



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITÉ ABDELHAMID IBN BADIS - MOSTAGANEM



Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et informatique
Filière : Informatique

MEMOIRE DE MASTER EN INFORMATIQUE
Option : **Ingénierie des Systèmes d'Information**

THEME :

« La reconnaissance des émotions faciales :
Application à la surveillance médicale »

Etudiants : « **MEKDAM REDA** »

« **YAYOUCHE FARES MEHDI** »

Encadrant : **Henni Fouad**

Co-encadrant : **Belmehidi Dhehbi**

Examineur : **Benkerdagh Saliha**

Président : **Habib Zahmani Mohamed**

Année Universitaire 2023-2024

Remerciements

Tout d'abord, nous remercions Allah, pour nous avoir donné la force, la patience et la sagesse nécessaires pour mener à bien cette mémoire.

Nous remercions tout particulièrement notre encadrant, Mr. Henni Fouad, pour son soutien indéfectible, ses conseils avisés, et sa patience tout au long de ce projet. Sa rigueur scientifique et ses encouragements nous ont été d'une aide précieuse pour mener à bien cette recherche.

Nous souhaitons également exprimer notre reconnaissance à notre co-encadrant, Mr. Belmehidi Dhehbi, pour ses précieuses orientations et son expertise, qui ont grandement enrichi notre travail.

Nous remercions les membres du jury pour avoir accepté d'évaluer notre travail et d'avoir consacré leur temps pour évaluer cette mémoire.

Nous sommes également reconnaissants envers l'ensemble des enseignants et le personnel de l'Université ABDELHAMID IBN BADIS pour leur soutien académique et administratif, ainsi qu'à nos collègues étudiants pour leur camaraderie et leurs échanges enrichissants.

Nous exprimons notre sincère gratitude envers nos parents, nos familles, et toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce mémoire. Leur soutien constant, leurs encouragements et leur compréhension ont été inestimables tout au long de nos études.

Merci à tous.

Dédicaces

On dédie ce travail à nos parents, dont l'amour inconditionnel, le soutien indéfectible et les sacrifices ont été les fondements de notre éducation et de notre réussite. Leurs encouragements constants et leur foi en notre potentiel nous ont inspirés à poursuivre nos rêves et à atteindre nos objectifs.

On dédie également cette mémoire à nos frères et sœurs, qui ont partagé avec nous chaque étape de ce voyage. Leur soutien, leurs conseils ont enrichi nos vies de manière inestimable. À nos frères et sœurs, nous exprimons notre gratitude pour leur présence constante et leur soutien inconditionnel.

On souhaite également exprimer notre reconnaissance à nos autres membres de famille, proches et éloignés, qui ont toujours été là pour nous encourager et nous soutenir dans nos projets.

Résumé

La reconnaissance des émotions faciales représente une avancée majeure dans divers domaines, y compris celui de la surveillance médicale. Ce projet se concentre sur la conception et l'implémentation d'un système de reconnaissance des émotions faciales, mettant particulièrement l'accent sur son intégration dans le domaine médical. L'objectif principal est de créer un modèle efficace capable d'évaluer l'état émotionnel des patients en temps réel. Nous avons pu concevoir modèle CNN et l'entraîner sur une base de données améliorée, basée sur Fer2013. Nos résultats ont montré une précision d'environ 68 % sur les données de test, ce qui indique une performance prometteuse pour notre approche. En utilisant ce modèle, notre application vise à fournir aux professionnels de la santé un outil précieux pour détecter et analyser les expressions faciales des patients de manière continue, améliorant ainsi la qualité des soins et la réactivité médicale.

Mots clé : Les émotions faciales, La reconnaissance des émotions faciales, Réseaux de neurones convolutifs, CNN, FER2013

Abstract

Facial emotion recognition represents a significant advancement in various fields, including medical monitoring. This project focuses on the design and implementation of a facial emotion recognition system, with a particular emphasis on its integration into the medical domain. The primary objective is to create an efficient model capable of assessing patients' emotional states in real-time. We were able to design CNN model and train it on an enhanced database based on Fer2013. Our results showed an accuracy of approximately 68% on test data, indicating promising performance for our approach. Using this model, our application aims to provide healthcare professionals with a valuable tool to continuously detect and analyze patients' facial expressions, thereby improving the quality of care and medical responsiveness.

Keywords: Facial emotions, Facial emotion recognition, Convolutional Neural Networks, CNN, FER2013

الملخص

يمثل التعرف على التعابير الوجهية تقدماً كبيراً في مجالات متعددة، بما في ذلك المراقبة الطبية. تركز هذه الأطروحة على تصميم وتنفيذ نظام للتعرف على التعابير الوجهية، مع التركيز بشكل خاص على دمجها في المجال الطبي. الهدف الرئيسي هو إنشاء نموذج فعال قادر على تقييم الحالات العاطفية للمرضى على المباشر. تمكنا من تصميم نموذج CNN وتدريبه على قاعدة بيانات محسنة بناءً على Fer2013. أظهرت نتائجنا دقة تبلغ حوالي 68% على بيانات الاختبار، مما يشير إلى أداء مشجع لنهجنا. باستخدام هذا النموذج، يهدف تطبيقنا إلى تزويد المهنيين الطبيين بأداة قيمة للكشف وتحليل تعابير الوجه للمرضى بشكل مستمر، مما يعزز بذلك جودة الرعاية والاستجابة الطبية.

الكلمات المفتاحية: التعرف على العواطف الوجهية، الشبكات العصبية التلافيفية، CNN،

FER2013

Liste des figures

Figure N°	Titre de la figure	Page
Figure 1.1	Exemple sur la variation d'éclairage sur le visage	10
Figure 1.2	Exemple sur la variation de pose	10
Figure 1.3	Exemple d'expressions faciales	11
Figure 1.4	Les émotions universelles	14
Figure 1.5	Liste des actions units relatives aux 6 expressions faciales	15
Figure 1.6	Modèle MPEG4 l'ensemble d'attributs faciaux	16
Figure 1.7	l'architecture d'un système de reconnaissance des émotions	17
Figure 2.1	Neurone biologique (a) et neurone artificiel (b)	24
Figure 2.2	L'architecture typique d'un CNN	26
Figure 2.3	Schéma de déplacement de filtre	27
Figure 2.4	Différents types de convolutions	29
Figure 2.5	Processus de Max-Pooling	30
Figure 2.6	La fonction d'activation ReLU	31
Figure 3.1	Echantillon de la base de données	41
Figure 3.2	Code de prétraitement et Augmentation des Données	42
Figure 3.3	Code de la création du modèle	44
Figure 3.4	Résultat de notre modèle	45
Figure 3.5	Code de l'entraînement avec SGD	47
Figure 3.6	Le résultat de l'entraînement avec SGD	47
Figure 3.7	Code de l'entraînement avec RMSprop	48
Figure 3.8	Résultat de l'entraînement avec RMSprop	48

Figure 3.9	Courbes de précision (accuracy) et de perte (loss) SGD	49
Figure 3.10	Matrice de confusion SGD	50
Figure 3.11	Courbes de perte (loss) et de précision (accuracy) RMSprop	51
Figure 3.12	Matrice de confusion RMSprop	52
Figure 3.13	Interface initiale de l'application	53
Figure 3.14	Test d'émotion capturée	54
Figure 3.15	Interface historique	55
Figure 3.16	Exemple de message reçu	57

Liste des abréviations

Abréviation	Expression Complète	Page
SVM	Support Vector Machine	8
FACS	Facial Action Coding System	14
FFP	Facial Feature Points	15
FER	Face Expression Recognition	17
ROI	Region of Interest	17
ASM	Modèle de forme active	18
AAM	Modèles d'apparence active	18
LBP	Motif binaire local	18
LPQ	Quantification de phase locale	18
HOG	Histogramme de gradient orienté	18
NN	Neural Networks	19
LDA	Linear discriminant analysis	19
HMM	Hidden Markov Model	19
MP	Maladie de Parkinson	20
MA	Maladie d'Alzheimer	20
CNN	Convolutional Neural Network	23
RNN	Recurrent Neural Network	25
ReLU	Rectified linear unit	26
VGG	Visual Geometry Group	32
GPU	Graphics processing unit	36
CPU	Central processing unit	36

Table des matières

INTRODUCTION GENERALE	4
CHAPITRE 1 LA RECONNAISSANCE DES EMOTIONS FACIALES	6
1.1 INTRODUCTION	6
1.2 LA DETECTION DE VISAGE	6
1.2.1 Méthodes de détection	6
1.2.1.1 Approches basée sur les caractéristiques invariantes.....	6
1.2.1.2 Approches basée sur les connaissances.....	7
1.2.1.3 Approches basée sur la mise en correspondance	7
1.2.1.4 Approches basée sur l'apparence	8
1.2.2 Difficultés de la détection de visage	9
1.2.2.1 Variation d'éclairage	9
1.2.2.2 Variation de pose	10
1.2.2.3 Les expressions faciales	10
1.2.2.4 Occlusion.....	11
1.3 EXPRESSION FACIALE ET EMOTION.....	11
1.3.1 Emotion.....	12
1.3.2 Expression faciale	12
1.3.3 Expressions faciales universelles.....	12
1.3.4 Les Systèmes de codification.....	14
1.3.4.1 Système de Codification des Actions Faciales FACS	14
1.3.4.2 MPEG-4	15
1.3.5 Système de reconnaissance d'expression faciale.....	16
1.3.5.1 Prétraitement.....	17
1.3.5.2 Extraction des caractéristiques	17
1.3.5.3 Classification	18
1.3.6 Domaine d'application.....	19
1.3.7 La reconnaissance des émotions dans le domaine médical.....	20
1.3.7.1 Surveillance de la douleur:.....	20
1.3.7.2 Surveillance de la santé mentale:	20
1.3.7.3 Surveillance des troubles du sommeil:	21
1.3.7.4 Surveillance des signes vitaux:	21

1.4	OBJECTIF	21
1.5	CONCLUSION.....	21
CHAPITRE 2 RESEAUX DE NEURONES CONVOLUTIFS.....		23
2.1	INTRODUCTION	23
2.2	DEEP LEARNING	23
2.2.1	<i>Les différents algorithmes de Deep Learning.....</i>	<i>24</i>
2.3	RESEAUX DE NEURONES CONVOLUTIFS.....	25
2.3.1	<i>Les différentes couches d'un CNN.....</i>	<i>26</i>
2.3.1.1	La couche de convolution	26
2.3.1.2	Les différentes convolutions	28
2.3.1.3	La couche de pooling	29
2.3.1.4	La couche ReLU	30
2.3.1.5	La couche entièrement connectée.....	31
2.3.2	<i>Avantages des CNN.....</i>	<i>31</i>
2.3.3	<i>Quelques réseaux convolutifs pré-entraînés connus.....</i>	<i>32</i>
2.4	CONCLUSION.....	33
CHAPITRE 3 CONCEPTION ET IMPLEMENTATION		35
3.1	INTRODUCTION.....	35
3.2	ENVIRONNEMENT DE TRAVAIL.....	35
3.2.1	<i>Environnement matériel</i>	<i>35</i>
3.2.2	<i>Environnement de développement.....</i>	<i>36</i>
3.2.2.1	Visual Studio Code	36
3.2.2.2	Google Colab.....	36
3.2.2.3	Pourquoi les GPU ?.....	36
3.2.2.4	Cloudinary.....	37
3.2.2.5	Twilio.....	37
3.2.2.6	Python.....	38
3.2.3	<i>Bibliothèques utilisées</i>	<i>38</i>
3.2.3.1	OpenCV (Open Source Computer Vision Library).....	38
3.2.3.2	NumPy.....	38
3.2.3.3	Pandas.....	39
3.2.3.4	Tensorflow	39
3.2.3.5	Keras	39
3.2.3.6	Matplotlib	40

3.2.3.7	Tkinter	40
3.3	BASE DE DONNEES.....	40
3.4	IMPLEMENTATION ET RESULTATS	42
3.4.1	<i>Prétraitement et Augmentation des Données</i>	42
3.4.2	<i>Création du modèle</i>	43
3.4.3	<i>Entraînement du modèle</i>	45
3.4.4	<i>Résultats et analyse</i>	49
3.4.4.1	Première expérience	49
3.4.4.2	Deuxième expérience	50
3.5	PRESENTATION DE L'APPLICATION.....	52
3.5.1	<i>Système de notification</i>	56
3.6	CONCLUSION.....	57
	CONCLUSION GENERALE	58
	BIBLIOGRAPHIE	60

Introduction Générale

Les expressions faciales sont des signaux visuels révélateurs de l'état émotionnel et psychologique d'une personne. Leur détection et analyse sont importantes dans divers domaines comme la formation assistée par ordinateur, le traitement d'images et la reconnaissance des formes. Cependant, cette tâche est complexe en raison des variations individuelles des expressions faciales. La complexité et la subtilité des expressions rendent leur détection et interprétation difficiles, nécessitant des techniques avancées d'analyse et de reconnaissance automatique.

La reconnaissance des émotions faciales va plus loin en analysant les expressions faciales pour identifier les états émotionnels. En utilisant divers algorithmes, elle détecte et analyse les micro-expressions faciales afin de déterminer des émotions telles que la joie, la tristesse, la colère, la surprise, la peur et le dégoût. Ses applications s'étendent à des domaines variés tels que le marketing, les interactions homme-machine, la sécurité, l'éducation et la médecine.

Dans le domaine de la surveillance médicale, la reconnaissance des émotions faciales présente des avantages significatifs pour améliorer la qualité des soins et la sécurité des patients. Intégrée aux systèmes de surveillance médicale, elle permet aux professionnels de détecter et d'évaluer les signes émotionnels des patients en temps réel, ce qui est crucial pour la prise en charge des situations médicales critiques.

Notre objectif est de concevoir un système de reconnaissance des émotions faciales en utilisant des réseaux de neurones convolutifs (CNN) et de proposer notre propre modèle. Nous mettons l'accent sur son application dans la surveillance médicale. Ce système sera conçu pour détecter et analyser les expressions faciales des patients, fournissant ainsi des informations cruciales pour évaluer leur état émotionnel en temps réel. En intégrant cette technologie de

pointe dans les pratiques de surveillance médicale, nous visons à améliorer la qualité des soins et à favoriser une approche plus proactive et personnalisée de la prise en charge des patients.

Ce mémoire est structuré en trois chapitres :

Dans le premier chapitre, nous introduisons la détection de visage et la reconnaissance des émotions faciales, explorant les méthodes de détection et les défis rencontrés. Nous examinons ensuite les concepts d'expression faciale et d'émotion, ainsi que leur utilisation dans le domaine médical. Enfin, nous définissons les objectifs du mémoire.

Dans le deuxième chapitre, nous plongeons dans le domaine du deep learning en mettant l'accent sur les réseaux de neurones convolutifs (CNN). Nous explorons en détail les principes fondamentaux des CNN, notamment leur architecture en couches et leur fonctionnement. En outre, nous concluons ce chapitre en mettant en avant quelques réseaux convolutifs pré-entraînés bien connus.

Dans le troisième et dernier chapitre, nous détaillons l'environnement de travail et la base de données utilisés dans notre implémentation, suivis des étapes de prétraitement, création et entraînement du modèle, avec une analyse des résultats. Enfin, nous présentons l'application développée pour pouvoir tester notre modèle.

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

Chapitre 1

La reconnaissance des émotions faciales

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, affectent nos pensées, nos actions et nos interactions sociales. Leur impact peut être significatif dans divers domaines, tels que la psychologie, la santé mentale, et bien plus encore.

Dans ce chapitre, nous présenterons un état de l'art sur les techniques de reconnaissance d'expressions faciales.

1.2 La détection de visage

La détection de visage est une tâche fondamentale en vision par ordinateur qui vise à localiser et à identifier les visages humains dans des images ou des vidéos. C'est une étape essentielle dans de nombreuses applications, notamment la photographie, la surveillance, la sécurité, l'interaction homme-machine et la réalité augmentée.

1.2.1 Méthodes de détection

Les méthodes de détection de visages ont été classées par Yang en quatre approches [1]:

1.2.1.1 Approches basée sur les caractéristiques invariantes

Ce type de méthodes a pour objectif de trouver les caractéristiques structurelles même si le visage est dans différentes positions, conditions lumineuses ou changement d'angle de

vue. Le problème avec cette méthode est que la qualité des images peut être sévèrement diminuée à cause de l'illumination, du bruit ou l'occlusion ce qui empêche l'algorithme de fonctionner correctement. Cependant, il existe plusieurs propriétés ou caractéristiques invariantes du visage dont les principales sont les suivantes [1] :

- La couleur de peau
- Les caractéristiques faciales
- La texture

1.2.1.2 Approches basée sur les connaissances

Cette méthode repose sur des règles qui identifient les éléments clés et les caractéristiques des visages humains. Ces règles sont généralement établies en se basant sur la relation entre les différentes caractéristiques faciales.

Ce processus rencontre des difficultés lorsqu'il s'agit de traduire les connaissances sur les visages humains en règles clairement définies. Cette difficulté peut entraîner des erreurs de détection et limiter l'efficacité globale du système [1].

1.2.1.3 Approches basée sur la mise en correspondance

Des modèles caractéristiques d'un visage entier ou d'une sous-partie de visage (bouche, œil, nez) sont créés. La localisation se fait par corrélation de ces modèles avec les candidats. On trouve deux techniques [2]:

- **Modèle prédéfini** : Cette technique est utilisée pour classer des objets, elle est très intéressante pour la détection de visage vu sa facilité d'application. Le principe de cette méthode est basé sur une comparaison effectuée entre une image quelconque et un modèle prédéfini, dont le but est de calculer la corrélation pour aboutir à une décision par oui/non. La correspondance est faite pixel par pixel. Cette méthode est utilisée par Sakai et al [2].

- **Modèle déformable** : Cette approche est utilisée dans le but de modéliser les caractéristiques faciales qui s'adaptent élastiquement par rapport au modèle du visage présent. Dans cette méthode, les caractéristiques faciales sont décrites par des gabarits paramétrés. La meilleure adaptation du modèle élastique est de trouver la fonction énergétique en minimisant les paramètres. Cette méthode a été utilisée par Kanade et al [2].

1.2.1.4 Approches basée sur l'apparence

Cette méthode vise à identifier les caractéristiques significatives des visages et des non-visages en utilisant des techniques d'analyse statistique et d'apprentissage. Ces techniques sont organisées à travers des modèles de distribution ou des fonctions discriminantes. La classification entre visages et non-visages est réalisée par une variable aléatoire x , dérivée soit d'une image soit d'un vecteur de caractéristiques. Parmi les méthodes utilisées dans ce contexte, on peut citer : [1]

- **Eigen faces** : Turk et Pentland ont été les premiers à développer la méthode Eigenface en 1991, Eigenface qui sera ensuite l'une des méthodes les plus connues de la détection de visage. Le principe de cette méthode est de projeter une image dans un espace puis on calcule la distance euclidienne entre l'image originale et sa projection. L'avantage de cette méthode est qu'elle donne des résultats très encourageants, mais le calcul prend beaucoup de temps [1].
- **Support Vector Machine (SVM)** : La méthode SVM fait partie de la gamme des méthodes à apprentissage statistique, proposé par V.Vapnik en 1995. Le système de détection est conçu avec des images de formation et aucune connaissance préalable n'est requise sur l'image [1].
- **Détecteur de Viola et Jones** : La méthode Viola-Jones consiste à utiliser les descripteurs de Haar qui sont constitués de deux rectangles adjacents, l'un noir et l'autre blanc. Ces descripteurs sont superposés aux différentes régions de l'image à la recherche d'une zone de forte corrélation [3].

- **Réseaux de neurones :** Les réseaux de neurones sont utilisés pour classifier les pixels de l'image en tant que visage ou non-visage. Dans toute utilisation de réseaux de neurones, il faut définir une topologie du réseau. Cette dernière est déterminée par des tests successifs et il n'y a aucune méthode standard à suivre pour définir la meilleure [1].

1.2.2 Difficultés de la détection de visage

La détection de visage chez les êtres humains est un processus complexe et fascinant qui est effectué naturellement et instinctivement par le cerveau, Alors que la détection et l'identification des visages dans une scène sont des tâches que les êtres humains réalisent sans difficulté notable, la conception d'un système automatique capable d'accomplir ces mêmes tâches constitue un défi considérable.

Les principales difficultés sont la robustesse aux différentes identités, poses du visage, expressions faciales et aux variations d'illumination. La sortie d'un détecteur de visage indique le nombre de visages présents dans l'image. De plus, la plupart des détecteurs de visage actuels sont aussi des localisateurs de visages : ils renvoient une localisation des visages détectés (une boîte englobant par exemple) [4].

1.2.2.1 Variation d'éclairage

La variation de l'apparence d'un visage dans une image est considérablement influencée par l'illumination de la scène au moment de la capture (Figure 1.1). Les variations d'éclairage rendent la tâche de reconnaissance de visage difficile. En effet, le changement d'apparence d'un visage dû à l'illumination se révèle parfois plus critique que la différence physique entre les individus, et peut entraîner une mauvaise classification des images d'entrée. Ceci a été expérimentalement observé dans Adini et al [1].



Figure 1.1 Exemple sur la variation d'éclairage sur le visage

1.2.2.2 Variation de pose

Les variations de pose représentent un défi majeur pour la détection de visages. En effet, les visages peuvent apparaître dans une multitude de poses différentes, allant de la face de profil à divers angles. Ces variations de pose entraînent des déformations géométriques qui rendent souvent difficile l'alignement et la détection précise des visages. Les méthodes de détection doivent être capables de modéliser efficacement ces variations de manière non linéaire pour assurer une détection fiable dans une variété de situations [1].



Figure 1.2 Exemple sur la variation de pose

1.2.2.3 Les expressions faciales

L'expression faciale de l'émotion, combinée avec la parole peut produire des changements d'apparence importants des visages.

La déformation du visage qui est due aux expressions faciales est localisée principalement sur la partie inférieure du visage. L'information faciale se situant dans la partie supérieure du visage reste quasi invariable. Elle est généralement suffisante pour effectuer une identification. Toutefois, étant donné que l'expression faciale modifie l'aspect du visage, elle entraîne forcément une diminution du taux de reconnaissance [1].



Figure 1.3 Exemple d'expressions faciales

1.2.2.4 Occlusion

Le visage peut être partiellement masqué par des objets dans la scène, ou par le port d'accessoires tels que lunettes, écharpe etc. Les résultats expérimentaux semblent indiquer que, dans ces conditions, les performances des algorithmes de reconnaissance restent faibles [1].

1.3 Expression faciale et émotion

Les expressions faciales et les émotions jouent un rôle crucial dans la communication humaine, permettant aux individus d'exprimer leurs sentiments, leurs intentions et leurs états émotionnels.

Expressions et émotions sont très liées et parfois confondues, c'est pour cette raison qu'on se tiendra aux définitions suivantes par la suite.

1.3.1 Emotion

Les émotions peuvent être définies comme des réponses psychophysiologiques complexes et subjectives à des stimuli internes ou externes, qui incluent des expériences affectives, des changements physiologiques, des réactions comportementales et des processus cognitifs. Elles sont souvent associées à des états mentaux et physiologiques spécifiques, tels que la joie, la tristesse, la colère, la peur, le dégoût, la surprise, entre autres, et peuvent influencer le comportement et la prise de décision. Les émotions peuvent être déclenchées par divers facteurs, y compris les événements de la vie quotidienne, les interactions sociales, les pensées et les souvenirs. Elles jouent un rôle essentiel dans la régulation du comportement, dans la communication interpersonnelle et dans la perception de soi et des autres [5].

On distingue six émotions identifiées par Ekman : la joie, la tristesse, la peur, la colère, la surprise et le dégoût.

1.3.2 Expression faciale

L'expression faciale peut être définie comme les configurations musculaires spécifiques du visage qui reflètent et communiquent les émotions, les intentions et les états mentaux d'un individu. Ces expressions sont souvent caractérisées par des mouvements et des positions des muscles du visage, tels que les sourcils, les yeux, le nez, la bouche et les joues, qui varient en fonction des émotions ressenties, telles que la joie, la tristesse, la colère, la peur, le dégoût et la surprise. Les expressions faciales jouent un rôle crucial dans la communication non verbale, permettant aux individus d'exprimer et de percevoir les émotions et les intentions des autres, ainsi que de réguler les interactions sociales. Elles sont également utilisées comme indices dans divers domaines, tels que la psychologie, la psychiatrie, l'anthropologie, la linguistique, la robotique et l'intelligence artificielle, pour comprendre le comportement humain et développer des systèmes d'interaction homme-machine plus intuitifs et efficaces [6].

1.3.3 Expressions faciales universelles

Les expressions faciales universelles font référence à un ensemble restreint d'expressions du visage qui sont largement reconnues et interprétées de la même manière à

travers différentes cultures et contextes sociaux. Le psychologue Paul Ekman a identifié six expressions faciales universelles, communément appelées "les six émotions de base", qui sont les suivantes [6]:

- **Joie** : Elle se caractérise par l'état d'une personne dans une condition de satisfaction intense. Elle est due par rapport au désir, à la réussite, au bien-être, et l'accomplissement, mais aussi l'approche [7].
- **Tristesse** : C'est l'état d'une personne qui souffre moralement suite à une insatisfaction et des soucis. Elle est souvent due soit à une perte, ou un deuil, ou un obstacle, la personne se replie souvent sur soi [7].
- **Colère** : C'est l'état d'une personne dans une réaction violente et agressive lors d'une contrariété. Elle est souvent due soit à une injustice, ou un dommage, atteinte au système de valeurs. La plupart des personnes en ce moment attaquent [7].
- **Dégoût** : C'est l'état d'une personne qui a une répugnance pour certains aliments ou à un manque d'appétit. Elle est souvent due soit à un rejet, ou contre quelqu'un, ou à une aversion. La plupart des personnes en ce moment préfèrent se retirer [7].
- **Peur** : C'est l'état d'une personne menacée par un danger réel ou imaginaire. Elle est souvent due soit à une menace, ou à un danger, ou à des inconnus. La plupart des personnes en ce moment préfèrent prendre la fuite [7].
- **Surprise** : C'est l'état d'une personne étonnée par quelque chose d'inattendu. Elle est souvent due soit à un danger immédiat, ou à un imprévu, ou à des inconnus. La plupart des personnes en ce moment préfèrent prendre la fuite ou elles sursautent [7].

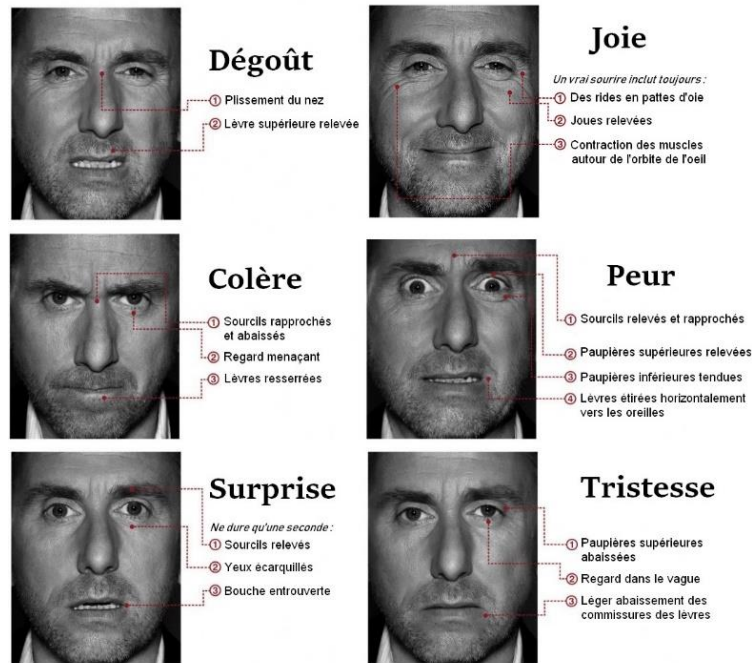


Figure 1.4 Les émotions universelles

1.3.4 Les Systèmes de codification

La compréhension de l'expression faciale nécessite une connaissance approfondie de l'anatomie du visage ainsi que de la structure des muscles faciaux sous-jacents. Les chercheurs se sont particulièrement intéressés au développement de systèmes de codage pour décrire et interpréter ces expressions. Parmi les différentes propositions, l'outil d'Ekman est largement reconnu et utilisé.

En 1978 Ekman a développé un outil de codification des expressions du visage largement utilisé aujourd'hui. Il s'intéresse désormais à l'analyse des expressions de manière informatique [8].

1.3.4.1 Système de Codification des Actions Faciales FACS

Ekman et Izard ont élaboré des méthodes pour mesurer les comportements du visage, notamment en créant le système FACS (Facial Action Coding System), largement reconnu et utilisé. Ce système utilise environ quarante caractéristiques anatomiques distinctes et établit

une taxonomie complète des expressions faciales. Le FACS est devenu le standard le plus courant pour classer de manière systématique les expressions physiques des émotions faciales, et il est employé par les psychologues ainsi que par les infographistes. Il définit 46 Unités d'Action, qui correspondent à des contractions ou des relaxations de un ou plusieurs muscles, dont les combinaisons définissent une expression faciale. Initialement, le système reposait sur la formation d'experts pour reconnaître et interpréter ces Unités d'Action, mais aujourd'hui, il est également utilisé pour automatiser la détection des Unités d'Action et donc des expressions, ainsi que pour la simulation graphique de visages [9].























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 9	Upper Lip Raiser AU 10	Nasolabial Deepener AU 11	Lip Corner Puller AU 12	Cheek Puffer AU 13	Dimpler AU 14
					
Lip Corner Depressor AU 15	Lower Lip Depressor AU 16	Chin Raiser *AU 17	Lip Puckerer *AU 18	Lip Stretcher *AU 20	Lip Funneler AU 22
					
Lip Tightener AU 23	Lip Pressor AU 24	Lips Part *AU 25	Jaw Drop *AU 26	Mouth Stretch *AU 27	Lip Suck AU 28
Lip Tightener	Lip Pressor	Lips Part	Jaw Drop	Mouth Stretch	Lip Suck

Figure 1.5 Liste des unités d'action relatives aux 6 expressions faciales

1.3.4.2 MPEG-4

La norme de codage vidéo MPEG-4 dispose d'un modèle du visage humain développé par le groupe d'intérêt Face and Body Ad Hoc 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 points sont effectuées pour former des unités de mesure qui servent à la

description des mouvements musculaires (Facial Animation Paramètres - équivalents des Actions Unitaires d'Ekman) [10].

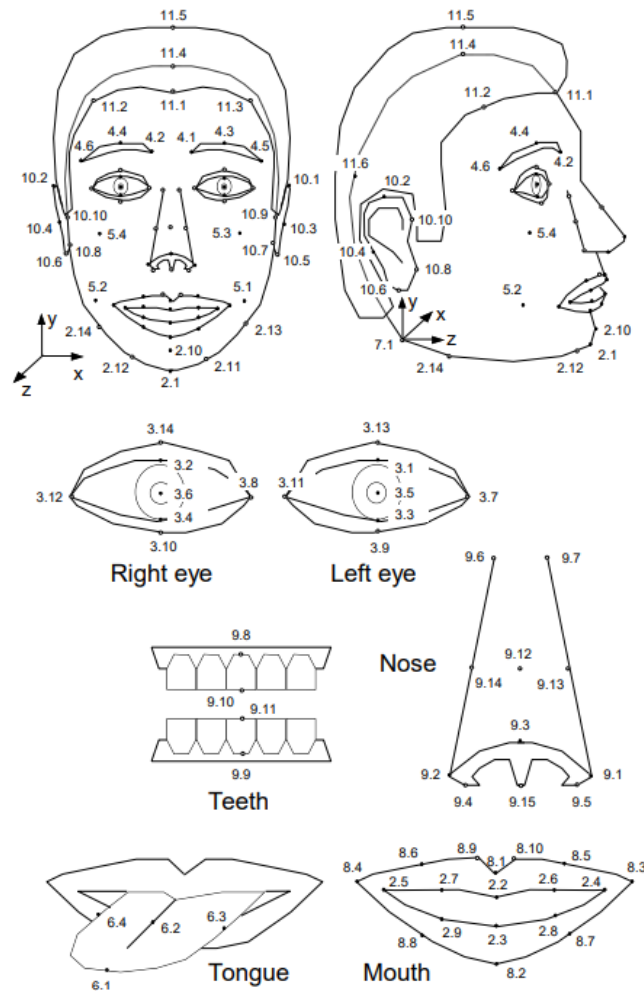


Figure 1.6 Modèle MPEG4 l'ensemble d'attributs faciaux [10]

1.3.5 Système de reconnaissance d'expression faciale

Le système d'analyse des expressions faciales fait référence à un ensemble de méthodes et de techniques utilisées pour évaluer et interpréter les expressions faciales humaines. L'objectif principal de ce système est de détecter, classifier et analyser les expressions faciales afin de comprendre les émotions, les intentions et les états mentaux des individus. Il est généralement composé de 3 étapes principales comme l'illustre la figure 1.7 ci-dessous.

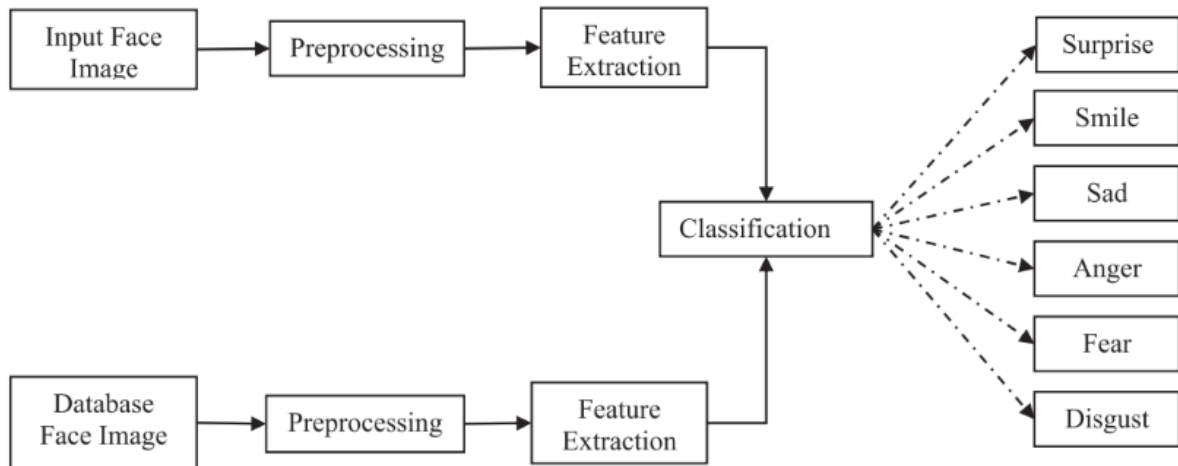


Figure 1.7 L'architecture d'un système de reconnaissance des émotions [11]

1.3.5.1 Prétraitement

Le prétraitement est un processus qui peut être utilisé pour améliorer les performances du système FER (Face Expression Recognition) et peut être effectué avant le processus d'extraction des caractéristiques. Le prétraitement de l'image comprend différents types de processus tels que la clarté de l'image et la mise à l'échelle, l'ajustement du contraste et des processus d'amélioration supplémentaires pour améliorer les cadres d'expression.

En FER, plusieurs méthodes de prétraitement sont utilisées mais le processus de segmentation de la ROI (Region Of Interest) est plus adapté car il détecte précisément les organes du visage qui sont principalement utilisés pour la reconnaissance des expressions. Ensuite, l'égalisation d'histogramme est également une autre technique de prétraitement importante pour la FER car elle améliore la distinction des images [11].

1.3.5.2 Extraction des caractéristiques

L'extraction des caractéristiques consiste à trouver et à décrire les caractéristiques positives d'intérêt au sein d'une image pour un traitement ultérieur. En traitement d'image et en vision par ordinateur, l'extraction des caractéristiques est une étape significative, car elle permet de passer de la représentation des données graphiques à une représentation implicite.

Ensuite, ces représentations de données peuvent être utilisées en tant qu'entrées pour la classification [11].

Les méthodes d'extraction des caractéristiques pour l'analyse d'expression peuvent être séparées en deux types d'approches : les méthodes basées sur les caractéristiques géométriques et les méthodes basées sur l'apparence [4].

A. Les caractéristiques géométriques

Capturent la forme et la position des composants faciaux. Ces caractéristiques du visage sont extraites pour former un vecteur représentant la géométrie faciale. Cette approche contient différents modèles pour obtenir la reconnaissance des expressions faciales, tels que [4] :

- Modèle de forme active (ASM)
- Modèles d'apparence active (AAM)

B. Les caractéristiques d'apparence

Captent les variations de l'apparence du visage, comme les rides et les sillons, qui sont des caractéristiques de texture cutanée. Ces aspects visuels peuvent être extraits sur l'ensemble du visage ou sur des zones spécifiques de celui-ci. Et pour cette approche il y a des techniques pour obtenir la reconnaissance des expressions faciale comme [4]:

- Motif binaire local (LBP)
- Quantification de phase locale (LPQ)
- Histogramme de gradient orienté (HOG)

1.3.5.3 Classification

C'est la phase finale d'un système de reconnaissance des expressions faciales. À ce stade, l'objectif est de reconnaître les six expressions prototypes. Les recherches dans ce domaine sont généralement divisées en trois catégories : les approches globales, les approches

locales et les approches hybrides. Chaque type d'approche présente ses propres avantages et inconvénients en ce qui concerne les défis liés aux conditions environnementales, aux variations d'échelle, aux orientations des images, aux angles de vue de la tête, et autres [4].

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

- NN : Réseaux de neurones (Neural Networks).
- LDA : Analyse Discriminante Linéaire (Linear discriminant analysis).
- SVM : Machines à vecteurs de support (Support Vector Machine)
- HMM : Modèles de Markov cachés (Hidden Markov Model).

1.3.6 Domaine d'application

La reconnaissance des émotions faciales, une branche émergente de l'informatique et de la psychologie, offre des possibilités passionnantes dans de nombreux domaines de la vie moderne. Voici quelques domaines d'application de cette technologie

- **Publicité et marketing** : Les entreprises utilisent la reconnaissance des émotions faciales pour évaluer les réactions émotionnelles des consommateurs aux publicités, aux produits et aux expériences de marque. Ces informations peuvent être utilisées pour optimiser les campagnes publicitaires et les stratégies de marketing afin d'augmenter l'engagement et les ventes.
- **Développement personnel** : Les applications et les dispositifs personnels basés sur la reconnaissance des émotions faciales peuvent aider les individus à mieux comprendre et à réguler leurs propres émotions. Cela peut inclure des applications de bien-être mental, des outils d'entraînement à la communication non verbale et des dispositifs de surveillance du bien-être émotionnel.
- **Psychologie et psychiatrie** : La reconnaissance des émotions faciales est utilisée pour étudier les réponses émotionnelles des individus dans différentes

situations, ainsi que pour évaluer les troubles mentaux tels que la dépression, l'anxiété et l'autisme. Les professionnels de la santé mentale peuvent utiliser cette technologie pour mieux comprendre les schémas émotionnels des patients et adapter leurs thérapies en conséquence.

- **Santé mentale** : Dans le domaine de la santé mentale, la reconnaissance des émotions faciales peut aider les cliniciens à évaluer l'état émotionnel des patients et à suivre l'efficacité des traitements. Elle peut également être intégrée dans des applications de télémédecine pour fournir un suivi à distance des patients.

1.3.7 La reconnaissance des émotions dans le domaine médical

Les émotions humaines peuvent révéler des informations cruciales sur les conditions de santé et de nombreux troubles. Des recherches ont été menées sur les variations émotionnelles dans la maladie de Parkinson (MP), la schizophrénie, la maladie d'Alzheimer (MA), le trouble de l'épilepsie et la dépression [12].

Dans le domaine de la surveillance médicale, l'analyse des expressions faciales peut être utilisée de plusieurs manières :

1.3.7.1 Surveillance de la douleur:

Les expressions faciales peuvent être un indicateur important de la douleur chez les patients, en particulier chez ceux qui ne peuvent pas communiquer verbalement, comme les nourrissons, les personnes âgées ou les patients inconscients. Les systèmes d'analyse des expressions faciales peuvent être utilisés pour surveiller les expressions faciales des patients et détecter les signes de douleur, ce qui permet aux professionnels de santé d'ajuster les traitements en conséquence.

1.3.7.2 Surveillance de la santé mentale:

Dans les environnements de soins de santé mentale, l'analyse des expressions faciales peut être utilisée pour surveiller les patients et détecter les signes de détresse émotionnelle ou

de crise. Les systèmes d'analyse des expressions faciales peuvent alerter le personnel médical en cas de changements significatifs dans les expressions faciales des patients, ce qui peut permettre une intervention précoce.

1.3.7.3 Surveillance des troubles du sommeil:

Certains troubles du sommeil, tels que l'apnée du sommeil, peuvent être associés à des expressions faciales spécifiques pendant le sommeil, telles que les grimaces ou les mouvements de la mâchoire. Les systèmes d'analyse des expressions faciales peuvent être utilisés pour surveiller les expressions faciales des patients pendant le sommeil et détecter les signes de troubles du sommeil, ce qui peut aider au diagnostic et à la gestion de ces conditions.

1.3.7.4 Surveillance des signes vitaux:

Bien que les expressions faciales ne soient pas des indicateurs directs des signes vitaux tels que la fréquence cardiaque ou la pression artérielle, elles peuvent fournir des informations contextuelles importantes sur l'état de santé général d'un patient. Par exemple, des expressions faciales pâles ou fatiguées pourraient indiquer une détresse physiologique.

1.4 Objectif

L'objectif de ce projet consiste à implémenter un système de reconnaissance des émotions faciales, avec un accent particulier sur son utilisation chez les patients en soins intensifs. Ce système vise à détecter les changements dans les expressions faciales des patients, fournissant ainsi aux cliniciens des informations supplémentaires pour évaluer leur état émotionnel et guider leur diagnostic et leur traitement.

1.5 Conclusion

Nous avons consacré ce chapitre aux principales notions et caractéristiques du domaine de la reconnaissance des émotions faciales. Par ailleurs, nous avons survolé les problèmes, et les différentes approches de détection de visages les plus utilisées, ainsi qu'une introduction sur l'expression et l'émotion. Et nous avons concentré notre étude sur l'apprentissage en

profondeur (Deep Learning) pour la détection de visage ainsi que les réseaux de neurones convolutifs (CNNs) dont l'étude détaillée feront objet du chapitre suivant

Chapitre 2

Réseaux de Neurones Convolutifs

2.1 Introduction

L'apprentissage est un domaine de l'intelligence artificielle permettant à une machine d'apprendre à partir d'exemples déjà disponibles. Dans ce chapitre, nous allons nous focaliser sur les réseaux de neurones utilisés pour la reconnaissance des émotions faciales, considérés comme l'un des algorithmes les plus performants de l'apprentissage approfondi, les réseaux de neurones convolutifs ou aussi appelés CNN.

2.2 Deep learning

Le deep learning est un sous-ensemble de l'apprentissage automatique qui utilise des réseaux neuronaux à plusieurs couches pour imiter le fonctionnement du cerveau humain. Bien qu'ils ne soient pas aussi performants que ce dernier, ces réseaux apprennent à partir de grandes quantités de données. Les couches supplémentaires améliorent la précision des prédictions. Le deep learning est à la base de nombreuses applications d'IA, comme les assistants numériques, les télécommandes vocales et la détection de fraudes, et joue un rôle clé dans des technologies émergentes [\[13\]](#).

Le deep Learning s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain (figure 2.1). 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

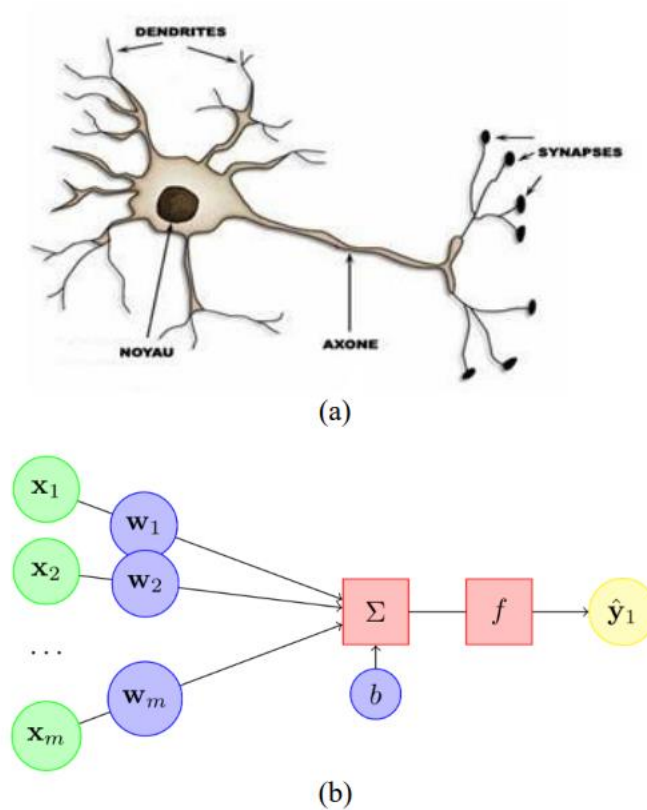


Figure 2.1 Neurone biologique (a) et neurone artificiel (b)

2.2.1 Les différents algorithmes de Deep Learning

Alors que le deep learning n'est certainement pas nouveau, il connaît une croissance explosive grâce à l'intersection de réseaux neuronaux profondément stratifiés et à l'utilisation de GPU pour accélérer leur exécution, Le nombre d'architectures et d'algorithmes utilisés dans le deep learning est vaste et varié. Chaque algorithme possède ses propres spécificités et applications [14].

- **Réseaux de neurones convolutifs** : les CNN sont constitués d'une multitude de couches chargées de traiter et d'extraire les caractéristiques des données. De manière spécifique, les réseaux neuronaux convolutifs sont utilisés pour l'analyse et la détection d'objets. Ils peuvent donc servir par exemple à reconnaître des

images satellites, traiter des images médicales, détecter des anomalies ou prédire des séries chronologiques [14].

- **Réseaux neuronaux récurrents** : Les RNN possèdent des connexions qui constituent des cycles dirigés. Cela permet aux sorties du LSTM d'être exploitées comme entrées au niveau de la phase actuelle. La sortie du LSTM se transforme en une entrée pour la phase actuelle. Elle peut donc mémoriser les entrées précédentes à l'aide de sa mémoire interne. Dans la pratique, les RNN sont utilisés pour le sous-titrage d'images, le traitement du langage naturel et la traduction automatique [14].
- **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 [14].

2.3 Réseaux de Neurones Convolutifs

Convolutional Neural Network (CNN) (réseaux de neurones convolutifs) est un type de réseau de neurones artificiels acycliques (feed-forward) dans lequel le motif de connexion entre les neurones est inspiré par le cortex visuel des animaux [15]. Le but d'un CNN est d'apprendre des fonctionnalités d'ordre supérieur dans les données via des convolutions. Ils sont bien adaptés à la reconnaissance d'objets et de classification d'images.

Ces structures permettent d'identifier des visages, des individus, des panneaux de signalisation et de nombreux autres aspects des données visuelles. L'architecture typique d'un CNN est donnée dans la Figure 2.2

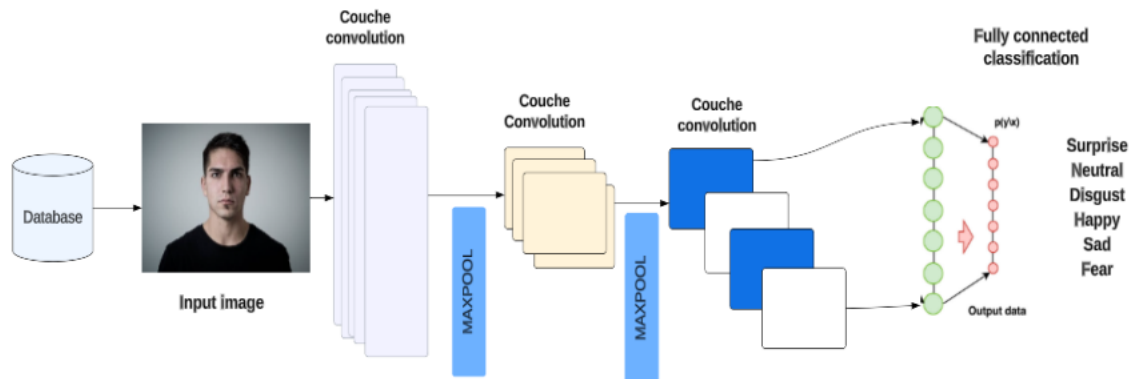


Figure 2.2 L'architecture typique d'un CNN

2.3.1 Les différentes couches d'un CNN

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.

2.3.1.1 La couche de convolution

La convolution est l'élément principal des réseaux de neurones convolutifs. La convolution peut être simplifiée en tant qu'application d'un filtre mathématique à une image, utilisée pour le traitement et la reconnaissance d'images. 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 [16].

Cette technique nous permet de détecter les parties de l'image qui pourraient être intéressantes. 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 [17].

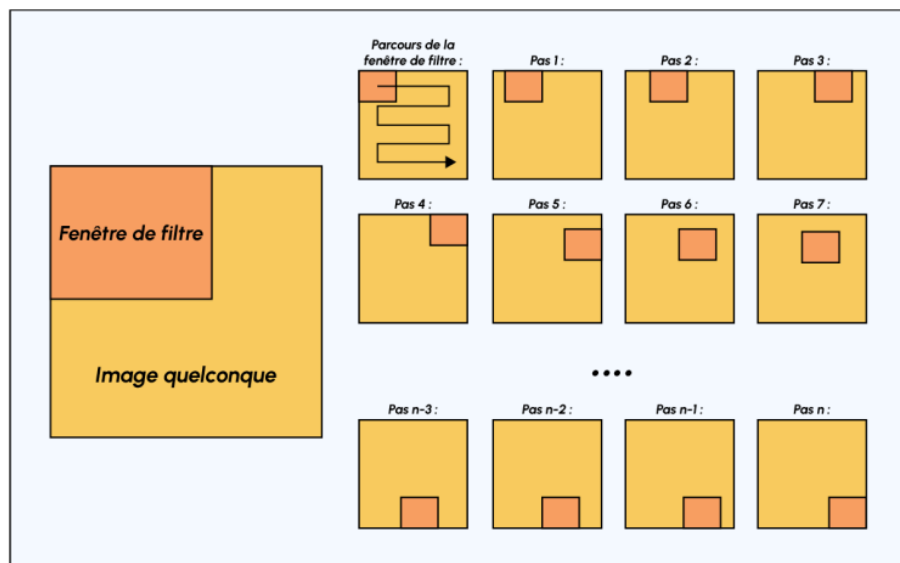


Figure 2.3 Schéma de déplacement de filtre [17]

Trois hyper paramètres permettent de dimensionner le volume de la couche de convolution (aussi appelé volume de sortie) :

- Profondeur de la couche** : nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
- Le pas** : contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.

- c. **La marge (à 0) ou ' zéro padding '** : parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. La taille de ce 'zero-padding' est le troisième hyper paramètre.

2.3.1.2 Les différentes convolutions

Bien que la convolution de base soit couramment utilisée, il est bénéfique de connaître les différentes variantes de convolutions disponibles car elles peuvent s'avérer utiles dans divers contextes [16].

- **Convolution classique avec padding** : Cette convolution est la plus courante et consiste à déplacer un noyau de convolution sur une image, calculant le produit scalaire entre le noyau et la zone correspondante de l'image. Le padding permet de contrôler la taille de la sortie en ajoutant des zéros autour de l'image.
- **Dilated convolution** : Aussi appelée convolution dilatée, elle est similaire à la convolution classique mais avec un paramètre de dilatation qui spécifie l'espacement entre les valeurs échantillonnées dans le noyau. Cela permet d'élargir le champ réceptif sans augmenter le nombre de paramètres.
- **Transposed convolution** : Aussi connue sous le nom de convolution transposée ou déconvolution, elle construit la sortie comme si on inversait une convolution sur l'image. Elle est souvent utilisée pour augmenter la taille de l'image
- **Séparable convolution** : Cette convolution décompose la convolution en deux étapes : une première convolution spatiale suivie d'une convolution en profondeur. Cela réduit le coût computationnel tout en maintenant les performances du modèle.

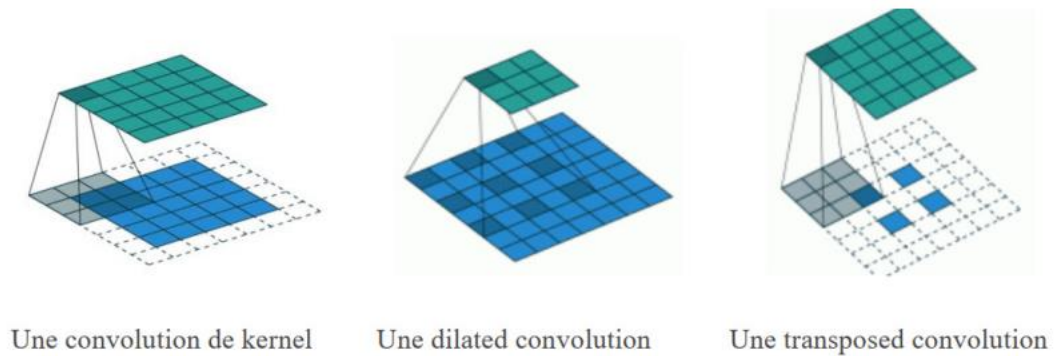


Figure 2.4 Différents types de convolutions [16]

2.3.1.3 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 [16].

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 [16].

Il existe plusieurs types de pooling : Max, Moyenne, Somme, etc.

- **Max pooling** : qui revient à prendre la valeur maximale de la sélection. C'est le type le plus utilisé, car il est rapide à calculer (immédiat), et permet de simplifier efficacement l'image.

- **Le mean-pooling** : (ou averagepooling), soit la moyenne des pixels de la sélection, on calcule la somme de toutes les valeurs et on divise par le nombre de valeurs. On obtient ainsi une valeur médiane pour représenter ce lot de pixels.
- **Le sum-pooling** : c'est la moyenne sans avoir divisé par le nombre de valeurs (on ne calcule que leur somme).

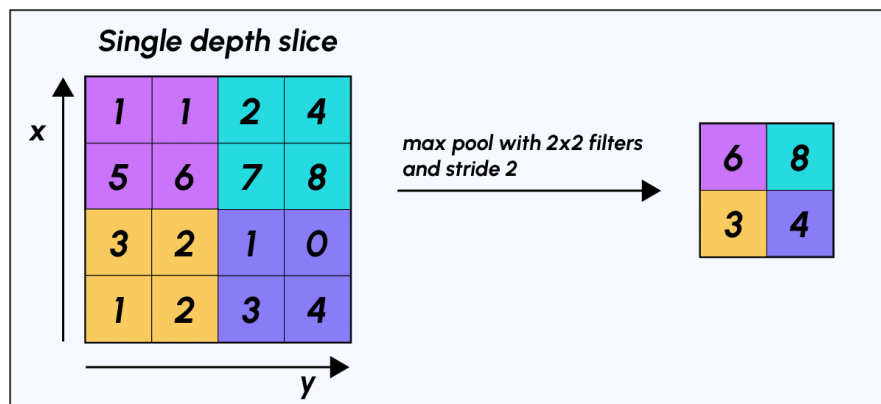


Figure 2.5 : Processus de Max-Pooling [17]

2.3.1.4 La couche ReLU

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 [16].

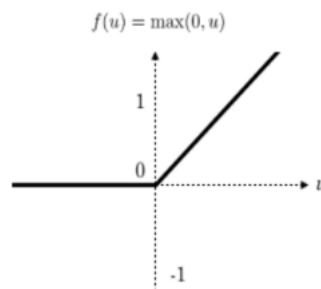


Figure 2.6 La fonction d'activation ReLU [17]

2.3.1.5 La couche entièrement connectée

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. 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. Chaque probabilité est calculée à l'aide de la fonction softmax. 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. Le but de la couche entièrement connectée est de pouvoir utiliser ces fonctions pour classer l'image d'entrée dans différentes classes en fonction de l'ensemble de données d'apprentissage [16].

2.3.2 Avantages des CNN

Les CNN (Convolutional Neural Networks) offrent plusieurs avantages significatifs qui les rendent extrêmement efficaces pour la tâche de traitement des données visuelles, notamment :

- **Extraction automatique des caractéristiques** : Les CNN sont capables d'apprendre automatiquement les caractéristiques pertinentes à partir des données brutes, éliminant ainsi le besoin d'ingénierie des fonctionnalités manuellement. Les couches de convolution permettent de capturer des motifs visuels à différentes échelles et orientations, ce qui rend les CNN adaptés à une grande variété de tâches de vision par ordinateur.
- **Invariance spatiale** : Les CNN sont capables de détecter des motifs dans une image indépendamment de leur position spatiale. Grâce à l'utilisation de filtres convolutifs et de couches de pooling, les CNN sont capables de généraliser efficacement à des images présentant des variations de translation, de rotation et de mise à l'échelle.
- **Capacité à gérer des données de grande taille** : Les CNN peuvent être entraînés sur des ensembles de données volumineux sans surajustement excessif grâce à l'utilisation de techniques telles que la régularisation et le dropout. Cela leur permet d'exploiter pleinement la richesse des données disponibles et d'obtenir des performances élevées sur des tâches complexes.

2.3.3 Quelques réseaux convolutifs pré-entraînés connus

Il existe plusieurs réseaux de classification d'images dont les plus répandus ont été entraînés avec plus d'un million d'images et peuvent classer les images en un ou plusieurs milliers de catégories d'objets

- **VGG** : Développés par le Visual Geometry Group (VGG) de l'Université d'Oxford, VGG16 et VGG19 sont des architectures de réseaux convolutifs profonds très populaires. Ils se caractérisent par leur simplicité et leur uniformité, avec une série de couches convolutives et de couches de pooling. Ces modèles sont souvent utilisés comme bases pour le transfert d'apprentissage [\[18\]](#).

- **ResNet** : Les réseaux ResNet, développés par Microsoft Research, introduisent des connexions résiduelles qui permettent de former avec succès des réseaux beaucoup plus profonds. Ces architectures ont remporté le premier prix à l'ImageNet Large Scale Visual Recognition Challenge (ILSVRC) en 2015 [19].
- **Inception** : Les architectures Inception, développées par Google, se caractérisent par l'utilisation de blocs d'inception qui permettent une extraction d'entités à différentes échelles spatiales. InceptionV3 et InceptionResNetV2 sont des versions populaires utilisées pour la classification d'images [20].
- **MobileNet** : Développés par Google, les modèles MobileNet sont conçus pour une utilisation sur des appareils mobiles et des applications nécessitant une faible consommation d'énergie. Ils sont basés sur des opérations de convolution profonde avec un nombre réduit de paramètres [21].
- **Xception** : Également développé par Google, Xception est une variante de l'architecture Inception qui remplace la convolution standard par des convolutions en profondeur séparable. Cette approche a permis de réduire considérablement le nombre de paramètres tout en maintenant des performances élevées [22].
- **DenseNet** : Les réseaux DenseNet, développés par l'Université Cornell, se distinguent par leur structure dense, où chaque couche est connectée à toutes les couches précédentes. Cela favorise le flux d'informations et facilite l'apprentissage [23].

2.4 Conclusion

Dans ce chapitre, nous avons introduit le concept de l'apprentissage profond et présenté diverses architectures, Nous nous sommes ensuite concentrés sur les réseaux de neurones

convolutifs (CNN), et leurs structures, et ses différentes couches. Et à la fin nous avons présenté quelques réseaux convolutifs pré-entraînés connus.

Dans le chapitre suivant, nous allons décrire en détail les étapes nécessaires à la construction de notre système de reconnaissance faciale basé sur les réseaux de neurones convolutifs.

Chapitre 3 Conception et Implémentation

3.1 Introduction

Ce chapitre vise à détailler les différentes étapes de la mise en œuvre d'un système de reconnaissance des émotions faciales basé sur des réseaux de neurones convolutifs (CNN) conçu pour le domaine médical. Nous commencerons par présenter les outils utilisés pour l'implémentation. Ensuite, nous décrirons en détail la conception du modèle proposé. Enfin, nous conclurons ce chapitre en présentant notre application finale.

3.2 Environnement de travail

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

3.2.1 Environnement matériel

Pour réaliser ce projet avec succès, Nous avons utilisé deux ordinateurs portables, Voici leurs spécifications:

- PC 1: ASUS Vivobook Flip 14
 - Processeur : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz
 - Mémoire : 8 Go
 - Disque dur : SSD 256 Go M.2 NVMe
 - Système d'exploitation : Windows 11 Pro
- PC 2: Apple MacBook Pro 2017

- Processeur : Intel Core i5 bicœur à 2,3 GHz
- Mémoire : 8 Go
- Disque dur : SSD 128 Go M.2 NVMe
- Système d'exploitation : macOS ventura

3.2.2 Environnement de développement

3.2.2.1 Visual Studio Code

Visual Studio Code (VS Code) est un environnement de développement intégré (IDE) créé par Microsoft, qui se distingue par sa légèreté, sa rapidité et sa grande adaptabilité aux besoins des développeurs. Conçu pour faciliter l'écriture et la modification de code source, VS Code offre une interface utilisateur intuitive et un ensemble complet de fonctionnalités pour soutenir le processus de développement logiciel [24].

3.2.2.2 Google Colab

Google Colab est une plateforme de notebooks Jupyter hébergée par Google, qui offre un environnement de développement interactif pour écrire, exécuter et partager du code Python. Basée sur le cloud, Colab permet aux utilisateurs d'accéder à leurs notebooks depuis n'importe quel navigateur web sans installation locale. Avec un accès gratuit aux ressources de calcul, y compris des GPU, Colab est largement utilisé pour des tâches de calcul intensif telles que l'apprentissage automatique. Son intégration avec Google Drive facilite le stockage et le partage des notebooks, tandis que sa compatibilité avec les bibliothèques populaires de Python en fait un choix populaire pour la recherche, le développement et l'enseignement en science des données et en apprentissage automatique [25].

3.2.2.3 Pourquoi les GPU ?

La plupart des approches d'apprentissage machine ou d'apprentissage en profondeur nécessitent l'utilisation de GPU en raison du volume de données à traiter. Les calculs mathématiques complexes du deep learning sur un processeur central (CPU) peuvent prendre

des mois. Cependant, ces calculs peuvent être effectués beaucoup plus rapidement en les exécutants sur des GPU dédiés. Il est donc vivement recommandé d'utiliser un GPU pour l'entraînement des réseaux de neurones, car cela permet d'obtenir des performances élevées. Bien que les GPU soient généralement coûteux, ainsi que les plateformes cloud comme AWS et GCP, Google Colab offre une solution gratuite et efficace. En effet, Colab met à disposition des utilisateurs un GPU Nvidia Tesla K80, offrant des performances remarquables. Cette ressource permet aux développeurs de former leurs réseaux de neurones dans le cloud avec une efficacité et une rapidité sans égal [26].

3.2.2.4 Cloudinary

Cloudinary est une plateforme en ligne spécialisée dans la gestion et la distribution de médias, notamment les images et les vidéos. Elle offre des outils complets pour le stockage sécurisé, la manipulation et la transformation (comme le redimensionnement, le recadrage, et l'application de filtres), ainsi que l'optimisation automatique pour améliorer les performances web. De plus, Cloudinary utilise un réseau de distribution de contenu (CDN) pour assurer une diffusion rapide et globale des médias. Son interface conviviale et ses API robustes facilitent l'intégration et la gestion des actifs numériques, en faisant une solution populaire parmi les développeurs, les marketeurs et les équipes créatives [27].

3.2.2.5 Twilio

Twilio est une plateforme de communication en tant que service (CPaaS) qui permet aux développeurs d'intégrer facilement des fonctionnalités de communication telles que les appels vocaux, les messages texte (SMS) et les messages multimédias (MMS) dans leurs applications et services. Avec des API et des SDK conviviaux, Twilio offre une grande souplesse d'intégration et de personnalisation, permettant aux développeurs de créer des expériences utilisateur riches et interactives. La plateforme est hautement évolutive et fiable, adaptée aux entreprises de toutes tailles, et propose une tarification transparente basée sur l'utilisation, permettant aux développeurs de maîtriser leurs coûts en fonction de leurs besoins réels [28].

3.2.2.6 Python

Python est un langage de programmation interprété, polyvalent et convivial, largement utilisé dans de nombreux domaines, notamment le développement web, l'analyse de données, l'intelligence artificielle et l'automatisation des tâches. Connu pour sa syntaxe claire et concise, Python favorise la lisibilité du code et la productivité des développeurs. Il offre une vaste bibliothèque standard ainsi qu'un riche écosystème d'extensions tierces, ce qui en fait un choix privilégié pour les projets de toutes tailles. Grâce à sa simplicité et sa puissance, Python est devenu l'un des langages de programmation les plus populaires au monde, apprécié tant par les débutants que par les professionnels expérimentés [29].

3.2.3 Bibliothèques utilisées

3.2.3.1 OpenCV (Open Source Computer Vision Library)

OpenCV (Open Source Computer Vision Library) est une bibliothèque open source populaire destinée au traitement d'images et à la vision par ordinateur. Cette bibliothèque offre une vaste gamme de fonctionnalités pour la manipulation d'images, telles que la lecture et l'écriture de fichiers image dans différents formats, le redimensionnement et la transformation d'images, la détection d'objets, le suivi de mouvement, la reconnaissance faciale, la stéréovision et bien plus encore. OpenCV est largement utilisé dans de nombreux domaines, notamment la robotique, la réalité augmentée, la surveillance vidéo, la reconnaissance optique de caractères (OCR) et la médecine [30].

3.2.3.2 NumPy

NumPy est une bibliothèque fondamentale en Python pour le calcul numérique. Elle fournit des structures de données puissantes pour représenter des tableaux multidimensionnels ainsi que des fonctions pour effectuer des opérations mathématiques sur ces tableaux. NumPy est largement utilisé dans le domaine de la science des données, de l'apprentissage automatique, de la modélisation mathématique et de nombreux autres domaines. Grâce à ses performances

élevées et à sa syntaxe concise, NumPy permet aux développeurs de réaliser efficacement des calculs numériques complexes [31].

3.2.3.3 Pandas

Pandas est une bibliothèque open source très utilisée en Python pour la manipulation et l'analyse de données. Elle offre des structures de données flexibles, notamment les DataFrame et les Series, qui permettent aux utilisateurs de travailler efficacement avec des données tabulaires et de les manipuler de manière intuitive. Pandas fournit une grande variété de fonctionnalités pour le nettoyage, la transformation, la fusion et l'agrégation de données, ainsi que pour l'indexation et la sélection des données [32].

3.2.3.4 Tensorflow

TensorFlow est une bibliothèque open source développée par Google pour le calcul numérique et le développement de modèles d'apprentissage automatique et de réseaux neuronaux. Elle offre une architecture flexible pour la construction et le déploiement de modèles d'apprentissage automatique sur une variété de plates-formes, y compris les ordinateurs de bureau, les serveurs et les appareils mobiles. TensorFlow prend en charge la création de modèles d'apprentissage automatique à la fois pour l'apprentissage supervisé et non supervisé, ainsi que pour des tâches telles que la classification, la régression, la segmentation d'images, la génération de texte et bien plus encore. Grâce à ses fonctionnalités avancées telles que l'optimisation automatique des graphiques de calcul, la distribution de calcul sur plusieurs processeurs et la compatibilité avec les GPU pour l'accélération matérielle [33].

3.2.3.5 Keras

Keras est une bibliothèque open source haut niveau pour le développement d'applications d'apprentissage automatique et de réseaux neuronaux en Python. Elle offre une interface simple et intuitive pour la création, la formation et l'évaluation de modèles d'apprentissage automatique, ce qui en fait un excellent choix pour les débutants et les experts en apprentissage automatique. Keras permet aux utilisateurs de créer des modèles d'apprentissage profond en utilisant des blocs de construction appelés "couches", qui peuvent

être combinés pour former des architectures de réseau complexes. Cette bibliothèque offre également une grande flexibilité en termes de choix de backend, permettant aux utilisateurs de travailler avec des moteurs de calcul tels que TensorFlow, Theano ou Microsoft Cognitive Toolkit (CNTK) [34].

3.2.3.6 Matplotlib

Matplotlib est une bibliothèque de visualisation de données en Python, offrant une grande flexibilité pour créer une grande variété de graphiques et de visualisations. Que ce soit pour tracer des courbes, des histogrammes, des diagrammes à barres, des nuages de points ou des diagrammes en boîte, Matplotlib fournit des outils puissants pour représenter visuellement les données de manière claire et efficace. Cette bibliothèque est largement utilisée dans de nombreux domaines, notamment la science des données, la recherche académique, la visualisation de données financières et bien d'autres [35].

3.2.3.7 Tkinter

Tkinter est la bibliothèque standard de Python pour la création d'interfaces graphiques (GUI). Intégrée directement dans Python, Tkinter permet aux développeurs de concevoir des applications de bureau avec des interfaces utilisateur interactives. Elle offre une variété de widgets, tels que des boutons, des labels, des champs de texte, des boîtes de dialogue, des menus et des canevas, permettant de construire des interfaces utilisateur complètes et fonctionnelles. Tkinter est apprécié pour sa simplicité et sa facilité d'utilisation, rendant le développement de GUI accessible même pour les débutants [36].

3.3 Base de données

Afin d'accroître la précision et d'optimiser la performance des modèles d'analyse des expressions faciales, il est essentiel d'entraîner les réseaux avec un grand nombre d'échantillons d'images. Cela nécessite des ensembles de données bien classifiés, avec des émotions clairement identifiées, et suffisamment volumineux pour permettre un entraînement optimal. Pour ce faire, il existe plusieurs bases de données disponibles sur Kaggle.

Dans notre projet, nous utilisons une version améliorée de la base de données Fer2013 proposé par PRAJWAL SOOD disponible sur Kaggle. Elle comprend un total de 38191 images en niveaux de gris, 30549 images d'apprentissage et 7642 images de teste pré-recadrées de 48x48 pixels de visages chacun étiquetés avec l'une des 7 classes d'émotion suivantes [37]:

- ❖ 'ANGER': 0,
- ❖ 'DISGUST': 1,
- ❖ 'FEAR': 2,
- ❖ 'HAPPINESS': 3,
- ❖ 'NEUTRAL': 4,
- ❖ 'SADNESS': 5,
- ❖ 'SURPRISE': 6

La figure 3.1 ci-dessous montre un échantillon de la base de données



Figure 3.1 Echantillon de la base de données

3.4 Implémentation et résultats

3.4.1 Prétraitement et Augmentation des Données

Le prétraitement des données est une étape cruciale dans la préparation des données pour l'entraînement d'un modèle. Il consiste à normaliser les valeurs des pixels des images et à appliquer des transformations pour augmenter la diversité des données d'entraînement, ce qui aide le modèle à généraliser mieux aux nouvelles données.

La figure ci-dessous (Figure 3.2) montre le code utilisé pour le prétraitement et l'augmentation des données à l'aide de la classe `ImageDataGenerator` de Keras. Nous avons normalisé les valeurs des pixels des images à l'intervalle $[0, 1]$ et appliqué des techniques d'augmentation aux données d'entraînement, telles que des rotations, des décalages, des cisaillements, des zooms, et des inversions horizontales et verticales. Ces augmentations augmentent la diversité des données d'entraînement, aidant le modèle à apprendre des caractéristiques plus robustes et à mieux généraliser.

```
] batch_size = 128

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   rotation_range=20,
                                   width_shift_range=0.1,
                                   height_shift_range=0.1,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True,
                                   vertical_flip=True,)

training_set = train_datagen.flow_from_directory(train_dir,
                                                batch_size=batch_size,
                                                target_size=(48,48),
                                                shuffle=True,
                                                color_mode='grayscale',
                                                class_mode='categorical')

test_datagen = ImageDataGenerator(rescale = 1./255)
test_set = test_datagen.flow_from_directory(test_dir,
                                           batch_size=batch_size,
                                           target_size=(48,48),
                                           shuffle=False,
                                           color_mode='grayscale',
                                           class_mode='categorical')
```

Figure 3.2 Code de prétraitement et Augmentation des Données

3.4.2 Création du modèle

Après plusieurs tests, nous avons sélectionné un modèle composé de quatre couches convolutionnelles et de deux couches entièrement connectées. Les couches convolutionnelles sont responsables de l'extraction des caractéristiques importantes des images en appliquant des filtres et des opérations non linéaires telles que ReLU. Chaque couche est suivie d'une normalisation par lots pour stabiliser l'entraînement et d'une couche de max pooling pour réduire la dimensionnalité des caractéristiques extraites. Les couches entièrement connectées prennent en charge la classification des caractéristiques extraites par les couches convolutionnelles en les transformant en une représentation appropriée pour la prédiction des classes d'images. Ces couches sont également soumises à une régularisation par dropout pour prévenir le surapprentissage en désactivant aléatoirement une fraction des neurones pendant l'entraînement. Enfin, la couche de sortie utilise une activation Softmax pour produire des probabilités de classe pour les différentes catégories d'images. La couche Softmax comporte 7 neurones, correspondant aux 7 classes de sortie de notre tâche de classification. Ce modèle a été choisi pour sa capacité à bien généraliser sur les données de test tout en évitant le surapprentissage sur les données d'entraînement.

La figure ci-dessous (Figure 3.3) montre le code de la création du modèle

```

model = Sequential()

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

# 2ème Couche Convolutionnelle
model.add(Conv2D(128,(5,5),activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

# 3ème Couche Convolutionnelle
model.add(Conv2D(512,(3,3),activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

# 4ème Couche Convolutionnelle
model.add(Conv2D(512,(3,3),activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

# Flattening
model.add(Flatten())

# 1ère Couche Entièrement Connectée
model.add(Dense(256,activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.25))

# 2ème Couche Entièrement Connectée
model.add(Dense(512,activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.25))

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

```

Figure 3.3 Code de la création du modèle

La figure 3.4 illustre les différentes couches de notre modèle ainsi que le nombre de paramètres

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 64)	640
batch_normalization (Batch Normalization)	(None, 48, 48, 64)	256
max_pooling2d (MaxPooling2D)	(None, 24, 24, 64)	0
dropout (Dropout)	(None, 24, 24, 64)	0
conv2d_1 (Conv2D)	(None, 24, 24, 128)	204928
batch_normalization_1 (Batch Normalization)	(None, 24, 24, 128)	512
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 128)	0
dropout_1 (Dropout)	(None, 12, 12, 128)	0
conv2d_2 (Conv2D)	(None, 12, 12, 512)	590336
batch_normalization_2 (Batch Normalization)	(None, 12, 12, 512)	2048
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 512)	0
dropout_2 (Dropout)	(None, 6, 6, 512)	0
conv2d_3 (Conv2D)	(None, 6, 6, 512)	2359808
batch_normalization_3 (Batch Normalization)	(None, 6, 6, 512)	2048
max_pooling2d_3 (MaxPooling2D)	(None, 3, 3, 512)	0
dropout_3 (Dropout)	(None, 3, 3, 512)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 256)	1179904
batch_normalization_4 (Batch Normalization)	(None, 256)	1024
dropout_4 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 512)	131584
batch_normalization_5 (Batch Normalization)	(None, 512)	2048
dropout_5 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 7)	3591

Total params: 4478727 (17.08 MB)		
Trainable params: 4474759 (17.07 MB)		
Non-trainable params: 3968 (15.50 KB)		

Figure 3.4 Résultat de notre modèle

3.4.3 Entraînement du modèle

Maintenant que l'architecture du modèle est définie, nous sommes prêts à passer à la phase d'entraînement. Cette étape essentielle consiste à ajuster les poids du modèle en utilisant les données d'entraînement afin qu'il puisse apprendre à reconnaître les motifs et les caractéristiques dans les images.

Une fois que le modèle est entraîné, nous procéderons à l'évaluation de ses performances en ajustant certains paramètres clés :

- **Epoch** : Nombre d'itérations sur l'ensemble de données d'entraînement.
- **Loss** : Mesure du taux d'erreur du modèle pendant l'entraînement.
- **Accuracy** : Taux de prédictions correctes du modèle par rapport à l'ensemble de données d'entraînement.
- **Val_loss** : Perte calculée sur un ensemble de données de validation distinct.
- **Val_acc** : Précision du modèle sur l'ensemble de données de validation.
- **Optimiseur** : L'optimiseur est un algorithme utilisé pour ajuster les poids du modèle pendant l'entraînement afin de minimiser la fonction de perte.
- **History** : Objet retourné par la fonction d'entraînement du modèle, contenant des informations sur les métriques telles que la perte et la précision à chaque époque.

Pour une évaluation plus approfondie des performances de notre modèle, nous avons décidé d'explorer son entraînement en utilisant deux optimiseurs différents. Cette démarche nous permettra de comparer l'efficacité de ces optimiseurs dans l'apprentissage du modèle

- A. Première expérience** : Dans cette expérience nous avons entraîné notre Modèle avec l'optimisateur SGD et avec 120 epoche comme illustre la figure 3.5 ci-dessous

```

opt = SGD(learning_rate=0.0001, momentum=0.9)
model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])

epochs = 120

from keras.callbacks import ModelCheckpoint

checkpoint = ModelCheckpoint("new_modele.h5", monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
callbacks_list = [checkpoint]

history = model.fit(
    training_set,
    steps_per_epoch=training_set.n // training_set.batch_size,
    epochs=epochs,
    validation_data=test_set,
    validation_steps=test_set.n // test_set.batch_size,
    callbacks=callbacks_list
)

```

Figure 3.5 Code de l'entraînement avec SGD

La figure 3.6 ci-dessous illustre le processus d'apprentissage de notre modèle

```

Epoch 1: val_accuracy improved from -inf to 0.23543, saving model to new_modele.h5
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()
  saving_api.save_model(
238/238 [=====] - 42s 132ms/step - loss: 2.7172 - accuracy: 0.1608 - val_loss: 1.9509 - val_accuracy: 0.2354
Epoch 2/120
238/238 [=====] - ETA: 0s - loss: 2.5320 - accuracy: 0.1767
Epoch 2: val_accuracy did not improve from 0.23543
238/238 [=====] - 30s 126ms/step - loss: 2.5320 - accuracy: 0.1767 - val_loss: 1.9975 - val_accuracy: 0.1748
Epoch 3/120
238/238 [=====] - ETA: 0s - loss: 2.4296 - accuracy: 0.1827
Epoch 3: val_accuracy did not improve from 0.23543
-----●-----●-----●-----
●
●
●
Epoch 119/120
238/238 [=====] - ETA: 0s - loss: 1.6412 - accuracy: 0.3493
Epoch 119: val_accuracy did not improve from 0.37182
238/238 [=====] - 29s 121ms/step - loss: 1.6412 - accuracy: 0.3493 - val_loss: 1.5888 - val_accuracy: 0.3602
Epoch 120/120
238/238 [=====] - ETA: 0s - loss: 1.6376 - accuracy: 0.3515
Epoch 120: val_accuracy did not improve from 0.37182
238/238 [=====] - 30s 126ms/step - loss: 1.6376 - accuracy: 0.3515 - val_loss: 1.6213 - val_accuracy: 0.3547
CPU times: user 1h 11min 2s, sys: 3min 12s, total: 1h 14min 14s

```

Figure 3.6 Le résultat de l'entraînement avec SGD

B. Deuxième expérience : Dans cette expérience nous avons entraîné notre Modèle avec l'optimisateur RMSprop et avec 120 epoche comme illustre la figure 3.7 ci-dessous

```

%%time

opt = RMSprop(learning_rate=0.0001)
model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])

epochs = 120

checkpoint = ModelCheckpoint("modele.h5", monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
callbacks_list = [checkpoint]

history = model.fit(
    training_set,
    steps_per_epoch=training_set.n // training_set.batch_size,
    epochs=epochs,
    validation_data=test_set,
    validation_steps=test_set.n // test_set.batch_size,
    callbacks=callbacks_list
)

```

Figure 3.7 Code de l'entraînement avec RMSprop

La figure 3.8 ci-dessous illustre le processus d'apprentissage de notre modèle

```

Epoch 1/120
238/238 [=====] - ETA: 0s - loss: 2.1082 - accuracy: 0.2249
Epoch 1: val_accuracy improved from -inf to 0.23848, saving model to new_modele.h5
238/238 [=====] - 42s 129ms/step - loss: 2.1082 - accuracy: 0.2249 - val_loss: 2.2617 - val_accuracy: 0.2385
Epoch 2/120
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()`
  saving_api.save_model(
238/238 [=====] - ETA: 0s - loss: 1.7765 - accuracy: 0.2983
Epoch 2: val_accuracy did not improve from 0.23848
238/238 [=====] - 28s 118ms/step - loss: 1.7765 - accuracy: 0.2983 - val_loss: 4.3354 - val_accuracy: 0.1615
Epoch 3/120
238/238 [=====] - ETA: 0s - loss: 1.6143 - accuracy: 0.3694
Epoch 3: val_accuracy improved from 0.23848 to 0.39566, saving model to new_modele.h5
238/238 [=====] - 28s 119ms/step - loss: 1.6143 - accuracy: 0.3694 - val_loss: 1.5144 - val_accuracy: 0.3957
●
●
●
Epoch 118/120
238/238 [=====] - ETA: 0s - loss: 0.8689 - accuracy: 0.6743
Epoch 118: val_accuracy did not improve from 0.67863
238/238 [=====] - 28s 118ms/step - loss: 0.8689 - accuracy: 0.6743 - val_loss: 0.9702 - val_accuracy: 0.6492
Epoch 119/120
238/238 [=====] - ETA: 0s - loss: 0.8676 - accuracy: 0.6766
Epoch 119: val_accuracy did not improve from 0.67863
238/238 [=====] - 31s 131ms/step - loss: 0.8676 - accuracy: 0.6766 - val_loss: 0.9314 - val_accuracy: 0.6655
Epoch 120/120
238/238 [=====] - ETA: 0s - loss: 0.8628 - accuracy: 0.6788
Epoch 120: val_accuracy did not improve from 0.67863
238/238 [=====] - 28s 118ms/step - loss: 0.8628 - accuracy: 0.6788 - val_loss: 0.9276 - val_accuracy: 0.6598
CPU times: user 1h 8min 26s, sys: 3min 4s, total: 1h 11min 31s

```

Figure 3.8 Résultat de l'entraînement avec RMSprop

3.4.4 Résultats et analyse

Dans cette partie nous nous intéressons aux résultats d'exécution de notre approche.

3.4.4.1 Première expérience

Ci-dessous, nous présentons les courbes de précision et de perte qui illustrent l'entraînement et les tests de notre modèle avec l'optimiseur SGD.

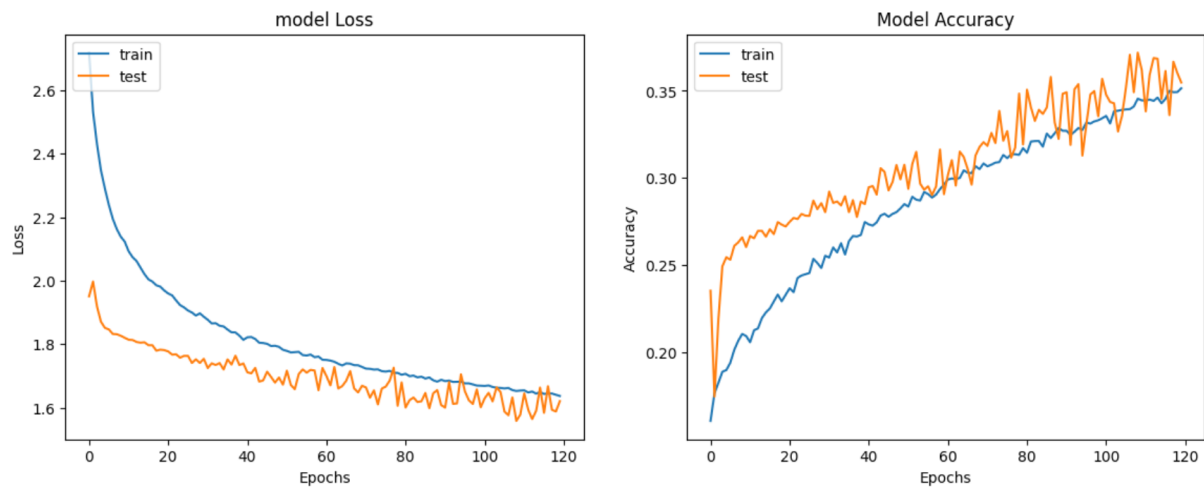


Figure 3.9 Courbes de précision (accuracy) et de perte (loss) SGD

La figure ci-dessus (Figure 3.9) représente les courbes de précision (accuracy) et de perte (loss) générées par le modèle. La courbe sur la gauche montre la perte de l'ensemble d'entraînement en bleu et celle de l'ensemble de test en orange, en fonction du nombre d'epochs. À droite, la courbe représente la précision de l'ensemble d'entraînement en bleu et celle de l'ensemble de test en orange, également en fonction du nombre d'epochs.

Il est important de noter que la précision de validation (val_accuracy) n'a pas dépassé 0.37, ce qui révèle une limite dans les performances du modèle sur les données de test.

La matrice de confusion ci-dessous (Figure 3.10) illustre la performance de notre modèle de classification sur les différentes catégories d'émotions. Chaque ligne de la matrice représente les vraies étiquettes, tandis que chaque colonne représente les prédictions faites par le modèle.

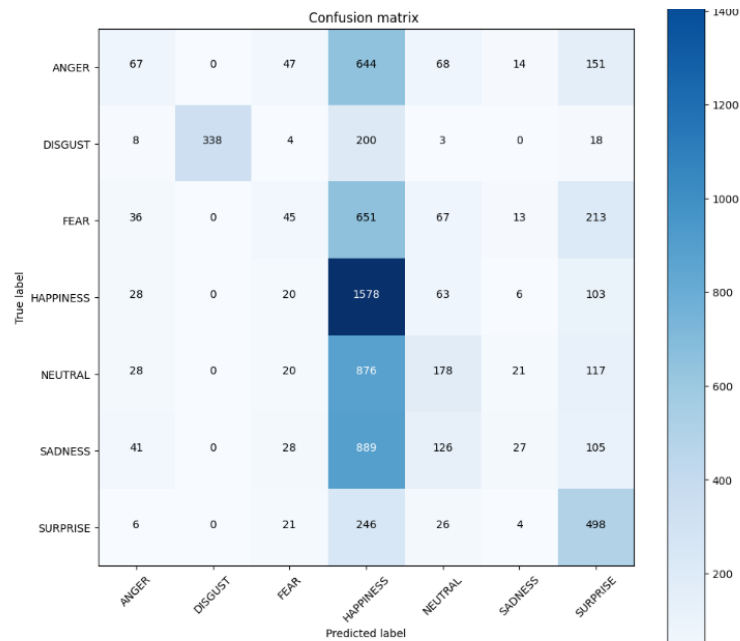


Figure 3.10 Matrice de confusion SGD

Bien que le modèle montre une bonne performance globale, il semble confondre plusieurs catégories avec "HAPPINESS". Cela indique un besoin d'amélioration pour mieux différencier les autres émotions des expressions de bonheur.

3.4.4.2 Deuxième expérience

Ci-dessous, nous présentons les courbes de précision (accuracy) et de perte (lose) qui illustrent l'entraînement et les tests de notre modèle avec l'optimiseur RMSprop.

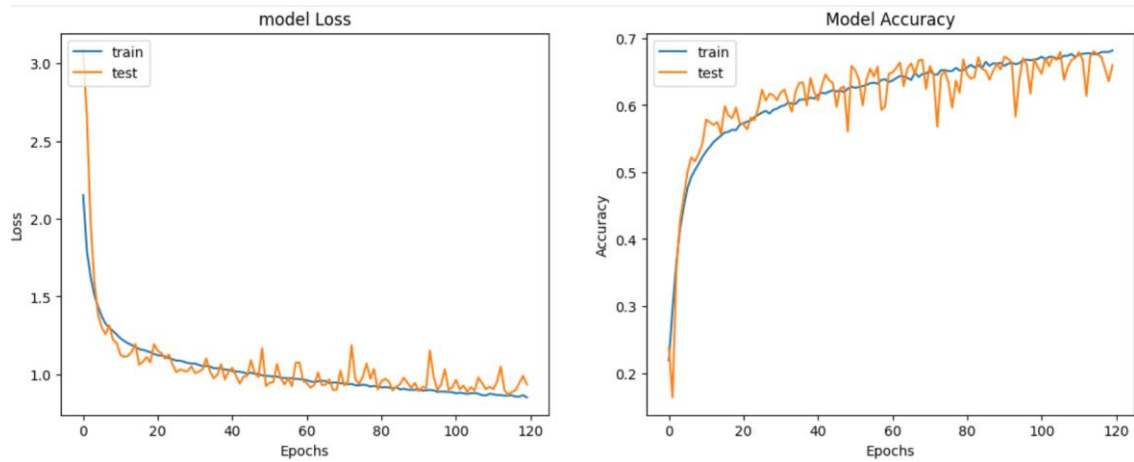


Figure 3.11 Courbes de perte (loss) et de précision (accuracy) RMSprop

La figure ci-dessus (Figure 3.11) présente les courbes de perte (loss) et de précision (accuracy) générées par le modèle au cours de l'entraînement. La courbe de gauche montre la perte de l'ensemble d'entraînement en bleu et celle de l'ensemble de test en orange. On observe que la perte diminue rapidement au début des epochs, puis continue à baisser plus progressivement, indiquant un apprentissage efficace. La courbe de droite illustre la précision de l'ensemble d'entraînement en bleu et celle de l'ensemble de test en orange. La précision augmente significativement au début des epochs, puis se stabilise à mesure que l'entraînement progresse. La précision de validation (val_accuracy) a atteint environ 0.68, ce qui montre une bonne performance du modèle sur les données de test.

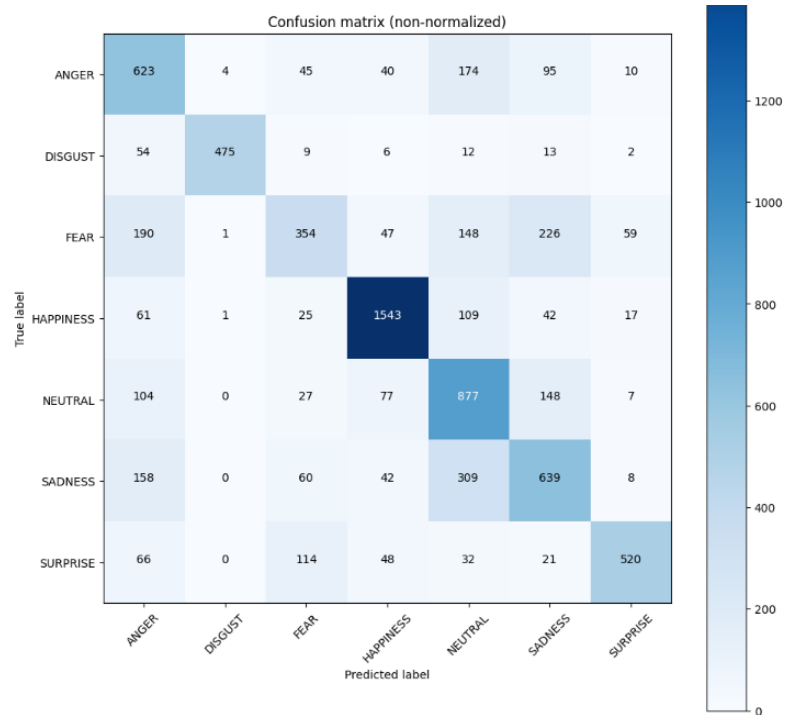


Figure 3.12 Matrice de confusion RMSprop

La figure ci-dessus (Figure 3.12) présente la matrice de confusion de notre modèle, illustrant les performances de classification des différentes émotions. La diagonale de la matrice, qui représente les classifications correctes, montre des valeurs élevées, ce qui indique que le modèle a une bonne capacité à identifier correctement la majorité des émotions. Les erreurs de classification, représentées par les valeurs hors diagonale, existent mais sont moins fréquentes en comparaison. La forte concentration de valeurs élevées sur la diagonale suggère que le modèle est bien entraîné et performant dans sa tâche principale de reconnaissance des émotions. Pour cette raison, nous avons décidé d'utiliser ce modèle dans notre application.

3.5 Présentation de l'application

Pour tester notre modèle et le rendre accessible à des utilisateurs, nous avons décidé de développer une petite application capable de détecter les changements d'émotions. Notre application permettra à toute personne de tester notre modèle en temps réel, en identifiant les émotions à partir séquences vidéo fournies.

En termes d'interface, nous avons souhaité faire simple et intuitive. L'objectif est de garantir une expérience utilisateur fluide et accessible, La figure ci-dessous (Figure 3.13) montre l'interface initiale de notre application.

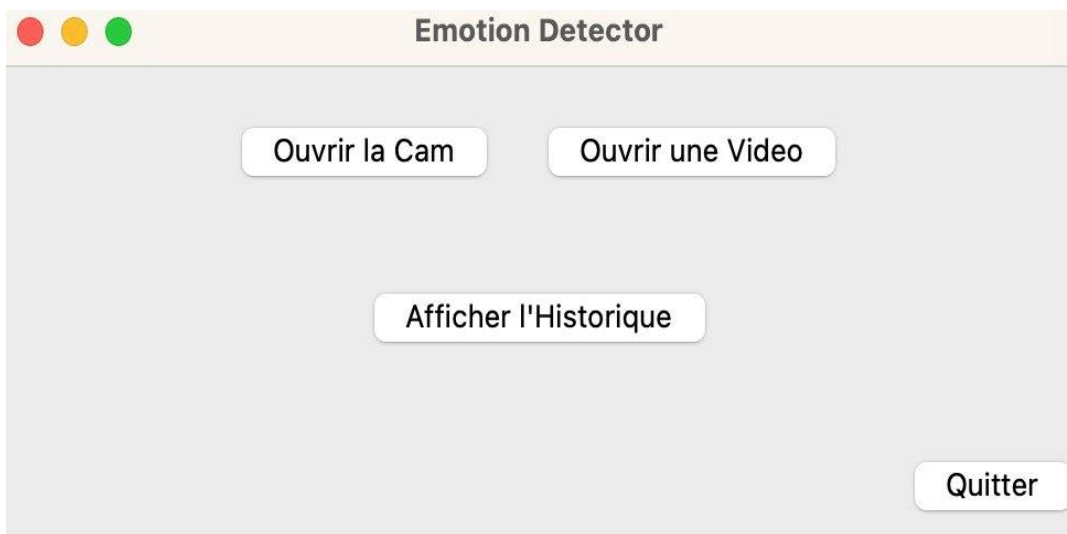


Figure 3.13 Interface initiale de l'application

L'interface contient quatre boutons :

- Le premier bouton « Ouvrir la cam » permet d'ouvrir la webcam du pc et tester le modèle en temps réel.
- Le deuxième bouton «Ouvrir une vidéo » permet de tester le modèle sur une vidéo déjà enregistré.
- Le troisième bouton «Afficher l'historique » permet d'afficher tous les images capturées par notre application.
- Le quatrième bouton «Quitter » permet de quitter l'application.

La figure ci-dessous (Figure 3.14) montre le résultat d'un test effectué avec une vidéo pré-enregistrée. Le visage du sujet est encadré en jaune, et l'émotion dominante, "Neutre", est affichée en vert au-dessus du cadre. En haut de l'image, l'heure de la capture est indiquée. En bas à gauche, les pourcentages représentant les différentes émotions détectées (surprise,

tristesse, neutralité, joie, peur, dégoût, colère) sont listés, accompagnés d'une évaluation générale de l'état émotionnel du patient.

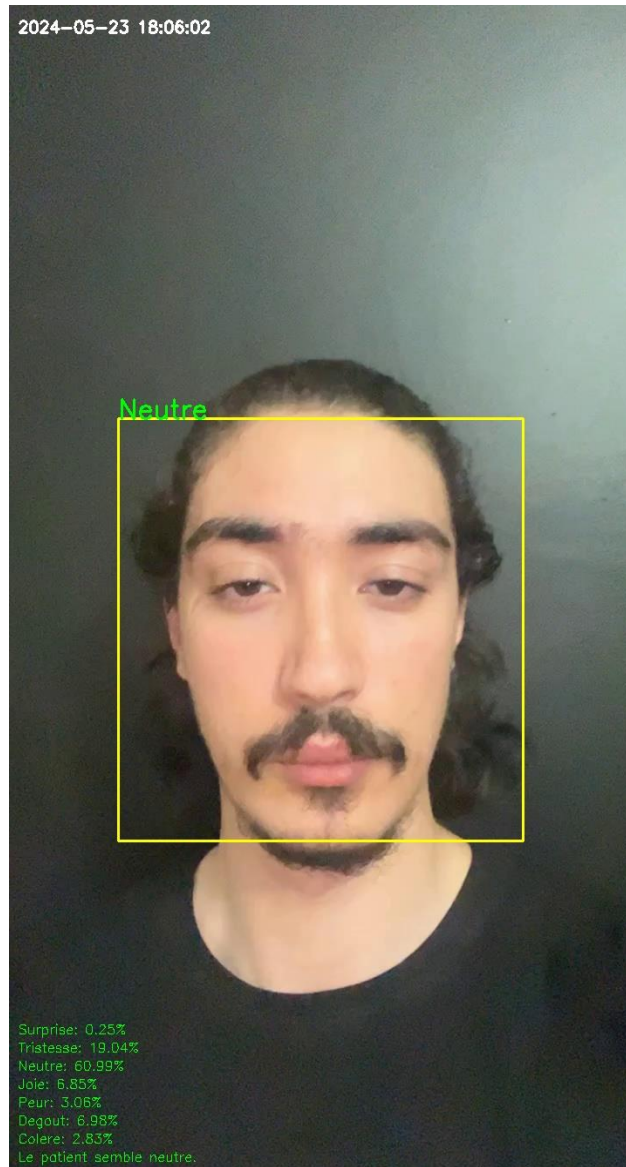


Figure 3.14 Test d'émotion capturée

Chaque changement d'émotions une image est capturée avec toutes les informations précisées avant.

La figure ci-dessous (Figure 3.15) montre l'interface historique de notre application. L'interface affiche les images capturées par l'application avec l'heure de capture, l'état émotionnel du patient, et un lien vers l'image correspondante. Deux boutons permettent de naviguer entre les différentes images.

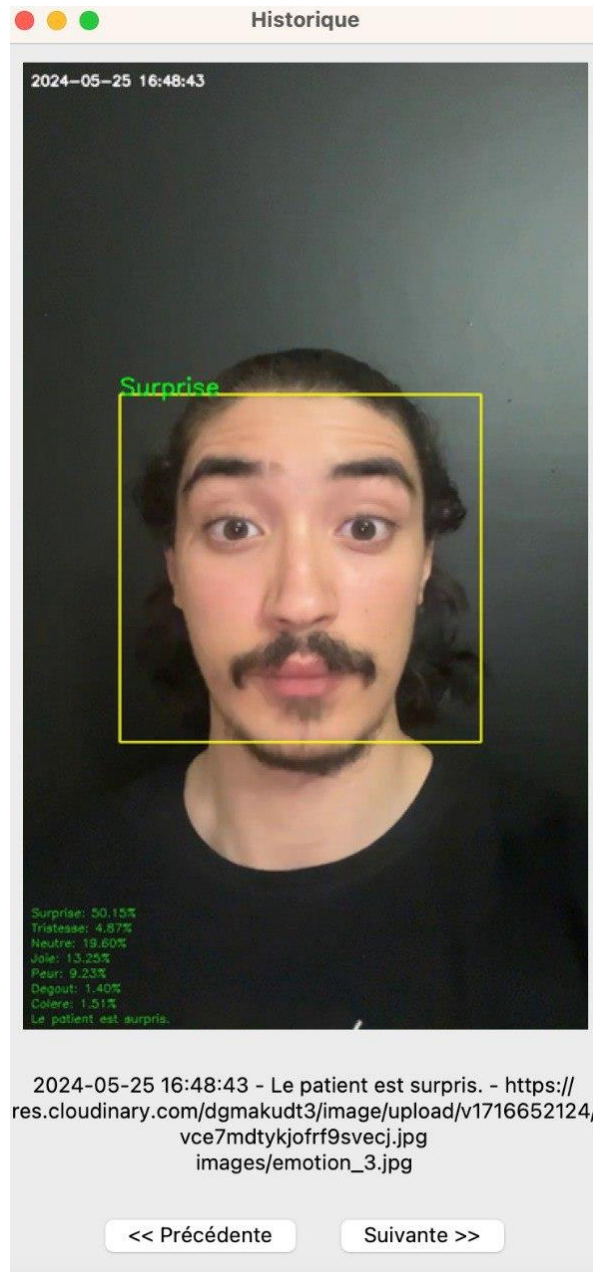


Figure 3.15 Interface historique

3.5.1 Système de notification

Lorsque des patients se trouvent en soins intensifs, la surveillance de leur état émotionnel revêt une importance cruciale. Dans ce contexte, la mise en place d'un système de notification efficace devient essentielle. Cette fonctionnalité permet aux cliniciens de recevoir des alertes en temps réel lorsqu'un patient présente des changements émotionnels importants, leur fournissant ainsi des informations supplémentaires précieuses pour évaluer leur état et ajuster leur traitement en conséquence.

Pour mettre en œuvre ce processus, nous avons utilisé le site Cloudinary pour télécharger les photos capturées par notre application, permettant ainsi d'obtenir un lien vers chaque image. Ensuite, nous avons intégré Twilio dans notre système pour envoyer des SMS au téléphone du personnel chargé de surveiller le patient, les notifiant ainsi des changements émotionnels détectés. Ces messages contiennent des informations essentielles telles que l'émotion du patient, l'heure précise du changement, ainsi qu'un lien direct vers l'image associée, permettant une évaluation immédiate de la situation.

La figure (Figure 3.16) ci-dessous illustre un exemple de message reçu lors de changement d'émotions chez un patient

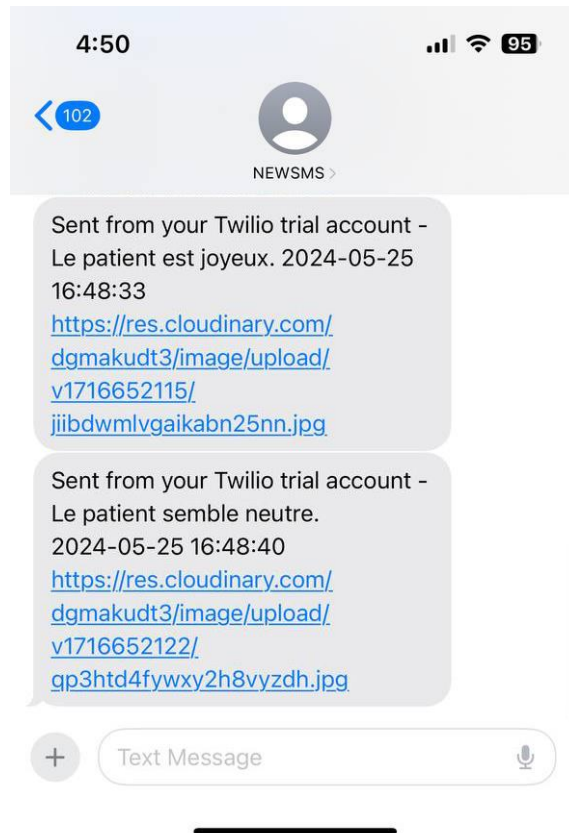


Figure 3.16 Exemple de message reçu

3.6 Conclusion

Ce chapitre a été dédié à la conception de notre propre modèle de réseau de neurones convolutifs pour la reconnaissance des expressions faciales. À travers des tests avec deux optimiseurs distincts, SGD et RMSprop, nous avons observé que RMSprop offrait des performances supérieures. Bien que notre modèle ait démontré une efficacité générale, mais qu'il peut être encore amélioré.

Conclusion Générale

Notre objectif était de concevoir un système de reconnaissance des émotions faciales, en mettant particulièrement l'accent sur son application dans le domaine de la surveillance médicale.

Au bout de ce projet, nous avons pu créer et entraîner notre propre modèle en utilisant des réseaux de neurones convolutifs (CNN) sur une version améliorée de la base de données Fer2013. Le modèle proposé est composé de quatre couches convolutionnelles et de deux couches entièrement connectées. Nous avons testé notre modèle avec deux optimiseurs, SGD et RMSprop, et avons obtenu un meilleur résultat avec RMSprop, atteignant une précision d'environ 68 % sur les données de test.

De plus, nous avons implémenté notre modèle dans une application avec une interface simple, permettant de le tester en temps réel ou sur une vidéo pré-enregistrée. Cette application a été spécialement conçue pour répondre aux besoins des professionnels de la santé en assurant une surveillance continue des émotions des patients, En détectant les changements émotionnels à travers les expressions faciales, elle alerte immédiatement le personnel médical , leur permettant ainsi de réagir rapidement et de prendre les mesures nécessaires.

Malgré les progrès réalisés dans notre projet de reconnaissance des émotions faciales, nous avons dû faire face à plusieurs obstacles significatifs. Notamment, la limitation de nos ressources matérielles s'est avérée être un défi majeur. Nos machines ne disposaient pas de GPU dédiés, ce qui a entravé notre capacité à accélérer les calculs nécessaires à l'entraînement de notre modèle. En conséquence, nous avons été contraints d'utiliser Google Colab, mais sa limitation en termes de temps d'exécution nous a souvent freinés dans nos expérimentations. De plus, la non disponibilité de datasets publics spécifiques aux émotions faciales dans le domaine médical a compliqué la conception de notre modèle.

En termes de perspectives, nous envisageons d'améliorer notre modèle en explorant l'ajout de couches supplémentaires et en l'entraînant sur des bases de données spécifiques à des patients. Parallèlement, pour l'application, nous chercherons à améliorer le système de notification, actuellement limité, afin de fournir une surveillance plus précise et réactive des émotions des patients.

En conclusion, ce projet nous a permis de développer une compréhension approfondie des concepts et des techniques liés à la reconnaissance des émotions faciales et à l'utilisation des réseaux de neurones convolutifs (CNN). Ce projet constitue un premier pas vers une surveillance médicale continue et automatisée, en particulier pour des patients en situation délicate (réanimation).

Bibliographie

- [1] Zabi, K., & Mazouz, K. (2021). « Détection de visage », Mémoire de fin de cycle, Université Larbi Ben M'hidi d'Oum el Bouaghi
- [2] Ghediri, I., & Semri, K. (2021). « La Reconnaissance Des émotions de base Par Les réseaux de neurones : Application de Deep Learning », Mémoire de Fin d'Etudes, Université L'arbi Ben M'Hidi Oum El Bouaghi.
- [3] Azzoune, I., & Kheldoun, N. (2020). « Une approche IA pour la reconnaissance des expressions faciales », Mémoire de Master, Université Akli Mohand Oulhadj de Bouira.
- [4] Kaarouf, M. A., & Bediaf, A. (2022). « Reconnaissance des expressions faciales », Mémoire Master Professionnel, Université Kasdi Merbah-Ouargla.
- [5] Ekman, P. (1992). An argument for basic emotions. *Cognition and Emotion*, 6(3-4), 169-200
- [6] Ekman, P. (1993). Facial expression and emotion. *American Psychologist*, 48(4), 384-392.
- [7] Andreas Lanitis, Christopher J. Taylor, and Timothy F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and machine intelligence*, 19(7) :743–756, 1997.
- [8] Chettouh, H. (2020). « Montage d'un Système de Reconnaissance des Expressions Faciales avec le Deep Learning », Mémoire de Master, Université Mohamed Khider – BISKRA.
- [9] Ghanem, K. (2010). « Reconnaissance des Expressions Faciales à Base d'Informations Vidéo ; Estimation de l'Intensité des Expressions Faciales », Thèse de Doctorat, Université Mentouri de Constantine
- [10] Visage Technologies AB. MPEG-4 Face and Body Animation (MPEG-4 FBA) An overview. Sweden
- [11] Revina I. Michael et al. A survey on human face expression recognition techniques. *J. King Saud Univ. Comput. Inf. Sci.* (2021)
- [12] Khare S.K. et al. Emotion recognition and artificial intelligence: A systematic review (2014–2023) and research recommendations. *Inf. Fusion* (2023)

- [13] « Qu'est-ce que l'apprentissage en profondeur? », <https://www.ibm.com/fr-fr/topics/deep-learning> , Consulté le 08/03/2024
- [14] « Deep learning architectures », <https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures/> , Consulté le 08/03/2024
- [15] Bouadi, D., & Imarazene, T. (2019). « Classification d'images agricoles avec le Deep Learning », Mémoire de master, Université Mouloud Mammeri - Tizi Ouzou.
- [16] Merzougui, Dr. (2021). Support de cours deep learning : Chapitre 3 : Réseau de neurones convolutionnel CNN. Master 2 SDS, Université Batna 2, Algérie.
- [17] « Convolutional Neural Network : Tout ce qu'il y a à savoir », <https://datascientest.com/convolutional-neural-network> , Consulté le 08/03/2024
- [18] «VGG16 and VGG19 », <https://keras.io/api/applications/vgg/> , Consulté le 13/03/2024
- [19] « ResNet and ResNetV2», <https://keras.io/api/applications/resnet/> , Consulté le 13/03/2024
- [20] « InceptionV3 », <https://keras.io/api/applications/mobilenet/> , Consulté le 13/03/2024
- [21] « MobileNet, MobileNetV2, and MobileNetV3 », <https://keras.io/api/applications/mobilenet/> , Consulté le 13/03/2024
- [22] « Xception », <https://keras.io/api/applications/xception/> , Consulté le 13/03/2024
- [23] « DenseNet », <https://keras.io/api/applications/densenet/> , Consulté le 13/03/2024
- [24] « VsCode », <https://code.visualstudio.com/Docs> , Consulté le 14/04/2024
- [25] « Colaboratory », <https://research.google.com/colaboratory/faq.html> , Consulté le 14/04/2024
- [26] « Google Colab Free GPU Tutorial », <https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d> , Consulté le 14/04/2024
- [27] « Cloudinary », <https://cloudinary.com/about> , Consulté le 18/04/2024
- [28] « About Twilio », <https://www.twilio.com/en-us/company> , Consulté le 18/04/2024
- [29] « Python », <https://www.python.org/about/> , Consulté le 18/04/2024
- [30] « About OpenCV », <https://opencv.org/about/> , Consulté le 20/04/2024
- [31] « NumPy », <https://numpy.org/about/> , Consulté le 18/04/2024
- [32] « About Pandas », <https://pandas.pydata.org/about> , Consulté le 18/04/2024

- [33] « Introduction a TensorFlow » , <https://www.tensorflow.org/learn?hl=fr> , Consulté le 18/04/2024
- [34] « About Keras 3 » ,<https://keras.io/about/> , Consulté le 18/04/2024
- [35] « Matplotlib: Visualization with Python » , <https://matplotlib.org/> , Consulté le 18/04/2024
- [36] « tkinter — Python interface to Tcl/Tk » , <https://docs.python.org/3/library/tkinter.html> , Consulté le 18/04/2024
- [37] « Extended and Augmented Google FER » , <https://www.kaggle.com/datasets/prajwalsood/google-fer-image-format> , Consulté le 20/04/2024