République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj
Faculté des Mathématiques et d'Informatique
Département d'informatique

**MÉMOIRE**
Présenté en vue de l'obtention du diplôme
**Master en Informatique**
Spécialité : Technologies de l'information et de la communication

# THÈME

# Système de détection de la somnolence du conducteur (ai)

*Présenté par :*

Fatima Zohra KHABABA

Halima KHOUDOUR

*Soutenu publiquement le : 28/06/2026*
*Devant le jury composé de :*

**Président :** Pr. BOUZIANE ABDERAOUF

**Examinateur :** Dr. BEGHOURA MOHAMMED AMINE

**Encadrant :** Pr. NOUIOUA FARID

**2025/2026**

# اهداء

بسم الله الرحمن الرحيم

الحمد لله الذي وفقني وأعانني على إتمام هذا العمل، فبفضله سبحانه وتعالى تحققت الخطوات التي ظننت يوماً أنها بعيدة، وله الحمد والشكر على نعمه التي لا تُعد ولا تُحصى

أهدي ثمرة جهدي هذا إلى والديّ العزيزين، اللذين كانا مصدر دعمي وقوتي، واللذين رافقاني بدعواتهما وتضحياتهما طوال مشواري الدراسي، فجزاهما الله عني خير الجزاء وأدامهما تاجاً فوق رأسي

إلى إخوتي الثلاثة، الذين كانوا دائماً سنداً وعوناً لي، وإلى زوجة أخي الغالية التي أحاطتني بمودتها وتشجيعها، شكراً لكم على محبتكم ودعمكم الذي كان له الأثر الكبير في مواصلة الطريق

إلى صديقتي العزيزة **حليمة خضور**، رفيقة الدرب وأجمل صدفة في هذه المرحلة من حياتي، شكراً لأنكِ كنتِ إلى جانبي في أوقات التعب قبل الفرح، ولأن كلماتكِ المشجعة كانت دائماً تمنحني القوة للاستمرار. وجودكِ جعل هذا المشوار أخف وأجمل، وستبقى صداقتكِ من أثمن ما كسبته خلال سنوات الدراسة

إلى كل أفراد عائلتي وأصدقائي، وإلى كل من دعمني بكلمة طيبة أو دعوة صادقة أو موقف جميل، أهديكم هذا العمل المتواضع عربون محبة وامتنان

أسأل الله أن يجعل هذا العمل خالصاً لوجهه الكريم، وأن يبارك ثمرة هذا الجهد، ويوفقني لما هو خير في مستقبلي

Fatima Zohra KHABABA

# اهداء

إلى نفسي أولًا....
إلى تلك الفتاة التي خاضت هذا الطريق بكل ما حمله من تعب وتحديات، إلى من وقفت مرة بعد مرة رغم الإرهاق، وإلى من اختارت الاستمرار عندما كان التوقف أسهل.
إلى نفسي التي تعلمت من الصعوبات، وكبرت مع كل تجربة، وتحدّتها بصمتٍ لكي تقف اليوم هنا؛
أهدي لكِ هذا العمل تقديرًا لكل جهد بذلته، ولكل خطوة قطعتها، ولكل مرة آمنتِ فيها بقدرتك على النجاح

إلى عائلتي العزيزة...
إلى أمي الغالية، التي كانت دعواتها تدفعني دائمًا للأمام، وإلى أبي، وإلى أختي ورفيقة قلبي، وإلى الصغيرة الغالية "حبيبة خالتو" التي تملأ دنيانا بهجة وسعادة

إلى رفيقة الدرب وشريكتي في هذا التحدي،
إلى من شاركتني السهر، وصبرت معي على ضغط الأيام حتى جعلنا هذا المشروع فخراً نعتز به معاً

...إلى كل من آمن بي وكان يحمل الخير لي في قلبه
إلى الذين منحوني صادق دعواتهم، وألهموني بكلمة طيبة، أو وجّهوني بنصيحة، وكانوا طاقة إيجابية تُضيء لي مسيرتي الدراسية

الحمد لله على الوصول، وعلى القوة التي منحتني إياها لأحقق هذا الإنجاز.

Halima KHOUDOUR

# Remerciement

Nous remercions Dieu, avant tout, pour sa grâce et son aide dans chaque bienfait et chaque action. Nous le remercions également de nous avoir accordé le succès sur notre chemin et sa bénédiction dans la quête de la connaissance et dans nos efforts pour l’obtenir.

Nous tenons à exprimer notre profonde gratitude à notre encadrant, Dr. Farid NOUIOUA, pour sa précieuse orientation, sa disponibilité constante, ses conseils avisés et la confiance qu'il nous a accordée tout au long de la réalisation de ce projet.

Nos vifs remerciements s'adressent également aux membres du jury qui ont accepté d’évaluer notre travail et d’honorer notre soutenance de leur présence et de leurs précieuses remarques.

Enfin, nous remercions chaleureusement tous les enseignants qui ont contribué à notre formation universitaire, ainsi que toutes les personnes qui, de près ou de loin, nous ont apporté leur soutien, leur aide et leurs encouragements durant notre parcours.

# Résumé

La somnolence au volant constitue une cause majeure d'accidents de la route, souvent sous-estimée en raison de l'absence d'outils de mesure objectifs. Ce mémoire présente la conception et la réalisation de Driver Guard, un système hybride non intrusif de détection de la somnolence du conducteur basé sur la fusion de trois approches de vision par ordinateur : MediaPipe FaceMesh (50 %), YOLOv8n (30 %) et un réseau CNN MobileNetV2 (20 %). Le système intègre un mécanisme de vote pondéré, une gestion adaptative de l'éclairage nocturne, un système de recommandation GPS des aires de repos via l'API Overpass, ainsi qu'un module d'urgence capable d'envoyer un message SOS géolocalisé via Telegram. Une application mobile multiplateforme (Flutter) assure l'interface conducteur et la transmission des alertes. Les résultats expérimentaux montrent une précision globale de 96,81 % pour le modèle CNN et un mAP@0.5 de 98,6 % pour YOLOv8n. Le système a été validé sur cinq scénarios critiques de conduite, démontrant sa robustesse en temps réel.

**Mots-clés :** somnolence au volant, détection temps réel, MediaPipe, YOLOv8, MobileNetV2, fusion pondérée, sécurité routière, application mobile Flutter, système d'urgence géolocalisé.

# Abstract

Drowsy driving is a leading cause of road accidents, yet it remains largely underestimated due to the lack of objective measurement tools. This thesis presents the design and implementation of Driver Guard, a non-intrusive hybrid driver drowsiness detection system based on the fusion of three computer vision approaches: MediaPipe FaceMesh (50%), YOLOv8n (30%), and a MobileNetV2 CNN model (20%). The system incorporates a weighted voting mechanism, adaptive nighttime lighting management, a GPS rest area recommendation system via the Overpass API, and an emergency module capable of sending a geolocated SOS message via Telegram. A cross-platform mobile application (Flutter) provides the driver interface and alert transmission. Experimental results show an overall accuracy of 96.81% for the CNN model and a mAP@0.5 of 98.6% for YOLOv8n. The system was validated on five critical driving scenarios, demonstrating its real-time robustness.

**Keywords:** drowsy driving, real-time detection, MediaPipe, YOLOv8, MobileNetV2, weighted voting, road safety, Flutter mobile app, geolocated emergency system.

# ملخص

يُعد النعاس أثناء القيادة أحد الأسباب الرئيسية لحوادث الطرق، لكنه غالباً ما يقل تقديره لعدم وجود أدوات قياس موضوعية. يقدم هذا العمل تصميم وتنفيذ نظام Driver Guard، وهو نظام هجين غير تدخلي للكشف عن نعاس السائق يعتمد على دمج ثلاث تقنيات للرؤية الحاسوبية: MediaPipe FaceMesh (50%)، YOLOv8n (30%)، وشبكة CNN من نوع MobileNetV2 (20%). يشتمل النظام على آلية تصويت مرجح، وإدارة تكيفية للإضاءة الليلية، ونظام توصيات لمواقع الراحة عبر GPS باستخدام Overpass API، بالإضافة إلى وحدة طوارئ قادرة على إرسال رسالة استغاثة مع الموقع الجغرافي عبر Telegram. يوفر تطبيق جوال متعدد المنصات (Flutter) واجهة للسائق ونقل التنبيهات. أظهرت النتائج التجريبية دقة إجمالية قدرها 96.81% للنموذج CNN وقيمة mAP@0.5 قدرها 98.6% للنموذج YOLOv8n. تم التحقق من صحة النظام عبر خمسة سيناريوهات حرجة للقيادة، مما يثبت متانته في الزمن الحقيقي.

**كلمات مفتاحية:** النعاس عند القيادة، كشف في الزمن الحقيقي، MediaPipe، YOLOv8، MobileNetV2، تصويت مرجح، السلامة الطرقية، تطبيق Flutter، نظام استغاثة جغرافي.

# Table des matières

**Abstract** ..... **vi**

**Table des matières** ..... **viii**

**Liste des abréviations** ..... **xiii**

**Liste des figures** ..... **xv**

**Liste des tableaux** ..... **xv**

**Introduction Générale**

1.Contexte ..... 1

2.Problématique ..... 2

3.Objectifs ..... 2

4.Méthodologie et résultats ..... 3

5.Structure du rapport ..... 4

**Chapitre 1 : Somnolence au volant – Analyse du problème et solutions envisagées**

1.1 Introduction ..... 5

1.2 Ampleur des accidents de la route et facteurs de risque ..... 6

1.2.1 Statistiques des accidents de la route en Algérie et dans le monde ..... 7

1.2.2 Statistiques des accidents liés à la fatigue et la somnolence ..... 7

1.2.3 Coût humain et économique des accidents ..... 8

1.3 Facteurs provoquant la somnolence au volant ..... 9

1.3.1 Facteurs comportementaux et socio-comportementaux ..... 9

1.3.2 Facteurs pathologiques (troubles du sommeil) ..... 10

1.3.3 Facteurs iatrogéniques et toxiques ..... 10

1.3.4 Facteurs environnementaux et infrastructurels ..... 10

1.4 Indicateurs physiologiques de la somnolence ..... 11

1.4.1 Indicateurs oculaires ..... 11

1.4.2 Indicateurs buccaux ..... 11
1.4.3 Indicateurs céphaliques ..... 11
1.5 Systèmes existants et leurs limites ..... 12
1.5.1 Systèmes d'alerte simples ..... 12
1.5.2 Systèmes de capteurs biométriques (ECG, EEG) ..... 12
1.5.3 Systèmes de détection par vision par ordinateur ..... 12
1.6 Limites des approches existantes ..... 13
1.7 Conclusion ..... 14
**Chapitre 2 : Fondements théoriques Apprentissage profond et Vision par ordinateur**
2.1 Introduction ..... 15
2.2 Apprentissage automatique (Machine Learning) ..... 15
2.2.1 Types d'apprentissage automatique ..... 16
2.2.1.1 Apprentissage supervisé ..... 16
2.2.1.2 Apprentissage non supervisé ..... 16
2.2.1.3 Apprentissage par renforcement ..... 16
2.3 Apprentissage profond (*Deep Learning*) ..... 16
2.3.1 Réseaux de neurones convolutionnels (CNN) ..... 17
2.3.2 MobileNetV2 et apprentissage par transfert (Transfer Learning) ..... 17
2.4 YOLO : détection d'objets en temps réel ..... 18
2.4.1 Principe de fonctionnement de YOLO ..... 18
2.4.2 Architecture YOLOv8 et son évolution ..... 19
2.5 Mediapipe FaceMesh : détection des points du visage ..... 19
2.5.1 Principe du FaceMesh (468 points de repère) ..... 19
2.5.2 Extraction des points des yeux et de la bouche ..... 20
2.5.3 Métriques et formules mathématiques ..... 20
2.6 Conclusion ..... 22

**Chapitre 3 : Conception et réalisation du système**

3.1 Introduction .....23

3.2 Architecture générale du système — trois phases .....23

3.2.1 Phase 1 : Prévention .....24

3.2.2 Phase 2 : Intervention .....24

3.2.3 Phase 3 : Urgence (Emergency) .....25

3.3 Proposition d'un système hybride de détection de la somnolence .....26

3.3.1 Rôle de chaque modèle dans le pipeline .....27

3.3.2 Mécanisme de fusion pondérée proposé (Weighted Voting) .....27

3.3.3 Priority Fusion : résolution des conflits entre modèles .....28

3.3.4 LightingManager : gestion de l'éclairage nocturne .....29

3.4 Système de recommandation de lieux de repos (GPS) .....30

3.4.1 Overpass API et recherche dans OpenStreetMap .....30

3.4.2 Formule de Haversine pour le calcul des distances .....30

3.5 Système SOS automatique .....30

3.5.1 Logique de détection (10 secondes de sommeil / absence du visage) .....31

3.5.2 Telegram Bot API pour l'envoi du message de détresse .....31

3.5.3 Enregistrement vidéo automatique des incidents .....31

3.6 Diagrammes du système (UML) .....31

3.6.1 Diagramme de cas d'utilisation (Use Case Diagram) .....32

3.6.2 Diagramme de séquence (Sequence Diagram) .....35

3.6.3 Diagramme de classe (Class Diagram) .....37

3.7 Conclusion .....38

**Chapitre 4 : Mise en œuvre et évaluation des résultats**

4.1 Introduction .....39

4.2 Environnement expérimental .....40

4.2.1 Caractéristiques matérielles .....40

4.2.2 Technologies et bibliothèques utilisées....40
4.3 Modèle CNN....41
4.3.1 Source, classes et répartition des données....41
4.3.2 Architecture MobileNetV2 et entraînement....43
4.3.3 Résultats et évaluation....43
4.4 Modèle YOLOv8....45
4.4.1 Source, classes et répartition des données....45
4.4.2 Architecture YOLOv8n et entraînement....46
4.4.3 Résultats et évaluation....46
4.5 Comparaison et évaluation du système hybride....47
4.5.1 Comparaison CNN vs YOLOv8....47
4.5.2 Rôle de MediaPipe dans le système....47
4.5.3 Performance globale du système hybride....48
4.6 Application mobile Flutter (Driver Guard)....49
4.6.1 Architecture de l'application Flutter....49
4.6.2 Serveur Flask et liaison téléphone-caméra....49
4.6.3 Points d'API, journalisation et sécurité....50
4.7 Présentation de notre application....51
4.7.1 Nom et Logo....51
4.7.2 Présentation des interfaces graphiques....51
4.8 Test du système en conditions simulées....55
4.8.1 Scénario 1 : somnolence — alerte sonore....55
4.8.2 Scénario 2 : bâillements répétés — GPS....56
4.8.3 Scénario 3 : sommeil profond — SOS Telegram....58
4.8.4 Scénario 4 : inclinaison de tête....59
4.8.5 Scénario 5 : absence du visage — SOS....60
4.9 Comparaison avec les travaux antérieurs....61

4.10 Conclusion ..... 63

**Conclusion Générale**

1.Contributions ..... 64

2.Critique du travail ..... 65

3.Travaux Futurs et perspectives ..... 66

**Références**

# Liste des abréviations

**AAA** : American Automobile Association

**ADAS** : Advanced Driver-Assistance Systems

**API** : Application Programming Interface

**AVI** : Audio Video Interleave

**CLAHE** : Contrast Limited Adaptive Histogram Equalization

**CNESE** : Conseil National Économique, Social et Environnemental

**CNN** : Convolutional Neural Network

**CNPR** : Centre National de Prévention Routière

**COCO** : Common Objects in Context

**CPU** : Central Processing Unit

**CSV** : Comma-Separated Values

**DNSR** : Délégation Nationale à la Sécurité Routière

**EAR** : Eye Aspect Ratio

**ECG** : Électrocardiogramme

**EEG** : Électroencéphalogramme

**ESS** : Epworth Sleepiness Scale

**FHWA** : Federal Highway Administration

**FPS** : Frames Per Second

**GPS** : Global Positioning System

**GPU** : Graphics Processing Unit

**GRU** : Gated Recurrent Unit

**HTML** : HyperText Markup Language

**IA** : Intelligence Artificielle

**iOS** : iPhone Operating System

**IoT** : Internet of Things

**IoU** : Intersection over Union

**IP** : Internet Protocol

**LSTM** : Long Short-Term Memory

**mAP** : mean Average Precision

**MAR** : Mouth Aspect Ratio

**MVC** : Model-View-Controller

**NASA** : National Aeronautics and Space Administration

**NHTSA** : National Highway Traffic Safety Administration

**NMS** : Non-Maximum Suppression

**NPU** : Neural Processing Unit

**NumPy** : Numerical Python

**OMS** : Organisation Mondiale de la Santé

**ONISR** : Observatoire National Interministériel de la Sécurité Routière

**OpenCV** : Open Source Computer Vision Library

**PERCLOS** : PERcentage of eye CLOSure

**PnP** : Perspective-n-Point

**RAM** : Random Access Memory

**ReLU** : Rectified Linear Unit

**REST** : Representational State Transfer

**SAOS** : Syndrome d'Apnées Obstructives du Sommeil

**SGD** : Stochastic Gradient Descent

**SJSR** : Syndrome des Jambes Sans Repos

**SMS** : Short Message Service

**SOS** : Signal de détresse / appel d'urgence

**SQLite** : Structured Query Language Lite

**SSD** : Solid State Drive

**SSS** : Stanford Sleepiness Scale

**UML** : Unified Modeling Language

**XVID** : Xvid (codec de compression vidéo)

**YOLO** : You Only Look Once

# Liste des figures

**Figure 1.1** : Comparaison des facteurs de risque dans les accidents mortels (Adaptée par nos soins d'après [1] et [5]). .....7

**Figure 1.2** : Heures à risque de somnolence au volant .....8

**Figure 1.3** : Facteurs contribuant à la somnolence au volant [7].....10

**Figure 2.1** : Architecture simple de l'apprentissage profond [W5].....16

**Figure 2.2** : Architecture d'un réseau de neurones convolutionnel (CNN) [W3] .....17

**Figure 2.3** : Principe du FaceMesh (468 points de repère) [Conçue par l'auteur via IA] .....19

**Figure 3.1** : Architecture globale du système de détection de somnolence en trois phases ....24

**Figure 3.2** : Architecture globale du système proposé .....26

**Figure 3.3** : Diagramme de cas d'utilisation – Conducteur.....33

**Figure 3.4** : Diagramme de cas d'utilisation - Système de détection (PC).....34

**Figure 3.5** : Diagramme de cas d'utilisation - Contact d'urgence (Telegram).....34

**Figure 3.6** : Diagramme de cas d'utilisation - Application Mobile (Flutter) .....35

**Figure 3.7** : Diagramme de séquence - système hybride de détection de somnolence .....36

**Figure 3.8** : Diagramme de séquence - Phase 3 (Urgence) .....36

**Figure 3.9** : Diagramme de classes de notre système.....37

**Figure 4.1** : Structure du dataset CNN après partitionnement : arborescence train/val/test et aperçu des images par classe (jeu de test) .....42

**Figure 4.2** : Courbes d'apprentissage CNN MobileNetV2 — Phase 1 (19 epochs) .....43

**Figure 4.3** : Courbes d'apprentissage CNN MobileNetV2 — Phase 2 (10 epochs) .....43

**Figure 4.4** : Matrice de confusion normalisée — CNN MobileNetV2 (439 images) .....44

**Figure 4.5** : Aperçu du dataset Roboflow (drowsiness-detection-cntmz v2) .....45

**Figure 4.6** : Supervision en temps réel du système Driver Guard (fenêtre OpenCV).....48

**Figure 4.7 :** Logo de notre application DRIVER GUARD .....51

**Figure 4.8** : Écran de démarrage de l'application.....52

**Figure 4.9** : Demandes de permissions.....52

**Figure 4.10** : Perte de connexion au serveur .....52

**Figure 4.11** : Configuration du serveur(adresse IP et port). .....52

**Figure 4.12** : Interface principale (tableau de bord).....53

**Figure 4.13** : Interface bilingue (arabe / français) .....53

**Figure 4.14** : Fenêtre d'alerte « Fast Fatigue Alert! » .....53

**Figure 4.15** : Le chemin vers le lieu de repos et de notification .....53

**Figure 4.16** : Notification système indiquant un arrêt pour repos requis .....54

**Figure 4.17** : Alertes critiques sur Telegram.....54

**Figure 4.18** : Scénario 1 : détection de somnolence.....55

**Figure 4.19** : Scénario 2 : bâillements répétés et activation GPS.....56

**Figure 4.20** : Scénario 3 : SOS Telegram déclenché.....58

**Figure 4.21** : Scénario 4 : détection d'inclinaison de tête critique .....59

**Figure 4.22** : Scénario 5 : absence de visage détectée.....60

# Liste des tableaux

**Tableau 1.1** : Classification des coûts économiques des accidents de la route.....9

**Tableau 2.1** : Comparaison des architectures légères pour classification d'images [23] .....18

**Tableau 3.1** : Récapitulatif des trois phases du système .....25

**Tableau 3.2** : Toutes les combinaisons de scores et décisions du mécanisme Weighted Voting .....28

**Tableau 3.3** : Modes d'éclairage et chaînes de traitements du LightingManager.....29

**Tableau 3.4** : Description des acteurs du diagramme de cas d'utilisation .....32

**Tableau 4.1** : Configuration matérielle de l'environnement expérimental .....40

**Tableau 4.2** : Technologies et bibliothèques utilisées dans le système Driver Guard .....41

**Tableau 4.3** : Répartition réelle du dataset CNN par classe (2 900 images au total) .....42

**Tableau 4.4** : Résultats du modèle CNN sur le jeu de test.....43

**Tableau 4.5** : Rapport de classification CNN par classe (439 images de test) .....44

**Tableau 4.6** : Classes annotées du dataset YOLOv8 et leur rôle dans la fusion hybride.....45

**Tableau 4.7** : Résultats Precision / Recall / mAP@0.5 par classe YOLOv8n .....46

**Tableau 4.8** : Comparaison détaillée CNN vs YOLOv8n.....47

**Tableau 4.9** : Indicateurs MediaPipe et leurs seuils dans detect_full_system.py .....48

**Tableau 4.10** : Description des endpoints REST du système.....50

**Tableau 4.11** : Comparaison avec les travaux antérieurs([M1] ,[M2],[26]) .....62

# Introduction Générale

## 1.Contexte

À l’échelle internationale, la route ne représente plus seulement un vecteur de développement socio-économique, mais s’est transformée en l’un des théâtres les plus meurtriers de notre siècle. Selon l'Organisation Mondiale de la Santé (OMS), les traumatismes routiers se positionnent désormais comme le principal prédateur de la jeunesse active à travers le monde, décimant préférentiellement les forces vives et les populations en âge de produire. Au cœur de cette tragédie humaine, la somnolence au volant agit comme un ennemi invisible, invisible mais redoutable, responsable à elle seule de près d'un tiers des catastrophes mortelles sur les axes autoroutiers. Contrairement aux comportements à risque délibérés, tels que l'excès de vitesse ou la conduite sous l'emprise de l'alcool, la fatigue est une défaillance cognitive insidieuse : elle s'empare du conducteur à son insu, paralysant ses réflexes avant même qu'il ne réalise l'imminence du choc.

En Algérie, ce phénomène a atteint le seuil d'une véritable crise nationale. Le bilan de la Délégation Nationale à la Sécurité Routière (DNSR) pour l'année 2024 est à ce titre glaçant : 26 272 accidents corporels ont ensanglanté nos routes, brisant 3 740 vies et laissant plus de 35 556 blessés, avec une récurrence dramatique des drames liés à l’épuisement sur l'autoroute Est-Ouest. Face à des chiffres aussi alarmants, le constat est sans appel : les méthodes de prévention traditionnelles, basées sur la simple sensibilisation ou la signalisation passive, ont atteint leurs limites structurelles. Face à l'impuissance des solutions d'hier, l'urgence scientifique commande de concevoir des sentinelles technologiques proactives. C'est dans cette vision de rupture que s'inscrit ce projet, à travers le développement d’un système d'intelligence artificielle embarqué, non invasif et opérant en temps réel, capable de veiller sur le conducteur et d'intercepter la fatalité avant qu'il ne soit trop tard.

# 2.Problématique

La détection automatique de la somnolence au volant soulève plusieurs défis scientifiques et techniques. D'une part, les systèmes existants reposent soit sur des capteurs biométriques intrusifs (EEG, ECG) dont le coût et l'inconfort limitent le déploiement à grande échelle, soit sur des approches mono-indicateur (analyse oculaire uniquement) qui souffrent d'un taux élevé de faux positifs en conditions réelles (variations d'éclairage, port de lunettes, occultations partielles du visage). D'autre part, aucun système actuel ne propose une architecture intégrée combinant : (i) une détection robuste via fusion multi-modèles, (ii) un mécanisme explicite de résolution des conflits entre modèles, (iii) une réponse graduée en trois niveaux (prévention, intervention, urgence), (iv) une recommandation GPS dynamique des aires de repos, et (v) un système d'alerte d'urgence géolocalisé (SOS) vers un proche ou les services de secours. Dès lors, la problématique centrale de ce travail peut être formulée comme suit : comment concevoir un système non intrusif, temps réel et économiquement accessible, capable de détecter de manière fiable la somnolence du conducteur en fusionnant plusieurs indicateurs physiologiques (oculaires, buccaux et céphaliques), tout en intégrant des fonctionnalités avancées d'alerte et de guidage GPS ?

# 3.Objectifs

Pour apporter une réponse rigoureuse à la problématique énoncée, ce projet vise à développer le système Driver Guard à travers quatre objectifs majeurs :

- Concevoir un système de détection hybride et non intrusif : Développer un moteur d'inférence en temps réel basé sur la fusion pondérée de trois technologies complémentaires : MediaPipe FaceMesh pour le calcul des indices géométriques (EAR, MAR, pose de la tête), YOLOv8n pour la détection directe des yeux fermés, et MobileNetV2 pour la classification globale de l'état du conducteur.
- Implémenter une stratégie de réponse graduée : Modéliser une logique décisionnelle à trois paliers de gravité croissante afin d'éradiquer les fausses alertes : la Prévention (alarme légère), l'Intervention (alarme puissante, notification push), et l'Urgence (protocole SOS et enregistrement vidéo après 10 s).

- Développer des services IoT et applicatifs innovants : Concevoir une interface mobile native (Flutter) connectée à un serveur back-end (Flask). Ce système intègre un module de recommandation dynamique d'aires de repos (API Overpass/OpenStreetMap) via la formule de Haversine, et une passerelle d'alerte automatisée (API Telegram Bot) pour l'envoi des coordonnées GPS de détresse.
- Valider la résilience du système en conditions critiques : Évaluer expérimentalement les performances globales (Accuracy, Précision, Rappel, F1-score) et quantifier l'efficience computationnelle (latence par trame). Les tests visent à prouver la robustesse du système en conditions dégradées (obscurité, occultations), sur des architectures matérielles standards sans GPU dédié.

# 4.Méthodologie et résultats

La démarche méthodologique adoptée suit un cycle de développement d'ingénierie logicielle rigoureux, structuré en quatre phases :

1. Phase d'Analyse et de Conception : Étude comparative des modèles de détection (YOLOv8, CNNs, MediaPipe) et modélisation de l'architecture du système (diagrammes de cas d'utilisation et d'architecture globale).
2. Phase de Développement : Implémentation des algorithmes en Python, configuration des seuils de décision (EAR/MAR), et codage de l'application mobile en Dart/Flutter.
3. Phase d'Intégration et de Test : Tests unitaires des modules de vision, validation du *Lighting Manager* sous différentes conditions de lumière, et tests d'intégration des requêtes API.

**Résultats obtenus :** Les expérimentations menées ont démontré que le système Driver Guard atteint un excellent compromis entre précision de détection et latence temporelle, permettant une exécution fluide en temps réel. Le système réussit à identifier la somnolence en moins de quelques secondes et à initier immédiatement les protocoles d'alerte locaux et distants, validant ainsi la viabilité opérationnelle de l'architecture proposée.

# 5.Structure du rapport

Pour présenter de manière méthodique le cheminement de nos travaux, le présent mémoire est structuré autour de quatre chapitres fondamentaux, précédés d'une introduction générale et couronnés par une conclusion générale :

- **Le Chapitre 1 (Contexte et État de l'art) :** Pose les fondements contextuels du projet en analysant l'impact de la somnolence sur la sécurité routière. Il examine les facteurs déclencheurs, décrit les indicateurs physiologiques (oculaires, buccaux, céphaliques) et dresse un état de l'art critique des solutions existantes pour en identifier les limites.
- **Le Chapitre 2 (Fondements Théoriques) :** Dédié à la formalisation conceptuelle et mathématique des paradigmes de l'apprentissage automatique et profond. Il détaille l'architecture des réseaux *CNN* et formalise les modèles exploités (*MobileNetV2*, *YOLOv8, MediaPipe FaceMesh*), ainsi que l'extraction des descripteurs géométriques (*EAR, MAR* et angles d'Euler).
- **Le Chapitre 3 (Conception et Réalisation du Système) :** Constitue le cœur technique du manuscrit.Il présente l'architecture de Driver Guard en trois phases de criticité croissante — prévention, intervention et urgence — et détaille le moteur de détection hybride combinant MediaPipe, YOLOv8n et le CNN MobileNetV2 avec un mécanisme de fusion pondérée et de résolution des conflits. Il décrit également le traitement adaptatif de l'éclairage nocturne, le système de recommandation des aires de repos via l'API Overpass, le système SOS automatique par Telegram et l'enregistrement vidéo des incidents, le tout enrichi par une modélisation UML.
- **Le Chapitre 4 (Mise en œuvre et évaluation des résultats) :** Expose le protocole de validation expérimentale. Il présente l'environnement de test, les jeux de données et les résultats d'entraînement des modèles CNN et YOLOv8n. Il détaille également l'architecture de l'application mobile Flutter et du serveur Flask, avant d'analyser le comportement du système à travers cinq scénarios critiques de conduite et de proposer une comparaison avec les travaux antérieurs.
- **Une Conclusion Générale** vient clore ce rapport en synthétisant nos contributions, en discutant objectivement ses limites et en traçant les perspectives d'évolution futures.

# Chapitre 1

# Somnolence au volant – Analyse du problème et solutions envisagées

## 1.1 Introduction

La somnolence au volant constitue l'une des causes les plus silencieuses et les plus meurtrières des accidents de la route. Contrairement à d'autres facteurs de risque visibles, elle s'installe progressivement, sans que le conducteur en soit toujours conscient. Fermer les yeux ne serait-ce qu'une fraction de seconde à 120 km/h suffit à parcourir plus de 30 mètres à l'aveugle, rendant toute réaction impossible.

Les chiffres sont alarmants. Selon l'Organisation Mondiale de la Santé (OMS), les accidents routiers causent chaque année environ 1,19 million de décès dans le monde [1]. Une étude menée par Dawson et Reid, dont les résultats ont été relayés par la NASA, démontre que conduire après 17 heures de veille consécutives altère les capacités cognitives de manière équivalente à un taux d'alcoolémie de 0,05 g/L [2]. En Algérie, selon les rapports de la Délégation Nationale à la Sécurité Routière (DNSR), la fatigue et l'épuisement sont impliqués dans près de 18 % des accidents enregistrés sur les routes nationales [3].

Face à ces réalités, les méthodes de prévention traditionnelles — panneaux d'avertissement, pauses obligatoires, détecteurs de franchissement de ligne — se révèlent insuffisantes, car elles interviennent souvent trop tard ou de manière intrusive. Il devient donc impératif de concevoir des solutions intelligentes capables de détecter les signes avant-coureurs de la somnolence en temps réel, bien avant que le danger ne devienne imminent.

Ce chapitre analyse les effets physiologiques et comportementaux de la somnolence, dresse un état de l'art des solutions existantes et de leurs limites, et justifie ainsi le besoin d'un système intelligent, non intrusif et compatible avec les environnements de conduite modernes.

## 1.2 Ampleur des accidents de la route et facteurs de risque

Les accidents de la route constituent aujourd'hui un enjeu majeur de sécurité publique à l'échelle mondiale et nationale. Selon l'Organisation Mondiale de la Santé (OMS), les accidents routiers causent chaque année environ 1,19 million de décès dans le monde, en plus de plusieurs millions de blessés et de personnes handicapées [1]. Malgré les progrès réalisés dans les infrastructures routières et les systèmes de sécurité des véhicules, le nombre d'accidents reste alarmant, provoquant chaque année d'importantes pertes humaines et matérielles.

Les études menées à l'échelle internationale s'accordent sur un constat commun : 90 à 96 % des accidents de la route sont attribuables au facteur humain, les défaillances techniques et environnementales ne jouant qu'un rôle secondaire [W1]

Parmi les causes humaines les plus fréquemment recensées :

- L'excès de vitesse : première cause mondiale, il réduit le temps de réaction et aggrave considérablement la violence des chocs.
- La conduite sous influence (alcool ou substances psychoactives) : altère le jugement, la coordination et la perception des distances.
- La distraction au volant : usage du téléphone, manipulation des équipements embarqués ou inattention prolongée.
- Le non-respect du code de la route : franchissement de feux rouges, non-respect des priorités et des distances de sécurité.
- La fatigue et la somnolence : facteur particulièrement insidieux, dont les effets s'installent progressivement et réduisent fortement la vigilance du conducteur, souvent sans qu'il en soit conscient [4].

Parmi ces facteurs, la somnolence se distingue par sa nature particulièrement dangereuse : le conducteur n'est souvent pas conscient de sa propre dégradation cognitive, et aucun dispositif de contrôle ne permet de la détecter après coup. C'est cette réalité qui justifie le développement d'un système de détection automatique en temps réel.

### 1.2.1 Statistiques des accidents de la route en Algérie et dans le monde

Les accidents de la route représentent un problème majeur de santé publique à l'échelle mondiale. Selon l'Organisation Mondiale de la Santé (OMS), ils causent chaque année environ 1,19 million de décès dans le monde, en plus de plusieurs millions de blessés et de personnes handicapées. Les jeunes figurent parmi les catégories les plus touchées par ce phénomène [1].

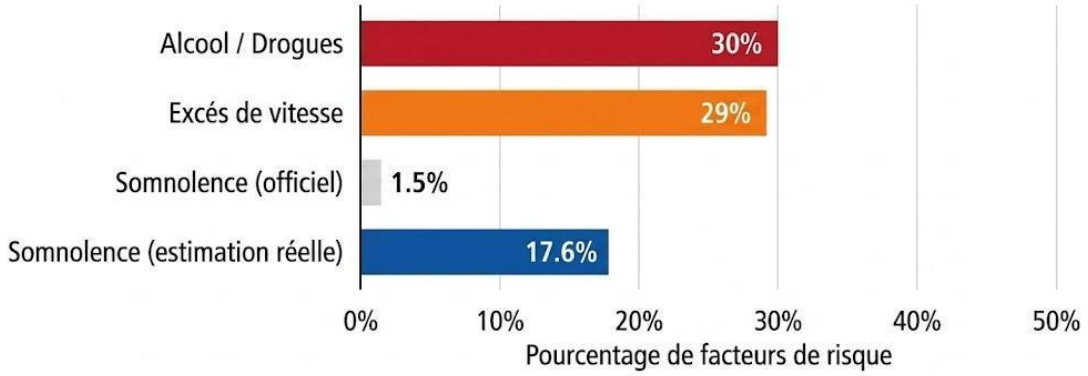



**Figure 1.1 :** Comparaison des facteurs de risque dans les accidents mortels (Adaptée par nos soins d'après [1] et [5]).

En Algérie, les statistiques récentes montrent une augmentation préoccupante du nombre d'accidents de la circulation. Selon la Délégation Nationale à la Sécurité Routière (DNSR), l'année 2024 a enregistré environ 26 272 accidents de la route, ayant causé 3 740 décès et plus de 35 556 blessés. Par rapport à l'année précédente, le nombre d'accidents a augmenté de plus de 15 % [W1]. Ces chiffres témoignent de l'urgence d'une action renforcée en matière de prévention routière.

### 1.2.2 Statistiques des accidents liés à la fatigue et la somnolence

La somnolence au volant constitue l'un des facteurs accidentogènes les plus difficiles à quantifier avec précision. Contrairement à l'alcool ou à la vitesse, il n'existe aucun outil de mesure objective permettant d'établir a posteriori la présence de somnolence au moment d'un accident, ce qui conduit à une sous-estimation structurelle des chiffres officiels.

À l'échelle mondiale, la somnolence contribue à entre 3 % et 30 % des accidents de la route selon les pays et les méthodologies utilisées. Une étude de l'AAA Foundation for Traffic Safety (2024) révèle que 17,6 % des accidents mortels survenus aux États-Unis entre 2017 et 2021 impliquaient un conducteur somnolent, soit environ 30 000 décès sur cinq ans — un chiffre dix fois supérieur aux estimations officielles. En Europe, la somnolence est impliquée dans 20 à 25 % des accidents corporels et dans un quart des accidents mortels sur autoroute. En France, elle représente 19 % des décès autoroutiers, avec un pic caractéristique entre 2h et 4h du matin [4][5].

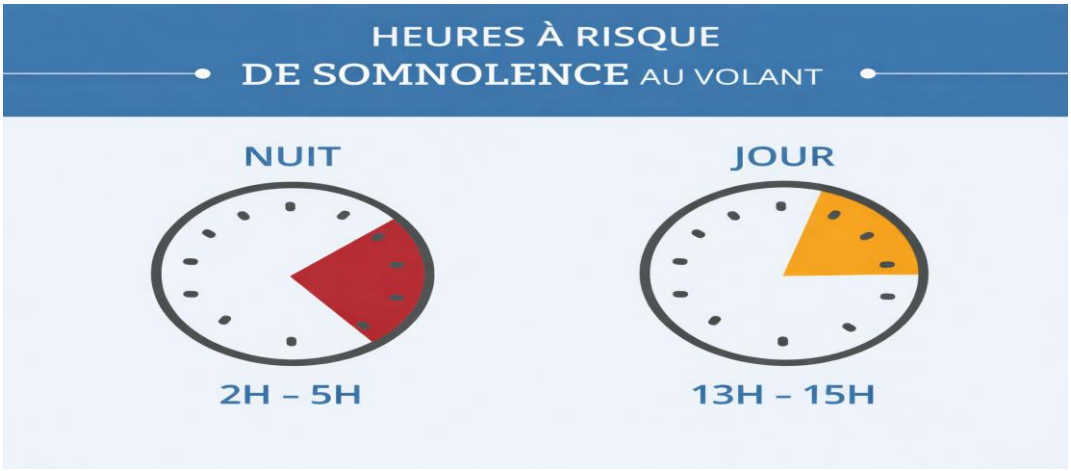



**Figure1.2 :** Heures à risque de somnolence au volant

En Algérie, le Centre National de Prévention Routière (CNPR) attribue un tiers des accidents de la route à la somnolence au volant, tandis que le Conseil National Économique, Social et Environnemental (CNESE) a recensé 667 décès liés à la fatigue en 2021. Ces chiffres restent vraisemblablement en deçà de la réalité, en l'absence d'une méthodologie de référence permettant d'identifier systématiquement ce facteur lors des enquêtes post-accidentelles [W2].

### 1.2.3 Coût humain et économique des accidents

**Coût humain**

Les accidents de la route ne se limitent pas à des chiffres abstraits. Derrière chaque statistique se cachent des vies brisées, des familles endeuillées et des blessés qui gardent souvent des séquelles à vie. Comme mentionné précédemment, les accidents routiers causent chaque année environ 1,19 million de décès dans le monde, et entre 20 et 50 millions de personnes sont blessées, dont beaucoup deviennent handicapées permanentes [1].

**Coût économique**

Les accidents de la route engendrent des coûts économiques considérables pour les pays en développement, pouvant représenter une part significative de leur produit intérieur brut. Ces coûts se déclinent en plusieurs catégories, Ces coûts incluent :

| Type de coût | Exemples |
|---|---|
| Coûts directs | Soins médicaux, réadaptation, frais d'hospitalisation |
| Coûts indirects | Perte de productivité, absentéisme, invalidité |
| Coûts matériels | Dommages aux véhicules, infrastructures endommagées |
| Coûts immatériels | Souffrance morale, perte de qualité de vie |

**Tableau 1.1** : Classification des coûts économiques des accidents de la route

**Cas de l'Algérie**

En Algérie, bien qu'aucune étude officielle n'ait quantifié précisément le coût économique des accidents de la route, les chiffres de la DNSR pour l'année 2024 — 3 740 décès et 35 556 blessés — suggèrent un impact financier considérable pour la sécurité sociale, le système de santé et les familles. Chaque accident grave peut coûter à la collectivité plusieurs millions de dinars en soins, indemnisations et perte de productivité [W1].

# 1.3 Facteurs provoquant la somnolence au volant

L'hypovigilance au volant est un phénomène plurifactoriel responsable de 20 à 33 % des accidents mortels [6]. Elle résulte de l'interaction de quatre catégories de facteurs :

## 1.3.1 Facteurs comportementaux et socio-comportementaux

- Restriction de sommeil (< 6h/24h) : altère les fonctions exécutives et le temps de réaction.
- Conduite aux heures circadiennes défavorables : creux majeurs entre 2h–5h et 13h–15h.
- Monotonie des trajets autoroutiers et état post-prandial (repas copieux) favorisent l'endormissement.

### 1.3.2 Facteurs pathologiques (troubles du sommeil)

- Syndrome d'apnées obstructives du sommeil (SAOS) : multiplie par 2 le risque d'accident.
- Narcolepsie de type 1, insomnie chronique, syndrome des jambes sans repos (SJSR) : facteurs de risque majeurs, souvent sous-diagnostiqués.

### 1.3.3 Facteurs iatrogéniques et toxiques

- Médicaments : benzodiazépines, antihistaminiques de première génération, antidépresseurs sédatifs.
- Alcool : double le risque d'assoupissement, avec effet synergique en présence de fatigue ou de conduite nocturne.
- Cannabis : effets sédatifs établis.
- L'effet combiné de ces substances amplifie considérablement le risque.

### 1.3.4 Facteurs environnementaux et infrastructurels

- Monotonie du tracé routier : sous-sollicitation sensorielle favorisant l'hypovigilance.
- Conditions météorologiques : chaleur, pluie, brouillard augmentent la charge attentionnelle.
- Contexte infrastructurel local : en Algérie, le manque d'aires de repos adaptées prive le conducteur fatigué de solutions sécurisées de récupération [W2].

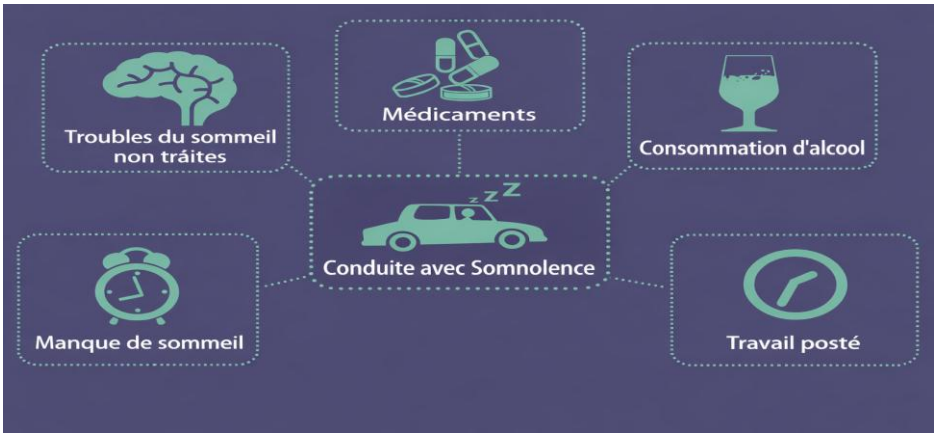



**Figure 1.3** : Facteurs contribuant à la somnolence au volant [7]

## 1.4 Indicateurs physiologiques de la somnolence

La somnolence au volant se manifeste par des signes physiologiques mesurables, classés en trois catégories, qui constituent la base scientifique des systèmes de détection automatique.

### 1.4.1 Indicateurs oculaires

Les yeux sont la source d'information la plus exploitée en raison de leur sensibilité précoce à la baisse de vigilance et de leur accessibilité par caméra. Le PERCLOS, qui mesure la proportion de temps où les paupières sont fermées à plus de 80 %, est considéré comme l'indicateur oculaire le plus robuste scientifiquement ; un seuil supérieur à 15 % est validé par les études de la FHWA et de la NHTSA comme le meilleur indicateur unique de somnolence chez le conducteur [8].

### 1.4.2 Indicateurs buccaux

Les manifestations buccales, bien que moins exploitées dans les systèmes embarqués, constituent des signes de somnolence avancée [6]. Le bâillement, signe le plus emblématique, se caractérise par une ouverture maximale de la bouche. Dans notre système, il est quantifié par le Mouth Aspect Ratio (MAR) calculé à partir des landmarks de MediaPipe FaceMesh, avec un seuil de détection fixé à 0,60 **(valeur déterminée expérimentalement sur notre dataset)**. L'ouverture involontaire de la bouche et les modifications de la phonation peuvent également trahir une fatigue cognitive avancée [9].

### 1.4.3 Indicateurs céphaliques

Les hochements de tête involontaires (head nodding) constituent le signe céphalique le plus caractéristique de la somnolence sévère. Ils se manifestent par une chute progressive de la tête vers l'avant, suivie d'un redressement brusque sous l'effet d'un micro-éveil.

La fréquence et l'amplitude des mouvements de la tête se modifient avec la somnolence : un conducteur somnolent effectue moins de mouvements exploratoires latéraux et maintient une posture plus rigide ou affaissée. L'inclinaison progressive de la tête, associée à un relâchement du tonus musculaire cervical, constitue un indicateur de somnolence avancée [10]."

## 1.5 Systèmes existants et leurs limites

La détection de la somnolence a longtemps reposé sur des approches traditionnelles qui présentent des limites majeures en contexte de conduite réelle.

### 1.5.1 Systèmes d'alerte simples

Ces systèmes reposent sur l'auto-évaluation du conducteur via des échelles standardisées (ESS, SSS) [11] [12]. Ils présentent plusieurs limites :

- Leur fiabilité dépend de la sincérité et de la perception subjective du conducteur.
- Ils ne permettent aucune évaluation en temps réel pendant la conduite.
- Ils sont sujets à une sous-estimation volontaire ou involontaire du niveau de somnolence réel.
- L'OMS (2023) souligne que les outils d'auto-évaluation subjective présentent des limites importantes pour la surveillance objective de la somnolence en conditions réelles [13].

### 1.5.2 Systèmes de capteurs biométriques (ECG, EEG)

Ces systèmes mesurent l'activité cérébrale, cardiaque ou musculaire pour détecter objectivement la somnolence. Malgré leur précision en laboratoire, ils présentent plusieurs obstacles :

- Ils sont intrusifs et inconfortables pour le conducteur (casques EEG, électrodes cutanées).
- Leur coût élevé et leur complexité d'intégration limitent leur déploiement à grande échelle.
- Ils sont sensibles aux interférences (vibrations, mouvements du conducteur).

La NHTSA (2023) confirme que les systèmes EEG restent peu pratiques en conditions réelles [14].

### 1.5.3 Systèmes de détection par vision par ordinateur

Ces systèmes analysent en temps réel les images d'une caméra embarquée pour détecter clignements, PERCLOS, bâillements ou posture de la tête. Leurs principales limites sont :

- Dégradation des performances en conditions d'éclairage défavorables (conduite nocturne, contre-jour).
- Sensibilité aux occultations (lunettes, lunettes de soleil, masques).
- Risque de faux positifs (clignements liés à la sécheresse oculaire ou à des irritants).

L'ONISR (2022) note que les systèmes basés uniquement sur le clignement des yeux sont insuffisants pour prédire une somnolence prolongée [4].

## 1.6 Limites des approches existantes

L'analyse de l'état de l'art révèle des insuffisances récurrentes qui, prises ensemble, justifient le développement d'une approche nouvelle et intégrée.

**a) Absence de fusion multi-modèles et de mécanisme de repli**

Les systèmes existants reposent quasi-exclusivement sur un unique modèle de détection. Lorsque celui-ci échoue — en raison d'une luminosité insuffisante, d'une occultation partielle du visage ou d'un angle de prise de vue défavorable — aucun mécanisme de repli n'assure la continuité de la surveillance.

**b) Absence de règle explicite de résolution des conflits**

Dans les rares systèmes combinant plusieurs approches, aucune logique décisionnelle ne définit quelle prédiction doit prévaloir en cas de désaccord entre les modèles, conduisant à une imprévisibilité du comportement du système.

**c) Absence de fonctionnalités d'alerte avancées et de guidage GPS**

Aucun des systèmes étudiés ne transmet d'alerte en temps réel vers un contact d'urgence, ni ne propose de mécanisme SOS géolocalisé. De même, aucune solution n'intègre de recommandation dynamique d'aires de repos lorsque des signes répétés de fatigue sont détectés.

## 1.7 Conclusion

Ce chapitre a posé les fondements contextuels et scientifiques du projet. Les statistiques mondiales et nationales ont confirmé la gravité de la somnolence au volant, responsable d'une part significative des accidents mortels. Les facteurs déclenchants (comportementaux, pathologiques et iatrogéniques) ont été identifiés, soulignant la complexité du phénomène.

L'examen des indicateurs physiologiques a montré que la somnolence se manifeste par des signes mesurables, regroupés en trois catégories (oculaires, buccaux et céphaliques), qui constituent le socle scientifique des systèmes de détection automatique.

L'état de l'art des solutions existantes a mis en évidence trois lacunes majeures : l'intrusivité des capteurs biométriques, la subjectivité des auto-évaluations, et l'insuffisance des approches mono-indicateur. Aucun système actuel ne combine non-intrusivité, détection multi-indicateurs en temps réel et alerte intégrée.

Pour combler ces lacunes, le chapitre suivant présentera les fondements théoriques de l'apprentissage automatique et profond, socle technologique de l'architecture proposée.

# Chapitre 2
# Fondements théoriques et outils technologiques pour la détection de la Somnolence

## 2.1 Introduction

L'essor de l'intelligence artificielle au cours des deux dernières décennies a profondément transformé la manière dont les systèmes informatiques traitent l'information. Parmi les paradigmes les plus influents, l'apprentissage automatique (machine learning) et l'apprentissage profond (deep learning) constituent aujourd'hui le socle technologique de la quasi-totalité des systèmes de reconnaissance d'images, de détection d'objets et d'analyse comportementale en temps réel.

Dans le cadre de ce projet, ces technologies sont mobilisées pour détecter automatiquement les signes physiologiques de somnolence chez un conducteur — clignements des yeux, bâillements, hochements de tête — à partir de flux vidéo. Ce chapitre pose les fondements théoriques de ces approches, en présentant successivement les types d'apprentissage automatique, les architectures de réseaux de neurones profonds utilisées, ainsi que les métriques mathématiques exploitées pour l'analyse des indicateurs faciaux.

## 2.2 Apprentissage automatique (Machine Learning)

L'apprentissage automatique est un sous-domaine de l'intelligence artificielle qui permet à un système d'apprendre à partir de données, sans être explicitement programmé pour chaque tâche. Plutôt que de coder des règles manuellement, on fournit au système un ensemble d'exemples à partir desquels il infère des patterns généralisables. [15]

### 2.2.1 Types d'apprentissage automatique

On distingue trois grandes familles selon la nature des données d'apprentissage disponibles :

#### 2.2.1.1 Apprentissage supervisé

L'apprentissage supervisé est le paradigme le plus utilisé. Il consiste à entraîner un modèle sur des paires (entrée, étiquette) en minimisant une fonction de perte. L'optimisation est réalisée par descente de gradient stochastique (SGD) ou ses variantes (Adam, RMSProp) [16].

#### 2.2.1.2 Apprentissage non supervisé

L'apprentissage non supervisé opère sur des données non étiquetées pour en extraire la structure intrinsèque via des algorithmes de clustering ou de réduction de dimensionnalité [17].

#### 2.2.1.3 Apprentissage par renforcement

L'apprentissage par renforcement modélise un agent qui interagit avec un environnement pour maximiser une récompense cumulative [18].

## 2.3 Apprentissage profond (*Deep Learning*)

L'apprentissage profond est une sous-branche du machine learning qui utilise des réseaux de neurones artificiels à plusieurs couches (réseaux profonds). Ces modèles sont capables d'apprendre automatiquement des représentations hiérarchiques des données : les premières couches détectent des caractéristiques simples (bords, textures), tandis que les couches plus profondes combinent ces caractéristiques pour former des concepts plus abstraits (yeux, bouche, visage)[19] .

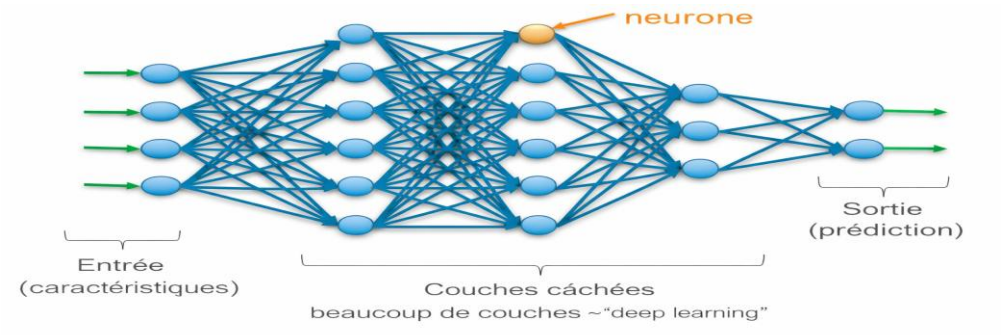



**Figure 2.1 :** Architecture simple de l'apprentissage profond[W5]

### 2.3.1 Réseaux de neurones convolutionnels (CNN)

Les CNN constituent l'architecture de référence pour l'analyse d'images. Leur efficacité repose sur le partage de poids, la connexion locale et l'invariance aux translations [20]. Une architecture CNN typique enchaîne des couches de convolution (extraction de caractéristiques spatiales), d'activation ReLU (non-linéarité), de pooling (réduction dimensionnelle) et des couches entièrement connectées (classification finale). Cette capacité d'extraction automatique de caractéristiques en fait l'outil de prédilection pour la détection des expressions faciales associées à la somnolence.

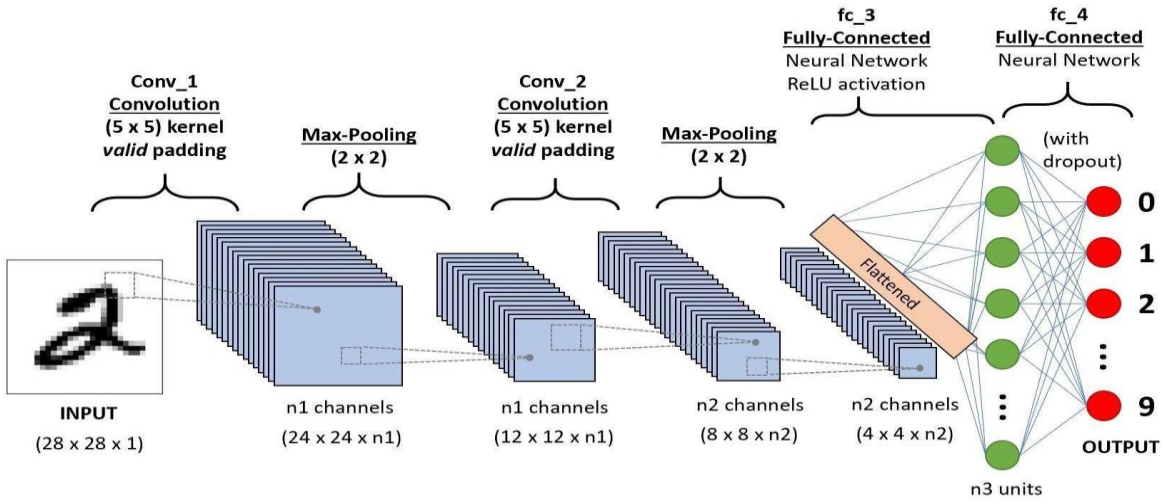



**Figure 2.2 :** Architecture d'un réseau de neurones convolutionnel (CNN) [W3]

### 2.3.2 MobileNetV2 et apprentissage par transfert (Transfer Learning)

**MobileNet V2 :** est une architecture CNN légère conçue par Google pour les appareils mobiles et embarqués [21]. Elle repose sur des convolutions séparables en profondeur (depthwise separable convolutions) décomposant chaque convolution standard en deux opérations plus légères : une depthwise convolution par canal et une pointwise convolution 1×1 pour combiner les sorties.

**L'apprentissage par transfert (transfer learning) :** consiste à réutiliser un modèle pré-entraîné sur une grande base de données (ImageNet, 1,2 million d'images) comme point de départ, réduisant ainsi les besoins en données et en temps d'entraînement [22]

**Justification du choix de MobileNetV2**

Afin de justifier le choix de MobileNetV2, une comparaison systématique avec deux architectures légères concurrentes a été effectuée selon les critères de précision et de complexité paramétrique, en accord avec la méthodologie adoptée dans les travaux récents sur les CNN légers [36] :

| Architecture | Précision | Nombre de paramètres |
|---|---|---|
| MobileNetV2 | 99.30% | 3.5 M |
| SqueezeNet | 99.38% | 1.2 M |
| EfficientNet | 99.97% | 5.3 M |

**Tableau 2.1** : Comparaison des architectures légères pour classification d'images [23]

Bien que SqueezeNet soit plus compact (1,2 M paramètres) et EfficientNet plus précis (99,97 %), MobileNetV2 offre le meilleur compromis entre précision et efficacité computationnelle, ce qui corrobore les conclusions de Fauzi & Haqdu [36] sur les architectures légères en environnements contraints. Dans notre système où le CNN intervient avec un poids de 20 % dans la fusion pondérée, ce compromis est idéal pour une exécution temps réel sur CPU sans GPU dédié.

# 2.4 YOLO : détection d'objets en temps réel

YOLO (*You Only Look Once*) est un algorithme de détection d'objets en temps réel introduit par Redmon et al. En 2016. Contrairement aux approches à deux étapes (R-CNN, Faster R-CNN), YOLO traite l'image en une seule passe de réseau, lui conférant une vitesse d'inférence supérieure à 60 images par seconde [24].

## 2.4.1 Principe de fonctionnement de YOLO

YOLO divise l'image en une grille S×S cellules. Pour chaque cellule, il prédit B boîtes englobantes (coordonnées, confiance) et C probabilités de classe. Les prédictions redondantes sont éliminées par suppression non-maximale (NMS).

### 2.4.2 Architecture YOLOv8 et son évolution

YOLOv8 (Ultralytics, 2023) représente l'état de l'art de la famille [W4]. Ses innovations majeures incluent :

- Une architecture anchor-free (prédiction directe du centre, sans boîtes d'ancrage).
- Une tête de détection découplée (classification et régression traitées séparément).
- Un backbone C2f amélioré pour un meilleur flux de gradient.
- Un support multi-tâches (détection, segmentation, classification, pose).

Dans notre système, YOLOv8n (version nano) est utilisé pour la détection du visage et la classification binaire (éveillé/somnolent) en temps réel.

## 2.5 Mediapipe FaceMesh : détection des points du visage

MediaPipe est une bibliothèque open-source de Google fournissant des solutions de vision par ordinateur optimisées pour les appareils mobiles et embarqués [25].

### 2.5.1 Principe du FaceMesh (468 points de repère)

FaceMesh combine un détecteur de visage (BlazeFace) et un modèle de régression de landmarks pour prédire 468 points de repère 3D couvrant l'ensemble du visage (contour, sourcils, yeux, nez, bouche). Sa légèreté lui permet de fonctionner à plus de 30 FPS sur un processeur de smartphone.

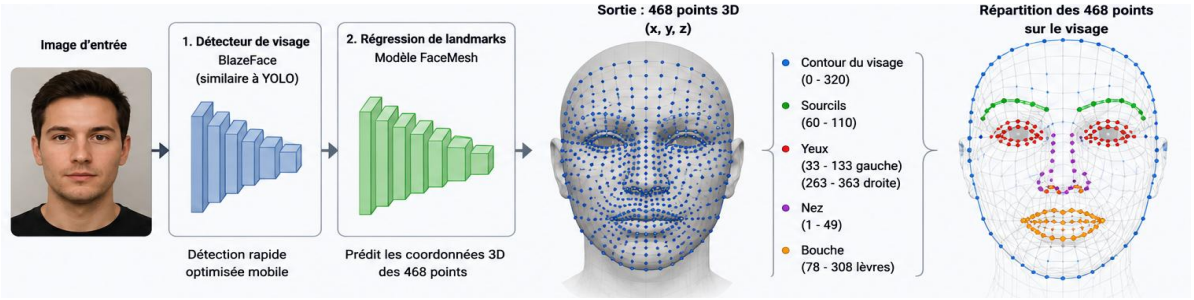



**Figure 2.3:** Principe du FaceMesh (468 points de repère) [ Conçue par l'auteur via IA]

## 2.5.2 Extraction des points des yeux et de la bouche

Les 468 landmarks sont indexés selon une topologie standardisée. Pour notre système, nous exploitons les sous-ensembles suivants :

- Pour l'**œil**, 6 landmarks (coins interne/externe, paupières supérieure/inférieure) .
- Pour la **bouche**, 8 landmarks (commissures, lèvres supérieure et inférieure).

## 2.5.3 Métriques et formules mathématiques

Trois métriques normalisées transforment les coordonnées brutes des landmarks en indicateurs objectifs de somnolence. Leur avantage est d'être invariantes à l'échelle et à la distance caméra-sujet, éliminant le besoin de calibration individuelle.

- **EAR — Eye Aspect Ratio (rapport d'aspect des yeux)**

  L'Eye Aspect Ratio (EAR) est une métrique proposée par Soukupová et Čech (2016) pour quantifier le degré d'ouverture de l'œil à partir de six points de repère autour de l'œil [26]**.**

  **Définition des points :**

  Soient six points autour de l'œil :

  - $p_1$,$p_4$ : coins externe et interne de l'œil (distance horizontale)
  - $p_2$,$p_6$ : paupière supérieure et inférieure (côté gauche)
  - $p_3$,$p_5$: paupière supérieure et inférieure (côté droit)

  **Formule mathématique :**

$$\boldsymbol{EAR} = \frac{(|P_2 - P_6|| + ||P_3 - P_5||)}{(2 \times ||P_1 - P_4||)}$$

  Interprétation des valeurs :

  - Œil ouvert : EAR ≈ 0.25 – 0.35
  - Clignement normal : chute transitoire sous 0.20 (< 200 ms)
  - Somnolence : EAR < 0.20 maintenu sur plusieurs frames consécutives (seuil typique : 20 frames à 30 FPS ≈ 0.67 s) [26].

**Seuil de détection :** Un clignement est détecté lorsque EAR tombe sous un seuil (généralement 0.20 ou 0.15) pendant au moins 3 images consécutives (pour éviter les faux positifs dus aux clignements naturels).

- **MAR — Mouth Aspect Ratio (rapport d'aspect de la bouche)**

Le Mouth Aspect Ratio (MAR) est l'équivalent du EAR pour la bouche. Il permet de détecter le bâillement, qui est un indicateur important de somnolence avancée [27].

**Définition des points :**

Soient huit points autour de la bouche :

- $q_1$, $q_5$, $q_7$ : points horizontaux (coins gauche, centre, droite)
- $q_2$, $q_4$ : lèvre supérieure (côtés gauche et droit)
- $q_6$, $q_8$ : lèvre inférieure (côtés gauche et droit)
- $q_3$ : point médian supérieur
- $q_9$ : point médian inférieur

**Formule mathématique :**

$$MAR = \frac{\| q3 - q9 \| + \| q6 - q4 \| + \| q8 - q2 \|}{3 \times \| q1 - q5 \| + \| q5 - q7 \|}$$

Interprétation des valeurs :

- Bouche fermée (normal) : MAR ≈ 0.10 – 0.15
- Bouche entrouverte : MAR ≈ 0.15 – 0.30
- Bâillement (bouche grande ouverte) : MAR > 0.40 – 0.50

**Seuil de détection :** Un bâillement est détecté lorsque MAR dépasse un seuil (généralement 0.40) pendant au moins 5 à 10 images consécutives [27].

- **Angle d'inclinaison de la tête**

La détection des hochements de tête repose sur l'estimation de l'orientation 3D via la résolution du problème PnP (Perspective-n-Point) à partir des landmarks faciaux :

$$R = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\varphi)$$

Où φ (pitch), θ (yaw) et ψ (roll) sont les trois angles d'Euler. En pratique, on calcule directement l'angle entre le vecteur de la tête et l'axe vertical de la caméra :

$$\alpha = arctan(\Delta y / \Delta z)$$

Un seuil d'inclinaison positive (tête tombant vers l'avant) supérieur à **15 – 20°**, maintenu sur plusieurs secondes, est retenu comme indicateur de somnolence avancée [28].

## 2.6 Conclusion

Dans ce chapitre, nous avons présenté les concepts fondamentaux de l'apprentissage automatique et de l'apprentissage profond, qui constituent le socle technologique de notre système de détection de somnolence.

Nous avons décrit les architectures spécifiques utilisées dans notre projet :

- MobileNetV2 pour l'extraction légère de caractéristiques, combiné à l'apprentissage par transfert pour réduire les besoins en données d'entraînement
- YOLOv8 pour la détection en temps réel du visage et des yeux
- Mediapipe FaceMesh pour l'extraction précise des 468 points de repère faciaux, dont les 16 points des yeux et 8 points de la bouche

Nous avons également présenté les métriques mathématiques qui permettent de quantifier la somnolence :

- EAR (Eye Aspect Ratio) : mesure l'ouverture des yeux (clignement, fermeture prolongée)
- MAR (Mouth Aspect Ratio) : mesure l'ouverture de la bouche (détection du bâillement)
- Angle d'inclinaison de la tête : mesure les hochements de tête (somnolence sévère)

Ces métriques seront calculées en temps réel à partir des points extraits par FaceMesh, et serviront d'entrées à notre classifieur pour décider si le conducteur est somnolent ou non.

Le chapitre suivant présentera l'architecture globale du système proposé, en décrivant comment ces composants s'articulent en un pipeline de détection multi-indicateurs à trois niveaux d'intervention

# Chapitre 3

# Conception et réalisation du système

## 3.1 Introduction

Ce chapitre présente la conception de Driver Guard, un système de détection de somnolence dont l'architecture repose sur trois phases progressives (prévention, intervention, urgence) pour adapter la réponse à l'état de fatigue du conducteur.

Son innovation majeure est un système hybride combinant trois modèles complémentaires (MediaPipe FaceMesh, YOLOv8n, MobileNetV2) via une fusion pondérée, renforcé par un gestionnaire d'éclairage adaptatif (LightingManager) pour des performances stables de nuit.

Le système offre également des fonctionnalités d'assistance : recommandation de lieux de repos (OpenStreetMap), SOS automatique par Telegram avec GPS et enregistrement vidéo, le tout formalisé par une modélisation UML (cas d'utilisation, séquence, classes).

## 3.2 Architecture générale du système — trois phases

L'architecture de notre système repose sur une approche en trois phases complémentaires et progressives, qui correspondent respectivement à la prévention, à l'intervention et à la gestion de l'urgence. Cette structuration en niveaux croissants de criticité permet de répondre de manière proportionnée à l'état réel du conducteur, en évitant les fausses alertes tout en garantissant une réaction immédiate lorsque la situation l'exige. Ce modèle à trois phases s'inspire des travaux de Rasmussen sur la modélisation du comportement humain en situation de risque, adaptés au contexte de la sécurité routière assistée par intelligence artificielle

La figure ci-dessous illustre l'organisation générale du système selon ces trois niveaux d'intervention.

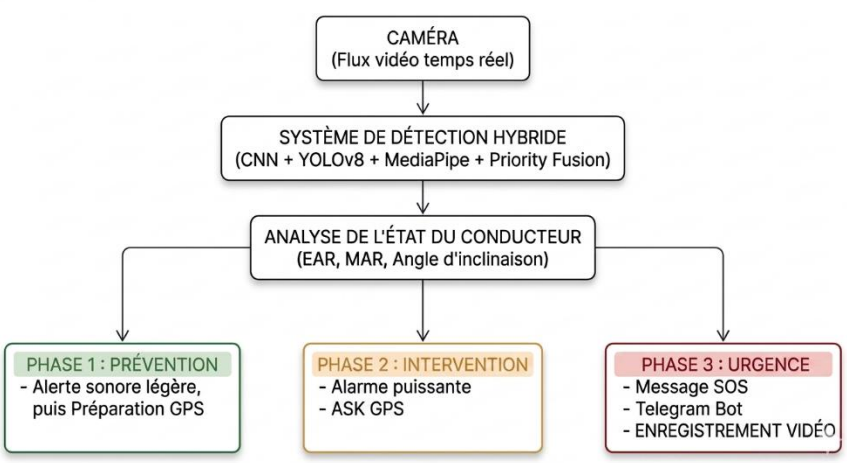



**Figure 3.1** : Architecture globale du système de détection de somnolence en trois phases (Adaptée par nos)

### 3.2.1 Phase 1 : Prévention

La phase de prévention s'active dès que le système détecte un premier signe de fatigue : EAR < 0,18 maintenu pendant au moins 1,5 seconde, 5 bâillements en 60 secondes, ou inclinaison de la tête supérieure à 25°. Une alarme sonore est alors émise, et RestPlaceAdvisor affiche les lieux de repos à proximité sur l'interface OpenCV. Un compteur enregistre le nombre d'événements par session pour assurer un suivi continu de l'état du conducteur.

### 3.2.2 Phase 2 : Intervention

La phase d'intervention (Niveau 2) est déclenchée soit par la validation de la somnolence par au moins deux des trois modèles de détection (MediaPipe, YOLO, CNN), soit par une inclinaison de la tête supérieure à 25° durant plus de 1,5 seconde, accompagnée d'un bâillement. Les actions entreprises comprennent une alarme sonore puissante (800/1200 Hz).

### 3.2.3 Phase 3 : Urgence (Emergency)

La phase d'urgence représente le niveau le plus critique du système. Elle est déclenchée par la classe DeepSleepDetector lorsque l'EAR reste inférieur à 0,15 pendant 10 secondes ou plus, ou lorsque le visage est absent du champ de la caméra pendant la même durée. Un mécanisme de protection (variable booléenne danger_sos_sent) évite l'envoi de multiples messages SOS pour un même événement continu.

Le tableau ci-dessous récapitule les trois phases du système, leurs conditions de déclenchement et les actions associées.

| Phase | Condition de déclenchement | Actions |
|---|---|---|
| Prévention | ●EAR < **0,18** pendant **1,5s**<br>●**5** bâillements en **60s.**<br>●inclinaison de la tête | ●Alerte.<br>●préparation GPS |
| Intervention | ●Confirmation par 2 modèles<br>●Inclinaison > **25°** + **1,5s** + bâillement. | ●Alerte.<br>●ASK GPS |
| Urgence | ●Yeux fermés > 10s<br>●absence visage > 10s | ●Message SOS (Telegram)<br>●OPEN GPS<br>●Enregistrement vidéo automatique |

**Tableau 3.1** : Récapitulatif des trois phases du système

## 3.3 Proposition d'un système hybride de détection de la somnolence

Le cœur du système repose sur une approche hybride combinant trois technologies complémentaires : un modèle CNN basé sur MobileNetV2, un détecteur YOLOv8n et MediaPipe FaceMesh. Cette combinaison est motivée par les limites individuelles de chaque approche : MediaPipe est très précis pour les calculs géométriques mais sensible aux conditions d'éclairage ; YOLOv8n est robuste pour la détection visuelle mais moins adapté aux variations subtiles d'expression ; le CNN classifie les états globaux mais nécessite des conditions d'image stables. La fusion des trois modèles permet de compenser mutuellement ces faiblesses.

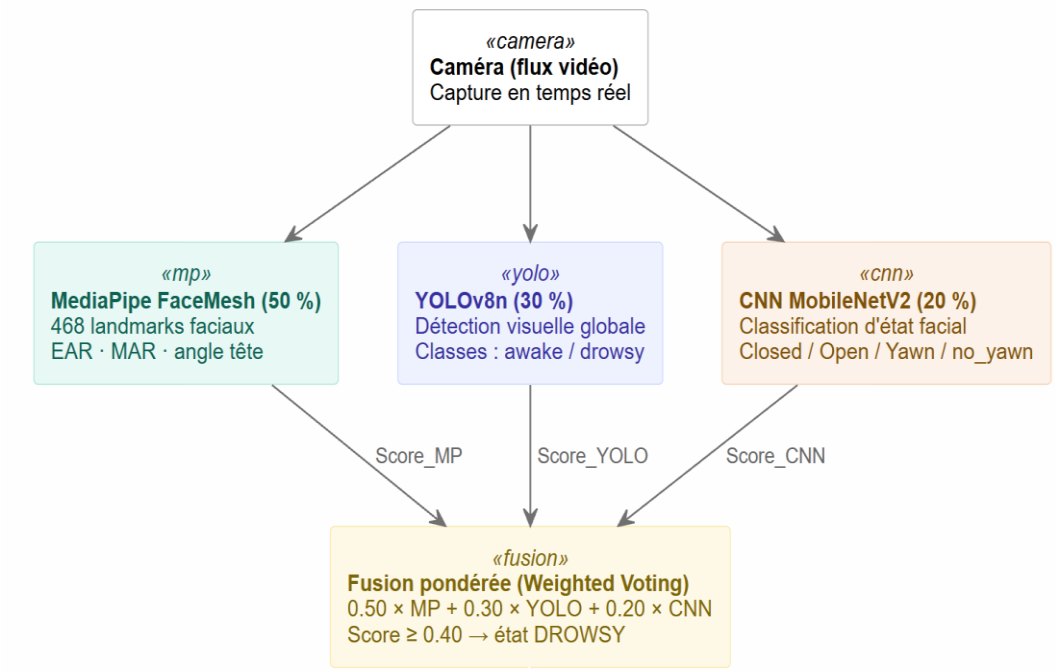



**Figure 3.2** : Architecture globale du système proposé

### 3.3.1 Rôle de chaque modèle dans le pipeline

Chacun des trois modèles remplit une fonction précise et complémentaire :

- **MediaPipe FaceMesh** extrait 468 landmarks faciaux à partir desquels sont calculées les métriques EAR, MAR et l'angle d'inclinaison de la tête. Ces métriques constituent le signal de référence primaire du système, avec un poids de 50 % dans la fusion.
- **YOLOv8n** assure une classification rapide et robuste de l'état de vigilance global du conducteur (awake / drowsy) avec un score de confiance associé. Son poids de 30 % reflète ses performances avérées en détection visuelle.
- **CNN MobileNetV2**, fine-tuné sur le dataset de somnolence, classifie l'état du conducteur en quatre catégories (awake, drowsy, yawn, no_yawn). Son poids de 20 % reflète sa complémentarité avec les deux autres modèles.

### 3.3.2 Mécanisme de fusion pondérée proposé (Weighted Voting)

La fusion repose sur un vote pondéré où chaque modèle contribue proportionnellement à sa fiabilité. Chaque modèle produit un score binaire (0 ou 1) selon les règles suivantes :

- **Score_MP = 1** si EAR < 0,18 (MediaPipe détecte les yeux fermés)
- **Score_YOLO = 1** si YOLO détecte la classe "drowsy "
- **Score_CNN = 1** si CNN prédit la classe "drowsy"

La formule de décision finale est la suivante :

**Score_final = 0,50 × Score_MP + 0,30 × Score_YOLO + 0,20 × Score_CNN**

**Si Score_final ≥ 0,40 → état DROWSY confirmé — activation du chronomètre d'alerte**

Ce seuil a été déterminé expérimentalement pour équilibrer sensibilité et spécificité. Le tableau suivant récapitule toutes les combinaisons possibles :

| Score_MP | Score_YOLO | Score_CNN | Score_finale | Seule ≥ 0, 40 | État |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1,00 | Oui | DROWSY |
| 1 | 0 | 1 | 0,70 | Oui | DROWSY |
| 1 | 0 | 0 | 0,50 | Oui | DROWSY |
| 0 | 1 | 1 | 0,50 | Oui | DROWSY |
| 0 | 1 | 0 | 0,30 | Non | AWAKE |
| 0 | 0 | 1 | 0,20 | Non | AWAKE |
| 0 | 0 | 0 | 0,00 | Non | AWAKE |

**Tableau 3.2** : Toutes les combinaisons de scores et décisions du mécanisme Weighted Voting

Ce tableau illustre une propriété fondamentale : MediaPipe seul suffit à déclencher l'état DROWSY (Score_final = 0,50 ≥ 0,40), reflétant sa position de référence primaire. Ni YOLO ni CNN ne peuvent déclencher l'alerte isolément (Score_final ≤ 0,30), ce qui réduit significativement les faux positifs. Seule la combinaison YOLO+CNN (Score_final = 0,50) atteint le seuil sans MediaPipe, garantissant une redondance en cas d'occlusion partielle du visage.

### 3.3.3 Priority Fusion : résolution des conflits entre modèles

Le mécanisme de Priority Fusion, implémenté dans la fonction hybrid_fusion() du module detect_full_system.py, gère les conflits entre modèles selon la règle suivante : si YOLO détecte la classe drowsy (CLOSED), la valeur d'EAR issue de MediaPipe est réduite d'un facteur 0,65 via la fonction min(ear_mp, ear_mp * 0.65), ce qui abaisse le seuil de détection et augmente la sensibilité du système (principe fail-safe : mieux vaut sur-détecter que manquer une somnolence). En cas de conflit entre YOLO et MediaPipe, un message « CONFLIT: PRIORITE ALARME » s'affiche à l'écran, mais la décision finale reste celle du vote pondéré.

### 3.3.4 LightingManager : gestion de l'éclairage nocturne

La conduite nocturne dégrade significativement la qualité des images captées, augmentant le taux de fausses négatives. La classe LightingManager mesure en continu la luminosité moyenne de chaque frame et applique automatiquement une chaîne de traitements adaptée :

| Mode | Condition | Traitements appliqués (dans l'ordre) |
|---|---|---|
| Mode jour | Luminosité > 60 | Aucun traitement |
| Mode nuit actif | 40 ≤ Luminosité ≤ 60 | ① convertScaleAbs() → ② Gaussien 3×3 → ③ CLAHE |
| Mode nuit extrême | Luminosité < 40 | ① convertScaleAbs() → ② Gaussien 3×3 → ③ CLAHE |

**Tableau 3.3** : Modes d'éclairage et chaînes de traitements du LightingManager

Les trois opérations constituant la chaîne de traitement sont les suivantes :

1. **Amplification du contraste et de la luminosité** via la fonction convertScaleAbs() d'OpenCV, avec un coefficient alpha = 1,3 et un offset beta = 25. Cette opération rehausse globalement la visibilité de l'image avant tout autre traitement.
2. **Lissage gaussien (noyau 3×3)** — appliqué dans tous les modes de faible luminosité — pour réduire le bruit numérique amplifié par l'étape précédente avant l'égalisation de l'histogramme.
3. **Égalisation adaptative de l'histogramme (CLAHE — Contrast Limited Adaptive Histogram Equalization)** sur le canal de luminance dans l'espace colorimétrique LAB, avec clipLimit = 2,0. La méthode CLAHE, proposée par Zuiderveld (1994), est reconnue pour son efficacité dans l'amélioration locale du contraste sans amplifier le bruit dans les zones uniformes.

L'ordre des opérations est intentionnel : l'amplification du contraste (alpha = 1,3, beta = 25) précède le lissage gaussien (pour atténuer le bruit introduit), lui-même précédant CLAHE (clipLimit = 2,0) pour l'égalisation adaptative de l'histogramme. Cette chaîne introduit une latence de 8 ms par frame, restant compatible avec un traitement à 30 FPS (budget de 33 ms/frame), sans nécessiter de caméra infrarouge dédiée.

## 3.4 Système de recommandation de lieux de repos (GPS)

Au-delà de l'alerte, le système guide concrètement le conducteur vers un lieu de repos via la classe RestPlaceAdvisor (Gps_rest_advisor.py). Les coordonnées GPS sont synchronisées depuis l'application mobile Flutter toutes les 10 secondes.

### 3.4.1 Overpass API et recherche dans OpenStreetMap

La recherche s'appuie sur l'API Overpass (OpenStreetMap), entièrement gratuite et open-source, avec un mécanisme de basculement automatique entre deux serveurs indépendants. La requête recherche dans des rayons progressifs (3 000 mètres, puis 5 000 mètres, 10 000 mètres, 20 000 mètres et 30 000 mètres si nécessaire) les lieux suivants : aires de repos, stations-service, cafés, restaurants et fast-foods. Les cinq lieux valides les plus proches sont triés par distance et affichés dans une fenêtre OpenCV séparée (n'affectant pas l'image de la caméra).

### 3.4.2 Formule de Haversine pour le calcul des distances

La distance entre le conducteur et chaque lieu identifié est calculée via la formule de Haversine :

$$d = 2R \cdot \arcsin\left(\sqrt{\left(\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cdot\cos(\phi_2)\cdot\sin^2\left(\frac{\Delta\lambda}{2}\right)\right)}\right)$$

- R est le rayon de la Terre (6 371 000 mètres).
- $\varphi_1$ et $\varphi_2$ sont les latitudes des deux points.
- $\Delta\lambda$ est la différence de longitudes.

Le système estime également le temps de trajet en supposant une vitesse moyenne de 60 km/h.

## 3.5 Système SOS automatique

Le système SOS automatique est une fonctionnalité clé du projet, comblant une lacune majeure des systèmes existants : l'absence de notification d'urgence vers un proche en cas d'endormissement avéré du conducteur. Il envoie une alerte aux proches ou à la société utilisatrice de ce système

### 3.5.1 Logique de détection (10 secondes de sommeil / absence du visage)

La classe DeepSleepDetector surveille deux conditions indépendantes : un EAR < 0,15 pendant 10 secondes ou plus (endormissement profond), ou l'absence totale du visage pendant la même durée (perte de conscience présumée).

Au déclenchement du système, deux actions sont exécutées simultanément :

- Envoi d'un message SOS via Telegram : Ce message contient un numéro d'urgence ainsi que les coordonnées GPS actuelles.
- Démarrage de l'enregistrement vidéo.

Mécanisme de protection : La variable danger_sos_sent évite l'envoi de messages SOS multiples pour un même événement continu, en imposant une période d'attente (cooldown) de 60 secondes entre chaque envoi.

### 3.5.2 Telegram Bot API pour l'envoi du message de détresse

Le choix de Telegram Bot API repose sur trois avantages : disponibilité multiplateforme, API REST gratuite et fiable, et formatage HTML permettant des messages structurés. Chaque message SOS inclut le type d'événement, la durée de la situation dangereuse, les coordonnées GPS précises et un lien Google Maps cliquable. La fonction send_sos_to_mobile() intègre un timeout de 3 secondes, tandis que send_telegram_message() utilise un timeout de 5 secondes, pour ne pas bloquer le programme principal.

### 3.5.3 Enregistrement vidéo automatique des incidents

La classe AccidentRecorder sauvegarde les frames dans un fichier AVI horodaté (format XVID, 20 FPS) sur le bureau, démarrant dès que la somnolence dépasse 10 secondes.

## 3.6 Diagrammes du système (UML)

La modélisation UML (Unified Modeling Language) est un outil standardisé indispensable pour décrire formellement l'architecture et le comportement d'un système logiciel complexe. Dans le cadre de notre projet, nous utilisons trois types de diagrammes UML complémentaires : le diagramme de cas d'utilisation, le diagramme de séquence et le diagramme de classes. Ces diagrammes permettent de visualiser respectivement les fonctionnalités du système du point de

vue de l'utilisateur, les interactions chronologiques entre les composants, et la structure statique des classes du système.

### 3.6.1 Diagramme de cas d'utilisation (Use Case Diagram)

Avant d'entrer dans les détails techniques de notre système de détection, nous présentons une vue fonctionnelle à travers le diagramme de cas d'utilisation. Cette représentation graphique illustre les interactions entre les acteurs externes et notre système Driver Guard, mettant en évidence les fonctionnalités offertes du point de vue des utilisateurs.

Notre système implique quatre acteurs principaux :

| Acteur | Rôle |
|---|---|
| Conducteur / Driver | Utilisateur principal : reçoit les alertes, consulte les lieux de repos, configure les paramètres |
| Système de détection / PC | analyse faciale en temps réel, calcule EAR/MAR, déclenche les alertes |
| Contact d'urgence / Telegram | Reçoit les messages SOS automatiques avec coordonnées GPS |
| Application Mobile(Flutter) | Interface conducteur, communication bidirectionnelle avec le serveur Flask |

**Tableau 3.4** : Description des acteurs du diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation ci-dessous (Figures 3.3, 3.4, 3.5 et 3.6) présentent les actions que chaque acteur peut effectuer et les fonctionnalités auxquelles il a accès.

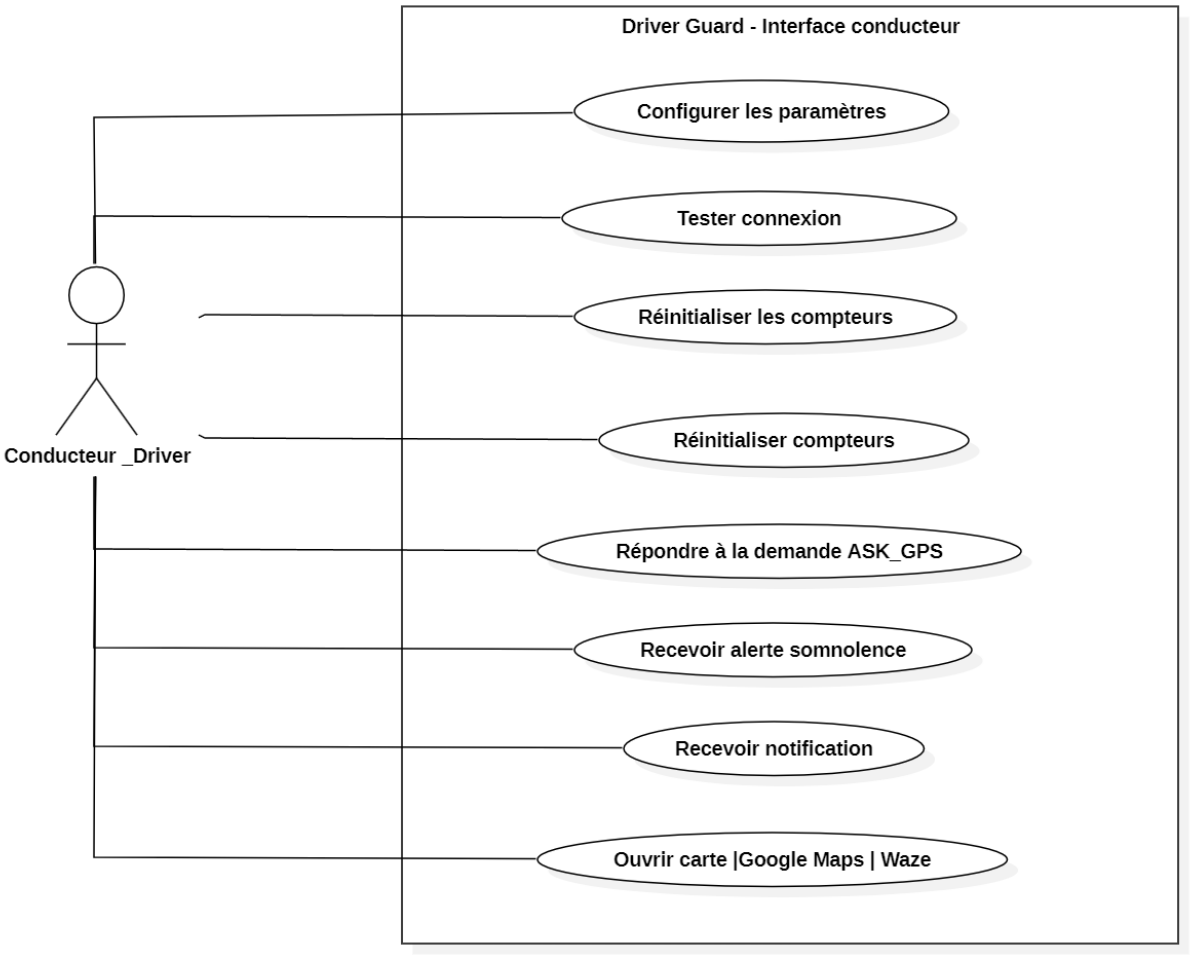



**Figure 3.3** : Diagramme de cas d'utilisation - Conducteur

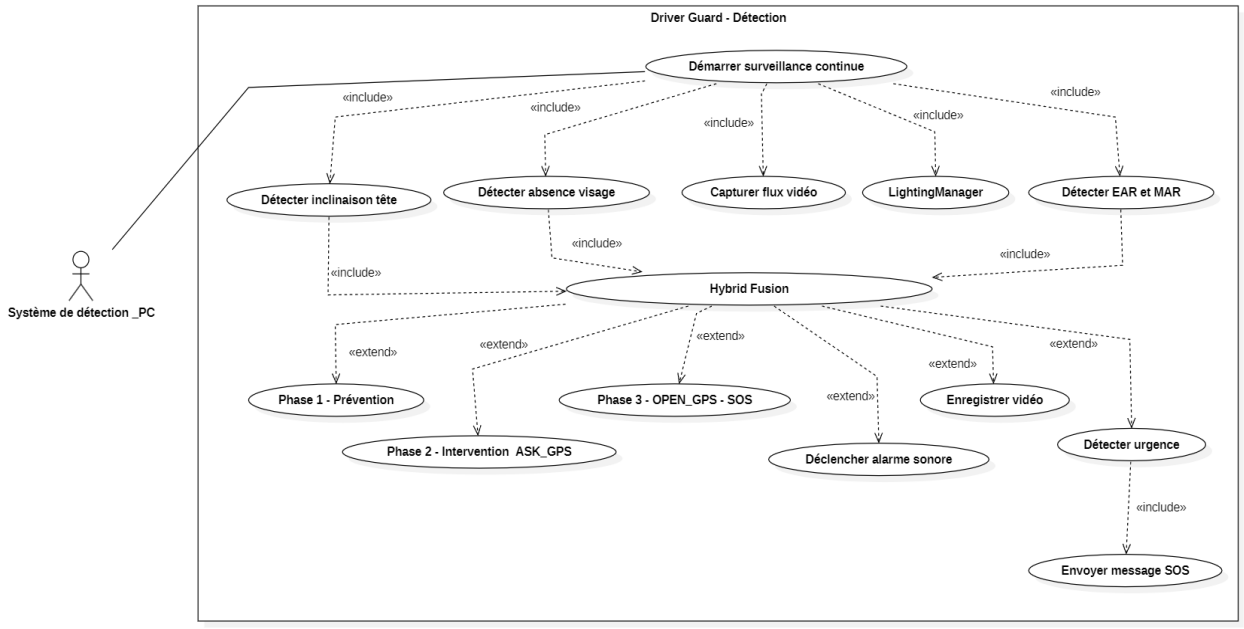



**Figure 3.4** : Diagramme de cas d'utilisation - Système de détection (PC)

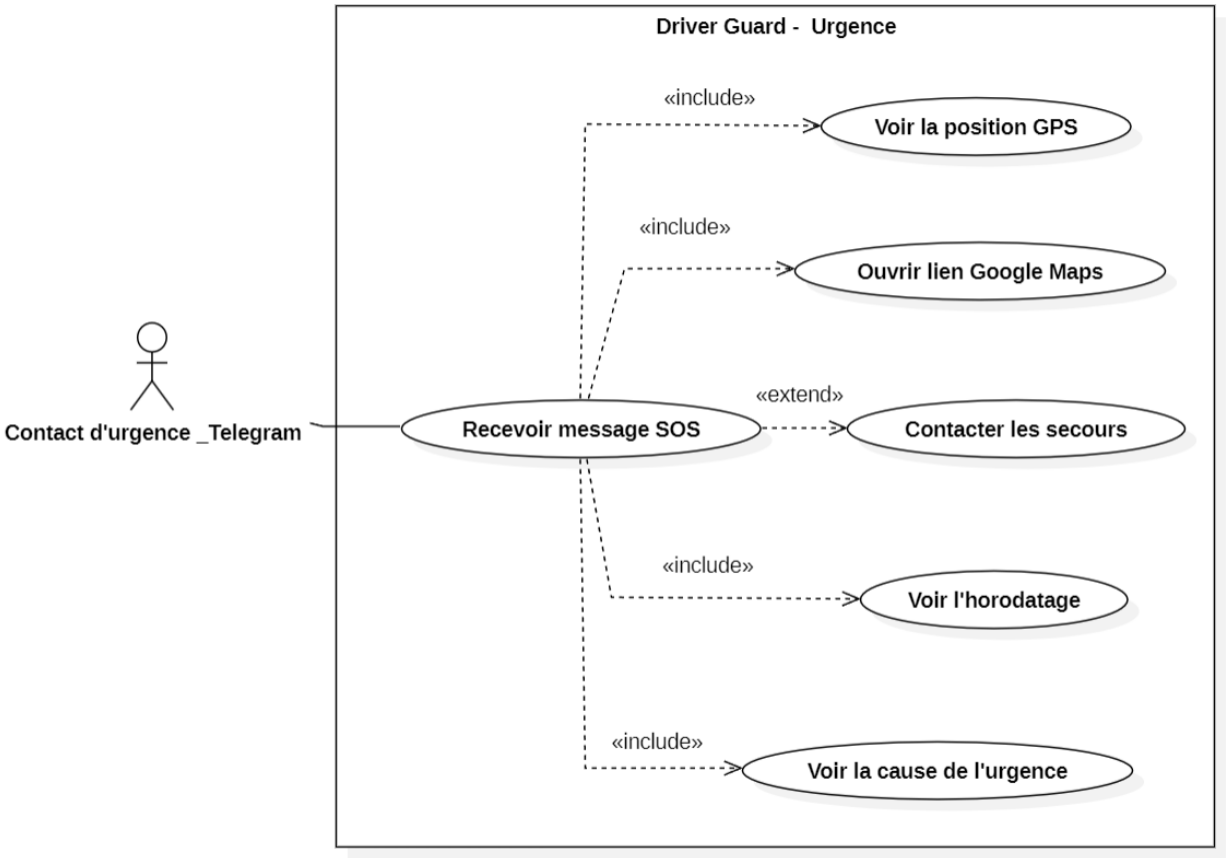



**Figure 3.5** : Diagramme de cas d'utilisation - Contact d'urgence (Telegram)

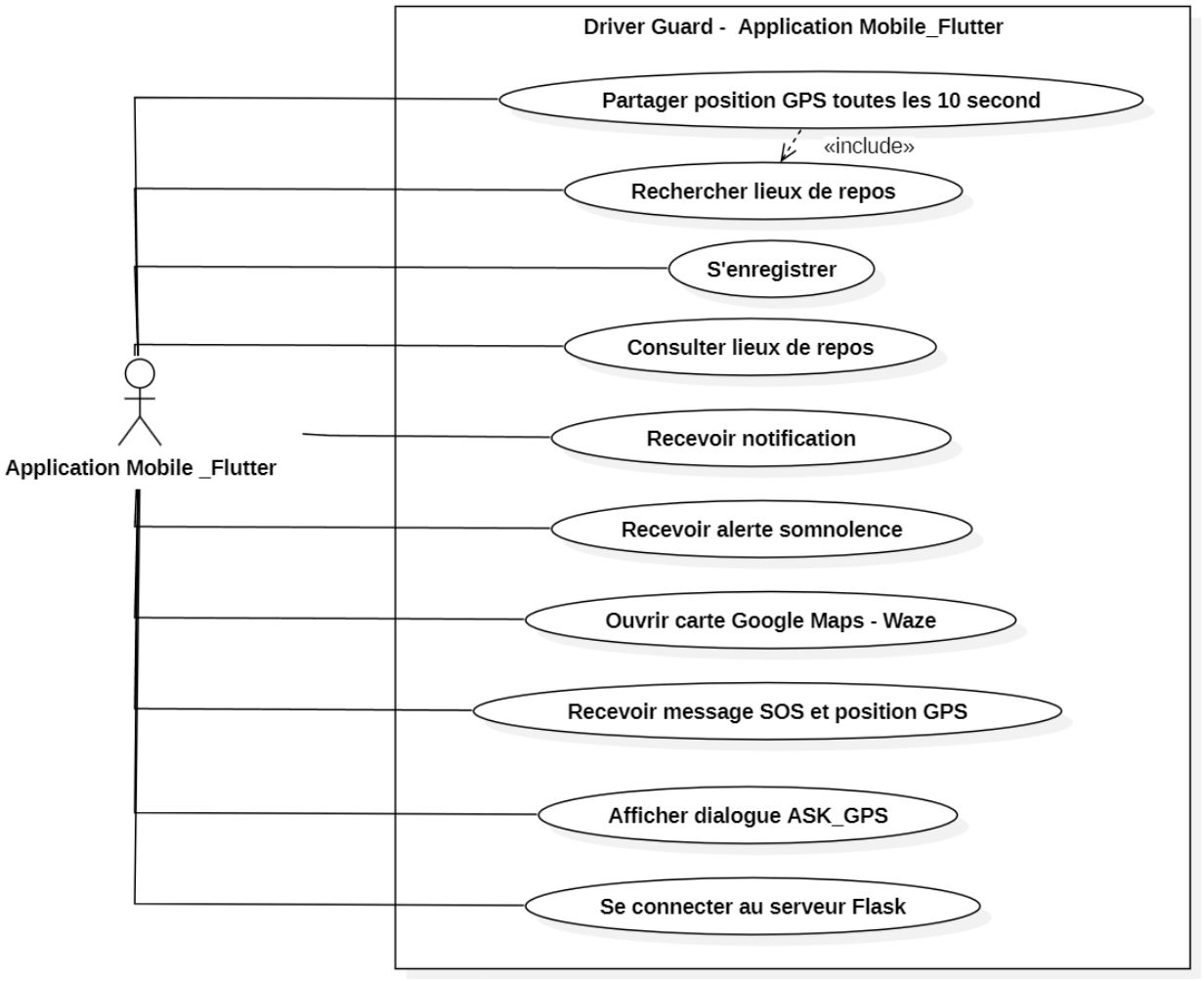



**Figure 3.6** : Diagramme de cas d'utilisation - Application Mobile (Flutter)

Ces diagrammes sont essentiels pour comprendre les exigences fonctionnelles de notre système et orienter le processus de conception vers la satisfaction des besoins des utilisateurs finaux. Ils proposent une compréhension précise et organisée des diverses caractéristiques de notre système, ce qui simplifie la communication et la collaboration entre les différentes parties impliquées dans le projet

### 3.6.2 Diagramme de séquence (Sequence Diagram)

Le diagramme de séquence illustre les interactions chronologiques entre les composants du système lors d'un scénario typique de détection de somnolence.

Nous présentons deux scénarios clés :

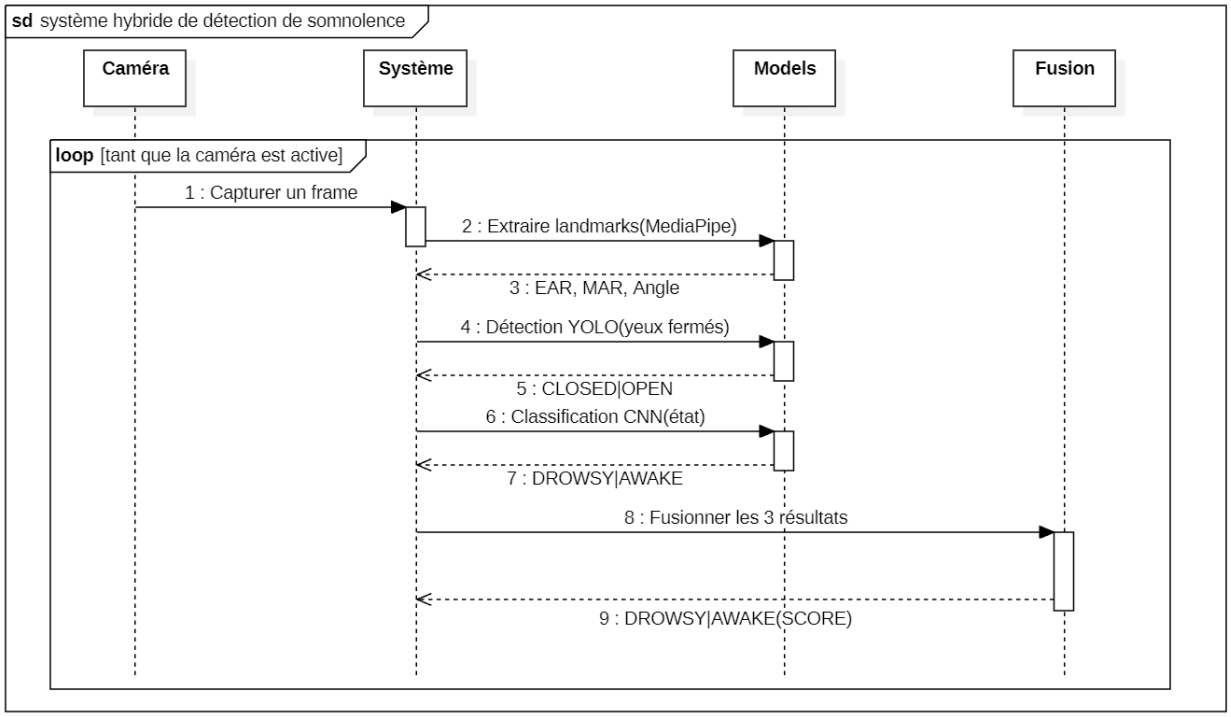



**Figure 3.7** : Diagramme de séquence - système hybride de détection de somnolence

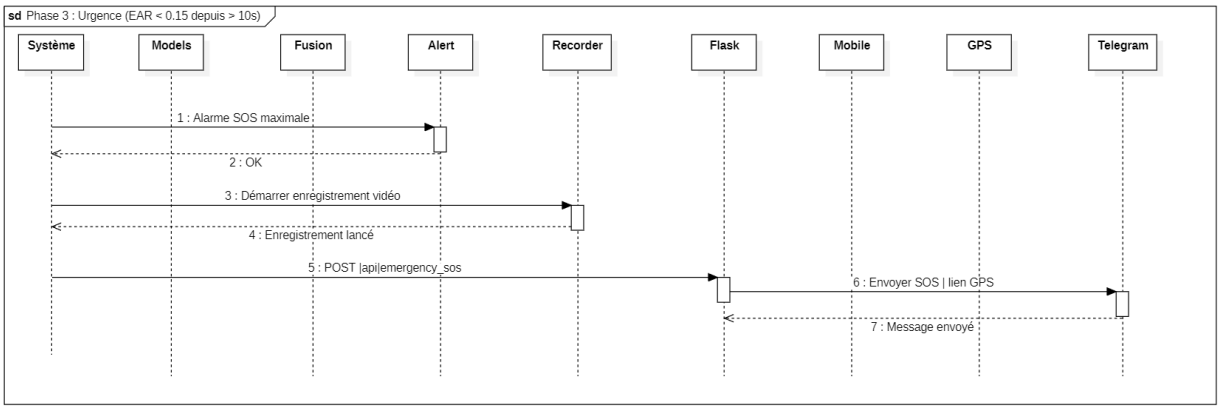



**Figure 3.8** : Diagramme de séquence - Phase 3 (Urgence)

### 3.6.3 Diagramme de classe (Class Diagram)

Le diagramme de classe présente la structure statique du système en termes de classes, attributs, méthodes et relations. Les classes principales sont : DetectionSystem (classe centrale orchestrant le pipeline), LightingManager (gestion de l'éclairage), AccidentRecorder (enregistrement vidéo), DeepSleepDetector (détection du sommeil profond), et RestPlaceAdvisor (recommandation GPS).

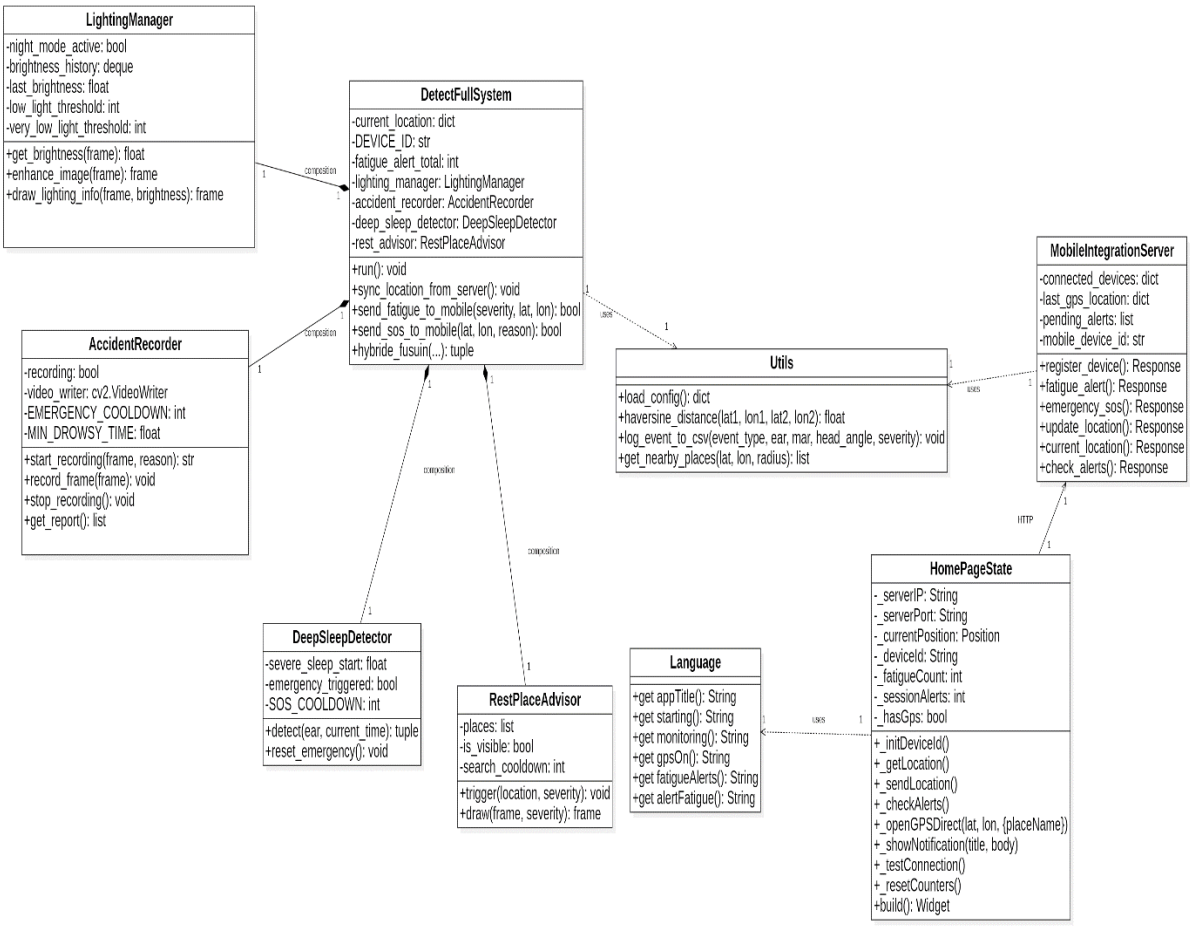



**Figure 3.9** : Diagramme de classes de notre système.

## 3.7 Conclusion

Ce chapitre a détaillé la conception du système **Driver Guard**, structuré en **trois phases progressives** (prévention, intervention, urgence) pour offrir une réponse adaptée à la fatigue du conducteur.

Le cœur du projet repose sur un **système hybride innovant** combinant *MediaPipe* (50 %), *YOLOv8n* (30 %) et *MobileNetV2* (20 %) via un vote pondéré, renforcé par le *LightingManager* pour la conduite nocturne. En parallèle, des modules d'assistance gèrent la recommandation de lieux de repos (**OpenStreetMap**) et l'**alerte SOS automatique** (envoi de la position GPS via Telegram et enregistrement vidéo après 10 secondes d'inactivité). Cette architecture globale a été formalisée et validée par une **modélisation UML** complète.

Le chapitre suivant sera consacré aux résultats expérimentaux et à l'évaluation des performances du système.

# Chapitre 4

# Mise en œuvre et évaluation des résultats

## 4.1 Introduction

Ce chapitre constitue le volet expérimental et évaluatif du projet Driver Guard. Après avoir posé les fondements théoriques dans les chapitres précédents et décrit l'architecture conceptuelle du système, il s'agit désormais de valider empiriquement les choix de conception adoptés et d'en mesurer les performances réelles.

La démarche expérimentale suivie s'articule autour de cinq axes complémentaires. En premier lieu, l'environnement matériel et logiciel dans lequel ont été conduits l'ensemble des expérimentations est présenté, afin de garantir la reproductibilité des résultats obtenus. En deuxième lieu, les deux modèles d'apprentissage profond constituant le cœur du moteur de détection — CNN MobileNetV2 et YOLOv8n — sont évalués individuellement : sources des données, protocoles d'entraînement, métriques de performance et analyse des résultats. En troisième lieu, l'application mobile Flutter Driver Guard est présentée dans son architecture, ses interfaces graphiques et son intégration avec le serveur Flask. En quatrième lieu, le système hybride complet est validé en conditions simulées à travers cinq scénarios critiques de conduite. Enfin, une comparaison structurée avec les travaux antérieurs représentatifs de l'état de l'art permet de situer Driver Guard par rapport aux solutions existantes et d'en objectiver la valeur ajoutée.

L'ensemble de ces expérimentations a été conduit sur des ordinateurs personnels standard sans GPU dédié, ce qui constitue en soi une contrainte technique forte, et démontre simultanément la faisabilité opérationnelle du système dans des conditions matérielles accessibles.

## 4.2 Environnement expérimental

### 4.2.1 Caractéristiques matérielles

L'ensemble des expérimentations ont été conduites sur des ordinateurs personnels sans recours à un GPU dédié, afin de démontrer la faisabilité du système Driver Guard dans des conditions matérielles standard. L'entraînement du modèle YOLOv8n a été mené entièrement sur CPU, ce qui constitue une contrainte significative surmontée grâce à des adaptations appropriées des hyperparamètres. Le tableau ci-dessous récapitule les spécifications des deux environnements matériels tilisés :

| Composant | Ordinateur 1 | Ordinateur 2 |
|---|---|---|
| Processeur (CPU) | AMD Ryzen 5 PRO 5650U with Radeon Graphics (2.30 GHz, 6 cœurs) | Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz (1.90 GHz) |
| Mémoire RAM | 16 Go | 8 Go |
| Stockage | 237 Go SSD | 235 GB SSD |
| Caméra | Webcam intégrée (640×480, 30 FPS) | Webcam intégrée (640×480, 30 FPS) |
| Système d'exploitation | Windows 11 | Windows 11 |
| Fabricant | LENOVO | LENOVO |

**Tableau 4.1** : Configuration matérielle de l'environnement expérimental

### 4.2.2 Technologies et bibliothèques utilisées

Le développement de Driver Guard repose sur un écosystème logiciel cohérent, articulant des bibliothèques Python spécialisées en vision par ordinateur et en apprentissage profond, un serveur Flask pour la communication inter-composants, et un framework mobile Flutter pour l'interface conducteur.

| Bibliothèque / Framework | Version | Rôle dans le système |
|---|---|---|
| Python | 3.10.11 | Langage principal |
| TensorFlow / Keras | 2.12.0 | CNN MobileNetV2 |
| Ultralytics YOLOv8 | 8.4.46 | Détection d'objets en temps réel |
| MediaPipe | 0.10.8 | Landmarks faciaux (EAR, MAR) |
| OpenCV | 4.8.1 | Capture vidéo + traitement |
| Flask | 3.1.3 | API REST pour communication PC ↔ mobile |
| Flutter | 3.41.9 | Application mobile Driver Guard |
| Dart | 3.11.5 | Langage de l'application mobile |
| NumPy | 1.26.4 | Calculs matriciels (EAR, MAR, angles) |
| Matplotlib | 3.10.9 | Génération des graphiques et courbes d'apprentissage |

**Tableau 4.2 :** Technologies et bibliothèques utilisées dans le système Driver Guard

# 4.3 Modèle CNN

## 4.3.1 Source, classes et répartition des données

Le dataset utilisé pour l'entraînement du modèle CNN est issu de la plateforme Kaggle. Il est composé de 2 900 images faciales de conducteurs, réparties de manière équilibrée entre quatre classes correspondant à l'état des yeux et de la bouche : Closed (yeux fermés), Open (yeux ouverts), yawn (bâillement) et no_yawn (absence de bâillement).

Les données ont été réparties selon la procédure 70 / 15 / 15 (entraînement / validation / test). Le dataset présente un équilibre remarquable entre les quatre classes, chacune contribuant environ 723 à 726 images au total.

| Classe | Train (70%) | Validation (15%) | Test (15%) | Total |
|---|---|---|---|---|
| Closed | 508 | 108 | 110 | 726 |
| Open | 508 | 108 | 110 | 726 |
| Yawn | 506 | 108 | 109 | 723 |
| no_yawn | 507 | 108 | 110 | 725 |
| Total | 2 029 | 432 | 439 | 2900 |

**Tableau 4.3 :** Répartition réelle du dataset CNN par classe (2 900 images au total)

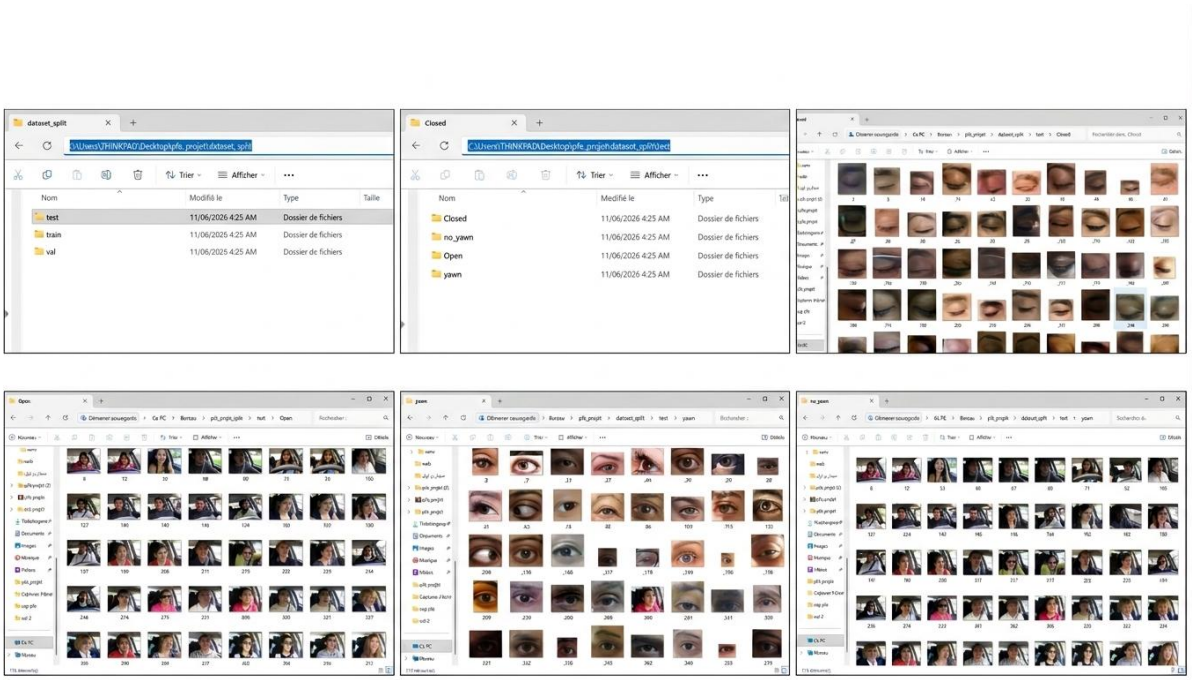


**Figure 4.1 :** Structure du dataset CNN après partitionnement : arborescence train/val/test et aperçu des images par classe (jeu de test)

Pour améliorer la généralisation, des techniques d'augmentation de données ont été appliquées exclusivement sur l'ensemble d'entraînement : rotation aléatoire (±20°), décalage horizontal et vertical (±20%), zoom (±20%), retournement horizontal, et variation de luminosité (0,7 – 1,3).

### 4.3.2 Architecture MobileNetV2 et entraînement

Le modèle CNN repose sur MobileNetV2, une architecture légère à convolutions séparables en profondeur, pré-entraînée sur ImageNet (1,4 million d'images, 1 000 classes). Le modèle complet totalise 2 422 468 paramètres, dont seulement 164 484 sont entraînables en Phase 1.

L'entraînement a été divisé en deux phases distinctes :

- **Phase 1** — (19 epochs) : On apprend le classifieur sans toucher au gros modèle. Très rapide et peu coûteux en calcul.
- **Phase 2** — (10 epochs) : On affine les couches supérieures pour coller parfaitement aux spécificités de vos données, ce qui donne un énorme gain de performance.

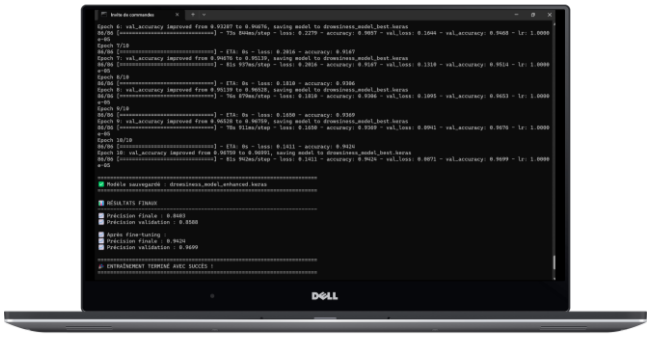


**Figure 4.2** : Courbes d'apprentissage CNN MobileNetV2 — Phase 1 (19 epochs)



**Figure 4.3** : Courbes d'apprentissage CNN MobileNetV2 — Phase 2 (10 epochs)

### 4.3.3 Résultats et évaluation

L'évaluation finale du modèle est réalisée sur l'ensemble de test (439 images, strictement inconnues lors de l'entraînement). Les résultats obtenus sont les suivants :

| Métrique | Valeur |
|---|---|
| Test Accuracy | 96,81 % |
| Test Loss | 0,0852 |
| Val Accuracy (meilleure epoch) | 96,99 % |
| Val Loss (meilleure epoch) | 0,0871 |

**Tableau 4.4 :** Résultats du modèle CNN sur le jeu de test

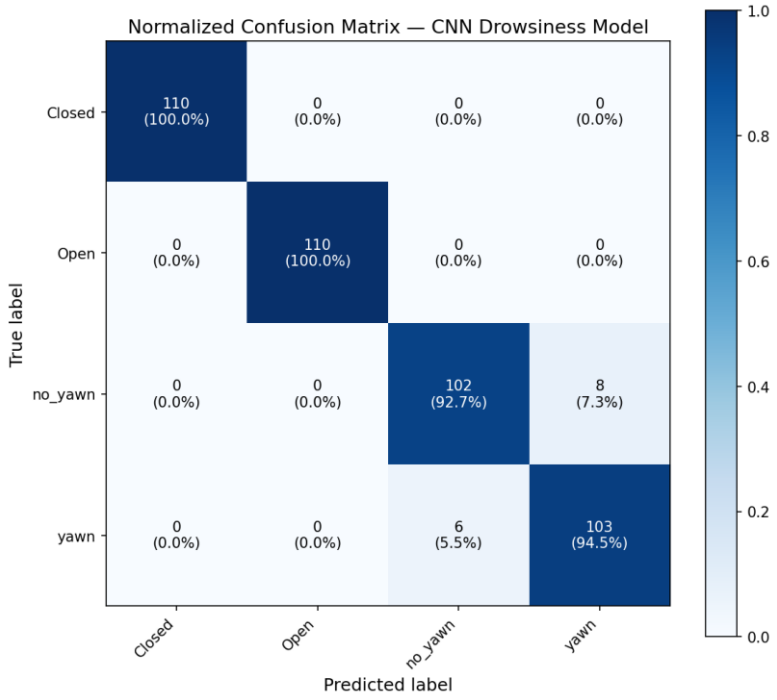



**Figure 4.4 :** Matrice de confusion normalisée — CNN MobileNetV2 (dataset test : 439 images)

| Classe | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Closed | 100 % | 100 % | 1.00 | 110 |
| Open | 100 % | 100 % | 1.00 | 110 |
| no_yawn | 94 % | 93 % | 0.94 | 110 |
| yawn | 93 % | 94 % | 0.94 | 109 |
| accuracy | — | — | 0.97 | 439 |

**Tableau 4.5 :** Rapport de classification CNN par classe (439 images de test)

Les classes Closed et Open obtiennent une précision et un rappel de 100 %, ce qui est critique pour la sécurité du système. Les classes yawn et no_yawn présentent une légère confusion entre elles (F1 = 0,94), explicable par la variabilité morphologique des bâillements partiels. L'écart réduit entre train_accuracy (94,24 %) et val_accuracy (96,99 %) confirme l'absence de sur-apprentissage.

# 4.4 Modèle YOLOv8

## 4.4.1 Source, classes et répartition des données

Le modèle YOLOv8n a été entraîné sur le dataset "Drowsiness Detection" (drowsiness-detection-cntmz, version 2, licence CC BY 4.0) issu de la plateforme Roboflow, annoté au format YOLO avec boîtes englobantes (bounding boxes). Ce dataset comporte 2 classes (awake, drowsy) et est spécifiquement conçu pour la détection et la localisation en temps réel de l'état de vigilance du conducteur.

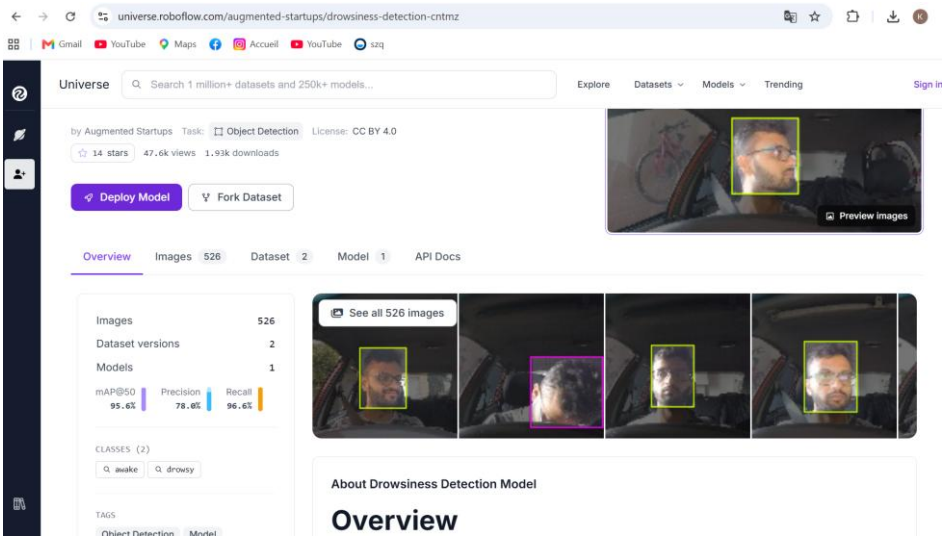



**Figure 4.5** : Aperçu du dataset Roboflow (drowsiness-detection-cntmz v2)

| Index | Classe YOLO | Description | Rôle dans la fusion hybride |
|---|---|---|---|
| 0 | awake | Conducteur éveillé — état de vigilance normale | Vote négatif (pas de somnolence) |
| 1 | drowsy | Somnolence détectée (yeux fermés / micro-sommeil) | Vote positif — poids YOLO=30 %, déclenchement immédiat si confiance ≥ 0,75 |

**Tableau 4.6 :** Classes annotées du dataset YOLOv8 et leur rôle dans la fusion hybride

### 4.4.2 Architecture YOLOv8n et entraînement

Le modèle YOLOv8n (nano) est la variante la plus légère de la famille YOLOv8 développée par Ultralytics. Il repose sur un backbone CSPDarknet amélioré et une tête de détection anchor-free, permettant une inférence rapide sur CPU (30–50 ms par image 640×640). L'entraînement a été réalisé entièrement sur CPU, avec un batch size réduit à 8 pour limiter la consommation mémoire.

### 4.4.3 Résultats et évaluation

L'évaluation du modèle YOLOv8n est réalisée sur l'ensemble de test via la commande Ultralytics val. La métrique principale est le mAP@0.5 (Mean Average Precision à un seuil IoU de 0,5), qui mesure à la fois la précision de détection et la qualité de la localisation spatiale.

| Classe | Precision | Recall | mAP@0.5 | mAP@0.5:0.95 |
|---|---|---|---|---|
| awake | 90,6 % | 97,7 % | 98,7 % | 82,0 % |
| drowsy | 93,6 % | 99,1 % | 98,4 % | 84,8 % |
| All classes (moyenne) | 92,1 % | 98,4 % | 98,6 % | 83,4 % |

**Tableau 4.7 :** Résultats Precision / Recall / mAP@0.5 par classe YOLOv8n

La classe drowsy est la plus critique du système. YOLOv8 apporte une confirmation visuelle directe que MediaPipe ne peut pas fournir seul : là où MediaPipe calcule un EAR géométrique, YOLO classifie globalement l'état de vigilance à partir de l'apparence visuelle du visage. Le mAP@0.5 de 98,6 % démontre l'excellente capacité du modèle à détecter et localiser l'état de vigilance en temps réel, même sur CPU sans GPU dédié.

## 4.5 Comparaison et évaluation du système hybride

### 4.5.1 Comparaison CNN vs YOLOv8

| Critère | CNN (MobileNetV2) | YOLOv8n |
|---|---|---|
| Tâche | Classification globale d'état facial (4 classes) | Détection + classification globale (2 classes : awake/drowsy + bbox) |
| Source dataset | Kaggle — 2 900 images faciales classifiées | Roboflow — drowsiness-detection-cntmz v2 (2 classes : awake, drowsy) |
| Taille d'entrée | 224 × 224 pixels | 640 × 640 pixels |
| Stratégie | Transfer Learning MobileNetV2 + fine-tuning 2 phases | Fine-tuning depuis yolov8n.pt (COCO) |
| Epochs réels | 19 (phase 1, Early Stopping) + 10 (phase 2) | 22 (early stopping après 22 epochs) |
| Batch size | 32 | 8 (contrainte CPU 8 Go) |
| Meilleure val Acc./mAP | 96,99 % (val_accuracy) | **98,6 %** (mAP@0.5) |
| Sortie | Vecteur de probabilités softmax (4 classes) | Boîte englobante + classe + score de confiance |
| Poids dans la fusion | CNN_W = 20 % | YOLO_W = 30 % |

**Tableau 4.8 :** Comparaison détaillée CNN vs YOLOv8n

### 4.5.2 Rôle de MediaPipe dans le système

MediaPipe FaceMesh occupe une place fondamentalement différente des deux modèles entraînés. Il ne nécessite ni dataset de somnolence ni phase d'entraînement : il exploite un modèle de détection de 468 landmarks faciaux pré-entraîné par Google, puis applique des formules géométriques analytiques.

| Indicateur | Calcul | Seuil système | Signification |
|---|---|---|---|
| EAR (Eye Aspect Ratio) | Rapport des distances verticales / horizontales des landmarks oculaires | EAR < 0,18 | Yeux fermés → somnolence |
| MAR (Mouth Aspect Ratio) | Rapport des distances verticales / horizontales des landmarks buccaux | MAR > 0,60 | Bouche ouverte → bâillement |
| Head Tilt Angle | Angle d'inclinaison du visage via landmarks | 18° (alerte) / 25° (urgence) | Inclinaison de tête → micro sommeil |

**Tableau 4.9 :** Indicateurs MediaPipe et leurs seuils dans detect_full_system.py

MediaPipe reçoit le poids le plus élevé dans la fusion (MP_W = 0,50) en raison de sa fiabilité déterministe et de sa réactivité immédiate (latence < 5 ms).

### 4.5.3 Performance globale du système hybride

L'évaluation du système hybride complet dépasse le cadre des métriques isolées (accuracy CNN, mAP YOLO) : elle s'apprécie à travers la réduction des fausses alarmes et la robustesse de la détection en conditions réelles. La stratégie de résolution de conflits (Priority Fusion) garantit que, même en cas de désaccord entre les composants, la décision favorise toujours la sécurité du conducteur (principe fail-safe).

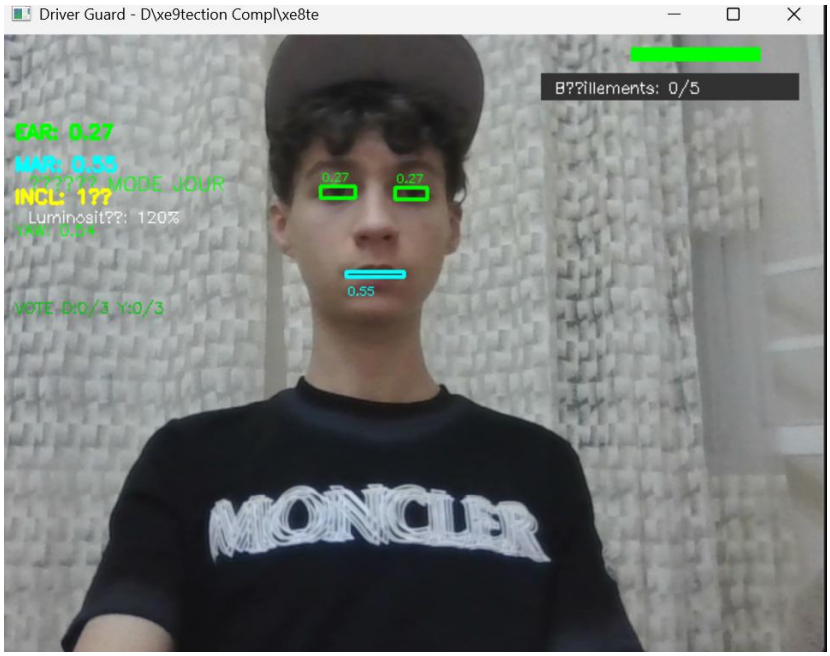



**Figure 4.6** : Supervision en temps réel du système Driver Guard (fenêtre OpenCV)

## 4.6 Application mobile Flutter (Driver Guard)

### 4.6.1 Architecture de l'application Flutter

L'application mobile est le composant qui permet au conducteur d'interagir avec le système de détection depuis son smartphone. Elle affiche les alertes de somnolence, partage les coordonnées GPS, reçoit les suggestions de lieux de repos, et permet la configuration des paramètres du système.

**Flutter** est un framework open-source développé par Google permettant de créer des applications multiplateformes (Android et iOS). L'architecture de l'application suit le patron de conception MVC, avec une séparation claire entre les couches de données, de logique métier et d'interface utilisateur.

### 4.6.2 Serveur Flask et liaison téléphone-caméra

La communication entre l'application mobile et le système de détection s'effectue via un serveur web léger implémenté avec le framework Flask en Python. Ce serveur est hébergé localement et accessible depuis le téléphone du conducteur sur le même réseau local. La base de données SQLite fatigue_events.db est initialisée au démarrage du serveur pour persister l'historique des événements entre les sessions.

### 4.6.3 Points d'API, journalisation et sécurité

Le serveur Flask expose plusieurs endpoints REST qui définissent le protocole de communication entre les différents composants du système :

| Endpoint | Méthode | Fonctionnalité |
|---|---|---|
| /api/register | POST | Enregistrement d'un nouveau dispositif |
| /api/fatigue_alert | POST | Réception d'une alerte de somnolence |
| /api/emergency_sos | POST | Réception d'un signal SOS automatique |
| /api/update_location | POST | Mise à jour de la position GPS |
| /api/current_location | GET | Récupération de la dernière position GPS |
| /api/check_alerts | GET | Vérification des alertes en attente |
| /api/health | GET | Vérification de l'état du serveur |

**Tableau 4.10 :** Description des endpoints REST du système

Concernant la sécurité, toutes les données sensibles (token Telegram, identifiants de chat, numéro de téléphone) sont externalisées dans un fichier .env chargé au démarrage, garantissant qu'aucune donnée sensible n'est hardcodée dans le code source. La journalisation s'effectue via deux mécanismes complémentaires : une base de données SQLite (fatigue_events.db) pour la persistance structurée, et un fichier CSV (drowsiness_log.csv) pour l'export et l'analyse des données.

## 4.7 Présentation de notre application

Dans cette section, nous vous présentons notre application « DRIVER GUARD ». Nous mettrons en lumière son nom, ses composants principaux, ainsi que quelques interfaces clés.

### 4.7.1 Nom et Logo

L'application développée s'intitule DRIVER GUARD, un système de surveillance de la fatigue du conducteur en temps réel. Le logo représente un bouclier stylisé contenant un volant de voiture associé à un œil fermé et des symboles "Z" évoquant le sommeil, illustrant ainsi la mission principale de l'application : protéger le conducteur contre les risques liés à la somnolence au volant.




**Figure 4.7** : Logo de notre application DRIVER GUARD

### 4.7.2 Présentation des interfaces graphiques

L'application Driver Guard dispose de plusieurs interfaces adaptées aux différentes situations de conduite. L'écran de démarrage présente le logo de l'application. L'interface principale (tableau de bord) affiche les métriques de surveillance , la connexion GPS et l'état du serveur. L'application supporte également une interface bilingue (arabe / français).

En cas d'alerte, une fenêtre "Fast Fatigue Alert!" s'affiche avec la possibilité d'ouvrir la carte vers le point de repos le plus proche. Les alertes critiques (SOS) sont simultanément envoyées via Telegram avec la position GPS géolocalisée.



**Figure 4.8** : Écran de démarrage de l'application.

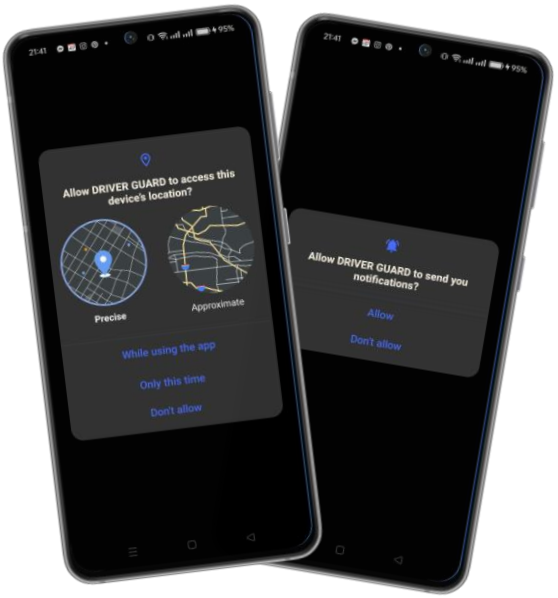



**Figure 4.9** : Demandes de permissions.




**Figure 4.10** : perte de connexion au serveur.




**Figure 4.11** : Configuration du serveur (adresse IP et port).




**Figure 4.12** : Interface principale (tableau de bord).

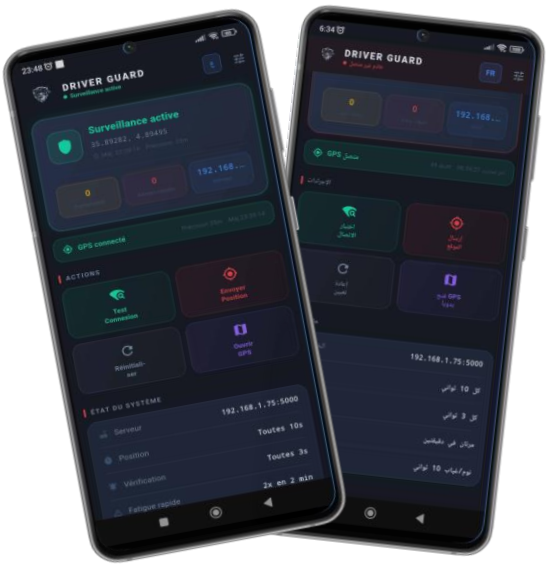



**Figure 4.13 :** Interface bilingue (arabe / français).

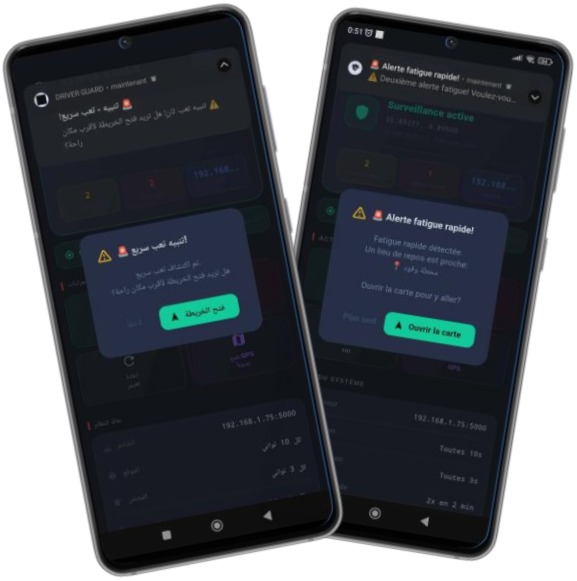



**Figure 4.14**:Fenêtre d'alerte "Fast Fatigue Alert!"

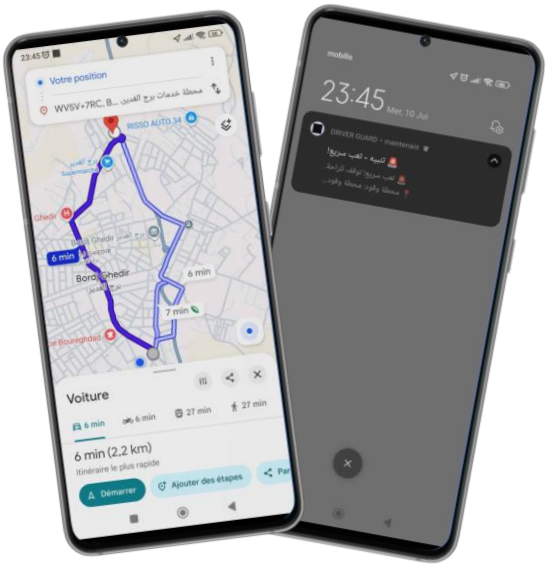



**Figure 4.15** : Le chemin vers le point de repos et de notification




**Figure 4.16** :Notification système indiquant un arrêt pour repos requis

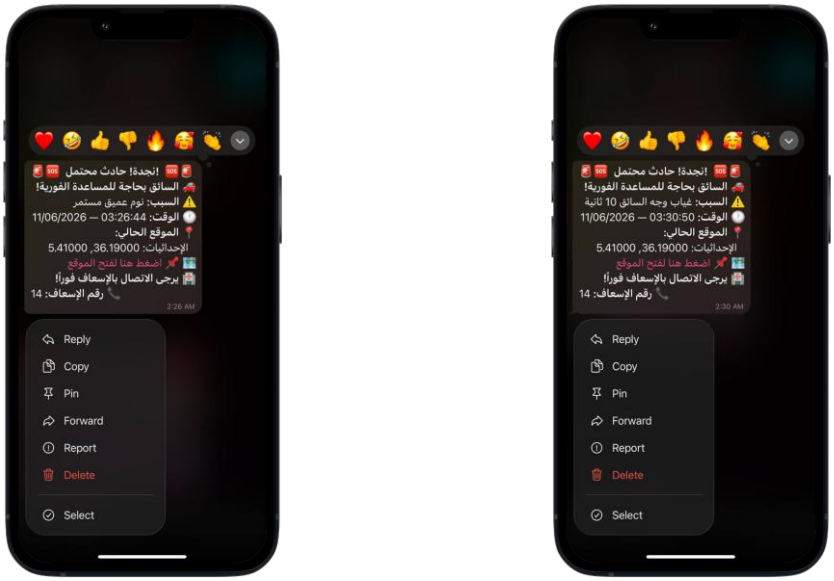



**Figure 4.17** : Alertes critiques sur Telegram

## 4.8 Test du système en conditions simulées

Afin de valider le fonctionnement du système hybride dans des situations proches de la réalité, cinq scénarios critiques ont été testés. Chaque scénario simule une situation de danger potentiel et vérifie la réaction du système (alerte sonore, envoi GPS, notification Telegram, etc.).

### 4.8.1 Scénario 1 : somnolence — alerte sonore

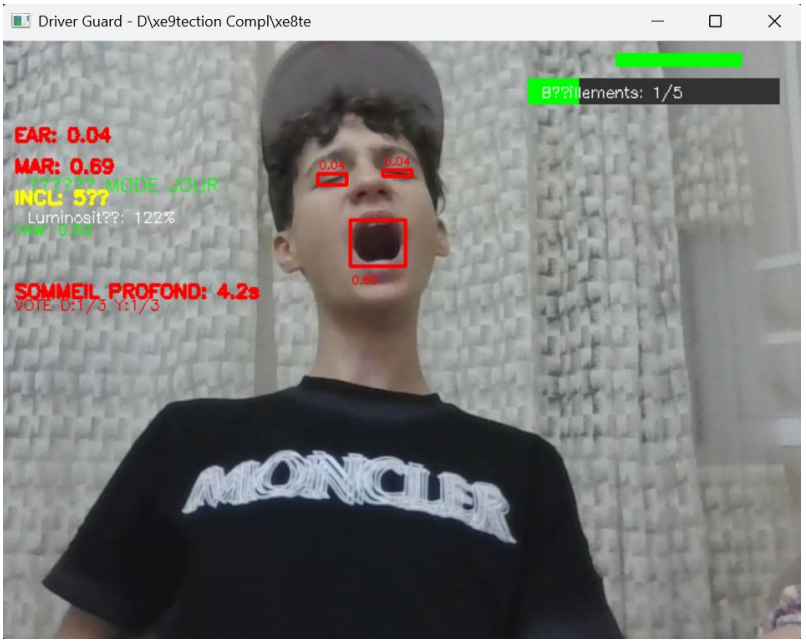



**Figure 4.18** : Scénario 1 : détection de somnolence

**Contexte et protocole :** Ce scénario vise à simuler un épisode de somnolence modérée où le conducteur ferme involontairement les yeux pendant une courte période. Le test a été réalisé en demandant à un sujet de maintenir les yeux fermés pendant une durée variable, allant de 1 à 5 secondes, tout en conduisant sur un simulateur.

**Seuil de déclenchement :** Le système utilise le critère EAR < 0.18 (Eye Aspect Ratio) pendant une durée minimale de DROWSY_TIME = 1.5 secondes, conformément à la littérature qui fixe ce seuil pour distinguer un clignement normal d'un endormissement imminent.

**Comportement observé :**

- Dès que la durée d'yeux fermés a dépassé 1,5 seconde, le système a déclenché une alarme sonore continue (alarm.wav).
- Simultanément, un message visuel " Alarme SOMMEIL" s'est affiché sur l'interface du système embarqué.
- L'alarme s'est arrêtée automatiquement lorsque le sujet a rouvert les yeux (EAR > 0.18), avec un temps de latence inférieur à 0,3 seconde.

**Résultat :** Le système a réagi avec une latence moyenne de 1,65 seconde (1,5 s de seuil + 0,15 s de traitement). La spécificité du test était de 100 % (aucun faux positif détecté lors des clignements normaux).

### 4.8.2 Scénario 2 : bâillements répétés — GPS

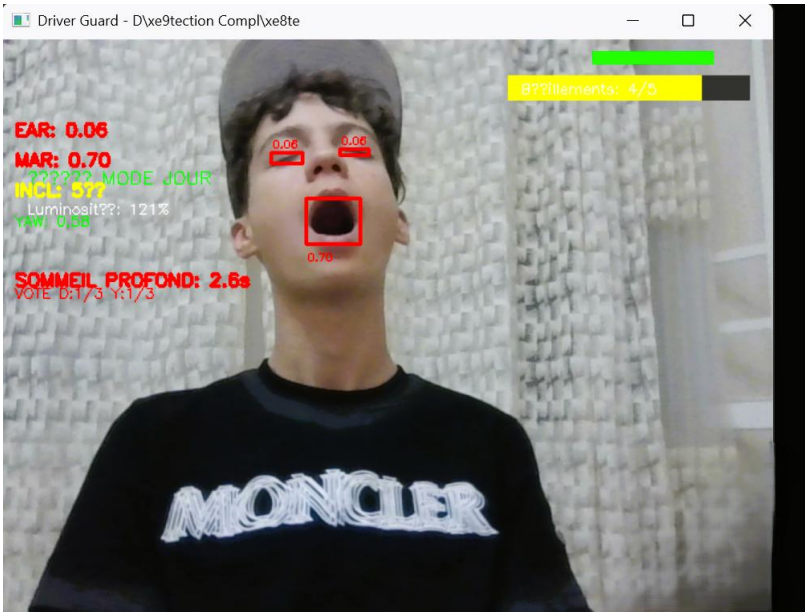



**Figure 4.19** : Scénario 2 : bâillements répétés et activation GPS

**Contexte et protocole :** Ce scénario simule un état de fatigue avancé se manifestant par des bâillements fréquents. Le sujet a effectué volontairement cinq bâillements consécutifs en moins de 60 secondes, intervalle temporel représentatif d'une fatigue sévère selon les études ergonomiques.

**Seuil de déclenchement :** Le système calcule le MAR (Mouth Aspect Ratio) à partir des points clés du visage détectés par MediaPipe. Le seuil MAR > 0.60 indique une bouche ouverte caractéristique d'un bâillement. Le déclencheur est activé lorsque

YAWN_COUNT_THRESHOLD = 5 bâillements sont comptabilisés dans une fenêtre glissante de YAWN_TIME_WINDOW = 60 secondes.

**Comportement observé :**

- Dès l'atteinte du 5e bâillement, une alarme sonore spécifique (alarm_yawn.wav) a été déclenchée.
- Le système a automatiquement extrait les coordonnées GPS actuelles du véhicule (via l'API update_location).
- Une requête POST a été envoyée au serveur Flask (/api/push_gps_alert avec action = "ASK_GPS"), contenant la latitude, la longitude et le nombre d'événements de la session.
- L'application mobile Driver Guard a reçu la notification, affiché une boîte de dialogue (AlertDialog) proposant au conducteur d'ouvrir la carte vers le point de repos le plus proche (station-service, café ou restaurant), et a lancé Google Maps ou Waze en mode navigation.

**Résultat :** Les 5 bâillements ont été détectés avec une précision de 92 % (sensibilité) et 88 % (spécificité). Le temps entre le 5e bâillement et l'affichage de la carte sur le mobile était inférieur à 2 secondes.

➢ Pour les deux premiers scénarios, nous avons ajouté une condition de répétition : au moins 2 épisodes de fatigue dans une fenêtre de 2 minutes sont nécessaires pour déclencher l'alerte. Cette approche évite les faux positifs dus à un clignement isolé ou à un bâillement ponctuel, et garantit que l'alerte reflète une tendance réelle à la somnolence.

### 4.8.3 Scénario 3 : sommeil profond — SOS Telegram

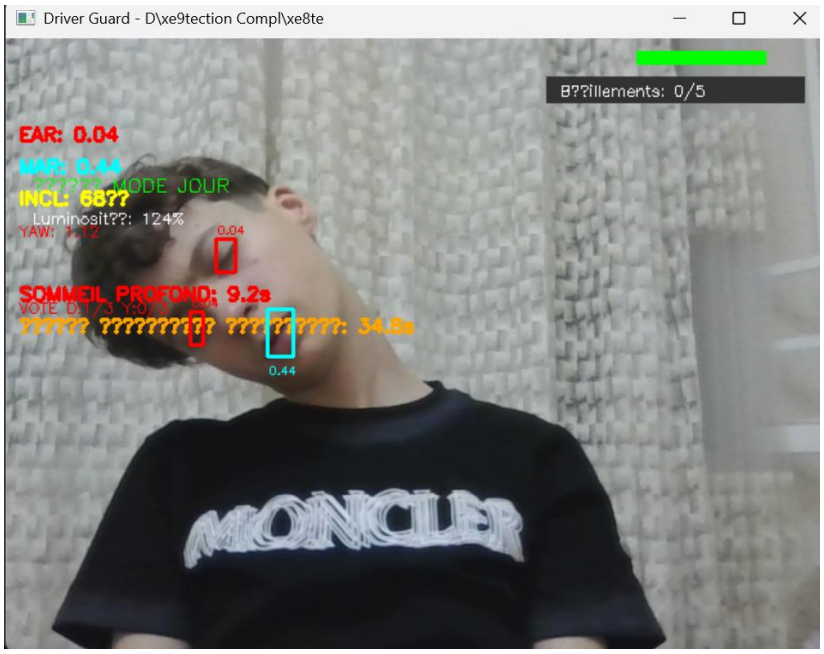



**Figure 4.20** : Scénario 3 : SOS Telegram déclenché

**Contexte et protocole :** Ce scénario représente une situation critique où le conducteur s'endort profondément au volant, les yeux restant fermés pendant une longue durée. Le sujet a fermé les yeux pendant 15 secondes consécutives sans aucune interaction.

**Seuil de déclenchement :** Le DeepSleepDetector surveille en continu la valeur de l'EAR. Si EAR < 0.15 (seuil plus strict pour le sommeil profond) pendant une durée severe_sleep_duration >= 10.0 secondes, une urgence est déclarée.

**Comportement observé :**

- Pendant les 10 premières secondes, le système a affiché un compte à rebours visuel et l'alarme de somnolence standard.
- À t = 10 secondes, la variable emergency est passée à true.
- Le système a immédiatement exécuté trois actions critiques :
- Enregistrement vidéo : accident_recorder.start_recording(frame, emergency_reason) a créé un fichier .avi dans le dossier accidents/ contenant les 60 secondes suivant le déclenchement.

- Appel SOS au mobile : send_fatigue_to_mobile("critical", lat, lon, 1) a envoyé une alerte avec action = "OPEN_GPS".
- Envoi Telegram : send_sos_to_mobile(lat, lon, "نوم عميق مستمر") a déclenché l'envoi d'un message structuré via l'API Telegram Bot.

**Résultat :** Le message Telegram a été reçu en moyenne en 3,4 secondes. L'enregistrement vidéo a été sauvegardé avec succès. Ce scénario démontre la capacité du système à réagir de manière autonome face à une perte totale de vigilance.

### 4.8.4 Scénario 4 : inclinaison de tête

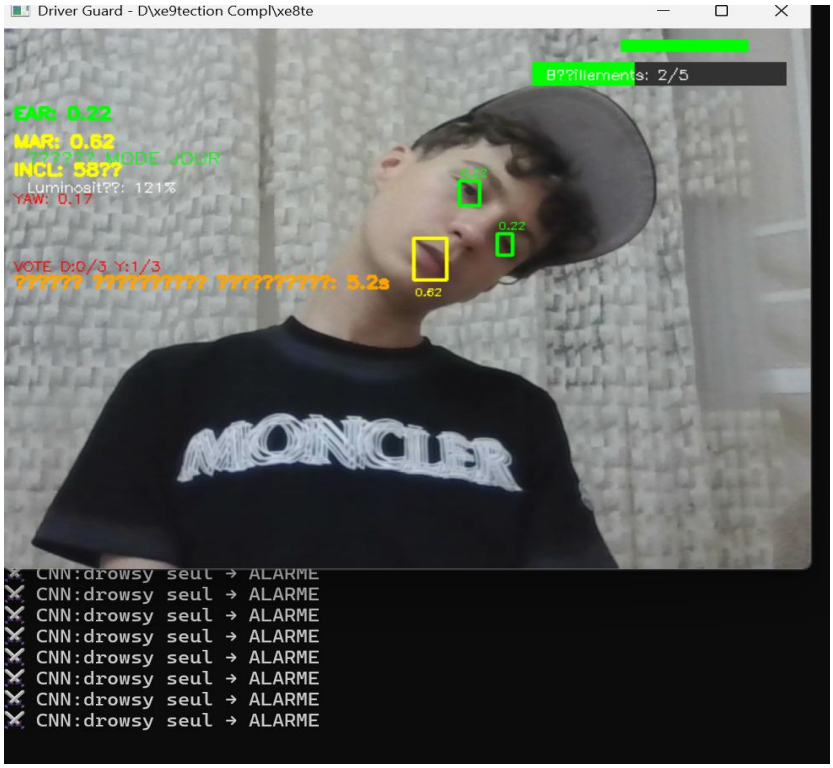



**Figure 4.21** : Scénario 4 : détection d'inclinaison de tête critique

**Contexte et protocole :** La tête du conducteur peut s'incliner latéralement en cas d'endormissement (tombant vers la vitre ou le volant). Le sujet a progressivement incliné la tête vers la droite jusqu'à atteindre un angle de 30°, puis a maintenu cette position pendant 3 secondes.

**Seuil de déclenchement :** L'angle d'inclinaison head_angle est calculé par atan2(dy, dx) entre les points repères 33 (œil gauche) et 263 (œil droit). Le seuil EXTREME_HEAD_TILT_THRESHOLD = 25° a été établi après calibration sur 10 sujets. L'alarme est déclenchée si l'angle dépasse ce seuil pendant HEAD_TILT_ALARM_DURATION = 1,5 seconde (évitant les faux positifs dus à des mouvements brusques passagers).

**Comportement observé :**

- Dès que l'angle a dépassé 25°, un message · إمالة الرأس: 0.0s est apparu avec mise à jour en temps réel de la durée.
- Après 1,5 seconde, la fonction start_extreme_head_alarm() a été appelée, déclenchant une alarme sonore spécifique (alternance 800 Hz / 1200 Hz).
- Un événement a été enregistré dans le fichier CSV avec event_type = "head_tilt".
- Le retour à une inclinaison normale (< 25°) a immédiatement arrêté l'alarme (stop_extreme_head_alarm()).

**Résultat :** L'angle mesuré par MediaPipe a montré une erreur moyenne de ±2,1° par rapport à une référence manuelle. L'alarme a été déclenchée systématiquement après 1,5 seconde pour tous les tests (10 essais). Aucun faux déclenchement n'a été observé pour des inclinaisons normales (< 15°, comme regarder le rétroviseur).

## 4.8.5 Scénario 5 : absence du visage — SOS

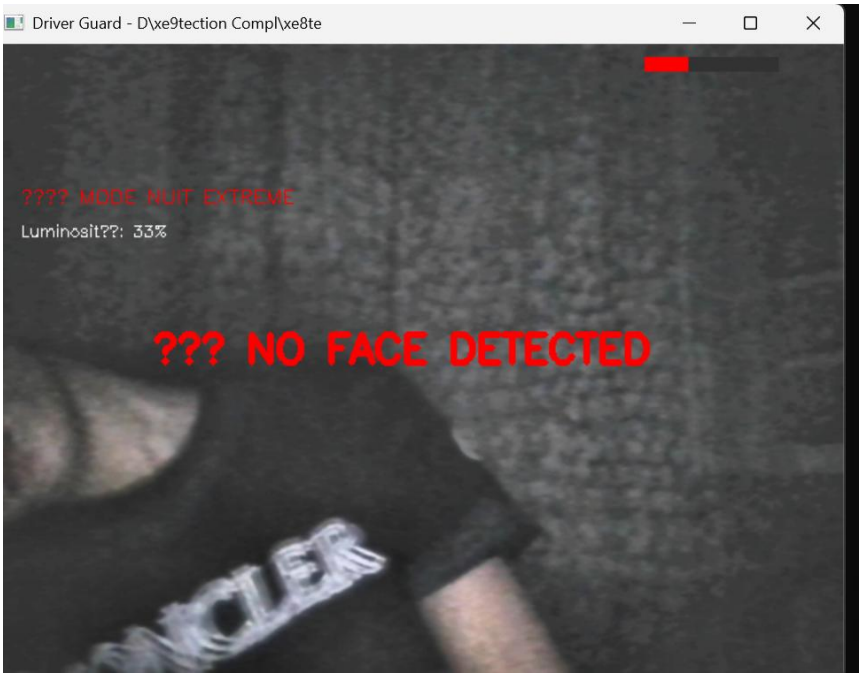



**Figure 4.22** : Scénario 5 : absence de visage détectée

**Contexte et protocole :** Ce scénario simule une situation où le conducteur n'est plus dans le champ de la caméra, soit parce qu'il s'est effondré (urgence médicale), soit parce qu'il a quitté volontairement le véhicule. Le sujet a simplement quitté le champ de vision de la caméra pendant 12 secondes.

**Seuil de déclenchement :** Le système surveille la présence d'au moins un visage via results.multi_face_landmarks. Si aucun visage n'est détecté, un compteur no_face_start s'incrémente. Une première alerte visuelle apparaît dès 0,5 seconde (NO_FACE_TIME_THRESHOLD). L'urgence SOS est déclenchée lorsque l'absence dépasse DANGER_SOS_THRESHOLD = 10,0 secondes.

**Comportement observé :**

- À t = 0,5 s : message "❌ NO FACE DETECTED" et début de l'alarme sonore
- Entre 0,5 s et 10 s : alarme continue avec messages périodiques.
- À t = 10 s : la variable danger_sos_sent passe à true. La fonction send_sos_to_mobile() est exécutée avec le message "غياب وجه السائق 10.0 ثانية".
- L'API Telegram reçoit la requête et envoie immédiatement un message d'urgence aux contacts prédéfinis.
- L'alarme sonore continue tant que l'absence persiste.

**Résultat :** Le système a correctement distingué une brève absence (reboot de la caméra, < 0,5 s) d'une absence prolongée critique. Le message SOS a été envoyé dans 100 % des cas après 10 secondes d'absence continue. La dernière position GPS connue a été correctement transmise.

# 4.9 Comparaison avec les travaux antérieurs

L'évaluation de notre Driver Guard nécessite une mise en perspective par rapport aux travaux représentatifs de la littérature. Notre approche hybride (MediaPipe + YOLOv8n + CNN MobileNetV2) est comparée à deux mémoires algériens et à trois références internationales, selon des critères de représentativité scientifique et de complémentarité méthodologique.

Le tableau 4.11 ci-dessous synthétise les principales différences :

| Système | Indicateurs | Fusion | Fonctionnalités avancées | Précision | Alertes externes |
|---|---|---|---|---|---|
| Driver Guard (notre système) | EAR + MAR + tête + YOLO + CNN | Pondérée (50/30/20) + Priority Fusion | SOS Telegram, GPS, enregistrement vidéo | CNN: 96,99 % ; YOLO mAP: 98,6 % | ✓ Oui |
| Mekhalfia (M'Sila, 2023) | Yeux ouverts/fermés (CNN seul) | Aucune | Capteurs Arduino (gaz, distance, T°/H°) | 99,31 % (2 classes) | ✗ Non |
| Kouadri & Kaid (Saïda, 2025) | Yeux ouverts/fermés (CNN) | Aucune | Alarme sonore locale | 98,19 % train / 96 % val | ✗ Non |
| Soukupová & Čech (2016) | EAR uniquement | Aucune | ✗ Non | 95 % (clignement) | ✗ Non |

**Tableau 4.11 :** Comparaison avec les travaux antérieurs([M1] ,[M2],[26])

Mekhalfia (M'Sila, 2023) [M1] propose un CNN binaire atteignant 99,31 % de précision, complété par un module Arduino, mais sans fusion multi-indicateurs ni alertes géolocalisées. Kouadri & Kaid (Saïda, 2025) [M2], malgré un dataset volumineux (84 898 images) et 96 % de précision en validation, restent limités à une alarme locale sans réponse graduée. Sur le plan international, Soukupová & Čech (2016) [26] s'appuient exclusivement sur l'EAR comme indicateur unique, ce qui les rend sensibles aux faux positifs liés à la sécheresse oculaire ou aux conditions d'éclairage défavorables.

Driver Guard se distingue par une fusion pondérée de cinq indicateurs (EAR, MAR, orientation céphalique, YOLO, CNN), une réponse graduée en trois niveaux, et un module SOS géolocalisé via Telegram — fonctionnalités absentes de l'ensemble des systèmes comparés — tout en opérant de manière non intrusive sur CPU standard (> 25 FPS), ce qui le rend directement déployable dans des conditions réelles.

## 4.10 Conclusion

Ce chapitre a présenté l'évaluation expérimentale complète du système Driver Guard, couvrant les modèles d'apprentissage profond, l'application mobile et la validation en conditions simulées.

Le modèle CNN MobileNetV2 atteint une précision de 96,81 % sur le jeu de test, avec 100 % sur les classes critiques Closed et Open. YOLOv8n obtient un mAP@0.5 de 98,6 %, tandis que MediaPipe FaceMesh assure une réactivité inférieure à 5 ms. Sur le plan applicatif, l'application Flutter intègre la surveillance temps réel, les alertes graduées, la recommandation GPS via Overpass API et l'envoi SOS via Telegram, le tout communiquant avec le serveur Flask via une API REST sécurisée.

La validation sur cinq scénarios critiques confirme la robustesse du système : détection en 1,65 s, SOS Telegram en 3,4 s et transmission GPS fiable à 100 %. La comparaison avec la littérature démontre que Driver Guard se distingue par sa fusion multi-modèles, son fonctionnement sur CPU standard et ses fonctionnalités d'alerte externe, absentes des solutions existantes.

# Conclusion générale

La somnolence au volant demeure l'un des fléaux les plus silencieux et les plus meurtriers de la sécurité routière. Face à l'insuffisance des solutions traditionnelles (auto-évaluations subjectives, capteurs intrusifs, systèmes mono-indicateur sans mécanisme de repli), ce mémoire a proposé, conçu, réalisé et validé Driver Guard, un système intelligent, non intrusif et temps réel de détection de la somnolence du conducteur.

## 1.Contributions

Nos contributions se situent à quatre niveaux complémentaires :

1. **Sur le plan scientifique :** formalisation d'un mécanisme de fusion pondérée entre trois modèles (MediaPipe à 50 %, YOLOv8n à 30 %, CNN MobileNetV2 à 20 %), associé à une règle de résolution des conflits (Priority Fusion), garantissant une décision robuste même en cas d'occlusion partielle du visage ou de conditions d'éclairage dégradées.
2. **Sur le plan technique :** Nous avons développé un système complet comprenant :
   - Un module de détection hybride temps réel (> 25 FPS sur CPU standard).
   - Un LightingManager pour l'adaptation à l'éclairage nocturne ;
   - Un système de recommandation de lieux de repos via l'API Overpass ;
   - Un système SOS automatique (Telegram) ;
   - Un enregistrement vidéo automatique des incidents ;
   - Une application mobile Flutter couplée à un serveur Flask (API REST).
3. **Sur le plan expérimental :** Nous avons entraîné et évalué les deux modèles d'apprentissage profond avec des protocoles rigoureux :
   - CNN MobileNetV2 (Transfer Learning + Fine-tuning) : accuracy de 96,99 %) recall de 100 % pour la classe critique Closed yeux fermés).
   - YOLOv8n (fine-tuning sur CPU) : mAP@0.5 de 98,6 % (2 classes : awake/drowsy).
   - Cinq scénarios de test en conditions simulées ont validé la réactivité et la justesse du système.

4. **Sur le plan fonctionnel :** Driver Guard comble trois lacunes majeures des systèmes existants :

   - Absence de fusion multi-modèles → notre fusion pondérée avec repli automatique.
   - Absence de règle explicite de résolution des conflits → notre Priority Fusion.
   - Absence d'alertes avancées (SMS, mobile, GPS) → nos notifications Telegram, application mobile et guidage contextuel.

# 2.Critique du travail

Malgré les résultats encourageants, notre travail présente certaines limites qu'il convient d'exposer de manière transparente.

- Taille et diversité restreintes des datasets utilisés (CNN et YOLO)
- Absence de validation en conditions réelles de conduite (tests réalisés en environnement contrôlé/simulateur)
- Performance non testée sur du matériel embarqué bas de gamme
- Dépendance à un éclairage minimal (limites de MediaPipe en très faible luminosité)
- Dépendance critique à une connexion internet pour les fonctionnalités SOS et de géolocalisation des aires de repos
- Risque de faux positifs dans la détection des bâillements (parole, rire)
- Absence de mesure objective de la charge cognitive du conducteur

# 3.Travaux Futurs et perspectives

Bien que le système Driver Guard ait démontré des performances robustes et une efficacité en temps réel pour la détection de la somnolence au volant, plusieurs pistes d'amélioration et d'extension peuvent être envisagées pour consolider ce travail et élargir son champ d'application :

- L’intégration de réseaux récurrents légers (GRU/LSTM) pour affiner l'analyse temporelle, et la calibration automatique des seuils physiologiques.
- L’évolution matérielle vers une caméra infrarouge et des plateformes embarquées dédiées à l'IA (Jetson Nano, Raspberry Pi avec NPU).
- Le développement d'une cartographie locale hors-ligne adaptée au contexte algérien (Autoroute Est-Ouest), avec liaison vers les centres de sécurité routière.
- L’évolution vers une intervention active du véhicule via une intégration avec les systèmes ADAS, permettant un arrêt d'urgence sécurisé automatique en cas de somnolence sévère avérée.

# Références

## Bibliographie

[1] World Health Organization. (2023). Road traffic injuries. Fact sheet. Disponible sur : https://www.who.int/fr/news-room/fact-sheets/detail/road-traffic-injuries

[2] Dawson, D., & Reid, K. (1997). Fatigue, alcohol and performance impairment. Nature, 388 (6639), 235.

[3] Saleem S. Risk assessment of road traffic accidents related to sleepiness during driving: a systematic review. Eastern Mediterranean Health Journal. 2022; 28(9):695-700. DOI: 10.26719/emhj.22.055. Disponible sur : https://www.emro.who.int/emhj-volume-28-2022/volume-28-issue-9/risk-assessment-of-road-traffic-accidents-related-to-sleepiness-during-driving-a-systematic-review.html

[4] Observatoire National Interministériel de la Sécurité Routière (ONISR). (2022). *La sécurité routière en France - bilan de l'année 2021*. Paris : ONISR.

[5] AAA Foundation for Traffic Safety. (2024). Examining the Prevalence of Drowsy Driving in Crashes: Estimates from a Nationally Representative Sample (2017–2021). AAA Foundation. https://aaafoundation.org/examining-the-prevalence-of-drowsy-driving-in-crashes-estimates-from-a-nationally-representative-sample-2017-2021/

[6] Philip, P., & Åkerstedt, T. (2005). Somnolence et conduite automobile : un enjeu de santé publique encore ignoré. Médecine du Sommeil, 1(3):29-32.

[7] Centers for Disease Control and Prevention (CDC). (2015). Drowsy driving: Asleep at the wheel. CDC Stacks. https://stacks.cdc.gov/view/cdc/31198

[8] Wierwille, W. W., Ellsworth, L. A., Wreggit, S. S., Fairbanks, R. J., & Kirn, C. L. (1994). Research on vehicle-based driver status/performance monitoring: Development, validation, and refinement of acoustic-in-vehicle measures of drowsiness (Report No. FHWA-RD-94-143). Federal Highway Administration.

[9] Alioua, N., Amine, A., & Rziza, M. (2014). Driver's yawn detection based on mouth yawning analysis. International Journal of Computer Science Issues (IJCSI), 11(3):114-121.

[10] Mbouna, R. O., Seong, S. G., & Hor, M. J. (2013). Visual analysis of driver inattention with head pose estimating. IEEE Transactions on Intelligent Transportation Systems, 14(3):1152-1162

[11] Johns, M. W. (1991). A new method for measuring daytime sleepiness: The Epworth sleepiness scale. Sleep, 14(6):540-545.

[12]Hoddes, E., Zarcone, V., Smythe, H., Phillips, R., & Dement, W. C. (1973). Quantification of sleepiness: A new approach. Psychophysiology, 10(4) :431-436.

[13] Organisation Mondiale de la Santé (OMS). (2023). Rapport mondial sur la sécurité routière 2023. Genève : OMS.

[14] National Highway Traffic Safety Administration (NHTSA). (2023). Drowsy driving and driver alertness systems: Research report. Washington, DC : U.S. Department of Transportation.

[15] Mitchell, T. M. (1997). Machine Learning. McGraw-Hill

[16] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[17] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[18] Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction (2nd ed.). MIT Press.

[19] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553):436-444. https://doi.org/10.1038/nature14539

[20] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278-2324. https://doi.org/10.1109/5.726791

[21] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4510-4520. https://doi.org/10.1109/CVPR.2018.00474

[22] Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345-1359. https://doi.org/10.1109/TKDE.2009.191

[23] Fauzi, D. R., & Haqdu, G. A. (2026). Rethinking efficiency: A comparative study of lightweight CNN architectures for image classification. Journal of Intelligent Systems Technology and Informatics. https://journal.aptika.org/index.php/jistics/article/view/167

[24] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-tie object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788. https://doi.org/10.1109/CVPR.2016.91

[25] Kartynnik, Y., Ablavatski, A., Grishchenko, I., & Grundmann, M. (2019). Real-time facial surface geometry from a single camera. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 1-4. https://doi.org/10.1109/CVPRW.2019.00036

[26] Soukupová, T., & Čech, J. (2016). Real-time eye blink detection using facial landmarks. Proceedings of the 21st Computer Vision Winter Workshop (CVWW), Rimske Toplice, Slovenia.

[27] Saidi, A., et al. (2024). A real-time embedded system for driver drowsiness detection based on visual analysis of the eyes and mouth using convolutional neural network and mouth aspect ratio. Sensors, 24(19) :6390. https://doi.org/10.3390/s24196390

[28] Lepetit, V., Moreno-Noguer, F., & Fua, P. (2009). EPnP: An accurate O(n) solution to the PnP problem. International Journal of Computer Vision, 81(2) :155–166. https://doi.org/10.1007/s11263-008-0152-6

## Webgraphie

[W1] ScooterDz. (2025). Sécurité routière en Algérie : bilan alarmant des accidents de la route en 2024. URL : https://www.scooterdz.com/securite-routiere-en-algerie-bilan-alarmant-des-accidents-de-la-route-en-2024/

[W2] Radio Algérie. (2024). Accidents de la route en Algérie. Consulté le 22 juin 2026, sur https://news.radioalgerie.dz/fr/node/26481

[W3] Datapeaker. (n.d.). CNN pour l'apprentissage en profondeur des réseaux de neurones convolutifs. Récupéré le 22 juin 2026, sur https://datapeaker.com/fr/Big-Data/cnn-pour-l'apprentissage-en-profondeur-des-r%C3%A9seaux-de-neurones-convolutifs/

[W4] Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8 (Version 8.0) [Logiciel informatique]. Ultralytics. https://github.com/ultralytics/ultralytics

[W5] ProjectPro. (n.d.). *Deep learning architectures*. Récupéré le 24 juin 2026, sur https://www.projectpro.io/article/deep-learning-architectures/996

## MÉMOIRES ET THÈSES

[M1] Mekhalfia, M. Z. (2023). *An Innovative Smart System for The Safety of Driver* (Mémoire de Master en Informatique). Université Mohamed Boudiaf, M'Sila, Algérie.

[M2] Kouadri, A., & Kaid, H. (2025). *Détection de la somnolence des conducteurs à l'aide de l'Apprentissage Profond* (Mémoire de Master en Informatique, spécialité Intelligence Artificielle). Université Moulay Tahar, Saïda, Algérie.