

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université Akli Mohand Oulhadj - Bouira -
Tasdawit Akli Muḥend Ulḥağ - Tubirett -



وزارة التعليم العالي والبحث العلمي
جامعة أكلي محمد أولحاج
- البويرة -

Faculté des Sciences et des Sciences Appliquées

كلية العلوم والعلوم التطبيقية

Référence :/MM/ 2023

المرجع :/م/م / 2023

Mémoire de Master

Présenté au

Département : Génie Électrique

Domaine : Sciences et Technologies

Filière : Electronique

Spécialité : Electronique des systèmes embarqués

Réalisé par :

HAKOUM Smail

Et

SKANDER Mustapha

Thème

**Utilisation de Deep Learning pour la reconnaissance et
classification des émotions faciales**

Soutenu le : **03/07/2023**

Devant le Jury composé de :

Mr :	REZKI Mohamed	M.C.A	Univ. Bouira	Président
Mr :	BENSAFIA Yassine	M.C.A	Univ. Bouira	Rapporteur
Mr :	TOUAFEK Mohamed Yaakoub	M.A.A	Univ. Bouira	Examineur

Année Universitaire : 2022-2023

Remerciements



Tout d'abord, nous tenons à remercier sincèrement Allah, le Tout-Puissant, le Miséricordieux, le Généreux, de nous avoir donné le courage de poursuivre et d'achever ce travail.

Nous tenons à exprimer notre profonde gratitude aux personnes qui nous ont soutenus tout au long de la préparation de ce projet

*Nous sommes reconnaissants envers notre encadrant **M. BENSAFIA Yassine**, nous avons l'honneur de bénéficier de votre riche enseignement, merci pour votre soutien, votre aide, votre orientation et vos conseils si précieux.*

*Nous souhaitons exprimer notre sincère gratitude et nos remerciements au **membre du jury**. Veuillez accepter notre respect le plus sincère et notre profonde reconnaissance pour votre contribution à ce travail.*

*Chaleureux remerciement à **nos familles** qui nous ont soutenus moralement, ainsi que **nos parents** qui nous ont encouragé et aidé à arriver à ce stade de notre formation.*

A handwritten signature in black ink, appearing to read 'Thank you', with a small heart symbol above the 'y'. The signature is written in a cursive, flowing style.

Dédicace

Je dédie ce travail à mes chers parents qui m'ont aimé et m'ont soutenu toute ma vie, je ne peux tout simplement pas exprimer ma gratitude avec des mots, que Dieu vous protège.

Je le dédie aussi : à ma famille, mes amis, en enfin et surtout mon binôme et toute sa famille.

Merci pour votre confiance et votre soutien tout au long de mon parcours, que Dieu vous bénisse tous.

Mustapha

”There is Always Something to Be Grateful For”

Je dédie ce travail à mes chers parents qui m'ont aimé et m'ont soutenu toute ma vie, je ne peux tout simplement pas exprimer ma gratitude avec des mots, que Dieu vous protège.

Je le dédie aussi : à ma famille, mes sœurs, mes deux belles nièces, mes amis, et surtout mon binôme et toute sa famille.

Merci pour votre confiance et votre soutien tout au long de mon parcours, que Dieu vous bénisse tous.

Smail

Résumé — Les expressions faciales sont l'un des moyens les plus efficaces et les plus naturels dont dispose l'être humain pour exprimer ses émotions et ses intentions. La reconnaissance des émotions est une tâche importante dans l'interaction homme-machine, où l'objectif est d'identifier et de comprendre les états émotionnels des individus à travers diverses modalités telles que les expressions faciales. L'objectif de cette étude est d'utiliser l'apprentissage profond, en particulier les réseaux de neurones convolutifs. Les CNN sont généralement utilisés pour les modalités basées sur l'image, telles que la reconnaissance des expressions faciales. Pour entraîner un réseau neuronal à la reconnaissance des émotions, il faut disposer d'un ensemble de données d'expressions émotionnelles étiquetées, en utilisant la base de données FER2013, Elle est utilisée pour entraîner le réseau à apprendre les caractéristiques et les modèles associés aux différentes émotions. Une fois le réseau formé, il peut être utilisé pour reconnaître des émotions dans de nouvelles données non vues. Dans l'ensemble, les approches basées sur l'apprentissage profond ont montré des résultats prometteurs dans la reconnaissance des émotions et ont le potentiel d'améliorer diverses applications telles que l'interaction homme-robot et la surveillance de la santé mentale.

Mots clés : émotions, expressions faciale, deep learning, réseaux de neurones, CNN, FER2013

Abstract — Facial expressions are one of the most effective and natural ways for humans to express their emotions and intentions. Emotion recognition is an important task in human-computer interaction, where the goal is to identify and understand the emotional states of individuals through various modalities such as facial expressions. The focus of this study is to use Deep Learning, specifically convolutional neural networks. CNN are typically used for image-based modalities such as facial expression recognition. Training a neural network for emotion recognition requires a dataset of labeled emotional expressions, using the FER2013 database. It is used to train the network to learn the features and patterns associated with different emotions. Once the network is trained, it can be used to recognize emotions in new, unseen data. Overall, deep learning-based approaches have shown promising results in emotion recognition and have the potential to improve various applications such as human-robot interaction, and mental health monitoring.

Keywords : emotions, facial expressions, deep learning, neural networks, CNN, FER2013

ملخص — تعابير الوجه هي واحدة من أكثر الطرق فعالية وطبيعية لدى البشر للتعبير عن مشاعرهم ونواياهم. يعد التعرف على المشاعر مهمة هامة في التفاعل بين الإنسان والحاسوب، حيث يتمثل الهدف في تحديد وفهم الحالات العاطفية للأفراد من خلال أساليب مختلفة مثل تعابير الوجه. الهدف من هذه الدراسة هو استخدام التعلم العميق، ولا سيما الشبكات العصبية الالتفافية. تُستخدم شبكات CNN عادةً الطرق القائمة على الصور، مثل التعرف على تعبيرات الوجه. لتدريب شبكة عصبية على التعرف على المشاعر، يحتاج المرء إلى مجموعة بيانات من التعبيرات العاطفية المصنفة، نستخدم مجموعة البيانات FER2013 الأساسية، والتي تُستخدم لتدريب الشبكة على تعلم الخصائص والأنماط مرتبطة بمشاعر مختلفة. بمجرد تدريب الشبكة، يمكن استخدامها للتعرف على المشاعر في البيانات الجديدة غير المرئية. عمومًا، أظهرت النهج المبنية على التعلم العميق نتائج واعدة في التعرف على العواطف، ولديها الإمكانية لتحسين التطبيقات المختلفة مثل التفاعل الإنسان مع الآلة، ومراقبة الصحة العقلية.

كلمات مفتاحية : مشاعر، تعابير الوجه، الشبكات العصبية، CNN، التعلم العميق، FER2013.

Table des matières

Résumé	iii
Liste des Figures	vii
Liste des Tableaux	viii
Liste des Abréviations	ix
Introduction Générale	1

Chapitre 1 : Notions sur les émotions

1.1	Introduction	3
1.2	Les émotions	3
1.2.1	Définition	3
1.2.2	La reconnaissance des émotions	3
1.2.3	La différence entre expressions faciales et émotions	4
1.3	Les expressions faciales	4
1.3.1	Définition	4
1.3.2	Analyse et description des six expressions faciales	5
1.4	Différentes approches de reconnaissance d'expressions faciales	7
1.4.1	Approches conventionnelles	7
1.4.1.1	Acquisition d'images et extraction du visage	8
1.4.1.2	Extraction des caractéristiques	9
1.4.1.3	Classification	9
1.4.2	Approches basées sur l'apprentissage profond (Deep learning)	10
1.5	Applications pratiques de la reconnaissance des expressions faciales	11
1.6	Conclusion	11

Chapitre 2 : L'Apprentissage Profond (Deep Learning)

2.1	Introduction	12
2.2	L'Apprentissage Profond (Deep Learning)	12

2.2.1	Définition -----	12
2.2.2	La relation entre l'Intelligence Artificielle (IA), Machine Learning (ML) et le Deep Learning (DL)-----	12
2.2.3	Pour quoi l'Apprentissage Profond ?-----	13
2.3	Réseaux de neurones -----	14
2.3.1	Principe des réseaux de neurones -----	14
2.3.2	L'architecture des RNA -----	16
2.3.3	Les différentes Architectures du Deep Learning -----	17
2.3.3.1	Réseaux de neurones convolutifs (CNN).-----	17
2.3.3.2	Réseau de neurones récurrents (RNN) -----	17
2.3.3.3	Modèle génératif-----	18
2.4	Les applications de Deep learning -----	19
2.5	Réseaux de Neurones Convolutifs CNN -----	19
2.5.1	Présentation des CNN-----	20
2.5.2	Architecture de CNN-----	21
2.5.3	Les Différentes Couches de CNN-----	21
2.5.3.1	Couche convolution (CONV)-----	21
2.5.3.2	La couche ReLu – Unité linéaire rectifiée -----	23
2.5.3.3	La couche de Pooling -----	24
2.5.3.4	Couches Entièrement Connectées (Fully connected) -----	25
2.5.4	Quelques réseaux convolutifs pré-entraînés – célèbres -----	25
2.6	Conclusion-----	26
Chapitre 3 : Implémentation et résultats expérimentaux		
3.1	Introduction-----	27
3.2	But du projet-----	27
3.3	Environnement de développement -----	27
3.3.1	Environnement matériel (HARDWARE)-----	27

3.3.2	Environnement logiciel (SOFTWARE)	28
3.4	Outils de développement	28
3.4.1	Visual Studio Code (VS Code)	28
3.4.2	Python	28
3.5	Bibliothèques utilisées	29
3.5.1	OpenCV (Open Source Computer)	29
3.5.2	TensorFlow	29
3.5.3	Keras	30
3.5.4	Numpy	30
3.5.5	Matplotlib	30
3.6	Installation des bibliothèques	31
3.7	Base de données (Database)	31
3.8	Conception du projet	32
3.8.1	Importer des bibliothèques	32
3.8.2	Création du modèle	33
3.8.3	L'entraînement du modèle	34
3.8.4	Résultats expérimentaux et analyse de performances	36
3.9	Tester le modèle pour la reconnaissance d'expressions faciales	37
3.9.1	Utilisation du modèle avec OpenCV et Webcam en temps réel	37
3.9.2	Utilisation du modèle en vidéo en format MP4	39
3.9.3	Utilisation du modèle en image en format PNG	40
3.10	Conclusion	40
	Conclusion et Perspectives	41
	Bibliographie	42

Liste des Figures

Figure 1. 1 - Les différentes expressions faciales. -----	5
Figure 1. 2 - Les six expressions de base. -----	6
Figure 1. 3 - Liste des Actions Units relatives aux 6 expressions faciales [10].-----	6
Figure 1. 4 - Modèle MPEG4 l'ensemble d'attributs faciaux [12]. -----	7
Figure 1. 5 - Structure d'un système de reconnaissance des expressions faciales. -----	8
Figure 2. 1 - La relation entre l'intelligence artificielle (AI), le ML et le DL.-----	13
Figure 2. 2 - Les performances de l'apprentissage profond par rapport à d'autres algorithmes d'apprentissage [22].-----	13
Figure 2. 3 - Comparaison entre apprentissage automatique (ML) et l'apprentissage profond (DL).-----	14
Figure 2. 4 - Comparaison entre Neurone biologique (a) et neurone artificiel (b). -----	15
Figure 2. 5 - Réseau de neurones artificiels (RNA).-----	16
Figure 2. 6 - Quelques topologies standards [24]. -----	16
Figure 2. 7 - Les différents modèles du Deep Learning. -----	19
Figure 2. 8 - Schéma représentant l'architecture d'un CNN [37].-----	21
Figure 2. 9 - Architecture de réseau proposée. -----	21
Figure 2. 10 - Schéma de déplacement de la fenêtre de filtre à travers l'image [37]. -----	22
Figure 2. 11 - Différents types de convolutions. -----	23
Figure 2. 12 - La fonction d'activation ReLU est définie comme suit :-----	24
Figure 2. 13 - Pooling avec un filtre de taille 2x2 et un pas de 2. -----	24
Figure 2. 14 - Architecture de VGG-16.-----	26
Figure 3. 1 - Logo de VS Code. -----	28
Figure 3. 2 - Logo de python. -----	28
Figure 3. 3 - Logo d'OpenCV.-----	29
Figure 3. 4 - Logo de TensorFlow.-----	29
Figure 3. 5 - Logo de Keras. -----	30
Figure 3. 6 - Logo de NumPy.-----	30
Figure 3. 7 - Logo de Matplotlib.-----	30

Figure 3. 8 - Installation de la bibliothèque Keras. -----	31
Figure 3. 9 - Exemples d'images pour la reconnaissance des émotions faciales à partir de quatre ensembles de base de données. -----	32
Figure 3. 10 – Importation des bibliothèques pour entraîner et valider notre modèle. -----	33
Figure 3. 11 - Code pour la création du modèle. -----	33
Figure 3. 12 - Résultat du modèle. -----	34
Figure 3. 13 - Code pour l'entraînement du modèle. -----	35
Figure 3. 14 - Résultat de l'entraînement du modèle. -----	36
Figure 3. 15 - Les courbes de précision (accuracy) et de perte (loss) générée du modèle.-----	37
Figure 3. 16 - La visualisation de nos images qui ont été traitées. -----	38
Figure 3. 17 - Résultats d'une vidéo externe.-----	39
Figure 3. 18 - Résultats de quelques images dans une seule image externe. -----	40

Liste des Tableaux

Tableau 3. 1 - Détails de la base de données utilisée. -----	32
Tableau 3. 2 - Tableau de paramètres utilisées. -----	35
Tableau 3. 3 - Résultats de l'apprentissage. -----	36

Liste des Abréviations

AI	Artificial Intelligence
API	Application Programming Interface
AU	Unité d'Action
BN	Bayesian Network
CK	Cohn-Kanade
CNN	Convolutional Neural Network
CNTK	Microsoft Cognitive Toolkit
CONV	Convolutional
CPU	Central Processing Unit
DL	Deep Learning
EEG	Électroencéphalogramme
FACS	Système de Codage des Actions Faciales
FAP	Facial Animation Parameters
FER2013	Facial Expression Reconition 2013 database
GPU	Graphics Processing Unit
HD	Haute Définition
JAFFE	Japanese Female Facial Expressions
KDEF	Karolinska Directed Emotional Faces
LDA	Linear Discriminant analysis
LSTM	Long-Short-Term-Memory
ML	Machine Learning
MLP	Multi Layer Perceptron
MPEG	Moving Pictures Experts Group

NN	Neural Networks
OS	Operating System
PNG	Portable Network Graphics
RAM	Random Access Memory
REF	Reconnaissance des Emotions Faciales
ReLU	Rectified Linear Units
RNA	Réseau de Neurones Artificiels
RNN	Recurrent Neural Network
ROI	Region of Interest
SVM	Support Vector Machine
VGG	Visual Geometry Group
VGG16	Visual Geometry Group (modèle 16 couches)
VS	Visual Studio
XOR	Exclusive OR

Introduction Générale

Ces dernières années ont été marquées par une révolution scientifique et technologique remarquable, qui a transformé divers domaines, notamment la vision par ordinateur. Cette technologie permet de créer des applications intelligentes capables de comprendre et d'extraire des objets de différents types à partir de données telles que des images ou des vidéos, à partir de bases de données. Au cours de cette décennie, l'informatique affective est devenue une réalité, permettant d'introduire les émotions dans le monde de l'informatique (à l'ordinateur). Les émotions sont des réactions psychologiques et physiologiques à des stimuli externes ou internes, les émotions de base comprennent la surprise, la tristesse, la peur, la colère, le dégoût et la joie jouent un rôle essentiel dans la communication interpersonnelle et la régulation du comportement chez les êtres humains. Elles sont indispensables pour exprimer nos sentiments, interagir avec les autres et adapter notre comportement aux situations que nous rencontrons.

La reconnaissance des émotions est une approche qui utilise l'analyse des expressions faciales pour détecter visuellement l'état émotionnel ou mental d'une personne. C'est une application importante dans de nombreux domaines tels que l'apprentissage assisté par ordinateur, la médecine, le traitement d'images, la sécurité, et l'interaction homme-machine. Cependant, la variabilité des expressions faciales entre les individus rend cette tâche difficile à réaliser. Cette méthode est un sujet de recherche actif qui remonte à 1872, date à laquelle Darwin a commencé à s'intéresser à cette question. Dans les années 1970, Suwa et Motoi ont proposé pour la première fois l'analyse automatique des expressions faciales (REF) à partir d'images en traçant le mouvement de 20 points identifiés sur une séquence d'images. Cette méthode a connu une croissance rapide ces dernières années grâce à l'utilisation de techniques de deep learning.

Le deep learning est une méthode de traitement de données qui permet de reconnaître des modèles complexes dans les données en utilisant des réseaux de neurones artificiels profonds, en particulier les réseaux de neurones convolutifs (CNN), ont été spécialement développés pour résoudre les problèmes rencontrés en machine learning. Le CNN est l'une des architectures de réseau les plus représentatives de la technologie d'apprentissage profond (deep learning), et a

connu un grand succès dans le domaine du traitement et de la reconnaissance d'images.

Notre intérêt de recherche a porté sur l'étude et la création d'un modèle capable de reconnaître les émotions faciales de base à partir d'images en utilisant un réseau de neurones convolutifs CNN avec la base des données FER2013.

La structure de ce mémoire se compose de trois chapitres :

- Dans *le chapitre 1*, nous abordons les notions sur les émotions faciales, et une analyse et description détaillées des six expressions faciales fondamentales. De plus, nous examinons les systèmes de codification utilisés, ainsi, que les approches utilisées dans les systèmes de reconnaissance des expressions faciales.
- Ensuite *le chapitre 2*, nous présentons un aperçu sur le Deep Learning, où nous donnerons plus de détails sur les réseaux de neurones convolutifs.
- Le dernier chapitre, *le chapitre 3*, illustre l'implémentation et l'expérimentation de notre approche de reconnaissance des émotions faciales basée sur les CNN en mettant en œuvre notre logiciel de programmation et les bibliothèques nécessaires pour l'implémentation de notre programme en python, et les différents résultats obtenus.

Enfin, nous concluons par une conclusion générale des résultats, ainsi que quelques perspectives pour de futures recherches dans ce domaine.

Chapitre 1

Notions sur les émotions

1.1 Introduction

La reconnaissance des émotions est un domaine passionnant de l'étude de la psychologie et des sciences cognitives. Les émotions occupent une position centrale dans notre vie quotidienne, affecte nos pensées, nos actions et nos interactions sociales. Leur impact peut être significatif dans divers domaines, tels que la psychologie, la santé mentale, l'éducation, la communication interpersonnelle, l'interaction homme-machine, et bien plus encore. La plupart des systèmes d'analyse des expressions faciales cherchent à regrouper les expressions dans quelques grandes catégories émotionnelles, telles que la surprise, la colère, la joie, le dégoût, la peur, la tristesse.

Dans ce chapitre, nous permettra d'explorer les différents aspects de la reconnaissance des émotions faciales, depuis la définition des émotions jusqu'aux méthodes utilisées pour les mesurer et les détecter.

1.2 Les émotions

1.2.1 Définition

D'après le psychologue James A. Russell, les émotions sont des états affectifs complexes qui englobent des changements dans les processus physiologiques, psychologiques (mental) et comportementaux. Ces changements reflètent la manière dont les individus perçoivent, évaluent et réagissent à leur environnement [1].

Les émotions peuvent être considérées comme des réponses adaptatives à des conditions particulières qui peuvent être déclenchées par des facteurs internes ou externes. Elles sont souvent liées à des expressions faciales, des mouvements et des réponses physiologiques telles que l'accélération du rythme cardiaque, les tremblements, la transpiration, etc. Les émotions positives comme la joie, l'amour et la gratitude, ainsi que les émotions négatives comme la peur, la colère et la tristesse, peuvent avoir des effets à court ou à long terme sur un individu [1].

Il existe de nombreuses façons de catégoriser les émotions, les trois types les plus courantes utilisant : émotions primaires, émotions secondaires et émotions sociales [2].

1.2.2 La reconnaissance des émotions

Les systèmes de reconnaissance d'émotions sont des technologies qui permettent de détecter et d'analyser les émotions des personnes à partir de diverses sources de données.

Trois systèmes de reconnaissance des émotions ont été développés [3].

- La reconnaissance des émotions à partir de la parole (Reconnaissance vocale) : cette technique permet de détecter les émotions à partir de la voix. Les algorithmes de reconnaissance vocale peuvent être employés pour détecter les émotions en analysant des caractéristiques telles que le volume de voix, le rythme, la tonalité.
- La reconnaissance des émotions à partir de EEG (signaux du système nerveux).
- Reconnaissance faciale : cette technique permet de détecter les expressions faciales des individus à partir de caméras vidéo ou de photos. Les algorithmes de reconnaissance faciale peuvent être utilisés pour détecter les différentes émotions telles que le dégoût, joie, la tristesse, la colère et la peur.

Ces systèmes utilisent souvent des algorithmes d'apprentissage automatique pour apprendre à reconnaître les émotions à partir de données d'entraînement et pour prendre des décisions en temps réel. Nous nous intéressons à la reconnaissance faciale.

1.2.3 La différence entre expressions faciales et émotions

Une étude publiée dans la revue "Journal of Personality and Social Psychology" en 1991 a exploré la relation entre les expressions faciales et les émotions ressenties. Les chercheurs ont constaté que les expressions faciales peuvent avoir un effet sur l'émotion ressentie [4]. Les expressions faciales et les émotions soient étroitement liées, elles sont des concepts différents. Selon une étude de 2014 publiée dans la revue Cortex, l'expression faciale est définie comme "un mouvement involontaire ou volontaire des muscles faciaux qui produit un changement dans l'apparence du visage". L'étude a également défini l'émotion comme "un état mental complexe impliquant une expérience subjective, des changements physiologiques et comportementaux et des tendances comportementales" [5].

1.3 Les expressions faciales

1.3.1 Définition

L'expression faciale désigne un ensemble de mouvements musculaires et de traits du visage qui reflètent les états émotionnels, les sentiments et les pensées d'une personne. Ces mouvements peuvent être volontaires ou involontaires, et ils sont souvent accompagnés d'autres signaux non verbaux tels que la posture et les gestes corporels. Les expressions faciales sont

visuellement reconnues et considérées comme un moyen de communication important qui permet le transfert d'informations verbales et non verbales [6].



Figure 1. 1 - Les différentes expressions faciales.

1.3.2 Analyse et description des six expressions faciales

Selon Ekman et Friesen, les expressions faciales peuvent être classées en six catégories émotionnelles de base : « le dégoût, la colère, la joie, la tristesse, la surprise et la peur ». Ces expressions sont considérées comme universelles dans toutes les cultures humaines, bien que leur intensité et leur signification puissent varier en fonction du contexte culturel et social [7].

1. La joie (joy/happiness) : une émotion positive, cette expression se caractérise par un sourire, des yeux pétillants et des joues légèrement relevées.
2. La tristesse (sadness) : elle se manifeste par un froncement de sourcils, des yeux fermés ou abaissés, une bouche tombante et des joues tombantes.
3. La peur (fear/scared) : elle est marquée par des sourcils levés et rapprochés, des yeux grands ouverts, une bouche ouverte et des joues tendues vers l'extérieur.
4. La colère (anger) : elle se traduit par des sourcils froncés, des yeux étroits et une bouche serrée.
5. Le dégoût (disgust) : cette expression est caractérisée par une moue, des sourcils rapprochés et une lèvre supérieure légèrement relevée.
6. La surprise (surprise) : elle se manifeste par des sourcils levés, des yeux grands ouverts et une bouche également ouverte.

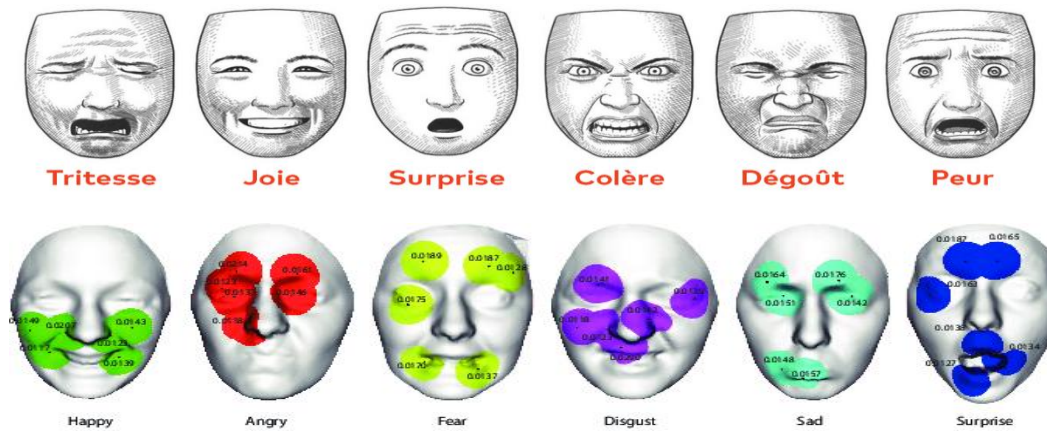


Figure 1. 2 - Les six expressions de base.

L'analyse de l'expression faciale implique l'examen des mouvements de la peau, tels que ceux des sourcils, des lèvres et des joues, ainsi que des rides, comme celles du front, des sourcils et du nez [8]. Pour cela, Le système de codage des actions faciales (FACs) est le système le plus utilisé qui code les différents mouvements faciaux possibles présenté par Ekman et Friesen en 1978 [9], FACs contient une liste de 46 mouvements faciaux appelés « unités d'action ». Chaque unité d'action (AU) correspond à un mouvement des muscles faciaux et possède un code numérique. Le FACs définit 32 actions atomiques plus les 14 descripteurs d'action tels que la direction du regard et la pose de la tête.

Upper Face Action Units					
AU 1	AU 2	AU 4	AU 5	AU 6	AU 7
Inner Brow Raiser *AU 41	Outer Brow Raiser *AU 42	Brow Lowerer *AU 43	Upper Lid Raiser AU 44	Cheek Raiser AU 45	Lid Tightener 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 AU 15	Upper Lip Raiser AU 16	Nasolabial Deepener AU 17	Lip Corner Puller AU 18	Cheek Puffer AU 20	Dimpler AU 22
Lip Corner Depressor AU 23	Lower Lip Depressor AU 24	Chin Raiser *AU 25	Lip Puckerer *AU 26	Lip Stretcher *AU 27	Lip Funneler AU 28
Lip Tightener	Lip Pressor	Lips Part	Jaw Drop	Mouth Stretch	Lip Suck

Figure 1. 3 - Liste des actions units relatives aux 6 expressions faciales [10].

Les paramètres d'animation faciale (FAPs) c'est le deuxième codage utilisé pour la détection des émotions, ce codage est défini par le standard MPEG-4 en 1998 où il est utilisé principalement pour l'animation des déformations du visage [11]. Les valeurs des paramètres d'animation du codage FAPs sont 84 points définis sur un visage neutre présentés dans la figure 1.4 :

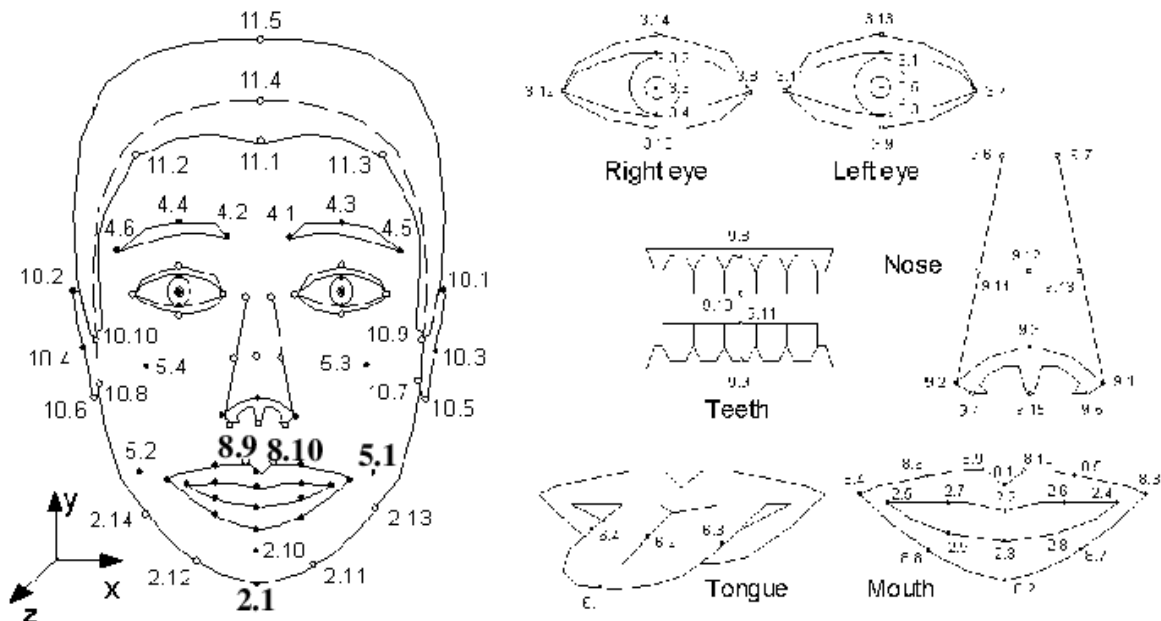


Figure 1. 4 - Modèle MPEG4 l'ensemble d'attributs faciaux [12].

1.4 Différentes approches de reconnaissance d'expressions faciales

1.4.1 Approches conventionnelles

Ces approches implémentent un processus en trois étapes, à savoir l'extraction de la région du visage, l'extraction des caractéristiques représentant les changements faciaux causés par les expressions faciales et la classification pour déterminer une similarité entre les caractéristiques extraites et celles de référence. Des modules de prétraitement de données peuvent également être utilisés pour améliorer les résultats de chaque étape. La Figure 1.5 illustre les trois modules principaux d'un système conventionnel de reconnaissance automatique des expressions faciales [13].

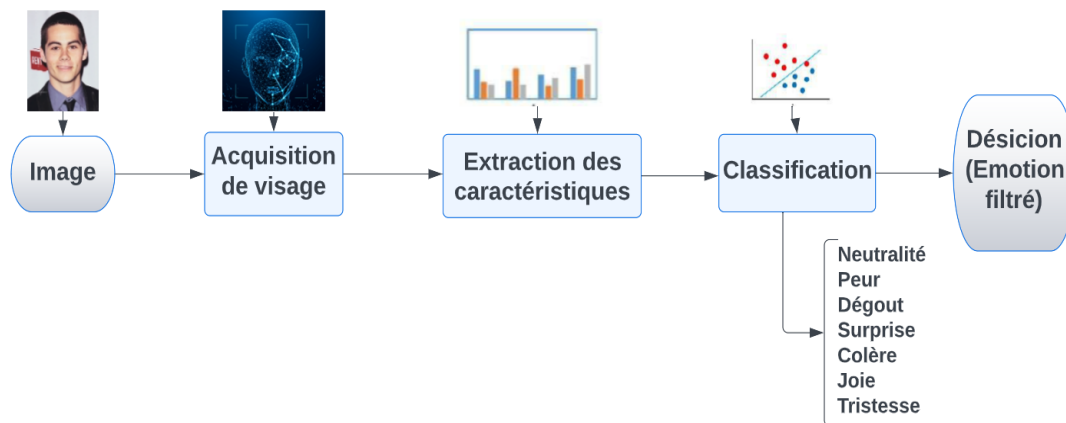


Figure 1. 5 - Structure d'un système de reconnaissance des expressions faciales.

1.4.1.1 Acquisition d'images et extraction du visage

Le système d'acquisition est équipé d'une caméra ou d'un appareil photo pour capturer une image 2D du visage. Cette image sera ensuite traitée pour détecter et localiser la région du visage ainsi que ses limites. Les algorithmes de détection de visages sont classés en quatre catégories, à savoir les méthodes basées sur la connaissance, les approches invariantes, les méthodes d'appariement de modèles et les méthodes basées sur l'apparence [4].

- a. Les méthodes basées sur la connaissance utilisent des règles prédéfinies basées sur la connaissance humaine afin de détecter un visage
- b. Les approches basées sur les caractéristiques invariantes visent à trouver des caractéristiques de structure du visage robustes aux conditions de pose et d'éclairage.
- c. La méthode basée sur un modèle utilise des modèles de visage préenregistrés. Calculer la valeur de corrélation entre le modèle et l'image d'entrée, déterminer la présence ou l'absence d'un visage en fonction de cette valeur de corrélation,
- d. Les techniques qui se basent sur l'apparence du visage font appel à l'apprentissage automatique et à des méthodes statistiques pour créer un modèle du visage.
- e. Les méthodes de détection de visage rigides, qui incluent des techniques de renforcement, reposent sur des modèles prédéfinis.

1.4.1.2 Extraction des caractéristiques

Les méthodes d'extraction des caractéristiques peuvent être classées en deux catégories distinctes : les méthodes basées sur les caractéristiques géométriques et les méthodes basées sur l'apparence [14].

Les caractéristiques géométriques décrivent la forme et l'emplacement des composants du visage tels que la bouche, le nez, les yeux, les sourcils, ainsi qu'à leurs relations mutuelles. Les caractéristiques d'apparence, représentent les variations d'apparence de la peau du visage, telles que les sillons et les rides. Ces caractéristiques peuvent être extraites sur l'ensemble du visage ou sur des régions spécifiques du visage [14].

1.4.1.3 Classification

Plusieurs classifieurs ont été utilisés à la reconnaissance des expressions faciales :

- LDA : Analyse Discriminante Linéaire (Linear discriminant analysis).
- SVM : Machines à vecteurs de support (Support Vector Machine)
- BN : Réseaux bayésiens (Bayesian Network).
- NN : Réseaux de neurone (Neural Networks).

Les avantages

En raison du modèle de comparaison sous-jacent, ces méthodes sont généralement insensibles au bruit et sont également robustes par rapport à l'occlusion des visages dans la base d'apprentissage et de l'occlusion des visages inconnus (hors de la base d'apprentissage) [15].

Les inconvénients

- Les méthodes utilisées ne nous fournissent pas de mesures précises, mais plutôt des paramètres. Par exemple, il peut être difficile de mesurer avec précision le degré d'ouverture de la bouche en pixels, mais il est plus facile de déterminer si la bouche est ouverte ou fermée [15].
- Lorsqu'on souhaite appliquer ces méthodes à des poses ou éclairages différents, il est nécessaire d'avoir une base de données d'apprentissage adaptée, ce qui peut rendre la construction de cette base difficile [15].
- Le principal problème des méthodes géométriques réside dans la détection imparfaite des points caractéristiques du visage, sur lesquels les mesures géométriques sont calculées.

Cette détection peut être affectée par divers facteurs tels que les variations d'éclairage, l'occlusion, etc. [15], et qu'aucune méthode n'arrive à reconnaître les 46 unités d'actions définies dans FACS.

1.4.2 Approches basées sur l'apprentissage profond (Deep learning)

A l'opposé des approches traditionnelles, l'approche basée sur l'apprentissage profond est devenue une approche générale de l'apprentissage par machine, donnant des meilleurs résultats dans de nombreuses études sur la vision par ordinateur tel que la REF automatiques avec la disponibilité de données volumineuses et le principal avantage des algorithmes d'apprentissage profond est de supprimer les techniques de prétraitement d'image en permettant un apprentissage de début à la fin directement à partir d'images d'entrées, ils existent des méthodes basées sur la détection des UAs afin de classifier l'émotion en utilisant des réseaux profonds. Les architectures les plus utilisées dans cette approche sont les réseaux neurones convolutifs CNN [16].

Les avantages

- Meilleure précision : Les réseaux de neurones profonds peuvent apprendre des caractéristiques plus complexes que les méthodes traditionnelles, ce qui permet une reconnaissance plus précise des expressions faciales.
- Adaptabilité : peuvent s'adapter à différentes conditions d'éclairage, d'orientation et de pose du visage, ce qui rend la reconnaissance des expressions plus robuste.
- Grande quantité de données : nécessitent une grande quantité de données d'entraînement pour fonctionner efficacement, mais une fois que ces données sont disponibles, le modèle peut être très précis.
- Les réseaux de neurones profonds peuvent être formés pour détecter automatiquement les expressions faciales sans intervention humaine, ce qui rend la reconnaissance des expressions plus rapide et plus efficace. Et peuvent être mis à jour avec de nouvelles données pour améliorer continuellement leurs performances en matière de reconnaissance des expressions faciales.

Les inconvénients

Ces méthodes sont évaluées uniquement sur des images qui ont été utilisées lors de l'apprentissage, ce qui rend difficile la prédiction de leur comportement lorsqu'elles sont

utilisées sur des sujets inconnus [15].

1.5 Applications pratiques de la reconnaissance des expressions faciales

Il existe de nombreuses utilisations pratiques de la technologie de reconnaissance d'expressions faciales [17], telles que :

- Sécurité : L'amélioration de la sécurité en détectant les comportements suspects ou en identifiant les personnes recherchées.
- Marketing : L'analyse des réactions des consommateurs à des publicités pour améliorer les campagnes marketing.
- Santé mentale : Le diagnostic des troubles mentaux en analysant les expressions faciales des patients.
- Education : La mesure de l'engagement des étudiants lors des cours en ligne.
- Divertissement : L'intégration dans les jeux vidéo pour permettre aux joueurs de contrôler les personnages avec leurs expressions faciales.

1.6 Conclusion

Dans ce chapitre, nous avons abordé la théorie des émotions ainsi qu'une brève analyse et description des expressions faciales et le système de codification. Par la suite, nous avons présenté les diverses méthodes conventionnelles et récentes utilisées pour la reconnaissance automatique des expressions faciales. Et que les méthodes basées sur le deep learning permettent de contourner les limitations des méthodes conventionnelles.

Le chapitre suivant portera sur l'apprentissage profond (deep learning), en citant les différents types de réseaux de neurones en détaillant notre approche basée sur les CNN pour la reconnaissance des émotions.

Chapitre 2

L'Apprentissage Profond
(Deep Learning)

2.1 Introduction

Avec le développement technologique, le terme apprentissage n'est pas exclusif à l'homme seulement mais, les machines aussi apprennent de leurs expériences, pour faire face aux besoins de leurs concepteurs. L'apprentissage profond, également connu sous le nom de Deep Learning, est un domaine de l'intelligence artificielle (IA) qui s'appuie sur des réseaux de neurones artificiels pour apprendre à partir de données. Cette méthode d'apprentissage automatique utilise des modèles mathématiques appelés réseaux de neurones artificiels, qui s'inspirent du fonctionnement du cerveau humain.

Dans ce chapitre, nous commencerons dans la première partie par présentation des quelques notions concernant l'apprentissage profond, puis nous nous focaliserons dans la deuxième partie, sur les réseaux de neurones convolutifs (CNN).

2.2 L'Apprentissage Profond (Deep Learning)

2.2.1 Définition

Le terme "Deep Learning" ou apprentissage profond a fait son apparition dans le domaine de l'apprentissage automatique (ML) grâce à Dechter en 1986 [18], et dans celui des réseaux neuronaux artificiels grâce à Aizenberg et al en 2000 [18].

L'apprentissage profond est une branche de l'apprentissage automatique (machine learning) qui utilise des réseaux de neurones artificiels profonds afin de résoudre des problèmes complexes [19]. Grâce à ses multiples couches de traitement composées de transformations linéaires et non linéaires, il est capable d'extraire des caractéristiques à partir des données brutes. De plus, il peut apprendre progressivement sur ces caractéristiques à travers chaque couche avec une intervention humaine minimale [20].

2.2.2 La relation entre l'Intelligence Artificielle (IA), Machine Learning (ML) et le Deep Learning (DL)

L'Intelligence Artificielle (IA) est un domaine de l'informatique qui vise à créer des machines capables de simuler des comportements et des capacités humaines. Le Machine Learning est un sous-ensemble de l'IA qui utilise des techniques d'apprentissage pour permettre aux ordinateurs d'effectuer des tâches sans être explicitement programmés pour le faire.

Le Deep Learning, est une technique avancée de Machine Learning qui utilise des réseaux de neurones artificiels pour apprendre à partir de données non structurées, telles que des images ou du texte.

Artificial intelligence

A science devoted to making machines think and act like humans

Machine Learning

Focuses on enabling computers to perform tasks without explicit programming

Deep Learning

A subset of machine learning based on artificial neural network

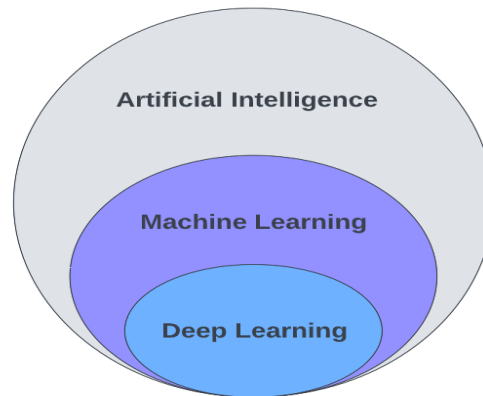


Figure 2. 1 - La relation entre l'intelligence artificielle (AI), le ML et le DL.

2.2.3 Pourquoi l'Apprentissage Profond ?

Bien que les algorithmes de ML soient efficaces pour résoudre de nombreux problèmes, ils n'ont pas réussi à résoudre certains des problèmes clés de l'IA, tels que la reconnaissance vocale et la reconnaissance d'objets [21]. L'un des grands avantages de l'apprentissage profond, est qu'il est alimenté par des quantités massives de données (big data) [22].

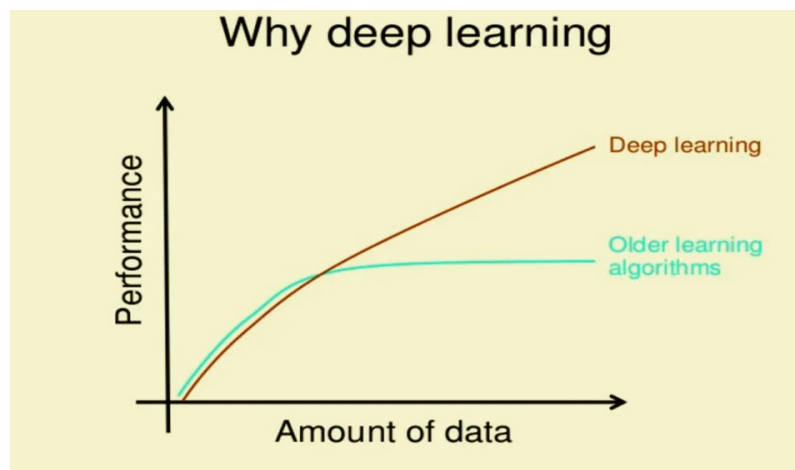


Figure 2. 2 - Les performances de l'apprentissage profond par rapport à d'autres algorithmes d'apprentissage [22].

Les différents algorithmes du deep Learning ne sont apparus que lorsque l'apprentissage automatique n'a pas réussi à résoudre une grande variété de problèmes en intelligence artificielle (IA). Ces algorithmes visaient à atteindre les objectifs suivants :

- D'améliorer les performances des algorithmes traditionnels dans de telles tâches de l'IA et s'adapter à n'importe quel type de problème.
- Pour développer une grande quantité de données telle que les big data.
- Extraire automatiquement les caractéristiques des données de manière automatisée [21].

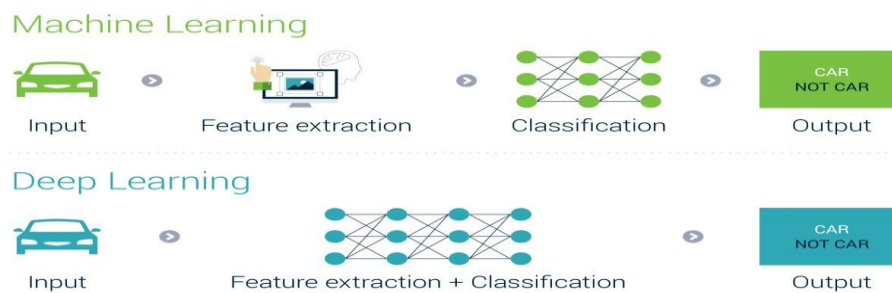


Figure 2. 3 - Comparaison entre apprentissage automatique (ML) et l'apprentissage profond (DL).

L'apprentissage profond utilise des réseaux de neurones afin d'obtenir des représentations pertinentes de caractéristiques à partir de données.

2.3 Réseaux de neurones

En 1943, W. McCulloch et W. Pitts ont développé le concept de neurone formel, qui est une abstraction mathématique du neurone biologique. Cette avancée a marqué l'apparition des réseaux de neurones artificiels et a ouvert la voie à l'utilisation de modèles mathématiques pour étudier le fonctionnement du cerveau et créer des systèmes d'intelligence artificielle (AI).

2.3.1 Principe des réseaux de neurones

Les réseaux de neurones sont des modèles informatiques qui sont conçus pour simuler le fonctionnement des neurones biologiques dans le cerveau humain (Figure 2.4). Et qui par la suite s'est rapproché des méthodes statistiques [23]. Les réseaux de neurones artificiels (RNA) sont des systèmes composés de multiples neurones artificiels fortement interconnectés. Chaque neurone artificiel, agissant comme un processeur élémentaire calcule une seule sortie en fonction des informations qu'il reçoit.

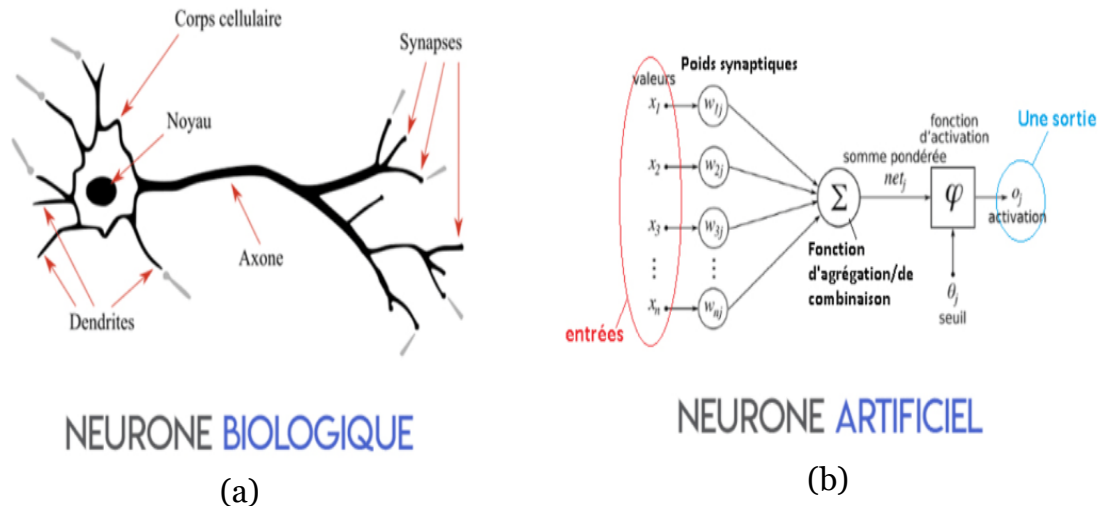


Figure 2. 4 - Comparaison entre Neurone biologique (a) et neurone artificiel (b).

Le neurone artificiel reprend l'architecture et le fonctionnement du neurone biologique où les synapses assurent les connexions avec les autres neurones, les dendrites sont les entrées, les axones sont les sorties et le noyau active les sorties selon les stimulations en entrées selon l'expression suivante [8] :

$$\hat{y} = f(w \cdot x + b)$$

Où

- x : le vecteur d'entrée.
- \hat{y} : la sortie (prédiction).
- w et b : respectivement, le poids et le biais (les paramètres) influençant le fonctionnement du neurone à travers la fonction d'activation associé f .

Un seul neurone ne suffit pas pour résoudre des problèmes complexes. Pour cette raison, les réseaux de neurones sont construits en associant plusieurs dizaines, voire centaines ou milliers de neurones, selon une structure plus ou moins complexe. Les réseaux de neurones les plus courants se distinguent par leur organisation et leur nombre de couches, leur nombre de neurones, ainsi que leur type d'apprentissage.

Le réseau de neurones est donc composé de plusieurs couches de neurones, où chaque couche reçoit et traite les informations provenant de la couche précédente (comme illustré dans la figure 2.5). Ainsi, le réseau multicouche, également connu sous le nom de perceptron multicouche (Multi Layer Perceptron MLP) apprendra par exemple à reconnaître les lettres avant de s'attaquer aux mots dans un texte, ou déterminent si une photo contient un visage, puis reconnaissent de qui il s'agit [8].

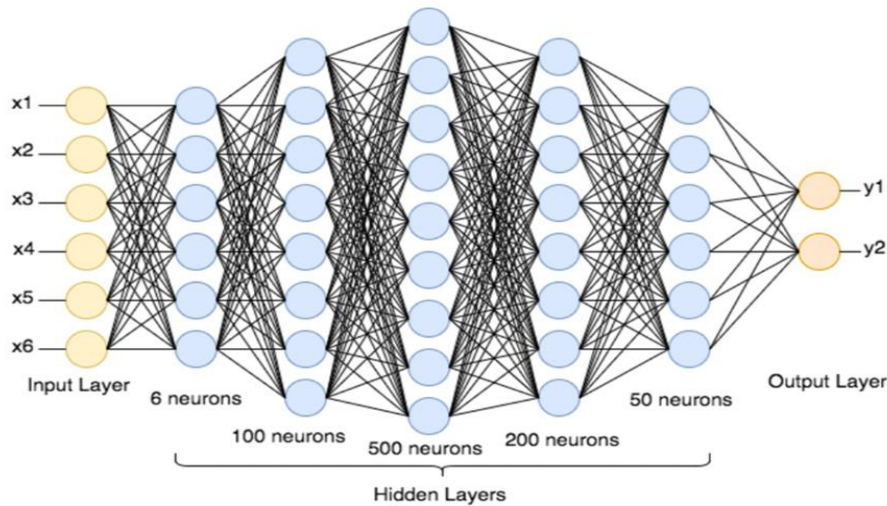


Figure 2. 5 - Réseau de neurones artificiels (RNA).

2.3.2 L'architecture des RNA

La topologie d'un réseau de neurones artificiels décrit son architecture, c'est-à-dire comment les neurones sont connectés entre eux. Il existe nombreuses types d'architecture de réseaux de neurones, chacune avec ses propres caractéristiques et applications.

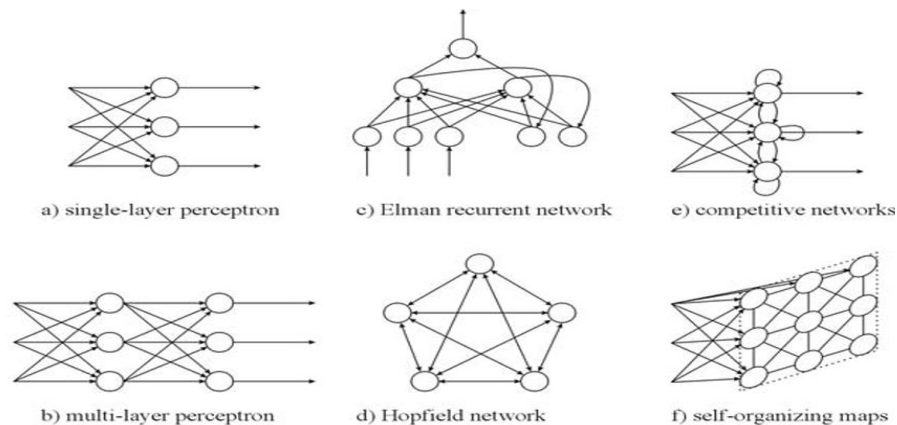


Figure 2. 6 - Quelques topologies standards [24].

L'architecture (b) représente une topologie standard largement utilisée dans les réseaux neuronaux, notamment dans les architectures de perceptron multicouche telles que celle étudiée dans ce projet. Les neurones sont structurés en couches, où l'information d'entrée est transmise de couche en couche jusqu'à obtenir le résultat en sortie. Cette architecture divise les neurones en trois catégories : les neurones d'entrée, les neurones cachés et les neurones de sortie (les neurones cachés se situent entre les neurones d'entrée et ceux de sortie). Les neurones sont

généralement organisés en couches comprenant la couche d'entrée, une ou plusieurs couches cachées, ainsi que la couche de sortie (input layer, hidden layer, output layer).

2.3.3 Les différentes Architectures du Deep Learning

En raison de la multitude d'architectures profondes disponibles, il peut être difficile de comparer leurs performances, étant donné que certaines ne sont pas testées sur les mêmes ensembles de données. Compte tenu de la rapidité de l'évolution du domaine du Deep Learning, de nouvelles architectures, variantes et algorithmes sont régulièrement introduits.

Dans la suite, nous présentons les architectures les plus courantes :

1. Réseaux de neurones convolutifs (CNN).
2. Réseau de neurones récurrents (RNN).
3. Modèle génératif.

2.3.3.1 Réseaux de neurones convolutifs (CNN).

Les réseaux de neurones convolutifs (CNN) sont une catégorie de réseaux de neurones artificiels qui ont révolutionné le domaine de la vision par ordinateur. Ils sont particulièrement bien adaptés pour traiter des données avec une structure en grille, comme les images [25]. CNN a identifié avec succès des visages, des objets, des panneaux de signalisation et des voitures autonomes [20]. De plus, il a récemment montré son efficacité dans diverses tâches de traitement du langage naturel, notamment la classification des phrases [26], [27], [28].

En ML, les réseaux convolutifs sont un type de réseau neuronal direct (feed forward) inspiré des processus biologiques [29]. Il existe quatre principales opérations (couches) illustrées dans le CNN à savoir : la couche convolution, la couche Rectified Linear Unit, la couche Pooling, la couche entièrement connectée.

2.3.3.2 Réseau de neurones récurrents (RNN)

Un RNN est composé de neurones qui sont connectés de manière récurrente pour former des boucles. Ces boucles permettent aux neurones de conserver une mémoire de l'information précédente et de l'utiliser pour traiter la donnée actuelle. Les RNN sont particulièrement utiles pour le traitement du langage naturel, la reconnaissance vocale, la prédiction de séries

temporelles, et bien d'autres encore [30]. Les RNN sont désignés comme récurrents car ils effectuent la même tâche pour chaque élément d'une séquence, où la sortie dépend des calculs précédents.

Le RNN le plus couramment utilisé est le réseau de neurones à mémoire à court terme (LSTM), introduit par Hochreiter et Schmidhuber en 1997. Les LSTM sont capables de conserver une mémoire à long terme des informations précédentes, en plus de leur mémoire à court terme [30].

En principe, les RNN peuvent utiliser des informations sur des séquences de longueur quelconque, mais dans la pratique, on les restreint généralement à prendre en compte uniquement les dernières étapes passées [31]. Ils sont utilisés pour : La modélisation du langage et génération de texte, la traduction automatique, la reconnaissance vocale, la description des images.

2.3.3.3 Modèle génératif

Un réseau de neurones génératif (Generative Neural Network) permet de générer de nouvelles données qui ressemblent à celles qui ont été utilisées pour entraîner le modèle. Le modèle utilise une architecture de réseau neuronal pour apprendre les caractéristiques et les relations sous-jacentes des données d'entraînement, puis utilise ces connaissances pour générer de nouvelles données.

Les modèles discriminatifs tels que les réseaux de neurones convolutifs (CNN) et récurrents (RNN) sont conçus pour prédire les données en fonction du label et de l'entrée fournie. En revanche, les modèles génératifs décrivent comment les données sont générées, et ils utilisent la loi de Bayes pour apprendre et effectuer des prédictions [32].

Quelques exemples de modèles génératifs :

- Generative Adversarial Networks.
- Deep Belief Networks.
- Boltzmann Machines.

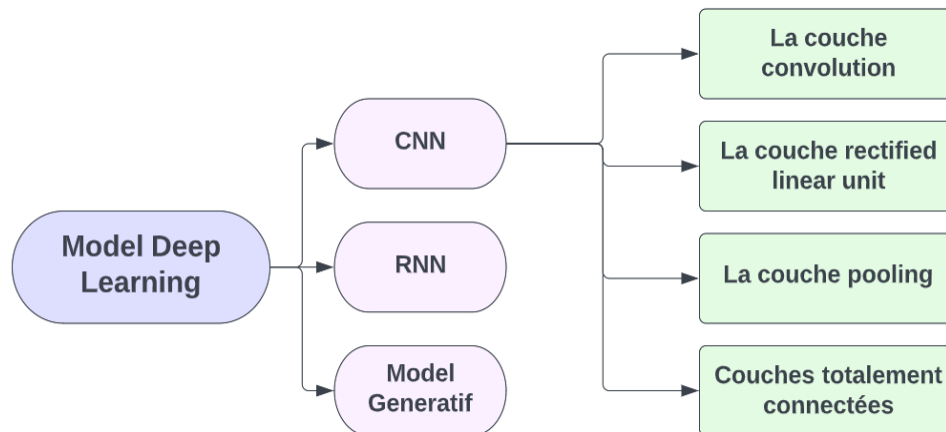


Figure 2. 7 - Les différents modèles du Deep Learning.

2.4 Les applications de Deep learning

Le Deep Learning est utilisé dans de nombreux domaines, offrant de nouvelles possibilités telles que :

- La reconnaissance faciale : Les algorithmes de DL peuvent apprendre à détecter et reconnaître les caractéristiques faciales à partir d'une photo [33].
- La détection d'objets (Les algorithmes de détection d'objets peuvent identifier et localiser précisément des éléments ou des personnes dans une image) [33].
- Colorisation d'images en noir et blanc en s'appuyant sur des modèles préalablement appris.
- Ajouter du son aux films muets (silencieux).
- La traduction automatique du texte d'une langue à une autre.
- Générer d'écriture automatique, imitant l'écriture humaine.
- Génération de légende d'image.
- Jeu automatique

2.5 Réseaux de Neurones Convolutifs CNN

Les réseaux de neurones convolutifs ont été introduits pour la première fois par Yann LeCun, Bengio Yoshua et al, en 1998 dans leur article intitulé "Gradient-based learning applied to document recognition". Depuis cette publication, ces réseaux ont été largement utilisés dans divers domaines tels que la reconnaissance d'images, la classification de vidéos, la détection d'objets, la segmentation d'images [25].

Aujourd'hui, les réseaux de neurones convolutifs, sont toujours les modèles les plus

performants pour la classification d'images. Google, Microsoft, Facebook, Nvidia (géant du processeur graphique), Alibaba (site marchand chinois), Baidu (le moteur de recherche chinois) utilisent les CNN dans leurs applications [34].

Durant cette phase, nous nous appuyerons exclusivement sur le CNN (l'approche utilisée dans notre système de reconnaissance des émotions faciales).

2.5.1 Présentation des CNN

Actuellement, les réseaux de neurones convolutifs, également connus sous le nom de CNN (Convolutional Neural Network), demeurent les modèles les plus performants pour la classification d'images [35]. Le terme "réseau de neurones convolutif" fait référence à l'utilisation d'une opération mathématique appelée convolution dans ce type de réseau neuronal. Les CNN sont simplement utilisés la convolution à la place de la multiplication matricielle dans au moins une de leurs couches.

Les CNN sont spécifiquement conçus pour traiter des images en tant qu'entrée, ce qui se reflète dans leur architecture distincte composée de deux blocs principaux.

- 1) Le premier bloc des CNN joue un rôle crucial, car il agit en tant qu'extracteur de caractéristiques. Il réalise cela en appliquant une opération de filtre de convolution. La première couche convolue l'image avec plusieurs noyaux de convolution, ce qui génère des "feature maps". Ensuite, ces feature maps sont soumises à des processus de normalisation (via une fonction d'activation) et/ou de redimensionnement. Ce processus itératif peut être répété plusieurs fois en filtrant les feature maps obtenues précédemment à l'aide de nouveaux noyaux, ce qui génère de nouvelles feature maps. Ces nouvelles feature maps subissent ensuite des étapes de normalisation et de redimensionnement. Cette séquence peut être répétée plusieurs fois. Enfin, les valeurs des dernières feature maps sont concaténées pour former un vecteur. Ce vecteur représente la sortie du premier bloc, et l'entrée du deuxième bloc.
- 2) Le deuxième bloc : traite le vecteur d'entrée en le transformant à l'aide de multiples combinaisons linéaires et fonctions d'activation, pour renvoyer un nouveau vecteur en sortie. Ce dernier vecteur est composé d'autant d'éléments qu'il y a de classes. Chaque élément du vecteur représente la probabilité que l'image appartienne à la classe correspondante.

2.5.2 Architecture de CNN

Les réseaux de neurones convolutifs sont basés sur des perceptrons multicouches (MLP) [36], l'architecture CNN est formée par un empilement de couches de traitement indépendantes.

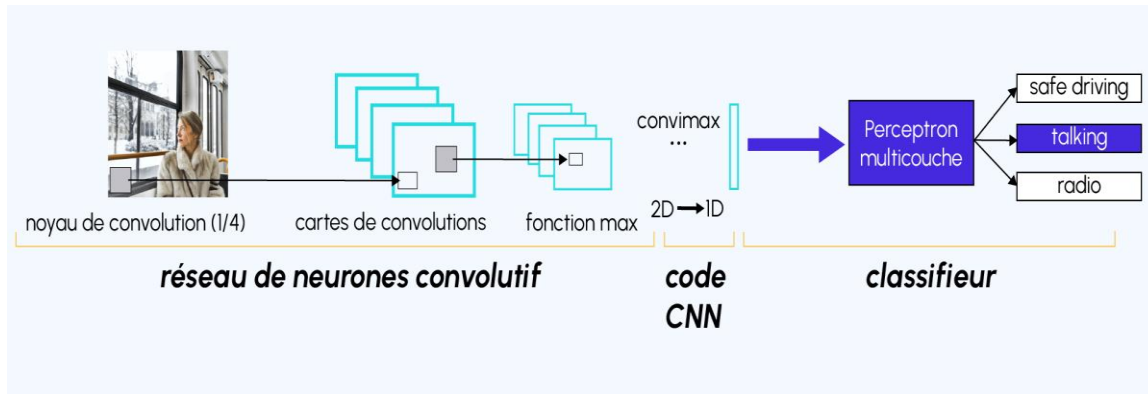


Figure 2. 8 - Schéma représentant l'architecture d'un CNN [37].

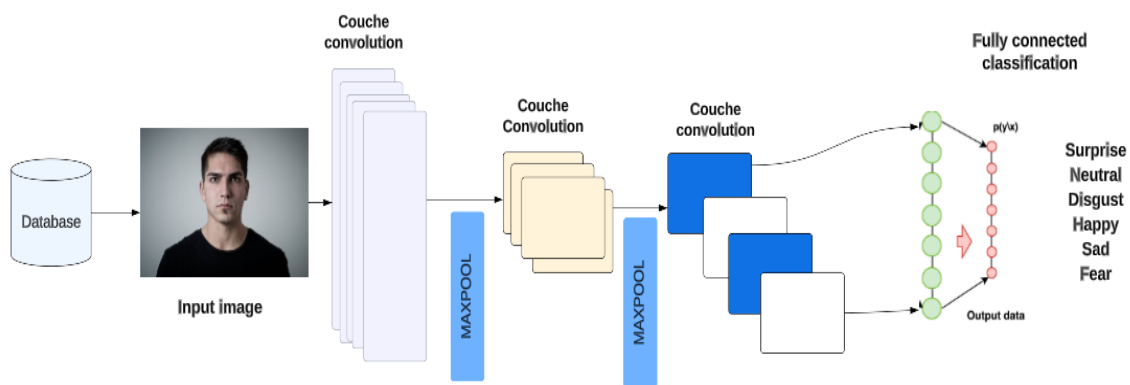


Figure 2. 9 - Architecture de réseau proposée.

Un réseau de neurones convolutif est composé de quatre types de couches : la couche de convolution, la couche de pooling, la couche de rectification linéaire (ReLU) et la couche entièrement connectée.

2.5.3.1 Couche convolution (CONV)

La convolution peut être simplifiée en tant qu'application d'un filtre mathématique à une image [35], utilisée pour le traitement et la reconnaissance d'images [37]. D'un point de vue plus technique, il s'agit de faire glisser une matrice sur une image, et pour chaque pixel, utiliser la somme de la multiplication de ce pixel par la valeur de la matrice, pour chaque pixel, d'effectuer la somme des produits entre ce pixel et les valeurs correspondantes de la matrice.

Cette technique nous permet de détecter les parties de l'image qui pourraient être intéressantes [35]. Son effet sur les images est similaire au filtrage et fonctionne comme suit :

- Tout d'abord, nous définissons la taille de la fenêtre de filtre, située en haut à gauche.
- La fenêtre de filtre, qui représente la caractéristique (feature), se déplace progressivement de gauche à droite sur un nombre prédéfini de cases (le pas), jusqu'à atteindre la fin de l'image.
- Un calcul de convolution est effectué sur chaque partie de l'image détectée et une carte d'activation ou une carte de caractéristiques est obtenue en sortie indiquant où les caractéristiques sont situées dans l'image : plus la valeur de la carte de caractéristiques (feature map) est élevée, plus la portion balayée de l'image ressemble à la caractéristique.

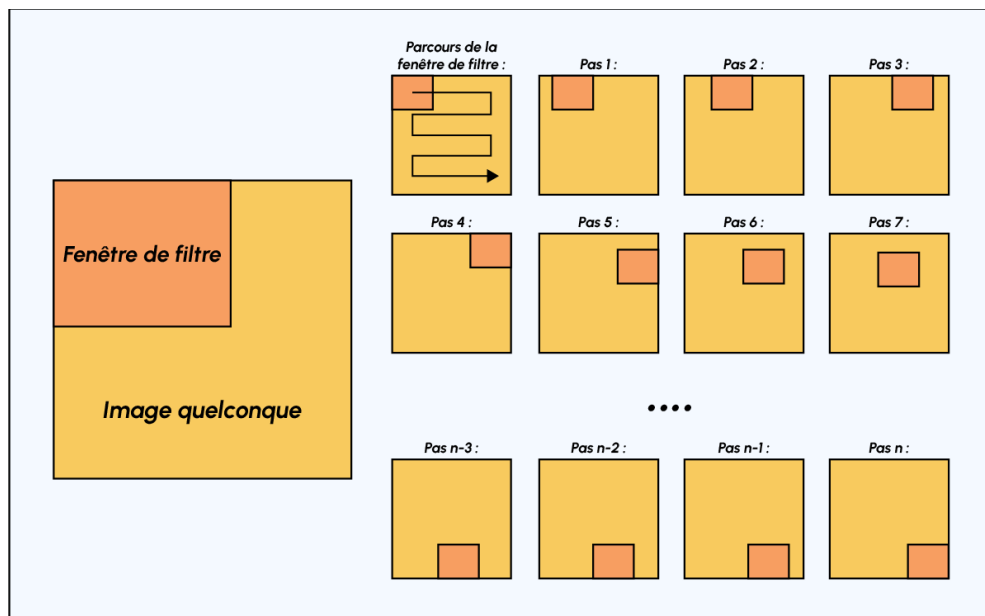


Figure 2. 10 - Schéma de déplacement de la fenêtre de filtre à travers l'image [37].

Les trois hyper paramètres utilisés pour la détermination des dimensions du volume de la couche convolution (volume de sortie) sont : la profondeur, le pas et la marge.

- La profondeur de la couche correspond au nombre de noyaux de convolution utilisés, ou au nombre de neurones associés à un même champ récepteur.
- Le pas détermine le niveau (dégré) de chevauchement des champs récepteurs. Lorsque le pas est petit, les champs récepteurs se chevauchent, ce qui entraîne une augmentation de la taille du volume de sortie.
- La marge (à 0 ou "zero padding" en anglais) consiste à ajouter des zéros à la frontière du volume d'entrée. Cette marge est souvent définie à zéro. Elle permet de contrôler la

dimension spatiale du volume de sortie. Parfois, il est souhaitable de maintenir la même taille (surface) que celle du volume d'entrée [38].

2.5.3.1.1 Les différentes convolutions

Il existe différents types de convolutions qui peuvent être utilisées en fonction de l'objectif souhaité :

- La convolution classique, qui consiste à décaler le noyau entre chaque calcul, et le padding, qui détermine comment la convolution est appliquée lorsque le noyau dépasse les limites de l'image.
- La convolution dilatée (dilated convolution), dans laquelle le noyau de convolution est élargi en insérant des zéros entre les éléments du noyau. Cette technique permet d'augmenter la taille effective du noyau de convolution sans augmenter le nombre de paramètres du modèle. Un paramètre supplémentaire, appelé le taux de dilution (dilation rate), détermine le nombre de pixels à ignorer entre les éléments du noyau.
- Convolution transposée (transposed convolution), permet de passer d'une image de petite taille à une image de plus grande taille.
- La séparable convolution, qui décompose la convolution en convolutions plus simples.

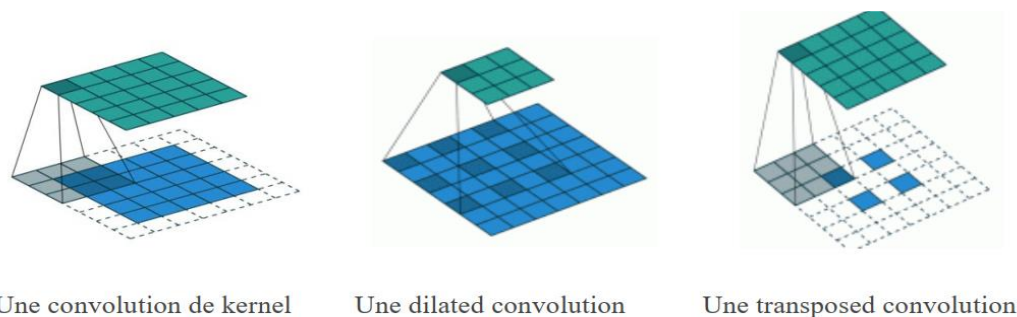


Figure 2. 11 - Différents types de convolutions.

2.5.3.2 La couche ReLu – Unité linéaire rectifiée

Cette couche applique une fonction d'activation non linéaire qui transforme les sorties de la couche précédente en activités sparses en définissant une valeur seuil.

Cette fonction d'activation est appliquée à chaque pixel de l'image après la convolution et remplace toutes les valeurs négatives par zéro. Si cette fonction n'est pas appliquée, la fonction créée serait linéaire, ce qui ne permettrait pas de résoudre le problème XOR, car aucune fonction

d'activation n'est appliquée dans la couche de convolution [35].

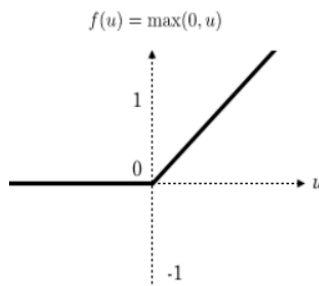


Figure 2. 12 - La fonction d'activation ReLU est définie comme suit :

$$f(x) = \max(0, x).$$

2.5.3.3 La couche de Pooling

La couche de Pooling est généralement placée entre deux couches de convolution. Elle prend en entrée plusieurs feature maps et applique l'opération de Pooling à chacune d'entre elles. L'objectif de l'opération de Pooling est de réduire la taille des images tout en préservant les caractéristiques essentielles. Pour ce faire, l'image est découpée en cellules régulières, et la valeur maximale à l'intérieur de chaque cellule est conservée [35].

Cette méthode implique l'utilisation d'une fenêtre de 2 ou 3 pixels qui se déplace sur l'image, similaire à la convolution. Cependant, cette fois-ci, nous effectuons des sauts de 2 pour une fenêtre de taille 2 et des sauts de 3 pour une fenêtre de taille 3. La taille de la fenêtre est appelée "kernel size" et les sauts sont appelés "strides". À chaque étape, la valeur maximale parmi celles présentes dans la fenêtre est sélectionnée, et cette valeur devient le pixel d'une nouvelle image. Cette opération est appelée Max Pooling [35].

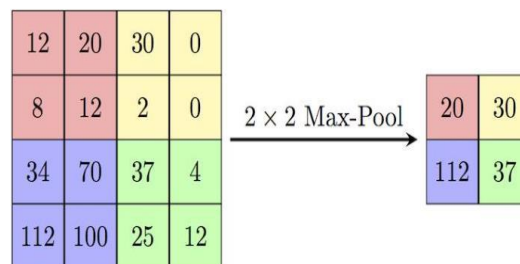


Figure 2. 13 - Pooling avec un filtre de taille 2x2 et un pas de 2.

2.5.3.4 Couches Entièrement Connectées (Fully connected)

La couche entièrement connectée, généralement située en dernière position dans un réseau de neurones, prend en entrée un vecteur et produit en sortie un nouveau vecteur. Pour cela, elle combine linéairement les valeurs d'entrée et applique éventuellement une fonction d'activation [35]. Une fois les caractéristiques de l'entrée extraites, un perceptron ou perceptron multicouche (MLP) est attaché à la fin du réseau. Ce perceptron prend en entrée les caractéristiques extraites et produit un vecteur de N dimensions, où N est le nombre de classes, et chaque élément représente la probabilité d'appartenance à une classe donnée [36]. Chaque probabilité est calculée à l'aide de la fonction softmax [21]. Une fonction softmax est utilisée pour prédire une classe parmi K classes mutuellement exclusives. La perte d'entropie croisée sigmoïdale est utilisée pour prédire K valeurs de probabilité indépendantes dans l'intervalle [0,1]. La perte euclidienne est utilisée pour effectuer une régression vers des valeurs réelles [39]. Le terme « entièrement connecté » implique que chaque neurone dans la couche précédente est connecté à chaque neurone sur la couche suivante.

2.5.4 Quelques réseaux convolutifs pré-entraînés – célèbres

Voici quelques exemples de réseaux de neurones convolutifs célèbres :

- LeNet [27] : Yann LeCun est l'un des pionniers dans le développement des premières applications réussies des réseaux de neurones convolutifs dans les années 1990. Parmi ses réalisations les plus célèbres, on trouve l'architecture LeNet, qui était utilisée pour la reconnaissance des codes postaux, des chiffres, et bien d'autres applications.
- AlexNet [40] : développé par Alex Krizhevsky, Ilya Sutskever et Geoff Hinton, a été le pionnier dans la popularisation des réseaux de neurones convolutifs dans le domaine de la vision par ordinateur. Marquant une avancée majeure dans la capacité des réseaux neuronaux à reconnaître et classer des images.
- Le réseau ResNet (Residual Network) : un réseau de neurones convolutionnels profond avec une architecture résiduelle, développé par Microsoft en 2015.
- GoogLeNet : créé par Google, ce module utilise un pooling global moyen, ce qui élimine une quantité considérable de paramètres. GoogLeNet possède également plusieurs versions, dont Inception-v4 et Xception.
- VGG16 : un modèle de réseaux de neurones très populaire utilisé pour la classification d'images, développé par l'Université d'Oxford en 2014.

Chapitre 3

Implémentations Et Résultats
Expérimentaux

3.1 Introduction

Ce chapitre a pour objectif de présenter les différentes étapes de l'implémentation d'un système de reconnaissance d'expressions faciales en temps réel à base de CNN. Nous allons utiliser un ensemble de donnée contenant 28709 images pour l'entraînement, et 7178 images pour les tests. La base de données que nous avons utilisée, déjà divisée en deux dossiers pour l'entraînement et le test est composée d'images représentant différentes expressions faciales téléchargées à partir de la plateforme Kaggle. Ensuite, nous détaillons les différentes étapes de l'implémentation de notre approche, ainsi que les différentes étapes de réalisation du modèle. Nous allons débiter premièrement en présentant les ressources, le langage de programmation et l'environnement de développement que nous avons utilisé, puis les étapes de la réalisation de notre modèle. Enfin, nous clôturerons ce chapitre en présentant les résultats expérimentaux obtenus pour notre approche.

3.2 But du projet

Le but de ce projet est de développer un système de reconnaissance des émotions faciales précis et fiable qui peut prendre en entrée des images faciales à partir d'une webcam, d'une séquence vidéo ou d'une image, contenant certaines expressions, les reconnaître et les classer dans sept classes d'expressions différentes, telles que : « Neutre », « joie », « tristesse », « colère », « peur », « surprise », et « dégoût », avec un taux de précision élevé. Le système utilisera des techniques de vision par ordinateur et d'apprentissage profond pour identifier les traits du visage et les mouvements associés à chaque expression faciale. Cela permettra au système de classer les images faciales humaines dans l'une des sept classes d'expressions différentes, afin d'être utilisé pour la surveillance de la sécurité, la reconnaissance des émotions et la recherche en psychologie.

3.3 Environnement de développement

Dans cette section, nous allons présenter les environnements matériels et logiciels que nous avons utilisés pour notre travail. Nous allons décrire les différents outils, langages de programmation, bibliothèques et Framework que nous avons utilisés pour la mise en place de notre système de reconnaissance des expressions faciales.

3.3.1 Environnement matériel (HARDWARE)

Afin de mener à bien ce projet, nous avons travaillé sur un MacBook Pro doté les caractéristiques suivantes :

- *Processeur : Intel® Core™ i5-3210M CPU @ 2.50GHz*
- *Mémoire physique (RAM) installée : 16Go*
- *Carte graphique : Intel® HD Graphics 4000*

3.3.2 Environnement logiciel (SOFTWARE)

- *OS : Microsoft Windows 10 Professionnel « 64bits »*
- *Logiciel : Python 3.10.2*

3.4 Outils de développement

3.4.1 Visual Studio Code (VS Code)

Visual Studio Code, créée par Microsoft, est un éditeur de code source gratuit et open-source qui permet écrire, modifier et déboguer du code dans une variété de langages de programmation.



Figure 3. 1 - Logo de VS Code.

3.4.2 Python

Python est un langage de programmation considéré comme de haut niveau. Cela signifie qu'il s'éloigne du langage machine et joutant des concepts et des outils facilitant la lecture et l'écriture, et s'apparentant davantage aux langages humains (l'anglais en l'occurrence) [41]. Il s'agit d'un langage de programmation couramment utilisé dans le développement d'applications web, la création de logiciels, l'analyse de données et l'apprentissage automatique (ML). Le logiciel Python est téléchargeable gratuitement, s'intègre bien à tous les types de systèmes et accélère le développement [42].



Figure 3. 2 - Logo de python.

3.5 Bibliothèques utilisées

3.5.1 OpenCV (Open Source Computer)

OpenCV est une bibliothèque graphique spécialisée dans le traitement d'images. Elle a été développée par Intel en 2000 et peut être utilisée avec les langages de programmation Python, C++ et Java. Pour comprendre comment fonctionne OpenCV, nous devons d'abord définir ce qu'est la vision par ordinateur.

La vision par ordinateur est une branche de l'IA (l'intelligence artificielle) qui permet aux machines d'analyser, traiter et comprendre des images. OpenCV met à disposition une bibliothèque de plus de 2500 algorithmes dédiés à la vision par ordinateur, ce qui permet d'effectuer divers traitements sur les images, on peut citer l'extraction de couleurs, l'application des filtres, la détection de visages. Ces algorithmes se basent principalement sur des calculs mathématiques complexes, notamment sur les traitements sur les matrices (car une image peut être considérée comme une matrice de pixels) [43].



Figure 3. 3 - Logo d'OpenCV.

3.5.2 TensorFlow

TensorFlow a été créé par une équipe de chercheurs et d'ingénieurs de Google Brain. Dans le domaine de DL, elle est la bibliothèque la plus largement utilisée. L'une des raisons pour lesquelles TensorFlow est si populaire est la prise en charge de plusieurs langages pour créer des modèles d'apprentissage profond, tels que Python, C++ et R. L'architecture flexible de TensorFlow nous permet de déployer nos modèles d'apprentissage en profondeur sur un ou plusieurs processeurs (ainsi que sur des GPU). Voici quelques cas d'utilisations populaires de TensorFlow :

- Analyse vidéo.
- Applications textuelles telles que le résumé de texte et la détection de langue.
- Reconnaissance d'image (la reconnaissance faciale, détection d'objets).
- Reconnaissance sonore.



Figure 3. 4 - Logo de TensorFlow.

3.5.3 Keras

Keras est écrit en Python et peut s'exécuter sur TensorFlow (ainsi que CNTK et Theano). C'est une API de haut niveau, développée dans le but de permettre une expérimentation rapide. Les réseaux de neurones convolutifs et les réseaux de neurones récurrents sont pris en charge par Keras. Fonctionne de manière transparente avec le CPU et le GPU. Keras possède plusieurs architectures, pour résoudre une grande variété de problèmes (ex : classification des images) : InceptionV3, Mobilenet, VGG16, VGG, ...etc.



Figure 3. 5 - Logo de Keras.

3.5.4 Numpy

NumPy est une bibliothèque Python open-source utilisée pour le calcul scientifique et l'analyse de données. Elle fournit des structures de données de haut niveau et des fonctions pour travailler avec des tableaux multidimensionnels. Elle est conçue pour travailler efficacement avec de grandes quantités de données et pour fournir des opérations numériques rapides. NumPy est également utilisé pour créer de puissants modèles d'apprentissage automatique et d'autres applications à forte intensité de données [44].



Figure 3. 6 - Logo de NumPy.

3.5.5 Matplotlib

Matplotlib est une bibliothèque de visualisation graphique (traçage) disponible dans le langage Python en tant que composant de NumPy, une ressource pour le traitement numérique de grandes quantités de données. Matplotlib utilise une API orientée objet pour intégrer des tracés dans des applications Python [45].



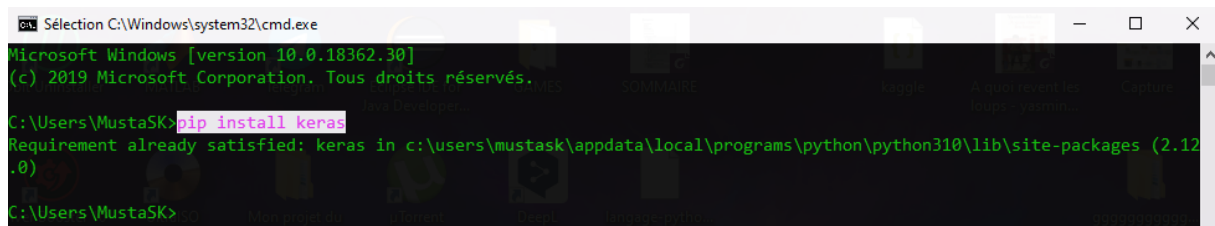
Figure 3. 7 - Logo de Matplotlib.

3.6 Installation des bibliothèques

L'installation des bibliothèques OpenCV, TensorFlow, Keras, NumPy et Matplotlib est essentielle pour notre projet. Ces bibliothèques sont utilisées pour traiter les images en temps réel, extraire les caractéristiques du visage, développer des modèles d'apprentissage automatique pour la reconnaissance des expressions faciales et visualiser les résultats. Dans ce guide, nous allons expliquer comment installer ces bibliothèques avec pip, un gestionnaire de packages Python, afin que nous puissiez commencer à les utiliser dans notre projet.

Pour installer la bibliothèque Keras, par exemple avec pip, il faut suivre les étapes suivantes :

- Ouvrir une ligne de commande ou un terminal sur notre système d'exploitation.
- Utiliser la commande "pip install keras" pour procéder à l'installation de Keras.



```
Sélection C:\Windows\system32\cmd.exe
Microsoft Windows [version 10.0.18362.30]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Users\MustaSK> pip install keras
Requirement already satisfied: keras in c:\users\mustask\appdata\local\programs\python\python310\lib\site-packages (2.12.0)

C:\Users\MustaSK>
```

Figure 3. 8 - Installation de la bibliothèque Keras.

3.7 Base de données (Database)

Pour pouvoir entraîner les modèles, il est nécessaire de disposer d'ensembles de données avec des émotions bien classées et suffisamment de données pour pouvoir entraîner les modèles de manière optimale. Dans cette section, nous présentons la base de données que nous avons utilisés au cours de l'exécution de ce travail.

Il existe nombreuses bases de données disponibles pour la reconnaissance des expressions faciales, notamment : Extended Cohn Kanade (**CK +**), JAFFE (**J**ApAnese **F**emale **F**acial **E**xpressions), FER 2013 (**F**acial **E**xpression **R**econition 2013 database), KDEF (**K**arolinska **D**irected **E**motional **F**aces), MMI ...etc.

Parmi ces plateformes d'apprentissage nous avons accès à la base de données « FER2013 » elle contient 35887 d'images d'expressions faciales en niveaux de gris de 48x48 pixels, qui sont réparties en deux parties. Le nombre d'images utilisées pour l'entraînement et le test du CNN est présentées dans le tableau 3.1 :

Tableau 3. 1 - Détails de la base de données utilisée.

Emotions	Entrainement	Test
<i>Angry</i>	3995	958
<i>Disgust</i>	436	111
<i>Fear</i>	4097	1024
<i>Happy</i>	7215	1774
<i>Sad</i>	4965	1233
<i>Surprise</i>	4830	1247
<i>Neutral</i>	3171	831
<i>Total</i>	25709	7178



Figure 3. 9 - Exemples d'images pour la reconnaissance des émotions faciales à partir de quatre ensembles de base de données.

3.8 Conception du projet

3.8.1 Importer des bibliothèques

Afin de réaliser cette tâche, nous écrivons un script Python en utilisant l'éditeur de code Visual Studio Code sur notre système local.

Le code ci-dessus est utilisé pour importer les bibliothèques nécessaires et préparer les données d'images pour l'entraînement et la validation d'un modèle de réseau de neurones convolutifs pour la classification d'images. Les bibliothèques importées incluent Keras, TensorFlow, NumPy, Seaborn, Matplotlib et d'autres bibliothèques Python courantes.

```

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

from keras.preprocessing.image import ImageDataGenerator
from keras.layers import Dense, Input, Dropout, Flatten, Conv2D
from keras.layers import BatchNormalization, Activation, MaxPooling2D
from keras.models import Model, Sequential
from keras.optimizers import Adam
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
from keras.utils import plot_model
from keras.applications import VGG16

from IPython.display import SVG, Image
from livelossplot import PlotLossesKerasTF
import tensorflow as tf
print("Tensorflow version:", tf.__version__)
✓ 0.3s Python

```

Figure 3. 10 – Importation des bibliothèques pour entraîner et valider notre modèle.

3.8.2 Création du modèle

Une fois les bibliothèques importées et le jeu de données obtenu, nous pouvons passer à la construction de notre modèle CNN, qui est une étape importante.

```

1 # Initialising the CNN
2 model = Sequential()
3
4 # 1 - Convolution
5 model.add(Conv2D(64,(3,3), padding='same', input_shape=(48, 48,1)))
6 model.add(BatchNormalization())
7 model.add(Activation('relu'))
8 model.add(MaxPooling2D(pool_size=(2, 2)))
9 model.add(Dropout(0.25))
10
11 # 2nd Convolution layer
12 model.add(Conv2D(128,(5,5), padding='same'))
13 model.add(BatchNormalization())
14 model.add(Activation('relu'))
15 model.add(MaxPooling2D(pool_size=(2, 2)))
16 model.add(Dropout(0.25))
17
18 # 3rd Convolution layer
19 model.add(Conv2D(512,(3,3), padding='same'))
20 model.add(BatchNormalization())
21 model.add(Activation('relu'))
22 model.add(MaxPooling2D(pool_size=(2, 2)))
23 model.add(Dropout(0.25))
24
25 # 4th Convolution layer
26 model.add(Conv2D(512,(3,3), padding='same'))
27 model.add(BatchNormalization())
28 model.add(Activation('relu'))
29 model.add(MaxPooling2D(pool_size=(2, 2)))
30 model.add(Dropout(0.25))
31
32 # Flattening
33 model.add(Flatten())
34
35 # Fully connected layer 1st layer
36 model.add(Dense(256))
37 model.add(BatchNormalization())
38 model.add(Activation('relu'))
39 model.add(Dropout(0.25))
40
41 # Fully connected layer 2nd layer
42 model.add(Dense(512))
43 model.add(BatchNormalization())
44 model.add(Activation('relu'))
45 model.add(Dropout(0.25))
46
47 model.add(Dense(7, activation='softmax'))
48
49 opt = Adam(lr=0.0005)
50 model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])
51 model.summary()

```

Figure 3. 11 - Code pour la création du modèle.

Le code présenté est une implémentation d'un réseau de neurones convolutifs (CNN) pour la classification d'images. Le modèle est construit en utilisant la bibliothèque Keras de Python, et est entraîné sur un ensemble de données d'entraînement pour prédire la catégorie de sept émotions différentes à partir d'images faciales.

```
-----  
Layer (type)                Output Shape                Param #  
-----  
conv2d (Conv2D)             (None, 46, 46, 32)         320  
conv2d_1 (Conv2D)           (None, 44, 44, 64)         18496  
max_pooling2d (MaxPooling2D) (None, 22, 22, 64)         0  
dropout (Dropout)           (None, 22, 22, 64)         0  
conv2d_2 (Conv2D)           (None, 20, 20, 128)        73856  
max_pooling2d_1 (MaxPooling2D) (None, 10, 10, 128)        0  
dropout_1 (Dropout)         (None, 10, 10, 128)        0  
conv2d_3 (Conv2D)           (None, 8, 8, 256)          295168  
max_pooling2d_2 (MaxPooling2D) (None, 4, 4, 256)          0  
dropout_2 (Dropout)         (None, 4, 4, 256)          0  
flatten (Flatten)           (None, 4096)                0  
dense (Dense)                (None, 512)                 2097664  
dropout_3 (Dropout)         (None, 512)                 0  
dense_1 (Dense)             (None, 7)                   3591  
-----  
Total params: 2,489,095  
Trainable params: 2,489,095  
Non-trainable params: 0  
-----
```

Figure 3. 12 - Résultat du modèle.

3.8.3 L'entraînement du modèle

Maintenant que nous avons réussi à construire l'architecture du modèle, il est temps de passer à la phase d'entraînement du modèle et d'évaluer les résultats obtenus.

Dans cette section nous nous intéressons aux résultats d'exécution de l'approche développée. Une fois que le modèle est formé, nous allons procéder à l'ajustement de certains paramètres :

Tableau 3. 2 - Tableau de paramètres utilisées.

Paramètres	Utilisations
Epoch	Le nombre d'itération dans notre base de données.
Loss	Le taux d'erreur.
Accuracy	Le taux de précision.
Val_loss	La valeur perdue.
Val_acc	La valeur de précision.

```

1  %%time
2
3  epochs = 15
4  steps_per_epoch = train_generator.n//train_generator.batch_size
5  validation_steps = validation_generator.n//validation_generator.batch_size
6
7  reduce_lr = ReduceLRonPlateau(monitor='val_loss', factor=0.1,
8                               patience=2, min_lr=0.0001, mode='auto')
9  checkpoint = ModelCheckpoint("model_weights.h5", monitor='val_accuracy',
10                             save_weights_only=True, mode='max', verbose=1)
11  callbacks = [PlotLossesKerasTF(), checkpoint, reduce_lr]
12
13  history = model.fit(
14      x=train_generator,
15      steps_per_epoch=steps_per_epoch,
16      epochs=epochs,
17      validation_data = validation_generator,
18      validation_steps = validation_steps,
19      callbacks=callbacks
20  )

```

Figure 3. 13 - Code pour l'entraînement du modèle.

Le code présenté est une partie de l'entraînement d'un modèle de réseau de neurones convolutifs (CNN) pour la classification d'images faciales en sept catégories d'émotions différentes, en utilisant la bibliothèque Keras de Python. Ce code définit les paramètres de l'entraînement, y compris le nombre d'epochs, les rappels (callbacks) pour réduire le taux d'apprentissage et enregistrer les poids du modèle, ainsi que le nombre de pas par epoch pour les ensembles de données d'entraînement et de validation. Le modèle est ensuite entraîné en utilisant la méthode fit() de Keras avec les paramètres définis et les résultats de l'entraînement sont enregistrés dans la variable history. Ce code permet de configurer et d'exécuter l'entraînement du modèle de classification d'émotions faciales.

- Une fois le modèle compilé, nous ajustons ensuite les données pour l'entraînement et la validation. Ici, nous prenons la taille du batch à 64 avec 15 itérations.

```

Epoch 1/15
897/897 [=====] - 1544s 2s/step - loss: 1.7940 - accuracy: 0.2561 - val_loss: 1.7140 - val_accuracy: 0.3023
Epoch 2/15
897/897 [=====] - 1318s 1s/step - loss: 1.6939 - accuracy: 0.3160 - val_loss: 1.5530 - val_accuracy: 0.3986
Epoch 3/15
897/897 [=====] - 1294s 1s/step - loss: 1.5925 - accuracy: 0.3751 - val_loss: 1.4364 - val_accuracy: 0.4410
Epoch 4/15
897/897 [=====] - 1252s 1s/step - loss: 1.5121 - accuracy: 0.4132 - val_loss: 1.3743 - val_accuracy: 0.4607
Epoch 5/15
897/897 [=====] - 1250s 1s/step - loss: 1.4495 - accuracy: 0.4399 - val_loss: 1.2932 - val_accuracy: 0.5081
Epoch 6/15
897/897 [=====] - 1237s 1s/step - loss: 1.4109 - accuracy: 0.4556 - val_loss: 1.2967 - val_accuracy: 0.4919
Epoch 7/15
897/897 [=====] - 1241s 1s/step - loss: 1.3787 - accuracy: 0.4719 - val_loss: 1.2306 - val_accuracy: 0.5269
Epoch 8/15
897/897 [=====] - 1337s 1s/step - loss: 1.3477 - accuracy: 0.4822 - val_loss: 1.2050 - val_accuracy: 0.5325
Epoch 9/15
897/897 [=====] - 1322s 1s/step - loss: 1.3300 - accuracy: 0.4892 - val_loss: 1.1871 - val_accuracy: 0.5481
Epoch 10/15
897/897 [=====] - 1246s 1s/step - loss: 1.3102 - accuracy: 0.4990 - val_loss: 1.1864 - val_accuracy: 0.5502
Epoch 11/15
897/897 [=====] - 1245s 1s/step - loss: 1.2914 - accuracy: 0.5080 - val_loss: 1.1721 - val_accuracy: 0.5525
Epoch 12/15
897/897 [=====] - 1341s 1s/step - loss: 1.2777 - accuracy: 0.5129 - val_loss: 1.1554 - val_accuracy: 0.5585
Epoch 13/15
897/897 [=====] - 1349s 2s/step - loss: 1.2635 - accuracy: 0.5197 - val_loss: 1.1788 - val_accuracy: 0.5540
Epoch 14/15
897/897 [=====] - 1276s 1s/step - loss: 1.2549 - accuracy: 0.5240 - val_loss: 1.1535 - val_accuracy: 0.5605
Epoch 15/15
897/897 [=====] - 1247s 1s/step - loss: 1.2416 - accuracy: 0.5287 - val_loss: 1.1483 - val_accuracy: 0.5614
    
```

Figure 3. 14 - Résultat de l’entraînement du modèle.

3.8.4 Résultats expérimentaux et analyse de performances

Le tableau ci-dessous montre que le réseau a été entraîné pendant seulement 15 epochs et a atteint une précision de 56,14% avec une faible perte qui suit la perte d'entraînement.

Tableau 3. 3 - Résultats de l’apprentissage.

Epoch	Perte d'apprentissage	Précision d'apprentissage	Valeur perdu	Valeur de précision
1	1.7940	0.2561	1.7140	0.3023
2	7.6939	0.3160	1.5530	0.3986
3	1.5925	0.3751	1.4364	0.4410
4	1.5121	0.4132	1.3743	0.4607
5	1.4495	0.4399	1.2932	0.5081
6	1.4109	0.4556	1.2967	0.4919
7	1.3787	0.4719	1.2306	0.5269
8	1.3477	0.4822	1.2050	0.5325
9	1.3300	0.4892	1.1871	0.5481
10	1.3102	0.4990	1.1864	0.5502
11	1.2914	0.5080	1.1721	0.5525
12	1.2777	0.5129	1.1554	0.5595
13	1.2635	0.5197	1.1788	0.5540
14	1.2549	0.5240	1.1535	0.5605
15	1.2416	0.5287	1.1483	0.5614

Après avoir analysé les résultats obtenus, on peut constater que la précision de l'apprentissage et des tests augmente avec le nombre d'epochs, ce qui indique que le modèle apprend plus d'informations à chaque epoch. Si la précision diminue, cela signifie que le modèle a besoin de plus d'informations pour apprendre, et donc le nombre d'epochs doit être augmenté, et vice versa. De même, l'erreur d'apprentissage et de validation diminue avec le nombre d'epochs. Ci-dessous, nous présentons les courbes de précision et de perte qui illustrent l'entraînement et les tests de notre modèle.

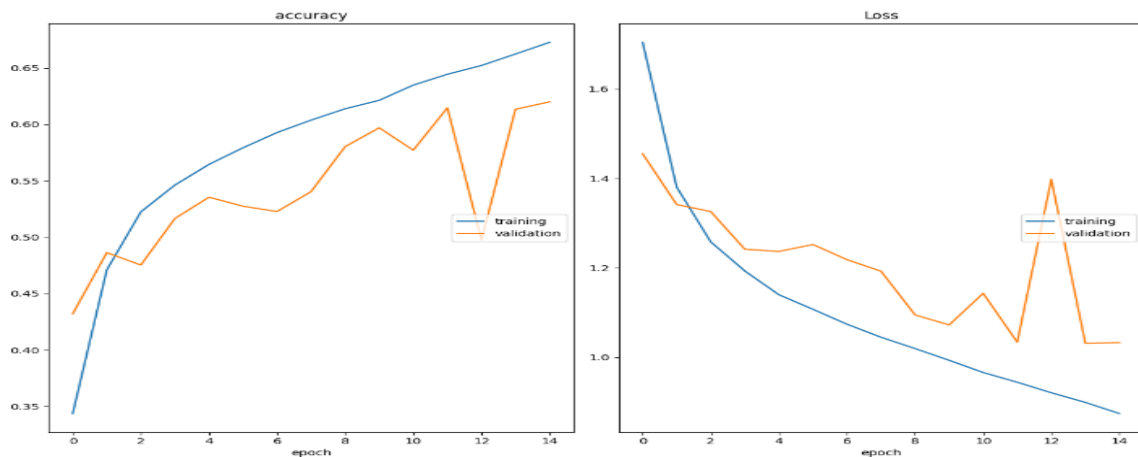


Figure 3. 15 - Les courbes de précision (accuracy) et de perte (loss) générée du modèle.

La figure 3.15 représente les courbes de précision (accuracy) et de perte (loss) générée du modèle, la courbe sur la gauche représente la précision de l'ensemble d'entraînement en bleu et la précision de l'ensemble de test en orange en fonction du nombre d'epoch. D'un autre côté la courbe sur la droite représente la courbe de perte de l'ensemble d'entraînement en bleu et la perte de l'ensemble de test en orange en fonction du nombre d'epoch.

D'après les deux figures, la précision de l'apprentissage et de la validation augmente avec le nombre d'epoch. Ceci reflète qu'à chaque epoch le modèle apprend plus d'informations. De même, nous observons que l'erreur d'apprentissage et de la validation diminue avec le nombre d'epoch.

3.9 Tester le modèle pour la reconnaissance d'expressions faciales

3.9.1 Utilisation du modèle avec OpenCV et Webcam en temps réel

Pour la mise en œuvre de notre modèle pour la reconnaissance des expressions faciales en temps réel, nous avons créé un autre code, à l'aide d'OpenCV et d'une webcam.

- Après avoir exécuté le code, nous avons obtenu les résultats comme suit :

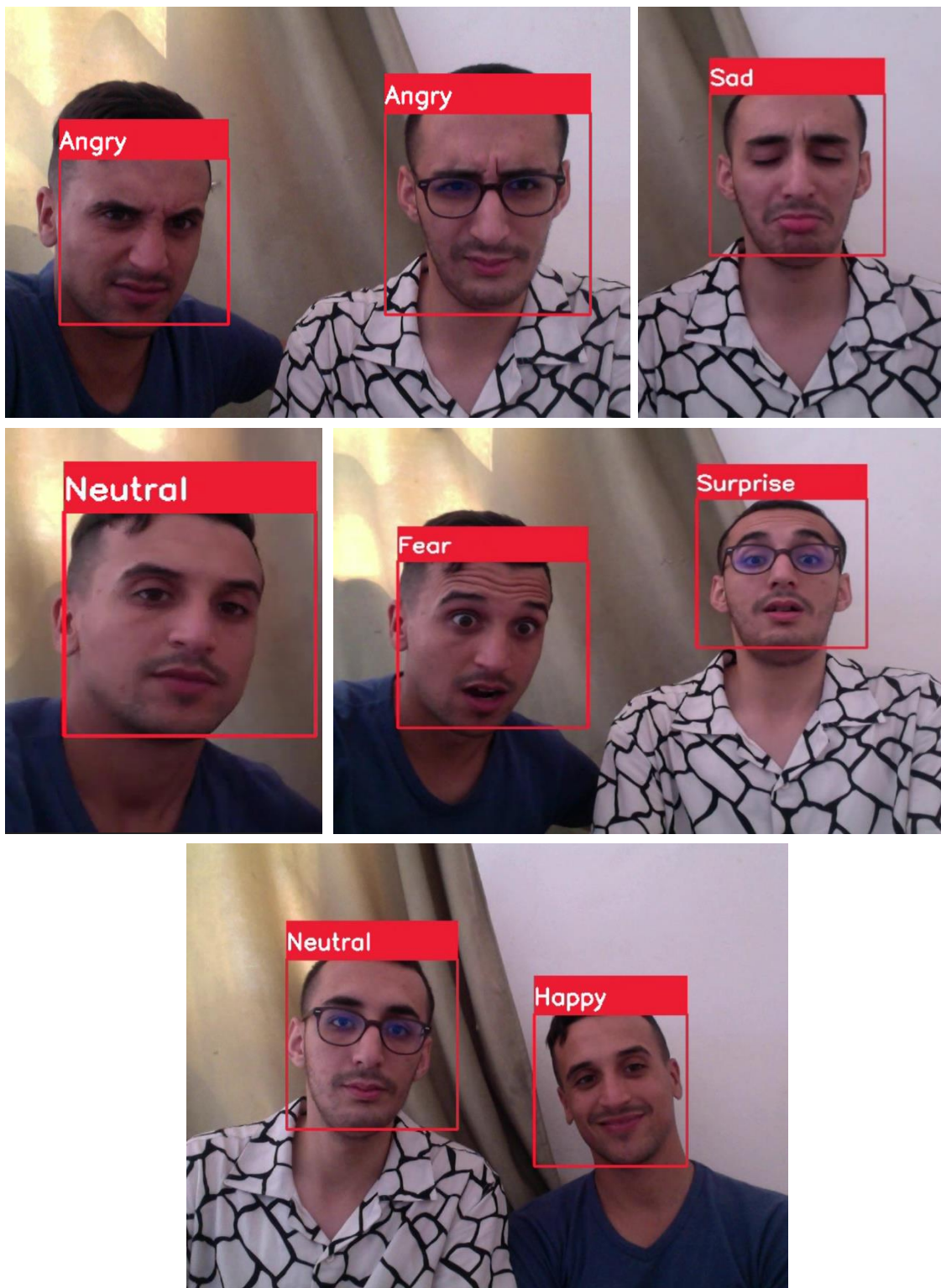


Figure 3. 16 - La visualisation de nos images qui ont été traitées.

3.9.2 Utilisation du modèle en vidéo en format MP4

Une fois que les poids du modèle ont été importés, nous avons également importé un fichier vidéo au format .mp4 appelé "vidéo.mp4". Cela a ouvert une nouvelle fenêtre où la vidéo est diffusée. Nous avons effectué le même travail qu'auparavant, c'est-à-dire traiter chaque trame de la vidéo en temps réel, comme nous le faisons avec la webcam, jusqu'à la fin de la vidéo.

- Après avoir exécuté le code, nous avons obtenu les résultats comme suit :



Figure 3. 17 - Résultats d'une vidéo externe.

3.9.3 Utilisation du modèle en image en format PNG

Une fois que le fichier "image.png" a été importé, une nouvelle fenêtre s'ouvrira affichant la photo avec les visages détectés et les émotions associées pour chaque visage (pour une seule image).

- Après avoir exécuté le code, nous avons obtenu les résultats comme suit :



Figure 3. 18 - Résultats de quelques images dans une seule image externe.

3.10 Conclusion

Le dernier chapitre de notre travail présente un modèle de classification des expressions faciales basé sur les techniques de Deep Learning, avec une architecture de réseau de neurones convolutifs (CNN) que nous avons proposée. Nous avons démontré que notre modèle est généralement efficace, mais qu'il peut être encore amélioré en augmentant le nombre d'itérations et en l'entraînant sur une base de données plus large.

Conclusion et Perspectives

Ce travail est un mémoire de Master réalisé à l'université de Bouira. Notre projet de fin d'étude sur la reconnaissance des émotions faciales en utilisant le deep learning a été une expérience enrichissante et passionnante. Grâce à l'utilisation de réseaux neurones convolutifs (CNN), nous avons pu développer un modèle capable de classifier avec précision différentes expressions faciales incluant la peur, la neutralité, la tristesse, la surprise, la colère, la joie, le dégoût.

Nous avons exploré les notions fondamentales de l'apprentissage profond et nous avons notamment mis en évidence le rôle des réseaux de neurones convolutifs dans le traitement des images et comment les CNN peuvent extraire automatiquement les caractéristiques importantes des images pour effectuer des classifications précises, en nous intéressant en particulier aux différentes couches qui les composent. En utilisant la base de données FER2013, nous avons entraîné notre modèle et avons surmonté le défi du sur apprentissage pour obtenir des performances fiables.

Cependant, il convient de noter que le succès de notre projet a été en partie tributaire des ressources de traitement disponibles. Avec des unités de traitement plus avancées telles que les GPU, nous aurions pu bénéficier de performances supérieures et d'un entraînement plus rapide. Au future, nous envisageons les perspectives de recherche suivantes :

- Augmenter les données d'entraînement en ajoutant des données supplémentaires à la base de données FER2013. Et nous prévoyons de soumettre le modèle à des tests sur des bases de données plus vastes et volumineuses.
- Ajouter de nouvelles couches afin d'évaluer leur impact sur les performances du modèle.
- Etendre le modèle pour inclure davantage de classes d'émotions secondaires, afin de pouvoir reconnaître et interpréter les micro-expressions faciales.

En conclusion, notre projet a démontré le potentiel du Deep Learning pour la reconnaissance des émotions faciales. Ce domaine de recherche offre de nombreuses possibilités d'application dans des domaines tels que l'analyse du comportement humain et les interfaces homme-machine. Notre travail ouvre la voie à des développements futurs dans ce domaine, et nous sommes fiers d'avoir contribué à cette avancée technologique.

Bibliographie

1. Russell, J. A. (2003). Core affect and the psychological construction of emotion. *Psychological Review*, 110(1), 145-172.
2. Khalf, F. (2018). Reconnaissance automatique des émotions par données multimodales : expressions faciales et des signaux physiologiques.
3. Hamdi, H. (2012). Plate-forme multimodale pour la reconnaissance d'émotions via l'analyse de signaux physiologiques. French.
4. Ekman, P., & Friesen, W. V. (1978). Facial action coding system: A technique for the measurement of facial movement. Consulting Psychologists Press.
5. Winkielman, P., & Cacioppo, J. T. (2001). Mind at ease puts a smile on the face: Psychophysiological evidence that processing facilitation elicits positive affect.
6. Ouakkaf, H., & Berkane, M. (2017). Reconnaissance automatique des expressions faciales par support vector machine. Université Larbi Ben M'hidi Oum El Bouaghi, faculté des sciences exactes et des sciences de la nature et de la vie, département informatique, Algérie.
7. Ekman, P., & Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2), 124-129
8. NOUIOUA, N., & MADI, M. (2020). Reconnaissance Automatique des Expressions Faciales. Faculté de Technologie, Département de Génie Electrique, Université Abderrahmane Mira de Bejaia, Algérie.
9. Rosenberg, E., & Ekman, P. (1997). What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS). Oxford University Press, USA.
10. Llatas Spiers, D. (2016). Facial emotion detection using deep learning.
11. Bettadapura, V. (2012). Face expression recognition and analysis: the state of the art. arXiv preprint arXiv:1203.6722.
12. Ghanem, K. (2010). Reconnaissance des expressions faciales à base d'informations vidéo ; estimation de l'intensité des expressions faciales.
13. Ekman, P. and Friesen, W. V. (2003). Unmasking the face : A guide to recognizing emotions from facial clues. Ishk.
14. Valstar, M. and Pantic, M. (2006). Fully automatic facial action unit detection and temporal analysis. In 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), pages 149–149. IEEE
15. Ghediri, I. R., & Semri, K. (2021). La Reconnaissance Des émotions de base Par Les

réseaux de neurones : Application de Deep Learning. Faculté des sciences exactes et sciences de la nature et de la vie, Département d'informatique, Université Oum El Bouaghi, Algérie.

16. Fasel, B. and Luetten, J. (2003). Automatic facial expression analysis : a survey. *Pattern recognition*, 36(1) : 259–275.
17. Barrett, L. F., Adolphs, R., Marsella, S., Martinez, A. M., & Pollak, S. D. (2019). Emotional Expressions [Reconsidered] : Challenges to Inferring Emotion From Human Facial Movements. *Psychological Science in the Public Interest*, 20(1), 1-68.
18. Aizenberg, I., Aizenberg, N. N., & Vandewalle, J. P. L. (2013). Multi-valued and universal binary neurons: theory, learning and applications. Springer Science & Business Media.
19. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning* (Vol. 1). MIT press.
20. Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8).
21. Moualek, D. Y. (2017). *Deep Learning pour la classification des images*. (Thèse de doctorat). Université de Tlemcen, Algérie.
22. Tiwari, A. (2019). Why Deep Learning Is Needed Over Traditional Machine Learning. *Towards Data Science* : <https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063> (Consulté le 05 Avril 2023).
23. NACER, F. (2019). *Reconnaissance d'expression faciale à partir d'un visage réel*. Département informatique, Université 08 mai 1945, Guelma, Algérie.
24. Bloechle, J. L. (2003). *Réseau de Neurones Artificiels pour la classification des fontes Arabes et la distinction entre la langue Arabe et les langues Latines*. Mémoire de maîtrise, Université de Fribourg, Département d'informatique, Fribourg, Suisse.
25. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324
26. Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818–833). Springer.
27. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
28. Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 160–167).
29. Hinton, G. E. (2012). A practical guide to training restricted Boltzmann machines. In *Neural networks : Tricks of the trade* (pp. 599–619). Springer.
30. Jordan, M. I., & Rumelhart, D. E. (1986). *Parallel distributed processing, Explorations in the microstructure of cognition. Volume 1 : Foundations*, 318-362.
31. DIALLO N. A. D., (2019). *La reconnaissance des expressions faciales*.

32. Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Advances in neural information processing systems (pp. 841–848).
33. Jedha. (2018). Les applications du Deep Learning. Medium : <https://medium.com/jedha/les-applications-du-deep-learning-d674483d44af>. (Consulté le 14 Avril 2023)
34. Noura, H., & Khelifa, N. (2018). Classification des textures par les réseaux de neurones convolutifs. Mémoire de master, Faculté du Génie Électrique et d'Informatique, Département d'Automatique, Université Mouloud Mammeri de Tizi-Ouzou, Algérie.
35. Merzougui, Dr. (2021). Support de cours deep learning : Chapitre 3 : Réseau de neurones convolutionnel CNN. Master 2 SDS, Université Batna 2, Algérie.
36. CHETTOUH, H. (2020). Montage d'un Système de Reconnaissance des Expressions Faciales avec le Deep Learning. Mémoire de master, Faculté des Sciences Exactes, Département d'informatique, Université Mohamed Khider – Biskra, Algérie.
37. DataScientest. Architecture d'un Convolutional Neural Network – CNN : <https://datascientest.com/convolutional-neuralnetwork> (Consulté le 20 Avril 2023)
38. Mokri, M. Z. (2017). Classification des images avec les réseaux de neurones convolutionnels. Mémoire de Master. Département d'informatique, Université de Tlemcen, Algérie.
39. Brahim, B. M. (2017). L'apprentissage profond pour la classification et la recherche d'images par le contenu. Département d'informatique, Université de Ouargla, Algérie.
40. Krizhevsky. A., Sutskever, I. and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105.
41. ENTWANNE (2022), Zeste de savoir, un zeste de python.
42. Amazon Web Services. Python on AWS. à partir de <https://aws.amazon.com/fr/what-is/python/> (Consulté le 27 April 2023)
43. Axopen. (2019, septembre 25). OpenCV, c'est quoi ? [Blog post]. à partir de <https://www.axopen.com/blog/2019/09/open-cv-cest-quoi/> (Consulté le 27 April 2023).
44. CommentOuvrir. Tout sur NumPy à partir <https://commentouvrir.com/definitions/tout-sur-numpy/> (Consulté le 27 April 2023) .
45. AstrologyPage. Matplotlib : <https://fr.theastrologypage.com/matplotlib> (Consulté le 27 April 2023)