


République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Akli Mohand Oulhadj de Bouira

Faculté des Sciences et des Sciences Appliquées

Département d'Informatique

Mémoire de Master

en Informatique

Spécialité : *GSI*

Thème

Une approche IA pour la reconnaissance des expressions faciales

---

Encadré par

— Aliouet Wahiba

Réalisé par

— Azzoune Ikram

— Kheldoun Nadia

2019/2020

Remerciements

Nous remercions d'abord le bon Dieu qui nous a aidé et qui nous a donné le courage et la patience pour réaliser ce modeste travail.

Nous continuerons à remercier nos enseignants de l'Université Akli Mohand Oulhadj Bouira.

En particulier, nous remercions notre chere encadreur Mme Aliouat Wahiba qui était toujours à notre disposition sans réservation pendant la préparation de ce mémoire de fin d'études.

Nos remerciements vont aussi aux membres de jury pour avoir accepter de juger ce modeste travail. On n'oublie pas nos parents pour leur contribution, leur soutien et leur patience. Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours soutenue et encouragées au cours de la réalisation de ce travail.

Merci à tous et à toutes.

Dédicaces

J'ai le plaisir de dédier ce travail à mes très chers parents à qui est l'expression de mes profonds sentiments de respect, de gratitude et de reconnaissance que dieu les garde pour nous.

A toute la famille.

A tous mes amis.

A tous ceux qui m'aiment et que j'aime.

Kheldoun Nadia.

Dédicaces

Je dédie ce mémoire à :

Mes parents :

Ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutient permanent venu de toi.

Mes grands parents, Puisse dieu vous accorder santé, longue vie et prospérité.

Mes sœurs (Nesrine, yasmine, souha) et mon neuveu Amir qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.

Mon binôme Nadia, Mes amis et à tous ceux qui, par un mot, m'ont donné la force de continuer .....

Azzoune Ikram.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

FACS	Facial Action Coding System
CNN	Convolutional Neural Networks
RNN	les réseaux de neurones récurrents
ECNN	EmotionsCNN

Introduction générale

Le visage étant la partie la plus expressive et communicative d'un être humain, parce qu'il peut lui montrer différentes expressions émotionnelles exprimant l'émotion intérieure de la personne. Les expressions faciales provoquent des changements physiologiques sur le visage, tels que le mouvement des sourcils, la position de la bouche ouverts ou fermés, ou encore la manière de regarder fixement les yeux. Selon le travail [3], l'auteur montre qu'il existe six expressions émotionnelles universelles : dégoût, colère, bonheur, tristesse, surprise et peur. Ces expressions sont identifiables en observant les signes du visage.

Aujourd'hui, la reconnaissance d'expression faciale s'avère être l'une des applications les plus pertinentes dans de nombreux domaines à savoir : Intéraction homme-machine, médecine, sécurité, education,... etc.

Ces dernières années, le deep Learning et plus particulièrement les réseaux de neurones convolutionnels (CNN) ont apparu spécialement pour résoudre les problèmes rencontrés du machine Learning. Le CNN est l'une des structures réseau les plus représentatives de la technologie d'apprentissage en profondeur et a connu un grand succès dans le domaine du traitement et de la reconnaissance d'images.

L'objectif de ce projet de fin d'étude est de concevoir et de réaliser une application permet de reconnaitre les expressions faciales d'un visage en utilisant la méthode d'apprentissage profond. Dans ce travail, nous comparons entre deux architectures CNN différentes inspirés du VGGnet [34] et ResNet50 [35] qui sont élaboré pour la classification des images.

Ce mémoire est constitué en trois chapitres, comme suit :

Chapitre 1 présente un bref aperçu des principales théories émotionnelles, avec un accent particulier sur les expressions faciales.ensuit, il décrit les étapes de traitement d'un système d'analyse d'expression faciale en discutant les diverses techniques utilisées à chaque étapes.

Chapitre 2 présente le deep learning et ces défferents architecteur, on basant sur les CNN, puis La conception de notre approche de reconnaissance des expressions faciales nomée ECNN.

Le dernier chapitre est dédié à la description des différents outils utilisés dans le développement de notre application ,ensuit, nous présenterons la base de données utilisée. Puis on va créer deux modèles avec des architectures différentes, par la suite, on appliquera ces modèles sur la base fer2013. À la fin de ce chapitre, nous présentons les résultats obtenus et discutés.

Nous terminerons par une conclusion générale dans laquelle nous résumerons le travail réalisé, et nous donnons quelques suggestions pour améliorer et développer ce système.

Chapitre 1

La reconnaissance d'expression faciale

1.1 Introduction

Au cours des dernières années, les capacités informatiques et l'intelligence artificielle dans le domaine de la reconnaissance des expressions faciales ont évolué rapidement.

L'expression faciale est l'une des plus puissantes moyens immédiats pour les êtres humains de communiquer leurs émotions et intentions et pour vraiment réaliser une interaction intelligente efficace homme-ordinateur, il est nécessaire que l'ordinateur doit être capable d'interagir naturellement avec l'utilisateur, de la même manière que l'interaction homme-homme a lieu.

Ce chapitre présente un bref aperçu des principales théories émotionnelles, avec un accent particulier sur les expressions faciales (Section 1.2). Ensuite, il décrit les étapes de traitement au cours desquelles un système d'analyse d'expression faciale peut être décomposé tout en discutant les diverses techniques utilisées à chaque étapes (Section 1.3).

1.2 Expressions faciales et émotions

1.2.1 L'expression faciale

L'expression faciale est produite par l'activation des muscles faciaux, qui sont déclenchés par les impulsions nerveuses. Les actions des muscles faciaux provoquent le mouvement

et les déformations de peau et traits du visage. Dans l'interprétation de l'expression faciale, ce sont ces déformations que nous observons et dont nous devons déduire l'émotion sous-jacente [1].

1.2.2 L'émotion

Les émotions, de façon générale, sont des états motivationnels. Elles sont constituées d'impulsions, de désirs ou d'aversions ou, plus généralement, elles comportent des changements de motivation. Elles poussent l'individu à modifier sa relation avec un objet, un état du monde, un état de soi, ou à maintenir une relation existante malgré des obstacles ou des interférences. Notons une caractéristique essentielle de ces motivations : les émotions sont relationnelles. Elles se jouent entre le sujet et le monde. Les émotions ne sont pas des états subjectifs, intérieurs à une personne, ou du moins pas en première instance [2].

1.2.3 Qu'est-ce qu'une expression faciale émotionnelle ?

Les êtres humains peuvent exprimer leurs émotions suivant différentes modalités telles que les expressions faciales, les expressions corporelles, la prosodie ou encore le langage. Ces différents modes d'expressions des émotions ne s'opposent pas et peuvent très bien être utilisés simultanément. Nous pouvons par exemple exprimer de la peur avec une voix effrayée, un visage effrayé et une expression corporelle de peur. L'incongruité entre ces modalités peut entraîner une perturbation de la reconnaissance de l'expression faciale, qui se trouve alors biaisée par l'expression émotionnelle corporelle (Meeren et coll., 2005) [3].

1.2.4 Représentation de l'Expression Faciale

La variété des émotions exprimées à l'aide du visage complexifie leur étude, en se s'inspirant des études de darwin (1872), ekman et ses collaborateurs ont mis en évidence le caractère universel d'un nombre d'émotions, considérées comme « émotions de base » (la joie, la peur, la colère, le dégoût, la tristesse, la surprise) [3]. figure 1.1

Les représentations faciales relient les mouvements des muscles faciaux aux émotions exprimées. Elles représentent une sorte de dictionnaire utile à la reconnaissance des émotions [3].

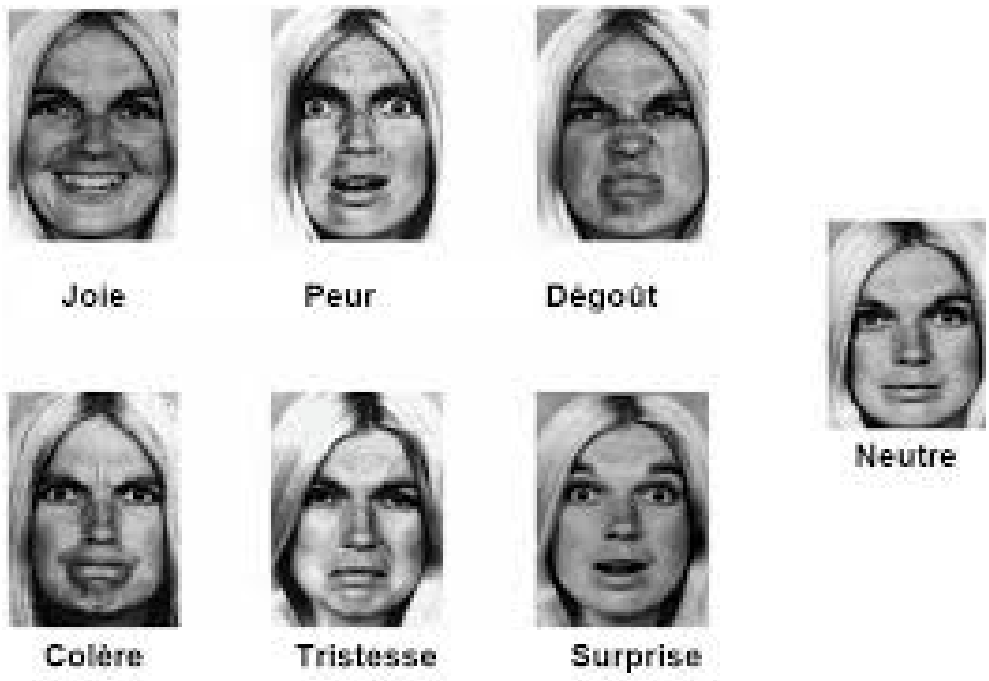


FIGURE 1.1 – Les six expressions faciales universelles et le neutre [4].

A.Représentation par le système FACS

Se basant sur l'universalité, Ekman et Izard ont développé des méthodes de mesure des comportements du visage. En particulier, ils ont créé le système FACS (Facial Action Coding System), largement utilisé et reconnu. Il utilise une quarantaine de caractéristiques anatomiques indépendantes, et définit une taxonomie de toutes les expressions faciales. Le Système FACS est probablement le standard le plus populaire utilisé pour classifier systématiquement les expressions physique des émotions du visage, et est utilisé par des psychologues mais aussi par des infographistes. Ce système définit 46 Unités d'action, qui sont autant de contraction ou de relaxation d'un ou plusieurs muscles, et dont l'association définit une expression faciale. La philosophie de base du système consistait à former des experts pour la reconnaissance et l'interprétation des Unités d'action, mais désormais, le système est aussi utilisé pour automatiser la reconnaissance des Action Units et donc des expressions, ainsi que pour la simulation graphique de visages [5].

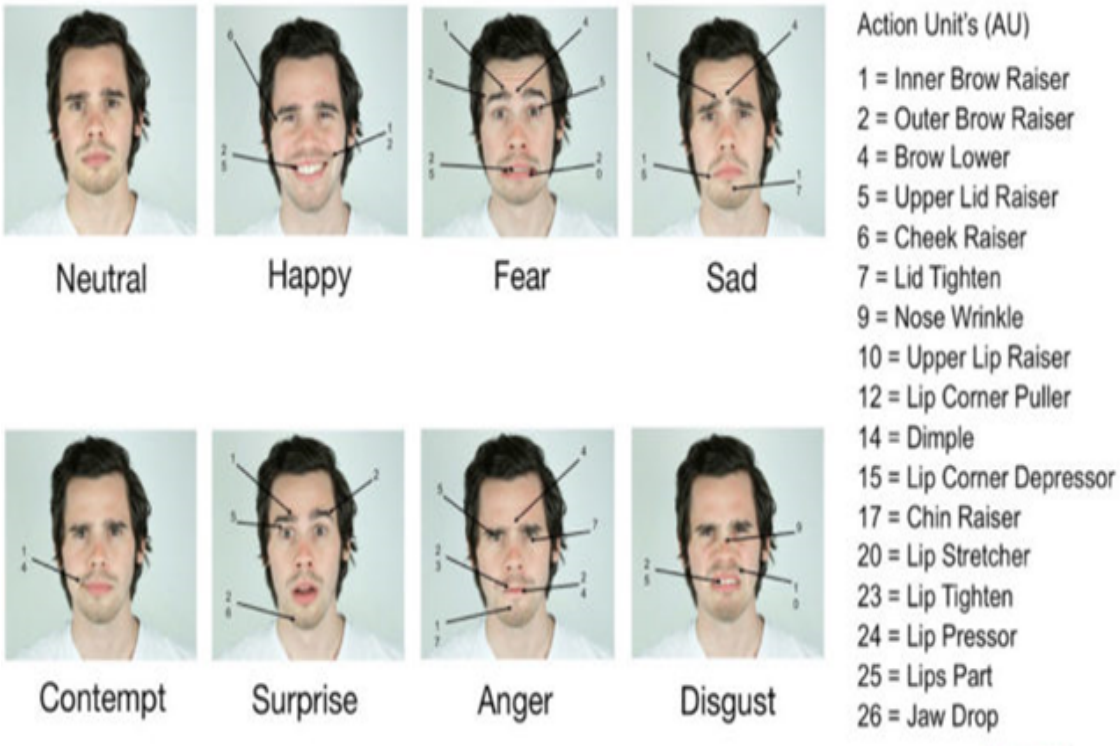


FIGURE 1.2 – Liste des Action Units relatives aux 6 expressions faciales + neutre [6].

Les références (unités d'action) à droite de l'image (figure 1.1)montrent les mouvements musculaires individuels de chaque expression. Elles indiquent les muscles spécifiques impliqués et les traits distinctifs visibles à la surface du visage. Elles sont dérivées du FACS (Facial Action Coding System), qui est méthode utilisée pour la recherche sur les expressions faciales développée par le Dr Paul Ekman et Wallace Friesen [6]. Bien que FACS soit un système de description bénéficiant d'une grande maturité (environ vingt années de développement), il souffre cependant de quelques inconvénients [7] :

- Complexité : on estime qu'il faut 100 heures d'apprentissage pour en maîtriser les principaux concepts.
- Difficulté de manipulation par une machine : FACS a d'abord été créé pour des psychologues, Certaines mesures restent floues et difficilement évaluables par une machine.
- Manque de précision : les transitions entre deux états d'un muscle sont représentées de manière linéaire, ce qui est une approximation de la réalité. En particulier les mesures temporelles de l'activation des muscles faciaux (onset, apex et offset) ne sont pas mises en évidence.

B. MPEG4

La norme de codage vidéo MPEG-4 dispose d'un modèle du visage humain développé par le groupe d'intérêt Face and Body AdHoc Group. C'est un modèle 3D articulé. Ce modèle est construit sur un ensemble d'attributs faciaux, appelés Facial Feature Points(FFP). Des mesures sur ces 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) [7].figure 1.3

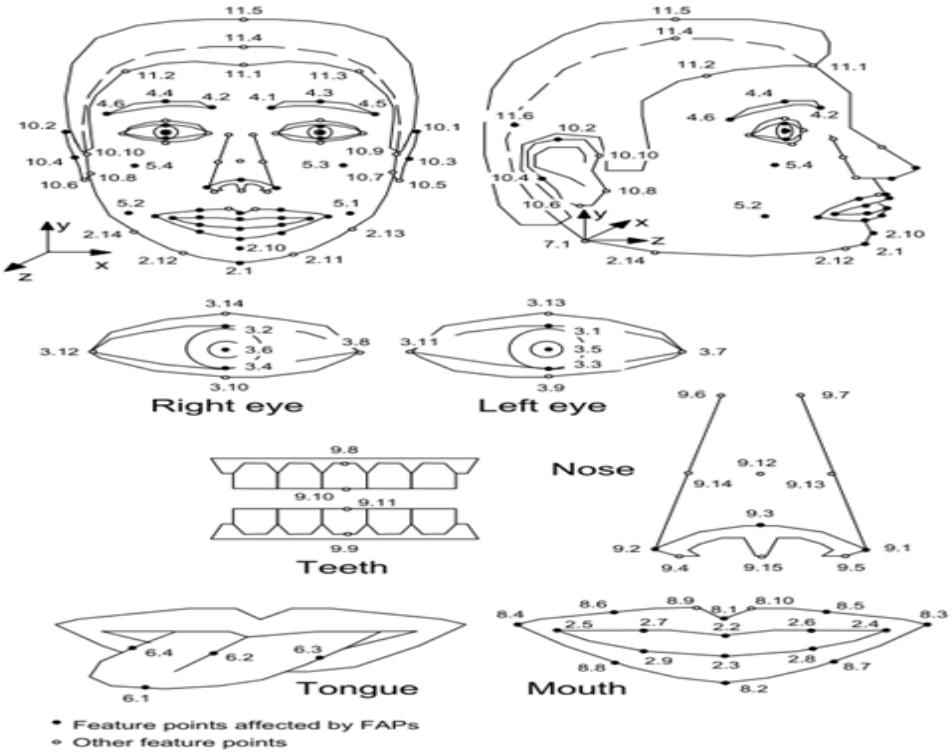


FIGURE 1.3 – MPEG-4 Points caractéristiques du visage [8].

C. Candide

Candide est un modèle du visage. Il est composé d'un modèle en fil de fer représentant un visage générique et d'un ensemble de paramètres [7] :

C.1. Paramètres de forme (Shape Units)

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

1. Hauteur de la tête.

2. Position verticale des sourcils.
3. Position verticale des yeux.
4. Largeur des yeux.
5. Hauteur des yeux.
6. Distance de séparation des yeux.
7. Profondeur des joues.
8. Profondeur du nez.
9. Position verticale du nez.
10. Degré de courbure du nez (s'il pointe vers le haut ou non).
11. Position verticale de la bouche.
12. Largeur de la bouche. [7]

C.2.Paramètres d'animation (Animation Units)

Ces paramètres représentent les différences intra-individus, i.e. les différentes actions faciales. Ils sont composés d'un sous-ensemble de FACS et d'un sous-ensemble des FAPs de MPEG-4. Les FAPs sont définis par rapport à leur FAPU correspondant. Ces paramètres, qu'ils soient d'animation ou de forme, sont représentés sous forme d'une liste de points du modèle de fil de fer à mettre à jour. Candide permet de voir clairement la différence entre les AUs de FACS et les FAPs de MPEG-4 : les Aus de FACS sont exprimées de manière absolue, à la différence des FAPs qui sont exprimés par rapport à des mesures du visage (les FAPUs) [7].
Quelque soit la représentation utilisée dans un processus d'analyse des expressions faciale, le but est la reconnaissance de l'expression faciale étudiée.

FIGURE 1.4 – CANDIDE-3 with 113 vertices and 168 surfaces [9].

1.3 Système de reconnaissance d'expression faciale

Un système de reconnaissance faciale doit pouvoir identifier des visages présents dans une image ou une vidéo de manière automatique.il est généralement composer de 3 étapes principales .figure 1.5

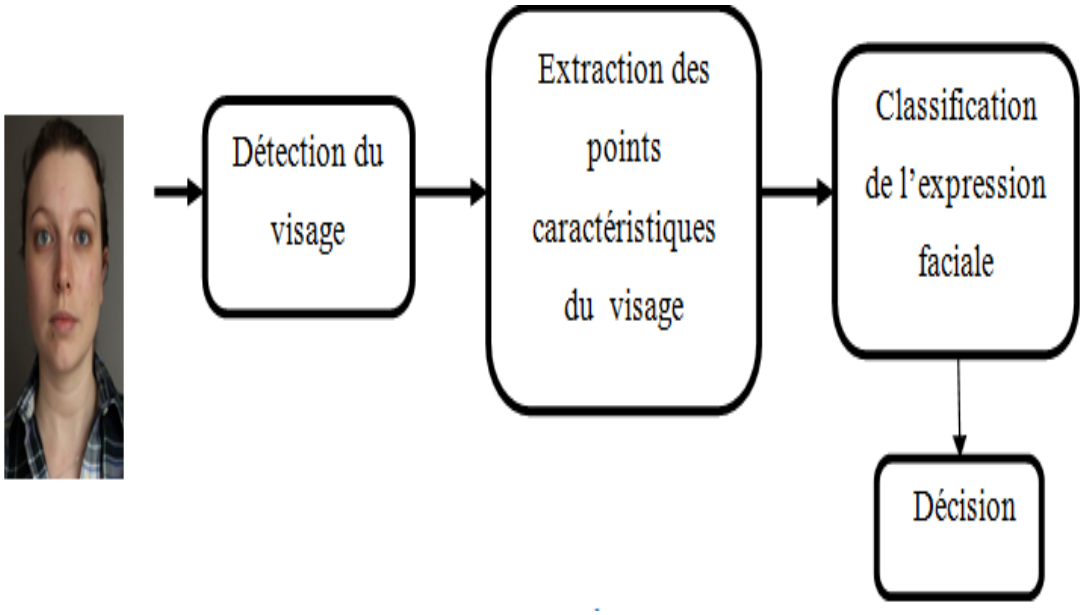


FIGURE 1.5 – Architecture d'un système de reconnaissance des expressions faciales.

1.3.1 Détection du visage

La détection de visage consiste a d´exterminer la présence ou l'absence de visages dans une image. C'est une tache préliminaire nécessaire à la plupart des techniques d'analyse du visage. Les techniques utilisées sont généralement issues du domaine de la reconnaissance des formes. En effet, le problème peut être vu comme la d´détection de caractéristiques communes à l'ensemble des visages humains : il s'agit de comparer une image à un modèle générique de visage et d'indiquer s'il y a ou non ressemblance.

Les principales difficultées sont la robustesse aux déférentes identités, poses du visage, expressions faciales et aux variations d'illumination. La sortie d'un d´détecteur de visage indique le nombre de visages présents dans l'image. De plus, la plupart des d´détecteurs de visage actuels sont aussi des localisateurs de visages : ils renvoient une localisation des visages détectes (une boite englobant par exemple) [10].

Un système complet de localisation du visage devrait faire face à certains défis décrits ci-dessous.

- Pose : Les traits du visage, y compris les yeux, le nez et la bouche, peuvent être partiellement invisibles ou déformés en raison de la pose relative du visage ou de la caméra.
- Occlusion : Les traits du visage peuvent être obstrués par une barbe, une moustache et des lunettes. De même, le maquillage peut provoquer l'apparition de régions artificielles sur le visage ou cacher les limites faciales normales.
- Expression : les traits du visage montrent de grands changements dans leur forme sous différentes expressions. Certaines caractéristiques peuvent devenir invisibles ou d'autres ne sont visibles que sous différentes expressions.
- Conditions d'acquisition de l'image : L'éclairage et les changements dans les caractéristiques de la caméra affectent de manière significative la chrominance des régions du visage. Certaines caractéristiques peuvent être masquées ou combinées avec des ombres ou des brillances sur le visage provoquant ainsi une perte d'informations sur les couleurs [11].

Plusieurs approches sont développées pour la détection de visage qui sont divisés en quatre catégories, certaines utilisent la forme, ou qui sont basés sur la couleur (couleur de peau), et l'autre se base sur l'apparence faciale, ou la dernière est une combinaison entre toutes les approches précédentes [12]. Les quatre subdivisions des approches de détection faciale sont : (figure 1.6)

- Approche basée sur les connaissances acquises.
- Template-matching.
- Approche basée sur l'apparence.
- Approche basée sur des caractéristiques invariantes.

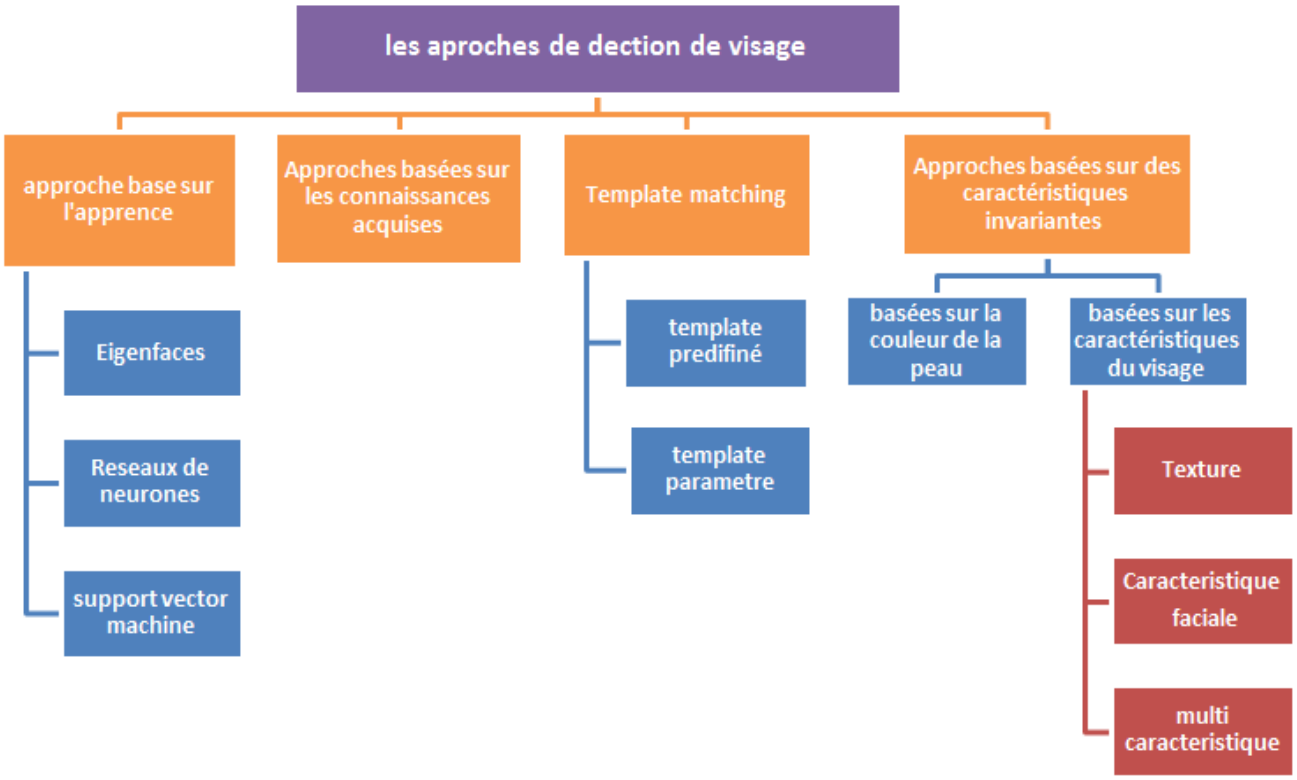


FIGURE 1.6 – Les approches de détection de visage.

A. Approche basée sur les caractéristiques invariantes

Ces approches utilisent les éléments invarintes aux variations d'illumination, d'orientation ou d'expression tels que la texture ou la signature de couleur de la peau pour la detection [13].

B. Approches basées sur les connaissances acquises

Ces méthodes se basent sur la connaissance des différents éléments qui constituent un visage et des relations qui existent entre eux. Ainsi, les positions relatives de différents éléments clés tels que la bouche, le nez et les yeux sont mesurées pour servir ensuite à la classification 'visage' 'non visage' chez Chiang et al. Le problème dans ce type de méthode est qu'il est difficile de bien définir de manière unique un visage. Si la définition est trop détaillée, certains visages seront ratés tandis que si la description est trop générale, le taux de faux positifs montera en flèche [14].

c. Approches basées sur le « Template matching »

Des modèles caractéristiques d'un visage entier ou de sous-partie de visage (bouche, œil, nez) sont créés. La localisation se fait ensuite sur base de la corrélation de ces modèles avec les candidats [14]. On trouve deux techniques de détection de visage appartenant à l'approche de détection basée sur le Template matching qui sont :
- Template prédéfinie
- Template déformable ou élastique

D. Approches basées sur l'apparence

Ces méthodes utilisent le même principe que celui présenté au point précédent mais se basent sur des modèles appris à partir d'un ensemble d'essai. Ces méthodes présentent l'avantage de s'exécuter très rapidement mais demandent un long temps d'entraînement. Les méthodes appartenant à cette catégorie ont montré de bons résultats par rapport aux trois autres types de méthodes. On peut citer parmi celles-ci, la méthode basée sur les réseaux de neurones de Rowley et al, la méthode de Schneiderman et Kanade basée sur un classifieur de Bayes naïf ainsi que le fameux algorithme de Viola et Jones fonctionnant en temps réel [14], et ce dernier sera détaillé ci-dessous.

• La methode Viola et Jones :

La methode de Viola et Jones est une méthode de détection d'objet dans une image numérique, proposée par les chercheurs Paul Viola et Michael Jones en 2001.Elle fait partie des toutes première méthodes capables de détecter efficacement et en temps réel des objets dans une image. Inventée à l'origine pour détecter des visages. La methode de Viola et Jones est l'une des méthodes les plus connues et le plus utilisées en particulier

pour la détection de visage et la détection de perssones [13].

E. Comparaison entre les différent Approche de détection de visage

Approche	Avantages	Inconvénients
Approches basées sur l'apparence	-Eigenface donne des bons résultats. -Les réseaux de neurones robustes au bruit	-Beaucoup temps de calcul -difficiles à construire. -phase d'apprentissage difficile à mener
Approches basées sur les connaissances acquis	-réduire le temps de calcul nécessaire par l'utilisation d'images sous Echantillonnées	-occasionne de nombreuses fausses détections - difficiles de traduire les connaissances humaines en des règles bien définies
Template-matching	-Simple au niveau de processus de détection. -Donne des résultats assez encouragent.	Mise à jour à chaque changement d'orientation.
Approches basées sur des caractéristiques invariantes	-Résiste contre les petits changements d'éclairage et la position de visage - la couleur de peau réduit la zone de recherche	- L'utilisation de la méthode basée sur la couleur de la peau nécessite des processus pour compléter la détection de visage.

TABLE 1.1 – Représente les avantages et les inconvénients de chaque méthode.

1.3.2 Extraction des points caractéristiques faciaux

Les points caractéristiques du visage sont principalement situés autour des composants faciaux tels que les yeux, la bouche, les sourcils, le nez et le menton. La détection des points caractéristiques du visage commence habituellement à partir d'une boite englobant rectangulaire renvoyée par un détecteur de visage qui localise ce dernier. L'extraction de caractéristiques géométriques telles que les contours des composants faciaux, les distances faciales, etc. fournit les emplacements ou les caractéristiques d'apparence peuvent être

calculées. De ce fait, les méthodes d'extraction des caractéristiques pour l'analyse d'expression peuvent êtres séparées en deux types d'approches : les méthodes basées sur les caractéristiques géométriques et les méthodes basées sur l'apparence [15] :

A. Les caractéristiques géométriques

Représentent la forme et l'emplacement des composants du visage (y compris la bouche, les yeux, les sourcils et le nez). Les composants faciaux ou les traits faciaux sont extraits pour former un vecteur de caractéristiques représentant la géométrie du visage [11]. Cette approche contient des différents modèles pour obtenir la reconnaissance des expressions faciales comme [15] :

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

B. Les caractéristiques d'apparence

Représentent les changements d'apparence (texture de la peau) du visage, tels que les rides et les sillons. Ces caractéristiques d'apparence peuvent être extraites sur tout le visage ou sur des régions spécifiques du visage. Selon les différentes méthodes d'extraction des caractéristiques, les effets de la rotation de la tête dans le plan et les différentes échelles de prise de vue du visage peuvent être éliminés par une normalisation de ce dernier avant l'extraction des caractéristiques ou par une représentation des caractéristiques avant l'étape de reconnaissance d'expression [11]. Et pour cette approche il y a des techniques pour obtenir la reconnaissance des expressions faciale comme [15] :

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

1.3.3 La classification

La dernière étape d'un système automatique d'analyse d'expression est la reconnaissance de l'expression faciale en fonction des caractéristiques extraites. Certains systèmes classent directement les expressions tandis que d'autres classent les expressions en recon-

naissant d'abord des unités d'action (AUs) particulières. De nombreux classifieurs ont été appliqués à la reconnaissance d'expression tels que : [11]

- réseaux de neurone (Neural Networks, NN).
- machines à vecteurs de support (Support Vector Machine, SVM).
- analyse Discriminante Linéaire (Linear discriminant analysis,LDA).
- analyse Discriminante Linéaire (Linear discriminant analysis, LDA).
- K-plus proche voisin (K Nearest Neighbor, KNN),
- régression logistique multinomiale (Multinomial Regression Logistic, MRL)
- modèles de Markov cachés (Hidden Markov Model, HMM).
- réseaux bayésiens (Bayesian Network, BN).

et d'autres. Ici, nous résumons les méthodes de reconnaissance d'expression à des méthodes basées sur des images statiques et sur des séquences vidéo. La méthode de reconnaissance basée sur des données statiques utilise uniquement l'image courante avec ou sans image de référence (il s'agit principalement d'une image de visage neutre) pour reconnaître l'expression d'une seule image. La méthode de reconnaissance basée sur des données dynamiques utilise les informations temporelles des séquences pour reconnaître les expressions d'une ou plusieurs images [11].

1.3.4 Bases des données des expressions faciales

Pour développer tout nouveau système de reconnaissance ou de détection des expressions faciales, il convient de choisir une base de données utilisée pour tester ce système. De plus, des bases de données communes sont nécessaires pour évaluer les algorithmes de manière comparative. Les bases de données ce qui est disponible peut être classé en deux catégories : les fondements des expressions faciales spontanéité et notions de base des expressions faciales.

Exemple de base des données [16] (Table1.2) :

La base de données	Description	Lien
DISFA	130 000 images vidéo stéréo en haute résolution 27 sujets adultes (12 femmes et 15 hommes) 66 points de repère du visage pour chaque image Résolution d'image de 1024X 768	http ://www.engr.du.edu/mmahoor/DISFA.htm
Ck+(Cohn -Kanade)	593 séquences vidéo à la fois posées et non posées émotions (spontanées) 123 sujets âgés de 18 à 30 ans Fournit des protocoles et des résultats de base pour le visage suivi des fonctionnalités, unités d'action et reconnaissance des émotions Résolutions d'image de 640 X480 et 640 X490	http ://www.consortium.ri.cmu.edu/ckagree/
B+	16 128 images faciales 28 sujets distincts pour 576 conditions d'observation Résolution d'image de 320 243	http ://vision.ucsd.edu/content/ extended-yale-face-database-b-b

Fer2013	compte environ 34034personnes bien structurées En Images de gris à 48X48 pixels les visages regroupés automatiquement par l'API Google de recherche d'image	https ://www.kaggle.com/datasets
DISFA	130 000 images vidéo stéréo en haute résolution 27 sujets adultes (12 femmes et 15 hommes) 66 points de repère du visage pour chaque image Résolution d'image de 1024X 768	http ://www.engr.du.edu/mmahoor/DISFA.htm
KDEF	4900 images d'expressions faciales humaines d'émotion 70 individus, sept expressions émotionnelles différentes avec 5 angles différents Résolution de l'image 562 762	http ://www.emotionlab.se/ resources/kdef
JAFFE	213 images de sept émotions faciales Dix modèles féminins japonais Six adjectifs d'émotion de 60 sujets japonais Résolution d'image 256 256	http ://www.kasrl.org/jaffe_info.html

TABLE 1.2 – Exemple de bases de données [16].

1.4 Conclusion

Dans ce chapitre nous avons présenté les théories et les représentations les plus connues dans la reconnaissance d'expressions faciales : les techniques de codifications, les approches de détections de visages ainsi que l'extraction des caractéristiques, Finalement, les bases de données fréquemment utilisées ont été décrites. A l'issue de cette étude, Dans le chapitre suivant nous présenterons le deep learning pour la reconnaissance des expressions faciales.

Chapitre 2

Deep learning et la reconnaissance des expressions faciales

2.1 Introduction

Le Deep Learning est un nouveau domaine de recherche du Machine Learning(ML), Il concerne les algorithmes inspirés par la structure et le fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données.

Les algorithmes de Deep Learning ont récemment contribué à faire progresser les performances des systèmes d'analyse des expressions du visage, plus précisément Les réseaux de neurones à convolution (Convolutional Neural Networks, CNN) sont une catégorie de réseaux de neurones qui se sont avérés très efficaces dans ces domaines.

Dans le deuxième chapitre nous présentons Le deep Learning (l'apprentissage profond) en citant ces différents Architectures et en se basant sur les réseaux de neurones convolutifs.Enfin, nous décrivons les étapes du notre système ECNN et les différents modèles qui l'utilisent.

2.2 L'apprentissage en profondeur (deep Learning)

2.2.1 Définition

Le deep learning est un sous-ensemble des techniques de machine learning à base de réseaux de neurones qui s'appuient sur des réseaux de neurones à plusieurs couches dites cachées.Celles-ci permettent par exemple de décomposer de manière hiérarchique le contenu d'une donnée complexe comme de la voix ou une image pour la classifier ensuite : identifier des mots pour la voix ou associer des tags descriptifs à des images.

C'est le principe de l'une des grandes catégories de réseaux de neurones de deep learning, les réseaux convolutionnels ou convolutifs (schéma ci-dessous). Un réseau peut être profond mais aussi large si le nombre de neurones est élevé dans chaque couche. Le deep learning remplace les méthodes antérieures du machine learning dites à base de « hand-craft features » qui consistaient à définir à la main les éléments à rechercher dans les objets (formes dans les images, etc).

Dans le deep learning, notamment pour la détection d'images, le réseau de neurones découvre tout seul ces composantes avec des niveaux d'abstraction évoluant de bas en haut et de couche en couche. Le deep learning sert le plus souvent au traitement du langage, de la parole, du bruit, de l'écriture et des images. Il a d'autres usages dans les outils d'aide à la décision [17].

2.2.2 Evolutions du deep learning

Les outils de deep learning s'appuient sur différentes variantes de réseaux de neurones pour leur mise en œuvre pratique. Leur histoire remonte aux perceptrons de Franck Rosenblatt de 1957.

L'histoire du deep learning a véritablement démarré plus de 20 ans plus tard, dans les années 1980. Il a cependant fallu attendre 1995 pour que l'on puisse les mettre en œuvre en pratique, sans doute, grâce aux progrès matériels, à la loi de Moore mais aussi aux progrès conceptuels, notamment aux travaux d'Alexander Weibel en 1989, Yann LeCun en 1988 et 1998 et à Geoff Hinton, particulièrement à partir de 1986. Le premier est le père des réseaux TDNN de reconnaissance de phonèmes, le second des réseaux convolutifs

à rétropropagation d'erreurs tandis que le dernier est considéré comme étant le père de nombreux concepts de base du deep learning [17].

Année	Contributeur	Contribution
300 AC	Aristotle	introduction de l'associationnisme, début de l'histoire des humains qui essayent de comprendre le cerveau
1873	Alexander Bain	introduction du Neural Groupings comme les premiers modèles de réseaux de neurones
1943	McCulloch and Pitts	introduction du McCulloch–Pitts (MCP) modèle considérer comme L'ancêtre des réseaux de neurones artificielles
1949	Donald Hebb	considérer comme le père des réseaux de neurones, il introduit la règle d'apprentissage de Hebb qui servira de fondation pour les réseaux de neurones modernes
1958	Frank Rosenblatt	introduction du premier perceptron 1974 Paul Werbos introduction de la retro propagation
1980	TeuvoKohonen	introduction des cartes auto organisatrices
1980	Kunihiko Fukushima	introduction du Neocognitron, qui a inspiré les réseaux de neuronesconvolutif
1982	John Hopfield	introduction des réseaux de Hopfield
1985	Hilton and Sejnowski	introduction des machines de Boltzmann
1986	Paul Smolensky	introduction de Harmonium, qui sera connu plus tard comme machines de Boltzmann restreintes

1986	Michael I. Jordan	définition et introduction des réseaux de neurones récurrent
1990	Yann LeCun	introduction de LeNet et montra la capacités des réseaux de neurones profond
1997	Schuster and Paliwal	introduction des réseaux de neurones récurrent bidirectionnelles
	Hochreiter and Schmidhuber	introduction de LSTM, qui ont résolu le problème du vanishing gradient dans les réseaux de neurones récurrent
2006	Geoffrey Hinton	introduction des Deepbelief Network
2009	Salakhutdinov and Hinton	introduction des Deep Boltzmann Machines
2012	Alex Krizhevsky	introduction de AlexNet qui remporta le challenge ImageNet

TABLE 2.1 – Les étapes majeurs du Deep Learning [18]

2.3 Pour quoi le choix deep Learning

Tout d'abord les différents algorithmes du deep Learning ne sont apparus qu'à l'échec de l'apprentissage automatique tentant de résoudre une grande variété de problèmes de l'intelligence artificielle (l'IA) :

- Afin d'améliorer le développement des algorithmes traditionnels dans de telles tâches de l'IA.
- De développer une grande quantité de données telle que les big data.
- De s'adapter à n'importe quel type de problème
- D'extraire les caractéristiques de façon automatique [19]

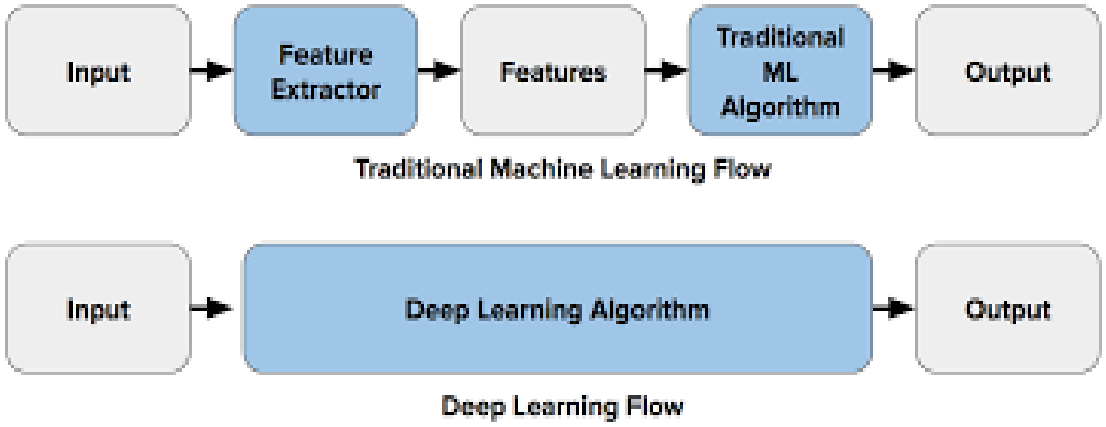


FIGURE 2.1 – Le procède du ML classique comparé à celui du Deep Learning[20].

2.4 Les différentes Architectures du Deep Learning

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

2.4.1 Les réseaux de neurones convolutifs

Convolutional Neural Network (CNN) (réseaux de neurones convolutifs) sont un type de réseau de neurones spécialisés pour le traitement de données ayant une topologie semblable à une grille. Les réseaux convolutifs ont connu un succès considérable dans les applications pratiques. Le nom « réseau de neurones convolutif » indique que le réseau emploie une opération mathématique appelée convolution. La convolution est une opération linéaire spéciale. Les réseaux convolutifs sont simplement des réseaux de neurones qui utilisent la convolution à la place de la multiplication matricielle dans au moins une de leurs couches. Ils ont de larges applications dans la reconnaissance de l'image et de la vidéo, les systèmes de recommandations [21] et le traitement du langage naturel (telles que la classification des phrases) [22].

Les CNN permettent de traiter directement de grandes quantités de données que sont les images pour plusieurs raisons [23] :

— Les images sont corrélées spatialement (les valeurs des pixels adjacents sont généralement très proches) et les couches de convolutions permettent de créer des liens entre ces données corrélées spatialement.

— Les couches de convolutions sont généralement suivies de couches de sous échantillonnage (pooling) qui diminuent grandement la taille des données.

A. Les différentes couches de CNN :

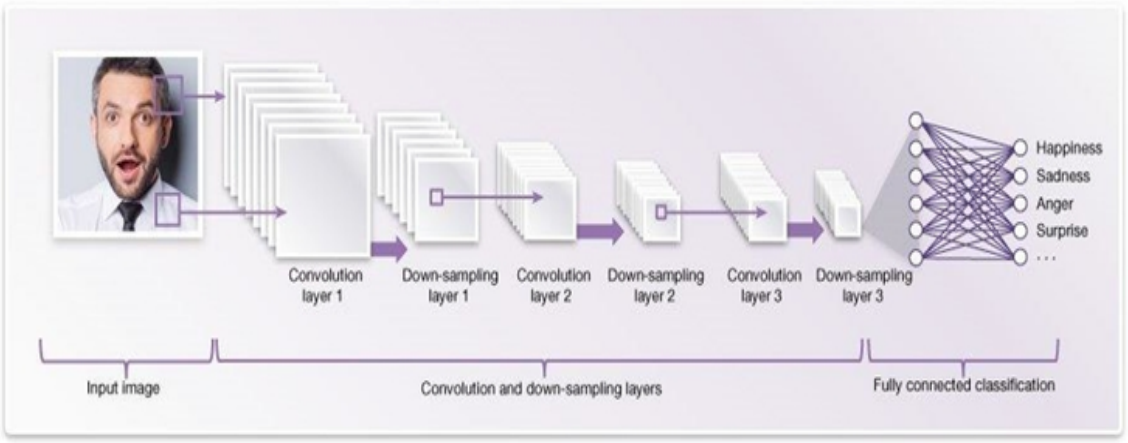


FIGURE 2.2 – Architecture d'un réseau de neurone convolutionnel [24].

A.1 la couche de convolution :

La convolution est le cœur du réseau de neurones convolutif, comme vous vous en doutez. À l'origine, une convolution est un outil mathématique (on parle de produit de convolution) très utilisé en retouche d'image, car il permet d'en faire ressortir l'extraction des caractéristiques à partir des images d'entrées, afin d'appliquer un bon filtre. En fait, une convolution prend simplement en entrée une image et un filtre (qui est une autre image), effectue un calcul, puis renvoie une nouvelle image (généralement plus petite) [25].

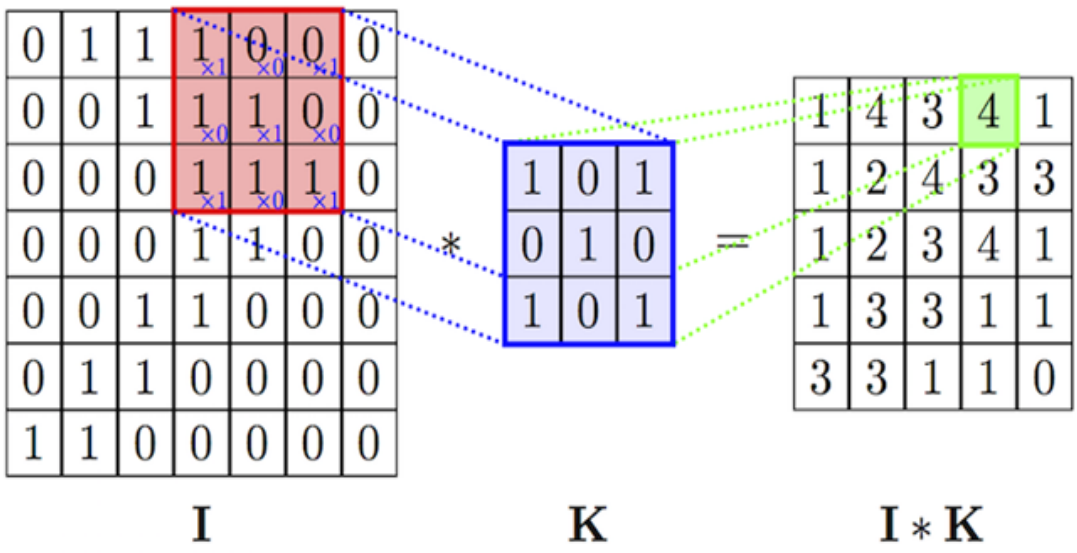


FIGURE 2.3 – Exemple d'une convolution[26]

A.2 Couche de pooling :

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs featuremaps, et applique à chacune d'entre elles l'opération de pooling.

L'opération de pooling (ousub-sampling)consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes.

Pour cela, on découpe l'image en cellules régulière, puis on garde au sein de chaque cellule la valeur maximale. En pratique, on utilise souvent des cellules carrées de petite taille pour ne pas perdre trop d'informations. Les choix les plus communs sont des cellules adjacentes de taille 2 × 2 pixels qui ne se chevauchent pas, ou des cellules de taille 3 × 3 pixels, distantes les unes des autres d'un pas de 2 pixels (qui se chevauchent donc). On obtient en sortie le même nombre de featuremaps qu'en entrée, mais celles-ci sont bien plus petites.

La couche de pooling permet de réduire le nombre de paramètres et de calculs dans le réseau. On améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage.

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

Par exemple, dans le cas du Max pooling, on définit une fenêtre glissante de taille mxm et on prend le plus grand élément du featuremap dans cette fenêtre [27] (figure 2.4).

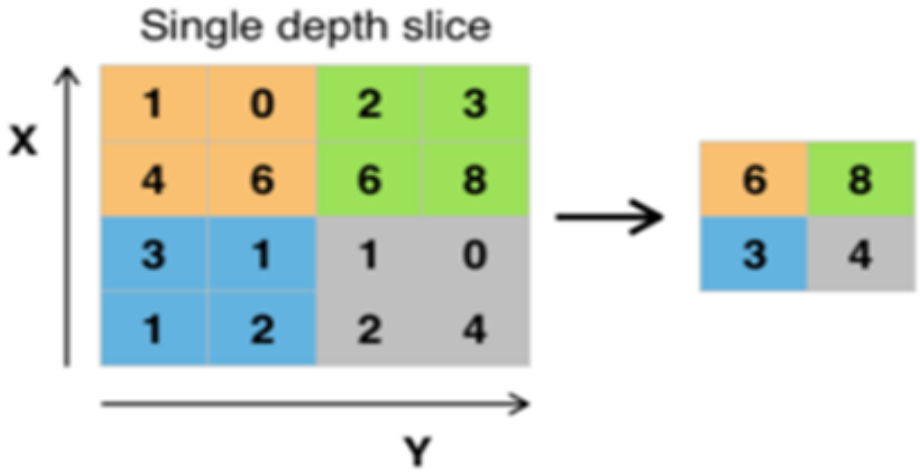


FIGURE 2.4 – Exemple d'un pooling [27].

A.3 La couche totalement connectée

Une couche entièrement connectée implique des poids, des biais et des neurones. Il connecte les neurones d'une couche aux neurones d'une autre couche. Il est utilisé pour classer les images entre différentes catégories par formation.

A.4 Fonctions d'activation

Une fonction d'activation est une fonction mathématique appliquée à un signal en sortie d'un neurone artificiel. Il dérive de l'équivalent biologique qui signifie "potentiel d'activation, lorsque le seuil de stimulation aura été atteint entraîne une réponse du neurone. Son but principal est de pouvoir permettre aux réseaux de neurones d'apprendre des fonctions plus complexes qu'une simple régression linéaire, car le simple fait de multiplier les poids d'une couche cachée est juste une transformation linéaire :

RectifiedLinear Unit (ReLU) : Elle est utilisée après chaque opération de convolution, ou toutes les valeurs de pixels négatifs sont mises à zéro. Le but de ReLU est d'introduire

la non-linéarité dans notre CNN, puisque la plupart des données du monde réel, puisque la plupart de caractéristiques appliquées à l'une des cartes d'entrée donne une carte de sortie qui est également appelée carte de caractéristiques rectifiées [28].

2.4.2 Réseau de neurones récurrents

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

Les RNN ont connu un grand succès dans de nombreuses tâches de traitement du langage naturel. Les plus grands exploits des RNN ont été accomplis par l'architecture LSTM car ils sont bien meilleurs pour capturer des dépendances à long terme [29].

- Modélisation du langage et génération de texte
- Traduction automatique
- La reconnaissance vocale
- Description des images

2.4.3 Modèle génératif

Si les modèles discriminatifs comme (CNN, RNN) sont utilisés pour prédire les données du label et de l'entrée, tant disque le modèle génératif décrit comment générer les données, il apprend et fait des prédictions en utilisant la loi de Bayes[30].

Les modèles génératifs sont capables de bien plus que la simple classification comme par exemple générer de nouvelles observations [29].

Voici quelques exemples de modèle génératif :

- Boltzmann Machines
- Restricted Boltzmann Machines
- Deep Belief
- Deep Boltzmann Machines
- Generative Adversarial Networks
- Generative Stochastic Networks
- Adversarial autoencoders

2.5 Domaine d'application de Deep Learning

L'apprentissage en profondeur investit progressivement notre quotidien :

- La reconnaissance vocale.
- Le tagging automatique de morceaux de musique.
- La synthèse vocale avancée.
- Deep Boltzmann Machines
- L'étiquetage automatique d'image.
- La conception de nouvelles molécules pharmaceutiques.

Toutes ces applications mettent aujourd'hui en œuvre des techniques de Deep Learning [31].

2.6 Système ECNN

2.6.1 Présentation du système ECNN

Nous suggérons une conception de ECNN un système de reconnaissance d'expression faciale à partir d'un visage en temps réels utilise les réseaux de neurones CNN, qui permet de détecter un visage d'une personne à partir d'une image ou via une caméra pour connaître l'expression avec un taux de précision associé au sept expressions universel à savoir la joie, le dégoût, la peur, la colère, la tristesse , la surprise et le neutre.

2.6.2 Architecteur du système ECNN

Le système de reconnaissance des expressions faciales est effectué en trois étapes principales :

- Détection du visage et des parties du visage.
- Extraction des caractéristiques et Classification des expressions.

Dans ce qui suit, nous détaillerons chacune des étapes du système ECNN

A. La Détection de visage

L'efficacité de système de reconnaissance faciale basée sur l'authentification faciale dépend essentiellement de la méthode utilisée pour localiser le visage dans l'image. Dans notre méthode, nous utilisons l'algorithme Viola-Jones pour détecter diverses parties du visage humain telles que la bouche, les yeux, le nez, les narines, les sourcils, la bouche, les lèvres, les oreilles, etc. Cet Algorithme explore les caractéristiques de type Haar via le classificateur Cascade, qui peut efficacement combiner de nombreuses fonctionnalités et déterminer les différents filtres sur un classificateur résultant [32].

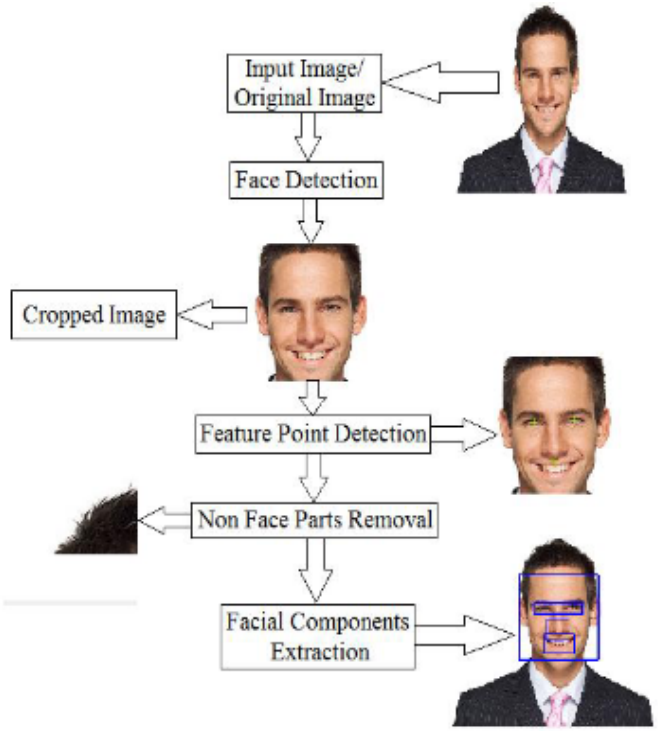


FIGURE 2.5 – La détection du visage avec Viola-Jones [33].

B. Extraction de caractéristiques faciales et Classification des expressions

Si le visage est détecté dans l'image, le système lance le processus d'extraction des caractéristiques qui va convertir les données des pixels à des représentations et configuration plus réduite et optimal pour que la représentation extraite soit utilisé dans le processus de la classification.

L'objectif de l'étape de la classification est la reconnaissance de l'expression faciale en fonction des caractéristiques extraites. Cette étape est basée sur l'utilisation de la technologie de réseau de neurones convolutionel (CNN). En effet, nous avons utilisé deux architectures de CNN, inspiré de VGGNet [34] et ResNet50 [35]. Dans ce qui suit, nous présentons les deux architectures VGGNet et ResNet pour faciliter la comprehension de nos deux modele nomé modele -A- et modele -B-.

B.1 Architecture VGGnet

VGG a été proposé par le groupe VGG à Oxford Université, VGG a été inventé dans le but d'améliorer la précision de la classification en augmentant la profondeur des CNN. Une amélioration de VGG par rapport à AlexNet consiste à remplacer les plus gros cœurs de convolution (11x11, 5x5) dans AlexNet avec plusieurs noyaux de convolution 3x3 consécutifs. Selon à la différence du nombre de couches du neural réseau, il peut être divisé en VGG16 et VGG19 ayant respectivement 16 et 19 couches de poids [34].

VGG Net prend l'entrée de 224 × 224 images RVB et les fait passer à travers une pile de couches convolutives avec la taille de filtre fixe de 3 × 3. Il y a cinq filtres de regroupement maximum intégrés entre les couches convolutives afin de sous-échantillonner la représentation d'entrée (image, matrice de sortie de couche cachée, etc.) L'empilement de couches convolutives est suivi de 3 couches entièrement connectées, ayant respectivement 4096, 4096 et 1000 canaux. La dernière couche est une couche soft-max La figure ci-dessous montre la structure du réseau VGG 19.

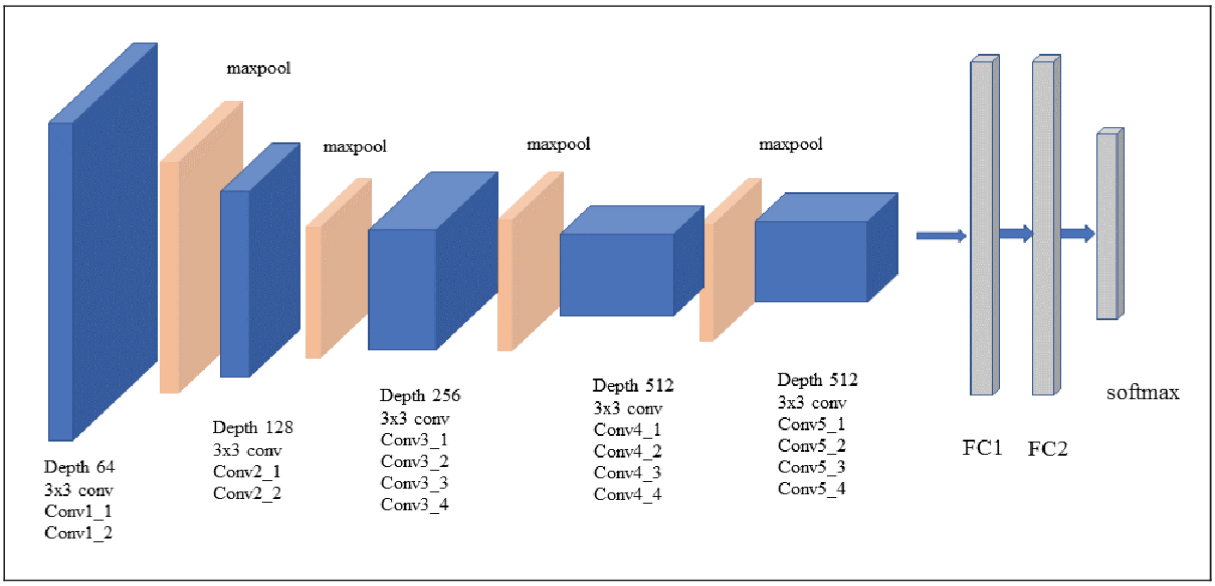


Fig. 3. VGG-19 network architecture

FIGURE 2.6 – Architecteur de VGG-19[36].

B.2 Architecture Resent50

Contrairement aux architectures de reseau sequentielles traditionnelles telles que VGG, RestNet est plutot une forme qui repose sur des modules de micro architecture (egalement appeles "architecture reseau dans reseau ",le terme micro-architecture désigne l'ensemble des "blocs de construction " utilisés pour construire le reseau [35] .

Le réseau résiduel profond est presque similaire aux réseaux qui ont des couches de convolution, pooling, d'activation et les couches entièrement connectées empilées les unes sur les autres. La seule construction du réseau simple pour en faire un réseau résiduel est la connexion d'identité entre les couches. La capture d'écran ci-dessous (figure 2.7)montre le bloc résiduel utilisé dans le réseau. Vous pouvez voir la connexion d'identité comme la flèche incurvée provenant de l'entrée et descendant jusqu'à la fin du bloc résiduel. Le ResNet-50 est fait de cinq blocs convolutifs empilés les uns sur les autres (figure 2.8) [35].

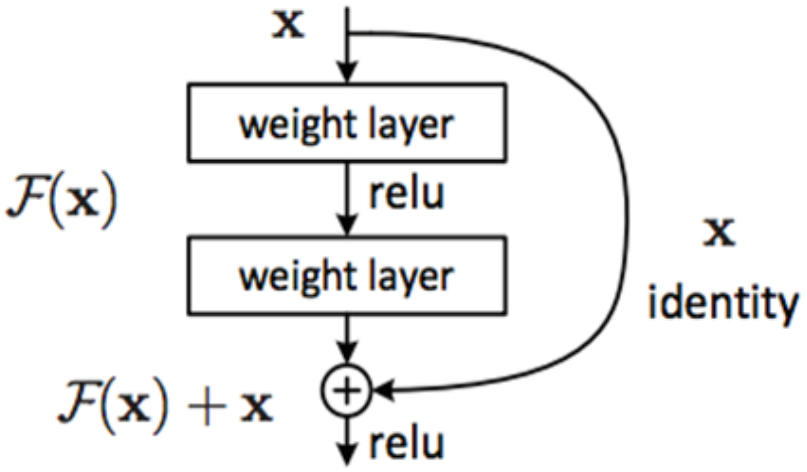


FIGURE 2.7 – Connexion résiduelle [35].

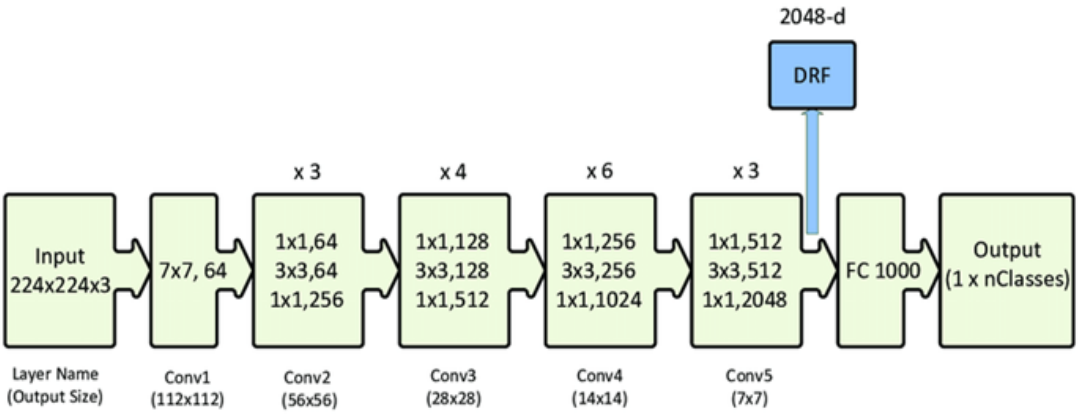


FIGURE 2.8 – Architecture ResNet-50 représentée avec les unités résiduelles, la taille des filtres et le sorties de chaque couche convolutive [35].

2.7 Conclusion

Dans ce chapitre, nous avons défini le deep learning, son historique ainsi que ses différentes architecteurs on a focalisé notre attention sur les réseaux de neurones convolutifs CNN et leurs structures, et ses différentes couches, puis quelques domaines de l'utilisation du deep learning. Enfin, nous avons présenté le système ECNN et leur architecteur

avec les différents modèles inspére de VGGnet et ResNet50.Le chapitre suivant sera la présentation de modèle -A- et -B- et l'implémentation et la réalisation de notre système de ECNN.

Chapitre 3

Implémentation et Réalisation de ECNN

3.1 Introduction

L'objectif de ce chapitre est de présenter les étapes de l'implémentation de système de reconnaissance des expressions faciales. Nous commençons tout d'abord par la présentation des ressources, du langage et de l'environnement de développement que nous avons utilisé. Puis on va créer deux modèles avec des architectures différentes, par la suite, on appliquera ces modèles sur la base fer2013. Et afin de réaliser tout ça nous utiliserons le langage python et les différentes bibliothèques comme Tensorflow et Keras, pour l'apprentissage et la classification, ainsi que quelque technique simple pour améliorer les performances des models comme Dropout.

3.2 Logiciels et outils

3.2.1 Google Colab

Google Colaboratory ou Colab est un service cloud gratuit pour vous initier au Deep Learning, il est prend désormais en charge le GPU gratuit [37]. Colab permet [37] :

- d'améliorer vos compétences de codage en langage de programmation Python.
- de développer des applications en Deep Learning en utilisant des bibliothèques Python populaires telles que Keras, TensorFlow, PyTorch et OpenCV.
- d'utiliser un environnement de développement (Jupyter Notebook) qui ne nécessite aucune configuration.

- les documents Colab (Jupyter Notebook) sont enregistrés directement sur votre compte Google Drive.
- la fonctionnalité qui distingue Colab des autres services est l'accès à un processeur graphique GPU, totalement gratuitement

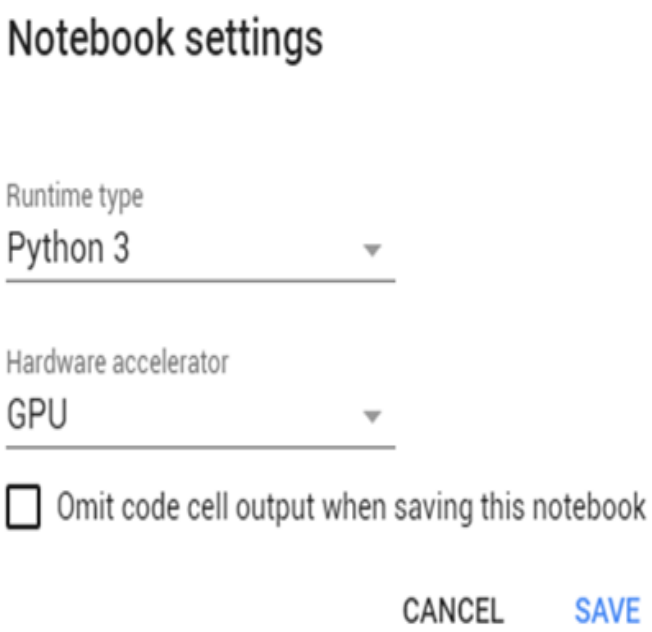


FIGURE 3.1 – Réglage du GPU libre.

pourquoi les GPUs ?

Pour la plupart des approches de l'apprentissage en profondeur, les GPU sont essentiels en raison de la quantité de données sur laquelle vous devez opérer. Un GPU contient des milliers d'unités de calcul contre une dizaine pour un CPU, le calcul mathématique de Deep Learning sur un CPU peut prendre des mois De quoi accélérer la vitesse d'apprentissage .

3.2.2 Python

Python est un langage portable, dynamique, extensible, gratuit, qui permet (sans l'imposer) une approche modulaire et orientée objet de la programmation. Python est développé depuis 1989 par Guido van Rossum et de nombreux contributeurs bénévoles [38].

- La syntaxe de Python est très simple et, combinée à des types de données évolués (listes, dictionnaires...), conduit à des programmes à la fois très compacts et très lisibles.
- Python gère ses ressources (mémoire, descripteurs de fichiers...) sans intervention du programmeur, par un mécanisme de comptage de références (proche, mais différent, d'un garbage collector).
- Il n'y a pas de pointeurs explicites en Python.
- Python est (optionnellement) multi-threadé.
- Python est orienté-objet. Il supporte l'héritage multiple et la surcharge des opérateurs. Dans son modèle objets, et en reprenant la terminologie de C++, toutes les méthodes sont virtuelles.
- Python intègre, comme Java ou les versions récentes de C++, un système d'exceptions, qui permettent de simplifier considérablement la gestion des erreurs
- Python est extensible : comme Tcl ou Guile, on peut facilement l'interfacer avec des bibliothèques C existantes. On peut aussi s'en servir comme d'un langage d'extension pour des systèmes logiciels complexes.
- Enfin, Python est un langage de choix pour traiter le XML.

FIGURE 3.2 – Le logo du python [39].

3.2.3 Opencv

OpenCV (Open Source Computer Vision Library) est une bibliothèque libre de logiciels de vision et d'apprentissage automatique. OpenCV a été conçu pour fournir une infrastructure commune pour les applications de vision par ordinateur et pour accélérer l'utilisation de la perception de la machine dans les produits commerciaux. Étant un produit sous licence BSD, OpenCV permet aux entreprises d'utiliser et de modifier facilement le code [40].

La bibliothèque compte plus de 2500 algorithmes optimisés, qui comprennent un ensemble complet d'algorithmes de vision informatique et d'apprentissage automatique classiques et de pointe. Ces algorithmes peuvent être utilisés pour détecter et reconnaître des visages, identifier des objets, classer des actions humaines dans des vidéos, suivre des mouvements de caméra, suivre des objets en mouvement, extraire des modèles 3D d'objets, produire des nuages de points 3D à partir de caméras stéréo, assembler des images ensemble pour produire une image haute résolution d'une scène entière,trouve des images similaires dans une base de données d'images,retirer les yeux rpoges des images prises au flash,suivre les mouvements des yeux,reconnaitre le décor et établir des repères pour des le superposer à la réalité augementée [40].

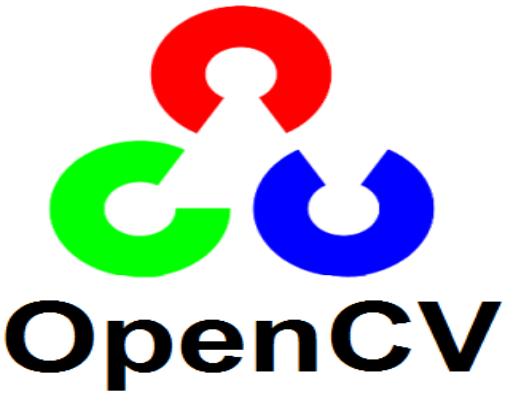


FIGURE 3.3 – Le logo du OpenCv [41].

3.2.4 TensorFlow

TensorFlow est une bibliothèque open source de Machine Learning, permettant de développer et d'exécuter des applications de Machine Learning et de Deep Learning, Créé par l'équipe Google Brain en 2011, sous la forme d'un système propriétaire dédié aux réseaux de neurones de Deep Learning. En 2015, il a été renommé TensorFlow et Google l'a rendu open source. Depuis lors, il a subi plus de 21000 modifications par la communication et est passé en version 1.0 en février 2017 [42].

TensorFlow est une bibliothèque de Machine Learning, il s'agit d'une boîte à outils permettant de résoudre des problèmes mathématiques extrêmement complexes avec aisance. Elle permet aux chercheurs de développer des architectures d'apprentissages expérimentaux et de les transformer en logiciels [42].

TensorFlow regroupe un grand nombre de modèles et d'algorithmes de Machine Learning et de Deep Learning. Son API front-end de développement d'applications repose sur le langage de programmation Python, tandis que l'exécution de ces applications s'effectue en C++ haute-performance [42].

Cette bibliothèque permet notamment d'entraîner et d'exécuter des réseaux de neurones pour la classification de chiffres écrits à la main, la reconnaissance d'image, les plongements de mots, les réseaux de neurones récurrents, les modèles séquence-to-séquence pour la traduction automatique, ou encore le traitement naturel du langage [42].

3.2.5 Keras

Keras est une librairie Python qui encapsule l'accès aux fonctions proposées par plusieurs librairies de machine learning, en particulier Tensorflow. De fait, Keras n'implémente pas nativement les méthodes. Elle sert d'interface avec Tensorflow simplement [43].

Keras est le cadre d'apprentissage profond le plus utilisé parmi les 5 meilleures équipes gagnantes sur Kaggle. Parce que Keras rend plus facile de mener de nouvelles expériences [43].

3.2.6 NumPy

La bibliothèque NumPy est une « bibliothèque fondamentale nécessaire pour l'informatique scientifique avec Python » fournir mise en œuvre efficace de tableaux multidimensionnels avec un large éventail de fonctions mathématiques pour fonctionner sur eux. NumPy interfaces avec des paquets d'algèbre linéaire bien développés, stables et largement utilisés : BLAS et LAPACK, ainsi que le paquet de transformation rapide de Fourier FFTPACK. D'où l'utilisation et le développement de NumPy exige une connaissance et une compréhension avancées des méthodes numériques et application [44].

3.3 Base de données utilisée

Pour augmenter la précision et améliorer les performances des modèles, il est nécessaire d'entrainer avec de nombreux d'échantillons d'images, nous avons donc utilisé la base de donnée fer2013 pour notre système pour l'apprentissage.
La base de données Fer2013 contient 34034 images expressions faciales chacune une taille de 48x48 pixels, qui sont reparties en trois parties à savoir :

FIGURE 3.4 – Echantillons d'mages fer2013.

- 80 % : représentent les images d'Entraînement(Training).
- 10 % : représentent les images de teste public (public Test).
- 10 % : représentent les images de teste privé (private Test).

3.4 Architecture des réseaux

Au cours de nos expérimentations, nous avons créé deux modèles avec différentes architectures où nous avons appliqué sur les deux modèles la base Fer2013 et pour chaque modèle, nous avons fait une évaluation sur le nombre d'époques. Dans ce qui suit nous présenterons l'architecture des 2 modèles et le résultat obtenu à chaque fois :

3.4.1 Modele -A-

Le modele A est inspiré du l'architecture VGG, l'architecture suivante représente les différentes couches que contient notre modèle :

- Le premier modèle que nous présentons est composé de sept couches convolutives suivie de cinq couches de max pooling et trois couches de entièrement connectée.
- l'image en entrée a une taille de 48*48 , chaque couche de convolution composée de plusieurs filtres (16 pour la première couche, 32 pour la deuxieme couche , 64 pour la troisiemme couche,128pour les deux avant derniere et 512 pour les 2 dernière couches), la taille de chaque filtre est inchangeable pour les sept couches de convolution (elle est de 3*3), la fonction d'activation utilisée pour toutes les couches de convolution est ReLU ainsi que le maxpooling afin de réduire la taille des images. nous allons utiliser un Dropout pour réduire le sur ajustement avec un pourcentage de 20 % à chaque fois après chaque couche de convolution, notre partie classification se compose de deux couches entièrement connectées. Cependant, ces couches ne peuvent accepter que des données à une dimension 1D. nous passerons par la fonction Flatten qui convertira les données à une dimension pour des donnees de 2D.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 16)	160
batch_normalization (BatchNo	(None, 48, 48, 16)	64
activation (Activation)	(None, 48, 48, 16)	0
dropout (Dropout)	(None, 48, 48, 16)	0
conv2d_1 (Conv2D)	(None, 48, 48, 32)	4640
batch_normalization_1 (Batch	(None, 48, 48, 32)	128
activation_1 (Activation)	(None, 48, 48, 32)	0
dropout_1 (Dropout)	(None, 48, 48, 32)	0
conv2d_2 (Conv2D)	(None, 48, 48, 64)	18496
batch_normalization_2 (Batch	(None, 48, 48, 64)	256
activation_2 (Activation)	(None, 48, 48, 64)	0
max_pooling2d (MaxPooling2D)	(None, 24, 24, 64)	0
dropout_2 (Dropout)	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 24, 24, 128)	73856
batch_normalization_3 (Batch	(None, 24, 24, 128)	512
activation_3 (Activation)	(None, 24, 24, 128)	0
max_pooling2d_1 (MaxPooling2	(None, 12, 12, 128)	0
dropout_3 (Dropout)	(None, 12, 12, 128)	0
conv2d_4 (Conv2D)	(None, 12, 12, 128)	147584
batch_normalization_4 (Batch	(None, 12, 12, 128)	512
activation_4 (Activation)	(None, 12, 12, 128)	0
max_pooling2d_2 (MaxPooling2	(None, 6, 6, 128)	0

dropout_4 (Dropout)	(None, 6, 6, 128)	0
conv2d_5 (Conv2D)	(None, 6, 6, 512)	590336
batch_normalization_5 (Batch	(None, 6, 6, 512)	2048
activation_5 (Activation)	(None, 6, 6, 512)	0
max_pooling2d_3 (MaxPooling2	(None, 3, 3, 512)	0
dropout_5 (Dropout)	(None, 3, 3, 512)	0
conv2d_6 (Conv2D)	(None, 3, 3, 512)	2359808
batch_normalization_6 (Batch	(None, 3, 3, 512)	2048
activation_6 (Activation)	(None, 3, 3, 512)	0
max_pooling2d_4 (MaxPooling2	(None, 1, 1, 512)	0
dropout_6 (Dropout)	(None, 1, 1, 512)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 256)	131328
batch_normalization_7 (Batch	(None, 256)	1024
activation_7 (Activation)	(None, 256)	0
dropout_7 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 512)	131584
batch_normalization_8 (Batch	(None, 512)	2048
activation_8 (Activation)	(None, 512)	0
dropout_8 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 7)	3591

Total params: 3,470,023
Trainable params: 3,465,703
Non-trainable params: 4,320

FIGURE 3.5 – L'architecture du modele -A-.

3.4.2 Modele -B-

Le modele -B- est inspiré de l'architecture Resnet50, voici un appercu du modele -B-

```
Model: "Net50"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 48, 48, 1)]  0
__________________________________________________________________________________________________
conv1 (Conv2D)                  (None, 46, 46, 8)    80          input_1[0][0]
__________________________________________________________________________________________________
bn_conv1 (BatchNormalization)   (None, 46, 46, 8)    32          conv1[0][0]
__________________________________________________________________________________________________
activation (Activation)         (None, 46, 46, 8)    0           bn_conv1[0][0]
__________________________________________________________________________________________________
res2a_branch2a (Conv2D)         (None, 46, 46, 32)   288         activation[0][0]
__________________________________________________________________________________________________
bn2a_branch2a (BatchNormalizati (None, 46, 46, 32)   128
res2a_branch2a[0][0]
.
.
.
.
__________________________________________________________________________________________________
flatten (Flatten)               (None, 9216)         0           avg_pool[0][0]
__________________________________________________________________________________________________
fc1024 (Dense)                  (None, 512)          4719104     flatten[0][0]
__________________________________________________________________________________________________
fc7 (Dense)                     (None, 7)            3591        fc1024[0][0]
==================================================================================================
Total params: 10,646,583
Trainable params: 10,620,071
Non-trainable params: 26,512
__________________________________________________________________________________________________
```

Le meme nombre des couches convonutives du ResNet50 a été utilisé.

Le modele -B- utilise 50 couches plus une couche de fully connected .l'architecture est montré dans la figure 3.6, le maxpooling a ete enlevé a cause de la taille du filtre dans la premiere convolution.

3.5 Implémentation et Réalisation deECNN

Le système ECNN au niveau de l'implémentation contient plusieurs modules, dans ce qui suit, nous allons spécifier chacun.

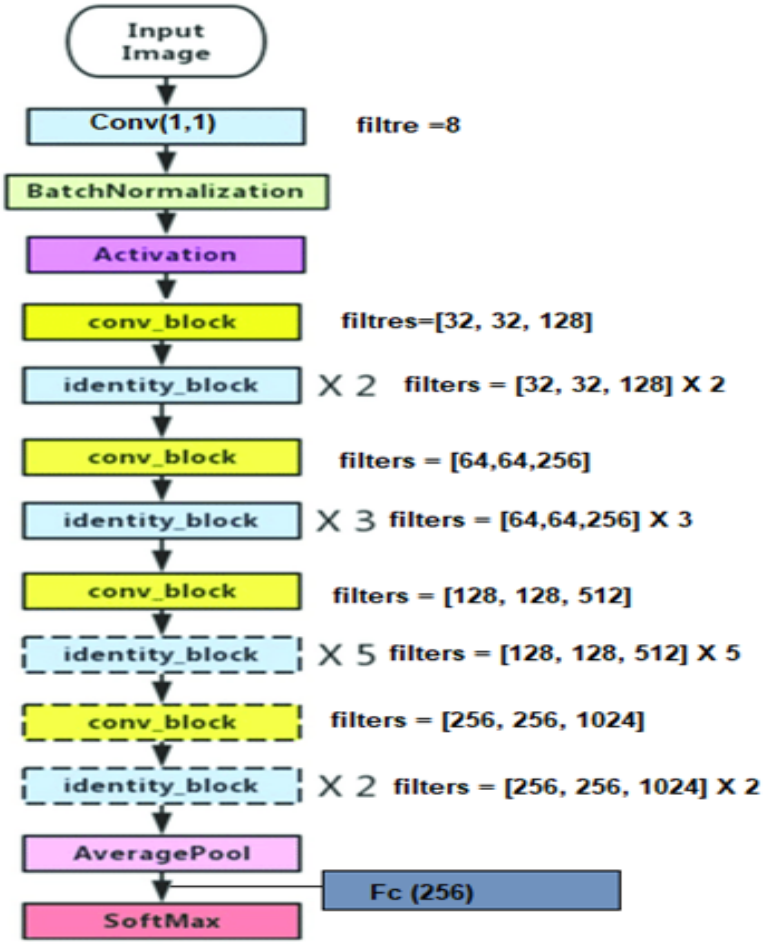


FIGURE 3.6 – L'architecture du modele -B-.

3.5.1 Le module charge et pretraitement de la base

Dans la base des données utilisées (FER2013), les images sont traitées de manière à ce que les visages soient presque centrés et chaque visage occupe environ la même quantité d'espace dans chaque image. Chaque image a été classé dans l'une des sept classes exprimer différentes émotions du visage. Ces émotions faciale sont été classés comme suit : 0 = en colère, 1 = dégoûté, 2 = Peur, 3 = Heureux, 4 = Triste, 5 = Surprise et 6 = Neutre. Les images données sont divisées en trois ensembles différents qui sont l'apprentissage, la validation, et ensembles de test. Il y a environ 28 709 images de formation, 3589 images de validation et 3589 images à tester. Chaque image est donnée sous forme de chaîne images de taille (48 × 48) stockée en tant que vecteur de ligne sous le format (.csv) [45]. Le code ci-dessous charge la base de données et faire un traitement des images et les préparé pour les servir au modèle CNN choisi par la suite.

A. La fonction 'load data'

Cette fonction lit le fichier fer2013.csv et convertit la séquence de pixels de chaque ligne en image de dimension 48 * 48 et retourne des images contiens des visages et des étiquettes d'émotion.

```
def load_fer2013():


    data = pd.read_csv('/content/drive/My Drive/fer2013.csv')
    data = (data[data['pixels'].notnull()])
    pixels = data['pixels'].tolist()
    width, height = 48, 48
    faces = []
    for pixel_sequence in pixels:
        face = [int(pixel) for pixel in pixel_sequence.split(' ')]
        face = np.asarray(face).reshape(width, height)
        face = cv2.resize(face.astype('uint8'),image_size)
        faces.append(face.astype('float32'))
    faces = np.asarray(faces)
    faces = np.expand_dims(faces, -1)
    emotions = (data['emotion'])#.values
    return faces, emotions
```

FIGURE 3.7 – La fonction load data.

B. La fonction preprocess input

Cette fonction est une procédure standard pour l'apprentissage elle est jugé la meilleur méthode pour le modèle de réseaux de neurones dans les problèmes de vision par ordinateur, Les images sont redimensionnées à [0,1] en les divisant par 255. La soustraction par 0,5 et la multiplication par 2 modifient les jusqu'à [-1,1]. [-1,1].

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

FIGURE 3.8 – La fonction preprocess input.

3.5.2 Le Module des Modeles CNNs

Ce module contient les deux architectures testées au court de notre étude.

A. Modele -A-

Un appercu sur le code du modele -A- figure 3.9

```
model = Sequential()

model.add(Conv2D(16, (3,3), padding='same', input_shape=(48,48,1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.2))

model.add(Conv2D(32, (3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.2))

model.add(Conv2D(64, (3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(128, (3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(128, (3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(512, (3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(512, (3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.2))

model.add(Flatten())
```

FIGURE 3.9 – Modele -A-.

B. Modele -B-

Un appercu sur le code du modele -B- figure 3.10

```
X = X_input
# Stage 1

X = Conv2D(8, (3, 3), strides = (1, 1), name = 'conv1', kernel_initializer = glorot_uniform(seed=0))(X)
X = BatchNormalization(axis = 3, name = 'bn_conv1')(X)
X = Activation('relu')(X)
# removed maxpool
#X = MaxPooling2D((3, 3), strides=(2, 2))(X)

# Stage 2
X = convolutional_block(X, f = 3, filters = [32, 32, 128], stage = 2, block='a', s = 1)
X = identity_block(X, 3, [32, 32, 128], stage=2, block='b')
X = identity_block(X, 3, [32, 32, 128], stage=2, block='c')


# Stage 3
X = convolutional_block(X, f = 3, filters = [64,64,256], stage = 3, block='a', s = 2)
X = identity_block(X, 3, [64,64,256], stage=3, block='b')
X = identity_block(X, 3, [64,64,256], stage=3, block='c')
X = identity_block(X, 3, [64,64,256], stage=3, block='d')

# Stage 4
X = convolutional_block(X, f = 3, filters = [128, 128, 512], stage = 4, block='a', s = 2)
X = identity_block(X, 3, [128, 128, 512], stage=4, block='b')
X = identity_block(X, 3, [128, 128, 512], stage=4, block='c')
X = identity_block(X, 3, [128, 128, 512], stage=4, block='d')
X = identity_block(X, 3, [128, 128, 512], stage=4, block='e')
X = identity_block(X, 3, [128, 128, 512], stage=4, block='f')

# Stage 5
X = convolutional_block(X, f = 3, filters = [256, 256, 1024], stage = 5, block='a', s = 2)
X = identity_block(X, 3, [256, 256, 1024], stage=5, block='b')
X = identity_block(X, 3, [256, 256, 1024], stage=5, block='c')

# AVGPOOL .
X = AveragePooling2D((2,2), name='avg_pool')(X)

# output layer
X = Flatten()(X)
X = Dense(512, activation = 'relu', name='fc1024' , kernel_initializer = glorot_uniform(seed=0))(X)
X = Dense(classes, activation='softmax', name='fc' + str(classes), kernel_initializer = glorot_uniform(seed=0))(X)
```

FIGURE 3.10 – Modele -B-.

3.5.3 Module d'apprentissage

Une fois que le modèle est formé, nous utilisons la fonction «fit ()», Pour l'apprentissage du réseau .

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

FIGURE 3.11 – un appercu de la fonction fit().

```
Epoch 82/300
898/897 [==============================] - 91s 102ms/step - loss: 0.4256 - accuracy: 0.8426 - val_loss: 1.0460 - val_accuracy: 0.6906
Epoch 83/300
898/897 [==============================] - 91s 102ms/step - loss: 0.4226 - accuracy: 0.8448 - val_loss: 1.0501 - val_accuracy: 0.6896
Epoch 84/300
898/897 [==============================] - 91s 102ms/step - loss: 0.4224 - accuracy: 0.8445 - val_loss: 1.0488 - val_accuracy: 0.6899
Epoch 85/300
898/897 [==============================] - 91s 102ms/step - loss: 0.4181 - accuracy: 0.8452 - val_loss: 1.0485 - val_accuracy: 0.6899
Epoch 86/300
898/897 [==============================] - 91s 102ms/step - loss: 0.4200 - accuracy: 0.8456 - val_loss: 1.0493 - val_accuracy: 0.6896
Epoch 87/300
898/897 [==============================] - 91s 102ms/step - loss: 0.4167 - accuracy: 0.8488 - val_loss: 1.0508 - val_accuracy: 0.6903
```

FIGURE 3.12 – appercu de la phase d'apprentissage.

La figure 3.12 représente notre modèle qui entraîne les données, puis les valide. Le nombre d'époques est le nombre de fois où le modèle parcourt les données. Plus nous avons d'époques, plus le modèle s'améliorera, jusqu'à un certain point, ou le modèle cessera de s'améliorer.

3.6 L'interface de l'application

Plutôt que d'adopter une approche purement théorique, nous avons pensé qu'il est mieux d'appliquer notre travail sur une application.

Comme présenté sur la figure qui suit l'interface initiale de l'application

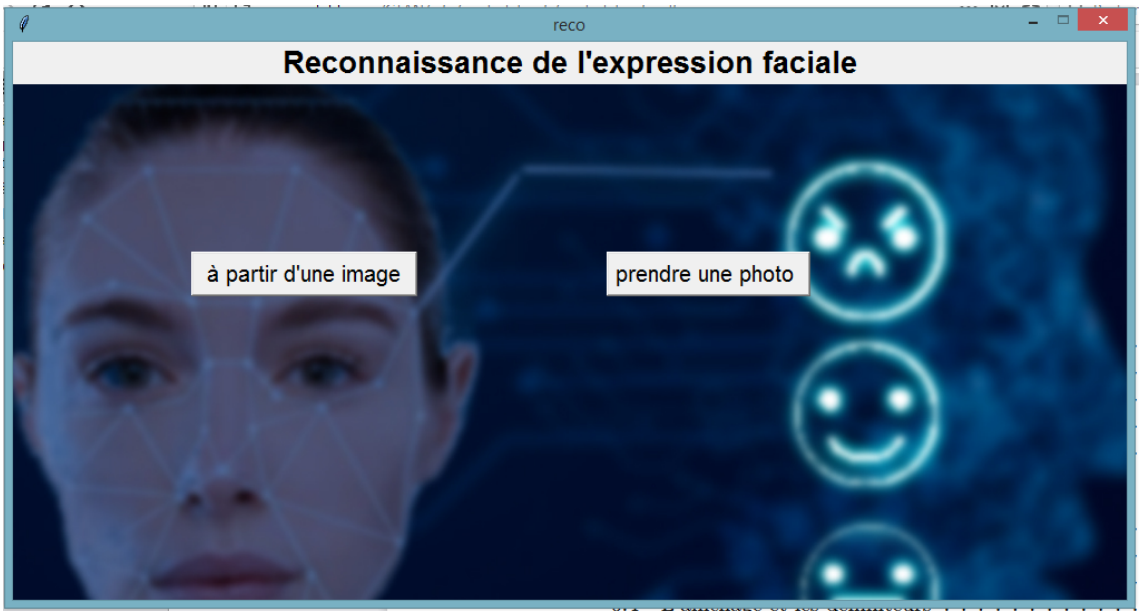


FIGURE 3.13 – Interface initiale.

Dans la figure 3.13 :

- Le bouton "a partire d'une image" permet de lancer l'interface 3.14.
- le bouton "prendre une photo " permet de capturer une image de la camera du pc .

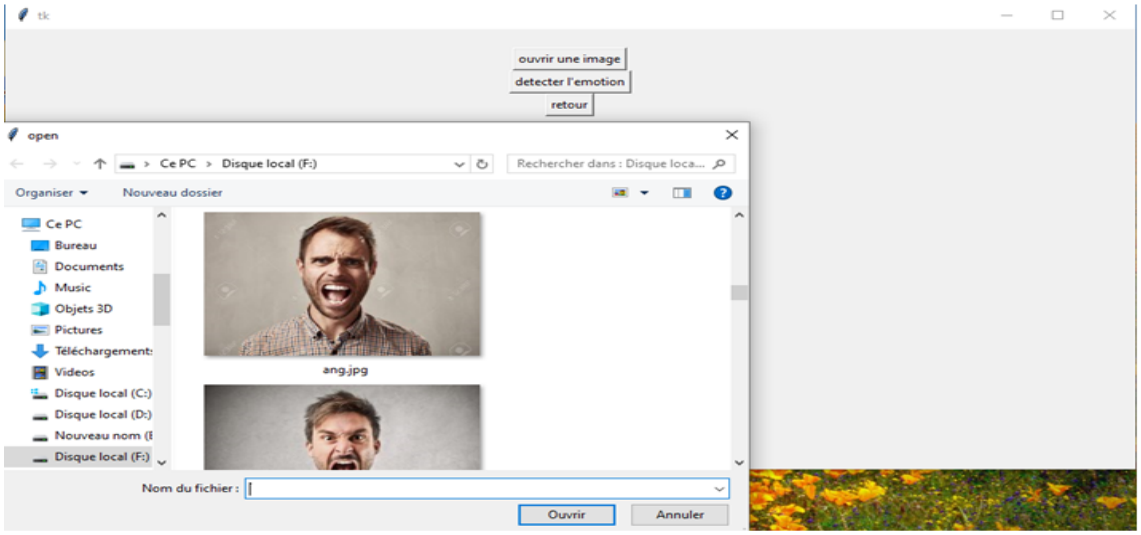
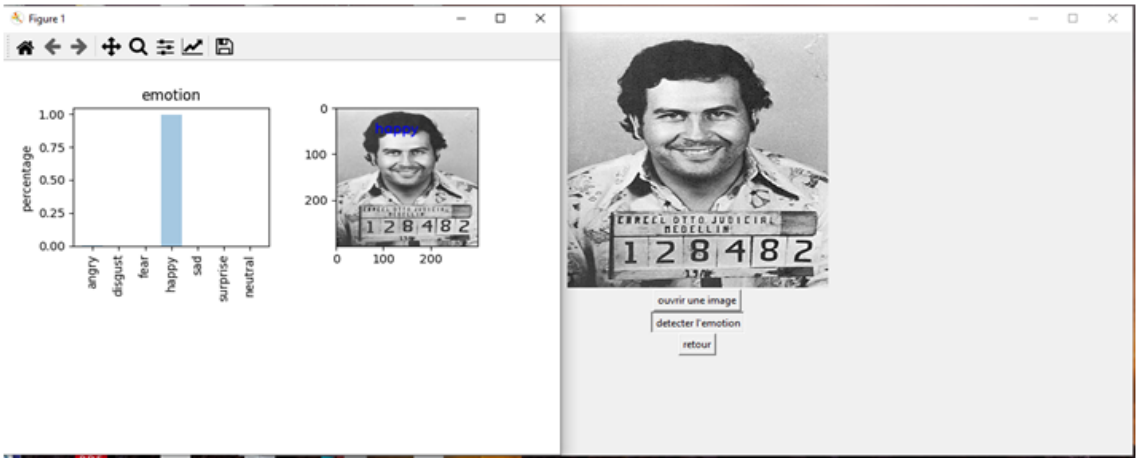


FIGURE 3.14 – Fenetre1.

Pour la figure 3.14 :

- Le button "ouvrir une image " nous permet de choisir une image pour la detection du l'émotion
- Le button "detecter l'emotion" fait la predection de l'émotion sur l'image et ajoute l'émotion detecté au dessous du visage.



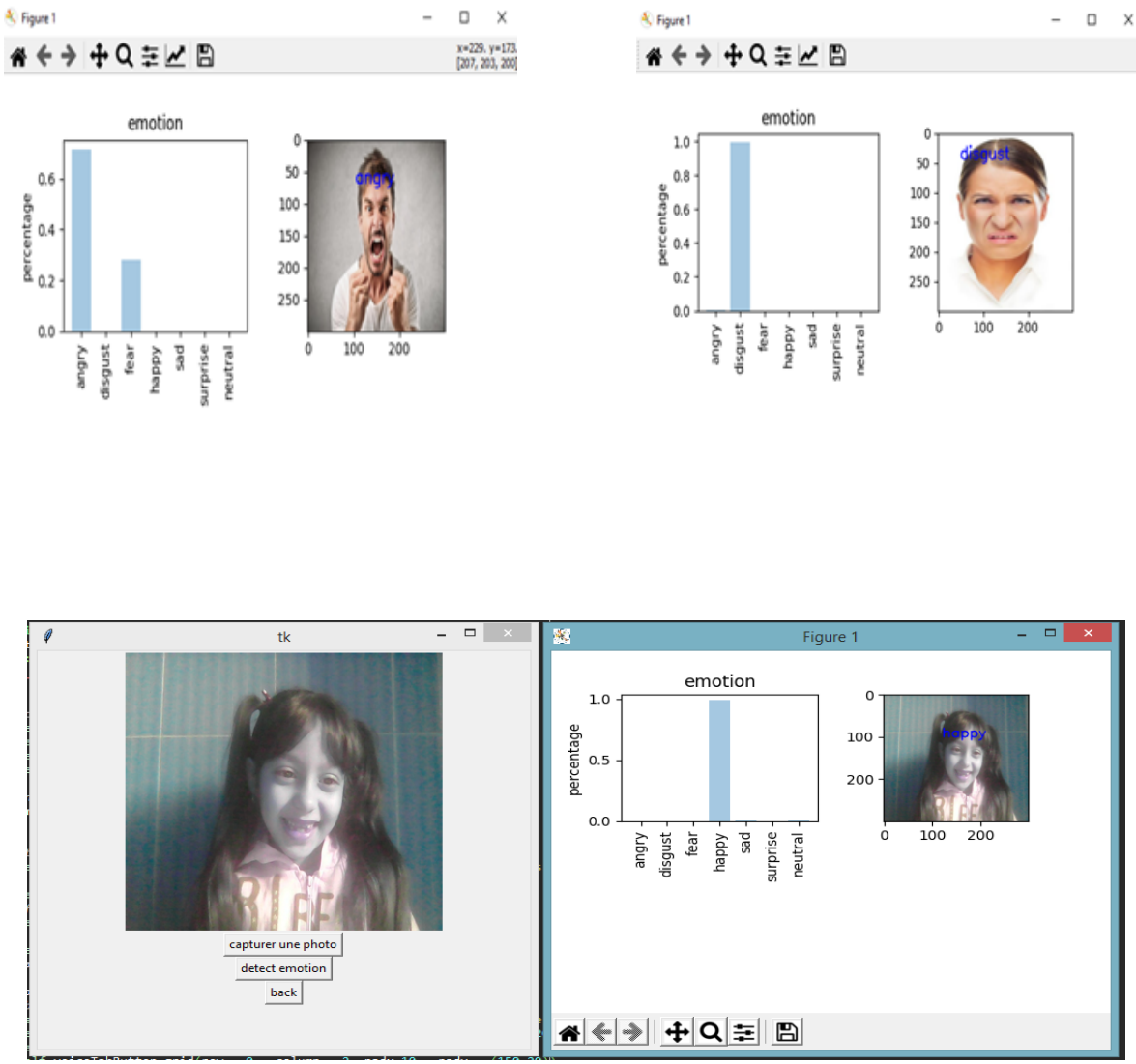


FIGURE 3.15 – La reconnaissance de quelques expressions

3.7 Résultats obtenus et discussion

Un système complet de localisation du visage devrait faire face à certains défis décrits ci-dessous.

3.7.1 Résultats obtenus pour le modèle -A- et -B-

A.Résultats obtenus pour le modèle -A-

A.1. Nombre d'époques = 20

D'après la figure 3.16 La précision de l'apprentissage et de la validation augmente avec le nombre d'époques .De même, l'erreur d'apprentissage et de la validation diminue avec

le nombre d'époque.

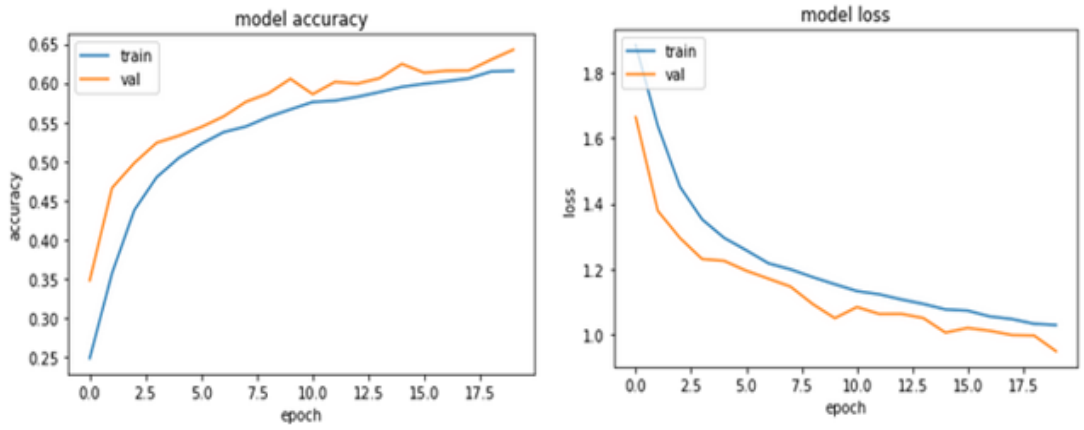


FIGURE 3.16 – Précision et erreur pour le modèle -A- (20 époques).

La matrice de confusion permet d'évaluer la performance du modèle, puisqu'elle reflète les métriques du Vrai positif, Vrai négatif, Faux positif et Faux négatif. La figure 3.17 illustre de près la position de ces métriques pour chaque classe. A titre d'exemple le modèle a bien classé l'emotion angry, happy et surprise, il a mal classé l'emotion disgust et fear.

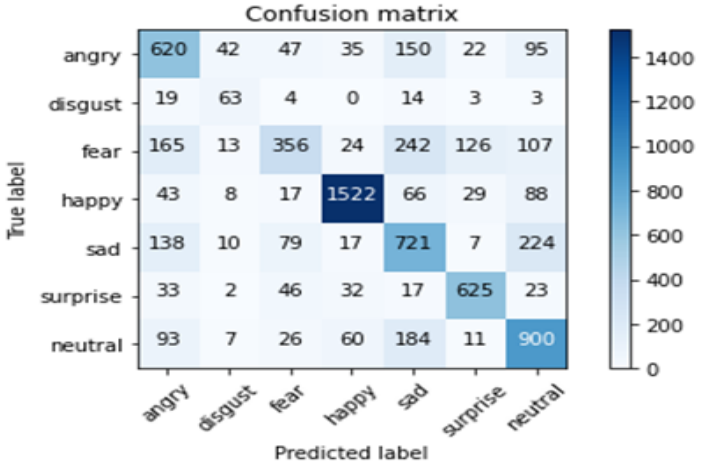


FIGURE 3.17 – Matrice de confusion modele-A-(20epoques)

A.2. Nombre d'époques = 100

D'après la figure 3.18, La précision de validation augmente avec le nombre d'époque jusqu'au quarantiéme epoques ou elle devient presque stable entre [0,67et 0,69]. De même, l'erreur de la validation diminue jusqua 'au quartieme époques ou elle devient stable.

La précision d'apprentissage augmente encore. De même, l'erreur d'apprentissage diminue avec le nombre d'époque.

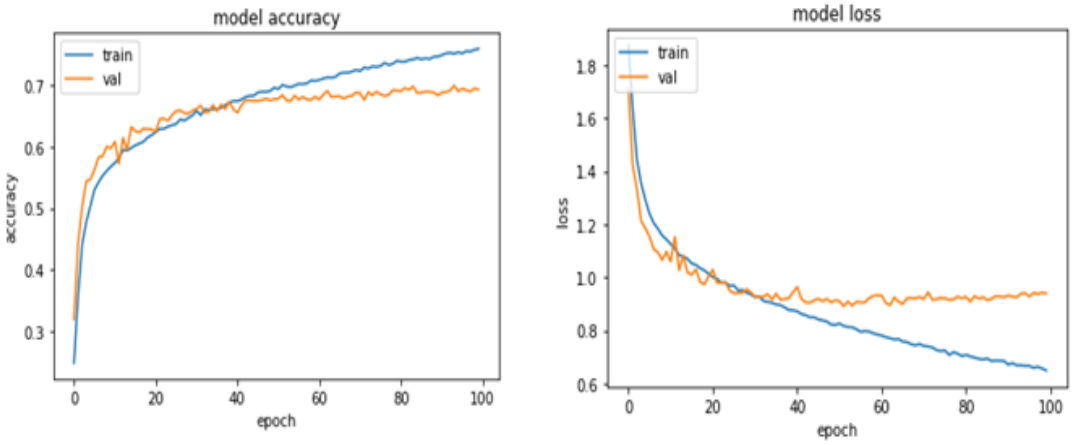


FIGURE 3.18 – Précision et erreur pour le modèle A (100époques)

La figure 3.19 illustre de près la position de ces métriques pour chaque classe. A titre d'exemple le modèle a bien classé presque tous les images.

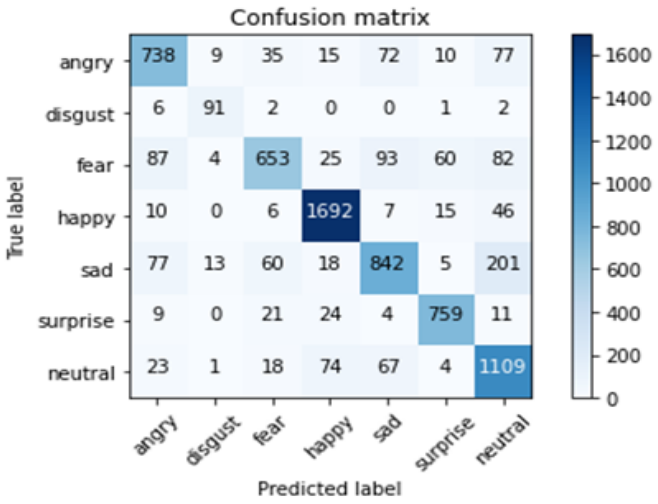


FIGURE 3.19 – Matrice de confusion modele-A-(100epoques)

A.3. Nombre d'époques = 300

D'après la figure 3.20 La précision de l'apprentissage et de la validation augmente dans l'intervalle du nombre d'époques [0-75] et dans l'intervalle [75-300] la validation se stabilise entre (0.69, 0.71) et l'apprentissage continue a augmente.

De même, l'erreur d'apprentissage et de la validation diminue dans l'intervalle du nombre d'époques [0-50], et dans l'intervalle [50-300] l'erreur d'apprentissage continue a se diminue,contrairement a l'erreur de validation qui se stabilse.

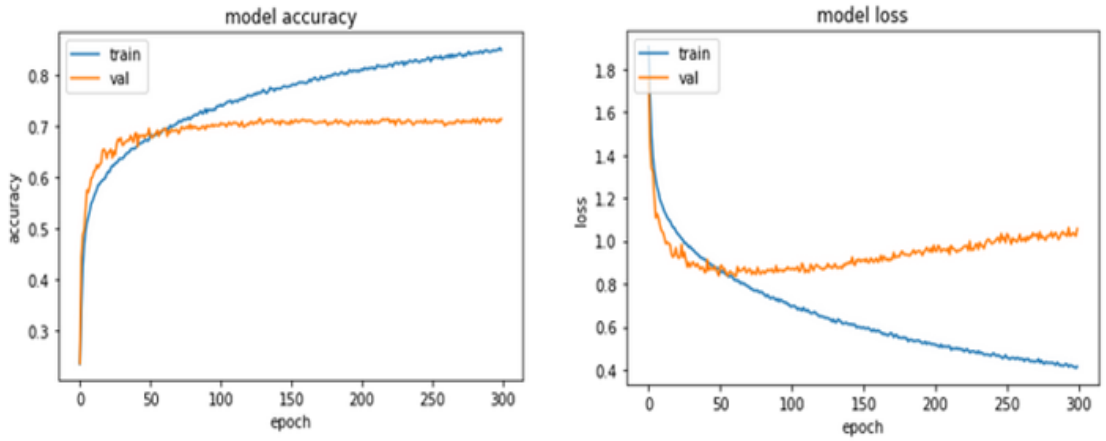


FIGURE 3.20 – Précision et erreur pour le modèle A (300époques)

La figure 3.21Nous montre que le modèle a bien classé presque toutes les images.

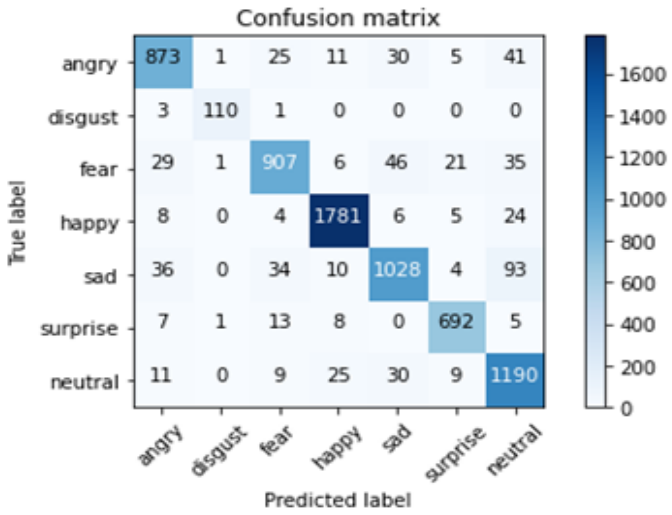


FIGURE 3.21 – Matrice de confusion modele-A-(300epoques)

B.Résultats obtenus pour le modèle -B-

B.1. Nombre d'époques = 20

D'après la figure 3.22 La précision de l'apprentissage et de la validation augmente avec le nombre d'époques .De même, l'erreur d'apprentissage et de la validation diminue avec le nombre d'époque.

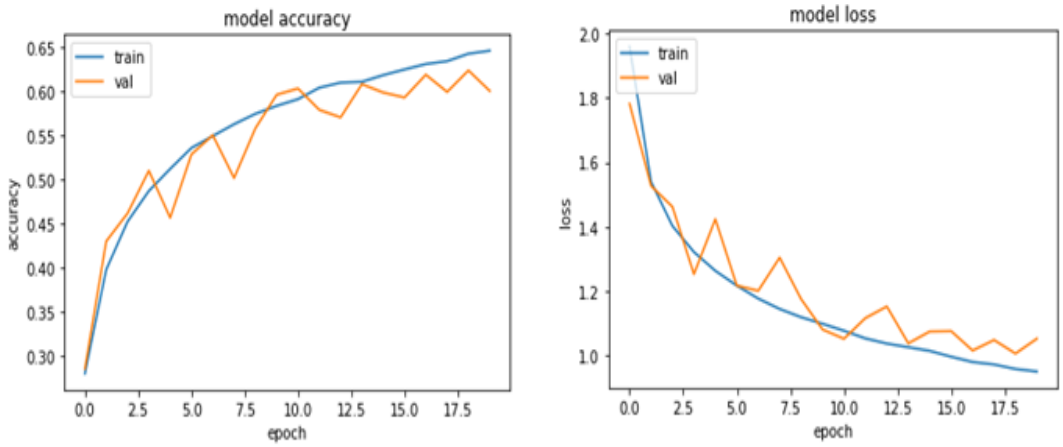


FIGURE 3.22 – Précision et erreur pour le modèle -B- (20 époques).

La figure 3.23 illustre de près la position de ces métriques pour chaque classe. A titre d'exemple le modèle a bien classé les emotions happy, neutral et surprise et il a mal classé les emotions disgust, fear et angry.

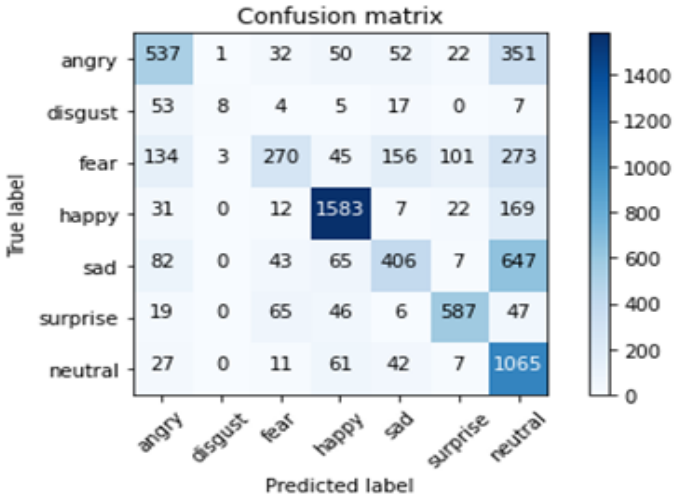


FIGURE 3.23 – Matrice de confusion modele-B-(20epoques)

B.2. Nombre d'époques = 100

D'après la figure 3.24, La précision de validation et d'apprentissage augmente dans l'intervalle du nombre d'époques [0-45] ou elle devient stable l'intervalle du nombre d'époques [45-100] . De même, l'erreur de la validation et d'apprentissage diminue jusqua 'au cinquantieme epoques ou elle devient stable .

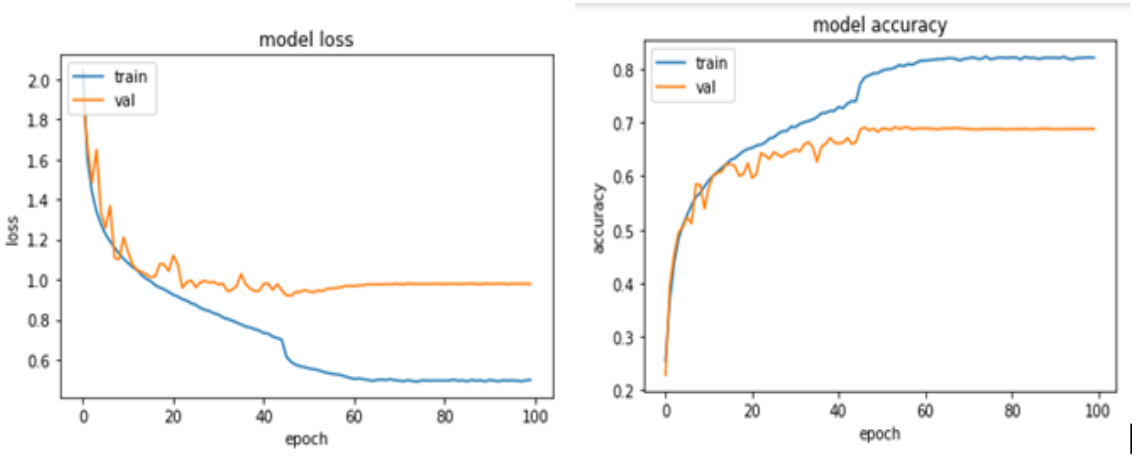


FIGURE 3.24 – Précision et erreur pour le modèle B (100époques)

La figure 3.25 montre que le modèle a bien classé presque toute les emotions .

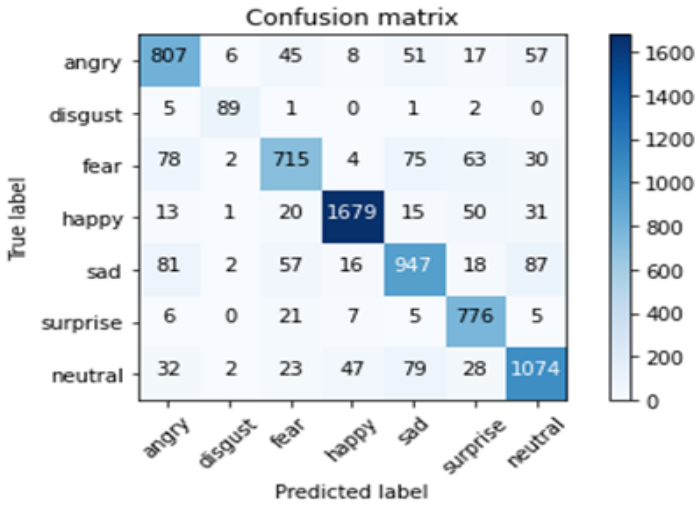


FIGURE 3.25 – Matrice de confusion modele-B-(100epoques)

B.3. Nombre d'époques = 300

La précision de validation et d'apprentissage augmente avec le nombre d'époque jusqu'au cinquantieme epoques ou elle devient stable. Notez que la perte de validation augmente tandis que la perte d'apprentissage diminue. Cela indique que le modèle souffre d'un surajustement -une situation dans laquelle le modèle prédira les données d'entraînement avec une grande précision, mais pas les données de validation.-

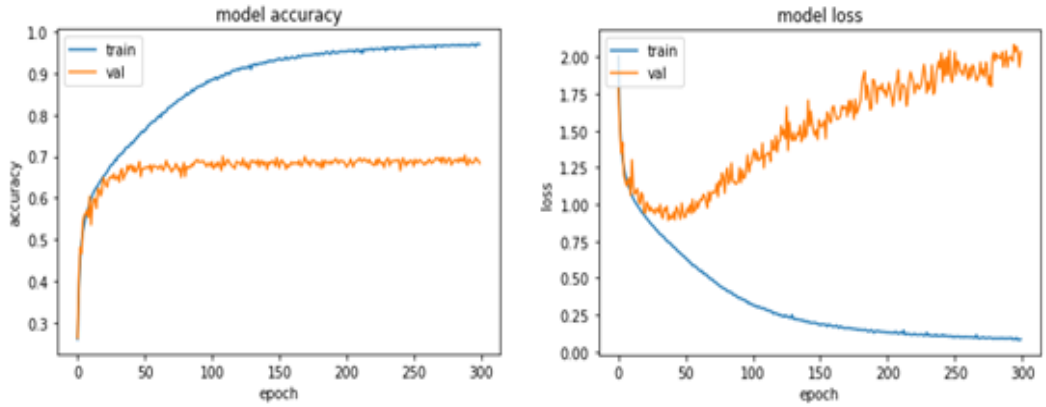


FIGURE 3.26 – Précision et erreur pour le modèle B (300époques)

La figure 3.27 montre que le modèle a bien classé presque toute les emotions .

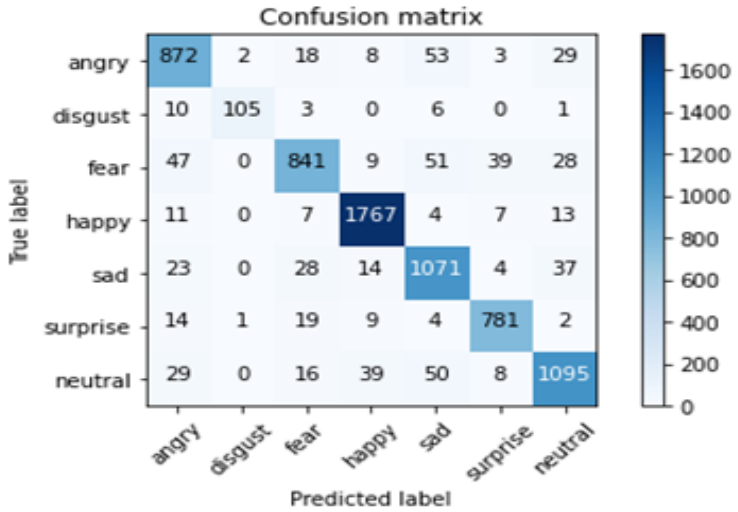


FIGURE 3.27 – Matrice de confusion modele-B-(300epoques)

3.7.2 Discussion

A.Tableau de comparaison des résultats

Modele	Nombre de couches	Nombre de paramètres	Paramètres traitable	Paramètres non traitable	Epoques	précision
A	7	3,470,023	3,465,703	4,320	20	64%
					100	69%
					300	71%
B	50	10,646,583	10,620,071	26,512	20	60%
					100	68%
					300	68%

TABLE 3.1 – Tableau de comparaison des résultats entre Modele -A- et -B-

B. discution 1 :

La performance du modèle -B- obtenue sur la base Fer2013 est relativement faible (68%) (Compte tenu de la taille réduite de l'échantillon d'images de la base) par rapport à la performance du modèle -A-, qui est de 71% sur la même base, bien que le nombre de paramètres traité dans le modèle -A- est beaucoup moins que celui du modele -B-.

C. discution 2 :

Nous remarquons qu'à chaque fois que nous augmentons le nombre d'époques, le taux de précision augmente, nous remarquons aussi que ceci n'est pas proportionnel car arrivé à un certain seuil d'époques, ça commence à se stabiliser et l'augmentation n'est pas aussi importante qu'au début.

D. discution 3 :

Si nous comparons nos deux modèles, nous remarquons que pour 20 époques le modèle -A- donne de meilleurs résultats (une différence de 4%) et quand le nombre d'époques dépasse 20 epoques, nous remarquons que l'augmentation est significative en ce qui concerne le modèle -B- et on enregistre un taux de 8%, le modèle -A- n'a fait qu'5% d'augmentation mais reste encore meilleur que le modele -B-. A 100 époques, le modèle -A- continue à

augmenter en taux de précision et fait un +2%, tandis que le modèle -B- n'as pas augmenter et enfin quand nous comparons les 2 modèles, nous trouvons que le modele -B- a cessé d'augemnter bien avant le modele -A- cela est du a la profondeur du modele -B-.

E. discution 4 :

Étant donné que seul la base de données FER-2013 a été utilisé dans ce cas sans l'utilisation d'autres base de données, une précision de 0,71 est admirable, comme la matrice de confusion (figure 3.21) démontre que le modele a bien classé tous les emotions .Si la base de donnes a fourni une grande quantite de donnees d'apprentissage et tout en conservant la même structure de réseau, l'efficacité du système proposé sera considérablement améliorée.

3.8 Conclusion

Nous avons présenté dans ce chapitre une approche de classification basée sur les réseaux neurones convolutionnels, pour cela, on a utilisé deux modèles avec différentes architectures et nombre d'époques puis nous avons interprété les différents résultats obtenus. Notre système à été testé sur la base de données FER2013 de Kaggle.les expérimentations ont montré que le modèle -A- est plus efficace que le modele -B- en terme de taux de precision.

Conclusion générale

La capacité de l'ordinateur à reconnaître les expressions faciales de la personne constitue un nouveau défi pour la recherche scientifique moderne. D'autant plus que la communication entre humaine et les appareils électroniques a augmenté considérablement, alors que les chercheurs cherchent à développer des programmes intelligents capables de comprendre les expressions faciales en peu de temps.

Ces dernières années, le deep Learning et plus particulièrement les réseaux de neurones convolutionnels (CNN) ont apparu spécialement pour résoudre les problèmes rencontrés du machine Learning. Le CNN est l'une des structures réseau les plus représentatives de la technologie d'apprentissage en profondeur et a connu un grand succès dans le domaine du traitement et de la reconnaissance d'images.

Dans ce travail, nous avons développé un système de reconnaissance des expressions faciales, nommé ECNN, basé principalement sur les modeles VGG et ResNet50. Le système ECCN a été testé sur la base de données Fer2013, et les résultats obtenus ont montrés que le modèle -A- est plus efficace que le modèle -B- en terme de taux de précision.

Comme percpectives, nous proposons de :

- Tester notre modèle sur d'autres bases des données plus volumineuse .
- Ajouter des donnees suplimentaire d'entrainement a la base de donnees fer2013.
- Inclure plus de classes des emotions secondaires pour reconnaitre les micros expressions.

Bibliographie

[1] Jenn-Jier James Lien Automatic Recognition of Facial Expressions Using Hidden Markov Models and Estimation of Expression Intensity CMU-RI-TR-98-31 p13.

[2] J. Colletta et A. Tcherkassof : Les émotions : cognition, langage et développement. Pierre Mardaga, 2003.

[3] Dorian Dozolme. La détection d'une expression faciale incongrue par rapport à un modèle de situationémotionnel : un défi neurocognitif ?. Neurosciences. Université Paris Sud - Paris XI, 2014.

[4] Anne Francou Vedjovsky, Nathalie Milleville Douillet. Perception des émotions faciales chez desadultes présentant une surdité évolutive. Sciences cognitives. 2016.

[5] Antonin Danalet ,Modèles de choix discrets pour la reconnaissance des expressions faciales statiques 2007.

[6] https ://www.eiagroup.fr/domaines-expertise/expressions-faciales-et-micro-expressions/

[7] KHADOUDJA GHANEMLE,Reconnaissance des Expressions Faciales àBase d'Informations Vidéo ; Estimation de l'Intensité des ExpressionsFaciales : Université Mentouri de Constantine 2010

[8] http ://visagetechnologies.com/mpeg-4-face-and-body-animation/

[9] Jörgen Ahlberg, CANDIDE-3 – AN UPDATED PARAMETERISED FACE

[10] Hugo Mercier, Outils informatiques d'analyse des expressions faciales en langue des signes , l'Université Paul Sabatier – Toulouse III 2007.

[11] Khadija Lekdioui. Reconnaissance d'états émotionnels par analyse visuelle du visage et apprentissage machine. Synthèse d'image et réalité virtuelle [cs.GR]. Université Bourgogne Franche-Comté ; Université Ibn Tofail. Faculté des sciences de

Kénitra, 2018.

[12] S.GUERFI « Authentification d'individus par reconnaissance de caractéristiques biométriques liées aux visages 2D/3D » THÈSE Doctorat ,l'Université Evry Val d'Essonne Spécialité : Sciences de l'Ingénieur, octobre 2008

[13] MAHI Abdelhakim, Détection de visage par l'algoritme de boosting, Université Aboubakr belkaid Tlemcen, Juin 2018.

[14] B. Soufiane, « Détection et Identification de Personne par la Méthode Biométrique », Mémoire de Magister en Electronique .Université Mouloud Mammeri de Tizi-Ouzou (UMMTO).

[15] BELHADJ Mahdi ,Etude et simulation d'un système de reconnaissance des expressions faciale.université de Biskra 2019.

[16] NACER Foued ,Reconnaissance d'expression faciale à partir d'un visage reel. Université de 8 Mai 1945 – Guelma.Juillet 2019.

[17] Olivier Ezratty,Les usages de l'intelligence artificielle Novembre 2018.

[18] H. Wang, B. Raj, and E. P. Xing, "On the origin of deeplearning," arXivpreprintarXiv :1702.07800, 2017.

[19] Moualek, D. Y. (2017). Deep Learning pour la classification des images .

[20] https ://zbigatron.com/has-deep-learning-superseded-traditional-computer-visiontechniques/

[21] A. van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in advances in neural information processing systems, pp. 2643–2651, 2013.

[22] R. Collobert and J. Weston, "A unified architecture for natural language processing : Deep neural networks with multi task learning," in Proceedings of the 25th international conference on Machine learning, pp. 160–167, ACM, 2008.

[23] V.BARRIERE, IENAC12L APPROCHES « DEEP LEARNING » APPLIQUEES AUX SIGNAUX AUDIO : PAROLE ET MUSIQUE.

[24] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv :1511.06434.

[25] https ://www.synopsys.com/designware-ip/technical-bulletin/ev-facial-expression-dwtb-q117.html

[26] https ://anhvnn.wordpress.com/2018/02/01/deep-learning-computer-vision-and-convolutional-neural-networks/

[27] https ://meritis.fr/ia/reconnaissance-des-emotions/

[28] Bengio, Y., Laufer, E., Alain, G., &Yosinski, J. (2014, January). Deep generative stochastic networks trainable by backprop. In International Conference on Machine Learning (pp. 226-234).

[29] MoualekDjaloul Youcef. Deep Learning pour la classification desimages. Université Abou BakrBelkaid– Tlemcen 2017.

[30] Ng, A. Y., '& Jordan, M. I. (2002). On discriminative vs. generative classifiers : A comparison of logistic regression and naive bayes. In Advances in neural information processing systems (pp. 841-848).

[31] A. Manuel, L. Pirmin, "Le deeplearning pas à pas", PARTIE I : Concepts, Des labos de R&D à la vie quotidienne. SQLI Digital experence, 2019.

[32] Mohammed Beladgham Khaled Merit Abdelmalik Taleb Ahmed,Improved Facial Expression Recognition Based on DWT Feature for Deep CNN Ridha Ilyas Bendjillali - University, Bechar 2019.

[33] Vikram, K., & Padmavathi, S. (2017). Facial parts detection using Viola Jones algorithm. 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS). doi :10.1109/icaccs.2017.

[34] Xiao Tian, Chao Chen. Modulation Pattern Recognition Based on Resnet50 Neural Network. 2019 2nd IEEE International Conference on Information Communication and Signal Processing.

[35] K. He, X. Zhang, S. Ren et J. Sun. Deep Residual Learning for ImageRecognition. CVPR, 2016.

[36] https ://medium.com/@saicharanars/building-vgg19-with-keras-f516101c24cf consulte le : 7/10/2020.

[37] https ://moov.ai/fr/blog/deep-learning-avec-google-colab/Consultée le : 13/10/2020

[38] http ://inforef.be/swi/python.htmConsultée le : 09 /10/2020.

[39] https ://urlz.fr/e6lvConsultée le : 26/10/2020.

[40] https ://opencv.org/about/Consultée le : 9/10/2020.

[41] https ://urlz.fr/e6mdConsultée le : 26/10/2020.

[42] https ://www.lebigdata.fr/tensorflow-definition-tout-savoirConsultée le : 09 /10/2020.

[43] https ://keras.io/ Consultée le : 9/10/2020.

[44] Pawlik, A., Segal, J., Sharp, H., & Petre, M. (2015). Crowdsourcing Scientific Software Documentation : A Case Study of the NumPy Documentation Project. Computing in Science & Engineering, 17(1), 28–36. doi :10.1109/mcse.2014.93 .

[45] https ://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data.

Résumé

Aujourd'hui, la reconnaissance d'expression faciale s'avère être l'une des applications les plus pertinentes dans de nombreux domaines à savoir : Intéraction homme-machine, médecine, sécurité, education,... etc. En plus, l'approche de deep Learning et plus particulièrement les réseaux de neurones convolutionnels (CNN) ont connu un grand succès dans le domaine du traitement et de la reconnaissance d'images.

Dans ce travail, nous allons développer un système de reconnaissance des expressions faciales (ECNN), basé sur les CNN à savoir : les modèles -A- et -B- inspére de VGGnet et ResNet50. Le système ECCN a été testé sur la base de données la Fer2013.

Mots clés : reseau de neurones , cnn, emotion, expressions faciale,intelligence artificielle . . . .

Abstract

Today, facial expression recognition turns out to be one of the most relevant applications in many areas namely : Human interaction machine, medicine, security, education, etc. In addition, the deep learning approach and more particularly convolutional neural networks (CNN) have been very successful in the field of image processing and recognition.

In this work, we will develop a system for recognizing facial expressions (ECNN), based on CNNs namely : models -A- and -B- inspired by VGGnet and ResNet50. The ECCN system has been tested on the Fer2013 database.

Key words : neural network ,cnn , emotion ,facial expression , artificial intelligence . . . .