification des expressions faciales en se basant sur les techniques développées dans le domaine du Deep Learning et en proposant notre architecture de réseau de neurones convolutif (CNN). Nous avons montré que notre modèle est globalement efficace mais qui peut s'améliorer encore plus, notamment en augmentant le nombre d'époques et en l'entraînant sur une base de données plus large.

L'objectif de ce mémoire est de concevoir et d'implémenter une application de détection des expressions faciales et vu la diversité des domaines où l'on peut l'utiliser (Sécurité, Médecine, Marketing, IHM, ...) cette application doit répondre à des exigences de rapidité et de robustesse des résultats.

Notre système utilise un classifieur d'expressions faciales créée à l'aide du Deep Learning avec une architecture CNN et un taux de précision avoisinant 68%

Ce travail nous a permis d'acquérir de nouvelles connaissances et d'approfondir d'autres, que ce soit dans l'aspect théorique : les expressions faciales, techniques et algorithmes utilisés, Deep Learning ou l'aspect pratique : Python, les diverses bibliothèques dédiées au traitement d'images et la vision artificielle, Jupyter Notebook

Parmi les perspectives ouvertes de ce projet :

- L'utilisation d'autres méthodes de reconnaissance, et la combinaison avec d'autres pour la conception d'un système hybride.
- Fiabiliser le système en diminuant la sensibilité aux contraintes d'éclairage, de pose et d'occultation par l'utilisation de nouvelles techniques de normalisation.
- Accroître le taux de précision en entraînant le modèle sur des bases de données plus volumineuses et en augmentant le nombre d'époques.
- Étendre notre système pour l'acquisition des images 3D et inclure plus de classes d'expressions.

-
- Bibliographie

- [1] G.-B. Duchenne, *Mécanisme de la physionomie humaine, ou Analyse électro-physiologique de l'expression des passions*, Paris: Veuve J. Renouard, 1862., 1862.
- [2] C. Darwin, *L'expression des émotions chez l'homme et les animaux*, sn, 1877.
- [3] P. Ekman et W. . V. Friesen, *Unmasking the Face: A Guide to Recognizing Emotions from Facial Expressions*, Prentice Hall, 1975.
- [4] M. NAVRAAN, C. N. MOGHADAM et M. MANSOORIZADEH, «AUTOMATIC FACIAL EMOTION RECOGNITION METHOD BASED ON EYE REGION CHANGES,» *JOURNAL OF INFORMATION SYSTEMS AND TELECOMMUNICATION (JIST)*, vol. 4, pp. 221-231, 2016.
- [5] S. Shiguang, Y. Peng, C. Xilin et G. Wen, «AdaBoost Gabor Fisher Classifier for Face Recognition,» chez *International Workshop on Analysis and Modeling of Faces and Gestures*, 2005.
- [6] B. Simon, *Forêts Aléatoires : De l'Analyse des Mécanismes de Fonctionnement à la*, PhD thesis, 2009.
- [7] T. Reiman, chez *The Power of Body Language*, Simon and Schuster, 2008, p. 44.
- [8] B. F. Lisa, A. Ralph, M. Stacy, M. M. Aleix et D. P. Seth, «Emotional Expressions Reconsidered: Challenges to Inferring Emotion From Human Facial Movements,» *Psychol Sci Public Interest*, pp. 1-68, Juillet 2019.
- [9] M.-H. Yang, D. J. Kriegman et N. Ahuja, «Detecting Faces in Images: A Survey,» *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 1, JANUARY 2002*.
- [10] Y. Ming-Hsuan, K. David J. et A. Narendra, «Detecting Faces in Images: A Survey,» *Transactions on Pattern Analysis and Machine Intelligence*, p. 34–58, January 2002.
- [11] S. Sallay, «Human Emotions Recognition through Facial Expressions and Sentiment Analysis for Emotionally Aware Deep Learning Models,» 31 Mai 2021. [En ligne]. Available: <https://medium.com/geekculture/human-emotions-recognition-through-facial-expressions-and-sentiment-analysis-for-emotionally-aware-4a41d7f8b80e>. [Accès le Septembre 2021].
- [12] K. Takeo, *Picture processing system by computer complex and recognition of human*, 1974.
- [13] E. Paul, F. . V. Wallace et H. C. Joseph, *Facial Action Coding System*, United States of America: Research Nexus division of Network Information Research Corporation, 2002.
- [14] «Facial action coding system,» Wikimedia Foundation, Inc, 17 Juin 2021. [En ligne]. Available: https://fr.wikipedia.org/wiki/Facial_action_coding_system. [Accès le 10 2022].

- [15] G. Khadoudja, Reconnaissance des expressions faciales à base d'informations video: estimation de l'intensité des expressions faciales, 2010, p. 681–685.
- [16] A. Jörgen, Candide-3-an updated parameterised face., 2001.
- [17] C. K. Byoung, «A brief review of facial emotion recognition based on visual information.,» *sensors*, 18(2), p. 401, 2018.
- [18] [En ligne]. Available: <https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/an-executives-guide-to-ai>. [Accès le Octobre 2022].
- [19] «Base de données des images pour apprentissage et test de Kaggle,» [En ligne]. Available: <https://www.kaggle.com/aadityasinghal/facial-expression-dataset>. [Accès le Juin 2021].
- [20] [En ligne]. Available: <https://answersingenesis.org/human-body/facial-expressions-universal-language/>.
- [21] D. S. e. R. Foulds, chez *Toward robust skin identification in video images, Proceeding of international conference on automatic face and gesture recognition*, 1996.
- [22] P. Viola et M. J. Jones , «Robust Real-Time Face Detection,» *International Journal of Computer Vision*, p. 137–154, 2004.
- [23] «Training support vector machines: an application to face detection,» *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 06 August 2002.
- [24] Y. Guangzheng et T. S. Huang, «Human Face Detection in Complex,» *Pattern Recognition*, pp. 53-63, 1994.
- [25] Z. Ruicong, L. Mengyi et Z. Dezheng, «A comprehensive survey on automatic facial action unit analysis,» *The Visual Computer* 36(1), Mai 2020.
- [26] «Keras Tutorial: Deep Learning in Python,» [En ligne]. Available: <https://www.datacamp.com/community/tutorials/deep-learning-python>. [Accès le Octobre 2022].
- [27] B. Facundo et G. M. Juan, «Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks,» *Energy and Buildings* 158, Novembre 2017.
- [28] H. Gu, Y. Wang, S. Hong et G. Gui, «Blind Channel Identification Aided Generalized Automatic Modulation Recognition Based on Deep Learning,» *IEEE Access*, p. 99, Aout 2019.
- [29] «Réseau de neurones – Deep Learning : Biologiques ou artificiels, même combat ?,» Septembre 2022. [En ligne]. Available: <https://datascientest.com/deep-learning-reseau-de-neurones-biologiques-ou-artificiels>.