



وزارة البحث العلمي والتعليم العالي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة عبد الحميد بن باديس مستغانم
Université Abdelhamid Ibn Badis Mostaganem
كلية العلوم والتكنولوجيا
Faculté des Sciences et de la Technologie
Département de Génie Électrique



MEMOIRE

Pour obtenir le diplôme de

MASTER EN ELECTRONIQUE

Spécialité : Electronique des systèmes embarqués

Présenté par

BENZAZA Mohammed Adlene

RAHOU Zahira

Conception et réalisation d'un IHM sous python pour la détection des émotions faciales

Soutenu le 04/07/2023

devant le jury composé de :

Président :	M ^{me} ABBAD AMEL	Professeur	Université de Mostaganem
Examineur :	M ^{lle} BENDANI DJAZIA	MAA	Université de Mostaganem
Rapporteur :	M ^r BENTOUMI Mohamed	MCA	Université de Mostaganem

Remercîments

Louange à allah tout puissant qui nous a guidées pour l'accomplissement de ce travail.

Au terme de notre cycle d'études, il nous parait opportun de nous acquitter d'un devoir noble, celui de remercier tous ceux qui ont contribué par leur assistance tant morale que physique à notre cursus universitaire et à la réalisation de ce mémoire.

Nos remerciements s'adressent à notre encadreur Dr BENTOUMI Mohamed pour sa disponibilité et son aide jusqu'à la finalisation de ce travail. Qu'il daigne de recevoir nos sentiments de gratitude et de respect profond

Nous remercions les membres de jury d'avoir bien voulu étudier ce travail et participer la commission du jury. Qu'ils en soient louablement gratifiés.

Nous remercions tous les enseignants de la Faculté des Sciences et de la Technologie de l'université de Mostaganem pour la qualité de l'enseignement.

Nous remercions toutes les personnes qui nous ont aidées de près ou d loïn pour la réalisation de notre travail

Résumé

L'expression faciale joue un rôle essentiel dans la communication des émotions et des intentions chez les êtres humains. Cependant, la reconnaissance des expressions faciales est un défi pour les machines. De nos jours, la reconnaissance d'expression faciale est devenue une application cruciale dans de nombreux domaines tels que l'interaction homme-machine, la médecine, la sécurité et l'éducation. Pour parvenir à une interaction homme-machine efficace, un système de reconnaissance automatique des expressions faciales doit passer par des étapes clés telles que la détection du visage, l'extraction des caractéristiques faciales et la classification des expressions.

Dans ce projet, nous avons utilisé l'architecture de réseau de neurones convolutifs (CNN) pour reconnaître les émotions de base. Nous avons choisi cette architecture en raison de son succès dans la classification d'images, comme l'ont démontré nos résultats.

Notre système de reconnaissance des émotions de base a été entraîné sur la base de données MMAFEDB.

Mots clés associés à ce projet sont les suivants : réseaux de neurones, CNN, émotion, expressions faciales, MTCNN.

Summary

Facial expression plays a vital role in conveying emotions and intentions among human beings. However, facial expression recognition presents a challenge for machines. Nowadays, facial expression recognition has become a crucial application in various fields, such as human-computer interaction, medicine, security, and education. To achieve effective human-machine interaction, an automatic facial expression recognition system must go through key steps such as face detection, facial feature extraction, and expression classification. In this project, we employed Convolutional Neural Network (CNN) architecture to recognize basic emotions. We chose this architecture due to its success in image classification, as demonstrated by our results. Our basic emotion recognition system was trained using the MMAFEDB database.

Keywords associated with this project include: neural networks, CNN, emotion, facial expressions, and MTCNN.

ملخص

التعبيرات الوجهية تلعب دورًا أساسيًا في توصيل المشاعر والنوايا بين البشر. ومع ذلك، تعتبر تمييز التعبيرات الوجهية تحديًا أمام الآلات. في الوقت الحالي، أصبح التعرف على التعبيرات الوجهية تطبيقًا حيويًا في مجموعة متنوعة من المجالات مثل التفاعل بين الإنسان والآلة والطب والأمان والتعليم. لتحقيق تفاعل فعال بين الإنسان والآلة، يجب على نظام التعرف التلقائي على التعبيرات الوجهية أن يمر بخطوات رئيسية مثل كشف الوجه واستخراج سمات الوجه وتصنيف التعبيرات.

في هذا المشروع، استخدمنا هندسة الشبكة العصبية المستدامة للاعتراف بالمشاعر الأساسية. اخترنا هذه الهندسة بناءً على نجاحها في تصنيف الصور، كما أظهرت نتائجنا. تم تدريب نظامنا للاعتراف بالمشاعر الأساسية على قاعدة بيانات

MMAFEDB

الكلمات الرئيسية المرتبطة بهذا المشروع تشمل الشبكات العصبية والهندسة الشبكية المستدامة والمشاعر والتعبيرات الوجهية

و.MTCNN

Table des matières

Remercîments	1
Résumé	2
Table des matières	3
Liste des figures	5
Liste des tableaux	7
Introduction générale	9
Chapitre 1 : Reconnaissance des expressions faciales	12
1. Introduction.....	12
2. Emotion	12
3. Expression faciale	13
3.1. Définition	13
3.2. Différents types des expressions faciales	13
3.3 Description des expressions faciales universelles	14
4. Reconnaissance des expressions faciales	15
4.1. Les étapes de la reconnaissance des expressions faciales.....	15
4.2. Détection du visage	16
4.2.1. HAAR Cascade.....	17
4.2.2. MTCNN	18
4.3. Extraction des caractéristiques	21
4.4. Classification.....	21
4.5. Base de données d'expressions faciales.....	22
4.6. Apprentissage automatique	24
4.6.1. Apprentissage profond (Deep Learning).....	24
4.6.2. Différence entre les méthodes de traitement d'image basé sur apprentissage automatique traditionnelle et le deep learning	25
4.6.3. Les réseaux de neurones convolutifs CNN	25
4.7. Domaines d'applications	26
5. Conclusion	27

Tables des matières

Chapitre 02 : Conception du système de reconnaissance des expressions faciales	29
1. Introduction.....	29
2. Conception	29
2.1. Présentation de notre système de reconnaissance des émotions	29
2.2 Architecture Globale du système de reconnaissance des émotions	30
2.3. Phase d'apprentissage et de prédiction	31
2.3.1 Phase d'apprentissage	31
2.3.2 Phase de prédiction	32
3. Conclusion	35
Chapitre 03 : Implémentation et représentation de l'interface	37
1 Introduction.....	37
2. Environnements de travail.....	37
2.1 Environnement matériel.....	37
2.2 Environnement logiciel.....	37
2.2.1 Python	37
2.2.2 Anaconda.....	38
2.3 Les bibliothèques.....	39
2.3.1 OpenCV	39
2.3.2 Tkinter	39
2.3.3 Keras	39
2.3.4 Tensorflow	40
3. Classifieur	40
3.1 Préparation des données pour l'apprentissage (l'entraînement)	40
3.2 L'apprentissage du classifieur d'images de réseau neuronal convolutionnel avec Keras et Tensorflow.....	41
4. Mise en œuvre du classifieur.....	43
5. Les tests : Présentation de l'application avec une interface homme-machine	44
5.1 Discussion des résultats.....	48
6. Conclusion	48
Conclusion Générale	51
Bibliographie	54

Liste des figures

Figure 1:: La Joie d'un enfant [1]	12
Figure 2:: Exemple des expressions faciales [3].....	13
Figure 3: Architecture d'un système de reconnaissance des expressions faciales.....	16
Figure 4: Détection des visages [9].....	17
Figure 5 :Les 5 différents types de caractéristiques de type Haar extraites d'un patch d'image [11].....	18
Figure 6: Introduites pour la première fois en 2001, les cascades de Haar sont une classe d'algorithmes de détection d'objets [12]	18
Figure 7: Représentation architecturale du réseau en cascade MTCNN [14].....	19
Figure 8: MTCNN : Les étapes de l'architecture en cascade prise pour garantir les meilleures performances dans la détection de visages humains [15]	20
Figure 9: exemple de base de données pour expressions faciales [21].....	23
Figure 10: Relation entre l'intelligence artificielle, l'apprentissage automatique et le Deep Learning [25]	24
Figure 11: La différence principale entre l'apprentissage automatique traditionnel et les algorithmes d'apprentissage profond [26]	25
Figure 12: organigramme du système de reconnaissance des expressions faciales en temps réel	30
Figure 13: Architecture globale de reconnaissance des émotions réalisées.....	31
Figure 14: Schéma globale du système de Reconnaissance des émotions réalisé	34
Figure 15: Schéma présentant l'architecture du modèle de classification des émotions	35
Figure 16: Logo de Python	39
Figure 17: Logo d'anaconda.....	39
Figure 18: Logo d'OpenCV	40
Figure 19:: Logo Keras.....	41
Figure 20: Logo Tenserflow	41

<u>Figure 21: Echantillon d'image de la Base de données MMA FACIAL EXPRESSION (MMAFEDB) [30]</u>	42
<u>Figure 22: Diagramme de cas d'utilisation avec le langage UML de notre système de reconnaissance des expressions faciales</u>	45
<u>Figure 23: Interface au démarrage du système</u>	46
<u>Figure 24: Détection de visage avec la méthode MTCNN</u>	48
<u>Figure 25: Détection de visage avec la méthode OPENCV</u>	48
<u>Figure 26: détection d'émotion « Happy » Joie</u>	49
<u>Figure 27: Détection d'émotion « Sad» Triste</u>	50

Liste des tableaux

Tableau 1: Différentes classifications d'expressions faciales.....	14
Tableau 2: Quelques bases de données qui utilise pour la reconnaissance des expressions faciales.....	23

Introduction générale

Introduction générale

La reconnaissance des émotions faciales est un domaine d'étude qui permet de mieux comprendre les influences des émotions sur nos vies. En comprenant les expressions faciales, nous pouvons améliorer nos interactions sociales et notre bien-être général. Cela ouvre de nouvelles perspectives pour optimiser nos relations avec les autres et améliorer notre compréhension mutuelle.

La reconnaissance faciale des émotions est un domaine de recherche qui vise à détecter et interpréter les expressions faciales afin de déterminer l'état émotionnel d'une personne. Cette technologie utilise des algorithmes d'apprentissage automatique pour analyser les caractéristiques faciales telles que les mouvements musculaires, les plis et les expressions, afin de reconnaître les émotions telles que la joie, la tristesse, la colère, la peur, etc.

Comme beaucoup de technologie, la reconnaissance faciale des émotions trouve des applications dans de nombreux domaines, tels que la psychologie, la recherche clinique, la sécurité, le marketing, et même dans les interfaces homme-machine. En psychologie et en recherche clinique, cette technologie peut être utilisée pour évaluer l'état émotionnel des individus, faciliter le diagnostic des troubles mentaux et suivre l'efficacité des thérapies. Dans le domaine de la sécurité, elle peut contribuer à la détection des comportements suspects en temps réel. Dans le domaine du marketing, elle peut aider à évaluer les réactions des consommateurs face à des produits, des publicités ou des expériences utilisateur.

Les méthodes utilisées dans la reconnaissance faciale des émotions peuvent être classées en deux catégories principales : les méthodes basées sur les traits géométriques du visage et les méthodes basées sur l'apprentissage automatique.

Les méthodes basées sur les traits géométriques se concentrent sur l'analyse des caractéristiques géométriques du visage, telles que la position des yeux, du nez et de la bouche, ainsi que sur les mouvements des muscles faciaux. Ces méthodes utilisent souvent des modèles mathématiques et des algorithmes spécifiques pour extraire les caractéristiques.

Les méthodes basées sur l'apprentissage automatique, en revanche, utilisent des réseaux de neurones profonds pour analyser les images faciales et extraire automatiquement les caractéristiques émotionnelles. Ces méthodes nécessitent un grand ensemble de données d'entraînement contenant des visages étiquetés avec les émotions correspondantes, afin de pouvoir apprendre à reconnaître et à prédire les émotions à partir des images étiquetées.

L'objectif de ce travail est de présenter ce mémoire dans le cadre de notre projet de fin d'étude et qui consiste à concevoir et à réaliser une IHM sous python pour la détection des émotions faciales, nous explorons aussi les méthodes utilisées pour la reconnaissance faciale des émotions.

Structure du mémoire

Ce mémoire se présente sous forme de trois chapitres :

Le chapitre 1 présente un aperçu général sur la reconnaissance des expressions faciale. Le chapitre 2 est consacré à la conception de notre système

Le chapitre 3 illustre l'implémentation et les résultats de notre système, les outils et logiciels que nous avons eu à utiliser pour le développement, et la réalisation de notre système.

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

Chapitre 1
Reconnaissance des expressions faciales

Chapitre 1 : Reconnaissance des expressions faciales

1. Introduction

Les émotions sont des expériences subjectives qui influencent notre état interne et notre comportement. Elles peuvent être déclenchées par divers stimuli et se manifestent à travers différents aspects.

L'expression faciale est un canal de communication qui permet de transmettre les émotions. Les théories et modèles des émotions cherchent à décrire et à catégoriser ces expressions, en identifiant des émotions universelles telles que la joie, la tristesse, la colère, la peur, le dégoût et la surprise.

Ce chapitre portera sur la reconnaissance des expressions faciales.

2. Emotion

Les émotions sont des réponses psychologiques et physiologiques complexes qui sont déclenchées en réponse à des stimuli internes ou externes. Les émotions peuvent inclure la joie, la tristesse, la colère, la peur, le dégoût, la surprise et bien d'autres.



Figure 1:: La Joie d'un enfant [1]

3. Expression faciale

3.1. Définition

L'expression faciale est une mimique faciale chargée de sens, c'est un changement dans le visage, perceptible visuellement, dû à l'activation ou mouvements de l'un ou de plusieurs muscles composant le visage. Ces mouvements transportent l'état émotionnel de l'individu aux observateurs. Les émotions sont au centre de l'explication des expressions faciales. [2]



Figure 2.: Exemple des expressions faciales [3]

3.2. Différents types des expressions faciales

Le tableau représenté ci-dessous (Tableau 1) [4] présente une classification des expressions faciales selon certains chercheurs à base d'émotion et d'inclusion.

Théoriciens	Emotions de base	Base d'inclusion
Plutchik	Acceptation, colère, anticipation, dégoût, joie, peur, tristesse, surprise	Relation aux processus biologiques adaptatifs
Arnold	Colère, aversion, courage, abattement, désir, désespoir, peur, haine, espoir, amour, tristesse	Relation aux tendances d'action
Frijda	Désir, bonheur, intérêt, surprise, émerveillement, chagrin	Désir, bonheur, intérêt, surprise, émerveillement, chagrin
McDougall	Colère, dégoût, exaltation, peur,	Relation à l'instinct

	soumission, émotion tendre, émerveillement	
Ekman , Friesen Et Ellsworth	dégoût, joie, peur, tristesse, surprise	Expressions faciales universelles

Tableau 1: Différentes classifications d'expressions faciales

3.3 Description des expressions faciales universelles

Lors de la production d'une expression faciale, il apparaît sur le visage un ensemble de déformation au niveau des traits permanents du visage.

Les émotions les plus fréquemment étudiées et utilisées sont les six émotions d'Ekman : la peur, la colère, la joie, la tristesse, le dégoût, la surprise. [5]

Joie : Elle se caractérise par l'état d'une personne dans une condition de satisfaction intense [5]. Elle est due par rapport au désir, à la réussite, au bien-être, et l'accomplissement [6] mais aussi l'approche.

Tristesse : c'est l'état d'une personne qui souffre moralement suite à une insatisfaction et des soucis [5]. Elle est souvent due soit à une perte, ou un deuil, ou un obstacle [6] la personne se replie souvent sur soi.

Colère : c'est l'état d'une personne dans une réaction violente et agressive lors d'une contrariété [5] Elle est souvent due soit à une injustice, ou un dommage, atteinte au système de valeurs [6] La plupart des personnes en ce moment attaquent.

Dégoût : c'est l'état d'une personne qui a une répugnance pour certains aliments ou à un manque d'appétit [5]. Elle est souvent due soit à un rejet, ou contre quelqu'un, ou à une aversion [6]. La plupart des personnes en ce moment préfèrent se retirer.

Peur : c'est l'état d'une personne menacée par un danger réel ou imaginaire [6]. Elle est souvent due soit à une menace, ou à un danger, ou à des inconnus [6]. La plupart des personnes en ce moment préfèrent prendre la fuite.

Surprise : c'est l'état d'une personne étonnée par quelque chose d'inattendu [5] Elle est souvent due soit à un danger immédiat, ou à un imprévu, ou à des inconnus [5]. La plupart des personnes en ce moment préfèrent prendre la fuite ou elles sursautent.

4. Reconnaissance des expressions faciales

La reconnaissance des expressions faciales désigne la capacité d'un système ou d'un modèle informatique à détecter, analyser et interpréter les différentes expressions du visage humain. Elle consiste à identifier et à comprendre les émotions et les états émotionnels d'une personne en se basant sur les mouvements et les configurations des muscles du visage.

La reconnaissance des expressions faciales utilise généralement des techniques d'intelligence artificielle, telles que la vision par ordinateur et l'apprentissage automatique. Ces techniques permettent au système d'analyser les différentes caractéristiques du visage, comme les sourcils, les yeux, la bouche, les rides, et d'associer ces informations à des émotions spécifiques, telles que la joie, la tristesse, la colère, la surprise, etc.

4.1. Les étapes de la reconnaissance des expressions faciales

La reconnaissance automatique des expressions faciale dans un système se décompose généralement de ces étapes (figure1.3):

-Acquisition

-Détection du visage

-Extraction des caractéristiques

-Reconnaissance ou classification d'expression faciale (émotion)

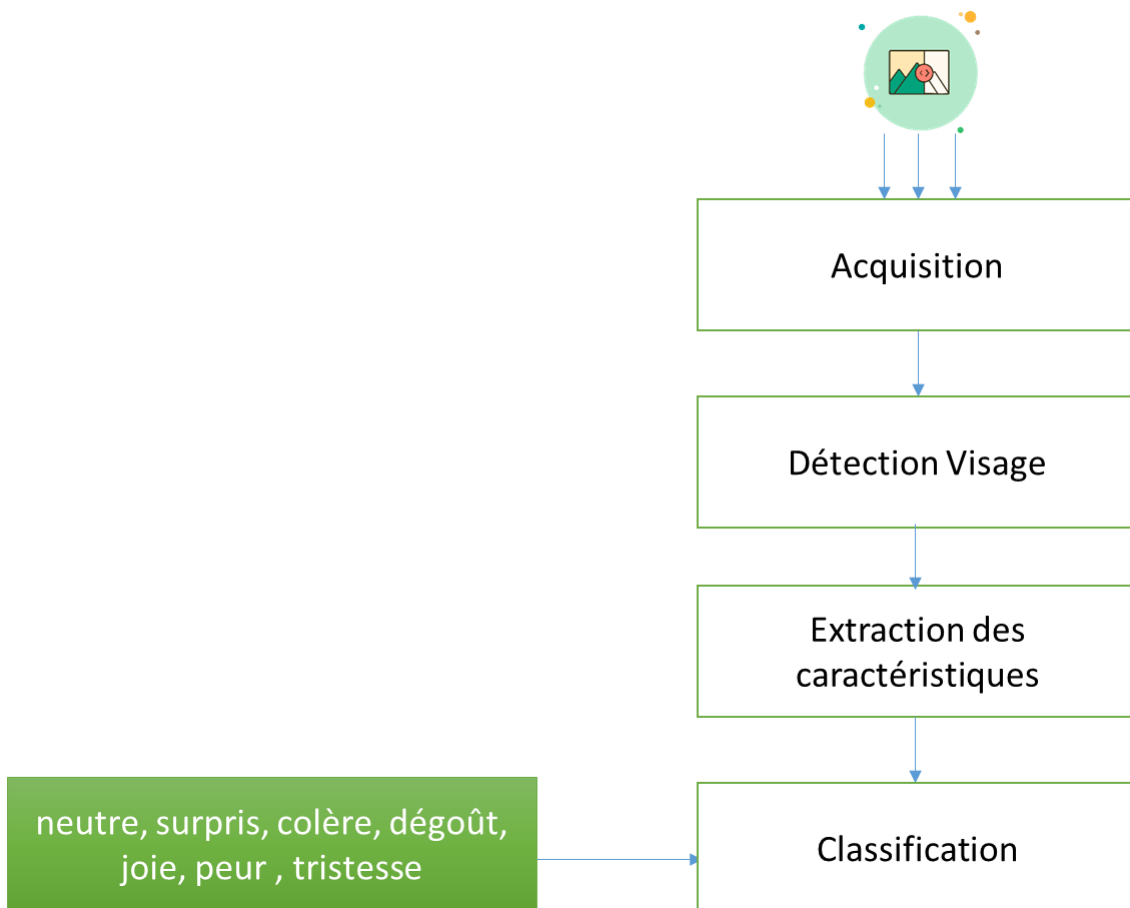


Figure 3: Architecture d'un système de reconnaissance des expressions faciales

4.2. Détection du visage

La première étape consiste à localiser les visages dans l'image. Différentes techniques peuvent être utilisées, telles que la méthode de Viola-Jones avec l'algorithme de Haar cascade ou des approches plus avancées telles que MTCNN (Multi-Task Cascaded Convolutional Networks). Nous présentons ci-dessous ces deux méthodes. [7] [8]



Figure 4: Détection des visages [9]

4.2.1. HAAR Cascade

La méthode des cascades de Haar a été proposée en 2001 dans un article de recherche intitulé "Rapid Object Detection using a boosted cascade of simple feature" par Paul Viola et Michael Jones. Elle est rapidement devenue célèbre, car en plus d'obtenir un taux de succès relativement élevé sur un Jeu de données difficiles "MIT+CMU Dataset", elle était 15 fois plus rapide que les autres méthodes de détection de visage à l'époque. Bien que la méthode ait été spécialement conçue pour détecter des visages, la communauté s'est rapidement rendu compte que la méthode s'appliquait aux autres membres du corps, aux objets, et de manière générale à tout type d'image. [7] [10]

Le principe de la méthode est le glissement d'une fenêtre de taille fixe sur une image à plusieurs échelles. À chacune de ces phases, la fenêtre s'arrête, calcule certaines caractéristiques, puis classe la région comme suit : Oui, cette région contient un visage, ou Non, cette région ne contient pas de visage. Cela nécessite un peu d'apprentissage automatique.

Nous avons besoin d'un classifieur entraîné à utiliser des échantillons positifs et négatifs d'un visage :

- Les points de données positifs sont des exemples de régions contenant un visage.
 - Les points de données négatifs sont des exemples de régions qui ne contiennent pas de visage.
- Avec ces points de données positifs et négatifs, nous pouvons "entraîner" un classifieur à reconnaître si une région donnée d'une image contient un visage. Pour chacun des arrêts le long du chemin de la fenêtre glissante, cinq caractéristiques rectangulaires sont calculées :

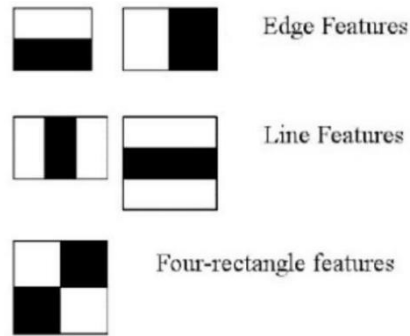


Figure 5 :Les 5 différents types de caractéristiques de type Haar extraites d'un patch d'image [11]

L'obtention des caractéristiques se fait par la soustraction de la somme des pixels dans la région blanche et celle des pixels dans la région noire. Ces caractéristiques ont une réelle importance dans le contexte de la détection des visages puisqu'elles permettent de classer les parties d'un visage, à savoir :

- Les régions des yeux ont tendance à être plus sombres que celles des joues.
- La région du nez est plus claire que la région des yeux.

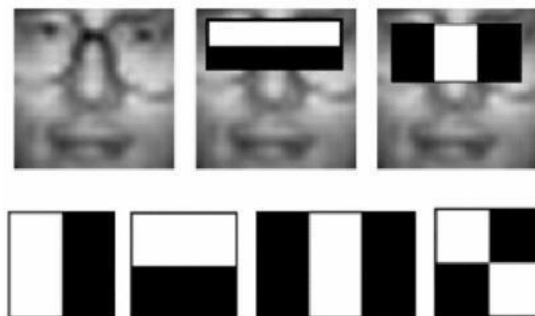


Figure 6: Introduites pour la première fois en 2001, les cascades de Haar sont une classe d'algorithmes de détection d'objets [12]

4.2.2. MTCNN

MultiTask Cascaded Convolutional Neural Network est un outil moderne de détection de visage, son architecture de réseaux de neurones combine trois réseaux à convolutions (P-Net,

R-Net et O-Net) connectés en cascade afin de sélectionner séquentiellement les meilleures fenêtres candidates à la présence de visages dans l'image. La Figure 1.5 développe la constitution de chacun de ces blocs de convolution. On y retrouve les trois réseaux P-Net, R- Net, O-Net, opérant des séries de convolutions aux images afin de produire par régression la hiérarchisation des visages, les valeurs des quatre coins du cadre contenant le visage potentiellement détecté, ainsi que les cinq points d'intérêt du visage (facial landmarks). Chaque CNN correspond à un étage pour lequel un ensemble de cadre vont être sélectionnés selon leur probabilité de contenir un visage. Le dernier étage va produire le meilleur cadre contenant l'intégralité du visage de l'image ainsi que les cinq coordonnées des principaux points d'intérêt du visage (ou facial landmarks) : deux pour les yeux, deux pour la bouche et un pour le nez. [13]

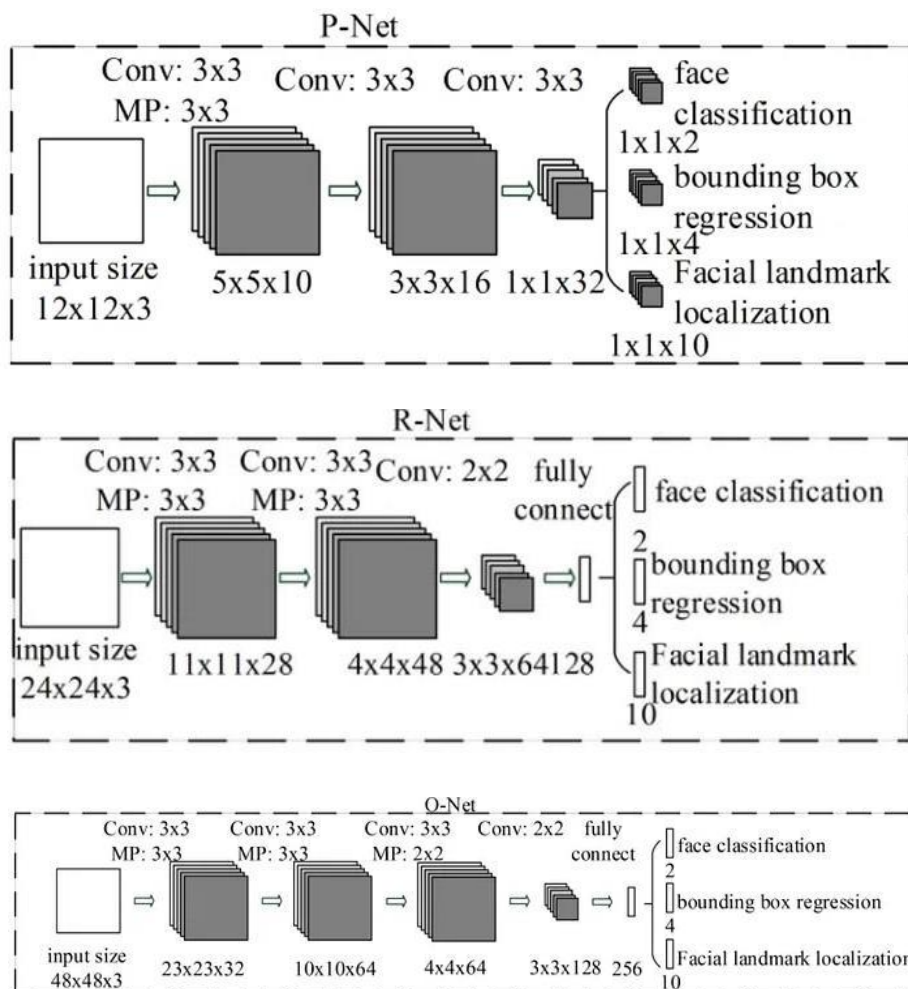


Figure 7: Représentation architecturale du réseau en cascade MTCNN [14]

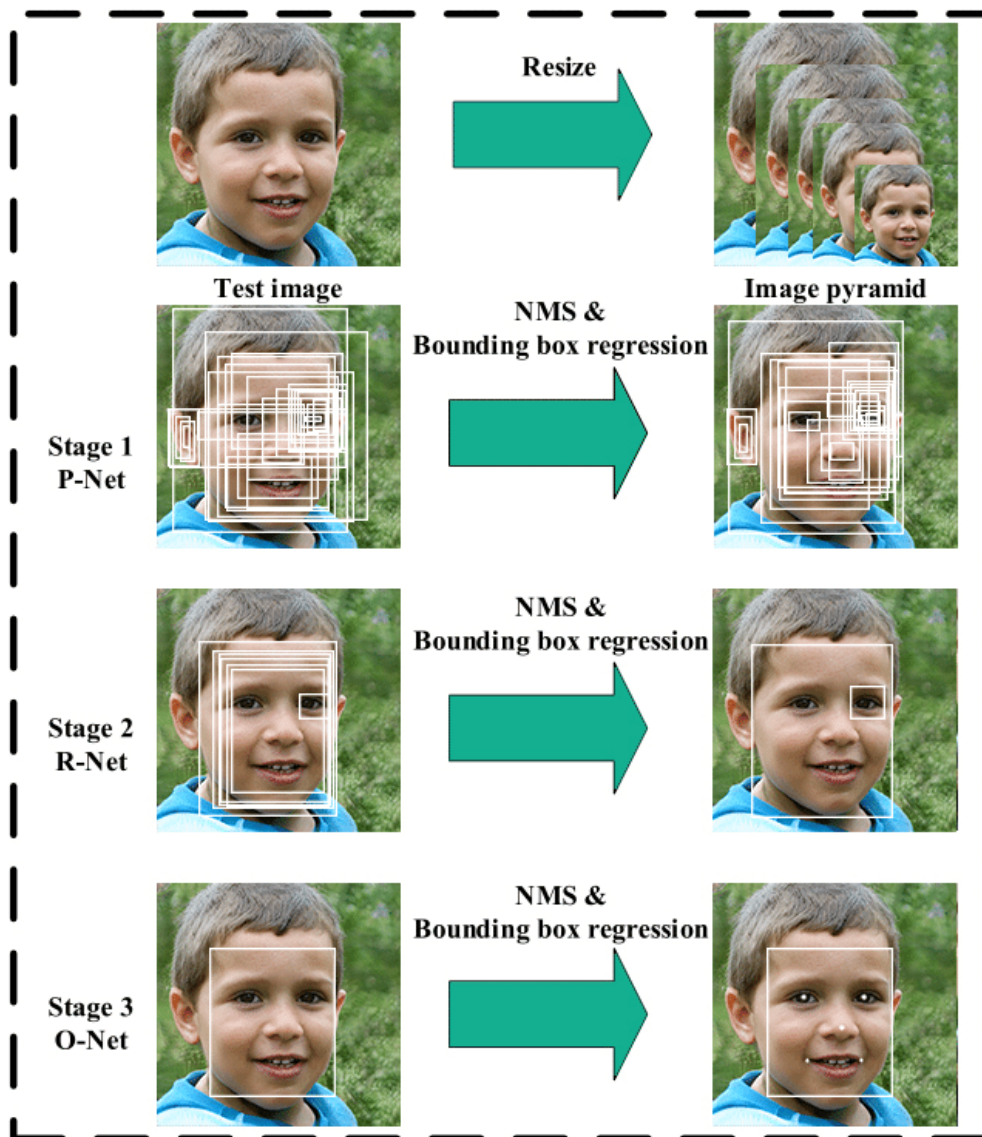


Figure 8: MTCNN : Les étapes de l'architecture en cascade prise pour garantir les meilleures performances dans la détection de visages humains [15]

L'image est redimensionnée plusieurs fois pour détecter des visages de différentes tailles. Ensuite, le P-Net (Proposal Net) scanne les images, effectuant la première détection. Il a un seuil de détection bas et détecte donc de nombreux faux positifs, même après NMS (NonMaximum Suppression), mais fonctionne comme ça à dessein. Les régions proposées (contenant de nombreux faux positifs) sont entrées pour le deuxième réseau, le R-Net (Refinement Network), qui, filtre les détections (également avec NMS) pour obtenir des boîtes englobantes assez précises.

La dernière étape, l'O-Net (Output Network) effectue le raffinement final des boîtes englobantes. De cette façon, non seulement les visages sont détectés, mais les cadres de délimitation sont très justes et précis.

Il est utilisable sous implémentation TensorFlow, et aussi sous celle de PyTorch qui est plus rapide. Il est utilisable aussi pour détection en temps réel. MTCNN est très précis et robuste. Il détecte correctement les visages même avec des tailles, un éclairage et des rotations importantes. Mais il est plus lent par rapport à quelques détecteurs comme Haar-cascade, HOG et DNN. Un autre désavantage est qu'il détecte des faux négatifs, chose à laquelle nous pouvons remédier en modifiant le seuil de MTCNN. [16]

4.3. Extraction des caractéristiques

Après la détection du visage, l'étape suivante consiste à extraire et à représenter les changements faciaux causés par une expression faciale. L'obtention de caractéristiques efficaces d'expression faciale, à partir de ROI(s) extraite(s), est cruciale pour une reconnaissance d'expressions faciales réussie. Les expressions faciales sont définies principalement par la contraction des muscles faciaux qui produisent des changements dans l'apparence et la forme du visage [17].

Les caractéristiques géométriques représentent la forme et l'emplacement des composants du visage (y compris la bouche, les yeux, les sourcils et le nez). Les composants faciaux ou les traits faciaux sont extraits pour former un vecteur de caractéristiques représentant la géométrie du visage.

Les caractéristiques d'apparence représentent les changements d'apparence (texture de la peau) du visage, tels que les rides et les sillons.

Selon les différentes méthodes d'extraction des caractéristiques, les effets de la rotation de la tête dans le plan et les différentes échelles de prise de vue du visage peuvent être éliminés par une normalisation de ce dernier avant l'extraction des caractéristiques ou par une représentation des caractéristiques avant l'étape de reconnaissance d'expression. [18]

4.4. Classification

La dernière étape d'un système automatique d'analyse d'expression faciale est la reconnaissance de l'expression faciale en fonction des caractéristiques extraites.

Certains systèmes classent directement les expressions. De nombreux classifieurs ont été appliqués à la reconnaissance d'expression tels que : [19]

- Réseaux de neurone (Neural Networks, NN).
- Machines à vecteurs de support (Support Vector Machine, SVM).
- Analyse discriminante linéaire (Linear discriminant analysis, LDA).
- K-plus proche voisin (K Nearest Neighbor, KNN).
- Régression logistique multinomiale (Multinomial Regression Logistic, MRL).
- Modèles de markov cachés (Hidden Markov Model, HMM).
- Réseaux bayésiens (Bayesian Network, BN).

4.5. Base de données d'expressions faciales

L'un des aspects les plus importants du développement de tout nouveau système de reconnaissance ou de détection d'expression faciale est le choix de la base de données qui sera utilisée pour tester ce nouveau système. De plus, des bases de données communes sont nécessaires pour évaluer les algorithmes de manière comparative. Les bases de données disponibles peuvent être classées en deux catégories : les bases d'expressions faciales spontanées et les bases d'expressions faciales posées. [20]

Le tableau suivant présente quelques bases de données qui sont utilisées pour la reconnaissance des expressions faciales [20] :

La base de données Description Lien	La base de données Description Lien	La base de données Description Lien
FER2013	compte environ 37 000 personnes bien structurées En Images de gris à 48X48 pixels les visages regroupés automatiquement par l'API Google de recherche d'image	https://www.kaggle.com/datasets
MMI	Plus de 2900 séquences vidéo et images haute	https://mmifacedb.eu/

	résolution images de 75 sujets _ 238 séquences vidéo sur 28 sujets, hommes et femmes _ Résolution d'image de 720 _ 576	
B+	16 128 images faciales 28 sujets distincts pour 576 conditions	http://vision.ucsd.edu/content/ extended-yale-face-database-b-b
JAFFE	213 images de sept émotions faciales Dix modèles féminins japonais Six adjectifs d'émotion de 60 sujets japonais Résolution d'image 256 256	http://www.kasrl.org/jaffe_info.htm 1

Tableau 2: Quelques bases de données qui utilise pour la reconnaissance des expressions faciales

























base de données	joie	tristesse	Dégoût	surprise	colère	peur
FER 2013						
						
MMA FACIAL EXPRESSION						
						

Figure 9: exemple de base de données pour expressions faciales [21]

4.6. Apprentissage automatique

L'apprentissage automatique (en anglais Machine Learning) est un champ d'études de l'intelligence artificielle qui se fonde sur des approches statistiques pour donner aux ordinateurs la capacité d'apprendre à partir de données. L'apprentissage automatique comporte généralement deux phases. La première consiste à calculer un modèle à partir de données, appelées l'apprentissage. La seconde phase correspond à la mise en production, le modèle étant déterminé, de nouvelles données qui peuvent alors être soumises afin d'obtenir le résultat correspondant à la tâche souhaitée (prédiction). [22]

4.6.1. Apprentissage profond (Deep Learning)

Le deep learning, également appelé apprentissage profond, est une approche de l'intelligence artificielle qui utilise des réseaux de neurones artificiels pour apprendre automatiquement à partir de données. Dans le contexte de la reconnaissance des émotions, le deep learning est utilisé pour analyser les expressions faciales, les sons ou les textes afin de détecter et de comprendre les différentes émotions exprimées par les individus. [24]

L'apprentissage profond est l'une des nombreuses approches de l'apprentissage automatique. [23]

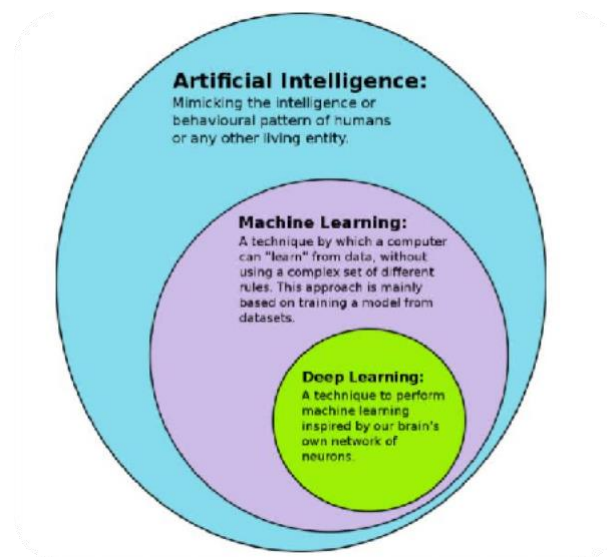


Figure 10: Relation entre l'intelligence artificielle, l'apprentissage automatique et le Deep Learning [25]

4.6.2. Différence entre les méthodes de traitement d'image basé sur apprentissage automatique traditionnelle et le deep learning

La différence principale entre les algorithmes de l'apprentissage automatique traditionnel et les algorithmes d'apprentissage profond, réside dans l'ingénierie des fonctions. Dans les algorithmes traditionnels d'apprentissage automatique, nous devons façonner les caractéristiques à la main (handcrafted). En revanche, dans les algorithmes d'apprentissage en profondeur, l'ingénierie des caractéristiques est effectuée automatiquement par l'algorithme. [26]

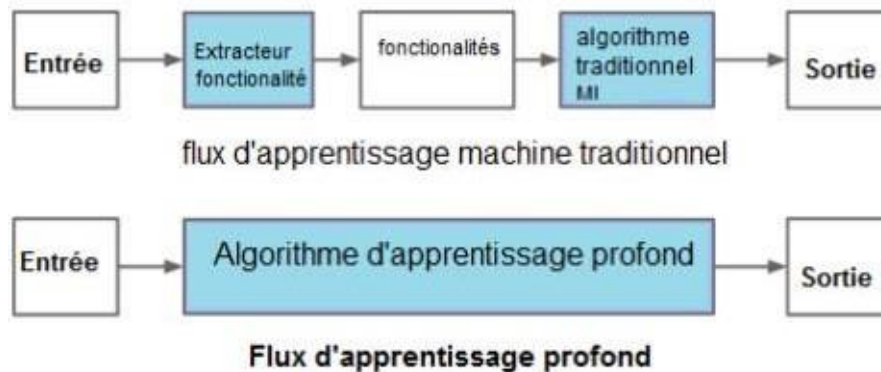


Figure 11: La différence principale entre l'apprentissage automatique traditionnel et les algorithmes d'apprentissage profond [26]

4.6.3. Les réseaux de neurones convolutifs CNN

Les réseaux de neurones convolutifs CNN (Convolutional Neural Network) , sont les structures les plus performantes pour faire la classification des images, Le nom « réseau de neurones convolutif » indique que le réseau emploie une opération mathématique appelée convolution. La convolution est une opération linéaire. Les réseaux convolutifs sont simplement des réseaux de neurones qui utilisent la convolution à la place de la multiplication matricielle dans au moins une de leurs couches. Ils comportent deux parties principales. S'il y a en entrée, une image qu'elle doit être sous la forme d'une matrice de pixels, de 2 dimensions pour une image en niveaux de gris et en 3 dimensions si elle est en couleur, pour représenter les couleurs fondamentales [Rouge, Vert, Bleu]. Cette image passe par la première partie d'un CNN qui est la partie convolutive Elle fonctionne comme un extracteur de caractéristiques des images.

L'image passe à travers une succession de filtres, ou noyaux de convolution, pour la transformer en nouvelles images appelées cartes de convolutions « feature maps ». Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN. Le résultat en sortie de la partie convolutive est branché en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron multicouche) qui consiste à combiner les caractéristiques de tout le réseau pour classer l'image et à la sortie qui est une couche comportant un neurone par classe, on obtient des valeurs numériques généralement normalisées entre 0 et 1, pour présenter la distribution de probabilité sur les classes. [27]

4.7. Domaines d'applications

Voici quelques-uns des domaines d'application courants de la reconnaissance des expressions faciales :

1- Interaction homme-machine : La reconnaissance des expressions faciales est utilisée pour permettre une interaction plus naturelle entre l'homme et la machine. Les systèmes d'interface utilisateur peuvent détecter les expressions faciales des utilisateurs pour comprendre leurs émotions, leurs intentions et leurs besoins, ce qui permet des interactions plus personnalisées et adaptatives.

2- Évaluation des émotions : La reconnaissance des expressions faciales est utilisée pour évaluer et analyser les émotions humaines. Elle peut être utilisée dans des domaines tels que la recherche en psychologie, la compréhension de la réaction des consommateurs à des produits ou des publicités, l'évaluation des réactions des patients dans le domaine de la santé mentale, etc.

3- Surveillance et sécurité : La reconnaissance des expressions faciales est utilisée pour la surveillance et la sécurité. Elle peut être utilisée pour détecter les émotions suspectes ou agressives dans des environnements publics ou sensibles, tels que les aéroports, les gares, les centres commerciaux, etc. Elle peut également être utilisée pour la détection de la fatigue au volant, la surveillance des conducteurs, etc.

4- Réalité virtuelle et jeux : La reconnaissance des expressions faciales est utilisée dans les applications de réalité virtuelle et de jeux pour améliorer l'expérience immersive. Elle permet

aux systèmes de détecter et de réagir aux expressions faciales des utilisateurs, ce qui permet des interactions plus naturelles et engageantes.

5- Santé et bien-être : La reconnaissance des expressions faciales est utilisée dans le domaine de la santé et du bien-être. Elle peut être utilisée pour détecter et surveiller les expressions faciales liées aux émotions, au stress, à la douleur, etc. Cela peut être utile dans les domaines de la santé mentale, de la médecine comportementale, de la réadaptation, etc.

6- Analyse du comportement : La reconnaissance des expressions faciales est utilisée pour l'analyse du comportement humain. Elle peut être utilisée pour détecter et analyser les expressions faciales liées à des comportements tels que l'attention, l'intérêt, l'ennui, la confusion, etc. Cela peut être appliqué dans des domaines tels que l'éducation, la recherche en marketing, les études sur l'interaction sociale, etc.

5. Conclusion

Dans ce chapitre on a présenté les théories et les représentations les plus utilisées dans le domaine de la reconnaissance des expressions faciales. Nous avons examiné quelques approches de détection des visages dans des images en temps réel, l'extraction des caractéristiques et la classification. De plus, nous avons décrit les bases de données couramment utilisées dans ce domaine. Nous avons choisi d'adopter le Deep Learning avec les CNN comme solution pour notre système de reconnaissance des expressions faciales.

Dans le prochain chapitre, nous aborderons la conception de notre système de reconnaissance des expressions faciales en détaillant les choix et les solutions spécifiques que nous avons adoptés.

Chapitre 02
***Conception du système de reconnaissance
des expressions faciales***

Chapitre 02 : Conception du système de reconnaissance des expressions faciales

1. Introduction

La reconnaissance des expressions faciales constitue un défi majeur dans le domaine de la vision artificielle, et de nombreuses méthodes ont été explorées pour le résoudre, en mettant notamment l'accent sur l'apprentissage profond, en particulier les réseaux de neurones convolutifs (CNN).

Dans ce chapitre, nous nous pencherons en détail sur la conception de notre approche visant à développer un système de reconnaissance des émotions de base.

2. Conception

2.1. Présentation de notre système de reconnaissance des émotions

Nous proposons une conception d'un système de reconnaissance des émotions de base en temps réel à partir de visages. Le système détecte les visages des personnes via une caméra et identifie leur expression faciale parmi les sept expressions universelles : joie, dégoût, peur, colère, tristesse, surprise et neutralité.

En utilisant des techniques avancées de traitement d'images et d'apprentissage automatique, le système analyse les caractéristiques faciales pour reconnaître et classifier les émotions. Les méthodologies, y compris les algorithmes et les modèles utilisés, sont présentés.

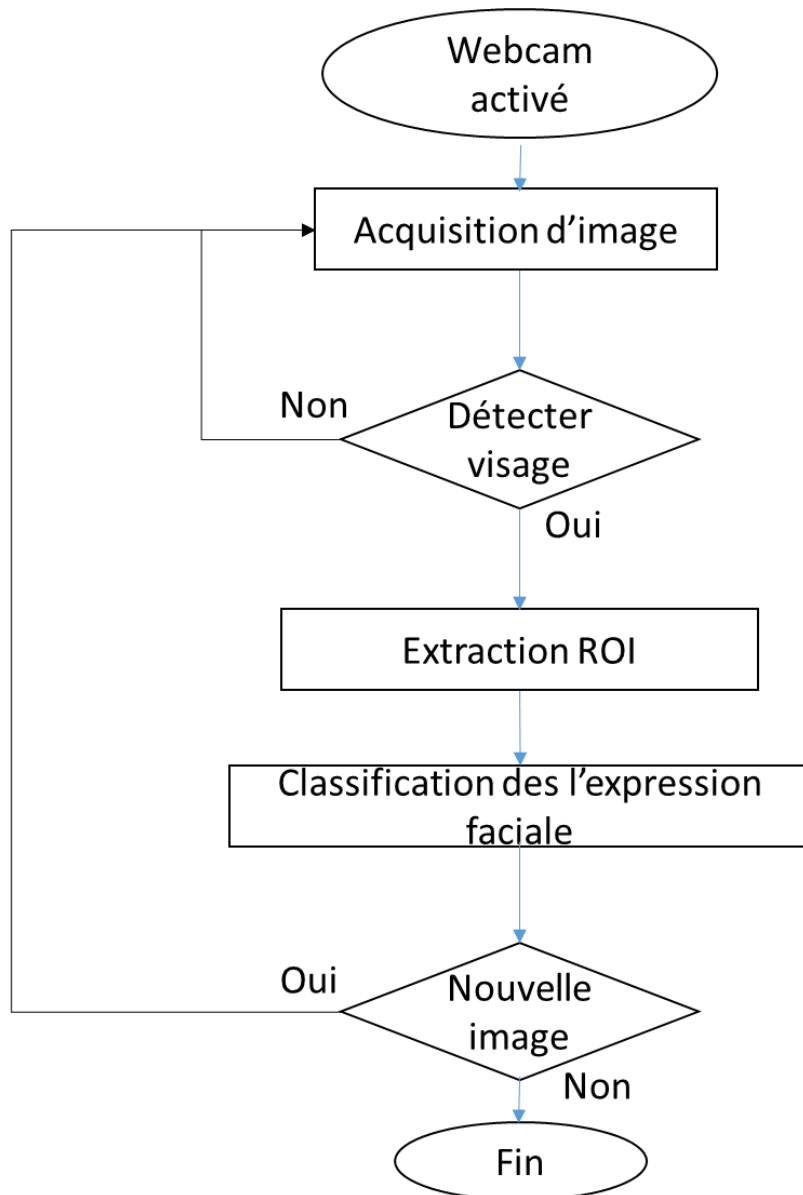


Figure 12: organigramme du système de reconnaissance des expressions faciales en temps réel

2.2 Architecture Globale du système de reconnaissance des émotions

La plupart des approches de reconnaissance des émotions suivent une architecture globale similaire, composée de deux modules distincts fonctionnant de manière autonome (figure 2.2 et figure 2.3) :

-La détection de visage

- La classification

Le premier module, la détection de visage, vise à localiser et à isoler les visages présents dans une image ou une vidéo pour obtenir les visages détectés, le deuxième module entre en jeu pour la classification, ce module utilise ces caractéristiques pour attribuer une émotion spécifique à chaque visage détecté. Cette approche en 2 étapes distinctes permet une détection et une reconnaissance précises des émotions, en séparant les tâches de manière claire et en utilisant des algorithmes spécifiques pour chaque module.

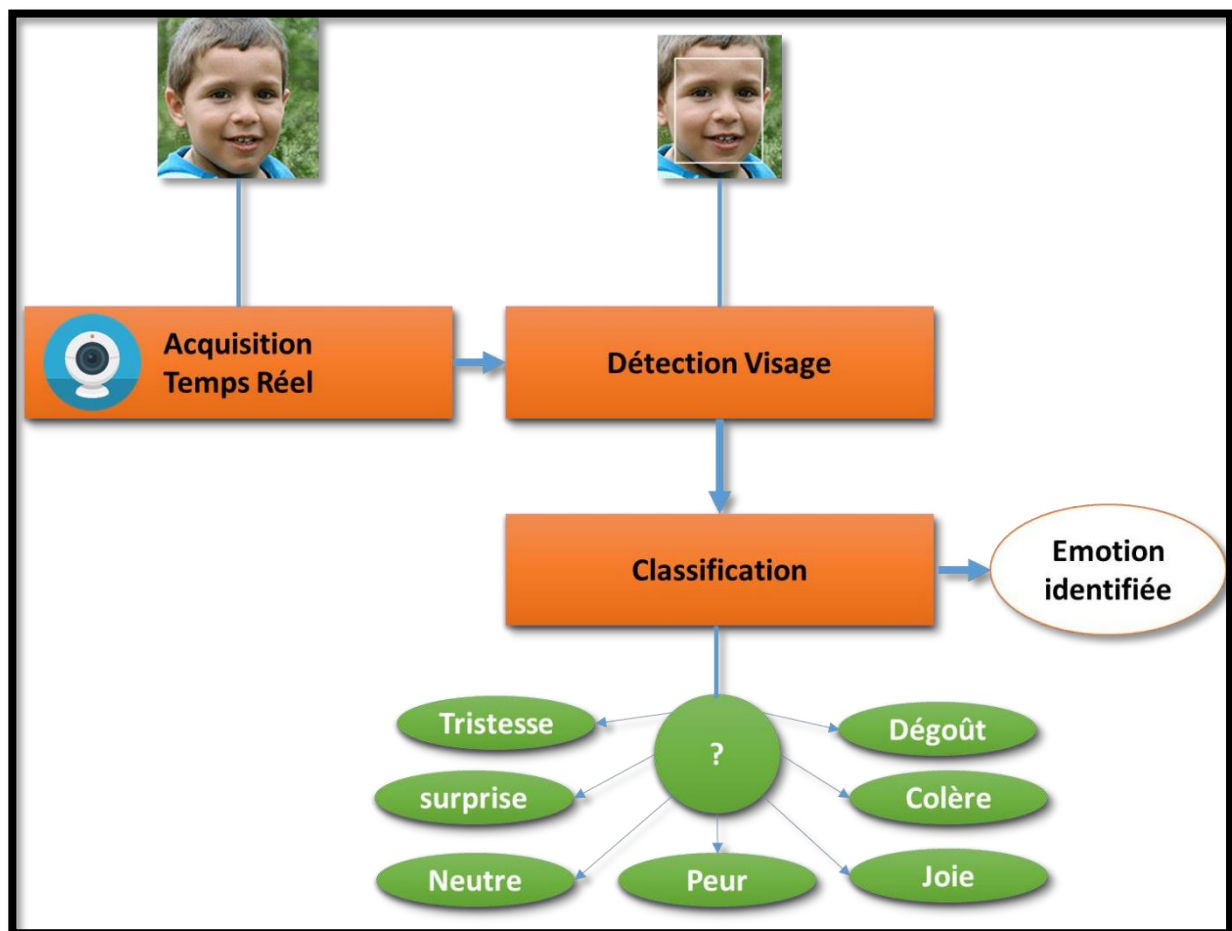


Figure 13: Architecture globale de reconnaissance des émotions réalisées

2.3. Phase d'apprentissage et de prédiction

2.3.1 Phase d'apprentissage

a- Préparation des données : Les images d'apprentissage sont collectées et annotées avec les expressions faciales correspondantes. Elles sont classées par classe (joie, tristesse, colère, etc.).

Dans notre cas nous utilisons la base de données MMFEDB téléchargeable à partir de Kaggle.com.

b- Modèle d'apprentissage : On va implémenter un classifieur des expressions faciales en utilisant un réseau de neurones convolutionnels (CNN) qui est conçu et configuré pour extraire les caractéristiques pertinentes des images d'expression faciale et effectuer la classification.

c-Entraînement du modèle :

L'entraînement du modèle consiste à ajuster les poids des neurones à l'aide des données d'apprentissage annotées. Dans notre cas, le modèle est entraîné à l'aide de TensorFlow et Keras sur la plateforme Kaggle.

Kaggle fournit un environnement prêt à l'emploi pour l'exécution de scripts TensorFlow et Keras, ce qui facilite l'entraînement et l'évaluation des modèles sur des ensembles de données disponibles sur la plateforme.

Pendant l'entraînement, le modèle est exposé aux images d'entraînement et à leurs étiquettes correspondantes. Les poids des neurones sont ajustés itérativement en utilisant l'algorithme d'optimisation Adam pour minimiser l'erreur de prédiction et maximiser la précision du modèle.

Le processus d'entraînement se déroule sur plusieurs époques, où chaque époque correspond à une passe complète sur l'ensemble des données d'entraînement. Au cours de chaque époque, les images sont divisées en lots (batches) de taille prédéfinie. Le modèle prédit les étiquettes correspondantes pour chaque image du lot.

Après chaque époque, le modèle est évalué sur l'ensemble de validation pour mesurer sa performance sur des données inconnues. Cela permet de surveiller la capacité du modèle à généraliser et à éviter le surapprentissage.

Une fois l'entraînement terminé, le modèle entraîné est sauvegardé dans un fichier, dans notre cas "emotion_model.h5", pour utilisations dans la phase de prédiction

2.3.2 Phase de prédiction

a- Acquisition d'images :

L'étape d'acquisition d'une image consiste à utiliser la bibliothèque OpenCV pour capturer une seule image en temps réel à partir de la source vidéo de la caméra. Cela permet de récupérer une seule image instantanée provenant de la caméra.

b- Détection de visage :

La détection des visages constitue un défi majeur qui consiste à identifier la présence d'un visage dans une image et à déterminer sa localisation et son échelle. L'efficacité des systèmes de reconnaissance des émotions dépend en grande partie de la méthode utilisée pour localiser les visages dans l'image.

Nous proposons deux méthodes distinctes pour la reconnaissance des émotions faciale :

- Haar Cascade

-MTCNN (Multi-task Cascaded Convolutional Networks)

La méthode de détection de visage basée sur Haar Cascade

La méthode de détection de visage que nous proposons comme première solution utilise les cascades de Haar qui repose sur l'algorithme de détection de visage Viola-Jones.

La méthode de Viola et Jones consiste à balayer une image à l'aide d'une fenêtre de détection de taille initiale 24px par 24px (dans l'algorithme original) et de déterminer si un visage y est présent.

La bibliothèque OpenCV présente une implémentation de cet algorithme sous le nom «détecteur en cascades de Haar ». Le point fort de cette méthode est la rapidité de détection ce qui la rend capable de s'exécuter en temps réel et de répondre aux exigences du traitement vidéo. Toutefois, elle présente quelques limites telles que la difficulté de détection simultanée de plusieurs vues de même objet, la durée nécessaire à la phase d'apprentissage des cascades est relativement assez grande et le nombre d'échantillons d'apprentissage est important. [7]

La méthode de détection de visage basée sur MTCNN

Cette méthode repose sur l'utilisation de l'algorithme MTCNN (Multi-task Cascaded Convolutional Networks), qui est largement utilisé pour la détection et la localisation des visages. Cette approche nous permet d'extraire les caractéristiques du visage, telles que les yeux, le nez et la bouche, afin d'identifier les expressions émotionnelles.

c- Prédiction de l'expression faciale : Les images prétraitées sont soumises au modèle de prédiction entraîné lors de la phase d'apprentissage. Le modèle classe l'expression faciale correspondante pour chaque image.

d- Affichage des résultats : Les résultats de la prédiction sont affichés sous forme de texte représentant les expressions faciales détectées. Dans notre cas les résultats de l'entraînement du modèle seront affichés dans une interface graphique en utilisant Tkinter. Cela nous permettra de visualiser les performances du modèle de manière interactive.

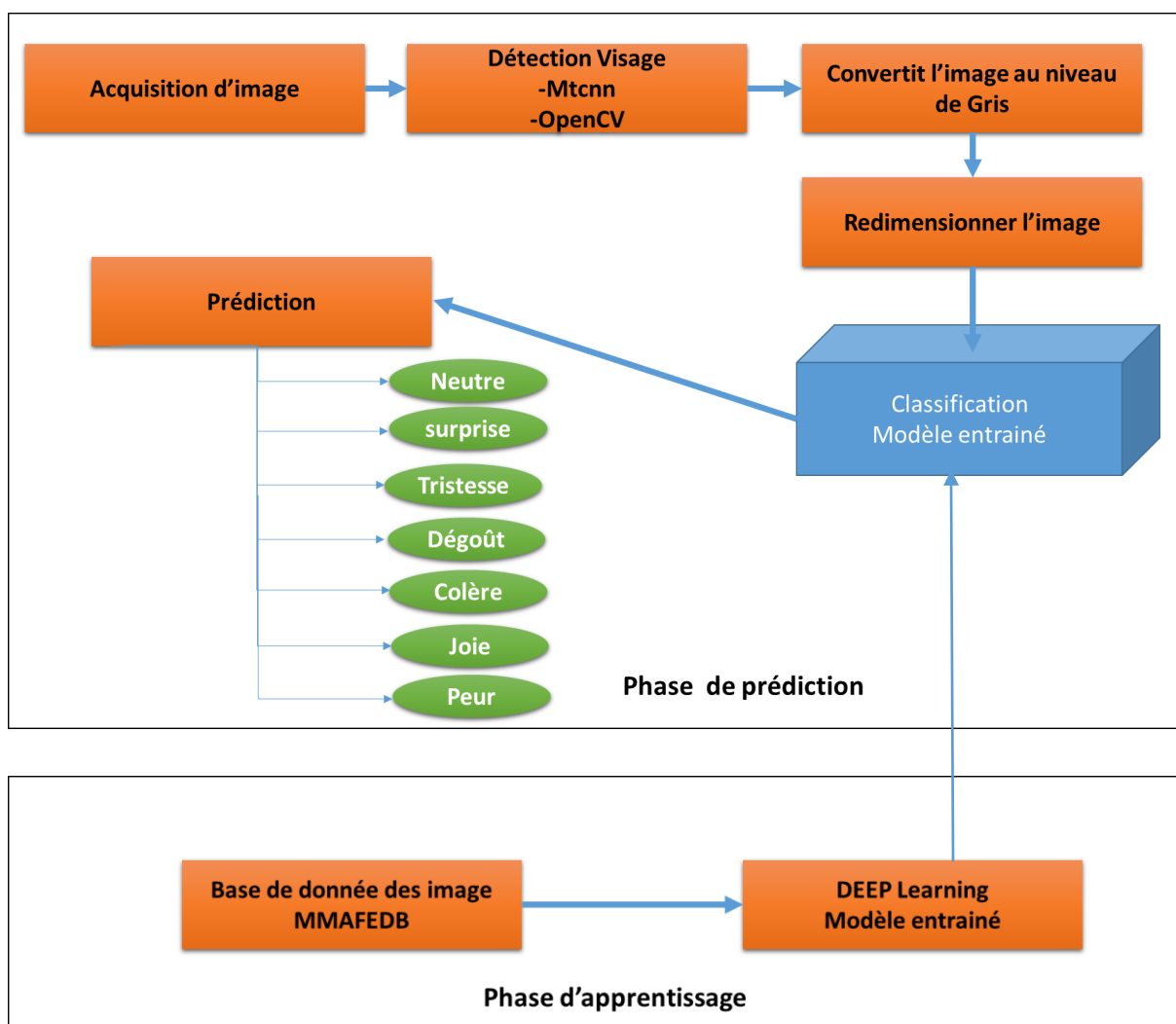


Figure 14: Schéma globale du système de Reconnaissance des émotions réalisé

input_shape = (48, 48, 1)

Premiere Couche

1. Conv2d(
 - Filters: 64
 - Kernel Size: (5, 5)
 - Activation: ReLU
 - Padding: Same)
2. MaxPooling2D(
 - Taille de la pool (2,2))

Deuxieme Couche

1. Conv2d(
 - Filters: 128
 - Kernel Size: (3, 3)
 - Activation: ReLU
 - Padding: Same)
3. MaxPooling2D(
 - Taille de la pool (2,2))

Troisieme couche

1. Conv2d(
 - Filters: 256
 - Kernel Size: (2, 2)
 - Activation: ReLU
 - Padding: Same)
2. MaxPooling2D(
 - Taille de la pool (2,2))

Flatten(x)

Quadrieme couche

- Dense(
 - Nombre de neurones: 256
 - Activation: ReLU)

Cinquieme couche

- Dense(
 - Nombre de neurones: 7
 - Activation: softmax)

Compiler le modèle

model.compile(

- Optimiseur : Adam
- Fonction de perte : categorical_crossentropy
- Métriques : accuracy

)

Figure 15: Schéma présentant l'architecture du modèle de classification des émotions

3. Conclusion

Ce chapitre a abordé la méthodologie de travail et les étapes nécessaires pour développer notre système de reconnaissance des expressions faciales. Nous avons également présenté un schéma général du processus de classification en utilisant des réseaux neuronaux convolutifs (CNN) et expliqué comment créer un modèle entraîné d'expression faciale.

Dans le prochain chapitre, nous nous concentrerons sur l'implémentation pratique du système et discuterons des résultats obtenus à partir de nos expérimentations.

Chapitre 03
***Implémentation et représentation de
l'interface***

Chapitre 03 : Implémentation et représentation de l'interface

1 Introduction :

Dans ce chapitre, Nous présenterons les outils logiciels et matériels utilisés dans l'implémentation de notre approche et pour améliorer l'expérience utilisateur, nous partagerons les résultats obtenus grâce à notre système de reconnaissance des émotions, mettant en évidence l'efficacité de l'interface graphique pour une utilisation conviviale et une interprétation visuelle des résultats.

2. Environnements de travail

Dans cette section, nous présentons l'environnement matériel et logiciel utilisé pour réaliser l'application.

2.1 Environnement matériel

La configuration de la machine utilisée est comme suit :

CPU	Intel® Core™ i3-4005U CPU @1.70GHz 1.70 GHz
Système d'exploitation	Windows 10 64 Bit
RAM	6.00 GB
Webcam	Width 640.0/ height : 480.0 / FPS : 30.0

2.2 Environnement logiciel

2.2.1 Python

Python est un langage de programmation interprété, orienté objet et de haut niveau. Il a été créé par Guido van Rossum et a été initialement publié en 1991. Python se distingue par sa syntaxe claire et simple, ce qui le rend facile à apprendre et à lire.



Figure 16: Logo de Python

2.2.2 Anaconda

Anaconda est une distribution open source de Python et de ses bibliothèques associées, conçue pour simplifier la gestion et l'utilisation des packages scientifiques et des outils de data science. Elle comprend un gestionnaire de packages, des environnements virtuels et une suite d'outils et de bibliothèques populaires préinstallés.



Figure 17: Logo d'anaconda

2.3 Les bibliothèques

2.3.1 OpenCV

Une bibliothèque qui permet d'effectuer des traitements particulièrement efficaces sur des images et des vidéos. En particulier, elle implémente des algorithmes de deep learning pré entraînés par exemple pour la reconnaissance faciale. [28]



Figure 18: Logo d'OpenCV

2.3.2 Tkinter

Tkinter est une bibliothèque standard de Python qui permet de créer des interfaces graphiques (GUI) pour des applications. Elle offre des outils et des widgets pour concevoir des fenêtres, des boutons, des champs de saisie, des listes déroulantes et d'autres éléments interactifs dans une application.

2.3.3 Keras

Keras est une API de réseaux de neurones de haut niveau, écrite en Python et interfaçable avec TensorFlow, CNTK et Theano. Elle a été développée avec pour objectif de permettre des expérimentations rapides. Être capable d'aller de l'idée au résultat avec le plus faible délai possible étant la clef d'une recherche efficace. [29]



Figure 19:: Logo Keras

2.3.4 Tenserflow

TensorFlow est une bibliothèque logicielle open-source utilisée pour créer et déployer des modèles d'apprentissage automatique. Il est principalement utilisé pour le développement de réseaux de neurones profonds et d'autres modèles d'apprentissage automatique. TensorFlow fournit des outils et des ressources pour faciliter la création, l'entraînement et le déploiement de ces modèles.

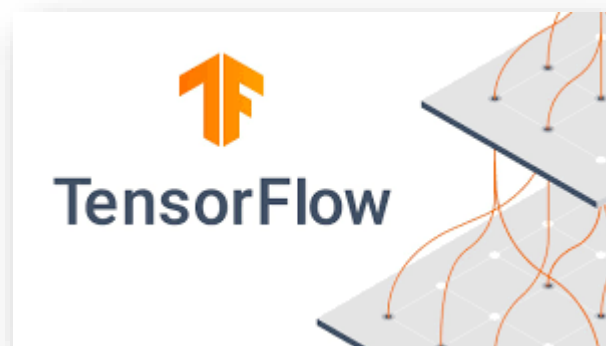


Figure 20: Logo Tenserflow

3. Classifieur

3.1 Préparation des données pour l'apprentissage (l'entraînement)

Nous avons utilisé pour l'entraînement la base de données « MMA FACIAL EXPRESSION » qui provient de la plateforme en Ligne Kaggle.com qui est destiné aux passionnés d'apprentissage automatique.

L'utilisation d'une base de données d'expressions faciales riche en données, telle que "MMA Facial expression" dans notre cas est bénéfique pour améliorer la précision et les performances de notre modèle de reconnaissance des émotions faciales, On peut télécharger gratuitement cette base à partir d'internet sur le site web Kaggle.

MMAFEDB contient 3 répertoires : Entraînement, validation et le test. Chaque répertoire contient sept sous-répertoires correspondant à sept catégories d'expressions faciales.

Les images de teste sont pré-recadrées de 48x48 pixels de visages chacun étiquetés avec l'une des 7 classes d'émotion suivantes :

0 : 1041 images → colère.

1 : 655 images → dégoût.

2 : 691 images → peur.

3 : 5459 images → heureuse.

4 : 5858 images → neutre.

5 : 2177 images → triste.

6 : 1475 images → surprise.



Figure 21: Echantillon d'image de la Base de données MMA FACIAL EXPRESSION (MMAFEDB) [30]

3.2 L'apprentissage du classifieur d'images de réseau neuronal convolutionnel avec Keras et Tensorflow

Une fois les données préparées, nous avons utilisé une architecture de réseau de neuronesconvolutifs (CNN) pour l'apprentissage

Notre modèle de reconnaissance des expressions faciales utilise le framework Keras avec TensorFlow comme backend. Le modèle est un réseau de neurones convolutif avec plusieurs couches de convolution, normalisation par lots et couches entièrement connectées.

Les images sont prétraitées à l'aide de générateurs d'images (ImageDataGenerator) pour la mise à l'échelle des pixels. Le modèle est compilé avec l'optimiseur Adam et la perte catégorielle croisée. Ensuite, il est entraîné pendant 10 époques sur les données d'entraînement et de validation.

Une fois l'entraînement terminé, le modèle est enregistré sous le nom de fichier "emotion_model.h5".

a. Importer les bibliothèques nécessaires :

- `os` : Pour effectuer des opérations sur le système de fichiers.
- `tensorflow.keras.models.Model` : Pour créer le modèle.
- `tensorflow.keras.layers` : Pour définir les couches du modèle.
- `tensorflow.keras.initializers.glorot_uniform` : Pour initialiser les poids du modèle.
- `keras.preprocessing.image.ImageDataGenerator` : Pour augmenter et prétraiter les images.

b. Définir les chemins d'accès aux répertoires de données :

- `train_data_dir` : Chemin d'accès au répertoire d'entraînement contenant les images d'entraînement.
- `validation_data_dir` : Chemin d'accès au répertoire de validation contenant les images de validation.

c. Définir la taille des images et la taille du lot (batch) :

- `image_size` : Une tuple représentant la taille des images (largeur, hauteur).
- `batch_size` : Le nombre d'images incluses dans chaque lot lors de l'entraînement.

d. Déterminer le nombre de classes d'émotions :

- `num_classes` : Le nombre de sous-répertoires dans le répertoire d'entraînement, qui représente les différentes classes d'émotions.

e. Créer le modèle :

- Définir la forme d'entrée du modèle en utilisant `Input(shape=input_shape)`.

- Appliquer des couches de convolution, de normalisation en lots et de mise en commun pour extraire les caractéristiques des images.
- Aplatir les caractéristiques extraites en utilisant Flatten().
- Appliquer des couches entièrement connectées (Dense) pour apprendre à partir des caractéristiques extraites.
- Définir la couche de sortie avec le nombre de classes d'émotions et la fonction d'activation softmax pour obtenir les probabilités des différentes classes.
- Créer le modèle en utilisant Model(inputs=inputs, outputs=outputs).

f. Compiler le modèle :

- Spécifier l'optimiseur (adam), la fonction de perte (categorical_crossentropy) et les métriques à utiliser pour évaluer le modèle (accuracy).

g. Créer des générateurs de données pour l'entraînement et la validation :

- Utiliser ImageDataGenerator pour augmenter et prétraiter les images.
- Définir les répertoires source, la taille des images, la taille du lot, le mode de couleur (grayscale) et le mode de classe (categorical).
- Les générateurs de données permettent de charger les images à la volée pendant l'entraînement, ce qui est utile pour les ensembles de données volumineux.

h. Entraîner le modèle :

- Utiliser la méthode fit pour entraîner le modèle.
- Passer le générateur d'entraînement, le nombre d'étapes par époque, le nombre d'époques, le générateur de validation et le nombre d'étapes de validation par époque.

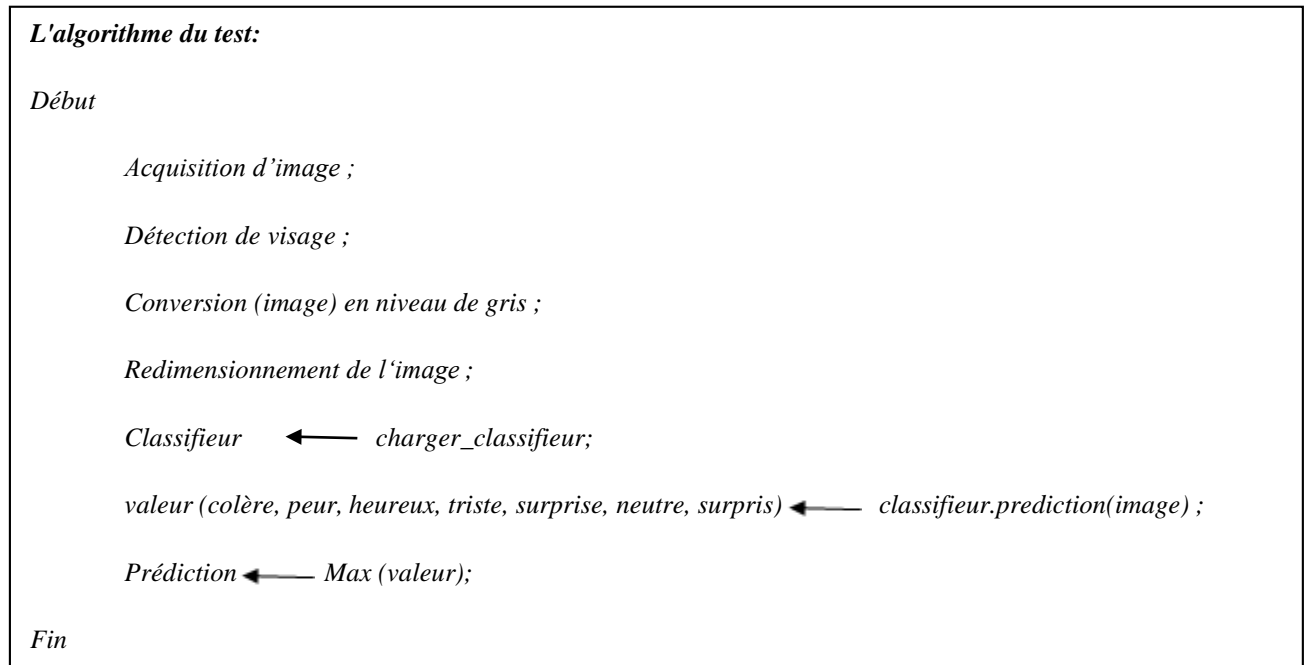
i. Sauvegarder le modèle :

- Utiliser la méthode save pour enregistrer le modèle entraîné dans un fichier "emotion_model.h5".

4. Mise en œuvre du classifieur

Pour la mise en œuvre de notre classifieur, nous utilisons une webcam et une interface graphique que nous avons développée avec Tkinter. Cette interface nous permet de capturer nos visages.

En utilisant la webcam, l'interface graphique affiche en temps réel les images capturées. Nous pouvons ainsi exprimer différentes expressions faciales devant la caméra et observer les prédictions correspondantes fournies par notre classifieur.



5. Les tests : Présentation de l'application avec une interface homme-machine

Pour tester notre approche et visualiser les résultats, nous avons développé une interface conviviale utilisant la bibliothèque Tkinter de python. Cette interface offre différentes fonctionnalités pour la détection en temps réel des visages à partir d'une webcam.

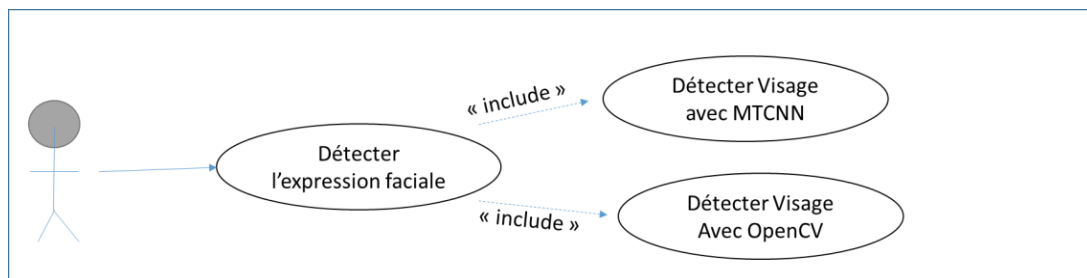


Figure 22: Diagramme de cas d'utilisation avec le langage UML de notre système de reconnaissance des expressions faciales

Après le démarrage de l'application, la webcam est automatiquement activée à l'intérieur de l'interface. L'interface présente 5 boutons comme on peut le voir dans la figure :

Bouton 1 « Enable MTCNN Face Detection » : Activer la détection avec la méthode MTCNN

Bouton 2 « Enable OpenCV Face Detection » : Activer la détection avec la méthode OpenCV

Bouton 3 « Enable Emotion Detection » : Activer la détection des émotions

Bouton 4 « Take Snapshot » : Prendre une

photo

Bouton 5 « Quit » : Quitter le système



Figure 23: Interface au démarrage du système

L'utilisateur peut choisir la méthode de détection des visages en temps réel en sélectionnant l'une des deux boutons correspondants. Lorsqu'aucune des deux méthodes n'est activée, le bouton de détection de l'expression faciale reste inactif.

En cliquant sur le bouton « Enable MTCNN Face Detection », un ou plusieurs rectangles apparaissent pour encadrer les visages détectés dans la séquence vidéo,

comme il est illustré dans la figure 3.9. Pendant que la méthode MTCNN est active, un message s'affiche sur le bouton correspondant pour indiquer l'exécution de cette méthode « Disable MTCNN Face Detection », et le bouton de détection de l'expression faciale devient actif. Le système affiche le nombre de visage détecté

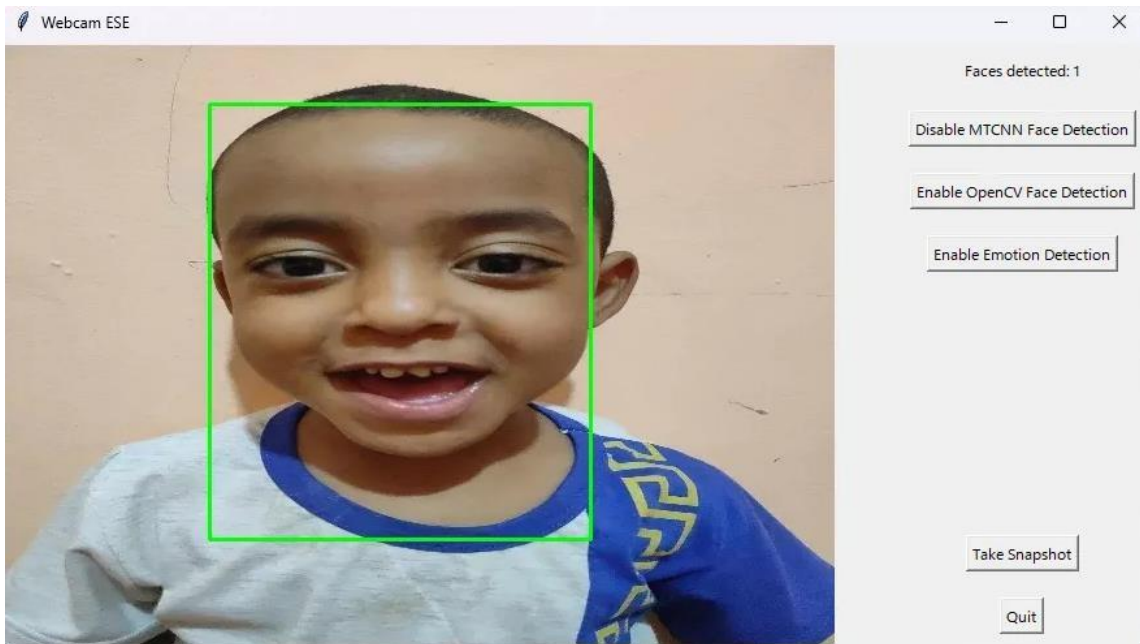


Figure 24: Détection de visage avec la méthode MTCNN

De même, en cliquant sur le bouton « Enable OpenCV Face Detection », le bouton de la méthode utilisant MTCNN devient inactif et affiche « Enable MTCNN Face Detection », de même, un message s'affiche sur le bouton associé à la méthode OpenCV pour indiquer son activation « disable OpenCV Face Detection ». Le bouton de détection de l'expression faciale devient également cliquable, comme illustré dans la figure 3.10.

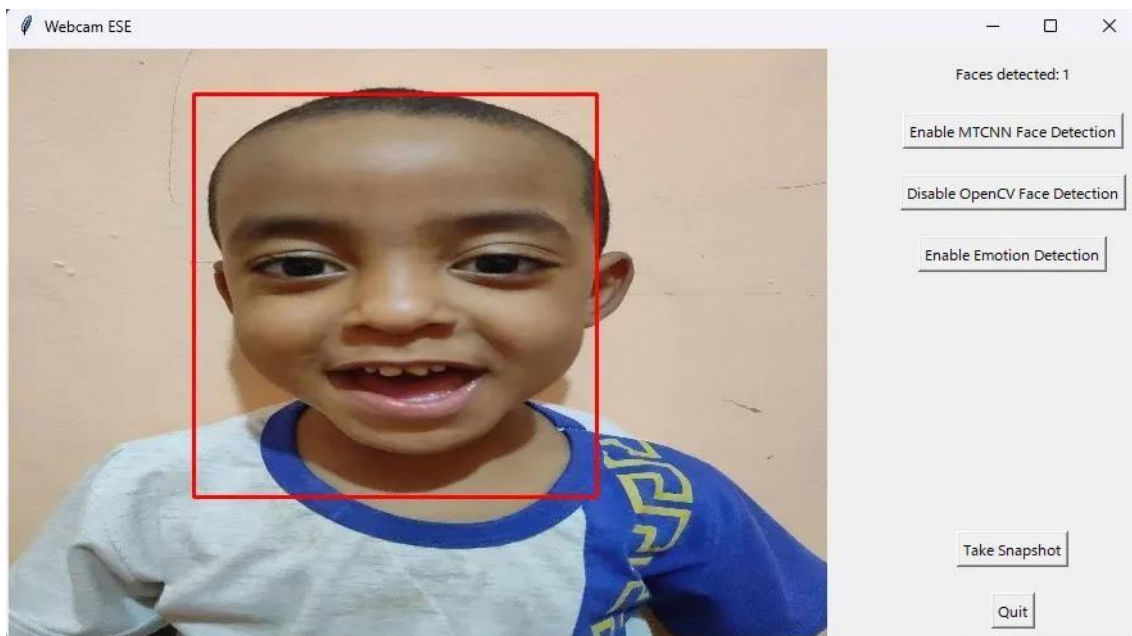


Figure 25: Détection de visage avec la méthode OPENCV

Comme mentionné précédemment, le bouton activant la fonction de détection de l'expression faciale est conditionné à l'activation d'une des méthodes de détection des visages. Cela signifie que lorsque l'utilisateur sélectionne l'une des méthodes de détection des visages (soit avec OpenCV, soit avec MTCNN), la fonction de détection de l'expression faciale devient disponible.

En cliquant sur le bouton détection des expressions faciales cela permettra à l'utilisateur de détecter et d'analyser les expressions faciales des visages détectés en temps réel.

Dans la figure 3.11 , nous pouvons observer une expression de joie clairement affichée. Cette expression est caractérisée par des traits faciaux tels que des yeux légèrement plissés, des lèvres étirées en un large sourire et peut-être des joues légèrement relevées. La méthode de détection de visage utilisée avec est l'OpenCV.

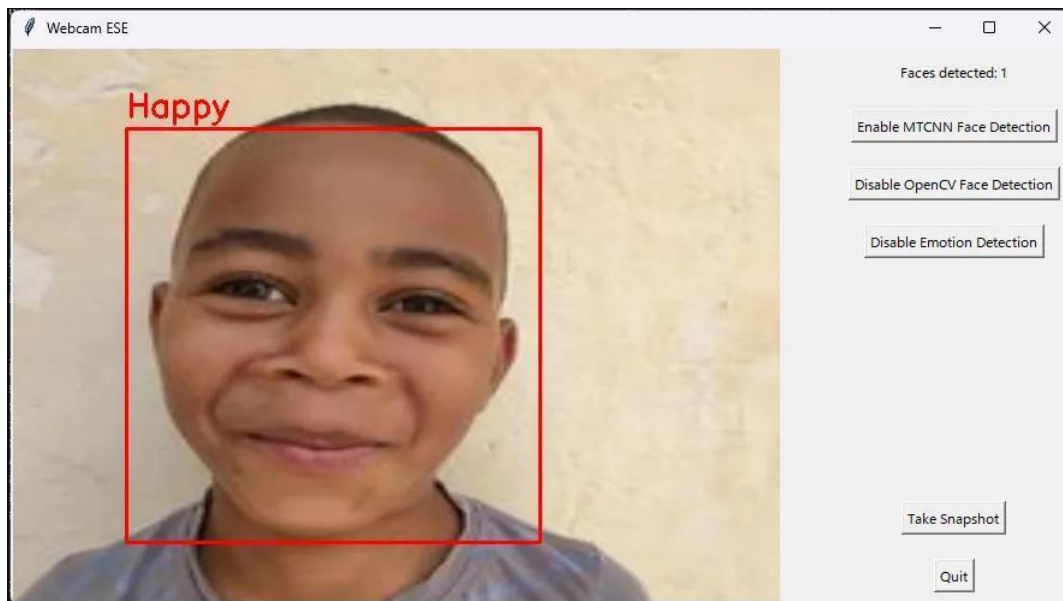


Figure 26: détection d'émotion « Happy » Joie

Quant à la figure 3.12 , elle représente une expression triste caractérisée par un positionnement neutre des traits du visage. La méthode de détection de visage utilisée avec est MTCNN.

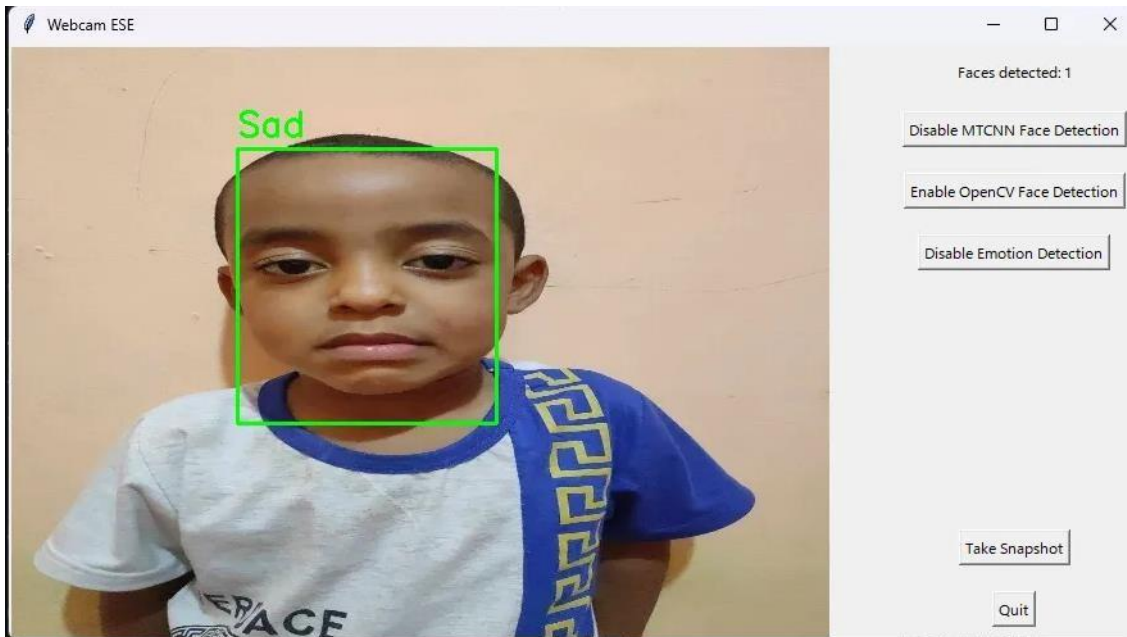


Figure 27: Détection d'émotion « Sad» Triste

Cette conception interactive assure que la détection de l'expression faciale est effectuée uniquement lorsque les visages sont détectés, garantissant ainsi une expérience utilisateur plus cohérente et informative.

5.1 Discussion des résultats

Pour la détection du visage, La méthode Haar Cascade est plus rapide mais peut-être moins précise dans des scénarios complexes, tandis que MTCNN offre une précision supérieure mais peut être plus lente en raison de l'utilisation de réseaux de neurones convolutifs.

Après plusieurs tests, notre classifieur a montré de bons résultats pour la classification des expressions faciales telles que la joie, tristesse et la surprise.

Cependant, le classifieur présente des performances plus faibles pour les expressions de colère, de dégoût de peur et de neutralité.

6. Conclusion

Ce chapitre a couvert l'environnement de développement et les outils utilisés pour la création de notre système de détection des visages et expressions faciales, ainsi que la conception de son interface graphique. Nous avons examiné en détail les étapes impliquées dans le développement de ce système, en mettant l'accent sur la mise en place d'un environnement propice à la détection

précise des visages et à l'analyse des émotions faciales.

Grâce à ces efforts, nous avons réussi à réaliser un système fonctionnel qui combine une détection précise des visages avec la reconnaissance des émotions faciales, le tout présenté dans une interface utilisateur conviviale qui permet de superviser facilement les différents traitements du système.

Conclusion Générale

Conclusion Générale

Dans notre mémoire, nous avons développé un système de reconnaissance des expressions faciales avec une interface graphique conviviale. La particularité de notre approche est qu'elle permet la détection en temps réel des expressions faciales sur des vidéos.

Dans le premier chapitre, nous avons introduit les concepts et les définitions essentiels de la reconnaissance des expressions faciales. Nous avons également passé en revue les méthodes de détection existantes, mettant en évidence les avancées réalisées dans ce domaine.

Le deuxième chapitre était consacré à la conception de notre système. Nous avons présenté l'architecture du système, en insistant sur les méthodes utilisées pour la détection des visages et des expressions faciales. Nous avons détaillé les différentes étapes du processus de détection et d'analyse des expressions faciales, en soulignant leur capacité à fonctionner en temps réel sur des vidéos.

Dans le troisième chapitre, nous avons abordé l'implémentation pratique du système. Nous avons décrit les outils et l'environnement utilisés, ainsi que l'interface graphique que nous avons développée. L'interface permet à l'utilisateur de visualiser en temps réel les résultats de la détection des expressions faciales sur des vidéos.

La principale différence entre MTCNN et OpenCV réside dans leur approche respective de détection des visages. MTCNN repose sur des réseaux de neurones convolutifs et utilise l'apprentissage en profondeur, ce qui lui permet d'obtenir des résultats de détection précis et robustes, même dans des scénarios complexes. En revanche, OpenCV se base sur des techniques classiques de vision par ordinateur, ce qui peut offrir une vitesse de traitement plus rapide, mais peut-être moins précis dans certaines situations.

En résumé, si vous avez besoin d'une détection de visage précise et fiable, MTCNN est une meilleure option. Son utilisation de l'apprentissage en profondeur lui permet de capturer les caractéristiques subtiles du visage, ce qui est crucial pour la reconnaissance des expressions faciales. Toutefois, si vous recherchez une solution plus rapide pour des applications plus simples, où la vitesse de traitement est un critère important, OpenCV peut être préférable.

Il est important de choisir la méthode de détection de visage en fonction des besoins spécifiques de votre application, en tenant compte à la fois de la précision et de la vitesse de traitement requises.

En conclusion, notre mémoire a démontré l'efficacité de notre approche pour la reconnaissance des expressions faciales en temps réel sur des vidéos. Les résultats obtenus ouvrent de nouvelles perspectives pour l'application de cette technologie dans divers domaines tels que l'analyse de l'émotion, la réalité virtuelle, et les interfaces homme-machine interactives. Des travaux futurs pourraient se concentrer sur l'amélioration des performances du système et l'exploration de nouvelles fonctionnalités pour une détection encore plus précise et robuste des expressions faciales dans des vidéos variées.

Bibliographie

Bibliographie

- [1] n. s. 2.0, *AmUnivers Fotopédia*.
- [2] K. GHANEM, «Reconnaissance des Expressions Faciales à Base d'Informations Vidéo, Estimation de l'Intensité des Expressions faciale.,» Constantine, 2010.
- [3] I. S. Rebahi Ghediri, *Thème La Reconnaissance Des émotions de base Par Les réseaux de neurones : Application de Deep Learning*.
- [4] R. Plutchik, *A general psychoevolutionary theory of emotion. In Theories of emotion*, Academic press, 1980.
- [5] C. & C. G. W. Padgett, *Representing face images for emotion classification.*, 1997.
- [6] A. T. C. J. & C. T. F. Lanitis, *Automatic interpretation and coding of*, 1997.
- [7] P. V. a. M. Jones, «"Rapid Object Detection using a Boosted Cascade of Simple Features",» chez *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.
- [8] Z. Z. Z. L. a. Y. Q. Kaipeng Zhang, «"Joint Face Detection and Alignment using Multi-task Cascaded Convolutional,» chez *IEEE Signal Processing Letters*, 2016.
- [9] [En ligne]. Available: <https://rassouldev.medium.com/1-introduction-%C3%A0-la-d%C3%A9tection-des-visages-b1e88be90e52>.
- [10] [En ligne]. Available: <https://kongakura.fr/article/haar-cascade>.
- [11] O. H. C. -. PyImageSearch. [En ligne]. Available: <https://www.pyimagesearch.com/2021/04/12/opencv-haar-cascades>. [Accès le 2 06 2023].
- [12] «ResearchGate,» [En ligne]. Available: https://www.researchgate.net/figure/Haar-features-used-for-Viola-Jones-face-detectionmethod_fig1_268348020. [Accès le 02 06 2023].
- [13] T. Teixeira, «Reconnaissance multi-dimensionnelle de l'émotion par apprentissage profond de caractéristiques spatio-temporelles sur séquences vidéo,» Montreal, 2020.

- [14] K. Zhang, Z. Zhang, Z. Li et Y. and Qiao, «Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters 23(10): 1499–1503,» 2016.
- [15] «Minor Privacy Protection by Real-time Children Identification and Face Scrambling at the Edge - Scientific Figure on ResearchGate. Available from».
- [16] S. Z. OUBAHA AMINA, «Systèmes intelligents & Réseaux Faculté des sciences et Techniques,» Fès.
- [17] M. Pantic, « Face for interface. The Encyclopedia of Multimedia Technology and Networking,» 2002.
- [18] K. LEKDIOUI, «Reconnaissance d'états émotionnels par analyse visuelle du visage et apprentissage machine,» Montbéliard, 2018.
- [19] A. I. Nadia, «Une approche IA pour la reconnaissance des expressions faciales,» Bouira, Algérie, 2019-2020.
- [20] « A Brief Review of Facial Emotion Recognition Based on Visual Information Byoung Chul Ko University,» Korea, 2018.
- [21] B. A. KARFOUF Mohammed Ali, «KARFOUF Mohammed Ali et BEDDIAF Ali, Reconnaissance des expressions faciales,» Mémoire de Master en Électronique Système Embarqué, Université Kasdi Merbah-Ouargla, 2022.
- [22] R. Agrawal, «Person Detection in Various Posture using HOG Feature And SVM Classifier,» 2018.
- [23] L. Heidmann, «AI vs. Machine Learning vs. Deep Learning.,» 2019.
- [24] M. D. Youcef, « Deep Learning pour la classification des images,» Université Abou Bakr Belkaid Tlemcen, 2017.
- [25] «Artificial Intelligence and Radiotherapy,» 2021.
- [26] M. M. Daha Boubaker, «Un miroir intelligent interactif basé sur Raspberry Pi,» UNIVERSITÉ ECHAHID HAMMA LAKHDAR EL OUED, 2018.
- [27] D. Kriesel, «A Brief Introduction to Neural Networks,» Bonn, 2017.

[28] [En ligne]. Available: <https://www.editions-eni.fr/>.

[29] [En ligne]. Available: <https://www.actuia.com/keras/>.

[30] [En ligne]. Available: <https://www.kaggle.com/datasets/mahmoudima/mma-facial-expression/>.

[31] M. & P. A. Turk, *Eigenfaces for recognition*. *Journal of cognitive*, 1991.

[32] O. A. -. S. ZINEB.

ANNEX :

Le code de l'application Python :

```
import tkinter as tk
from datetime import datetime
import os
import cv2
import PIL.Image, PIL.ImageTk
from facenet_pytorch import MTCNN
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import img_to_array
window = tk.Tk()
window.title("Webcam ESE")
canvas_width = int(window.winfo_screenwidth() * 0.5)
canvas_height = int(window.winfo_screenheight() * 0.6)
canvas = tk.Canvas(window, width=canvas_width, height=canvas_height)
canvas.pack(side=tk.LEFT)
face_cascade = cv2.CascadeClassifier('C:/ProgramData/anaconda3/Lib/site-
packages/cv2/data/haarcascade_frontalface_default.xml')
model = load_model('emotion_model2.h5')
emotion_labels = ['Angry', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral']
video_capture = cv2.VideoCapture(0)
opencv_face_detection_enabled = False
emotion_detection_enabled = False
face_detection_enabled = False # Set initial state of face detection
mtcnn = MTCNN() # Load the MTCNN face detector
mtcnn.p_threshold = 0.9
mtcnn.r_threshold = 0.9
mtcnn.o_threshold = 0.9

def take_snapshot():
    # Capture a frame from the video feed
    ret, frame = video_capture.read()
    if ret:
        # Save the current frame to a file in the specified directory
        timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
        filename = f"{timestamp}.jpg"
        filepath = os.path.join(snapshot_directory, filename)
        cv2.imwrite(filepath, frame)
        print(f"Snapshot saved to {filepath}")

def toggle_emotion_detection():
    global emotion_detection_enabled, face_detection_enabled, opencv_face_detection_enabled

    emotion_detection_enabled = not emotion_detection_enabled

    if emotion_detection_enabled:
        emo_toggle_button.configure(text="Disable Emotion Detection")
    else:
        emo_toggle_button.configure(text="Enable Emotion Detection")
```

```

def toggle_face_detection():
    global face_detection_enabled
    face_detection_enabled = not face_detection_enabled
    if face_detection_enabled:
        fd_toggle_button.configure(text="Disable MTCNN Face Detection")
        global opencv_face_detection_enabled
        opencv_face_detection_enabled = False
        opencv_fd_toggle_button.configure(text="Enable OpenCV Face Detection")
    else:
        fd_toggle_button.configure(text="Enable MTCNN Face Detection")
    if not face_detection_enabled and not opencv_face_detection_enabled:
        emo_toggle_button['state'] = 'disabled' # Disable the toggle button
    else:
        emo_toggle_button['state'] = 'normal' # Enable the toggle button

def toggle_opencv_face_detection():
    global opencv_face_detection_enabled
    opencv_face_detection_enabled = not opencv_face_detection_enabled
    if opencv_face_detection_enabled:
        opencv_fd_toggle_button.configure(text="Disable OpenCV Face Detection")
        global face_detection_enabled
        face_detection_enabled = False
        fd_toggle_button.configure(text="Enable MTCNN Face Detection")
    else:
        opencv_fd_toggle_button.configure(text="Enable OpenCV Face Detection")
    if not face_detection_enabled and not opencv_face_detection_enabled:
        emo_toggle_button['state'] = 'disabled' # Disable the toggle button
    else:
        emo_toggle_button['state'] = 'normal' # Enable the toggle button

def quit_app():
    video_capture.release()
    window.quit()
    window.destroy()
    exit()

face_counter = tk.Label(window, text="Faces detected: 0")
face_counter.pack(side=tk.TOP, padx=10, pady=10)

def detect_emotion_mtcnn(frame):
    boxes, _ = mtcnn.detect(frame)
    num_faces = 0

    if boxes is not None:
        num_faces = len(boxes)
        for box in boxes:
            x1, y1, x2, y2 = box.astype('int')

            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

    if emotion_detection_enabled:
        face_roi = frame[y1:y2, x1:x2]

```

```

    face_roi = cv2.cvtColor(face_roi, cv2.COLOR_BGR2GRAY)
    face_roi = cv2.resize(face_roi, (48, 48))
    face_roi = face_roi.astype('float') / 255.0
    face_roi = img_to_array(face_roi)
    face_roi = np.expand_dims(face_roi, axis=0)

    # Predict the emotion
    emotion_prediction = model.predict(face_roi)
    emotion_index = np.argmax(emotion_prediction)
    emotion_label = emotion_labels[emotion_index]

    # Draw emotion label on the face rectangle
    cv2.putText(frame, emotion_label, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)

# Update the face counter label
face_counter.configure(text=f"Faces detected: {num_faces}")

def detect_emotion_opencv(frame):
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # Convert frame to grayscale
    faces = face_cascade.detectMultiScale(gray)
    num_faces = len(faces)

    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)

        if emotion_detection_enabled:
            face_roi = gray[y:y + h, x:x + w]
            face_roi = cv2.resize(face_roi, (48, 48))
            face_roi = face_roi.astype('float') / 255.0
            face_roi = img_to_array(face_roi)
            face_roi = np.expand_dims(face_roi, axis=0)

            # Predict the emotion
            emotion_prediction = model.predict(face_roi)
            emotion_index = np.argmax(emotion_prediction)
            emotion_label = emotion_labels[emotion_index]

            # Draw emotion label on the face rectangle
            cv2.putText(frame, emotion_label, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 0, 255), 2)

        # Update the face counter label
        face_counter.configure(text=f"Faces detected: {num_faces}")

def process_frame():
    ret, frame = video_capture.read()

    if ret:
        if face_detection_enabled:
            detect_emotion_mtcnn(frame)
        elif opencv_face_detection_enabled:
            detect_emotion_opencv(frame)

    # Convert the frame to PIL format
    # Convert the frame from BGR to RGB for display

```

```

frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
photo = PIL.ImageTk.PhotoImage(image=PIL.Image.fromarray(frame))
canvas.create_image(0, 0, image=photo, anchor=tk.NW)
canvas.image = photo

# Schedule the next frame update
window.after(10, process_frame)

snapshot_directory = "snapshots"
if not os.path.exists(snapshot_directory):
    os.makedirs(snapshot_directory)

quit_button = tk.Button(window, text="Quit", command=quit_app)
quit_button.pack(side=tk.BOTTOM, padx=10, pady=10)

snapshot_button = tk.Button(window, text="Take Snapshot", command=take_snapshot)
snapshot_button.pack(side=tk.BOTTOM, padx=10, pady=10)

fd_toggle_button = tk.Button(window, text="Enable MTCNN Face Detection", command=toggle_face_detection)
fd_toggle_button.pack(side=tk.TOP, padx=10, pady=10)

opencv_fd_toggle_button = tk.Button(window, text="Enable OpenCV Face Detection",
command=toggle_opencv_face_detection)
opencv_fd_toggle_button.pack(side=tk.TOP, padx=10, pady=10)

emo_toggle_button = tk.Button(window, text="Enable Emotion Detection", command=toggle_emotion_detection,
state='disabled')
emo_toggle_button.pack(side=tk.TOP, padx=10, pady=10)

# Start processing the video feed
process_frame()

# Start the Tkinter event loop
window.mainloop()

```