

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي



جامعة سعيدة د. مولاي الطاهر

كلية الرياضيات و الإعلام الآلي و الاتصالات السلكية و اللاسلكية

قسم: الإعلام الآلي

Mémoire de Master en informatique

Spécialité : Réseaux informatiques et systèmes distribués

T h è m e

Système de Reconnaissance Faciale Basé sur le Cloud Computing

■ Présenté par :

RAHIL Nacera

DOUAR Yousra

■ Dirigé par :

DIB Soumai

Année universitaire



2025-2026

ملخص

يهدف هذا المشروع إلى تطوير نظام ذكي للتعرف على الوجوه يعتمد على تقنيات التعلم العميق والحوسبة السحابية. تم تصميم وتنفيذ ثلاث معماريات مختلفة هي السحابة المحلية، السحابة الحقيقية، والنظام الهجين من أجل مقارنة أدائها واختيار مع 'Face Recognition وOpenCV وFlask وPython الحل الأنسب. يعتمد النظام على تقنيات حديثة مثل تخزين البيانات والصور. كما تم اعتماد أسلوب قائم على قياس المسافة Cloudinary وFirebase استخدام لتحسين دقة اتخاذ القرار وتقليل (Thresholds) مع استخدام عتبات (Distance-Based Recognition) لتحسين التمييز بين الوجوه المتشابهة، خاصة Confidence Gap أخطاء التعرف. إضافةً إلى ذلك، تم تطبيق آلية التوائم. وقد أظهرت النتائج أن النظام الهجين يوفر أفضل توازن بين السرعة والمرونة والموثوقية.

Abstract

This project focuses on developing a facial recognition system based on Deep Learning and integrated with Cloud Computing technologies. Three architectures were designed and implemented: Local Cloud, Real Cloud, and Hybrid Cloud. The system uses Python, Flask, OpenCV, Face Recognition, Firebase, and Cloudinary. To improve recognition reliability, a Distance-Based Recognition approach with decision thresholds was adopted. A Confidence Gap mechanism was also implemented to better distinguish highly similar faces, especially twins. Experimental results demonstrate that the Hybrid Cloud architecture provides the best balance between performance, flexibility, and reliability.

Résumé

Ce projet consiste à développer un système de reconnaissance faciale basé sur le Deep Learning et intégré dans un environnement Cloud Computing. Trois architectures ont été étudiées et réalisées : Cloud Local, Cloud Réel et Cloud Hybride. Le système utilise Python, Flask, OpenCV, Face Recognition, Firebase et Cloudinary. Afin d'améliorer la fiabilité de la reconnaissance, une approche basée sur la distance (Distance-Based Recognition) a été adoptée avec l'utilisation de seuils de décision (Thresholds). Un mécanisme Confidence Gap a également été implémenté pour mieux distinguer les visages très similaires, notamment les jumeaux. Les résultats obtenus montrent que l'architecture hybride offre le meilleur compromis entre performance, flexibilité et fiabilité.



REMERCIEMENT

À la source de l'amour et de la tendresse,
au battement de mon cœur et à mes
prières constantes ; ma mère bien-aimée...

À la source de la force et du soutien, et à
celui sous l'ombre duquel j'ai eu
l'honneur de cheminer ; mon cher père...

À mes frères, piliers de ma vie, et à ma
chère sœur, chacun par son nom et sa
haute valeur, et leur place précieuse dans
mon cœur, les âmes sœurs et mon
trésor dans ce monde...

À ma première inspiration, et la raison
de ma persévérance et de ma lutte pour
être une fierté à suivre ; mon cher fils
« Yacine »...

Aux fleurs de ma vie et au sourire de
mon monde : mes deux belles petites
« Anfal » et « Israa »...

À toute la famille, aux amis et aux
proches, qui nous ont comblés de leurs
prières sincères, soutenus par leurs
encouragements **bienveillants** et
poussés vers le sommet...

Je vous dédie à tous le fruit de ce succès,
car vous êtes les partenaires du chemin
et avec vous la joie est complète.

RAHIL
Nacera

إِهْدَاء

إلى منبع الحب والحنان، ونبض قلبي ودعائي
الدائم: أمي الحبيبة...

إلى منبع القوة والسند، ومن تشرفت
بمسيرتي تحت ظله: أبي الغالي...

إلى إخوتي سند حياتي وأختي الغالية،
كلٌّ باسمه وقدره الرفيع ومقامه المحفوظ
في القلب، شقائق الروح وذخري في
الدنيا...

إلى ملهمي الأول، وسبب استمرار
وكفاحي لأكون فخراً يقتدي به؛ ابني
الغالي "ياسين"...

إلى زهرتي حياتي وبسمة دُنْيَاي:
صغيرتي الجميلتين "أنفال" و "إسراء"...

إلى كافة الأهل والأصدقاء والأحبة، من
غمرونا بصدق دعواتهم، ودعمونا بطيب
تشجيعهم ودفعوا بنا نحو القمة...

أهديكم جميعاً ثمرة هذا النجاح، فأنتم
شركاء المسيرة وبكم بكتمل الفرح.

رَحِيلُ نَصِيرَةٍ





2026
class of

Je dédie ce modeste travail :

À mes très chers parents, pour leur amour inconditionnel, leurs sacrifices, leurs encouragements et leur soutien permanent tout au long de mon parcours académique. Que Dieu les protège et les récompense pour tout ce qu'ils ont fait pour moi.

À mes frères Lotfi, Ali et Ibrahim, pour leur affection, leur présence et leurs encouragements constants.

À toute ma famille, qui a toujours cru en moi et m'a soutenue dans les moments les plus difficiles.

À mes amis et amies et collègues d'études, avec qui j'ai partagé de précieux moments de travail, d'entraide et de réussite.

À tous mes enseignants, qui ont contribué à ma formation et à mon développement académique par leurs connaissances et leurs conseils.

Enfin, à toute personne ayant participé de près ou de loin à la réalisation de ce mémoire.



Félicitations
YOUSRA DOUAR



Table des Matières

Introduction générale:	7
Chapitre 1: Reconnaissance faciale	1
1.Introduction :	1
1.1 Définition de la reconnaissance faciale :	3
1.2 Historique :	5
1.3 Fonctionnement et analyse des systèmes de reconnaissance faciale :	7
1.3.1 Détection des visages :	9
1.3.2. Analyse et extraction des caractéristiques :	9
1.3.3. Conversion en représentation numérique :	10
1.3.4. Comparaison et identification :	10
1.4 Types de systèmes biométriques :	10
1.5 Techniques classiques (Eigenfaces , Fisherfac , BPH) :	12
1.6 Approches modernes (Deep Learning, CNN) :	14
1.7 Avantages et inconvénients de la reconnaissance faciale :	15
1.8 Domaines de la reconnaissance faciale :	16
Chapitre 2 : Le Cloud Computing	18
1 Introduction:	18
1.1 Définition du Cloud Computing :	20
1.2 Historique :	21
1.3 Les Caractéristiques Du Cloud Computing :	23
1. Service à la demande (On-Demand Self-Service) :	24
2. Large accès réseau (Broad Network Access) :	24
3. Mutualisation des ressources (Resource Pooling) :	25
4. Élasticité rapide (Rapid Elasticity) :	25
5. Service mesuré (Measured Service) :	25
1.4 Types De Cloud Computing :	26
1. Cloud Public :	27
2. Cloud Privé :	27
3. Cloud Hybride :	28
4. Cloud Communautaire :	28
5. Multi-Cloud :	28
1.5 Types de services de Cloud Computing :	29
1. Infrastructure as a Service (IaaS) :	30
2. Platform as a Service (PaaS) :	30
3. Software as a Service (SaaS) :	31
4. Function as a Service (FaaS) / Serverless :	32
1.6 Avantages du Cloud Computing :	32
1. Réduction des coûts :	33

2. Scalabilité et flexibilité :	33
3. Accessibilité et mobilité :	34
4. Stockage et sauvegarde des données :	34
5. Maintenance simplifiée :	34
1.7 Limites et défis du Cloud Computing :	35
1.7.1 Sécurité et confidentialité des données :	35
1.7.2 Dépendance à la connexion Internet :	35
1.7.3 Latence et performance :	35
1.7.4 Dépendance au fournisseur (Vendor Lock-in) :	36
1.7.5 Conformité réglementaire :	36
1.8 Architecture du Cloud Computing :	36
1.8.1 Définition de l'architecture cloud :	37
1.8.2 Composants de l'architecture cloud :	38
1.8.2.1 Frontend (interface utilisateur) :	38
1.8.2.2 Backend (infrastructure cloud) :	38
Chapitre3 : Conception, Implémentation et Évaluation du Système de Reconnaissance Faciale Basé sur le Cloud Computing	42
1. Introduction:	42
1.2 Problématique:	43
1.3 Système de reconnaissance faciale dans un cloud local:	45
1.3.1 Architecture du système:	46
1.3.1.1 Le serveur Cloud (Machine virtuelle - VM):	47
1.3.1.2 Interface d'administration (Web - Windows):	47
1.3.1.3 Client de capture (Système local):	48
1.3.2 Outils et Technologies Utilisés:	46
1.3.2.1 Python:	46
1.3.2.2 OpenCV:	46
1.3.2.3 Bibliothèque Face Recognition:	47
1.3.2.4 Flask:	48
1.3.2.5 Machine Virtuelle (Virtual Machine):	48
1.3.2.6 HTML, CSS et JavaScript:	49
1.4 Conception du Système:	49
1.4.1 Module de Détection des Visages:	49
1.4.2 Module d'Extraction des Caractéristiques:	50
1.4.3 Module de Reconnaissance Faciale:	50
1.4.4 Module de Traitement Cloud:	51
1.5 Implémentation du Système:	51
1.5.1 Gestion des données:	54
1.6 Workflow du Système:	54
1.7 Diagramme UML (class + use case):	55

1.8 Conclusion:	56
2.Système de reconnaissance faciale dans un cloud réel:	56
2.1 Introduction:	57
2.2 Architecture du système:	58
2.2.1 Aperçu général de l'architecture:	58
2.2.2 Couche client:	59
2.2.3 Couche de communication:	60
2.2.4 Couche de traitement cloud:	61
2.2.5 Couche de stockage:	62
2.3 Outils et technologies utilisés:	65
2.3.1 Langage de programmation Python:	65
2.3.2 Bibliothèque OpenCV:	66
2.3.3 Bibliothèque Face Recognition:	66
2.3.4 Framework Flask:	67
2.3.5 Plateforme PythonAnywhere:	68
2.3.6 Bibliothèque Pillow:	68
2.3.7 Format JSON:	69
2.4 Conception du système:	70
2.4.1 Module d'authentification:	70
2.4.2 Module de gestion des visages connus:	71
2.4.3 Module de détection des visages:	71
2.4.4 Module d'extraction des caractéristiques:	71
2.4.5 Module de reconnaissance faciale:	72
2.4.6 Amélioration de la précision du système en cas de visages similaires et de jumeaux:	72
2.5 Mise en œuvre du système:	74
2.6 Flux de travail du système:	75
2.7 Les Diagrammes de systeme:	75
2.8 Conclusion:	76
.3Système hybride:	77
3.1 Introduction	77
3.2 Architecture Générale du Système:	79
3.2.1 Vue d'Ensemble:	79
3.2.2 Couche Présentation (Frontend):	80
3.2.3 Couche Traitement (Backend):	80
3.2.4 Couche Stockage Cloud:	81
3.2.5 Architecture Hybride:	82
3.3 Fonctionnement Détaillé du Système:	82
3.3.1 Authentification de l'Administrateur:	83
3.3.2 Ajout d'une Nouvelle Personne:	83

3.3.3 Stockage des Images dans Cloudinary:	84
3.3.4 Enregistrement des Métadonnées dans Firestore:	84
3.3.5 Encodage des Visages:	84
3.3.6 Acquisition d'une Image Inconnue:	85
3.3.7 Comparaison et Reconnaissance	85
3.3.8 Enregistrement de la Présence:	86
3.4 Technologies Utilisées:	86
3.4.1 Python:	87
3.4.2 Flask:	87
3.4.3 Face Recognition (Dlib):	88
3.4.4 OpenCV:	89
3.4.5 Firebase Firestore:	89
3.4.6 Cloudinary:	90
3.4.7 NumPy:	92
3.4.8 HTML5:	92
3.4.9 Bootstrap:	93
3.4.10 JavaScript:	93
3.4.11 Cloud Local:	94
3.4.12 Architecture Hybride Cloud:	94
3.5 Diagrammes UML :	95
3.5.1 Diagramme de cas d'utilisation :	95
3.5.2 Diagramme de classes:	96
3.5.3 Diagramme de séquence:	96
3.6 Réalisation et captures d'écran :	97
3.6.1 Interface Dashboard :	97
3.6.2 Ajout de personne :	98
3.6.3 Reconnaissance faciale :	100
3.6.4 Gestion de l'assiduité :	100
4.Comparaison des approches proposées:	101
5.Tests et Résultats:	103
5.1Captures d'écran:	103
5.1.1Cloud Local:	103
5.1.2Cloud Réel:	107
5.2Temps de réponse:	109
5.3 Résultats obtenus:	111
6.1 Difficultés rencontrées:	114
6.2 Perspectives d'amélioration:	114
7 Conclusion:	114
Conclusion générale:	116
Références	1

Liste des figures

Figure 1 : image de face	5
Figure 2 : Types d'accès biométriques	12
Figure 3 : Les Caractéristiques Du Cloud Computing	24
Figure 4 : types of cloud	27
Figure 5 : Types de services de Cloud Computing	29
Figure 6 :Architecture du Cloud Computing	37
Figure 7 : image Le système de reconnaissance faciale la résidence universitaire Ahmed Medghri	45
Figure 8 : Architecture du système sur le cloud local	49
Figure 9 :Diagramme UML (class + use case)	55
Figure 10 :Architecture du système	64
Figure 11 :Les Diagrammes de systeme	76
Figure 12 :Architecture Générale du Système	79
Figure 13 :capture Firebase Firestore (Cloud Database)	90
Figure 14 :capture de Stockage d'images dans un Cloudinary	91
Figure 15 :capture de Stockage d'images dans un Cloudinary	91
Figure 16 :Diagramme de cas d'utilisation	95
Figure 17 :Diagramme de classes	96
Figure 18 :Diagramme de séquence	96
Figure 19 :capture d'écran de l'interface de connexion utilisateur	97
Figure 20 :capture d'interface dashboard	98
Figure 21 :capture d'écran 1 du téléchargement des images d'une personne	98
Figure 22 :capture d'écran 2 du téléchargement des images d'une personne	99
Figure 23 :capture d'écran d'une personne téléchargeant	99
Figure 24 :capture d'écran de reconnaissance faciale1	100
Figure 25 :capture d'écran d'affichage des logs d'entrée 1	100
Figure 26 :capture d'écran d'affichage des logs d'entrée 2	101
Figure 27 : login cloud local	104
Figure 28 :Dashboard cloud local 1	104
Figure 29 :Dashboard cloud local 1	105
Figure 30 :Ajout Personne cloud local	105
Figure 31 :Reconnaissance cloud local 1	106
Figure 32 :Reconnaissance cloud local 2	106
Figure 33 :login Cloud Réel	107
Figure 34 :Dashboard Cloud Réel 1	108
Figure 35 :Dashboard Cloud Réel 2	108
Figure 36 : Ajout Personne Cloud Réel	108
Figure 37 :Reconnaissance Cloud Réel	109

Figure 38 : Comparision de Tests	110
--	-----

Introduction générale:

Avec l'évolution rapide des technologies de l'intelligence artificielle et du Cloud Computing, les systèmes biométriques sont devenus des solutions incontournables pour l'identification et le contrôle d'accès. Parmi ces technologies, la reconnaissance faciale occupe une place importante grâce à sa capacité à identifier automatiquement les individus à partir de leurs caractéristiques faciales.

Cependant, les systèmes de reconnaissance faciale traditionnels rencontrent encore plusieurs défis, notamment la gestion des données à grande échelle, les performances en temps réel et la difficulté de distinguer des personnes présentant une forte ressemblance faciale, comme les jumeaux.

Dans ce contexte, notre projet vise à concevoir et développer un système intelligent de reconnaissance faciale basé sur les techniques de Deep Learning et intégré dans différents environnements de Cloud

Computing. Trois architectures ont été étudiées : un Cloud Local, un Cloud Réel et une Architecture Hybride combinant les avantages des deux approches.

L'objectif principal est d'améliorer la précision de reconnaissance, d'assurer une gestion centralisée des données et de proposer une solution performante, flexible et évolutive adaptée aux besoins des applications réelles.

Chapitre 1:

Reconnaissance faciale

Chapitre 1: Reconnaissance faciale

1.Introduction :

Avec l'évolution rapide des technologies de l'information depuis les années 1980 — des ordinateurs personnels jusqu'à Internet, au commerce électronique et à l'Internet des objets — les systèmes numériques sont devenus une partie essentielle de la vie quotidienne. Alors que la frontière entre le monde réel et le monde numérique devient de plus en plus floue, la nécessité d'identifier les utilisateurs de manière précise et sécurisée s'est imposée comme un enjeu majeur de recherche.

Dans des domaines sensibles comme la sécurité et le contrôle d'accès, les méthodes traditionnelles basées sur les mots de passe, les cartes d'identité ou les clés montrent de nombreuses limites. Elles peuvent être perdues, volées ou utilisées à mauvais escient, ce qui augmente le risque de fraude d'identité et de violations de sécurité. Pour cette raison, les gouvernements et les organisations cherchent de plus en plus des solutions d'identification plus fiables.

Les systèmes biométriques apparaissent alors comme une alternative intéressante. Au lieu de s'appuyer sur des objets ou des informations à mémoriser, ils se basent sur des caractéristiques physiologiques ou comportementales uniques, comme les empreintes digitales, la texture de l'iris ou les traits du visage. Étant intimement liées à la personne, ces caractéristiques sont difficiles à perdre, à oublier ou à copier, ce qui améliore à la fois la sécurité et la facilité d'utilisation.

Parmi les méthodes biométriques, la reconnaissance faciale suscite un intérêt particulier en raison de son coût modéré, de sa simplicité d'implémentation et de son acceptation élevée par les utilisateurs. Elle utilise des techniques de vision par ordinateur pour détecter les visages dans des images ou des vidéos et analyser leurs caractéristiques afin d'identifier ou de vérifier une personne. Dans certains cas, elle peut aussi estimer des informations supplémentaires telles que le sexe, l'âge ou l'état émotionnel.

Depuis plusieurs décennies, la reconnaissance faciale a évolué à partir de méthodes classiques comme Eigenfaces, Fisherfaces et Local Binary Patterns (LBPH) — efficaces principalement dans

des environnements bien contrôlés — vers des approches basées sur le deep learning, qui apprennent automatiquement des représentations plus riches du visage à partir de grandes bases de données. Ces nouvelles méthodes améliorent nettement la précision et la robustesse dans des conditions réelles. Néanmoins, des défis restent à relever, notamment en matière de performance dans des environnements non contrôlés, ainsi que vis-à-vis des problèmes liés à la confidentialité et à l'éthique de l'utilisation des données biométriques.

Dans ce chapitre, nous présentons une vue d'ensemble de la reconnaissance faciale, en abordant ses fondements, son principe de fonctionnement, son évolution et les principales approches utilisées dans ce domaine .

1.1 Définition de la reconnaissance faciale :

La reconnaissance faciale est un moyen d'identifier ou de confirmer l'identité d'une personne à l'aide d'une image de son visage. Les systèmes de reconnaissance faciale peuvent être utilisés pour identifier des personnes en temps réel ou pour

identifier des personnes sur des photos ou des vidéos. [1]

La reconnaissance faciale est un procédé biométrique au même

titre que la reconnaissance empreintes digitales, iris, vocale...Cela consiste donc à déterminer l'identité d'une personne. Le système de reconnaissance faciale est une application informatique qui à l'aide d'algorithme, analyse toutes les caractéristiques faciales telles que l'écartement des yeux, des arêtes du nez, des commissures des lèvres, des oreilles. menton, à partir d'une image de son visage qui peut provenir à la fois d'une photo ou d'une vidéo. Donc de nombreux formats d'image peuvent être utilisés qu'ils soient statiques (JPG, bmp, png, GIF ...) ou en mouvement (MPEG4 Part 2).[2]

La reconnaissance faciale est une catégorie de sécurité biométrique. D'autres formes de logiciels biométriques incluent la reconnaissance vocale, la reconnaissance d'empreintes digitales, et la reconnaissance de la rétine ou de l'iris de l'œil. Cette technologie est principalement utilisée pour la sécurité et l'application de la loi, bien qu'il y ait un intérêt croissant pour d'autres domaines d'utilisation.[3]



Figure 1 : image de face

1.2 Historique :

Les premières tentatives de reconnaissance faciale remontent à 1973, lorsque Takeo Kanade, dans le cadre de sa thèse de doctorat à l'Université de Kyoto, a réalisé une étude pionnière visant à analyser et reconnaître les visages à partir de leurs caractéristiques. Ce travail est considéré comme l'une des premières contributions dans ce domaine.

Par la suite, ce domaine a connu une évolution progressive, notamment avec l'essor des grandes entreprises technologiques. En

2011, Facebook a lancé un service de reconnaissance faciale accessible au grand public.

Cependant, ce service a suscité de nombreuses critiques liées à la protection de la vie privée et des données personnelles, ce qui a conduit à son arrêt en Europe en 2012.

En 2015, la même entreprise a développé un système plus avancé appelé DeepFace, basé sur les réseaux de neurones artificiels et l'apprentissage profond. Ce système permet d'identifier un visage avec une grande précision, même sous différentes orientations, grâce à l'analyse de millions d'images pour l'apprentissage des caractéristiques faciales.

Avec les progrès rapides de l'intelligence artificielle entre 2010 et 2020, les systèmes de reconnaissance faciale sont devenus plus performants et largement utilisés dans divers domaines. Ils ont été intégrés dans de nombreuses applications et services du quotidien, notamment sur les appareils mobiles, rendant cette technologie plus accessible au grand public.

En 2020, l'entreprise Clearview AI a suscité une controverse importante en développant une application capable d'identifier des personnes à partir de simples photographies, utilisée notamment par certains services de sécurité, mais fortement critiquée en raison des atteintes à la vie privée.

La même période a également vu l'émergence de systèmes plus avancés comme Aware, développé par Herta Security, capable non seulement de reconnaître des visages, mais aussi d'analyser des foules et de surveiller le respect des mesures sanitaires, notamment la distanciation sociale durant la pandémie de Covid-19.[4]

1.3 Fonctionnement et analyse des systèmes de reconnaissance faciale :

La reconnaissance faciale est un domaine de recherche qui a émergé dans les années 1960–1970 avec les premières tentatives d'automatisation de l'identification humaine à partir des caractéristiques du visage. Depuis, ce domaine a connu plusieurs évolutions majeures, passant des méthodes classiques basées sur

des caractéristiques manuelles vers des approches modernes fondées sur l'intelligence artificielle et l'apprentissage profond [5][7].

Les méthodes traditionnelles de reconnaissance faciale reposent principalement sur l'extraction de caractéristiques géométriques et structurelles du visage, telles que la position des yeux, du nez, des pommettes et de la mâchoire. Ces caractéristiques sont ensuite utilisées pour comparer les visages et effectuer l'identification ou la vérification d'identité [5].

Cependant, ces approches présentent plusieurs limites. Elles nécessitent souvent une forte puissance de calcul et sont sensibles aux variations d'éclairage, aux changements d'expression faciale ainsi qu'aux différentes poses du visage, ce qui peut réduire leur efficacité dans des environnements réels [5][8].

Avec les progrès de l'intelligence artificielle, les approches modernes basées sur le deep learning ont permis d'améliorer considérablement les performances des systèmes de reconnaissance

faciale. Ces méthodes utilisent des réseaux de neurones profonds capables d'extraire automatiquement les caractéristiques discriminantes du visage à partir des images, sans intervention humaine directe [6][5].

Un système complet de reconnaissance faciale comprend généralement trois étapes principales : la détection du visage, le prétraitement des images et enfin la reconnaissance ou l'identification. Chaque étape joue un rôle essentiel dans la performance globale du système [8].

1.3.1 Détection des visages :

Cette étape consiste à localiser un ou plusieurs visages dans une image ou une vidéo, qu'ils soient isolés ou présents dans une scène contenant plusieurs personnes. Elle constitue la première étape essentielle de tout système de reconnaissance faciale [5][9].

1.3.2. Analyse et extraction des caractéristiques :

Après la détection du visage, le système extrait des caractéristiques discriminantes telles que la distance entre les yeux, la forme du nez, les contours du visage et d'autres points faciaux. Ces informations

permettent de représenter chaque individu de manière unique [5][7].

1.3.3. Conversion en représentation numérique :

Les caractéristiques extraites sont ensuite transformées en une représentation mathématique appelée empreinte faciale (face embedding), qui permet de coder le visage sous forme de vecteur numérique pour faciliter la comparaison [6][9].

1.3.4. Comparaison et identification :

Enfin, l'empreinte faciale est comparée à une base de données à l'aide d'algorithmes de correspondance afin d'identifier ou de vérifier l'identité d'une personne [5][6].

1.4 Types de systèmes biométriques :

Les systèmes biométriques utilisés dans la reconnaissance faciale font partie d'un ensemble plus large de technologies d'identification. Selon la littérature, ils peuvent être classés en trois grandes catégories : morphologiques, comportementales et biologiques [8].

Les systèmes morphologiques reposent sur des caractéristiques physiques stables telles que le visage, les empreintes digitales, l'iris ou la rétine. Parmi ces méthodes, les empreintes digitales sont très précises mais nécessitent un contact direct, tandis que la reconnaissance faciale est plus pratique car non intrusive [8].

Les systèmes basés sur l'iris offrent une très grande précision, mais leur coût et leur complexité d'acquisition sont élevés. À l'inverse, la reconnaissance faciale est plus facile à déployer mais reste sensible aux variations d'éclairage, de pose et d'expression [8].

Les systèmes comportementaux, tels que la reconnaissance vocale ou la signature, dépendent du comportement de l'utilisateur et peuvent varier avec le temps [8].

Enfin, les systèmes biologiques comme l'analyse ADN offrent une précision très élevée mais ne sont pas adaptés à une utilisation en temps réel [8].

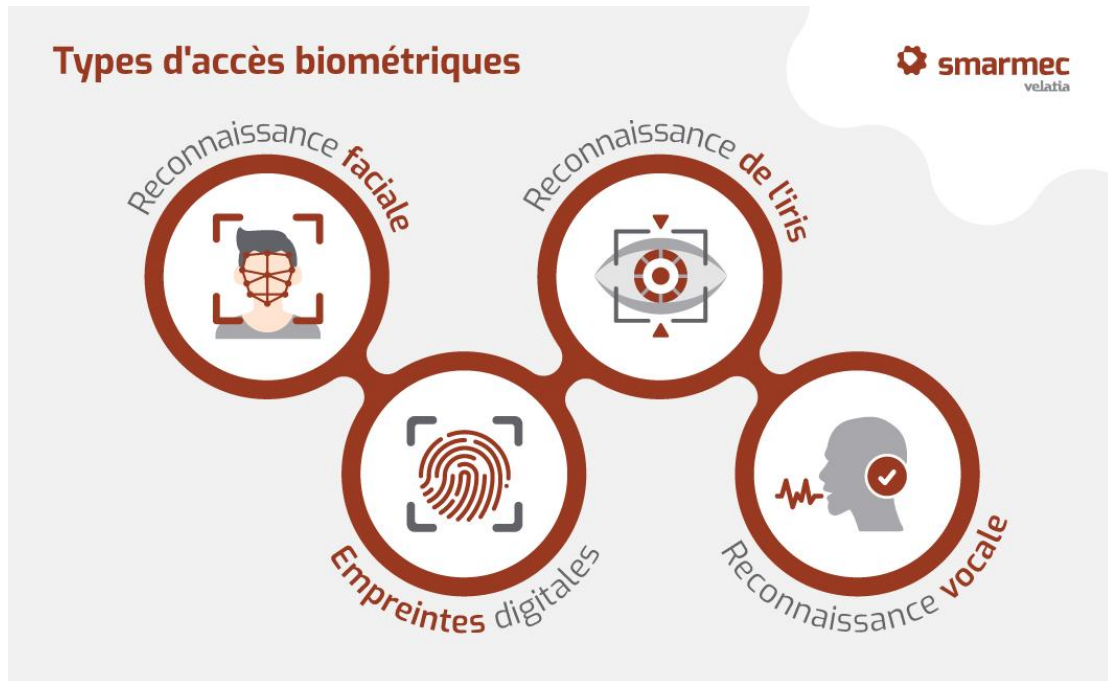


Figure 2 : Types d'accès biométriques

1.5 Techniques classiques (Eigenfaces , Fisherfac , BPH) :

Les méthodes classiques de reconnaissance faciale reposent sur l'extraction de caractéristiques à partir des images du visage sans utilisation de réseaux de neurones profonds. Ces approches utilisent des techniques statistiques et mathématiques pour représenter les visages.

Parmi les méthodes les plus connues, la méthode Eigenfaces est basée sur l'Analyse en Composantes Principales (PCA), qui permet de réduire la dimension des images tout en conservant les informations essentielles [3].

La méthode Fisherfaces repose sur l'Analyse Discriminante Linéaire (LDA), qui améliore la séparation entre différentes classes de visages en maximisant la variabilité inter-classes [3].

La méthode LBPH (Local Binary Pattern Histogram) est une approche locale robuste aux variations d'éclairage et simple à implémenter, largement utilisée dans les systèmes de reconnaissance faciale en temps réel [5].

Ces méthodes sont rapides et peu coûteuses, mais elles restent limitées face aux variations complexes d'éclairage, de pose et d'expression faciale [3].

1.6 Approches modernes (Deep Learning, CNN) :

Les approches modernes de reconnaissance faciale reposent sur l'intelligence artificielle, notamment le Machine Learning et le Deep Learning. Ces méthodes permettent d'extraire automatiquement les caractéristiques des visages à partir des données.

Les réseaux de neurones convolutifs (CNN) sont les architectures les plus utilisées pour la reconnaissance faciale grâce à leur capacité à apprendre des représentations hiérarchiques des images [2].

D'autres approches incluent les auto-encodeurs (AE), les réseaux antagonistes génératifs (GAN) ainsi que des architectures hybrides, améliorant ainsi la performance des systèmes [9].

Ces méthodes offrent une très haute précision mais nécessitent de grandes bases de données et une puissance de calcul importante

[2][9].

1.7 Avantages et inconvénients de la reconnaissance faciale :

La reconnaissance faciale est une technologie non intrusive, rapide et facile à utiliser, ce qui la rend adaptée à de nombreuses applications modernes.

Cependant, elle présente certaines limites, notamment sa sensibilité aux variations d'éclairage, aux angles de prise de vue et aux expressions faciales.

De plus, les modèles basés sur le Deep Learning nécessitent des ressources de calcul importantes ainsi que de grandes quantités de données pour l'entraînement [8].

1.8 Domaines de la reconnaissance faciale :

La reconnaissance faciale est une technologie biométrique largement utilisée dans de nombreux domaines de la vie quotidienne et professionnelle grâce à sa rapidité et sa facilité d'utilisation.

Dans les appareils mobiles, elle est utilisée pour le déverrouillage des smartphones afin de renforcer la sécurité des données personnelles [1][3].

Dans le domaine de la sécurité et de l'application de la loi, elle est utilisée pour identifier des suspects et comparer des images avec des bases de données criminelles [8].

Dans les aéroports et les frontières, elle est intégrée dans les systèmes de contrôle automatisé afin d'améliorer la sécurité et réduire le temps d'attente.

Dans le secteur bancaire, elle permet une authentification sécurisée lors des transactions en ligne.

Dans le domaine de la santé, elle est utilisée pour l'accès aux dossiers médicaux et le suivi des patients.

Enfin, elle est également utilisée dans le commerce, l'éducation et les systèmes de surveillance intelligents [8][9].

Chapitre 2 : Le Cloud Computing.

Chapitre 2 : Le Cloud Computing.

1 Introduction:

Bien qu'il soit présent sur le marché depuis plusieurs années, le Cloud Computing reste encore une source de confusion pour un grand nombre d'utilisateurs. En réalité, beaucoup de personnes utilisent déjà des services cloud sans même s'en rendre compte, à travers des applications quotidiennes telles que les réseaux sociaux, les services de messagerie ou encore le stockage en ligne.

Il est difficile de trouver une définition unique et universellement acceptée du Cloud Computing, car ce domaine évolue constamment avec les avancées technologiques. Cependant, selon le National Institute of Standards and Technology (NIST), le Cloud Computing peut être défini comme un modèle permettant un accès réseau pratique, à la demande et en libre-service à des ressources informatiques partagées et configurables, telles que des serveurs, du stockage, des applications et des services, pouvant être rapidement provisionnées et libérées avec un minimum d'effort de gestion.

Ce chapitre vise à présenter l'état de l'art du Cloud Computing, à clarifier ses concepts fondamentaux et à proposer une compréhension unifiée de ses objectifs. Il met également en évidence les principaux avantages qu'il offre, notamment en termes de flexibilité, de réduction des coûts et d'accessibilité.

Le Cloud Computing repose sur l'utilisation de serveurs distants accessibles via Internet, permettant aux utilisateurs de stocker, traiter et gérer leurs données sans avoir besoin d'infrastructures physiques locales. Les données et les applications sont hébergées dans des centres de données répartis dans le monde entier, offrant ainsi une grande disponibilité et une continuité de service.

Ce modèle technologique transforme profondément la manière dont les entreprises et les particuliers utilisent les ressources informatiques. En effet, il permet aux utilisateurs d'accéder à leurs fichiers et applications depuis n'importe quel appareil connecté à Internet, tout en garantissant une synchronisation en temps réel. Pour les entreprises, le Cloud Computing représente une solution efficace pour réduire les coûts liés à la maintenance des serveurs, tout en facilitant la collaboration et l'expansion internationale.

Ainsi, le Cloud Computing s'impose aujourd'hui comme une technologie incontournable, au cœur de la transformation numérique des organisations modernes.

Ce chapitre sera structuré en plusieurs sections présentant les modèles, les types et les caractéristiques du Cloud Computing.

1.1 Définition du Cloud Computing :

Le Cloud Computing, ou informatique en nuage, est un paradigme technologique permettant un accès pratique et à la demande à un ensemble partagé de ressources informatiques configurables. Ces ressources incluent notamment les réseaux, les serveurs, le stockage, les applications et les services, qui peuvent être rapidement fournis et libérés avec un minimum d'effort de gestion ou d'intervention du fournisseur [10][11].

Selon le National Institute of Standards and Technology (NIST), le Cloud Computing est défini comme :

« Un modèle permettant un accès réseau omniprésent, pratique et à la demande à un ensemble partagé de ressources informatiques configurables, qui peuvent être rapidement provisionnées et libérées avec

un minimum d'effort de gestion ou d'interaction avec le fournisseur de service » [10].

Cette définition met en évidence plusieurs caractéristiques essentielles du Cloud Computing, notamment l'accès à la demande, la mutualisation des ressources, la flexibilité et la scalabilité [11][12].

Le modèle proposé par le NIST constitue une référence fondamentale dans le domaine du Cloud Computing, car il permet de standardiser la compréhension de cette technologie et de faciliter la comparaison entre les différents services cloud proposés sur le marché. Il sert également de base pour distinguer les différents modèles de services et de déploiement utilisés dans les environnements cloud modernes [10][12].

Ainsi, la définition du NIST joue un rôle central dans la compréhension du Cloud Computing, en fournissant un cadre clair et structuré permettant d'analyser ses composants et ses caractéristiques principales [10].

1.2 Historique :

Les origines du Cloud Computing remontent aux années 1960 avec l'apparition du concept de partage de temps (Time-Sharing). À cette époque, plusieurs utilisateurs pouvaient accéder à distance à des ordinateurs centraux (mainframes) afin d'exécuter leurs tâches. Le

modèle reposait principalement sur des centres de données centralisés, où les utilisateurs envoyaient leurs travaux pour traitement, constituant ainsi les premières formes d'accès partagé aux ressources informatiques [11].

Dans les années 1990, le terme « cloud » a commencé à être utilisé de manière métaphorique. En 1994, la société General Magic l'a employé pour désigner un environnement virtuel dans lequel évoluaient des services logiciels. Par la suite, en 1996, Compaq a contribué à populariser cette idée en anticipant le développement des services informatiques basés sur Internet, notamment le stockage en ligne et les applications accessibles à distance [14].

Au début des années 2000, le Cloud Computing a commencé à prendre une forme concrète avec l'émergence d'Amazon Web Services (AWS) en 2002, permettant aux développeurs de créer des applications sans dépendre d'une infrastructure physique dédiée. En 2006, Amazon a lancé des services majeurs tels qu'Amazon S3 (stockage) et Amazon EC2 (calcul), marquant une étape importante dans l'évolution du Cloud Computing [18].

En 2008, la NASA a également contribué au développement des technologies cloud en participant à des projets open source destinés aux environnements cloud privés et hybrides [16].

Au cours de la décennie suivante, le Cloud Computing s'est fortement développé avec le lancement de plusieurs plateformes majeures, notamment Microsoft Azure en 2010, IBM SmartCloud en 2011 et Oracle Cloud en 2012 [13][14][26].

Ainsi, le Cloud Computing a évolué progressivement d'un simple concept de partage de ressources à une technologie essentielle et incontournable dans les systèmes informatiques modernes [11][12].

1.3 Les Caractéristiques Du Cloud Computing :

Le cloud computing représente aujourd'hui une technologie essentielle dans le domaine des technologies de l'information. Il permet aux utilisateurs d'accéder à des ressources et des services informatiques via Internet de manière flexible et efficace. Le National Institute of Standards and Technology (NIST) a défini cinq caractéristiques essentielles qui distinguent le cloud computing des infrastructures informatiques traditionnelles [10].

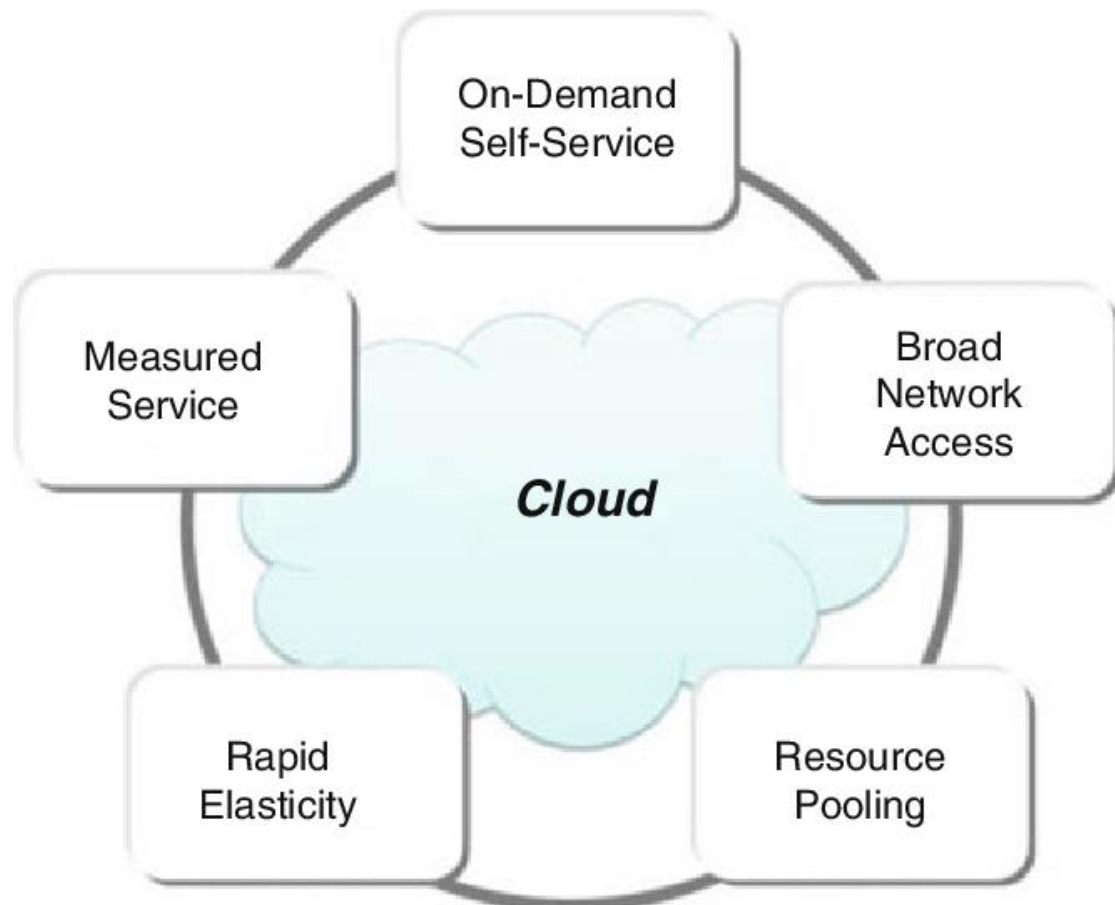


Figure 3: Les Caractéristiques Du Cloud Computing

1. Service à la demande (On-Demand Self-Service) :

Le cloud computing permet aux utilisateurs d'obtenir automatiquement des ressources informatiques telles que les serveurs, le stockage ou les bases de données, sans intervention directe du fournisseur de service. Les utilisateurs peuvent également gérer et configurer ces services à travers des plateformes en ligne [10].

2. Large accès réseau (Broad Network Access) :

Les services cloud sont accessibles via Internet à partir de différents appareils comme les ordinateurs, les smartphones et les tablettes. Cette caractéristique facilite le travail à distance ainsi que la collaboration entre les utilisateurs [11].

3. Mutualisation des ressources (Resource Pooling) :

Le cloud computing repose sur le partage des ressources entre plusieurs utilisateurs grâce à une infrastructure commune. Les ressources physiques et virtuelles sont attribuées dynamiquement selon les besoins des clients, ce qui permet d'optimiser les performances et de réduire les coûts tout en garantissant la sécurité et la confidentialité [12].

4. Élasticité rapide (Rapid Elasticity) :

Les ressources cloud peuvent être augmentées ou réduites rapidement selon la demande. Les entreprises ont ainsi la possibilité d'adapter facilement leurs capacités de stockage ou de traitement sans avoir à acheter de nouveaux équipements, ce qui offre une grande flexibilité [11][18].

5. Service mesuré (Measured Service) :

Les systèmes cloud utilisent des mécanismes de surveillance et de mesure pour contrôler l'utilisation des ressources telles que le stockage, la bande passante et la puissance de calcul. La facturation est basée sur la consommation réelle des utilisateurs selon le modèle « pay-as-you-go », permettant une meilleure gestion des coûts [10][18].

1.4 Types De Cloud Computing :

Le cloud computing repose sur plusieurs modèles essentiels qui permettent de fournir des services informatiques de manière flexible, évolutive et efficace. Ces modèles se divisent principalement en modèles de déploiement (public, privé, hybride, communautaire et multi-cloud) et en modèles de services (IaaS, PaaS, SaaS et FaaS). Chaque modèle répond à des besoins spécifiques en fonction des exigences des organisations [12].

Le choix d'un modèle de cloud dépend de plusieurs facteurs, notamment la sensibilité des données, les contraintes de sécurité et de conformité, les besoins en performance et en évolutivité, ainsi que le budget disponible [11].

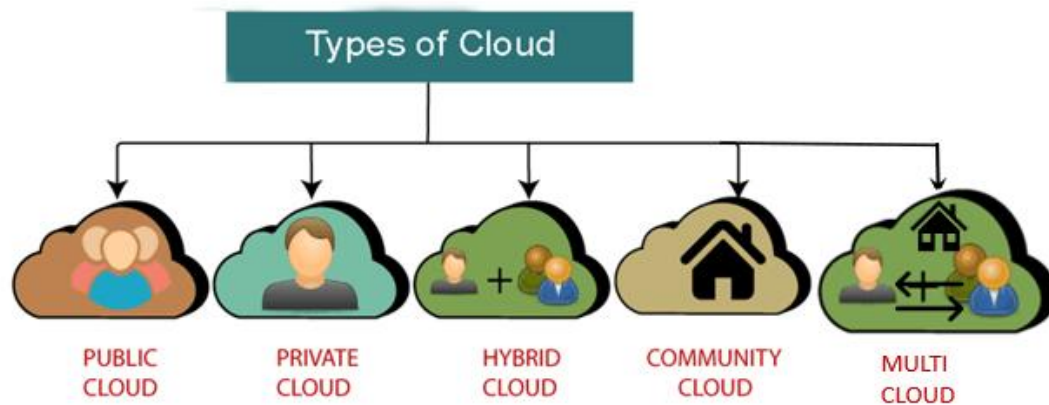


Figure 4: types of cloud

1. Cloud Public :

Le cloud public est un modèle dans lequel les services informatiques sont fournis par des prestataires externes via Internet. Les ressources (serveurs, stockage, applications) sont partagées entre plusieurs utilisateurs selon un modèle de paiement à l'usage [18].

Ce modèle est apprécié pour sa flexibilité, sa scalabilité et son coût réduit, car il ne nécessite pas d'investissement initial important. Il est particulièrement adapté aux petites entreprises, aux startups et aux environnements à charge variable [13][18].

2. Cloud Privé :

Le cloud privé est une infrastructure dédiée à une seule organisation. Elle peut être gérée en interne ou par un fournisseur externe [14].

Il offre un niveau élevé de sécurité, de contrôle et de personnalisation, ce qui le rend idéal pour les secteurs sensibles comme la finance, la santé ou les administrations publiques [12][14].

3. Cloud Hybride :

Le cloud hybride combine les environnements public et privé. Il permet de conserver les données sensibles dans un cloud privé tout en utilisant les ressources du cloud public pour des besoins supplémentaires [13].

Ce modèle offre un équilibre entre sécurité, flexibilité et optimisation des coûts, et permet notamment le concept de cloud bursting pour gérer les pics de charge [12].

4. Cloud Communautaire :

Le cloud communautaire est partagé entre plusieurs organisations ayant des besoins communs (sécurité, conformité, mission). Les ressources sont mutualisées entre les membres de la communauté, ce qui permet de réduire les coûts tout en maintenant un niveau de sécurité adapté [12].

5. Multi-Cloud :

Le multi-cloud consiste à utiliser plusieurs fournisseurs de services cloud en parallèle afin d'améliorer la flexibilité et d'éviter la dépendance à un seul fournisseur [15][26].

1.5 Types de services de Cloud Computing :

Les services de cloud computing représentent les différentes façons dont les ressources informatiques sont fournies aux utilisateurs via Internet. Ils couvrent principalement les infrastructures, les plateformes de développement et les applications. Ces modèles peuvent être utilisés séparément ou combinés selon les besoins des organisations [10][12].

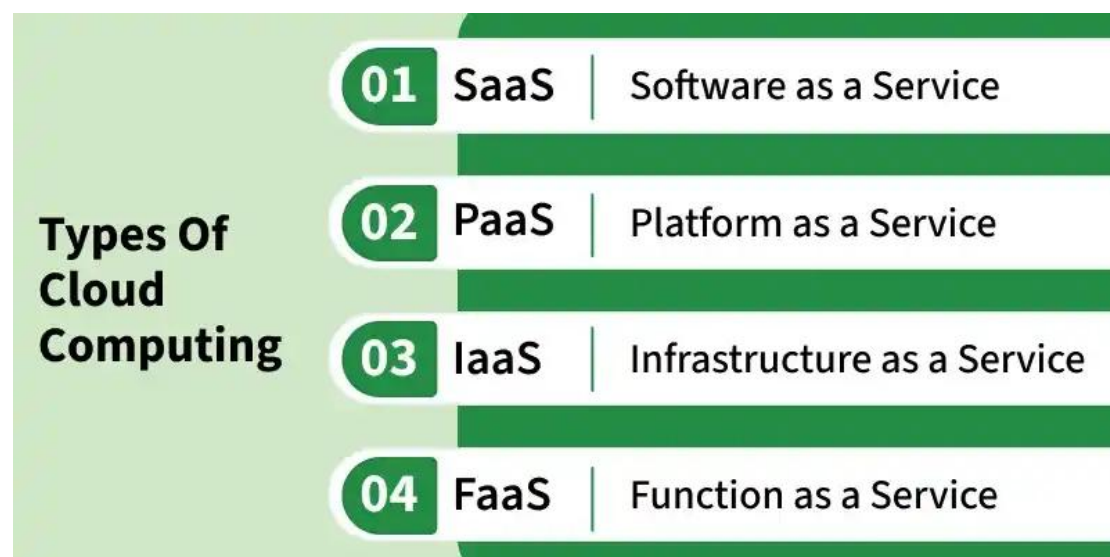


Figure 5: Types de services de Cloud Computing

1. Infrastructure as a Service (IaaS) :

L'IaaS fournit des ressources informatiques de base à la demande, telles que les serveurs, le stockage, les réseaux et la virtualisation [18].

Dans ce modèle, le fournisseur est responsable de l'infrastructure physique, tandis que l'utilisateur gère les systèmes d'exploitation, les applications et les données [12].

Points essentiels :

Accès à des ressources informatiques virtualisées

Gestion flexible de l'environnement utilisateur

Paieement selon la consommation réelle

Avantages :

Grande flexibilité

Réduction des coûts d'équipement

Adapté aux besoins techniques avancés [11][18]

2. Platform as a Service (PaaS) :

Le PaaS fournit un environnement complet destiné au développement, au test et au déploiement des applications, sans que l'utilisateur ait à gérer l'infrastructure [13].

Le fournisseur s'occupe des serveurs, du stockage et des outils nécessaires au développement.

Points essentiels :

Environnement de développement prêt à l'emploi

Outils intégrés pour les développeurs

Gestion automatisée de l'infrastructure

Avantages :

Simplification du développement

Gain de temps important

Déploiement rapide des applications [13][15]

3. Software as a Service (SaaS) :

Le SaaS propose des applications complètes accessibles directement via Internet, sans installation locale [14].

Le fournisseur gère entièrement l'infrastructure, la maintenance et les mises à jour.

Points essentiels :

Accès via navigateur ou application

Aucune installation requise

Service entièrement géré

Avantages :

Facilité d'utilisation

Accessibilité depuis n'importe où

Réduction des coûts de maintenance [11][14]

4. Function as a Service (FaaS) / Serverless :

Le modèle serverless permet d'exécuter des fonctions spécifiques sans gestion de serveurs [15].

L'infrastructure est automatiquement gérée par le fournisseur, et la facturation se fait uniquement selon l'exécution réelle.

Points essentiels :

Exécution basée sur des événements

Absence de gestion de serveurs

Mise à l'échelle automatique

Avantages :

Optimisation des coûts

Haute efficacité

Idéal pour les applications modernes [15][18]

1.6 Avantages du Cloud Computing :

Le cloud computing est une technologie clé des systèmes d'information modernes. Il permet aux organisations d'accéder à des ressources informatiques flexibles, évolutives et économiques, tout en améliorant la gestion des données et la performance globale des systèmes [11][12].

1. Réduction des coûts :

Le cloud computing permet de diminuer les coûts liés à l'achat, l'installation et la maintenance des infrastructures informatiques traditionnelles [18].

Il repose sur un modèle de paiement à l'usage (pay-as-you-go), où les entreprises ne paient que pour les ressources réellement consommées, ce qui réduit les investissements initiaux [10].

2. Scalabilité et flexibilité :

Les ressources cloud peuvent être ajustées rapidement en fonction des besoins [11].

Cette élasticité permet aux organisations de s'adapter aux variations de charge de travail et d'améliorer les performances des applications en période de forte demande [12].

3. Accessibilité et mobilité :

Les services cloud sont accessibles via Internet depuis n'importe quel endroit et à partir de différents appareils (ordinateurs, smartphones, tablettes) [13].

Cela facilite le travail à distance et améliore la collaboration entre les utilisateurs.

4. Stockage et sauvegarde des données :

Le cloud offre des capacités de stockage importantes ainsi que des mécanismes de sauvegarde automatique [14].

Cela permet de sécuriser les données et de réduire les risques de perte liés aux pannes ou aux incidents techniques.

5. Maintenance simplifiée :

Les fournisseurs de services cloud assurent la gestion de l'infrastructure, les mises à jour et la maintenance des systèmes [15].

Cela permet aux entreprises de se concentrer sur leurs activités principales.

1.7 Limites et défis du Cloud Computing :

Malgré ses nombreux avantages, le cloud computing présente également certaines limites qu'il est important de prendre en compte [11].

1. Sécurité et confidentialité des données :

Le stockage des données sur des serveurs externes peut exposer les informations à des risques de cyberattaques ou d'accès non autorisé [17][28].

La sécurité des données reste donc un enjeu majeur dans les environnements cloud.

2. Dépendance à la connexion Internet :

L'accès aux services cloud dépend entièrement d'une connexion Internet stable.

Toute interruption ou faiblesse du réseau peut affecter l'utilisation des services [12].

3. Latence et performance :

Le traitement à distance peut entraîner des délais de réponse, en particulier pour les applications nécessitant un traitement en temps réel [11].

4. Dépendance au fournisseur (Vendor Lock-in) :

Certaines organisations peuvent devenir dépendantes d'un fournisseur cloud, ce qui rend difficile le changement de plateforme en raison des coûts ou des différences technologiques [12][15].

5. Conformité réglementaire :

Certaines entreprises doivent respecter des réglementations strictes concernant le stockage et le traitement des données, notamment dans les secteurs bancaire, médical et gouvernemental [17].

1.8 Architecture du Cloud Computing :

L'architecture cloud désigne l'organisation et la structuration des différents composants et technologies nécessaires au fonctionnement du cloud computing. Elle définit la manière dont les ressources informatiques (matériel, logiciels, réseaux et services) sont

interconnectées et utilisées pour fournir des services cloud efficaces, évolutifs et sécurisés [10][12].

On peut considérer l'architecture cloud comme un plan d'organisation permettant de concevoir, déployer et gérer des applications dans un environnement cloud.

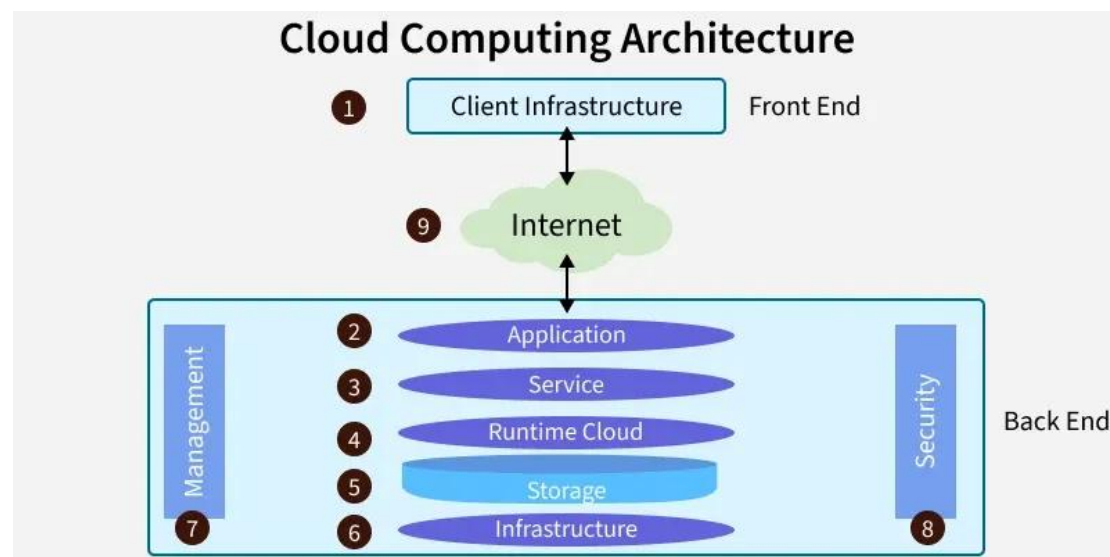


Figure 6:Architecture du Cloud Computing

8.1 Définition de l'architecture cloud :

L'architecture cloud représente la structure globale qui relie les différents éléments du cloud computing, tels que les serveurs, le stockage, les applications, les réseaux virtuels et les mécanismes de sécurité [12].

Elle permet de déterminer la meilleure façon d'assembler ces ressources afin de répondre aux besoins d'une organisation en termes de performance, de sécurité et d'évolutivité [11].

8.2 Composants de l'architecture cloud :

L'architecture cloud est généralement divisée en deux parties principales : le frontend (côté utilisateur) et le backend (côté fournisseur).

1. Frontend (interface utilisateur) :

Le frontend correspond à la partie visible par l'utilisateur. Il regroupe les éléments qui permettent d'accéder aux services cloud.

Composants :

Interface utilisateur (navigateur, application mobile, etc.)

Applications client

Appareils utilisés (ordinateur, smartphone, tablette)

Rôle :

Permet à l'utilisateur d'interagir avec les services cloud et d'envoyer des requêtes au système.

2. Backend (infrastructure cloud) :

Le backend représente l'ensemble des ressources et systèmes qui constituent le cloud lui-même.

a) Applications :

Elles traitent les demandes des utilisateurs et assurent la communication entre les différentes parties du système.

b) Services :

Ils représentent le cœur du cloud computing, car ils gèrent les ressources comme le stockage, les applications et les environnements de développement [10].

c) Environnement d'exécution (Runtime Cloud) :

Il permet l'exécution des services dans un environnement virtualisé, similaire à un système d'exploitation, en utilisant la virtualisation et les hyperviseurs [12].

d) Stockage :

Il correspond à l'espace où sont stockées les données des utilisateurs. Il est conçu pour être flexible, sécurisé et évolutif.

e) Infrastructure :

Elle comprend les composants matériels essentiels tels que les processeurs, les GPU, les serveurs et les équipements réseau [29].

Chapitre3 :

Conception

Implémentation et

Évaluation du

Système de

Reconnaissance

Faciale Basé sur

le Cloud Computing

Chapitre3 : Conception, Implémentation et Évaluation du Système de Reconnaissance Faciale Basé sur le Cloud Computing

1. Introduction:

Dans ce chapitre, nous présentons un système de reconnaissance faciale basé sur les techniques de Deep Learning, intégré dans un environnement de Cloud Computing, avec une touche d'intelligence artificielle.

Avant de commencer la conception du système, nous nous sommes inspirés d'un cas réel suite à une visite à la résidence universitaire Ahmed Medghri. Cette expérience a été réalisée en collaboration avec le responsable du système de reconnaissance faciale des étudiantes de la résidence. Elle nous a permis d'identifier l'idée du projet, que nous avons ensuite cherché à améliorer en proposant des solutions aux différents problèmes rencontrés dans le système existant.

Le développement du système s'est articulé autour de trois principales étapes.

Premièrement, nous avons mis en place un système de reconnaissance faciale dans un environnement de cloud local, en conservant

volontairement les mêmes problématiques existantes afin de les étudier en conditions réelles. Cette étape nous a également permis de simuler le fonctionnement d'un environnement Cloud et de mieux comprendre ses principes.

Deuxièmement, nous avons développé un système de reconnaissance faciale dans un environnement de cloud réel, dans lequel nous avons proposé des solutions aux problèmes identifiés lors de la première étape, tout en améliorant les performances globales du système.

Troisièmement, nous avons conçu une approche hybride combinant le cloud local et le cloud réel, afin d'optimiser davantage les performances, la flexibilité et la robustesse du système.

Dans la suite de ce chapitre, nous détaillerons chacune de ces étapes.

1.2 Problématique:

Le système de reconnaissance faciale utilisé au sein de la résidence universitaire Ahmed Medghri (section féminine) présente une limite importante liée à la confusion entre des individus ayant une forte ressemblance faciale, notamment dans le cas de vrais jumeaux ou de

personnes très similaires. En effet, le système tend à les identifier comme une seule et même personne, ce qui engendre des erreurs d'identification.

Cette problématique a entraîné certaines difficultés dans la gestion quotidienne de la résidence, notamment lors de la distribution des repas aux étudiantes, où des cas d'erreurs d'accès ou de confusion ont été observés. Afin de contourner temporairement ce problème, des solutions non optimales ont été mises en place, telles que le changement d'angle de prise de vue, l'ajout d'accessoires (comme des lunettes), ou d'autres ajustements visuels, sans toutefois résoudre le problème de manière définitive.

Dans le cadre de ce travail, nous proposons d'aborder cette problématique et de développer des solutions plus robustes à travers la conception d'un système de reconnaissance faciale basé sur le Cloud Computing réel, ainsi qu'une approche hybride combinant cloud local et cloud réel.



Figure 7: image Le système de reconnaissance faciale la résidence universitaire Ahmed Medghri

1. Système de reconnaissance faciale dans un cloud local:

Suite à la problématique présentée précédemment, liée à l'incapacité du système à distinguer efficacement les individus présentant une forte ressemblance faciale, notamment les jumeaux, nous avons jugé nécessaire de commencer par une étude expérimentale de ce problème à travers une première implémentation.

Dans cette première étape, nous avons développé un système de reconnaissance faciale dans un environnement de cloud local (Local

Cloud). L'objectif principal n'était pas de proposer une solution définitive, mais plutôt d'analyser le comportement du système dans un environnement simulé proche du Cloud Computing réel. Cette approche nous a permis de conserver les mêmes limitations existantes afin de mieux les étudier et d'évaluer leur impact sur les performances du système.

Dans le prolongement de la première étape expérimentale présentée précédemment, basée sur un environnement de cloud local, nous avons procédé à une conception plus détaillée de l'architecture du système développé. Cette architecture décrit l'organisation globale des différents composants ainsi que leurs interactions afin d'assurer le bon fonctionnement du système de reconnaissance faciale.

1.2 Architecture du système:

L'architecture du système développé repose sur une approche distribuée combinant un environnement local, une machine virtuelle et une interface Web d'administration. Cette architecture permet de séparer les différentes responsabilités du système afin d'améliorer la performance, la flexibilité et la maintenance.

Le système est composé de trois modules principaux :

1.2.1 Le serveur Cloud (Machine virtuelle - VM):

Le serveur constitue le cœur du système et est implémenté à l'aide du framework Flask. Il est déployé sur une machine virtuelle simulant un environnement cloud. Ce serveur est responsable de la réception des images, du traitement de la reconnaissance faciale, ainsi que de la gestion des bases de données contenant les visages connus et inconnus.

Il expose plusieurs API REST telles que :

1. add pour l'ajout d'une nouvelle personne
2. delete pour la suppression d'un utilisateur
3. list pour afficher les personnes enregistrées
4. recognize pour la reconnaissance faciale en temps réel

Le serveur assure également la journalisation des événements (log.txt), notamment les détections de visages inconnus avec horodatage.

1.2.2 Interface d'administration (Web - Windows):

L'interface d'administration est une application Web exécutée sur un système Windows. Elle permet à l'administrateur de gérer facilement le système via un navigateur.

Cette interface permet :

l'ajout de nouvelles personnes avec leurs images.

la suppression d'utilisateurs enregistrés.

la consultation des visages connus.

la visualisation des visages inconnus détectés en temps réel.

le déclenchement d'une alarme sonore en cas de détection d'un intrus.

L'interface communique directement avec le serveur via des requêtes HTTP en utilisant l'adresse IP de la machine virtuelle.

1.2.3. Client de capture (Système local):

Le client est une application développée en Python utilisant OpenCV. Il est responsable de la capture vidéo en temps réel à partir de la webcam.

Chaque image capturée est envoyée au serveur pour traitement. En retour, le serveur renvoie les résultats de reconnaissance contenant le nom de la personne détectée ainsi que les coordonnées du visage. Ces informations sont ensuite affichées directement sur le flux vidéo en temps réel.

Cette architecture distribuée permet de combiner les avantages du traitement local et du cloud computing. Elle assure une reconnaissance faciale en temps réel tout en garantissant une gestion centralisée et sécurisée des données.

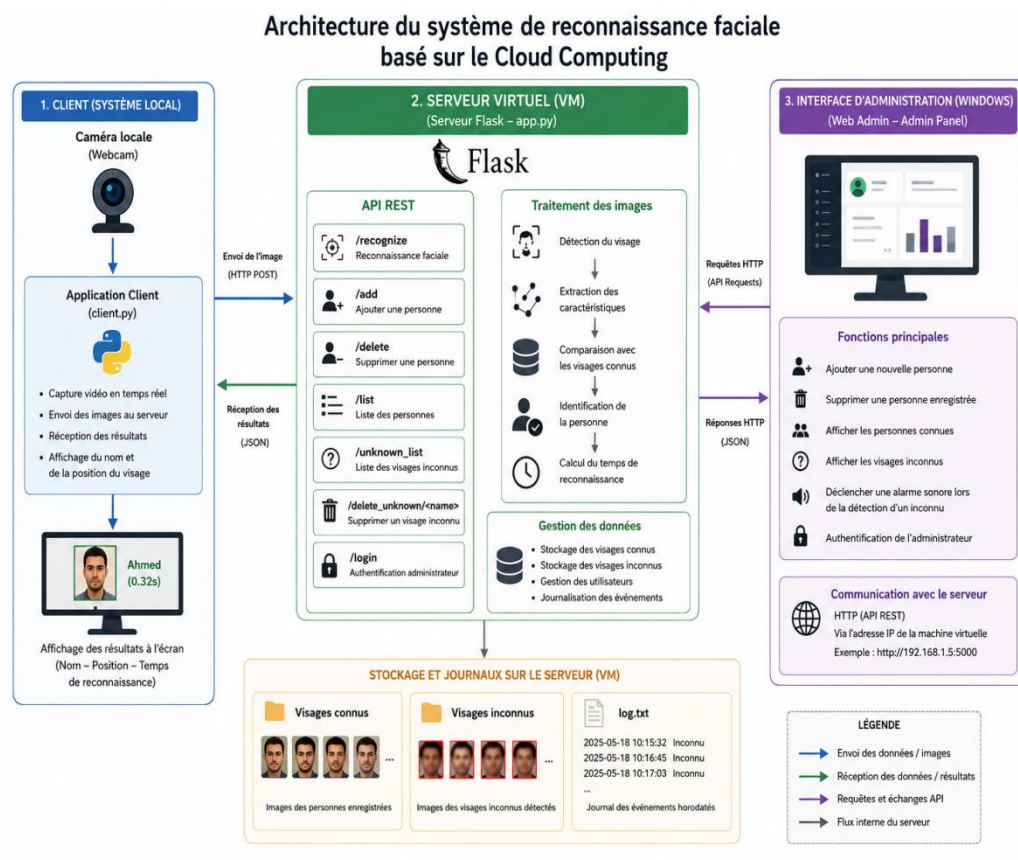


Figure 8: Architecture du système sur le cloud local

1.3 Outils et Technologies Utilisés:

Dans cette section, nous présentons les technologies et outils utilisés pour le développement du système, en expliquant le rôle de chaque technologie dans le projet.

1.3.1 Python:

Python est le langage principal utilisé pour le développement du système. Il a été choisi grâce à sa simplicité, sa lisibilité et surtout la richesse de ses bibliothèques dédiées à l'intelligence artificielle et au traitement d'images.

Dans ce projet, Python a permis de :

gérer les algorithmes de reconnaissance faciale,

manipuler les images et les flux vidéo,

créer la communication entre client et serveur.

De plus, sa compatibilité avec les frameworks modernes facilite le développement rapide et efficace des applications basées sur l'IA.

1.3.2 OpenCV:

OpenCV (Open Source Computer Vision Library) est une bibliothèque

spécialisée dans le traitement d'images et la vision par ordinateur.

Dans ce système, OpenCV a été utilisé pour :

capturer les images via la webcam,

détecter les visages en temps réel,

prétraiter les images (redimensionnement, conversion en niveaux de gris),

afficher les résultats de reconnaissance.

OpenCV joue donc un rôle essentiel dans la partie acquisition et préparation des données visuelles.

1.3.3 Bibliothèque Face Recognition:

La bibliothèque Face Recognition est basée sur des modèles de deep learning permettant d'encoder les visages sous forme de vecteurs numériques.

Elle est utilisée pour :

détecter automatiquement les visages dans une image,

extraire les caractéristiques faciales (face encodings),

comparer les visages détectés avec ceux stockés dans la base de données,

déterminer l'identité de la personne.

Cette bibliothèque simplifie fortement la mise en place d'un système de reconnaissance faciale précis et rapide.

1.3.4 Flask:

Flask est un micro-framework web en Python utilisé pour créer des API REST légères et efficaces.

Dans ce projet, Flask permet :

la communication entre le client et le serveur,

l'envoi et la réception des images,

le retour des résultats en format JSON,

la gestion des utilisateurs et des opérations administratives.

Flask a été choisi pour sa flexibilité et sa facilité d'intégration avec Python.

1.3.5 Machine Virtuelle (Virtual Machine):

La machine virtuelle représente l'environnement serveur du système.

Elle joue plusieurs rôles :

hébergement du serveur Flask,

centralisation des traitements,

simulation d'un environnement cloud local,

gestion des ressources système de manière sécurisée.

L'utilisation d'une VM permet de reproduire une architecture cloud réelle tout en gardant un environnement de test contrôlé.

1.3.6 HTML, CSS et JavaScript:

Ces technologies ont été utilisées pour développer l'interface web d'administration.

HTML : structure des pages web

CSS : design et mise en forme

JavaScript : interaction dynamique avec le serveur

Cette interface permet à l'administrateur de :

gérer les utilisateurs,

consulter les visages inconnus,

visualiser les logs,

contrôler le système à distance.

1.4 Conception du Système:

Le système est conçu sous forme d'une architecture modulaire, où chaque module remplit une tâche spécifique dans le processus global de reconnaissance faciale.

1.4.1 Module de Détection des Visages:

Ce module a pour objectif de localiser les visages dans une image ou une

vidéo.

Il fonctionne en plusieurs étapes :

réception de l'image,

conversion en format exploitable,

application d'algorithmes de détection,

extraction des zones contenant des visages.

Ce module est fondamental car il prépare les données pour les étapes suivantes.

1.4.2 Module d'Extraction des Caractéristiques:

Une fois le visage détecté, le système extrait des caractéristiques uniques appelées face encodings.

Ces caractéristiques représentent :

la forme du visage,

la distance entre les yeux,

les traits biométriques globaux.

Chaque visage est transformé en vecteur numérique, ce qui permet une comparaison efficace.

1.4.3 Module de Reconnaissance Faciale:

Ce module compare les encodings extraits avec ceux stockés dans la base

de données.

Le processus est le suivant :

calcul de la distance entre vecteurs,

définition d'un seuil de similarité,

identification si correspondance trouvée.

Résultat :

si correspondance → personne identifiée,

sinon → classé comme Unknown.

1.4.4 Module de Traitement Cloud:

Ce module représente le cœur de l'architecture cloud.

Il permet :

le traitement centralisé des données,

la réception des requêtes client,

l'exécution des algorithmes sur le serveur,

le renvoi des résultats en JSON.

Cette approche améliore la performance et permet une gestion centralisée du système.

1.5 Implémentation du Système:

L'implémentation a été réalisée en plusieurs étapes progressives afin de garantir la stabilité du système.

Étape 1 : Développement du serveur Flask

Le serveur Flask a été développé comme point central du système.

Il propose plusieurs API :

reconnaissance faciale,

ajout d'utilisateurs,

suppression d'utilisateurs,

authentification,

gestion des visages inconnus.

Chaque API joue un rôle précis dans le fonctionnement global.

Étape 2 : Création du client de capture

Un client Python basé sur OpenCV a été développé.

Ses fonctions :

capture d'images en temps réel,

envoi automatique vers le serveur,

affichage des résultats reçus.

Étape 3 : Développement de l'interface d'administration

Une interface web interactive a été développée pour faciliter la gestion du système.

Elle permet :

contrôle du système via navigateur,

consultation des données,

gestion des utilisateurs.

Étape 4 : Gestion des données

Le système organise les données comme suit :

Known Faces : stockage des visages connus,

Unknown Faces : stockage des visages non reconnus,

log.txt : enregistrement des événements.

Cette organisation facilite la traçabilité et la maintenance.

Étape 5 : Déploiement sur la machine virtuelle

Le serveur Flask est déployé sur une machine virtuelle locale.

Avantages :

simulation d'un cloud privé,

accès distant,

sécurité et isolation du système.

Étape 6 : Tests du système

Les tests ont couvert plusieurs aspects :

détection des visages,

précision de reconnaissance,

performance du serveur,

communication client-serveur,

gestion des cas inconnus.

Les résultats ont montré que le système fonctionne de manière stable et efficace.

1.5.1 Gestion des données:

Les données du système sont stockées localement sur la machine virtuelle. Les images des personnes connues sont enregistrées dans un dossier dédié (Known Faces), tandis que les visages non reconnus sont sauvegardés dans un répertoire séparé (Unknown Faces). Les encodages faciaux sont générés automatiquement au démarrage du système à partir des images disponibles.

1.6 Workflow du Système:

Le fonctionnement global suit le pipeline suivant :

Capture d'image via webcam

Envoi vers serveur Flask

Détection des visages

Extraction des caractéristiques

Comparaison avec base de données

Identification ou classification Unknown

Enregistrement si nécessaire

Retour du résultat en JSON

Affichage à l'utilisateur

Ce workflow garantit un traitement structuré et rapide des données en temps réel.

1.7 Diagramme UML (class + use case):

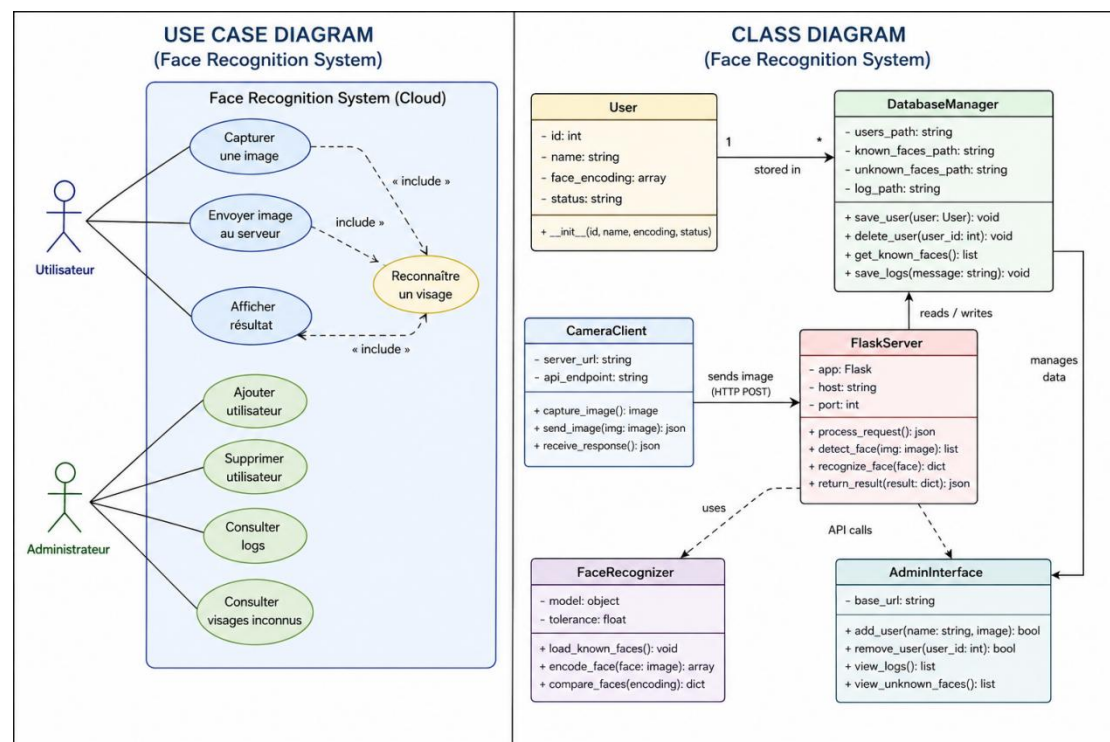


Figure 9: Diagramme UML (class + use case)

1.8 Conclusion:

Ce chapitre a présenté en détail la conception et l'implémentation du système de reconnaissance faciale basé sur le Cloud Computing.

Nous avons décrit :

les outils utilisés,

l'architecture du système,

les différents modules fonctionnels,

ainsi que les étapes de développement et de test.

Cette architecture démontre que l'intégration de Python, OpenCV, Face Recognition et Flask permet de construire un système intelligent, scalable et efficace pour la reconnaissance faciale en temps réel

2.Système de reconnaissance faciale dans un cloud réel:

2.1 Introduction:

Après avoir passé en revue les concepts théoriques relatifs au cloud computing et aux techniques de reconnaissance faciale dans les chapitres précédents, ce chapitre se concentre sur l'aspect applicatif du projet à travers la conception et la mise en œuvre d'un système intégré de reconnaissance faciale au sein d'un environnement de cloud computing réel.

Le système proposé vise à tirer parti des possibilités offertes par le cloud computing en termes de flexibilité, d'accessibilité et de mise à disposition des ressources informatiques, en transférant les opérations de traitement lourd liées à la reconnaissance faciale des appareils des utilisateurs vers un serveur cloud dédié. Cette approche permet de réduire la charge sur les appareils locaux et d'améliorer les performances du système, en particulier lors du traitement d'images et de l'extraction de caractéristiques faciales.

La plateforme PythonAnywhere a été adoptée comme environnement d'hébergement cloud réel pour déployer l'application et la faire fonctionner sur le réseau Internet, car cette plateforme fournit un environnement approprié pour exécuter des applications Python et gérer les services réseau de manière stable et sécurisée. Une interface de programmation (API) a également été développée à l'aide du framework Flask pour assurer la communication entre le client et le serveur selon le

modèle Client-Serveur.

Ce chapitre présente en détail l'architecture du système proposé, les outils et technologies utilisés pour son développement, ainsi que l'explication des différentes unités composant le système et leur mécanisme de fonctionnement, puis la clarification des étapes de mise en œuvre et des expériences menées pour vérifier l'efficacité de la solution proposée.

2.2 Architecture du système:

2.2.1 Aperçu général de l'architecture:

Le système a été conçu selon une architecture distribuée basée sur le modèle client-serveur, l'un des modèles les plus utilisés dans les applications de cloud computing moderne.

Dans ce modèle, les fonctions du système sont séparées en deux parties principales : une partie utilisateur représentée par le client, et une autre partie responsable du traitement et du stockage des données représentée par le serveur cloud. Cette séparation permet d'assurer l'indépendance entre les composants du système et de faciliter les opérations de maintenance et de développement futur.

Le mécanisme de fonctionnement repose sur le client qui capture les images et les envoie au serveur cloud via Internet, où les opérations de détection des visages, d'extraction des caractéristiques et de comparaison

avec la base de données située sur le serveur sont effectuées. Après la fin du traitement, les résultats sont envoyés au client pour les afficher à l'utilisateur.

Cette architecture se caractérise par plusieurs avantages, dont les plus importants sont :

Réduction de la consommation des ressources locales

Possibilité d'accéder au système de n'importe où

Facilité de mise à jour du système sans avoir besoin de le réinstaller chez les utilisateurs

Évolutivité pour accueillir un plus grand nombre d'utilisateurs à l'avenir

Centralisation de la gestion des données et des modèles utilisés pour la reconnaissance faciale

Le système se compose de quatre couches principales interconnectées, comme illustré dans la figure architecturale du système :

Couche client

Couche de communication

Couche de traitement cloud

Couche de stockage

2.2.2 Couche client:

Cette couche représente le point d'interaction direct entre l'utilisateur et le

système. Elle a été développée sous forme d'un programme en Python nommé `client.py`.

La mission principale de cette couche est de collecter les données brutes et de les envoyer au serveur cloud sans effectuer d'opérations de traitement complexes localement.

Les fonctions principales de cette couche incluent :

Capturer des images ou des trames vidéo depuis la caméra

Effectuer des opérations préliminaires sur l'image si nécessaire

Compresser les données et les préparer pour l'envoi via le réseau

Envoyer les images au serveur en utilisant des requêtes HTTP

Recevoir les résultats de la reconnaissance depuis le serveur

Afficher l'identité de la personne détectée ou un message indiquant qu'elle n'est pas reconnue

Cette conception contribue à rendre l'application légère et facile à exécuter, même sur des appareils aux capacités limitées.

2.2.3 Couche de communication:

Cette couche représente l'intermédiaire responsable de l'échange de données entre le client et le serveur via le réseau Internet.

Le système repose sur le protocole HTTP comme l'un des protocoles les plus utilisés dans les applications web modernes, où les images sont

envoyées via des requêtes POST à une interface de programmation (REST API) développée avec Flask.

Les fonctions les plus importantes de cette couche sont les suivantes :

Transférer les images du client vers le serveur

Transférer les résultats du traitement du serveur vers le client

Assurer l'indépendance des différents composants du système

Fournir un canal de communication unifié et facile à étendre

Supporter le fonctionnement à distance via Internet

Cette couche est considérée comme un facteur essentiel pour concrétiser le concept du cloud computing en reliant directement l'utilisateur aux ressources de traitement situées sur le serveur cloud.

2.2.4 Couche de traitement cloud:

Cette couche est le composant le plus important du système, car elle contient toutes les opérations d'intelligence artificielle et de traitement d'images.

Cette couche a été déployée sur la plateforme PythonAnywhere. Elle reçoit les images envoyées par les clients pour les traiter et exécuter les différentes étapes de la reconnaissance faciale.

Le processus de traitement passe par les étapes suivantes :

Première étape : Réception de l'image : Le serveur reçoit l'image envoyée par le client et vérifie l'intégrité des données reçues.

Deuxième étape : Détection des visages : Les algorithmes de détection de visages sont appliqués pour localiser les positions des visages dans l'image et extraire la région de chaque visage.

Troisième étape : Extraction des caractéristiques : Après avoir localisé le visage, il est converti en une représentation numérique appelée "Face Encoding", qui est un vecteur numérique représentant les caractéristiques uniques du visage.

Quatrième étape : Comparaison : Le vecteur résultant est comparé avec les vecteurs stockés précédemment dans la base de données des personnes connues.

Cinquième étape : Prise de décision : Sur la base des résultats de la comparaison, l'identité de la personne est déterminée ou elle est classée comme personne inconnue.

Sixième étape : Retour du résultat : Le résultat final est envoyé au client au format JSON pour l'afficher à l'utilisateur.

Ce mécanisme permet d'exploiter les ressources informatiques du serveur au lieu de consommer les ressources de l'appareil local.

2.2.5 Couche de stockage:

Cette couche est responsable du stockage des données nécessaires au fonctionnement du système et à la garantie de la continuité de son fonctionnement.

Les données stockées sont divisées en deux types principaux :

Base de données des visages connus : Contient les images des personnes autorisées ou préalablement enregistrées dans le système, ainsi que les vecteurs numériques extraits de ces images qui sont utilisés lors du processus de correspondance.

Base de données des visages inconnus : Le système sauvegarde les images des personnes qui n'ont pas été reconnues dans un dossier dédié, ce qui permet de les consulter ultérieurement et de les ajouter à la base de données si nécessaire.

Cette couche contribue à améliorer la gestion des données et à faciliter les opérations de mise à jour et d'expansion future du système.

ARCHITECTURE DU SYSTÈME DE RECONNAISSANCE FACIALE BASÉ SUR LE CLOUD

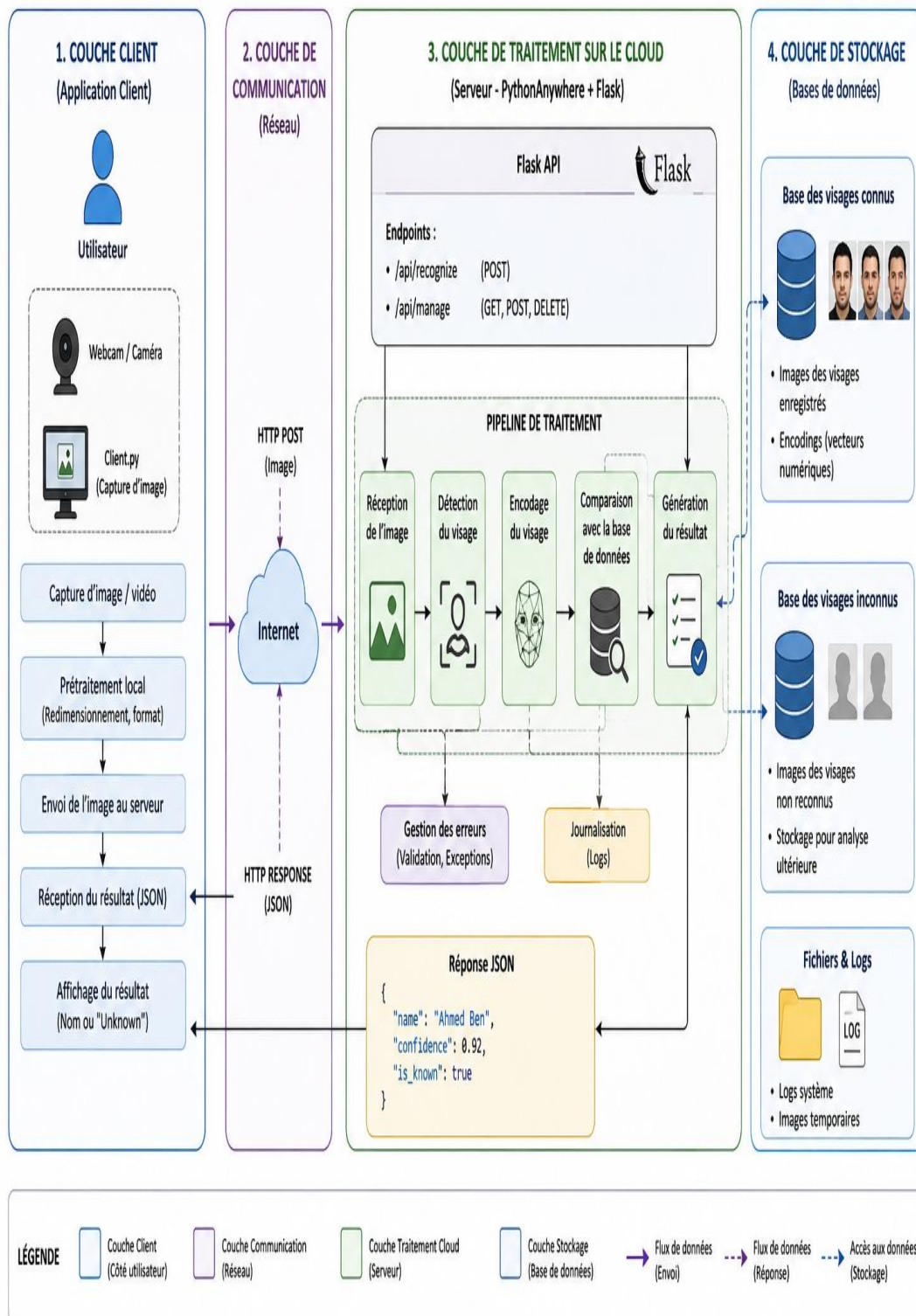


Figure 10: Architecture du système

2.3 Outils et technologies utilisés:

Le système proposé repose sur un ensemble d'outils et de technologies logicielles soigneusement choisis pour répondre aux exigences du projet et garantir l'efficacité des opérations de reconnaissance faciale dans l'environnement cloud. Cette section présente les différents outils utilisés et leur rôle au sein du système.

2.3.1 Langage de programmation Python:

Python est le langage principal utilisé pour le développement du système, en raison de sa facilité d'apprentissage, sa flexibilité et son grand nombre de bibliothèques spécialisées dans les domaines de l'intelligence artificielle, du traitement d'images et du développement d'applications web.

Python se caractérise également par sa capacité à intégrer plusieurs technologies différentes au sein d'un même projet, ce qui en a fait un choix approprié pour développer tous les composants du système, tant au niveau du client que du serveur.

Les raisons les plus importantes de ce choix sont :

Facilité d'écriture et de maintenance du code

Disponibilité de bibliothèques avancées pour le traitement d'images et l'apprentissage profond

Large soutien de la communauté scientifique et de recherche

Possibilité d'exécuter les applications sur différents systèmes d'exploitation

2.3.2 Bibliothèque OpenCV:

OpenCV est l'une des bibliothèques open source les plus célèbres utilisées dans le domaine de la vision par ordinateur, fournissant un large ensemble d'outils et d'algorithmes pour le traitement d'images et de vidéos.

Cette bibliothèque a été utilisée dans le projet pour :

Capturer des images depuis la caméra de l'appareil

Traiter les trames reçues

Convertir différents formats d'image

Améliorer la qualité des images avant de les envoyer au serveur

OpenCV contribue à faciliter les opérations de manipulation des images de manière rapide et efficace.

2.3.3 Bibliothèque Face Recognition:

Cette bibliothèque représente l'élément principal responsable des opérations de reconnaissance faciale au sein du système.

La bibliothèque repose sur des modèles d'apprentissage profond avancés capables d'extraire les caractéristiques distinctives de chaque visage et de les convertir en vecteurs numériques pouvant être comparés mathématiquement.

La bibliothèque fournit un ensemble de fonctions importantes telles que :

La détection des visages dans les images

L'extraction des caractéristiques faciales

Le calcul du degré de similarité entre les visages

La reconnaissance de l'identité des personnes

Elle a été choisie en raison de sa grande précision et de sa facilité d'intégration avec Python.

2.3.4 Framework Flask:

Flask est un framework léger pour le développement d'applications web et d'interfaces de programmation (API).

Il a été utilisé pour créer le serveur cloud chargé de recevoir les images provenant du client, de les traiter et de retourner les résultats.

Parmi les principaux avantages qu'il offre :

Facilité de développement et de déploiement

Faible consommation de ressources

Support de la construction d'API REST

Possibilité de l'intégrer facilement avec les bibliothèques d'intelligence artificielle

2.3.5 Plateforme PythonAnywhere:

PythonAnywhere a été adoptée comme plateforme d'hébergement cloud réel pour déployer l'application sur Internet.

Cette plateforme fournit un environnement prêt à exécuter des applications Python sans avoir besoin de configurer des serveurs complexes.

Son utilisation a contribué à :

Rendre le système accessible via Internet

Tester l'application dans un environnement cloud réel

Gérer facilement les fichiers et les bases de données

Garantir la continuité de fonctionnement du serveur

La plateforme permet également d'accéder à l'application depuis n'importe quel appareil connecté à Internet.

2.3.6 Bibliothèque Pillow:

La bibliothèque Pillow est utilisée pour traiter les images numériques et effectuer certaines opérations auxiliaires avant ou après l'étape de

reconnaissance.

Ses fonctions incluent :

Ouvrir et lire les images

Redimensionner les images

Convertir différents formats

Sauvegarder les images résultantes

Cette bibliothèque aide à améliorer la gestion des fichiers image au sein du système.

2.3.7 Format JSON:

Le système repose sur JSON comme format standard pour l'échange de données entre le client et le serveur.

Ce format a été choisi pour ses caractéristiques suivantes :

Simplicité et facilité de lecture

Rapidité de transfert sur les réseaux

Compatibilité avec la plupart des langages de programmation

Les données échangées incluent :

Les noms des personnes détectées

Les résultats de correspondance

Les messages d'erreur

Les informations sur l'état du système

2.4 Conception du système:

La conception du système vise à garantir les plus hauts niveaux d'efficacité et de précision lors du processus de reconnaissance faciale, en divisant les fonctions en unités indépendantes qui coopèrent entre elles pour exécuter les différentes étapes du traitement.

Cette division aide à faciliter la maintenance et le développement futur, et permet également d'ajouter de nouvelles fonctionnalités sans affecter les autres parties du système.

2.4.1 Module d'authentification:

Ce module représente la première ligne de défense du système, visant à protéger le panneau d'administration et à empêcher tout accès non autorisé.

Le module repose sur un système de connexion basé sur le nom d'utilisateur et le mot de passe, et la validité des données est vérifiée avant d'autoriser l'accès aux fonctions administratives.

Ce module offre :

La protection des données sensibles

Le contrôle des autorisations des utilisateurs

La prévention des accès non autorisés

2.4.2 Module de gestion des visages connus:

Ce module permet de gérer la base de données des personnes enregistrées dans le système.

Il offre les fonctions suivantes :

Ajouter de nouvelles personnes

Supprimer les personnes enregistrées

Mettre à jour les images

Afficher la liste des personnes stockées

Ce module est essentiel pour maintenir une base de données à jour et précise.

2.4.3 Module de détection des visages:

Ce module représente la première étape du processus de reconnaissance.

Lors de la réception d'une nouvelle image, le système recherche les visages qu'elle contient et détermine les coordonnées de chaque visage.

L'importance de cette étape réside dans l'isolation du visage du reste des éléments de l'image pour réduire l'influence du bruit et des différents arrière-plans.

2.4.4 Module d'extraction des caractéristiques:

Après avoir localisé le visage, ses caractéristiques uniques sont extraites.

Ce processus repose sur des modèles d'apprentissage profond qui convertissent l'image en un vecteur numérique contenant des informations décrivant les traits du visage.

Ce vecteur représente l'empreinte numérique de la personne dans le système.

2.4.5 Module de reconnaissance faciale:

Ce module compare le vecteur résultant avec les vecteurs stockés dans la base de données.

Le processus de comparaison repose sur le calcul de la distance mathématique entre les vecteurs.

Plus la distance est petite, plus le degré de similarité est grand.

Si le pourcentage de similarité dépasse le seuil requis, l'identité de la personne est reconnue.

2.4.6 Amélioration de la précision du système en cas de visages similaires et de jumeaux:

Le problème de la distinction entre les personnes ayant une grande similarité est l'un des défis les plus importants auxquels sont confrontés les systèmes de reconnaissance faciale modernes.

Dans certains cas, les distances calculées entre plusieurs personnes peuvent être très proches, ce qui augmente la probabilité de prendre une

décision erronée.

Pour surmonter ce problème, un mécanisme supplémentaire a été développé qui repose sur l'analyse des résultats les plus proches au lieu de se fier uniquement au résultat le plus proche.

Le système calcule toutes les distances entre le visage actuel et tous les visages stockés, puis classe les résultats par ordre croissant et sélectionne les deux meilleures correspondances.

Ensuite, la différence entre elles est calculée, appelée "Confidence Gap" (écart de confiance).

Si la différence entre la meilleure correspondance et la deuxième meilleure correspondance est très petite, le système considère que la décision n'est pas suffisamment fiable et classe le visage comme "Inconnu".

Cette approche contribue à :

Réduire le taux d'erreurs

Augmenter le niveau de fiabilité

Améliorer les performances dans les cas complexes

Augmenter la précision de la reconnaissance des jumeaux et des visages similaires

2.5 Mise en œuvre du système:

Après avoir terminé la phase de conception, on est passé à la phase de mise en œuvre pratique du système.

Le projet a été réalisé selon les étapes suivantes :

Première phase : Développement du serveur cloud : Une application Flask a été créée, chargée de recevoir les images, traiter les requêtes, gérer la base de données et exécuter les opérations de reconnaissance faciale.

Deuxième phase : Développement du programme client : Le programme client.py a été créé pour fonctionner sur l'appareil de l'utilisateur et se charger de capturer les images, les envoyer au serveur, recevoir et afficher les résultats.

Troisième phase : Déploiement du projet dans le cloud : L'application a été déployée sur la plateforme PythonAnywhere et liée à une URL accessible via Internet.

Quatrième phase : Tests : Une série de tests a été effectuée pour vérifier la réussite de la communication entre le client et le serveur, la précision de la reconnaissance faciale, la rapidité de réponse et la stabilité du système pendant le fonctionnement.

2.6 Flux de travail du système:

Le flux de travail au sein du système peut être résumé selon les étapes suivantes :

L'utilisateur exécute le programme client

Une image est capturée depuis la caméra

L'image est envoyée au serveur cloud

Le serveur reçoit l'image et commence à la traiter

Les visages présents dans l'image sont détectés

Les caractéristiques faciales de chaque visage sont extraites

Les résultats sont comparés avec la base de données

La décision finale est prise

Une réponse au format JSON est créée

Le résultat est envoyé au client

Le nom de la personne ou un message "Inconnu" est affiché à l'utilisateur

2.7 Les Diagrammes de systeme:

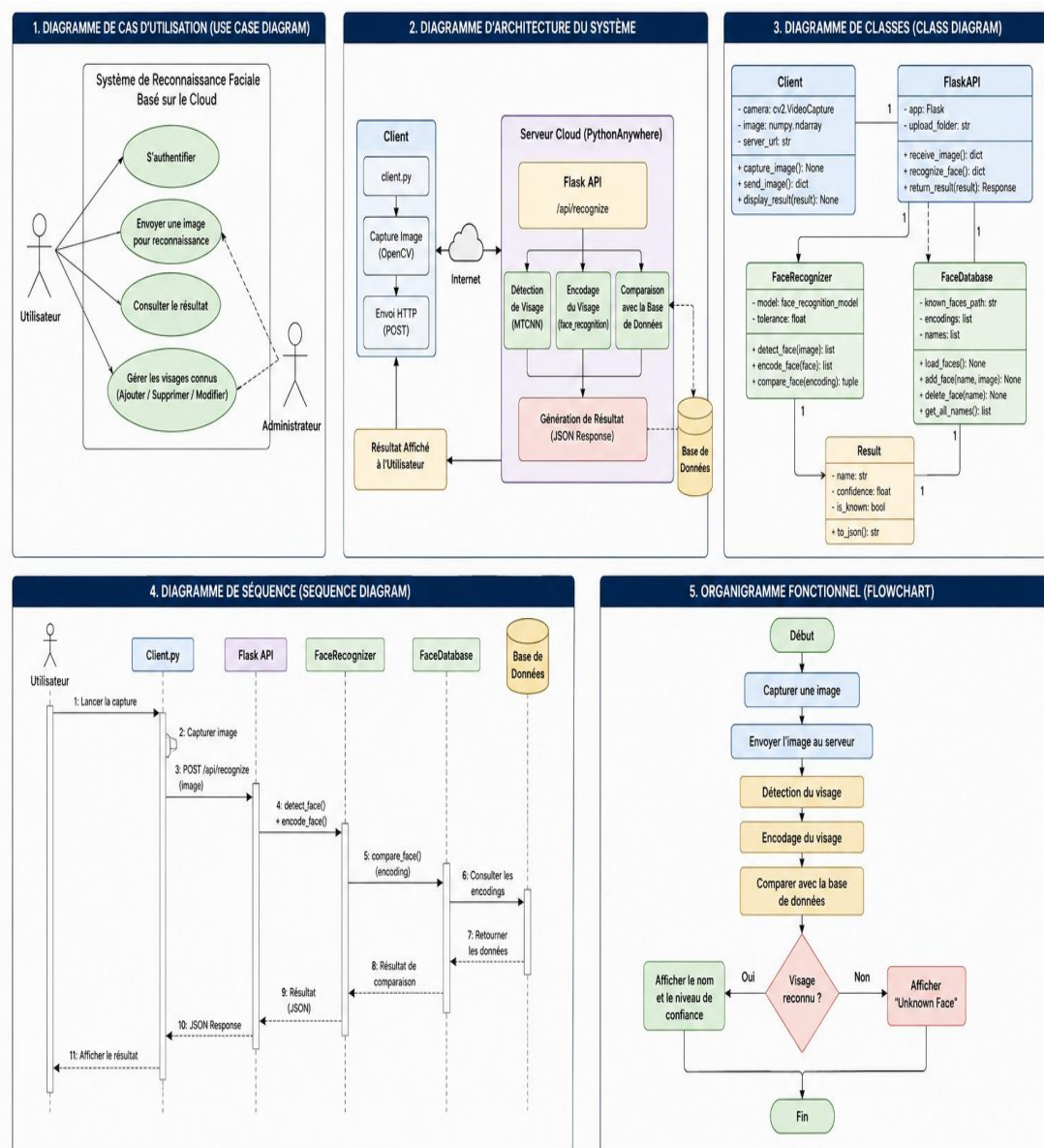


Figure 11 :Les Diagrammes de système

2.8 Conclusion:

Dans ce chapitre, les aspects applicatifs du projet ont été présentés à travers la conception et la mise en œuvre d'un système intégré de

reconnaissance faciale au sein d'un environnement de cloud computing réel. L'architecture générale du système, les outils et technologies utilisés pour son développement ont également été clarifiés, ainsi que l'explication des modules internes et de leur mécanisme de fonctionnement.

Les résultats des tests ont montré que l'adoption du cloud computing a contribué à améliorer les performances du système et à faciliter son accès via Internet, tout en fournissant un environnement approprié pour exécuter les opérations de reconnaissance faciale avec une grande efficacité. Le mécanisme d'amélioration pour le traitement des cas de jumeaux et de visages similaires a également contribué à augmenter le niveau de précision et de fiabilité, rendant le système plus adapté à une utilisation dans des applications réelles qui nécessitent un haut niveau de fiabilité et de sécurité.

3. Système hybride:

3.1 Introduction

Dans ce chapitre, nous présentons la phase de conception et de

réalisation de notre système de reconnaissance faciale basé sur les techniques de Deep Learning et intégré dans un environnement de Cloud Computing hybride.

L'objectif principal de cette approche est de combiner les avantages du traitement local et des services cloud afin d'obtenir un système performant, flexible et facilement évolutif.

L'architecture proposée repose sur deux environnements complémentaires :

Un Cloud Local, utilisé pour l'exécution du backend Flask, le traitement des images et les tests en environnement contrôlé.

Un Cloud Réel, basé sur les services Firebase Firestore et Cloudinary, permettant le stockage distant des données et des images.

Cette architecture hybride offre plusieurs avantages, notamment la rapidité du traitement local, la synchronisation en temps réel des données, ainsi qu'une meilleure scalabilité du système grâce à l'utilisation de services cloud modernes.

Dans ce chapitre, nous détaillons :

l'architecture générale du système hybride ;

les différentes composantes de l'application ;

le fonctionnement détaillé du processus de reconnaissance faciale ;

les technologies utilisées ;

les diagrammes UML représentant la structure du système ;

ainsi que la réalisation pratique et les interfaces développées.

3.2 Architecture Générale du Système:

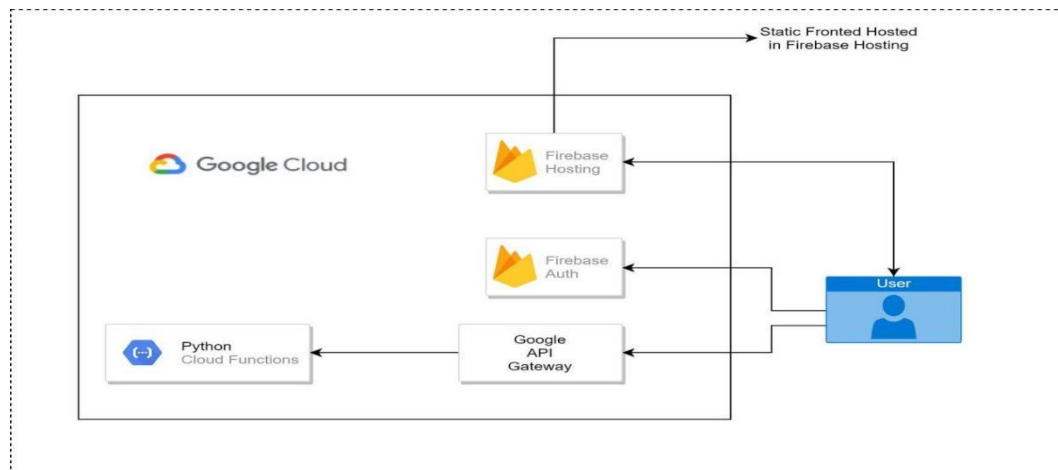


Figure 12: Architecture Générale du Système

3.2.1 Vue d'Ensemble:

Le système développé adopte une architecture distribuée de type Client–Serveur intégrée dans un environnement Cloud Hybride.

Cette architecture est composée de trois couches principales :

Couche Présentation (Frontend)

Couche Traitement (Backend)

Couche Stockage Cloud

L'objectif est de séparer clairement les responsabilités entre l'interface utilisateur, les traitements de reconnaissance faciale et le stockage des données.

Cette séparation facilite la maintenance, l'évolution et le déploiement

futur du système.

3.2.2 Couche Présentation (Frontend):

La couche Frontend représente l'interface web utilisée par l'administrateur.

Elle permet :

l'authentification des utilisateurs ;

l'ajout de nouvelles personnes ;

l'importation des images faciales ;

le lancement de la reconnaissance ;

la consultation des résultats ;

le suivi des présences enregistrées.

Cette interface a été développée à l'aide des technologies HTML, CSS, Bootstrap et JavaScript afin d'offrir une expérience utilisateur simple et intuitive.

3.2.3 Couche Traitement (Backend):

Cette couche constitue le cœur fonctionnel du système.

Elle est développée avec Flask et assure :

le traitement des requêtes HTTP ;

la gestion de l'authentification ;

la communication avec Firebase et Cloudinary ;

l'exécution du modèle de reconnaissance faciale ;

la gestion des présences.

Le backend joue le rôle d'intermédiaire entre l'interface utilisateur et les services cloud.

3.2.4 Couche Stockage Cloud:

Cette couche est responsable du stockage et de la gestion des données.

Elle repose sur deux services cloud complémentaires :

Firebase Firestore

Utilisé pour :

stocker les profils des utilisateurs ;

enregistrer les personnes reconnues ;

conserver les historiques de présence ;

assurer la synchronisation des données.

Cloudinary

Utilisé pour :

stocker les images faciales ;

générer des URLs sécurisées ;

optimiser le stockage des images ;

faciliter l'accès distant aux ressources visuelles.

3.2.5 Architecture Hybride:

L'architecture hybride combine :

Cloud Local

Responsable de :

l'exécution du serveur Flask ;

le calcul des encodages faciaux ;

la comparaison des visages ;

les tests en environnement local.

Cloud Réel

Responsable de :

la persistance des données ;

le stockage sécurisé des images ;

la synchronisation distante ;

la disponibilité des ressources sur Internet.

Cette combinaison permet de bénéficier simultanément des performances du traitement local et de la flexibilité des services cloud.

3.3 Fonctionnement Détaillé du Système:

Le fonctionnement global du système est organisé en plusieurs étapes successives allant de l'authentification de l'utilisateur jusqu'à

l'enregistrement de la présence.

3.3.1 Authentification de l'Administrateur:

Lors du démarrage de l'application, l'administrateur doit s'authentifier à l'aide de son identifiant et de son mot de passe.

Les informations saisies sont vérifiées dans la collection users de Firebase Firestore.

En cas de validation réussie, l'accès au tableau de bord est accordé.

Cette étape garantit la sécurité et le contrôle d'accès au système.

3.3.2 Ajout d'une Nouvelle Personne:

Après authentification, l'administrateur peut enregistrer une nouvelle personne.

Cette opération comprend :

la saisie du nom ;

l'importation de plusieurs images ;

la préparation des données pour le stockage cloud.

L'utilisation de plusieurs images améliore la robustesse du système de reconnaissance.

3.3.3 Stockage des Images dans Cloudinary:

Les images téléchargées sont envoyées vers Cloudinary.

Pour chaque image, le service génère :

une URL sécurisée (secure_url) ;

un identifiant unique (public_id).

Ces informations seront utilisées lors des étapes de reconnaissance.

3.3.4 Enregistrement des Métadonnées dans Firestore:

Les informations associées à chaque personne sont enregistrées dans la collection persons.

Chaque document contient :

le nom de la personne ;

les URLs des images ;

les identifiants Cloudinary.

Cette structure permet d'assurer la liaison entre les données stockées et les traitements réalisés par le système.

3.3.5 Encodage des Visages:

Les images sont récupérées depuis Cloudinary puis analysées localement.

À l'aide des bibliothèques OpenCV et face_recognition, le système :

détecte les visages ;

extraire les caractéristiques faciales ;

génère un vecteur numérique de 128 dimensions (Face Encoding).

Ces vecteurs sont ensuite utilisés comme référence lors de la reconnaissance.

3.3.6 Acquisition d'une Image Inconnue:

Lors de la reconnaissance, l'utilisateur peut :

capturer une image depuis la webcam ;

importer une image depuis un fichier local.

Le visage détecté est ensuite converti en vecteur numérique.

3.3.7 Comparaison et Reconnaissance

Le vecteur obtenu est comparé à tous les encodages enregistrés.

La comparaison repose sur la distance euclidienne calculée à l'aide de NumPy.

Si la distance minimale est inférieure au seuil fixé :

La personne est reconnue.

Sinon :

Le résultat est classé comme "Unknown".

3.3.8 Enregistrement de la Présence:

Lorsqu'une personne est reconnue avec succès, le système ajoute automatiquement une entrée dans la collection attendance.

Chaque enregistrement contient :

le nom ;

la date ;

l'heure de reconnaissance.

Ces données servent à produire les statistiques de présence.

3.3.9 Avantages de l'Architecture Hybride:

L'architecture proposée présente plusieurs avantages :

Réduction du temps de traitement.

Amélioration des performances.

Scalabilité élevée.

Synchronisation en temps réel.

Sécurité renforcée des données.

Maintenance simplifiée.

Possibilité d'évolution future du système.

3.4 Technologies Utilisées:

Le développement du système de reconnaissance faciale repose sur un ensemble de technologies complémentaires combinant l'intelligence

artificielle, le traitement d'images, le développement web et les services cloud. Cette combinaison permet de construire une architecture hybride performante, évolutive et adaptée aux besoins des systèmes modernes de reconnaissance faciale.

3.4.1 Python:

Python constitue le langage principal utilisé dans le développement du système. Il a été choisi en raison de sa simplicité, de sa lisibilité et de la richesse de son écosystème logiciel, notamment dans les domaines de l'intelligence artificielle et du traitement d'images.

Dans ce projet, Python assure plusieurs fonctions essentielles :

Développement du backend de l'application ;

Intégration des bibliothèques de reconnaissance faciale ;

Communication avec les services cloud ;

Traitement des données et gestion des requêtes.

Grâce à sa flexibilité, Python facilite également l'intégration de nouvelles fonctionnalités et l'évolution future du système.

3.4.2 Flask:

Flask est un micro-framework web léger développé en Python. Il a été utilisé pour implémenter le serveur backend de l'application.

Son rôle principal consiste à :

Gérer les requêtes HTTP ;

Fournir les interfaces API du système ;

Assurer la communication entre le frontend et les services cloud ;

Contrôler les opérations d'authentification et de reconnaissance faciale.

Flask présente plusieurs avantages tels que sa simplicité d'utilisation, sa modularité et son faible coût en ressources système.

3.4.3 Face Recognition (Dlib):

La bibliothèque Face Recognition constitue le noyau principal du système de reconnaissance faciale.

Elle repose sur les modèles de Deep Learning fournis par Dlib et permet :

La détection automatique des visages ;

L'extraction des caractéristiques faciales ;

La génération des encodages faciaux ;

La comparaison entre plusieurs visages.

Chaque visage est représenté par un vecteur numérique de 128 dimensions appelé Face Encoding, utilisé pour mesurer le degré de similarité entre deux individus.

Cette bibliothèque offre une précision élevée tout en restant relativement simple à intégrer dans les applications Python.

3.4.4 OpenCV:

OpenCV (Open Source Computer Vision Library) est une bibliothèque spécialisée dans la vision par ordinateur et le traitement d'images.

Dans notre projet, elle est utilisée pour :

Capturer les images à partir de la webcam ;

Lire et manipuler les fichiers image ;

Convertir les formats d'image ;

Afficher les résultats de reconnaissance ;

Dessiner les cadres et les informations sur les visages détectés.

OpenCV joue un rôle fondamental dans la gestion des flux vidéo et l'interaction visuelle avec l'utilisateur.

3.4.5 Firebase Firestore:

Firebase Firestore est une base de données NoSQL hébergée dans le cloud et développée par Google.

Elle est utilisée pour stocker :

Les comptes utilisateurs ;

Les informations des personnes enregistrées ;

Les historiques de présence ;

Les métadonnées des images.

L'utilisation de Firestore offre plusieurs avantages :

Synchronisation en temps réel ;

Haute disponibilité ;

Scalabilité automatique ;

Sécurité intégrée ;

Facilité d'intégration avec les applications web.

Cette solution permet au système de conserver les données de manière fiable et accessible à distance.

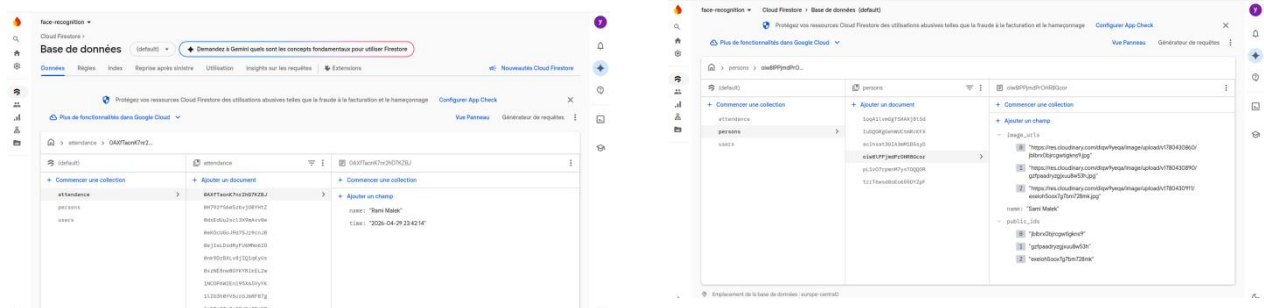


Figure 13: capture Firebase Firestore (Cloud Database)

3.4.6 Cloudinary:

Cloudinary est une plateforme cloud spécialisée dans le stockage et la gestion des ressources multimédias.

Dans ce projet, elle est utilisée pour :

Héberger les images des visages ;

Générer des liens sécurisés ;

Optimiser automatiquement les images ;

Réduire la charge de stockage locale.

Lors de l'ajout d'une nouvelle personne, les images sont transférées vers

Cloudinary puis référencées dans Firebase Firestore à travers leurs URLs et identifiants uniques.

Cette approche permet une meilleure organisation des données et une gestion efficace des ressources visuelles.

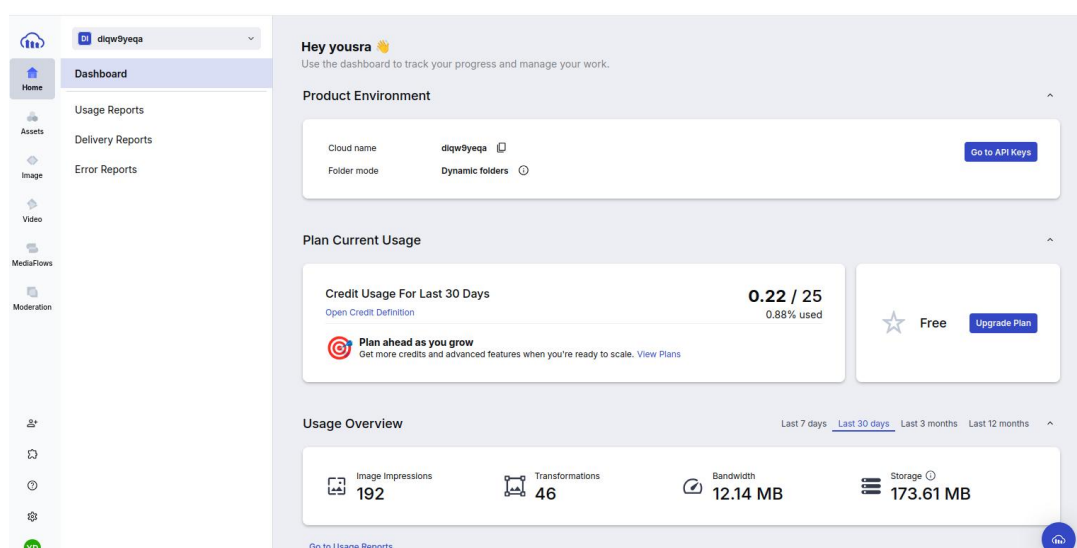


Figure 14:capture de Stockage d'images dans un Cloudinary

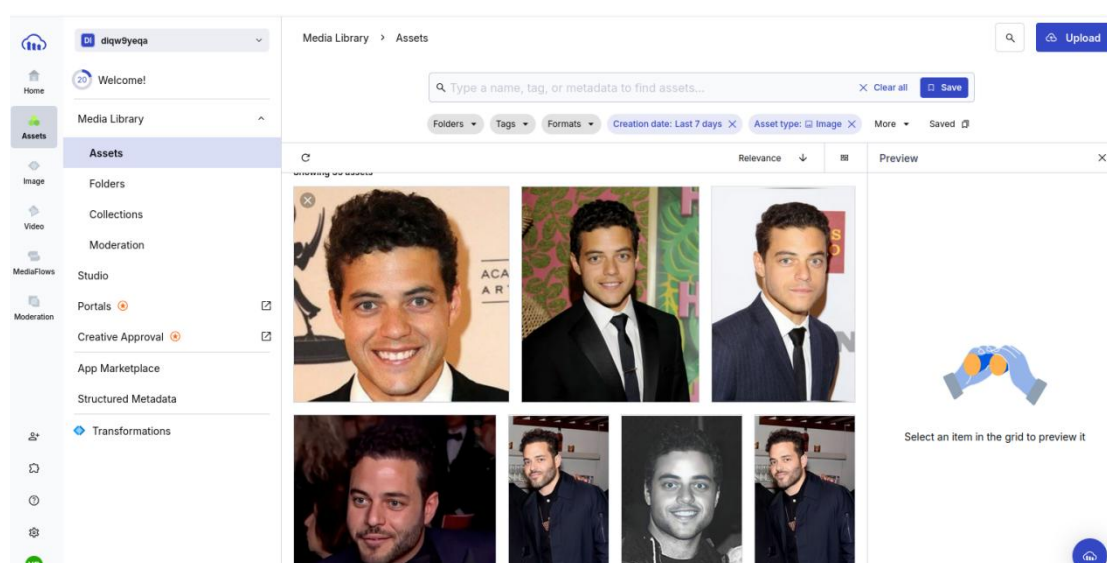


Figure 15:capture de Stockage d'images dans un Cloudinary

3.4.7 NumPy:

NumPy est une bibliothèque scientifique destinée au calcul numérique haute performance.

Elle intervient principalement dans :

La manipulation des vecteurs faciaux ;

Le calcul des distances entre les encodages ;

L' exécution des opérations mathématiques nécessaires à la reconnaissance ;

Le tri et l'analyse des résultats.

Grâce à son optimisation interne, NumPy permet d'effectuer rapidement des calculs sur de grands ensembles de données.

3.4.8 HTML5:

HTML5 constitue le langage de structuration des pages web de l'application.

Il permet :

La création des formulaires ;

L'organisation des interfaces ;

L'affichage des données utilisateur ;

L'intégration des composants visuels.

Il représente la base de l'interface graphique du système.

3.4.9 Bootstrap:

Bootstrap est un framework CSS utilisé pour améliorer l'apparence et l'ergonomie de l'interface utilisateur.

Ses principales contributions sont :

La création d'interfaces modernes ;

L'adaptation automatique aux différentes tailles d'écran ;

La mise en forme rapide des composants ;

L'amélioration de l'expérience utilisateur.

Grâce à Bootstrap, l'application bénéficie d'une interface professionnelle et responsive.

3.4.10 JavaScript:

JavaScript assure les fonctionnalités dynamiques du frontend.

Il permet notamment :

L'envoi des requêtes vers le serveur Flask ;

La communication avec Firebase ;

La gestion des événements utilisateur ;

La mise à jour dynamique des pages ;

L'affichage instantané des résultats.

L'utilisation de JavaScript améliore considérablement l'interactivité et la fluidité de l'application.

3.4.11 Cloud Local:

Le Cloud Local représente l'environnement d'exécution local utilisé durant les phases de développement et de test.

Il héberge :

Le serveur Flask ;

Les traitements de reconnaissance faciale ;

Les opérations d'encodage ;

Les expérimentations et validations du système.

Cette approche permet de réduire les temps de réponse et de faciliter le débogage avant le déploiement final.

3.4.12 Architecture Hybride Cloud:

L'une des particularités de ce projet réside dans l'adoption d'une architecture hybride combinant les ressources locales et les services cloud.

Cette architecture permet :

D'effectuer les calculs intensifs localement ;

De stocker les données dans le cloud ;

D'assurer la synchronisation en temps réel ;

D'améliorer la disponibilité des ressources ;

De faciliter la montée en charge du système.

L'intégration conjointe de Flask, Firebase Firestore et Cloudinary permet ainsi de bénéficier des avantages du Cloud Computing tout en conservant les performances du traitement local.

3.5 Diagrammes UML :

Dans cette section, nous présentons les diagrammes UML du système :

3.5.1 Diagramme de cas d'utilisation :

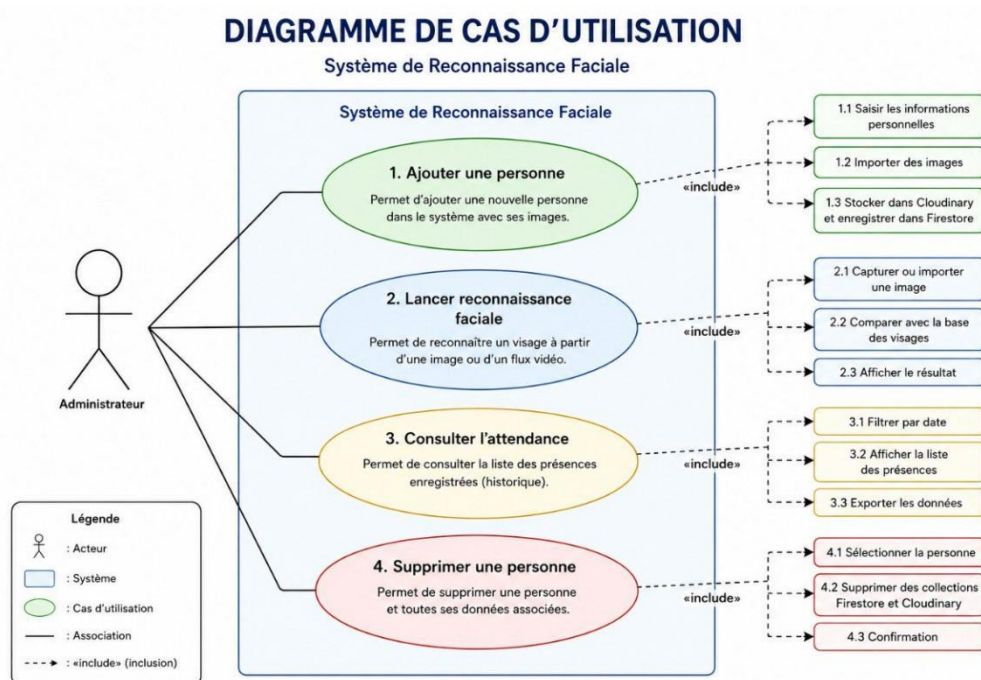


Figure 16:Diagramme de cas d'utilisation

3.5.2 Diagramme de classes:

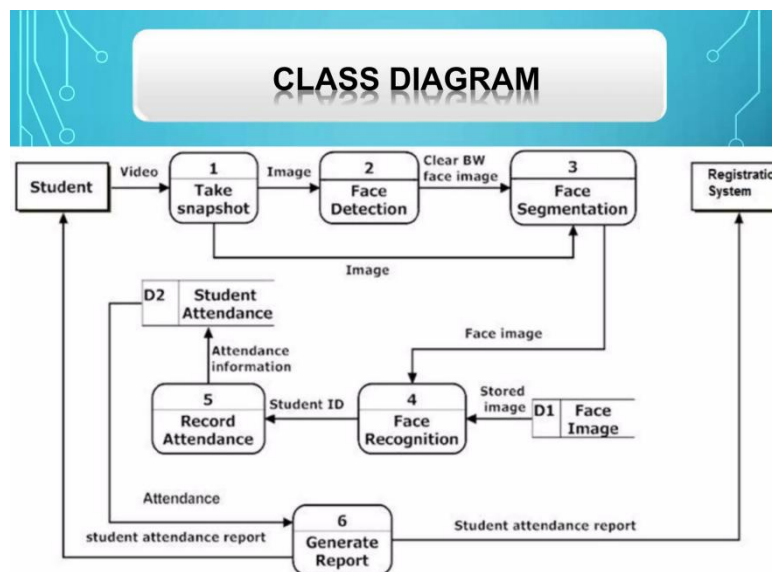


Figure 17:Diagramme de classes

3.5.3 Diagramme de séquence:

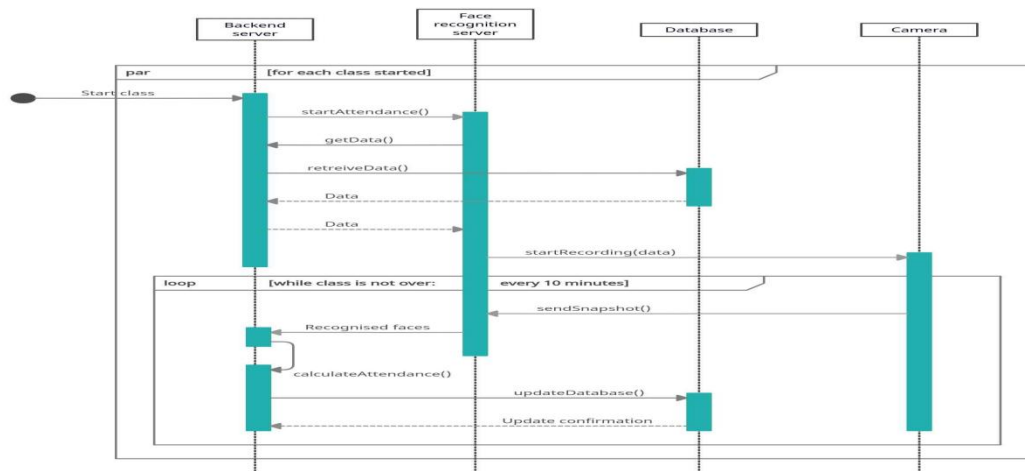


Figure 18:Diagramme de séquence

3.6 Réalisation et captures d'écran :

Le système a été implémenté sous forme d'une application web fonctionnelle.

3.6.1 Interface Dashboard :

Affiche :

nombre total de personnes

présence du jour

statut du système

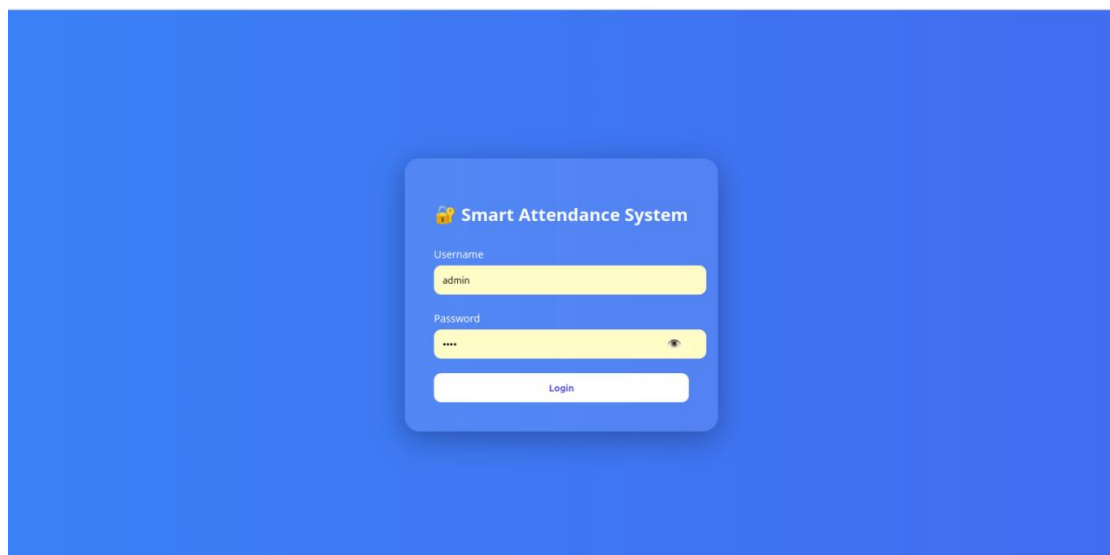


Figure 19:capture d'écran de l'interface de connexion utilisateur

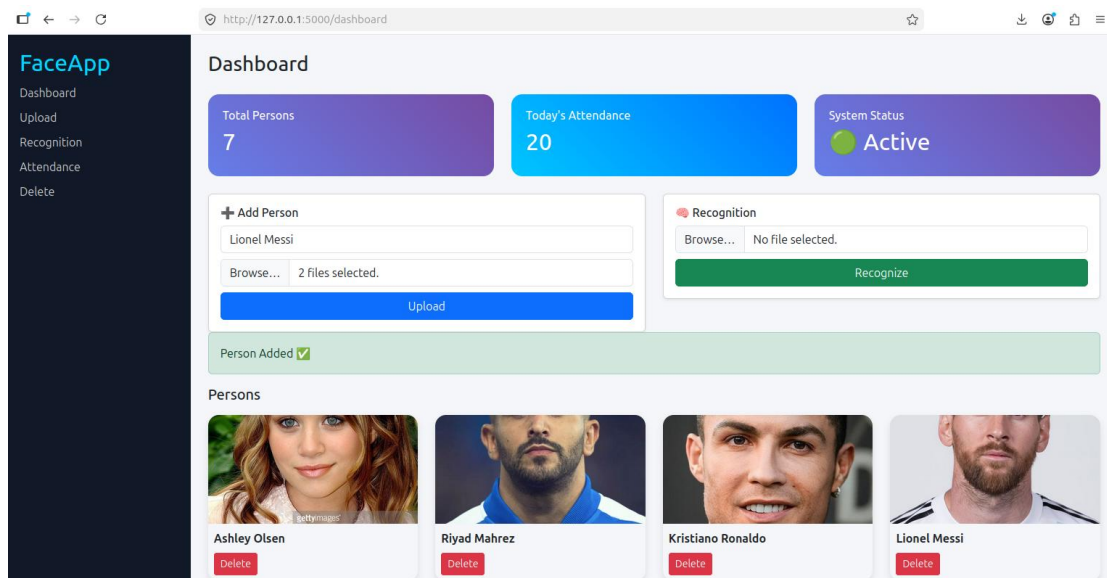


Figure 20:capture d'interface dashboard

3.6.2 Ajout de personne :

Permet l'ajout de plusieurs images par utilisateur avec stockage cloud.

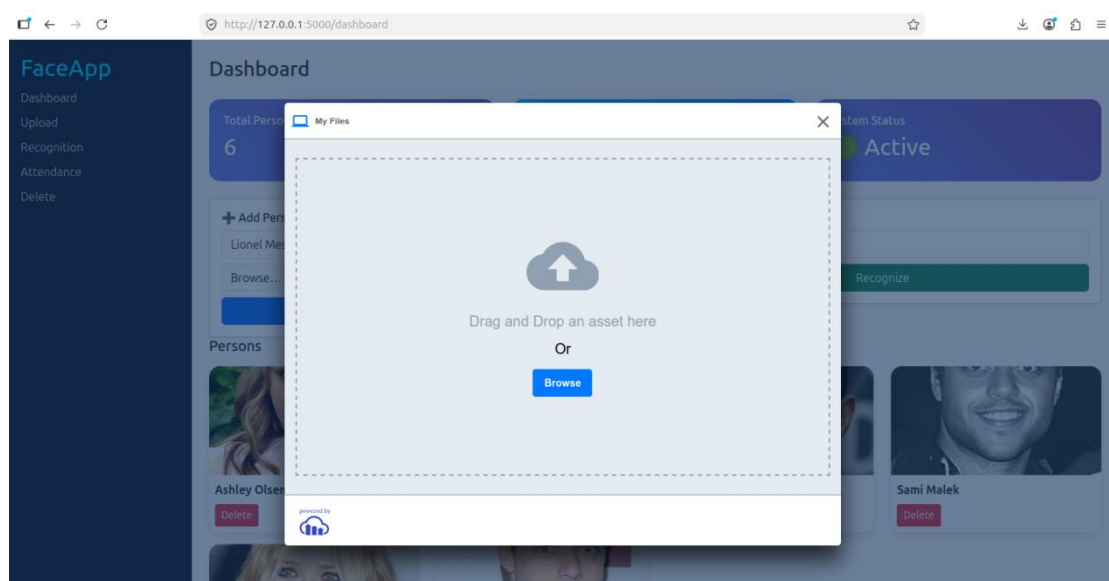


Figure 21:capture d'écran 1 du téléchargement des images d'une personne

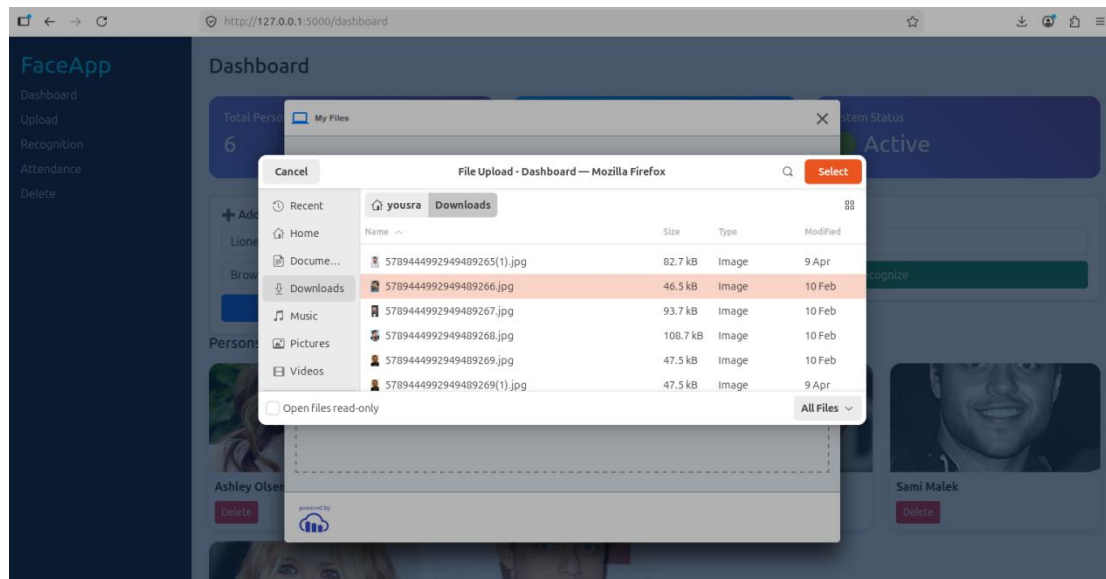


Figure 22:capture d'écran 2 du téléchargement des images d'une personne

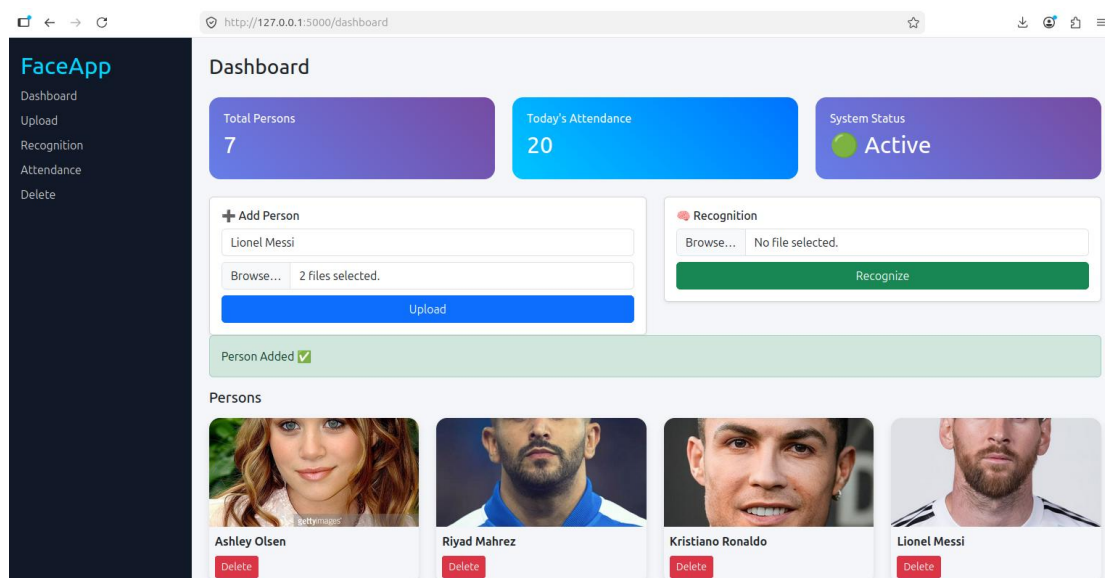


Figure 23:capture d'écran d'une personne téléchargeant

3.6.3 Reconnaissance faciale :

Le système identifie la personne ou retourne “ Unknown” en cas d’absence de correspondance.

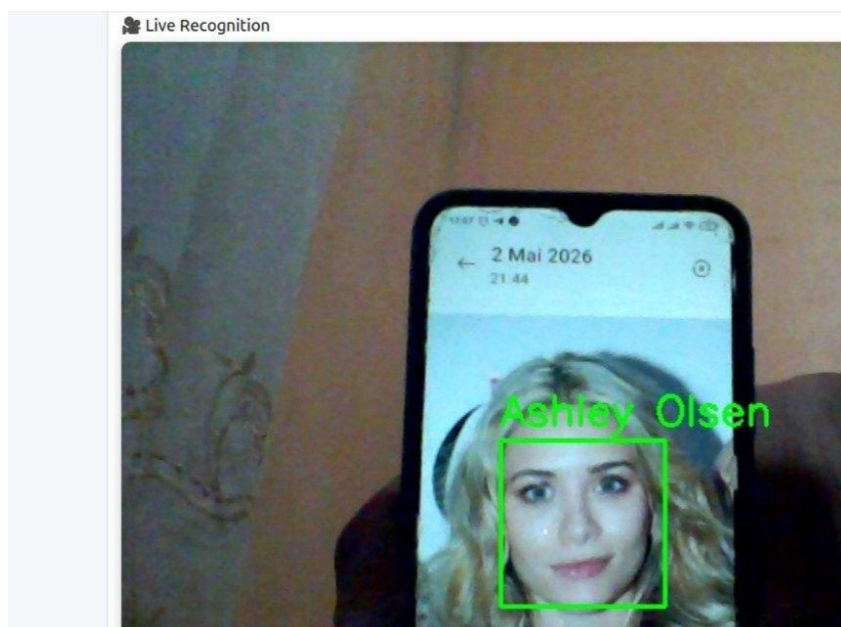


Figure 24:capture d’écran de reconnaissance faciale1

3.6.4 Gestion de l’assiduité :

Affichage des logs d’entrée avec date et heure.

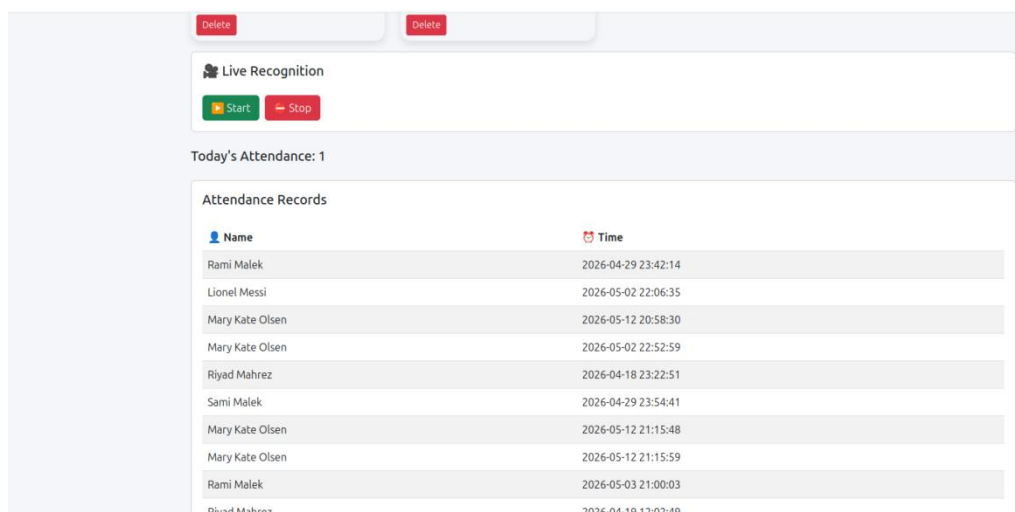


Figure 25:capture d’écran d’affichage des logs d’entrée 1

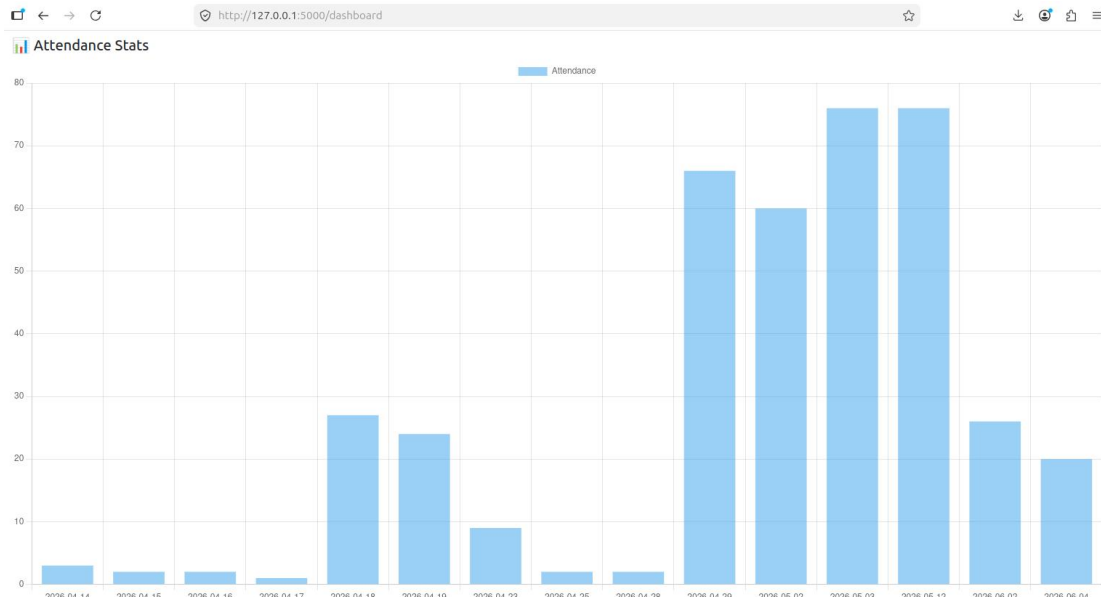


Figure 26:capture d'écran d'affichage des logs d'entrée 2

4.Comparaison des approches proposées:

Afin d'évaluer les différentes solutions développées dans ce projet, une comparaison a été réalisée entre le système basé sur un cloud local et le système déployé sur un cloud réel. Cette étude permet d'identifier les avantages et les limites de chaque approche, ainsi que l'intérêt de la solution hybride proposée.

Critère	Cloud Local	Cloud Réel	Système Hybride
Environnement d'exécution	Machine virtuelle locale	PythonAnywhere	Local + Cloud réel
Accessibilité	Réseau local uniquement	Accessible via Internet	Accessible localement et à distance
Dépendance à Internet	Faible	Élevée	Moyenne
Temps de réponse	Rapide	Plus lent à cause du réseau	Optimisé
Scalabilité	Limitée	Élevée	Élevée
Coût	Faible	Moyen	Moyen
Maintenance	Simple	Centralisée	Plus complexe
Sécurité	Contrôlée localement	Assurée par le fournisseur cloud	Renforcée
Gestion des utilisateurs	Locale	Centralisée	Centralisée
Reconnaissance des jumeaux	Non optimisée	Mécanisme amélioré (Confidence Gap)	Optimisée

L'analyse montre que le système basé sur un cloud local constitue une bonne solution pour les phases de développement et de test. Il offre une faible latence et un contrôle total de l'environnement, mais reste limité en termes d'accessibilité et d'évolutivité.

Le système déployé sur un cloud réel apporte plusieurs améliorations importantes. Il permet un accès à distance via Internet, une gestion centralisée des données et une meilleure évolutivité. De plus, l'intégration du mécanisme « Confidence Gap » améliore la précision de reconnaissance dans les cas complexes tels que les jumeaux ou les personnes ayant des caractéristiques faciales très similaires.

Enfin, le système hybride combine les avantages des deux approches. Les traitements rapides peuvent être réalisés localement tandis que les

ressources du cloud réel sont utilisées pour les opérations plus exigeantes. Cette architecture permet d'obtenir un meilleur compromis entre performance, flexibilité et fiabilité.

5. Tests et Résultats:

Critère	Local	Réel	Hybride
Temps réponse	Excellent	Moyen	Très bon
Accès Internet	Non	Oui	Oui
Stockage Cloud	Non	Oui	Oui
Scalabilité	Faible	Élevée	Élevée
Maintenance	Simple	Moyenne	Moyenne

5.1 Captures d'écran:

5.1.1 Cloud Local:

Login:

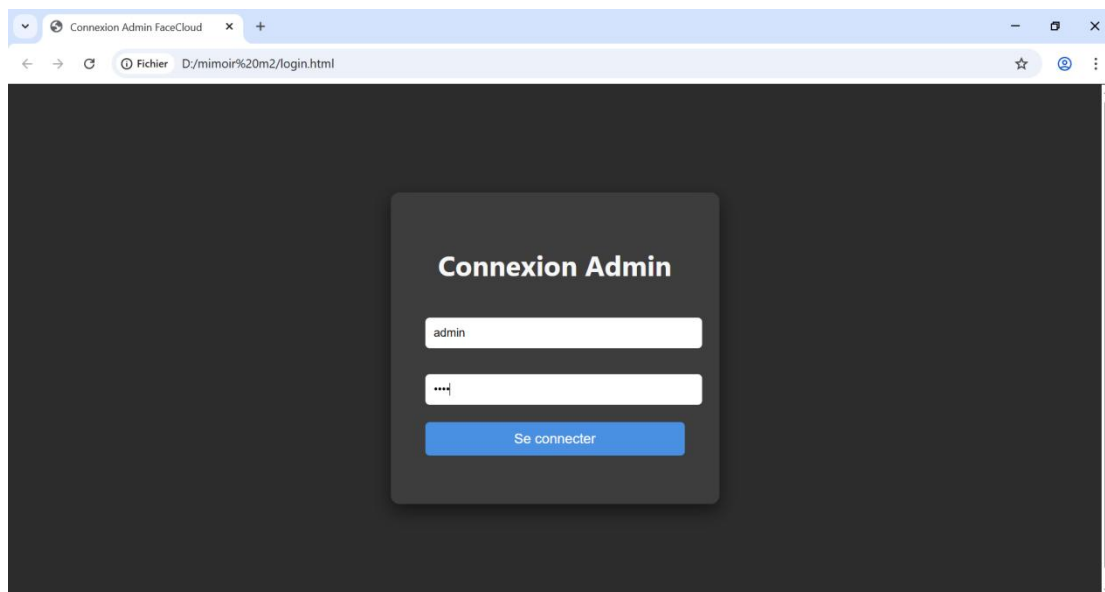


Figure 27: login cloud local

Dashboard:

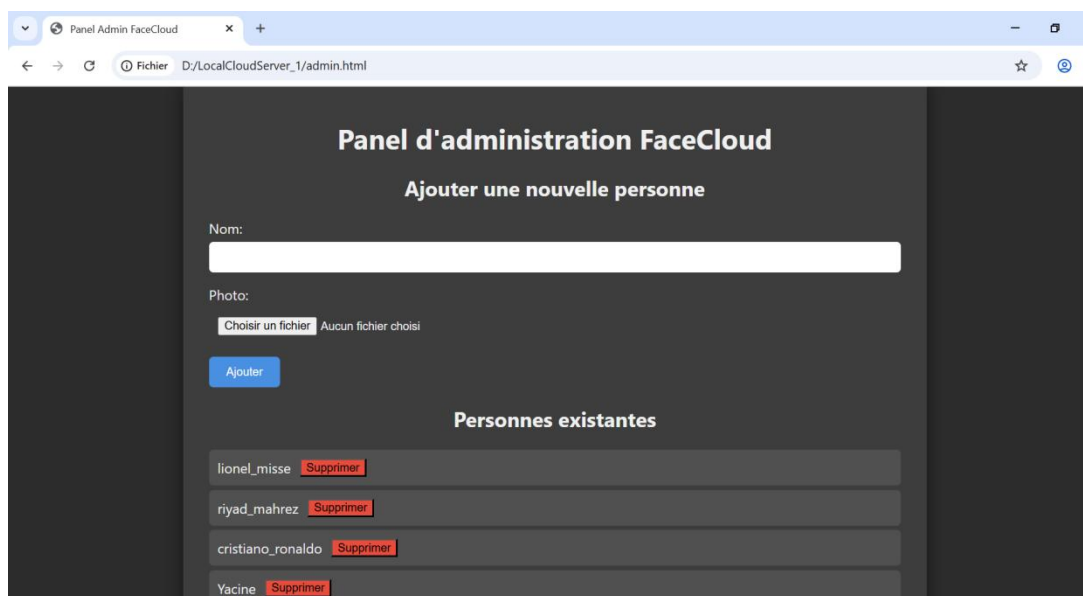


Figure 28:Dashboard cloud local 1

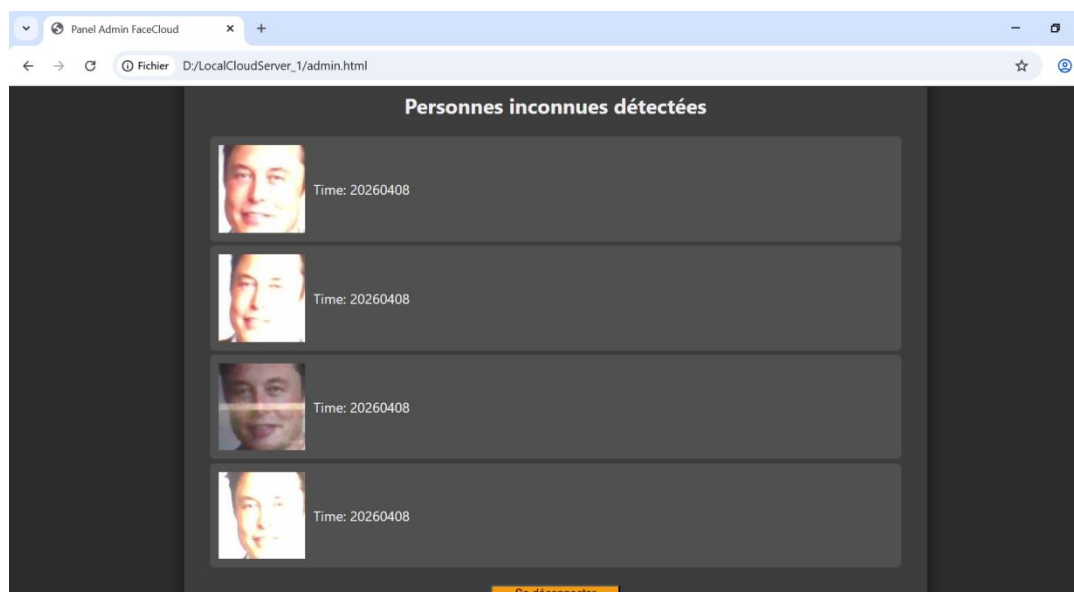


Figure 29:Dashboard cloud local 1

Ajout Personne:

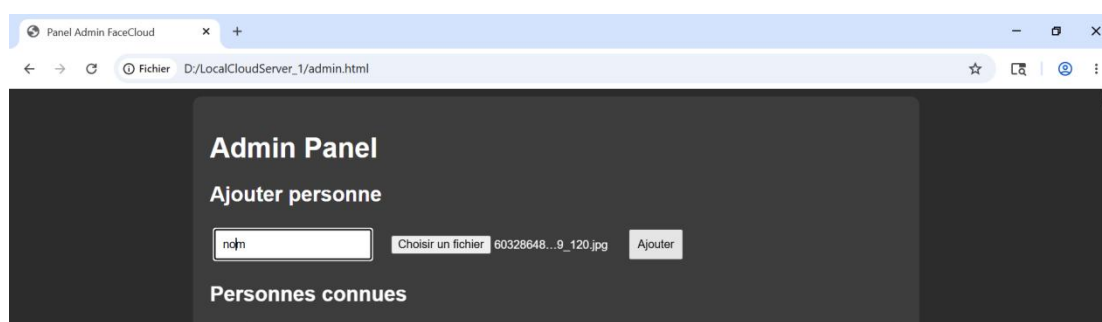


Figure 30:Ajout Personne cloud local

Reconnaissance:

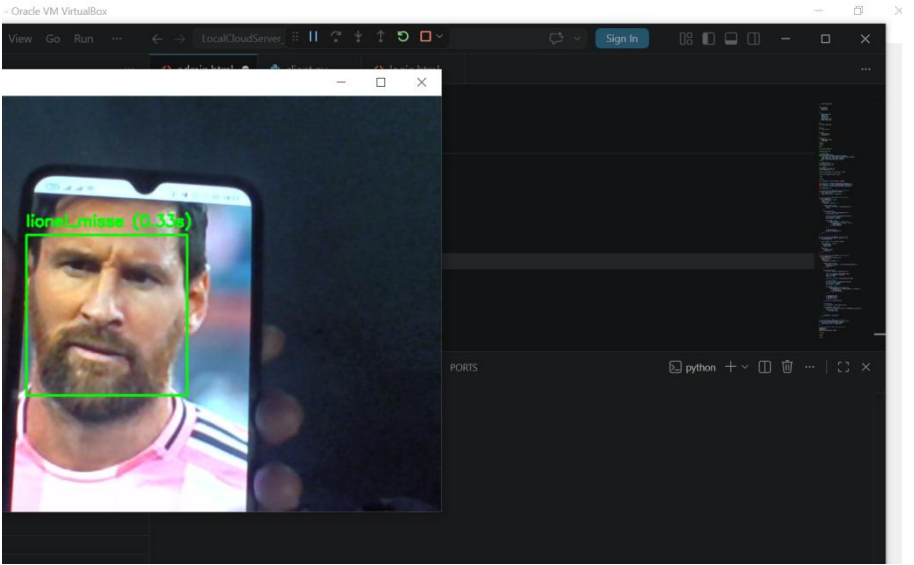


Figure 31:Reconnaissance cloud local 1

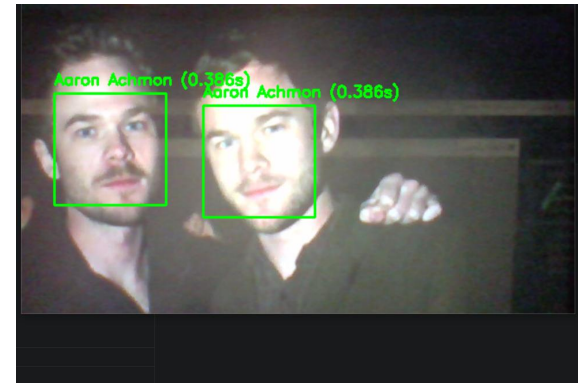
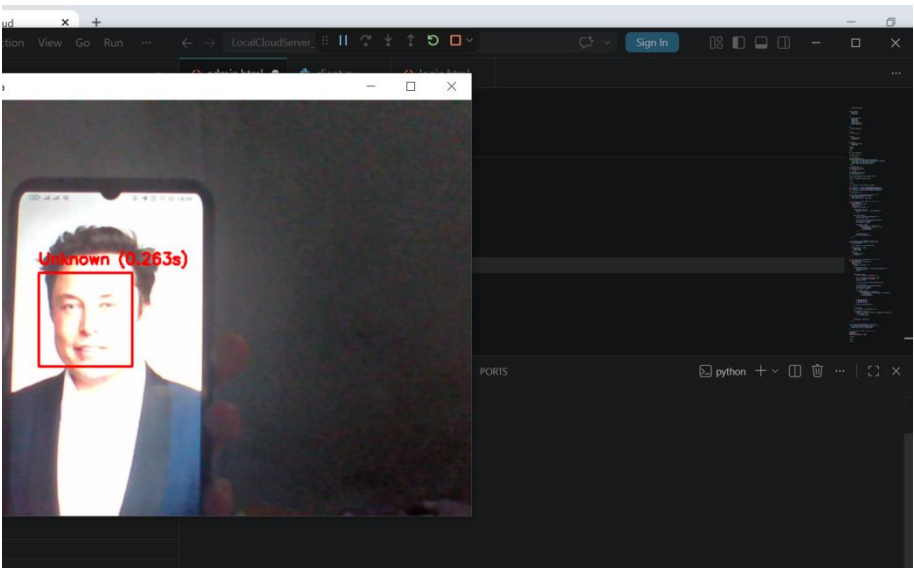


Figure 32 :Reconnaissance cloud local 2

5.1.2 Cloud Réel:

Login:

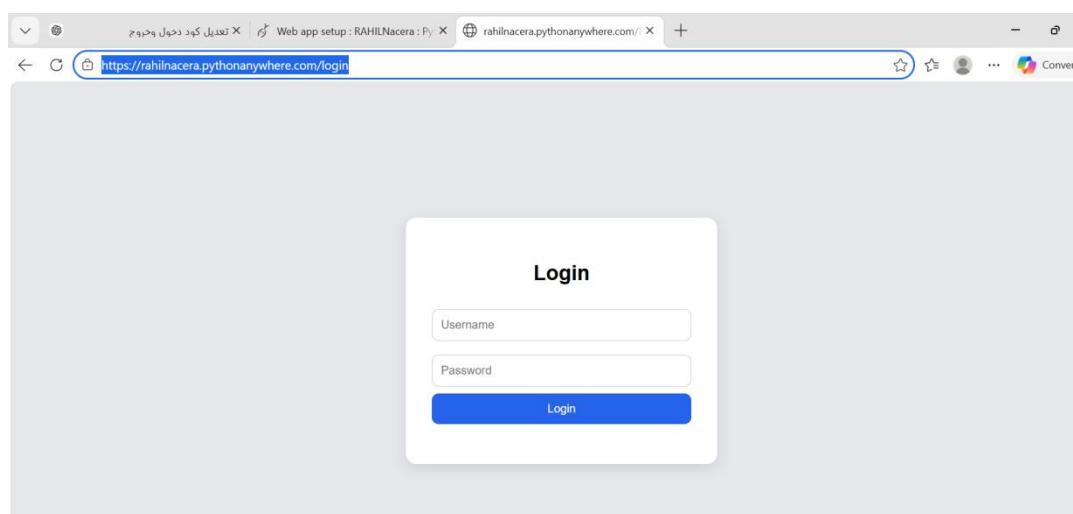


Figure 33:login Cloud Réel

Dashboard:

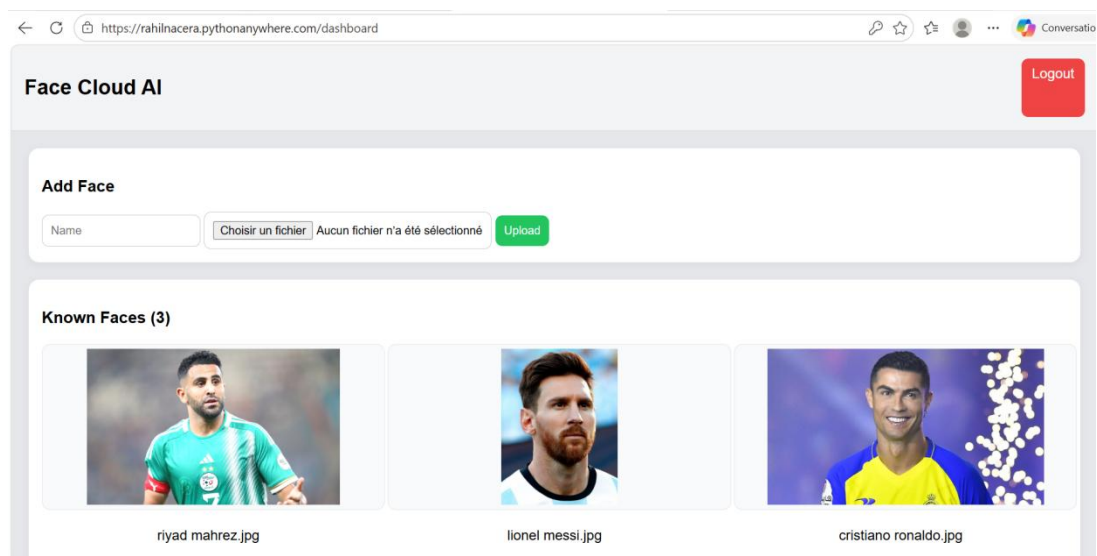


Figure 34:Dashboard Cloud Réel 1

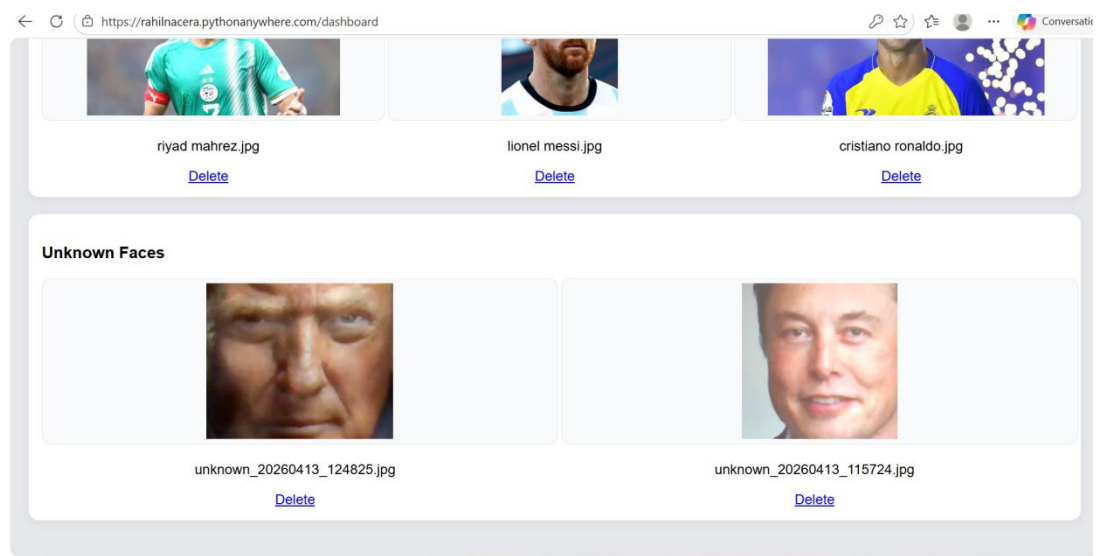


Figure 35 :Dashboard Cloud Réel 2

Ajout Personne:

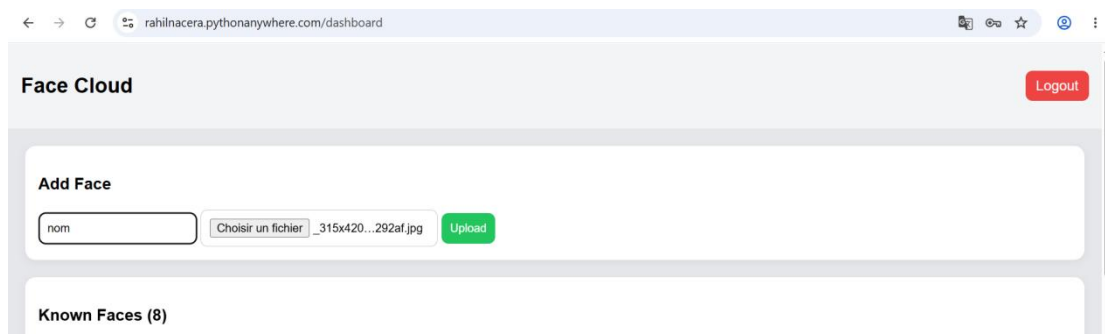


Figure 36: Ajout Personne Cloud Réel

Reconnaissance:

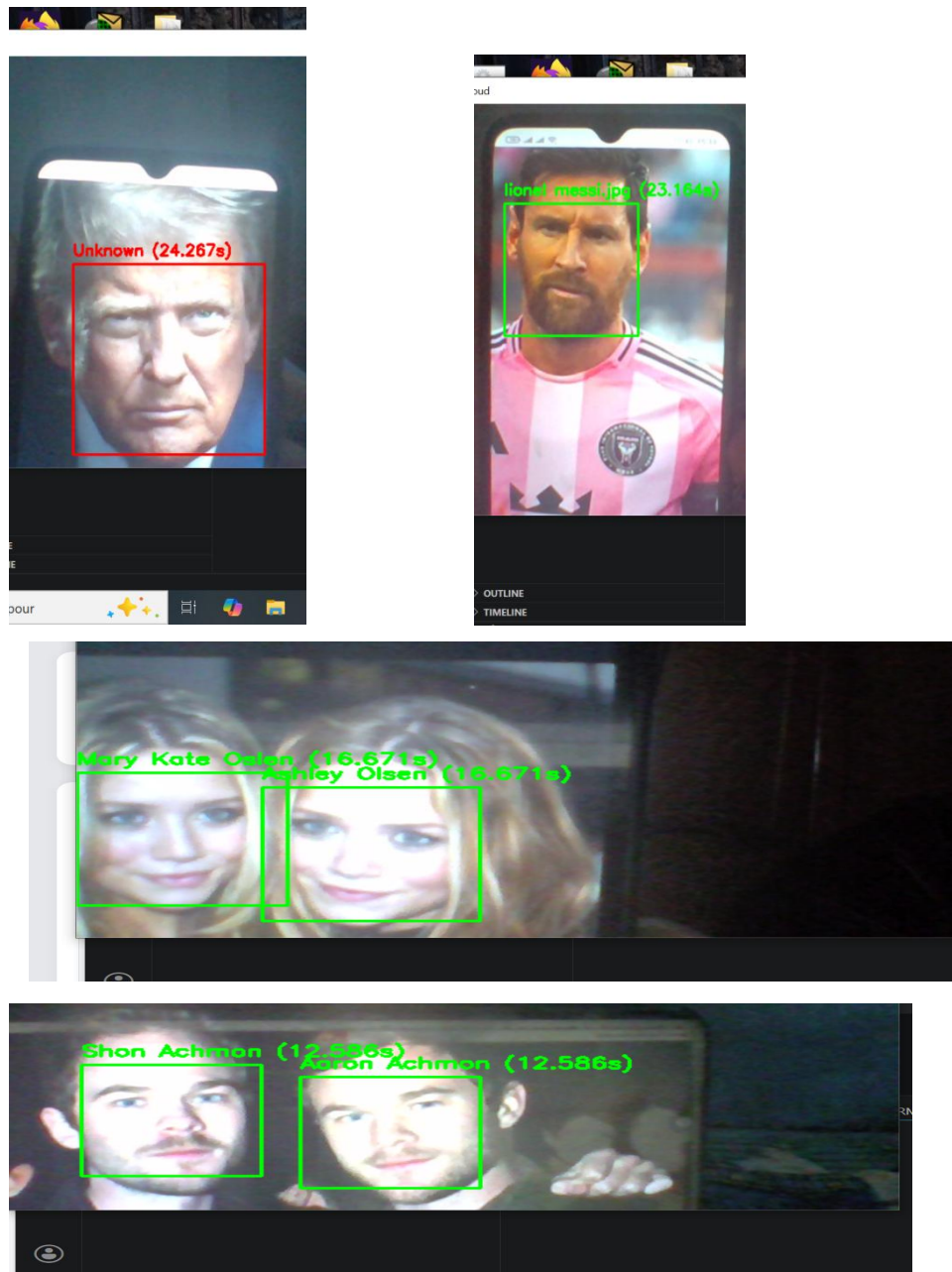


Figure 37 :Reconnaissance Cloud Réel

5.2Temps de réponse:

Afin de comparer les performances des différentes architectures, le temps

de réponse du système a été mesuré lors de plusieurs opérations de reconnaissance faciale. Les résultats obtenus sont présentés dans le tableau suivant :

Test	Cloud Local	Cloud Réel
riyad mahrez	0.31s	26.28s
lionel messi	0.40s	25.759s
cristiano ronaldo	0.42s	25.063s
Mary Kate Oslen	0.41s	17.875s
Ashley Olsen	/	17.875s
Shon Achmon	0.37s	17.368s
Aaron Achmon	/	17.368s

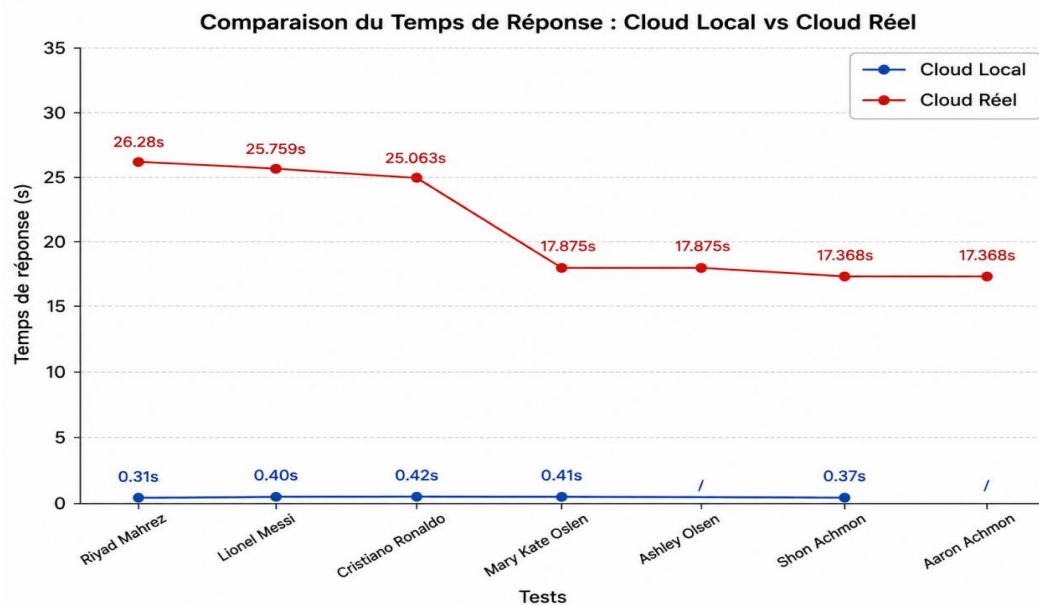


Figure 38: Comparision de Tests

Afin d'évaluer l'impact des conditions d'éclairage sur les performances du système, plusieurs tests ont été réalisés dans différents environnements lumineux. Les résultats observés sont présentés dans le tableau suivant :

Condition d'éclairage	Résultat observé
Bonne luminosité	Reconnaissance correcte et rapide
Luminosité moyenne	Reconnaissance acceptable
Faible luminosité	Diminution de la précision
Contre-jour important	Risque de classification « Unknown »

5.3 Résultats obtenus:

Les différents tests réalisés ont montré que le système fonctionne correctement dans les trois environnements étudiés.

Les résultats obtenus indiquent que :

La reconnaissance faciale est réalisée avec succès pour les personnes enregistrées.

Les visages inconnus sont correctement classés comme « Unknown ».

Le système hybride offre un meilleur compromis entre rapidité et accessibilité.

Le stockage cloud permet une gestion centralisée des données.

Le mécanisme Confidence Gap améliore la distinction entre les personnes résentant de fortes ressemblances facial

5.4 Analyse:

L'analyse des résultats montre que chaque architecture présente des avantages spécifiques.

Le cloud local offre les meilleurs temps de réponse grâce à l'absence de communication Internet. Toutefois, il reste limité en termes d'accessibilité et de scalabilité.

Le cloud réel permet un accès distant et une gestion centralisée des données, mais les temps de réponse sont légèrement plus élevés à cause des échanges réseau.

Le système hybride combine les avantages des deux approches. Les traitements intensifs sont exécutés localement tandis que les données sont stockées dans le cloud. Cette architecture fournit ainsi un bon équilibre entre performance, disponibilité et flexibilité.

L'analyse des tests a également montré que les conditions d'éclairage ont un impact direct sur les performances du système de reconnaissance faciale. Les meilleurs résultats ont été obtenus lorsque les images étaient capturées dans un environnement bien éclairé avec une luminosité homogène. En revanche, dans les situations de faible éclairage ou de contre-jour important, certaines caractéristiques faciales deviennent

moins visibles, ce qui peut réduire la qualité des encodages générés et augmenter le risque d'erreurs de reconnaissance ou de classification des visages comme « Unknown ».

Malgré l'utilisation des bibliothèques OpenCV et Face Recognition pour le traitement des images, l'éclairage reste un facteur important influençant la précision du système. Cette observation confirme la nécessité de capturer les images dans des conditions lumineuses adaptées afin d'obtenir des résultats plus fiables.

6 Discussion:

Les résultats obtenus confirment l'intérêt de l'intégration du Cloud Computing dans les systèmes de reconnaissance faciale.

L'utilisation d'une architecture distribuée permet d'améliorer la gestion des données, de faciliter l'accès aux ressources et de simplifier les opérations de maintenance.

L'approche hybride s'est révélée particulièrement efficace puisqu'elle exploite les ressources locales pour les traitements critiques tout en bénéficiant des services cloud pour le stockage et la synchronisation.

6.1 Difficultés rencontrées:

- * Configuration de l'environnement cloud.
- * Déploiement de Flask sur PythonAnywhere.
- * Variations des conditions d'éclairage lors de la capture des images.
- * Gestion des bibliothèques Face Recognition et Dlib.
- * Temps de transfert des images via Internet.
- * Cas complexes de jumeaux et visages très similaires.
- * Synchronisation entre Firebase et Cloudinary.

6.2 Perspectives d'amélioration:

- 1 Intégration d'un modèle Deep Learning plus avancé.
- 2 Déploiement sur AWS ou Google Cloud.
- 3 Authentification multi-facteurs.
- 4 Développement d'une application mobile Android/iOS.

7 Conclusion:

Dans ce chapitre, nous avons présenté la conception, l'implémentation et l'évaluation de trois architectures de reconnaissance faciale : le cloud

local, le cloud réel et le système hybride.

Les expérimentations réalisées ont démontré que chaque approche possède ses avantages spécifiques. Toutefois, l'architecture hybride s'est révélée la plus adaptée puisqu'elle combine la rapidité du traitement local avec la flexibilité et la disponibilité des services cloud.

Les résultats obtenus confirment la faisabilité de l'intégration du Deep Learning et du Cloud Computing dans un système moderne de reconnaissance faciale capable de répondre aux exigences de performance, de sécurité et de scalabilité.

Conclusion générale:

Au terme de ce travail, nous avons réussi à concevoir et développer un système intégré de reconnaissance faciale reposant sur les techniques de Deep Learning et de Cloud Computing. Trois architectures distinctes ont été mises en œuvre et étudiées : le Cloud local, le Cloud réel et l'architecture hybride, dans le but d'évaluer leurs performances et de comparer leurs caractéristiques.

Le projet ne s'est pas limité à l'utilisation des algorithmes classiques de reconnaissance, mais a introduit une amélioration du processus décisionnel grâce à une approche basée sur la reconnaissance par distance (Distance-Based Recognition) entre les représentations faciales, avec l'application de seuils (Thresholds) adaptés pour contrôler l'acceptation ou le rejet de l'identité. Cette stratégie a permis de réduire les erreurs de reconnaissance erronée (False Recognition) et d'accroître la fiabilité du système, notamment dans les cas de visages fortement similaires.

Les résultats ont montré que l'architecture hybride constitue la solution la plus équilibrée, puisqu'elle combine la rapidité du traitement local avec la flexibilité et la disponibilité des services cloud. Elle offre ainsi de bonnes performances et une forte capacité d'évolution pour les applications réelles.

Quant aux perspectives futures, nous envisageons l'intégration de modèles de Deep Learning plus avancés, le déploiement sur des plateformes cloud professionnelles telles que AWS et Google Cloud, ainsi que le développement d'une application mobile permettant une utilisation plus large du système.

Références

- [1] <https://aws.amazon.com/what-is/facial-recognition>
- [2] reconnaissance faciale [en ligne] <
<https://bibliométrie.wordpress.com/2017/02/12/reconnaissance-faciale/> >
- [3] <https://www.kaspersky.com/resource-center/definitions/what-is-facial-recognition>
- [4] https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_reconnaissance_faciale#Historique
- [5] Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face Recognition: A Literature Survey. ACM Computing Surveys, 35(4), 399–458.
- [6] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.
- [7] Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1), 71–86.
- [8] Jain, A. K., Ross, A., & Nandakumar, K. (2011). Introduction to Biometrics. Springer.
- [9] Wang, M., Deng, W. (2021). Deep face recognition: A survey.

Neurocomputing, 429, 215–244.

[10] Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing. NIST Special Publication 800-145.

[11] Armbrust, M. et al. (2010). A View of Cloud Computing. Communications of the ACM.

[12] Buyya, R. et al. (2011). Cloud Computing: Principles and Paradigms. Wiley.

[13] Microsoft Azure Documentation.

[14] IBM Cloud Documentation.

[15] Google Cloud Documentation.

[16] OpenStack Documentation.

[17] Cloud Security Alliance (CSA).

[18] Amazon Web Services Documentation.

[19] Google Cloud Vision API Documentation.

[20] Microsoft Azure Face API Documentation.

[21] LeCun, Y. et al. (2015). Deep Learning. Nature.

[22] Wang, M., & Deng, W. (2021). Deep Face Recognition: A Survey. Neurocomputing.

[23] Zhao, W. et al. (2003). Face Recognition: A Literature Survey. ACM Computing Surveys.

- [24] Jain, A. K. et al. (2011). Introduction to Biometrics. Springer.
- [25] Turk, M., & Pentland, A. (1991). Eigenfaces for Recognition. Journal of Cognitive Neuroscience.
- [26] Oracle Cloud Documentation.
- [27] Cisco Systems – Secure Access Service Edge (SASE).
- [28] Trend Micro – Cloud Security Solutions Overview.
- [29] NVIDIA – GPU Accelerated Computing for AI.

