

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Électronique
Spécialité Electronique des Systèmes Embarqués

présenté par

Saada Fawzi

&

Siouane Mouhssin

Reconnaissance des expressions faciales en temps réel

Proposé par : Naceur Djamila

Année Universitaire 2019-2020

Remerciements

Nous remercions tout d'abord **DIEU** qui nous a donné la force et le courage pour pouvoir réaliser ce travail.

Nous tenons à exprimer nos sincères remerciements et notre reconnaissance notre encadreur **Mme Naceur Djamila** d'avoir proposé ce sujet qui nous a mené vers la découverte du monde de l'expérimental, de nous avoir bien dirigé afin de pouvoir réaliser ce travail, nous n'oublions pas de la remercier pour sa disponibilité, ses conseils et son expérience.

Nous remercions également les membres du jury d'avoir accepté d'évaluer notre travail.

Sur le plan plus personnel, nous remercions vivement nos parents pour leurs encouragements et leur soutien pendant notre cursus, ainsi que tous les membres de nos chères familles et tous nos amis.

Enfin, on remercie tous ceux qui ont contribué de près ou de loin au bon déroulement de ce travail.

Merci

Dédicaces

J'exprime mes sentiments les plus profonds et je dédie ce

Modeste travail :

*A mon père, ma mère, pour l'éducation qu'ils ont su me donner et qui m'a permis avec l'aide de **DIEU** d'arriver là où je suis.*

*A Ma chères sœur **Hadjer, Nermine, Ryan** et mon Chèr frère **Fouad***

A toute ma famille.

A tous mes amis et tous ceux qui m'ont aidé afin de réaliser ce travail.

Mouhssin

Dédicaces

J'exprime mes sentiments les plus profonds et je dédie ce

Modeste travail :

*A mon père, ma mère, pour l'éducation qu'ils ont su me donner et qui m'a permis avec l'aide de **DIEU** d'arriver là où je suis.*

*A Ma chère sœur **SARA** et mon Chère frère **MOUHAMED***

A toute ma famille.

A tous mes amis et tous ceux qui m'ont aidé afin de réaliser ce travail.

FAWZI

ملخص:

تعبيرات الوجه هي الطريقة غير اللفظية الأكثر شيوعًا، والتي يستخدم فيها الإنسان حركة أو وضعًا واحدًا أو أكثر للعضلات تحت جلد الوجه. يتكون مشروعنا من توفير نهج التعرف على الوجه بناءً على التعلم العميق وبشكل أكثر تحديدًا على الشبكات العصبية التلافيفية بما في ذلك بنيتان لشبكة CNN: VGG16 و Bottleneck، واختيارنا لهذين البنيتين مدفوعان بهما. النجاح في التعرف على الصور. يتم تنفيذ تدريب الشبكة العصبية باستخدام مكتبة TensorFlow والمكتبات الأخرى مثل keras و Transfer Learning كنموذج يتم استخدامه في مرحلة التدريب.

الكلمات المفتاحية: التعلم العميق، معالجة الصور، CNN، تعبير الوجه، VGG16، Python، IA

Résumé

L'expression faciale est la méthode non verbale la plus courante, dans laquelle les humains utilisent un ou plusieurs mouvements ou positions des muscles sous la peau du visage.

Notre projet consiste à fournir une approche de reconnaissance faciale basée sur l'apprentissage profond (deep Learning) et plus particulièrement sur les réseaux de neurones convolutifs dont deux architectures du réseau CNN : VGG16 et Bottleneck, Notre choix de ces deux architectures est motivé par leur succès dans la reconnaissance d'images.

L'entraînement du réseau de neurones est implémenté à l'aide de la bibliothèque TensorFlow et d'autres bibliothèques comme keras et L'apprentissage par transfert comme modèle est utilisé dans la phase de l'entraînement.

Les mots clé : Deep Learning, Traitement d'image, CNN, expressions faciales, VGG16, Python, IA

Abstract

Facial expression is the most common non-verbal method in which humans use one or more movements or positions of the muscles under the skin of the face.

Our project consists in providing a facial recognition approach based on deep learning and more particularly on convolution neural networks, including two CNN network architectures: VGG16 and Bottleneck. Our choice of these two architectures is motivated by their success in image recognition.

The neural network training is implemented using the Tensor Flow library and other libraries such as keras and Transfer Learning as a model is used in the training phase.

Listes des acronymes et abréviations

ANN : Neurones Artificiels Artificielle

API : Application Programming Interface

CAN : Convertisseur Analogique Numérique

CPU : Central Processing Unit

CNN : Réseau de Neurones Convolutifs

EER: Equal Error Rate

GPU: Graphics Processing Units

GNU:Gyeongsang National University

IA: Intelligence Artificielle

IOS: Internetwork-Operating System

LSTM: Long Short Term Memory

ML: Machine Learning

MLP: Multi Layers Perceptron

MAC: Multiply and Accumulate: Medium Access Control

OS: Operating System

RVB: Rouge Rert Blanc

RNN : Réseau de Neurones récursif

RN : Réseaux de Rorone

RNC : Réseau de Reurones Convolutif

TAJ : Traitement des Antécédents Judiciaires

Table de matières

Remerciements.....	I
Dédicaces.....	II
Résumé.....	V
Listes des acronymes et abréviations.....	VII
Introduction générale.....	1
CHAPITRE I Généralités sur la reconnaissance faciale	
I.1 Introduction.....	3
I.2 Historique.....	3
I.3. Définition de la reconnaissance faciale.....	4
I.4 Architecture générale d'un système de reconnaissance faciale.....	4
I.4.1 Détection de visage.....	4
I.4.2 Extraction des caractéristiques du visage.....	5
I.4.3 Comparaison des caractéristiques.....	5
I.5 L'utilisation opérationnelle de la reconnaissance faciale.....	5
I.6 Avantages et inconvénients de la reconnaissance faciale.....	6
I.7 Observation.....	7
I.8 Problématique.....	7
I.9. Reconnaissance des visages.....	7
I.10 Le fonctionnement de la reconnaissance de visage.....	8
I.10.1. Le monde physique : (L'extérieur)	8
I.10.2 .L'Acquisition de l'image.....	8
I.10.3. Les prétraitements.....	8
I.10.4. L'extraction de paramètres.....	9
I.10.5. La classification : (Modélisation).....	9
I.10.6. L'apprentissage.....	9
I.10.7. La décision	9
I.11 Vision par ordinateur.....	9
I.12 Traitement d'image.....	11
I.13 Intelligence Artificielle.....	11
Conclusion.....	12
CHAPITRE II Expression faciale et Deep Learning	
II.1 Introduction.....	13

II.2 Expression faciale.....	13
II.2.1 Les six expressions faciales universelles.....	13
II.2.2 Première liste des émotions de base.....	14
II.3 Les différentes méthodes développées pour la reconnaissance faciale.....	16
II.3.1 Approches locales.....	16
II.3.2 Approches globales.....	16
II.3.3 Approches hybrides.....	17
II.4 Réseaux de neurones.....	18
II.4.1 Pourquoi les réseaux de neurones sont-ils importants ?.....	20
II.4.2 Principe de fonctionnement et architectures des réseaux de neurones.....	20
II.5 L'apprentissage en profondeur (Deep Learning).....	21
II.5.1 Relation entre l'intelligence artificielle, le ML et le deep Learning.....	21
II.5.2 Fondements de l'apprentissage profond en vision.....	22
II.5.3 choix deep Learning ?.....	22
II.5.4 Les réseaux de neurones convolutifs.....	23
II.5.4.1 Architecture d'un réseau de neurones convolutif.....	24
II.5.4.2 Caractéristiques et avantage de CNN.....	24
II.5.5 Réseau de neurones récurrents.....	25
II.5.6 Modèle génératif.....	25
II.5.7 Exemples d'application de Deep Learning.....	26
II.6. L'apprentissage par transfert.....	27
II.7 Présentation l'architecture VGG16.....	31
II.7.1 Le Graphics Processing Units.....	32
Conclusion.....	32
 CHAPITRE III Implémentation des résultats expérimentaux	
III.1 Introduction.....	33
III.2 Objectifs et contraintes de travail.....	33
III.3 Environnement de développement.....	34
III.3.1 Environnement matériel (hardware).....	34
III.3.2 Environnement immatériel (software).....	34
III.3.3. Outils de développement.....	34
III.3.4 Bibliothèques utilisées.....	35
III.3.4.1 Installation des bibliothèques.....	37
III.4 partie du programme.....	40
III.4.1 Initialisation.....	40
III.4.2 Base de données utilisée.....	41

III.4.3 La distribution de Dataset en Train data, Validation data et Test data.....	42
III.4.4 Prétraitement des images.....	45
III.4.5Concaténation de toutes les data frames.....	46
III.4.6 Création de fonctionnalités de Bottleneck à partir du modèle.....	47
VGG-16. (Transfer Learning).....	47
III.4.7 Création du modèle.....	48
III.4.8 L'entraînement.....	49
III.4.9 Résultats de test.....	52
III.5 l'utilisation du model pour la reconnaissance en temps réel.....	56
III.6 Conclusion.....	59
CONCLUSION Général.....	60
BIBLIOGRAPHIE.....	61

Liste des Figure

CHAPITRE I Généralités sur la reconnaissance faciale

FIGURE I.1 : Système de reconnaissance faciale [19].....	4
FIGURE I.2 : Schéma de fonctionnement de la reconnaissance de visage.....	8
FIGURE I.3 : Les domaines de la vision par ordinateur [9].....	10
FIGURE I.4 : Schéma représentant l'approche de vision par ordinateur [10].....	11

CHAPITRE II Expression faciale et Deep Learning

FIGURE II.1 : Les six expressions faciales primaires proposées par Ekman[21].....	16
FIGURE II.2 : Nous donne la classification des algorithmes principaux utilisés en reconnaissance faciale.....	18
FIGURE II.3 : Model d'un neurone artificiel [4].	19
FIGURE II.4 : Structure d'un neurone artificiel [14].....	19
FIGURE II.5 : Différentes architectures des réseaux de neurones [24].....	21
FIGURE II.6 : La relation entre l'IA, le ML et le deep Learning.....	22
FIGURE II.7 : Comparaison entre la machine Learning et le Deep Learning.....	23
FIGURE II.8 : Réseau de neurones convolutifs [4].....	23
FIGURE II.9 : Architecture d'un CNN [24].....	24
FIGURE II.10 : résume les différents modèles du deep Learning.....	26
FIGURE II.11 : Différents process entre l'apprentissage classique et l'apprentissage par transfert [25].....	27
FIGURE II.11 : Représentation 3D de l'architecture de VGG-16.....	31

CHAPITRE III Implémentation des résultats expérimentaux

FIGURE III.1 : Les bibliothèques utilisées.....	37
FIGURE III.2 : Interface d'Anaconda Navigator.....	38
FIGURE III.3 : Fenêtre de la création d'un environnement.....	38
FIGURE III.4 : L'installation de bibliothèque keras.....	39
FIGURE III.5 : L'installation de bibliothèque keras.....	39
FIGURE III.6 : l'installation d'Open CV.....	40
FIGURE III.7 : Capture du code d'installation des bibliothèques.....	40
FIGURE. III.8 : Expressions humaines.....	41
FIGURE. III.9 : Expressions animées.....	42
FIGURE III.10 : Distribution d'images humaines.....	43

FIGURE III.11 : Distribution d'images animées.....	44
FIGURE III.12 : code de la conversion des images en niveau de gris.....	45
FIGURE III.13 : code du recadrage et redimensionnement des images.....	45
FIGURE III.14 : code de la concaténation de toutes les data frames.....	46
FIGURE III.15 : VGG-16 Bottleneck Features.....	47
FIGURE III.16 : code pour la création de bottleneck features pour les images d'entraînement.....	48
FIGURE III.17 : code pour la création du modèle.....	49
FIGURE III.18 : code pour l'entraînement du modèle.....	50
FIGURE III.19 : Journal de train.....	51
FIGURE III.20 : Précision de la perte d'époque.....	51
FIGURE III.21 : Images humaines de la matrice de confusion et de rappel.....	53
FIGURE III.22 : Images animées de la matrice de confusion et de rappel.....	55
FIGURE III.23 : code d'installation des bibliothèques.....	56
FIGURE III.24 : code de chargement du modèle.....	56
FIGURE III.25 : code de la définition d'interface.....	57
FIGURE III.26 : conversion de toutes les images en échelle de gris.....	57
FIGURE III.27 : Code de la détection du visage dans l'image.....	58
FIGUREI III.28 : Affichage de nos résultats.....	58
FIGURE III.29 : Visualisation de nos images traitées.....	59

Introduction générale

Les agents d'animation par ordinateur et les robots ajoutent une nouvelle dimension à l'interaction homme-machine, ce qui en fait un élément essentiel de la façon dont les ordinateurs affectent nos vies sociales dans les activités quotidiennes.

Le contact face-à-face est un processus en temps réel qui fonctionne sur une échelle de temps de quelques millièmes de seconde. Le niveau d'incertitude à cette échelle de temps est important, forçant les humains et les machines à s'appuyer sur des primaires sensorielles riches en sens plutôt que sur des inférences symboliques lentes.

Parmi les tâches résolues par la vision par ordinateur dans ce cas, se trouve le problème de la reconnaissance des expressions faciales en temps réel, ce problème est l'un des plus connus et largement discuté par de nombreux chercheurs.

Cependant, les principaux problèmes avec de tels systèmes sont la faible précision de détection et une forte demande d'ordinateur haute performance, ainsi que l'incapacité de certains systèmes à classer les expressions faciales dans l'une des sept expressions humaines de base. La reconnaissance automatique des expressions faciales suit le modèle général de Processus de reconnaissance de formes étape par étape: extraire des fichiers, apprentissage, classification et décision des données et des tests. Plusieurs travaux ont été développés dans le domaine de l'apprentissage automatique sur la reconnaissance des expressions faciales à l'aide de plusieurs algorithmes d'extraction des caractéristiques (statistiques ou structurelles) et de classificateurs.

Au début des années 60, des experts en intelligence artificielle se sont concentrés sur la possibilité de simuler les capacités sensorielles humaines car la vision est le sens humain le plus important, donc bien sûr, il y a eu beaucoup de travail dans cette direction. À la fin des années 80, Yan le Cun a développé un type de réseau particulier qui s'appelle le réseau de neurones convolutionnel CNN (convolutional neural network), ces réseaux sont une forme

particulière du réseau neuronal multicouche. Les réseaux CNN sont des structures de réseaux plus représentatives de la technologie d'apprentissage en profondeur 'deep Learning', ils ont eu un grand succès dans le traitement d'image et la reconnaissance. Mais ces méthodes étaient quelque peu difficiles à mettre en œuvre avec les ordinateurs.

Après la publication d'un article en 2005 expliquant l'intérêt des GPU pour l'apprentissage automatique, Ces puissants processeurs spécialisés, initialement conçus pour rendre des graphiques pour les jeux vidéo, se sont avérés très efficaces dans les calculs de réseaux neuronaux. Le but de cette thèse est de proposer une approche pour reconnaître les expressions faciales basées sur la méthode des réseaux de neurones convolutifs. Ce mémoire est organisé comme suit :

Le chapitre 1 présente l'état de l'art de la reconnaissance faciale dans sa forme la plus générale et introduit la vision par ordinateur ainsi que l'intelligence artificielle

Le chapitre 2 est consacré aux généralités sur la reconnaissance des expressions faciales, sur les réseaux de neurones à Convolution, sur le Deep Learning et à l'étude du modèle proposé.

Le chapitre 3 expose la mise en œuvre et le pilotage de notre système en présentant les outils et le logiciel que nous avons dû utiliser pour développer et réaliser le système.

Enfin, nous terminerons ce mémoire par une conclusion générale et quelques points de vue.

Chapitre I

Généralités sur la reconnaissance faciale

Chapitre I Généralités sur la reconnaissance faciale

I.1 Introduction

La communauté de recherche en vision par ordinateur a montré un vif intérêt pour l'analyse et la reconnaissance automatique des expressions faciales.

La reconnaissance faciale est une tâche que les humains accomplissent naturellement et sans effort dans leur vie quotidienne, et c'est la technique la plus courante et la plus populaire, et elle reste la plus acceptable car elle correspond à ce que les humains utilisent dans l'interaction visuelle.

La grande disponibilité d'ordinateurs puissants et peu coûteux ainsi que de systèmes informatiques a suscité un grand intérêt pour le traitement automatique des images numériques et vidéo dans de nombreuses applications, notamment l'identification, la surveillance et l'interaction humaine-Machine. La reconnaissance faciale, en tant que l'une des principales technologies biométriques, joue un rôle de plus en plus important dans la recherche.

I.2 Historique

La reconnaissance faciale automatique est un concept relativement nouveau, le premier système semi-automatisé de la reconnaissance faciale a été développé dans les années 1960, il nécessite à l'administrateur de localiser les yeux, les oreilles, le nez et la bouche sur la photo et de saisir les distances calculées et les ratios à un point de référence commun, qui ont ensuite été comparés aux données de référence.

Ce problème a été traité pour la première fois en 1988 par Francis Galton, celui-ci avait proposé la collection de profils faciaux en tant que courbes, trouver leur normes, et ensuite classer les nouveaux profils par leurs dérivation par rapport à cette norme, depuis une multitude d'approches ont été élaborées et plusieurs méthodes ont été adoptées pour résoudre ce problème [1].

En 1991, Türk et Pentland ont découvert que lorsque vous utilisez la technique 'Eigen faces' (ACP), l'erreur résiduelle peut être utilisée pour détecter un visage dans une image.

Cette technologie a été mise en essai en janvier 2001 lors de la finale du championnat de football américain SUPER BOWL en capturant des images de surveillance puis les comparées à une base de données numérique [2]. Aujourd’hui la reconnaissance faciale est utilisée dans plusieurs domaines.

I. 3 Définition de la reconnaissance faciale

La reconnaissance faciale est en fait l'un des nombreux types d'analyse d'images du domaine de la vision par ordinateur qui permet de constituer et de reconnaître automatiquement les personnes à partir d'images de leurs visages. La reconnaissance faciale utilise des algorithmes qui analysent les visages existants Images ou vidéos pour extraire un ensemble de caractéristiques distinctives [3].

I.4. Architecture générale d'un système de reconnaissance faciale

Un système de reconnaissance faciale peut être structuré comme le montre la figure I.1

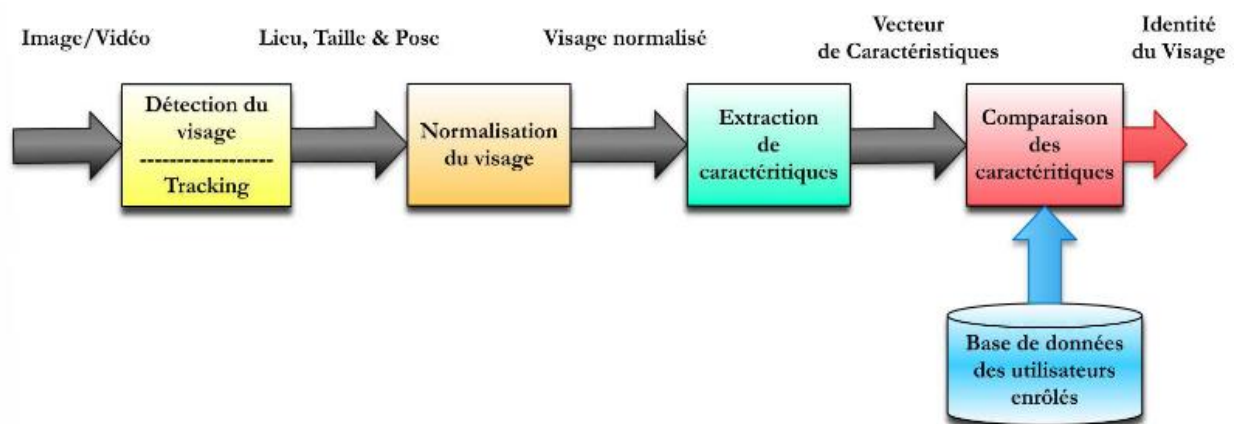


Figure I.1 : Système de reconnaissance faciale [4].

I.4.1. détection de visage

La détection des visages est une étape essentielle et importante, la reconnaissance faciale et finale ne pourrait pas être efficace que si l'étape de détection est meilleure. L'étape de détection commence par la capture de la scène qui contient un visage, puis extrait le visage de la photo prise par un détecteur afin de préserver une zone contenant des composants du visage [19].

I.4.2. Extraction des caractéristiques du visage

Nous pouvons le faire de deux manières diverses: La première consiste à extraire des régions entières du Visage, souvent réalisée avec une approche de reconnaissance universelle. La deuxième méthode dite de reconnaissance locale, extrait quelques points de différence des zones caractéristiques du visage, telles que la bouche et le nez [4].

I.4.3. Comparaison des caractéristiques

Selon les caractéristiques extraites précédemment, les algorithmes de comparaison diffèrent. On trouve dans la littérature plusieurs approches : calcul de distance, calcul de similarité. D'autres méthodes se basent sur la classification des caractéristiques par un seul classifieur (SVM, classifieur bayésien, etc.) ou par plusieurs (Adaboost) [4].

I.5. Utilisation opérationnelle de la reconnaissance faciale

L'utilisation opérationnelle de la reconnaissance faciales constate pour l'identification, en effet l'identification peut donner lieu à des applications nombreuses et plus diverses. On peut notamment citer les cas d'usages suivants, constatés ou envisagés dans certains pays.

La reconnaissance automatique de personnes présentes sur une image aux fins d'identifier par exemple ses relations sur un réseau social, à l'instar de Face book qui l'utilise, par comparaison entre l'image et les gabarits de toutes les personnes présentes sur le réseau ayant consenti à cette fonctionnalité, pour suggérer l'identification nominative de ces relations.

L'accès à des services, certains distributeurs de billets reconnaissant leurs clients, par comparaison entre un visage capté par une caméra et la base de données de visages détenue par la banque.

Le suivi du parcours d'un passager d'un service de transport à toutes les étapes de ce parcours, par comparaison entre le gabarit calculé en temps réel de toute personne se présentant à des portiques présents à certaines étapes du parcours (déposes bagages, portiques d'embarquement, etc.) et les gabarits des personnes enrôlées au préalable au sein du dispositif.

La recherche, dans une base de données comportant des photographies, de l'état civil d'une personne (victime, suspecte, etc.) non identifiée, ainsi que le permet par exemple en France le traitement TAJ (Traitement des antécédents judiciaires).

Le suivi des déplacements d'une personne dans l'espace public, par comparaison entre son visage et les gabarits biométriques des personnes circulant ou ayant circulé dans la zone surveillée, par exemple en cas d'oubli d'un bagage ou à la suite de la commission d'un délit.

La reconstitution du parcours d'une personne et de ses interactions successives avec des personnes tierces, par une comparaison des mêmes éléments mais réalisée en différé, pour identifier ses contacts par exemple.

L'identification sur la voie publique de personnes recherchées, par confrontation en temps réel de tous les visages captés à la volée par des caméras de vidéo protection et une base de données détenue par les forces de l'ordre.

Pour l'authentification, il s'agit d'effectuer une comparaison 1-to-1 où la personne présente un titre d'accès, tel qu'un badge, contenant son empreinte biométrique (l'image du visage). La caméra compare cette donnée biométrique avec le visage qu'elle va capturer. La personne va être authentifiée si les deux images présentent des points de similitudes suffisantes.

I.6. Avantages et inconvénients de la reconnaissance faciale

- ✓ Les avantages de la reconnaissance faciale sont que :
 - cette technologie est peu intrusive, et peut même se dérouler à l'insu de l'utilisateur,
 - certains de ces systèmes sont aujourd'hui capables de reconnaître des individus même avec des subterfuges (fausses moustaches, barbe, lunettes...),
 - elle ne demande aucune action de l'utilisateur,
 - il n'y a pas de contact physique,
 - cette technique est peu coûteuse
- ✓ Les inconvénients de la reconnaissance faciale sont que :
 - en fonction de l'inclinaison et de la distance, la technique de la vidéo présente des difficultés,

- le visage change beaucoup trop pour être un repère biométrique suffisamment fiable,
- c'est une technologie qui est sensible à l'environnement (éclairage, expression du visage),
- elle est sensible aux changements (barbe, moustache, chirurgie, piercing...) pour des appareils de base,
- les vrais jumeaux ne sont pas différenciés

I.7.Observation

La reconnaissance faciale est une technologie biométrique de reconnaissance des visages

La reconnaissance faciale est une technique informatique et probabiliste qui permet de reconnaître automatiquement une personne sur la base de son visage, pour l'authentifier ou l'identifier.

I.8. Problématique

Le problème de la reconnaissance de visages peut être formulé comme suit : Etant donnée une ou plusieurs images d'un visage, la tâche est de trouver ou de vérifier l'identité d'une personne par comparaison de son visage à l'ensemble des images de visages stockées dans une base de données. Il faut être capable de trouver de manière fiable un visage et ses repères. Ceci est essentiellement un problème de segmentation et de systèmes pratiques, la plupart des efforts sont déployés pour résoudre cette tâche. En fait, la reconnaissance réelle basée sur les caractéristiques extraites de ces repères faciaux est seulement une dernière étape mineure. Il existe deux types de problèmes de détection de visage:

- 1- Détection de visage dans les images.
- 2- Détection de visage en temps réel.

I.9. Reconnaissance des visages

Cette technologie qui représente quinze pour cent des applications consiste à faire une photographie plus ou moins évoluée pour en extraire un ensemble de facteurs qui se veulent propres à chaque individu. Ces facteurs sont choisis pour leur forte invariabilité et

concernent des zones du visage tel que le haut des joues, les coins de la bouche, l'écartement des yeux et la taille de la bouche par exemple.

I.10. Le fonctionnement de la reconnaissance de visage

Dans un système de reconnaissance de visages, une image suit -depuis son entrée- un processus bien précis pour arriver à déterminer l'identité du porteur de visages. Ce processus comporte plusieurs étapes qui peuvent être illustrées par le schéma suivant :



Figure I.2 : Schéma de fonctionnement de la reconnaissance de visage.

I.10.1. Le monde physique : (L'extérieur)

C'est le monde réel en dehors du système avant l'acquisition de l'image. Dans cette étape, on tient compte généralement de trois paramètres essentiels: L'éclairage, la variation de posture et l'échelle.

I.10.2 .L'Acquisition de l'image :

Cette étape consiste à extraire l'image de l'utilisateur du monde extérieur dans un état statique à l'aide d'un appareil photo ou dynamique à l'aide d'une caméra.

I.10.3. Les prétraitements :

Le rôle de cette étape est d'éliminer les parasites causés par la qualité des dispositifs optiques ou électroniques lors de l'acquisition de l'image en entrée, dans le but de ne conserver que les informations essentielles et donc préparer l'image à l'étape suivante. Elle est indispensable car on ne peut jamais avoir une image sans bruit à cause du background et de la lumière qui est généralement inconnue. Il existe plusieurs types de traitement et d'amélioration de la qualité de l'image, telle que : la normalisation, l'égalisation et le filtre médian. Cette étape peut également contenir la détection et la localisation du visage dans une image, surtout là où le décor est très complexe.

I.10.4. L'extraction de paramètres :

En plus de la classification, l'étape de l'extraction des paramètres représente le cœur du système de reconnaissance, elle consiste à effectuer le traitement de l'image dans un autre espace de travail plus simple et qui assure une meilleure exploitation de données, et donc permettre l'utilisation, seulement, des informations utiles, discriminantes et non redondantes.

I.10.5. La classification : (Modélisation)

Cette étape consiste à modéliser les paramètres extraits d'un visage ou d'un ensemble de visages d'un individu en se basant sur leurs caractéristiques communes. Un modèle est un ensemble d'informations utiles, discriminantes et non redondantes qui caractérise un ou plusieurs individus ayant des similarités.

I.10.6. L'apprentissage:

C'est l'étape où on fait apprendre les individus au système, elle consiste à mémoriser les paramètres, après extraction et classification, dans une base de données bien ordonnées pour faciliter la phase de reconnaissance et la prise d'une décision, elle est en quelque sorte la mémoire du système.

I.10.7. La décision :

C'est l'étape qui fait la différence entre un système d'identification d'individus et un autre de vérification. Dans cette étape, un système d'identification consiste à trouver le modèle qui correspond le mieux au visage pris en entrée à partir de ceux stockés dans la base de données, il est caractérisé par son taux de reconnaissance. Par contre, dans un système de vérification il s'agit de décider si le visage en entrée est bien celui de l'individu (modèle) proclamé ou il s'agit d'un imposteur, et il est caractérisé par son EER (equal error rate).

I.11. Vision par ordinateur :

La vision par ordinateur est un ensemble de techniques conçues pour l'interprétation automatique d'images. Au début de l'année 1964, les experts en intelligence artificielle se sont concentrés sur la possibilité de simuler les capacités sensorielles humaines.

Comme la vision est le sens humain le plus important, il est naturel que les travaux dans cette direction aient été nombreux. La démocratisation des appareils photo numériques a encore accéléré les possibilités offertes par le traitement des images, non seulement grâce

aux processus de correction intégrés dans les capteurs eux-mêmes, mais également grâce à la nouvelle facilité de post-traitement informatique.

La vision par ordinateur est l'émulation de la capacité du cerveau humain à interpréter une scène dynamique à partir de signaux visuels, en particulier pour identifier rapidement des objets et anticiper leurs mouvements. Ces fonctions cognitives sont essentielles pour la navigation en robotique autonome, mais profitent également aux applications d'exploration de données.

La numérisation automatique de vieux documents, la recherche d'images similaires en ligne ou la description audio automatique sont des exemples de tâches pouvant s'appuyer sur une brique d'interprétation des images. En particulier, les efforts de la communauté de la vision par ordinateur se sont concentrés sur la reconnaissance des objets dans les images, y compris leur identification et leur localisation.

De nombreux descripteurs ad hoc ont été introduits pour des applications aussi diverses que la détection de visage, la classification automatique de photographies d'animaux ou la reconnaissance optique de caractères [5].

Le dénominateur commun de ces œuvres est d'essayer de donner un sens aux images.

L'extraction de la sémantique à partir d'informations enregistrées sur une base de données constitue également l'objectif de la reconnaissance d'une scène.

Ce projet se situe à l'intersection de la vision par ordinateur et de l'apprentissage automatique. En particulier, nous proposons de mettre en œuvre des méthodes d'apprentissage en profondeur pour l'interprétation automatique des images de reconnaissance d'une scène [5].

Dans la [figure I.3] nous montrons les domaines les plus connus de la vision par ordinateur ainsi que la différence entre eux et leurs multiples niveaux.

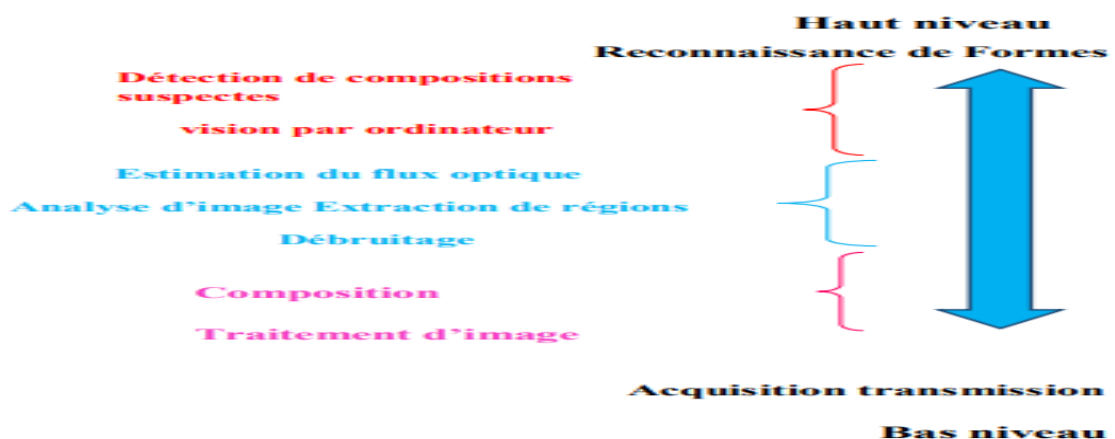


Figure I.3 : Les domaines de la vision par ordinateur. [5]

Dans la [figure I.4] nous donnons un exemple d'une approche de vision par ordinateur dans le domaine médical suivi d'un schéma qui montre les différentes étapes pour réaliser cette approche

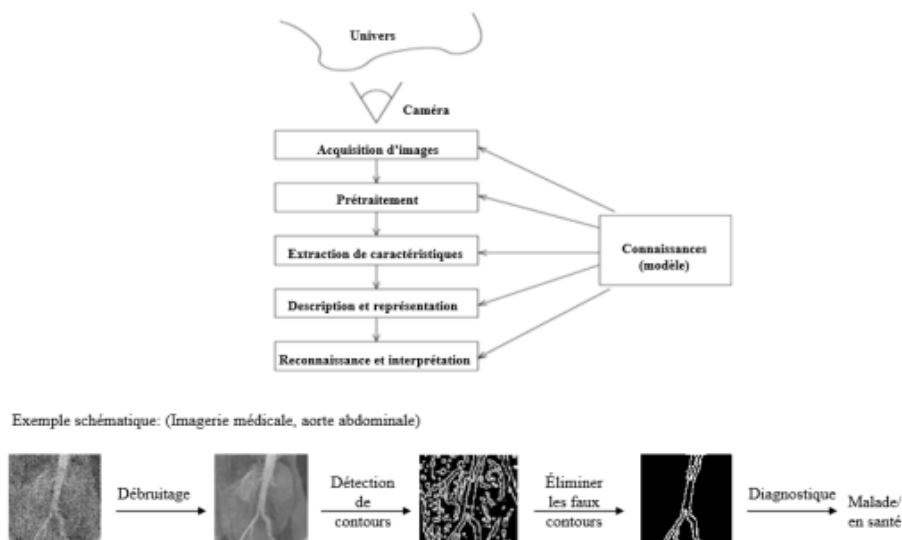


Figure I.4 Schéma représentant l'approche de vision par ordinateur [6]

I.12. Traitement d'images

Le traitement des images est l'ensemble des méthodes et techniques qui les exploitent, afin de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'extraire les informations jugées pertinentes [7].

D'une manière inverse, le traitement des images et la vision exploitent les connaissances et les techniques de l'intelligence artificielle pour l'adaptation et l'amélioration de la reconnaissance du monde extérieur par les machines intelligentes [8].

I.13. Intelligence Artificielle

L'intelligence pourrait être considérée comme un ensemble de techniques permettant aux machines d'exécuter des tâches et de résoudre des problèmes normalement réservés à l'homme et à certains animaux [9].

Les tâches d'intelligence artificielle sont parfois très simples pour l'homme, telles que reconnaître et localiser des objets dans une image, planifier les mouvements d'un robot pour rattraper un objet ou conduire une voiture. Ils nécessitent parfois une planification

complexe, telle que jouer aux échecs. Les tâches les plus complexes nécessitent beaucoup de connaissances et de bon sens, par exemple pour traduire un texte ou mener un dialogue. Ces dernières années, l'intelligence a presque toujours été associée à des capacités d'apprentissage. C'est grâce à l'apprentissage qu'un système intelligent capable d'effectuer une tâche peut améliorer ses performances avec l'expérience. C'est par l'apprentissage qu'il peut apprendre à effectuer de nouvelles tâches et à acquérir de nouvelles compétences [10]. Le domaine de l'IA n'a pas toujours considéré l'apprentissage comme essentiel à l'intelligence. Construire un système intelligent consistait autrefois à écrire un programme "à la main" pour jouer aux échecs, pour reconnaître les caractères imprimés ou pour poser un diagnostic médical à partir des symptômes. Mais cette approche «manuelle» a ses limites [11].

I.14 conclusion

Dans ce chapitre, nous avons présenté des généralités sur la reconnaissance faciale ainsi que sur la vision par ordinateur et son importance dans la reconnaissance des objets dans les images, comme la reconnaissance des expressions faciales. L'approche de reconnaissance est une approche de l'intelligence artificielle qui servira de base à la suite de notre travail.

Dans le prochain chapitre, nous introduirons l'apprentissage profond en citant les différents types de réseaux neuronaux et en détaillant plus particulièrement les réseaux neuronaux convolutifs.

Chapitre II
Expression faciale et Deep
Learning

II.1 Introduction

Dans ce chapitre, nous allons présenter quelques notions concernant les expressions faciales telles que leurs définitions, leurs différentes théories, et leurs composantes. Ensuite, nous décrivons un état de l'art sur les méthodes développées dans le domaine de la reconnaissance des expressions faciales et plus particulièrement nous abordons la notion du deep Learning et ses différentes architectures existantes.

II.2 Expression faciale

Une expression faciale est une manifestation visible d'un visage qui reproduit l'état d'esprit, l'activité cognitive et physiologique, la personnalité et la psychopathologie d'une personne. Elle repose sur trois principales caractéristiques influençant la nature de l'expression faciale à savoir : la bouche, les yeux et les sourcils, ainsi caractérisant une déformation du visage par un changement perceptible visuel [12].

Lorsque l'on parle d'expressions faciales et des sept émotions universelles on ne peut pas passer à côté des micro-expressions. Les micro-expressions sont de rapides expressions faciales qui surviennent pour deux raisons : Un refoulement émotionnel inconscient, ou une dissimulation volontaire de ses émotions.

Mais quand on les observe, que leur apparition vient d'une dissimulation volontaire ou bien d'un refoulement, elles s'expriment de façon identique sur le visage. Ce sont donc de formidables indices qui apparaissent sur une personne quand elle a beaucoup à gagner ou à perdre. Ce sont donc quand vous les repérez, des indices de vérité. [4]

II.2.1 Les six expressions faciales universelles

Dès le vingtième siècle, Ekman et Friesen définissent six émotions de base sur la base d'une étude culturelle et psychologique. , Ils ont expliqué que les humains perçoivent certaines émotions de base de la même manière indépendamment de la culture de

l'individu. Les tentatives pour décrire l'émotion humaine reposent sur l'hypothèse que les humains expriment universellement un ensemble d'émotions primaires discrètes qui incluent le bonheur, la tristesse, la peur, la colère, le dégoût et la surprise (voir figure II.1). Principalement à cause de sa simplicité et de revendication de l'universalité, cette hypothèse des émotions primaires universelle a été largement exploitée dans l'informatique affective.

II.2.2 Première liste des émotions de base

La liste des émotions de base établie en 1972, a pu être définie :

La peur

Elle met le corps en alerte maximale, prêt à fuir. Elle fait relever les sourcils, soulever les paupières supérieures en découvrant le blanc des yeux et en plissant le front, ouvrir la bouche et relever les lèvres, couler la sueur sur le front ou les mains. Le cœur et la respiration courte s'accélèrent, le corps s'immobilise, le tonus musculaire augmente, la bouche est sèche, la peau pâlit car le sang qu'elle contient est détourné vers les muscles.

La colère

Elle met le corps en état d'agression. Les sourcils sont tirés vers le bas, provoquant des rides verticales au-dessus du nez, les yeux mi-clos semblent furieux, le visage devient rouge. La bouche reste fermée ou entrouverte, mais les lèvres retroussées découvrent les dents comme un animal prêt à mordre. Le cœur et la respiration ample s'accélèrent, le corps ramassé sur lui-même par des muscles tendus semble prêt à bondir.

Le dégoût

Il s'exprime souvent par des yeux fermés, les sourcils tombants, de gros plis sur le front et au-dessus du nez, des pommettes et des joues bombées. Les lèvres pincées peuvent laisser sortir le bout de la langue. La paume des mains se tourne vers l'avant, comme pour repousser un objet. Une sensation de nausée est par ailleurs possible.

✚ La tristesse

Paupières baissées, les yeux rougis brillent ou larmoient. Seul l'angle interne des sourcils se soulève, alors que les coins de la bouche pointent vers le bas. Le cœur ralentit, la respiration est ample et lente, entrecoupée de profonds soupirs et de petites inspirations très superficielles. Le corps plutôt immobile semble replié sur lui-même, le tonus musculaire est faible.

✚ La joie

Un large sourire éclaire le visage et la lèvre supérieure retroussée découvre les dents du haut, sauf quand le rire ouvre grand la bouche et laisse voir toute la dentition. Les yeux mi-clos sont soulignés par des rides de la patte d'oie à leur angle externe et des poches sous la paupière inférieure. Les joues sont bombées par la contraction des muscles peauciers. La respiration est lente et ample, entrecoupée de petites pauses, le rythme cardiaque est souvent accéléré.

✚ La surprise

Les deux paupières grandes ouvertes laissent voir une large zone du blanc des yeux écarquillés. La bouche est entrouverte par la contraction des joues et le retroussement de la lèvre supérieure, mais l'aspect est plus proche d'un sourire forcé que de l'expression de la peur. Après une brève accélération, le cœur ralentit rapidement alors que la respiration est brièvement bloquée en inspiration.

Une septième expression faciale, associée au **mépris**, est aussi reconnue universellement[3].

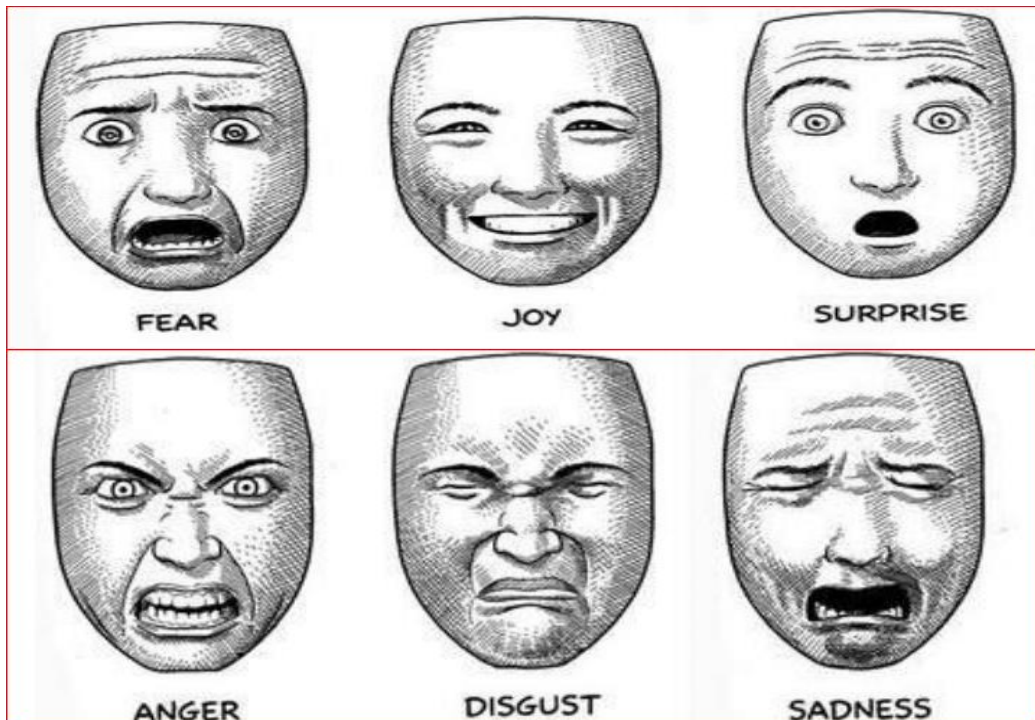


FIGURE II.1 Les six expressions faciales primaires proposées par Ekman [13].

II.3 Les différentes méthodes développées pour la reconnaissance faciale

Ils existent différentes approches qui ont été développées dans le domaine de la reconnaissance des expressions faciales. Elles peuvent être séparées en deux grandes familles, à savoir : les méthodes globales et les méthodes locales.

II.3.1 Approches locales

Les approches locales de la reconnaissance de visages sont basées sur des modèles et reposent sur un traitement séparé des différentes régions de l'image du visage. Les modèles utilisés reposent sur les connaissances que l'on possède à priori de la morphologie des visages. La plupart du temps, cela implique la détection/extraction de caractéristiques faciales locales [3].

II.3.2 Approches globales

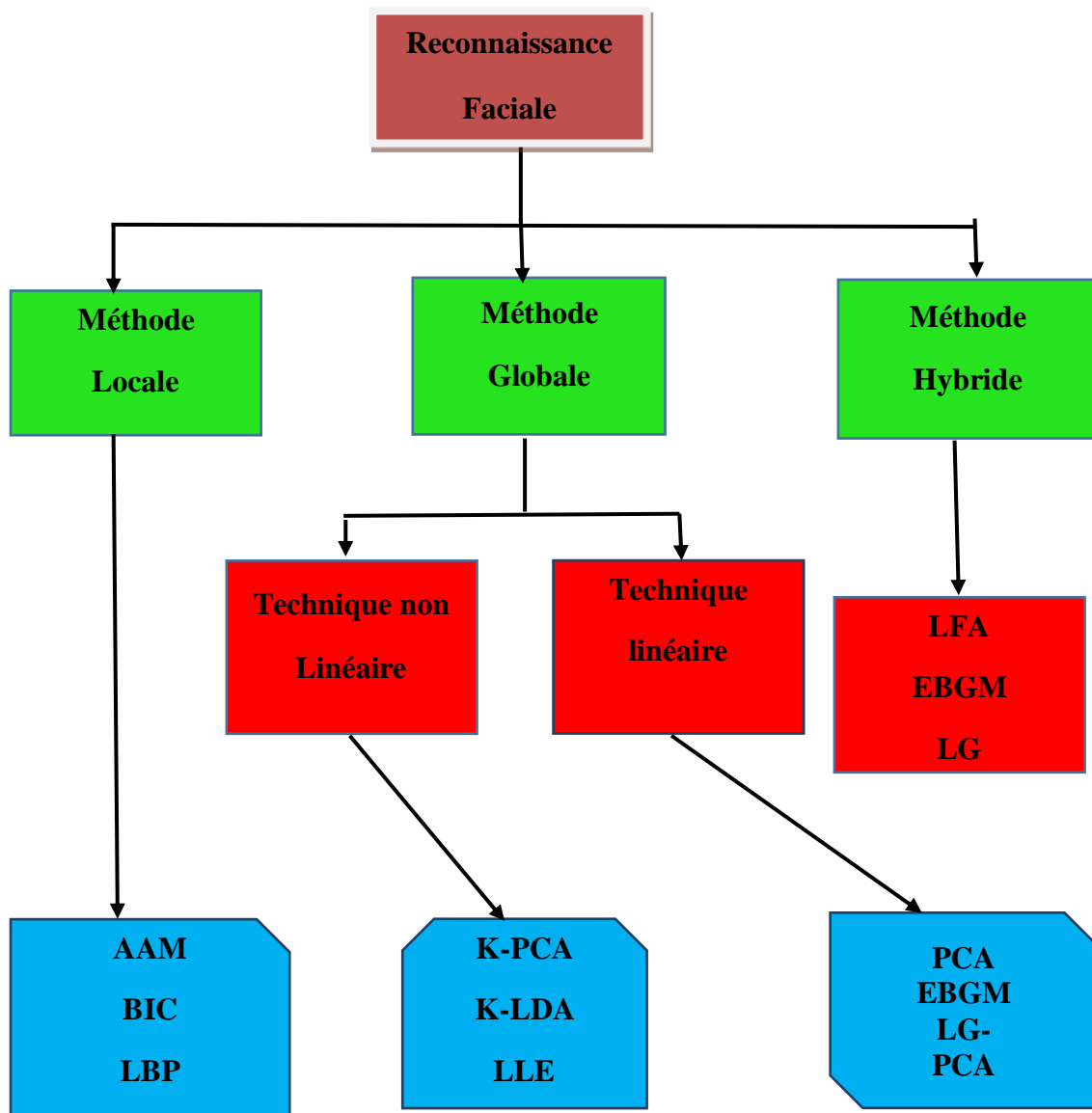
Les approches globales prennent l'image du visage comme un tout et utilisent des techniques d'analyse statistique bien connues. L'idée est généralement de projeter l'image d'entrée du visage, préalablement vectorisée, dans un espace de plus faible dimension, où

la reconnaissance est supposée être plus aisée. La projection est souvent conçue pour ne pas sélectionner que les caractéristiques importantes et suffisamment discriminantes pour différencier les personnes entre elles. Un des avantages des méthodes globales est qu'elles sont rapides à mettre en œuvre, les calculs reposant sur des opérations matricielles relativement simples. Cependant, étant donné qu'elles considèrent le visage comme un tout, elles sont sensibles aux conditions de luminosité, de pose ou encore d'expression faciale.

La plupart de ces méthodes réalisent une analyse de sous-espaces de visage. Cette analyse découle de la constatation d'un fait relativement simple : la classe des visages réside dans un sous-espace de l'espace de l'image d'entrée. Prenons par exemple une image de taille [100,100] en niveaux de gris. Le nombre de configurations possibles est égal à 25610000 Configuration. Cependant, parmi toutes ces configurations possibles, seule une petite partie correspond aux visages. L'information contenue dans les images de visages est donc très redondante, la dimension de ces images peut donc être réduite en se concentrant uniquement sur ce qui nous intéresse. Le sous-espace est souvent appelé espace de visages. Les méthodes globales peuvent se décomposer en deux types de techniques : les techniques linéaires et les techniques non linéaires [3].

II.3.3 Approches hybrides

Les méthodes hybrides résultent de l'association des méthodes locales et des méthodes globales. Elles combinent la détection de caractéristiques locales avec l'extraction de caractéristiques globales. Ces techniques essaient finalement de tirer partie des avantages des deux types de méthodes citées plus haut [3].



La figure II.2. Nous donne la classification des algorithmes principaux utilisés en reconnaissance faciale.

II.4 Réseaux de neurones

Le réseau de neurones artificiels (ANN) est inspiré des neurones biologiques. L'élément de base de l'ANN est un neurone artificiel. Chaque neurone artificiel possède des entrées qui sont pondérées, suivies d'une fonction de transfert ou d'une fonction d'activation, le neurone produit une valeur d'échelle. Un exemple de modèle neural est illustré par la (Figure II.3)

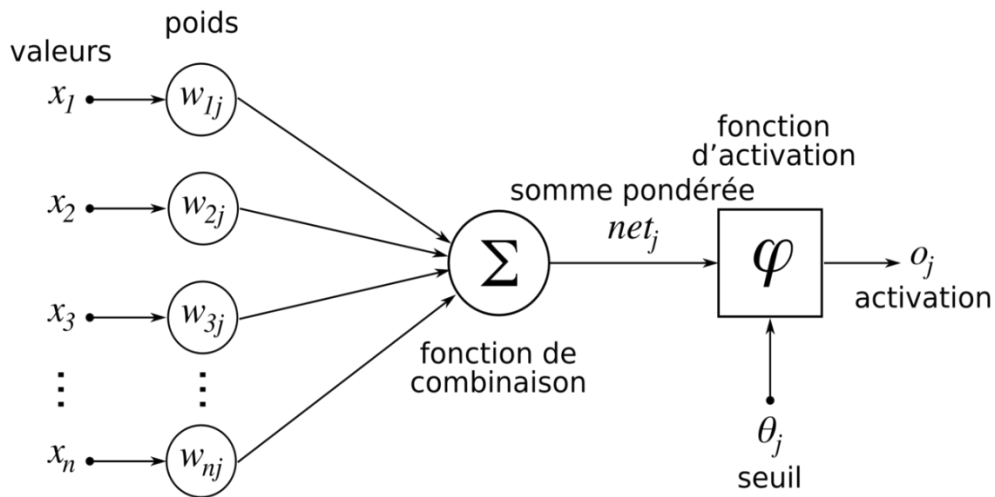


Figure II.3 Modèle d'un neurone artificiel [12].

Basé sur un neurone artificiel, différents empilements de neurones forment un auto-encodeur [16]. Il existe plusieurs types de réseaux de neurones les réseaux récurrents RNN et les réseaux à convolution CNN. L'architecture de base est illustrée à la Figure II.4.

L'Auto-encodeur : est un réseau de neurones artificiels utilisé pour l'apprentissage automatique d'analyse et de reconnaissance dans une image. Le CNN utilisé dans plusieurs architectures de segmentation et de traitement d'image, s'inspire des processus biologiques de l'animal. Grâce à l'excellente structure, le CNN a obtenu des résultats remarquables en matière de classification, de segmentation et de détection d'images [22] [15].

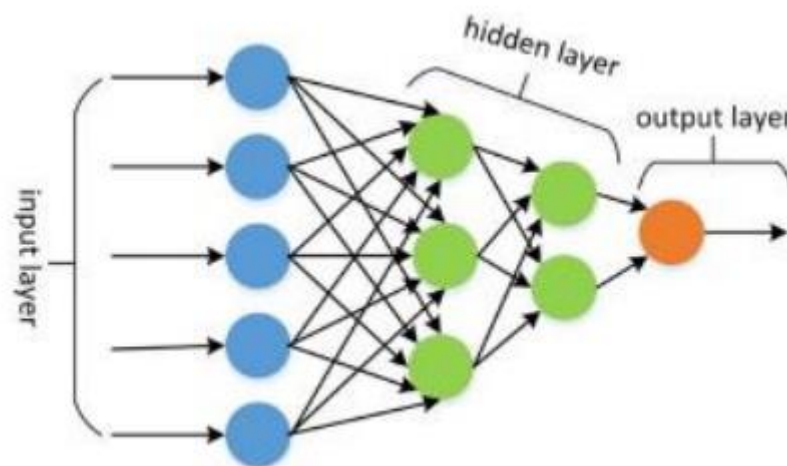


Figure II.4 . Structure d'un réseau de neurones artificiels [10].

II.4.1 Pourquoi les réseaux de neurones sont-ils importants ?

Les réseaux de neurones sont particulièrement bien adaptés à la reconnaissance de formes pour identifier et classer des objets ou des signaux dans les systèmes de la parole, de la vision et du contrôle. Ils peuvent également être utilisés pour la prédiction et la modélisation de séries chronologiques.

Voici quelques exemples d'utilisation des réseaux de neurones :

- ❖ Les compagnies d'électricité anticipent avec précision la charge sur leurs réseaux électriques pour assurer la fiabilité et optimiser l'efficacité des générateurs d'électricité qu'ils exploitent.
- ❖ Les guichets automatiques bancaires acceptent de façon fiable les dépôts bancaires via la lecture du numéro de compte et du montant du dépôt sur votre chèque.
- ❖ Les pathologistes se fient aux applications de détection du cancer pour les orienter dans la classification des tumeurs, de bénignes à malignes, en fonction de l'uniformité de la taille des cellules, de l'épaisseur de la masse, de la mitose et d'autres facteurs [23].

II.4.2 Architectures des réseaux de neurones

Un réseau de neurones permet de résoudre de nombreux problèmes d'intelligence artificielle. Schématiquement, le fonctionnement est le suivant :

- Les neurones sont organisés en couches successives : chaque neurone reçoit une information issue des neurones de la couche qui précède.
- Chacune de ces informations est pondérée : elle est multipliée par une valeur qui lui confère un "poids" particulier.
- Les entrées ainsi pondérées sont additionnées.
- Elles sont traitées par une fonction objective dont le but est d'adapter la valeur de sortie à une plage de valeurs.
- La valeur de sortie issue de cette fonction constitue l'entrée de l'ensemble des neurones de la couche suivante [18].

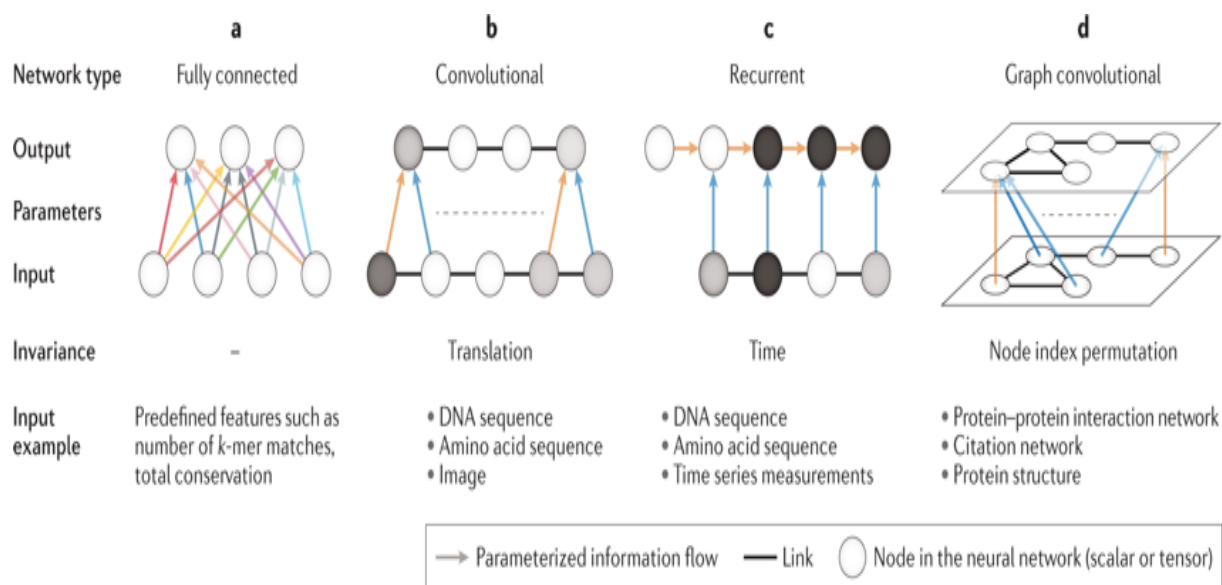


Figure II.5 . Différentes architectures des réseaux de neurones[18].

II.5 L'apprentissage profond (Deep Learning)

L'apprentissage profond permet aux modèles informatiques composés de plusieurs couches de traitement d'apprendre des représentations des données avec plusieurs niveaux d'abstraction.

Il permet de découvrir des structures complexes au sein de grands ensembles de données en utilisant la rétro-propagation qui dicte à la machine comment celle-ci doit modifier ses paramètres internes pour calculer la représentation de chaque couche à partir de la représentation de la couche précédente.

II.5.1 Relation entre l'intelligence artificielle, le ML et le Deep Learning

Le Deep Learning est un nouveau domaine de recherche de la ML, qui a été introduit dans le but de rapprocher le ML de son objectif principal à savoir : l'intelligence artificielle. Il concerne les algorithmes inspirés par la structure et du fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données.

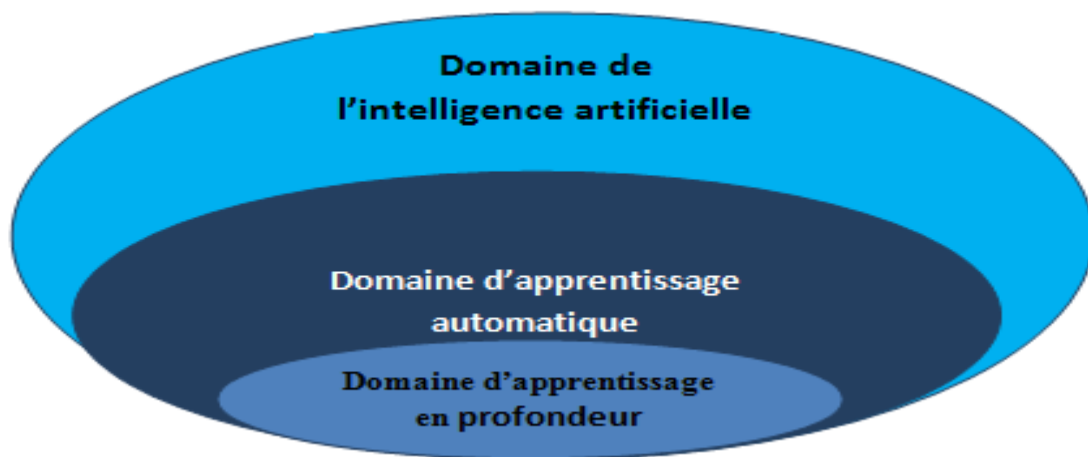


Figure II.6 . La relation entre l'IA, le ML et le Deep Learning.

II.5.2 Fondements de l'apprentissage profond en vision

Le domaine de l'apprentissage profond propose des méthodes efficaces pour obtenir des systèmes intelligents pour effectuer des tâches de classification, de régression ou de regroupement. En augmentant le nombre de neurones et le nombre de couches dans un réseau, la fonction de décision augmente elle-même en complexité.

Spécifiquement dans le domaine de l'apprentissage supervisé, les tâches qui nécessitent de réaliser une correspondance entre une donnée en entrée et une représentation choisie en sortie peuvent être accomplies à l'aide des réseaux de neurones. La précision et l'exactitude de cette fonction de correspondance retrouvée demeurent limitées par la quantité et la qualité des données d'entraînement.

II.5.3 Choix du Deep Learning ?

- 1) Les fonctionnalités conçues manuellement sont souvent sur-spécifiées, incomplètes et prennent beaucoup de temps à concevoir et à valider.
- 2) Les fonctionnalités apprises sont faciles à adapter, rapides à apprendre.
- 3) L'apprentissage en profondeur fournit un cadre très souple, universel et pouvant être appris pour représenter des informations mondiales, visuelles et linguistiques.
- 4) Peut apprendre à la fois sans surveillance et sous surveillance.

- 5) Apprentissage efficace du système conjoint de bout en bout.
- 6) Utiliser de grandes quantités de données d'entraînement

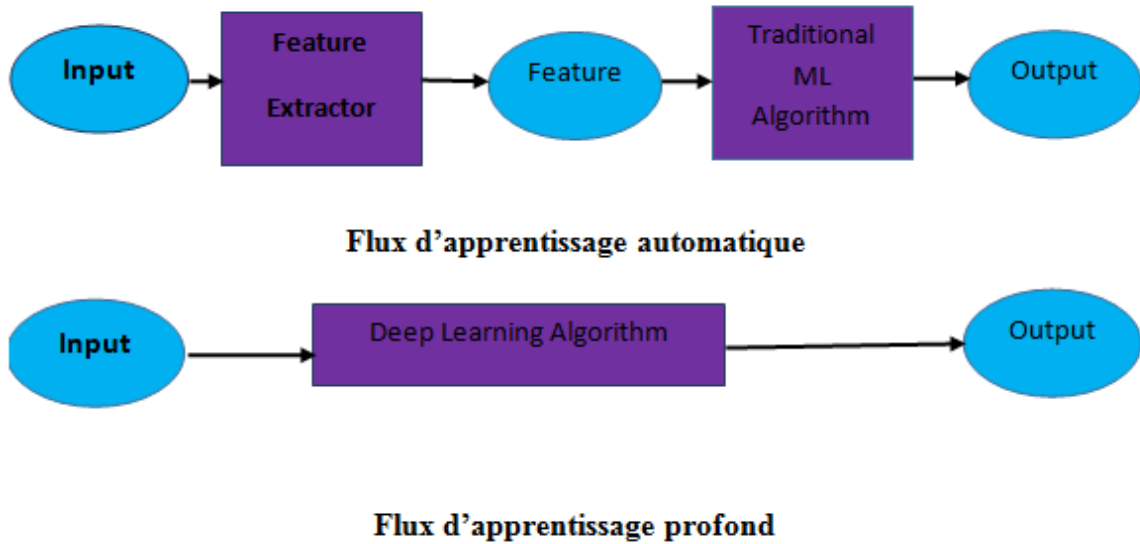


Figure II.7. Comparaison entre le machine Learning et le Deep Learning.

II.5.4 Les réseaux de neurones convolutifs

Les réseaux de neurones convolutifs sont directement inspirés du cortex visuel des vertébrés. Un réseau de neurones à convolution, appelé aussi convnet, ou encore CNN se représente en général sous la forme suivante :

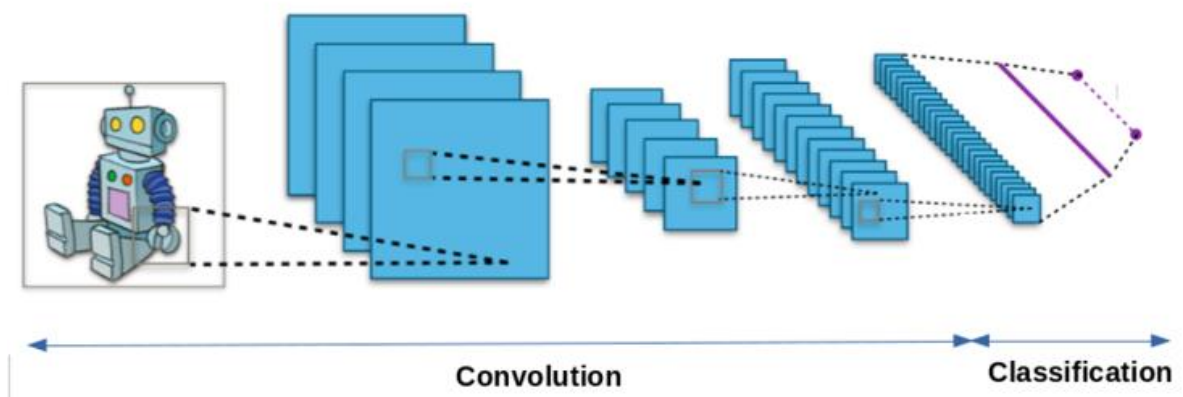


FIGURE II.8. Réseau de neurones convolutifs [12].

II.5.4.1 Architecture d'un réseau de neurones convolutifs.

On distingue deux parties, une première partie que l'on appelle la partie convolutive du modèle et la seconde partie, que l'on va appeler la partie classification du modèle qui correspond à un modèle MLP.

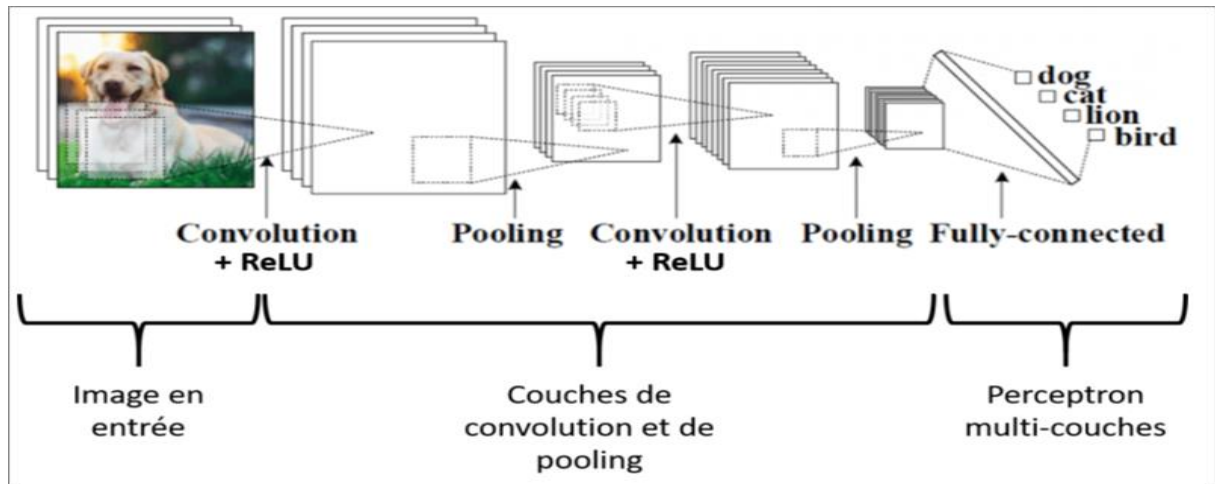


FIGURE II.9 Architecture d'un CNN [18].

II.5.4.2 Caractéristiques et avantages des CNN

Un avantage majeur des CNN est l'utilisation d'un poids unique associé aux signaux entrants dans tous les neurones d'un même noyau de convolution. Cette méthode réduit l'empreinte mémoire, améliore les performances et permet une invariance du traitement par translation. C'est le principal avantage du CNN par rapport au perceptron multicouche, qui, lui, considère chaque neurone indépendant et affecte donc un poids différent à chaque signal entrant.

Lorsque le volume d'entrée varie dans le temps, il devient intéressant de rajouter un paramètre le long de l'échelle de temps dans le paramétrage des neurones. On parlera dans ce cas de réseau neuronal à retard temporel.

Comparés à d'autres algorithmes de classification d'images, les CNN utilisent relativement peu de prétraitement. Cela signifie que le réseau est responsable de faire évoluer tout seul ses propres filtres, ce qui n'est pas le cas d'autres algorithmes plus traditionnels. L'absence de paramétrage initial et d'intervention humaine est un atout majeur des CNN.

II.5.5 Les Réseaux de neurones récurrents

Les RNN sont des réseaux de neurones dans lesquels l'information peut se propager dans les deux sens, y compris des couches profondes aux premières couches. En cela, ils sont plus proches du vrai fonctionnement du système nerveux, qui n'est pas à sens unique. Ces réseaux possèdent des connexions récurrentes au sens où elles conservent des informations en mémoire : ils peuvent prendre en compte à un instant t un certain nombre d'états passés. Pour cette raison, les RNNs sont particulièrement adaptés aux applications faisant intervenir le contexte, et plus particulièrement au traitement des séquences temporelles comme l'apprentissage et la génération de signaux, c'est à dire quand les données forment une suite et ne sont pas indépendantes les unes des autres. Néanmoins, pour les applications faisant intervenir de longs écarts temporels, cette « mémoire à court-terme » n'est pas suffisante.

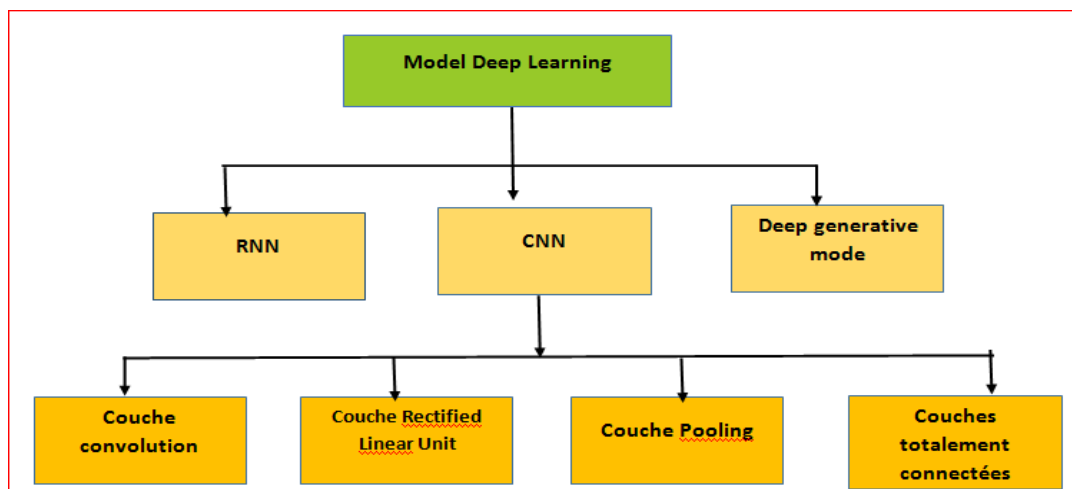
En effet, les RNNs « classiques » ne sont capables de mémoriser que le passé dit proche, et commencent à « oublier » au bout d'une cinquantaine d'itérations environ. Ce transfert d'information à double sens rend leur entraînement beaucoup plus compliqué, et ce n'est que récemment que des méthodes efficaces ont été mises au point comme les LSTM. Ces réseaux à large « mémoire court-terme » ont notamment révolutionné la reconnaissance de la voix par les machines ou la compréhension et la génération de texte.

D'un point de vue théorique, les RNNs ont un potentiel bien plus grand que les réseaux de neurones classiques : des recherches ont montré qu'ils sont « Turing-complet », c'est à dire qu'ils permettent théoriquement de simuler n'importe quel algorithme. Cela ne donne néanmoins aucune piste pour savoir comment les construire pour cela dans la pratique [18].

II.5.6 Modèle génératif

Si les modèles discriminatifs comme (CNN, RNN) sont utilisés pour prédire les données du label et de l'entrée, le modèle génératif décrit comment générer les données, il apprend et fait des prédictions en utilisant la loi de Bayes. Cependant les modèles génératifs sont capables de bien plus que la simple classification comme par exemple générer de nouvelles observations. Voici quelques exemples de modèle génératif :

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



La figure II.10 résume les différents modèles du deep Learning.

II.5.7 Exemples d'application du Deep Learning

Le deep Learning est utilisé dans de nombreux domaines :

- reconnaissance d'image,
- traduction automatique.
- voiture autonome.
- diagnostic médical.
- recommandations personnalisées.
- modération automatique des réseaux sociaux.
- prédiction financière.
- identification de pièces défectueuses.
- robots intelligents.

II.6. L'apprentissage par transfert

La façon la plus populaire de résoudre un problème en appliquant l'apprentissage profond est la Formation à partir de zéro. Dans cette stratégie, tous les paramètres de la couche sont initialisés de manière aléatoire. Pendant la formation, les valeurs de paramètres seront mises à jour dans la phase rétrospective afin de fournir les meilleures valeurs pour l'exactitude du modèle. L'utilisation des valeurs aléatoires des paramètres est extrêmement risquée. le modèle en besoin d'un long temps d'apprentissage pour optimiser ces valeurs. De plus, si nous formons les sur un ensemble de données qui ne pouvait pas fournir suffisamment d'échantillons pour le processus de formation, le pourrait souvent tomber dans les minimaux locaux. Ces problèmes peuvent être évités par la formation de modèle sur un grand ensemble de données, qui comprend plusieurs milliers d'échantillons. Cependant, Cette condition est difficile à remplir dans la pratique, car il faut recueillir des exemples dans le monde réel et c'est coûteux et cela prend beaucoup de temps. Heureusement, nous pouvons réutiliser les valeurs des paramètres d'une autre expérience pour initialiser les valeurs des paramètres lorsque deux tâches ont une relation. C'est connu sous le nom d'apprentissage par transfert, une autre façon d'appliquer l'apprentissage profond [19].

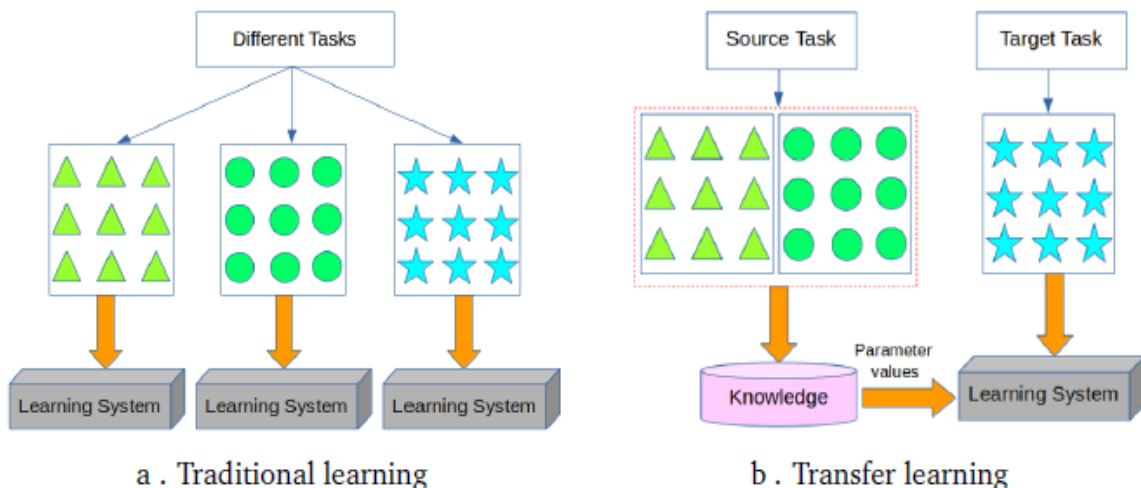


Figure II.11 : Différents procès entre l'apprentissage classique et l'apprentissage par transfert [19].

L'apprentissage par transfert est une technique d'apprentissage approfondie où un modèle formé sur une tâche (appelée tâche source) est réorienté vers une deuxième tâche connexe (appelée tâche cible). Il s'agit de l'amélioration de l'apprentissage dans la tâche cible entrainant les connaissances, qui ont été acquises lors de la tâche source. L'une des plus célèbres est l'apprentissage par transfert qui utilise l'ensemble de données ImageNet pour former différents modèles populaires, par exemple AlexNet, ResNet. Ensuite, les valeurs des paramètres de ces modèles sont fournies pour servir de modèle préétabli à résoudre les divers problèmes. La figure II.11 montre une comparaison entre 2 stratégies à appliquer l'apprentissage en profondeur. Dans la stratégie traditionnelle, chaque modèle est construit et formé pour chaque couche (Figure II.11.a). A l'opposé, la stratégie d'apprentissage par transfert utilise les connaissances sur la tâche source à utiliser dans le système d'apprentissage pour résoudre la tâche cible lorsque Les tâches source et cible ont une relation (figure II.11 .b). En pratique, l'apprentissage par transfert est principalement ciblé sur 2 méthodes :

***L'utilisation des réseaux CNN comme extracteur de caractéristiques fixes :** Prenez un CNN préformé sur un grand ensemble de données, puis retirez la dernière couche entièrement connectée et utilisez les autres couches de CNN comme un extracteur pour le nouveau jeu de données.

***Mise au point d'une CNN :** Cette situation est la même que la première. Cependant, non seulement elle remplace et recycle la dernière couche, mais aussi affine les poids des modèles préformés en étendant la rétropropagation. On peut noter que pour réutiliser un modèle préformé, les paramètres ont été adaptés entre deux tâches. Ces paramètres peuvent être la taille des images en entrée, le nombre de sorties ou les paramètres de chaque couche. Comme d'habitude, les valeurs des paramètres à chaque couche, par exemple le rembourrage ou les valeurs de foulée, sont sélectionnées pour changer leurs valeurs afin de déclarer les différences entre les deux tâches.

Le choix d'une stratégie d'apprentissage par transfert dépend sur divers facteurs, mais le plus important est la taille de l'ensemble de données cible (petite ou grand) et sa similarité avec l'ensemble de données source. La caractéristique d'un réseau CNN est le caractère générique extrait par les premières couches, tandis que les caractéristiques spécifiques sont données par les dernières couches. Il existe donc quatre grands scénarios :

1. L'ensemble de données cible est petit et similaire à l'ensemble de données source. Ce n'est pas une bonne idée d'affiner le CNN en raison d'inquiétudes quant au surdimensionnement, car les données sont peu nombreuses. Étant donné que l'est similaire aux données originales, nous attendons des caractéristiques de plus haut niveau dans le CNN pour être également lié à cet ensemble de données. Ainsi, la meilleure idée pourrait être de former un sur les fonctionnalités de CNN.
2. L'ensemble de données cible est important et similaire à l'ensemble de données source. Comme nous disposons de plus de données, nous peut croire que nous n'allons pas nous surpasser lorsque nous aurons mis au point l'ensemble du réseau.
3. L'ensemble de données cible est petit et très différent de l'ensemble de données source. Il semble que la formation le classificateur en haut du réseau n'est pas le meilleur moyen de remplacer le réseau devrait être formé à un niveau plus élevé de l'architecture.
4. L'ensemble de données cible est important et très différent de l'ensemble de données source. Comme nous avons un énorme de données, nous pouvons complètement former CNN à partir de zéro. Cependant, il est bénéfique d'initialiser les poids à partir d'un modèle préformé pour stimuler le processus d'apprentissage et pour gagner du temps. Ainsi, nous devrions disposer de suffisamment de données et de confiance pour affiner à travers l'ensemble du réseau.

En pratique, l'apprentissage par transfert a été appliqué dans différentes applications : Ng.Hong-Wei et ses collaborateurs ont appliqué l'apprentissage par transfert en deux étapes pour reconnaître les émotions sur les visages humains. En partant d'un processus générique de préformation de deux architectures CNN (AlexNet et VGG) sur le jeu de données ImageNet, La première étape consiste à affiner les modèles préformés sur un ensemble de données d'expressions faciales.

La deuxième étape se déroule ensuite en se basant uniquement sur la formation d'une partie de la reconnaissance des émotions dans la nature (EmotiW), en adaptant les poids du réseau aux caractéristiques du sous-défi de la reconnaissance statique des expressions faciales dans la nature (SFEW). Leur projet expérimental Les résultats ont montré que l'approche de réglage fin en cascade permet de mieux que l'étape unique de réglage fin des ensembles de données combinés, avec une amélioration jusqu'à 16 %. Girshick et ses collaborateurs ont mis au point le modèle préformé de AlexNet sur le jeu de données PASCAL pour effectuer la détection et la segmentation d'objets tâches. Les résultats

obtenus ont prouvé que l'utilisation de l'apprentissage par transfert de manière significative améliorée par rapport aux autres méthodes sans CNN. Shin.Hoo-Chang.ala appliqué la CNN profonde sur des images médicales pour détecter le ganglion lymphatique thoraco-abdominal et de classer la maladie pulmonaire interstitielle. Ils ont choisi les deux modèles célèbres : AlexNet et GoogLeNet. Leurs études ont porté sur trois facteurs importants de l'emploi un CNN profond pour résoudre un problème : formation à partir de zéro, utilisation de CNN "sur étagère", et apprentissage par transfert.

En apprenant à partir de zéro, tous les paramètres des modèles CNN est aléatoirement initialisé et formé sur l'ensemble des données. Dans l'apprentissage par transfert, ils ont suivi une hypothèse que : "malgré la disparité entre les images naturelles, les CNN formés sur le réseau ImageNet, bien connu et à grande échelle, peuvent encore être transférés à rendre plus efficaces les tâches de reconnaissance d'images médicales". C'est pourquoi ils ont utilisé le préformées des modèles AlexNet et Google Net, pour affiner leur l'ensemble des données d'images médicales.

La performance de l'utilisation de données "standard" a également été prise en compte en utilisant le modèle préformé d'AlexNet comme extracteur de caractéristiques et en ne formant que les le classificateur final. À la fin des processus, ils ont constaté que l'apprentissage par transfert est la plus performante. S. Lin et ses collaborateurs ont proposé le transfert et le réseau spécialisé (TS-Net) qui fusionne les connaissances générales et spécialisées en combinant un FaceNet de transfert et un FaceNet spécialisé.

Le premier est obtenu par la mise au point du réseau GoogLeNet préformé pour transférer les connaissances en matière de reconnaissance d'objets pour la reconnaissance des visages, et ce dernier est formé sur les correctifs globaux et locaux pour fournir les connaissances spécialisées discriminantes pour la reconnaissance des visages.

Afin d'évaluer les différentes stratégies d'apprentissage approfondi et d'améliorer les résultats, nous avons essayé de transférer les connaissances d'AlexNet sur le jeu de données d'ImageNet pour affiner nos images, mais les résultats n'ont pas été satisfaisants. Cependant, nous pensons que l'apprentissage par transfert est une bonne solution pour notre application basée sur que nous avons étudiés. Pour conserver l'idée de l'apprentissage par transfert, nous avons choisi un autre scénario pour l'appliquer [19].

II.7 Présentation de l'architecture VGG16

VGG16 est un modèle de réseau neuronal convolutionnel proposé par K. Simonyan et A. Zisserman de l'Université d'Oxford dans l'article intitulé «Verydeepconvolutional networks for large-scale image recognition ». Le modèle atteint une précision de 92,7% sur la base de données ImageNet qui contient plus de 14 millions d'images appartenant à 1000 classes. C'est un modèle de 16 couches qui reçoit à l'entrée une taille d'image RVB de taille 224 x 224 VGG16 utilise les filtres de noyaux 3X3 dans les couche de convolution et un filtre de taille 2X2 en max pooling, toutes les couches cachées sont équipées de la non-linéarité de rectification.

Et à la fin du réseau, trois couches entièrement connectées qui calcule le score de chaque classe des caractéristiques extraites de la convolution dans les étapes précédentes. Les cartes de caractéristiques de la couche finale sont représentées sous forme de vecteurs avec des valeurs scalaires, la fonction de classification soft-max. qui formule le score final entre 0 et 1 pour présenter la classification sous forme un pourcentage partagé entre les différentes classes,

Les couches entièrement connectées sont coûteuses en termes de calcul, des approches alternatives ont été proposées au cours des dernières années. Par exemple la mise en œuvre d'une couche Globale de pooling moyenne qui aide à réduire le nombre de paramètres dans le réseau de manière significative.

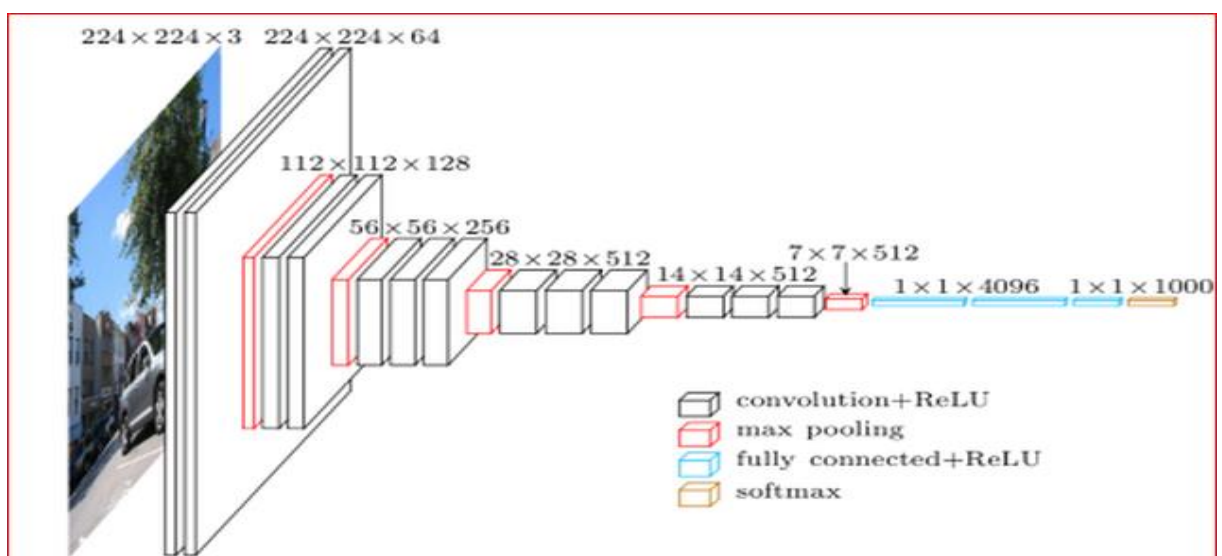


FIGURE II.7 Représentation 3D de l'architecture de VGG-16

II.7.1 Le GraphicsProcessingUnits

Les GPU sont les processeurs adaptés pour entraîner des réseaux de neurones. Initialement développés pour les opérations graphiques, NVIDIA a lancé en 2007 des GPU destinés à des calculs plus génériques. Ces GPU accélèrent les portions de code intensives en calculs parallélisables, les portions séquentielles restantes étant traitées par les CPU. Les réseaux de neurones nécessitent de nombreuses opérations matricielles, qui sont accélérées d'un facteur dix lorsqu'elles sont distribuées sur les milliers de cœurs de GPU.

Conclusion

Dans cette partie, nous avons présenté quelques méthodes existantes sur la reconnaissance de l'expression faciale. Ainsi nous avons eu à définir qu'est-ce qu'une expression faciale et qu'elles sont les plus utilisés, et qu'est-ce que le Deep Learning et comment il se différencie des algorithmes de ML traditionnelles. Nous avons également parlé des trois familles majeures de modèle à savoir les réseaux convolutifs, les réseaux récurrents ainsi que les modèles génératifs, ces derniers font encore l'objet de recherches intensives.

Chapitre III

Implémentation et résultats expérimentaux

Chapitre III Implémentation résultats expérimentaux

III.1 Introduction

Dans ce chapitre, nous présentons la reconnaissance des expressions faciales en temps réel de sept expressions humaines les plus élémentaires : En colère, Dégoût, Peur, Heureux, Neutre, Triste, Surpris. Nous commençons tout d'abord par la présentation des ressources, du langage et de l'environnement de développement que nous avons utilisé, puis par les étapes de réalisation du modèle et on termine par les tests effectués. Ce chapitre est composé de deux parties, l'implémentation du système et les résultats expérimentaux des tests.

III.2 Objectifs et contraintes de travail :

✚ Les Objectifs :

Notre objectif est de prédire l'expression du visage humain en temps réel le plus rapidement et le plus précisément possible.

✚ Les Contraintes:

- **Latence:** étant donné une image, le système doit être en mesure de prédire l'expression immédiatement et de transférer le résultat. Par conséquent, il existe une faible latence requise.
- **Interopérabilité:** L'interopérabilité est importante pour les images fixes mais pas en temps réel. Pour les images fixes, la probabilité d'expressions prédites peut être donnée.
- **Précision:** Notre objectif est de prédire l'expression d'un visage dans l'image aussi précisément que possible. Plus la précision du test est élevée, meilleures seront les performances de notre modèle dans le monde réel.

III.3 Environnement de développement:

III.3.1 Environnement matériel (hardware):

Un ordinateur avec les caractéristiques suivantes:

- **Processeur:** Processeur: Intel i7 8550u Ram
- **RAM:**8go
- **Disque Dur:**1000 go HDD Webcam intégrée de marque

III.3.2 Environnement immatériel (software) :

- **Système d'exploitation :** Windows 10 64bit
- **Logiciel :** python 3.7 64bit

III.3.3. Outils de développement

➤ Jupyter Notebook

Est une application Web tournée essentiellement vers l'apprentissage de langages de programmation, nous citons par exemple Julia, Java, Python, ... et notamment R. Cette application se présente comme une méthode innovatrice et vivante ayant comme fonctionnalité l'intégration de live code dans les Notebooks pour ce genre d'enseignement basé sur la technologie et l'informatique.

➤ Python

Est un langage de programmation interprété, multi paradigme et multiplateforme. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Schème, Small talk et Tcl.

Le langage Python est placé sous une licence libre proche de la licence BSD4 et fonctionne sur la plupart des plates-formes informatiques, des smartphones aux ordinateurs centraux⁵, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser, il est également apprécié par certains pédagogues qui y trouvent un langage où la syntaxe, clairement séparée des

mécanismes de bas niveau, permet une initiation aisée aux concepts de base de la programmation.

➤ **Anaconda**

Est une distribution libre et open source des langages de programmation Python et R appliquée au développement d'applications dédiées à la science des données et à l'apprentissage automatique, qui vise à simplifier la gestion des paquets et de déploiement. Les versions de paquetages sont gérées par le système de gestion de paquets conda. La distribution Anaconda est utilisée par plus de 6 millions d'utilisateurs et comprend plus de 250 paquets populaires en science des données adaptés pour Windows, Linux.

III.3.4 Bibliothèques utilisées

➤ **Open CV**

Open CV est une bibliothèque libre de vision par ordinateur. Cette bibliothèque est écrite en C et C++ et peut être utilisée sous Linux, Windows et Mac OS X. Des interfaces ont été développées pour Python, Ruby, Matlab et autre langage. Elle est orientée vers des applications en temps réel. Un des buts d'Open CV est d'aider les gens à construire rapidement des applications sophistiquées de vision à l'aide d'infrastructure simple de vision par ordinateur.

La bibliothèque d'Open CV contient plus de 500 fonctions, elle met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques. Elle propose la plupart des opérations classiques en traitement bas niveau des images et des vidéos. Ses fonctionnalités sont :

- Le traitement d'images.
- Le traitement vidéo.
- Les algorithmes d'apprentissage.
- Les calculs Matriciels.

➤ **Tensorflow**

TensorFlow est un cadre de travail (framework) de programmation pour le calcul numérique de Google, initié et développé par l'équipe « Google Brain » spécialisé dans l'intelligence artificielle, et rendu Open Source en Novembre 2015, et est devenu très rapidement l'un des frameworks les plus utilisés pour le Deep Learning, Les caractéristiques principales de Tensorflow sont :

- Multiplate formes (Windows, Linux, Mac OS, et même Android et iOS).
- APIs en Python, C++, Java et Go (l'API Python est plus complète cependant, c'est sur celle-ci que nous allons travailler).
- Temps de compilation très court dû au backend en C/C++.
- Supporte les calculs sur CPU, GPU et même le calcul distribué sur cluster.
- Une documentation extrêmement bien fournie avec de nombreux exemples et tutoriels.
- Enfin et surtout: Le fait que le framework vienne de Google et que ce dernier ait annoncé avoir migré la quasi-totalité de ses projets liés au Deep Learning en TensorFlow est quelque peu rassurant. Cependant dans FACECNN nous utilisons cette plateforme via la bibliothèque Keras ce qui est une sur-couche à TensorFlow donc un niveau plus haut que Tensorflow.

➤ **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 +.

➤ **Keras**

Keras est un framework open source d'apprentissage profond pour le Python, capable de s'exécuter sur TensorFlow. Il est écrit par Francis Chollet, membre de l'équipe Google. Keras est utilisé dans un grand nombre de startups, de laboratoires de recherche (dont le CERN, Microsoft Research et la NASA) et de grandes entreprises telles que Netflix, Yelp, Square, Uber, Google, etc. Il a été développé dans le cadre de

l'effort de recherche du projet ONEIROS (Open-ended Neuro Electronic Intelligent Robot Operating System). En 2017, l'équipe TensorFlow de Google a pris la décision de fournir un support pour Keras et de l'intégrer dans la bibliothèque principale de TensorFlow. Il présente un ensemble d'abstractions de plus haut niveau et plus intuitives qui facilitent la configuration des réseaux neuronaux.

III .3.4 .1 Installation des bibliothèques

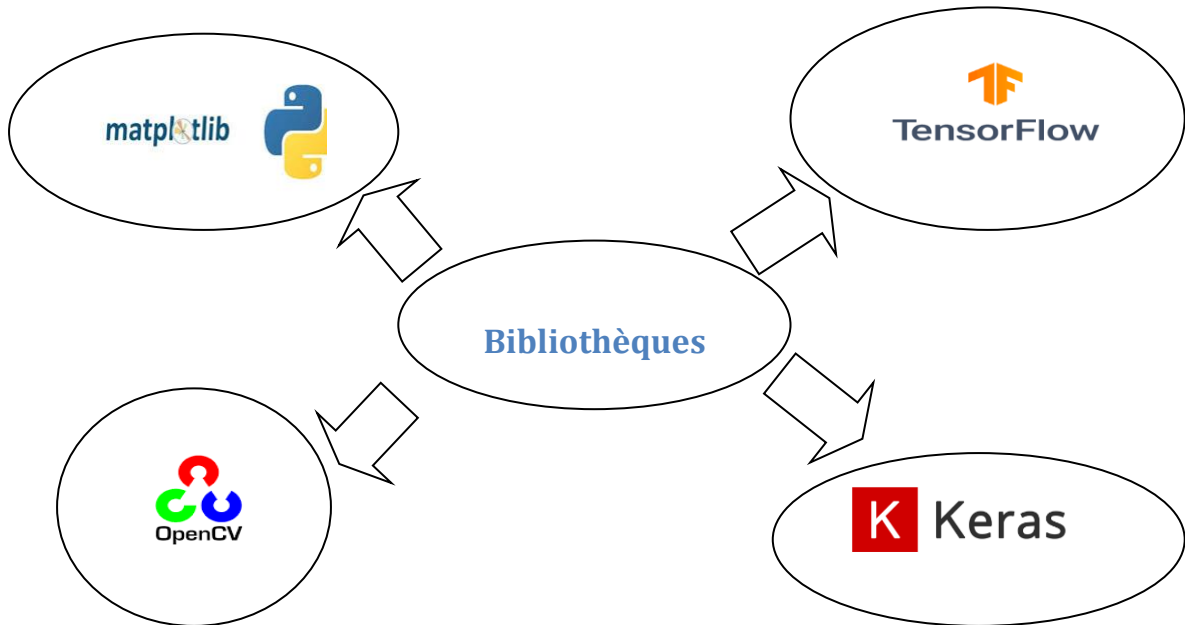


Figure III.1 : Les bibliothèques utilisées.

- **Création d'un nouvel environnement :**

Il y a beaucoup de méthodes pour la création des environnements Python nous utilisons la méthode 'Anaconda Navigator'

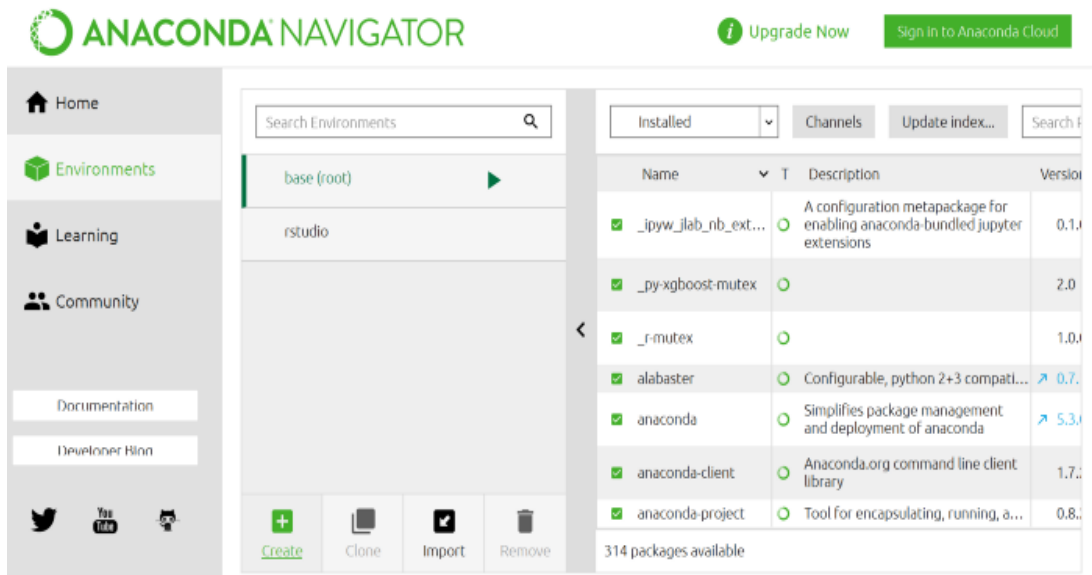


FIGURE III.2. : Interface d'Anaconda Navigator.

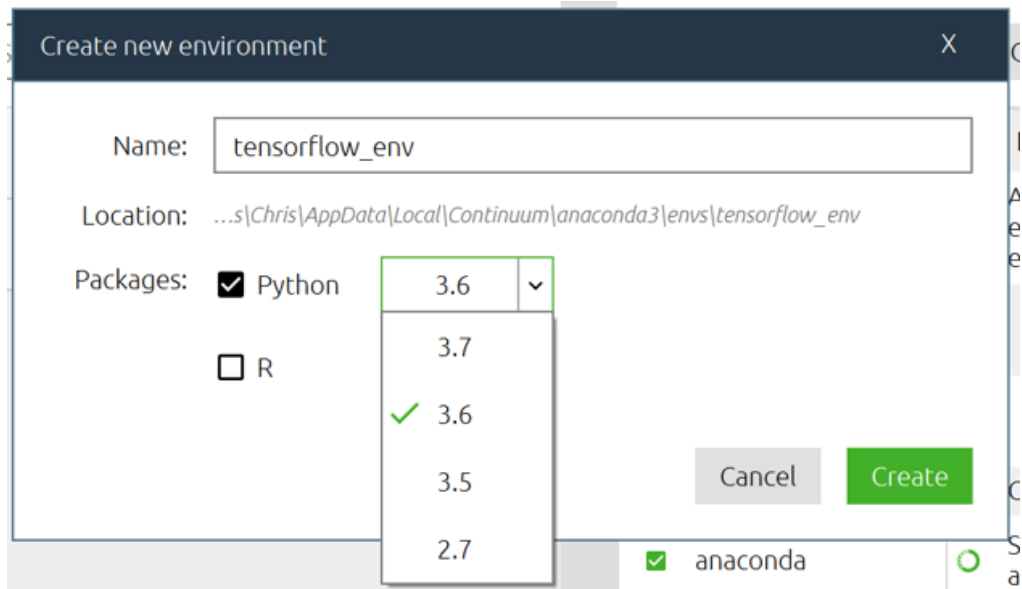


FIGURE III.3. : Fenêtre de la création d'un environnement.

- **Installation de Tensorflow CPU :**

Il y a beaucoup de méthodes pour installer Tensorflow pour le CPU nous utilisons la méthode 'Anaconda Navigator'

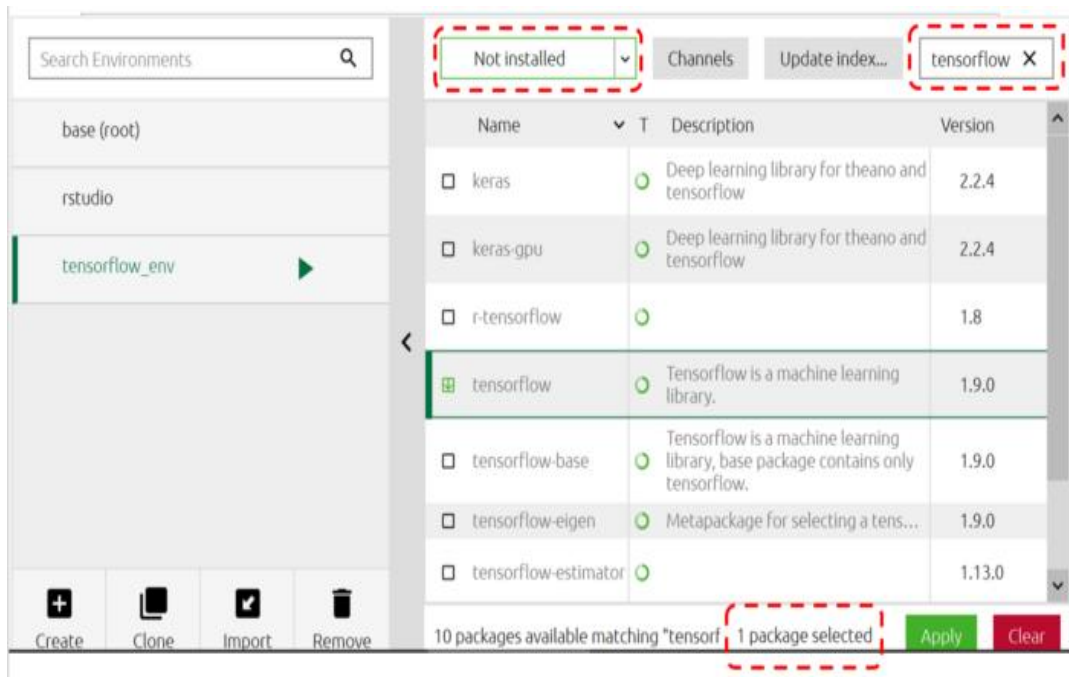


FIGURE III.4 : L'installation de la bibliothèque keras.

- **Installation de keras :**
Pour installer keras, nous utilisons la méthode 'Anaconda Navigator'

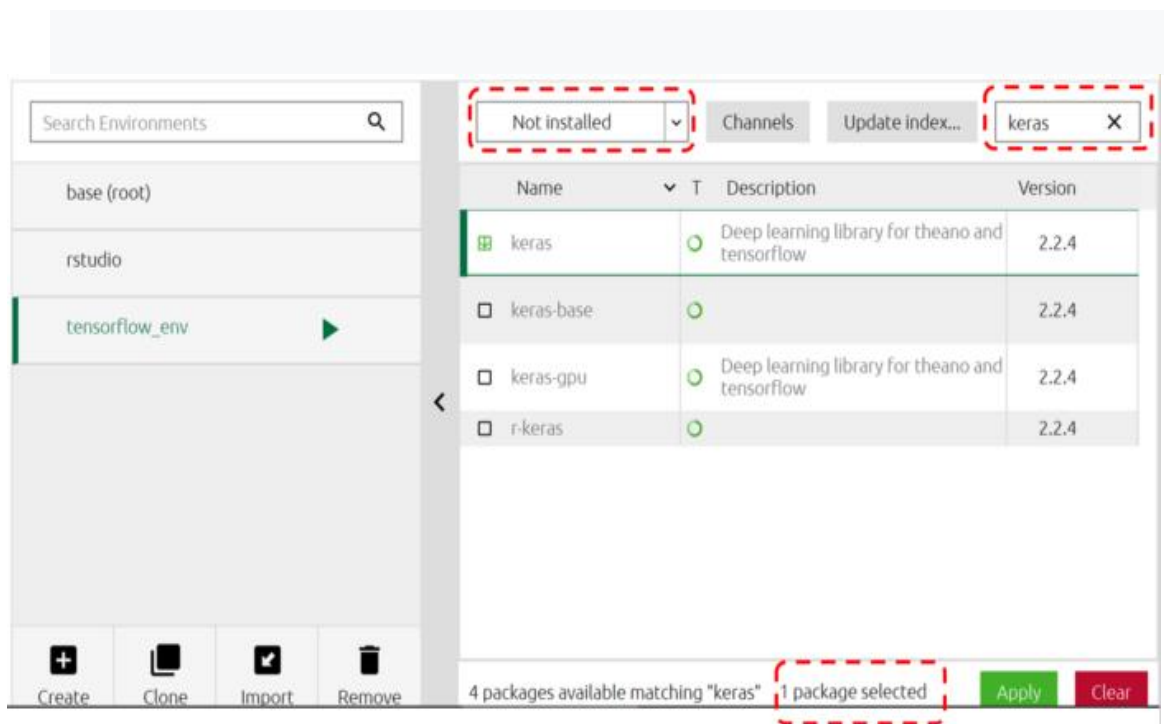


FIGURE III.5 : L'installation de la bibliothèque keras.

▪ Installation d'open cv :

La méthode utilisée est « Anaconda navigator » pour installer open cv

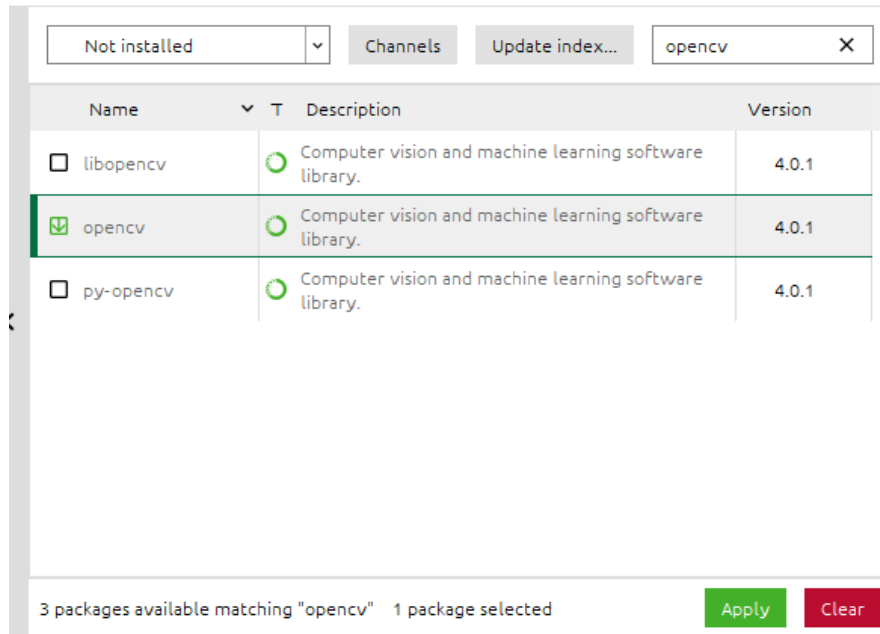


FIGURE III.6 l'installation d'Open CV.

III.4 partie du programme

III.4.1 Initialisation :

Dans cette partie on importe les bibliothèques qu'on a installé et aussi celles qui sont préinstallées :

```
: import os
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from PIL import Image
import glob
import cv2
from sklearn.model_selection import train_test_split
from keras.layers import Dropout, Dense
from keras.layers.normalization import BatchNormalization
from keras.models import Sequential, load_model
from keras.applications import VGG16
from sklearn.metrics import accuracy_score, confusion_matrix
```

FIGURE III.7: Capture du code d'installation des bibliothèques.

III.4.2 Base de données utilisée :

Pour de meilleurs résultats, il est préférable d'entraîner le réseau avec beaucoup d'échantillons d'images. Cela augmenterait la précision et améliorerait la performance des modèles, nous avons alors téléchargé des données de 4 sources différentes.

1. **Images humaines Source-1:**<http://www.consortium.ri.cmu.edu/ckagree/>
2. **Images humaines Source-2:**<http://app.visgraf.impa.br/database/faces/>
3. **Images humaines Source-3:**<http://www.kasrl.org/jaffe.html>
4. **Source d'images animées:**<https://grail.cs.washington.edu/projects/deepexpr/ferg-db.html>

Nous avons un total de 10596 images. Dont 1496 images humaines et 9100 images animées. Par conséquent, nous avons deux types d'images. L'une concerne les images humaines et l'autre les images animées.

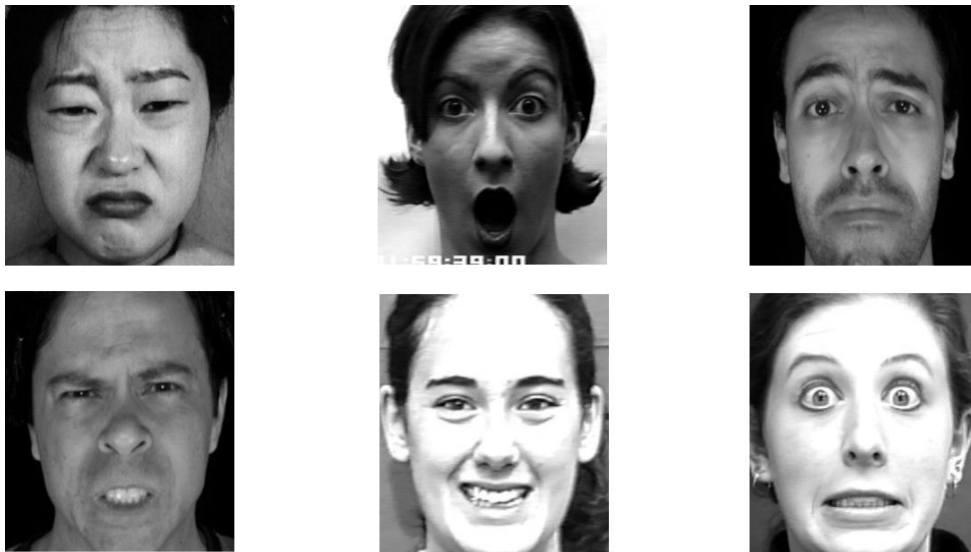


Figure III.8 Expressions humaines.

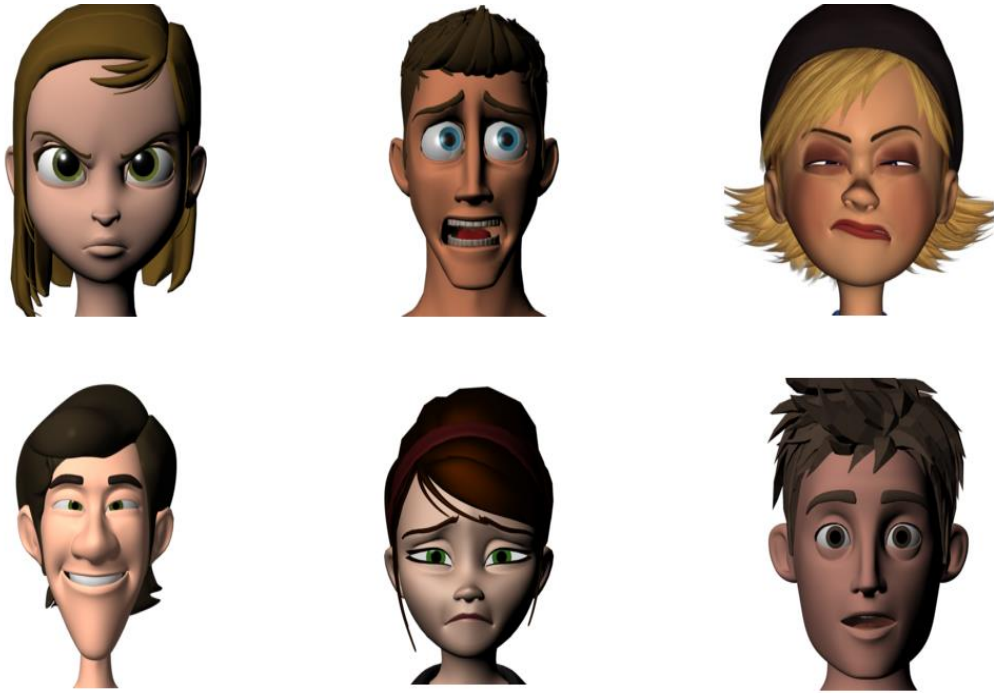


Figure III.9 Expressions animées.

III.4.3 Distribution des données Data set en Train data, Validation data et Test data

Pour les images humaines et animées, nous avons divisé nos données de telle sorte que les trois, à savoir les données de train, de Cross Validation et de test, devraient obtenir des images d'étiquettes de classe uniformément réparties.

Train Data :

L'échantillon de données utilisé pour ajuster le modèle.

Validation Data :

Échantillon de données utilisé pour fournir une évaluation non biaisée de l'ajustement d'un modèle sur l'ensemble de données d'entraînement lors du réglage des hyper paramètres du modèle. L'évaluation devient plus biaisée au fur et à mesure que la compétence sur l'ensemble de données de validation est intégrée à la configuration du modèle.

Test Data :

Échantillon de données utilisé pour fournir une évaluation non biaisée d'un ajustement final du modèle sur l'ensemble de données d'apprentissage.

La distribution des images humaines est la suivante :

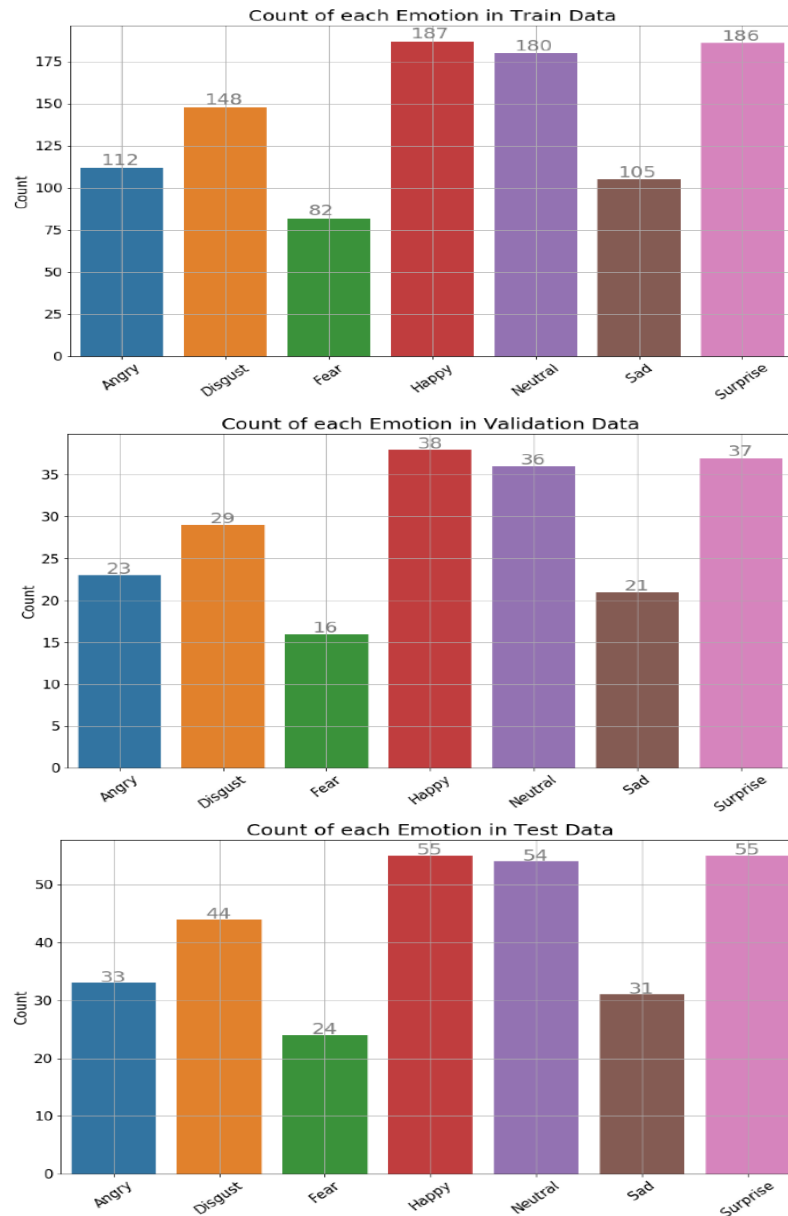


FIGURE III.10 Distribution d'images humaines.

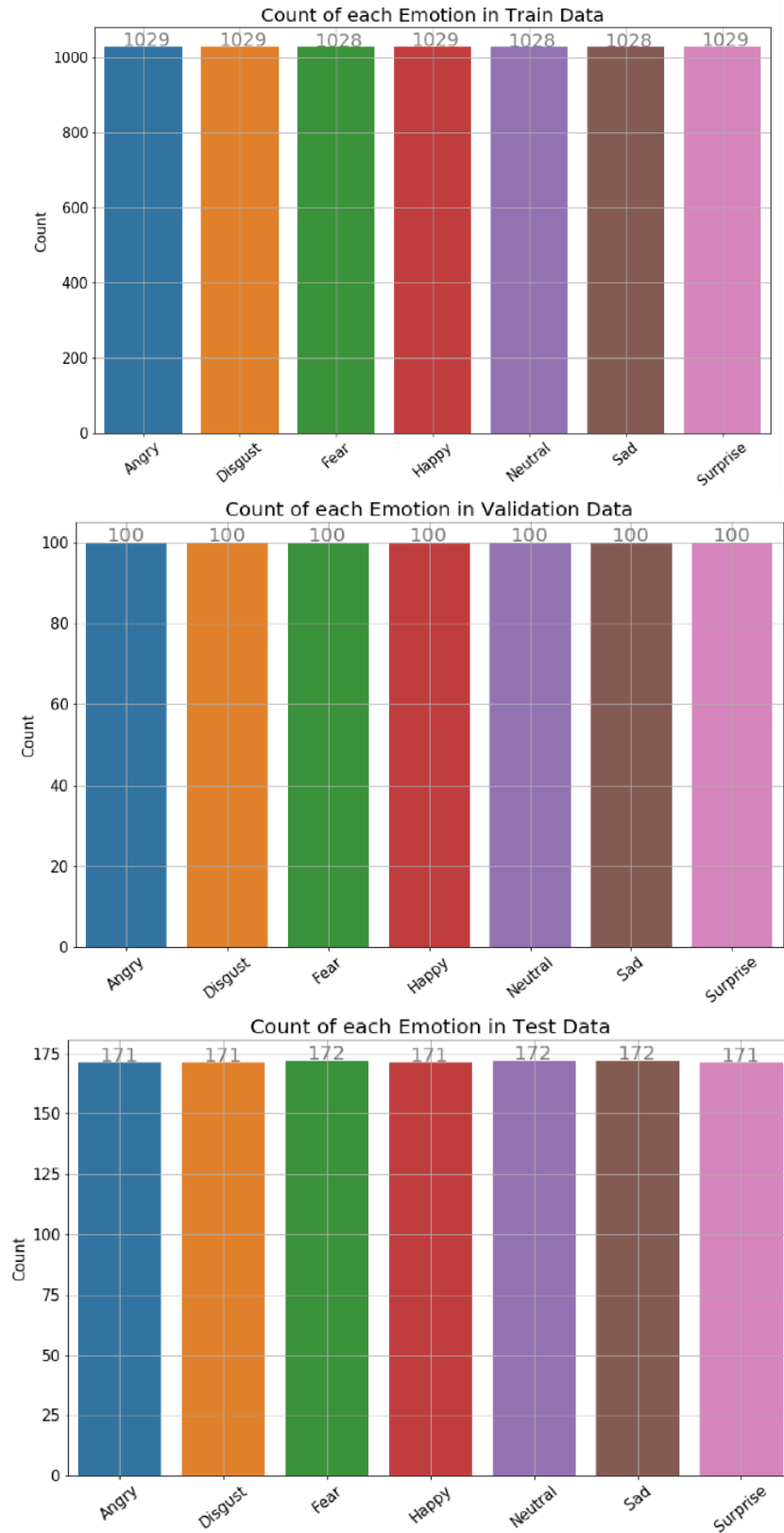


FIGURE III.11 Distribution d'images animées.

Nous avons maintenant au total six data frames :trois data frames - train, test et validation - pour les images humaines et trois data frames - train, test et Validation pour les images animées.

III.4.4 Prétraitement des images

Le code de la conversion des images en niveau de gris est comme suit :

```
def convt_to_gray(df):
    count = 0
    for i in range(len(df)):
        path1 = df["folderName"][i]
        path2 = df["imageName"][i]
        img = cv2.imread(os.path.join(path1, path2))
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        cv2.imwrite(os.path.join(path1, path2), gray)
        count += 1
    print("Total number of images converted and saved = "+str(count))
```

```
convt_to_gray(df_anime_train)
```

Total number of images converted and saved = 7200

```
convt_to_gray(df_anime_cv)
```

Total number of images converted and saved = 700

```
convt_to_gray(df_anime_test)
```

Total number of images converted and saved = 1200

FIGURE III.12 : code de la conversion des images en niveau de gris.

Le code pour le recadrage, le redimensionnement des images et leur enregistrement est comme suit :

```
def change_image(df):
    count = 0
    for i, d in df.iterrows():
        img = cv2.imread(os.path.join(d["folderName"], d["imageName"]))
        face_clip = img[40:240, 35:225] #cropping the face in image
        face_resized = cv2.resize(face_clip, (350, 350))
        cv2.imwrite(os.path.join(d["folderName"], d["imageName"]), face_resized) #resizing and saving the image
        count += 1
    print("Total number of images cropped and resized = {}".format(count))
```

```
change_image(df_anime_train)
```

Total number of images cropped and resized = 7200

```
change_image(df_anime_cv)
```

Total number of images cropped and resized = 700

```
change_image(df_anime_test)
```

Total number of images cropped and resized = 1200

FIGURE III.13 : code du recadrage et redimensionnement des images.

III.4.5 Concaténation de toutes les dataframes

La Combinaison de données de l'entraînement du modèle humaines et animées a pour code comme suit :

Après la visualisation des expressions, on va concaténer les data frame.

```
frames = [df_human_train, df_anime_train]
combined_train = pd.concat(frames)
combined_train.shape
```

```
(8200, 4)
```

```
Final_human.reset_index(inplace = True, drop = True)
Final_human = Final_human.sample(frac = 1.0) #shuffling the dataframe
Final_human.reset_index(inplace = True, drop = True)
Final_human.head()
```

	folderName	\
0	../../../../Desktop/Python/Data/Human/Happy/	
1	../../../../Desktop/Python/Data/Human/Neutral/	
2	../../../../Desktop/Python/Data/Human/Sad/	
3	../../../../Desktop/Python/Data/Human/Neutral/	
4	../../../../Desktop/Python/Data/Human/Neutral/	

	imageName	Emotion	Labels
0	Happy.26033.jpg	Happy	4
1	29735Exp6heartbroken_African_190.jpg	Neutral	5
2	37111Exp4crying_lady_29.jpg	Sad	6
3	74581Exp6amazed_family_433.jpg	Neutral	5
4	73442Exp6amazed_american_25.jpg	Neutral	5

FIGURE III.14 : code de la concaténation de toutes les data frames.

Nous avons combiné les trames de données des images humaines et animées. Nous n'avons combiné que les images de train à la fois humaines et animées afin de pouvoir former notre modèle à la fois sur des images humaines et animées. Cependant, nous avons séparé les images de Validation et les images de test des images humaines et animées afin de pouvoir croiser la validation de nos résultats sur des images humaines et animées séparément. En même temps, nous pourrions également tester l'efficacité de notre modèle séparément sur des images humaines et animées. Par cela, nous apprendrons à savoir à quel point notre modèle fonctionne séparément sur des images humaines et animées.

III.4.6 Création de fonctionnalités de Bottleneck à partir du

Modèle VGG-16. (Transfer Learning)

Le Bottleneck est la dernière couche d'activation avant les couches entièrement connectées dans un modèle vgg16. Le fait d'avoir une telle couche encourage le réseau à comprimer les représentations des caractéristiques pour les adapter au mieux à l'espace disponible, les caractéristiques apprises des images qui sont ensuite transmises au MLP qui agit comme un modèle de pointe. Ce MLP réduit ensuite la fonction de perte et met à jour les poids dans le MLP et les filtres dans CNN. Si nous utilisons uniquement le modèle vgg16 jusqu'aux couches entièrement connectées, nous pouvons convertir l'entrée X (image de taille 224 x 224 x 3, par exemple) en sortie Y de taille 512 x 7 x 7 (figure 15).

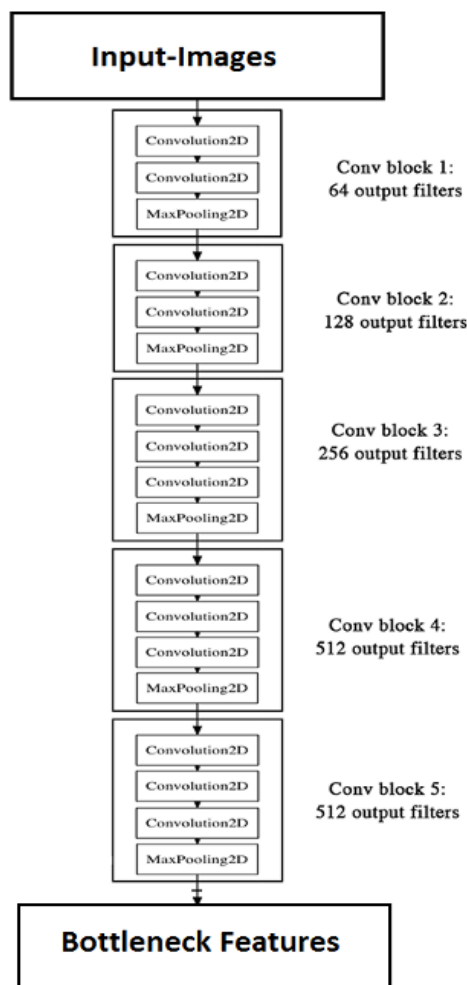


Figure III.15: VGG-16 Bottleneck Features.

Nous avons transmis chaque image une par une à travers ce réseau et nous avons généré des Bottleneck features, nous les avons stockées dans un tableau numpy. Nous devons utiliser la fonction **model.predict () de VGG-16** et générer des Bottleneck features pour nos images.

Enfin, pour l'ensemble de nos 10596 images, nous avons généré des Bottleneck features qu'on a ensuite enregistré sur notre disque dur. De cette façon, nous avons utilisé l'apprentissage par transfert du modèle VGG-16 pour notre propre tâche.

```
def loadCombinedTrainBatch(batch_size):
    global TrainCombined_batch_pointer
    batch_images = []
    batch_labels = []
    for i in range(batch_size):
        path1 = Train_Combined.iloc[TrainCombined_batch_pointer + i]["folderName"]
        path2 = Train_Combined.iloc[TrainCombined_batch_pointer + i]["imageName"]
        read_image = cv2.imread(os.path.join(path1, path2))
        read_image_final = read_image/255.0 #here, we are normalizing the images
        batch_images.append(read_image_final)

        batch_labels.append(TrainCombined_Labels[TrainCombined_batch_pointer + i]) #appending corresponding Labels

    TrainCombined_batch_pointer += batch_size

    return np.array(batch_images), np.array(batch_labels)
```

```
#creating bottleneck features for train data using VGG-16- Image-net model
model = VGG16(weights='imagenet', include_top=False)
SAVEDIR = "../Data/Bottleneck_Features/Bottleneck_CombinedTrain/"
SAVEDIR_LABELS = "../Data/Bottleneck_Features/CombinedTrain_Labels/"
batch_size = 10
for i in range(int(len(Train_Combined)/batch_size)):
    x, y = loadCombinedTrainBatch(batch_size)
    print("Batch {} loaded".format(i+1))

    np.save(os.path.join(SAVEDIR_LABELS, "bottleneck_labels_{}".format(i+1)), y)

    print("Creating bottleneck features for batch {}".format(i+1))
    bottleneck_features = model.predict(x)
    np.save(os.path.join(SAVEDIR, "bottleneck_{}".format(i+1)), bottleneck_features)
    print("Bottleneck features for batch {} created and saved\n".format(i+1))
```

FIGURE III.16 : code pour la création de Bottleneck features pour les images d'entraînement.

III.4.7 Création du modèle :

Nous avons utilisé la fonction d'activation relu pour les couches cachées et la fonction d'activation softmax pour la classification (couche entièrement connectée)

```
no_of_classes = 7
```

```
#model architecture
def model(input_shape):
    model = Sequential()

    model.add(Dense(512, activation='relu', input_dim = input_shape))
    model.add(Dropout(0.1))

    model.add(Dense(256, activation='relu'))

    model.add(Dense(128, activation='relu'))
    model.add(BatchNormalization())

    model.add(Dense(64, activation='relu'))
    model.add(Dense(output_dim = no_of_classes, activation='softmax'))

    return model
```

FIGURE III.17 : code pour la création du modèle.

III.4.8 L'entraînement :

Dans cette phase on a utilisées ces codes pour entrainer le modèle a reconnaître les objets intérêts :

```
#training the model
SAVEDIR_COMB_TRAIN = "../Data/Bottleneck_Features/Bottleneck_CombinedTrain/"
SAVEDIR_COMB_TRAIN_LABELS = "../Data/Bottleneck_Features/CombinedTrain_Labels/"

SAVEDIR_CV_HUMANS = "../Data/Bottleneck_Features/Bottleneck_CVHumans/"
SAVEDIR_CV_HUMANS_LABELS = "../Data/Bottleneck_Features/CVHumans_Labels/"

SAVEDIR_CV_ANIME = "../Data/Bottleneck_Features/Bottleneck_CVAnimated/"
SAVEDIR_CV_ANIME_LABELS = "../Data/Bottleneck_Features/CVAnimated_Labels/"

SAVER = "../Data/Model_Save/"

input_shape = 10*10*512 #this is the shape of bottleneck feature of each image which comes after passing the image thro

model = model(input_shape)
# model.Load_weights(os.path.join(SAVER, "model.h5"))
model.summary()
model.compile(loss = 'categorical_crossentropy', optimizer = "adam", metrics = ["accuracy"])

epochs = 20
batch_size = 10
step = 0
combTrain_bottleneck_files = int(len(Train_Combined) / batch_size)
CVHuman_bottleneck_files = int(len(CV_Humans) / batch_size)
CVAnime_bottleneck_files = int(len(CV_Animated) / batch_size)
epoch_number, CombTrain_loss, CombTrain_acc, CVHuman_loss, CVHuman_acc, CVAnime_loss, CVAnime_acc = [], [], [], [], [], [

for epoch in range(epochs):
    avg_epoch_CombTr_loss, avg_epoch_CombTr_acc, avg_epoch_CVHum_loss, avg_epoch_CVHum_acc, avg_epoch_CVAnime_loss, avg_e
    epoch_number.append(epoch + 1)

    for i in range(combTrain_bottleneck_files):

        step += 1

        #Loading batch of train bottleneck features for training MLP.
        X_CombTrain_load = np.load(os.path.join(SAVEDIR_COMB_TRAIN, "bottleneck_{}.npy".format(i+1)))
        X_CombTrain = X_CombTrain_load.reshape(X_CombTrain_load.shape[0], X_CombTrain_load.shape[1]*X_CombTrain_load.shap
        Y_CombTrain = np.load(os.path.join(SAVEDIR_COMB_TRAIN_LABELS, "bottleneck_labels_{}.npy".format(i+1)))

        #Loading batch of Human CV bottleneck features for cross-validation.
```



```

X_CVHuman_load = np.load(os.path.join(SAVEDIR_CV_HUMANS, "bottleneck_{}.npy".format((i % CVHuman_bottleneck_files
X_CVHuman = X_CVHuman_load.reshape(X_CVHuman_load.shape[0], X_CVHuman_load.shape[1]*X_CVHuman_load.shape[2]*X_CV
Y_CVHuman = np.load(os.path.join(SAVEDIR_CV_HUMANS_LABELS, "bottleneck_labels_{}.npy".format((i % CVHuman_bottlen

#Loading batch of animated CV bottleneck features for cross-validation.
X_CVAnime_load = np.load(os.path.join(SAVEDIR_CV_ANIME, "bottleneck_{}.npy".format((i % CVAnime_bottleneck_files)
X_CVAnime = X_CVAnime_load.reshape(X_CVAnime_load.shape[0], X_CVAnime_load.shape[1]*X_CVAnime_load.shape[2]*X_CVA
Y_CVAnime = np.load(os.path.join(SAVEDIR_CV_ANIME_LABELS, "bottleneck_labels_{}.npy".format((i % CVAnime_bottlene

CombTrain_Loss, CombTrain_Accuracy = model.train_on_batch(X_CombTrain, Y_CombTrain) #train the model on batch
CVHuman_Loss, CVHuman_Accuracy = model.test_on_batch(X_CVHuman, Y_CVHuman) #cross validate the model on CV Human
CVAnime_Loss, CVAnime_Accuracy = model.test_on_batch(X_CVAnime, Y_CVAnime) #cross validate the model on CV Animat

print("Epoch: {}, Step: {}, CombTr_Loss: {}, CombTr_Acc: {}, CVHum_Loss: {}, CVHum_Acc: {}, CVAni_Loss: {}, CVAni

avg_epoch_CombTr_loss += CombTrain_Loss / combTrain_bottleneck_files
avg_epoch_CombTr_acc += CombTrain_Accuracy / combTrain_bottleneck_files
avg_epoch_CVHum_loss += CVHuman_Loss / combTrain_bottleneck_files
avg_epoch_CVHum_acc += CVHuman_Accuracy / combTrain_bottleneck_files
avg_epoch_CVAnime_loss += CVAnime_Loss / combTrain_bottleneck_files
avg_epoch_CVAnime_acc += CVAnime_Accuracy / combTrain_bottleneck_files

print("Avg_CombTrain_Loss: {}, Avg_CombTrain_Acc: {}, Avg_CVHum_Loss: {}, Avg_CVHum_Acc: {}, Avg_CVAnime_Loss: {}, AV

CombTrain_loss.append(avg_epoch_CombTr_loss)
CombTrain_acc.append(avg_epoch_CombTr_acc)
CVHuman_loss.append(avg_epoch_CVHum_loss)
CVHuman_acc.append(avg_epoch_CVHum_acc)
CVAnime_loss.append(avg_epoch_CVAnime_loss)
CVAnime_acc.append(avg_epoch_CVAnime_acc)

model.save(os.path.join(SAVER, "model.h5")) #saving the model on each epoc
model.save_weights(os.path.join(SAVER, "model_weights.h5")) #saving the weights of model on each epoch
print("Model and weights saved at epoch {}".format(epoch + 1))

log_frame = pd.DataFrame(columns = ["Epoch", "Comb_Train_Loss", "Comb_Train_Accuracy", "CVHuman_Loss", "CVHuman_Accuracy"]
log_frame["Epoch"] = epoch_number
log_frame["Comb_Train_Loss"] = CombTrain_loss
log_frame["Comb_Train_Accuracy"] = CombTrain_acc
log_frame["CVHuman_Loss"] = CVHuman_loss
log_frame["CVHuman_Accuracy"] = CVHuman_acc
log_frame["CVAnime_Loss"] = CVAnime_loss

log_frame["CVAnime_Accuracy"] = CVAnime_acc
log_frame.to_csv("../Data/Logs/Log.csv", index = False)

```

FIGURE III.18 : code pour l'entraînement du modèle.

- ✓ Nous avons obtenu une précision de 100% sur les données de Validation des images animées. La précision de notre Validation sur les images humaines a été bonne, nous avons alors obtenu une précision finale de 87% sur les données CV des images humaines.
- ✓ Nous avons exécuté notre modèle jusqu'à 20 époques et nous avons obtenu les chiffres suivants de l'époque 1 à l'époque 20:

	CVHuman_Accuracy	CVAnime_Loss	CVAnime_Accuracy
0	0.15	2.21	0.19
1	0.21	1.55	0.34
2	0.25	1.09	0.51
3	0.29	0.70	0.71
4	0.35	0.40	0.85
5	0.48	0.22	0.91
6	0.50	0.11	0.93
7	0.54	0.07	0.97
8	0.55	0.04	0.98
9	0.57	0.01	0.99
10	0.60	0.01	0.99
11	0.61	0.01	0.99
12	0.69	0.01	1.00
13	0.75	0.00	1.00
14	0.83	0.00	1.00
15	0.85	0.00	1.00
16	0.84	0.00	1.00
17	0.86	0.00	1.00
18	0.87	0.00	1.00
19	0.87	0.00	1.00

✓ **FIGURE III.19** : Journal de train.

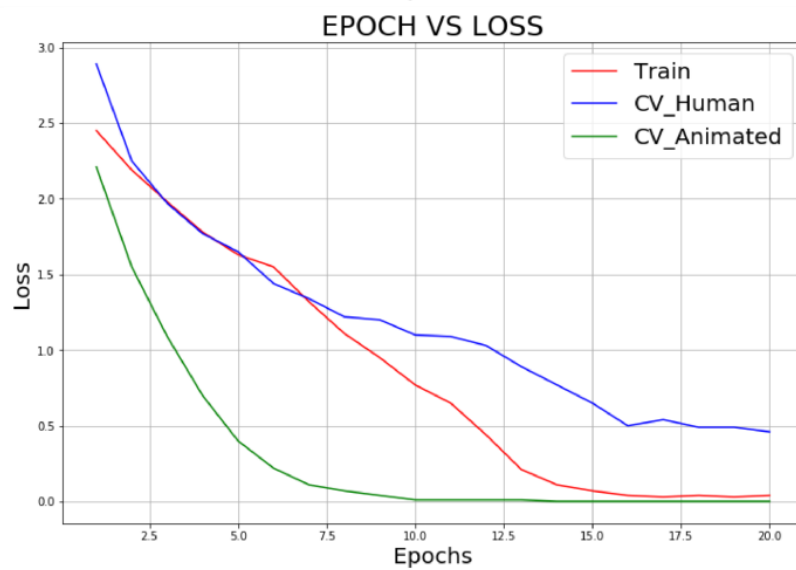
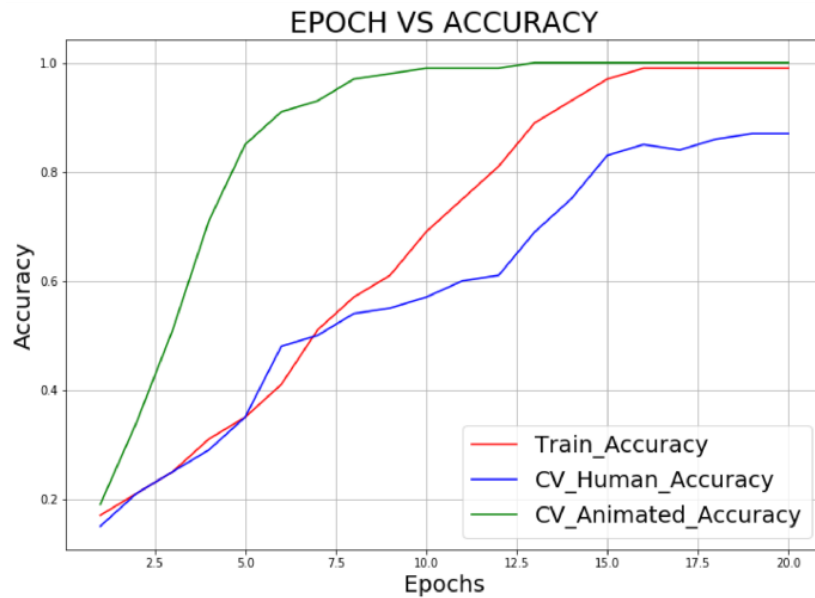


FIGURE III.20 : Précision de la perte d'époque.

- ✓ Nous avons pu réduire les pertes de la 1ère à la 20ème époque comme suit:
 - Perte d'entraînement: 2,45 à 0,04.
 - CV perte humaine: 2,89 à 0,46.
 - CV Animated loss: 2,21 à 0,00.
- ✓ Quant à la précision, elle a augmenté de la 1ère à la 20ème époque comme suit:
 - Précision du train: 17% à 99%
 - Précision humaine CV: 15% à 87%
 - Précision animée CV: 19% à 100%

III.4.9 Résultats de test

Comme mentionné ci-dessus, nous avons séparé les données de test des images animées et humaines afin de pouvoir tester notre modèle sur des images humaines et animées séparément.

Après avoir testé notre modèle sur des données de test humain, nous avons obtenu les résultats suivants:

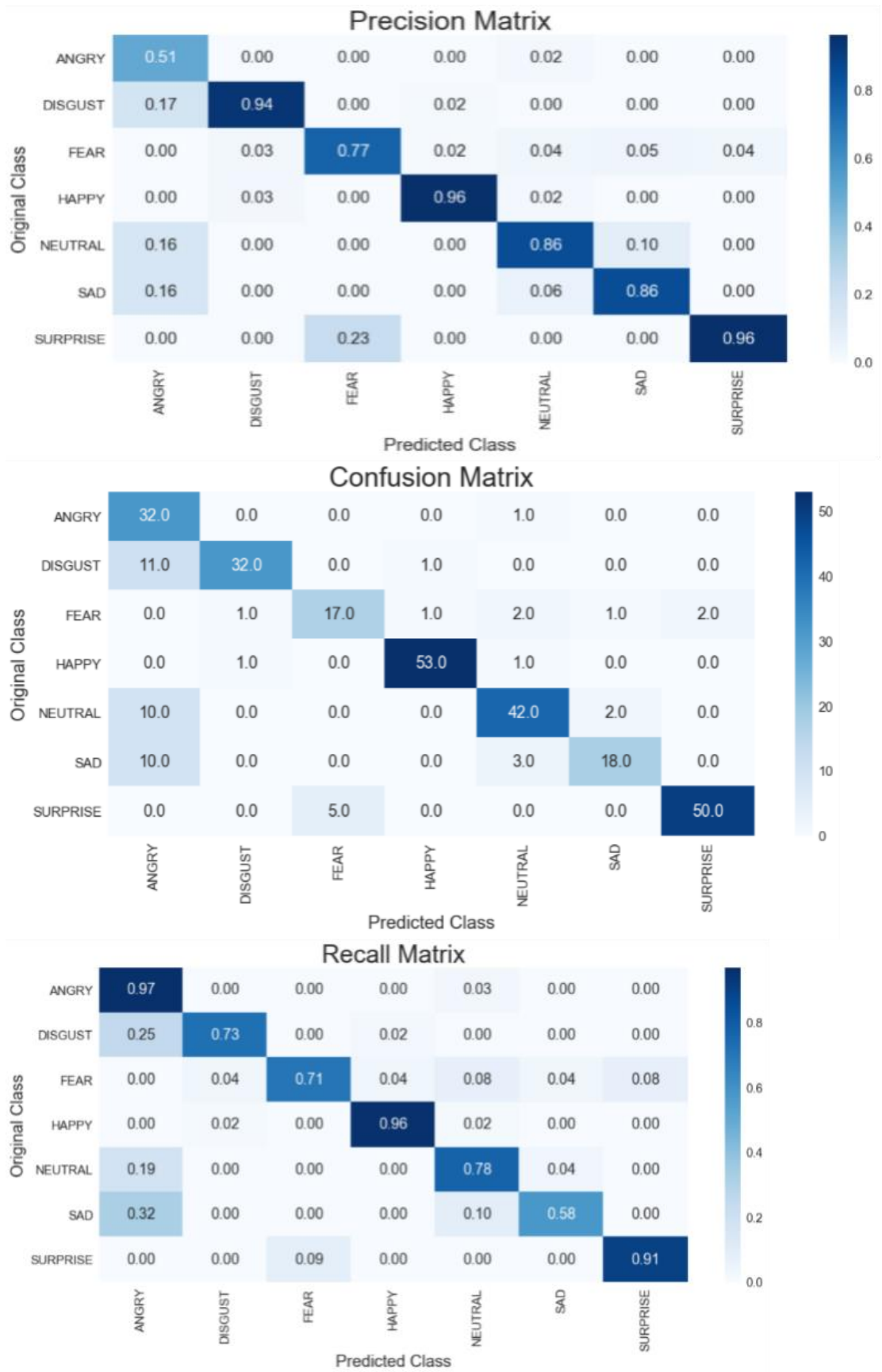


FIGURE III.21 : Images humaines de la matrice de confusion et de rappel.

Dans la matrice de confusion, nous pouvons observer que certaines des classes sont plus biaisées vers la classe «Neutre». La même chose peut être vérifiée à partir de la matrice de rappel. Il existe de nombreuses images de la classe «TRISTE» qui sont prédites dans «Neutre». Le modèle n'est pas capable de faire une distinction complète entre les classes «Neutre» et «TRISTE», peut-être parce qu'il n'existe pas une différence subtile entre les expressions en Neutre et TRISTE

Après avoir testé notre modèle sur des données de test Animé, nous avons obtenu les résultats suivants:

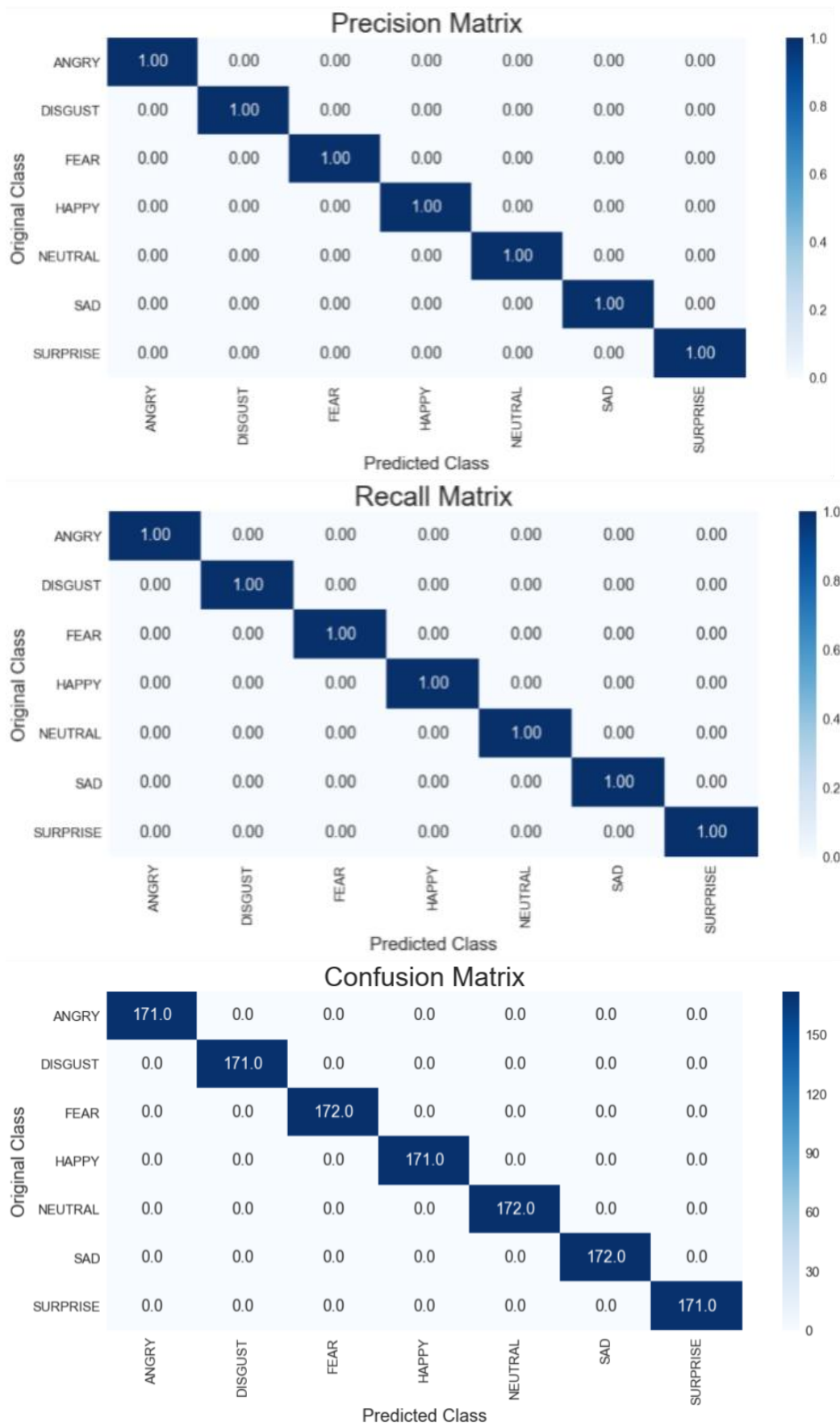


FIGURE III.22 : Images animées de la matrice de confusion et de rappel.

Dans les images animées, nous avons une précision de 100%. La même chose peut être vérifiée à partir de la matrice de confusion et de la matrice de rappel. Notre modèle fonctionne parfaitement sur les images animées.

III.5 Utilisation du modèle pour la reconnaissance en temps réel :

Pour la mise en œuvre de notre modèle nous avons créé un autre code où la reconnaissance des expressions faciales va être établie.

Importation des bibliothèques nécessaires

```
import cv2
import numpy as np
from keras.models import load_model
from keras.applications import VGG16
```

FIGURE III.23 : code d'installation des bibliothèques.

Chargement l du modèle obtenu

```
EMOTION_DICT = {1:"ANGRY", 2:"DISGUST", 3:"FEAR", 4:"HAPPY", 5:"NEUTRAL", 6:"SAD", 7:"SURPRISE"}
model_VGG = VGG16(weights='imagenet', include_top=False)
model_top = load_model("../Data/Model_Save/model.h5")
```

FIGURE III.24 : code de chargement du modèle.

Définition de l'interface d'utilisation :

```
def rerun(text, cap):
    while(True):
        ret, img = cap.read()
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        font = cv2.FONT_HERSHEY_SIMPLEX
        cv2.putText(img, "Last Emotion was "+str(text), (95,30), font, 1.0, (255, 0, 0), 2, cv2.LINE_AA)

        cv2.putText(img, "Press SPACE: FOR EMOTION", (5,470), font, 0.7, (255, 0, 0), 2, cv2.LINE_AA)

        cv2.putText(img, "Hold Q: To Quit", (460,470), font, 0.7, (255, 0, 0), 2, cv2.LINE_AA)

        faces = face_cascade.detectMultiScale(gray, 1.3, 5)
        for x,y,w,h in faces:
            cv2.rectangle(img, (x,y), (x+w, y+h), (255, 0, 0), 2)

        cv2.imshow("Image", img)

        if cv2.waitKey(1) == ord(' '):
            cv2.imwrite("test.jpg", img)
            text = return_prediction("test.jpg")
            first_run(text, cap)
            break

        if cv2.waitKey(1) == ord('q'):
            cap.release()
            cv2.destroyAllWindows()
            break
```

FIGURE III.25 : du code de la définition d'interface.

Les étapes du fonctionnement du programme sont :

- **Etape 01** : convertir l'image en niveau de gris et l'enregistrer

```
#converting image to gray scale and save it
img = cv2.imread(path)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imwrite(path, gray)
```

FIGURE III.26 : conversion de toutes les images en échelle de gris

- **Etape 02** : détecter le visage dans l'image, la rogner puis la redimensionner puis l'enregistrer


```

#detect face in image, crop it then resize it then save it
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
img = cv2.imread(path)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(gray, 1.3, 5)
for (x,y,w,h) in faces:
    face_clip = img[y:y+h, x:x+w]
    cv2.imwrite(path, cv2.resize(face_clip, (48,48)))

```

FIGURE III.27 : Code de la détection du visage dans l'image

- **Etape 03:**enfin enregistrer et lire l'image traitée puis faire une prédiction et afficher le résultat

```

#read the processed image then make prediction and display the result
read_image = cv2.imread(path)
read_image = read_image.reshape(1, read_image.shape[0], read_image.
↪shape[1], read_image.shape[2])
read_image_final = read_image/255.0
VGG_Pred = model_VGG.predict(read_image_final)
VGG_Pred = VGG_Pred.reshape(1, VGG_Pred.shape[1]*VGG_Pred.shape[2]*VGG_Pred.
↪shape[3])
top_pred = model_top.predict(VGG_Pred)
emotion_label = top_pred[0].argmax() + 1

```

FIGURE III.28: Affichage de nos résultat.

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



FIGURE III.29 Visualisation de nos images traitées.

III.6 CONCLUSION

Nous avons obtenu un très bon résultat, mais il y a toujours une énorme marge d'amélioration. Afin d'obtenir une meilleure précision, nous avons besoin de beaucoup plus d'images humaines avec une bonne variance entre elles. Nous pouvons également affiner les 2 ou 3 derniers blocs de convolution de la couche VGG-16 pour augmenter la précision. Nous pouvons également concevoir notre propre modèle CNN sur un ordinateur avancé en puissance de calcul en réglant soigneusement les hyperparamètres et en entraînant le modèle sur des 100,000 images humaines avec une bonne variance entre elles et en gardant la taille de chaque image supérieure à 400 * 400.

Conclusion générale

De nos jours, il existe des ordinateurs capables d'effectuer des tâches artificielles qui nécessitent de l'intelligence. Malgré tous ces progrès, la machine serait toujours située à des années-lumière du cerveau humain si une comparaison était faite entre les deux, principalement en raison de la complexité des tâches que nous essayons d'automatiser et de notre ignorance de cet organe mystérieux qu'est le cerveau et des mécanismes qui régissent son activité. Il ne fait aucun doute que la motivation derrière l'investissement dans la reconnaissance faciale est la multiplicité et la diversité des domaines d'application. La mise en place d'un système de reconnaissance faciale automatique et fiable est un obstacle technique non résolu. La reconnaissance automatique des visages à l'aide d'images 2D a été soigneusement étudiée au cours des dernières décennies. Cependant, en raison du manque inhérent d'informations faciales dans les images 2D, les algorithmes de reconnaissance faciale ont de grandes difficultés à gérer les différences d'éclairage, les changements de position, les expressions faciales, les temps de latence et l'occlusion. La surface de face 3D permet de surmonter les difficultés de position et de tête d'éclairage différentes. Avec l'avancement des technologies d'imagerie 3D, les données faciales 3D sont devenues de plus en plus disponibles ces dernières années. Il existe maintenant plusieurs bases de données publiques de référence de visage 3D disponibles.

L'objectif du projet est de prédire l'expression faciale humaine en temps réel aussi rapidement et précisément que possible.

Dans ce projet, nous avons introduit certaines méthodes actuelles de reconnaissance des expressions faciales qui ont atteint une précision très acceptable. La même chose a pu être vérifiée à partir de la matrice de confusion et de la matrice de récupération. Notre modèle a très bien fonctionné pour l'animation.

Enfin, ce projet nous a permis de faire progresser nos connaissances en abordant plusieurs domaines, le deep learning, la reconnaissance faciale, le langage de programmation python, et dans l'espoir qu'il nous offrira plus d'opportunités dans le futur.

Bibliographie

- [1] Khefif Bouchra « Mise au point d'une application de reconnaissance faciale "Université Abou Bakr Belkaid – Tlemcen
- [2] Riadh HAMZAOUI « Reconnaissance des personnes par le visage dans des séquences vidéo » Institut Supérieur des Sciences Appliquées et de Technologies de Sousse .
- [3] H. Ouamane et M. Benatia, « Identification De Reconnaissance Faciale Avec Des Expressions », Mémoire de Master en Electronique, Option : Télécommunication, Université Mohamed Khider Biskra, Soutenu le 07 Juin 2012.
- [4] W. Hizem et B. Dorizzi, « Capteur Intelligent pour la Reconnaissance de Visage », Thèse présentée pour l'obtention du diplôme de Docteur De L'Institut National Des Télécommunications Spécialité : Electronique/Informatique, Université Pierre et Marie Curie - Paris 6, 2009.
- [5] Liu, Xiaolong& Deng, Zhidong& Yang, Yuhan. Recent progress in semantic imagesegmentation. Artificial Intelligence Review. 1-18. 10.1007/s10462-018-9641-3.(2018).
- [6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, SanjeevSatheesh, Sean Ma,Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C.Berg, Fei-Fei Li: ImageNet Large Scale Visual Recognition Challenge. InternationalJournal of Computer Vision 115(3): 211-252 (2015)
- [7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, SanjeevSatheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, Fei-Fei Li: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision 115(3): 211-252 (2015)

- [8] Ian T. Young and Jan J. Gerbrands and Lucas J. van Vliet and Cip-data Koninklijke Bibliotheek and Den Haag and Young Ian Theodore and Gerbrands Jan Jacob and Van Vliet and Lucas Jozef, *Fundamentals Of Image Processing*, 1995
- [9] Russell, S. et Norvig, P., *Artificial Intelligence: A Modern Approach* (2nd ed.) ISBN 0-13-790395-2; Prentice Hall, 2003
- [10] Bengio: *Learning Deep Architectures for AI. Foundations and Trends in Machine Learning* 2(1): 1-127 (2009)
- [11] **Ghimire, D., Lee, J., Li, Z. N., & Jeong, S.** (2017). Recognition of facial expressions based on salient geometric features and support vector machines. *Multimedia Tools and Applications*, 76(6), 7921-7946.
- [12] Ian T. Young and Jan J. Gerbrands and Lucas J. van Vliet and Cip-data Koninklijke Bibliotheek and Den Haag and Young Ian Theodore and Gerbrands Jan Jacob and Van Vliet and Lucas Jozef, *Fundamentals Of Image Processing*, 1995
- [13] *Deep Learning with Applications Using Python <<Chatbots and Face, Object, and Speech Recognition With TensorFlow and Keras>>*
- [14] Liu, Xiaolong & Deng, Zhidong & Yang, Yuhan. Recent progress in semantic image segmentation. *Artificial Intelligence Review*. 1-18. 10.1007/s10462-018-9641-3. (2018).
- [15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, Fei-Fei Li: ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115(3): 211-252 (2015)
- [16] **Diane Lingrand**, «Introduction au Traitement d'images» Livre. Vuibert . ISBN-13:978-2711748662, 2e édition (2008)

- [17] Reconnaissance des Expressions Faciales à Base d'Informations Vidéo ; Estimation de l'Intensité des Expressions Faciales: KHADOUJIA GHANEMLE Université Mentouri de Constantine 2010.
- [18] Practical Machine Learning and Image Processing<< For Facial Recognition, Object Detection, and Pattern Recognition Using Python>> — Himanshu Singh
- [19] THÈSE PRÉSENTÉE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX <<Marquage automatique des images biologiques 2D: traitement d'images avec et sans méthodes d'apprentissage profond >> Soutenue le 19 Novembre 2019
- [20] **Nicolas Audebert** : Classification de données massives de télédétection. (Classification of big remote sensing data). Bretagne Loire University, Rennes, France (2018)
- [21] **Diaporama Automne 2011** Vision par ordinateur IMN 559 Presenter Par Pierre-Marc Jodoi page 4 et 5
- [22] Russell, S. et Norvig, P., Artificial Intelligence: A Modern Approach (2nd ed.) ISBN 0-13-790395-2; Prentice Hall, 2003
- [23] Bengio: Learning Deep Architectures for AI. Foundations and Trends in Machine Learning 2(1): 1-127 (2009)
- [24] Crevier, Daniel, AI: The Tumultuous Search for Artificial Intelligence, New York, NY: Basic Books, ISBN 978-0-465-02997-6, 1993
- [25] Nicolas Audebert: Classification de données massives de télédétection. (Classification of big remote sensing data). Bretagne Loire University, Rennes, France (2018)

