REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : siod11/M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Systèmes d'information, Optimisation et Décision (SIOD)**

Reconnaissance multi-dimensionnelle de l'émotion par apprentissage profond de caractéristiques spatio-temporelles sur séquences vidéo

---

Par :

TOUMI ABDERRAHIM

Soutenu le 26/06/2022 devant le jury composé de :

DJEROU LEILA	Professeur	Président
HATTAB DALILA	MAA	Rapporteur
ben taraah laila	MAA	Examinateur

Année universitaire 2021-2022

Remerciements

Tout d'abord, nous devons dire merci à ***Allah*** que j'ai été et que je suis assez bon physiquement et mentalement pour étudier et atteindre cet endroit, paix sur notre ***prophète Mohammed***.

Je tiens également à remercier ma superviseure de projet, ***HATTAB DALILA***, pour m'avoir guidé, donné un coup de main et avoir été une bonne écoute pour moi, sans sa supervision, je ne ferais pas ce projet au maximum.

Enfin, je tiens à remercier ma famille et mes amis qui m'ont soutenu pendant mon séjour ici à l'université. Je voudrais remercier mon père et ma mère, mes frères et mes sœurs, Et je veux remercier mes amis qui ne peuvent pas être que des amis, ce sont mes frères qui étaient mes soutiens à l'université « ***Sofiane*** et ***Mohammed Isaac*** », sans oublier mes fidèles amis « ***Amine***, ***Zouhir***, ***Massoud***, ***Rachid***, ***Bilal***, ***Mohammed Lakhdar, KISRAN Mohammed, karim, amdjed, raafet, didin, mostafa, ishak, imad*** », et bien sûr mon amie.

Table des matières

Liste des tableaux

Chapitre 1

Chapitre 3

LA LISTE DES FIGURES

Chapitre1

Chapitre 2

Chapitre 3

Résumé

L'informatique affective et la reconnaissance des émotions suscitent un intérêt croissant plusieurs domaines de recherche au cours des dernières décennies. En particulier, les traits du visage sont l'un des moyens les plus efficaces d'enregistrer les éléments caractéristiques comportement humain et décrire un état émotionnel.

Algorithmes d'apprentissage profond sont mis en œuvre pour classer les émotions exprimées par le visage en temps réel capturé via une webcam. En effet, Le réseau neuronal convolutif (CNN) est utilisé pour détecter les émotions en temps réel exprimé par le visage en sept émotions différentes obtenues à partir des images traitées.

Abstract

Emotional computing and emotion recognition have attracted increasing interest in several areas of research in recent decades. In particular, facial features are one of the most effective ways to record characteristic elements of human behavior and describe an emotional state.

Deep learning algorithms are implemented to classify the emotions expressed by the face in real time captured via a webcam. Indeed, the convolutive neural network (CNN) is used to detect emotions in real time expressed by the face in seven different emotions obtained from the processed images.

ملخص

أظهرت الحوسبة المؤثرة والتعرف على المشاعر اهتماً ما متزايًدا بالعديد من مجالات البحث على مدى العقود الماضية. و تعبيرات الوجه هي واحدة من أقوى الطرق لتصوير أنماط معينة في السلوك البشري ووصف الحالة العاطفية للإنسان.

يتم تنفيذ الخوارزميات القائمة على التعلم العميق لتصنيف المشاعر التي يعبر عنها الوجه في الوقت الحقيقي التي يتم للكشف في الوقت الحقيقي عن المشاعر (CNN) التقاطها عبر كاميرا الويب. في الواقع ، تُستخدم الشبكة العصبية التالفيفية التي يعبر عنها الوجه في سبعة مشاعر مختلفة تم الحصول عليها من الصور المعالجة.

.

Introduction générale :

Les humains ont inventé le langage et ils en sont fiers. Ils parlent tellement qu'ils ont abandonné les méthodes de communication de base telles que : le toucher, les gestes, le contact visuel et les expressions faciales. Les expressions de base peuvent transmettre plus d'informations, plus rapidement que la parole, mais elles peuvent également exprimer des messages qui ne peuvent pas être communiqués par le langage. Bien que personne ne puisse les voir, les enfants expriment constamment leurs sentiments à travers des expressions faciales et des gestes, et les adultes et leurs visages changent juste pour penser à quelque chose, même si personne ne regarde. Les expressions faciales peuvent être définies comme un signe visible sur le visage qui indique comment une personne se sent, donc les expressions faciales peuvent montrer la joie, la tristesse, la douleur, la fatigue, etc.

L'utilisation d'algorithmes du domaine de la vision artificielle pour reconnaître diverses expressions faciales attire l'attention des spécialistes de ce domaine, ainsi que du grand public : ce type d'opération peut être très utile pour la sécurité, la médecine, les communications, l'éducation, etc. Sur les technologies de vision par ordinateur qui permettent la reconnaissance de certaines expressions faciales afin de détecter d'éventuels cas de fatigue du conducteur.

En raison du besoin croissant d'utilisation des transports et de l'augmentation des accidents de la route pour diverses raisons, notamment la vitesse excessive, la somnolence ou la fatigue, il est préférable d'équiper chaque conducteur du matériel nécessaire pour éviter de tels accidents. Une possibilité est de concevoir des systèmes capables d'alerter le conducteur et de surveiller sa vigilance pour le maintenir éveillé et ainsi réduire le nombre d'accidents de la route, nous proposons donc un système qui s'appuie sur des technologies de vision artificielle pour reconnaître certaines expressions faciales afin de détecter un conducteur potentiel fatigue.

Pour cela, l'un des problématiques rencontrées est la détéction de l'expression faciale du conducteur afin de pouvoir agir rapidement pour ne pas avoir de conséquences grave.

Notre travail consiste à modéliser un système de detection d'expressions faciale dans une vidéo en se basant sur le paradigme du deep learning.

- ## Structure du mémoire :

Nous avons choisi d'articuler notre étude autour de trois chapitres principaux

Le premier chapitre est consacré à un aperçu du Dee p Learning et de l'intelligence artificielle Parce qu'il démontrer la reconnaissance des émotions du point de vue d'une machine et une introduction à réseaux de neurones convolutions.

Chapitre II : Le deuxième chapitre est consacré à l'explication et à l'étude de la méthode des réseaux de neurones convolutifs 3D et de ses relations avec le Deep Learning.

Chapitre III : Il lustre l'implémentation et l'expérimentation de notre système, les outils et logiciels que nous avons eu à utiliser pour le développement, et la réalisation de notre système. Et enfin, nous terminerons ce mémoire par une conclusion générale.

1.1 Introduction :

La reconnaissance des formes est devenue un élément important en raison du besoin croissant d'apprentissage automatique et d'intelligence artificielle dans les problèmes pratiques. La reconnaissance faciale est l'un de ces problèmes qui a été souligné par les nombreux travaux sur les figures publiés par divers auteurs .Ce domaine couvre diverses applications telles que la vérification d'identité, la parenté, l'âge des jumeaux, la chirurgie plastique faciale, ainsi que la détection, le suivi, l'identification de personnes, l'analyse des émotions et la détection de maladies par analyse faciale. Le problème principal de ces investigations reste la manipulation. De données volumineuses. Certaines méthodologies telles que l'apprentissage profond (DL), les réseaux de neurones convolutifs (CNN) et leurs extensions ainsi que pour résoudre le problème de propriété et de classification, Ils sont bien préparés pour ce type d'analyse de données et sont plus proches de l'intelligence humaine.

Dans ce **premier chapitre**, nous donnons un aperçu de l'apprentissage en profondeur et de l'intelligence artificielle, car ils permettent la reconnaissance des émotions du point de vue de la machine, ainsi qu'une introduction aux réseaux de neurones convolutifs.

1.2 Qu'est-ce que L'Intelligence Artificielle

Définir l'intelligence artificielle (IA) n'est pas facile, le domaine est si vaste qu'il est impossible de le restreindre à un domaine de recherche précis ; Il s'agit plutôt d'un programme multidisciplinaire. Si son ambition initiale était d'imiter les processus cognitifs des humains, ses objectifs actuels sont plutôt de développer des automates qui résolvent certains problèmes bien mieux que les humains, par tous les moyens disponibles.

Ainsi l'IA se trouve au carrefour de différentes disciplines : informatique, mathématiques (logique, optimisation, analyse, probabilités, algèbre linéaire), sciences cognitives... sans oublier les connaissances pointues des domaines auxquels on veut l'appliquer. Reposant sur des approches tout aussi variées : analyse sémantique, représentation symbolique, apprentissage statistique ou exploratoire, réseaux de neurones, etc. (1)

L'intelligence artificielle est à l'origine d'avancées importantes sur les plans technologique et commercial, qu'il s'agisse de véhicules autonomes, de diagnostics médicaux ou d'industrie manufacturière de pointe. À mesure que l'intelligence artificielle passe du domaine du virtuel au marché mondial, sa croissance est alimentée par une profusion de données numérisées et une puissance de traitement informatique qui progresse rapidement, avec un effet potentiellement révolutionnaire: en détectant des tendances parmi des milliards de points de données apparemment sans rapport, l'intelligence artificielle peut améliorer les prévisions météorologiques, accroître le rendement des cultures, améliorer la détection du cancer, prévoir une épidémie et améliorer la productivité industrielle (2)

1.3 La reconnaissance des formes

1.3.1 Définition

La reconnaissance des formes est l'une des branches de l'apprentissage automatique qui met l'accent sur la reconnaissance des formes et la régularité des données .Il est également considéré comme le processus de classification des données d' entrée dans certains modèles en fonction des caractéristique fondamentales de, dans certains cas, la reconnaissance des modèles est synonyme d'apprentissage automatique, alors que l'apprentissage automatique ce concentre sur la maximisation des taux de reconnaissance, les algorithmes de reconnaissance de formes fournissent normalement les meilleurs résultats pour leurs données d'entrée. Faites correspondre les entrées en tenant compte de la variation statistique, donc ils ont pu obtenir de meilleurs résultats. (3)

1.3.2 Système de reconnaissance des formes :

Figure 1.1 présente le système de reconnaissance des formes.

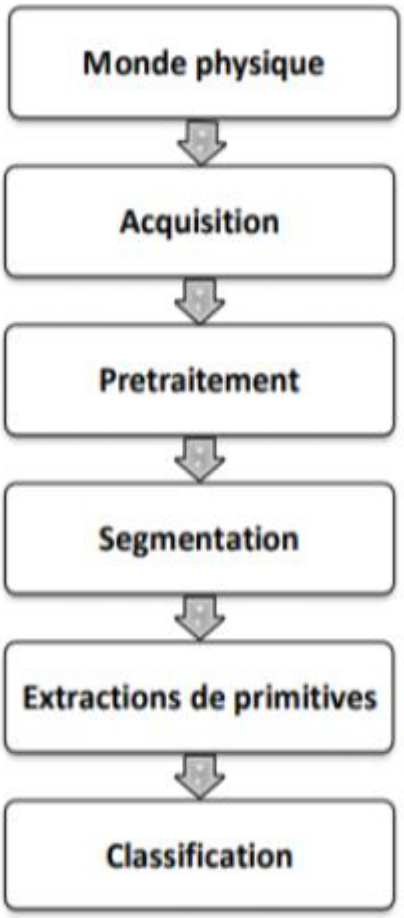


Figure1. 1 .Système de reconnaissance des formes (4)

:

1.4 L'apprentissage automatique

Est une étape clé dans le système de reconnaissance. Il consiste à fournir au système un ensemble de formes connues à priori (on connaît la classe de chacune d'elles). C'est cet ensemble d'apprentissage qui va permettre de régler le système de reconnaissance de façon à ce qu'il soit capable de reconnaître ultérieurement des formes de classes inconnues. On distingue généralement deux types d'apprentissage: apprentissage supervisé et apprentissage non supervisé (4)

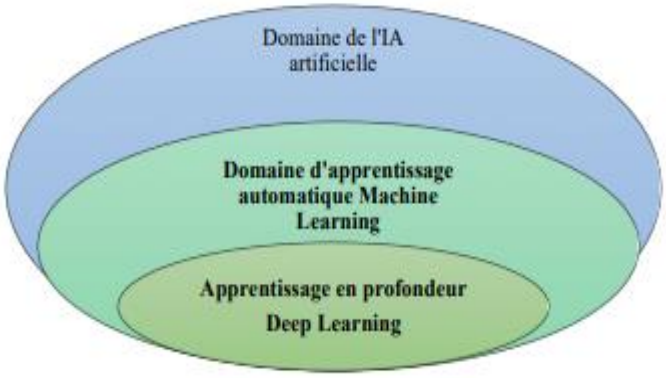


Figure1. 2 : La relation entre l'IA et l'apprentissage profond (4)

1.4.1 Type d'Apprentissage automatique

Les types les plus importants et essentiel pour l'apprentissage sont :

- L'apprentissage supervisé
- Apprentissage non-supervisé
- Apprentissage du renforcement.
- Apprentissage profondeur

❖ ***L'apprentissage supervisé :***

Consiste à fournir au module apprentissage un échantillon représentatif de l'ensemble des formes à reconnaître. Où opérateur de supervision ou professeur, indique étiquette correcte de chaque exemple, qui sera utilisée par le module apprentissage pour identifier la classe dans laquelle opérateur de supervision, souhaite que exemple soit arrangé. Donc la phase apprentissage, a pour objectif analysé les ressemblances entre les formes d'une même famille et les dissemblances entre les formes de Familles différentes pour en déduire les meilleures séparations de l'espace de représentations. Alors objectif général des méthodes d'apprentissage supervisé, est de construire ou approximer à partir de la base d'apprentissage, une règle ou une fonction de classification qui permet à partir de la description d'une forme, affecter la bonne étiquette ou classe à cette forme inconnue par
Le module apprentissage Figure 1.4

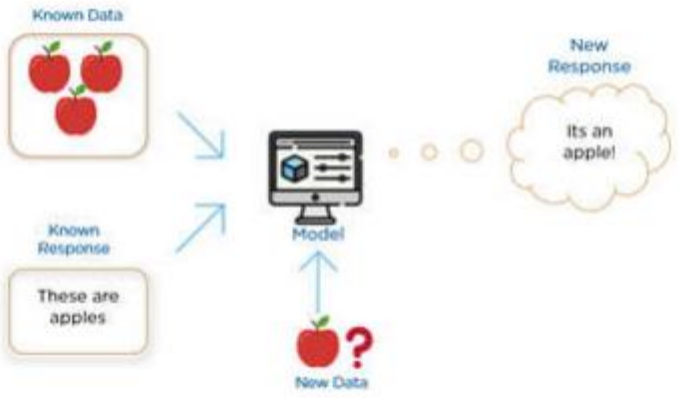


Figure1. 3 Apprentissage supervisé (5)

❖ Apprentissage non-supervisé :

D'autre part, l'apprentissage non supervisé tente de données non étiquetées. Ce type d'apprentissage fait appel à de nombreuses techniques (5) Viser à résumer, expliquer les principales caractéristiques ou déterminer la répartition

D'un ensemble de données. Différentes méthodes incluses dans l'apprentissage non supervisé Y compris le partitionnement des données, le modèle de Markov caché, la réduction dimensions, estimations de distribution, etc.

Tableau 1.1 Différences entre l'Apprentissage Supervisé et non Supervisé

Apprentissage Supervisé	Apprentissage non Supervisé
Données d'entrée sont étiquetées	Données d'entrée sont **non étiquetées**
Utilise le jeu de données d'apprentissage	Utilise tout le jeu de données en entrée
Utilisé pour la prédiction	Utilisé pour l'analyse
Classification et régression	Regroupement, Estimation de la densité et **Réduction de la dimensionnalité**

❖ ***Apprentissage du renforcement :***

Dans l'apprentissage par renforcement, l'objectif est d'apprendre quelles actions effectuer, compte tenu d'un contexte, afin de maximiser une mesure de récompense qui peut à son tour dépendre des actions passées .Au cours de l'apprentissage, des prédictions sont ensuite faites pour prendre des décisions et explorer l'espace de solutions. L'approche implique le compromis exploitation, c'est-à-dire essayer de nouvelles stratégies pour trouver une meilleure solution, quitte à commettre des erreurs, ou maximiser la récompense avec la solution la plus efficace actuellement apprise apprendre des tâches de contrôle, comme marcher ou apprendre à jouer (6)

1.4.2 Méthodes d'apprentissage machine :

1.4.2.1 Généralités :

Le système REF (Reconnaissance d'Expressions Faciales) automatique est conçu pour apprendre à partir de données annotées. Cette méthode particulière de développement d'algorithmes est appelée modèle supervisé dans le domaine de l'apprentissage automatique. Cependant, depuis 2005 et les années suivantes, les émotions du système de reconnaissance sont principalement liées à deux facteurs clés : les bases de données et les algorithmes. Lorsque le premier enregistre des informations visuelles du visage, l'autre vise à modéliser les données sous la forme d'un ensemble de caractéristiques logiques projetées dans un espace multidimensionnel. La représentation nouvellement classée peut être traitée par un classificateur (tel que SVM) ou une couche de neurones densément connectés pour prédire le sentiment final (7)

1.4.2.2 Les machines à vecteur de support(SVM) :

SVM est une méthode de classification binaire à apprentissage supervisé, elle a été introduite par Vapnik en 1995, cette méthode est donc une alternative récente pour la classification. Cette méthode est basée sur l'existence d'un classifier linéaire dans un espace approprié d'un problème de dé classification à deux classes , cette méthode utilise un jeu de données d'apprentissage pour apprendre les paramètres du modèle et est basée sur l'utilisation de ce que l'on appelle des noyaux (kernel) qui permettent une séparation optimale des données Dans la présentation des principes de fonctionnement, nous décrirons les données par des "points" dans un plan (8)

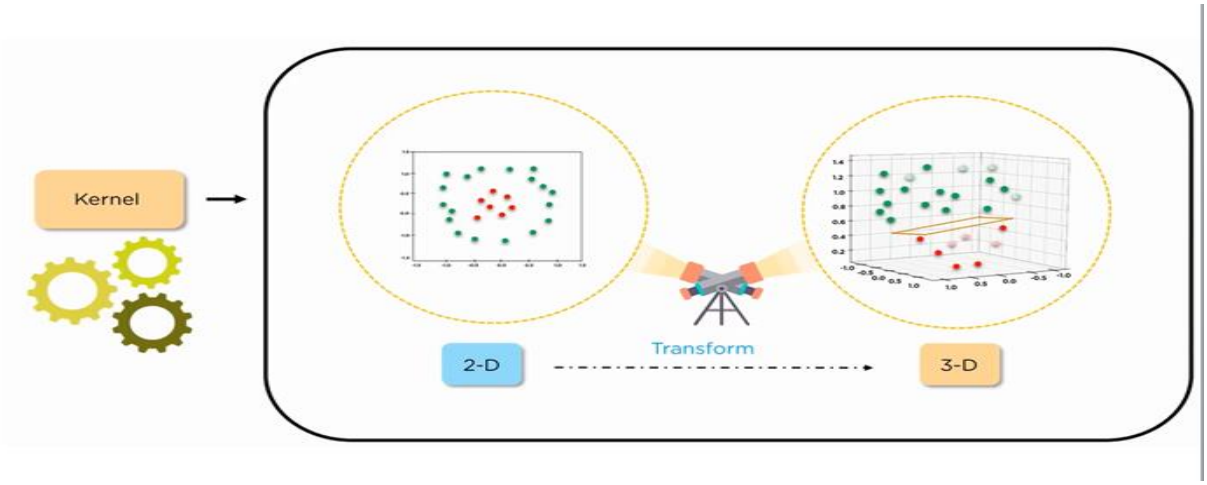


Figure1. 4 Exemple de transforme d'un SVM (8)

1.4.2.3 Les arbres de décision :

Un arbre de décision est un algorithme utiliser dans la classification qui représente un ensemble de décision sous la forme d'un arbre ou chaque nœud final représente une décision et chaque nœud interne représente un test les branche représente le résultat des tests.

Tableau1.2 BD pour jouer au golf :

N°	Ensoleillement	Température	Humidité	Vent	Jouer
1	Soleil	75	70	Oui	Oui
2	Soleil	80	90	Oui	Non
3	Soleil	85	85	Non	Non
4	Soleil	72	95	Non	Non
5	Soleil	69	70	Non	Oui
6	Couvert	72	90	Oui	Oui
7	Couvert	83	78	Non	Oui
8	Couvert	64	65	Oui	Oui
9	Couvert	81	75	Non	Oui
10	Pluie	71	80	Oui	Non
11	Pluie	65	70	Oui	Non
12	Pluie	75	80	Non	Oui
13	Pluie	68	80	Non	Oui
14	Pluie	70	96	Non	Oui

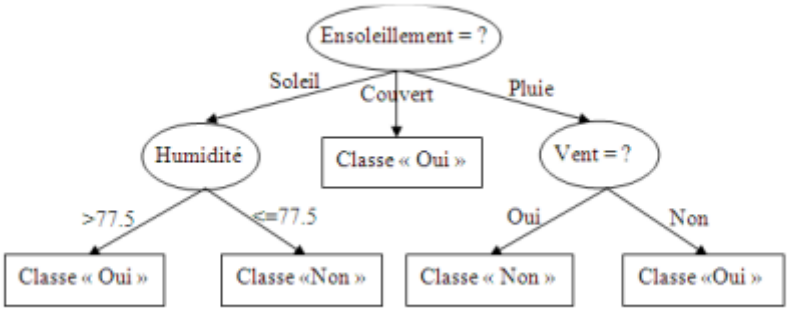


Figure 1.5 Arbre de Décision pour la base de donnes pour jouer au golfe

Avantage:

- Lisibilité
-Capacité a sélectionnée automatiquement les variables.
-Robuste au bruit et aux valeurs manquantes.
-Classification rapide (parcours d'un chemin dans un arbre)

Inconvénients :

-Sensibles au nombre de classes
-Evolutivité dans le temps (le cas où les données évolue dans le temps il faut refaire la phase d'apprentissage)

1.4.2.4 La méthode des k plus proches voisins :

C'est une approche très simple et directe. Il ne nécessite aucune formation mais simplement le stockage des données de formation. Son principe est le suivant. Les données de classe inconnue sont comparées à toutes les données stockées. On choisit pour les nouvelles données la classe majoritaire parmi ses K voisins les plus proches (elle peut donc être lourde pour les grosses bases de données) au sens d'une distance choisie (9)

Quelle distance ?

Afin de trouver les K plus proches d'une donnée à classer, on peut choisir v la distance euclidienne. Soient deux données représentées par deux vecteurs xi et xi, la distance entre ces deux données est donnée par :

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2}$$

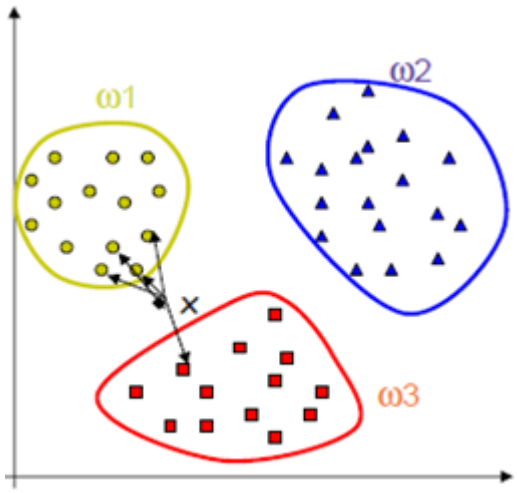


Figure1. 6 Exemple de KNN

1.4.2.5 Méthode statistique

Cette méthode crée un modèle statistique de l'objet à distinguer au lieu de créer un modèle en rassemblant un ensemble de modèles pour former un ensemble de données statistiques, ce qui permet de créer un mécanisme de décision. De cette façon, le modèle est représenté par un ensemble de propriétés extrait de l'élément à distinguer, de sorte qu'il apparaisse comme un point dans un espace multidimensionnel et que le nombre de dimensions corresponde au nombre de propriétés du motif (10)

Un ensemble de motifs forme un seul élément pour donner des informations sur cet élément, il est important de sélectionner ou de créer des propriétés qui permettent aux motifs d'appartenir à différents groupes et d'occuper des régions compressées et non emboîtées dans l'espace. La figure 8 montre un exemple de séparation des bonnes et des mauvaises propriétés.

Après le processus de filtrage, les limites de la résolution sont définies dans espace des propriétés, de sorte que les motifs appartenant à différentes classes soient séparés.

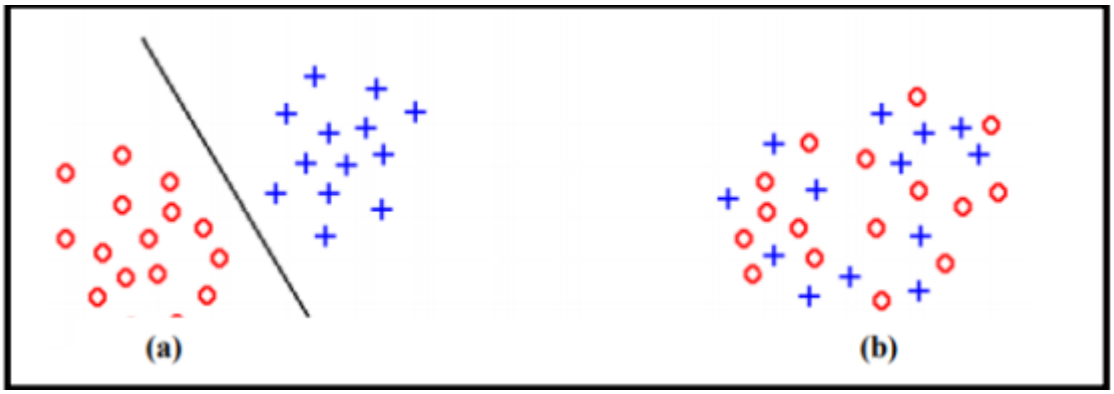


Figure1.7 Séparation des bonnes propriétés et de mauvaises propriétés (10)

1.4.2.6 Les méthodes statistiques :

La régression logistique est définie comme un ajustement La surface de régression des données lorsque la variable dépendante est dichotomique. Cette La technique est utilisée pour vérifier si la variable indépendante eut prédire des variables dépendantes dichotomiques. Différent de la régression multiple et l'analyse discriminante, cette technique ne nécessite pas la distribution normale des prédicteurs ce n'est pas non plus l'homogénéité de la variance. Il existe différents types de régression logistique, il y a

Chacun de leurs processus statistiques conduit à l'élaboration de modèles théoriques différents. Par conséquent, les types directs, séquentiels et automatiques (stepwise) seront discutés. Un exemple explique l'utilisation de cette technologie dans le logiciel SPSS et les procédures d'analyse Les résultats seront expliqués en détail, notamment en ce qui a trait à l'interprétation des rapports de cote (11).

1.5 Reconnaissance des émotions :

Les émotions sont la force puissante qui détermine le comportement humain. Nous les associons souvent au bonheur, à la tristesse, à la colère, à la peur, à la joie, à la haine et à la surprise. Différentes émotions peuvent être identifiées en mesurant les changements dans les expressions faciales et le langage corporel. Ce processus est communément appelé reconnaissance des sentiments ou analyse des sentiments et constitue une caractéristique importante des systèmes d'IA.

L'un des principaux moyens de déduire les émotions des gens est de regarder leurs expressions faciales et c'est quelque chose que nous pouvons mesurer en traitant les images faciales. Les algorithmes derrière l'analyse des sentiments peuvent différer d'une application à l'autre, ils utilisent : des images faciales, pour reconnaître les expressions faciales (REF).

Traditionnellement, les systèmes automatiques REF sont conçus pour apprendre données annotées. Cette méthode particulière pour développer des algorithmes est appelée mode d'apprentissage automatique. Cependant, depuis 2005, Dans les années suivantes, les systèmes de reconnaissance des émotions sont principalement liés à deux facteurs clés : bases de données et algorithmes. Lorsque la première enregistre l'information d'autres

visent à modéliser les données comme un ensemble logique de caractéristiques projetées dans des espaces multidimensionnels (7)

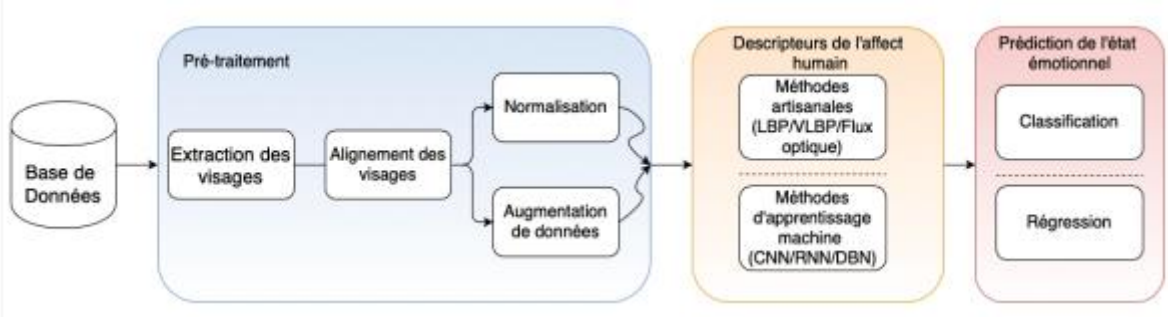


Figure 1.8 Présentation du pipeline sommaire d'un système REF (7)

1.5.1 Pré traitement :

D'une manière générale, la diffusion spécifique des données fait intervenir un certain nombre de phénomènes, qui peuvent être une source importante de bruit et se produire ne convient pas pour détecter les expressions faciales dans les images. Certaines modifications préliminaires de l'image peuvent maximiser le potentiel de l'information émotionnelle des données, normalisant ainsi l'information détectée qui est invalide émotion. Par exemple, en raison de l'existence de différents arrière-plans, éclairages, poses de tête Vers différents environnements et prises de vue. Pour cette raison, la technologie d'alignement de visage Et standardiser. De plus, l'augmentation des données peut être possible dans le cas des modèles d'apprentissage en profondeur. (7)

1.5.2 Descripteurs de l'affect humain

Une fois les données standardisées sur le même champ, Le processus suivant consiste à extraire toutes les fonctionnalités inhérentes à chaque instance une base de données, sous forme de vecteurs de dimension fixe appartenant à un espace de caractéristiques spécifique, dans laquelle ces vecteurs peuvent être projetés. Comme nous avons Comme on peut le voir sur la figure ci-dessus, différentes technologies ont émergé et ont fait des développements incroyables dans l'histoire De l'apprentissage en profondeur marque une incroyable révolution dans la description Image espace-temps. D'une part, les méthodes manuelles (telles que LBP, LBP-TOP et flux optique) s'intéressent à la texture de l'image et fournissent une base pour l'analyse des traits du visage. Mais d'un autre côté, celles-ci laissent peu à peu place à des compétences l'apprentissage en profondeur est utilisé pour détecter automatiquement les descripteurs de visage. (7)

1.5.3 Prédiction de l'émotion

Une fois l'ensemble de données transféré dans un nouvel espace Caractéristiques, la dernière étape consiste à obtenir des prédictions liées à un modèle de représentation émotionnelle spécifique. La méthode courante d'apprentissage automatique est comprise la

classification et la régression des modèles supervisés Aura un modèle de représentation des caractéristiques extraites et "Le cluster ING" en mode non supervisé, signifie que le modèle est issu de la distribution Même les vecteurs de caractéristiques. (7)

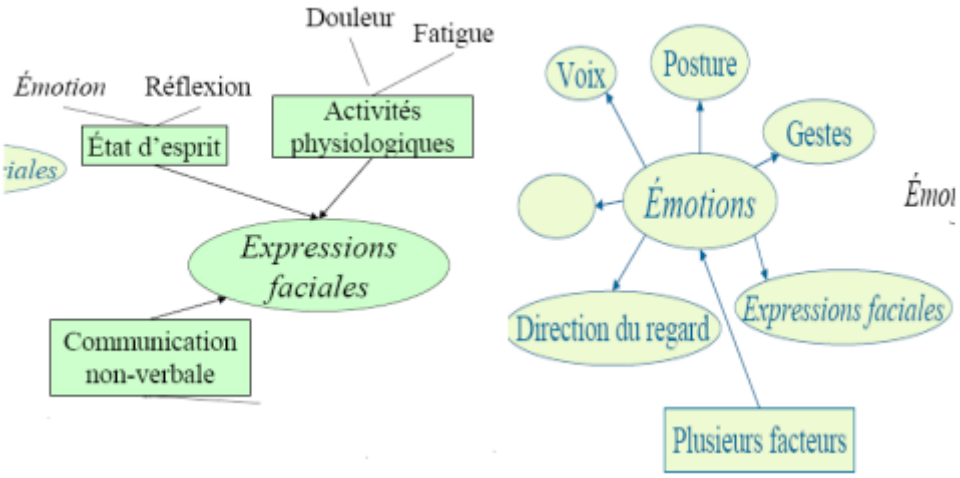


Figure1.9 Générateurs de l'expression faciale et de l'émotion (12)

1.6 Expressions faciales:

Il n'y a rien de plus naturel que d'utiliser la reconnaissance faciale pour une personne. Image les traits du visage sont probablement les traits biologiques les plus courants L'homme procède à une identification personnelle. Utiliser la caméra peut capturez la forme du visage d'un individu et reconnaissez certaines caractéristiques. Selon le système utilisé, l'individu doit être devant l'appareil ou peut être en mouvement Une certaine distance. Comparez ensuite les données biométriques obtenues Vers le fichier de référence. Au début des années 1970, la reconnaissance faciale était Principalement basé sur des attributs faciaux mesurables, tels que la distance entre les yeux, sourcils, lèvres, position du menton, forme, etc. Depuis les années 1990, divers la technologie utilisée met à profit toutes les découvertes dans le domaine thérapeutique Images et réseaux de neurones récents.

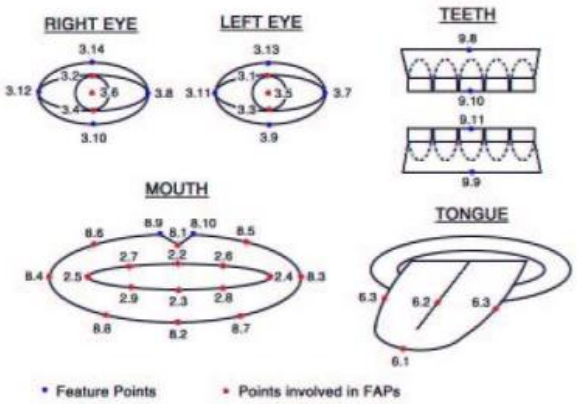


Figure1.10 Modèle du visage MPEG-4 Définition des distances Di (13)

Ces dernières années, les technologies d'apprentissage en profondeur ont été activées

Des progrès rapides dans la reconnaissance faciale, dépassant même les performances

Humain.

1.6.1 Description des six expressions faciales

Les émotions les plus étudiées et utilisées sont : la peur, Colère, joie, tristesse, dégoût, surprise.

Table1.3 Les 6 émotions de base (12)

Émotion	Déclencheurs et circonstances d'apparition	Comportement
Joie	Désir Réussite Bien-être Accomplissement	Approche
Tristesse	Perte Deuil	Repli sur soi
Colère	Obstacle Injustice Dommage Atteindre à son intégrité physique ou psychique Limites de la personne Atteinte au système de valeurs	Attaque
Peur	Menace Danger Inconnu	Fuite Sidération Évitement Parfois attaque
Dégoût	Substance ou personne nuisible Aversion physique ou psychique Contre quelqu'un Reje	
Surprise	Danger immédiat Inconnu Imprévu	Retrait Sursaut

Figure1. 11 Mouvements faciaux globaux (12)

1.6.2 Modèle du circulé pour la représentation de l'émotion

Approche dimensionnelle est une autre approche théorique très populaire en psychologie des émotions humaines qui propose une représentation continue sur plusieurs axes ou dimensions (par contraste aux catégories discrètes des émotions de base)

Trois facteurs ont été utilisés afin de mieux rendre compte des effets psychophysiologiques des différentes émotions : la valence, le degré d'activation physiologique (ou l'arousal) et la dominance (contrôle). En général, deux dimensions principales sont mises en avant (figure 12). D'une part, la valence émotionnelle, c'est-à-dire le caractère positif ou négatif de l'expérience émotionnelle et, d'autre part, la dimension de l'intensité ou le degré d'activation de l'expérience émotionnelle (l'arousal). L'approche dimensionnelle permet de représenter facilement des émotions nuancées mais également des transitions entre différents états émotionnels. (12)

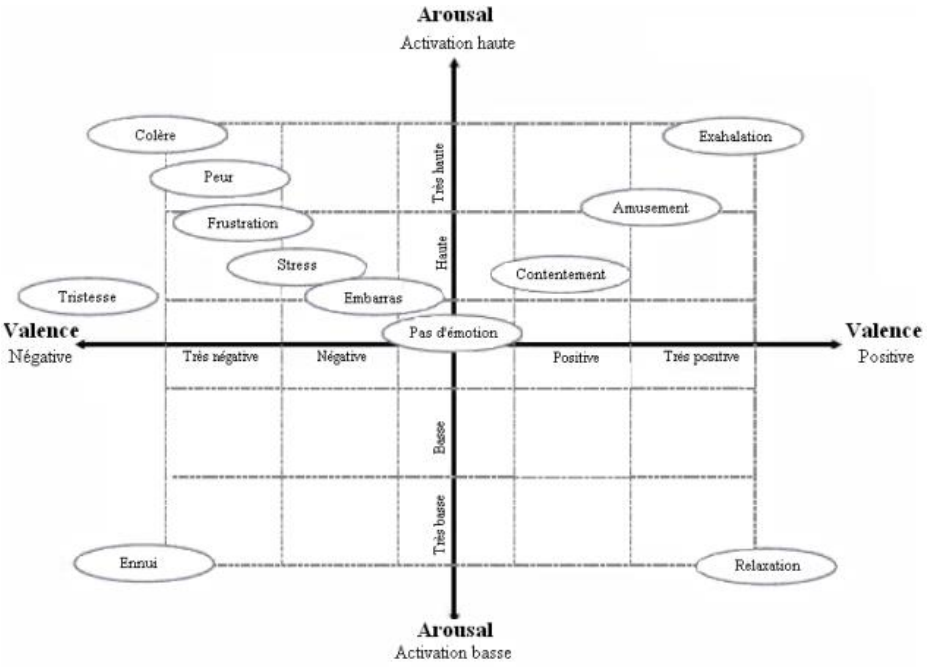


Figure 1.12 La représentation de quelques émotions sur deux axes (12)

En 1980 place les émotions primaires sur les différents secteurs d'un cercle. Dans les encadrés de forme rectangulaire, on trouve les dyades primaires qui correspondent à des émotions secondaires. Elles résultent de la combinaison de deux émotions primaires représentées par des secteurs adjacents sur le cercle .Par exemple, la déception résulte de la tristesse et de la surprise.

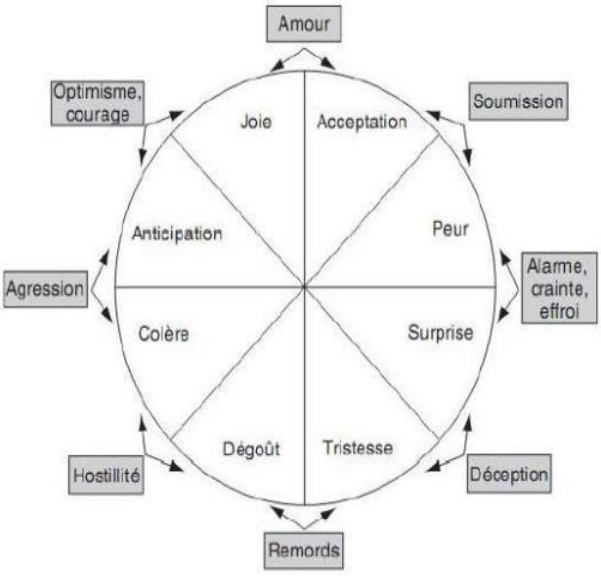


Figure 1.13 La représentation des émotions mixtes (12)

1.7 Apprentissage profond

L'apprentissage en profondeur est une branche de l'apprentissage automatique qui enseigne aux ordinateurs à faire ce qui vient naturellement aux humains : apprendre de l'expérience. Les algorithmes d'apprentissage automatique utilisent des méthodes de calcul

Pour "apprendre" des informations directement à partir de données sans s'appuyer sur une

Équation prédéterminée comme modèle. L'apprentissage profond est particulièrement adapté à la reconnaissance d'image, qui est importante pour résoudre des problèmes tels que la reconnaissance faciale, la détection de mouvement et de nombreuses technologies avancées d'assistance au conducteur telles que conduite autonome, détection de voie, détection de piétons et stationnement autonome (14).

L'apprentissage en profondeur utilise les réseaux de neurones pour apprend Représentations utiles des propriétés directement à partir des données. Réseaux Les neurones combinent plusieurs couches de traitement non linéaire à l'aide d'éléments

Processus parallèle simple inspiré des systèmes biologiques.

Réseaux de neurones convolutions

Les réseaux de neurones convolutions, ou CNN en abrégé, sont le choix populaire des réseaux de neurones pour diverses tâches de vision par ordinateur telles que la reconnaissance d'images. Le nom "convolution" est dérivé d'une opération arithmétique impliquant diverses fonctions de convolution. 4 étapes majeures dans la conception d'un réseau CNN (15).

Convolution : le signal d'entrée est reçu à ce stade.

Sous-échantillonnage: les entrées reçues de la couche de convolution sont

Lissées afin de réduire la sensibilité des filtres au bruit ou à toute autre variation.

Activation: cette couche contrôle la façon dont le signal passe d'une couche à l'autre, de la même manière que les neurones de notre cerveau.

Entièrement connecté: à cette étape, toutes les couches du réseau sont connectées à chaque neurone d'une couche précédente aux neurones de la couche suivante.

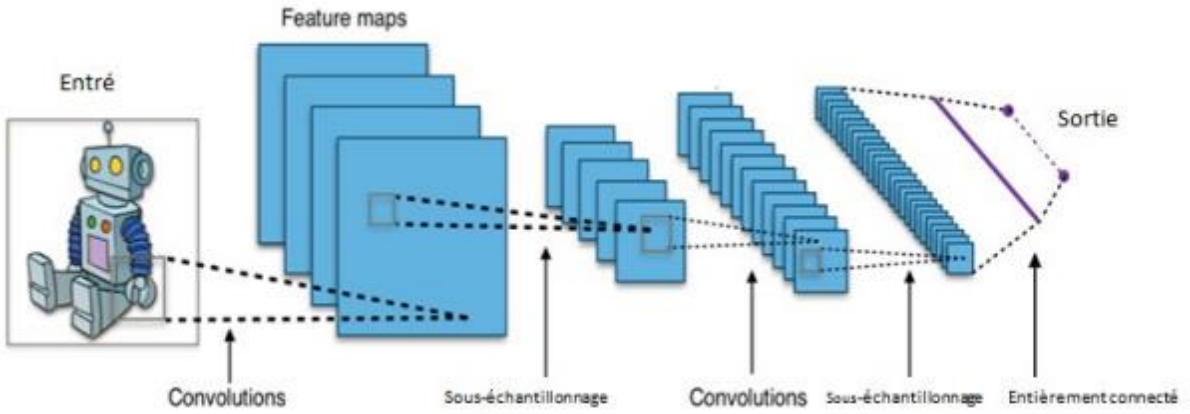


Figure 1.14 Architecture d'un réseau de neurones à convolution (15).

1.8 Conclusion

Dans ce premier chapitre, nous définissons l'apprentissage automatique et les types d'apprentissage, nous définissons l'émotion ainsi que les principaux concepts qui lui sont associés tels que l'expression faciale, les types d'émotions de base, les travaux développés pour la reconnaissance faciale. Le chapitre suivant sera consacré Réseaux de neurones convolutifs et 3D-CNN et Transfert d'apprentissag.

2 Chapiter 2 : CNN pour la reconnaissance d'émotions

2.1 Introduction :

Étant donné que les traits du visage contiennent un grand nombre d'informations d'identification et ne peuvent complètement être représentée par une seule caractéristique, la fusion de multiples caractéristiques est particulièrement importante pour obtenir une performance de reconnaissance du visage robuste, en particulier lorsqu'il existe une grande différence entre les ensembles de test et les ensembles de formation. Cela a été prouvé à la fois dans l'apprentissage traditionnel et approches en profondeur. Ces dernières années, des progrès ont été réalisés dans le traitement et reconnaissance des faces 3D-2D. L'apprentissage en profondeur a connu une croissance rapide en raison de l'émergence de jeux de données de visage à grande échelle. L'apprentissage puissant des données a entraîné une poussée de la recherche en reconnaissance faciale 2D. Il peut effectivement résoudre les problèmes des algorithmes traditionnels d'apprentissage automatique. Notre travail se concentre sur l'application de technologie d'apprentissage en profondeur à la pointe de la technologie en reconnaissance du visage.

Après avoir expliqué les concepts de base, nous passons à expliquer comment reconnaître les sentiments par 3D-CNN, mais avant cela, il y a beaucoup d'étapes.

2.2 L'expression faciale

Une expression faciale est une manifestation visible d'un visage de l'état d'esprit (émotion, réflexion), l'activité cognitive, physiologique (fatigue, douleur), personnalité et psychopathologie d'une personne. Elle repose sur trois principales caractéristiques influençant sur la nature de l'expression faciale à savoir :

La bouche, les yeux et les sourcils.

2.2.1 Paramètres de forme

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

- Hauteur de la tête,
- Position verticale des sourcils,
- Position verticale des yeux,
- Largeur des yeux,
- Hauteur des yeux,
- Distance de séparation des yeux,
- Profondeur des joues,
- Profondeur du nez,
- Position verticale du nez,
- Degré de courbure du nez (s'il pointe vers le haut ou non),
- Position verticale de la bouche,

- Largeur de la bouche (16)

2.3 Réseaux de neurones

Les réseaux de neurones sont un type d'algorithme assez complexe utilisé pour résoudre des Problèmes non soumis à des lois fixes. Ils simulent la façon dont le cerveau humain reconnaît les sons, la parole et les images via un processus énorme et distribué, constitué de simples unités de traitement appelées neurones. Qui a une caractéristique neurologique car il stocke les connaissances scientifiques et les informations expérimentales et les met à la disposition de utilisateur en ajustant les poids (17).

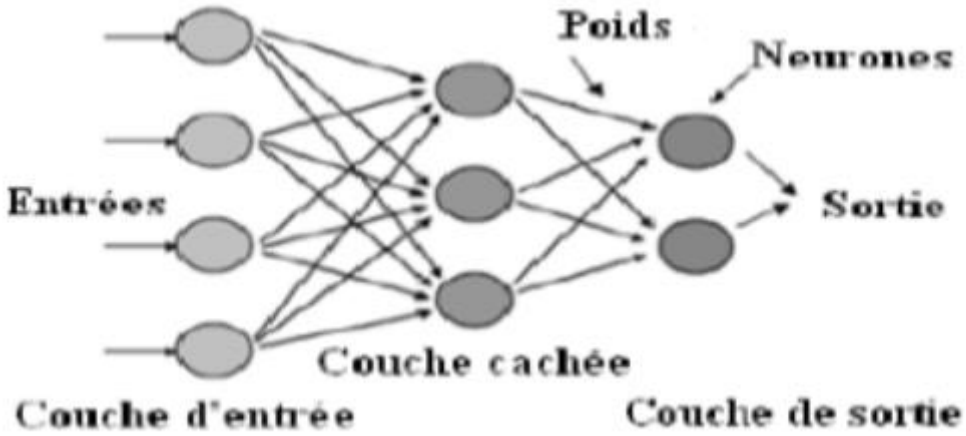


Figure 2 1Composants de réseau de neurone artificiel (17)

2.4 Réseaux de Neurones Convolutifs

Le réseau de neurones convolutifs (CNN) est un sous-type de réseaux de neurones artificiels (ANN). L'innovation des réseaux de neurones convolutifs est la capacité d'apprendre automatiquement un grand nombre de filtres en parallèle spécifiques à un ensemble de données apprises sous les contraintes d'un spécifique. Méthode prédictive de problème de modélisation, telle que la classification d'images. Ils sont constitués de neurones avec des poids et des biais apprenables. Chaque neurone spécifique reçoit de nombreuses entrées, puis prend une somme pondérée sur eux, où il le passe à travers une fonction d'activation et répond avec une sortie (18).

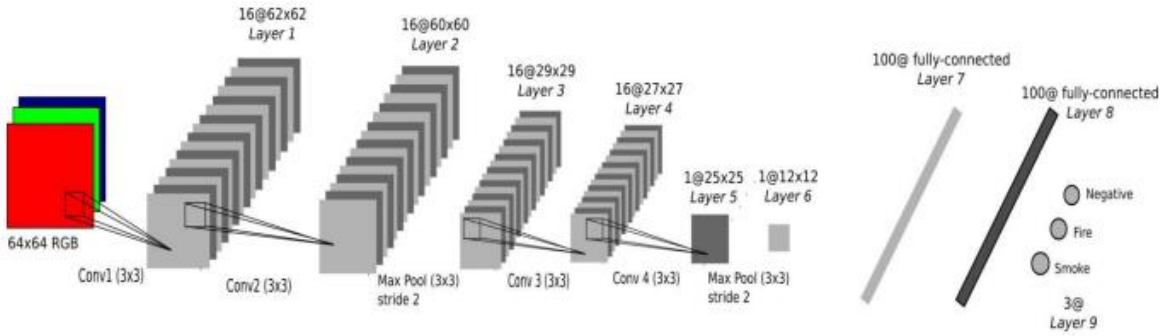


Figure 2 2. Architecture de notre réseau convolutionnel (19)

2.4.1 Types de couches dans le réseau neuronal convolutif :

L'architecture CNN comprend plusieurs types de couches. Une architecture typique consiste à itérer à partir d'une pile de plusieurs couches d'enveloppement et d'une couche de regroupement, suivies d'une ou plusieurs couches entièrement connectées.

A. Couche convolutive (Convolutional layer CONV) :

La couche de convolution a des noyaux (filtres) et chaque noyau a une largeur, une profondeur et une hauteur. Cette couche produit les cartes de caractéristiques à la suite du calcul du produit scalaire entre les noyaux et les régions locales de l'image. Leur tâche est d'extraire les informations pertinentes de l'image (caractéristiques) par une opération de convolution. Cette opération consiste à faire glisser une série de filtres sur une image. Le poids de ces filtres est mis à jour lors de l'apprentissage et c'est grâce à eux que le réseau a pu pour reconnaître les images plus tard (18).

B. Couche d'unité linéaire rectifiée (Rectified Linear Unit layer ReLU) :

La couche d'unité linéaire rectifiée (ReLU) est une fonction d'activation qui est utilisée sur tous les éléments du volume pour éliminer toutes les valeurs négatives et maintenir les valeurs positives dans le but d'introduire une complexité non linéaire dans le réseau (18).

C. Couche de Pooling :

L'étape de regroupement est une technique de sous-échantillonnage. Habituellement, une couche de regroupement est insérée uniformément entre les couches de correction et de convolution. En réduisant la taille des cartes d'entités, donc le nombre de paramètres réseau, cela accélère le calcul du temps et réduit le risque de surapprentissage.

Il y a de nombreuses opérations dans cette couche, par exemple Max pooling et Average pooling.

- **Mise en commun maximale (Max-pooling) :** Il s'agit d'une opération de pooling qui ne prélèvera que le maximum d'un pool. Cela se fait en fait avec l'utilisation de filtres glissant à

travers l'entrée et à chaque foulée, le paramètre maximum est retiré et le reste est abandonné. Cela sous-échantillonne en fait le réseau.

- **Mise en commun moyenne (Average pooling**) : Est une opération de regroupement qui calcule la valeur moyenne d'un pool et l'utilise pour créer un sous-échantillonnage (regroupé), elle extrait les fonctionnalités plus facilement que Max Pooling. La figure 2.3 montre un exemple d'opération de max pooling et Average pooling avec une taille de filtre 2x2 pixels à partir d'une entrée de pixels 4x4 (18).

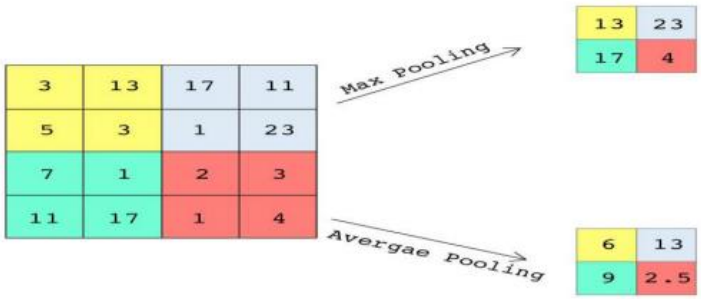


Figure 2 3Exemple de fonctionnement de Max pooling et Average pooling (20)

D. Couche entièrement connectée (Fully-connected layer FC)

Cette couche est située en bout de réseau, elle permet de classer l'image par les caractéristiques extraites de la succession de blocs de traitement, elle est complètement connexe, car toutes les entrées de la couche sont reliées aux neurones de sortie de cette couche. Ils ont accès à toutes les informations d'entrée Chaque neurone attribue à l'image une valeur de probabilité d'appartenir à la classe i parmi les classes C possibles Chaque probabilité est calculée à l'aide de la fonction "softmax" dans le cas des classeselles sont exclusivement réciproques.(18)

E. Couche de normalisation par lots (Batch normalisations layer BN) :

La normalisation par lots est une technique d'apprentissage des réseaux de neurones profonds pour améliorer la vitesse, les performances et la stabilité des réseaux de neurones artificiels. Il est utilisé pour normaliser le volume d'entrée avant de le passer au niveau suivant du réseau (18).

F. Couche d'abandon (Dropout layer DO) :

Est une méthode pour réduire le surajustement et améliorer l'erreur de généralisation dans les réseaux neuraux profonds dont il déconnecte aléatoirement les entrées de la couche précédente avec une probabilité p, ce qui permet d'éviter le sur-apprentissage.

En particulier, les couches CONV, FC et BN effectuent des transformations qui sont fonction non seulement des activations dans le volume d'entrée, mais aussi des paramètres (les poids et biais des

neurones), d'autre part, Les couches RELU/POOL implémenteront une fonction fixe. Notez que certaines couches contiennent des paramètres et d'autres non (18).

2.4.2 Les Architectures de CNN :

De nos jours, les CNN sont considérés comme les algorithmes les plus largement utilisés parmi les inspirés des techniques d'Intelligence Artificielle (IA). L'histoire de CNN commence par les neurobiologiques expériences menées par Hubel et Wiesel (1959, 1962) (21).

Leur travail a fourni une plate-forme pour de nombreux modèles cognitifs, et CNN a remplacé presque tous ceux-ci. Au fil des décennies, différents efforts ont été menés pour améliorer les performances des CNN.

Il existe de nombreuses architectures CNN réputées. Les architectures CNN les plus populaires sont données dans la figure 2.4

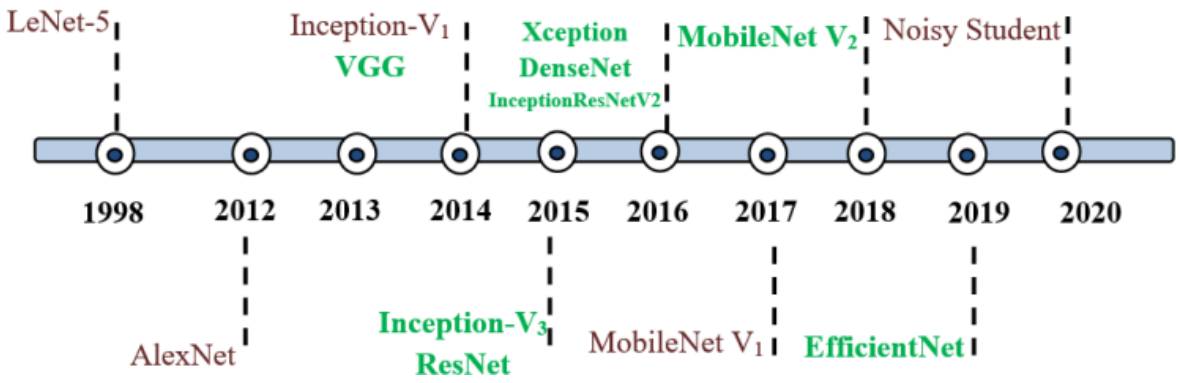


Figure 2 4 Histoire évolutive des CNNs montrant les innovations architecturales (21)

A. LeNet-5 :

LeNet était le réseau neuronal convolutif le plus archétype développé par Yann LeCun en 1990 (22) et amélioré plus tard en 1998 (23) .L'architecture LeNet la plus efficace et la plus connue et elle a été utilisé pour lire les codes postaux, les chiffres, etc.

Cette architecture contient 4 couches convolutive (CONV) et mise en commun (Pooling) alternées, suivies de 3 couches entièrement connectées (fully-connected). LeNet était la première architecture CNN, qui non seulement réduit le nombre de paramètres mais a pu apprendre les caractéristiques de pixels bruts automatiquement.

B. AlexNet :

La première architecture CNN célèbre est AlexNet, qui popularise le réseau de neurones convolutifs en vision par ordinateur (computer vision), développé par Alex Krizhevsky, Ilya Sutskever et Geoff Hinton (24). Plus tard, en 2012, AlexNet a été présenté au défi ImageNet ILSVRC et il a considérablement dépassé les performances du deuxième finaliste.

AlexNet contient 5 couches convolutives avec des unités linéaires rectifiées (ReLU) comme fonctions d'activation, 3 couches Max Pooling et 3 couches entièrement connectées (FC).

C. VGG :

L'utilisation réussie des CNN dans les tâches de reconnaissance d'image a accéléré la recherche en conception architecturale. À cet égard, Simonyan et al ont proposé un principe de conception simple et efficace pour les architectures CNN. Leur architecture, nommée comme le Groupe visuel Géométrie (VGG) de l'université d'Oxford (25). Sa principale réalisation a été de remplacé les filtres 11x11 et 5x5 avec une pile de 3x3 couche de filtres. L'utilisation de filtres de petite taille offre un avantage supplémentaire de faible complexité de calcul en réduisant le nombre de paramètres.

D. GoogleNet :

GoogleNet a été le gagnant du concours 2014-ILSVRC, connu également comme Inception-V1, Il a été développé par une équipe de Google (Christian Szegedy et al). (26)

C'est un type de réseau de neurones convolutif basé des modules Inception, ces blocs encapsulent des filtres de différentes tailles (1x1, 3x3 et 5x5) pour capturer des informations spatiales à différentes échelles, suivi du filtre Concat qui permet de concaténer les résultats des filtres. En outre, la densité de la connexion a été réduite en utilisant la mise en commun moyenne globale à la dernière couche, au lieu d'utiliser une couche entièrement connectée. Ces réglages de paramètres ont provoqué une diminution significative du nombre de paramètres de 60 millions à 4 millions de paramètres.

E. Inception-v3

Inception-V3 est une version améliorée d'Inception-V1 et V2 proposée par Christian Szegedy et al. (27) L'idée d'Inception-V3 était de réduire le coût de calcul des réseaux profonds (Deep networks) sans affecter la généralisation. À cette fin, Szegedy et al. Remplacement des filtres de grande taille (5x5 et 7x7) par des filtres petits et asymétriques (1x7 et 1x5) et la convolution 5 × 5 est transformée en deux opérations de convolution 3 × 3.

F. ResNet

Kaiming He et al. (28) Ont développé un réseau résiduel (ResNet). Cette architecture CNN présente des connexions de saut uniques et une utilisation essentielle de la normalisation par lots (Batch Normalization). Encore une fois, l'architecture n'a pas de couches

entièrement connectées à la fin du réseau. Le principal inconvénient de ce réseau est qu'il est très coûteux à évaluer en raison de la vaste gamme de paramètres. Cependant, jusqu'à présent, ResNet est considéré comme un modèle de réseau neuronal convolutif à la pointe de la technologie et constitue l'option par défaut pour l'utilisation des ConvNets dans la pratique. Il avait été le gagnant de l'ILSVRC 2015.

G. FaceNet :

FaceNet est un système de reconnaissance faciale développé en 2015 par Florian Schroff et al. Chez Google (29). Est un système qui, étant donné une image d'un visage, extraira des caractéristiques de haute qualité du visage et prédira une représentation vectorielle de 128 éléments de ces caractéristiques, appelée incrustation de visage, qui peut ensuite être utilisée pour former un système d'identification de visage.

Le système FaceNet peut être largement utilisé grâce à de multiples implémentations open source tierces du modèle et à la disponibilité de modèles préformés.

H. DenseNet

Le réseau convolutif dense (DenseNet) introduit par Huang et al (30), est une architecture de réseau où chaque couche est directement connectée à toutes les autres couches à l'avance (dans chaque bloc dense). Ce type de connexion est appelé connectivité dense. Pour chaque couche, les cartes d'entités de tous les couches précédentes sont traitées comme des entrées distinctes tandis que ses propres cartes d'entités sont passé en entrée à toutes les couches suivantes.

I. MobileNet-v1

MobileNet est une architecture légère développée par Howard et al (31)de Google, le modèle MobileNet est conçu pour être utilisé dans des applications mobiles, cette architecture utilise des convolutions séparables en profondeur. Il réduit considérablement le nombre de paramètres par rapport au réseau avec des convolutions régulières avec la même profondeur dans les filets. Il en résulte des réseaux de neurones profonds légers. Howard et al introduisent deux hyperparamètres globaux simples qui font un compromis efficace entre latence et précision. Ces hyperparamètres permettent au constructeur de modèles de choisir le modèle de la bonne taille pour leur application sur les contraintes du problème.

Ces paramètres sont le multiplicateur de largeur « α » et le multiplicateur de résolution « ρ ».

J. MobileNet-v2

La deuxième version de MobileNet proposée par Sandler et al. (32) Il est basé sur une structure résiduelle inversée où les connexions résiduelles sont entre les couches de goulot d'étranglement (bottleneck layer). La couche d'expansion intermédiaire utilise des convolutions légères en profondeur pour filtrer les entités en tant que source de non-linéarité.

2.5 Détection et prétraitement du visag :

La détection des visages est la première étape essentielle de la reconnaissance faciale. Il est utilisé pour détecter les visages dans les images. Il fait partie de la détection d'objets et peut être utilisé pour détecter des visages en temps réel à des fins de surveillance et de détection. Suivi de personnes ou d'objets.

L'augmentation des données d'image est une technique qui peut être utilisée pour agrandir artificiellement la taille d'un ensemble de données d'apprentissage en créant des versions modifiées des images dans l'ensemble de données. La formation de modèles de réseaux neuronaux d'apprentissage en profondeur sur plus de données peut entraîner des modèles avec de meilleures performances, et l'augmentation des techniques peut créer des variations dans les images qui peuvent améliorer la capacité d'adapter les modèles pour généraliser ce qu'ils ont appris à de nouvelles images.

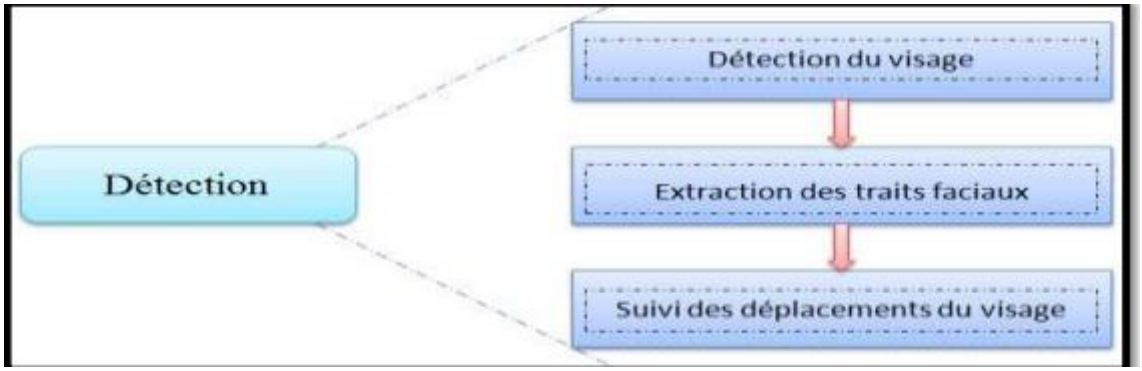


Figure 2 5 Détails de l'étape « Détection »

2.5.1 Méthode de Viola et Jones

Nous nous intéressons particulièrement à décrire de manière brève certains travaux existants à partir des années 2000. La célèbre technique de détection de visage dans la dernière décennie est celle proposée par Paul Viola et Michael Jones en 2001 (33) (34) Cette méthode combine quatre principes clés qui sont les caractéristiques rectangulaires simples appelées des caractéristiques pseudo-Haar en raison de leur similitude avec les ondelettes de Haar, l'approche d'image intégrale pour la détection rapide et efficace des caractéristiques, la méthode d'apprentissage adaptative AdaBoost (35) (publiés par Freund et Schapire en 1996) cherchant à minimiser l'erreur de classification, et l'algorithme en cascade de classifieurs : c'est une chaine qui combine plusieurs classifieurs par étages triés par ordre croissant de complexité.

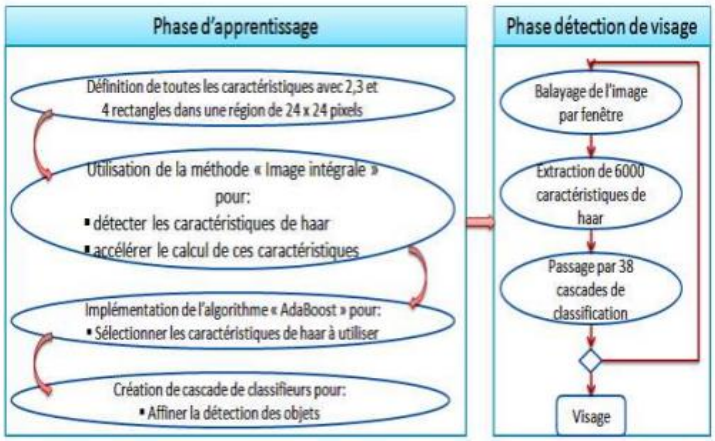


Figure 2 6 La chaine d'exécution de la méthode « Viola&Jones »

La méthode de Viola et Jones nécessite de quelques centaines à plusieurs milliers d'exemples de l'objet que l'on souhaite détecter, d'entraîner un classifieur qui, une fois son apprentissage terminé, sert à détecter la présence éventuelle de l'objet dans une image en naviguant de manière exhaustive, dans toutes les positions et dimensions possibles.Considérée comme l'une des méthodes de détection d'objets les plus importantes, la méthode de Viola et Jones est notamment connue pour avoir introduit plusieurs notions reprises par la suite par de nombreux chercheurs en vision par ordinateur comme la notion d'image intégrale ou la méthode de classement construit comme une cascade de classificateurs améliorés. Cette méthode bénéficie d'une implémentation sous licence BSD dans OpenCV, la bibliothèque utilisée dans notre application (36).

Figure 2 7 Détection de visage avec la méthode de Viola et Jones (36).

2.5.2 Prétraitement vidéo

Toutes les images sont initialement extraites du signal visuel pour les étapes suivantes. Étant donné que ces images extraites contiennent encore des informations redondantes considérables pour la détection des émotions, nous extrayons uniquement les régions faciales comme suit : Toutes les boîtes englobantes contenant des régions faciales dans

chaque image ont été extraites à l'aide de l'algorithme de Viola et Jones et une région faciale a ensuite été détectée (37).

2.6 Extraction des Composants du Visage

2.6.1 Extraction des Yeux et des Sourcils

Afin d'obtenir des contours de qualité suffisante pour être utilisés dans le cadre de la reconnaissance des expressions faciales, les travaux existants souffrent de deux limitations principales: certains proposent une localisation globale approximative de ces caractéristiques en extrayant une boîte qui enferme ces caractéristiques (38), (39)d'autres essaient d'extraire plus précisément les contours mais les modèles choisis sont trop simples et peu réalistes et les algorithmes nécessitent une phase de sélection manuelle de points dans la première image. La méthode proposée s'efforce de pallier ces problèmes.

La zone de recherche de chaque iris est limitée aux parties hautes gauche et droite de la boite englobant le visage. Les dimensions de chaque boite de recherche de l'iris ont été déterminées suite à une phase d'apprentissage. Sur chaque image de la base ORL [ORL base], une boite englobant chaque œil a été sélectionnée à la main et les dimensions respectives de ces boites ont été étudiées. Il a été déduit les relations suivantes entre les dimensions de la boite englobant le visage et celles de la boite englobant chaque œil

$$Hauteur_{visage} = 4 * Hauteur_{oeil} \; ; \; L\arg eur_{visage} = 2.5 * L\arg eur_{oeil}$$

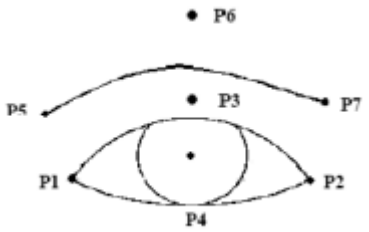


Figure 2 8 Modèle pour l'œil et le sourcil et points caractéristiques Pi

2.6.2 Détection des Lèvres (de la bouche)

Associé à l'algorithme de segmentation des yeux et des sourcils citée, un autre algorithme est utilisé également pour la détection des lèvres.

Plusieurs modèles paramétriques ont déjà été proposés pour modéliser le contour des lèvres. Des auteurs ont proposé de modéliser les lèvres par deux paraboles (40), d'autres ont proposé de modéliser le contour supérieur des lèvres à l'aide de deux paraboles au lieu d'une (41), ou encore d'utiliser des quartiques (42) . Un gain en précision a été obtenu par rapport à la première idée, néanmoins tous ces modèles sont encore limités par leur trop grande rigidité, en particulier dans le cas d'une bouche non symétrique.

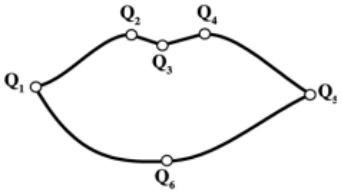


Figure 2 9 Modèle choisi pour la bouche

2.7 Identification en profondeur des visages générés par CNN

Avec l'évaluation exploratoire dans (43), sur deux différents ensembles de données communes on peut soutenir les idées selon lesquelles (i) les auto-encodeurs peuvent être utilisés pour la réduction de dimension et fusion de caractéristiques et que (ii) ces auto-encodeurs peuvent être formés sur des domaines différents (mais sans doute similaires) que ceux du modèle utilisé. Les auteurs (43) affirment que les deux ensembles de données sélectionnés ne sont pas représentatifs pour une évaluation à grande échelle, mais ils fournissent une première approche pour investiguer sur une idée nouvelle dans le traitement des données d'image rares, c'est-à-dire qu'il n'y en a pas assez données pour former un modèle, et des jeux de données d'image, où seules les fonctionnalités pour la reconstruction sont disponibles.

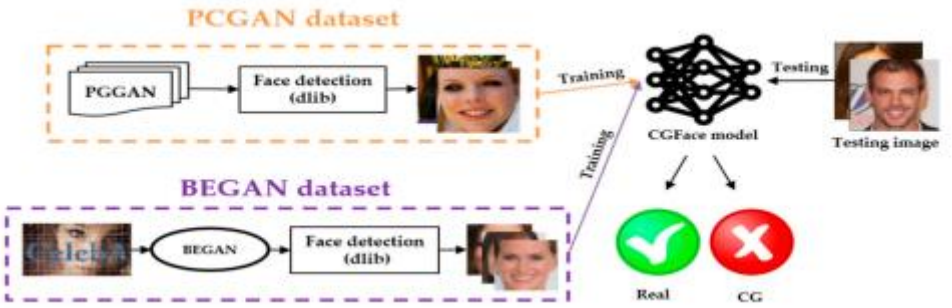


Figure 2 10 Architecture globale de la méthode proposée

La **figure 2. 11** illustre un modèle de détection de visage généré par ordinateur, appelé CGFace, et présente chaque couche avec sa taille d'entrée, de noyau et de sortie correspondante. Globalement, le modèle fonctionne bien, car on maintient un degré de diversité des visages sur toute la gamme des valeurs γ (le rapport entre deux pertes de reconstruction dans le temps). À faible valeur (0,3), les visages générés sont généralement uniformes. Cependant, avec une valeur plus élevée, par exemple 0,7, la variété des visages augmentent, mais les artefacts et le bruit apparaissent également plus fréquemment.

Figure 2 11 Images générées avec différentes valeurs de l'hyper paramétré γ (43)

2.7.1 L'apprentissage par transfert learning :

Conception d'algorithmes d'apprentissage automatique et d'apprentissage profond méthodes traditionnelles de distribution de l'espace des fonctionnalités spécifique. Une fois que la distribution de l'espace des caractéristiques change, le modèle nécessite une refonte à partir de zéro et il est également fastidieux de collecter les données d'apprentissage requises Par conséquent, compte tenu du modèle d'apprentissage en profondeur des données étiquetées suffisantes sont requises pendant la formation. Il est donc presque impossible de créer des modèles basés sur l'apprentissage automatique pour le domaine une cible composée de très peu de données étiquetées pour un apprentissage supervisé. Exister dans ce cas, l'apprentissage par transfert améliorera considérablement Apprendre L'idée principale derrière l'apprentissage par transfert est d'emprunter données ou connaissances étiquetées extraites d'un domaine connexe pour aider les algorithmes d'apprentissage automatique à obtenir de meilleures performances dans le domaine d'intérêt.

La **figure 2.12** montre la comparaison entre les processus d'apprentissage automatique apprentissage traditionnel et apprentissage par transfert. Comme nous sommes dans un l'apprentissage automatique traditionnel, qui essaie d'apprendre chaque tâche séparément avec différents systèmes d'apprentissage, et l'apprentissage par transfert tente d'extraire les connaissances des tâches source précédentes dans les tâches cibles Ce dernier dispose de peu de données étiquetées pour l'apprentissage supervisé.

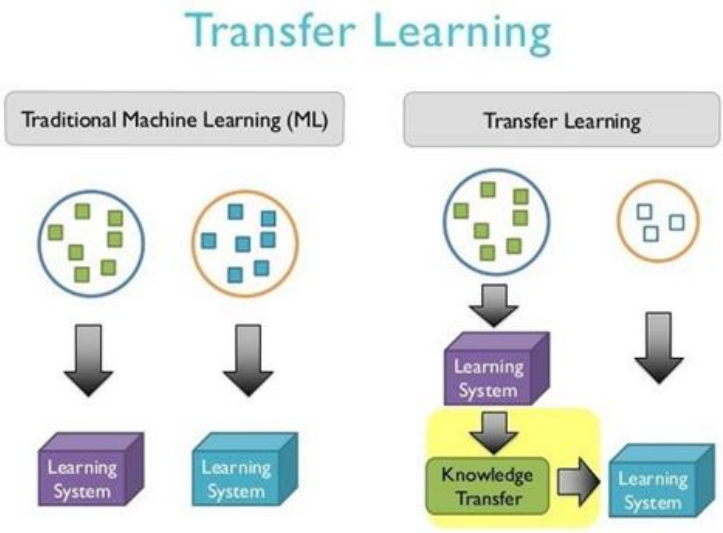


Figure 2 12 Transfer learning

2.8 Architectures de classification des actions

Alors que le développement des architectures de représentation d'images a mûri rapidement ces dernières années, il n'existe toujours pas architecture frontale claire pour la vidéo. Certaines des principales différences dans les architectures vidéo actuelles sont de savoir si Les opérateurs de convolution et de couches utilisent 2D (basé sur l'image) ou Noyaux 3D (basés sur la vidéo); si l'entrée du réseau n'est qu'une vidéo RVB ou inclut également un flux optique précalculé et, dans le cas des ConvNets 2D, comment les informations est propagé à travers les trames, ce qui peut être fait soit en utilisant réseau neuronal récurrent telles que les LSTM, soit agrégation dans le temps (44)

2.8.1 Apprentissage 3D-CNN

Gonfler les ConvNets 2D en 3D. Un certain nombre d'architectures de classification d'images très réussies ont été développées au fil des ans, en partie par des essais. Au lieu de répéter le processus pour les modèles spatio-temporels nous proposons de convertir simplement les modèles de classification d'images (2D) réussis en ConvNets 3D. Cela peut être fait par partir d'une architecture 2D, et gonfler tous les filtres et la mise en commun des noyaux - en les dotant d'une dimension temporelle. Les filtres sont généralement carrés et nous rendez-les cubiques - les filtres N × N deviennent N × N × N (44).

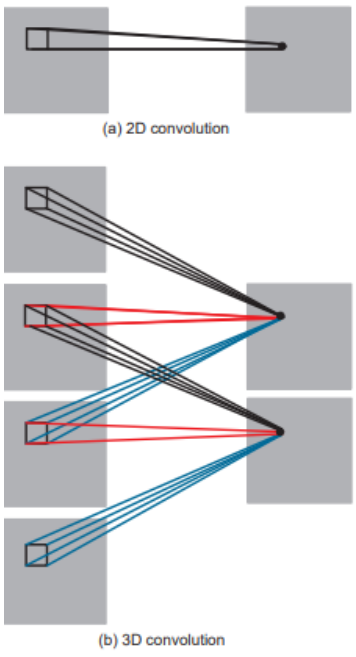


Figure 2 13 Comparaison de 2D et 3D (45)

Figure 2.13 Comparaison des convolutions 2D (a) et 3D (b). En (b) la taille du noyau de convolution dans le temps la dimension est de 3 et les ensembles de connexions sont codés par couleur afin que les poids partagés soient de la même couleur. En 3D convolution, le même noyau 3D est appliqué au chevauchement cubes 3D dans la vidéo d'entrée pour extraire les caractéristiques de mouvement (45)

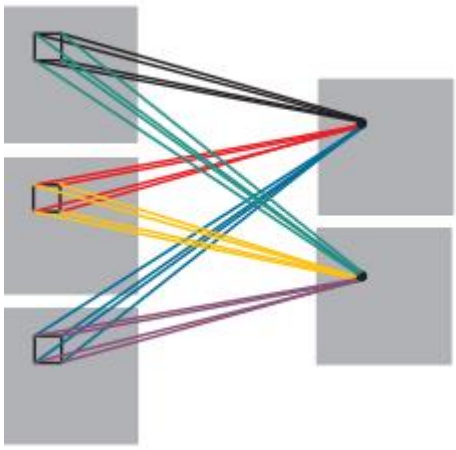


Figure 2 14 3D convolution

Extraction de plusieurs fonctionnalités à partir de contiguës cadres. Plusieurs convolutions 3D peuvent être appliquées à des cadres contigus pour extraire plusieurs caractéristiques. Comme sur la figure 2.13, les ensembles de connexions sont codés par couleur afin que le partage les poids soit de la même couleur. Notez que tous les 6 ensembles des connexions ne partagent pas de poids, ce qui donne deux différentes caractéristiques des cartes sur la droite

Ensemble de cartes d'entités de niveau inférieur. Similaire au cas de convolution 2D, cela peut être réalisé en appliquant plusieurs convolutions 3D avec des noyaux distincts même emplacement dans la couche précédente (Figure 2.14) (45).

2.8.2 Réseau neuronal récurrent (RNN)

RNN est l'une des architectures fondamentales du réseau neuronal à partir de laquelle d'autres architectures d'apprentissage en profondeur sont construites (46).

- il pourrait avoir des connexions qui se reproduisent dans des couches antérieures ou dans la même couche, qui permettent aux RNN de maintenir la mémoire des inputs précédents et des problèmes de modèle à temp.
- Il peut être déroulé dans le temps et entrainé avec une back-propagation standard ou en utilisant une back-propagation dans le temps.

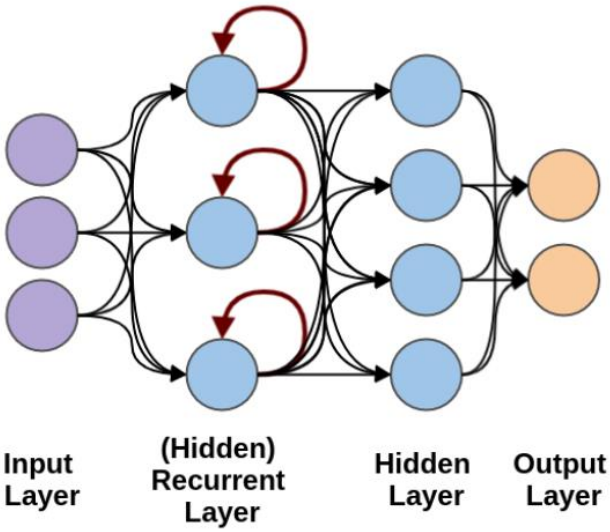


Figure 2 15 Réseau de neurones récurrent

2.8.3 Mémoire à long et court terme (LSTM)

Utilisé lorsque les données problématiques présentent des retards ou des lacunes nécessitant le stockage de certaines données. Même principe que RNN mais avec une nouvelle cellule. Cellules de mémoire capables de stocker les données de nombreuses itérations précédentes, de décider de la quantité d'informations à conserver des itérations précédentes, de réguler la quantité d'informations transmises au niveau suivant et la quantité à détruire. (18)

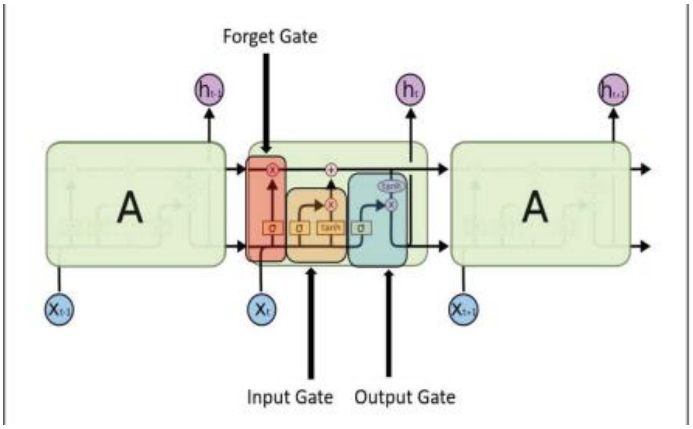


Figure 2 16 Représentation d'une cellule LSTM (7)

2.9 Architectures 3D-CNN

Les 3d cnn sont des réseaux formés de convolution 3D dans toute l'architecture. En convolution 3D, les filtres sont conçus en 3D, et les canaux et temporels les informations sont représentées sous différentes dimensions. Par rapport au temporel techniques de fusion, les 3d CNN traitent les informations temporelles de manière hiérarchique et engager

l'ensemble du réseau. Avant les architectures 3d CNN, la modélisation temporelle était généralement réalisée en utilisant un flux supplémentaire de flux optique ou en en utilisant des couches de regroupement temporel. Cependant, ces méthodes étaient limitées à la 2D la convolution et les informations temporelles ont été placées dans la dimension du canal. Le l'inconvénient des architectures 3d CNN est qu'elles nécessitent d'énormes calculs les coûts et la demande de mémoire par rapport à ses homologues 2D (47).

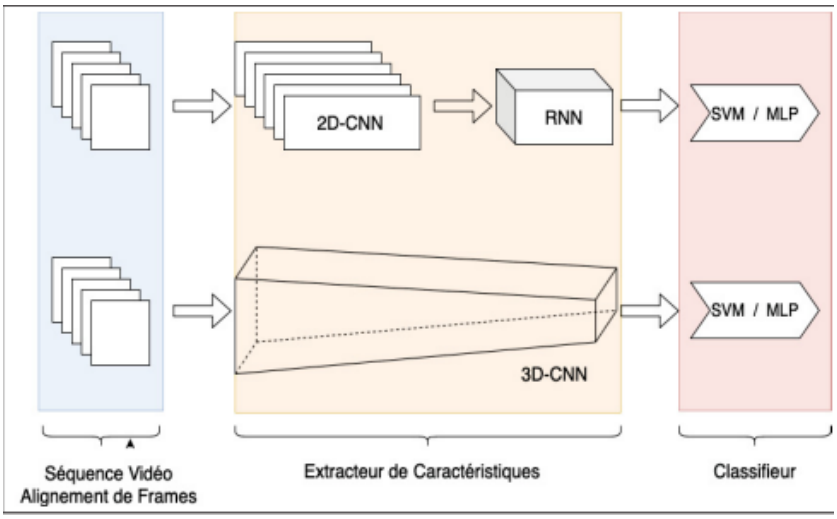


Figure 2 17 Comparaison schématique de l'analyse de séquences vidéo (47)

La **figure 2.17** confronte les architectures CNN-RNN au 3D-CNN. Avec CNN-RNN,, les caractéristiques de chacune des trames sont d'abord extraites avec CNN, puis la concaténation de l'ensemble est vue comme un seul vecteur d'entrée du RNN qui établira la relation temporelle entre chaque trame pour la séquence prédiction d'étiquettes Avec 3DCNN, les caractéristiques spatio-temporelles de chaque séquence sont extraites en une seule étape (7).

2.9.1 Chaîne de traitement des donnée

Dans cette étude nous nous intéressons particulièrement à deux types d'architectures pour la représentation spatio-temporelle de l'émotion : le réseau en cascade de type CNN-RNN et le 3D-CNN

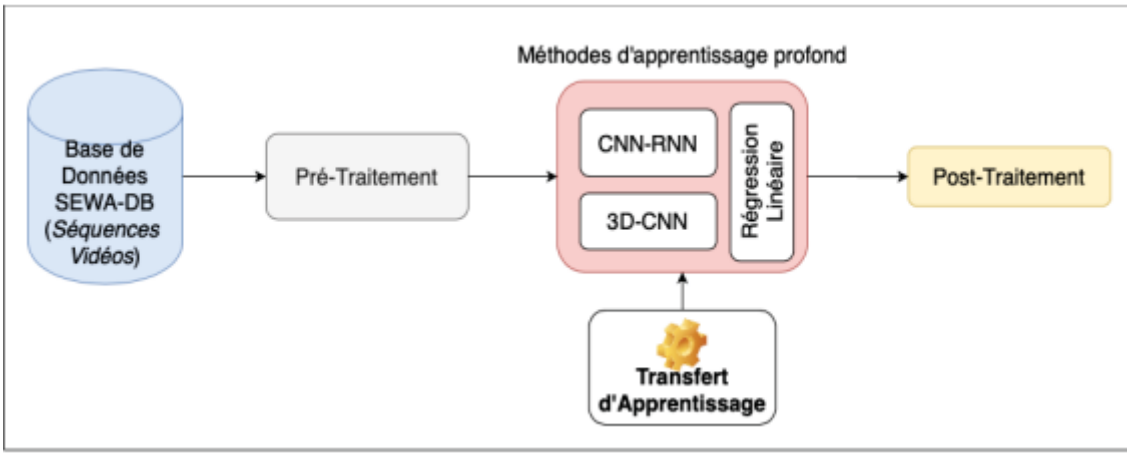


Figure 2 18 Étapes du processus d'apprentissage développé pour l'étude (7)

La Figure 2.1 développe la structure globale de notre chaîne de traitement des données

Nous transférons d'abord les connaissances acquises à travers des domaines plus ou moins proches de notre tâche cible (VGGFace, VGGFace + RAFDB, ImagNet). Ensuite, dans la deuxième étape, les données de notre tâche cible sont organisées dans un nouveau format, ou pour obtenir des prédictions souhaitables pour nos modèles (orientation du visage, normalisation, expansion des données, sélection de mini-clips). Troisièmement, nous détaillerons le point central de l'approche, qui est le développement du modèle d'apprentissage profond basé sur les réseaux de neurones convolutifs (CNN-RNN et 3D-CNN).Enfin, dans la dernière étape, expliquerons les méthodes de post-traitement nécessaires pour optimiser les résultats. (7)

2.10 Conclusion

Dans ce chapitre, nous expliquons les principaux concepts d'un réseau de neurones convolutifs 3D CNN et les architectures de CNN, et transfert d'apprentissag, Nous avons expliqué quelles sont les opérations de base de l'analyse qui sont la convolution, la reconnaissance des changements par RNN

3 CHAPITRE 03 : Implémentation et résultats expérimentaux

3.1 Introduction

Ce chapitre porte sur l'implémentation d'une application de reconnaissance faciale avec des réseaux de neurones convolutifs, deux bibliothèques principales sont utilisées dans le L'implémentation sont Keras et OpenCv.

Nous commençons tout d'abord par la présentation des ressources, du langage et de l'environnement de développement que nous avons utilitaire. Puis les étapes de la réalisation du modèle et on termine par les tests effectués.

3.2 Environnements et outils de développement

Pour l'implémentation de notre système nous avons utilisé une machine présente les caractéristiques suivantes :

Processeur	Intel(R) Core(TM) i3-6006U CPU @ 2.00GHz 2.00 GHz
RAM :	12,0 Go
Type du système :	Système d'exploitation 64 bits, processeur x64

3.3 Mise en place de l'environnement de programmation

Dans cette section on va présenter les outils et les bibliothèques utilisés dans le développement de notre système.

Visual Studio Code : Visual Studio Code (VSCode) est un éditeur de code source léger et puissant développé par Microsoft. Il est doté d'une interface interactive très agréable et possède plusieurs fonctions d'aide et d'extensions. Il supporte un large éventail de langages de programmation, y compris celui que nous allons utiliser, Python.

Figure 3 1 logo Visual Studio Code

Python : Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. C'est est un langage qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. Il est cependant particulièrement utilisé comme langage de script pour automatiser des taches simples.

Figure 3 2 logo Python

OpenCV : Une bibliothèque de vision par ordinateur à code source ouvert qui a été créé pour regrouper les applications de vision par ordinateur. Elle dispose de tonnes d'algorithmes optimisés (plus de 2500) à cet égard. Elle supporte une variété de systèmes d'exploitation, de langages de programmation, de matériels et de logiciel.

TensorFlow : TensorFlow est un outil open source d'apprentissage automatique développé par Google. Le code source à et é ouvert le 9 novembre 2015 par Google et publié sous licence Apache. Initiée par Google en 2011, et est doté d'une interface pour Python et Julia. TensorFlow est l'un des outils les plus utilisés en IA dans le domaine de l'apprentissage machine.

Figure 3 3 logo TensorFlow

Keras : Est une API d'apprentissage profond de haut niveau, simple, flexible et puissante, dont le backend est TensorFlow. Elle vise à faciliter et à minimiser les actions requises pour résoudre les problèmes, ce qui en fait le Framework de Deep Learning le plus utilisé.

Figure 3 4 logo keras

Numpy :

Est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion facilitée des tableaux de nombres, des fonctions sophistiquées (diffusion), on peut aussi l'intégrer le code C / C ++ et Fortran.

MatplotLib : Est une bibliothèque de traçage pour le langage de programmation Python et son extension mathématique numérique NumPy. Il fournit une API orientée objet permettant d'incorporer des graphiques dans des applications à l'aide de kits d'outils d'interface graphique à usage général tels que Tkinter, wxPython, Qt ou GTK +.

Figure 3 5 logo Matplotlib

3.4 Préparation des données

3.4.1 La base de données utilisée :

Fer2013 est un ensemble de données open source d'abord créé par PierreLuc Carrier et Aaron Courville, puis partagé publiquement pour un concours Kaggle juste avant ICML 2013Visage et bien d'autres Dans le domaine du FER également, les résultats sont jusqu'à présent prometteurs, la plupart des gagnants du défi de reconnaissance de l'expression faciale utilisant des réseaux de neurones profonds des photos 48×48 avec différentes émotions comme (joie, colère, peur, tristesse, surprise, dégoût, neutre).

Tableau 3 1 Etiquette d'émotion dans l'ensemble de Donnée Fer 2013

Étiquette	Nombre de photos	expression de faciel
0	3955	colère
1	436	dégoût
2	4097	peur
3	7215	joie
4	4830	tristesse
5	3171	surprise
6	4965	neutre

Figure 3 6Exemple de la base Fer2013

3.4.2 Comparaison entre les différentes architectures

Dans le Chapitre 2, nous avons présenté les architectures CNN les plus récentes et décrites dans le site KERAS officiel et TensorFlow, ces architectures seront utilisées dans notre étude expérimentale.

Il est à noter que toutes ces architectures sont formées sur la base ImageNet.

- **ImageNet :** ImageNet : est une base de données visant à améliorer la recherche en apprentissage profond, il a plus de 14 millions d'images hiérarchiques. (48)
  L'entraînement des CNNs sur des grands ensembles de donnés peut prendre des jours, voire des semaines, même sur des ordinateurs multi-gpu très performants. L'apprentissage par transfert est donc un moyen de raccourcir cette opération chronophage.
- **L'apprentissage par transfert** consiste à prendre des modèles prés-entraînés sur une base de données de référence standard de vision par ordinateur (tel que : ImageNet, COCO, FMNIST, etc.) congeler les couches inférieures pour ne pas être entrainées à nouveaux et à réutiliser ses poids en plus d'ajouter quelques couches en haut.

I. Les métriques utilisées

Le choix de la meilleure architecture était basé sur 4 métriques que nous expliquerons pour mieux comprendre la décision prise.

- **Précision** est la métrique utilisée pour évaluer les modèles de classification (comme le nôtre), elle représente le nombre de prédictions vraies obtenues par le modèle (49). Elle est décrite par :

Précision = $Nombre\ de\ prédiction\ correctes / Nombre\ total\ de\ prédiction$

- **L'erreur :** La valeur de l'erreur est la pénalité pour les mauvaises prédictions, décrivant à quel point la prédiction est mauvaise ou éloignée de sa valeur réelle (49). Pour notre problème, nous avons utilisé la fonction d'entropie croisée catégorielle qui calcule cette somme comme pertes :

$$Perte = -\sum_{i=1}^{TailleDeSortie} y_i * log\hat{y}_i$$

- **Précision de la validation :** est la précision calculée pendant l'entrainement mais sur des données qui ne sont pas utilisées pour l'entrainement. Elle est utilisée pour valider le degré de généralisation de notre modèle sur des données non vues (49).
- **Erreur de validation** : Calculée de la même manière que la perte d'apprentissage mais n'est pas utilisée pour mettre à jour les poids, elle sert uniquement à voir les pertes généralisées (49).

II. Résultats des tests des architectures

Dans le chapitre 2, nous avons introduit les dernières structures CNN et maintenant comparer les modèles de structures CNN avec les yeux et de la bouche

Tableau des résultats des modèles des yeux :

	L'erreur	*Précision*	*L'erreur de validation*	*Précision de la validation*
VGG16	0.3096	0.8658	1.0220	0.5475
VGG19	0.4589	0.7903	0.5729	0.7225
EfficientNetB7	0.5516	0.7385	0.6228	0.6700
DenseNet	0.4785	0.7717	0.6181	0.6550
InceptionV3	0.4780	0.7778	0.4895	0.7763
ResNet50V2	0.3721	0.8455	0.6258	0.5612
InceptionResnetV2	0.4627	0.7965	0.6363	0.5525
Xception	0.4704	0.7918	0.3769	0.9175
NasNetMobile	0.5779	0.6975	0.6240	0.6413
MobileNetV2	0.4148	0.8002	0. 4703	0.7862

Tableau 3 2 Tableau des résultats des modèles des yeux : (49)

Tableau des résultats des modèles de la bouche*

	L'erreur	*Précision*	*L'erreur de validation*	*Précision de la validation*
VGG16	0.4028	0.8215	0.5232	0.7250
VGG19	0.3308	0.8568	0.7770	0.5063
EfficientNetB7	0.5942	0.6973	0.6222	0.7175
DenseNet	0.3177	0.8767	0.2568	0.9175
InceptionV3	0.2820	0.8915	0.3228	0.8763
ResNet50V2	0.3617	0.8443	0.3939	0.8175
InceptionResnetV2	0.3407	0.8622	0.5234	0.7350
Xception	0.4877	0.7878	0.5568	0.7387
NasNetMobile	0.3130	0.8680	0.6198	0.6800
MobileNetV2	0.3088	0.8740	0.2153	0.9475

Tableau 3 3 Tableau des résultats des modèles de la bouche (49)

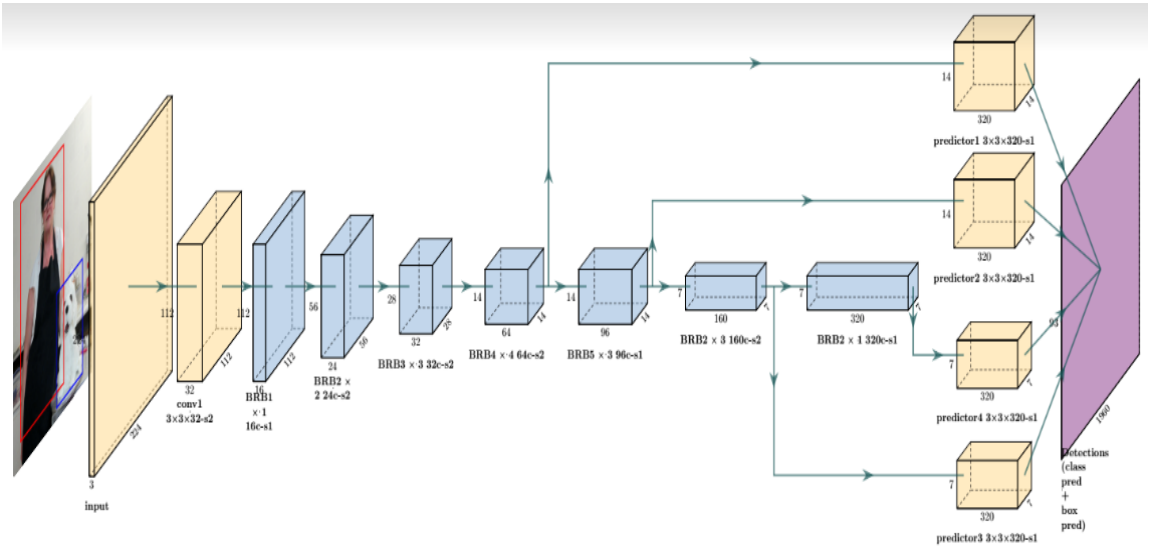
En analysant les résultats des tests pour toutes les structures, qui sont présentés dans le formulaire et les tableaux

Ci-dessus, nous avons fini par choisir **MobileNetV2.**

MobileNetV2 : Cette architecture obtient un très bon score pour l'évaluation de l'état des yeux et

La bouche, en plus, c'est une construction légère qui convient très bien dans notre projet d'ingénierie.

- **Architecture MobileNetV2**



3.5 Implémentation

3.5.1 Import libraries

La première étape consiste à importer ce dont nous avons besoin à partir des bibliothèques Python pour commencer notre formation en apprentissage approfondi, ceci sont des bibliothèques utilisées dans notre formation

```
my emotion recognition test > rec.py > ...
1   import tensorflow as tf
2   import keras
3   from tensorflow.keras import layers
4   import cv2
5   import numpy as np
6   import matplotlib.pyplot as plt
7   import os
8   import tkinter
9   from tkinter import filedialog
10  from tkinter.ttk import *
11  from tkinter import *
12  import random
```

Figure 3 7 Importer des bibliothèques

3.5.2 Chargement du jeu de données

La deuxième étape consiste à charger notre ensemble de données qui va être utilisé dans la formation, mais comme nous ne le faisons pas avoir un fichier np pour notre ensemble de données

nous avons des images dans des dossiers), nous devons créer notre data.np pour simplifier les étapes de pré-traitement avec un simple code python

```
create_training_data()
random.shuffle(training_date)
#print(len(training_date))
x=[]
y=[]
for features,labels in training_date:
    x.append(features)
    y.append(labels)

x=np.array(x).reshape(-1,img_size,img_size,3)
x=x/255.0
y=np.array(y)
model = tf.keras.applications.MobileNetV2()
```

Figure 3 8 Des images au fichier npy

Une fois que nous avons préparé nos données de formation et nos données de validation dans des fichiers np, l'ensemble de données doit déjà charger comme décrit dans le code suivant :

```
datadir = "my emotion recognition test/training/train"
classes = ["0","1","2","3","4","5","6"]
reactions = ["colère","dégoût","peur","joie","tristesse","surprise","neutre"]
```

Figure 3 9 Chargement de l'ensemble de données

3.5.3 Formation CNN

Pour cette étape, nous allons commencer à écrire le fichier train.py, et importer nos packages

```
from keras.applications import mobilenetV2
from keras.models import Sequential,Model
from keras.layers import Dense,Dropout,Activation,Flatten,GlobalMaxPooling2D
from keras.layers import Conv2D,MaxPooling2D,ZeroPadding2D
from keras.layers.normalization import BatchNormalization
from keras.preprocessing.image import ImageDataGenerator
```

Figure 3 10 Importer des bibliothèques train

La classe ImageDataGenerator sera utilisée pour l'augmentation des données, une technique utilisée pour prendre des images existantes dans notre ensemble de données et appliquer aléatoirement Transformations pour générer plus de données d'entraînement qui réduiront le surapprentissage.

Puisque notre ensemble de données est maintenant prêt, nous allons concevoir un modèle de réseau neuronal convolutif architecture utilisant le modèle séquentiel de Keras avec les configurations de réglage comme indiqué dans la prochaine code:

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3,3),input_shape=(img_width,
img_height,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(32, kernel_size=(3,3), activation='relu'))
model.add(Conv2D(32, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(32, kernel_size=(3,3), activation='relu'))
model.add(Conv2D(32, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.7))
model.add(Dense(classes_num, activation='softmax'))
```

Le train-test-split sera utilisé pour créer nos ensembles de données d'entraînement et de test.

Après avoir importé les packages nécessaires, nous allons initialiser certaines variables :

```
newmod= keras.Model(inputs=base_input,outputs=final_output)

#newmod.summary()


newmod.compile(loss="sparse_categorical_crossentropy", optimizer="adam" , metrics = ["accuracy"])
newmod.fit(x,y,epochs=200)
newmod.save('mymodel.h5')
newmod = tf.keras.models.load_model('final.h5')
```

Epoch : Le nombre total d'époques où nous formerons notre réseau pour (c'est-à-dire combien de fois notre réseau "voit" chaque exemple de formation et en tire des modèles).

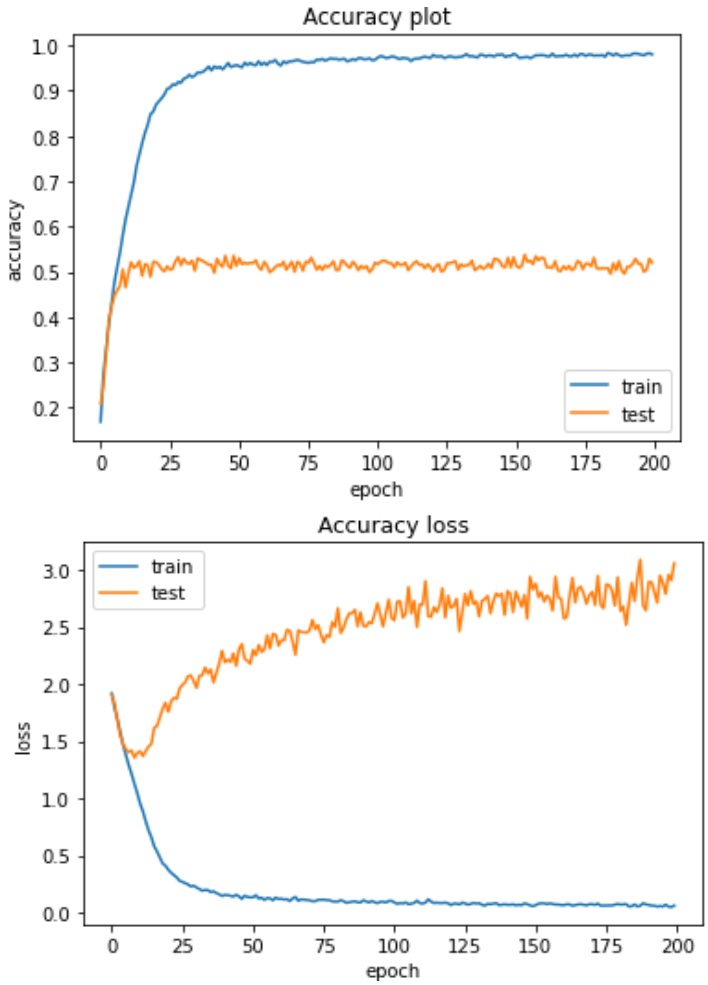
Former le CNN sur mon ordinateur portable :

```
Epoch 8/25
145/145 [==============================] - 1043s 7s/step - loss: 0.7311 - accuracy: 0.7371
Epoch 9/25
145/145 [==============================] - 1138s 8s/step - loss: 0.6850 - accuracy: 0.7560
Epoch 10/25
145/145 [==============================] - 1095s 8s/step - loss: 0.5874 - accuracy: 0.7839
Epoch 11/25
145/145 [==============================] - 1059s 7s/step - loss: 0.4934 - accuracy: 0.8292
Epoch 12/25
145/145 [==============================] - 944s 7s/step - loss: 0.4628 - accuracy: 0.8367
Epoch 13/25
145/145 [==============================] - 1134s 8s/step - loss: 0.3847 - accuracy: 0.8676
Epoch 14/25
145/145 [==============================] - 1026s 7s/step - loss: 0.3428 - accuracy: 0.8792
Epoch 15/25
145/145 [==============================] - 1072s 7s/step - loss: 0.3206 - accuracy: 0.8900
Epoch 16/25
145/145 [==============================] - 1081s 7s/step - loss: 0.2611 - accuracy: 0.9098
Epoch 17/25
145/145 [==============================] - 1105s 8s/step - loss: 0.2586 - accuracy: 0.9111
Epoch 18/25
145/145 [==============================] - 1096s 8s/step - loss: 0.2477 - accuracy: 0.9165
Epoch 19/25
145/145 [==============================] - 1064s 7s/step - loss: 0.2088 - accuracy: 0.9271
Epoch 20/25
145/145 [==============================] - 1275s 9s/step - loss: 0.2057 - accuracy: 0.9316
Epoch 21/25
145/145 [==============================] - 925s 6s/step - loss: 0.1548 - accuracy: 0.9472
Epoch 22/25
145/145 [==============================] - 909s 6s/step - loss: 0.1799 - accuracy: 0.9418
Epoch 23/25
145/145 [==============================] - 867s 6s/step - loss: 0.1829 - accuracy: 0.9418
Epoch 24/25
145/145 [==============================] - 903s 6s/step - loss: 0.1618 - accuracy: 0.9528
Epoch 25/25
145/145 [==============================] - 1132s 8s/step - loss: 0.1061 - accuracy: 0.9679
```

Figure 3 11 Précision et perte avant et après l'entraînement

3.5.4 Analyse de performance

Des résultats acceptables et fiables ont été obtenus tels que représentés dans la courbe graphique représentant notre courbe d'entraînement et notre parcours d'acceptation, qui représente l'évolution de la précision de la reconnaissance des émotions et du taux d'échec.



3.5.5 Le fonctionnement et les résultats

Dans cette partie, nous expliquerons en général ce que fait le programme, qui dépend de méthode 2D-CNN, qui connaît les sentiments d'une personne en identifiant les sentiments du visage. En utilisant la fonction que nous avons mentionnée dans la première chapiter, Viola et Jones qui détermine le visage à l'aide du fichier.

opencv/haarcascade_frontalface_default.xml : qui est un fichier qui a maîtrisé la reconnaissance faciale, ou nous l'appelons le meilleur dossier appris pour la reconnaissance facial, qui a prouvé son exactitude dans de nombreux programmes et langues**.**

Mais avant que le processus d'identification des sentiments ne commence, vous devez télécharger le fichier entrain Que nous avons déjà formé sur mon ordinateur et après expérimentation, le fichier peut être entrainement dans l'ordinateur, mais il ne faut que beaucoup de temps, Pour former le fichier en fonction de l'apprentissage par transfert et de l'apprentissage profond, premier apprentissage par transfert afin de tirer parti des expériences précédentes, apprentissage profond

afin d'enseigner le fichier spécifiquement après avoir téléchargé le modèle de **model.h5** nous entraînons le fichier à travers la base de données qui est des images de **data/train** que nous avons apportées.

```
newmod= keras.Model(inputs=base_input,outputs=final_output)

#newmod.summary()


newmod.compile(loss="sparse_categorical_crossentropy", optimizer="adam" , metrics = ["accuracy"])
newmod.fit(x,y,epochs=200)
newmod.save('mymodel.h5')
newmod = tf.keras.models.load_model('final.h5')
```

Après avoir importé le fichier haarcascade nous aurons écrit un code pour détecter les visages et classer les émotions souhaitées pour chaque frame respectivement. Nous avons attribué les étiquettes qui seront différentes émotions comme (colère, heureuse, triste, surprise, neutre, dégoute, peureux). Dès que vous exécutez le code, une nouvelle fenêtre apparaîtra et votre webcam s'allumera. Il détectera ensuite le visage de la personne, tracera un cadre de délimitation sur la personne détectée, puis convertira l'image RVB en niveaux de gris et la classera en temps réel. Veuillez-vous référer au code ci-dessous pour les mêmes et exemples de sorties qui sont affichés dans les images. Pour arrêter le code, vous devez appuyer sur « q »

```
classes = ["0","1","2","3","4","5","6"]
reactions = ["colère","dégoût","peur","joie","tristesse","surprise","neutre"]
```

Numéro de 0 à 6, qui est le nombre de sentiments qui ont été étudiés.

Exécution la reconnaissance d'émotions en temps réel :

```
= cv2.VideoCapture(thesubject)

le True:
ret, frame = cap.read()
if thesubject != 0:
    frame = cv2.resize(frame,(720,680))
if not ret:
    print("no objects detected")
    break
facecascade = cv2.CascadeClassifier(cv2.data.haarcascades+path)
gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
faces = facecascade.detectMultiScale(gray,1.1,4)
for x,y,w,h in faces:

    roi_gray = gray[y:y+h, x:x+w]
    roi_color = frame[y:y+h, x:x+w]
    cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
    facess=facecascade.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in facess:
        face_roi = roi_color[ey:ey+eh,ex:ex+ew]
        final_image = cv2.resize(face_roi,(224,224))
        final_image = np.expand_dims(final_image,axis=0)
        final_image=final_image/255.0
        font=cv2.FONT_HERSHEY_SIMPLEX
        Predictions = newmod.predict(final_image)
```

```
cv2.imshow('face emotion',frame)
if cv2.waitKey(2) & 0xFF ==ord ('q'):
    break
.release()
.destroyAllWindows()
```

Figure 3 12 Figure 3 12 Exécution la reconnaissance d'émotions en temps réel

3.5.6 Le modèle en video.mp4 :

Après avoir importé les poids du modèle, nous avons importé un fichier .mp4 (vidéo.mp4), une nouvelle fenêtre s'ouvrira et votre vidéo s'allumera. On fait le même travail qu'avant en webcam en temps réel jusque a la fin de la vidéo pour chaque frame respectivement :

```
def videoprediction(self):
    chemin = UploadAction()
    cap = cv2.VideoCapture(chemin)

    while True:
        ret, frame = cap.read()
        if chemin != 0:
            frame = cv2.resize(frame,(900,600))
        if not ret:
            print("no objects detected")
            break
        facecascade = cv2.CascadeClassifier(cv2.data.haarcascades+path)
        gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
        faces = facecascade.detectMultiScale(gray,1.1,4)
        for x,y,w,h in faces:

            roi_gray = gray[y:y+h, x:x+w]
            roi_color = frame[y:y+h, x:x+w]
            cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
            facess=facecascade.detectMultiScale(roi_gray)
            for (ex,ey,ew,eh) in facess:
                face_roi = roi_color[ey:ey+eh,ex:ex+ew]
                final_image = cv2.resize(face_roi,(224,224))
                final_image = np.expand_dims(final_image,axis=0)
                final_image=final_image/255.0
                font=cv2.FONT_HERSHEY_SIMPLEX
                Predictions = newmod.predict(final_image)
```

Figure 3 13 Insertions des vidéos externes et applique le test.

3.5.7 Le modèle en image.png

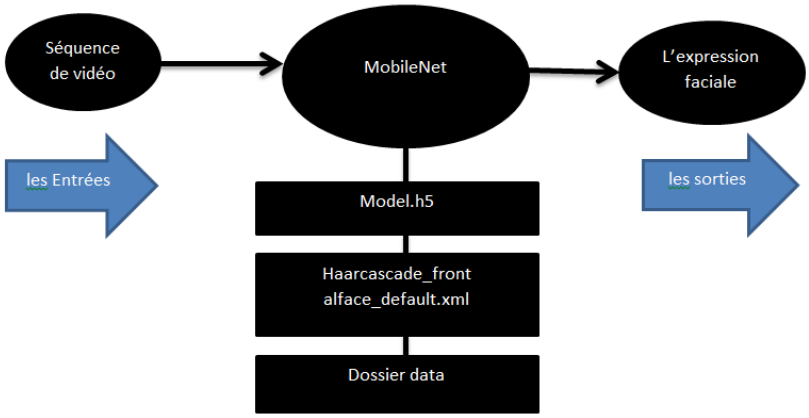
Importer le fichier image.png, une nouvelle fenêtre s'ouvrira et votre photo s'allumera avec les visages détectés et les émotions sélectionnées pour chaque visage (un seul frame) :

```
def imageprediction(self):
    chemin = UploadAction()
    cap = cv2.imread(chemin)
    #print(cap);
    predictions =   DeepFace.analyze(cap)
    #print(predictions)
    facecascade = cv2.CascadeClassifier(cv2.data.haarcascades+
    gray=cv2.cvtColor(cap,cv2.COLOR_BGR2GRAY)
    faces = facecascade.detectMultiScale(gray,1.1,4)
    for x,y,w,h in faces:
        cv2.rectangle(cap,(x,y),(x+w,y+h),(255,0,0),2)
    font = cv2.FONT_HERSHEY_SIMPLEX

    prd = []
    reaction=""
    prd.append(predictions['emotion']['disgust'])
    prd.append(predictions['emotion']['fear'])
    prd.append(predictions['emotion']['happy'])
    prd.append(predictions['emotion']['sad'])
    prd.append(predictions['emotion']['surprise'])
    prd.append(predictions['emotion']['angry'])
    if(predictions['dominant_emotion'] == "disgust"):
        reaction = "degout"
    elif(predictions['dominant_emotion'] == "fear"):
        reaction = "peur"
    elif(predictions['dominant_emotion'] == "happy"):
        reaction = "joie"
    elif(predictions['dominant_emotion'] == "sad"):
        reaction = "tristesse"
    elif(predictions['dominant_emotion'] == "surprise"):
        reaction = "surprise"
```

Figure 3 14 Insertions des images externes et applique le test.

3.5.8 Architecture



3.5.9 Résultat

Si nous faisons une comparaison de la (Figure) nous constatons qu'entre ces deux itérations nous constatons que le taux d'erreur baisse tant disque notre précision augmente cela signifie que notre modèle a été bien entrainer et réponds d'ailleurs à la définition du réseau de neurones qui disait que plus le réseau de neurones est profond meilleur sont ses performances.

Exactitude et perte avant et après de trainement :

1.

```
Epoch 2/25
145/145 [==============================] - 856s 6s/step - loss: 1.3496 - accuracy: 0.4950
..                                  ..                                  ..
..                                  ..                                  ..
..                                  ..                                  ..
Epoch 22/25
145/145 [==============================] - 909s 6s/step - loss: 0.1799 - accuracy: 0.9418
Epoch 23/25
145/145 [==============================] - 867s 6s/step - loss: 0.1829 - accuracy: 0.9418
Epoch 24/25
145/145 [==============================] - 903s 6s/step - loss: 0.1618 - accuracy: 0.9528
Epoch 25/25
 76/145 [==============>...............] - ETA: 7:24 - loss: 0.0937 - accuracy: 0.9708
```

Figure 3 15 Exactitude et perte avant et après de trainement

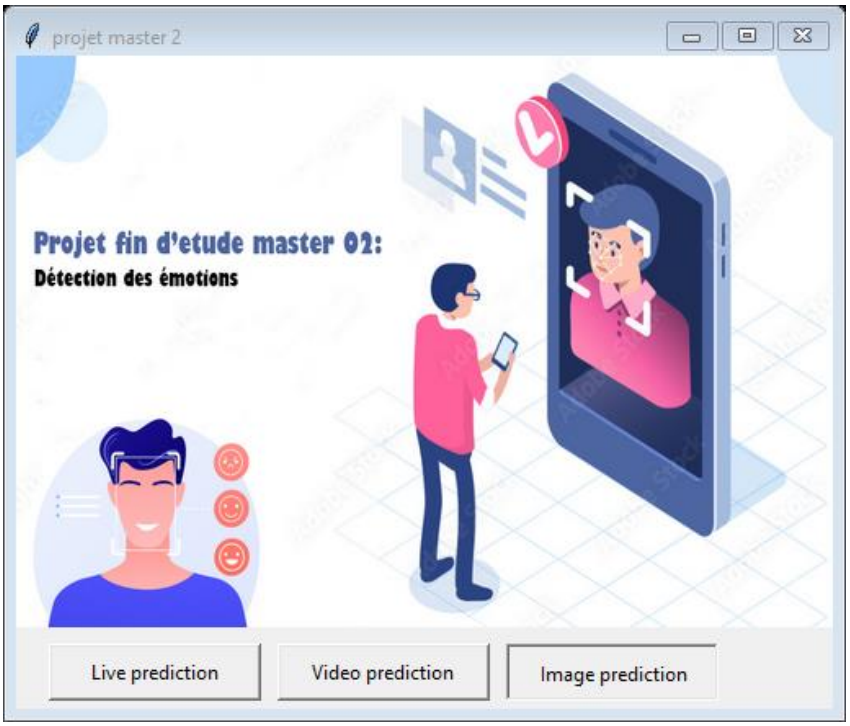


Figure 3 16 Page 1 de l'application

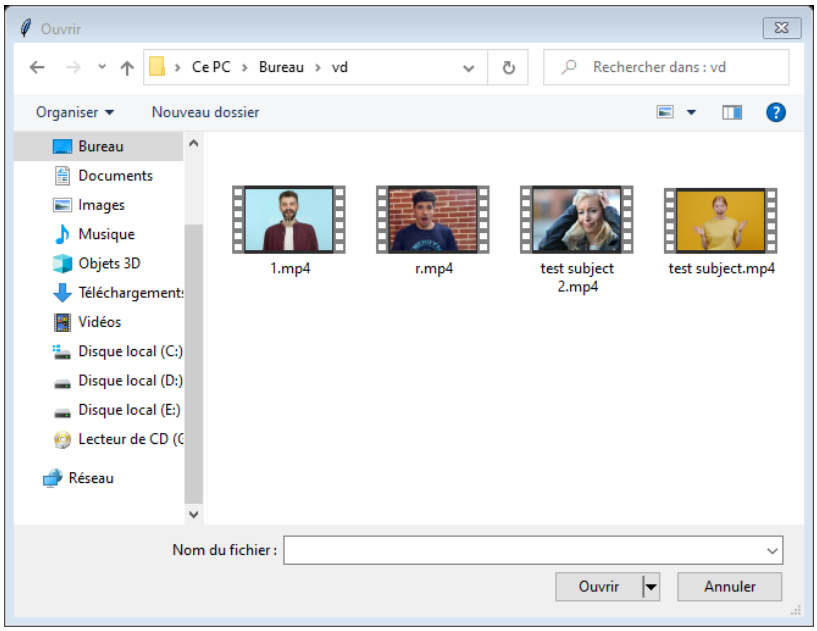


Figure 3 17 Vidéo prediction

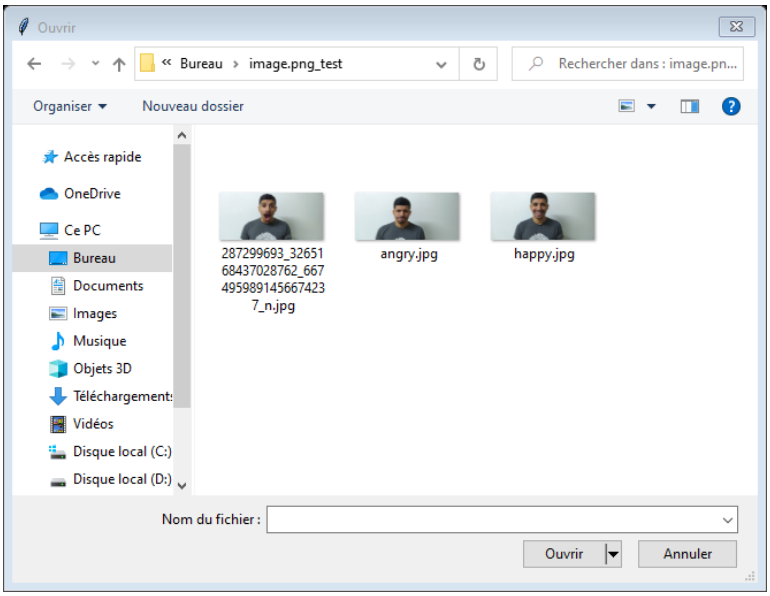
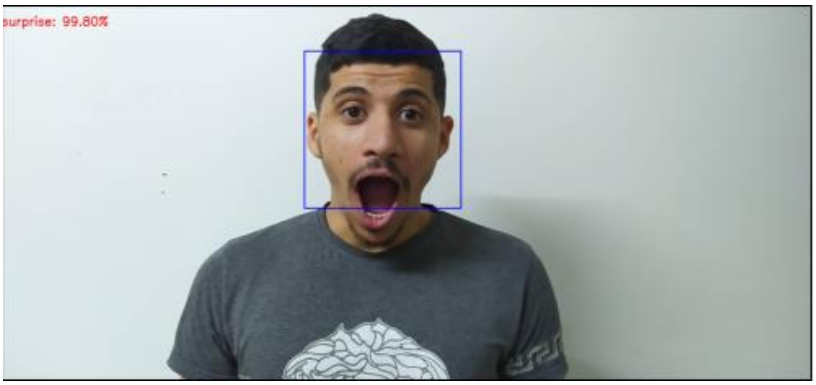
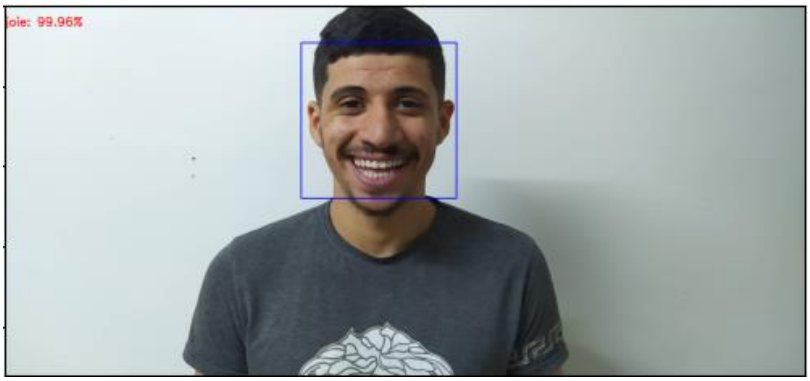


Figure 3 18 Image prediction





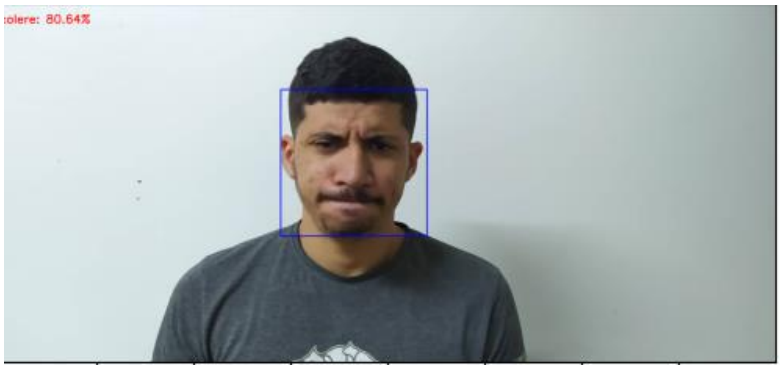


Figure 3 19 Les résultats de ma photo.

Figure 3 20 Résultats de la photo externe.

3.6 Conclusion

Nous avons présenté dans ce chapitre la mise en œuvre de l'approche de reconnaissance des expressions faciales basée sur des réseaux de neurones convolutifs 2D , Parce que nous n'avons pas trouvé d'images 3D, la méthode d'apprentissage était des images de qualité moyenne (224 * 224) mais à l'avenir il a été suggéré que l'apprentissage sur des images de haute qualité soit plus efficace

.

4 Conclusion Générale

Nous avons présenté à travers ce mémoire de fin d'étude une étude théorique ce que concerne les expressions faciales et l'apprentissage approfondie en se basant sur les réseaux de neurone convolutifs (CNN).

Nous avons par la suite utilisé un modèle de CNN approfondi avec une base de données fer-2013 ce qui nous fournit un très bon résultat d'entrainement et un classificateur à la capacité de classifier les expressions faciales avec un gain de performance arrive à 70% de précision.

Ce travail a connu des difficultés à cause de

- Les taux de puissance et de reconnaissance sont faibles dans les grandes bases de données.
- Les résultats de ces méthodes traditionnelles restent limités dans le cadre d'un traitement à très grande échelle
- Les caractéristiques sont définies par des méthodes traditionnelles Le succès actuel dans la résolution de ces problèmes est dans le Deep Learning ou plus particulièrement les réseaux de neurones convolutifs CNN car:
- CNN apprend à les extraire directement de la base de données d'apprentissage.
- Le réseau CNN est généralement associé à un classificateur pour former le modèle de bout en bout sur l'ensemble de données.

• La disponibilité des bases de données et la puissance de calcul des machines actuelles ont des valeurs à ajouter pour les CNN.

Comme perspectives de recherches futures, nous envisageons de :

1. Tester sur notre modèle d'autre base de données autre que celle de Fer2013.
2. Augmenter de nouvelle couche afin de voir ce que cava donnée.
3. Travailler avec les modèles déjà entrainés tels que le VGG16 et faire une comparaison.

5 Bibliographie

1. **Folleto, Villani.** *Livret de vulgarisation Mission Villani sur l'intelligence artificielle.* mars 2018.

2. **l'OMPI.** *Rapport 2019 de l'OMPI sur les tendances technologiques Résumé Intelligence artificielle.* 2019.

3. **Konstantinos, Theodoridis Sergios & Koutroumbas.** *Pattern Recognition Fourth Edition.* 2003.

4. **rahid, Zaghdoudi.** *Les techniques de reconnaissance de Formes Application.* Guelma : université 08 mai 1945., 2018.

5. **Guillaume Brigot.** *Prédire la structure des forêts à partir d'images PolInSAR par apprentissage de descripteurs LIDAR.* Français : Université Paris Saclay y (COmUE), 2017.

6. **Xavier, Glorot.** *Apprentissage des reseaux de neurones profonds et applications entraitement automatique de la langue naturelle.* s.l. : Universit´e de Montréal, Novembre, 2014.

7. **Teixeira, Thomas.** *Reconnaissance multi-dimensionnelle de l'émotion par apprentissage profond de caractéristiques spatio-temporelles sur séquences vidéo.* 10 SEPTEMBRE 2020.

8. **Boris, Mohamadally Hasan Fomani.** *Machines à Vecteurs de Support ou Séparateurs à Vastes Marges.* 16 janvier 2006.

9. **Chamroukhi, Faïcel.** *Classification supervisée : Les K-plus proches voisins.*

10. **Muallim Mohammad Tarek.** *Pattern Recognition using Artificial Immune.* 2008-2009.

11. **Desjardins, Julie.** *L'analyse de régression logistique Université de Montréa.* 2005. Vol. 1(1), p. 35-41..

12. **ABDAT, Faiza.** *Reconnaissance automatique des émotions par données multimodales : expressions faciales et signaux physiologiques.* 15 - 06 - 2010.

13. **GHANEM, KHADOUDJA.** *Reconnaissance des Expressions Faciales à Base d'Informations Vidéo Estimation de l'Intensité des Expressions .*

14. **bengio, yoshua.** *Learning Deep architectures for AI. in Foundations and Trends in Machine Learning.* 2009.

15. **Touzet, Claude.** *LES RESEAUX DE NEURONES ARTIFICIELS,INTRODUCTION AU CONNEXIONNISME.* Juillet 1992.

16. **Ahlberg, Jörgen.** *CANDIDE-3 - An Updated Parameterised Face (2001).*

17. *<Digital image processing>.* **yadav, A.** new delhi boston usa. : s.n., 2019.

18. **Mahmoud, SEKKIL Hicham Mohamed MEBROUKI.** *Etude comparative entre les différentes architectures des réseaux de neurones convolutifs (CNNs) pour la détection de convolutifs (CNNs) pour la détection de.* 26 / 09 / 2021.

19. *Détection de la fumée et du feu par réseau de neurones convolutifs.* **Sébastien Frizzi, Rabeb Kaabi, Moez Bouchouicha, Jean-Marc Ginoux, Farhat Fnaiech, Eric Moreau.**

20. **Amjad, Sohaib Asif Kamran.** *Automatic COVID-19 Detection from chest radiographic images using Convolutional Neural Network.*

21. *Receptive fields of single neurones in the cat's striate cortex.* **Hubel, Wiesel et.** p. 574–591, 22 April 1959.

22. *Handwritten digit recognition with a backpropagation network.* **al, Y. LeCun et.** pp. 396-404, 1990.

23. *Gradient-based learning applied to document recognition,.* **al, Y. LeCun et.** pp. 2278-2324, 1998.

24. *Imagenet classification with deep convolutional neural networks.* **KRIZHEVSKY, Alex, SUTSKEVER, Ilya, HINTON et E.Geoffrey.** pp. 1097-1105, 2012.

25. *Very deep convolutional networks for large-scale image recognition.* **A.Zisserman, K.Simonyan et.** 2014.

26. *Going Deeper with Convolutions.* **C.Szegedy, W.Liu, Y.Jia, P.Sermane, S.Reed, D.Anguelov, D.Erhan, V.Vanhoucke et A.Rabinovich.** 1-9, 2015.

27. *Rethinking the Inception Architecture for Computer Vision.* **C.Szegedy, V.Vanhoucke, S.Ioffe, J.Shlens et Z.Wojna,.** p. 2818–2826, 2016.

28. *Deep residual learning for image recognition.* **HE, Kaiming, ZHANG, Xiangyu, REN, Shaoqing et al.** pp. 770-778,, 2016.

29. *FaceNet : A Unified Embedding for Face Recognition and Clustering.* **SCHROFF, Florian, KALENICHENKO, Dmitry, PHILBIN et James.** 2015.

30. **Weinberger, Gao Huang et Zhuang Liu et Laurens van der Maaten Kilian Q.** *Densely Connected Convolutional Networks.* 2017. p. 4700–4708.

31. **Kalenichenko, Andrew G. Howard Menglong Zhu et Bo Chen et Dmitry.** *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.* 2017. arXiv:1704.04861.

32. **Chen, Mark Sandler Andrew Howard Menglong Zhu Andrey Zhmoginov Liang-Chieh.** *MobileNetV2: Inverted Residuals and Linear Bottlenecks .* 2018. p. 4510–4520,.

33. **Jones, P. Viola et M.** *Robust real-time face detection.* Juillet 2001 . page 747.

34. **Jones, Viola et M.** *Rapid object detection using a boosted cascade of simple features .* Décembre 2001 . pages 511–518.

35. **Schapire, Y. Freund et R.** *Experiments with a new boosting algorithm .* Janvier 1996. pages 148–156.

36. **Gelly, Gregory.** *Reseaux de neurones recurrents pour le traitement automatique de la parole .* 12 Oct 2017 .

37. **Nguyen, Dung.** *Multimodal Emotion Recognition Using Deep Learning Techniques.*

38. **M. Pantic, L.J.M. Rothkrantz.** *Expert system for automatic analysis of facial expressions.* December 2000. pp. 1424-1445.

39. **Essa, Antonio Haro Myron Flickner Irfan.** *Detecting and Tracking Eyes By Using Their Physiological Properties, Dynamics, and Appearance .* June 2000.

40. **Y. Tian, T. Kanade, J. Cohn.** *Robust Lip Tracking by Combining Shape, Color and Motion .* 2000.

41. **T. Coianiz, L Torresani , B. Caprile.** *2D Deformable Models for Visual Speech Analysis" In NATO Advanced Study Institute : Speech reading by Man and Machine.* 1995. pp.391-398..

42. **M.E. Hennecke, K.V Prasad, D.G. Storck.** *"Using Deformable Templates to Infer Visual Speech Dynamics". In Proc. 28th Annual Asilomar Conference on Signals, Systems and computers.* 1994. pp. 578-582..

43. **al, L. Dang et.** *Deep Learning Based Computer Generated Face Identification Using Convolutional Neural Network .* 2018 .

44. **Zisserman, Joao Carreira Andrew.** *Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset .* 2017.

45. **Yu, Shuiwang Ji Wei Xu Ming Yang Kai.** *3D Convolutional Neural Networks for Human Action Recognition.* 2012.

46. **jones, M. tim.** *Deep learning architectures.* https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures/ : s.n.

47. **Alatan, M. Esat Kalfaoglu Sinan Kalkan and A. Aydin.** *Late Temporal Modeling in 3D CNN Architectures with BERT for Action Recognition .*

48. ImageNet. [En ligne] 11 mars 2021. https://image-net.org/.

49. [En ligne] 16 July 2021. https://developers.google.com/machine-learning/crash-course.