

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة عين تموشنت - بلحاج بوشعيب
Université Aïn Temouchent – Belhadj Bouchaib
كلية العلوم والتكنولوجيا
Faculté des Sciences et de Technologie
قسم الرياضيات والإعلام الآلي
Département de Mathématiques et Informatique



Mémoire de Fin d'Études

Présenté en vue de l'obtention du diplôme de

Master en Informatique

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Cybersécurité et Intelligence Artificielle (CYSIA)

Thème

Conception et réalisation d'un système intelligent de
gestion automatisée des présences basé sur la
reconnaissance faciale :
ClassTrack DZ

Présenté par :

YAGOUBI Asmaa

ZIGH Hanaa

Devant le jury composé de :

Dr MESSAOUDI M.A UAT

Président

Dr BARNOUSSI M. UAT

Examineur

Dr MERAD BOUDIA UAT

Encadreur

Djalal

Année universitaire : 2025–2026

Remerciements

وَقُلْ رَبِّ زِدْنِي عِلْمًا

(Sourate Taha, 20:114)

Au nom d'Allah, le Tout Miséricordieux, le Très Miséricordieux, nous Lui rendons grâce pour nous avoir accordé la santé, la patience et la force nécessaires à l'accomplissement de ce travail.

Ce mémoire est le résultat d'un parcours universitaire riche en apprentissages et en efforts continus. Il n'aurait pu être réalisé sans le soutien et l'encouragement de plusieurs personnes auxquelles nous exprimons notre profonde gratitude.

Nous adressons nos remerciements à nous-mêmes pour les efforts fournis, la rigueur et la persévérance tout au long de ce projet.

Nous remercions également nos familles pour leur soutien constant, leurs prières, leur confiance et leurs sacrifices, qui ont été essentiels à notre réussite.

Nous tenons à exprimer notre profonde reconnaissance à notre encadrant **M. Djalal MERAD BOUDIA**, pour son encadrement, ses conseils précieux et sa disponibilité.

Enfin, nous remercions toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce mémoire.

Dédicace

Au nom d'Allah, le Tout Miséricordieux, le Très Miséricordieux.

Je dédie ce modeste travail, fruit de plusieurs années de efforts, de patience et de persévérance, aux personnes les plus chères à mon cœur.

À mes parents, qui ont été et resteront toujours mon plus grand soutien dans la vie. À mon père, pour sa force, sa sagesse et tous les sacrifices qu'il a consentis pour m'offrir un avenir meilleur. À ma mère, pour son amour inconditionnel, ses prières constantes, sa tendresse et sa présence dans chaque étape de mon parcours. Aucun mot ne peut exprimer toute ma gratitude envers eux.

À ma sœur et à mon frère, pour leur affection, leur soutien moral et leur présence qui ont illuminé mon chemin durant les moments difficiles.

À mon fiancé, pour sa patience, son soutien constant, son encouragement et sa compréhension tout au long de ce travail, et pour avoir été à mes côtés dans les moments de doute comme dans les moments de réussite.

À la mémoire de mon grand-père رحمه الله, que Dieu lui fasse miséricorde, dont l'absence laisse un grand vide mais dont les souvenirs et les valeurs continuent de m'accompagner et de m'inspirer chaque jour.

À mes amies, pour leur soutien, leur présence, leurs encouragements et les beaux moments partagés tout au long de mon parcours universitaire.

À mon binôme Zigh Hanaa, avec qui j'ai partagé les défis, les efforts et les réussites de ce parcours académique.

À toutes les personnes qui me sont chères, merci du fond du cœur.

Y. Asmaa

En premier lieu, je dédie ce travail à moi-même, pour avoir affronté les difficultés, les doutes et la fatigue sans jamais abandonner. Pour avoir continué à avancer malgré les obstacles, avec patience, persévérance et détermination. Ce travail est le fruit de mes efforts, de ma résilience et de ma volonté d'atteindre cet objectif.

C'est également avec une immense gratitude et une profonde émotion que je dédie ce travail :

À ma tendre mère Zoulikha, qui a éclairé mon chemin de ses prières et de son amour inconditionnel. Tu es ma force et mon modèle.

À mon cher père Tayeb, pour sa sagesse, ses sacrifices précieux et son soutien indéfectible tout au long de mon parcours. Que Dieu te préserve et t'accorde une longue vie pleine de santé.

À mon précieux époux Amin, pour ta patience infinie, tes encouragements quotidiens et pour avoir toujours cru en moi. Ce travail est aussi le tien.

À la prunelle de mes yeux, mon tendre fils Mohammed, qui illumine ma vie et me donne la force de me surpasser chaque jour.

À mes piliers, mon frère Adel et mes merveilleuses sœurs Sanaa et Sara, pour leur complicité, leur amour et leur présence chaleureuse à mes côtés.

À mon binôme et amie Asmaa, avec qui j'ai partagé les défis et les réussites de ce parcours académique.

À tous les membres de ma famille, à mes amis, et à toutes les personnes qui portent un amour sincère dans leur cœur.

Z. Hanaa

Résumé

La gestion manuelle des présences dans les établissements d'enseignement supérieur demeure chronophage, sujette aux erreurs et vulnérable aux fraudes. Ce mémoire propose le développement de *ClassTrack DZ*, un système intelligent de gestion automatisée des présences fondé sur la reconnaissance faciale. L'étude débute par une revue théorique de l'intelligence artificielle et de l'apprentissage profond, suivie d'une analyse critique de l'état de l'art des algorithmes de détection et de reconnaissance. Une démarche expérimentale rigoureuse a permis de reproduire les travaux de référence tout en évaluant une innovation majeure : l'intégration de l'architecture YOLOv8 pour remplacer les détecteurs conventionnels (Viola-Jones, HOG, MTCNN). Les pipelines hybrides ont été comparés sur deux jeux de données à l'aide de métriques standardisées (exactitude, précision, rappel, F1-Score et temps d'inférence). Les résultats démontrent que YOLOv8 accélère significativement la détection multi-visages tout en maintenant une précision optimale. La combinaison YOLOv8 + Dlib s'est imposée comme le compromis idéal pour un déploiement rapide en amphithéâtre. L'implémentation modulaire, réalisée en Python, garantit la reproductibilité scientifique et intègre une interface pédagogique sécurisée respectant les exigences éthiques relatives aux données biométriques. Ce travail valide la pertinence opérationnelle de la reconnaissance faciale pour moderniser la gestion académique et ouvre la voie à des extensions futures incluant la détection de vivacité et le déploiement sur plateformes embarquées.

Mots-clés : Reconnaissance faciale, YOLOv8, Gestion automatisée des présences, Apprentissage profond, Dlib, ClassTrack DZ, Système temps réel.

Abstract

Manual attendance tracking in higher education institutions remains time-consuming, error-prone, and susceptible to fraud. This thesis presents the development of *ClassTrack DZ*, an intelligent automated attendance system based on facial recognition technology. The study begins with a theoretical overview of artificial intelligence and deep learning, followed by a critical analysis of state-of-the-art detection and recognition algorithms. A rigorous experimental methodology was employed to reproduce reference benchmarks while evaluating a key innovation: the integration of the YOLOv8 architecture to replace conventional detectors (Viola-Jones, HOG, MTCNN). Hybrid pipelines were compared across two datasets using standardized metrics (accuracy, precision, recall, F1-score, and inference time). Results demonstrate that YOLOv8 significantly accelerates multi-face detection while maintaining optimal accuracy. The YOLOv8 + Dlib configuration emerged as the most suitable compromise for real-time deployment in lecture halls. The modular Python-based implementation ensures scientific reproducibility and features a secure pedagogical interface compliant with ethical standards for biometric data. This work validates the operational viability of facial recognition for modernizing academic administration and paves the way for future enhancements, including liveness detection and embedded platform deployment.

Keywords: Facial Recognition, YOLOv8, Automated Attendance Management, Deep Learning, Dlib, ClassTrack DZ, Real-Time System.

الملخص

لا تزال إدارة الحضور اليدوية في مؤسسات التعليم العالي تستغرق وقتاً طويلاً، وعرضة للأخطاء، وعرضة للاحتيال. تقترح هذه الأطروحة تطوير نظام DZ، ClassTrack وهو نظام ذكي لإدارة الحضور آلياً يعتمد على تقنية التعرف على الوجوه. تبدأ الدراسة بمراجعة نظرية للذكاء الاصطناعي والتعلم العميق، تليها دراسة تحليلية لأحدث التقنيات في خوارزميات الكشف والتعرف.

وقد سمح منهج تجريبي دقيق بتكرار العمل المرجعي مع تقييم ابتكار رئيسي: دمج بنية YOLOv8 لاستبدال أجهزة الكشف التقليدية، Viola-Jones، HOG، MTCNN. تمت مقارنة مسارات المعالجة المهيمنة على مجموعتي بيانات باستخدام مقاييس موحدة (الدقة، والضبط، والاستدعاء، ومقياس F1، ووقت الاستدلال).

تُظهر النتائج أن YOLOv8 يُسرّع بشكل ملحوظ عملية الكشف عن الوجوه المتعددة مع الحفاظ على دقة مثالية. وقد برزت تركيبة Dlib + YOLOv8 كحل وسط مثالي للتطبيق السريع في قاعات المحاضرات.

يضمن التطبيق المعياري، المكتوب بلغة Python، إمكانية تكرار النتائج العلمية، ويتضمن واجهة تعليمية آمنة تحترم المتطلبات الأخلاقية للبيانات البيومترية. يؤكد هذا العمل على الأهمية العملية لتقنية التعرف على الوجه في تحديث الإدارة الأكاديمية، ويمهد الطريق لتوسعات مستقبلية، تشمل كشف حيوية الوجه ونشرها على المنصات المدججة.

الكلمات المفتاحية:

التعرف على الوجه، YOLOv8، إدارة الحضور الآلية، التعلم العميق، DZ، ClassTrack Dlib، نظام الوقت الحقيقي.

Table des matières

Introduction générale	2
Chapitre 1 : Fondements théoriques de l'Intelligence Artificielle et de la Reconnaissance Faciale	
1.1 Introduction	6
1.2 Définition de l'Intelligence Artificielle	6
1.2.1 Types d'Intelligence Artificielle	7
1.2.2 Historique et évolution de l'IA	8
1.2.3 Applications de l'Intelligence artificielle dans la gestion des absences des étudiants	9
1.3 Apprentissage automatique (Machine Learning)	10
1.3.1 Apprentissage supervisé	10
1.3.2 Apprentissage non supervisé	11
1.3.3 Apprentissage par renforcement	11
1.4 Réseaux de neurones artificiels (ANN)	11
1.4.1 Structure d'un réseau de neurones	12
1.5 Apprentissage profond (Deep Learning)	13
1.5.1 Fonctionnement de l'apprentissage profond	13
1.5.2 Architectures de Deep Learning	15
1.5.2.1 Réseaux de neurones multicouches (MLP Multi-Layer Perceptron)	15
1.5.2.2 Réseaux de neurones convolutifs (CNN Convolutional Neu- ral Networks)	16
1.6 Reconnaissance faciale	19
1.6.1 Définition de la reconnaissance faciale	19
1.6.2 Étapes du système de reconnaissance faciale	19
1.6.3 Capture d'image	20
1.6.4 Détection du visage	20
1.6.5 Extraction des caractéristiques et conversion en vecteur (embedding)	21
1.6.6 Comparaison et identification	21
1.6.7 Avantages de la reconnaissance faciale	22
1.6.8 Limites de la reconnaissance faciale	22

1.7	Conclusion	24
Chapitre 2 : État de l'art des systèmes de reconnaissance faciale		26
2.1	Introduction	26
2.2	Méthodes de détection des visages dans une image	26
2.2.1	Haar Cascade (ViolaJones)	26
2.2.2	HOG (Histogram of Oriented Gradients)	29
2.2.3	MTCNN (Multi-task Cascaded CNN)	30
2.2.4	YOLO (You Only Look Once)	32
2.3	Méthodes de reconnaissance faciale	35
2.3.1	Local Binary Pattern Histogram (LBPH) :	35
2.3.2	Algorithme FaceNet :	36
2.3.3	Modèle Dlib basé sur ResNet	37
2.4	Étude comparative	39
2.4.1	Métriques de comparaison	39
2.4.2	Expérimentations	40
2.4.2.1	Environnement expérimental	41
2.4.2.2	Bases de données utilisées	41
2.4.2.3	Application des algorithmes sur les bases de données	42
2.4.3	Contribution	43
2.4.3.1	Analyse comparative des performances	45
2.4.4	Analyse et discussion des résultats	52
2.4.4.1	Analyse des résultats sur la base de données 1	52
2.4.4.2	Analyse des résultats sur la base de données 2	53
2.4.5	Synthèse comparative des approches	54
2.4.6	Interprétation globale	54
2.5	Conclusion	56
Chapitre 3 : Conception, Implémentation		58
3.1	Introduction	58
3.2	Analyse et définition des besoins	59
3.2.1	Contexte et problématique	59
3.2.2	Objectifs du système	59
3.2.3	Besoins fonctionnels	59
3.2.4	Besoins non fonctionnels	60
3.3	Conception	60
3.3.1	Étude des acteurs	60
3.3.2	Diagrammes UML	61
3.3.2.1	Diagramme d'activités	61
3.3.2.2	Diagramme de cas d'utilisation	61

3.3.2.3	Diagramme de classes	62
3.3.2.4	Diagramme de séquence - Pointage facial	63
3.3.3	Modélisation de la base de données	64
3.4	Implémentation et développement	65
3.4.1	Environnement de développement	65
3.4.2	Langage de programmation	65
3.4.2.1	Bibliothèques utilisées	66
3.4.3	Matériel utilisé	67
3.4.4	Présentation de l'application	67
3.4.4.1	Écran d'accueil	67
3.4.4.2	Interface administrateur	68
3.4.4.3	Interface enseignant	70
3.5	Résultats	72
3.6	Difficultés rencontrées	73
3.7	Perspectives futures	74
3.8	Conclusion	75
Conclusion générale		76

Table des figures

1.1	Schéma général de l'Intelligence Artificielle	7
1.2	Applications de l'IA dans la gestion éducative et la prise de présence automatisée	10
1.3	Types d'apprentissage automatique	11
1.4	Structure générale d'un réseau de neurones artificiels (ANN)	12
1.5	Schéma illustratif du Deep Learning à plusieurs couches	13
1.6	Fonctionnement du processus d'apprentissage profond basé sur Forward et Backpropagation	14
1.7	Architecture générale d'un réseau MLP	16
1.8	Architecture d'un réseau neuronal CNN	17
1.9	Illustration de la convolution dans un réseau CNN	17
1.10	Processus de reconnaissance faciale	19
1.11	Étapes d'un système de reconnaissance faciale	20
1.12	Exemple de détection automatique de visages	21
1.13	Extraction des caractéristiques faciales à partir des points clés (landmarks)	21
2.1	Architecture du réseau MTCNN	31
2.2	Pipeline du système basé sur YOLOv8	33
2.3	Principe de l'opérateur LBP appliqué à une image faciale	36
2.4	Principe de la Triplet Loss	38
2.5	Comparaison de l'Accuracy – Base de données 1	45
2.6	Comparaison de la Precision – Base de données 1	46
2.7	Comparaison du Recall – Base de données 1	47
2.8	Comparaison du F1-Score – Base de données 1	47
2.9	Comparaison du temps de traitement (s) – Base de données 1	48
2.10	Comparaison de l'Accuracy – Base de données 2	49
2.11	Comparaison de la Precision – Base de données 2	49
2.12	Comparaison du Recall – Base de données 2	50
2.13	Comparaison du F1-Score – Base de données 2	51
2.14	Comparaison du temps de traitement (s) sur la Base de données 2	51
3.1	Diagramme d'activités du système	61
3.2	Diagramme de cas d'utilisation du système	62
3.3	Diagramme de classes du système	63

3.4	Diagramme de séquence du processus de pointage facial	64
3.5	Interface de l'éditeur Visual Studio Code	65
3.6	Logo du langage Python	66
3.7	Écran d'accueil	68
3.8	Interface de connexion et tableau de bord administrateur	68
3.9	Formulaire d'ajout d'un étudiant	69
3.10	Interface d'annuaire et d'archivage	69
3.11	Interface de gestion des comptes utilisateurs	70
3.12	Interface de connexion et tableau de bord enseignant	70
3.13	Interface de reconnaissance faciale	71
3.14	Interface d'affichage de la liste des présences	71
3.15	Architecture du pipeline de reconnaissance faciale	72
3.16	Résultat de reconnaissance faciale avec la combinaison YOLOv8 + Dlib . .	73

Liste des tableaux

1.2	Types d'intelligence artificielle	8
1.3	Évolution historique de l'intelligence artificielle (1950–2026)	9
1.4	Comparaison des types de pooling dans un réseau CNN	18
2.1	Interprétation des métriques dans le contexte de la gestion automatisée des présences	39
2.2	Résultats de classification obtenus sur la base de données 1	43
2.3	Résultats de classification obtenus sur la base de données 2	43
2.4	Performances des configurations YOLOv8 sur la base de Lateef et Kamil (31 étudiantes)	44
2.5	Performances des configurations YOLOv8 sur la base locale (8 sujets, 24 images)	45
2.6	Synthèse comparative des approches de détection et de reconnaissance faciale	54
3.1	Structure des tables de la base de données	64

Liste des abréviations

AGI	Intelligence artificielle générale
ANN	Réseau de neurones artificiels
ASI	Intelligence artificielle super-intelligente
CNN	Réseau de neurones convolutifs
Dlib	Digital Library (bibliothèque de vision par ordinateur)
DL	Apprentissage profond (Deep Learning)
FaceNet	Réseau d'embedding pour reconnaissance faciale
FN	Faux négatifs
FP	Faux positifs
FPS	Images par seconde (Frames Per Second)
GPU	Processeur graphique
HOG	Histogramme des gradients orientés
IA	Intelligence artificielle
LBP	Motif binaire local (Local Binary Pattern)
LBPH	Histogramme de motifs binaires locaux
mAP	Moyenne des précisions (Mean Average Precision)
ML	Apprentissage automatique (Machine Learning)
MLP	Perceptron multicouche
MTCNN	Réseau convolutif en cascade multi-tâches
O-Net	Output Network
P-Net	Proposal Network
RAM	Mémoire vive

ReLU	Unité linéaire rectifiée
ResNet	Réseau neuronal résiduel (Residual Network)
RGB	Rouge Vert Bleu
R-Net	Refine Network
ROI	Région d'intérêt
SVM	Machine à vecteurs de support
TN	Vrais négatifs
TP	Vrais positifs
YOLO	You Only Look Once

Introduction générale

La transformation numérique des établissements d'enseignement supérieur constitue aujourd'hui un levier essentiel pour améliorer l'efficacité des services administratifs et pédagogiques. Cette mutation s'inscrit dans une dynamique plus large de digitalisation qui touche l'ensemble des secteurs d'activité, et l'université ne fait pas exception. Les établissements universitaires, confrontés à des effectifs étudiants en constante augmentation et à des exigences croissantes en matière de traçabilité pédagogique, sont aujourd'hui amenés à repenser leurs processus de gestion afin de gagner en efficacité, en fiabilité et en réactivité.

Parmi les activités administratives les plus sensibles, la gestion des présences des étudiants représente une activité primordiale, permettant d'assurer le suivi de l'assiduité et de contribuer à l'évaluation du parcours académique. En effet, l'assiduité aux enseignements constitue, dans de nombreux établissements, un critère déterminant pour l'autorisation de passage aux examens, l'attribution de bourses, ou encore l'évaluation continue des étudiants. Il s'agit donc d'une donnée stratégique qui doit être collectée de manière précise, fiable et exploitable.

Les méthodes traditionnelles de gestion des présences, fondées sur les appels manuels ou les feuilles d'émargement, présentent plusieurs limites en termes de temps de traitement, de fiabilité et de sécurité. D'une part, l'appel nominal effectué par l'enseignant en début de séance consomme un temps précieux qui pourrait être consacré à l'enseignement lui-même, particulièrement dans les amphithéâtres regroupant un grand nombre d'étudiants. D'autre part, les feuilles d'émargement, bien que largement répandues, demeurent vulnérables à la fraude : il est en effet courant qu'un étudiant signe pour le compte d'un de ses camarades absent, ce qui fausse considérablement les statistiques d'assiduité. Enfin, la centralisation, la saisie et le traitement manuel de ces données introduisent des risques d'erreurs humaines et retardent la disponibilité des informations pour les services pédagogiques et administratifs.

Face à ces contraintes, les avancées récentes de l'intelligence artificielle et de la vision par ordinateur ouvrent de nouvelles perspectives pour l'automatisation de ce processus. Les progrès réalisés ces dernières années dans le domaine de l'apprentissage profond, et plus particulièrement dans les réseaux de neurones convolutifs, ont permis des avancées considérables en matière de détection et de reconnaissance faciale. Ces technologies, désormais matures et accessibles, offrent la possibilité de concevoir des systèmes capables

d'identifier automatiquement des individus à partir d'images ou de flux vidéo, avec des niveaux de précision élevés et des temps de traitement compatibles avec une utilisation en temps réel. Leur intégration dans le contexte universitaire constitue ainsi une opportunité pertinente pour repenser la gestion des présences de manière plus moderne, plus rapide et plus fiable.

Dans ce cadre, notre projet de fin d'études, intitulé *Conception et réalisation d'un système intelligent de gestion automatisée des présences basé sur la reconnaissance faciale : ClassTrack DZ*, vise à concevoir et développer une plateforme dédiée à la gestion automatisée des présences des étudiants. Ce projet s'inscrit dans une démarche d'innovation technologique appliquée au contexte universitaire algérien, en proposant une solution pragmatique, adaptée aux contraintes réelles d'un environnement académique, tant sur le plan matériel que sur le plan organisationnel.

La problématique traitée dans ce mémoire peut être formulée comme suit : *Comment développer une solution intelligente pour l'automatisation de la gestion des présences au sein d'un environnement universitaire réel, garantissant précision et efficacité ?*

Cette problématique soulève plusieurs questionnements sous-jacents qui ont guidé notre démarche de recherche et de développement. Quelles sont les approches de détection et de reconnaissance faciale les plus adaptées à un contexte de salle de classe, caractérisé par des variations d'éclairage, d'angle de prise de vue et de densité d'occupants ? Comment garantir un taux de reconnaissance satisfaisant tout en limitant les faux positifs et les faux négatifs, qui pourraient respectivement valider la présence d'un étudiant absent ou, à l'inverse, marquer comme absent un étudiant pourtant présent ? Enfin, comment concevoir une architecture logicielle robuste, capable de s'intégrer aisément dans le fonctionnement quotidien d'un établissement universitaire ?

Afin d'apporter une réponse à cette problématique, nous avons étudié les principales approches de détection et de reconnaissance faciale, puis conçu et développé une solution complète destinée à la gestion automatisée des présences des étudiants. Notre démarche s'est articulée autour de trois axes complémentaires. Le premier axe a consisté en une étude théorique approfondie des fondements de l'intelligence artificielle et de l'apprentissage profond, nécessaire à la compréhension des mécanismes sous-jacents aux systèmes de reconnaissance faciale. Le deuxième axe a porté sur une analyse comparative des principales méthodes existantes dans la littérature, en vue de sélectionner les approches les plus pertinentes pour notre contexte d'application. Le troisième axe, à caractère pratique, a consisté en la conception, l'implémentation et l'évaluation expérimentale de l'application *ClassTrack DZ*, à travers des tests réalisés sur des corpus de données représentatifs.

Ce mémoire est structuré en trois chapitres principaux :

- **Le premier chapitre**, intitulé *Fondements théoriques de l'Intelligence Artificielle et de la Reconnaissance Faciale*, expose les concepts fondamentaux liés à l'intelligence artificielle et à l'apprentissage profond. Il pose les bases théoriques indis-

pensables à la compréhension des chapitres suivants, en présentant notamment les principes des réseaux de neurones, de l'apprentissage supervisé, ainsi que les spécificités des architectures dédiées au traitement d'images.

- **Le deuxième chapitre**, consacré à l'*État de l'art des systèmes de reconnaissance faciale*, dresse une étude critique des principales méthodes de détection et de reconnaissance, suivie d'une analyse comparative de leurs performances. Ce chapitre permet de justifier les choix techniques retenus pour la conception de notre solution, sur la base de critères tels que la précision, la rapidité de traitement et la facilité d'intégration.
- **Le troisième chapitre** détaille la conception et l'implémentation technique de l'application *ClassTrack DZ*, ainsi que l'analyse des résultats expérimentaux obtenus sur nos corpus de test. Il présente l'architecture globale du système, les choix technologiques effectués, ainsi qu'une évaluation rigoureuse des performances obtenues, accompagnée d'une discussion critique des résultats.

Nous terminerons ce mémoire par une conclusion générale qui synthétisera les principaux apports de ce travail, tout en proposant des perspectives d'amélioration et d'évolution future du système développé.

Chapitre 1

Fondements théoriques de l'Intelligence
Artificielle et de la Reconnaissance Faciale

1.1 Introduction

L'intelligence artificielle (IA) est désormais considérée comme un élément crucial de la transition numérique des structures organisationnelles et des systèmes éducatifs. Les progrès significatifs en matière d'apprentissage automatique (Machine Learning) et d'apprentissage profond (Deep Learning) ont conduit à la création de modèles capables d'analyser et de comprendre des données complexes, y compris les images numériques. La reconnaissance faciale, grâce à son aptitude à reconnaître automatiquement des personnes en se basant sur des traits biométriques distinctifs, occupe une position dominante parmi ces applications.

Dans le secteur de l'éducation, cette technologie présente des possibilités importantes pour automatiser la gestion des présences, remplaçant ainsi les techniques traditionnelles souvent longues et sujettes à des erreurs humaines. L'objectif de ce chapitre est d'exposer les bases théoriques indispensables à la compréhension du système suggéré dans ce mémoire. D'où les points successivement traités :

- Les principes de base de l'intelligence artificielle.
- Les notions d'apprentissage automatique (Machine Learning) et d'apprentissage profond (Deep Learning).
- Les structures des réseaux de neurones artificiels, en particulier les MLP et les CNN.
- Le fonctionnement détaillé d'un système de reconnaissance de visages.

Ainsi que les bénéfices et les contraintes liés à leur utilisation dans la gestion des absences des étudiants. Ces aspects constituent le socle théorique de notre étude intitulée : « Utilisation de l'Intelligence Artificielle pour la Gestion des Absences ».

1.2 Définition de l'Intelligence Artificielle

L'intelligence artificielle (IA) est un domaine de l'informatique consacré à la conception de systèmes capables de reproduire certaines fonctions cognitives humaines, telles que l'apprentissage, le raisonnement et la prise de décision. Ces systèmes analysent des données, identifient des régularités et adaptent leur comportement en fonction des expériences acquises.

Selon Russell et Norvig, l'IA peut être définie comme :

L'intelligence artificielle est un domaine scientifique dont l'objectif est de concevoir des agents capables d'appréhender leur environnement, de raisonner et d'agir rationnellement afin d'atteindre des objectifs spécifiques, en fusionnant l'analyse des processus mentaux avec le comportement observable. [1] L'IA constitue un cadre général englobant plusieurs approches d'apprentissage automatisé, notamment le Machine Learning et le Deep Lear-

ning, qui seront détaillées dans les sections suivantes. La Figure 1.1 présente un schéma général de l'Intelligence Artificielle.

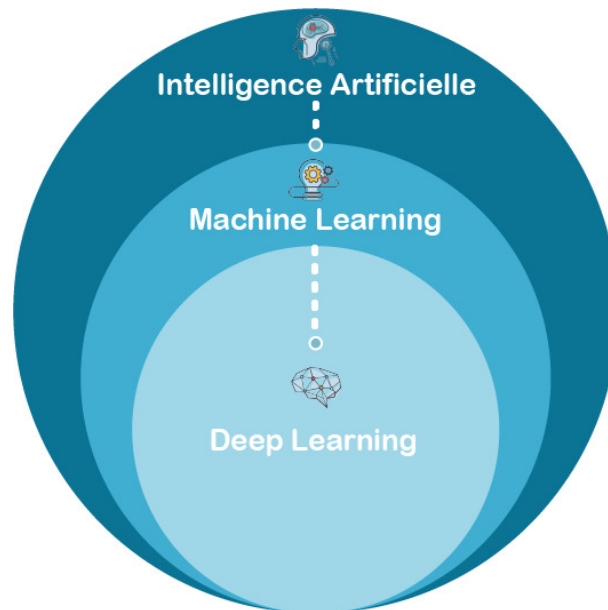


FIGURE 1.1 – Schéma général de l'Intelligence Artificielle

1.2.1 Types d'Intelligence Artificielle

On distingue traditionnellement trois niveaux d'intelligence artificielle : IA faible, IA forte et IA super-intelligente. Le tableau ci-dessous résume ces types avec leurs caractéristiques principales.

TABLEAU 1.2 – Types d'intelligence artificielle

Type d'IA	Description
IA faible (Narrow AI)	— Systèmes conçus pour exécuter une tâche spécifique. — Ne disposent ni de conscience ni de capacités générales d'adaptation. — Exemple : reconnaissance faciale, assistants vocaux.
IA générale (AGI)	— Intelligence théorique capable de réaliser toute tâche cognitive humaine. — Possède une capacité d'adaptation et d'apprentissage autonome. — Encore inexistante à ce jour.
IA super-intelligente (ASI)	— Concept hypothétique dépassant les capacités cognitives humaines. — Capable de surpasser l'humain dans tous les domaines intellectuels. — Sujet encore purement théorique en recherche.

Dans le cadre de ce projet, nous utilisons une intelligence artificielle faible (Narrow AI), car le système est spécifiquement programmé pour la reconnaissance faciale afin de gérer les absences, sans ambition d'imiter le raisonnement humain.

1.2.2 Historique et évolution de l'IA

L'intelligence artificielle (IA) se développe depuis les années 1950, notamment à travers les travaux d'Alan Turing et la conférence de Dartmouth, souvent cités comme des étapes fondatrices pour la définition des concepts et l'organisation du champ de recherche. Après des périodes de ralentissement dues aux limites techniques, souvent qualifiées d'('AI winter'), les avancées en Machine Learning et en Deep Learning depuis les années 2000, avec l'accès à de grands volumes de données et à une puissance de calcul accrue, ont rendu possible le développement de systèmes qui apprennent à partir de leurs expériences. Dans des domaines comme la reconnaissance faciale, ces systèmes atteignent désormais une vitesse de traitement et une précision qui peuvent dépasser celles d'un humain. [2]

TABLEAU 1.3 – Évolution historique de l'intelligence artificielle (1950–2026)

Période	Étape clé	Idée principale	Résultat / Impact
1950–1960	Naissance de l'IA	Travaux d'Alan Turing et conférence de Dartmouth	Fondements théoriques de l'intelligence artificielle et développement des premiers programmes capables de résoudre des problèmes simples.
1980–2000	Systèmes experts et réseaux de neurones	Introduction de la rétropropagation (<i>backpropagation</i>) pour l'apprentissage des réseaux neuronaux	Développement d'applications spécialisées dans plusieurs domaines, malgré des limitations liées aux données et à la puissance de calcul disponible.
2010–2026	Apprentissage profond (<i>Deep Learning</i>)	Utilisation des réseaux de neurones profonds, notamment les CNN et le transfert d'apprentissage	Amélioration significative des performances en vision par ordinateur, traitement du langage naturel et reconnaissance faciale.

1.2.3 Applications de l'Intelligence artificielle dans la gestion des absences des étudiants

Depuis quelques années, l'intelligence artificielle (IA) occupe une place croissante dans le domaine de l'éducation, notamment parce qu'elle permet d'automatiser certaines tâches administratives. La gestion des absences en est un exemple courant, car elle repose souvent sur des opérations répétitives qui peuvent être traitées plus rapidement et avec moins d'erreurs grâce à des outils numériques. Intégrer la reconnaissance faciale dans les systèmes de prise de présence peut limiter les erreurs liées aux saisies manuelles, réduire le temps consacré au pointage et améliorer la fiabilité des données d'assiduité enregistrées. [3]

Parmi les applications les plus pertinentes, nous pouvons notamment citer :

- **Automatisation de la présence** : analyse et suivi de l'assiduité des étudiants. La précision de ce suivi pourrait être renforcée en s'appuyant sur des réseaux de neurones convolutifs (CNN), qui sont bien adaptés à l'analyse d'images et à l'extraction de caractéristiques pertinentes [4].
- **Suivi de la participation** : des plateformes numériques peuvent mesurer la fréquence des contributions et le niveau d'engagement des étudiants dans les activités. Ces informations permettent à l'enseignant de repérer plus tôt les situations de décrochage et d'ajuster l'accompagnement pédagogique en fonction des besoins.
- **Authentification biométrique** : réduit les fausses déclarations de présence et

contribue à une évaluation plus équitable.

- **Gestion des rapports et des notifications** : des outils automatisés génèrent des rapports et envoient des alertes au personnel, y compris aux étudiants, afin d'assurer un suivi en temps réel.

La Figure 1.2 illustre les principales applications de l'intelligence artificielle dans la gestion éducative et la prise de présence automatisée.

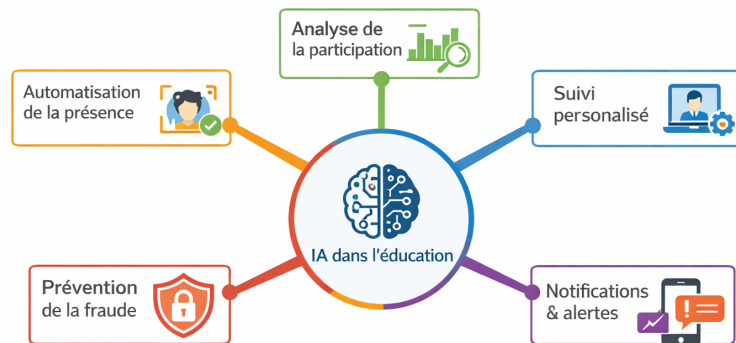


FIGURE 1.2 – Applications de l'IA dans la gestion éducative et la prise de présence automatisée

1.3 Apprentissage automatique (Machine Learning)

L'apprentissage automatique (Machine Learning) est un domaine central de l'intelligence artificielle. Il repose sur l'idée qu'un système peut repérer des régularités dans des données et améliorer ses résultats au fil des exemples, sans dépendre uniquement de règles fixées à l'avance.

Il s'appuie sur la conception de modèles mathématiques qui permettent d'extraire des connaissances à partir des données et d'améliorer progressivement leurs performances au fil de l'expérience. Dans le cadre d'un dispositif de reconnaissance faciale pour la gestion des absences, cette méthode vise à améliorer l'identification des étudiants au fil du temps, grâce à l'apprentissage progressif du système et à une précision qui augmente avec l'accumulation des données. [5]

1.3.1 Apprentissage supervisé

En apprentissage supervisé, on entraîne un modèle à partir d'un ensemble de données annotées, dans lequel chaque observation est associée à la réponse attendue, ce qui permet

au modèle d'apprendre la correspondance entre les entrées et les sorties. Par exemple, on peut constituer un ensemble d'images de visages d'étudiants, chacune associée à l'identité de la personne, afin d'entraîner le système à reconnaître ces mêmes individus lorsqu'il reçoit de nouvelles images.

1.3.2 Apprentissage non supervisé

L'apprentissage non supervisé s'appuie surtout sur des méthodes comme le clustering, qui consiste à regrouper des données, par exemple des visages, selon leurs similarités, sans disposer d'étiquettes au départ. Cette approche permet d'analyser la structure des données et de repérer des régularités qui ne sont pas visibles au premier regard.

1.3.3 Apprentissage par renforcement

L'apprentissage par renforcement (LR) est une approche d'apprentissage automatique dans laquelle un agent apprend à choisir des actions en interagissant avec un environnement, puis en ajustant sa stratégie à partir des récompenses (ou des pénalités) obtenues après chaque action.

Il existe aussi des approches comme l'apprentissage semi-supervisé et l'apprentissage par transfert, qui permettent de combiner ces méthodes ou de les adapter en fonction de l'objectif et des données disponibles.

Dans notre projet, nous utilisons principalement l'apprentissage supervisé, car chaque étudiant est identifié de manière explicite dans la base de données. La Figure 1.3 présente les différents types d'apprentissage automatique.

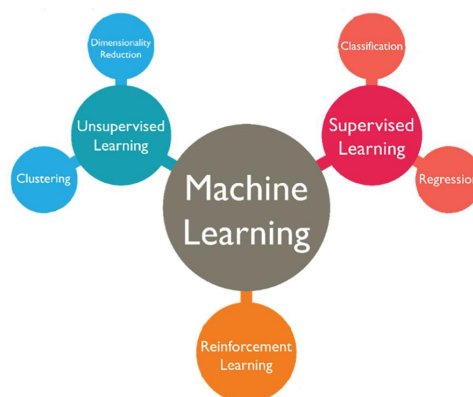


FIGURE 1.3 – Types d'apprentissage automatique

1.4 Réseaux de neurones artificiels (ANN)

Les réseaux de neurones artificiels (ANN), au cœur de l'apprentissage profond, s'inspirent du fonctionnement du cerveau humain. Chaque neurone artificiel reçoit des signaux

entrants, les modifie en suivant une procédure mathématique et transmet le résultat aux neurones subséquents en recourant à une fonction d'activation. Cette structure en couches aide le système à apprendre progressivement les liens complexes entre les données et le résultat souhaité. Cela lui confère une puissante capacité de modélisation pour des travaux tels que la reconnaissance faciale.

1.4.1 Structure d'un réseau de neurones

Un réseau de neurones artificiels standard se compose de trois types de couches organisées séquentiellement :

Couche d'entrée (Input Layer) : cette couche reçoit les données initiales représentées sous forme de vecteur de caractéristiques

$$\mathbf{x} = [x_1, \dots, x_d]^\top$$

Chaque neurone de cette couche correspond à une dimension de l'entrée, par exemple un pixel dans le cas d'une image.

Couches cachées (Hidden Layers) : Ils appliquent des opérations mathématiques aux données reçues. Chaque neurone calcule une combinaison pondérée de ses entrées, puis applique une fonction d'activation (sigmoïde, ReLU ou tanh). La transformation peut être formalisée par :

$$\mathbf{a}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)})$$

où $\mathbf{W}^{(l)}$ représente la matrice des poids, $\mathbf{b}^{(l)}$ le vecteur de biais, et $\sigma(\cdot)$ la fonction d'activation non linéaire. Ces couches permettent au réseau de capturer des relations complexes et non linéaires entre les entrées et les sorties.

Couche de sortie (Output Layer) : La dernière couche fournit la prédiction en sortie du réseau, en fonction de la tâche analysée. À titre d'exemple, l'identification d'un étudiant au sein d'une application de reconnaissance faciale. Cette couche traduit l'information traitée par les couches précédentes en résultat exploitable.

La Figure 1.4 illustre la structure générale d'un réseau de neurones artificiels.

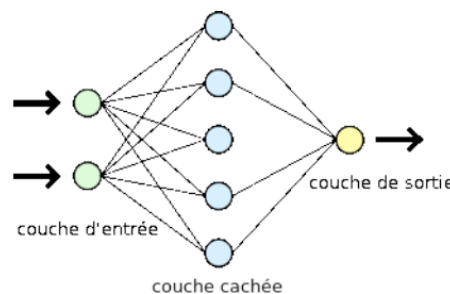


FIGURE 1.4 – Structure générale d'un réseau de neurones artificiels (ANN)

1.5 Apprentissage profond (Deep Learning)

L'apprentissage profond (Deep Learning) est un domaine avancé de l'apprentissage automatique reposant sur des réseaux de neurones artificiels profonds, capables d'apprendre des représentations hiérarchiques à travers plusieurs couches successives, avec des niveaux d'abstraction progressivement plus élevés. [6]

Cette architecture multicouche permet au modèle d'extraire automatiquement des caractéristiques complexes à partir de données brutes, sans nécessiter d'intervention humaine dans la phase d'extraction des attributs.

Le Deep Learning (DL) permet ainsi de modéliser des relations non linéaires complexes et de généraliser à partir d'exemples variés, grâce à un ajustement progressif des paramètres lors de l'entraînement sur de vastes ensembles de données. Cette capacité en fait une approche particulièrement adaptée aux applications de vision par ordinateur, notamment la reconnaissance faciale, où la précision dépend fortement de la qualité des représentations apprises par le modèle. [7]

La Figure 1.5 illustre un schéma représentant un réseau de Deep Learning composé de plusieurs couches successives.

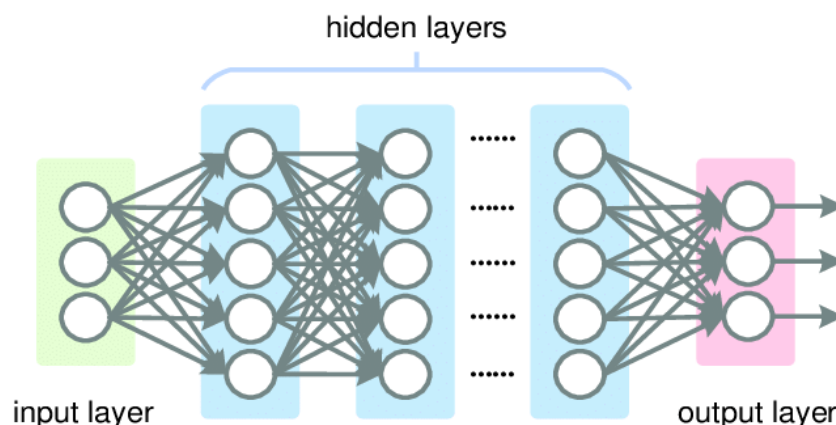


FIGURE 1.5 – Schéma illustratif du Deep Learning à plusieurs couches

1.5.1 Fonctionnement de l'apprentissage profond

Le fonctionnement de l'apprentissage profond repose sur un processus d'ajustement répété des paramètres d'un réseau de neurones à plusieurs couches. Cela se fait grâce à deux méthodes principales : la propagation avant (forward propagation) et la rétropropagation du gradient (backpropagation). Cette structure hiérarchique permet au modèle de déduire des représentations pertinentes à partir de vastes ensembles de données, sans nécessiter d'extraction manuelle ni de traits distinctifs. [8]

Propagation avant (Forward propagation) : La propagation avant désigne le processus par lequel un réseau neuronal fait passer les données d'entrée, couche après couche, de

l'entrée vers la sortie, afin de calculer une valeur de sortie correspondant à la prédiction. [9] Dans ce cadre déterministe, chaque couche effectue une transformation non linéaire. Cela permet d'obtenir, étape par étape, des représentations de plus en plus abstraites. On passe de motifs simples, comme les bords et les textures, à des structures sémantiques plus complexes, comme les objets et les scènes. Tout cela se fait sans extraction manuelle de caractéristiques. Lors de cette phase d'inférence, le modèle produit une prédiction finale qui sert ensuite à calculer l'erreur au moyen d'une fonction de perte, ce calcul constitue l'étape nécessaire avant la mise à jour des paramètres durant l'apprentissage. [10]

Rétropropagation (Backpropagation) : L'apprentissage des réseaux de neurones repose sur la rétropropagation du gradient. C'est elle qui permet d'affiner progressivement les connexions du modèle afin de réduire les erreurs de prédiction. Concrètement, après que le réseau a produit une prédiction, l'écart entre la sortie obtenue et la valeur réelle est mesuré à l'aide d'une fonction de perte. Cette erreur est ensuite propagée en sens inverse à travers les couches du réseau afin d'ajuster les paramètres dans la direction qui minimise l'erreur. Ce processus itératif permet au modèle de s'améliorer progressivement au fil des données. [11]

Ce mécanisme d'apprentissage itératif permet au système de devenir de plus en plus performant dans la reconnaissance faciale.

Dans le cadre de notre projet de gestion des absences par reconnaissance faciale, ce processus permet au modèle de corriger ses erreurs et d'améliorer la précision de l'identification des étudiants.

La Figure 1.6 illustre le fonctionnement du processus d'apprentissage profond basé sur la propagation avant (Forward Propagation) et la rétropropagation (Backpropagation).

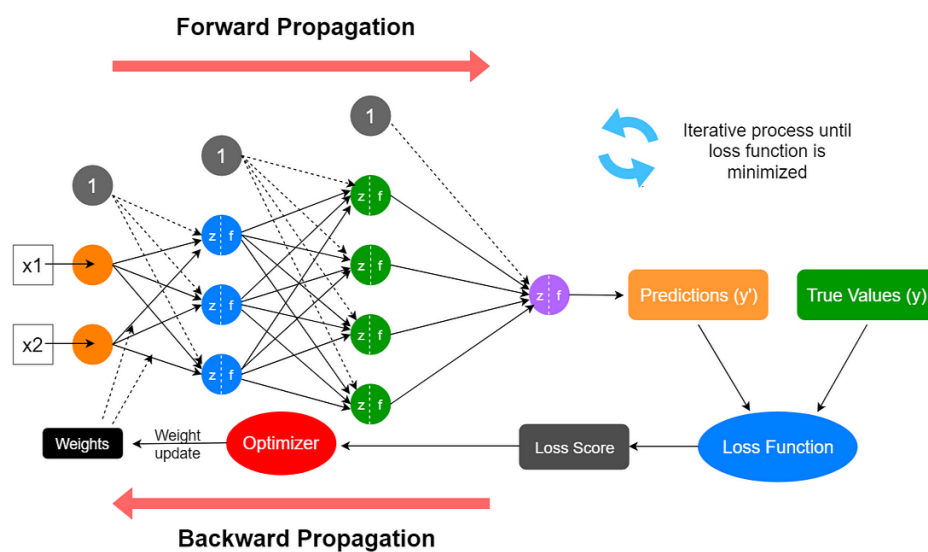


FIGURE 1.6 – Fonctionnement du processus d'apprentissage profond basé sur Forward et Backpropagation

1.5.2 Architectures de Deep Learning

Les architectures de Deep Learning fonctionnent sur des réseaux de neurones artificiels multicouches, destinés à extraire automatiquement des représentations hiérarchisées à partir des données. À l'inverse des techniques traditionnelles de Machine Learning requérant une extraction manuelle des caractéristiques, le Deep Learning offre une approche d'apprentissage end-to-end qui permet d'obtenir les caractéristiques pertinentes directement à partir des données brutes. Cette compétence est à l'origine de son triomphe dans des secteurs comme la vision numérique, l'identification de visages et l'examen d'images médicales. [12].

Plusieurs structures de Deep Learning ont été conçues en tenant compte du type de données et des missions à accomplir. Dans cette partie nous présentons les principales structures architecturales qui sont en lien avec notre travail.

1.5.2.1 Réseaux de neurones multicouches (MLP Multi-Layer Perceptron)

Les MLP (Multilayer Perceptron) constituent une architecture fondamentale du Deep Learning. Ils se caractérisent par une organisation en couches entièrement connectées, Chaque neurone d'une couche est en lien avec l'ensemble des neurones de la couche suivante.

Chaque neurone réalise une combinaison linéaire pondérée de ses entrées, suivie d'une fonction d'activation non linéaire (ReLU, Sigmoid ou Tanh), permettant de modéliser des relations complexes entre les variables.

Grâce à leur structure flexible, les MLP peuvent être optimisés afin d'améliorer la précision et de réduire l'erreur de classification, ce qui les rend adaptés à diverses tâches.

Cependant, leur architecture totalement connectée ne tient pas compte de la structure spatiale des données. Cette limitation réduit leur efficacité pour le traitement d'images, en particulier lorsqu'il s'agit de la reconnaissance faciale. Cela justifie l'utilisation d'architectures plus adaptées aux données visuelles, telles que les réseaux de neurones convolutifs (CNN), présentés ci-après [13]. L'architecture générale d'un MLP est représentée dans la Figure 1.7.

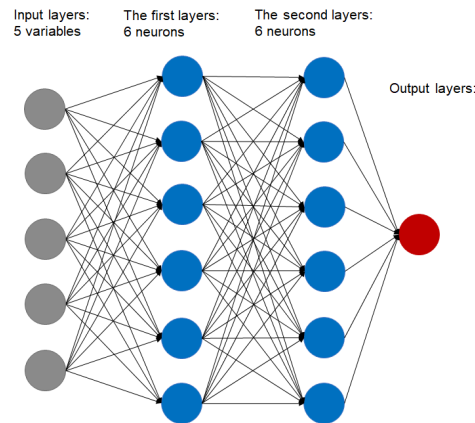


FIGURE 1.7 – Architecture générale d'un réseau MLP

1.5.2.2 Réseaux de neurones convolutifs (CNN Convolutional Neural Networks)

L'architecture avancée de Deep Learning représentant le Réseau de Neurones Convolutifs (CNN) est spécifiquement dédiée à l'analyse et au traitement des données visuelles. Contrairement aux réseaux complètement connectés comme les MLP, les CNN exploitent la structure spatiale intégrée des images, ce qui leur offre la capacité de repérer des motifs locaux tout en minimisant le nombre de paramètres requis pour l'apprentissage. Cette méthode hiérarchique, tirée des réseaux neuronaux du cortex visuel humain, utilise des couches de convolution qui peuvent détecter des caractéristiques de plus en plus abstraites, passant des simples contours et textures aux formes et structures complexes.

Les CNN, grâce à leur capacité d'extraire automatiquement des représentations hiérarchiques, assurent une performance optimale pour diverses tâches de vision par ordinateur comme la reconnaissance d'objets, la segmentation d'images et en particulier la reconnaissance faciale. Cette dernière requiert une précision et une robustesse face aux variations de lumière, d'angle et d'expression qui sont incontournables. Plusieurs recherches récentes attestent que les réseaux de neurones convolutifs (CNN) surpassent les techniques traditionnelles basées sur des descripteurs manuels, grâce à leur capacité à apprendre directement à partir de données non traitées sans nécessiter une étape de prétraitement complexe. [14]. [15]

Architecture typique d'un CNN L'architecture standard d'un CNN se compose de diverses couches interliées, chacune ayant un rôle spécifique dans le traitement de l'image, comme l'illustre la Figure 1.8.

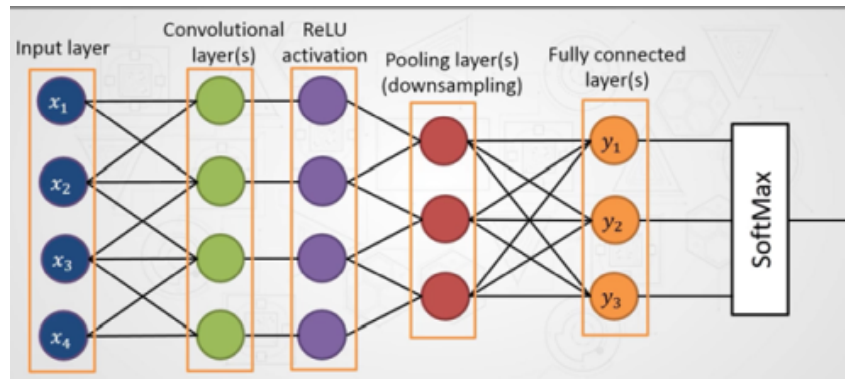


FIGURE 1.8 – Architecture d'un réseau neuronal CNN

Un CNN comprend généralement les couches suivantes :

- **Couche de convolution** : Cette couche utilise des filtres (ou kernels) sur l'image d'entrée pour identifier des motifs locaux comme les contours, les textures ou les formes de base. Chaque filtre génère une carte de caractéristiques illustrant la présence de motifs précis dans diverses zones de l'image, comme l'illustre la Figure 1.9 [16].

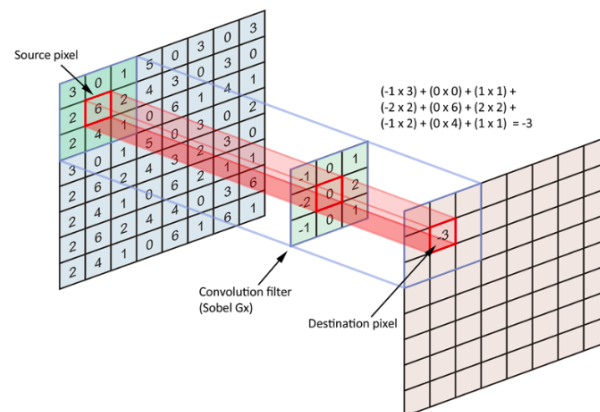


FIGURE 1.9 – Illustration de la convolution dans un réseau CNN

- **Fonction d'activation ReLU** : Suite à la convolution, la fonction d'activation ReLU (Rectified Linear Unit) est mise en oeuvre pour injecter de la non-linéarité dans le modèle, évitant ainsi les complications associées aux relations strictement linéaires. Elle facilite aussi l'accélération de l'apprentissage et l'amélioration du rendement du réseau.
- **Couche de sous-échantillonnage (pooling)** : Le pooling diminue la dimensionnalité spatiale des cartes de caractéristiques tout en préservant l'information la plus significative. Cela réduit la complexité du modèle, prévient le surajustement et renforce sa résilience face aux légers changements de l'image. On utilise fréquemment deux types de pooling : le max et l'average.

TABLEAU 1.4 – Comparaison des types de pooling dans un réseau CNN

Type de pooling	Principe	Objectif / Avantage	Exemple pratique
Max Pooling	Sélectionne la valeur maximale dans chaque fenêtre de pooling	Met en évidence les caractéristiques les plus discriminantes tout en réduisant la dimension des cartes de caractéristiques (<i>feature maps</i>)	Détection des contours ou motifs les plus marqués
Average Pooling	Calcule la moyenne des valeurs présentes dans chaque fenêtre de pooling	Réduit la dimension des données tout en conservant une représentation globale des caractéristiques	Obtention d'une représentation moyenne d'une région de l'image

— **Couches entièrement connectées (Fully Connected) :**

Suite à une série de convolutions, ReLU et pooling, les caractéristiques obtenues sont aplaties pour former un vecteur qui est ensuite dirigé vers des couches entièrement liées. Ces strates intègrent les données globales pour effectuer une analyse à un niveau supérieur.

— **Couche de classification Softmax :**

La couche finale utilise la fonction Softmax pour transformer les résultats du réseau en probabilités attribuées à chaque classe (comme l'identité de l'étudiant). Le modèle attribue donc l'image au visage qui présente la probabilité la plus forte.

- Rôle des CNN dans la reconnaissance faciale Dans les systèmes automatisés basés sur la reconnaissance faciale, les CNN occupent une place cruciale et facilitent trois étapes clés :

— **Identification du visage**

Le visage est localisé et isolé dans l'image ou la vidéo enregistrée par la caméra grâce à l'utilisation initiale du modèle CNN. Il fait la distinction entre le visage et les autres éléments de l'arrière-plan (corps, mur, objets, etc.).

— **Détaillement des traits du visage**

Dès que le visage est repéré, le CNN passe à l'analyse de l'image et dégage automatiquement des traits distinctifs comme la configuration des yeux, du nez, de la bouche et la structure globale du visage. Ces traits forment une empreinte distincte pour chaque personne.

— **Identification de l'étudiant(e)** Les caractéristiques déduites sont comparées avec celles stockées dans la base de données du système. Le modèle attribue par la suite

l'identité la plus probable à l'image examinée et signale automatiquement la présence de l'étudiant. Ce mécanisme complet est schématisé dans la Figure 1.10.

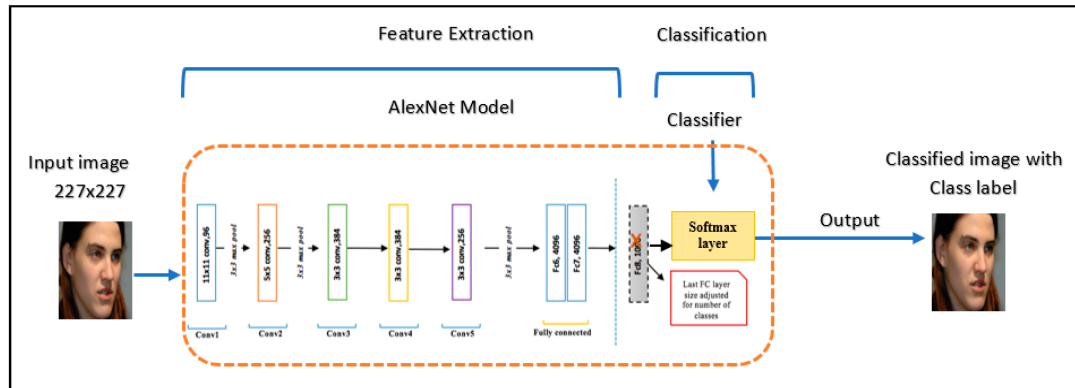


FIGURE 1.10 – Processus de reconnaissance faciale

1.6 Reconnaissance faciale

1.6.1 Définition de la reconnaissance faciale

La reconnaissance faciale est une technologie biométrique basée sur l'intelligence artificielle et la vision par ordinateur, permettant d'identifier ou de vérifier l'identité d'un individu à partir de son visage. À la différence des techniques biométriques traditionnelles, telles que celles basées sur les empreintes digitales ou l'iris, la reconnaissance faciale s'appuie sur l'analyse d'images ou de flux vidéo pour identifier automatiquement des traits distinctifs propres à chaque individu.

Actuellement, la reconnaissance faciale est largement utilisée dans divers secteurs, tels que la sécurité, la surveillance intelligente, les systèmes d'identification et les milieux éducatifs intelligents. [17].

1.6.2 Étapes du système de reconnaissance faciale

Un système moderne de reconnaissance faciale fondé sur l'apprentissage profond suit généralement un processus organisé en différentes phases, comme le montre la Figure 1.11.

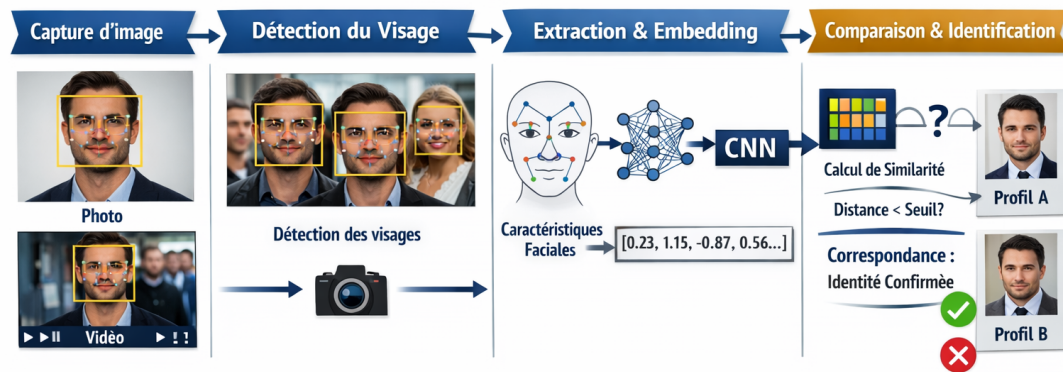


FIGURE 1.11 – Étapes d'un système de reconnaissance faciale

1.6.3 Capture d'image

La reconnaissance faciale débute par l'acquisition d'images fixes ou de séquences vidéo. Les images statiques offrent une meilleure qualité pour l'extraction précise des caractéristiques, tandis que les vidéos apportent des informations temporelles utiles pour gérer les variations d'éclairage et de pose. Cependant, la qualité parfois limitée des vidéos peut réduire les performances. Le choix entre ces deux modes dépend principalement de la résolution disponible, des ressources de calcul et du nombre d'images exploitables. [18]

1.6.4 Détection du visage

L'identification du visage constitue une phase initiale essentielle qui vise à repérer de manière automatique les zones faciales dans une image ou un enregistrement vidéo. Les techniques conventionnelles utilisant des caractéristiques manuelles ont progressivement cédé la place à des méthodes basées sur l'apprentissage profond. Des architectures telles que MTCNN (Multi-task Cascaded Convolutional Networks) [19] ou RetinaFace analysent l'image en temps réel pour délimiter chaque visage à l'aide d'un cadre englobant (*bounding box*) et localiser des points caractéristiques (yeux, nez, coins de la bouche) [20]. Cette étape est cruciale pour garantir la fiabilité du pipeline complet de reconnaissance faciale, comme le détaille la Figure 1.12.

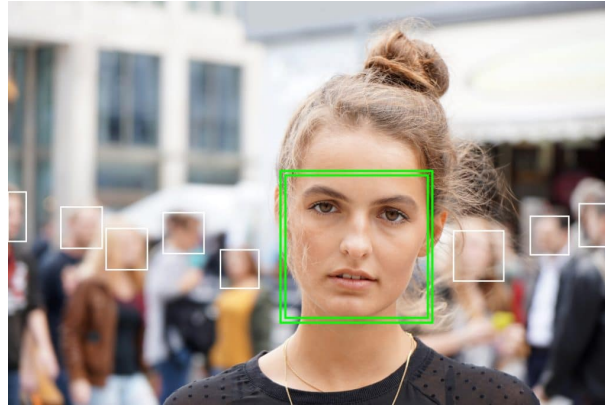


FIGURE 1.12 – Exemple de détection automatique de visages

1.6.5 Extraction des caractéristiques et conversion en vecteur (embedding)

Après la détection du visage, les traits distinctifs (comme l'écartement des yeux, la forme du nez ou des pommettes) sont préparés pour l'étape suivante. Le réseau neuronal convolutif (CNN) utilise ces informations pour les convertir en un vecteur numérique, connu sous le nom d'*embedding*, qui traduit mathématiquement l'image du visage. Cette représentation facilite une comparaison efficace de divers visages dans un espace vectoriel et représente l'étape cruciale pour l'identification. Ce processus de transformation est schématisé sur la Figure 1.13.

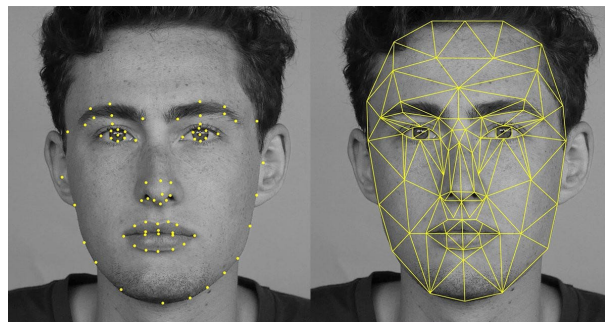


FIGURE 1.13 – Extraction des caractéristiques faciales à partir des points clés (landmarks)

1.6.6 Comparaison et identification

L'étape de comparaison repose sur l'évaluation de la similarité entre les vecteurs caractéristiques (embeddings) faciaux extraits. Cette similarité est généralement mesurée à l'aide de métriques de distance telles que la distance euclidienne (L2) ou la distance de Manhattan (L1), qui quantifient l'écart entre deux représentations vectorielles dans l'espace des caractéristiques. Lorsque la distance calculée est inférieure à un seuil prédéfini, l'identité est considérée comme correspondante. Cette procédure permet d'assurer la reconnaissance automatique des étudiants au sein d'un système de gestion des présences.

1.6.7 Avantages de la reconnaissance faciale

La technologie de reconnaissance faciale aide à la digitalisation des institutions éducatives en automatisant le contrôle d'assiduité. Elle optimise l'efficacité et la précision des enregistrements tout en diminuant les charges administratives [21]. L'intégration de cette technologie apporte plusieurs bénéfices stratégiques pour les établissements d'enseignement :

- **Prise de présence automatique** : La technologie de reconnaissance faciale facilite l'automatisation de l'enregistrement des présences en reconnaissant les étudiants via des modèles d'apprentissage profond, tels que les CNN. Cette automatisation diminue les erreurs faites par l'homme et améliore le temps dédié aux actions pédagogiques.
- **Réduction de la fraude** : Contrairement aux méthodes classiques d'appel ou de pointage manuel, souvent sujettes à des substitutions ou à des fraudes (comme un étudiant signant pour un autre), la technologie de reconnaissance faciale fournit une identification biométrique précise pour chaque personne. Cette automatisation diminue fortement les fraudes et renforce la fiabilité des données de présence.
- **Gain de temps** : Les systèmes automatisés offrent un gain de temps important pour les enseignants et les responsables éducatifs. Répondre aux appels, qui prenait des minutes avec les méthodes classiques, se fait maintenant en quelques secondes, même dans de grandes classes. Ce gain de temps facilite aussi la gestion des données pour des analyses futures et des actions pédagogiques plus rapides.
- **Traçabilité et analyse des données** : Les données de reconnaissance faciale sont organisées et datées. Cela facilite les rapports sur la présence, l'absence et les tendances. On peut aussi élaborer des initiatives pour accroître l'engagement des étudiants. Cette traçabilité améliore le suivi pédagogique et la prise de décisions basées sur les données.

1.6.8 Limites de la reconnaissance faciale

Bien qu'il y ait de nombreux bénéfices à utiliser la reconnaissance faciale pour le suivi des absences, il est essentiel de prendre en compte ses limitations tant techniques qu'organisationnelles. L'étude des limites aide à saisir les conditions dans lesquelles ce système peut être implanté de manière efficace :

- Sensibilité aux variations de lumière, d'angle de vue et d'expression faciale ;
- Coût initial élevé lié à l'acquisition et à la mise en place du matériel ;
- Enjeux éthiques et contraintes liées à la protection des données personnelles, nécessitant le respect des réglementations en vigueur ;

- Dépendance à la qualité des images capturées, influençant directement la précision de l'identification.

Par ailleurs, les systèmes sont vulnérables aux attaques de spoofing utilisant des photos imprimées, des vidéos enregistrées ou des masques, et disposaient de peu de moyens pour les détecter. Actuellement, les méthodes de détection utilisent des modèles d'apprentissage profond qui peuvent examiner des traits du visage et des actions humaines, tout en étant capables de s'adapter à des attaques non identifiées. Cependant, la plupart des études se focalisent toujours sur des attaques basiques, telles que les images fixes (photographies RGB) ou les vidéos de relecture présentant un visage capturé. [22]

Par conséquent, même si la reconnaissance faciale représente une solution avant-gardiste et efficace, son application doit respecter ces restrictions afin d'assurer précision, fiabilité ainsi que conformité aux standards éthiques et juridiques. Ces contraintes guident aussi les améliorations futures et les ajustements indispensables pour une intégration optimale dans le cadre éducatif.

1.7 Conclusion

Ce chapitre a permis de présenter les fondements théoriques de l'intelligence artificielle ainsi que les principales approches d'apprentissage automatique et d'apprentissage profond. Une attention particulière a été accordée aux réseaux de neurones convolutifs, en raison de leur rôle central dans les applications de vision par ordinateur, notamment la reconnaissance faciale.

L'analyse des différents concepts et architectures a montré que ces technologies constituent une base solide pour le développement de systèmes intelligents capables d'automatiser la gestion des présences avec un haut niveau de précision et de fiabilité.

Toutefois, certaines limites subsistent, notamment en lien avec les conditions d'acquisition des images, les contraintes matérielles et les enjeux liés à la protection des données personnelles. La prise en compte de ces aspects est essentielle pour garantir une utilisation optimale et éthique de ces systèmes.

Le chapitre suivant sera consacré à l'étude de l'état de l'art des méthodes de détection et de reconnaissance faciale, afin d'identifier les approches les plus adaptées à notre problématique.

Chapitre 2

État de l'art des systèmes de reconnaissance
faciale

2.1 Introduction

La reconnaissance faciale constitue aujourd'hui l'une des technologies les plus prometteuses pour l'automatisation des processus d'identification, notamment dans le domaine éducatif. Grâce aux avancées récentes en vision par ordinateur et en apprentissage profond, les systèmes de détection et de reconnaissance de visages ont connu des améliorations significatives en termes de précision, de robustesse et de rapidité d'exécution.

L'objectif de ce chapitre est de présenter une synthèse critique des principaux travaux scientifiques et des approches algorithmiques utilisées dans les systèmes modernes de reconnaissance faciale, afin d'identifier les méthodes les plus adaptées à la conception d'un système fiable de gestion automatisée de la présence des étudiants.

Pour répondre à cet objectif, le chapitre s'articule autour de quatre axes principaux :

- Étudier les algorithmes de détection faciale (Haar Cascade, HOG, MTCNN et YOLOv8), en présentant leurs principes de fonctionnement ainsi que leurs performances et limites opérationnelles .
- Analyser les méthodes de reconnaissance d'identité (LBPH, FaceNet et Dlib ResNet), en mettant l'accent sur les techniques d'extraction de caractéristiques et de comparaison .
- Réaliser une étude comparative et une validation expérimentale, incluant la reproduction des résultats de Lateef et Kamil [23] ainsi que nos propres expérimentations avec YOLOv8, afin d'évaluer les compromis entre précision, robustesse et coût computationnel .
- Synthétiser les résultats obtenus afin de justifier les choix de conception retenus pour le pipeline de notre système.

Cette analyse permettra d'identifier les avantages et les limites de ces technologies dans le contexte spécifique de la gestion automatisée des présences en milieu universitaire. Les résultats de cette étude comparative orienteront les choix méthodologiques et techniques adoptés dans la conception du système proposé, qui sera présenté en détail dans le chapitre suivant.

2.2 Méthodes de détection des visages dans une image

La détection constitue la première étape critique du pipeline de reconnaissance. Quatre principales approches dominent la littérature récente :

2.2.1 Haar Cascade (ViolaJones)

La méthode Haar Cascade, proposée par Viola et Jones, constitue l'une des premières approches efficaces de détection faciale en temps réel. Elle repose sur l'extraction de ca-

ractéristiques rectangulaires de type Haar combinées à l'algorithme d'apprentissage Ada-Boost, permettant de sélectionner les descripteurs les plus discriminants.

Principe de fonctionnement : Le processus débute par la conversion de l'image en image intégrale, ce qui permet d'accélérer le calcul des différences d'intensité entre régions adjacentes. L'algorithme applique ensuite une fenêtre glissante sur l'ensemble de l'image afin d'extraire les caractéristiques de Haar pour chaque sous-région. Un classifieur en cascade est alors utilisé pour rejeter progressivement les zones non pertinentes et conserver uniquement celles susceptibles de contenir un visage. Cette structure hiérarchique optimise considérablement le temps de calcul.

Algorithme Haar Cascade :

Algorithme 1 : Détection de visage par Haar Cascade (Viola–Jones) [24]

Input : I : image en niveaux de gris, M : modèle Haar**Output** : D : coordonnées du visage détecté, I_{crop} : visage recadré

Début- Calculer l'image intégrale $II(x, y)$ telle que :

$$II(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} I(x', y')$$

- **Pour** chaque fenêtre glissante W de taille $w \times h$:

- Calculer la valeur d'une caractéristique Haar :

$$f = \sum_{\text{zone blanche}} II - \sum_{\text{zone noire}} II$$

- Initialiser les poids $w_i = \frac{1}{N}$ - **Pour** chaque itération t :

$$\varepsilon_t = \sum_i w_i |h_t(x_i) - y_i|$$

- Mettre à jour les poids :

$$w_i \leftarrow w_i \beta_t^{(1-e_i)}$$

- **Pour** chaque étage k du classifieur :

$$H_k(x) = \begin{cases} 1 & \text{si } \sum \alpha_t h_t(x) \geq \theta_k \\ 0 & \text{sinon} \end{cases}$$

- Rejeter immédiatement les régions non faciales

- Déterminer les coordonnées finales $D = (x, y, w, h)$

- Extraire la région :

$$I_{crop} = I(x : x + w, y : y + h)$$

Fin

Performances et applications : Suna ryono et al . [25] ont montré l'efficacité de cette méthode dans le domaine des applications mobiles pour la reconnaissance faciale à partir d'images capturées par smartphone, mettant en avant sa rapidité d'exécution. De même, Alniemi et Mahmood. [26] rapportent un temps de traitement inférieur à 50 ms en conditions réelles, ce qui confirme son adéquation avec les systèmes embarqués à ressources limitées.

Limites : Malgré sa rapidité, plusieurs travaux soulignent ses limites. Sa et al. [27]

observent une diminution notable de la précision en présence de variations d'éclairage (entre 15 et 22 points). Par ailleurs, Lateef et Kamil [23] indiquent une dégradation des performances lorsque les visages sont éloignés ou présentent des variations importantes de pose. Ces résultats montrent que la méthode demeure sensible aux conditions environnementales et aux changements d'orientation.

2.2.2 HOG (Histogram of Oriented Gradients)

La méthode HOG repose sur l'analyse des gradients d'intensité afin de capturer les contours et la structure locale des objets, notamment des visages. Lateef et Kamil (2023) [23] combinent HOG avec le détecteur Dlib (Bibliothèque Python), basé sur une architecture ResNet-34 [28], afin d'améliorer la robustesse de la détection faciale.

Principe de fonctionnement : L'image est d'abord divisée en petites cellules. Pour chaque cellule, les orientations des gradients sont calculées, puis un histogramme représentant les directions dominantes est construit. Ces histogrammes sont ensuite normalisés sur des blocs plus larges afin de réduire l'effet des variations d'éclairage et des contrastes locaux. La représentation finale est transmise à un classifieur, souvent un SVM ou un réseau de neurones profond, pour identifier la présence d'un visage. Cette approche permet de capturer efficacement la structure locale du visage tout en restant robuste aux petites variations de position ou d'expression.

Algorithme HOG :

Algorithme 2 : Détection de visage par HOG [29]

Input : Image I (couleur ou niveaux de gris)**Output** : Liste $Faces$ des coordonnées des visages détectés

Début

- Convertir I en niveaux de gris $\rightarrow Gray$
- **Pour** chaque pixel de $Gray$:
 - Calculer le gradient horizontal Gx et vertical Gy
 - Calculer la magnitude $M = \sqrt{Gx^2 + Gy^2}$
 - Calculer l'orientation $\theta = \arctan(Gy/Gx)$
- Diviser $Gray$ en cellules de petite taille (ex. 8 x 8 pixels)
- **Pour** chaque cellule :
 - Calculer un histogramme des orientations de gradient
 - Pondérer chaque pixel par sa magnitude M
- Regrouper les cellules en blocs plus larges
- Normaliser les histogrammes pour réduire l'effet des variations d'éclairage et de contraste
- Concaténer les histogrammes normalisés de tous les blocs $\rightarrow HOG_Feature$
- **Pour** chaque fenêtre glissante dans l'image (multi-échelle) :
 - Extraire le vecteur HOG de la fenêtre.
 - Appliquer un classifieur (SVM ou réseau de neurones).
 - Si un visage est détecté : Ajouter les coordonnées à $Faces$.
- Retourner la liste $Faces$.

Fin

Performances et applications : Selon Lateef et Kamil [23], cette combinaison HOG + Dlib permet une détection fiable jusqu'à une distance d'environ 7 mètres, avec une précision de 94% sur un échantillon de 31 étudiants dans des conditions réelles de classe. Cette méthode s'avère donc particulièrement adaptée aux environnements d'enseignement de taille moyenne, offrant un bon compromis entre précision et complexité computationnelle.

Limites : Les performances diminuent significativement au-delà de 10 mètres, distance courante dans les grands amphithéâtres universitaires. De plus, HOG reste sensible aux variations importantes de pose, d'angle de vue ou d'éclairage, ce qui limite son applicabilité dans les environnements de grande capacité et en conditions d'éclairage difficile.

2.2.3 MTCNN (Multi-task Cascaded CNN)

MTCNN est une architecture profonde composée de trois réseaux successifs (P-Net, R-Net et O-Net) permettant de détecter le visage et de localiser cinq points caractéristiques (yeux, nez et coins de la bouche) [23]. Lateef et Kamil [23] rapportent que l'association de MTCNN avec FaceNet atteint une précision de 100% sur un échantillon de 31 étu-

dians en conditions réelles. La méthode reste performante même lorsque les visages sont partiellement masqués ou incliné jusqu'à des angles importants, grâce à l'apprentissage automatique des caractéristiques qui rend la détection plus robuste face aux variations que les approches classiques.

Principe de fonctionnement : La première phase, assurée par le P-Net (*Proposal Network*), a pour vocation de générer un ensemble initial de régions d'intérêt candidates (*Regions of Interest* - ROI) à partir d'une analyse pyramidale de l'image d'entrée. La deuxième phase, confiée au R-Net (*Refine Network*), procède à un raffinement de ces propositions par un classement plus discriminant et une suppression des détections erronées (*false positives*). La troisième et dernière phase, implémentée par le O-Net (*Output Network*), effectue une régulation fine des coordonnées des boîtes englobantes (*bounding box regression*) et réalise simultanément la localisation des cinq points de repère faciaux (*facial landmarks*), à savoir les deux yeux, l'extrémité du nez et les deux commissures des lèvres [30]. Ce traitement en cascade améliore progressivement la précision tout en maintenant un temps de calcul raisonnable, comme le schématise la Figure 2.1.

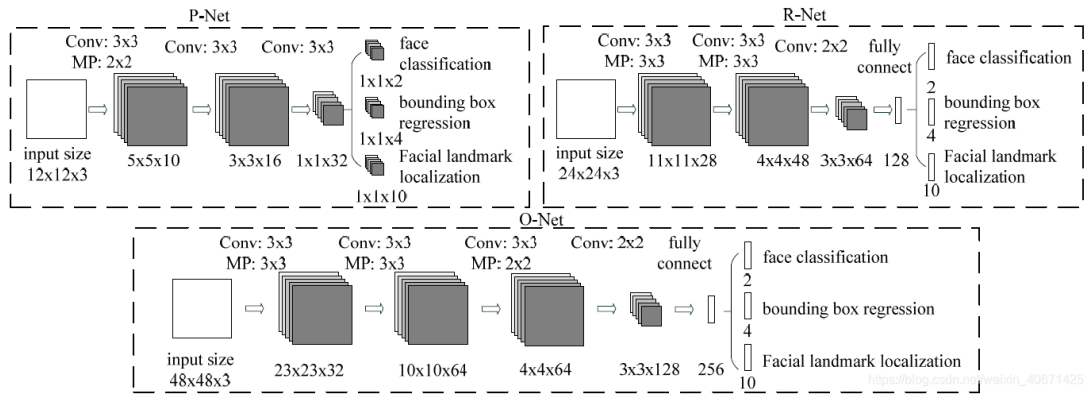


FIGURE 2.1 – Architecture du réseau MTCNN

Algorithme MTCNN :

Algorithme 3 : Détection de visage par MTCNN [31]

Input : I : Image originale**Output** : D : Coordonnées des visages détectés et points de repère faciaux

Début**Phase 1 : P-Net (Proposal Network)**

- Générer une pyramide d'images à différentes échelles
- **Pour** chaque échelle, détecter les régions candidates (ROI) pour les visages
- Conserver les propositions avec scores supérieurs à un seuil prédéfini

Phase 2 : R-Net (Refine Network)

- Prendre les ROI proposées par P-Net.
- Raffiner les boîtes englobantes et recalculer les scores de détection
- Supprimer les fausses détections (false positives)

Phase 3 : O-Net (Output Network)

- **Pour** chaque ROI restante, ajuster finement les coordonnées des boîtes englobantes (bounding box regression)
- Localiser les cinq points de repère faciaux : Les deux yeux, L'extrémité du nez, Les deux commissures des lèvres

Retourner D : ensemble des visages détectés avec leurs points de repère**Fin**

Performances et applications : MTCNN se distingue par son efficacité dans les environnements d'enseignement et les systèmes de surveillance en temps réel, en maintenant une grande précision même en présence de masques partiels ou de fortes inclinaisons du visage, Elle peut être combinée avec des réseaux d'embedding, comme FaceNet, pour la reconnaissance faciale et l'identification d'individus dans des bases de données.

Limites : Bien que robuste, MTCNN peut voir sa précision diminuer pour des visages très éloignés ou dans des conditions d'éclairage extrêmes. Le traitement en cascade, bien qu'optimisé, reste plus coûteux en termes de ressources que des méthodes plus simples comme HOG ou Haar Cascade.

2.2.4 YOLO (You Only Look Once)

YOLOv8 est un algorithme récent (2025) de détection d'objets en temps réel qui analyse l'image en une seule passe (*single forward pass*), contrairement aux méthodes basées sur des fenêtres glissantes. Cette approche prédit simultanément les boîtes englobantes et les classes des objets, ce qui permet un traitement rapide et efficace. Selon Wei et al. [32], YOLOv8 atteint une précision moyenne (mAP@0.5) de 98,11 % sur le dataset Mask-Imgs, composé de 6 385 images collectées en scènes réelles pour la détection de port de masque, tout en maintenant une vitesse de 135,61 images par seconde. Ces performances,

obtenues malgré des occlusions et des variations d'éclairage, confirment l'adaptabilité de l'architecture YOLOv8 aux environnements denses comme les amphithéâtres, où la détection simultanée de plusieurs visages et la réactivité sont essentielles pour les systèmes automatisés de gestion des présences.

Principe de fonctionnement : L'image est divisée en une grille. Pour chaque cellule, le réseau neuronal prédit simultanément les coordonnées des boîtes englobantes, la probabilité de présence d'un objet et sa classe. Cette approche en propagation unique permet d'effectuer la détection très rapidement, jusqu'à 45 images par seconde sur GPU, ce qui constitue l'un des principaux avantages de YOLO par rapport aux méthodes traditionnelles. Le pipeline complet du système basé sur YOLOv8 est illustré dans la Figure 2.2 [33].

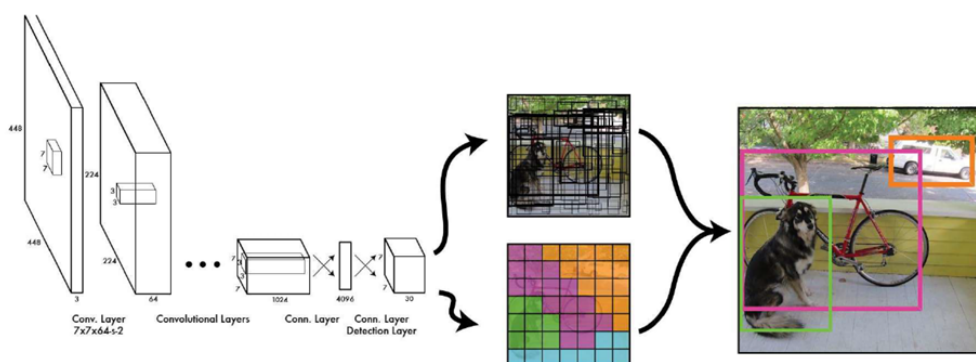


FIGURE 2.2 – Pipeline du système basé sur YOLOv8

Algorithme YOLOv8 :

Algorithme 4 : Algorithme YOLOv8 [34]

Input : M : Modèle YOLOv8 pré-entraîné DS : Échantillons du jeu de données personnalisé E : Nombre d'époques BS : Taille du batch**Output :** W_f : Fichier des poids entraînés

Début**Étape 1 :** Assigner les échantillons au jeu de données personnalisé DS **Étape 2 :** Initialiser les poids du modèle

- Charger le fichier de poids initial pré-entraîné

Étape 3 : Configurer les hyperparamètres du modèle selon le Tableau 2**Étape 4 :** Définir le nombre de batches

$$Batch_Number = \frac{|DS|}{BS}$$

Étape 5 : Entraîner le modèle pour chaque époque $EpNo \leftarrow 0$ **while** $EpNo < E$ **do** $EpNo \leftarrow EpNo + 1$ $BaNo \leftarrow 0$ **while** $BaNo < Batch_Number$ **do** $BaNo \leftarrow BaNo + 1$ Extraire un batch d'images de taille BS

Passer le batch courant à travers les couches du modèle

Enregistrer les sorties des cellules sélectionnées et leurs classes associées

Calculer la fonction de perte à partir des annotations réelles

Appliquer la méthode d'optimisation (mise à jour des poids)

end

Évaluer le modèle sur l'ensemble de validation

end**Étape 6 :** Sauvegarder et retourner le fichier des poids entraînés W_f **Fin**

Performances et applications : YOLO se révèle particulièrement adapté aux systèmes de surveillance et aux environnements éducatifs, offrant la possibilité de détecter plusieurs individus en temps réel. Son usage est idéal lorsque la rapidité de traitement est critique, notamment pour des applications interactives ou de suivi automatisé des présences.

Limites : Le principal inconvénient de YOLO réside dans sa dépendance à un GPU

pour atteindre ses performances optimales. Cette exigence matérielle limite son déploiement sur des dispositifs embarqués à faible coût, tels que le Raspberry Pi, et peut constituer un obstacle dans des environnements avec des ressources informatiques limitées.

2.3 Méthodes de reconnaissance faciale

La reconnaissance faciale constitue l'étape suivante après la détection et l'alignement des visages. Elle consiste à tirer des traits discriminants et à les comparer avec une base de données préexistante pour identifier l'individu. La littérature récente met en évidence trois approches dominantes :

2.3.1 Local Binary Pattern Histogram (LBPH) :

LBPH (Local Binary Pattern Histogram) est un opérateur de texture très précis qui convertit chaque pixel en valeur binaire en le comparant à ses voisins directs, générant un histogramme représentant les motifs texturaux du visage [26]. Cette technique est particulièrement résistante aux variations d'éclairage et ne nécessite pas de calculs mathématiques complexes, ce qui la rend particulièrement adaptée aux systèmes embarqués à faible coût.

Principe de fonctionnement : L'algorithme convertit d'abord l'image en niveaux de gris, puis calcule pour chaque pixel une valeur binaire selon la fonction $F(x) = 1$ si $x \geq 0$, sinon 0. Le LBP est ensuite calculé en encerclant le pixel central avec des valeurs binaires et en convertissant le résultat en nombre décimal selon l'équation :

$$\text{LBP}_{P,R}(g_c) = \sum_{p=0}^{P-1} 2^p F(g_p - g_c)$$

$$R = \sqrt{(x_p - x_c)^2 + (y_p - y_c)^2}$$

où g_c représente la valeur du pixel central, g_p la valeur du p^e pixel voisin, P le nombre de voisins, R le rayon de l'opérateur (distance entre le pixel central et chacun de ses voisins), avec $p = 0, 1, \dots, P-1$. L'image transformée est ensuite représentée par un histogramme de fréquences des motifs LBP. La similarité entre deux images est évaluée en comparant leurs histogrammes à l'aide de mesures de distance, telles que la distance euclidienne ou la distance basée sur les valeurs absolues [23]. Le principe de l'opérateur LBP appliqué à une image faciale est illustré dans la Figure 2.3.

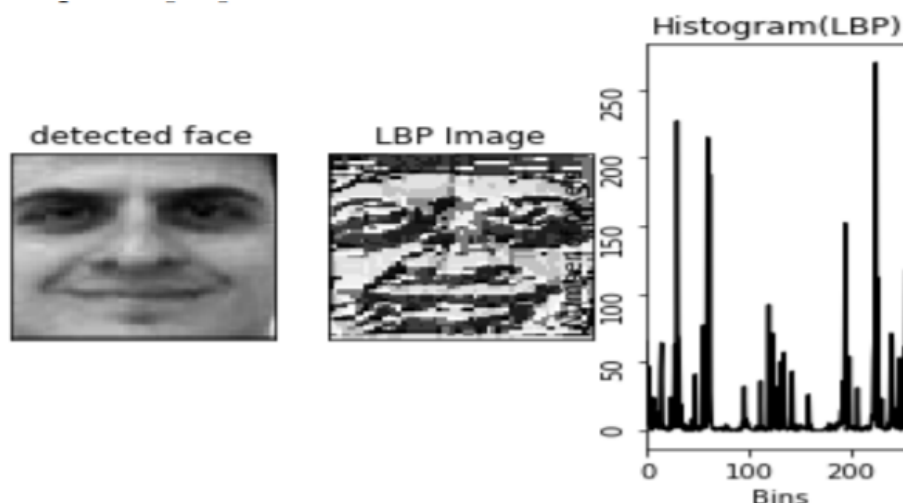


FIGURE 2.3 – Principe de l'opérateur LBP appliqué à une image faciale

Performances et applications : D'après le travail de lateef et Kamil, cette méthode permet d'atteindre 91 % de précision, 97 % de rappel(Recall), et de F1-score 96 % sur un échantillon de 31 étudiants évalués en conditions réelles de classe. Les résultats montrent que la méthode est particulièrement performante pour la reconnaissance de visages frontaux capturés à courte distance (inférieure à 5 mètres), avec des temps de traitement rapides adaptés aux systèmes intégrés [23]. Elle est spécialement employée dans les systèmes économiques tels que Raspberry Pi [35].

Limites : Deux limitations majeures ont été relevées dans l'étude : d'une part, une difficulté à reconnaître les visages non frontaux ou partiellement occultés, notamment lorsque les conditions d'éclairage ou la distance affectent la qualité de l'image ; d'autre part, la présence de faux positifs, illustrée par des cas d'erreurs d'identification tels qu'un enseignant classé comme étudiant et une étudiante incorrectement reconnue .[23]. Lateef et Kamil indiquent que ces limitations peuvent être atténuées par l'amélioration de la qualité et de la diversité des images d'entraînement, ainsi que par l'application de techniques de prétraitement, telles que l'égalisation d'histogramme, afin d'optimiser le contraste et la normalisation de l'éclairage [23].

2.3.2 Algorithme FaceNet :

FaceNet est un système de reconnaissance faciale créé par Google en 2015, permettant de produire à partir d'images de visages des vecteurs d'embedding de haute qualité. Ces vecteurs permettent non seulement une vérification d'identité, mais aussi l'identification et le regroupement de visages avec une grande précision.

Principe de fonctionnement : FaceNet fonctionne en utilisant un réseau neuronal profond de 22 couches pour mapper les images faciales dans un espace euclidien de faible dimension. Les distances entre les points dans cet espace correspondent à une mesure de

similarité faciale. FaceNet adopte une approche basée sur les données, en apprenant directement les représentations faciales à partir des pixels du visage. Les images de la même personne sont proches dans cet espace, tandis que celles de personnes différentes sont éloignées. Cette méthode permet à FaceNet de rester relativement robuste face aux variations de pose et de luminosité, sans nécessiter un alignement manuel préalable. Toutefois, en pratique, il est fréquemment couplé à un algorithme de détection tel que MTCNN afin d'améliorer la robustesse globale du système. . [26]

Performances et applications : Couplé à l'algorithme MTCNN pour la phase de détection, FaceNet atteint une précision de 100 % sur un échantillon de 31 étudiants évalués en conditions réelles, incluant des distances comprises entre 1 et 10 mètres, des variations de pose pouvant atteindre $\pm 60^\circ$, ainsi que des conditions d'éclairage variables. Le système démontre également une capacité remarquable à reconnaître des visages partiellement occultés (par exemple, mains couvrant partiellement le visage ou yeux fermés) et situés à des distances supérieures à 10 mètres [23]. Cette robustesse élevée en environnement non contrôlé en fait une solution particulièrement adaptée aux contextes éducatif.

Limites : Malgré ses bonnes performances, FaceNet présente certaines limites. Son entraînement nécessite une grande puissance de calcul, ce qui peut compliquer son utilisation sur des dispositifs simples [36]. De plus, ses performances dépendent fortement de la qualité et de la diversité des images utilisées pour l'apprentissage. Si les données ne sont pas variées ou suffisantes, la précision du système peut diminuer. Enfin, des conditions difficiles comme un éclairage très faible ou des occlusions importantes peuvent encore affecter ses performances.

2.3.3 Modèle Dlib basé sur ResNet

Cette approche repose sur l'architecture ResNet-34 entraînée selon une stratégie d'apprentissage métrique par fonction de perte triplet (triplet loss). Pour chaque identité, le réseau utilise trois images : une image ancre (anchor), une image positive (même identité que l'ancre) et une image négative (identité différente) [23]. Le réseau génère un vecteur d'embedding de 128 dimensions en minimisant la distance euclidienne entre l'ancre et l'image positive, tout en maximisant la distance entre l'ancre et l'image négative, avec une marge de séparation prédéfinie. La fonction de perte est définie comme suit :

$$\mathcal{L} = \max \left(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha, 0 \right)$$

où :

- A est l'entrée d'ancrage (*anchor*), que l'on veut tester,
- P est une entrée positive de la même classe que A ,
- N est une entrée négative d'une classe différente de A ,

- α correspond à une marge entre les couples positifs et négatifs,
- f Fonction de projection du réseau (vecteur 128-D), (*embedding function*).

Cette optimisation permet de projeter les visages d'une même personne dans des régions proches de l'espace de caractéristiques, tout en éloignant ceux de personnes différentes [36], Le principe de cette fonction de perte est représenté dans la Figure 2.4.

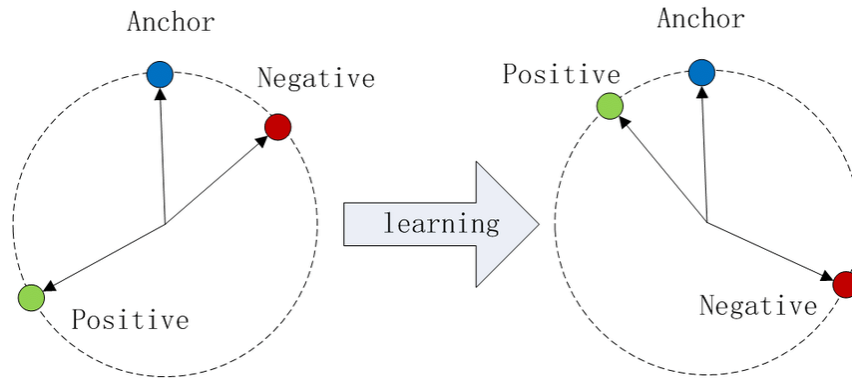


FIGURE 2.4 – Principe de la Triplet Loss

Principe de fonctionnement : Associé à l'algorithme de détection HOG pour la localisation des visages, ce modèle s'appuie sur l'extraction de 68 points caractéristiques (notamment les yeux, le nez et la bouche) afin d'assurer l'alignement préalable des visages avant la phase d'extraction des embeddings. Cette étape d'alignement permet de normaliser la position et l'orientation du visage, améliorant ainsi la robustesse du système face aux variations de pose. Les représentations vectorielles générées sont ensuite comparées à l'aide de mesures de similarité, telles que la distance euclidienne ou la similarité cosinus, afin d'identifier l'individu correspondant . [23]

Performances et applications : Grâce à cette architecture, le système atteint une précision de reconnaissance satisfaisante dans des conditions contrôlées, avec un temps de traitement compatible avec des applications en temps réel. Cette méthode est particulièrement adaptée aux scénarios de gestion de présence en salle de classe, où les distances de capture sont modérées (1 à 5 mètres) et les angles de vue relativement frontaux. Comme l'ont démontré Lateef et Kamil , l'association de Dlib avec un détecteur HOG permet un déploiement efficace sur des matériels standards, sans nécessiter de ressources GPU importantes, ce qui en fait une solution économiquement viable pour les établissements éducatifs. [23]

Limites : La contrainte majeure de cette méthode ne provient pas de l'algorithme de reconnaissance, mais plutôt de la phase de détection gérée par HOG. Effectivement, cette technique présente des difficultés pour détecter les visages situés à grande distance (plus de 10 mètres) ou fortement inclinés. Selon Lateef et Kamil, cette restriction est intrinsèque à l'algorithme de détection HOG et non au système de reconnaissance qui s'appuie sur Dlib. Par conséquent, la collaboration stratégique entre HOG pour la détection et Dlib

pour la reconnaissance semble appropriée pour des scénarios à distance modérée, typiques des environnements de classe, en tenant compte des contraintes spécifiques liées à l'étape de localisation des visages. [23]

2.4 Étude comparative

Cette section présente une étude comparative des différentes approches de détection et de reconnaissance faciale implémentées dans ce travail. Elle vise à évaluer et comparer les performances de plusieurs algorithmes selon des critères de précision, de robustesse et d'efficacité computationnelle. Dans ce cadre, nous présentons les métriques d'évaluation retenues ainsi que les résultats expérimentaux obtenus à partir de nos implémentations.

2.4.1 Métriques de comparaison

Afin de faciliter l'interprétation des résultats obtenus, les principales métriques utilisées sont adaptées au contexte spécifique de la gestion automatisée des présences. Le tableau 2.1 présente la signification pratique de chaque indicateur dans le cadre de notre application.

TABLEAU 2.1 – Interprétation des métriques dans le contexte de la gestion automatisée des présences

Métrique	Signification pratique
Exactitude (Accuracy)	Pourcentage global d'étudiants correctement identifiés, incluant les présences validées et les absences correctement détectées.
Rappel (Recall)	Capacité du système à détecter tous les étudiants présents, en minimisant les faux négatifs.
Précision (Precision)	Fiabilité des identifications réalisées par le système, en réduisant les erreurs d'attribution entre étudiants.
F1-Score	Mesure synthétique représentant le compromis entre la précision et le rappel du système.

Définitions des termes fondamentaux : Pour l'analyse des résultats de classification, nous considérons les quatre catégories suivantes :

- **Vrai Positif (TP)** : Étudiant présent dans la base de données et correctement identifié par l'algorithme.
- **Faux Positif (FP)** : Étudiant incorrectement identifié comme étant une autre personne enregistrée (erreur d'identification inter-classe).

- **Vrai Négatif (TN)** : Personne non enregistrée dans la base et correctement rejetée comme « personne inconnue » par le système.
- **Faux Négatif (FN)** : Étudiant enregistré dans la base que l'algorithme n'a pas pu reconnaître ou qu'il a marqué à tort comme « inconnu » (échec de reconnaissance intra-classe).

Formules des métriques : À partir de ces valeurs fondamentales, nous calculons les indicateurs suivants :

$$\text{Exactitude (Accuracy)} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

$$\text{Rappel (Recall)} = \frac{TP}{TP + FN} \times 100\%$$

$$\text{Précision (Precision)} = \frac{TP}{TP + FP} \times 100\%$$

$$\text{F1-Score (F1-Score)} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} \times 100\%$$

Remarque : Afin de comparer objectivement les différentes architectures évaluées, une démarche d'évaluation hiérarchisée est adoptée, reposant sur des indicateurs complémentaires. Le F1-Score constitue le critère principal de classement, car il représente la moyenne harmonique de la précision et du rappel, offrant une mesure synthétique robuste face aux déséquilibres potentiels de distribution des classes [37]. En analyse secondaire, la précision et le rappel sont examinés conjointement afin de caractériser le compromis opérationnel propre à chaque modèle : une précision élevée limite les fausses identifications, tandis qu'un rappel élevé garantit la détection exhaustive des instances cibles [38]. L'exactitude (accuracy) est rapportée à titre indicatif, en précisant systématiquement la répartition des classes pour éviter toute interprétation biaisée, notamment lorsque les échantillons majoritaires dominent le jeu de données. Cette méthodologie de comparaison, conforme aux pratiques recommandées en évaluation de systèmes de classification, assure une interprétation fiable, reproductible et adaptée aux exigences spécifiques de la reconnaissance faciale.

2.4.2 Expérimentations

Dans le cadre de cette étude, nous avons implémenté des approches de détection et de reconnaissance faciale afin d'évaluer leurs performances dans un système de gestion automatisée des présences. Les méthodes de détection utilisées sont **MTCNN**, **HOG** et **Viola-Jones**, tandis que les méthodes de reconnaissance sont **FaceNet**, **Dlib** et **LBPH**.

Le choix de ces algorithmes repose sur leur large utilisation dans la littérature scientifique et dans les systèmes de reconnaissance faciale en temps réel, garantissant ainsi une évaluation pertinente et comparable aux travaux existants, tout en couvrant des approches classiques et modernes.

Afin de valider la fiabilité de notre pipeline, nous avons retenu comme référence principale les travaux de Lateef et Kamil [23], en réimplémentant leurs configurations expérimentales. Ce choix se justifie par le fait que cette étude propose l'une des rares comparaisons homogènes de plusieurs pipelines de reconnaissance faciale, évalués sur une même base de données et selon un protocole expérimental identique, ce qui est essentiel pour garantir une analyse objective des différentes combinaisons détecteur-reconnaisseur. De plus, cet article est publié dans The Iraqi Journal for Computer Science and Mathematics, une revue scientifique internationale indexée, classée Q2 selon le Scimago Journal Rank (SJR 2025 = 0.482) avec un H-index de 22, ce qui atteste de sa rigueur méthodologique et de sa reconnaissance académique.

Cependant, cette étude présente une limite importante liée à l'absence de code source ouvert, ce qui restreint la reproductibilité directe des résultats. Notre démarche vise précisément à surmonter cette contrainte en proposant une implémentation indépendante en Python, entièrement vérifiable. Cette réimplémentation ne se limite pas à la reproduction des résultats publiés, mais permet également d'évaluer la cohérence et la stabilité des performances obtenues, ainsi que la reproductibilité effective des méthodes dans notre contexte d'application.

2.4.2.1 Environnement expérimental

Les expérimentations ont été conduites sur la plateforme cloud Google Colab, offrant un accès à des ressources matérielles mutualisées (GPU Tesla T4, 12 Go RAM) sans configuration locale. L'environnement logiciel, basé sur Python 3, intègre les bibliothèques suivantes : OpenCV et Ultralytics YOLO pour la détection et le traitement d'images, FaceNet et Dlib pour l'extraction de caractéristiques et la reconnaissance, NumPy et Matplotlib pour les calculs numériques et la visualisation, ainsi que les modules OS et Time pour la gestion des fichiers et l'évaluation temporelle. Cette configuration garantit la reproductibilité des tests et l'efficacité du pipeline de reconnaissance dans un contexte temps réel.

2.4.2.2 Bases de données utilisées

Deux bases de données distinctes ont été utilisées dans le cadre de cette étude afin de garantir une évaluation cohérente et comparative des différentes approches de détection et de reconnaissance faciale.

- **Base de données 1 : article de Lateef et Kamil** La base de données utilisée, proposée par Lateef et Kamil [23], est composée de 31 étudiants. Chaque étudiant est représenté par un ensemble de 70 images faciales, ce qui correspond à un total de 2170 images. Les images présentent des variations en termes d'éclairage, de distance, de pose et d'expressions faciales, permettant d'évaluer la robustesse des différentes approches dans des conditions proches d'un environnement réel.

Le choix de cette base s'inscrit dans une démarche de reproduction expérimentale, visant à valider notre approche en obtenant des résultats comparables à ceux rapportés par Lateef et Kamil, tout en assurant une évaluation cohérente et fiable des performances du système proposé.

Note : nous avons utilisé la base de données disponible via le lien suivant : <https://doi.org/10.34740/kaggle/dsv/4925762>

- **Base de données 2 : base de données proposée** La seconde base de données a été constituée dans le cadre de ce travail. Elle est composée de 8 étudiants de niveau Licence 2 en informatique. Pour chaque étudiant, trois images faciales ont été collectées, correspondant à différentes orientations du visage (face, profil gauche et profil droit). Cette base permet d'introduire des variations simples de pose et de tester le bon fonctionnement du système dans un environnement académique contrôlé, avec des images individuelles et des images du groupe. En complément, une image globale prise dans un amphi a été utilisée pour évaluer le système dans des conditions plus réalistes.

Note : Dans le cadre des expérimentations, ces images, capturées dans l'Amphi 1, ont été utilisées comme échantillons de test. Toutefois, elles ne peuvent être reproduites dans ce mémoire afin de préserver la confidentialité des données personnelles des étudiants, conformément aux exigences éthiques relatives au respect de la vie privée et au consentement des personnes concernées.

2.4.2.3 Application des algorithmes sur les bases de données

Afin d'évaluer les performances des différentes approches de détection et de reconnaissance faciale, les expérimentations ont été réalisées sur deux bases de données distinctes.

Application des algorithmes sur la base de données 1 : Les différentes configurations de détection et de reconnaissance faciale ont été appliquées sur la base de données proposée par Lateef et Kamil [23]. Cette étape a pour objectif principal de reproduire les résultats de leurs travaux afin de vérifier la cohérence et la fiabilité de notre implémentation.

TABLEAU 2.2 – Résultats de classification obtenus sur la base de données 1

Technique	TP	TN	FP	FN	Préc.	Rappel	F1-score	Acc.	Temps (s)
MTCNN + FaceNet	31	1	0	0	100 %	100 %	100 %	100 %	1704,6
HOG + Dlib	29	1	0	2	100 %	94 %	97 %	94 %	741,5
Viola-Jones + LBPH	29	0	2	1	94 %	97 %	96 %	91 %	222,2

Après l'implémentation des différentes configurations dans notre environnement expérimental, nous avons obtenu des résultats identiques à ceux rapportés dans les travaux de Lateef et Kamil [23]. Cette concordance confirme la cohérence de notre implémentation ainsi que la stabilité des performances des algorithmes étudiés.

L'évaluation a été réalisée sur l'ensemble de la base de données constituée de 31 individus, chacun représenté par 70 images présentant des variations de pose, d'illumination et d'expression. Aucun découpage de type 80 %/20 % n'a été appliqué dans cette étude. Par ailleurs, une évaluation complémentaire en conditions réelles a été effectuée dans un environnement de classe afin de tester la robustesse du système dans un scénario d'utilisation réel.

Application des algorithmes sur la base de données 2 : Les différentes configurations de détection et de reconnaissance faciale ont été appliquées sur la base de données proposée de 8 étudiants. Cette étape a pour objectif d'évaluer les performances des algorithmes dans un environnement académique contrôlé tout en introduisant des variations simples de pose faciale. Les résultats obtenus après l'implémentation sont présentés dans le tableau suivant :

TABLEAU 2.3 – Résultats de classification obtenus sur la base de données 2

Technique	TP	TN	FP	FN	Préc.	Rappel	F1-score	Acc.	Temps (s)
MTCNN + FaceNet	8	0	0	0	100 %	100 %	100 %	100 %	87,6
HOG + Dlib	7	1	0	0	94 %	94 %	94 %	100 %	55,2
Viola-Jones + LBPH	6	0	2	0	91 %	97 %	94 %	75 %	8,4

Aucun découpage préalable des données de type 80%/20% en ensembles d'entraînement et de test n'a été appliqué dans cette étude. L'ensemble des images disponibles a été exploité pour l'évaluation des modèles de reconnaissance faciale, afin d'analyser leurs performances globales dans différentes conditions de prise de vue.

2.4.3 Contribution

Notre contribution consiste à remplacer les détecteurs faciaux conventionnels (MTCNN, HOG et Viola-Jones) par l'architecture YOLOv8, afin d'optimiser la vitesse d'inférence

et d'améliorer la robustesse de la localisation des visages dans des environnements complexes, notamment en forte densité. Ce choix s'appuie sur les performances reconnues de YOLOv8 dans la détection d'objets en traitement rapide, qui offre un compromis optimal entre précision et efficacité computationnelle. Cette approche prometteuse bénéficie d'une architecture neuronale avancée, de mécanismes d'extraction de caractéristiques optimisés et de récentes améliorations en apprentissage profond, assurant ainsi une détection multi-visages rapide et fiable, même dans des conditions dégradées (variations d'éclairage, occlusions partielles, changements de pose). Sa faible empreinte mémoire et son orientation vers des systèmes rapides en font une solution particulièrement adaptée aux systèmes automatisés de gestion des présences, visant à renforcer la scalabilité et la robustesse du pipeline initial.

Le pipeline proposé s'articule autour d'une chaîne de traitement modulaire : détection des visages via YOLOv8, extraction des régions d'intérêt, génération d'embeddings à l'aide de FaceNet, Dlib ou LBPH, puis comparaison avec les images de la base de données. L'évaluation expérimentale repose sur des métriques standard (matrice de confusion, exactitude, précision, rappel et score F1), permettant de quantifier objectivement les gains en performance et de valider l'adéquation de cette approche prometteuse aux exigences opérationnelles du système.

- Base de données 1 : article de Lateef et Kamil

Cette première évaluation a été conduite sur le corpus de référence de Lateef et Kamil [23], composé de 31 étudiants, Les résultats des trois configurations intégrant YOLOv8 comme détecteur facial sont présentés dans le tableau. 2.4.

TABLEAU 2.4 – Performances des configurations YOLOv8 sur la base de Lateef et Kamil (31 étudiantes)

Technique	TP	TN	FP	FN	Acc.	Préc.	Rappel	F1	Temps (s)
YOLO + FaceNet	31	1	0	0	100 %	100 %	100 %	100 %	465,22
YOLO + Dlib	29	3	0	0	100 %	100 %	100 %	100 %	244,03
YOLO + LBPH	30	2	0	0	100 %	100 %	100 %	100 %	202,52

Les résultats confirment une exactitude de 100 % pour l'ensemble des combinaisons, avec un temps d'inférence minimal pour YOLO+Dlib (244 s), validant l'efficacité du détecteur YOLOv8 en contexte temps réel.

- Base de données 2 : base de données proposée

Afin d'évaluer l'adaptabilité du système dans un contexte académique francophone, une seconde validation a été menée sur un corpus constitué localement auprès de 8 étu-

dants de Licence 2 en informatique. Pour chaque individu, trois acquisitions ont été réalisées (vue frontale, profil gauche, profil droit), soit un total de 24 images. Les performances des mêmes configurations hybrides sont rapportées dans le tableau 2.5.

TABLEAU 2.5 – Performances des configurations YOLOv8 sur la base locale (8 sujets, 24 images)

Technique	TP	TN	FP	FN	Acc.	Préc.	Rappel	F1	Temps (s)
YOLO + FaceNet	8	0	0	0	100%	100%	100%	100%	61.2
YOLO + Dlib	8	0	0	0	100%	100%	100%	100%	49.8
YOLO + LBPH	6	2	0	0	100%	100%	100%	100%	7.8

2.4.3.1 Analyse comparative des performances

- **Base de données 1 :** Afin de visualiser les performances des différentes configurations étudiées, une représentation graphique des principales métriques d'évaluation a été réalisée sur la base de données de Lateef et Kamil [23]. Les graphiques suivants présentent une comparaison entre les approches classiques (MTCNN, HOG et Viola-Jones, respectivement) et les configurations intégrant le détecteur YOLOv8 appliquées à cette même base de données.

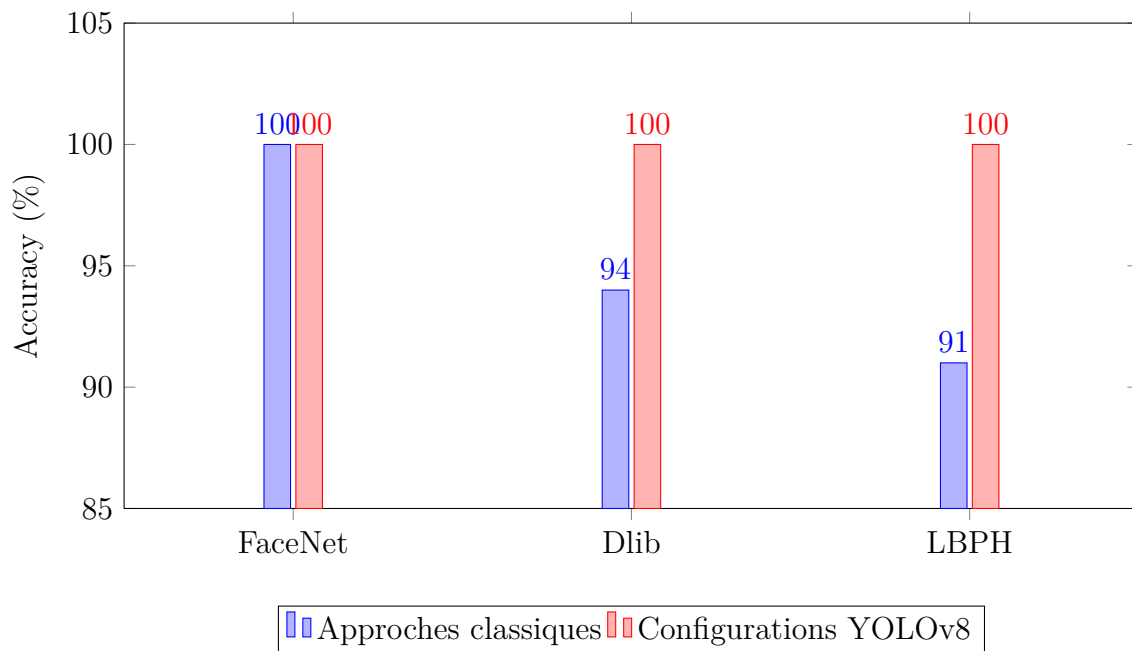


FIGURE 2.5 – Comparaison de l'Accuracy – Base de données 1

La figure 2.5 présente une comparaison de l'accuracy obtenue par les différentes méthodes de reconnaissance faciale sur la base de données 1. Les résultats montrent que l'approche classique basée sur MTCNN+FaceNet atteint une accuracy de 100 %, tandis

que HOG+Dlib et Viola-Jones+LBPH obtiennent respectivement 94 % et 91 %. En revanche, les configurations intégrant YOLOv8 pour la détection des visages atteignent une accuracy de 100 % pour les trois méthodes de reconnaissance (FaceNet, Dlib et LBPH).

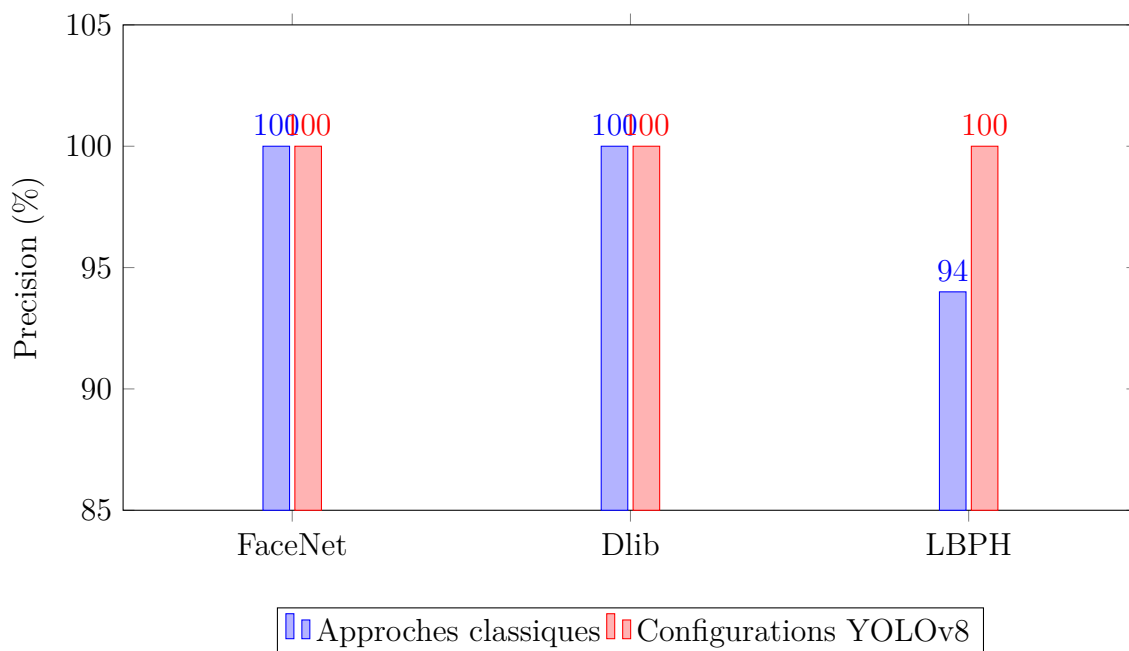


FIGURE 2.6 – Comparaison de la Precision – Base de données 1

La figure 2.6 présente une comparaison de la précision (Precision) des différentes méthodes de reconnaissance faciale sur la base de données 1. Les approches classiques montrent des performances élevées : MTCNN+FaceNet et HOG+Dlib atteignent une précision de 100 %, tandis que Viola-Jones+LBPH obtient 94 %. Après l'intégration de YOLOv8 pour la détection des visages, les trois méthodes atteignent une précision parfaite de 100 %.

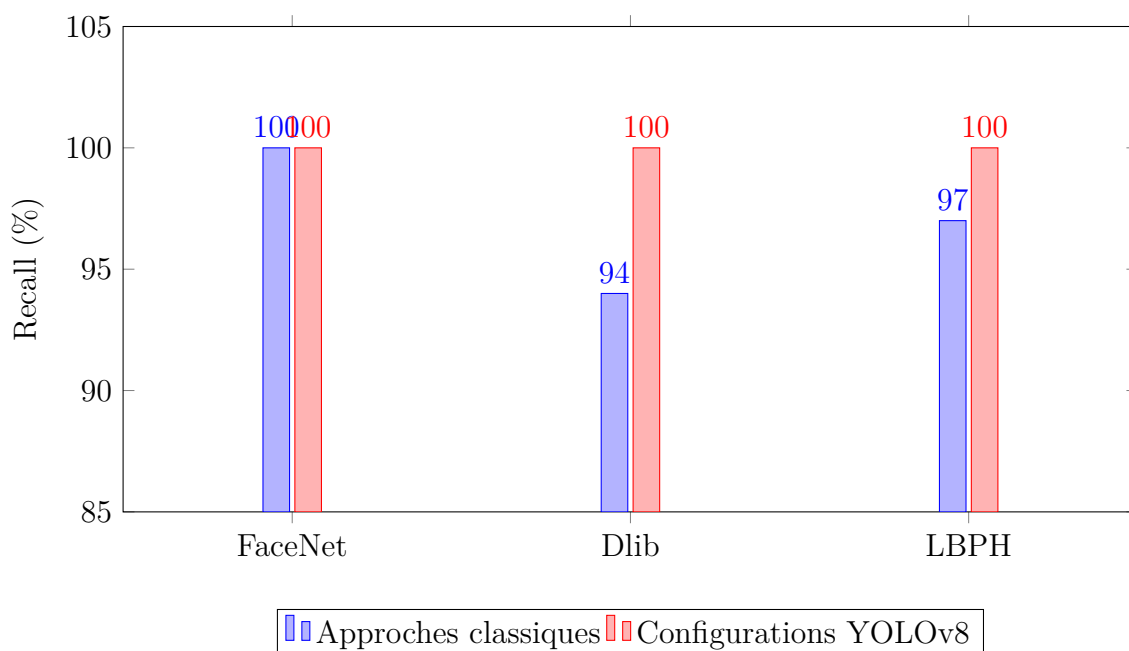


FIGURE 2.7 – Comparaison du Recall – Base de données 1

La figure 2.7 illustre la comparaison du taux de rappel (Recall) des différentes méthodes de reconnaissance faciale sur la base de données 1. Les résultats montrent que l'approche classique utilisant FaceNet atteint un rappel de 100 %, tandis que Dlib et LBPH obtiennent respectivement 94 % et 97 %. Après l'intégration de YOLOv8 pour la détection des visages, les trois méthodes atteignent un rappel de 100 %.

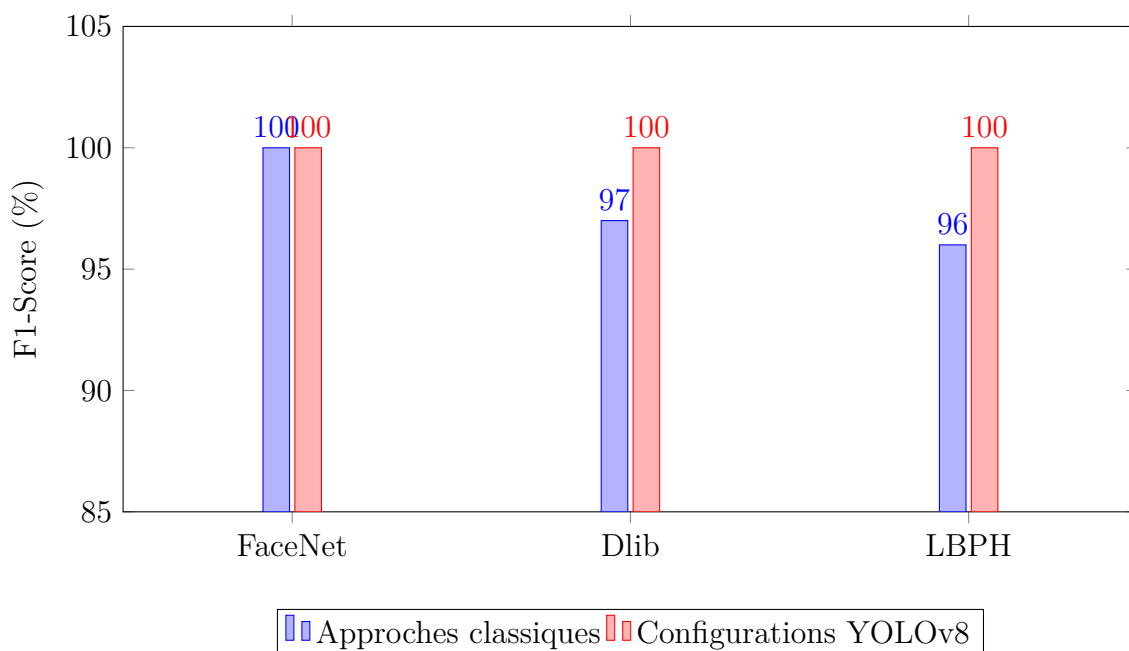


FIGURE 2.8 – Comparaison du F1-Score – Base de données 1

La figure 2.8 présente la comparaison du F1-Score des différentes méthodes de reconnaissance faciale sur la base de données 1. Les résultats obtenus avec les approches

classiques montrent que MTCNN+FaceNet atteint un F1-Score parfait de 100 %, tandis que HOG+Dlib et Viola-Jones+LBPH obtiennent respectivement 97 % et 96 %, ce qui indique un équilibre légèrement moins optimal entre précision et rappel pour ces deux méthodes. En revanche, après l'intégration de YOLOv8 pour la détection des visages, toutes les méthodes atteignent un F1-Score de 100 %, traduisant une amélioration globale des performances.

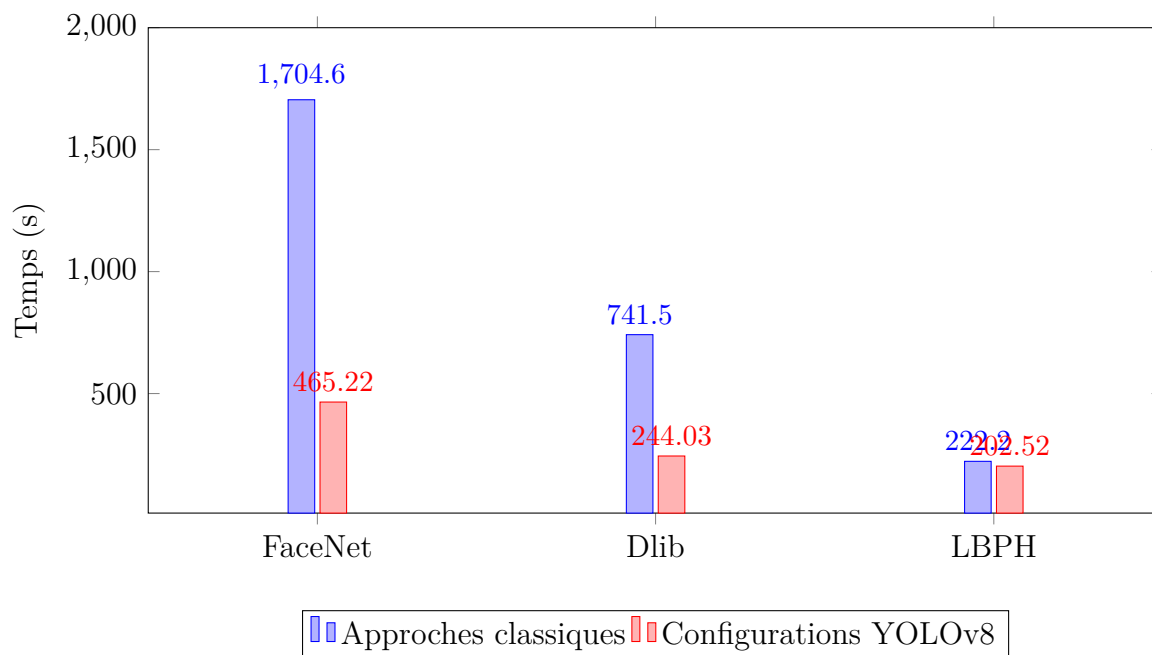


FIGURE 2.9 – Comparaison du temps de traitement (s) – Base de données 1

La figure 2.9 illustre la comparaison du temps de traitement des différentes méthodes de reconnaissance faciale sur la base de données 1. Les résultats montrent que, dans les approches classiques, FaceNet est la méthode la plus coûteuse en temps avec 1704.6 s, suivi de Dlib avec 741.5 s, tandis que LBPH est la plus rapide avec 222.2 s. Après l'intégration de YOLOv8, une réduction significative du temps de traitement est observée pour toutes les méthodes, avec 465.22 s pour FaceNet, 244.03 s pour Dlib et 202.52 s pour LBPH. Cette amélioration s'explique par une meilleure détection et localisation des visages en amont, ce qui réduit le coût global du traitement.

- Base de données 2 : Pour évaluer les performances des différentes configurations étudiées, des représentations graphiques des principales métriques ont été réalisées sur la base de données proposée dans ce travail. Ces résultats permettent de comparer les approches classiques de détection et de reconnaissance faciale avec les configurations basées sur YOLOv8.

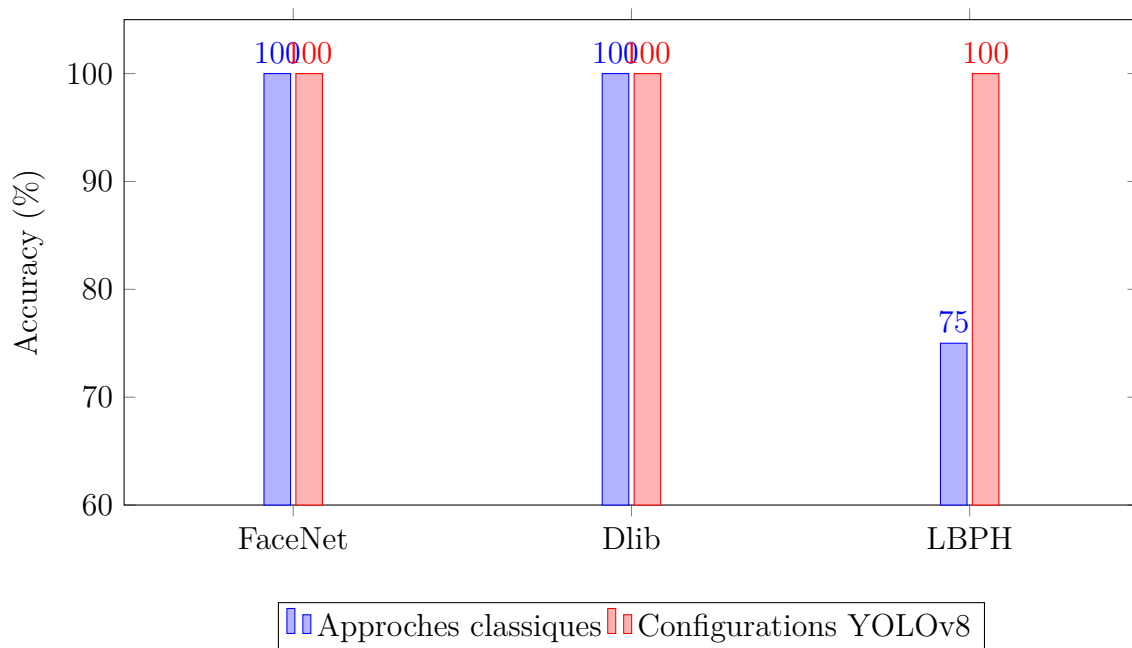


FIGURE 2.10 – Comparaison de l'Accuracy – Base de données 2

La figure 2.10 présente la comparaison de l'accuracy des différentes méthodes de reconnaissance faciale sur la base de données 2. Les résultats des approches classiques montrent que FaceNet et Dlib atteignent une précision parfaite de 100 %, tandis que LBPH présente une performance plus faible avec 75 %, ce qui reflète sa sensibilité aux variations des conditions de détection. Après l'intégration de YOLOv8, une amélioration est observée uniquement pour LBPH, qui atteint désormais 100 %, alors que FaceNet et Dlib restent stables à 100 %.

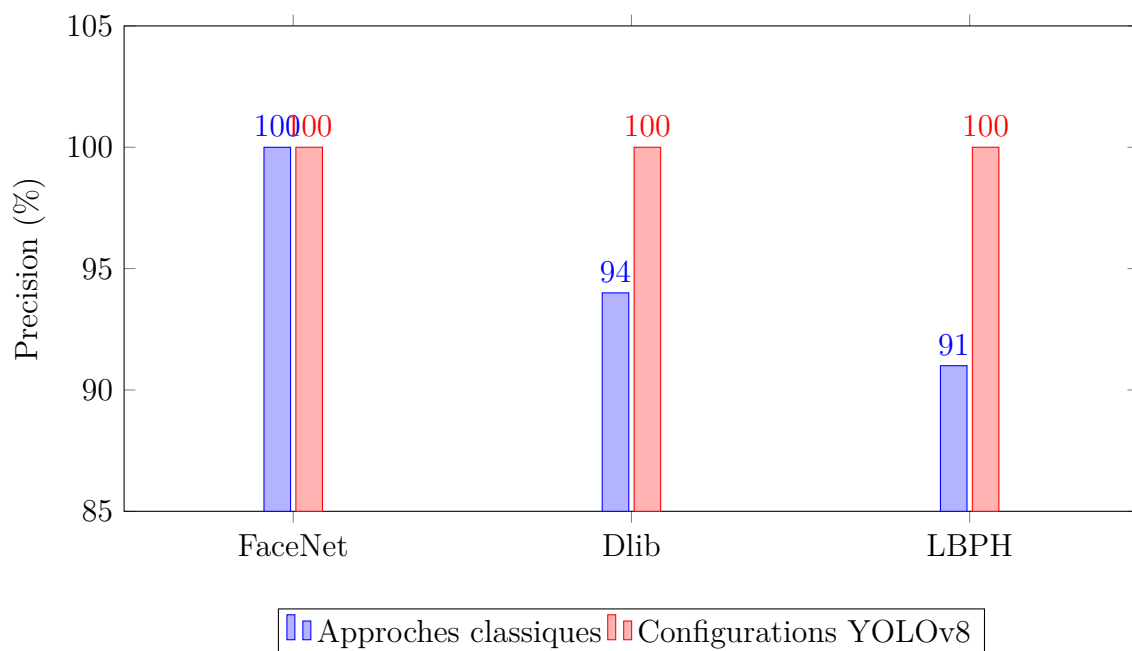


FIGURE 2.11 – Comparaison de la Precision – Base de données 2

La figure 2.11 présente la comparaison de la précision des différentes méthodes de reconnaissance faciale sur la base de données 2. Les approches classiques montrent des performances variables : FaceNet atteint une précision de 100 %, tandis que Dlib et LBPH obtiennent respectivement 94 % et 91 %, ce qui indique une diminution de la qualité de classification pour ces deux dernières méthodes. Après l'intégration de YOLOv8, une amélioration nette est observée puisque toutes les méthodes atteignent une précision de 100 %.

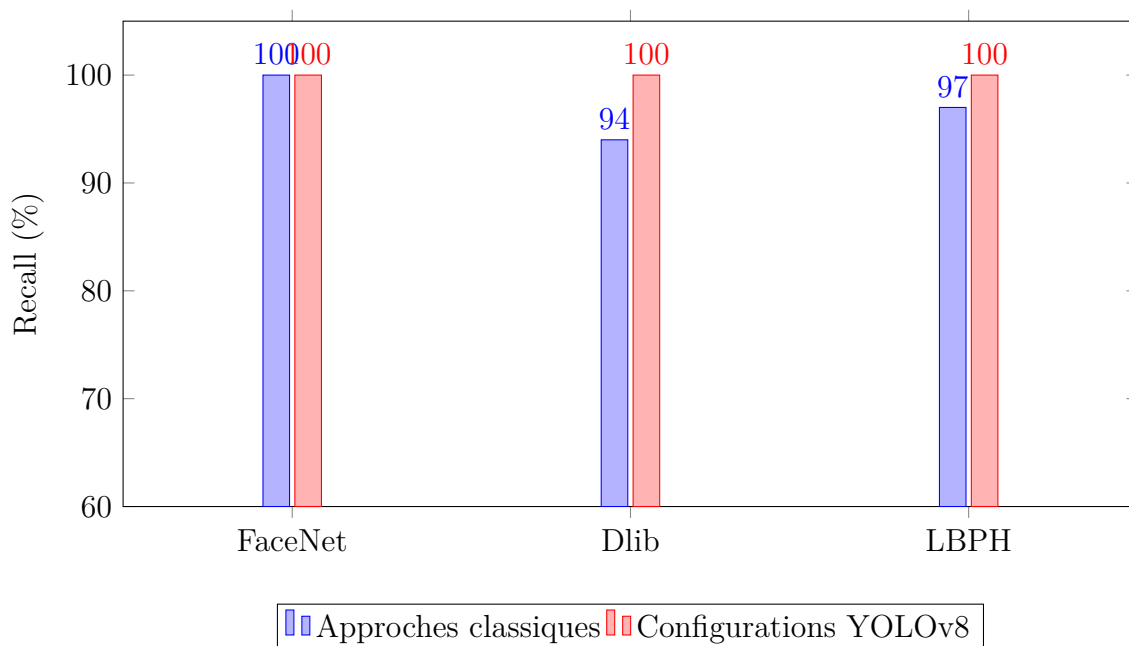


FIGURE 2.12 – Comparaison du Recall – Base de données 2

La figure 2.12 présente la comparaison du rappel (Recall) des différentes méthodes de reconnaissance faciale sur la base de données 2. Les résultats des approches classiques montrent que FaceNet atteint un rappel parfait de 100 %, tandis que Dlib et LBPH obtiennent respectivement 94 % et 97 %, ce qui indique une capacité légèrement variable à détecter correctement tous les visages présents. Après l'intégration de YOLOv8, toutes les méthodes atteignent un rappel de 100 %, traduisant une amélioration notable.

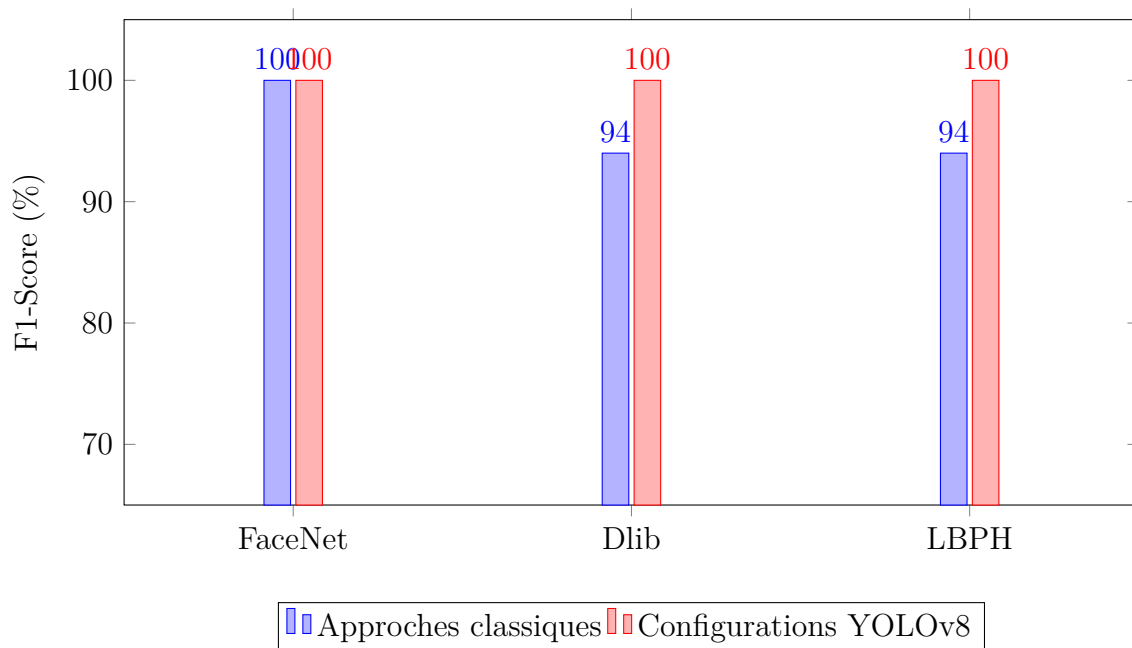


FIGURE 2.13 – Comparaison du F1-Score – Base de données 2

La figure 2.13 présente la comparaison du F1-Score des différentes méthodes de reconnaissance faciale sur la base de données 2. Les résultats des approches classiques montrent que FaceNet et Dlib atteignent un F1-Score parfait de 100 %, tandis que LBPH obtient une valeur plus faible de 94 %, indiquant un équilibre moins optimal entre précision et rappel pour cette méthode. Après l'intégration de YOLOv8, toutes les méthodes atteignent un F1-Score de 100 %, ce qui traduit une amélioration globale des performances.

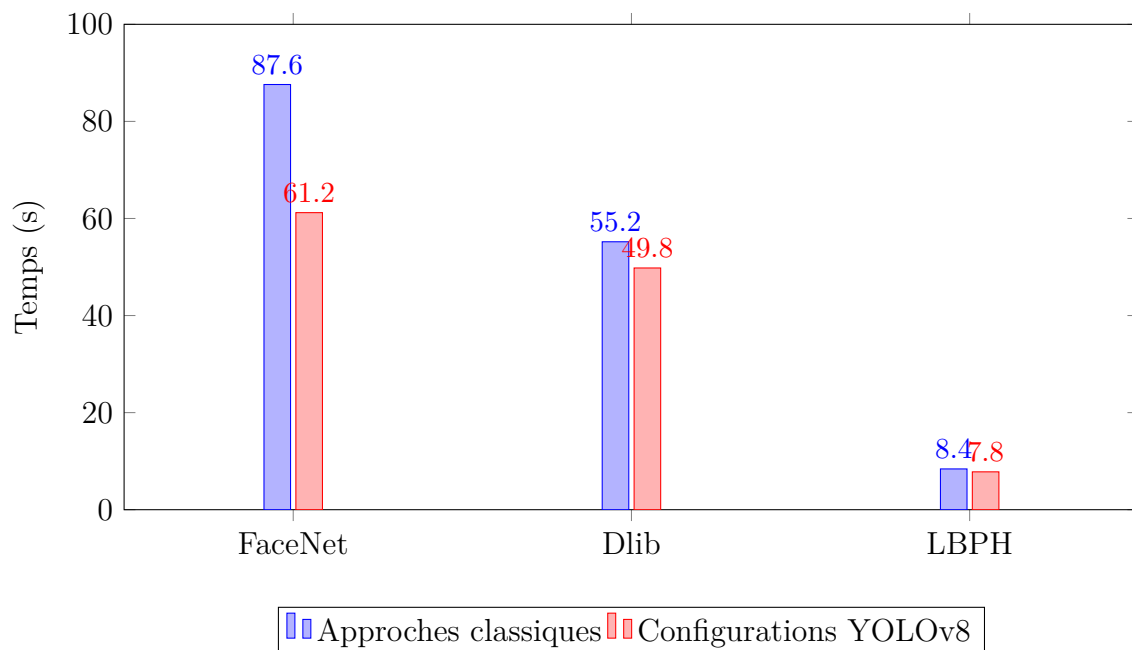


FIGURE 2.14 – Comparaison du temps de traitement (s) sur la Base de données 2

La figure 2.14 présente la comparaison du temps de traitement des différentes méthodes

de reconnaissance faciale sur la base de données 2. Les résultats montrent que, dans les approches classiques, FaceNet est la méthode la plus lente avec 87.6 s, suivi de Dlib avec 55.2 s, tandis que LBPH est la plus rapide avec 8.4 s. Après l'intégration de YOLOv8, une réduction globale du temps de traitement est observée pour toutes les méthodes, avec 61.2 s pour FaceNet, 49.8 s pour Dlib et 7.8 s pour LBPH.

2.4.4 Analyse et discussion des résultats

L'analyse comparative des résultats obtenus sur les deux bases de données permet de dégager plusieurs observations relatives aux performances des différentes configurations de détection et de reconnaissance faciale étudiées, tant sur le plan de la précision que de l'efficacité computationnelle.

2.4.4.1 Analyse des résultats sur la base de données 1

Les expérimentations menées sur la base de données de Lateef et Kamil [23], composée de 31 étudiants, confirment dans l'ensemble les conclusions de l'étude de référence et permettent d'établir une comparaison rigoureuse entre les approches classiques et les configurations intégrant YOLOv8.

Approches classiques : Parmi les approches classiques, la combinaison **MTCNN + FaceNet** se distingue par ses performances optimales, atteignant 100 % sur l'ensemble des métriques évaluées (précision, rappel, F1-score et exactitude). Ces résultats s'expliquent par la robustesse du détecteur MTCNN face aux variations d'éclairage et de pose, couplée à la puissance discriminante des embeddings profonds générés par FaceNet. Toutefois, cette combinaison présente un temps de traitement prohibitif de 1 704,6 s, inhérent à la complexité computationnelle des architectures de Deep Learning mobilisées, ce qui la rend peu adaptée aux contraintes d'un système rapide sans infrastructure matérielle dédiée (GPU).

La combinaison **HOG + Dlib** offre un compromis intéressant entre précision et efficacité : elle affiche une exactitude de 94 %, une précision de 100 %, un rappel de 94 % et un F1-score de 97 %, pour un temps de traitement de 741,5 s. Les deux faux négatifs observés (FN = 2) traduisent une sensibilité résiduelle aux variations de pose, caractéristique connue du descripteur HOG.

La combinaison **Viola-Jones + LBPH** présente les performances les plus modestes parmi les approches classiques, avec une exactitude de 91 %, une précision de 94 %, un rappel de 97 % et un F1-score de 96 %. Néanmoins, son temps d'exécution de 222,2 s constitue son principal atout, la rendant viable pour des systèmes embarqués ou à ressources matérielles limitées.

Configurations YOLOv8 : Les trois configurations intégrant YOLOv8 (**YOLO + FaceNet**, **YOLO + Dlib** et **YOLO + LBPH**) atteignent toutes une exactitude de 100 % sur cette base de données, avec des temps d'inférence significativement réduits par rapport à leurs équivalents classiques. Cette amélioration est directement imputable à la capacité de YOLOv8 à localiser les visages en une seule passe de réseau (single-stage detection), contrairement aux méthodes classiques qui reposent sur des approches en cascade ou par fenêtre glissante. En particulier, **YOLO + LBPH** enregistre le temps de traitement le plus court (202,52 s), suivi de **YOLO + Dlib** (244,03 s), tout en maintenant des performances parfaites pour les deux configurations, confirmant la synergie entre la rapidité du détecteur YOLOv8 et l'efficacité des modules de reconnaissance associés dans un système temps réel effectif.

2.4.4.2 Analyse des résultats sur la base de données 2

Sur la base de données locale composée de 8 étudiants, les résultats révèlent des tendances similaires, avec toutefois des nuances importantes liées à la nature de cette base, qui introduit des variations explicites de pose (vue frontale, profil gauche, profil droit).

Approches classiques : La combinaison **MTCNN + FaceNet** atteint une exactitude parfaite de 100 %, avec un temps de traitement de 87,6 s. La combinaison **HOG + Dlib** obtient également une exactitude de 100 %, bien qu'avec une précision, un rappel et un F1-score légèrement inférieurs (94 % chacun), pour un temps de traitement réduit de 55,2 s. Ces résultats confirment la robustesse de ces deux approches face aux variations de pose sur un corpus de taille réduite.

En revanche, la combinaison **Viola-Jones + LBPH** n'obtient qu'une exactitude de 75 %, en raison de la présence de deux faux positifs ($FP = 2$), malgré une précision de 91 % et un rappel de 97 %. Ce résultat met en évidence la sensibilité accrue du détecteur de Viola-Jones aux conditions de prise de vue non frontales, une limitation structurelle de ce classifieur entraîné principalement sur des visages de face. Cependant, son temps de traitement (8,4 s) demeure le plus faible de toutes les approches classiques évaluées, soulignant le compromis inhérent entre efficacité computationnelle et robustesse de détection.

Configurations YOLOv8 : Les trois configurations **YOLO + FaceNet**, **YOLO + Dlib** et **YOLO + LBPH** atteignent toutes une exactitude parfaite de 100 %, avec des temps d'exécution respectifs de 61,2 s, 49,8 s et 7,8 s. Ces résultats confirment la supériorité de YOLOv8 en tant que module de détection dans des conditions de variation de pose, grâce à son entraînement sur des jeux de données multi-vues et ses mécanismes d'attention intégrés. Il est à noter que, contrairement à son comportement observé avec les détecteurs classiques, le module **LBPH** couplé à YOLOv8 atteint ici des performances parfaites tout

en conservant le temps de traitement le plus faible de l'ensemble des configurations testées (7,8s), ce qui illustre l'impact déterminant de la qualité de la détection en amont sur les performances globales du pipeline de reconnaissance.

2.4.5 Synthèse comparative des approches

Afin de synthétiser les caractéristiques intrinsèques des différentes méthodes étudiées, le tableau 2.6 présente une comparaison qualitative selon plusieurs critères : le type d'approche, les avantages et limites de chaque méthode, ainsi que son adéquation au déploiement en temps réel.

TABLEAU 2.6 – Synthèse comparative des approches de détection et de reconnaissance faciale

Méthode	Type	Avantages	Limites	Rapidité	Robust	Temps réel
Viola-Jones + LBPH	Classique	Très rapide, faible coût computationnel, déploiement léger	Sensible aux variations d'éclairage et de pose; précision limitée	Très élevée	Faible	Oui
HOG + Dlib	Hybride	Bonne précision, descripteurs robustes aux transformations géométriques modérées	Sensible aux fortes variations de pose; temps de traitement moyen	Moyenne	Moyenne	Oui
MTCNN + FaceNet	Deep Learning	Très haute précision; robuste aux variations d'éclairage, de pose et d'expression	Temps d'inférence élevé; nécessite des ressources GPU importantes	Moyenne	Très élevée	Moyen
YOLOv8 + FaceNet	Deep Learning	Détection rapide et robuste en une seule passe; précision maximale sur les deux bases	Requiert un GPU pour des performances optimales	Très élevée	Très élevée	Oui
YOLOv8 + Dlib	Hybride	Meilleur compromis vitesse-précision; temps d'inférence minimal parmi les configurations Deep Learning	Requiert un GPU; performances du module Dlib sensibles aux profils extrêmes	Très élevée	Élevée	Oui
YOLOv8 + LBPH	Hybride	Très faible coût computationnel; temps de traitement le plus bas de toutes les configurations	Reconnaissance LBPH sensible aux variations de pose; faux négatifs sur vues de profil	Très élevée	Moyenne	Oui

Bien que FaceNet présente généralement de meilleures performances en termes de précision de reconnaissance, le choix de Dlib (ResNet) a été retenu en raison de son meilleur compromis entre précision, simplicité d'intégration et efficacité computationnelle. De plus, Dlib offre une implémentation légère et facilement exploitable en temps réel, ce qui le rend plus adapté aux contraintes d'un système embarqué de gestion des présences. Par ailleurs, ce choix est également justifié par les ressources matérielles disponibles, notamment 8 Go de RAM, qui rendent FaceNet moins adapté aux contraintes de traitement en temps réel.

2.4.6 Interprétation globale

À l'issue de cette analyse, plusieurs conclusions peuvent être formulées :

1. **Supériorité globale de YOLOv8** : L'intégration de YOLOv8 comme module de détection améliore systématiquement les performances de précision tout en réduisant les temps d'inférence, quels que soient le module de reconnaissance associé et la base de données utilisée. Cette observation valide l'intérêt de l'approche proposée comme extension du système.
2. **Rôle déterminant du module de reconnaissance** : Les limitations observées avec la configuration YOLO + LBPH sur la base de données 2 démontrent que la performance globale du pipeline est contrainte par le module de reconnaissance. Le choix d'un extracteur de caractéristiques adapté aux conditions d'acquisition (notamment les variations de pose) s'avère donc critique.
3. **Compromis précision-efficacité** : Les approches classiques (Viola-Jones + LBPH) conservent un intérêt opérationnel dans des environnements à ressources matérielles restreintes, en dépit de leurs performances inférieures, grâce à leur rapidité d'exécution remarquable.
4. **Recommandation pour un déploiement réel** : YOLO + Dlib offre un excellent compromis entre fiabilité de reconnaissance et performances temporelles compatibles avec une utilisation rapide dans des environnements réels tels que les amphithéâtres.

2.5 Conclusion

Ce chapitre a présenté une revue systématique des principales méthodes de détection et de reconnaissance faciale, évaluées selon des critères de précision, de robustesse environnementale et d'efficacité computationnelle. L'analyse théorique et expérimentale menée a permis de caractériser les forces et les limites de chaque approche dans le contexte spécifique de la gestion automatisée des présences.

La reproduction des travaux de Lateef et Kamil [23], enrichie par nos propres expérimentations, a confirmé la pertinence des architectures hybrides couplant détection et extraction de caractéristiques. Les résultats obtenus démontrent la capacité des méthodes modernes à maintenir des performances élevées dans des conditions réelles complexes, marquées par des variations significatives de pose, d'éclairage et de distance.

L'intégration du détecteur **YOLOv8** a par ailleurs validé l'efficacité des approches de détection en traitement rapide pour optimiser la latence globale du pipeline. Parmi les configurations évaluées, la combinaison **YOLOv8 + Dlib** s'est distinguée comme la plus adaptée aux contraintes d'un système automatisé d'émargement, grâce à son compromis optimal entre fiabilité de l'identification, rapidité d'exécution et viabilité technique de déploiement.

Ces conclusions étayent les choix méthodologiques retenus dans ce travail et fournissent un fondement solide pour la phase de conception. Le chapitre suivant exposera l'architecture globale de la solution proposée, les décisions techniques d'implémentation ainsi que le protocole de validation du système développé.

Chapitre 3

Conception et implémentation d'une
application de reconnaissance faciale dans le
cadre de la gestion des absences des
étudiantes lors des séances universitaires

3.1 Introduction

Ce chapitre présente les différentes étapes de conception, de développement et de réalisation du système intelligent de gestion automatisée des présences ClassTrack DZ, basé sur les techniques de reconnaissance faciale et d'intelligence artificielle. L'objectif principal est de concevoir une solution capable d'identifier automatiquement les étudiants et d'enregistrer leur présence de manière fiable, rapide et sécurisée dans un environnement universitaire.

Dans ce cadre, une attention particulière a été accordée aux choix méthodologiques et technologiques adoptés afin de garantir de bonnes performances en matière de détection et de reconnaissance faciale en temps réel. Le système proposé repose principalement sur l'utilisation de l'algorithme YOLOv8 pour la détection des visages et de la bibliothèque Dlib pour la reconnaissance faciale.

Ce chapitre est structuré en plusieurs parties principales :

- Dans un premier temps, nous présentons l'analyse des besoins du système, incluant le contexte du projet, les objectifs visés, les besoins fonctionnels et non fonctionnels ainsi que l'identification des différents acteurs intervenant dans le système.
- Ensuite, nous abordons la phase de conception à travers la présentation de l'architecture générale du système, des diagrammes UML ainsi que de la modélisation de la base de données.
- Par la suite, nous décrivons les étapes d'implémentation et de développement de l'application, notamment l'environnement matériel et logiciel utilisé, les technologies adoptées ainsi que les différents modules réalisés.
- Enfin, nous présentons les expérimentations effectuées et les résultats obtenus afin d'évaluer les performances du système en termes de précision, de rapidité et de fiabilité.

Ce chapitre permet ainsi de mettre en évidence la démarche suivie pour la réalisation du système proposé, depuis l'analyse des besoins jusqu'à la validation expérimentale de la solution développée.

3.2 Analyse et définition des besoins

3.2.1 Contexte et problématique

Dans les dernières années, l'intelligence artificielle a connu un développement remarquable et s'est intégrée dans plusieurs domaines tels que la santé, la sécurité, l'industrie et l'éducation. Dans ce contexte, nous avons voulu exploiter cette technologie dans le milieu universitaire, plus précisément dans la gestion des présences des étudiants, afin de moderniser et faciliter le processus de suivi des absences.

En effet, les méthodes traditionnelles utilisées pour enregistrer la présence présentent plusieurs problèmes : l'enseignant perd souvent un temps important en effectuant l'appel oral ou en faisant circuler une feuille de présence, et cette méthode favorise parfois la fraude. De plus, les feuilles de présence peuvent être perdues ou endommagées, et la gestion manuelle rend difficile l'organisation et l'archivage des données.

Face à ces problèmes, notre projet propose une solution intelligente basée sur l'intelligence artificielle permettant d'automatiser le processus de gestion des absences, d'améliorer la fiabilité des données, de réduire le temps de traitement et de limiter les cas de fraude.

3.2.2 Objectifs du système

- Automatiser le processus de gestion des absences des étudiants.
- Réduire le temps perdu lors de l'enregistrement manuel des présences.
- Limiter les cas de fraude et les erreurs dans le suivi des absences.
- Faciliter le travail des enseignants et de l'administration.
- Assurer un stockage sécurisé et organisé des données.
- Permettre une consultation rapide des informations et des historiques.
- Générer automatiquement des statistiques et des rapports.

3.2.3 Besoins fonctionnels

- Authentification sécurisée des utilisateurs (administrateur et enseignant).
- Gestion des comptes utilisateurs.
- Enregistrement automatique des présences des étudiants.
- Consultation de la liste des présences et des absences.
- Recherche des étudiants par nom, matricule ou groupe.
- Modification et mise à jour des informations des étudiants.
- Génération automatique des rapports et statistiques.

- Archivage des données de présence dans une base de données.
- Notification des absences importantes.
- Consultation de l'historique des absences.
- Gestion des classes, groupes et modules.
- Exportation des données et rapports.

3.2.4 Besoins non fonctionnels

- Le système doit être simple et facile à utiliser.
- L'interface doit être claire, moderne et ergonomique.
- Le système doit garantir la sécurité et la confidentialité des données.
- Les informations doivent être stockées de manière fiable et sécurisée.
- Le temps de réponse du système doit être rapide.
- Le système doit être disponible et accessible à tout moment.
- L'application doit être compatible avec différents appareils et systèmes.
- Le système doit permettre une maintenance et des mises à jour faciles.
- Le système doit assurer une bonne performance même avec un grand nombre d'utilisateurs.
- L'application doit être évolutive afin d'ajouter de nouvelles fonctionnalités dans le futur.

3.3 Conception

3.3.1 Étude des acteurs

Dans notre application de gestion des absences, on distingue principalement deux acteurs essentiels : **l'administrateur** et **l'enseignant**.

- **L'administrateur** est responsable de la gestion globale du système, notamment la création et la gestion des comptes utilisateurs, l'ajout des étudiants, la supervision des données et la configuration générale de la plateforme.
- **L'enseignant** est l'acteur principal dans le processus de gestion des présences, car il est chargé d'enregistrer les absences, de consulter les listes des étudiants et de suivre leur assiduité.

Cette organisation des rôles permet de garantir une meilleure répartition des tâches, une utilisation efficace du système et une gestion plus structurée des données.

3.3.2 Diagrammes UML

3.3.2.1 Diagramme d'activités

La Figure 3.1 présente le fonctionnement global du système de gestion des absences basé sur la reconnaissance faciale. Il met en évidence les deux principaux acteurs du système, à savoir l'administrateur et l'enseignant, ainsi que les différentes étapes de leurs interactions.

Après une phase d'authentification, l'administrateur accède à un tableau de bord lui permettant de gérer les étudiants, les comptes utilisateurs et de superviser les données du système. De son côté, l'enseignant initialise une séance, configure les paramètres du cours puis capture les images de la salle de classe. Ces images sont ensuite traitées par un module de détection et de reconnaissance faciale afin d'identifier les étudiants présents.

Enfin, les résultats obtenus sont enregistrés dans la base de données et des rapports peuvent être générés automatiquement, ce qui permet d'assurer une gestion des absences à la fois automatisée, fiable et efficace.

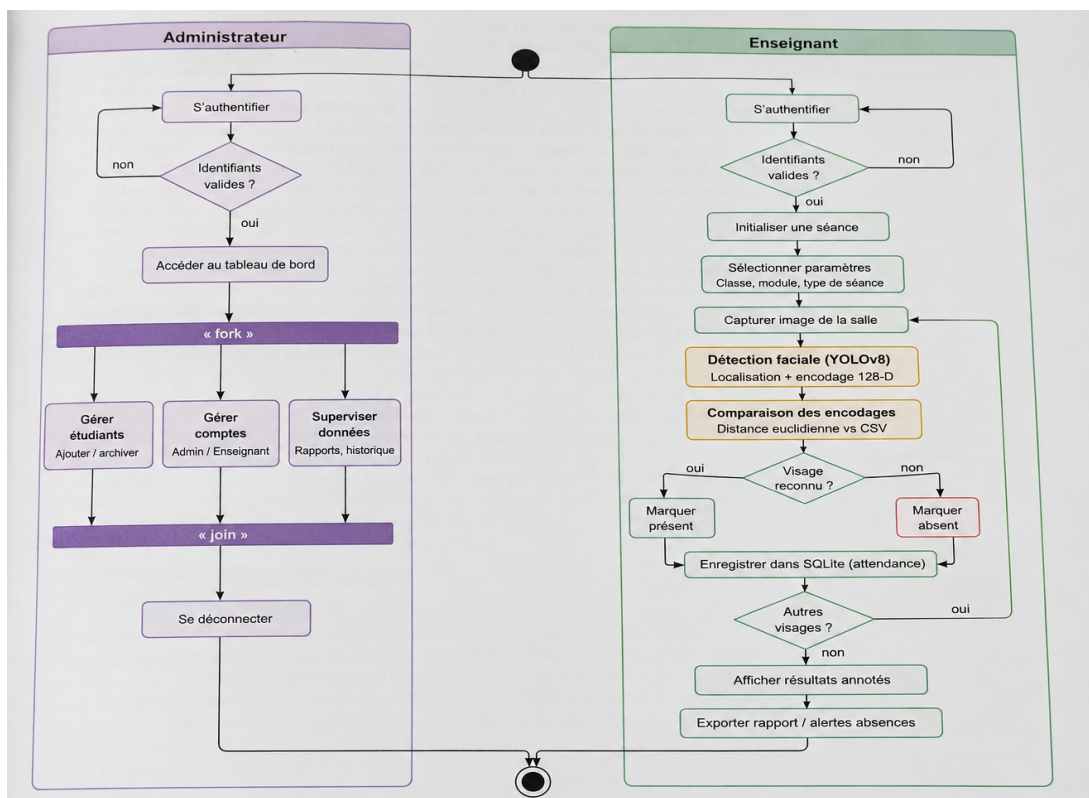


FIGURE 3.1 – Diagramme d'activités du système

3.3.2.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation présenté dans la figure 3.2 illustre les différentes interactions entre les acteurs du système et les fonctionnalités offertes par la plateforme de gestion intelligente des absences basée sur la reconnaissance faciale.

- L'administrateur assure la gestion globale de la plateforme, notamment l'administration des comptes utilisateurs et la gestion des étudiants.
- L'enseignant intervient dans le suivi pédagogique en lançant les séances et en consultant les absences.

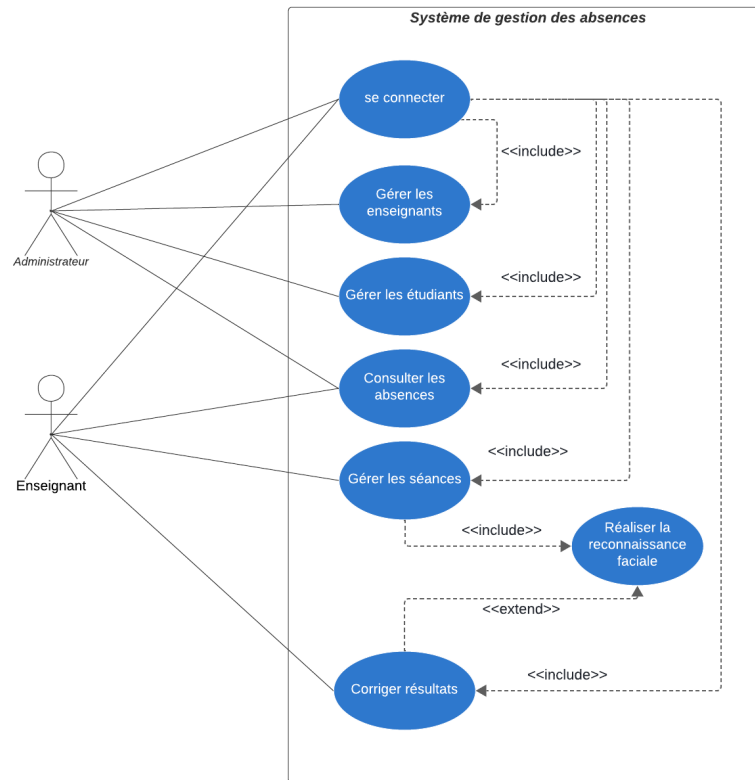


FIGURE 3.2 – Diagramme de cas d'utilisation du système

3.3.2.3 Diagramme de classes

Le diagramme de classes (figure 3.3) décrit la structure statique du système. Il met en évidence les principales entités du domaine, leurs attributs ainsi que les relations logiques qui assurent le bon fonctionnement du système.

Le système repose sur une hiérarchie d'utilisateurs fondée sur le principe d'héritage. La classe Utilisateur représente la classe mère, regroupant les attributs communs (nom, e-mail, mot de passe) ainsi que les fonctionnalités de base comme l'authentification. Deux classes dérivées en héritent : *Administrateur* et *Enseignant*.

Au cur du système, la classe Étudiant contient les informations académiques ainsi qu'un attribut photo utilisé pour la reconnaissance faciale. La classe Séance modélise un cours planifié à une date donnée. La classe Présence joue un rôle central en établissant une association entre un étudiant et une séance, en stockant le statut (présent, absent, retard) ainsi qu'un scoreReconnaissance représentant le niveau de confiance généré par l'algorithme d'IA.

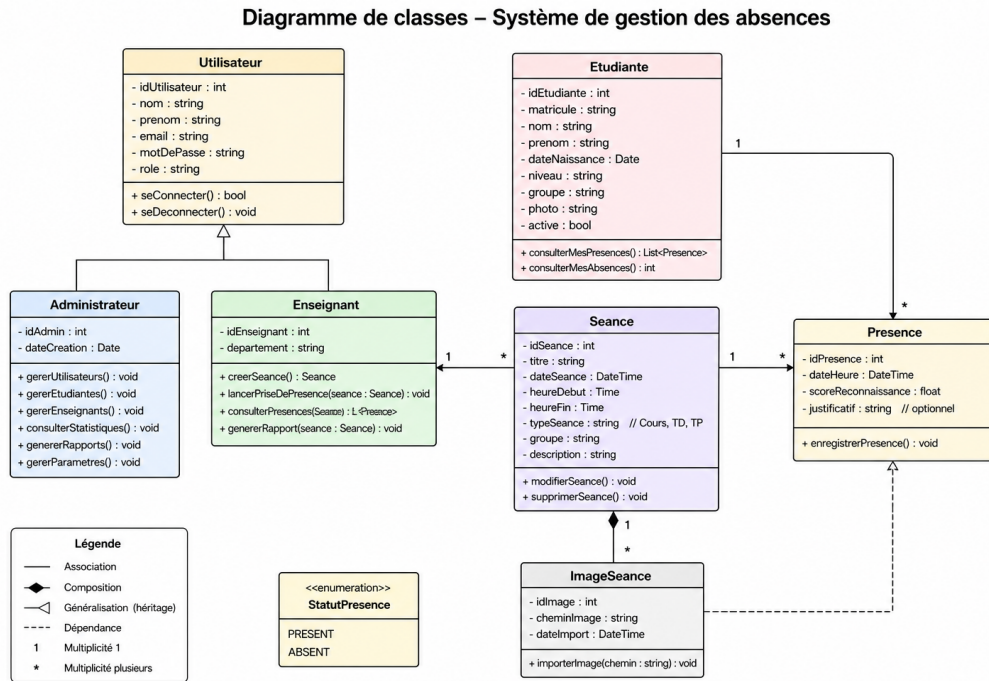


FIGURE 3.3 – Diagramme de classes du système

3.3.2.4 Diagramme de séquence - Pointage facial

Le diagramme de séquence (figure 3.4) illustre le déroulement dynamique du processus de pointage automatisé basé sur la reconnaissance faciale.

1. Initialisation et préparation des données : Le processus débute par l'action de l'enseignant qui initialise une séance via l'interface. Cette dernière enregistre immédiatement l'événement dans SQLite et charge les descripteurs faciaux de référence depuis le fichier CSV.

2. Traitement par le module de vision par ordinateur : Une fois l'image capturée, le module d'IA détecte les visages à l'aide de YOLOv8, extrait les vecteurs d'encodage pour chaque visage, puis les compare aux données de référence pour identifier les étudiants.

3. Enregistrement et restitution des résultats : La liste des étudiants reconnus est enregistrée dans la table *attendance*, et le système fournit à l'enseignant une image annotée ainsi qu'une liste récapitulative des présences.

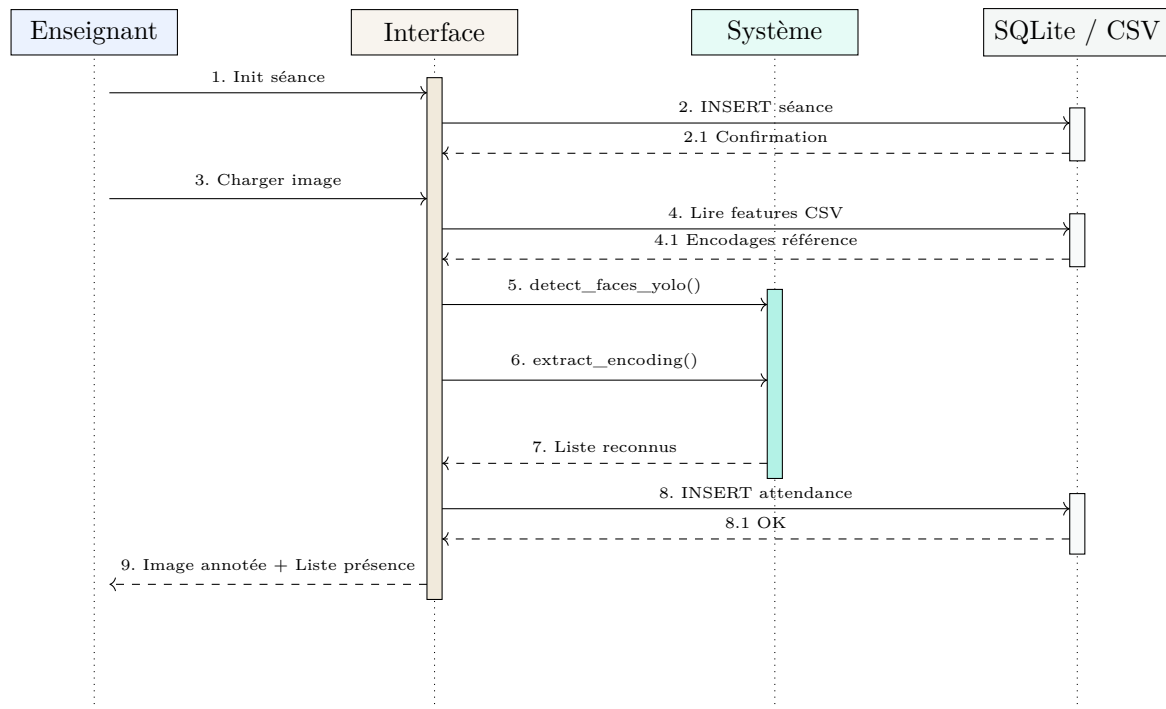


FIGURE 3.4 – Diagramme de séquence du processus de pointage facial

3.3.3 Modélisation de la base de données

Dans le cadre de ce projet, nous avons utilisé le système de gestion de base de données léger **SQLite**. Ce choix s'explique par sa simplicité d'intégration, sa rapidité ainsi que son absence de configuration serveur. SQLite permet de stocker localement les données de l'application tout en garantissant leur persistance et leur sécurité. La structure de la base de données est présentée dans le tableau ci-dessous.

TABLEAU 3.1 – Structure des tables de la base de données

Table	Colonnes / Description
students	id, name, student_id, email, class, group_name, photo, active
attendance	id, student_id, class, date, module, session_type, groupe
grades	id, student_id, class, subject, grade, semester Contrainte : UNIQUE(student_id, subject, semester)
users	id, username, password (SHA-256), full_name, email, role, photo, active

3.4 Implémentation et développement

3.4.1 Environnement de développement

Visual Studio Code (VS Code) est un éditeur de code source gratuit et open source développé par **Microsoft**. Il se caractérise par sa légèreté et sa compatibilité avec plusieurs systèmes d'exploitation, notamment Windows, macOS et Linux. Grâce à son riche écosystème d'extensions, il facilite le développement logiciel en offrant un support avancé pour plusieurs langages, en particulier Python, ainsi que des outils intégrés pour le débogage, la gestion de versions via Git et l'utilisation d'un terminal intégré. L'interface de cet environnement de développement est présentée dans la Figure 3.5.



FIGURE 3.5 – Interface de l'éditeur Visual Studio Code

3.4.2 Langage de programmation

Python s'est imposé comme un langage de programmation incontournable dans le développement des applications modernes, notamment en intelligence artificielle. Grâce à sa syntaxe simple, sa portabilité et sa gestion automatique de la mémoire, il permet un développement rapide et efficace. De plus, son riche écosystème de bibliothèques spécialisées en traitement de données et en vision par ordinateur en fait un choix idéal pour la mise en œuvre de systèmes de reconnaissance faciale. Le logo de ce langage est représenté dans la Figure 3.6.

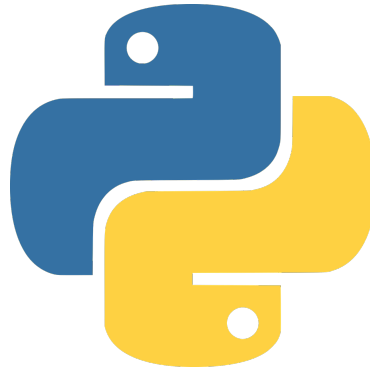


FIGURE 3.6 – Logo du langage Python

3.4.2.1 Bibliothèques utilisées

Les principales bibliothèques utilisées dans le développement de l'application sont les suivantes :

- **tkinter / ttk** : utilisées pour la conception et la gestion de l'interface graphique de l'application.
- **sqlite3** : utilisée pour la gestion et le stockage local des données au sein de la base de données.
- **face_recognition** : utilisée pour l'extraction des encodages faciaux 128 dimensions ainsi que pour la reconnaissance des visages.
- **ultralytics (YOLOv8)** : utilisée pour la détection rapide et performante des visages.
- **cv2 (OpenCV)** : utilisée pour le traitement, l'analyse et la manipulation des images et des flux vidéo.
- **numpy** : utilisée pour les calculs mathématiques, les opérations vectorielles et la manipulation des matrices.
- **PIL / Pillow** : utilisée pour la manipulation, le redimensionnement et l'affichage des images dans l'interface graphique.
- **matplotlib** : utilisée pour la génération de graphiques statistiques et de visualisations de données.
- **pandas** : utilisée pour la manipulation des données structurées et l'exportation vers des fichiers Excel (**.xlsx**).
- **reportlab** : utilisée pour la génération automatique des rapports PDF.
- **threading** : utilisée pour l'exécution asynchrone de certaines tâches afin d'éviter le blocage de l'interface utilisateur.
- **csv / json** : utilisées pour le stockage léger des données et la sauvegarde des fichiers de configuration.

- **hashlib** : utilisée pour le renforcement de la sécurité grâce au hachage des mots de passe avec l'algorithme SHA-256.
- **shutil** / **os** : utilisées pour la gestion des fichiers, des répertoires et des opérations système.

3.4.3 Matériel utilisé

Le système utilisé pour le développement et les expérimentations présente les caractéristiques matérielles suivantes :

- **Processeur** : 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz
- **Mémoire RAM** : 4 Go (3,76 Go utilisables)
- **Stockage** : SSD NVMe 238 Go
- **Carte graphique** : Intel(R) UHD Graphics (128 MB)
- **Système d'exploitation** : Windows 64 bits, architecture x64

Pour la capture des images de test dans un amphithéâtre réel, le dispositif suivant a été utilisé :

- **Téléphone** : Samsung Galaxy A14
- **Capteur principal** : 50 MP
- **Résolution des images utilisées** : 4080×3060 pixels
- **Qualité de l'image** : Haute qualité (images RGB en couleur)

3.4.4 Présentation de l'application

3.4.4.1 Écran d'accueil

L'écran d'accueil propose une interface épurée composée de deux cartes principales (Administrateur / Enseignant). Il prend en charge trois thèmes visuels (**clair**, **normal**, **sombre**) ainsi que trois langues (FR, EN, AR), commutables de manière dynamique via des listes déroulantes (Combobox). Cette interface est présentée dans la Figure 3.7.

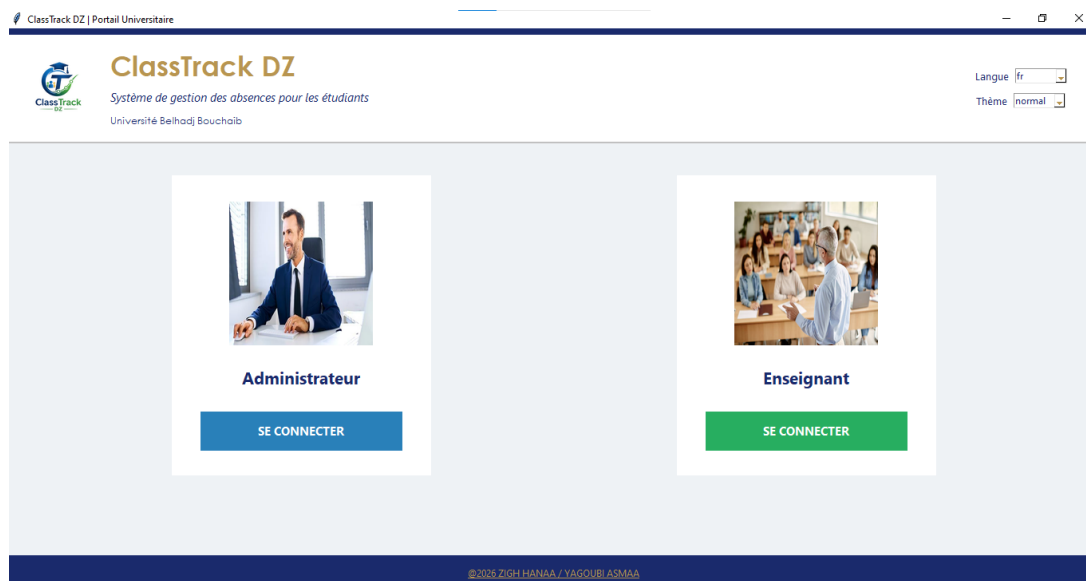


FIGURE 3.7 – Écran d'accueil

3.4.4.2 Interface administrateur

A - Authentification

L'interface d'authentification permet de sécuriser l'accès au système en exigeant un nom d'utilisateur et un mot de passe avant toute connexion. Après la vérification des informations enregistrées dans la base de données, l'utilisateur est redirigé vers le tableau de bord administrateur. Ces deux interfaces sont présentées dans la Figure 3.8.

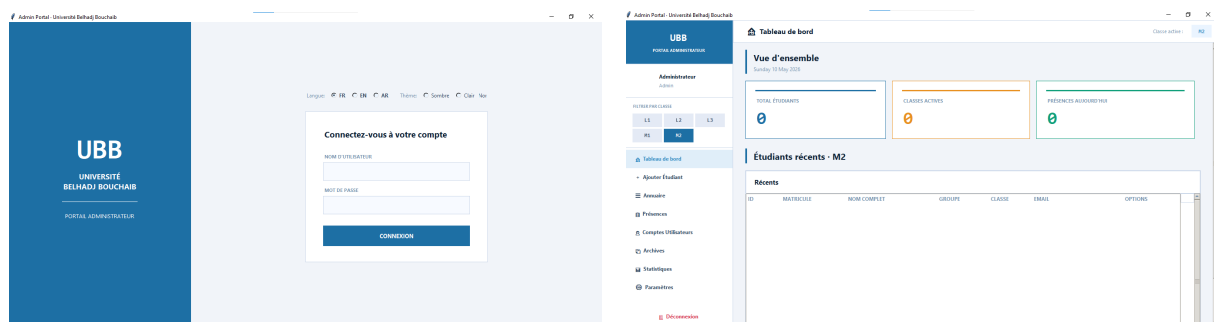


FIGURE 3.8 – Interface de connexion et tableau de bord administrateur

B - Gestion des étudiants

Le formulaire d'ajout d'étudiant comprend :

- des champs à remplir : nom complet, matricule, e-mail, groupe et classe,
- une galerie défilante permettant de sélectionner jusqu'à 20 photos de l'étudiant.

Après validation, les photos sont automatiquement rattachées au dossier de l'étudiant, et l'enregistrement des caractéristiques du visage est effectué de manière transparente. Ces deux interfaces sont illustrées dans la Figure 3.9.

FIGURE 3.9 – Formulaire d'ajout d'un étudiant

C - Annuaire et archivage

L'annuaire présente la liste de tous les étudiants actifs de la classe sélectionnée sous forme d'un tableau filtrable en temps réel. Pour chaque étudiant, on peut ouvrir une fenêtre de détail affichant la galerie de photos ou proposer un dialogue avec trois choix :

- **Archiver** (le compte d'utilisateur est désactivé),
- **Supprimer définitivement** (toutes ses données sont effacées),
- **Annuler**.

Ces deux interfaces sont détaillées dans la Figure 3.10.

FIGURE 3.10 – Interface d'annuaire et d'archivage

D - Gestion des comptes utilisateurs

Cette section permet de créer deux types de comptes : *Enseignant* ou *Admin*. Le mot de passe est sécurisé lors de la création. Des outils de filtrage par rôle ainsi qu'un export au format Excel sont disponibles. Les écrans dédiés à cette administration sont présentés dans la Figure 3.11.

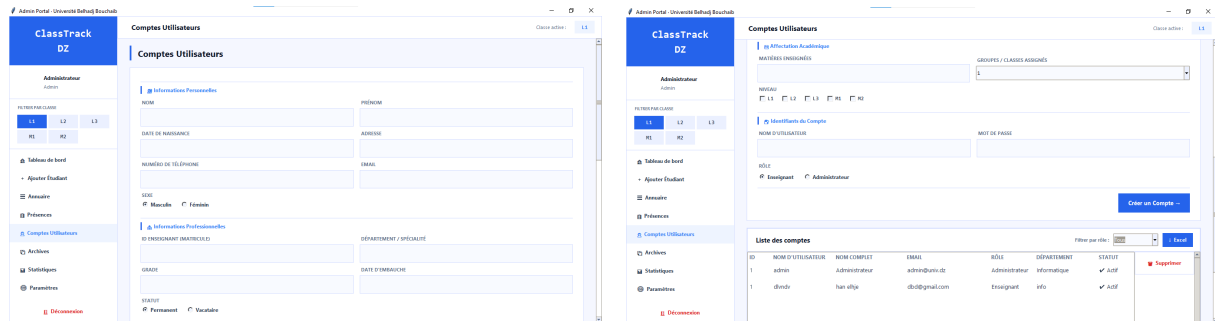


FIGURE 3.11 – Interface de gestion des comptes utilisateurs

3.4.4.3 Interface enseignant

A - Authentification

Cette interface est destinée à l'enseignant, qui peut accéder à son compte en utilisant son nom d'utilisateur et son mot de passe. Ces informations d'authentification sont fournies par l'administrateur. Les écrans dédiés à l'espace enseignant sont détaillés dans la Figure 3.12.

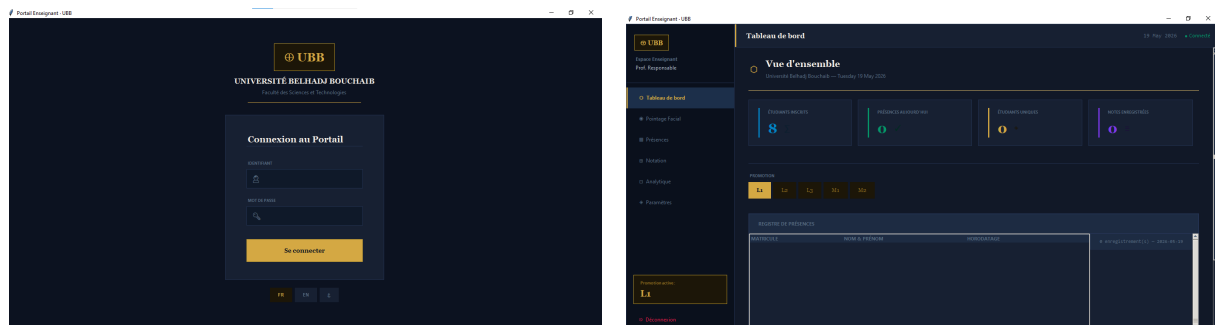


FIGURE 3.12 – Interface de connexion et tableau de bord enseignant

B - Reconnaissance faciale

Cette page est dédiée au processus de reconnaissance faciale. L'enseignant commence par renseigner le nom du module, puis sélectionne le type de séance (cours, travaux dirigés ou travaux pratiques). Dans le cas d'une séance de type TP ou TD, il doit également choisir le groupe concerné.

Une fois ces informations saisies, l'enseignant clique sur le bouton *Initialize session* afin de lancer la séance. Il charge ensuite une photo de la classe prise en amphithéâtre.

À ce niveau, le système utilise la combinaison de **YOLOv8** et **Dlib** pour détecter les visages et reconnaître les étudiants présents sur l'image.

Enfin, les résultats de reconnaissance sont enregistrés dans un tableau, comprenant le nom de l'étudiant ainsi que l'heure de présence. L'interface dédiée à ce processus est montrée dans la Figure 3.13.

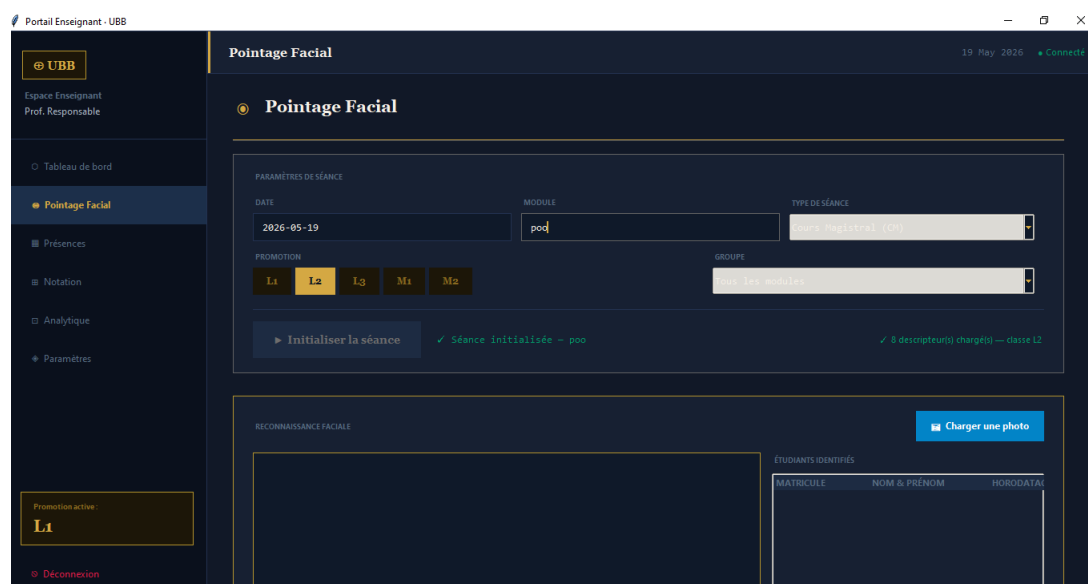


FIGURE 3.13 – Interface de reconnaissance faciale

C - Liste de présence

Lorsque les présences sont enregistrées, elles sont automatiquement sauvegardées dans la base de données du système. Cette page permet à l'enseignant de consulter la liste de présence associée à une séance donnée.

L'enseignant peut ainsi accéder aux enregistrements des présences en recherchant par nom de matière et par date. Cela facilite le suivi des séances et permet une meilleure organisation ainsi qu'un accès rapide aux informations relatives à l'assiduité des étudiants. L'interface d'affichage de cette liste est visualisée dans la Figure 3.14.

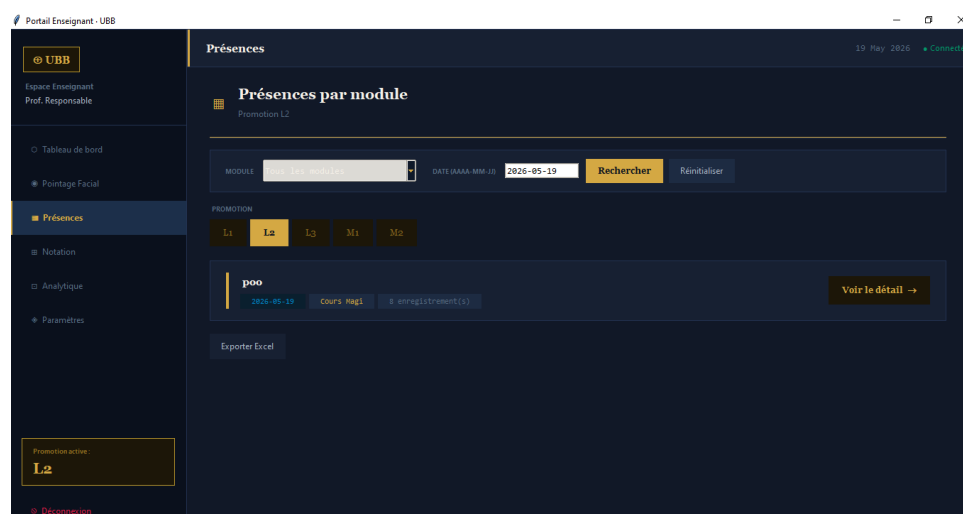


FIGURE 3.14 – Interface d'affichage de la liste des présences

3.5 Résultats

Afin d'évaluer les performances du système de reconnaissance faciale proposé, plusieurs expérimentations ont été réalisées en utilisant la combinaison retenue suite à l'étude comparative, à savoir **YOLOv8** pour la détection des visages et **Dlib** pour la reconnaissance faciale.

La figure 3.15 présente l'architecture générale du pipeline de reconnaissance faciale mis en uvre dans cette étude.

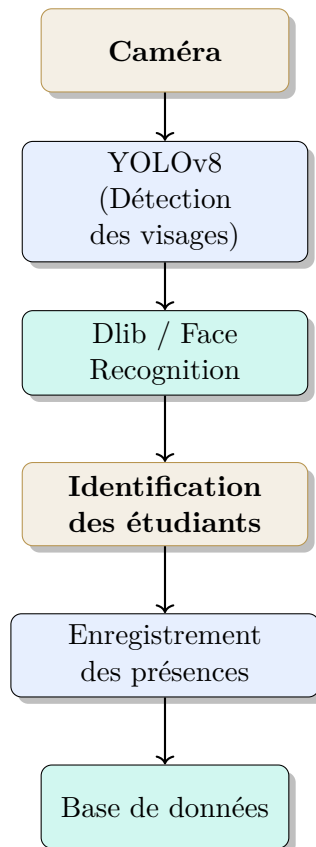


FIGURE 3.15 – Architecture du pipeline de reconnaissance faciale

Cette architecture commence par l'acquisition des images à partir d'une caméra. Les visages sont ensuite détectés à l'aide du modèle YOLOv8. Les régions faciales détectées sont transmises à Dlib pour l'extraction des caractéristiques biométriques et la reconnaissance des individus. Une fois l'étudiant identifié, sa présence est automatiquement enregistrée dans la base de données du système.

Expérimentation sur la base de données de Lateef et Kamil : Dans un premier temps, le système a été testé sur la base de données utilisée par Lateef et Kamil. Après le chargement des descripteurs faciaux enregistrés dans cette base, une image de test contenant plusieurs individus a été soumise au système.

Les résultats obtenus montrent que la combinaison **YOLOv8 + Dlib** a permis de reconnaître correctement **29 personnes** présentes dans l'image de test, comme le montre la Figure 3.16. Cette expérience confirme l'efficacité de l'approche proposée sur une base de données de référence comportant un nombre important d'individus.

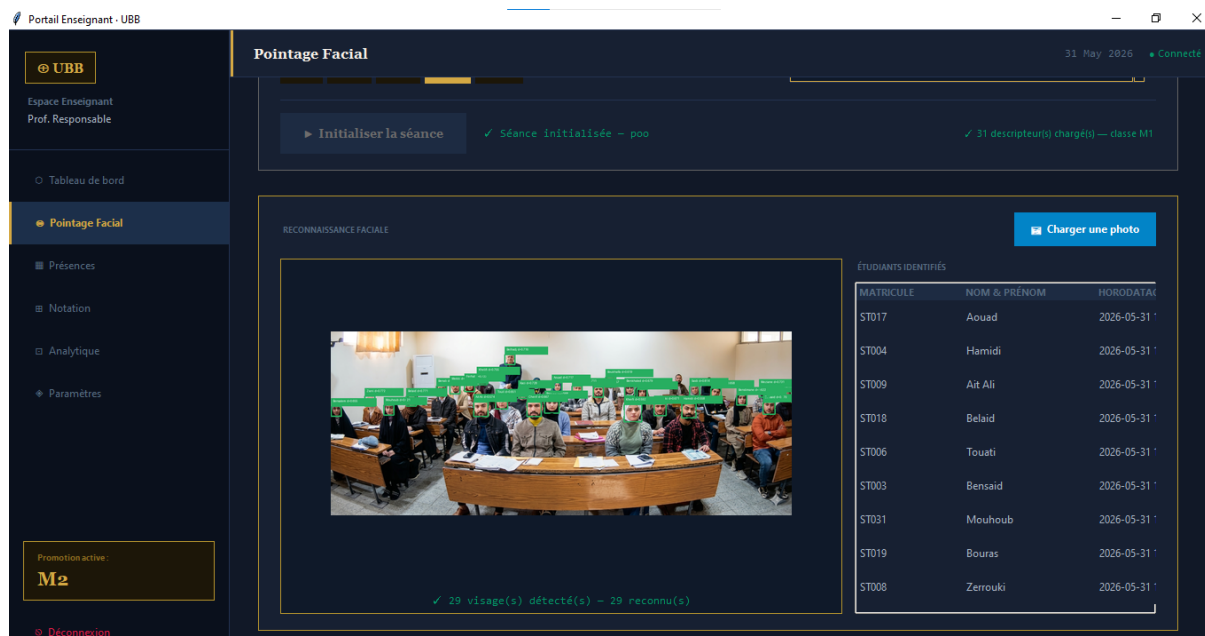


FIGURE 3.16 – Résultat de reconnaissance faciale avec la combinaison YOLOv8 + Dlib

Expérimentation sur la base de données proposée : Une seconde série de tests a été réalisée sur la base de données construite dans le cadre de ce travail. Cette base est composée de **8 étudiants de Licence 2 Informatique**. Pour chaque étudiant, trois images faciales ont été collectées selon différentes orientations du visage : face, profil gauche et profil droit.

Après l'entraînement et le chargement des descripteurs faciaux correspondants, le système a été appliqué à une image contenant les huit étudiants. Les résultats montrent que le système a correctement identifié **l'ensemble des 8 étudiants présents**, soit un taux de reconnaissance de 100 %.

Les expérimentations réalisées sur les deux bases de données montrent que l'approche proposée offre une bonne précision de reconnaissance. En particulier, les résultats obtenus sur la base de données développée dans ce travail confirment la fiabilité du système pour l'automatisation du pointage des étudiants dans les salles de cours universitaires.

3.6 Difficultés rencontrées

Au cours du développement du projet, plusieurs difficultés ont été rencontrées :

- **Limitations matérielles :** La machine utilisée dispose de ressources limitées (4

Go de RAM et carte graphique intégrée), ce qui rend difficile l'exécution de certains algorithmes d'intelligence artificielle gourmands en calcul.

- **Qualité des données** : Les images des bases de données présentent des variations (éclairage, angles, résolution), ce qui a influencé la performance des modèles de reconnaissance.
- **Gestion des dépendances** : L'intégration de bibliothèques telles que OpenCV et YOLOv8 a nécessité une attention particulière à la compatibilité des versions de Python et des packages.
- **Performance en traitement rapide** : L'exécution des algorithmes en temps réel a parfois présenté des lenteurs, nécessitant des optimisations du pipeline de traitement.

3.7 Perspectives futures

Dans le cadre de l'amélioration et du développement futur du système, plusieurs perspectives peuvent être envisagées :

- **Amélioration des performances** : Optimiser les algorithmes utilisés afin de réduire le temps de traitement et améliorer la précision, notamment pour les systèmes rapides.
- **Intégration de modèles plus avancés** : Utiliser des architectures plus récentes et plus performantes en deep learning, telles que les Transformers pour la vision ou des modèles optimisés pour les appareils embarqués.
- **Déploiement sur mobile** : Adapter le système pour une utilisation sur smartphones afin de faciliter son accessibilité dans les environnements académiques et professionnels.
- **Renforcement de la base de données** : Augmenter la taille et la diversité des datasets afin d'améliorer la robustesse du système face aux variations d'éclairage, de pose et d'environnement.
- **Sécurisation du système** : Intégrer des mécanismes de sécurité avancés pour protéger les données personnelles et garantir la confidentialité des utilisateurs.
- **Intégration de nouvelles fonctionnalités** : Ajouter des fonctionnalités supplémentaires comme la reconnaissance multi-caméras, le suivi rapide ou encore des tableaux de bord analytiques.

3.8 Conclusion

Ce chapitre a présenté l'ensemble du processus de conception, de développement et de réalisation du système intelligent de gestion automatisée des présences **ClassTrack DZ**. L'analyse des besoins a permis d'identifier les limites des méthodes traditionnelles de gestion des absences, notamment la perte de temps, les erreurs humaines et les risques de fraude, ainsi que de définir les objectifs principaux du système.

La phase de conception a ensuite permis de structurer la solution proposée à travers une architecture générale claire et des diagrammes UML décrivant les interactions entre les différents acteurs et composants du système. Cette étape a facilité la compréhension globale du fonctionnement de l'application et de son organisation interne.

L'implémentation a mis en évidence les choix techniques adoptés, ainsi que l'intégration des différentes technologies utilisées, notamment les bibliothèques de vision par ordinateur et de reconnaissance faciale. Elle a également montré la manière dont les différents modules ont été développés et intégrés afin de garantir le bon fonctionnement du système.

En conclusion, ce chapitre a permis de transformer une solution conceptuelle en une application concrète basée sur l'intelligence artificielle, capable d'automatiser le processus de gestion des présences. Le chapitre suivant sera consacré à l'évaluation des performances du système ainsi qu'à l'analyse des résultats expérimentaux obtenus.

Conclusion générale

La gestion manuelle des présences dans les établissements d'enseignement supérieur demeure un processus chronophage, source d'erreurs et vulnérable aux fraudes. L'appel nominal effectué en début de séance, ainsi que l'usage de feuilles d'émargement, mobilisent un temps précieux qui pourrait être consacré à l'enseignement, tout en exposant le processus à des falsifications difficiles à détecter, telles que la signature pour le compte d'un camarade absent. Ces limites, largement documentées dans la littérature et confirmées par notre étude de l'existant, justifient la recherche de solutions technologiques alternatives capables de fiabiliser ce processus tout en allégeant la charge administrative des enseignants.

Face à ces limites, la reconnaissance faciale s'impose comme une alternative technologique prometteuse pour automatiser et fiabiliser le suivi académique. Les avancées récentes en matière d'intelligence artificielle et de vision par ordinateur, en particulier dans le domaine de l'apprentissage profond, ont rendu possible la conception de systèmes de détection et de reconnaissance faciale suffisamment précis et rapides pour être déployés dans des conditions réelles, y compris dans des environnements aussi exigeants qu'un amphithéâtre universitaire regroupant un grand nombre d'étudiants.

L'objectif principal de ce mémoire consistait à concevoir et valider un système intelligent de gestion automatisée des présences basé sur la reconnaissance faciale, capable d'assurer une identification fiable, rapide et adaptée aux contraintes des environnements universitaires réels. Pour atteindre cet objectif, notre démarche s'est articulée en trois étapes complémentaires : une étude théorique approfondie des fondements de l'intelligence artificielle et de l'apprentissage profond, une analyse comparative rigoureuse des principales approches existantes de détection et de reconnaissance faciale, et enfin la conception, l'implémentation et l'évaluation expérimentale d'une solution logicielle complète, dénommée ClassTrack DZ.

En réponse à cette problématique, les résultats expérimentaux démontrent qu'un pipeline hybride associant YOLOv8 pour la détection et Dlib pour la reconnaissance répond pleinement aux exigences de fiabilité et de rapidité. Ce choix architectural résulte d'une analyse comparative méthodique de plusieurs approches candidates, menée sur la base de critères tels que la précision de détection, la robustesse face aux variations d'éclairage et d'angle de prise de vue, ainsi que le temps de traitement, déterminant pour une utilisation

en conditions réelles de classe.

Cette combinaison a atteint une précision de 100% sur les deux corpus expérimentaux, tout en réduisant significativement le temps d'inférence par rapport aux détecteurs conventionnels tels que ViolaJones, HOG ou MTCNN. Ce résultat confirme la pertinence du choix architectural retenu et valide empiriquement les hypothèses formulées en amont du projet quant à la supériorité des approches fondées sur les réseaux de neurones convolutifs modernes par rapport aux méthodes plus classiques de traitement d'image.

Les expérimentations ont clairement mis en évidence la supériorité de YOLOv8 en détection multi-visages, grâce à son approche en une seule passe (single forward pass), qui lui permet de traiter simultanément l'ensemble des visages présents dans une image, contrairement aux approches séquentielles plus lentes et moins adaptées à un contexte de salle de classe dense. Cette caractéristique s'est révélée déterminante dans le cadre de notre application, où le système doit être capable d'identifier rapidement et de manière fiable un grand nombre d'étudiants à partir d'une seule capture d'image. Ainsi, faisant de l'architecture YOLOv8 + Dlib le compromis optimal pour un déploiement en amphithéâtre, ces résultats constituent une contribution scientifique notable, en apportant des éléments de réponse concrets et quantifiés à la question du choix d'architecture pour ce type d'application.

Le développement de l'application ClassTrack DZ a permis de concrétiser ces contributions théoriques en une solution opérationnelle, testée et évaluée dans des conditions proches du réel. Implémentée de manière modulaire en Python, l'architecture garantit la reproductibilité scientifique et intègre une interface pédagogique sécurisée, conforme aux exigences éthiques liées aux données biométriques. Cette modularité facilite par ailleurs la maintenance évolutive du système, ainsi que l'intégration future de nouvelles fonctionnalités ou l'adaptation à d'autres contextes d'usage.

Au-delà de l'aspect purement technique, ce travail a également nécessité une réflexion approfondie sur les questions éthiques et réglementaires liées au traitement de données biométriques, en particulier en ce qui concerne le consentement des étudiants, la sécurisation du stockage des encodages faciaux, ainsi que la transparence du système vis-à-vis des utilisateurs finaux. Cette dimension éthique, souvent négligée dans les travaux purement techniques, a été intégrée dès la phase de conception, conformément aux bonnes pratiques recommandées dans le domaine de l'intelligence artificielle responsable.

Le système répond ainsi aux besoins actuels de transformation numérique des universités en proposant une solution automatisée, flexible et évolutive, susceptible d'être déployée à l'échelle d'un établissement, voire généralisée à d'autres institutions confrontées à des problématiques similaires de gestion des présences.

Il convient toutefois de souligner certaines limites inhérentes à ce travail. Les expérimentations menées, bien que représentatives, reposent sur des corpus de taille limitée et acquis dans des conditions spécifiques, ce qui invite à la prudence quant à la généralisation

des résultats obtenus à d'autres contextes, notamment en présence de conditions d'éclairage plus difficiles, d'occlusions partielles du visage, ou de variations démographiques plus importantes au sein de la population étudiante. De plus, le système actuel ne dispose pas encore de mécanismes de détection de tentatives de fraude par usurpation d'identité, ce qui constitue une vulnérabilité potentielle dans un contexte d'évaluation académique.

Enfin, ce travail ouvre plusieurs perspectives de recherche, notamment l'intégration de mécanismes de liveness detection pour contrer les tentatives de spoofing, qui consisteraient à tromper le système à l'aide d'une photographie ou d'une vidéo préenregistrée du visage d'un étudiant. Cette amélioration renforcerait significativement la robustesse du système face aux tentatives de fraude, en garantissant que seule une personne physiquement présente puisse être reconnue comme telle.

Une seconde perspective concerne l'optimisation du pipeline pour des plateformes embarquées, en vue d'un déploiement sur des dispositifs à ressources limitées tels que des caméras intelligentes ou des terminaux mobiles, ce qui permettrait de réduire les coûts d'infrastructure et de faciliter l'adoption du système à plus grande échelle. Enfin, le déploiement en flux vidéo continu constitue une évolution naturelle du système actuel, permettant un suivi de présence en temps réel tout au long de la séance, plutôt qu'à un instant unique, et ouvrant la voie à des analyses plus fines du comportement et de l'engagement des étudiants en classe.

Il confirme que l'alliance de l'intelligence artificielle et de la vision par ordinateur constitue un levier stratégique pour moderniser la gestion académique et bâtir des environnements éducatifs plus intelligents et plus efficaces. Ce mémoire constitue ainsi une contribution à la fois théorique et pratique à la problématique de l'automatisation de la gestion des présences, et pose les bases d'un système destiné à évoluer pour répondre aux besoins croissants de digitalisation des établissements d'enseignement supérieur.

Bibliographie

- [1] Russell, Stuart J. Artificial intelligence a modern approach. Pearson Education, Inc., 2010.
- [2] Chumakov, E. "History and definitions of AI." *European Psychiatry* 68.S1 (2025) : S61-S61.
- [3] Al-Nayyef, Huda. "Advancing Attendance : A Facial Recognition System Empowered by Deep Learning Techniques." *Journal of Al-Qadisiyah for Computer Science and Mathematics* 16.1 (2024) : 61-71.
- [4] Tolulope, T. Oluyemi, T. Oyediji Funke, and J. Oyebiyi Adewale. "Development of an Attendance Management System Using Facial Recognition Technology." *Journal of Engineering Research and Reports* 26.10 (2024) : 297-307.
- [5] Zhou, Zhi-Hua. *Machine learning*. Springer nature, 2021.
- [6] Chen, Xiangping, et al. "Deep learning-based software engineering : progress, challenges, and opportunities." *Science China Information Sciences* 68.1 (2025) : 111102.
- [7] Feri, Muhammad, et al. "Implementing deep learning approaches in primary education : A literature review." *Jurnal Varidika* (2025) : 178-195.
- [8] Mehra, Aditya. "Hybrid AI models : Integrating symbolic reasoning with deep learning for complex decision-making." *Journal of Emerging Technologies and Innovative Research* 11.8 (2024) : f693-f695.
- [9] Golilarz, Noorbakhsh Amiri, et al. "AI Learning Algorithms : Deep Learning, Hybrid Models, and Large-Scale Model Integration." *arXiv preprint arXiv :2410.09186* (2024).
- [10] Younesi, Abolfazl, et al. "A comprehensive survey of convolutions in deep learning : Applications, challenges, and future trends." *IEEE Access* 12 (2024) : 41180-41218.
- [11] Renner, A., Sheldon, F., Zlotnik, A. et al. The backpropagation algorithm implemented on spiking neuromorphic hardware. *Nat Commun* 15, 9691 (2024).
- [12] Deng, C. (2024). A review of face recognition technologies based on deep learning. *Applied and Computational Engineering*, 46, 297-303.

- [13] Zhao, Qihao, et al. "Research on intrusion detection model based on improved MLP algorithm." *Scientific reports* 15.1 (2025) : 5159.
- [14] Mienye ID, Swart TG. A Comprehensive Review of Deep Learning : Architectures, Recent Advances, and Applications. *Information*. 2024; 15(12) :755.
- [15] Khan, Saddam Hussain, and Rashid Iqbal. "A Comprehensive Survey on Architectural Advances in Deep CNNs : Challenges, Applications, and Emerging Research Directions." *arXiv preprint arXiv :2503.16546* (2025).
- [16] Monbroussou, Léo, et al. "Subspace preserving quantum convolutional neural network architectures." *Quantum Science and Technology* 10.2 (2025) : 025050.
- [17] Sun, Zhifang, and Zhe Liu. "Ensuring privacy in face recognition : a survey on data generation, inference and storage." *Discover Applied Sciences* 7.5 (2025) : 441.
- [18] Chen, Shaokang, et al. "Face recognition from still images to video sequences : A local-feature-based framework." *EURASIP journal on image and video processing* 2011.1 (2010) : 790598.
- [19] Karamizadeh, Sasan, Saman Shojae Chaeikar, and Hamidreza Salarian. "Combining MTCNN and Enhanced FaceNet with Adaptive Feature Fusion for Robust Face Recognition." *Technologies* 13.10 (2025) : 450.
- [20] Ren, Zhengwei, et al. "LittleFaceNet : A Small-Sized Face Recognition Method Based on RetinaFace and AdaFace." *Journal of Imaging* 11.1 (2025) : 24.
- [21] Agustiyar, Agustiyar, R. Rizal Isnanto, and Catur Edi Widodo. "Face Recognition for Attendance Systems : A Bibliometric Review of Research Trends and Opportunities." *Jurnal Sisfokom (Sistem Informasi dan Komputer)* 15.01 (2026) : 8-13.
- [22] Xing, Huifen, et al. "Face anti-spoofing based on deep learning : A comprehensive survey." *Applied Sciences* 15.12 (2025) : 6891.
- [23] Lateef, Ahmad S., and Mohammed Y. Kamil. "Facial Recognition Technology-Based Attendance Management System Application in Smart Classroom." *Iraqi Journal for Computer Science and Mathematics* 4.3 (2023) : 12.
- [24] Singh, Ranjeet, and Mandeep Kaur. "Face recognition and detection using Viola-Jones and cross correlation method." *International Journal of Science and Research (IJSR)* 4.1 (2015) : 2498-2501.
- [25] Sunaryono, Dwi, Joko Siswantoro, and Radityo Anggoro. "An android based course attendance system using face recognition." *Journal of King Saud University-Computer and Information Sciences* 33.3 (2021) : 304-312.
- [26] Alniemi, Omar, and Hanaa F. Mahmood. "Class Attendance System Based on Face Recognition." *Revue d'intelligence artificielle* 37.5 (2023).

- [27] Lalitha, S., et al. "AI-Driven Attendance Tracking with Haar Cascade and ResNet." *Procedia Computer Science* 258 (2025) : 3544-3553.
- [28] Shafiq, Muhammad, and Zhaoquan Gu. "Deep residual learning for image recognition : A survey." *Applied sciences* 12.18 (2022) : 8972.
- [29] Kachouane, M., et al. "HOG based fast human detection." 2012 24th International Conference on Microelectronics (ICM). IEEE, 2012.
- [30] Li, Xiaochao, Zhenjie Yang, and Hongwei Wu. "Face detection based on receptive field enhanced multi-task cascaded convolutional neural networks." *IEEE access* 8 (2020) : 174922-174930.
- [31] Karamizadeh, Sasan, Saman Shojae Chaeikar, and Hamidreza Salarian. "Combining MTCNN and Enhanced FaceNet with Adaptive Feature Fusion for Robust Face Recognition." *Technologies* 13.10 (2025) : 450.
- [32] Wei, Yingjie, et al. "Robust face mask detection in complex scenarios using YOLOv8 and context-aware convolutions." *Scientific Reports* 15.1 (2025) : 21350.
- [33] Tran, Duy Anh, et al. "Real-time object detection for autonomous driving using deep learning." *Institute of Computer Science, Department of Computer Science and Mathematics, Goethe University Frankfurt* 15 (2021).
- [34] Jocher, G. "Ultralytics YOLOv8 documentation. Ultralytics." 2023.
- [35] Pecolt, Sebastian, et al. "Personal identification using embedded raspberry pi-based face recognition systems." *Applied Sciences* 15.2 (2025) : 887.
- [36] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet : A unified embedding for face recognition and clustering." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
- [37] Grandini, Margherita, Enrico Bagli, and Giorgio Visani. "Metrics for multi-class classification : an overview." *arXiv preprint arXiv :2008.05756* (2020).
- [38] Powers, David MW. "Evaluation : from precision, recall and F-measure to ROC, informedness, markedness and correlation." *arXiv preprint arXiv :2010.16061* (2020).