

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj

Faculté des Sciences et de la technologie

Département d'Electronique

Mémoire

Présenté pour obtenir

LE DIPLOME DE MASTER

FILIERE : **Electronique**

Spécialité : Electronique industrielle

Par

- **ZAZ Sofia Hanene**
- **BENKAHLA Lilia**

Intitulé

*Etude et réalisation d'un système de reconnaissance faciale basé
sur l'apprentissage profond*

Evalué le :

Par le Jury composé de :

<i>Nom & Prénom</i>	<i>Grade</i>	<i>Qualité</i>	<i>Etablissement</i>
AZOUG Seifeddine	MCB	Président	Univ-BBA
TALBI Mohamed Lamine	MCA	Encadreur	Univ-BBA
Sid Ahmed Soumia	MCB	Examineur	Univ-BBA

Année Universitaire:2022 / 2023

Remerciement

Nous remercions Dieu tout puissant qui nous a aidés à mener à terme ce travail, en nous protégeant et en nous donnant la santé, la force et le courage qui nous ont permis de l'achever au temps qui nous était déterminé.

Nous exprimons notre profonde gratitude au « Dr. TALBI Mohamed Lamine », notre encadreur, qui nous a suggéré ce sujet et qui nous a guidés et conseillés tout au long de sa réalisation. Monsieur TALBI a fait preuve d'une grande disponibilité et d'un grand soutien pour nous aider à progresser et à enrichir notre travail.

Nous remercions nos professeurs de l'institut qui nous ont transmis leurs connaissances et leurs compétences durant nos études universitaires.

Nous remercions Messieurs (et mesdames) les membres du jury qui ont bien voulu être présents avec nous pour l'évaluation de notre mémoire.

Enfin, nous remercions chaleureusement nos parents qui ont été présents à nos côtés durant toute notre scolarité, en nous offrant les moyens nécessaires à notre réussite et en nous témoignant leur affection et leur encouragement. C'est grâce à eux que nous avons pu réussir ce parcours.

Dédicace

Ce travail est dédié, avec le plus grand plaisir du monde, aux familles « ZAZ et BENKAHLA ».

À nos parents, les êtres les plus chers au monde, qui ont toujours été notre source de vie et d'espoir. Leur amour inconditionnel et leur soutien indéfectible nous ont guidés tout au long de ce parcours.

À notre grand-mère ZAHIA, qui incarne la sagesse, la tendresse et la générosité. Ses précieux enseignements et sa présence bienveillante ont façonné notre personnalité.

À l'oncle ABDELGHANI, dont la bienveillance, l'intégrité et la persévérance ont toujours été une source d'admiration pour nous. Par son encouragement constant, il nous a donné une force supplémentaire pour la réalisation de notre projet.

À nos frères et sœurs, qui ont partagé avec nous des moments inoubliables, des rires et des larmes. Leur soutien indéfectible et leur complicité nous ont boostés pour arriver à terme de ce travail.

À tous nos proches et amis, qui nous ont toujours fait sentir leur présence chaleureuse à nos côtés, tout au long de nos études. Cette présence et cette amitié précieuse ont été traduites par un appui indispensable et bénéfique pour aller de l'avant.

Enfin, en nous adressant à tous ceux qui ont contribué à ce travail de près ou de loin, nous disons tout simplement que cette dédicace est un témoignage de notre gratitude éternelle envers chacun d'entre vous. Vous avez joué un rôle essentiel dans notre vie et dans la réalisation de ce travail. Que nos liens perdurent et que notre amour continue de grandir au fil des années.

TABLE DES MATIERES

Introduction générale	1
-----------------------------	---

CHAPITRE I : LA RECONNAISSANCE FACIALE ET L'INTELLIGENCE ARTIFICIELLE

I.1 La reconnaissance faciale	3
I.1.1 Introduction	3
I.1.2 Définition	3
I.1.3 Historique	3
I.1.4 Mode de fonctionnement d'un système de reconnaissance de visage	4
I.1.5 Techniques de reconnaissance faciale	5
I.1.6 Les domaines d'application de la reconnaissance faciale	6
I.1.7 Les avantages et les inconvénients de la reconnaissance faciale	7
I.1.8 Les difficultés de la reconnaissance faciale	8
I.2 L'intelligence artificielle	10
I.2.1 introduction	10
I.2.2 Définition	10
I.2.3 L'histoire et l'évolution de l'intelligence artificielle	10
I.2.4 Les domaines d'application de l'intelligence artificielle	11
I.2.5 La reconnaissance faciale et l'intelligence artificielle	12
I.3 Conclusion.....	12

CHAPITRE II : APPRENTISSAGE BASE SUR LES RESEAUX DE NEURONES CONVOLUTIFS ET RASPBERRY PI

II.1 Apprentissage basé sur les réseaux de neurones	13
II.1.1 Introduction.....	13
II.1.2 Définition des mots clés.....	13
II.1.3 Différents types d'apprentissage	14
II.1.4 La différence entre ML et Deep Learning	15
II.1.5 Les avantages de Deep Learning	16

.II.1.6 Domaines d'application du Deep Learning	16
II.1.7 Réseaux de neurones.....	17
II.2 Le Raspberry Pi.....	23
II.2.1 Définition	24
II.2.2 Les avantages du Raspberry PI	25
II.3 Conclusion	25

CHAPITRE III : SIMULATION ET IMPLEMENTATION DU SYSTEME DE RECONNAISSANCE FACIALE

III.1 Introduction	26
III.2 Description de la méthodologie du travail.....	26
III.3 Choix des paramètres du modèle FaceNet	27
III.4 Apprentissage du modèle.....	27
III.5 Description de la base de données utilisée	28
III.6 Description du prétraitement de la base de données.....	28
III.7 Résultats et discussion	29
III.7.1 Simulation de l'algorithme sous Colab	29
III.7.2 Implémentation de l'algorithme de reconnaissance faciale.....	35
III.8 Conclusion	44
Conclusion générale	45
Bibliographie	47

LISTE DES FIGURES

<i>Figure 1:</i> Exemple sur la reconnaissance faciale	6
<i>Figure 2:</i> Variation de pose	9
<i>Figure 3:</i> Variation d'éclairage	9
<i>Figure 4:</i> Variation d'expression	9
<i>Figure 5:</i> Domaines d'application de l'IA	11
<i>Figure 6:</i> La relation entre IA, ML et Deep learning	14
<i>Figure 7:</i> Machine Learning et Deep Learning	15
<i>Figure 8:</i> Mise en correspondance neurone biologique / neurone artificiel	17
<i>Figure 9:</i> Structure simple d'un réseau de neurones	18
<i>Figure 10:</i> Structure d'un CNN	19
<i>Figure 11:</i> Exemples max pooling et average pooling	20
<i>Figure 12:</i> Evolution des architectures des CNNs au fil du temps	21
<i>Figure 13:</i> Raspberry pi modèle B	24
<i>Figure 14:</i> Les différentes étapes de l'implémentation	27
<i>Figure 15:</i> Matrice de confusion de FaceNet	35
<i>Figure 16:</i> Logo Raspbian	35
<i>Figure 17:</i> Fenêtre du Raspberry Pi imager	36
<i>Figure 18:</i> Kit complet du Raspberry Pi4	37
<i>Figure 19:</i> Exécution du système pour la première fois	38
<i>Figure 20:</i> Interface terminale	38
<i>Figure 21:</i> Résultats d'apprentissage sur Raspberry Pi	40
<i>Figure 22:</i> Test 1 Identification réussite	42
<i>Figure 23:</i> Test 2 Identification réussite	42
<i>Figure 24:</i> Test 3 Identification non réussite	43

LISTE DES TABLEAUX

<i>Tableau 1:</i> Comparaison entre les architectures CNN	29
<i>Tableau 2:</i> Les performances de FaceNet.....	30
<i>Tableau 3:</i> Paramètres du modèle FaceNet.....	30
<i>Tableau 4:</i> Les performances du système en fonction de nombres de couches.....	32
<i>Tableau 5:</i> Les performances du système en fonction de nombres de classes	33
<i>Tableau 6:</i> les performances du système en fonction de nombres d'epochs et batch_size	33
<i>Tableau 7:</i> Meilleurs paramètres pour le système à implémenter sur la carte Raspberry.....	34
<i>Tableau 8:</i> Les performances du système à implémenter sur Raspberry.....	34
<i>Tableau 9:</i> Les résultats d'exécution sous Python	40

LISTE DES ABREVIATIONS

RF:Reconnaissance Faciale

IA:Intelligence Artificielle

PIN:Personnal Identification Number

2/3 D: 2/3 Dimensions

ML:Machine Learning

DL: Deep Learning

CNN: Convolutional Neural Network

RNN: Reccurent Neural Network

FC: Fully Connected

BN: Batch Normalization

DO: Droupout

RGB: Red/Green/Blue

LFW: Labeled Faces in the Wild

SD:Secure Digital

RAM:Random Access Memory

RÉSUMÉ

L'objectif principal de cette étude est de développer un système de reconnaissance faciale en utilisant les réseaux de neurones convolutionnels (CNN) et de l'implémenter sur une carte Raspberry Pi 4. La méthodologie se divise en deux volets : le développement d'un algorithme de reconnaissance faciale basé sur les CNN et l'apprentissage profond dans le premier volet, et l'implémentation de cet algorithme sur une carte Raspberry Pi 4 dans le deuxième volet. Une attention particulière a été accordée à l'optimisation des paramètres afin d'améliorer les performances sous l'environnement de programmation googleColab.

Les résultats obtenus ont démontré l'efficacité du système développé, avec une précision de reconnaissance des visages en temps réel atteignant 86%.

Mots clés :

Reconnaissance Faciale, Deeplearning, CNN, Raspberry, Python

المخلص

الهدف الرئيسي من هذه الدراسة هو تطوير نظام التعرف على الوجه باستخدام الشبكات العصبية التلافيفية (CNN) وتنفيذه على لوحة Raspberry Pi 4. تنقسم المنهجية إلى جزأين: تطوير خوارزمية التعرف على الوجه بناءً على شبكات CNN و التعلم العميق في الجزء الأول ، وتنفيذ هذه الخوارزمية على لوحة Raspberry Pi 4 في الجزء الثاني. تم إيلاء اهتمام خاص لتحسين المعلمات من أجل تحسين الأداء في ظل بيئة برمجة google Colab. أظهرت النتائج التي تم الحصول عليها كفاءة النظام المطور، حيث وصلت دقة التعرف على الوجوه في الوقت الفعلي إلى 86%.

الكلمات المفتاحية :

التعرف على الوجه ، التعلم العميق ، Python ، CNN ، Raspberry Pi

ABSTRACT

The main objective of this study is to develop a facial recognition system using convolutional neural networks (CNN) and to implement it on a Raspberry Pi 4 board. The methodology is divided into two parts: the development of a facial recognition algorithm based on CNNs and deep learning in the first part, and the implementation of this algorithm on a Raspberry Pi 4 board in the second part. Particular attention has been paid to optimizing parameters to improve performance under the google Colab programming environment.

The results obtained demonstrated the efficiency of the developed system, with real-time face recognition accuracy reaching 86%.

Keywords:

Face recognition, deep learning, CNN, Raspberry, Python

Introduction générale

L'intelligence artificielle (IA) joue un rôle crucial au 21^e siècle, avec les réseaux de neurones convolutifs (CNN) qui ont révolutionné le traitement d'images. Grâce à leur architecture spécialisée, les CNN sont des outils puissants pour la reconnaissance faciale, notamment dans la sécurité. Les progrès des CNN ont amélioré la précision et l'efficacité des systèmes de reconnaissance faciale, permettant la détection et l'identification des visages dans divers contextes, tels que la sécurité dans les aéroports ou l'authentification biométrique. Les systèmes de reconnaissance faciale basés sur l'IA analysent rapidement et avec précision de grandes quantités de données visuelles, renforçant ainsi les mesures de sécurité et prévenant les intrusions.

Cette étude vise à développer un système de reconnaissance faciale utilisant les CNN et la carte Raspberry Pi, avec une attention particulière portée à l'optimisation des paramètres pour améliorer les performances.

L'idée de base consiste à développer une version légère du réseau CNN pour une implémentation sur la carte Raspberry Pi. Cette version optimisée garantit des performances efficaces tout en prenant en compte les contraintes matérielles de la carte. La mise en œuvre de cette version légère permet une reconnaissance faciale en temps réel, offrant une solution pratique d'identification.

Afin de parvenir à ces objectifs, le système sera d'abord soumis à une simulation et à une exécution dans l'environnement de développement Google Colab. Cette étape permettra de tester et d'optimiser le modèle, en évaluant sa performance et en apportant les ajustements nécessaires. Une fois le modèle optimisé, il sera ensuite déployé sur une carte Raspberry Pi 4. Cette étape garantira une exécution en temps réel et une utilisation pratique du système, en le rendant accessible dans des scénarios réels. Ainsi, la combinaison de la simulation sous Google Colab et de l'implémentation sur Raspberry Pi 4 permettra de mettre en œuvre et de valider efficacement le système dans des conditions réelles d'utilisation.

Ce mémoire est divisé en trois chapitres de la manière suivante :

Le chapitre 1 donne un aperçu de la reconnaissance faciale, de son importance, de ses applications et de ses défis, ainsi que du rôle de l'intelligence artificielle dans ce domaine.

Le chapitre 2 se concentre sur l'apprentissage profond et les réseaux de neurones, en mettant en avant les architectures utilisées dans la reconnaissance faciale, telles que les réseaux de neurones convolutifs (CNN) et récurrents (RNN).

Le chapitre 3 aborde la mise en œuvre pratique du système de reconnaissance faciale en utilisant l'apprentissage profond, en présentant les résultats et les tests de simulation réalisés sur Google Colab ainsi que l'implémentation du système sous une carte Raspberry Pi 4.

Le mémoire se conclut avec une conclusion générale qui résume les résultats, souligne les limitations et propose des perspectives futures pour la recherche.

Chapitre I



*La reconnaissance faciale et
l'intelligence artificielle*

1.1 La reconnaissance faciale

1.1.1 Introduction

À notre époque, et grâce à l'évolution de la technologie, l'homme a créé quelques nouveaux systèmes qui donnent aux êtres humains une opportunité qui les aident dans plusieurs domaines tels que la sécurité, les systèmes de contrôles... etc. Parmi ces évolutions « La reconnaissance faciale (RF) », est l'une des applications les plus populaires au temps présent, elle peut être utilisée pour identifier et reconnaître les visages dans des images ou des vidéos, elle a pour objectif de permettre l'identification et la vérification des individus en se basant sur leurs caractéristiques faciales uniques.

1.1.2 Définition

La reconnaissance faciale est un développement d'un des domaines de l'intelligence artificielle pour la sécurité biométrique, c'est une méthode biométrique qui permet d'identifier, d'analyser ou de confirmer l'identité de l'individu grâce aux traits de visage de personnes sur des photos, dans des vidéos ou en temps réel en faisant un traitement sur les données d'une image capturée en direct avec celles sauvegardées dans le système.

La reconnaissance faciale à partir du visage sert à [1] :

- Authentifier une personne : c'est-à-dire, vérifier qu'une personne est bien celle qu'elle prétend être (dans le cadre d'un contrôle d'accès).
- Identifier une personne : c'est-à-dire, de retrouver une personne au sein d'un groupe d'individus, dans un lieu, une image ou une base de données.

1.1.3 Historique

La 'RF' a été développée par 'Benton et Van Allen' en 1968 pour évaluer la capacité d'identification des visages non familiers, il s'agit d'une épreuve consistant à apparier des photographies de visages non familiers présentés sous différents éclairages et selon des angles différents et nécessitant une bonne capacité d'intégration Visio-spatiale.

L'utilisation des techniques de reconnaissance faciale a connu un développement à grande échelle depuis le milieu des années 90, avec l'utilisation efficace de nouvelles technologies, notamment l'ordinateur et sa capacité de traitement d'images. L'utilisation de ces techniques existe depuis qu'une machine est capable de comprendre ce qu'elle « voit » lorsqu'on la

connecte à une ou plusieurs caméras, c'est-à-dire que les premiers essais datent du début des années 70 (Benton et Van Allen en 1968), et sont basés sur des attributs faciaux mesurables comme l'écartement des yeux, des sourcils, des lèvres, la position du menton, la forme, etc.

Ces méthodes sont très peu robustes, car elles font de nombreuses suppositions en se plaçant dans des cas très simples (visage de face, bonnes conditions d'illuminations, etc.). L'une des premières tentatives de reconnaissance de visage est faite par Takeo Kanade en 1973 lors de sa thèse de doctorat à l'Université de Kyoto [2].

En 2011, tout s'accélère grâce au Deep Learning, une méthode d'apprentissage automatique s'appuyant sur des réseaux de neurones artificiels. C'est l'ordinateur qui sélectionne les points à comparer : plus il est alimenté en images, mieux il apprend.

1.1.4 Mode de fonctionnement d'un système de reconnaissance de visage

La technique de la reconnaissance faciale se réside en associant une face à son identité, ce qui paraît simple pour les êtres humains, mais pour un système automatique, ce dernier aura besoin d'une telle base de données contenant des images classifiées des identités.

Ce processus peut fonctionner en différents modes : un mode enrôlement, un mode identification, et un mode d'authentification

- Le mode enrôlement appuie sur l'extraction des caractéristiques d'une certaine personne, puis les rangées sous la forme d'un vecteur caractéristique appelé la signature, qui est associée à une étiquette d'identité, sera stocké dans une base de données destinée
- En mode authentification, le système compare l'identité d'une personne obtenue avec celles stockées dans la base de données qui est représentée généralement sous forme d'un code PIN, un nom d'utilisateur, une carte à puce... etc. afin de déterminer si la déclaration est vraie ou non
- Le mode d'identification sert à identifier une personne à partir de son image faciale, ce système fait une comparaison entre sa signature avec celles qui existent dans la base de données, afin d'évaluer la correspondance. [3]

Dans chacun des modes précédents, le système exécute des opérations essentielles, à savoir :

- l'acquisition de l'image.
- le prétraitement.
- l'extraction des caractéristiques.
- la classification.

- l'apprentissage.
- la décision. [4]

1.1.5 Techniques de reconnaissance faciale

Reconnaissance faciale depuis images fixes et des séquences d'images :

La reconnaissance automatique des visages comporte trois étapes/sous tâches essentielles :

- Repérage et normalisation approximative des visages.
- Extraction et normalisation fine des caractéristiques faciales.
- Identification et/ou vérification.

La détection des visages et l'extraction des caractéristiques peuvent se faire en même temps, selon le type d'application, tels que les dimensions des bases de données d'apprentissage et de test, la complexité et la variabilité de l'arrière-plan, le bruit, l'occlusion et les contraintes de rapidité, certaines des sous-tâches peuvent être très difficiles. [5]

Reconnaissance faciale, basée sur la 3D :

La reconnaissance faciale présente des difficultés qui limitent son utilisation à grande échelle du point de vue de l'opérateur. La plus importante de ces difficultés est la précision. [5] Les systèmes actuels de reconnaissance faciale 2D peuvent être faussés par des variations de pose, d'éclairage, d'expressions et d'autres caractéristiques qui peuvent changer entre les prises d'un visage humain. On reconnaît aujourd'hui que, pour surmonter les problèmes de précision, il faut recourir à différentes modalités de capture (comme la 3D ou l'infrarouge).

Reconnaissance faciale, basée sur la vidéo :

La reconnaissance faciale à partir de la vidéo a fait l'objet de nombreuses recherches au cours de la dernière décennie. Ces travaux ont généralement visé à obtenir une reconnaissance faciale précise dans des conditions de vision très dégradées. Dans les environnements classiques de capture d'images faciales, comme les bureaux de passeport ou les commissariats, les facteurs de nuisance tels que la pose de la tête ou l'expression du visage sont maîtrisés. En revanche, les systèmes de vidéosurveillance ne peuvent pas contrôler ces facteurs, de sorte que les activités des individus filmés et les effets de l'environnement peuvent varier fortement [5].

1.1.6 Les domaines d'application de la reconnaissance faciale

La reconnaissance des visages suscite un grand intérêt car le visage est la partie du corps humain la plus significative et la plus utile pour l'identification d'une personne. Aujourd'hui, ce système est de plus en plus utilisé pour assurer la sécurité, non seulement sur les Smartphones comme auparavant, mais dans de nombreux domaines.

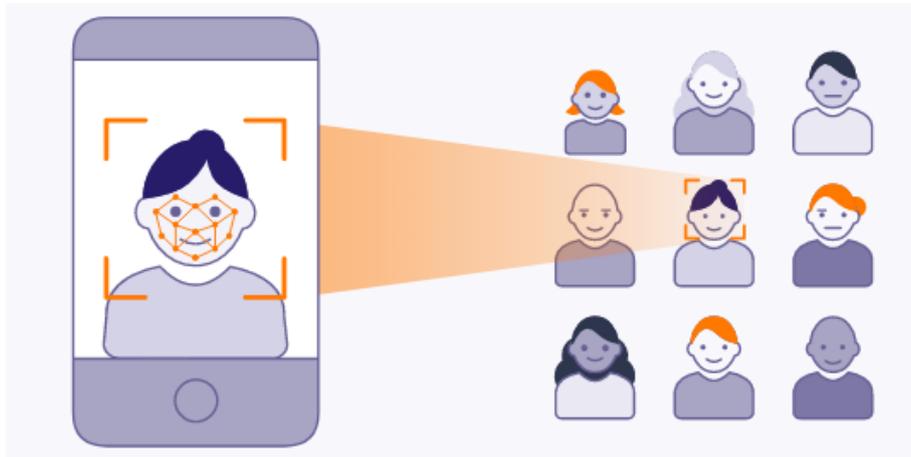


Figure 1: Exemple sur la reconnaissance faciale

Voici quelques exemples :

- **Les systèmes de contrôle d'accès automatique:**

Une caméra doit être située à l'entrée transfère des images à un ordinateur qui fait une reconnaissance de visage et puis décide l'accord d'accès

- **Les systèmes de télésurveillance et d'identification criminelle:**

Pour les enquêteurs la vérification des identités des suspects peut être assistée par ordinateur

- **Les interfaces homme-machine :**

On peut rendre la communication entre l'homme et la machine plus attractive, plus naturelle et plus confortable. [6]

Et autres domaines aussi tels que : l'accès à un compte bancaire, l'authentification des passagers dans les aéroports... etc.

1.1.7 Les avantages et les inconvénients de la reconnaissance faciale

1. Les avantages de la reconnaissance faciale:

La technique de la détection de visage a été faite pour faciliter la vie au quotidien, et essentiellement pour bien gérer les diverses activités qui dépendent du système de biométrie.

Une meilleure sécurité

Comme on a dit avant, la reconnaissance faciale permet d'identifier les personnes, donc peut servir à bien sécuriser l'entourage ou on l'utilise en verrouillant les outils personnels par exemple.

Plus de facilité

En utilisant cette technique, les gens peuvent gérer leurs besoins justes avec leurs visages, comme dans les magasins ou dans les salles de sport, il suffit juste d'être en face la caméra et tout est réglé sans avoir besoin de prendre les clés ou bien de l'argent ou même un mot de passe. Ça évite les queues d'attente ainsi qu'elle personnalise le service client.

Un traitement plus rapide

Le principe de détection est très rapide, le temps d'exécution de ce concept ne dure que quelques secondes, ce qui lui donne privilège

Intégration avec d'autres technologies

La méthode de reconnaissance faciale peut être intégré avec plusieurs logiciels, elle est compatible avec les grands logiciels de sécurité, elle peut aussi être utilisé comme outil de déverrouillage des appareils mobiles et aussi dans autres divers domaines

La détermination et la précision

Si l'on élabore une petite comparaison entre l'identification et la biométrie, on constate largement que cette dernière est plus sécurisante et plus fiable par rapport à l'identification, puisque la biométrie nous permet de déterminer l'identité exacte et directe d'une personne pour pouvoir accéder à son compte, tandis que l'identification se base sur l'utilisation de données et informations que quiconque peut détenir telles que mot de passe, nom d'utilisateur ...etc.

2. Les inconvénients de la reconnaissance faciale

Si on parle des avantages de la reconnaissance faciale, on voit qu'il y en a plusieurs, mais malgré tous ces bienfaits, on n'oublie jamais que cette technique possède quelques côtés négatifs par rapport à son concept

La surveillance, violation de la vie privée

Plusieurs personnes n'acceptent pas le fait d'être pris en photo ou bien d'être surveillé par une caméra et n'aiment pas que leurs visages soient stockés dans une base de données, puisque la reconnaissance faciale permet de détecter des gens spéciaux et la même chose pour les gens ordinaires ce qui menace la vie privée des êtres humains

Stockage massif des données

La technologie de la reconnaissance faciale exige l'intelligence artificielle, cette dernière possède un énorme ensemble de données qui a un stockage massif pour « apprendre » à donner des résultats précis, et cela n'est pas utile pour les établissements.

Piratage

Comme on le sait tous, cette technologie est un outil technique informatisé et programmé qui utilise de l'intelligence artificielle, cette dernière peut être piratée et manipulée par des hackers, et l'utiliser pour des actes illégaux, ce qui la rend un danger pour la société.

1.1.8 Les difficultés de la reconnaissance faciale

La détection des visages pour le cerveau humain est une technique très facile à faire, d'ailleurs l'être humain généralement ne trouve aucune difficulté pour reconnaître des personnes d'après la tâche visuelle. Par contre, construire un système automatique de détection peut faire face à un défi par rapport aux conditions d'acquisition des images, et cela constitue l'une des oppositions que le système puisse affronter. À signaler aussi, le taux de la reconnaissance faciale dépend de la variation de pose. Sans oublier l'expression faciale qui est un facteur qui a une grande influence, et autres difficultés qui déstabilisent la structure de détection [7].

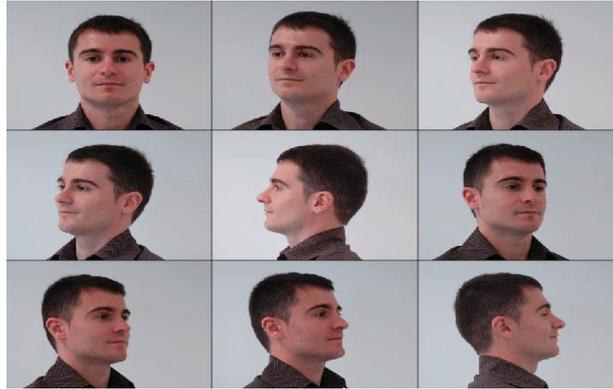


Figure 2: Variation de pose



Figure 3: Variation d'éclairage



Figure 4: Variation d'expression

En tenant compte les autres facteurs intervenants dans l'identification des visages tels que :

- le vieillissement.
- la chirurgie plastique.
- le maquillage.
- les effets de la consommation de drogues ou de tabac.
- L'occultation partielle. [1]

Donc, il est essentiel de mettre des images de haute qualité. Une qualité moyenne ou médiocre peut empêcher leur prise en compte lors des recherches dans le système, cela peut grandement compromettre la précision de cette opération et affecter les résultats.

L'idéal est d'avoir une photo d'identité avec un éclairage homogène et un fond neutre.

1.2 l'intelligence artificielle

1.2.1 introduction

La technologie s'impose de jour en jour grâce à son développement, la valeur qu'elle a dans notre vie est plus en plus discutée, surtout qu'on parlant précisément sur l'intelligence artificielle (IA). Cette dernière est un domaine de recherche qui vise à réaliser des systèmes et des machines autonomes capables de rendre facile la vie des gens, compléter leurs cognitifs et même les dépasser, L'IA englobe des sous-domaines variés, tels que l'apprentissage automatique, la vision par ordinateur, le traitement du langage naturel, la robotique, les systèmes experts, les réseaux de neurones artificiels, etc. Ce domaine est le sujet abordé par tout le monde, en demandant si ça va changer notre quotidien au meilleur ou au pire.

1.2.2 Définition

L'intelligence artificielle est un ensemble de théories et de stratégies qui a pour objectif de reproduire les capacités humaines, ainsi que les comportements intelligents observés dans la vie quotidienne, notamment ce qui concerne sa capacité de reconnaissance des visages, des objets...etc., et à résoudre des problèmes complexes.

Sa définition reste assez floue, elle est aussi large que la définition que vous pouvez avoir de l'intelligence en général [8], mais on peut la définir comme une technique qui permet au machine ou ordinateur d'imiter l'intelligence humaine. C'est un champ de recherche qui regroupe l'ensemble des techniques et méthodes qui servent à comprendre et reproduire le fonctionnement d'un cerveau humain.

1.2.3 L'histoire et l'évolution de l'intelligence artificielle

L'intelligence artificielle (IA) est une discipline jeune qui a connu plusieurs étapes depuis sa création dans les années 1940-1960. Elle a été marquée par des phases de déception, selon les avancées scientifiques, les financements et les attentes du public. Elle a exploré différents domaines et techniques, tels que la logique mathématique, la traduction automatique, la vision artificielle, la reconnaissance vocale, l'apprentissage automatique ou la robotique. Elle a aussi soulevé des questions éthiques, sociales et philosophiques sur le

rapport entre l'homme et la machine, les limites de l'intelligence et la responsabilité des décisions. [9]

Aujourd'hui, l'IA bénéficie de la puissance de calcul des ordinateurs et de l'accès à des données massives, mais elle reste loin d'atteindre une intelligence artificielle forte, c'est-à-dire capable de comprendre et de modéliser le monde dans sa complexité.

1.2.4 Les domaines d'application de l'intelligence artificielle

L'intelligence artificielle est présente en permanence dans notre vie quotidienne, elle est utilisée dans de nombreux domaines [10], notamment :

- Le traitement du langage naturel, qui permet aux machines de comprendre et de produire du texte ou de la parole
- La vision par ordinateur, qui permet aux machines de reconnaître et d'analyser des images ou des vidéos
- Les agents intelligents, qui sont des entités autonomes capables d'interagir avec leur environnement et d'atteindre des objectifs
- Les systèmes multi-agents, qui sont des ensembles d'agents intelligents qui coopèrent ou se concurrencent pour résoudre des problèmes complexes
- Les réseaux de neurones artificiels, qui sont des modèles inspirés du fonctionnement du cerveau humain et qui permettent d'apprendre à partir de données.[10]
- Technologie d'automatisation de la maison intelligente : Presque tout ce qui automatise votre maison utilise l'intelligence artificielle.

On trouve l'IA dans d'autres domaines tels que l'aviation, la sécurité, la reconnaissance, l'informatique ou les réseaux sociaux.

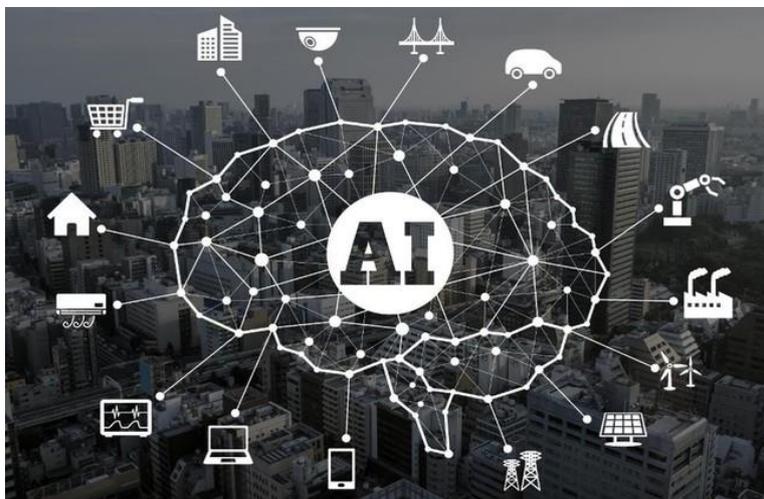


Figure 5: Domaines d'application de l'IA

1.2.5 La reconnaissance faciale et l'intelligence artificielle

La reconnaissance faciale avec l'intelligence artificielle est donc une technologie qui offre de nombreuses possibilités, mais qui pose aussi des questions éthiques et juridiques sur le respect de la vie privée et des droits humains. Il est important de comprendre son fonctionnement et ses implications pour en faire un usage responsable et éclairé [11,12].

1.3 Conclusion

La reconnaissance faciale est une technologie de surveillance et de sécurité qui fonctionne de manière fiable et sécurisée par rapport à d'autres technologies utilisées auparavant. Aujourd'hui, la reconnaissance faciale est largement utilisée dans de nombreux domaines de sécurité, tels que le contrôle d'accès aux bâtiments et aux entreprises, etc.

Ce chapitre est divisé en deux parties essentielles. Dans la première partie, nous avons d'abord défini ce qu'est la reconnaissance faciale, puis nous avons discuté de son fonctionnement général et de ses domaines d'application. Enfin, nous avons abordé les défis auxquels elle peut être confrontée.

Dans la deuxième partie, sans entrer dans les détails, nous avons mentionné l'intelligence artificielle et ses diverses utilisations, parmi lesquelles la reconnaissance faciale. Pour mettre en place un système de reconnaissance faciale, une étude approfondie des réseaux de neurones doit être réalisée, ce qui sera présenté dans le chapitre suivant.

Chapitre II



Apprentissage basé sur les réseaux de neurones convolutifs et Raspberry Pi

II.1 Apprentissage basé sur les réseaux de neurones

II.1.1 Introduction

Le Deep Learning est une technique d'intelligence artificielle, qui permet à des systèmes d'apprendre et d'évoluer à partir de leurs données, sans avoir besoin de modifier leur code. Il repose sur l'utilisation de réseaux de neurones artificiels, capables de modéliser des représentations complexes des données. Dans ce chapitre, nous allons explorer le domaine du Deep Learning et des réseaux de neurones convolutionnels (CNN). Nous allons également présenter la carte Raspberry Pi, qui peut exécuter des applications de Deep Learning.

II.1.2 Définition des mots clés

Machine Learning (ML)

L'apprentissage automatique ou machine Learning (ML), est un domaine qui combine l'intelligence artificielle et l'informatique, utilisant des algorithmes et des modèles statistiques pour permettre aux ordinateurs d'apprendre à partir de données sans qu'on leur dise explicitement comment apprendre [13].

Deep Learning (apprentissage profond)

L'apprentissage profond, aussi appelé en anglais 'Deep Learning', désigne des techniques d'apprentissage automatique qui utilisent des réseaux neuronaux. C'est un type d'intelligence artificielle où la machine est capable d'apprendre par elle-même. [14]

Cette méthode permet de créer des systèmes informatiques et des applications qui effectuent des tâches complexes.

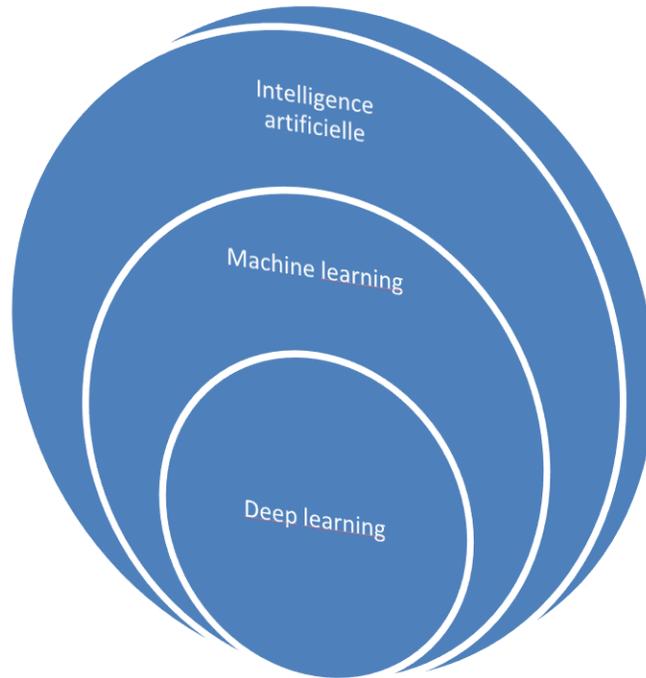


Figure 6: La relation entre IA, ML et Deep learning

II.1.3 Différents types d'apprentissage

Il existe essentiellement deux types d'apprentissage :

L'apprentissage non supervisé :

Dans l'apprentissage non supervisé, les données sont analysées et regroupées sans étiquettes ni intervention humaine. Les algorithmes d'apprentissage non supervisé cherchent à découvrir des motifs cachés ou des groupements de données dans un ensemble d'apprentissages formé uniquement de données. Aucun résultat attendu n'est donné au réseau. Dans ce cas, l'apprentissage consiste à identifier les ressemblances et les différences dans l'ensemble d'apprentissages. Les poids et les sorties du réseau tendent, en théorie, vers les représentations qui capturent les régularités statistiques des données [20], exemple :

- Carte auto organisatrice (Self Organizing Maps SOM)
- Auto encodeur (Auto encoder AE)

L'apprentissage supervisé :

Lorsqu'on présente un motif à un réseau et qu'on lui impose un état final souhaité vers lequel il doit tendre, on parle d'apprentissage supervisé. Ce type d'apprentissage repose sur plusieurs exemples de paires entrées-sorties. [20]

- Réseau de neurones récurrent (RNN).
- Réseau de neurone convolutif (CNN).

II.1.4 La différence entre ML et Deep Learning

Le Deep Learning est un mode de ML. Les deux approches s'appuient sur le principe de l'apprentissage machine. Il existe des différences notables entre ces deux concepts qu'on peut récapituler par ces deux graphes suivants

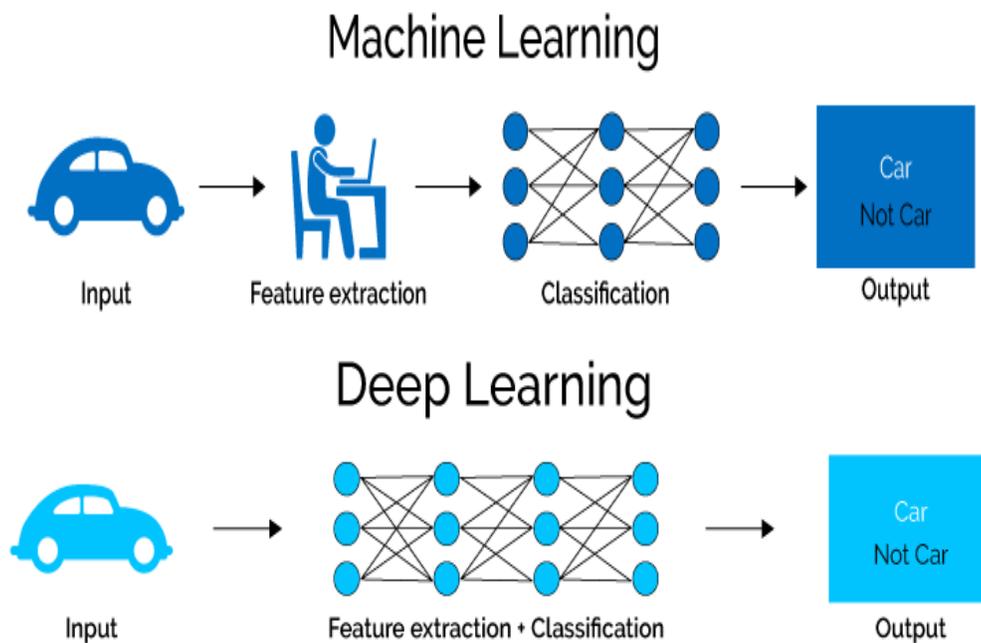


Figure 7: Machine Learning et Deep Learning

II.1.5 Les avantages de Deep Learning

L'apprentissage profond a la capacité d'analyser, de traiter et d'apprendre des représentations de données dans leur forme brute, sans modification. Cette méthode est très restreinte dans l'apprentissage machine et pourrait être impossible.

L'avantage de DL sur ML est qu'il est capable de saisir les caractéristiques des données brutes sans les extraire manuellement [18].

Cela permet de gagner du temps et des efforts, surtout pour les données très volumineuses. De plus, l'apprentissage profond permet généralement d'obtenir des résultats plus performants et plus exacts.

II.1.6 Domaines d'application du Deep Learning

Les performances des systèmes de Deep Learning dépassent largement celles des systèmes de Machine Learning classiques, car elles exigent d'activer des milliers d'hyper paramètres, et d'algorithmes pour qu'un modèle de Deep Learning devienne efficace.

L'algorithme permet à l'ordinateur d'apprendre à partir des nouvelles informations qu'il reçoit. Le réseau de neurones dont est doté l'ordinateur fait en sorte qu'il analyse des exemples de manière à devenir capable d'effectuer une tâche. Ces exemples sont étiquetés. C'est ce processus qui a rendu les ordinateurs capables de reconnaître des objets sur des images, parfois mieux que le cerveau humain lui-même [1]

Le Deep Learning est basé sur les réseaux de neurones dont le principe est d'apprendre au système à reconnaître, et à prendre des décisions de manière autonome, il est donc utilisé dans plusieurs domaines tels que :

- Reconnaissance d'image.
- Traduction automatique.
- Voiture autonome.
- Diagnostic médical.

II.1.7 Réseaux de neurones

1) Définition

Les neurones biologiques

Sont des cellules excitables connectées l'une à l'autre, et ayant un rôle de transmettre des informations dans notre système nerveux [16]

Les réseaux de neurones artificiels

Sont des modèles mathématiques qui ont été inspirés par des neurones biologiques, ce que veut dire plusieurs neurones travaille ensemble, reçoivent des signaux d'entrées traitent des informations et déclenchent un signal de sortie

Ils sont utilisés dans la prédiction et la classification dans le cadre de découverte de connaissances dirigée [17]. Cependant, on leur reproche souvent d'être une "boîte noire" : il est difficile de savoir comment les résultats sont produits, ce qui rend les explications délicates, même si les résultats sont bons.

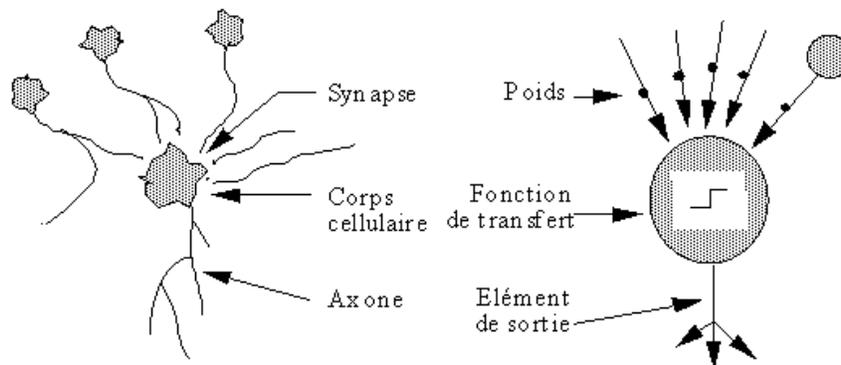


Figure 8: Mise en correspondance neurone biologique / neurone artificiel

2) Architecture d'un réseau de neurones artificielle

Un réseau de neurones se constitue d'une structure qui se divise en trois parties essentielles qui sont : une couche d'entrées qui est l'intermédiaire entre le réseau et la donnée dont elle transforme cette dernière simplement sans la modifier, cette couche se suit par des couches de traitement (autrement dit couches cachées ou bien HiddenLayers). Les couches cachées sont

désignées pour effectuer de nombreuses transformations intéressantes et nécessaires pour répondre aux exigences de la solution pour le problème donné [18].

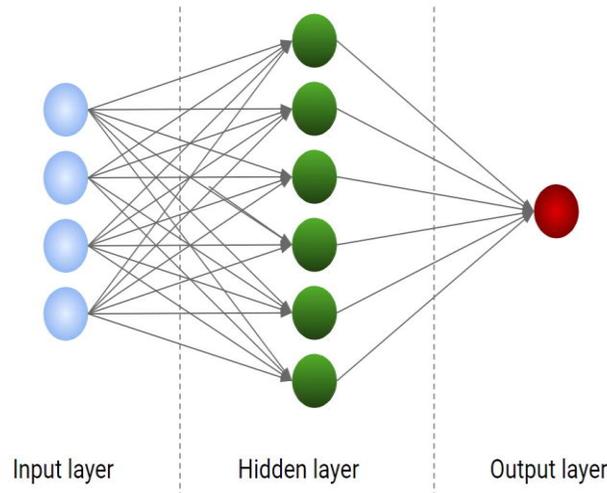


Figure 9: Structure simple d'un réseau de neurones

3) *Les réseaux de neurones convolutifs (CNN)*

Les réseaux de neurones convolutif ou (CNN pour Convolutional Neural Network en anglais), sont des réseaux de neurones multicouches spécialisés dans la classification des images et la reconnaissance de formes.

Contrairement aux techniques d'apprentissage supervisé, les réseaux de neurones convolutifs apprennent à partir des caractéristiques de chaque image. C'est là que réside leur force : les réseaux font tout le boulot d'extraction des caractéristiques automatiquement, contrairement aux autres techniques d'apprentissage.

Structure de base d'un CNN

Un réseau de neurones convolutif (CNN) possède au moins une couche de convolution, qui utilise des filtres pour transformer l'entrée, et souvent des couches de pooling, qui diminuent la dimension de l'entrée. Il possède aussi des couches entièrement connectées, et des couches de classification, comme la couche Softmax. Les CNN sont adaptés aux entrées qui ont une forme

bidimensionnelle, comme les images ou les sons. Ils sont connus pour être faciles à entraîner. La figure 12 présente la structure typique d'un CNN. [19]

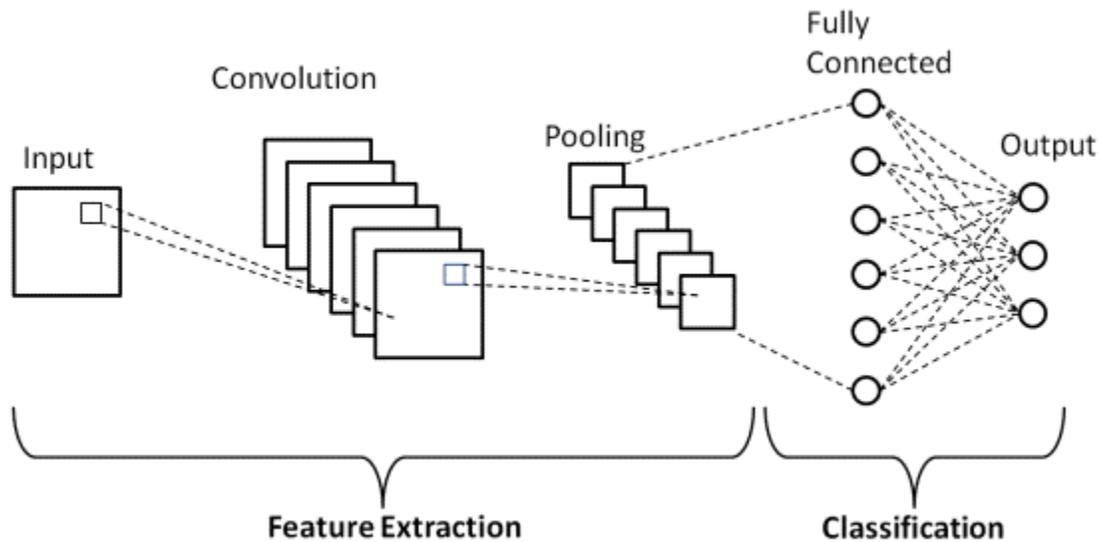


Figure 10: Structure d'un CNN

Architecture d'un CNN

Un simple ConvNet est une séquence de couches, et chaque couche d'un CNN transforme un volume d'activation en un autre grâce à une fonction différentiable. Trois principaux types de couches sont utilisés pour créer l'architecture de ce dernier : la couche convolutive (Convolutional layer), la couche de regroupement (Pooling layer) et la couche entièrement connectée (Fully-connected layer) (exactement comme dans les réseaux de neurones classiques). [18]

- **Les couches convolutives :** reçoivent en entrée plusieurs images. Les réseaux de neurones convolutifs utilisent ces couches afin de filtrer l'image d'entrée avec des noyaux de convolution qui apprennent à reconnaître des caractéristiques visuelles pertinentes, comme les angles, les contours ou les motifs. Ces caractéristiques sont ensuite combinées en représentations de plus en plus complexes, ce qui permet au réseau CNN de comprendre, et d'interpréter efficacement les images.
- **Couche de Pooling :** La mise en commun est une opération qui réduit la taille des caractéristiques extraites (en faisant du sous-échantillonnage) tout en gardant les

informations les plus pertinentes (en général, avec la mise en commun maximale), ce qui rend le réseau moins sensible au bruit on peut utiliser différentes fonctions pour faire la mise en commun :

- **Max Pooling** : On garde la valeur la plus élevée de la zone de pooling (comme sur la Figure d'exemple). [1]
- **Average Pooling** : C'est une opération qui réduit la taille d'un pool en prenant la moyenne des valeurs, ce qui permet de simplifier les caractéristiques extraites par rapport au Max Pooling. [18]

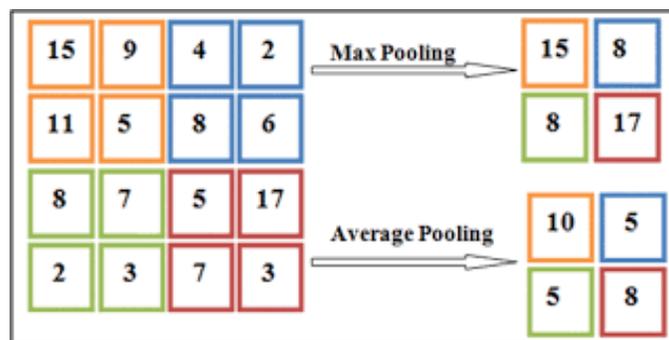


Figure 11: Exemples max pooling et average pooling

- **Couche entièrement connectée (Fully-connected layer FC)** : Les couches entièrement connectées sont les dernières couches d'un réseau de neurones convolutif. Elles sont reliées à toutes les activations de la couche précédente, comme dans un réseau de neurones ordinaire. Elles ont la même structure qu'une couche d'un réseau de neurones classique. Les CNN cherchent à réduire une fonction objective, souvent une fonction de perte. La fonction de perte la plus fréquente est la fonction Softmax. Elle transforme les sorties en une distribution de probabilité sur les différentes classes, dont les valeurs sont entre 0 et 1. La couche entièrement connectée permet d'apprendre des combinaisons non linéaires des caractéristiques extraites, ce qui peut renforcer la performance de classification. [8]

Il existe d'autres couches qui ont été ajoutées comme des cas spéciaux afin d'améliorer les performances du modèle, telles que

- **Couche de normalisation par lots (Batch normalization layer BN)** : La normalisation par lots est une technique d'apprentissage de réseaux de neurones

profonds, cette méthode sert à rendre les réseaux de neurones plus rapides, performants et stables.

- **Une couche de désactivation (Dropout layer DO) :** est une technique utilisée pour réduire le sur apprentissage et améliorer la capacité de généralisation des réseaux neuronaux profonds. Elle consiste à désactiver de manière aléatoire certaines entrées de la couche précédente ce qui permet de limiter le sur apprentissage [21]. On peut distinguer les couches qui contiennent des paramètres de celles qui n'en contiennent pas.
- **Couche d'unité linéaire rectifiée (Rectified Linear Unit layer ReLU) :** c'est une couche à laquelle la fonction d'activation est appliquée sur les éléments du volume d'entrée. Cette fonction conserve les valeurs positives et met à zéro les valeurs négatives. Elle permet d'ajouter des non-linéarités au réseau. Cette couche est un cas spécial qui a été utilisé pour la première fois par AlexNet en 2012.

4) Les modèles CNN utilisés pour la reconnaissance faciale

Les réseaux CNN ont connu une évolution des architectures depuis leur apparition. La figure 12 montre cette évolution au fil du temps.

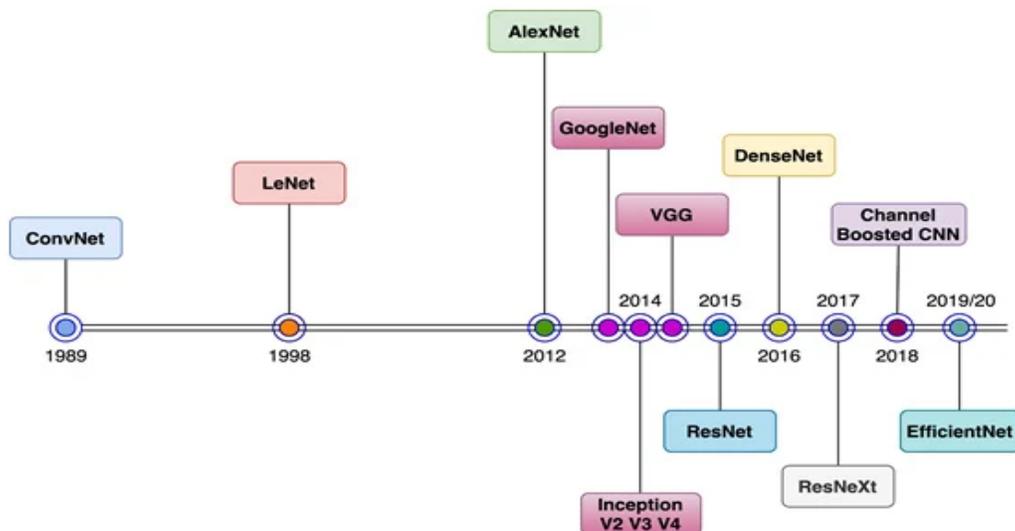


Figure 12: Evolution des architectures des CNNs au fil du temps

- **AlexNet :** Ce réseau se compose de cinq blocs de convolution-pooling, suivis de trois couches entièrement connectées pour la classification. [22]. Il a été entraîné sur le jeu de

données Imagenet, qui contient 15 millions d'images en haute résolution de taille $256 \times 256 \times 3$. Pour la première fois, la fonction d'activation ReLU a été utilisée entre les couches de convolution et les couches de pooling, ainsi que le pooling chevauchant avec un pas inférieur à la taille de la fenêtre.

- **VGGNet** : Ce réseau contient 16 couches de convolution, et se distingue par son architecture très régulière. Ce réseau ressemble à AlexNet mais il améliore les performances en utilisant des filtres plus petits. [23].
- **DenseNet** : Le nombre de couches dans un DenseNet peut varier selon le modèle, mais il est généralement composé de plusieurs blocs denses, séparés par des couches de transition. Dans ces blocs, chaque couche est reliée à toutes les autres couches qui viennent après elle. Cela permet d'avoir des réseaux plus profonds et plus efficaces à entraîner. [19]
- **ResNet** : L'idée principale de ResNet est de résoudre le problème de la dégradation de la performance qui survient lorsqu'un réseau devient plus profond, il utilise des liens résiduels entre les couches pour faciliter la mise à jour des poids. [15]
- **FaceNet** : est un modèle basé sur un réseau de convolution profonde, pour reconnaître spécialement les visages. Il peut obtenir des représentations de visage très distinctes dans un espace de caractéristiques.

L'architecture de FaceNet emploie des techniques avancées. Il a pour but de placer chaque visage d'une même personne dans un espace de caractéristiques, où les visages proches sont rassemblés et les visages lointains sont séparés par une marge. Cela permet d'avoir une représentation concise et discriminante pour chaque visage, sans être affectée par les variations d'image comme la lumière, la position ou l'expression. [24]

Voici en détail l'architecture du modèle :

- **Couche de convolution** : Le modèle commence par des couches de convolution qui extraient les caractéristiques visuelles des images d'entrée. Elles appliquent des filtres de petite taille (3×3) sur les images, en déplaçant le filtre d'un pas (stride) d'une unité à la fois. Cette opération permet de capturer des informations locales et de détecter des motifs visuels spécifiques tels que les contours, les textures et les formes dans les différentes parties de l'image. Au fur et à mesure que l'information passe à travers les couches de convolution successives, les filtres deviennent plus complexes et peuvent

détecter des caractéristiques plus abstraites et significatives pour la reconnaissance des visages.

- **Couches de normalisation par lot (BatchNormalization)** : Après chaque couche de convolution, une couche de normalisation par lot est appliquée pour accélérer la convergence du modèle en normalisant les activations.
- **Fonctions d'activation ReLU** : Les fonctions d'activation ReLU introduisent la non-linéarité dans le modèle en remplaçant les valeurs négatives par zéro, ce qui permet d'apprendre des relations non linéaires entre les caractéristiques. Ainsi, en utilisant des fonctions d'activation ReLU, le modèle est capable d'apprendre des représentations plus riches et non linéaires des caractéristiques des visages, ce qui améliore ses performances de reconnaissance.
- **Couches de pooling (MaxPooling2D)** : Les couches de pooling réduisent la dimension spatiale des caractéristiques tout en préservant les informations les plus importantes.
- **Couches entièrement connectées** : Les couches entièrement connectées apprennent des combinaisons linéaires des caractéristiques extraites et projettent les données dans un espace de dimension réduite. Dans le modèle FaceNet, il y a une couche Dense avec 128 neurones.
- **Couche d'embedding (Lambda)** : La couche d'embedding normalise les embeddings en utilisant la fonction L2-norm, ce qui facilite les calculs de similarité et de distance entre les visages.
- **Couche de classification (Dense)** : Le modèle se termine par une couche Dense avec une fonction d'activation softmax qui effectue la classification des visages en utilisant le nombre de classes spécifié.

En résumé, le modèle FaceNet est optimisé pour la reconnaissance faciale en créant des représentations distinctes et discriminantes des visages. Son architecture utilise des couches de convolution, de normalisation, des fonctions d'activation, des couches de pooling et des couches entièrement connectées pour extraire et représenter les caractéristiques des visages.

II.2 Le Raspberry Pi

II.2.1 Définition

C'est un ordinateur de petite taille, il est de faible coût qui fonctionne avec un processeur ARM. Il peut être connecté à un écran via le connecteur HDMI, ainsi qu'à un clavier et une souris classique. Il existe plusieurs versions de Raspberry Pi (Raspberry Pi 4, Raspberry Pi 3 b, B+, 4, zéro ...etc). [25]

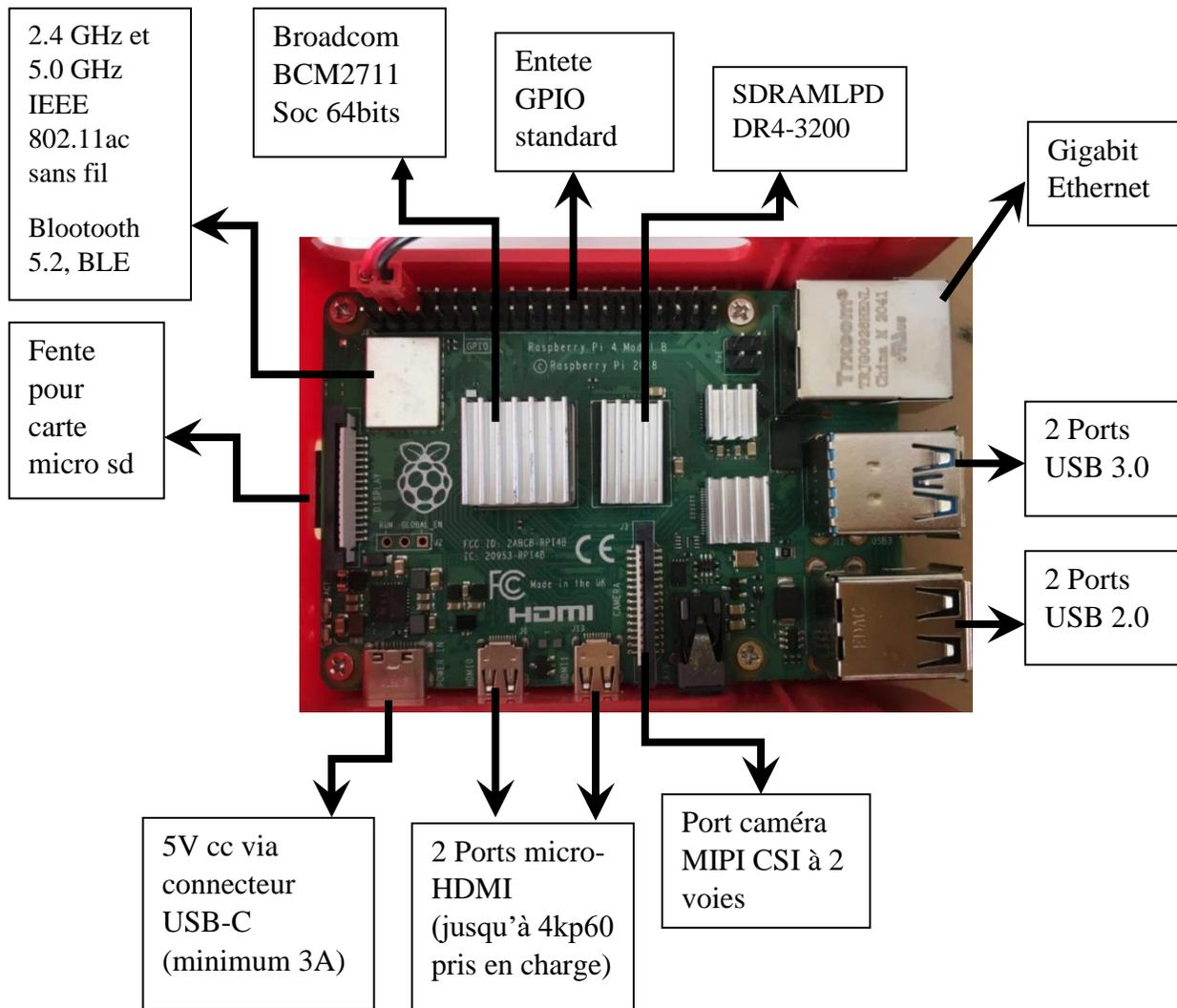


Figure 13: Raspberry pi modèle B

II.2.2 Les avantages du Raspberry PI

- Petit et facile à transporter, ce qui le rend adapté pour des applications de reconnaissance faciale nomades ou intégrées
- Compatible avec l'écran tactile de 7 pouces qui peut se connecter et montrer les images et les noms des visages reconnus.
- Peut utiliser l'apprentissage automatique pour analyser des images et des vidéos en temps réel grâce à la bibliothèque OpenCV
- Compatible avec la bibliothèque face_recognition qui permet de détecter les visages.

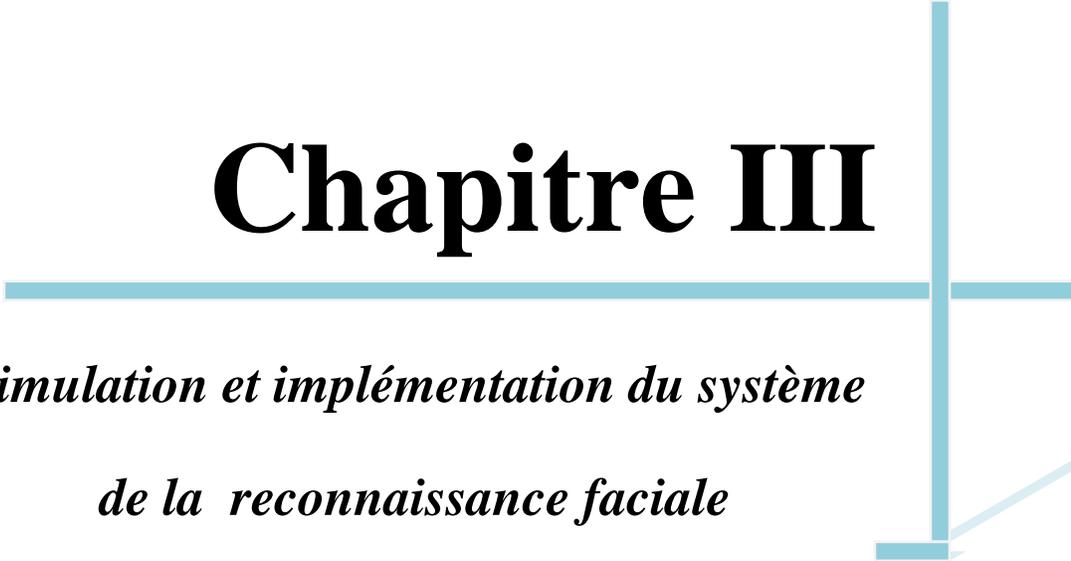
II.3 Conclusion

Dans ce chapitre, nous avons abordé les notions du deep learning, les réseaux de neurones convolutionnels (CNN), ainsi que quelques architectures adaptées à la reconnaissance faciale.

Nous avons aussi introduit brièvement le Raspberry Pi 4 que nous allons utiliser par la suite dans la partie pratique.

Dans le chapitre suivant, nous fournirons plus de détails sur la réalisation et les étapes d'implémentation sur la carte Raspberry Pi, ainsi que les tests réalisés dans Google Colab.

Chapitre III



*Simulation et implémentation du système
de la reconnaissance faciale*

III.1 Introduction

L'objectif principal de cette étude est de développer un système de reconnaissance faciale en utilisant les réseaux de neurones convolutionnels (CNN) et en l'implémentant sur une carte Raspberry Pi 4. Ce système permettra l'accès à des immeubles ou zones sécurisées une fois développé. La méthodologie de travail se divise en deux volets : le développement d'un algorithme de reconnaissance faciale basé sur les CNN et l'apprentissage profond dans le premier volet, et l'implémentation de cet algorithme sur une carte Raspberry Pi 4 dans le deuxième volet.

Dans la suite, nous présenterons d'abord la méthodologie de traitement et la base de données utilisée, puis nous passerons à l'application de l'algorithme dans l'environnement Colab et sur Raspberry Pi. Pour le développement des algorithmes, nous avons utilisé le langage de programmation Python avec différentes bibliothèques telles que TensorFlow et Keras. L'annexe contient les bibliothèques utilisées.

III.2 Description de la méthodologie du travail

L'algorithme du système étudié est principalement composé de quatre étapes : le chargement de la base de données, le prétraitement des images, le modèle CNN et l'évaluation des performances.

Les étapes de traitement comprennent :

- Chargement de la base de données
- Prétraitement
- Modèle CNN
- Évaluation des performances

Ces étapes de traitement seront appliquées dans l'environnement Colab pour étudier et optimiser les performances de l'algorithme. L'algorithme optimisé sera ensuite implémenté sur la carte Raspberry Pi.

La Figure 14 présente en détail les différentes étapes de traitement que nous avons suivies pour développer notre système de reconnaissance.

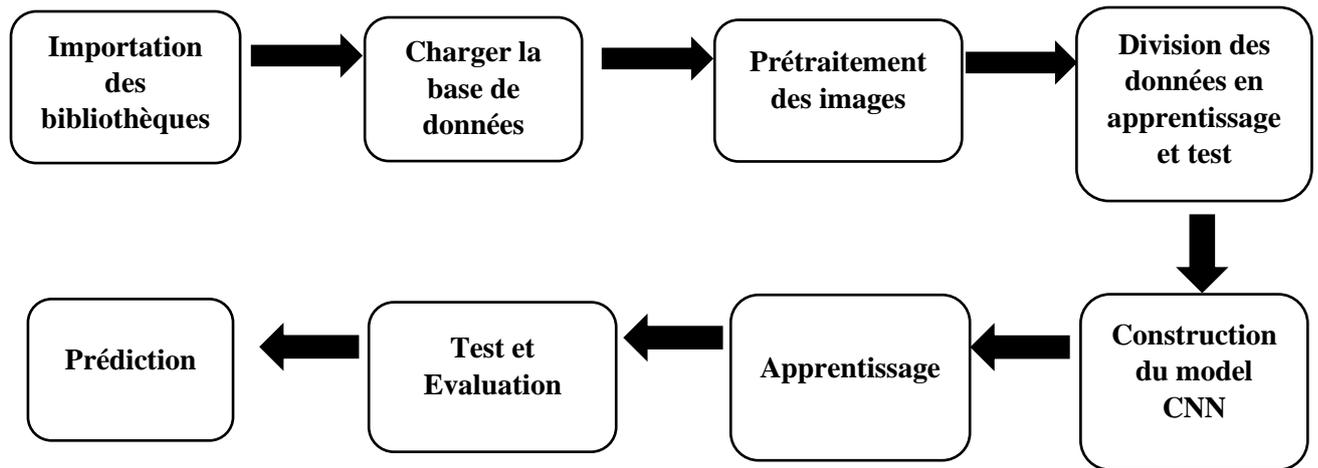


Figure 14: Les différentes étapes de l'implémentation

III.3 Choix des paramètres du modèle FaceNet

Les paramètres du modèle sont définis en fonction des besoins spécifiques de la tâche de reconnaissance faciale. Nous avons utilisé la configuration suivante :

- **Entrée du modèle** : La taille de l'image d'entrée est définie comme étant de 250x250 pixels avec 3 canaux de couleur (RGB).
- **Nombre de classes** : Nous avons choisi d'avoir 20 classes correspondant au nombre de personnes dans notre base de données.
- **Méthode d'optimisation** : Nous avons opté pour la méthode ADAM (Adaptive Moment Estimation), un algorithme d'optimisation couramment utilisé dans l'apprentissage profond. Ce choix dépend du problème spécifique et de la taille des données.
- **Minimisation de la fonction perte** : Pour minimiser la fonction perte, nous avons utilisé une méthode basée sur l'entropie croisée (cross-entropy), afin d'améliorer les performances de classification.
- **Évaluation des performances** : Nous avons mesuré les performances du modèle tout au long de l'entraînement en utilisant l'exactitude (accuracy), qui représente le rapport entre le nombre de prédictions correctes et le nombre total de prédictions.

III.4 Apprentissage du modèle

L'apprentissage du modèle a été réalisé en suivant les étapes suivantes :

- **Création des bases apprentissage et test** : Le jeu de données a été divisé en ensembles d'apprentissage et de test, où 80 % des données ont été réservées pour l'apprentissage et 20 % pour le test. Cette division permet d'évaluer les performances du modèle sur des données indépendantes.
- **Définition des Epochs et de la taille du lot (Batch size)** : Les Epochs représentent le nombre de passages complets du modèle sur l'ensemble des données d'entraînement. La taille du lot indique le nombre d'exemples d'entraînement utilisés pour mettre à jour les poids du modèle à chaque itération. Ces paramètres sont choisis en fonction de la nature du problème, de la taille des données et de la complexité du modèle. L'objectif est de trouver un équilibre pour éviter le surapprentissage et obtenir de bonnes performances sur de nouvelles données.
- **Entraînement du modèle** : Le modèle a été entraîné en utilisant les images normalisées en tant que données d'entrée et les étiquettes encodées en tant que données de sortie. La normalisation des images garantit une mise à l'échelle cohérente des valeurs des pixels, favorisant ainsi la convergence de l'apprentissage.

Il est essentiel de souligner que ces étapes sont cruciales pour optimiser les performances du modèle en termes de précision et de capacité à généraliser à de nouvelles données.

III.5 Description de la base de données utilisée

La base de données utilisée dans cette étude est la base LFW (Labeled Faces in the Wild). Cette base de données a été spécifiquement créée pour étudier le problème de la reconnaissance de visages dans des conditions non contrôlées. LFW contient plus de 13 000 photos de visages provenant de 5 749 personnes différentes, collectées à partir du Web. Chaque image est associée au nom de la personne qu'elle représente.

III.6 Description du prétraitement de la base de données

Dans cette étape de prétraitement, les opérations suivantes ont été effectuées :

- Conversion des images en niveaux de gris : Les images ont été converties en niveaux de gris. Cette transformation permet de simplifier les données d'entrée en ne conservant qu'une seule composante de couleur par pixel. Cette étape facilite ainsi le traitement et l'apprentissage du modèle en réduisant la complexité des données.

- Normalisation des valeurs des pixels : Les valeurs des pixels ont été normalisées pour être comprises dans la plage de 0 à 1. Cette étape de normalisation permet d'harmoniser l'échelle des valeurs des pixels, favorisant ainsi la convergence de l'algorithme d'apprentissage et facilitant la comparaison des valeurs entre les différents pixels de l'image.

III.7 Résultats et discussion

III.7.1 Simulation de l'algorithme sous Colab

1) Choix du type d'architecture

Dans cette section, nous allons comparer les performances de quatre architectures de réseaux CNN (AlexNet, DenseNet, ResNet et FaceNet) dans le contexte d'une configuration matérielle mobile. Nous allons utiliser plusieurs critères d'évaluation, tels que le temps d'exécution, la précision, l'erreur de précision en fonction du nombre d'époques, ainsi que le nombre de paramètres.

L'objectif de cette comparaison est de déterminer quelle architecture offre les meilleures performances dans un environnement mobile, où les contraintes de temps d'exécution et de ressources matérielles sont importantes. En analysant ces critères, nous serons en mesure d'identifier l'architecture qui présente le meilleur équilibre entre précision et efficacité, ce qui est crucial pour une implémentation réussie dans un contexte mobile.

Nous utiliserons des techniques de visualisation telles que des graphiques et des tableaux récapitulatifs pour présenter les résultats de manière claire et concise. Cette analyse comparative nous permettra de prendre des décisions éclairées quant au choix de l'architecture la plus adaptée à notre configuration matérielle mobile.

Tableau 1: Comparaison entre les architectures CNN

L'architecture	Temps d'exécution (s)	Nombre de paramètres (M)	La précision	L'erreur de la précision
AlexNet	55.52 s	58,398,484 M	0.2289	10.8602
DenseNet	530.02 s	8,107,604 M	0.1687	17.2889
ResNet	358.25 s	23,628,692 M	0.3253	5.3754
FaceNet	214.41 s	8,114,196 M	0.6867	1.2539

En fonction des données du tableau 1, on observe que FaceNet présente la meilleure précision parmi les architectures étudiées. Par contre, l'architecture AlexNet affiche le temps

d'exécution le plus court. Il est essentiel de trouver un compromis entre la précision et le temps d'exécution de l'architecture, afin de pouvoir l'implémenter efficacement sur la carte Raspberry Pi.

Étude et simulation du modèle de base de FaceNet

Afin d'établir un point de référence concernant le nombre de paramètres et la précision, nous allons utiliser le modèle de base de FaceNet tel qu'introduit dans la référence [26]. Cela nous permettra d'obtenir une idée précise du nombre de paramètres et de la valeur de la précision initiale. Ensuite, nous procéderons à la réduction de l'architecture, dans le but de trouver une configuration plus légère tout en maintenant des performances acceptables. Cette étape vise à optimiser à la fois l'efficacité des calculs et l'utilisation des ressources.

Pour l'entraînement du modèle, nous avons utilisé 50 epochs et un batch size de 64. Ces valeurs ont été choisies pour permettre une convergence adéquate du modèle tout en évitant un temps d'entraînement excessif.

Les résultats de cette expérience obtenus sont résumés dans le tableau 2 :

Tableau 2: Les performances de FaceNet

Model	Temps d'exécution	Précision	Erreur	Nombre de paramètres
FaceNet	214.41 s	0.6867	1.2539	8,114,196 M

Tableau 3: Paramètres du modèle FaceNet

Layer (type)	Output Shape	Parameters
input_1 (InputLayer)	[(None, 250, 250, 3)]	0
conv1 (Conv2D)	(None, 250, 250, 64)	1792
bn1 (BatchNormalization)	(None, 250, 250, 64)	256
re_lu (ReLU)	(None, 250, 250, 64)	0
max_pooling2d (MaxPooling2D)	(None, 125, 125, 64)	0
conv2 (Conv2D)	(None, 125, 125, 128)	73856

bn2 (BatchNormalization)	(None, 125, 125, 128)	512
re_lu_1 (ReLU)	(None, 125, 125, 128)	0
max_pooling2d_1 (MaxPooling 2D)	(None, 63, 63, 128)	0
conv3 (Conv2D)	(None, 63, 63, 256)	295168
bn3 (BatchNormalization)	(None, 63, 63, 256)	1024
re_lu_2 (ReLU)	(None, 63, 63, 256)	0
max_pooling2d_2 (MaxPooling 2D)	(None, 32, 32, 256)	0
conv4 (Conv2D)	(None, 32, 32, 512)	1180160
bn4 (BatchNormalization)	(None, 32, 32, 512)	2048
re_lu_3 (ReLU)	(None, 32, 32, 512)	0
max_pooling2d_3 (MaxPooling 2D)	(None, 16, 16, 512)	0
conv5 (Conv2D)	(None, 16, 16, 512)	2359808
bn5 (BatchNormalization)	(None, 16, 16, 512)	2048
re_lu_4 (ReLU)	(None, 16, 16, 512)	0
max_pooling2d_4 (MaxPooling 2D)	(None, 8, 8, 512)	0
flatten (Flatten)	(None, 32768)	0
fc1 (Dense)	(None, 128)	4194432
bn6 (BatchNormalization)	(None, 128)	512
re_lu_5 (ReLU)	(None, 128)	0
embedding (Lambda)	(None, 128)	0
classifier (Dense)	(None, 20)	2580

Total params: 8,114,196

Trainable params: 8,110,996

Non-trainable params: 3,200

2) Influence du nombre de couches sur la précision

Dans cette expérience, nous étudions l'impact du nombre de couches du bloc Pooling sur la précision du système. Nous avons réalisé plusieurs tests en modifiant le nombre de couches, allant de 4 à 6, dans différentes architectures du modèle. Les résultats obtenus sont présentés dans le tableau 4.

Tableau 4: Les performances du système en fonction de nombres de couches

Nombre de couches convolutives	Nombre de paramètres	Précision	Erreur	Temps d'exécution
6	9,918,868 M	0.1143	2.9930	214.36 s
5	8,114,196 M	0.6265	1.6053	182.17 s
4	4,459,220 M	0.7229	1.2469	59.18 s

D'après les résultats du tableau 4, nous constatons que le nombre de couches a un impact significatif sur la précision du système. Plus précisément, nous n'observons qu'un nombre de couches égal à 4 conduits à de meilleurs résultats.

3) Influence du nombre de classes sur la précision

Dans cette section, nous examinons l'effet du nombre de classes sur les performances du réseau. Notre objectif est de déterminer le nombre optimal de classes pour obtenir les meilleurs résultats. À cet effet, nous avons mené plusieurs expériences en faisant varier le nombre de classes de 20 à 52. Nous avons mesuré l'exactitude du réseau pour chaque nombre de classes, et les résultats sont présentés dans le tableau 5.

Tableau 5: Les performances du système en fonction de nombres de classes

Nombre de classe	Nombre de paramètres	Précision	Erreur	Temps d'exécution
52	8,118,324 M	0.8752	0.6679	1359.27 s
39	8,116,647 M	0.9168	0.3889	1242.47s
20	8,114,196 M	0.6286	1.7017	144.50 s

Nous avons constaté que l'exactitude du réseau augmente avec le nombre de classes jusqu'à un certain seuil, puis diminue. Le nombre optimal de classes est donc celui qui maximise l'exactitude du réseau. Dans notre cas, on voit que 39 classes est le nombre idéal pour de bons résultats, mais son temps d'exécution est très grand. Donc nous avons déterminé que le nombre de classes optimal est de 20, ce qui conduit aux meilleurs résultats en un minimum de temps.

4) Influence du nombre d'Epochs et le batch_size sur les performances de l'algorithme de reconnaissance

Le nombre d'epochs et le batch size jouent un rôle très important dans la détermination de l'exactitude de l'algorithme de reconnaissance. Il est donc essentiel de trouver un compromis entre le choix de ces deux paramètres d'une part, et l'exactitude et le temps d'exécution d'autre part. Dans cette section, nous allons réaliser une expérience afin d'étudier l'évolution des performances de FaceNet en fonction du nombre d'epochs et du batch size. Le tableau 6 résume les résultats obtenus en termes de précision et de temps d'exécution.

Tableau 6: les performances du système en fonction de nombres d'epochs et batch_size

Nombre d'epochs et batch_size	Nombre de paramètres	Précision	Erreur	Temps d'exécution
80/128	8,114,196 M	0.4699	1.8795	262.43 s
50/64	8,114,196 M	0.6265	1.6053	182.17 s
10/32	8,114,196 M	0.4940	2.0160	41.91 s

D'après les résultats du tableau 6, nous constatons que le couple Epoch/Batch size égal à 50 / 64 donne les meilleurs paramètres pour le système afin de bien l'implémenter sur la carte Raspberry.

Le tableau 7 présente les paramètres obtenus pour un système performant.

Tableau 7: Meilleurs paramètres pour le système à implémenter sur la carte Raspberry

Modèle	Nombres de couches	Nombre de classe	Nombred'epochs / batch_size
FaceNet	4	20	50 / 64

D'après les résultats des différentes expériences précédentes, nous avons sélectionné les performances résumées dans le tableau 8 pour la partie d'implémentation sur la carte Raspberry. La Figure 15 représente la matrice de confusion de la configuration que nous allons mettre en œuvre. Nous pouvons constater, d'après cette figure, notamment par la diagonale, que la majorité des classes sont correctement classifiées.

Tableau 8: Les performances du système à implémenter sur Raspberry

Modèle	Temps d'exécution	Précision	Erreur
FaceNet	142.46 s	0.7229	1.2469

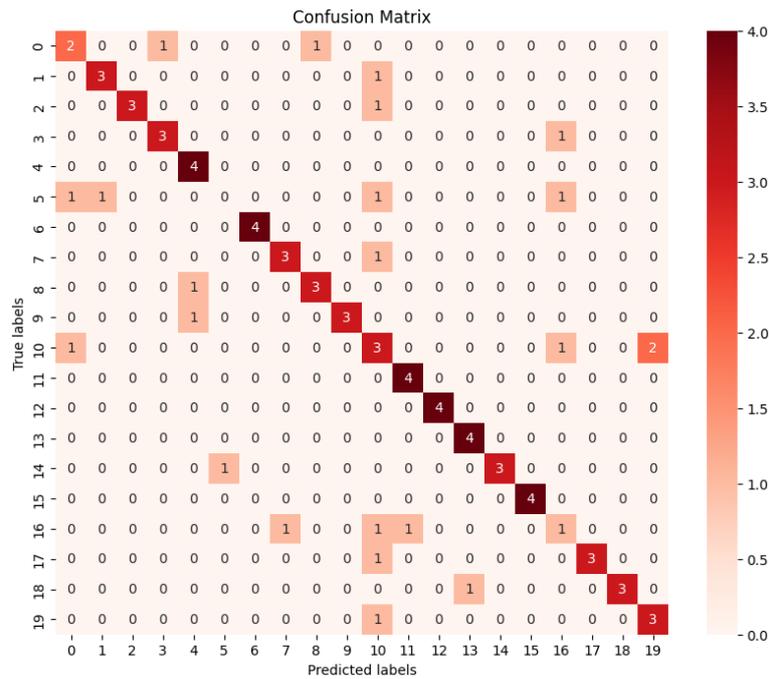


Figure 15: Matrice de confusion de FaceNet

III.7.2 Implémentation de l'algorithme de reconnaissance faciale

Pour la partie hardware nous avons utilisé le matériel suivant :

1) Carte Raspberry pi 4 (modèle B)

➤ Systèmed'exploitation de Raspberry Pi

Dans le cadre de ce projet sur le Raspberry Pi, nous avons choisi Raspbian comme système d'exploitation GNU/Linux. Raspbian est spécifiquement conçu et optimisé pour les appareils à carte unique comme le Raspberry Pi.

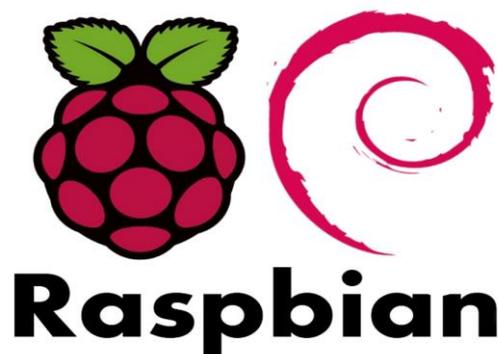


Figure 16: Logo Raspbian

Pour récapituler le processus d'installation et de configuration de Raspbian sur le Raspberry Pi 4, nous avons suivi les étapes suivantes :

2) Téléchargement du système Raspbian et transfert sur la carte SD :

- Raspbian est le système d'exploitation officiel pour les Raspberry Pi. Nous l'avons téléchargé depuis le site officiel de la fondation Raspberry Pi [27].
- Tout d'abord, nous avons téléchargé et installé Raspberry Pi Imager sur notre ordinateur.
- Ensuite, nous avons inséré la carte microSD dans le lecteur de carte SD de notre ordinateur.
- Nous avons lancé Raspberry Pi Imager et sélectionné l'option d'installation de Raspberry Pi OS.
- Le logiciel s'est chargé de télécharger l'image du système d'exploitation et de l'installer sur la carte microSD.
- Une fois l'installation terminée, nous avons retiré la carte microSD de l'ordinateur et l'avons insérée dans le Raspberry Pi.
- La figure 17 montre la fenêtre de Raspberry Pi Imager utilisée pour l'installation du système d'exploitation.

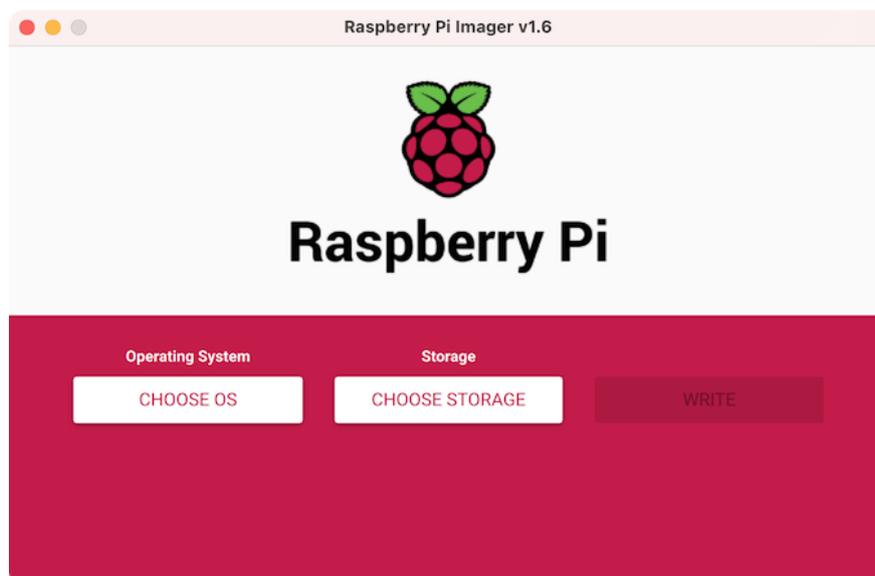


Figure 17: Fenêtre du Raspberry Pi imager

3) Configuration initiale et démarrage :

Après avoir préparé la carte SD avec le système d'exploitation Raspbian en suivant les étapes précédentes, nous avons effectué la configuration initiale et procédé au démarrage du Raspberry Pi 4. Voici les étapes que nous avons suivies :

- Insertion de la carte SD dans le Raspberry Pi :

Une fois la carte SD préparée, nous l'avons retirée de notre ordinateur et insérée dans le slot dédié sur notre Raspberry Pi 4.

- Connexion des périphériques nécessaires :

Nous avons connecté les périphériques essentiels au Raspberry Pi, tels qu'un clavier, une souris, un écran et une alimentation. Ces composants sont nécessaires pour la configuration initiale du Raspberry Pi et pour pouvoir l'utiliser de manière autonome.

- Utilisation du kit complet :

La figure 18 présente le kit complet que nous avons utilisé, comprenant le Raspberry Pi 4, le clavier, la souris, l'écran et l'alimentation.



Figure 188: Kit complet du Raspberry Pi4

En suivant ces étapes, nous avons réussi à préparer notre Raspberry Pi avec le système d'exploitation Raspbian, à effectuer la configuration initiale et à le démarrer correctement. Cette configuration initiale nous permet de tirer pleinement parti des fonctionnalités offertes

par le Raspberry Pi et d'utiliser l'ensemble du kit de manière optimale. Nous avons ainsi pu commencer à exploiter les capacités de notre Raspberry Pi 4 avec Raspbian.

- Une fois le Raspberry Pi alimenté, le système d'exploitation s'est automatiquement lancé. La figure 19 illustre le processus de démarrage du système, confirmant ainsi que le Raspberry Pi fonctionne correctement avec Raspbian installé.



Figure 19: Exécution du système pour la première fois

4) Mise à jour du système :

Afin de préparer l'environnement de programmation sur Raspberry Pi 4, nous avons effectué les opérations suivantes :

La figure 20 montre l'interface du terminal sur Raspberry Pi.

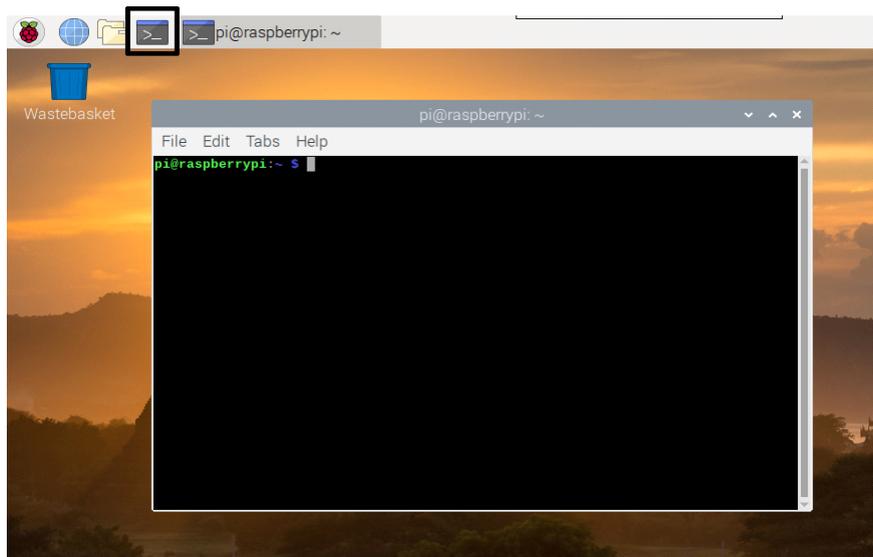


Figure 19: Interface terminale

- Nous avons mis à jour le système en ouvrant le Terminal sur Raspberry Pi, et en exécutant les commandes

```
'sudo apt update'
```

```
'sudo apt upgrade'
```

Ensuite nous avons installé Python, Open CV et Tensorflow.

- **Python** : On vérifie d'abord si Python était déjà installé avec la commande

```
'Python3 --version'
```

Si ce n'était pas le cas, il faut l'installer avec la commande

```
'sudo apt install python3'
```

- **OpenCV** : on a installé d'abord les dépendances d'Open CV avec les commandes

```
'sudo apt install libatlas-base-dev libjasper-dev libqtgui4 libqt4-test'
```

```
'sudo apt install libhdf5-dev libhdf5-serial-dev libhdf5-103'
```

```
sudo apt install libatlas-base-dev libjpeg-dev libpng-dev libtiff-dev'
```

Puis installé OpenCV avec la commande

```
'pip3 install opencv-python'
```

- **TensorFlow** : pour tensorflow nous avons utilisé seulement la commande

```
'pip3 install tensorflow'
```

5) Résultat d'apprentissage sur Raspberry Pi 4

Après avoir installé le système, la prochaine étape consiste à installer les bibliothèques nécessaires, puis à démarrer l'entraînement du modèle. Nous avons rapidement constaté que l'espace de stockage du Raspberry Pi était limité. Par conséquent, nous avons décidé d'effectuer l'apprentissage avec une base de données contenant uniquement 13 classes, en utilisant 50 époques et une taille de lot (batch size) de 64. Ces paramètres ont été choisis après avoir effectué plusieurs tests pour évaluer les limites de stockage du Raspberry Pi 4. Nous avons commencé avec 5 classes, puis augmenté à 10 classes, avant de constater une saturation

de la mémoire RAM avec 15 classes. Nous avons donc réduit le nombre de classes à 13, ce qui s'est avéré être un choix optimal. Nous avons procédé de la même manière pour le nombre d'époques et la taille de lot.

La figure 21 représente les résultats que nous avons obtenus lors de cet apprentissage.

```

face.py x part test.py x
105 # Create the FaceNet model
106 model = face_net(input_shape, num_classes)
107
108 # Print the model summary
109 model.summary()
110
111 # Compile the model
112 model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
113
114 # Define the number of epochs and batch size
115 epochs = 50
116 batch_size = 64
117
118 # Train the model
119 start_time = time.time()
120 history= model.fit(x_train, y_train, epochs=epochs, batch_size=batch_size, validation_data=(x_test, y_test))
121 end_time = time.time()
122 execution_time = end_time - start_time
123 print("execution time: %.2f seconds" % execution_time)
124 # Evaluate the model on the test data using `evaluate`
125 print("Evaluate on test data")
126 results = model.evaluate(x_test, y_test, batch_size=64)
127 print("test loss, test acc:", results)
128
129
130 # Save the trained model
131 model.save('/home/ETA_Laboratory/Documents')
132
133
134
Shell
Epoch 50/50
11/11 [=====] - 492s 45s/step - loss: 0.2156 - accuracy: 1.0000 - val_loss: 0.5284 - val
execution time: 24382.25 seconds
Evaluate on test data
3/3 [=====] - 29s 9s/step - loss: 0.5284 - accuracy: 0.8696
test loss, test acc: [0.5283728837966919, 0.8695651888847351]

```

Figure 20: Résultats d'apprentissage sur Raspberry Pi

6) Examiner les résultats d'apprentissage:

Les résultats d'exécution sont regroupés dans le tableau 9

Tableau 9: Les résultats d'exécution sous Python

Temps d'exécution	Précision	Erreur
24382.25 secondes (6h)	0,86	0,52

Ces résultats démontrent que notre modèle est capable de reconnaître les visages avec une précision satisfaisante de 86%, malgré les limitations du Raspberry Pi 4 en termes de puissance de calcul et de stockage. Nous avons réussi à entraîner notre modèle en 6 heures, ce qui est raisonnable pour un système embarqué. De plus, la perte est relativement faible, ce qui indique que le modèle n'est pas en surapprentissage ni en sous-apprentissage.

7) Partie Tests

Après la phase d'apprentissage de notre modèle, le test sur une image se déroule selon les étapes suivantes :

- Chargement de l'image de test à partir de son emplacement spécifié.
- Conversion de l'image en un format compatible avec le modèle.
- Normalisation des valeurs des pixels pour les ajuster à l'intervalle attendu par le modèle.
- Ajout d'une dimension supplémentaire à l'image pour correspondre à la dimension de lot (batch) attendue par le modèle.
- Passage de l'image au modèle pour obtenir une prédiction.
- Affichage du résultat de la prédiction, qui, dans notre cas, correspond au nom et au prénom de la personne reconnue.

Ces étapes permettent de tester le modèle en lui fournissant une image de test et en obtenant une prédiction basée sur ce qu'il a appris lors de l'entraînement. Les figures 22, 23 et 24 représentent les résultats des tests pour trois personnes différentes.

```

69
70 # Load the image from its path
71 img = tf.keras.preprocessing.image.load_img('/home/ETA_Laboratory/Documents/test 1.jpg')
72 # Convert the image to a NumPy array
73 img = tf.keras.preprocessing.image.img_to_array(img)
74
75 # Normalize the pixel values
76 img = img / 255.0
77
78 # Add a batch dimension
79 img = np.expand_dims(img, axis=0)
80
81 # Generate prediction on the image
82 pred = model.predict(img)
83
84 # Get the predicted label
85 pred_label = np.argmax(pred, axis=-1)
86
87 predicted_noun = label_encoder.inverse_transform(pred_label)
88 # Print the predicted label
89 print("Predicted label:", predicted_noun)
90
91 end_time = time.time()
92 execution_time = end_time - start_time
93 print("execution time: %.2f seconds" % execution_time)
94

```



```

Shell
>>> %Run 'part test.py'
1/1 [=====] - 1s 694ms/step
Predicted label: ['Ann_Veneman']
execution time: 0.88 seconds

```

Figure 21: Test 1 Identification réussite

```

68 start_time = time.time()
69
70 # Load the image from its path
71 img = tf.keras.preprocessing.image.load_img('/home/ETA_Laboratory/Documents/test 13.jpg')
72 # Convert the image to a NumPy array
73 img = tf.keras.preprocessing.image.img_to_array(img)
74
75 # Normalize the pixel values
76 img = img / 255.0
77
78 # Add a batch dimension
79 img = np.expand_dims(img, axis=0)
80
81 # Generate prediction on the image
82 pred = model.predict(img)
83
84 # Get the predicted label
85 pred_label = np.argmax(pred, axis=-1)
86
87 predicted_noun = label_encoder.inverse_transform(pred_label)
88 # Print the predicted label
89 print("Predicted label:", predicted_noun)
90
91 end_time = time.time()

```



```

Shell
>>> %Run 'part test.py'
1/1 [=====] - 1s 1s/step
Predicted label: ['Jack_Straw']
execution time: 1.25 seconds

```

Figure 22: Test 2 Identification réussite

```

68 start_time = time.time()
69
70 # Load the image from its path
71 img = tf.keras.preprocessing.image.load_img('/home/ETA_Laboratory/Documents/test 7.jpg')
72 # Convert the image to a NumPy array
73 img = tf.keras.preprocessing.image.img_to_array(img)
74
75 # Normalize the pixel values
76 img = img / 255.0
77
78 # Add a batch dimension
79 img = np.expand_dims(img, axis=0)
80
81 # Generate prediction on the image
82 pred = model.predict(img)
83
84 # Get the predicted label
85 pred_label = np.argmax(pred, axis=-1)
86
87 predicted_noun = label_encoder.inverse_transform(pred_label)
88 # Print the predicted label
89 print("Predicted label:", predicted_noun)
90
91 end_time = time.time()

```



```

Shell
>>> %Run 'part test.py'
1/1 [=====] - 1s 695ms/step
Predicted label: ['George_W_Bush']
execution time: 1.25 seconds

```

Figure 23: Test 3 Identification non réussite

8) Examiner les résultats du test :

Pour évaluer les performances de notre système, nous avons effectué 13 tests correspondant aux 13 classes de notre base de données. Notre système a réussi à reconnaître correctement 10 personnes sur 13, ce qui correspond à une précision de 77%. Les trois erreurs commises par notre système sont attribuables à des facteurs tels que la qualité de l'image, l'angle de vue ou la ressemblance entre les personnes.

Nous avons analysé ces erreurs et identifié des pistes d'amélioration pour notre système, telles que l'ajout d'une étape de prétraitement des images ou l'optimisation des paramètres du modèle.

Le temps d'exécution approximatif pour notre modèle lors du test sur le Raspberry Pi 4 est d'environ 1 seconde. C'est un temps assez rapide qui démontre que le modèle peut traiter l'image de test et générer une prédiction en un temps très court. Cela est particulièrement utile pour une utilisation en temps réel ou dans des applications telles que le déverrouillage d'une porte par reconnaissance faciale pour l'accès à un immeuble ou une zone sécurisée. Il convient de noter que le temps d'exécution peut varier en fonction de la taille de l'image de test et d'autres facteurs tels que la charge du système.

III.8 Conclusion

Dans ce chapitre, nous avons présenté les résultats et les tests de simulation de notre algorithme de reconnaissance faciale basé sur les réseaux CNN sur deux plateformes : Google Colab et Raspberry Pi 4. L'objectif était de comparer les performances de notre système avec celles d'autres architectures de réseaux CNN, et d'implémenter notre algorithme sur une carte Raspberry Pi 4, pour évaluer son fonctionnement en termes de précision, de temps d'exécution et de consommation de ressources.

Les principaux résultats obtenus dans ce chapitre sont les suivants :

- Notre algorithme a atteint une précision remarquable de 86%.
- Notre algorithme a démontré sa capacité à reconnaître des visages en temps réel sur le Raspberry Pi, avec un temps d'exécution moyen très court par image.

Ces résultats indiquent que ce système est capable de répondre aux exigences de performances attendues pour des applications de reconnaissance faciale. Il peut être appliqué dans divers domaines tels que la sécurité, la biométrie, le divertissement, et bien d'autres.

Cependant, ce système présente également quelques limites et des pistes d'amélioration possibles. Premièrement, il dépend fortement de la qualité des données utilisées pour l'entraînement et le test du modèle. Deuxième, le nombre de classes traité est limité par les ressources matérielles disponibles.

Les résultats obtenus ont confirmé l'efficacité de notre approche et ouvrent la voie à de futures améliorations et applications dans ce domaine en constante évolution.

Conclusion générale

Ce mémoire de Master a été consacré au développement d'un système de reconnaissance faciale basé sur les réseaux de neurones convolutifs (CNN) et implémenté sur la carte Raspberry Pi 4. Le travail s'est déroulé en deux étapes clairement définies.

Dans la première étape, nous avons comparé différentes architectures CNN et avons sélectionné FaceNet comme l'architecture la plus appropriée pour notre étude. Par la suite, le système a été soumis à des simulations et à des exécutions dans l'environnement de développement Google Colab. Cette phase nous a permis de tester et d'optimiser le modèle en évaluant ses performances et en apportant les ajustements nécessaires. Grâce à ces efforts, nous avons réussi à réduire le nombre de paramètres tout en maintenant une précision élevée et un temps d'exécution court.

La deuxième étape a consisté à mettre en pratique le travail en implémentant le modèle sur la carte Raspberry Pi 4. Nous avons confirmé le bon fonctionnement du système en testant des images préalablement entraînées, ainsi que des images externes provenant de notre base de données. Les résultats obtenus ont démontré l'efficacité de notre approche, avec une remarquable précision de 86% dans la reconnaissance en temps réel des visages.

Le système de reconnaissance faciale développé ouvre des perspectives d'application dans divers domaines tels que la sécurité, la biométrie et le divertissement. Cependant, il reste encore des possibilités d'amélioration et de développement pour ce projet. Parmi les pistes à explorer, on peut citer :

- L'amélioration de la qualité des données utilisées pour l'entraînement et les tests du modèle.
- L'adaptation du système à d'autres supports matériels tels que les smartphones ou les caméras embarquées.
- L'intégration d'autres fonctionnalités telles que la détection des émotions ou des expressions faciales.
- Le développement d'une interface graphique conviviale pour faciliter l'utilisation du système par les utilisateurs.

En conclusion, cette étude a réussi de développer un système de reconnaissance faciale performant, et adapté aux contraintes de la carte Raspberry Pi, ouvrant ainsi de nouvelles perspectives pour l'utilisation de cette technologie dans des applications du monde réel. Ce projet nous a également permis d'acquérir de solides compétences en programmation, en manipulation des bibliothèques OpenCV et TensorFlow, ainsi qu'en entraînement de réseaux de

Conclusion générale

neurones, tout en nous familiarisant avec le domaine de l'intelligence artificielle et de l'apprentissage automatique.

Bibliographie

- [1] LAOUADI, H. et SAADI, L., 2022. La reconnaissance faciale. Mémoire de master. Université Larbi Ben M'hidi-Oum El Bouaghi.
- [2] BOUKERROUCHE, Y. et ZERRIOUH, A., 2018. Mémoire de master en informatique. Université AboubekrBelkaid de Tlemcen.
- [3] MORIZET, N., 2009. Reconnaissance Biométrique par fusion multimodale du visage et de l'iris. Thèse de doctorat en signal et images. École doctorale d'informatique, télécommunications et électronique de Paris.
- [4] RABEARIVELO, M.D., 2022. Reconnaissance automatique de visages. Mémoire de master. Université d'Antananarivo.
- [5] BEN MOUFFOK, M.C. et SAYAH, A.R., 2022. Mémoire de master en télécommunication. Université Mohamed Khider de Biskra.
- [6] MEKKANI, S., 2014. Reconnaissance de visage. Mémoire de master en informatique. Université Larbi Ben M'hidi-Oum El Bouaghi.
- [7] OUAMANE, H., 2012. Identification de reconnaissance faciale avec des expressions. Mémoire de master en télécommunication. Université Mohamed Khider Biskra.
- [8] SELLAMI, Z., 2019. Optimisation du CNN par l'algorithme génétique pour la reconnaissance de visage. Thèse de master en réseaux et télécommunication. Université Mohamed Khider de Biskra.
- [9] ENCYCLOPÉDIE UNIVERSALIS, 2021. Intelligence artificielle (IA), Évolution de l'intelligence artificielle [en ligne]. Disponible sur : <https://www.universalis.fr/encyclopedie/intelligence-artificielle-ia-evolution-de-l-intelligence-artificielle/> [Consulté le 13/05/2023].
- [10] BOIFALTINGS, B. et SCHUMACHER, M., 2009. L'intelligence artificielle par la pratique. Paris : Polytechniques.
- [11] Le guide complet de la reconnaissance faciale - intelligence artificielle [en ligne]. Disponible sur : <https://www.appvizer.fr/magazine/operations/securite/reconnaissance-faciale> [Consulté le 13/05/2023].

- [12] Reconnaissance faciale : Identification avec IA - Digital Cook [en ligne]. Disponible sur : <<https://digital-cook.fr/reconnaissance-faciale-identification-avec-ia/>> [Consulté le 13/05/2023].
- [13] Machine Learning : tout savoir [en ligne]. Disponible sur : <<https://datascientest.com/machine-learning-tout-savoir>> [Consulté le 20/05/2023].
- [14] Définition | Deep Learning - Apprentissage profond | Futura Tech [en ligne]. Disponible sur : <<https://www.futura-sciences.com/tech/definitions/informatique-deep-learning-17060/>> [Consulté le 20/05/2023].
- [15] BEN LAZREG, M., 2020. Recherche de l'information dans les réseaux de neurones convolutifs pré-entraînés. Thèse de maîtrise en génie logiciel et des TI. École de technologie supérieure Université du Québec.
- [16] BLANC-DURAND, P., 2018. Réseaux de neurones convolutifs en médecine nucléaire : applications à la segmentation automatique des tumeurs gliales et à la correction d'atténuation en TEP/IRM. Thèse de docteur en médecine. Université Paris Descartes.
- [17] BELHADJER, H. et SAROUER, B., 2018. Classification des images avec les réseaux de neurones convolutionnels. Thèse de master en système informatique. Université Mouloud Mammeri de Tizi-Ouzou.
- [18] SEKKIL, H.M. et MEBROUKI, M., 2021. Etude comparative entre les différentes architectures des réseaux de neurones convolutifs (CNN) pour la détection de la fatigue du conducteur. Thèse de master en automatique. École supérieure en science appliquée Tlemcen.
- [19] HMIDI, A., 2021. Conception d'un système de reconnaissance des images et application des FPGA à l'accélération de l'algorithme d'apprentissage profond. Thèse de doctorat en électronique et micro-électronique. Université de Monastir – Tunisie.
- [20] LAIB, A. et MENNANA, T., 2018. Contribution à l'amélioration d'un distillateur simple pente : prédiction par la méthode des réseaux de neurones artificiels. Thèse de master en énergétique. Université EchahidHamma Lakhdar d'El-Oued.
- [21] CHEN, Y., 2015. Réseaux de neurones convolutifs pour classement des phrases. Thèse de Master en Informatique. Université de Waterloo Ontario Canada.

- [22] Wikidocs, 2023. Réseaux de neurones convolutionnels [en ligne]. Disponible sur : <<https://wikidocs.net/165403>> [Consulté le 04/06/2023].
- [23] FIFALIANARIVONY, N., 2020. Détection et reconnaissance de plaque d'immatriculation des véhicules circulants dans l'enceinte du ministère de la défense nationale en utilisant la machine learning et l'intelligence artificielle. Thèse de master en télécommunication. École supérieure polytechnique.
- [24] FaceNet : utilisation du système de reconnaissance faciale [en ligne]. Disponible sur : <<https://stacklima.com/facenet-utilisation-du-systeme-de-reconnaissance-faciale/>> [Consulté le 14/05/2023].
- [25] BENYELLES, N. et AICHOUBA, A., 2021. Détection et signalisation des absences scolaires par reconnaissance faciale. Thèse de master en automatique. École supérieure en science appliquée Tlemcen.
- [26] SCHROFF, F., KALENICHENKO, D. et PHILBIN, J., 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering [enligne]. Disponible sur : <<https://arxiv.org/pdf/1503.03832.pdf>> [Consulté le 04/06/2023].
- [27] Raspberry Pi Software [en ligne]. Disponible sur : <<https://www.raspberrypi.com/software/>> [Consulté le : 01/06/2023].

Annexe

a) Python :

Python est un langage polyvalent utilisé dans l'intelligence artificielle (IA) et l'apprentissage profond. Il permet de développer des applications IA en utilisant des réseaux de neurones convolutionnels (CNN) pour analyser de grandes quantités de données. Python offre des bibliothèques spécialisées telles que TensorFlow et Keras qui facilitent l'implémentation de CNN. Avec Python, les développeurs peuvent entraîner, valider et déployer des modèles de CNN, exploitant ainsi les capacités avancées de l'apprentissage profond.

b) Google Colaboratory (Colab):

Colab est un environnement de développement basé sur le « Cloud » qui permet d'exécuter des codes Python directement dans le navigateur, sans nécessiter de configuration ou d'installation.

c) Open CV (Open Source Computer Vision Library) :

C'est une bibliothèque open source qui se spécialise dans le traitement d'images et la vision par ordinateur. Elle offre des fonctionnalités pour capturer et manipuler des images et des vidéos, ainsi que des algorithmes de vision par ordinateur comme la détection d'objets, la reconnaissance faciale, le suivi d'objets, etc.

d) Tensorflow

Tensorflow est une bibliothèque open source qui a été développée par Google, a fin de créer et d'entraîner des modèles d'apprentissage automatique, spécialement des réseaux de neurones profonds. Elle fournit un ensemble d'outils et de bibliothèques pour construire, entraîner, évaluer et déployer de manière efficace ces modèles, par exemple :

- **Numpy** C'est une bibliothèque de calcul numérique en Python. Elle fournit des structures de données de tableau multidimensionnel (ndarray), et des fonctions mathématiques pour manipuler efficacement ces tableaux. NumPy est largement utilisé pour le traitement des données, les calculs scientifiques et l'analyse de données en Python.
- **Keras** est une bibliothèque de haut niveau pour les réseaux de neurones écrite en Python. Elle fournit une interface facile à utiliser pour créer et entraîner des modèles d'apprentissage profond. Keras permet de définir et de configurer facilement différents types de réseaux de neurones, comme les réseaux convolutifs (CNN).

- **Matplotlib** C'est une bibliothèque de traçage en 2D pour Python. Elle permet de créer des graphiques, des diagrammes, des visualisations de données et des tracés scientifiques de haute qualité.

e) Scikit-learn (sklearn)

Sklearn est une bibliothèque de python pour l'apprentissage. Elle offre des outils et des algorithmes pour différentes tâches comme la classification, le regroupement, la sélection de modèles. Scikit-learn s'appuie sur d'autres bibliothèques scientifiques comme NumPy, SciPy et matplotlib, et il est facile et rapide à utiliser.