

FACULTÉ DES SCIENCES ET TECHNIQUES D'ERRACHIDIA

Mémoire de fin d'études pour l'obtention du diplôme de LST
Option Génie Logiciel

Reconnaissance des émotions par les réseaux de neurones



Réalisé par :

Salhi Abdelmounaim
Sadiki Abdelkarim

Encadré par :

M. Ait khouya Youssef

Soutenu le 15 juillet 2024

Jury composé de :

Mme. Taifi Khaddouj : Présidente

M. Ait Oussous Mohammed : Examinateur

M. Ait Khouya Youssef : Encadrant

Remerciements

Tout d'abord, nous exprimons notre gratitude envers Dieu, le Tout-Puissant et Miséricordieux, pour nous avoir accordé la force, le courage et la patience nécessaires à la réalisation de ce travail.

Un chaleureux remerciement est également adressé aux membres du jury, Mme. Taifi Khaddouj et M. Mohammed Ait Oussous, pour leur intérêt et leur contribution à notre recherche. Leurs suggestions ont apporté une valeur ajoutée à notre travail.

Ensuite, nous tenons à remercier sincèrement notre encadrant, M. AIT KHOUTA YOUSSEF, pour sa confiance, sa disponibilité et ses précieux conseils tout au long de cette période d'encadrement. Ses recommandations avisées ont grandement enrichi notre réflexion.

Nous souhaitons également exprimer notre gratitude envers l'équipe pédagogique de la licence en sciences et techniques "Génie logiciel" ainsi qu'à la faculté des sciences et techniques d'Errachidia pour nous avoir offert cette opportunité enrichissante tout au long de nos études.

Enfin, nous sommes profondément reconnaissants envers nos familles pour leur soutien indéfectible, leurs encouragements et leurs prières. Leur appui moral et spirituel a été d'une importance capitale pour mener à bien ce travail.

Dédicace

Nous dédions ce modeste travail à tous nos enseignants durant nos années d'études avec lesquels nous avons beaucoup appris .

A nos très chers parents pour leur soutien et encouragement durant toutes mes années d'études et sans lesquels nous n'aurons jamais réussis.

A toute la famille, a tous nos amis ainsi qu'à toutes les personnes que nous avons connues, qui nous ont aidées, soutenues et encouragées.

Table des matières

Remerciements	i
Dédicace	ii
Introduction générale	1
1 Vision par ordinateur (Computer Vision)	2
1.1 Introduction	2
1.2 Caractéristiques de l'image	2
1.2.1 Définition d'une image	2
1.2.2 Types d'images	3
1.2.3 Dimension et résolution	3
1.3 Les émotions de base	3
1.4 Architecture de système de reconnaissance des expressions faciales	4
1.4.1 Détection du visage	4
1.4.2 Extraction des caractéristiques	5
1.4.3 Classification	5
1.5 Intelligence artificiel	6
1.6 Machine learning	6
1.7 Deep learning	6
1.7.1 Algorithmes populaires de Deep learning	7
1.7.2 Différence entre Deep learning et Machine learning	7
1.8 Conclusion	8
2 Réseau de neurones convolutifs	10
2.1 Introduction	10
2.2 L'algorithme CNN Convolutional Neural NetWork	10
2.2.1 Définition	10
2.2.2 Principe d'architecture d'un CNN	11
2.2.3 Les architectures populaires de CNN	11
2.2.3.1 Architecture adopté dans notre Projet	11

2.2.3.2	Architecture VGG-16	12
2.2.4	Les couches de CNN	12
2.2.4.1	Couche de convolution (CONV)	12
2.2.4.2	Couche Pooling	14
2.2.4.3	Fonction de correction Relu	15
2.2.4.4	Couche Flatten	16
2.2.4.5	Couche entièrement connectée (FC)	16
2.2.4.6	Dropout	17
2.3	Classification	18
2.3.1	Introduction	18
2.3.2	Définition	18
2.3.3	Différent type de classification	18
2.3.4	Différentes approches de classification	19
2.3.5	Propagation directe	20
2.3.6	Fonction Softmax	20
2.3.7	Fonction coût(LOSS FUNCTION)	21
2.3.8	Rétro propagation (Backpropagation)	21
2.3.9	Optimisation	21
2.3.9.1	Descente du gradient (Gradient Descent)	21
2.3.9.2	Le taux d'apprentissage	22
2.3.9.3	Optimiseur momentum	23
2.3.9.4	Optimiseur Rmsprop	23
2.3.9.5	Adam	24
2.4	Conclusion	24
3	Conception de l'application	26
3.1	Outils de conception	26
3.2	Équipe de développement	27
3.3	Diagramme de gantt	27
3.4	Diagrammes	28
3.4.1	Diagramme de classe	28
3.4.2	Diagramme de cas d'utilisation	29
3.4.3	Diagramme de séquence	30
3.4.3.1	Diagramme de séquence de la fonctionnalité 's'inscrire'	30
3.4.3.2	Diagramme de séquence de la fonctionnalité 's'authentifier'	31
3.4.3.3	Diagramme de séquence de la fonctionnalité 'Détecter émotion'	32
3.5	Conclusion	32

4	Implémentation et Résultats	34
4.1	Outils de développement	34
4.1.1	Langages de programmation et bibliothèques	34
4.1.2	Environnements	36
4.2	Matériel et méthodes	36
4.3	Le jeu de donnée(DataSet)	37
4.4	Base de données SQL	38
4.5	Définition des couches	39
4.6	Entraînement du modèle	40
4.7	Résultats du modèle	41
4.8	Matrice de confusion	43
4.9	Présentation des maquettes	44
4.10	Conclusion	46
	Conclusion générale	47

Abréviations

AI Artificiel Intelligence

CNN Convolutional Neural Network

RNN Recurrent Neural Network

FC Fully Connected Layers

ReLU Fonction réelle non-linéaire (Rectified Linear Units)

UML Unified Modeling Language

RVB Rouge Vert Bleu

SGBD Système de Gestion de Base de Données

SQL Structured Query Language

Adam Adaptive Moment

RMSprop Root Mean Square Propagation

Table des figures

1.1	Architecture d'un système de reconnaissance des expressions faciales	4
1.2	Détection des visages	5
1.3	Intelligence artificielle, Machine Learning et Deep Learning.	7
1.4	Difference entre deep learning et Machine learning.	8
2.1	Architecture VGG-16	12
2.2	Filtre de Convolution.	13
2.3	Application du filtre sur le visage.	14
2.4	Différence entre le Max pooling et l'Average pooling	15
2.5	L'addition de la couche pooling au modèle	15
2.6	Fonction ReLU	16
2.7	La Couche Flatten	16
2.8	Couches entièrement connectée (FC)	17
2.9	code d'empilement de la couche entièrement connecter	17
2.10	Opération du Dropout	18
2.11	Propagation directe	20
2.12	Différentes tailles de taux d'apprentissage	22
3.1	Diagramme de Gantt	27
3.2	Diagramme de classes.	28
3.3	Diagramme de cas d'utilisation.	29
3.4	Diagramme de séquence de la fonctionnalité 's'inscrire'.	30
3.5	Diagramme de séquence de la fonctionnalité 's'authentifier'.	31
3.6	Diagramme de séquence de la fonctionnalité 'Détecter émotion'.	32
4.1	Fiche technique GOOGLE COLAB	36
4.2	Exemple des images dans la BDD fer-2013	38
4.3	Capture du base de données	38
4.4	Structure du modèle adopté	39
4.5	Entraînement du modèle	41
4.6	Résultats du modèle	41

4.7	Graphe de précision et perte	42
4.8	Matrice de confusion	43
4.9	Page d'inscription	44
4.10	Page de login	44
4.11	Page d'accueil	45
4.12	Page 'MODE IMAGE'	45
4.13	Page 'MODE LIVE'	46

Introduction générale

La détection des émotions, également connue sous le nom de reconnaissance des émotions faciales, est un domaine fascinant dans le domaine de l'intelligence artificielle et de la vision par ordinateur. Cela implique l'identification et l'interprétation des émotions humaines à partir des expressions faciales. La détection précise des émotions a de nombreuses applications pratiques, notamment l'interaction homme-machine, l'analyse des commentaires des clients et la surveillance de la santé mentale. Les réseaux de neurones convolutifs (CNN) sont devenus un outil puissant dans ce domaine, révolutionnant la façon dont nous comprenons et traitons les signaux émotionnels des images.

Les émotions sont un aspect fondamental de la communication et du comportement humain. Ils s'expriment à travers les expressions faciales, le langage corporel et le ton de la voix. Bien que tous ces signaux soient importants, les expressions faciales sont souvent les indicateurs d'émotion les plus visibles et les plus fiables. Pour cette raison, le visage est devenu un sujet de recherche dans de nombreux domaines scientifiques tels que la psychologie, la médecine et l'informatique. La détection des émotions à l'aide des CNN se concentre principalement sur l'analyse des expressions faciales pour déterminer l'état émotionnel d'un individu.

Notre projet vise principalement à créer une application fiable pour détecter les émotions à partir des expressions faciales en utilisant l'apprentissage automatique profond (CNN) et divers environnements et bibliothèques tels que Tensorflow, Keras, Tkinter, CV2, Google colab, JupyterNotebook. .Nous avons considéré FER-13 comme le jeu de données qui contient 35 887 images en gris de taille 48*48 pixels et Mysql un SGBD pour stocker les informations des utilisateurs pour qu'ils puissent s'authentifier. La répartition de ce travail est comme suit :

- Dans le chapitre initial, nous aborderons la vision par ordinateur, où nous examinerons comment l'ordinateur perçoit et traite les images, ainsi qu'une vision globale de l'intelligence artificielle. .
- Le second chapitre, est consacré à la présentation de réseaux de neurones convolutifs ou on va mettre l'accent sur notre modèle.
- Dans le troisième chapitre, nous aborderons tout d'abord la conception de notre application, puis nous présenterons les divers outils utilisés pour le développement de l'application.
- Le chapitre suivant vise à exposer la mise en œuvre et les résultats de notre projet.
- Et enfin, nous terminerons ce mémoire par une conclusion générale.

Chapitre 1

Vision par ordinateur (Computer Vision)

1.1 Introduction

La vision par ordinateur est un domaine en plein essor de l'intelligence artificielle et de l'informatique, qui vise à permettre aux machines de comprendre et d'interpréter des données visuelles issues du monde réel. Elle englobe une multitude de techniques pour acquérir, traiter, analyser et comprendre les images ou les vidéos afin d'en extraire des informations significatives.

L'une des approches les plus prometteuses et puissantes dans ce domaine est l'utilisation des réseaux de neurones convolutifs (CNN). Les CNN sont une classe de réseaux de neurones profonds, spécialement conçus pour traiter les données structurées sous forme de grille, comme les images. Grâce à leur capacité à apprendre automatiquement des caractéristiques discriminantes à partir des données visuelles, les CNN ont révolutionné plusieurs applications de vision par ordinateur, notamment la reconnaissance d'objets, la segmentation d'images, et la détection de visages.

Une des applications spécifiques et fascinantes de la vision par ordinateur est la détection des émotions. Ce domaine vise à développer des systèmes capables d'identifier et de classifier les émotions humaines à partir d'images ou de vidéos de visages. En utilisant les CNN, ces systèmes peuvent apprendre à reconnaître des expressions faciales subtiles et complexes, permettant ainsi une analyse émotionnelle précise et en temps réel. La détection des émotions trouve des applications dans des domaines variés tels que la santé mentale, l'interaction homme-machine, le marketing, et la surveillance.

1.2 Caractéristiques de l'image

1.2.1 Définition d'une image

Une image est une représentation visuelle d'une scène, d'un objet ou d'une personne, capturée à travers divers moyens tels que la photographie, la peinture ou les dispositifs numériques. Dans le contexte de l'informatique et de la vision par ordinateur, une image est généralement définie comme une matrice de pixels, chaque pixel représentant un point de couleur ou de luminosité. Les images peuvent être en niveaux de gris, où chaque pixel a une valeur unique représentant l'intensité de la lumière, ou en couleur, où chaque pixel est décrit par un ensemble de valeurs correspondant aux composantes rouge, verte et bleue (RGB). Les images sont des données fondamentales utilisées dans de nombreuses applications de traitement d'image et de vision par ordinateur.[19]

1.2.2 Types d'images

Image couleur RVB : l'œil humain analyse la couleur à l'aide de trois types de cellules photoélectriques "Cône". Ces cellules sont sensibles aux basses, moyennes ou hautes fréquences (rouge, vert, bleu). Par conséquent, pour représenter la couleur d'un pixel, il est nécessaire d'entrer trois nombres correspondants avec dosage de trois couleurs primaires : rouge, vert, bleu. De cette façon, nous pouvons présenter l'image couleurs avec trois matrices correspondant chacune à une couleur primaire .

Image d'Intensité : C'est une matrice dans laquelle chaque élément est un nombre réel compris entre 0 (noir) et 1 (blanc). On parle aussi d'image en niveaux de gris car les valeurs comprises entre 0 et 1 stand pour différentes nuances de gris .

Image binaire : Une image binaire est un tableau rectangulaire avec une valeur d'élément de 0 ou 1. Lors de la visualisation d'une telle image, les zéros sont représentés par du noir et les uns par du blanc.[19]

1.2.3 Dimension et résolution

La dimension est la taille de l'image, elle se présente sous forme d'une matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels).

Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image .

Par contre, la résolution est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateur, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre) .

On utilise aussi le mot résolution pour désigner le nombre total de pixels horizontaux et verticaux sur un moniteur. Plus ce nombre est grand, plus la résolution est meilleure.[19]

1.3 Les émotions de base

Les expressions faciales dans diverses cultures et a dénombré 7 émotions fondamentales : La neutralité, la joie, la colère, la peur, la tristesse, la surprise, le dégoût.[3]

1. neutralité : est un état émotionnel caractérisé par l'absence de sentiments intenses ou marqués, représentant un équilibre et un calme émotionnel.

2. Peur : Une inquiétude plus ou moins réaliste face à la situation. Les personnes peuvent ressentir des frissons, une accélération du rythme cardiaque, des changements dans le rythme respiratoire et même une perte de conscience.

3. Joie : Une manifestation de joie profonde et d'abondance, exprimée par une augmentation ou une diminution de la fréquence cardiaque. La respiration est souvent plus ample et parfois accompagnée d'une sensation de calme ou d'excitation.

4. Dégoût : Rejet d'une situation ou d'une personne, parfois de manière déraisonnable. Vous pouvez être dégoûté par l'apparence de la nourriture, des odeurs, des lieux, des personnes ou

des actions.

5. La tristesse : Douleur émotionnelle causée par un sentiment de manque ou de perte. Parfois, il se manifeste par des pleurs, une perte d'appétit ou des symptômes de sevrage (parfois bénéfiques, mais pas toujours).

6. Colère : exprimée face à l'agressivité, la colère ou la frustration. Le rythme cardiaque et la respiration augmentent alors, souvent accompagnés de contractions corporelles, de poings et de mâchoires serrés et de sourcils froncés.

7. La surprise : Cela se produit lorsque vous êtes confronté à un événement inattendu. Elle peut évoquer les mêmes qualités que la peur car elle ne dure que peu de temps et peut muter en une autre émotion (colère, peur, dégoût).

1.4 Architecture de système de reconnaissance des expressions faciales

Un système qui effectue une reconnaissance automatique des expressions faciales est généralement composé de trois modules principaux, comme illustre dans la figure ci-dessous. Le premier module consiste à détecter et enregistrer la région du visage dans les images ou les séquences d'images d'entrée. Il peut s'agir d'un détecteur pour détecter le visage dans chaque image ou simplement détecter le visage dans la première image, puis suivre le visage dans le reste de la séquence vidéo. Le deuxième module consiste à extraire et représenter les changements faciaux causés par les expressions faciales. Le dernier module détermine une similarité entre l'ensemble des caractéristiques extraites et un ensemble de caractéristiques de référence. D'autres filtres ou modules de prétraitement de données peuvent être utilisés entre ces modules principaux pour améliorer les résultats de détection, d'extraction de caractéristiques ou de classification. (Voir la figure 1.1)

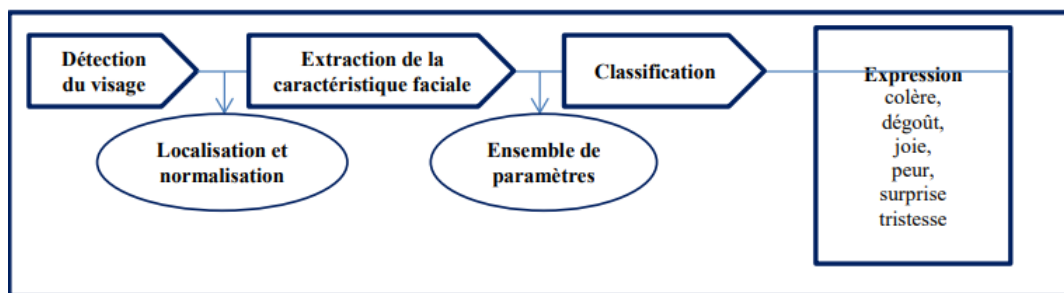


FIGURE 1.1 – Architecture d'un système de reconnaissance des expressions faciales

1.4.1 Détection du visage

La détection de visage est le processus par lequel un système informatique identifie et localise les visages humains dans des images ou des vidéos. C'est une étape essentielle pour de nombreuses applications de vision par ordinateur, telles que la reconnaissance faciale, l'analyse des expressions et la réalité augmentée.



FIGURE 1.2 – Détection des visages

L'utilisation de Haar Cascade pour la détection de visage repose sur un ensemble de caractéristiques appelées Haar-like features. Ces caractéristiques sont utilisées pour classifier les différentes régions d'une image en fonction de leur similarité avec les motifs des visages humains. L'algorithme Haar Cascade, proposé par Paul Viola et Michael Jones, utilise une approche de type "cascade" où des classificateurs simples et rapides sont appliqués en série pour éliminer rapidement les régions non faciales et concentrer le calcul sur les régions potentiellement intéressantes. (**Voir la figure 1.2**)

Cette méthode est particulièrement efficace et rapide, permettant une détection en temps réel des visages dans des applications pratiques.

1.4.2 Extraction des caractéristiques

L'extraction des caractéristiques est le processus par lequel des informations pertinentes et significatives sont extraites des images de visages afin de permettre à un modèle d'apprentissage automatique de reconnaître et de classifier les émotions. Dans le cadre de notre projet de détection des émotions par réseaux de neurones convolutifs (CNN), l'extraction des caractéristiques implique l'utilisation des couches convolutives du réseau pour identifier automatiquement les motifs et les détails dans les images de visages qui sont les plus représentatifs des différentes émotions humaines. Ces caractéristiques peuvent inclure des éléments tels que les expressions faciales, les contours des yeux, de la bouche et d'autres parties du visage, qui sont ensuite utilisés pour entraîner le modèle à distinguer entre des états émotionnels variés tels que la joie, la tristesse, la colère, etc.[8]

1.4.3 Classification

La classification est le processus de classification des éléments en groupes spécifiques, et dans le contexte de l'apprentissage automatique, cette classification est effectuée par un ordinateur. Pensez à quel point ce serait génial si votre ordinateur pouvait faire la différence entre vous et un étranger, ou distinguer une pomme de terre d'une tomate, ou décider si une certaine performance méritait un A ou un F. Soudain, le concept devient beaucoup plus intéressant. Dans l'apprentissage automatique supervisé, la classification est une tâche essentielle qui implique l'utilisation de données étiquetées pour l'apprentissage. Dans les domaines de l'apprentissage automatique et des statistiques, la classification fait référence au problème consistant à attribuer une nouvelle observation à l'une de plusieurs classes ou souspopulations prédéfinies. Cette détermination est basée sur un ensemble de données de d'entraînement, où les observations sont connues pour appartenir à des catégories spécifiques.[8]

Les étapes d'une classification

- Sélection des données à utiliser.
- Calcul de la similarité entre les n individus en se basant sur les données initiales.
- Choix d'un algorithme de classification et exécution de celui-ci.
- Interprétation des résultats obtenus.
- Évaluation de la qualité de la classification.
- Description des classes obtenues.

1.5 Intelligence artificiel

En premier lieu ,nous devons définir clairement de quoi nous parlons lorsque nous mentionnons L'Intelligence Artificielle(IA).

L'intelligence artificielle englobe l'ensemble des théories et techniques visant à simuler l'intelligence humaine, incluant le raisonnement, l'apprentissage, la planification etc..., à l'aide des programmes informatiques complexes. Cependant, le terme "intelligence artificielle" est souvent utilisé pour désigner spécifiquement l'apprentissage automatique et l'apprentissage en profond.

1.6 Machine learning

Le Machine Learning ou apprentissage automatique est un domaine scientifique, et plus particulièrement une sous-catégorie de l'intelligence artificielle. Elle consiste à laisser des algorithmes découvrir des « patterns », à savoir des motifs récurrents, dans les ensembles de données. Ces données peuvent être des chiffres, des mots, des images, des statistiques...

Tout ce qui peut être stocké numériquement peut servir de données pour le Machine Learning. En décelant les patterns dans ces données, les algorithmes apprennent et améliorent leurs performances dans l'exécution d'une tâche spécifique.

Pour résumer, les algorithmes de Machine Learning apprennent de manière autonome à effectuer une tâche ou à réaliser des prédictions à partir de données et améliorent leurs performances au fil du temps. Une fois entraîné, l'algorithme pourra retrouver les patterns dans de nouvelles données.

1.7 Deep learning

Le deep learning ou apprentissage profond est un type d'intelligence artificielle dérivé du machine learning (apprentissage automatique) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées.

Le deep Learning s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain. Ce réseau est composé de dizaines voire de centaines de «couches» de neurones, chacune recevant et interprétant les informations de la couche précédente. Le système apprendra par exemple à reconnaître les lettres avant de s'attaquer aux mots dans un texte, ou détermine s'il y a un visage sur une photo avant de découvrir de quelle personne il s'agit.(Voir la figure 1.3)

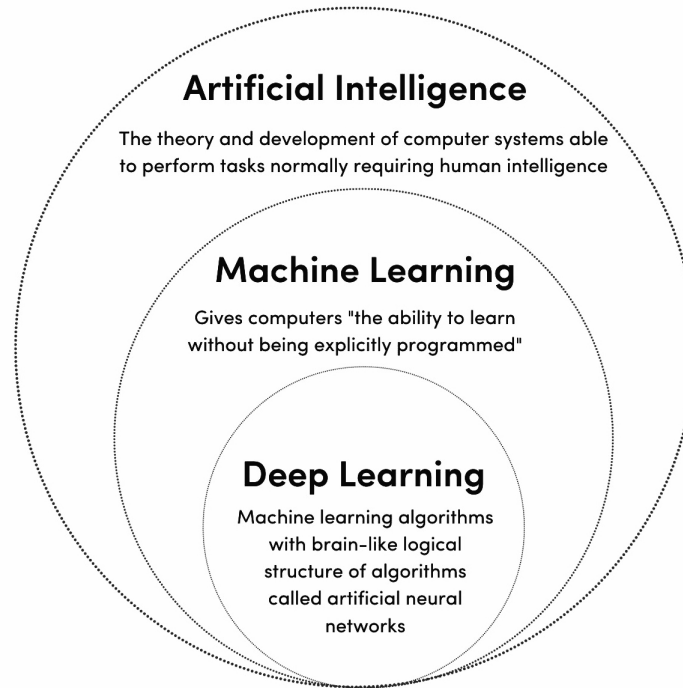


FIGURE 1.3 – Intelligence artificielle, Machine Learning et Deep Learning.

1.7.1 Algorithmes populaires de Deep learning

- 1. Convolutional Neural Networks (CNNs)** sont largement utilisés pour les tâches de vision par ordinateur telles que la détection des émotions, la classification d'images, la détection d'objets et la segmentation d'images.
- 2. Recurrent Neural Networks (RNNs)** sont conçus pour traiter des données séquentielles et sont couramment utilisés pour les tâches de traitement du langage naturel (NLP) et de reconnaissance vocale.
- 3. Transformer Networks** ont révolutionné le traitement du langage naturel en éliminant la nécessité des structures récurrentes et convolutives, en se concentrant sur des mécanismes d'attention.

1.7.2 Différence entre Deep learning et Machine learning

Le Deep Learning et le Machine Learning sont deux sous-domaines de l'Intelligence Artificielle (IA) qui utilisent des algorithmes et des données pour apprendre et faire des prédictions. Voici quelques différences clés entre eux :

- 1. Intervention humaine :** Dans les systèmes de Machine Learning, un humain doit identifier et coder à la main les caractéristiques appliquées en fonction du type de données (par exemple, la valeur des pixels, la forme, l'orientation), tandis qu'un système de Deep Learning essaie d'apprendre ces caractéristiques sans intervention humaine supplémentaire.
- 2. Données :** Le Machine Learning peut être formé sur des ensembles de données plus petits, tandis que le Deep Learning nécessite de grandes quantités de données.

3. Apprentissage : Le Machine Learning nécessite plus d'intervention humaine pour corriger et apprendre, tandis que le Deep Learning apprend de ses propres erreurs environnementales et passées.

4. Corrélations : Le Machine Learning fait des corrélations simples et linéaires, tandis que le Deep Learning fait des corrélations non linéaires et complexes.

5. Matériel : Le Machine Learning peut être formé sur une unité centrale de traitement (CPU), tandis que le Deep Learning nécessite une unité de traitement graphique (GPU) spécialisée pour la formation.

6. Précision : Le Deep Learning a généralement une précision plus élevée et nécessite une formation plus longue, tandis que le Machine Learning a une précision plus faible et nécessite une formation plus courte. En termes généraux, le Deep Learning est un sous-ensemble du Machine Learning, et le Machine Learning est un sous-ensemble de l'IA. Vous pouvez les visualiser comme une série de cercles concentriques qui se chevauchent, avec l'IA occupant le plus grand, suivi du Machine Learning, puis du Deep Learning.

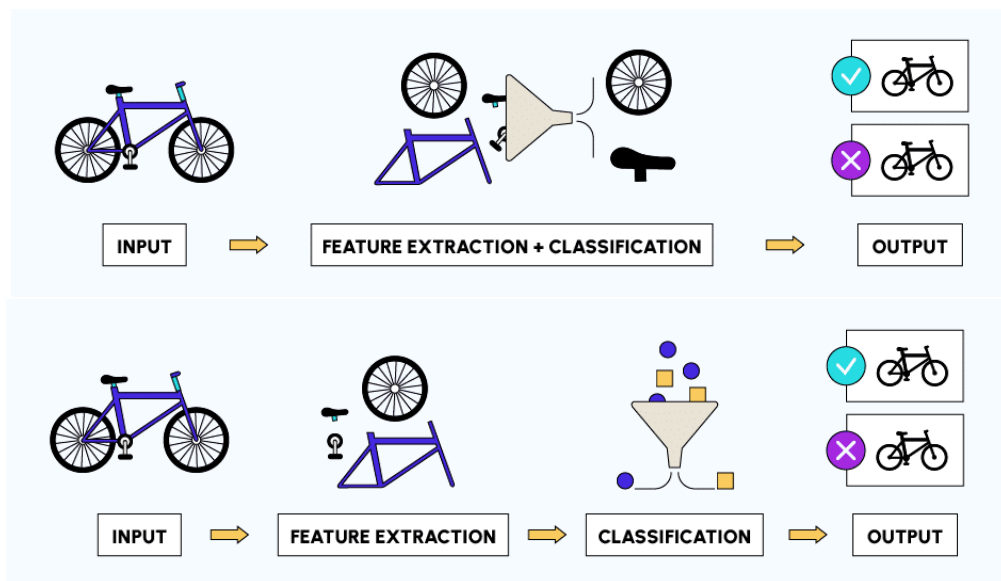


FIGURE 1.4 – Difference entre deep learning et Machine learning.

1.8 Conclusion

En conclusion, ce chapitre a fourni une vue d'ensemble des concepts fondamentaux de l'intelligence artificielle (IA), du machine learning (ML) et du deep learning (DL), ainsi que de leur application dans le domaine de la vision par ordinateur. Nous avons défini l'image comme une matrice de pixels représentant des données visuelles et exploré l'importance des caractéristiques extraites de ces images pour permettre une analyse efficace par les systèmes d'apprentissage automatique.

Nous avons mis en lumière les sept émotions de base , ainsi que la neutralité, et expliqué comment ces états émotionnels peuvent être détectés à partir des expressions faciales en utilisant des réseaux de neurones convolutifs (CNN).

L'algorithme de détection de visage basé sur Haar Cascade a été présenté comme une méthode efficace pour identifier les régions faciales dans les images, tandis que l'extraction des caractéristiques et la classification ont été décrites comme des étapes cruciales pour la reconnaissance et l'analyse des émotions.

En synthèse, ce chapitre établit les bases théoriques et techniques essentielles pour comprendre et développer des systèmes de détection des émotions par vision par ordinateur, posant ainsi les fondations pour les chapitres suivants, qui aborderont les aspects pratiques et expérimentaux de notre projet.

Chapitre 2

Réseau de neurones convolutifs

2.1 Introduction

L'apprentissage profond, également connu sous le nom de deep learning, est une branche de l'intelligence artificielle qui vise à imiter le fonctionnement du cerveau humain pour traiter des données et effectuer des tâches complexes. En utilisant des réseaux de neurones artificiels, le deep learning permet aux machines d'apprendre à reconnaître des motifs et des caractéristiques dans les données de manière autonome, sans être explicitement programmées pour des tâches spécifiques. Cette capacité à extraire des informations à partir de grandes quantités de données non structurées a révolutionné de nombreux domaines, y compris la vision par ordinateur, le traitement du langage naturel, la reconnaissance vocale, la recommandation de contenu et bien plus encore. Cette introduction donne un aperçu général de la puissance et du potentiel de l'apprentissage profond dans le domaine de l'intelligence artificielle.[16]

2.2 L'algorithme CNN Convolutional Neural NetWork

2.2.1 Définition

Dans le domaine de l'apprentissage en profondeur, les réseaux de neurones convolutifs (CNN/ConvNet) sont une classe de réseaux de neurones profonds largement utilisés pour l'analyse d'images visuelles. Contrairement à ce que l'on pourrait penser généralement lorsqu'on évoque les réseaux de neurones, les CNN ne se contentent pas d'utiliser des multiplicateurs matriciels. Ils font appel à une technique spéciale appelée "bypass" ou "skip connection".

La convolution, en mathématiques, est une opération arithmétique qui permet de combiner deux fonctions pour en produire une troisième, exprimant ainsi la manière dont la forme de l'une est modifiée par l'autre. Dans le contexte des CNN, la convolution est utilisée pour extraire des caractéristiques des images en appliquant des filtres spécifiques sur des régions locales. Cela permet aux CNN de capturer des motifs et des informations significatives au sein des images. [7]

En résumé, les réseaux de neurones convolutifs (CNN/ConvNet) sont une classe de réseaux de neurones profonds largement utilisés dans l'analyse d'images. Ils se distinguent par l'utilisation de la convolution, une opération mathématique qui permet de capturer des caractéristiques visuelles importantes. Les CNN utilisent également des connexions spéciales appelées "bypass" ou "skip connection" pour améliorer la performance et la capacité d'apprentissage du réseau.

2.2.2 Principe d'architecture d'un CNN

Un réseau de neurones convolutif n'est pas seulement un réseau neuronal profond avec de nombreuses couches cachées. Il s'agit plutôt d'un réseau profond qui simule le fonctionnement du cortex visuel du cerveau pour reconnaître et classifier des images ou des vidéos, et pour découvrir un objet ou même une partie dans une image.

Le concept et le fonctionnement des réseaux de neurones convolutifs est différent des autres réseaux de neurones, en effet un réseau neuronal convolutif comporte deux parties distinctes avec une entrée dans laquelle une image en forme de matrice de pixels bidimensionnelle (avec 2 dimensions, noir et blanc), ou une image couleur avec 3 dimensions (couleurs : rouge, vert et bleu) ou une image multidimensionnelle (image satellitaire) .

La première partie d'un réseau de neurones convolutif est la partie convolutionnelle qui sert à extraire les caractéristiques de l'image. Ensuite, l'image passe par le fichier de séquence de filtre, ou le noyau d'enroulement, ce qui conduit à la création d'une nouvelle image appelée cartes de convolution. Généralement, les filtres intermédiaires réduisent la résolution de l'image.

Ensuite, les cartes des caractéristiques sont aplaties dans un vecteur de caractéristiques pour former les données d'entrée de la partie de couche entièrement connectée. Le rôle principal de cette couche (complètement connectée) est de combiner les caractéristiques contenues dans le vecteur de son entrée pour la classification des images.[12]

2.2.3 Les architectures populaires de CNN

2.2.3.1 Architecture adopté dans notre Projet

L'architecture du modèle de détection des émotions qu'on ait est utilisé pour ce projet est définie comme suit :

Input Layer (Couche d'entrée)

Input Shape : (48, 48, 1) — Les images d'entrée sont de taille 48x48 pixels en échelle de gris (1 canal).

Premier Block Convolutionnel

Couche Conv2D : 32 filtres, taille de noyau 3x3, activation ReLU.

Couche Conv2D : 64 filtres, taille de noyau 3x3, activation ReLU.

Couche MaxPooling2D : Taille de la fenêtre de pooling 2x2.

Couche Dropout : Taux de dropout de 25

Second Block Convolutionnel

Couche Conv2D : 128 filtres, taille de noyau 3x3, activation ReLU.

Couche MaxPooling2D : Taille de la fenêtre de pooling 2x2.

Couche Conv2D : 128 filtres, taille de noyau 3x3, activation ReLU.

Couche MaxPooling2D : Taille de la fenêtre de pooling 2x2.

Couche Dropout : Taux de dropout de 25%.

Les couches entièrement connectées

Couche Flatten : Aplatie les résultats des couches précédentes pour les préparer à la couche dense.

Couche Dense : 1024 neurones, activation ReLU.

Couche Dropout : Taux de dropout de 50%.

Couche Dense : 7 neurones, activation Softmax (correspondant aux 7 classes d'émotions : colère, dégoût, peur, bonheur, tristesse, surprise, et neutre).[4]

2.2.3.2 Architecture VGG-16

L'architecture VGG-16 est un Réseau de neurones convolutifs (CNN) conçu pour les tâches de classification d'images. Il a été introduit par le Visual Geometry Group de l'Université d'Oxford. VGG-16 se caractérise par sa simplicité et son architecture uniforme, ce qui le rend facile à comprendre et à mettre en œuvre.

La configuration VGG-16 se compose généralement de 16 couches, dont 13 couches convolutives et 3 couches entièrement connectées. Ces couches sont organisées en blocs, chaque bloc contenant plusieurs couches convolutives suivies d'une couche de pooling maximum pour le sous-échantillonnage[14]. (Voir la figure 2.1)

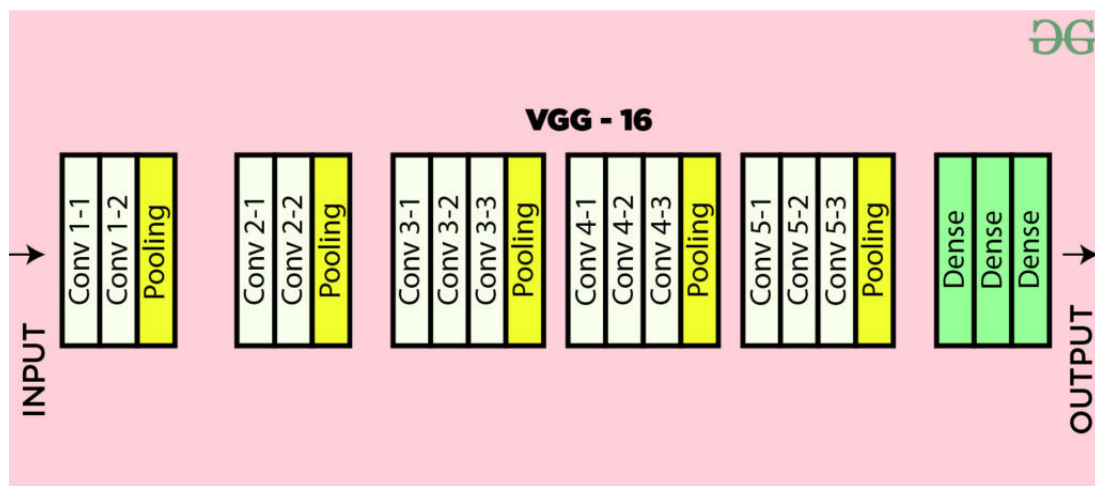


FIGURE 2.1 – Architecture VGG-16

2.2.4 Les couches de CNN

2.2.4.1 Couche de convolution (CONV)

La couche de convolution est un élément essentiel des réseaux neuronaux convolutifs (CNN) et est généralement la première couche de ces réseaux.

Son objectif est de détecter la présence d'un ensemble de caractéristiques dans les images en entrée. Pour cela, elle effectue une opération de filtrage par convolution : elle fait "glisser" une fenêtre représentant la caractéristique souhaitée sur l'image et calcule le produit de convolution entre la fenêtre et chaque région balayée de l'image. Dans ce contexte, une caractéristique est considérée comme un filtre, les deux termes étant équivalents.

La couche de convolution reçoit plusieurs images en entrée et effectue la convolution de chaque image avec chaque filtre. Les filtres correspondent exactement aux caractéristiques que l'on souhaite détecter dans les images.

Pour chaque paire (image, filtre), on obtient une carte d'activation, également appelée carte de caractéristiques, qui indique l'emplacement des caractéristiques dans l'image : plus la valeur est

élevée, plus la région correspondante de l'image ressemble à la caractéristique. Contrairement aux méthodes traditionnelles, les caractéristiques ne sont pas prédéfinies selon un formalisme particulier, mais sont apprises par le réseau pendant la phase d'entraînement. Les noyaux des filtres représentent les poids de la couche de convolution. Ils sont initialisés puis mis à jour à l'aide de la rétro propagation du gradient.

C'est là toute la puissance des réseaux neuronaux convolutifs : ils sont capables de déterminer automatiquement les éléments discriminants d'une image en s'adaptant au problème donné. Par exemple, si le problème consiste à distinguer les chats des chiens, les caractéristiques automatiquement apprises peuvent décrire la forme des oreilles ou des pattes.[13]

Lors de l'utilisation d'un CNN, il existe trois hyper paramètres importants que nous devons choisir :

Profondeur de la couche : il s'agit du nombre de noyaux de convolution utilisés, ce qui correspond également au nombre de neurones associés à un même champ récepteur. Une profondeur plus élevée permet au réseau d'apprendre des caractéristiques plus complexes.

Le pas (Stride) : il contrôle le chevauchement des champs récepteurs lors de la convolution. Un pas plus petit entraîne un chevauchement plus important et génère un volume de sortie plus grand.

La marge à 0 (Rembourage) ou zéro padding : parfois, il est utile d'ajouter des zéros autour du volume d'entrée. La taille de ce "zéro padding" est le troisième hyper paramètre. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En conservant la même surface que celle du volume d'entrée, elle peut aider à préserver les informations sur les bords de l'image.

En résumé, la couche de convolution est essentielle dans un CNN. Elle effectue la convolution des images avec des filtres pour extraire des caractéristiques importantes. Les hyper paramètres tels que la profondeur, le pas et la marge jouent un rôle crucial dans la manière dont la convolution est appliquée et dans la dimension du volume de sortie.[5]

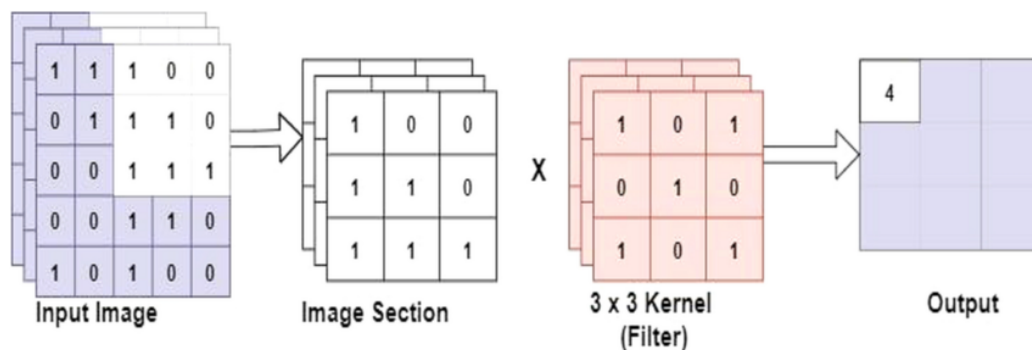


FIGURE 2.2 – Filtre de Convolution.

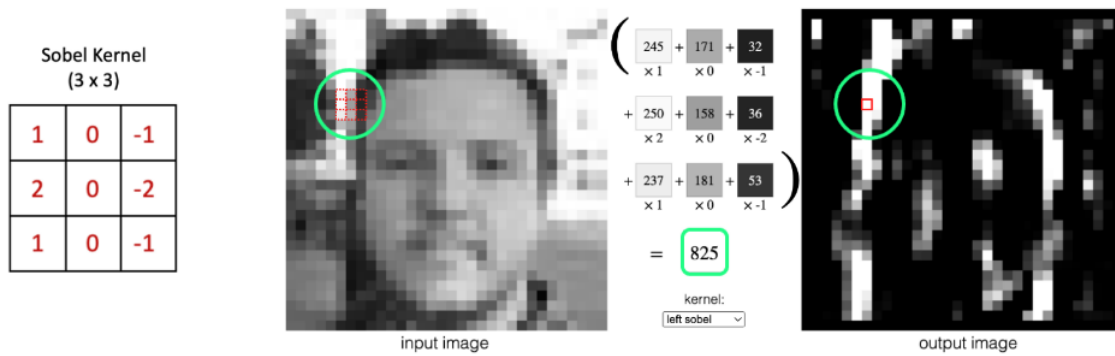


FIGURE 2.3 – Application du filtre sur le visage.

```
emotion_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48, 48, 1)))
```

2.2.4.2 Couche Pooling

Le regroupement de couches est une opération couramment utilisée après une couche convolutive dans un réseau de neurones convolutif (CNN). Cette opération permet de réduire la dimensionnalité des caractéristiques extraites et d'introduire une certaine invariance aux translations dans les données.

Il existe principalement deux types de regroupement couramment utilisés : le regroupement maximum (max pooling) et le regroupement moyen (average pooling).

Dans le regroupement maximum, une fenêtre (généralement de taille 2x2 ou 3x3) glisse sur la carte d'activation générée par la couche convolutive, et la valeur maximale dans chaque fenêtre est sélectionnée pour former une nouvelle carte d'activation réduite. Cela permet de conserver les caractéristiques les plus dominantes tout en réduisant la taille spatiale.

Dans le regroupement moyen, la même fenêtre glisse sur la carte d'activation, mais cette fois-ci, la valeur moyenne des activations dans chaque fenêtre est calculée pour former la nouvelle carte d'activation réduite. Cette opération permet de prendre en compte l'information globale de chaque région et de produire une représentation plus lisse.

Le regroupement de couches permet donc de réduire la dimension spatiale des caractéristiques tout en conservant les informations les plus importantes. Cela aide à réduire le nombre de paramètres et de calculs nécessaires dans le réseau, tout en introduisant une certaine invariance aux translations dans les données.[1]

Type	Max pooling	Average pooling
But	Chaque opération de pooling sélectionne la valeur maximale de la surface	Chaque opération de pooling sélectionne la valeur moyenne de la surface
Illustration		
Commentaires	<ul style="list-style-type: none"> • Garde les caractéristiques détectées • Plus communément utilisé 	<ul style="list-style-type: none"> • Sous-échantillonne la <i>feature map</i> • Utilisé dans LeNet

FIGURE 2.4 – Différence entre le Max pooling et l'Average pooling

Types de pooling :

- Le « max pooling »** : qui revient à prendre la valeur maximale de la sélection. C'est le type le plus utilisé car il est rapide à calculer (immédiat), et permet de simplifier efficacement l'image
- Le « meanpooling » (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 intermédiaire pour représenter ce lot de pixels[15]

```
# Une couche de pooling avec une taille de fenêtre 2x2.
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
```

FIGURE 2.5 – L'addition de la couche pooling au modèle

2.2.4.3 Fonction de correction Relu

Il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie.

La fonction ReLU $F(u) = \max(0, u)$ Cette fonction force les neurones à retourner des valeurs positives[10]. (Voir la figure 2.6)

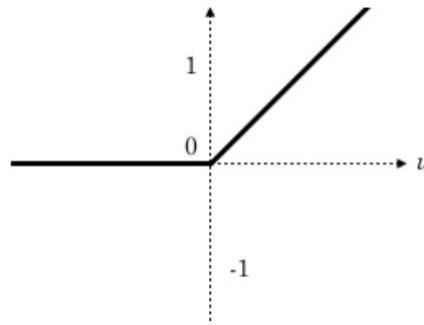


FIGURE 2.6 – Fonction ReLU

Dans notre modèle on a utilisé cette fonction plusieurs précise,ent dans les couches de convolution

2.2.4.4 Couche Flatten

Dans les réseaux de neurones convolutifs (CNN), la couche de "flatten" est utilisée pour convertir une matrice de caractéristiques en un vecteur unidimensionnel. Cela permet de passer des couches convolutives ou de pooling aux couches denses (fully connected). (**Voir la figure 2.7**)

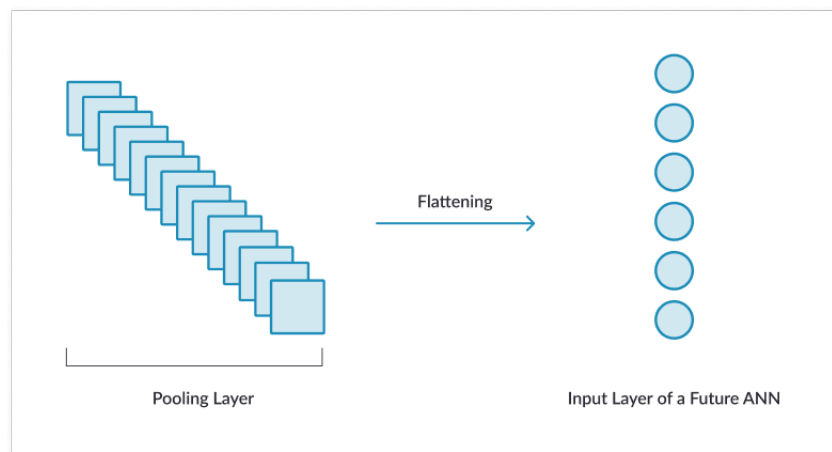


FIGURE 2.7 – La Couche Flatten

2.2.4.5 Couche entièrement connectée (FC)

Après plusieurs couches de convolution et de max -pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées.

Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente (comme on le voit régulièrement dans les réseaux réguliers de neurones). Leurs fonctions d'activations peuvent donc être calculées avec une multiplication matricielle suivie d'un décalage de polarisation[11]. (**Voir la figure 2.8**)

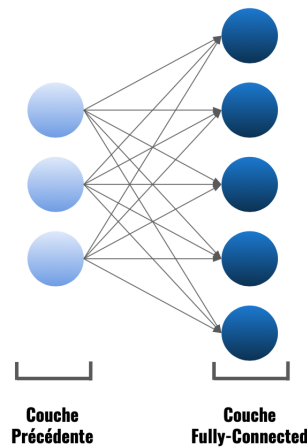


FIGURE 2.8 – Couches entièrement connectée (FC)

```
# Aplatit les données en un vecteur 1D.
emotion_model.add(Flatten())

# Une couche dense avec 1024 neurones et une activation ReLU.
emotion_model.add(Dense(1024, activation='relu'))

emotion_model.add(Dropout(0.5))

# La couche de sortie avec 7 neurones (pour les 7 émotions) et une activation
softmax pour produire des probabilités.
emotion_model.add(Dense(7, activation='softmax'))
```

FIGURE 2.9 – code d'empilement de la couche entièrement connecter

2.2.4.6 Dropout

Dans les réseaux de neurones convolutifs (CNN), le dropout est une technique de régularisation utilisée pour réduire le surapprentissage (overfitting). Elle consiste à désactiver aléatoirement un certain pourcentage de neurones pendant l'entraînement, ce qui force le réseau à ne pas trop dépendre de certaines connexions. Cela améliore la robustesse du modèle en l'empêchant de mémoriser les données d'entraînement et en favorisant une généralisation meilleure aux données de test. Le taux de dropout, souvent noté p , est un hyperparamètre qui définit la proportion de neurones à désactiver. (**Voir la figure 2.10**)

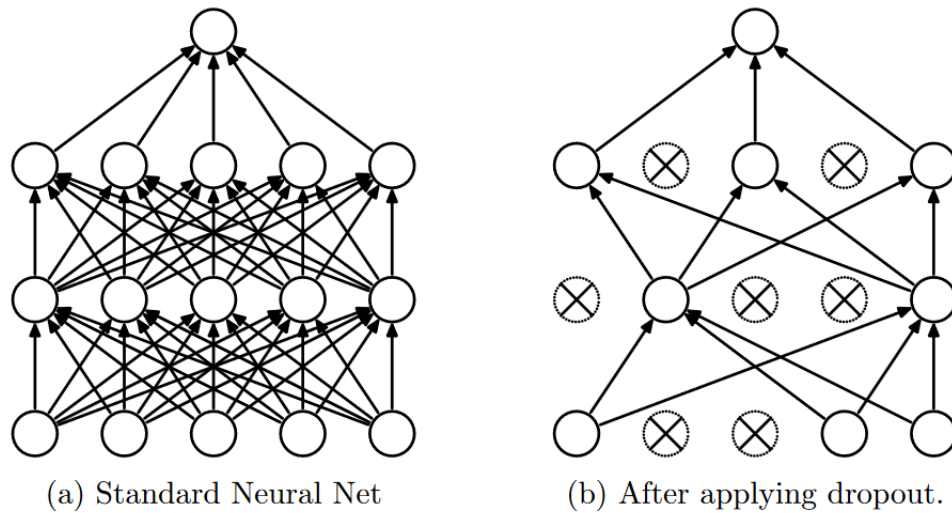


FIGURE 2.10 – Opération du Dropout

2.3 Classification

2.3.1 Introduction

La classification est le processus de classification des éléments en groupes spécifiques, et dans le contexte de l'apprentissage automatique, cette classification est effectuée par un ordinateur. Pensez à quel point ce serait génial si votre ordinateur pouvait faire la différence entre vous et un étranger, ou distinguer une pomme de terre d'une tomate, ou décider si une certaine performance méritait un A ou un F. Soudain, le concept devient beaucoup plus intéressant. Dans l'apprentissage automatique supervisé, la classification est une tâche essentielle qui implique l'utilisation de données étiquetées pour l'apprentissage. Dans les domaines de l'apprentissage automatique et des statistiques, la classification fait référence au problème consistant à attribuer une nouvelle observation à l'une de plusieurs classes ou souspopulations prédéfinies. Cette détermination est basée sur un ensemble de données de d'entraînement, où les observations sont connues pour appartenir à des catégories spécifiques.[2]

2.3.2 Définition

La classification est une discipline largement utilisée dans de nombreux domaines. Elle est souvent connue sous différents termes tels que classification, segmentation et regroupement. Pour donner une définition précise de la classification, il est nécessaire de comprendre ses racines, qui dérivent du verbe "classer", englobant plus qu'un seul domaine.

En mathématiques, la classification est la catégorisation des objets. Elle consiste à attribuer une classe à chaque objet ou individu à classer, sur la base de données d'apprentissage. Les méthodes d'apprentissage sont couramment utilisées pour accomplir cette tâche.[6]

2.3.3 Différent type de classification

Classification binaire : la classification binaire consiste à catégoriser les données en deux classes ou catégories distinctes. C'est la forme de classification la plus simple, où le but est d'af-

affecter chaque point de données à l'une des deux classes prédéfinies. Par exemple, déterminer si un e-mail est un spam ou non, classer une transaction comme frauduleuse ou légitime, ou prédire si un patient a ou non une certaine condition médicale.

Classification multi classe : la classification multi classe implique la catégorisation des données en plus de deux classes ou catégories. Dans ce type de classification, l'objectif est d'affecter chaque point de données à l'une de plusieurs classes prédéfinies. Par exemple, classer des images en différents types d'animaux (chat, chien, oiseau, etc.), reconnaître des chiffres manuscrits (0-9) ou identifier le genre d'une chanson (rock, pop, jazz, etc.). Les problèmes de classification multi classe peuvent être résolus à l'aide de divers algorithmes tels que les arbres de décision, la régression logistique, les machines à vecteurs de support ou les modèles d'apprentissage en profondeur comme les réseaux de neurones profonds. Ces deux types de classification fournissent une base pour organiser et analyser les données en fonction de leurs catégories ou classes distinctes.

2.3.4 Différentes approches de classification

Il existe différentes approches en matière de classification et d'apprentissage automatique. Voici quelques distinctions courantes :

Apprentissage supervisé : Dans l'apprentissage supervisé, un modèle est entraîné à partir d'un ensemble de données étiquetées, où chaque exemple est associé à une étiquette ou à un résultat correct. L'objectif est de prédire les étiquettes des nouvelles données non étiquetées. Cela est couramment utilisé pour la classification, la régression et d'autres tâches de prédiction.

Apprentissage non supervisé : Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé utilise un ensemble de données non étiquetées. L'objectif est de découvrir des structures, des patterns ou des relations inhérentes aux données. Les algorithmes de regroupement (clustering) et les techniques de réduction de dimensionnalité font partie de l'apprentissage non supervisé.

Apprentissage par renforcement : Dans l'apprentissage par renforcement, un agent interagit avec un environnement et apprend à prendre des actions afin de maximiser une récompense cumulative. L'agent explore l'environnement, reçoit des récompenses ou des pénalités en fonction de ses actions, et ajuste sa politique pour prendre des décisions optimales.

Apprentissage semi-supervisé : L'apprentissage semi-supervisé est une combinaison d'apprentissage supervisé et non supervisé. Il utilise à la fois des données étiquetées et non étiquetées pour entraîner le modèle. Cela peut être utile lorsque l'obtention de données étiquetées est coûteuse ou difficile, mais que des données non étiquetées sont disponibles en abondance.

2.3.5 Propagation directe

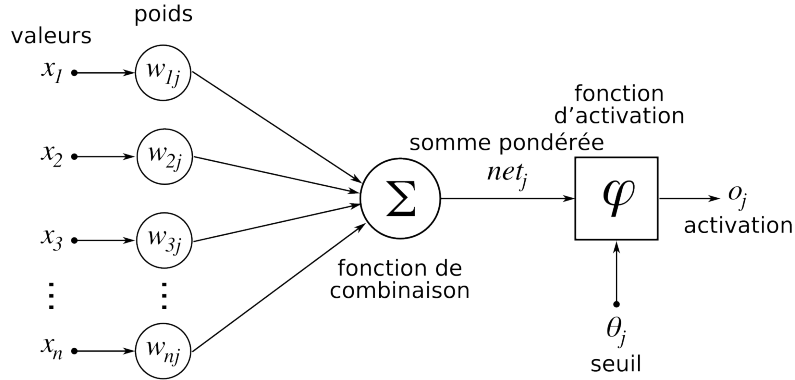


FIGURE 2.11 – Propagation directe

$$z = \sum_{i=1}^n w_i x_i + b \quad (2.1)$$

La propagation directe est le processus dans un réseau de neurones où les données d'entrée sont transmises à travers les couches du réseau pour générer une sortie. Et elle est dans laquelle on a initialisé les poids w et on introduit la fonction de perte.

z : préactivation.

w_i : Le poids.

x_i : L'entrée du neurone.

n : Nombre d'entrées.

Activation

$$a = \phi\left(\sum_{i=1}^n w_i x_i + b\right) \quad (2.2)$$

ϕ : Fonction d'activation.

2.3.6 Fonction Softmax

La fonction d'activation softmax, qui convertit la sortie du modèle en probabilités de classe.

$$P(y_i) = \frac{\exp(y_i)}{\sum_{j=1}^n \exp(y_j)} \quad (2.3)$$

y_i : Valeur de sortie.

y_j : Valeur de sortie de tout les classes.

2.3.7 Fonction coût (LOSS FUNCTION)

Dans notre modèle on a utilisé la fonction de perte "**CROSS ENTROPY**", car on a une classification et plusieurs classes (happy, neutral, disgusted, sad, angry, surprised).

$$L = - \sum_{i=1}^n P^* \log(P) \quad (2.4)$$

P^* : La probabilité de la classe.

P : La probabilité prédite.

n : Nombre de classes.

Nb : Le signe négatif est là comme $P(x) \leq 1$, donc $\log(P(x)) \leq 0$. Donc, pour avoir une valeur positive, le signe négatif est utilisé.

voici un code Python de la fonction de perte "CROSS ENTROPY" :

2.3.8 Rétro propagation (Backpropagation)

Après l'initialisation des poids et le calcul de la fonction de perte, On va essayer à l'aide d'un optimiseur (qui est ADAM dans notre modèle) de optimiser les poids et les biais d'un modèle basé sur l'erreur entre la sortie prédite et le réel sortie.

2.3.9 Optimisation

2.3.9.1 Descente du gradient (Gradient Descent)

La descente de gradient est une méthode pour minimiser une fonction objectif $L(w)$ paramétrée par les paramètres d'un modèle en mettant à jour les paramètres dans la direction opposée au gradient de la fonction objectif $\nabla_w L(w)$ par rapport aux paramètres. Le taux d'apprentissage η détermine la taille des pas que nous faisons pour atteindre un minimum (local). En d'autres termes, nous suivons la direction de la pente de la surface créée par la fonction objectif vers le bas jusqu'à atteindre une vallée.[17]

$$w_{i+1} = w_i - \eta \frac{\partial L}{\partial w_i} \quad ; \quad \eta > 0 \quad (2.5)$$

Jusqu'à : $|w_{i+1} - w_i| < \varepsilon$

w_i : Le poids actuel.

w_{i+1} : Nouveau poids.

η : taux d'apprentissage.

L : Fonction de perte.

$$b_{i+1} = b_i - \eta \frac{\partial L(b)}{\partial b_i} \quad ; \eta > 0 \quad (2.6)$$

b_i : Le biais actuel.

b_{i+1} : Nouveau biais.

2.3.9.2 Le taux d'apprentissage

Le taux d'apprentissage est la taille des étapes de la descente en gradient dans la direction du minimum local est déterminée par le taux d'apprentissage, qui détermine la vitesse ou la lenteur avec laquelle nous nous dirigerons vers les poids optimaux. (**Voir la figure 2.12**)

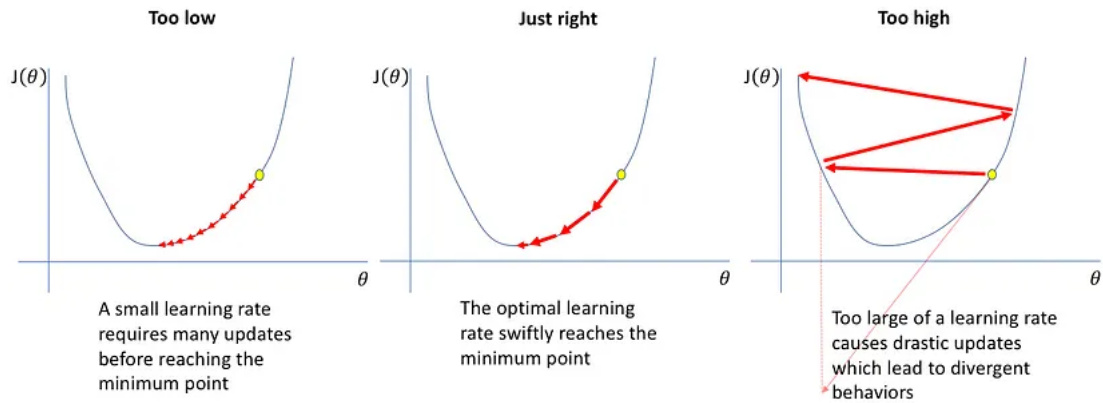


FIGURE 2.12 – Différentes tailles de taux d'apprentissage

2.3.9.3 Optimiseur momentum

L'idée principale derrière le momentum est de calculer une moyenne pondérée de façon exponentielle des gradients et de l'utiliser pour mettre à jour les poids. En prenant en compte les gradients passés, les étapes de descente de gradient deviennent lissées, ce qui peut réduire la quantité d'oscillations vues dans les itérations.[9]

$$w_{t+1} = w_t - \eta(Vw_{t+1}) \quad (2.7)$$

Avec :

$$Vw_{t+1} = \beta Vw_t + \left(\frac{\partial L}{\partial w_t}\right) \quad (2.8)$$

Vw_t : Variance (initiale=0).

β : le terme momentum $\beta = 0.9$.

2.3.9.4 Optimiseur Rmsprop

RMSprop divise également le taux d'apprentissage par une moyenne de gradients carrés en décroissance exponentielle. Hinton suggère β être réglé sur 0.9, alors qu'une bonne valeur par défaut pour le taux d'apprentissage η est 0.001.

$$w_{t+1} = w_t - \frac{\eta_{initial}}{\sqrt{Sdw_{t+1} + \varepsilon}} * \left(\frac{\partial L}{\partial w_t}\right) \quad (2.9)$$

$$b_{t+1} = b_t - \frac{\eta_{initial}}{\sqrt{Sdb_{t+1} + \varepsilon}} * \left(\frac{\partial L}{\partial b_t}\right) \quad (2.10)$$

Avec :

$$Sdw_{t+1} = \beta Sdw_t + (1 - \beta) \left(\frac{\partial L}{\partial w_t}\right)^2 \quad (2.11)$$

$$Sdb_{t+1} = \beta Sdb_t + (1 - \beta) \left(\frac{\partial L}{\partial b_t}\right)^2 \quad (2.12)$$

2.3.9.5 Adam

Adaptive Moment Estimation (Adam) est une méthode qui calcule les taux d'apprentissage adaptatif pour chaque paramètre. En plus de stocker une moyenne en décroissance exponentielle des carrés des gradients passés Sdw_t comme Adadelta et RMSprop, Adam conserve également une moyenne en décroissance exponentielle des gradients passés Vw_t , similaire à Momentum :

$$w_{t+1} = w_t - \frac{\eta_{initial}}{\sqrt{(Sdw_{t+1})^{corr} + \varepsilon}} * (Vw_{t+1})^{corr} \quad (2.13)$$

$$b_{t+1} = b_t - \frac{\eta_{initial}}{\sqrt{(Sdb_{t+1})^{corr} + \varepsilon}} * (Vb_{t+1})^{corr} \quad (2.14)$$

Avec :

$$Vw_{t+1} = \beta_1 Vw_t + (1 - \beta_1) \left(\frac{\partial L}{\partial w_t} \right) \quad (2.15)$$

$$(Vw_{t+1})^{corr} = \frac{Vw_{t+1}}{1 - \beta_1} \quad (2.16)$$

Vw_t : Rapidité (initiale=0).

$$Sdw_{t+1} = \beta_2 Sdw_t + (1 - \beta_2) \left(\frac{\partial L}{\partial w_t} \right)^2 \quad (2.17)$$

$$(Sdw_{t+1})^{corr} = \frac{Sdw_{t+1}}{1 - \beta_2} \quad (2.18)$$

2.4 Conclusion

En conclusion de ce chapitre, nous avons fourni une analyse détaillée des composants fondamentaux des réseaux de neurones convolutifs (CNN) et expliqué la procédure de classification des émotions faciales, en présentant l'architecture spécifique utilisée dans notre projet, basée sur le modèle CNN CLASSIQUE.

Tout d'abord, nous avons exploré les différentes couches qui constituent un CNN : les couches convolutives, les couches de pooling, et les couches entièrement connectées. Les couches convolutives jouent un rôle essentiel en identifiant et en extrayant les caractéristiques locales des images de visages à l'aide de filtres appliqués sur les pixels. Les couches de pooling, en réduisant la dimensionnalité des données tout en préservant les informations importantes, améliorent l'efficacité et réduisent le risque de surapprentissage. Les couches entièrement connectées, situées à la fin du réseau, permettent d'intégrer les caractéristiques extraites pour effectuer la classification finale.

Nous avons ensuite détaillé la procédure de classification, où les caractéristiques extraites par les couches convolutives et de pooling sont passées à travers les couches entièrement connectées pour prédire l'émotion présente dans l'image. Cette procédure repose sur l'utilisation de fonctions d'activation et de techniques de régularisation pour optimiser les performances du modèle.

L'architecture spécifique utilisée dans notre projet, est composée de plusieurs couches convolutives et de pooling, suivies de couches entièrement connectées. Cette structure permet au réseau de capturer efficacement les variations subtiles des expressions faciales et de généraliser ses connaissances pour prédire avec précision les émotions sur de nouvelles images.

En synthèse, ce chapitre a mis en lumière l'importance des différentes couches dans un CNN, expliqué la procédure de classification des émotions, et présenté l'architecture spécifique adoptée pour notre projet de détection des émotions. Cette compréhension approfondie des composants, de la procédure de classification, et de la structure du réseau pose les bases théoriques nécessaires pour aborder les aspects expérimentaux et les optimisations pratiques dans les chapitres suivants. En exploitant les capacités des CNN, notre projet vise à atteindre une reconnaissance précise et robuste des émotions, contribuant ainsi aux avancées dans le domaine de la vision par ordinateur et de l'intelligence artificielle.

Chapitre 3

Conception de l'application

3.1 Outils de conception



Le langage UML (Unified Modeling Language) est constitué de diagrammes intégrés utilisés par les développeurs informatiques pour la représentation visuelle des objets, des états et des processus dans un logiciel ou un système. Le langage de modélisation peut servir de modèle pour un projet et garantir une architecture d'information structurée ; il peut également aider les développeurs à présenter leur description d'un système d'une manière compréhensible pour les spécialistes externes. UML est principalement utilisé dans le développement de logiciels orientés objet. Les améliorations apportées à la norme dans la version 2.0 la rendent également adaptée à la représentation des processus de gestion



Visual Paradigm est un outil logiciel de modélisation et de gestion de projet qui prend en charge une variété de techniques et de langages de modélisation, tels que UML (Unified Modeling Language), BPMN (Business Process Model and Notation), et ERD (Entity-Relationship Diagram). Il est utilisé par les développeurs de logiciels, les analystes d'affaires et les gestionnaires de projet pour concevoir, documenter et gérer des systèmes complexes. Visual Paradigm offre des fonctionnalités pour la création de diagrammes, la génération de code, la gestion des exigences, et la collaboration en équipe, facilitant ainsi le développement et la maintenance des logiciels



Microsoft Project (ou MS Project ou MSP) est un logiciel de gestion de projets édité par Microsoft. Il permet aux chefs de projet et aux planificateurs de planifier et piloter les projets, de gérer les ressources et le budget, ainsi que d'analyser et communiquer les données des projets.

Utilisé aujourd'hui par plus de 20 millions de chefs de projet, Microsoft Project est le logiciel de gestion de projet le plus utilisé au monde^{1,2}. Plus de 10 000 entreprises ont aussi déployé la version serveur de Microsoft Project, nommée Microsoft Project Server.

3.2 Équipe de développement

M. Ait Khouya Youssef : l'encadrant du groupe, Professeur à la Faculté des Sciences et Techniques Errachidia .

M. Salhi Abdelmounaim : membre de l'équipe, étudiant à la Faculté des Sciences et Techniques Errachidia .

M. Sadiki Abdelkarim : membre de l'équipe, étudiante à la Faculté des Sciences et Techniques Errachidia.

3.3 Diagramme de gantt

Le diagramme de Gantt est largement utilisé dans la gestion de projet pour représenter visuellement l'avancement des différentes activités (tâches) qui composent un projet. Il offre une vue d'ensemble de la planification, des dépendances entre les tâches, des durées prévues et réelles, permettant ainsi de suivre et de visualiser l'état d'avancement du projet de manière efficace.

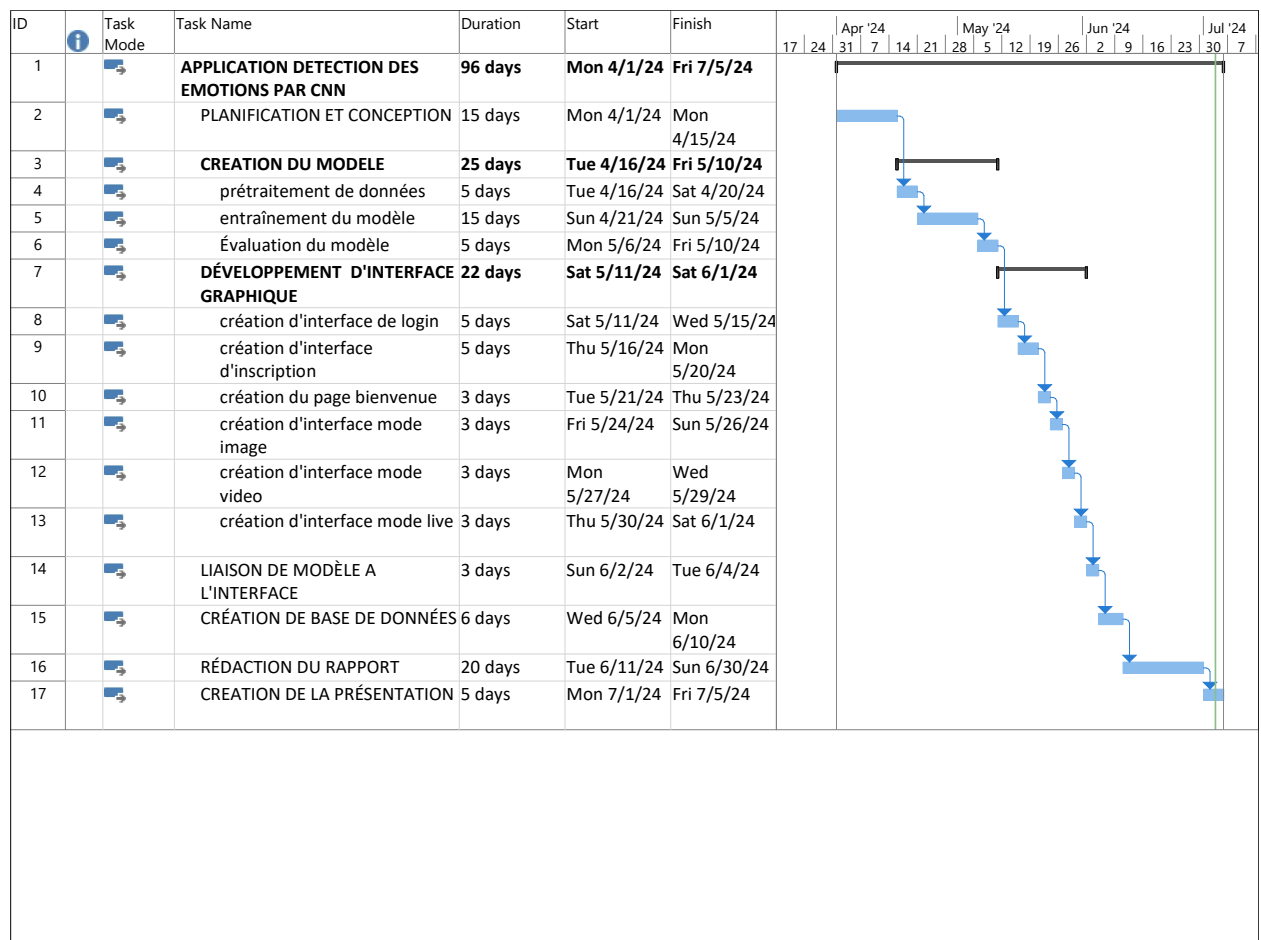


FIGURE 3.1 – Diagramme de Gantt

3.4 Diagrammes

3.4.1 Diagramme de classe

Un diagramme de classe est une représentation graphique utilisée en génie logiciel et en modélisation UML (Unified Modeling Language) pour décrire la structure statique d'un système. Il montre les classes du système, leurs attributs, méthodes et les relations qui les lient, comme les associations, les héritages et les dépendances. (Voir la figure 3.2)

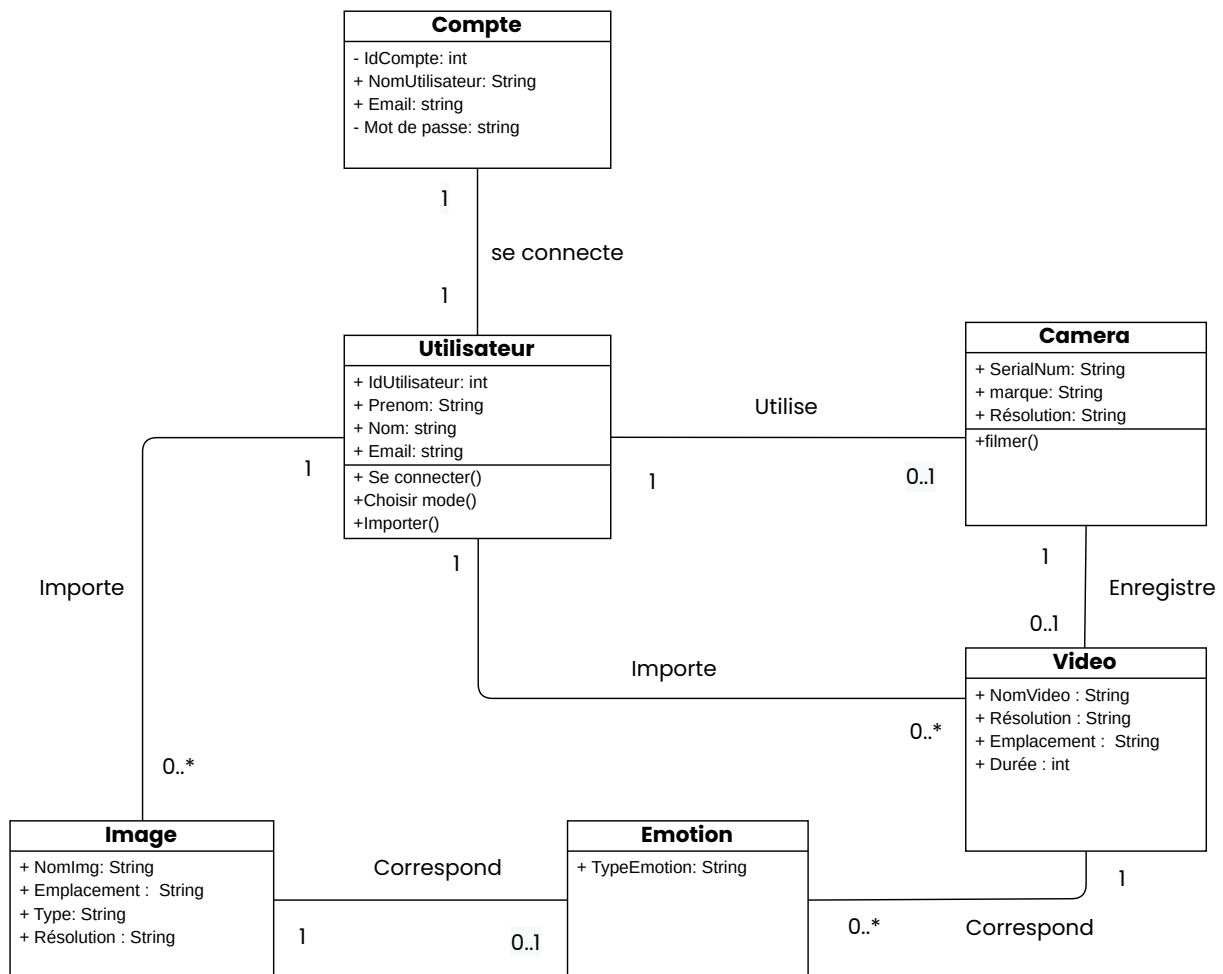


FIGURE 3.2 – Diagramme de classes.

3.4.2 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est une représentation graphique utilisée en ingénierie logicielle pour décrire les interactions entre les utilisateurs (ou acteurs) et un système. Il montre les différents cas d'utilisation (ou fonctionnalités) que le système offre aux utilisateurs et comment ces utilisateurs interagissent avec ces fonctionnalités. (Voir la figure 3.3)

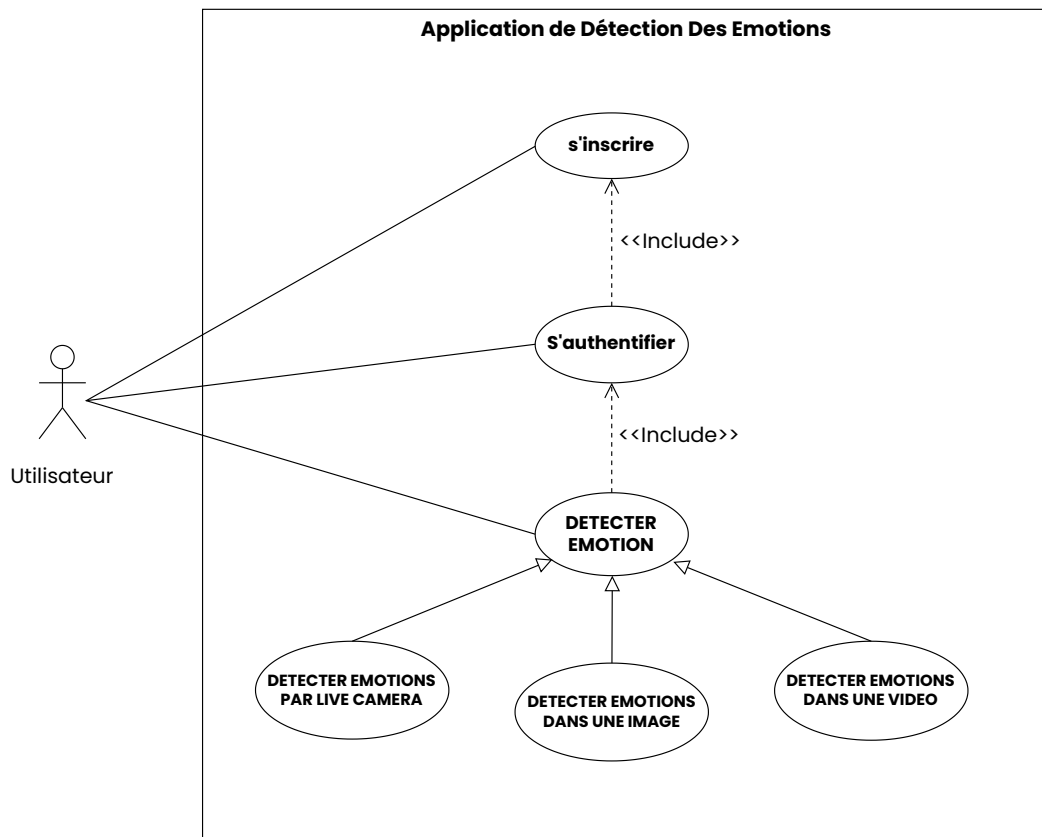


FIGURE 3.3 – Diagramme de cas d'utilisation.

3.4.3 Diagramme de séquence

Un diagramme de séquence est une représentation graphique utilisée en ingénierie logicielle pour illustrer la séquence d'interactions entre différents objets ou composants d'un système au fil du temps. Il montre les objets impliqués et les messages échangés entre eux dans un ordre chronologique, permettant de visualiser le flux de contrôle et de données dans un scénario spécifique. Ce type de diagramme aide à comprendre et à documenter la dynamique des opérations dans le système.

3.4.3.1 Diagramme de séquence de la fonctionnalité 's'inscrire'

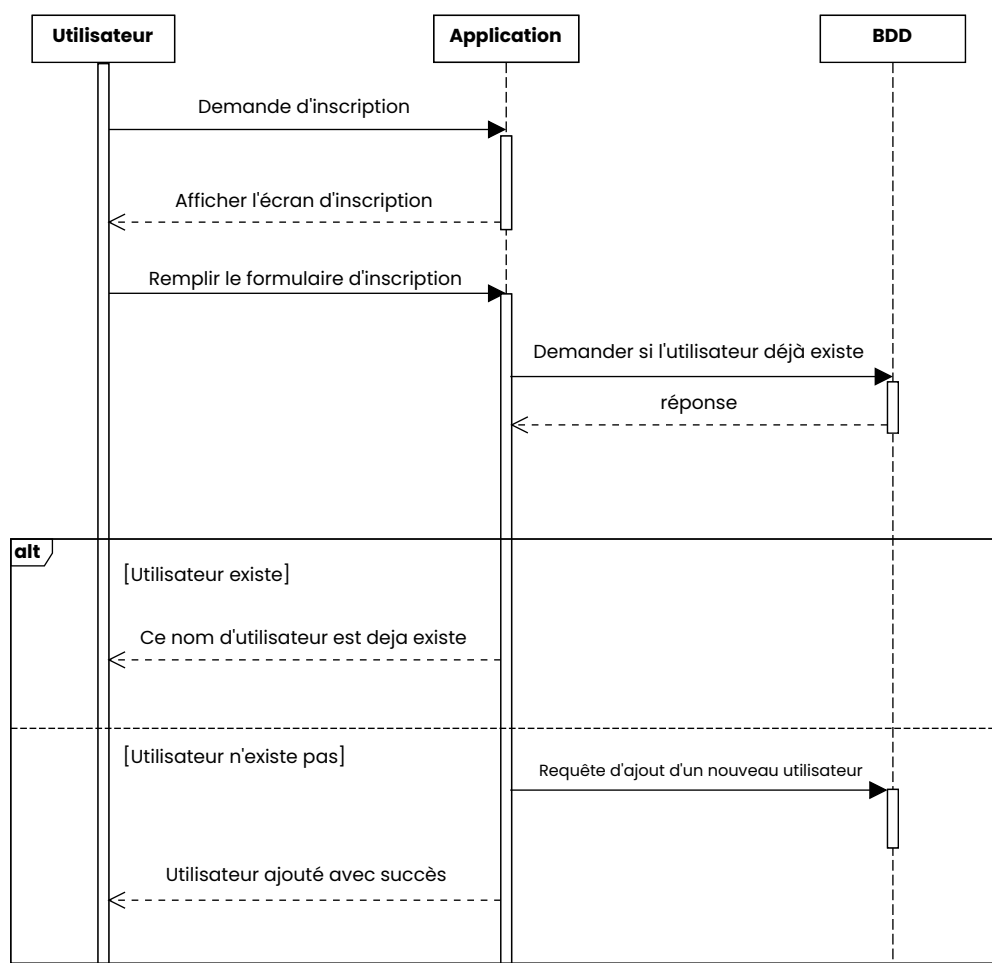


FIGURE 3.4 – Diagramme de séquence de la fonctionnalité 's'inscrire'.

3.4.3.2 Diagramme de séquence de la fonctionnalité 's'authentifier'

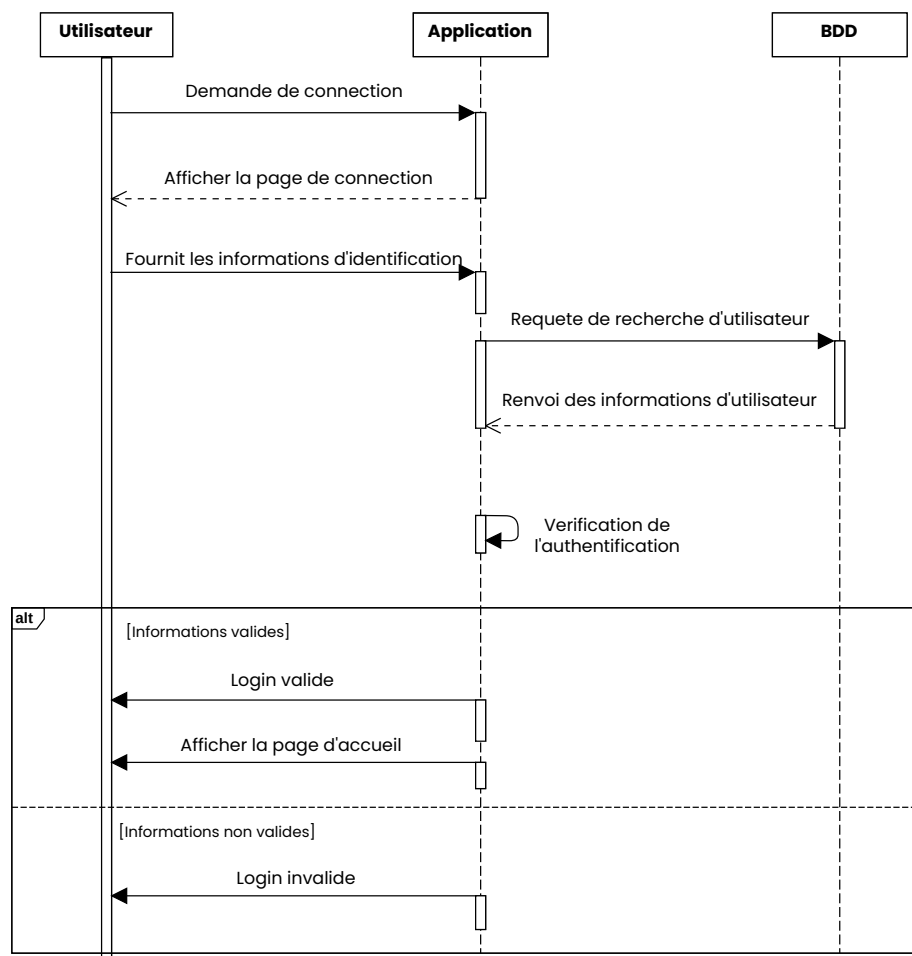


FIGURE 3.5 – Diagramme de séquence de la fonctionnalité 's'authentifier'.

3.4.3.3 Diagramme de séquence de la fonctionnalité 'Détecter émotion'

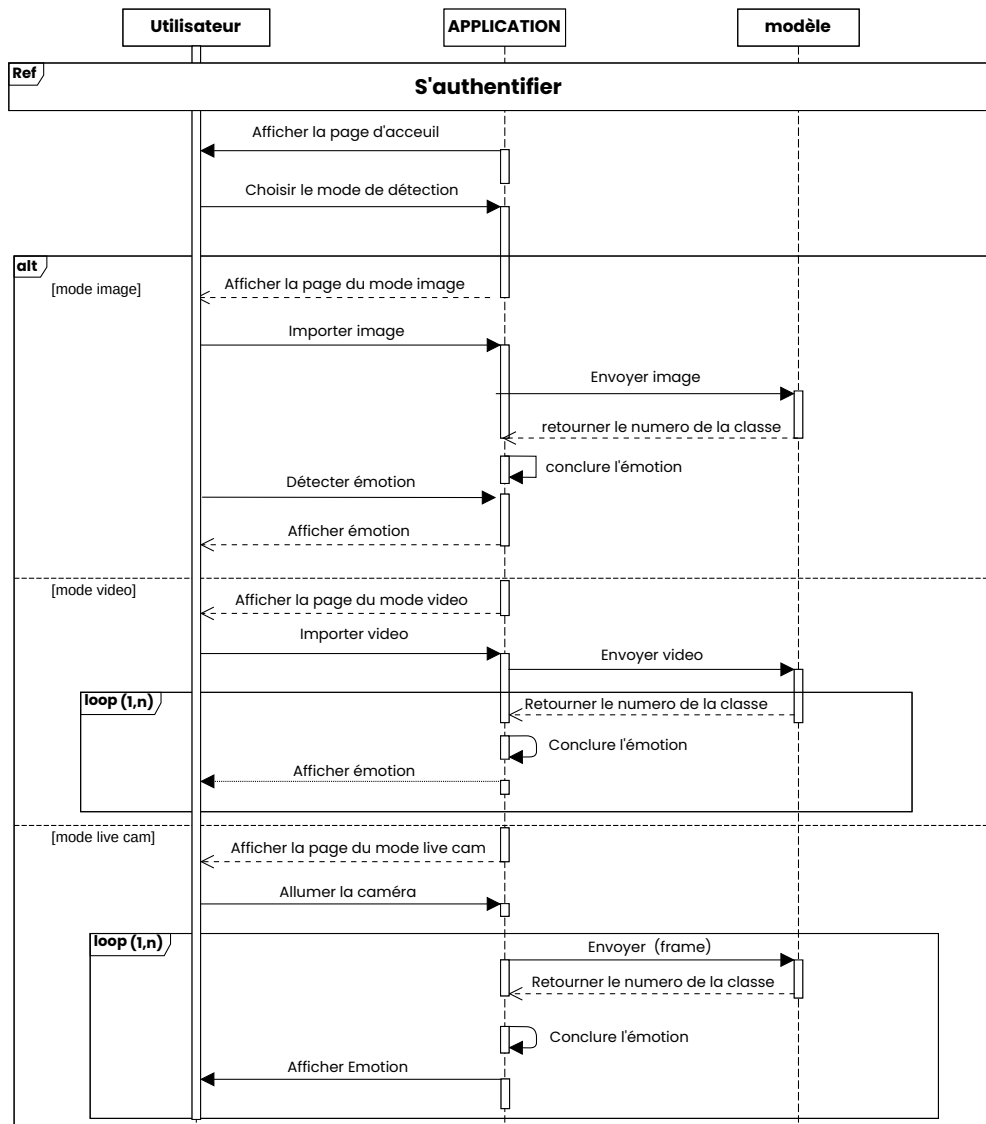


FIGURE 3.6 – Diagramme de séquence de la fonctionnalité 'Détecter émotion'.

3.5 Conclusion

En conclusion de ce chapitre de conception, nous avons détaillé les éléments essentiels pour la planification et la structuration de notre projet de détection des émotions. Nous avons présenté un diagramme de Gantt et divers diagrammes UML (Unified Modeling Language), qui ensemble fournissent une vue complète et organisée de notre projet.

Le diagramme de Gantt a été utilisé pour planifier et visualiser les différentes phases du projet, y compris les tâches spécifiques, les délais, et les dépendances entre les activités. Cette approche nous permet de gérer efficacement le temps et les ressources, de suivre l'avancement du projet, et de s'assurer que chaque étape est réalisée dans les délais prévus.

Les diagrammes UML, tels que les diagrammes de cas d'utilisation, de classes, de séquence, ont été élaborés pour modéliser les aspects structurels et comportementaux du système. Les

diagrammes de cas d'utilisation ont permis d'identifier et de définir les interactions entre les utilisateurs et le système, tandis que les diagrammes de classes ont détaillé la structure statique du système en montrant les classes, leurs attributs, et leurs relations. Les diagrammes de séquence ont offert une représentation dynamique des interactions entre les différents composants du système.

En synthèse, ce chapitre de conception a fourni une base solide pour le développement et l'implémentation de notre projet. Le diagramme de Gantt et les diagrammes UML ont permis de clarifier les exigences, de structurer les tâches, et de visualiser les interactions et les flux de données. Cette approche méthodique et organisée assure une compréhension commune parmi les membres de l'équipe et facilite la gestion du projet, posant ainsi les fondations pour une mise en œuvre réussie et une réalisation efficace de notre système de détection des émotions.

Chapitre 4

Implémentation et Résultats

4.1 Outils de développement

4.1.1 Langages de programmation et bibliothèques



Python : Est un langage de programmation très polyvalent et modulaire, qui est utilisé aussi bien pour écrire des applications comme YouTube, que pour traiter des données scientifiques.

Par conséquent, il existe de multiples installations possibles de Python. Le langage Python est placé sous une licence libre proche de la licence BSD4 et fonctionne sur la plupart des plates-formes informatiques, des smartphones aux ordinateurs centraux⁵, de Windows à Unix avec notamment GNU/Linux en passant par MacOS, ou encore

Android, iOS, et peut aussi être traduit en Java ou NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.



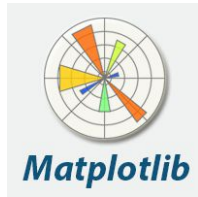
tensorflow : Est une bibliothèque open-source développée par Google, conçue principalement pour le calcul numérique et l'apprentissage automatique. Elle permet de créer, d'entraîner et de déployer des modèles d'apprentissage automatique, en particulier des réseaux de neurones profonds, en utilisant une approche basée sur les graphes de calcul. TensorFlow offre une grande flexibilité et une performance élevée, ce qui en fait l'une des bibliothèques les plus populaires pour le développement d'applications d'apprentissage automatique et d'in-

telligence artificielle.



Keras : est une bibliothèque open-source haut niveau pour le développement rapide d'applications d'apprentissage automatique et de réseaux de neurones artificiels. Elle offre une interface simple et intuitive, permettant aux utilisateurs de créer et de former des modèles d'apprentissage automatique avec une facilité accrue. Keras fonctionne comme une interface d'abstraction sur d'autres bibliothèques d'apprentissage automatique sous-jacentes, telles que TensorFlow, Theano ou Microsoft Cognitive Toolkit (CNTK), simplifiant ainsi le

processus de développement tout en offrant une grande flexibilité et une performance élevée.



Matplotlib : Est une bibliothèque open-source de visualisation de données en 2D intégrée à Python. Elle offre une vaste gamme de fonctionnalités pour la création de graphiques statiques, interactifs et animés, permettant aux utilisateurs de représenter graphiquement leurs données de manière claire et efficace. Matplotlib prend en charge divers types de graphiques, notamment les graphiques linéaires, les histogrammes, les diagrammes en barres, les nuages de points, les diagrammes à secteurs, les diagrammes en boîte, etc. Cette bibliothèque est largement utilisée dans les domaines de la science des données, de la recherche, de la finance et de nombreux autres domaines pour visualiser et explorer les données de manière professionnelle et esthétique.



opencv : est une bibliothèque open-source spécialisée dans le traitement d'images et la vision par ordinateur. Elle fournit un ensemble d'outils et de fonctions hautement optimisés pour la manipulation, l'analyse et la compréhension des images et des flux vidéo en temps réel. OpenCV prend en charge de nombreuses opérations de traitement d'images, telles que la détection d'objets, la reconnaissance faciale, le suivi de mouvement, la stéréovision, la calibration de caméra, la segmentation d'images, la correspondance d'images, etc. Cette bibliothèque est largement utilisée dans divers domaines tels que la robotique, la surveillance, la réalité augmentée, la vision industrielle, la recherche académique, et bien d'autres, en raison de sa grande efficacité, sa polyvalence et sa facilité d'utilisation.



MySQL : prononcé «My SQL» ou «My Sequel», est un système de gestion de base de données relationnelle open source. Il est basé sur le langage de requête de structure (SQL), utilisé pour ajouter, supprimer et modifier des informations dans la base de données. Les commandes SQL standard, telles que ADD, DROP, INSERT et UPDATE peuvent être utilisées avec MySQL. MySQL peut être utilisé pour diverses applications, mais se trouve généralement sur les serveurs Web. Un site Web utilisant MySQL peut inclure des pages Web donnant accès aux informations d'une base de données. Ces pages sont souvent qualifiées de "dynamiques", ce qui signifie que le contenu de chaque page est généré à partir d'une base de données lors du chargement de la page. Les sites Web utilisant des pages Web dynamiques sont souvent appelés sites Web gérés par une base de données.



CostumTkinter :est une bibliothèque UI moderne et personnalisable pour Tkinter de Python. Elle étend les capacités de Tkinter en fournissant de nouveaux widgets, des thèmes et des options de personnalisation, permettant aux développeurs de créer des interfaces utilisateur plus attrayantes et fonctionnelles. CustomTkinter offre une variété de widgets, y compris des boutons, des étiquettes, des entrées, des cadres et plus, avec des options de personnalisation améliorées. Et elle propose des thèmes intégrés et la possibilité de créer des thèmes personnalisés, ce qui facilite la conception d'interfaces utilisateur cohérentes

4.1.2 Environnements



Visual studio : Visual Studio est un outil de développement puissant qui permet d'effectuer l'ensemble du cycle de développement au même endroit. Il s'agit d'un environnement de développement intégré (IDE) complet permettant d'écrire, de modifier, de déboguer et de générer du code, puis de déployer votre application. En plus de l'édition et du débogage du code, Visual Studio comprend des compilateurs, des outils de complétion de code, un contrôle de code source, des extensions et de nombreuses autres fonctionnalités qui améliorent chaque étape du processus de développement logiciel.



Jupyter notebook : Est une application web open-source qui permet de créer et de partager des documents interactifs contenant du code, des visualisations et du texte explicatif. Il prend en charge plusieurs langages de programmation, mais est particulièrement populaire dans le domaine de la science des données et de l'apprentissage automatique, grâce à son intégration transparente avec des langages comme Python, R et Julia. Les utilisateurs peuvent exécuter du code pas à pas, afficher les résultats intermédiaires et créer des visualisations directement dans le même document. Jupyter Notebook favorise la collaboration et la reproductibilité en permettant aux utilisateurs de partager facilement leurs travaux, tout en facilitant l'exploration interactive des données et des modèles.

4.2 Matériel et méthodes



Google Colab : abréviation de Google Colaboratory, est une plateforme basée sur le cloud fournie par Google qui permet aux utilisateurs d'écrire et d'exécuter du code Python dans leur navigateur, sans aucune configuration requise.

performance techniques en google colab :

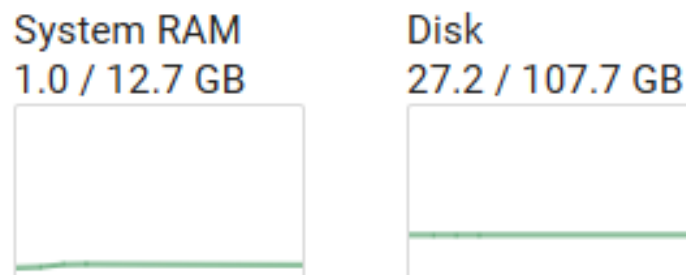


FIGURE 4.1 – Fiche technique GOOGLE COLAB

performance techniques PC 1 :

Le HP ProBook 6470b est un ordinateur portable conçu pour une utilisation professionnelle. Voici quelques spécifications clés :

processeur :

3e génération d'Intel Core i7-3520M (2,90 GHz, 4 Mo de cache L3, 35 W, 2 cœurs) jusqu'à 3,60 GHz avec la technologie Intel Turbo Boost

Graphiques :

Graphiques intégrés Intel HD 4000 (pour les processeurs Intel Core i7 et Core i5 de 3e génération)

Option d'affichage :

Écran interne de 14,0 pouces en diagonale, rétroéclairé par LED, HD 16 :9, antireflet (1366 x 768)

Stockage et disques :

Disque dur : 500 Go (SATA II)

Memoire :

4GB SDRAM DDR3 standard (1600 MHz) avec deux emplacements SODIMM prenant en charge la mémoire double canal Configurations de mémoire standard jusqu'à 8192 Mo (SODIMMs de 2 Go, 4 Go et 8 Go)

performance techniques PC 2 :

Le MSI Cyborg 15 A12V est un modèle d'ordinateur portable de la gamme Cyborg de MSI, principalement conçu pour les jeux et les performances élevées. Voici quelques caractéristiques typiques :

processeur :

Équipé d'un processeur Intel Core de 12e génération

Graphiques :

NVIDIA GeForce RTX 4060

Option d'affichage :

Dispose d'un écran de 15,6 pouces, généralement avec une résolution Full HD (1920 x 1080 pixels) et un taux de rafraîchissement élevé (144 Hz) pour une expérience fluide.

Stockage et disques :

Inclut un SSD NVMe de 512 Go, permettant des temps de chargement rapides et un accès rapide aux données.

Memoire :

Offre souvent 16 Go de RAM DDR4 pour assurer une multitâche fluide et des performances optimales

4.3 Le jeu de donnée(DataSet)

Le jeu de données FER-2013 (Facial Expression Recognition 2013) est une ressource largement utilisée dans le domaine de la reconnaissance des émotions faciales et du machine learning. Elle

a été publiée dans le cadre de la compétition Kaggle en 2013 et reste une référence pour les recherches et les développements dans ce domaine.

Les données consistent en des images de visages en niveaux de gris de 48 x 48 pixels. Les visages ont été automatiquement enregistrés de manière à ce que le visage soit plus ou moins centré et occupe à peu près le même espace dans chaque image.

La tâche consiste à classer chaque visage en fonction de l'émotion montrée dans l'expression faciale dans l'une des sept catégories (0 = Colère, 1 = Dégoût, 2 = Peur, 3 = Heureux, 4 = Triste, 5 = Surprise, 6 = Neutre). . L'ensemble de formation comprend 28 709 exemples et l'ensemble de test public comprend 7 178 exemples (**Voir la figure 4.2**).



FIGURE 4.2 – Exemple des images dans la BDD fer-2013

4.4 Base de données SQL

La base de données que Nous Avons utilisée pour les opérations de login et d'inscription dans notre projet de détection des émotions est de la forme suivante (**Voir la figure 4.3**) :

Table : users Cette table contient les colonnes suivantes :

id : Un identifiant unique pour chaque utilisateur de type entier auto-incrémenté.

username : Le nom d'utilisateur choisi par l'utilisateur de type chaîne de caractères.

password : Le mot de passe de l'utilisateur de type chaîne de caractères.

email : L'adresse e-mail de l'utilisateur qui est également une chaîne de caractères et elle doit être unique pour chaque utilisateur.

id	username	password	email
1	ABDELKARIM	12345	karimsadiki2077@gmail.com
2	SALHI	123	salhiabde@gmail.com

FIGURE 4.3 – Capture du base de données

4.5 Définition des couches

```

emotion_model = Sequential()

emotion_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu',
                        input_shape=(48, 48, 1)))
emotion_model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Flatten())
emotion_model.add(Dense(1024, activation='relu'))
emotion_model.add(Dropout(0.5))
emotion_model.add(Dense(7, activation='softmax'))

emotion_model.summary()
emotion_model.compile(loss='categorical_crossentropy', optimizer = 'adam', metrics = ["accuracy"])

```

FIGURE 4.4 – Structure du modèle adopté

Comprenons le code pour définir et compiler un modèle de réseau neuronal convolutif (CNN) pour une tâche spécifique,

emotion_Model = Sequential()

Cette ligne crée un modèle séquentiel en utilisant Keras. Un modèle séquentiel est une pile linéaire de calques, dans laquelle vous pouvez ajouter des calques un par un de manière séquentielle.

Couches convolutives :

```

emotion_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu',
                        input_shape=(48, 48, 1)))
emotion_model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))

```

Ces lignes ajoutent deux couches convolutives au modèle. Les couches convolutives sont fondamentales dans les CNN et sont utilisées pour détecter des modèles et des caractéristiques dans les images. Les paramètres de chaque couche Conv2D sont les suivants : 32 et 64 sont le nombre de filtres ou noyaux dans les couches. Ces filtres sont responsables de l'apprentissage de différentes fonctionnalités dans les images d'entrée.

kernel_size=(3, 3) définit la taille du noyau ou du filtre convolutif.

activation='relu' spécifie la fonction d'activation de l'unité linéaire rectifiée (ReLU), qui introduit la non-linéarité dans le modèle.

input_shape=(48, 48, 1) définit la forme d'entrée du premier calque sur 48×48 pixels avec un canal (images en niveaux de gris).

Couche MaxPooling :

```
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
```

Cette ligne ajoute une couche MaxPooling2D après la deuxième couche convolutive. La mise en pool maximale réduit les dimensions spatiales des cartes de fonctionnalités et permet de conserver les fonctionnalités importantes tout en réduisant la complexité de calcul.

Couche Dropout :

```
emotion_model.add(Dropout(0.25))
```

Ces lignes ajoutent des couches d'abandon avec un taux d'abandon de 25%. Le dropout est une technique de régularisation utilisée pour éviter le surapprentissage en désactivant aléatoirement une fraction de neurones pendant l'entraînement.

Couches convolutionnelles et MaxPooling supplémentaires :

```
emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
```

Ces lignes ajoutent deux paires supplémentaires de couches convolutives et de pooling maximum, suivies d'une autre couche d'abandon. Ces couches capturent probablement des fonctionnalités de niveau supérieur à partir des images d'entrée.

Couche d'aplatissement et couches denses :

```
emotion_model.add(Flatten())
emotion_model.add(Dense(1024, activation='relu'))
emotion_model.add(Dropout(0.5))
emotion_model.add(Dense(7, activation='softmax'))
```

Flatten() aplatit la sortie des couches précédentes en un vecteur 1D.

Les couches denses sont des couches entièrement connectées. Le premier dispose de 1 024 unités avec activation ReLU et le second de 7 unités (représentant probablement le nombre de classes d'émotions) avec une fonction d'activation softmax, qui convertit la sortie du modèle en probabilités de classe.[4]

4.6 Entraînement du modèle

La fonction "**fit_generator**" de Keras est utilisée pour entraîner un modèle sur des données générées par un générateur Python de manière batch par batch. Cette fonction était particulièrement utile pour gérer de grands ensembles de données qui ne pouvaient pas être entièrement chargés en mémoire à la fois (**Voir la figure 4.5**).

Paramètres Principaux :

train_generator : Le générateur de données qui produit les batches d'échantillons et de labels.

steps_per_epoch : Nombre de batches de données à générer par epoch.

epochs : Nombre d'epochs pour l'entraînement.

validation_data : Données de validation ou générateur pour la validation.

validation_steps : Nombre de batches de validation à générer par epoch de validation.

```
emotion_model_info = emotion_model.fit_generator(  
    train_generator,  
    steps_per_epoch=28709 // 64,  
    epochs=30,  
    validation_data=validation_generator,  
    validation_steps=7178 // 64)
```

FIGURE 4.5 – Entraînement du modèle

4.7 Résultats du modèle

pour obtenir les résultats du modèle on va utiliser la commande **emotion_model.evaluate** (Voir la figure 4.6)

```
113/113 [=====] - 14s 124ms\step - loss: 1.0744 -  
accuracy: 0.6610  
[1.0743528604507446, 0.6610337376594543]
```

FIGURE 4.6 – Résultats du modèle

le résultat final du modèle est **66%** de précision.

Pour analyser les performances de notre modèle d'apprentissage automatique sur le jeux de données FER2013, les graphiques de précision (accuracy) et de perte (loss) sont essentiels. Ils montrent comment le modèle s'améliore ou se détériore au fil des époques d'entraînement (Voir figure 4.7).

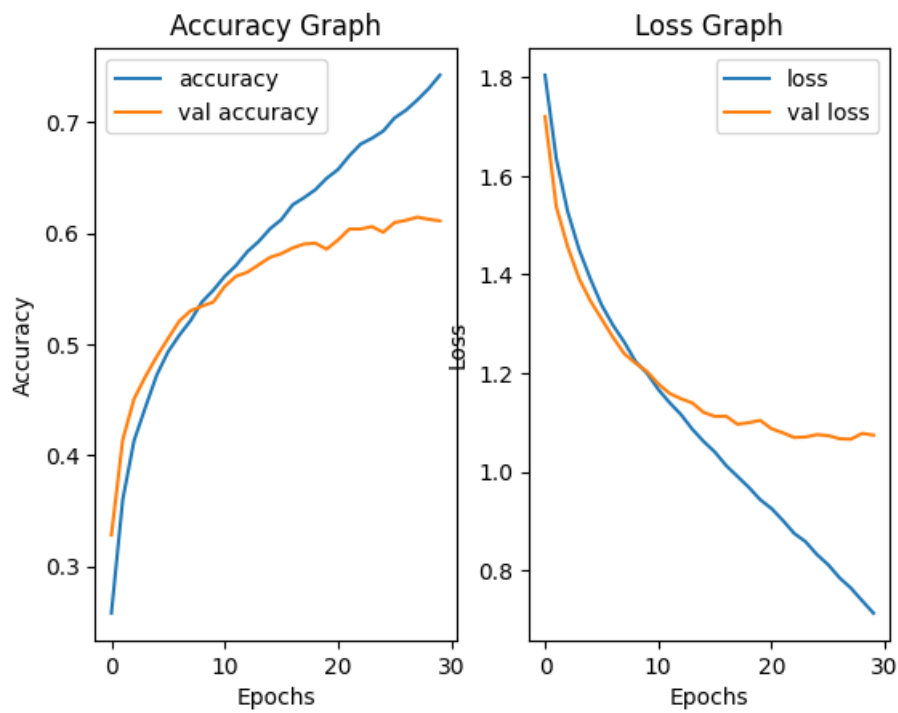


FIGURE 4.7 – Graphe de précision et perte

Les meilleurs résultats publiés sur le dataset FER2013 ont atteint une précision d'environ 72% à 74% sur le jeu de test, par l'utilisation des algorithmes complexes comme VGG16 mais cela peut varier légèrement en fonction des implémentations et des techniques spécifiques utilisées.[18]

TABLE 4.1 – Les meilleurs résultats publiés sur le dataset FER2013

Methode	Taux de précision
CNN	62.44%
GoogleNet	65.20%
VGG + SVM	63.31%
modèle adopté	66.1%
Conv + inception layer	66.40%
Bag of Words	67.40%
Attentional ConvNet	70.02%
CNN + SVM	71.20%
ARM (ResNet-18)	71.38%
Inception	71.60%
ResNet	72.40%
VGG	72.70%
VGG Avancé	73.28%

4.8 Matrice de confusion

La matrice de confusion est un outil utilisé pour évaluer la performance d'un modèle de classification. Elle permet de visualiser les prédictions du modèle en comparant les valeurs réelles et les valeurs prédites (**Voir la figure 4.8**).

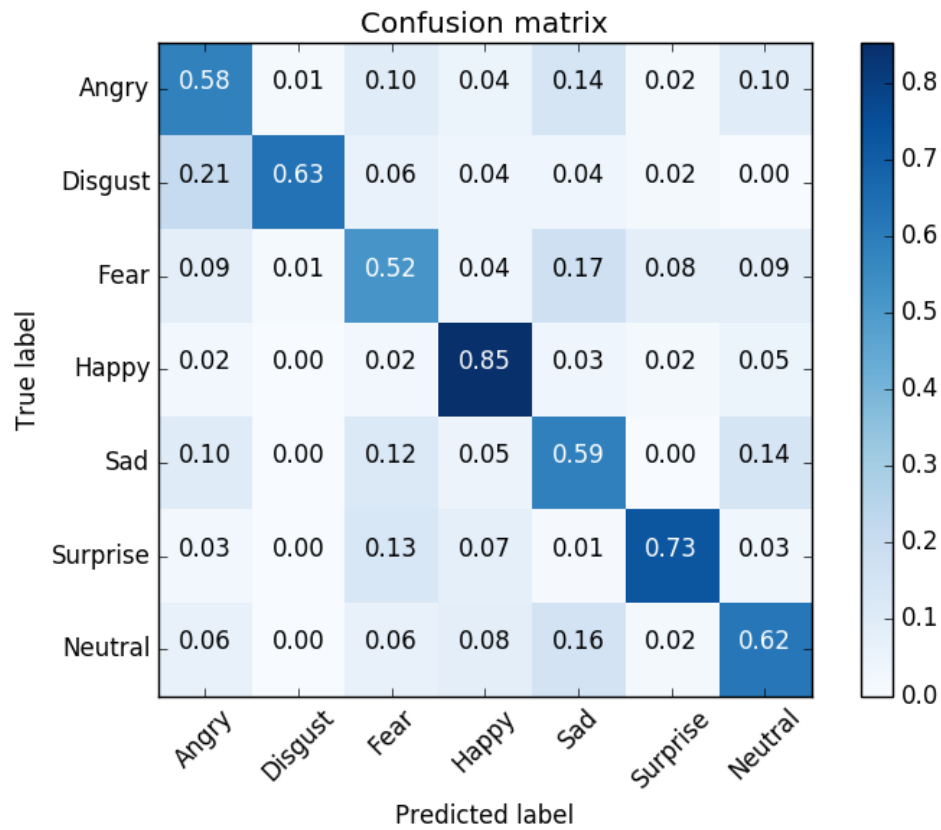


FIGURE 4.8 – Matrice de confusion

4.9 Présentation des maquettes

Page Inscription

Cette page permet à tout utilisateur de s'inscrire et de créer un compte sur l'application afin de pouvoir effectuer une détection d'émotion par la suite (**Voir la figure 4.9**).

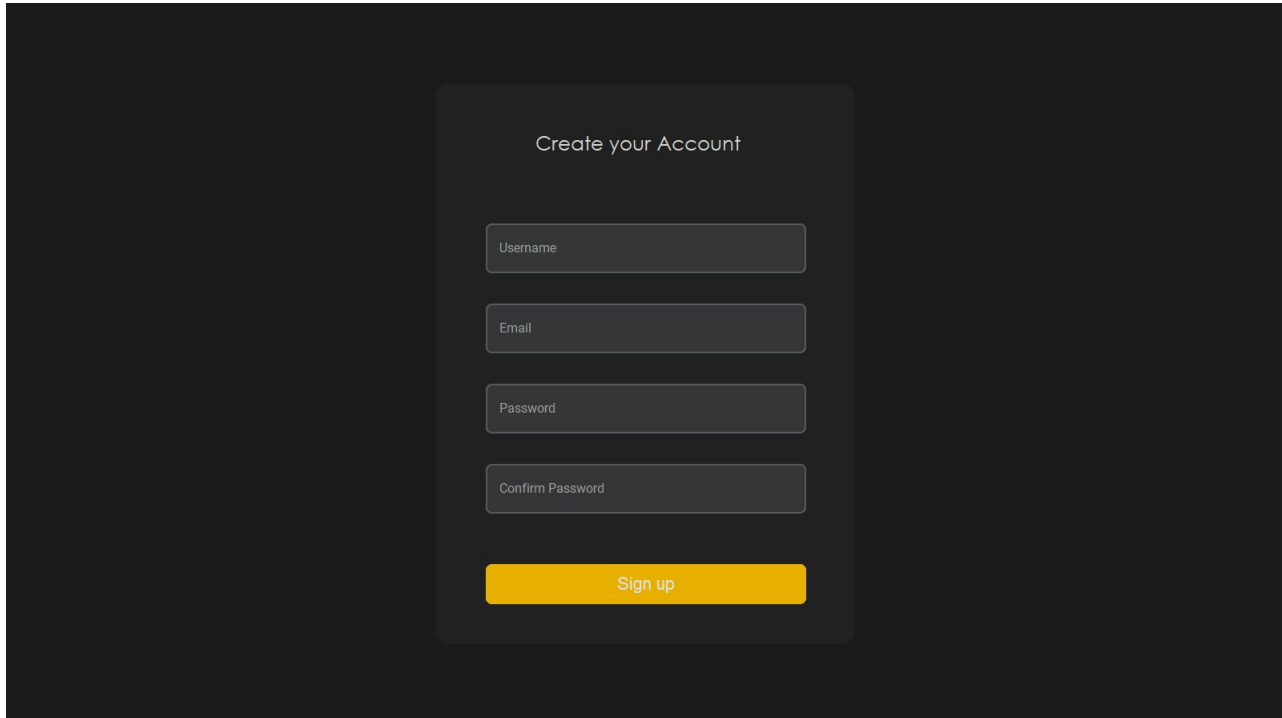
La maquette de la page d'inscription est présentée sur un fond noir. Au centre, un rectangle gris foncé contient le titre "Create your Account" en blanc. En dessous, quatre champs de saisie gris sont alignés verticalement, étiquetés "Username", "Email", "Password" et "Confirm Password". À la base de ce rectangle se trouve un bouton rectangulaire orange avec l'inscription "Sign up" en blanc.

FIGURE 4.9 – Page d'inscription

Page Authentication

Une fois l'utilisateur est inscrit, ce dernier à le droit de s'authentifier via l'interface ci-dessous (**Voir la figure 4.10**).

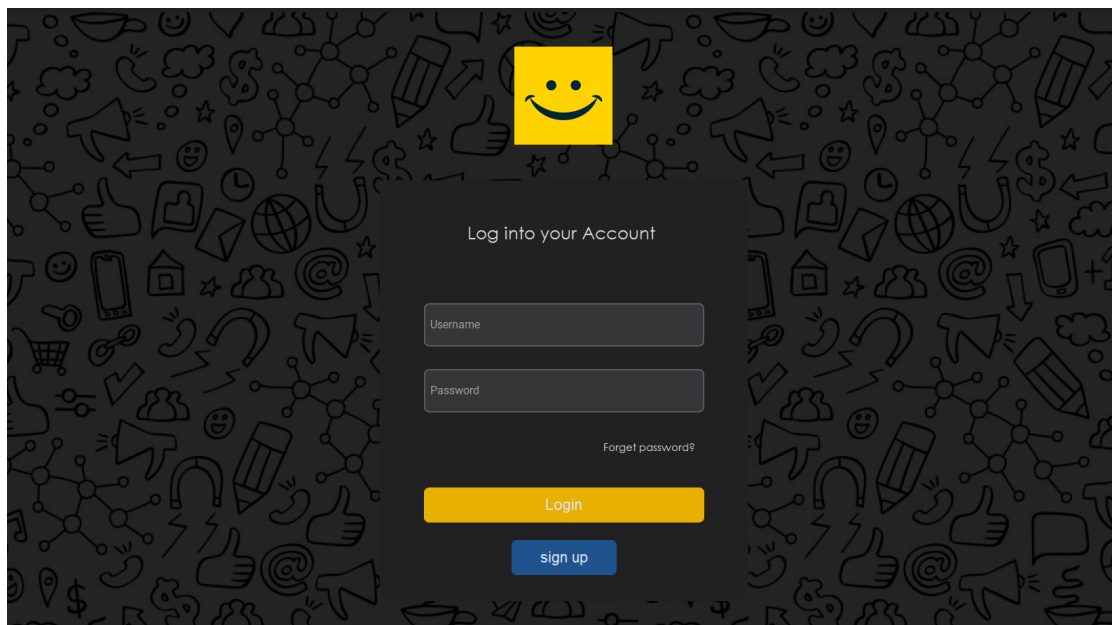
La maquette de la page de login est présentée sur un fond noir orné d'un motif répétitif de petits pictogrammes blancs (sourires, mains, globe, etc.). Au centre, un rectangle gris foncé contient le titre "Log into your Account" en blanc. En dessous, deux champs de saisie gris sont alignés verticalement, étiquetés "Username" et "Password". À la droite du champ "Password", un lien "Forget password?" est visible. En dessous des champs se trouvent deux boutons : un bouton rectangulaire orange "Login" et un bouton rectangulaire bleu "sign up".

FIGURE 4.10 – Page de login

Page d'accueil

Cette page permet à l'utilisateur de choisir le mode de détection (Voir la figure 4.11).

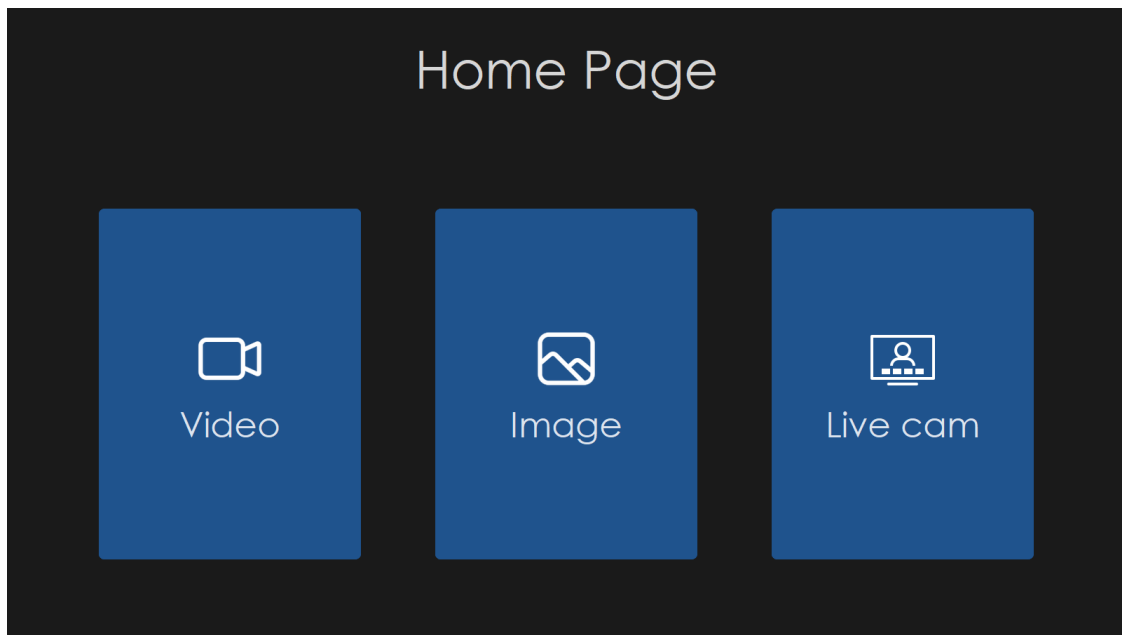


FIGURE 4.11 – Page d'accueil

Page image mode

Cette page permet à l'utilisateur de détecter l'émotion dans une image (Voir la figure 4.12).

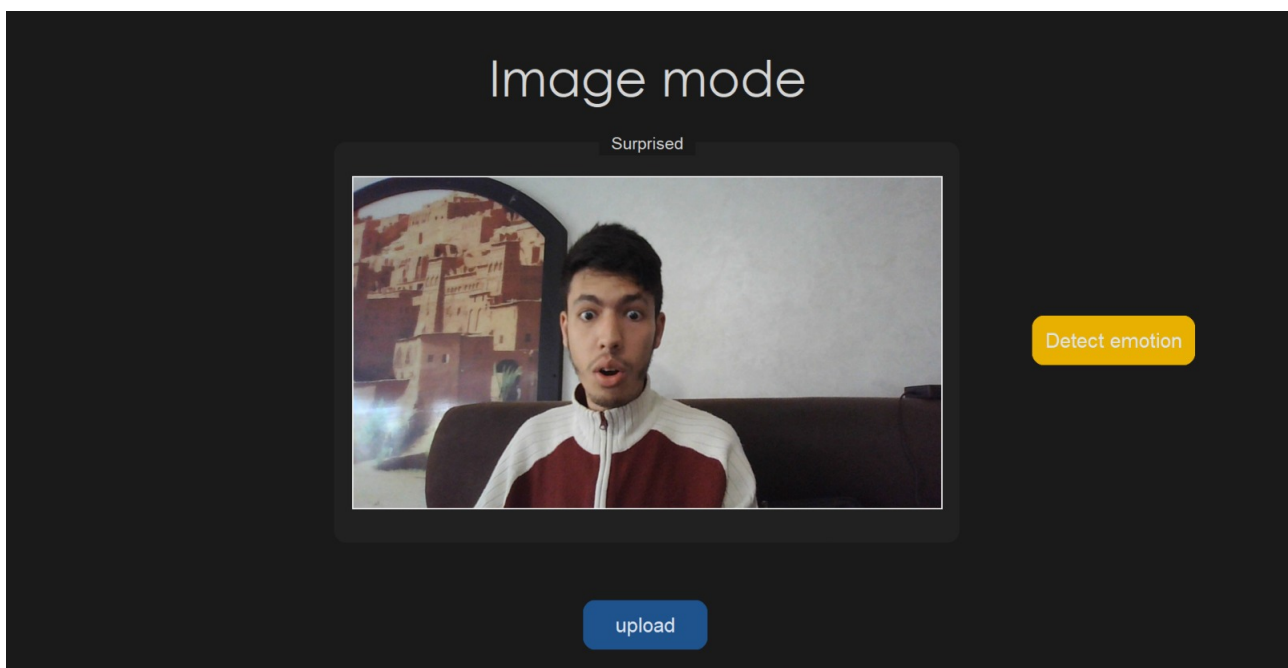


FIGURE 4.12 – Page 'MODE IMAGE'

Page live mode

Cette page permet à l'utilisateur de détecter l'émotion directement en utilisant la caméra.

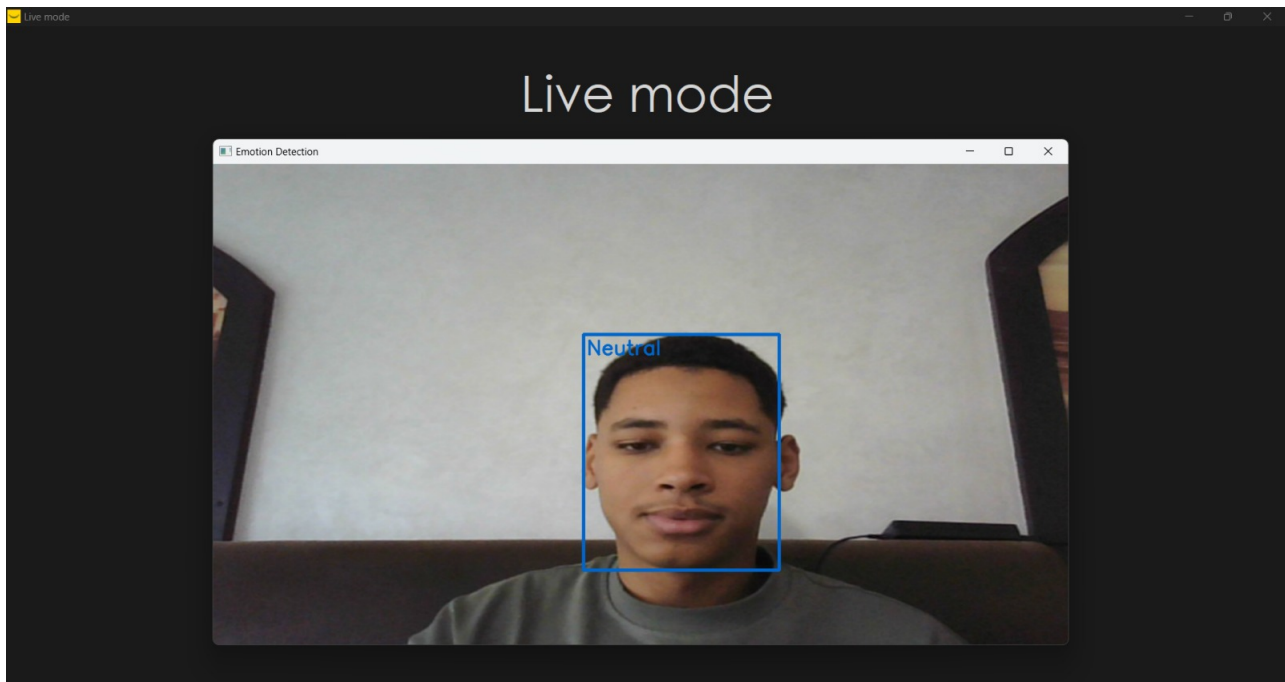


FIGURE 4.13 – Page 'MODE LIVE'

4.10 Conclusion

En conclusion de ce chapitre d'implémentation et résultats, nous avons présenté un aperçu exhaustif des étapes clés et des résultats obtenus dans notre projet de détection des émotions par réseaux de neurones convolutifs (CNN).

Nous avons d'abord décrit les outils de développement utilisés, qui incluent des bibliothèques et des frameworks de deep learning tels que TensorFlow et Keras, ainsi que des environnements de développement intégrés (IDE) comme Jupyter Notebook. Ces outils ont été essentiels pour le prototypage, l'entraînement et l'évaluation de notre modèle.

Ensuite, nous avons détaillé le dataset utilisé pour l'entraînement et la validation du modèle. Nous avons employé un dataset riche et diversifié contenant des images de visages annotées avec les émotions correspondantes (FER-2013), ce qui a permis à notre modèle d'apprendre à reconnaître une variété d'expressions faciales.

Nous avons également décrit en détail la structure de notre modèle CNN, basée sur l'architecture CNN CLASSIQUE. Cette architecture comprend plusieurs couches convolutives et de pooling suivies de couches entièrement connectées, optimisées pour la tâche de classification des émotions.

Les résultats de notre modèle ont été présentés, montrant les performances en termes de précision et de matrice de confusion. Ces résultats indiquent que notre modèle a atteint un niveau de précision satisfaisant, démontrant sa capacité à détecter et classifier les émotions de manière fiable.

Enfin, nous avons présenté les maquettes et les interfaces utilisateur développées pour visualiser les résultats de la détection des émotions. Ces maquettes illustrent comment notre système peut être intégré dans des applications pratiques, offrant une interface intuitive et conviviale pour les utilisateurs.

Conclusion générale et perspectives

La réalisation de cette application de détection des émotions par CNN a permis de démontrer l'efficacité des réseaux de neurones convolutifs dans l'analyse et l'interprétation des expressions faciales. En utilisant des bibliothèques comme TensorFlow, Keras et OpenCV, nous avons pu construire et entraîner un modèle capable de reconnaître différentes émotions humaines à partir d'images.

Nous avons utilisé la bibliothèque OpenCV et de l'algorithme Haar Cascade pour la détection des visages dans les images. Cette étape a permis de normaliser les données d'entrée et de se concentrer uniquement sur les régions d'intérêt.

Après nous avons utilisé TensorFlow et Keras pour construire un réseau de neurones convolutif (CNN) classique. Cette architecture a été choisie en raison de sa capacité à capturer les caractéristiques spatiales des images, essentielles pour la reconnaissance des émotions.

Le modèle a été entraîné sur un ensemble de données d'images étiquetées FER-2013, permettant au réseau d'apprendre à identifier des motifs spécifiques associés à chaque émotion. Le modèle a atteint une précision finale de 66%, ce qui, bien que perfectible, démontre une capacité significative à détecter les émotions.

Ensuite nous avons réalisé des diagrammes UML pour modéliser les aspects structurels et comportementaux de l'application. En plus d'utiliser un diagramme de Gantt pour planifier et suivre l'avancement du projet, assurant une gestion efficace du temps et des ressources.

Comme perspectives, nous souhaitons de :

Amélioration du Modèle : Explorer les architectures de réseaux neuronaux, telles que VGG-16 ou les modèles Transformer, pour voir si elles peuvent améliorer la précision et la robustesse de la détection des émotions

Augmentation des Données : En augmentant la base de données d'entraînement avec des données plus diversifiées et équilibrées, il serait possible d'améliorer la généralisation du modèle. L'utilisation de techniques comme l'augmentation de données pourrait également aider.

Déploiement et Scalabilité : Déployer l'application sur des plateformes cloud pour permettre une utilisation à grande échelle. Cela inclut l'utilisation de services comme AWS, Google Cloud ou Azure pour héberger l'application et gérer les charges de travail.

Développement d'une application Web : En utilisant la bibliothèque Flask on peut développer une application Web et la lier au modèle .

Bibliographie

- [1] Shervine Amidi AFSHINE AMIDI. “Pense-bête de réseaux de neurones convolutionnels”. In : (1 jan 2010).
- [2] GEEKSFORGEEKS. “Getting started with Classification”. In : (24 jan 2024).
- [3] Semri Khawla Rebahi ghdiri IMANE. “Rapport sur apprentissage profond appliquea la reconnaissance emotionnelle faciale”. In : (2021).
- [4] Shibsian KARSAW. “Emotion Detection Using Convolutional Neural Networks (CNNs)”. In : (12 jan 2024).
- [5] D. R LILIA. “La Détection de la colère chez le conducteur en utilisant le Deep Learning”. In : (2002).
- [6] Mr BELHADJER Hakim ET Mr SAROUER Brahim MADAME FELLAG. “Classification des images avec les réseaux de neurones Convolutionnels”. In : (2018).
- [7] Manav MANDAL. “INTRODUCTION TO CNN”. In : (1May 2021).
- [8] Francis CHARETTE MIGNEAULT. “Conception de Système de reconnaissance de visage spatio-temporelle sur vidéos à partir d’une seule image de référence”. In : (15 DEC 2017).
- [9] Kevin P. MURPHY’S. “Machine Learning : Une perspective probabiliste.” In : *MIT Press* (2012).
- [10] MALKI NARIMENE. “Classification automatique des textes par Les réseaux de neurones à convolution”. In : (2018/2019).
- [11] Abdelaziz HABBA et OMAR ISHAK ENCADRÉ PAR MR. OUAHAB ABDELWHAB. “La classification des images satellitaires par l’apprentissage profonde (deeplearning)”. In : (2018/2019).
- [12] Harnane Zahra OUNISSI MOHAMMED. “Modélisation et classification avec Deep Learning Application à la détection du Coronavirus Covid-19”. In : (2020).
- [13] Kimia Nadjahi PASCAL MONASSE. “Classez et segmentez des données visuelles”. In : (26 jun 2024).
- [14] Mohammed PAWAN. “VGG-16 | CNN model”. In : (21 mar 2024).
- [15] Lambert R. “Focus : Le Réseau de Neurones Convolutifs”. In : (11 jan 2019).
- [16] Jérémy ROBERT. “Deep Learning ou Apprentissage Profond : qu’est-ce que c’est ?” In : (28 Sep 2020).
- [17] Sebastian RUDER. “An overview of gradient descent optimization algorithms”. In : (2017). Insight Centre for Data Analytics, NUI Galway ; Aylien Ltd., Dublin ;
- [18] Zhuofa Chen YOUSIF KHAIREDDIN. “Facial Emotion Recognition : State of the Art Performance on FER2013”. In : (8mai 2021).
- [19] Mokri Mohammed ZAKARIA. “Classification des images avec les réseaux de neurones Convolutionnels”. In : (2017).