

الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université –Ain Temouchent- Belhadj Bouchaib
Faculté des Sciences et de Technologie
Département d'Électronique et des Télécommunications



Projet de Fin d'Etudes
Pour l'obtention du diplôme de Master en :
Domaine : Sciences et Technologie
Filière : Électronique
Spécialité : Instrumentation
Thème

**Etude et Développement d'un Système de Détection de la
Somnolence chez le Conducteur**

Présenté par :

- 1) Melle MILOUD ABID Abir
- 2) Melle BAHAOUS Chaimaa Fatima Zahra

Devant le jury composé de :

Dr MEKAMI Hayat	UAT.B.B (Ain Temouchent)	Présidente
Dr YAGOUB Reda	UAT.B.B (Ain Temouchent)	Examineur
Dr BENTAIEB Samia	UAT.B.B (Ain Temouchent)	Encadrante
Mme BOUTKHIL Malika	UAT.B.B (Ain Temouchent)	Co-encadrante

Année Universitaire 2025/2026

C'est avec une profonde gratitude et une immense émotion que je dédie ce travail de fin d'études avant tout à ma très chère mère, celle qui a toujours été présente à mes côtés, qui m'a soutenue, encouragée et accompagnée avec amour, patience et sacrifices infinis et vos prières qui ont éclairé mon chemin. ma force et ma plus grande source d'inspiration. Ce succès est avant tout le vôtre. Que Dieu la protège et vous accorde une longue vie pleine de santé et la récompense pour tout ce qu'elle a fait pour moi.

À mon cher père, pour ses conseils, son soutien et les valeurs qu'il m'a transmises.

À mon frère Abdellah et à ma sœur Sarra, pour leur affection, leurs encouragements et leur présence dans les moments difficiles comme dans les moments de joie.

Enfin, je dédie ce mémoire à toutes les personnes qui m'ont tendu la main, à mes amis, à mes collègues ainsi qu'à tous ceux qui ont contribué, de près ou de loin, par leurs encouragements, leur soutien moral ou leur assistance technique, à surmonter les obstacles rencontrés au cours de cette étude et à transformer les défis en réussite.

Abir

Au terme de ce travail, je tiens à le dédier avec une immense gratitude à toutes les personnes qui ont contribué, de près ou de loin, à mon parcours académique et personnel.

À mes très chers parents, pour leur amour inconditionnel, leur soutien indéfectible, leurs sacrifices et leurs encouragements constants. Vous avez toujours été ma source de force et de motivation.

À ma sœur Chifaa , à mon frères et mes petites sœurs , pour leur affection, leur présence et leur confiance en moi.

À mes amies et collègues d'études, avec qui j'ai partagé de nombreux moments de travail, d'entraide et de persévérance.

Enfin, à toutes les personnes qui m'ont soutenue, encouragée et inspirée tout au long de mon parcours.

Chaimaa

Remerciements

قال الله تعالى: وَمَنْ يَشْكُرْ فَإِنَّمَا يَشْكُرُ لِنَفْسِهِ
وقال رسول الله صلى الله عليه وسلم: لَا يَشْكُرُ اللهَ مَنْ لَا يَشْكُرُ النَّاسَ.

En reconnaissance du mérite et en témoignage de gratitude, nous tenons à exprimer nos sincères remerciements et notre profonde reconnaissance à notre encadrante de recherche, Dr BENTAIEB Samia, pour avoir accepté d'encadrer ce mémoire, ainsi que pour ses orientations judicieuses, ses conseils précieux, ses remarques pertinentes et son suivi constant tout au long de ce travail, qu'elle a considéré comme le sien. Elle n'a ménagé aucun effort pour le mener à bien dans les meilleures conditions, toujours avec bienveillance et ouverture. Qu'Allah la récompense abondamment, bénisse son temps et son travail, et lui accorde davantage de réussite et de succès dans sa carrière académique.

Nous adressons également nos vifs remerciements à notre co-encadrante, Mme BOUTKHIL Malika, pour son aide précieuse.

Nous adressons également nos remerciements aux membres du jury de soutenance, qui ont bien voulu consacrer du temps à la lecture et à l'évaluation de ce mémoire :

- Dr MEKAMI Hayat – Présidente
- Dr YAGOUB Reda – Examineur

Nous tenons également à exprimer nos sincères remerciements et notre profonde gratitude à l'ensemble des enseignants du département d'Électronique et de Télécommunications pour leurs efforts sincères et leur dévouement dans l'accomplissement de leurs missions pédagogiques envers les étudiants.

Nous tenons aussi à exprimer notre reconnaissance à tous nos enseignants, depuis l'école primaire jusqu'à l'université, qui ont participé à notre formation et à notre développement académique. Grâce à leur savoir, leur patience et leurs efforts, nous avons pu avancer et atteindre cette étape importante de notre parcours.

Enfin, nous remercions toutes les personnes qui nous ont soutenus, de près ou de loin, avec bienveillance, sincérité et assistance tout au long de la réalisation de ce mémoire, en espérant qu'il pourra servir de référence utile pour les travaux futurs.

ملخص

يهدف هذا العمل إلى تطوير نظام ذكي للكشف عن حالة نعاس السائق أثناء القيادة، وذلك بهدف تعزيز السلامة الطرقيّة عبر المراقبة المباشرة. يقدم هذا المشروع حلاً مدمجاً يعتمد على تقنيات الرؤية الحاسوبية والتعلم العميق. استندت منهجيتنا إلى تقييم مقارن بين طريقتين: اعتمد النموذج الأول على هيكلية MobileNetV2، الذي أظهر استجابة سريعة أثناء مرحلة التدريب، لكنه لم يحقق الفعالية المطلوبة عند التشغيل الفعلي. بناءً على ذلك، تم تطوير النموذج الثاني باستخدام تقنية MediaPipe، التي أثبتت دقة عالية في دراسة معالم الوجه أثناء التشغيل العملي، مما جعلها الخيار الأمثل لهذا النظام. تم التحقق من فعالية النظام من خلال تجارب عملية في ظروف قيادة حقيقية داخل مركبة، مع دمج نظام تنبيه صوتي للتحذير الفوري. تبرز النتائج أن الاعتماد على تقنية MediaPipe يمثل الحل الأكثر كفاءة وموثوقية، مما يساهم بشكل فعلي في تطوير تقنيات مراقبة السائق والحد من حوادث السير الناتجة عن التعب.

الكلمات المفتاحية: الكشف عن النعاس، الرؤية الحاسوبية، MobileNet، MediaPipe، الانظمة المدمجة، سلامة السائق.

Abstract

This work aims to develop an intelligent system for detecting driver drowsiness while driving, with the goal of enhancing road safety through direct monitoring. This project offers an embedded solution based on computer vision and deep learning techniques. Our methodology is based on a comparative evaluation between two methods: the first model relied on the MobileNet architecture, which showed a quick response during the training phase but did not achieve the required effectiveness during actual operation. Consequently, the second model was developed using MediaPipe technology, which proved highly accurate in studying facial landmarks during practical operation, making it the optimal choice for this system. The system's effectiveness was verified through practical tests in real driving conditions inside a vehicle, coupled with an audible alert system for immediate warning. The results highlight that relying on MediaPipe technology represents the most efficient and reliable solution, effectively contributing to the development of driver monitoring techniques and the reduction of traffic accidents caused by fatigue.

Keywords: Drowsiness detection, computer vision, MobileNet, MediaPipe, embedded systems, driver safety.

Résumé

Ce travail vise à développer un système intelligent de détection de la somnolence du conducteur pendant la conduite, dans le but d'améliorer la sécurité routière par une surveillance directe. Ce projet propose une solution embarquée basée sur des techniques de vision par ordinateur et d'apprentissage profond. Notre méthodologie repose sur une évaluation comparative entre deux méthodes : le premier modèle est basé sur l'architecture MobileNet, qui a montré une réponse rapide lors de la phase d'entraînement, mais n'a pas atteint l'efficacité requise lors du fonctionnement réel. Par conséquent, le deuxième modèle a été développé en utilisant la technologie MediaPipe, qui a prouvé une grande précision dans l'étude des points de repère faciaux lors du fonctionnement pratique, ce qui en fait le choix optimal pour ce système. L'efficacité du système a été vérifiée par des tests pratiques dans des conditions de conduite réelles à bord d'un véhicule, couplés à un système d'alerte sonore pour un avertissement immédiat. Les résultats démontrent que l'utilisation de la technologie MediaPipe constitue la solution la plus efficace et la plus fiable, contribuant concrètement au développement des techniques de surveillance du conducteur et à la réduction des accidents de la route liés à la fatigue.

Mots clés : Détection de la somnolence, vision par ordinateur, MobileNet, MediaPipe, systèmes embarqués, sécurité du conducteur.

Table des figures	ix
Liste des tableaux	xi
I Partie Théorique	3
1 État de l'art sur la détection de la somnolence au volant	4
1.1 Introduction	5
1.2 La somnolence au volant : aspects théoriques	5
1.2.1 Définition et causes de la somnolence	5
1.2.2 Phases de la somnolence	8
1.2.3 Impact sur la conduite automobile	9
1.2.4 Statistiques et études mondiales	9
1.3 Indicateurs de somnolence au volant	11
1.3.1 Indicateurs physiologiques	12
1.3.2 Indicateurs comportementaux	13
1.3.3 Indicateurs liés au comportement du véhicule	14
1.4 Méthodes de détection de la somnolence	15
1.4.1 Approches basées sur les capteurs physiologiques	15
1.4.2 Approches basées sur la vision par ordinateur	15
1.4.3 Approches hybrides	16
1.4.4 Systèmes commerciaux existants	17
1.5 Bases de données publiques	19
1.5.1 UTA Real-Life Drowsiness Dataset (UTA-RLDD)	19
1.5.2 NTHU-DDD Dataset (National Tsing Hua University)	19
1.5.3 YawDD Dataset (Yawning Detection Dataset)	19
1.5.4 Autres datasets pertinents	20
1.6 Conclusion	20
2 Bases Théoriques du Système de Détection	21
2.1 Introduction	22
2.2 Intelligence artificielle	22
2.2.1 Machine Learning	22
2.3 Vision par ordinateur	23
2.3.1 Acquisition et traitement d'images	23
2.3.2 Détection faciale	24
2.4 MediaPipe : pipeline de détection faciale	24
2.4.1 Architecture BlazeFace	25
2.4.2 Face Mesh et les 468 points de repère	25
2.4.3 Calcul des métriques EAR et MAR	25
2.4.4 Avantages de l'approche MediaPipe	27

2.5	Apprentissage profond	28
2.5.1	Classification et régression	28
2.5.2	Apprentissage supervisé et non supervisé	28
2.5.3	Transfer Learning	29
2.5.4	Réseaux de neurones convolutifs (CNN)	30
2.5.5	Métriques d'évaluation (Accuracy, Precision, Recall, F1-Score)	34
2.6	Conclusion	35
3	Conception du système de détection	37
3.1	Introduction	38
3.2	Architecture globale du système	38
3.2.1	Vue d'ensemble comparative des deux approches	38
3.2.2	Pipeline Méthode 1 : MobileNetV2	38
3.2.3	Pipeline Méthode 2 : MediaPipe	39
3.3	Méthode 1 : approche Deep Learning (MobileNetV2)	39
3.3.1	Dataset : Driver Drowsiness Dataset (DDD)	39
3.3.2	Prétraitement des images	41
3.3.3	Transfer Learning et architecture MobileNetV2	41
3.3.4	Entraînement et résultats	42
3.3.5	Optimisation : Quantification TensorFlow Lite	44
3.4	Méthode 2 : approche Landmarks (MediaPipe)	46
3.4.1	Pipeline de détection faciale	46
3.4.2	Logique de décision	47
3.4.3	Génération des alertes	47
3.4.4	Tests en conditions réelles (<i>In the Wild</i>)	48
3.5	Comparaison des deux méthodes	49
3.6	Conclusion	50

II Partie Pratique 51

4	Environnement de Développement	52
4.1	Introduction	53
4.2	Environnement matériel	53
4.2.1	Présentation du Raspberry Pi	53
4.2.2	Écran TFT LCD 7 pouces	56
4.2.3	La caméra Pi	56
4.2.4	Carte Micro SD	57
4.3	Configuration matérielle	58
4.3.1	Téléchargement de l'image de distribution	58
4.3.2	Enregistrement du contenu de l'image sur la carte SD	58
4.3.3	Configuration initiale	58

4.4	Environnement de travail	60
4.4.1	Langage Python	61
4.4.2	L'IDE Thonny	61
4.4.3	Bibliothèques utilisées	62
4.4.4	Environnement de développement	63
4.4.5	Environnement virtuel : <code>drowsy</code>	64
4.5	Conclusion	64
5	Évaluation Comparative et Discussion	65
5.1	Introduction	66
5.2	Protocole expérimental	66
5.3	Résultats	68
5.3.1	Performances sur Raspberry Pi	73
5.3.2	Journalisation des données (log CSV)	75
5.4	Discussion	75
5.5	Conclusion	76
	Bibliographie	78

Table des figures

1.1	Le système tripartite de la sécurité routière (conducteur-véhicule-environnement)	5
1.2	Variations cycliques de la vigilance suivant le rythme biologique.	6
1.3	Identification des plages horaires à haut risque d'endormissement (nuit et début d'après-midi).	7
1.4	Classification des niveaux de risque liés à la consommation médicamenteuse.	7
1.5	Reconnaître la fatigue au volant.	8
1.6	Évolution de la performance du conducteur en fonction de son niveau d'éveil.	9
1.7	Classification des indicateurs de somnolence au volant.	12
1.8	Bandes de fréquence EEG et leur rôle dans la variation de la vigilance.	13
1.9	Analyse comparative des mouvements du volant entre l'état d'éveil et de somnolence.	14
1.10	Système de surveillance du visage et des yeux du conducteur.	16
1.11	Le système commercial Attention Assist et les signaux utilisés (Mercedes-Benz, 2008).	18
2.1	Pipeline général du système de vision par ordinateur pour la surveillance du conducteur.	23
2.2	Localisation du visage et extraction de la région d'intérêt (ROI) pour l'analyse du visage du conducteur.	24
2.3	Pipeline de détection MediaPipe : image originale, détection BlazeFace et extraction des landmarks faciaux	26
2.4	Calcul des indicateurs EAR et MAR : comparaison entre conducteur somnolent et conducteur éveillé	27
2.5	Classification par régression	28
2.6	Apprentissage supervisé vs non-supervisé	29
2.7	Principe du transfert d'apprentissage	29
2.8	Architecture des réseaux de neurones convolutifs (CNN).	30
2.9	Modèle mathématique d'un neurone artificiel (perceptron de Rosenblatt) avec ses poids et son biais.	31
2.10	Opération de convolution bidimensionnelle entre la matrice d'entrée I et le noyau de filtrage K	31
2.11	Principe du sous-échantillonnage par la méthode du Max Pooling (filtre 2×2)	32

2.12	Principe du sous-échantillonnage par la méthode de l’Average Pooling (filtre 2×2)	32
2.13	Représentation graphique des principales fonctions d’activation non linéaires . . .	33
2.14	Concept de la régularisation par Dropout pour la réduction du surapprentissage (Overfitting)	33
2.15	Matrice de confusion pour un problème de classification binaire.	35
3.1	Exemples d’images de visages de conducteurs extraites du Real-Life Drowsiness Dataset (DDD).	40
3.2	Répartition des classes dans le Driver Drowsiness Dataset.	41
3.3	Courbes d’apprentissage (Train vs Validation) du modèle	43
3.4	Matrice de confusion du modèle	43
3.5	Échantillon de prédictions du modèle MobileNetV2 sur des images de test, avec comparaison entre l’étiquette réelle (True) et la prédiction du modèle (Pred) . .	44
3.6	Pipeline complet du système de détection par landmarks : acquisition, traitement MediaPipe et alertes.	46
3.7	Échantillons testés en temps réel.	49
4.1	Évolution du Raspberry Pi.	54
4.2	Description des composants du Raspberry Pi 4 Model B.	55
4.3	Écran TFT LCD 7 pouces pour Raspberry Pi.	56
4.4	Raspberry Pi Camera Module 2.	57
4.5	Carte Micro SD.	58
4.6	Modification du nom d’hôte.	59
4.7	Activation du SSH.	60
4.8	Activation du VNC.	60
4.9	Illustration du bureau du Raspberry Pi.	61
4.10	L’IDE Thonny.	61
5.1	Architecture générale du système embarqué basé sur Raspberry Pi.	66
5.2	Dispositif matériel du système embarqué : Raspberry Pi 4, écran TFT 7 pouces et module caméra monté sur support, exécutant l’interface de détection en temps réel.	67
5.3	Interface de model MobileNetV2	68
5.4	Validation expérimentale du système MobileNetV2	69
5.5	Interface de model Mediapipe	70
5.6	Validation expérimentale du système Mediapipe	72
5.7	Tests expérimentaux du système MediaPipe sur un sujet portant des lunettes de soleil	73
5.8	Tests expérimentaux du système MediaPipe sans port de lunettes : fermeture prolongée des yeux et bâillement, sous différents angles de tête	74

Liste des tableaux

1.1	Synthèse des causes de la somnolence au volant.	6
1.2	Comparaison des approches de détection de la somnolence.	18
2.1	Comparaison des architectures CNN de référence.	34
3.1	Vue d'ensemble comparative des deux approches.	38
3.2	Rapport de performance global du modèle MobileNetV2	44
3.3	Comparaison des modèles TensorFlow Lite quantifiés.	46
3.4	Comparaison détaillée des deux approches de détection de somnolence.	49
5.1	Extrait du fichier de log (<code>drowsy_log.csv</code>).	75

Liste des Abréviations

AAA Foundation American Automobile Association Foundation for Traffic Safety

AI Artificial Intelligence (Intelligence Artificielle)

API Application Programming Interface

AUC-DDD Autonomous University of Chile - Driver Drowsiness Dataset

CAN Controller Area Network

CNN Convolutional Neural Network

CV Computer Vision

CVL Computer Vision Laboratory

DBSCAN Density-Based Spatial Clustering of Applications with Noise

DDD Driver Drowsiness Dataset

DL Deep Learning

DMD Driver Monitoring Dataset

DMS Driver Monitoring System

DNSR Délégation Nationale à la Sécurité Routière

DSP Digital Signal Processor

EAR Eye Aspect Ratio

ECG Électrocardiogramme

EEG Électroencéphalogramme

EOG Électrooculogramme

ERT Ensemble of Regression Trees

Float16 Floating Point 16 bits

Float32 Floating Point 32 bits

FN False Negative

FP False Positive

GPU Graphics Processing Unit

GSM Global System for Mobile Communications

IDE Integrated Development Environment

ILSVRC ImageNet Large Scale Visual Recognition Challenge

ImageNet Image Database for Visual Recognition

INT8 Integer 8 bits

IR Infrarouge

k-NN k-Nearest Neighbors (k plus proches voisins)

KB Kilobyte

KSS Karolinska Sleepiness Scale

LCD Liquid Crystal Display

LSTM Long Short-Term Memory

MAR Mouth Aspect Ratio

ML Machine Learning

MobileNet Réseau de neurones léger pour appareils mobiles

ms milliseconde

NHTSA National Highway Traffic Safety Administration

NPU Neural Processing Unit

NTHU-DDD National Tsing Hua University - Driver Drowsiness Detection Dataset

OMS Organisation Mondiale de la Santé

OpenCV Open Source Computer Vision Library

ORL Olivetti Research Laboratory

PCA Principal Component Analysis (Analyse en Composantes Principales)

PERCLOS Percentage of Eye Closure (pourcentage de fermeture des yeux)

Python Langage de programmation

QT Framework pour le développement d'interfaces graphiques

ResNet Residual Neural Network

RGB Red Green Blue (modèle de couleur)

ROI Region of Interest (Région d'Intérêt)

RPi Raspberry Pi

RSA Réseau de Sécurité Active

RVB Rouge Vert Bleu

SD Secure Digital

SDLP Standard Deviation of Lane Position (écart-type de la position latérale)

SONAR Sound Navigation And Ranging

SSD Single Shot MultiBox Detector

SSH Secure Shell

SVD Singular Value Decomposition (Décomposition en valeurs singulières)

SWM Steering Wheel Movement (mouvement du volant)

TFLite TensorFlow Lite

TFT Thin-Film Transistor

TN True Negative

TP True Positive

ULg Université de Liège

UTA-RLDD Université de Texas at Arlington - Real-Life Drowsiness Dataset

VFC Variabilité de la Fréquence Cardiaque

VGGNet Visual Geometry Group Network

VLC VideoLAN Client

VNC Virtual Network Computing

YawDD Yawning Detection Dataset

Introduction Générale

Avec l'évolution croissante du parc automobile mondial et l'augmentation continue du trafic routier, la sécurité des usagers de la route demeure une priorité majeure. Parmi les facteurs de risque critiques, la somnolence au volant est reconnue comme une cause principale d'accidents mortels. En altérant la vigilance du conducteur et en ralentissant ses temps de réaction, cet état physiologique nécessite des solutions de surveillance proactives, capables d'intervenir avant la survenue d'un événement critique. Dans ce contexte, l'intégration des systèmes embarqués et de la vision par ordinateur offre des opportunités prometteuses pour automatiser la détection de la fatigue en temps réel, sans contrainte pour le conducteur. L'utilisation de technologies légères et performantes permet aujourd'hui de transformer des dispositifs embarqués standards, tels que le Raspberry Pi, en outils efficaces de prévention routière.

L'objectif principal de ce travail consiste à développer un système intelligent de détection de la somnolence au volant, basé sur l'intelligence artificielle et le Raspberry Pi 4, afin d'identifier automatiquement l'état de vigilance du conducteur à partir des flux vidéo capturés en conditions réelles.

Le présent mémoire est organisé en deux parties et cinq chapitres principaux :

1. **Partie Théorique**

Chapitre 1 : État de l'art sur la détection de la somnolence au volant 4

Ce chapitre présente les fondements théoriques de la somnolence au volant, ses causes, ses différentes phases ainsi que son impact sur la conduite automobile. Il expose également les principaux indicateurs de fatigue, les méthodes de détection existantes et les bases de données utilisées dans ce domaine de recherche.

Chapitre 2 : Bases théoriques du système de détection 21

Ce chapitre introduit les concepts théoriques nécessaires à la compréhension du système développé. Il présente les notions d'intelligence artificielle, de Machine Learning, de Deep Learning, les réseaux de neurones convolutifs (CNN), ainsi que les technologies de vision par ordinateur utilisées pour la détection de la somnolence, notamment MobileNetV2 et MediaPipe.

Chapitre 3 : Conception du système de détection 37

Ce chapitre décrit les différentes étapes de développement des modèles utilisés. Il présente la préparation des données, l'entraînement du modèle MobileNetV2 pour la classification des états de vigilance ainsi que l'approche basée sur MediaPipe pour

l'analyse des points de repère faciaux. Une comparaison des performances des deux approches y est également réalisée.

2. **Partie Pratique**

Chapitre 4 : Environnement de développement 52

Ce chapitre présente l'architecture matérielle et logicielle du système développé. Il détaille l'intégration des modèles sur Raspberry Pi 4, la configuration de la caméra, la mise en place du système d'alerte sonore ainsi que le fonctionnement global du dispositif de détection en temps réel.

Chapitre 5 : Évaluation comparative et discussion 65

Ce chapitre est consacré à l'évaluation du système proposé. Il présente les résultats obtenus lors des expérimentations, compare les performances de MobileNetV2 et MediaPipe en conditions réelles et discute les avantages, les limites ainsi que les perspectives d'amélioration du système.

À travers ce travail, nous proposons une solution intelligente de détection de la somnolence au volant combinant la vision par ordinateur, l'intelligence artificielle et les systèmes embarqués. Les résultats obtenus démontrent la faisabilité et l'efficacité de la solution développée pour la surveillance en temps réel de l'état de vigilance du conducteur, tout en mettant en évidence le potentiel des technologies modernes dans l'amélioration de la sécurité routière.

Première partie

Partie Théorique

Chapitre 1

État de l’art sur la détection de la sommolence au volant

1.1	Introduction	5
1.2	La somnolence au volant : aspects théoriques	5
1.2.1	Définition et causes de la somnolence	5
1.2.2	Phases de la somnolence	8
1.2.3	Impact sur la conduite automobile	9
1.2.4	Statistiques et études mondiales	9
1.3	Indicateurs de somnolence au volant	11
1.3.1	Indicateurs physiologiques	12
1.3.2	Indicateurs comportementaux	13
1.3.3	Indicateurs liés au comportement du véhicule	14
1.4	Méthodes de détection de la somnolence	15
1.4.1	Approches basées sur les capteurs physiologiques	15
1.4.2	Approches basées sur la vision par ordinateur	15
1.4.3	Approches hybrides	16
1.4.4	Systèmes commerciaux existants	17
1.5	Bases de données publiques	19
1.5.1	UTA Real-Life Drowsiness Dataset (UTA-RLDD)	19
1.5.2	NTHU-DDD Dataset (National Tsing Hua University)	19
1.5.3	YawDD Dataset (Yawning Detection Dataset)	19
1.5.4	Autres datasets pertinents	20
1.6	Conclusion	20

1.1 Introduction

La sécurité routière est l'une des priorités majeures de la société mondiale. Elle englobe à la fois la sécurité passive des véhicules (sièges et dispositifs de retenue, airbags, ceintures de sécurité, structures déformables) et la sécurité active, qui exige une attention constante du conducteur. Parmi les risques les plus critiques figure la somnolence au volant, un phénomène complexe situé aux confins de la biologie humaine et de la dynamique des véhicules. Comme l'illustre la Figure 1.1, ce risque s'inscrit dans l'interaction constante entre les trois piliers majeurs de la sécurité routière : le conducteur, le véhicule et l'environnement extérieur.

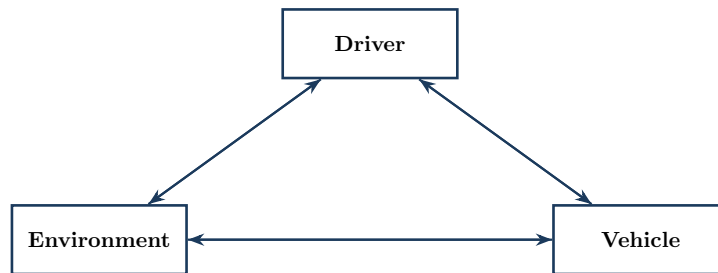


FIGURE 1.1 – Le système tripartite de la sécurité routière (conducteur-véhicule-environnement)

1.2 La somnolence au volant : aspects théoriques

1.2.1 Définition et causes de la somnolence

La somnolence est un état intermédiaire entre l'état de veille et le sommeil, également qualifié de « veille altérée » [1]. Elle se manifeste par une difficulté croissante à rester éveillé et peut conduire à un endormissement involontaire. Sur le plan cognitif, elle correspond à une diminution progressive de l'efficacité du traitement cérébral, rendant l'exécution de tâches normalement simples plus difficile. La somnolence s'accompagne d'un allongement du temps de réaction, d'une baisse de la vigilance, d'un déficit dans le traitement des informations disponibles et d'erreurs de la mémoire à court terme. Dans le contexte de la conduite, cet état entraîne un comportement moins vigilant et une perte partielle du contrôle du véhicule [2]. Dans le domaine de l'instrumentation appliquée à la santé et à la sécurité, plusieurs facteurs contribuant à l'apparition de la somnolence peuvent être distingués :

1.2.1.1 Les causes de la somnolence

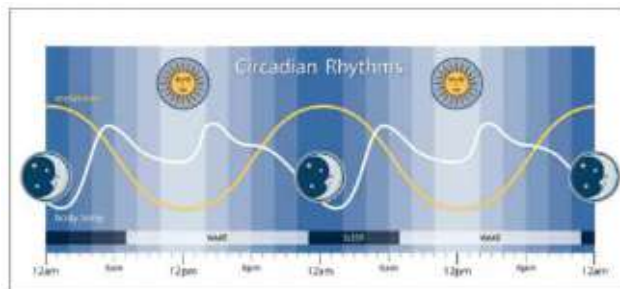
Les causes de la somnolence au volant peuvent être classées en trois catégories principales, résumées dans le Tableau 1.1.

TABLEAU 1.1 – Synthèse des causes de la somnolence au volant.

Catégorie	Facteurs identifiés
Causes physiologiques	Rythmes circadiens (baisses naturelles de vigilance, nuit et milieu de journée)
	Dette de sommeil accumulée (créneaux critiques 2h–5h et 13h–15h)
	Prédispositions personnelles (âge, sexe)
	Troubles respiratoires (apnée du sommeil)
Causes comportementales	Hygiène de vie et alimentation (repas riches, alcool)
	Consommation de substances (alcool, drogues, médicaments)
Causes environnementales	Facteurs ambiants (chaleur, faible luminosité, route monotone)
	Exposition à un environnement sombre (sécrétion de mélatonine)

1. **Causes physiologiques** : Elles sont liées à l'état biologique du conducteur :

- **Rythmes circadiens** : Le corps humain suit un cycle de 24 heures géré par l'horloge interne. Comme l'illustre la Figure 1.2, la vigilance n'est pas constante et subit des baisses naturelles, particulièrement marquées durant la nuit et en milieu de journée [3].

**FIGURE 1.2** – Variations cycliques de la vigilance suivant le rythme biologique.

- **Dette de sommeil** : Le manque de sommeil accumulé réduit le temps de réaction. Les périodes les plus critiques où le risque d'endormissement est maximal sont représentées dans la Figure 1.3, ciblant spécifiquement les créneaux 2h–5h et 13h–15h [4]. Cela peut être causé par un sommeil insuffisant, de longs trajets ou des périodes de travail prolongées, travail de nuit ou travail posté [5].



FIGURE 1.3 – Identification des plages horaires à haut risque d'endormissement (nuit et début d'après-midi).

- **Prédispositions personnelles** : Des facteurs tels que l'âge ou le sexe influencent la résistance à la fatigue [6].
 - **Troubles respiratoires** : Une mauvaise oxygénation durant le sommeil (comme l'apnée) provoque une fatigue chronique durant la journée [7].
2. **Causes comportementales** : Elles concernent les habitudes du conducteur, telles que la conduite prolongée sans pause :
- **Hygiène de vie et alimentation** : La consommation de repas riches, surtout s'ils sont associés à l'alcool, accentue la somnolence postprandiale (après le repas) [1].
 - **Consommation de substances** : L'alcool et les drogues diminuent drastiquement les réflexes [2,8]. De plus, certains traitements médicaux sont classés par niveaux de danger pour la conduite, comme on peut le voir sur la signalétique de la Figure 1.4, allant de la simple prudence à l'interdiction totale de conduire [1].



FIGURE 1.4 – Classification des niveaux de risque liés à la consommation médicamenteuse.

3. **Causes environnementales** : Elles dépendent des conditions extérieures :
- **Facteurs environnementaux** : Une température élevée dans l'habitacle, une faible luminosité [8,9], ou encore une conduite prolongée sur une route monotone peuvent accentuer la somnolence, phénomène souvent qualifié d'« hypnose autoroutière » [10].
 - **L'exposition à un environnement sombre** : L'exposition à un environnement sombre favorise la sécrétion de l'hormone mélatonine, produite par la glande pinéale et sécrétée préférentiellement pendant la nuit avec un pic vers 3 h du matin. La lumière artificielle, dans des conditions précises d'administration, peut supprimer ou décaler cette sécrétion [11], notamment dans les habitacles équipés de vitres filtrant la lumière.

1.2.2 Phases de la somnolence

Le premier signal d'alerte au volant est la fatigue, comme l'illustre la Figure 1.5. Tout commence par des bâillements, des paupières lourdes et des yeux qui piquent. Rapidement, le besoin de bouger s'installe et garder la tête droite devient difficile. C'est à ce moment que la notion du temps lui échappe : la vitesse devient instable (on accélère ou on ralentit sans s'en rendre compte) et la concentration s'évapore. Des tensions physiques apparaissent aussi, comme des courbatures ou des raideurs dans la nuque et le dos, signalant un besoin urgent de faire une pause. Si l'on ignore ces symptômes qui s'intensifient, l'endormissement complet devient inévitable, augmentant drastiquement le risque d'accident. C'est ce qu'on appelle la somnolence au volant [12].



FIGURE 1.5 – Reconnaître la fatigue au volant.

La somnolence est un processus progressif et non instantané. Elle évolue selon trois phases distinctes, allant de la simple fatigue cognitive jusqu'aux micro-sommeils, décrites comme suit :

- **Fatigue cognitive** : Le sujet commence à ressentir une baisse de vigilance et de concentration [13].
- **Somnolence légère** : Le sujet tente de maintenir les yeux ouverts malgré la fatigue [9].
- **Micro-sommeils en conduite réelle** : Les micro-sommeils constituent des phases critiques durant la conduite, pendant lesquelles le niveau de danger devient particulièrement élevé. Il s'agit de très courts épisodes de sommeil, généralement compris entre 0,5 et 15 secondes, durant lesquels le cerveau se déconnecte temporairement des stimulations sensorielles extérieures. Dans certains cas, les yeux peuvent rester ouverts, ce qui rend leur détection plus difficile par des systèmes basés uniquement sur la caméra [14].

La somnolence se manifeste différemment d'un conducteur à l'autre, tant par la nature des symptômes que par leur intensité. On peut toutefois regrouper ces alertes en trois grandes catégories [15] :

- **Signes physiques et physiologiques** : Bâillements fréquents, paupières lourdes, sensation de brûlure aux yeux, balancements du corps, ainsi qu'une modification du rythme

cardiaque et de la respiration (qui devient plus superficielle).

- **Altération des facultés mentales** : Pensées confuses ou « brumeuses », sautes d'humeur (irritabilité, déprime), moments d'absence (rêveries) et baisse globale de la concentration.
- **Dégradation de la conduite** : Réflexes plus lents, difficultés à maintenir une trajectoire droite et une vitesse stable, manque d'attention à la signalisation ou aux piétons, et écarts de conduite involontaires.

1.2.3 Impact sur la conduite automobile

Conduire fatigué, c'est conduire diminué. La fatigue agit directement sur les fonctions cognitives et motrices qui permettent de réagir, d'anticiper et de décider au volant [2], comme l'illustre la Figure 1.6. Dès que le niveau d'éveil baisse, le conducteur sort de sa zone optimale. Ses facultés se dégradent. Et plus il passe de temps derrière le volant, au-delà de deux heures par jour, plus cet état devient fréquent [1]. Le temps de réaction s'allonge de façon significative. Les études le comparent aux effets d'une imprégnation alcoolique, avec un risque de collision multiplié en conséquence [5]. La prise de décision suit la même tendance : le conducteur peine à évaluer les risques en temps réel, ce qui favorise les erreurs critiques [5]. Pire encore, ce phénomène reste largement invisible dans les statistiques. De nombreux accidents liés à la somnolence sont classés sous d'autres causes, ce qui rend difficile toute évaluation fiable de son ampleur réelle [5].

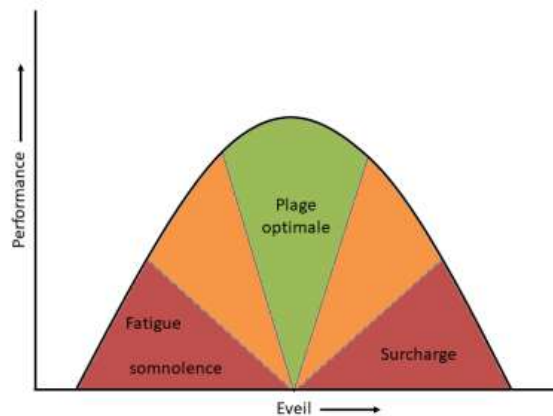


FIGURE 1.6 – Évolution de la performance du conducteur en fonction de son niveau d'éveil.

À l'échelle mondiale, la situation est préoccupante. L'Organisation Mondiale de la Santé (OMS) place la somnolence au volant parmi les facteurs majeurs d'accidents de la route à travers le monde [5].

1.2.4 Statistiques et études mondiales

Les statistiques les plus récentes (2024–2025) sont utilisées pour illustrer l'état actuel du phénomène, tandis que les données antérieures permettent d'analyser son évolution.

1.2.4.1 Statistiques mondiales sur la somnolence au volant

La somnolence au volant est aujourd'hui considérée comme l'un des facteurs importants des accidents de la route à l'échelle mondiale. Selon les estimations de l'Organisation mondiale de la santé (OMS), environ 1,3 million de personnes meurent chaque année dans des accidents de la route dans le monde [16]. Plusieurs études scientifiques indiquent que la somnolence au volant contribue à environ 3% à plus de 30% des accidents selon les pays étudiés et les méthodes d'évaluation utilisées [17], et que plus de 20% des conducteurs déclarent avoir dû s'arrêter au moins une fois pendant la conduite en raison de la somnolence [18].

Aux États-Unis, les données de la National Highway Traffic Safety Administration (NHTSA) indiquent qu'environ 100 000 accidents sont causés chaque année par des conducteurs somnolents, entraînant plus de 70 000 blessés et plus de 1 500 décès (NHTSA, Drowsy Driving Statistics). En 2017, environ 91 000 accidents signalés par la police impliquaient des conducteurs somnolents, causant près de 50 000 blessés et environ 800 décès [19]. Plus récemment, 633 décès liés à la somnolence ont été enregistrés en 2023 selon les rapports officiels de la NHTSA. Les études de la AAA Foundation for Traffic Safety indiquent que 17,6% des conducteurs impliqués dans des accidents mortels étaient estimés somnolents, et qu'environ 29 834 personnes ont perdu la vie dans des accidents impliquant un conducteur somnolent entre 2017 et 2021 [20]. Par ailleurs, environ 60% des conducteurs adultes déclarent avoir conduit en étant somnolents au moins une fois au cours de l'année, et plus d'un tiers (37%) affirment s'être déjà endormis au volant au moins une fois dans leur vie [21].

Ces statistiques montrent que la somnolence représente un facteur de risque important pour la sécurité routière mondiale, ce qui explique l'intérêt croissant pour le développement de systèmes intelligents de détection de la fatigue chez les conducteurs.

1.2.4.2 Statistiques sur la somnolence au volant en Algérie

En Algérie, les accidents de la route représentent un problème majeur de santé publique. Selon les statistiques publiées par la Délégation Nationale à la Sécurité Routière (DNSR), 26 272 accidents ont été enregistrés en 2024, causant 3 740 décès et 35 556 blessés [22]. D'autres données indiquent que 26 976 accidents ont été enregistrés à l'échelle nationale, entraînant 3 838 décès et 37 020 blessés [23].

Les analyses réalisées par les autorités de sécurité routière montrent que plus de 96% des accidents sont liés au facteur humain, notamment l'excès de vitesse, le manque d'attention et la fatigue du conducteur [24]. Les rapports du Groupement de la Gendarmerie nationale indiquent que la négligence humaine constitue la cause principale d'environ 93% des accidents. Selon les données de la Direction Générale de la Sûreté Nationale [25], plus de 95% des accidents sont liés au facteur humain, parmi lesquels figurent la fatigue, le manque de concentration et la conduite prolongée sans repos [26]. Cependant, les accidents directement liés à la somnolence restent souvent sous-déclarés, car ils sont généralement classés dans d'autres catégories comme la perte de contrôle du véhicule ou l'inattention du conducteur.

1.2.4.3 Statistiques dans les pays ayant mis en place des systèmes de détection de la somnolence au volant

Certains pays ont introduit des réglementations ou des normes pour les systèmes de surveillance du conducteur (Driver Monitoring Systems – DMS) :

- L'Union européenne a rendu ces systèmes obligatoires pour tous les nouveaux véhicules à partir de 2024–2026 [27].
- La législation japonaise inclut la surveillance de l'état du conducteur pour évaluer sa vigilance [28].
- La Chine a mis en place un standard national GB/T 40429-2021 relatif aux systèmes de surveillance et d'attention du conducteur [29].

Face à l'impact de la fatigue sur la sécurité routière, plusieurs pays ont commencé à développer et à intégrer des systèmes de surveillance du conducteur dans les véhicules modernes :

- En Europe, les analyses de la commission européenne indiquent que la fatigue du conducteur est impliquée dans environ 20% à 25% des accidents corporels, et dans près d'un accident mortel sur quatre sur les autoroutes [30].
- Des enquêtes récentes menées en Belgique (Wallonie) montrent que 46% des automobilistes ont admis avoir conduit en état de somnolence au cours du dernier mois [31].
- En France, une étude réalisée par la Fondation VINCI Autoroutes indique que près d'un conducteur sur trois a déjà connu un épisode de somnolence sévère nécessitant un arrêt immédiat, et que 40% de ces situations se terminent par un quasi-accident ou un accident [32].

Les systèmes basés sur l'intelligence artificielle peuvent atteindre une précision de détection de la fatigue allant jusqu'à 96,8% [33], et il a été démontré qu'après leur mise en œuvre, les accidents liés à la fatigue ont été réduits d'environ 35%, permettant ainsi de sauver des centaines de vies chaque année.

1.3 Indicateurs de somnolence au volant

L'évaluation du niveau de fatigue et de vigilance chez un conducteur s'appuie sur différentes métriques extraites de l'activité biologique de l'organisme, des manifestations physiques externes ou de la dynamique de conduite. La Figure 1.7 présente une synthèse de ces indicateurs.

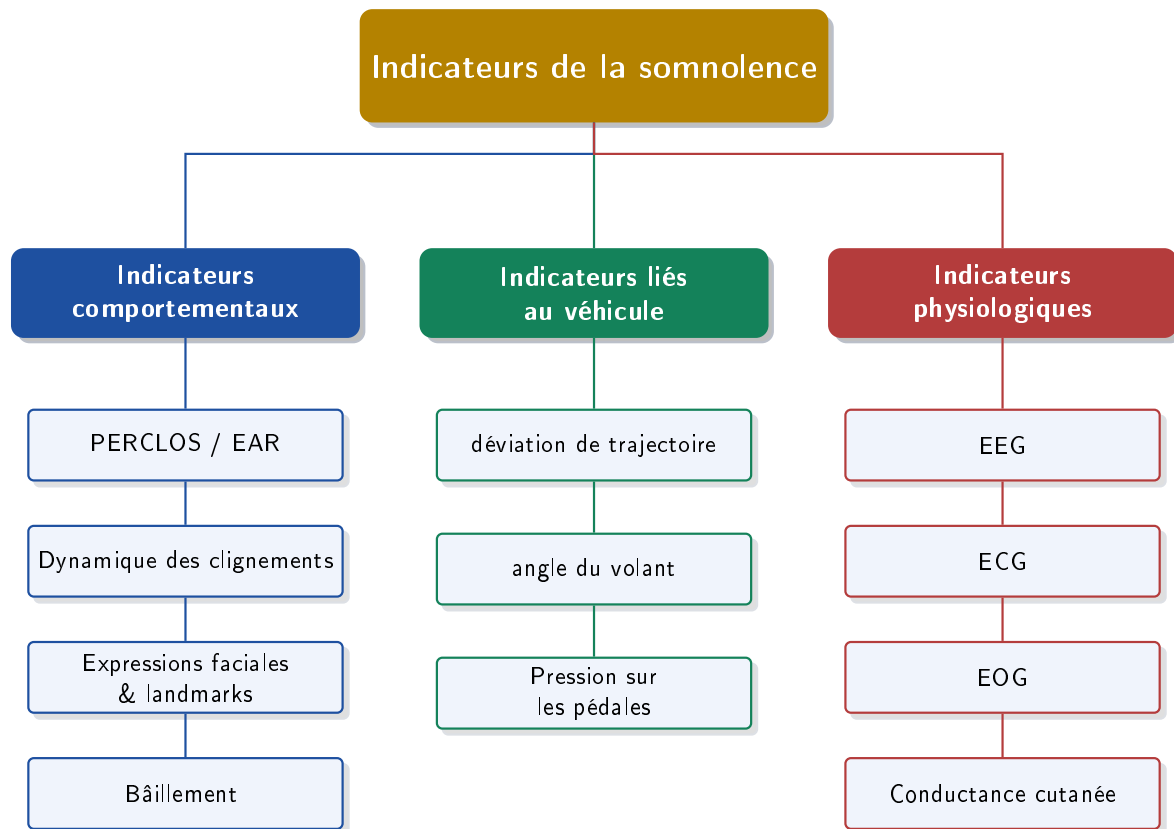


FIGURE 1.7 – Classification des indicateurs de somnolence au volant.

1.3.1 Indicateurs physiologiques

Ces indicateurs sont considérés comme les plus objectifs car ils découlent directement des fluctuations des systèmes nerveux central et autonome. Ils servent généralement de vérité terrain lors des expérimentations scientifiques.

- **L'Électroencéphalogramme (EEG)** : Il représente le signal de référence absolu pour mesurer la vigilance. La transition vers la somnolence se traduit par une baisse d'activité des ondes Bêta (synonymes de concentration) et une augmentation significative de l'amplitude des ondes de basse fréquence Alpha et Theta.

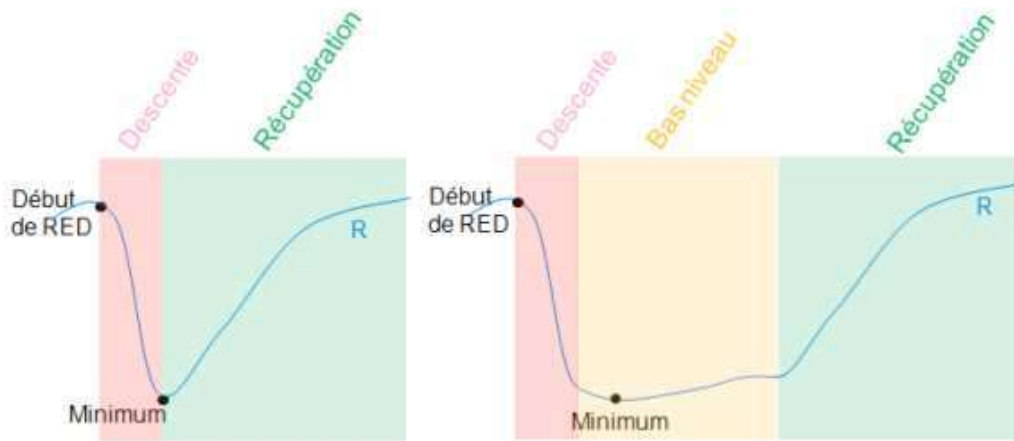


FIGURE 1.8 – Bandes de fréquence EEG et leur rôle dans la variation de la vigilance.

Le tableau présenté dans la Figure 1.8 résume les différentes bandes de fréquence de l'EEG selon Larue (2010). On observe clairement que la transition vers la somnolence se traduit par une augmentation des ondes Thêta (θ) et Alpha (α).

- **L'Électrocardiogramme (ECG)** : Il permet d'analyser la variabilité de la fréquence cardiaque. L'endormissement provoque un ralentissement du rythme cardiaque et une hausse de la VFC, ce qui témoigne de l'activation du système nerveux parasympathique.
- **L'Électrooculogramme (EOG) et conductance cutanée** : L'EOG quantifie précisément les mouvements lents des yeux et les fermetures prolongées des paupières. En parallèle, la conductance cutanée suit l'activité électrodermale liée à la transpiration pour évaluer le stress et la fatigue.

Bien que d'une précision médicale indiscutable, ces indicateurs restent extrêmement difficiles à déployer en conditions réelles. Leur caractère hautement intrusif (nécessité de poser des électrodes ou des capteurs inconfortables sur le conducteur) et le coût élevé des équipements limitent leur utilisation aux phases de laboratoire.

1.3.2 Indicateurs comportementaux

Contrairement aux signaux biologiques, les indicateurs comportementaux se basent sur l'observation des signes physiques externes et visibles du conducteur à l'aide de capteurs optiques non intrusifs.

- **L'indicateur Percentage of Eye Closure (PERCLOS)** : Proposé par Wierwille et Ellsworth en 1994, il calcule le pourcentage de temps où les paupières recouvrent la pupille à plus de 80%. C'est la métrique la plus validée et la plus sensible pour identifier l'installation d'une somnolence involontaire. En pratique, le PERCLOS est estimé à partir de l'Eye Aspect Ratio (EAR), un indicateur géométrique calculé à partir des points de repère faciaux situés autour de l'œil, et représentant le rapport entre la hauteur et la largeur de l'ouverture oculaire.
- **La dynamique des clignements d'yeux** : La fatigue induit un allongement anormal de la durée d'un clignement (souvent supérieure à 300 ms) combiné à une baisse de leur fréquence globale.

- **Expressions faciales et landmarks** : L'extraction automatique de points de repère faciaux permet de suivre la géométrie des traits. Ces modèles calculent notamment l'indice d'ouverture de la bouche pour détecter et quantifier la cinématique des bâillements, ainsi que les inclinaisons ou relâchements suspects de la tête.

L'efficacité de ces indicateurs dépend entièrement de la qualité du flux vidéo. Les variations de luminosité ambiante, la conduite de nuit ou le port d'accessoires (comme des lunettes de soleil) représentent des obstacles majeurs pouvant fausser l'analyse des algorithmes de vision.

1.3.3 Indicateurs liés au comportement du véhicule

Ces indicateurs mesurent l'impact de la fatigue sur l'action globale de conduite en analysant les données mécaniques issues du véhicule, généralement accessibles via le bus CAN de l'automobile.

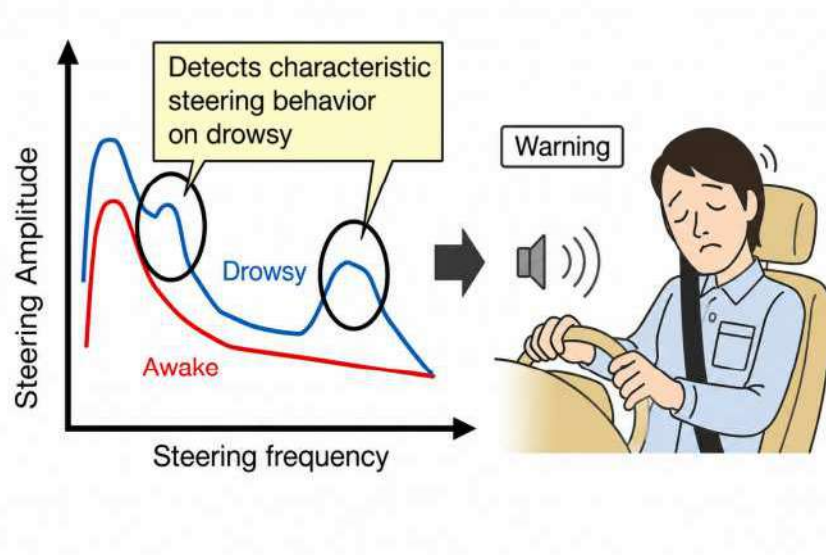


FIGURE 1.9 – Analyse comparative des mouvements du volant entre l'état d'éveil et de somnolence.

La Figure 1.9 montre la variation de l'amplitude et de la fréquence de braquage, mettant en évidence les micro-corrrections brusques caractérisant l'état de somnolence.

- **Déviations de trajectoire** L'écart-type de la position latérale du véhicule par rapport au centre de la voie augmente de manière critique avec la fatigue, en particulier lorsque le conducteur atteint un score élevé sur l'échelle de somnolence de Karolinska ($KSS > 7$) [34] [35].
- **Variabilité de l'angle du volant** Un conducteur éveillé effectue des micro-corrrections continues (entre 1° et 5°). En état de somnolence, ces petits mouvements diminuent au profit de corrections brusques et amples (de 6° à 10°) pour rattraper les dérives. Cette instabilité est mesurable via le calcul de l'entropie de direction [36] [37].
- **Pression sur les pédales** : La fatigue engendre une baisse de réactivité et des variations irrégulières ou des relâchements soudains de la pression exercée sur les pédales d'accélérateur et de frein [38].

Bien que ces mesures soient totalement transparentes et non invasives pour le conducteur, elles présentent des signes tardifs. De plus, elles peuvent être biaisées par la distraction ou le type

de route, et s'avèrent insuffisantes dans le contexte des véhicules autonomes où le conducteur n'intervient plus activement sur les commandes [39].

Une baisse de pression ou des variations anormales indiquent un déclin des performances de conduite dû à la fatigue. La plupart des études combinent plusieurs sources de mesures (vitesse, pression sur les pédales, mesures physiologiques) pour réduire les fausses alertes et améliorer la détection de la somnolence. Ces mesures sont non invasives et souvent déjà disponibles via le réseau CAN du véhicule, permettant une utilisation en temps réel. Certaines mesures peuvent être influencées par d'autres facteurs que la somnolence, comme la distraction ou le déclin attentionnel, d'où l'importance de les combiner avec d'autres indicateurs. Dans le contexte des véhicules autonomes ou partiellement autonomes, ces mesures seules peuvent être insuffisantes pour évaluer l'état réel du conducteur.

1.4 Méthodes de détection de la somnolence

Les méthodes actuelles de détection de la somnolence se divisent en trois approches principales : l'approche comportementale (basée sur le visage et les yeux), l'approche véhiculaire (la dynamique du véhicule) et l'approche physiologique. Chaque approche repose sur une technique précise, soit par un capteur, soit par la vision par ordinateur, ou bien sur la combinaison de deux techniques ou plus, dans ce cas appelée approche hybride.

1.4.1 Approches basées sur les capteurs physiologiques

L'approche physiologique fait référence à la méthode de détection de la somnolence en utilisant des données recueillies par des capteurs fixés sur le corps du conducteur, tel que le capteur de rythme cardiaque. Selon Arunasalam et al. (2019), la somnolence peut être détectée en utilisant les données collectées par un capteur de rythme cardiaque et un capteur de clignement des yeux. Ces données sont ensuite traitées ; si la somnolence est détectée, la vitesse du véhicule est réduite jusqu'à son arrêt complet, et l'état actuel ainsi que l'emplacement du conducteur sont à la fois suivis et envoyés au Cloud via le module GSM [40].

1.4.2 Approches basées sur la vision par ordinateur

Les systèmes basés sur la vision par ordinateur surveillent en permanence le visage du conducteur. Comme illustré dans la Figure 1.10, ces dispositifs permettent de suivre l'état des yeux et les expressions faciales afin de détecter les premiers signes de fatigue sans aucun contact physique.



FIGURE 1.10 – Système de surveillance du visage et des yeux du conducteur.

Cette méthode utilise une caméra vidéo (souvent couplée à un éclairage LED infrarouge de 850 nm pour la conduite de nuit) orientée vers le visage. La première étape, l'extraction des caractéristiques, repose sur l'utilisation des modèles Dlib ou MediaPipe pour détecter les 68 points des traits du visage avec un taux de 89,26% et 95% de précision. La seconde étape, la classification, fait appel à des réseaux de neurones (CNN ou LSTM pour la dimension temporelle) afin de faire la distinction entre un clignement normal et un début d'endormissement [41]. Cette méthode offre l'avantage de surveiller l'état de vigilance du conducteur de manière continue et non intrusive, permettant ainsi une réaction rapide en cas de signes précoces de somnolence.

1.4.3 Approches hybrides

La somnolence provoque une vision en tunnel (rétrécissement du champ visuel) ainsi qu'une diminution de la coordination main-œil. Côté instrumentation, cela explique pourquoi le temps de réponse conducteur-véhicule décroît, en passant d'environ 1 seconde à plus de 3 secondes, ce qui est souvent fatal à vitesse élevée [42].

Aucune mesure unique n'est capable de refléter avec précision les variations de l'état de vigilance du conducteur, car elle peut être imprécise, incomplète ou incertaine. De plus, la variabilité interindividuelle dans la réponse à la somnolence ou au manque de sommeil rend les mesures individuelles insuffisantes. Ainsi, les études se sont tournées vers la combinaison de plusieurs mesures (comportementales, physiologiques et issues du véhicule), ce qui augmente les sources d'information, réduit le nombre de fausses alertes et améliore les performances de détection de la somnolence. L'utilisation de mesures hybrides permet de compenser les limites de chaque mesure prise isolément et offre une estimation plus fiable de l'état de vigilance en temps réel [43] [44].

Dans le cadre de la détection de la somnolence chez le conducteur, plusieurs approches hybrides ont été développées pour améliorer la précision et la robustesse des systèmes de surveillance. Ces approches combinent différentes sources de données ou techniques afin de tirer profit des avantages de chacune.

- **Hybride Vision & ML** : Cette approche repose sur l'extraction de caractéristiques faciales pertinentes (état des yeux, clignement, ouverture de la bouche, etc.) à l'aide de techniques de vision par ordinateur. Ces caractéristiques servent ensuite de données d'entrée pour des algorithmes de Machine Learning, permettant de classifier l'état

de vigilance du conducteur. Ce procédé améliore l'efficacité par rapport aux méthodes traditionnelles et offre une alternative pratique pour la détection automatique de la somnolence [45].

- **Hybride Vision & DL** : L'approche Vision + Deep Learning combine l'utilisation d'images hybrides (par exemple, images visibles et images infrarouges) avec des réseaux de neurones profonds pour extraire automatiquement des caractéristiques pertinentes. Cette combinaison permet de conserver les avantages de chaque modalité d'image tout en améliorant la précision et en réduisant les erreurs de détection. Elle est particulièrement utile pour la détection en temps réel de signes de somnolence [46].

Le Tableau 1.2 résume et compare ces trois approches selon leur mécanisme de détection, les outils et techniques utilisés, ainsi que leurs avantages et inconvénients respectifs.

- **Hybride Physiologique & Vision** : Cette approche repose sur l'étude des phénomènes visuels à travers les propriétés physiologiques de l'appareil visuel. Elle combine l'analyse des signaux physiologiques, tels que la réponse de la rétine ou les mouvements oculaires, avec l'observation des caractéristiques visuelles pour mieux évaluer la vigilance du conducteur. Ce couplage direct entre les réponses biologiques et les stimuli visuels offre une méthode intégrée pour l'estimation de l'état de somnolence [47].

1.4.4 Systèmes commerciaux existants

Plusieurs systèmes commerciaux ont été développés pour détecter la somnolence et améliorer la sécurité des conducteurs. Voici quelques exemples :

- **Attention Assist (Mercedes-Benz)** : est un système conçu pour avertir le conducteur des premiers signes de somnolence (Assistant d'attention). Le capteur de direction est relié à un programme intelligent qui utilise 70 paramètres afin d'établir un profil unique du conducteur durant les 20 premières minutes de conduite. Entre 80 et 180 km/h, le système détecte les corrections de trajectoire irrégulières effectuées par les conducteurs lorsqu'ils commencent à ressentir de la somnolence, émet un avertissement sonore et affiche le message « Temps pour une pause ? » accompagné d'un symbole de tasse de café sur le tableau de bord. À titre d'exemple industriel, le système Attention Assist développé par Mercedes-Benz, illustré dans la Figure 1.11, combine plusieurs capteurs véhiculaires pour évaluer en temps réel le comportement du conducteur dès le début du trajet.

TABLEAU 1.2 – Comparaison des approches de détection de la somnolence.

Approche	Mécanisme de détection	Outils et techniques	Avantages
<i>Physiologique</i>	Utilisation de capteurs fixés sur le corps du conducteur.	Capteurs de rythme cardiaque et de clignement des yeux.	Mesure directe des changements biologiques.
<i>Vision par ordinateur</i>	Surveillance continue sans contact physique.	Caméras, LED infrarouges, modèles Dlib ou MediaPipe (CNN/LSTM).	Méthode non intrusive, surveillance continue et réaction rapide.
<i>Hybride</i>	Combinaison de plusieurs sources de données (visuelles, physiologiques, véhicule).	Vision & ML, Vision & DL, ou Physiologique & Vision.	Réduction des fausses alertes, robustesse accrue, précision optimale.

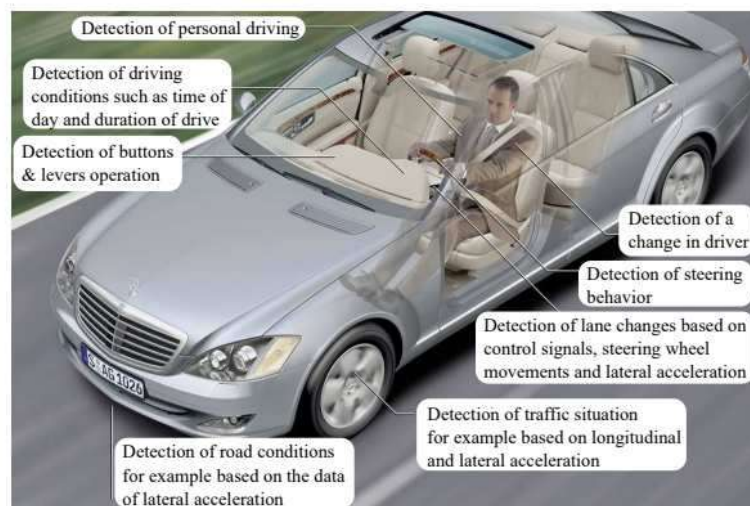


FIGURE 1.11 – Le système commercial Attention Assist et les signaux utilisés (Mercedes-Benz, 2008).

- **Systèmes DMS (Driver Monitoring Systems)** : chez les équipementiers Bosch ou Valeo, développement de caméras à l'intérieur de l'habitacle capables de détecter la distraction, la fatigue et la somnolence, et d'émettre des alertes ainsi que des mesures

pour renforcer la sécurité.

1.5 Bases de données publiques

L’utilisation de bases de données de référence constitue une étape essentielle dans le développement et la validation de tout nouveau système de détection de la somnolence, en permettant de comparer les performances obtenues à des données de test standardisées et reconnues par la communauté scientifique [48]. Ces bases de données jouent un rôle déterminant : elles fournissent les exemples nécessaires à l’entraînement des modèles, permettent d’évaluer leur capacité de généralisation face à des conditions variées (éclairage, morphologies, occlusions), et garantissent une comparaison équitable entre les différentes approches proposées dans la littérature.

1.5.1 UTA Real-Life Drowsiness Dataset (UTA-RLDD)

L’UTA-RLDD a été développé pour la détection à plusieurs niveaux de la somnolence. L’objectif principal de cet ensemble de données est de capturer les micro-expressions subtiles indiquant la fatigue, plutôt que les seuls signes extrêmes et facilement perceptibles de somnolence. Il comprend 60 participants en bonne santé qui ont enregistré un total de 30 heures de séquences vidéo RVB, en utilisant leurs téléphones personnels ou leurs webcams pour capturer leurs expressions faciales dans des scénarios réels. En raison de la nature physiologique et instinctive des expressions liées à la fatigue, les participants ont eu du mal à reproduire artificiellement les micro-expressions subtiles associées à la somnolence. Les données sont issues d’un corpus vidéo constitué de 60 sujets dans un état d’alerte, un état intermédiaire peu somnolent et un état très somnolent. Ces données sont intéressantes car le corpus aborde un grand nombre de conditions réelles [49].

1.5.2 NTHU-DDD Dataset (National Tsing Hua University)

L’ensemble de données NTHU-DDD, accessible au public, a été collecté par le laboratoire de vision par ordinateur (CV) de l’université nationale Tsing Hua. Il comprend 36 enregistrements vidéo infrarouges réalisés dans diverses conditions de conduite simulées, telles que la conduite normale, les bâillements lents, l’endormissement et le rire. Les vidéos ont été capturées sous des conditions d’éclairage diurnes et nocturnes, avec tous les scénarios impliquant une fatigue simulée [50].

1.5.3 YawDD Dataset (Yawning Detection Dataset)

Ensemble de données vidéo standard utilisé par les chercheurs pour la détection du bâillement, à partir de vidéos d’une caméra montée sous le rétroviseur du véhicule. Le YawDD, géré par le *Distributed & Collaborative Virtual Environments Research Laboratory* (DISCOVER Lab) de l’Université d’Ottawa, comprend deux sous-ensembles distincts. Le premier sous-ensemble comprend 322 vidéos montrant des expressions faciales normales, tandis que le second contient 29 vidéos de conducteurs en train de bâiller. Les deux sous-ensembles présentent un groupe diversifié de participants, comprenant des conducteurs et des conductrices de différentes origines ethniques, avec ou sans lunettes de vue ou de soleil [44].

1.5.4 Autres datasets pertinents

- **Driver Monitoring Dataset (DMD)** : Le jeu de données Driver Monitoring Dataset est le plus grand jeu de données visuelles sur les actions de conduite réelles, avec des séquences filmées par plusieurs caméras synchronisées (corps, visage, mains) et plusieurs flux (RVB, profondeur, IR) enregistrées dans deux scénarios (voiture réelle, simulateur de conduite). Différentes étiquettes annotées liées à la distraction, à la fatigue et à la position du regard et de la tête peuvent être utilisées pour entraîner des modèles d'apprentissage profond pour les systèmes de surveillance des conducteurs [51].
- **AUC-DDD dataset (AUC Distracted Driver Detection dataset)** : l'un des ensembles de données les plus connus sur la distraction au volant, créé par Abouelnaga et al. à l'Université américaine du Caire. L'ensemble de données contient au total 17 308 images provenant de 31 participants dans sept pays différents. Ces images ont été réparties de manière aléatoire entre un ensemble d'apprentissage (12 977 images) et un ensemble de validation (4 331 images). Tous les échantillons ont été collectés à partir de véhicules réels garés dans des zones sûres pendant le processus de collecte des données, chaque image ayant une résolution de 1920×1080 . L'ensemble de données est divisé en 10 classes représentant différentes postures des conducteurs, notamment la conduite en toute sécurité, parler au téléphone avec la main gauche ou droite, envoyer des SMS avec la main gauche ou droite, manger ou boire, se pencher en arrière, se recoiffer ou se remaquiller, régler la radio et parler aux passagers [52].
- **ULg Multimodality Drowsiness Database** : également appelée DROZY, est une base de données contenant divers types de données liées à la somnolence (signaux, images faciales, etc.) et destinée à aider les chercheurs à mener des expériences, ainsi qu'à développer et évaluer des systèmes (c'est-à-dire des algorithmes) dans le domaine de la surveillance de la somnolence [53].

1.6 Conclusion

Nous avons vu dans cet état de l'art que la tendance actuelle va vers des systèmes hybrides. Si les signaux physiologiques permettent de mesurer la vérité terrain, la vision par ordinateur est, de par sa non-invasivité, la solution actuellement la plus répandue en instrumentation automobile. Cela reste à nuancer avec des difficultés telles que l'occlusion (lunettes de soleil) ou la grande variabilité suivant les individus. Ce point servira de support aux développements de notre étude expérimentale dans les chapitres suivants.

Chapitre 2

Bases Théoriques du Système de Détection

2.1	Introduction	22
2.2	Intelligence artificielle	22
2.2.1	Machine Learning	22
2.3	Vision par ordinateur	23
2.3.1	Acquisition et traitement d'images	23
2.3.2	Détection faciale	24
2.4	MediaPipe : pipeline de détection faciale	24
2.4.1	Architecture BlazeFace	25
2.4.2	Face Mesh et les 468 points de repère	25
2.4.3	Calcul des métriques EAR et MAR	25
2.4.4	Avantages de l'approche MediaPipe	27
2.5	Apprentissage profond	28
2.5.1	Classification et régression	28
2.5.2	Apprentissage supervisé et non supervisé	28
2.5.3	Transfer Learning	29
2.5.4	Réseaux de neurones convolutifs (CNN)	30
2.5.5	Métriques d'évaluation (Accuracy, Precision, Recall, F1-Score)	34
2.6	Conclusion	35

2.1 Introduction

Dans ce chapitre, nous présentons le cadre théorique utilisé dans ce travail. Nous abordons les concepts fondamentaux liés à la vision par ordinateur, à l'apprentissage automatique et à l'apprentissage profond, qui constituent les bases scientifiques de notre étude. Nous décrivons également le framework MediaPipe, en détaillant son architecture et ses composantes principales, notamment BlazeFace et Face Mesh, ainsi que les indicateurs géométriques EAR et MAR exploités pour l'analyse de l'état de vigilance du conducteur.

2.2 Intelligence artificielle

L'intelligence artificielle (IA) fait référence à la simulation de l'intelligence humaine dans des machines programmées pour penser comme les humains et imiter leurs actions. Le terme peut également s'appliquer à toute machine qui présente des caractéristiques associées à l'esprit humain, comme l'apprentissage et la résolution de problèmes.

Pour atteindre l'intelligence artificielle, un système d'IA utilise principalement l'apprentissage automatique (Machine Learning) et d'autres techniques d'analyse de données.

2.2.1 Machine Learning

L'apprentissage automatique est un sous-domaine de l'intelligence artificielle. L'idée derrière l'apprentissage automatique est de créer une application qui apprend de son expérience. Pour que cette application puisse apprendre, un ensemble de données doit être fourni ; il n'est donc pas nécessaire de créer explicitement des algorithmes pour prédire le résultat de nouveaux échantillons de données.

L'apprentissage automatique peut être appliqué selon trois approches principales : supervisée, non supervisée ou semi-supervisée.

2.2.1.1 Apprentissage supervisé

L'apprentissage supervisé est couramment utilisé pour les problèmes de classification ou de régression. Lorsque les étiquettes sont représentées sous forme de catégories, le problème est un problème de classification. Lorsque les étiquettes sont représentées par une valeur numérique (un prix, une hauteur, etc.), le problème est un problème de régression.

2.2.1.2 Apprentissage non supervisé

Dans l'approche non supervisée, l'ensemble de données se compose uniquement de caractéristiques d'entrée sans étiquettes. Les algorithmes d'apprentissage automatique recherchent alors une structure ou des regroupements au sein des données.

2.2.1.3 Apprentissage semi-supervisé

L'apprentissage semi-supervisé utilise à la fois des données étiquetées et non étiquetées. En combinant ces deux types de données, les algorithmes peuvent apprendre à étiqueter les données non étiquetées.

2.3 Vision par ordinateur

La vision par ordinateur est une branche de l'intelligence artificielle qui repose sur l'apprentissage automatique et les réseaux de neurones, permettant aux ordinateurs de comprendre et d'analyser les images et les vidéos. Elle est utilisée pour traiter les données visuelles et prendre des décisions précises dans divers domaines, tels que le suivi des stocks dans les magasins, l'amélioration des systèmes de paiement automatisés, l'ajout de filtres aux photos et le contrôle de qualité dans les usines. Cette technologie peut également être employée dans les systèmes de détection de somnolence, notamment pour l'identification du visage et des yeux ainsi que l'analyse des signes de fatigue.

L'intérêt de la vision par ordinateur réside dans sa capacité à réaliser des tâches nécessitant une surveillance continue avec rapidité et précision, sans être affectée par la fatigue ou la pression humaine, comme l'analyse du trafic en temps réel pour détecter les embouteillages ou les accidents, ce qui reste difficile pour un observateur humain.

Pour concevoir un système de surveillance du conducteur efficace, la vision par ordinateur suit un pipeline précis allant de l'acquisition à la décision. Comme l'illustre la Figure 2.1, le flux commence par la capture vidéo via une caméra embarquée, suivie par la détection du visage et l'extraction des caractéristiques clés (yeux et bouche) afin de déclencher une alerte en cas de somnolence.

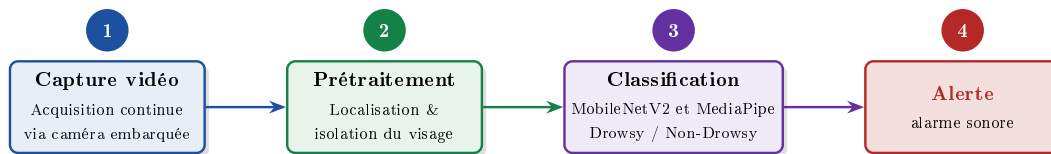


FIGURE 2.1 – Pipeline général du système de vision par ordinateur pour la surveillance du conducteur.

2.3.1 Acquisition et traitement d'images

Le processus d'acquisition d'images consiste à obtenir des données d'image numériques à partir de différentes sources telles que les caméras, les capteurs, les satellites, les scanners, etc. Ces données sont ensuite converties en image numérique, représentée sous forme d'une matrice bidimensionnelle de nombres appelée pixels, où chaque pixel correspond à la valeur numérique de l'intensité lumineuse à un emplacement donné. Après l'acquisition, l'image est traitée à l'aide de techniques de traitement numérique afin d'éliminer les distorsions, les erreurs et le bruit pouvant apparaître lors de la collecte des données, ainsi que d'améliorer la qualité des images numériques et leur interprétabilité pour l'utilisateur. Le système de traitement d'images comprend des éléments essentiels tels que la source des données, l'unité de traitement (ordinateur), ainsi que la destination des résultats, qui peut être un écran ou tout autre dispositif d'affichage de l'image traitée.

2.3.2 Détection faciale

Dans le domaine de la vision par ordinateur, les études se concentrent sur la localisation précise des visages dans des environnements complexes.

La détection de visages consiste à repérer la zone de l'image contenant un visage et à déterminer sa position (voir la Figure 2.2). Cette étape permet d'extraire une région d'intérêt (ROI) utilisée pour la suite du traitement, tout en éliminant les éléments non pertinents comme l'arrière-plan ou les cheveux.

Dans certains travaux, l'extraction des yeux est utilisée afin d'améliorer la localisation du visage dans des scènes complexes. Les systèmes de détection utilisent des caractéristiques visuelles pour isoler la région du visage. Parmi les méthodes utilisées, le modèle SSD (Single Shot MultiBox Detector) permet de détecter le visage en une seule passe du réseau, avec rapidité et précision, ce qui le rend adapté aux applications en temps réel [54]. De même, YOLO (You Only Look Once) est une architecture de détection en temps réel qui traite l'image en une seule passe, divisant celle-ci en une grille de cellules dont chacune prédit simultanément les boîtes englobantes et les probabilités de classe, offrant ainsi un excellent compromis entre vitesse et précision [55].

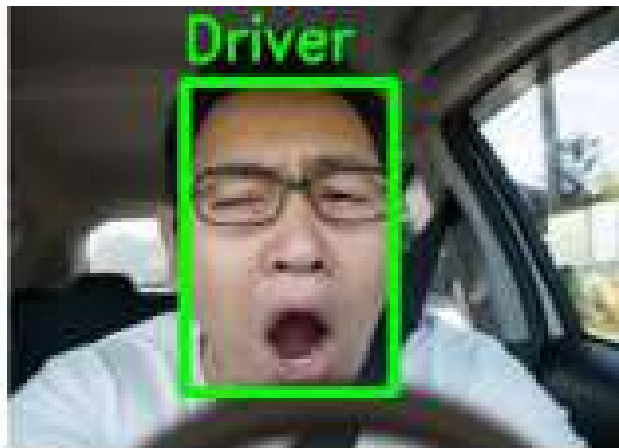


FIGURE 2.2 – Localisation du visage et extraction de la région d'intérêt (ROI) pour l'analyse du visage du conducteur.

2.4 MediaPipe : pipeline de détection faciale

MediaPipe est un framework de vision par ordinateur développé par Google pour le traitement des images et des vidéos en temps réel [56]. Il permet de détecter rapidement et efficacement les visages dans une image ou un flux vidéo, tout en conservant une faible consommation de ressources.

Son pipeline repose sur deux étapes principales. La première consiste à utiliser le détecteur BlazeFace pour localiser la position du visage à l'aide d'une boîte englobante. La seconde étape fait appel à Face Mesh afin d'extraire les points de repère faciaux permettant de décrire avec précision la géométrie du visage.

Grâce à cette architecture, MediaPipe offre une solution performante et adaptée aux appli-

cations nécessitant une détection faciale en temps réel sur des appareils mobiles ou embarqués.

2.4.1 Architecture BlazeFace

BlazeFace est un détecteur de visages léger conçu pour répondre aux contraintes des applications mobiles en temps réel. Son architecture repose sur un réseau de neurones convolutif (CNN) optimisé pour les GPU mobiles, garantissant un excellent compromis entre précision et rapidité d'exécution.

Le modèle utilise un extracteur de caractéristiques inspiré de MobileNet et basé sur des convolutions séparables en profondeur, ce qui réduit considérablement le nombre de paramètres et le coût computationnel.

BlazeFace adopte également un schéma d'ancrage optimisé spécialement conçu pour la détection des visages, permettant une localisation rapide et précise des régions faciales. Grâce à ces optimisations, le modèle atteint des performances élevées tout en conservant une taille réduite, ce qui en fait la composante principale du module de détection faciale de MediaPipe.

2.4.2 Face Mesh et les 468 points de repère

MediaPipe Face Mesh est une solution de suivi facial avancée permettant d'estimer la géométrie tridimensionnelle du visage en temps réel. Contrairement à la simple détection faciale, cette approche génère un maillage facial dense composé de 468 points de repère répartis sur l'ensemble du visage, notamment autour des yeux, des sourcils, du nez, des lèvres et du contour facial.

Chaque point est défini par des coordonnées normalisées (x, y) ainsi qu'une coordonnée z représentant la profondeur relative, ce qui permet de reconstruire une représentation 3D du visage et de suivre avec précision ses mouvements et ses expressions.

La structure fixe du maillage facilite l'identification automatique de régions faciales spécifiques. Dans le cadre de la détection de la somnolence, les points correspondant aux yeux et à la bouche peuvent être exploités pour calculer des indicateurs géométriques tels que l'EAR (Eye Aspect Ratio) et le MAR (Mouth Aspect Ratio), utilisés pour évaluer l'état de vigilance du conducteur.

La Figure 3.6 illustre les quatre étapes du pipeline MediaPipe appliqué à des images du dataset : l'image originale, la détection du visage par BlazeFace avec sa boîte englobante, l'extraction des 468 points de repère faciaux, et enfin le maillage facial généré par Face Mesh.

2.4.3 Calcul des métriques EAR et MAR

Afin d'évaluer l'état de vigilance du conducteur, le système utilise les points de repère faciaux extraits par MediaPipe Face Mesh pour calculer deux indicateurs géométriques : l'EAR (Eye Aspect Ratio) et le MAR (Mouth Aspect Ratio). Ces métriques permettent d'analyser respectivement l'ouverture des yeux et de la bouche, deux paramètres étroitement liés aux signes de fatigue et de somnolence.

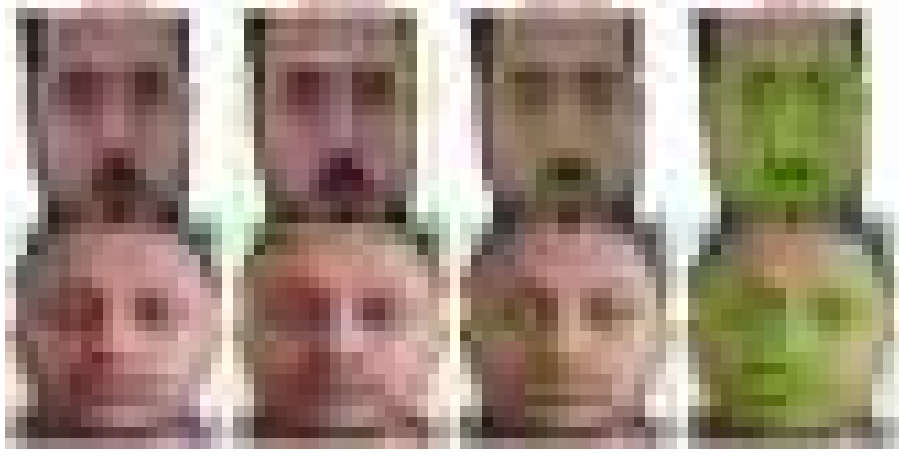


FIGURE 2.3 – Pipeline de détection MediaPipe : image originale, détection BlazeFace et extraction des landmarks faciaux

2.4.3.1 Eye Aspect Ratio (EAR)

L'EAR est une mesure géométrique utilisée pour détecter les clignements ainsi que la fermeture prolongée des yeux. Il est calculé à partir de six points de repère localisés autour de chaque œil selon la formule suivante :

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2 \times \|p_1 - p_4\|} \quad (2.1)$$

où p_1 et p_4 représentent les points horizontaux de l'œil, tandis que p_2 , p_3 , p_5 et p_6 correspondent aux points verticaux.

Lorsque l'œil est ouvert, la valeur de l'EAR reste relativement stable. En revanche, lors d'un clignement ou d'une fermeture prolongée des yeux, cette valeur diminue considérablement. Ainsi, un EAR inférieur à un seuil prédéfini pendant plusieurs images consécutives peut indiquer un état de somnolence.

2.4.3.2 Mouth Aspect Ratio (MAR)

La méthode MAR est utilisée pour détecter le bâillement, un indicateur important de la somnolence du conducteur. Cette approche repose sur le calcul des distances euclidiennes entre les points de repère faciaux situés autour de la bouche.

La formule du MAR est donnée par :

$$MAR = \frac{\|p_2 - p_8\| + \|p_3 - p_7\| + \|p_4 - p_6\|}{3 \times \|p_5 - p_1\|} \quad (2.2)$$

Cette équation établit le rapport entre la hauteur et la largeur de la bouche.

Les points utilisés mettent en évidence les deux zones d'intérêt suivantes :

- Zone d'intérêt de la bouche fermée.
- Zone d'intérêt de la bouche ouverte (bâillement).

Cette mesure permet au système de distinguer précisément les différentes phases d'ouverture de la bouche, améliorant ainsi la fiabilité de la détection de la fatigue en temps réel.

L'utilisation conjointe de l'EAR et du MAR permet ainsi de surveiller simultanément les

mouvements des yeux et de la bouche. Cette approche fournit une évaluation plus robuste de l'état de vigilance du conducteur et améliore les performances du système de détection de la somnolence en temps réel. La Figure 2.4 présente des exemples de calcul de l'EAR et du MAR, illustrant la variation de ces indicateurs entre un conducteur somnolent et un conducteur éveillé.



FIGURE 2.4 – Calcul des indicateurs EAR et MAR : comparaison entre conducteur somnolent et conducteur éveillé

2.4.4 Avantages de l'approche MediaPipe

L'intégration du pipeline MediaPipe au sein d'un système embarqué constitue un choix stratégique qui répond aux contraintes opérationnelles de la détection de la somnolence. Cette approche offre quatre avantages majeurs :

- **Temps réel et latence réduite** : En effectuant l'ensemble du traitement en local (Edge Computing), le système s'affranchit des contraintes liées à la bande passante et aux délais de transmission, garantissant ainsi une réactivité immédiate indispensable pour la sécurité routière.
- **Faible consommation énergétique** : Grâce à l'architecture légère de BlazeFace, optimisée par des convolutions séparables en profondeur, la charge computationnelle est minimisée. Cela permet une exécution fluide sur des dispositifs aux ressources limitées tout en optimisant la consommation électrique.
- **Protection de la vie privée** : La nature locale du traitement garantit qu'aucune donnée visuelle n'est transmise vers des serveurs externes. Cette architecture assure une confidentialité totale des données du conducteur, un critère essentiel pour l'acceptation et la conformité du système.
- **Robustesse et autonomie** : Le système opère en totale autonomie, sans dépendance vis-à-vis d'une connexion réseau. Cette fiabilité opérationnelle assure un fonctionnement continu et stable, même dans des conditions de conduite isolées ou instables.

2.5 Apprentissage profond

L'apprentissage profond (Deep Learning) est une branche de l'intelligence artificielle qui repose sur des réseaux de neurones multicouches capables d'apprendre automatiquement des représentations hiérarchiques à partir des données. Contrairement aux approches classiques du Machine Learning, l'apprentissage profond ne nécessite pas d'extraction manuelle des caractéristiques, ce qui le rend particulièrement adapté au traitement des données visuelles complexes telles que les images et les vidéos.

Un modèle de deep learning est conçu pour analyser continuellement les données avec une structure logique similaire à la façon dont un humain tirerait des conclusions. Pour effectuer cette analyse, les applications d'apprentissage profond utilisent une structure d'algorithmes en couches appelée réseau neuronal artificiel, dont la conception s'inspire du réseau biologique de neurones du cerveau humain.

2.5.1 Classification et régression

Les problèmes d'apprentissage supervisé se divisent en deux types selon la nature de la sortie comme le montre la Figure 2.5. La régression est utilisée lorsque la variable de sortie est continue. Dans le domaine de la sécurité routière, elle est appliquée par exemple pour estimer le niveau de fatigue du conducteur sous forme de score numérique, ou pour prédire le temps de réaction avant un incident. La classification consiste quant à elle à attribuer une entrée à l'une des catégories prédéfinies. Elle est largement utilisée dans les systèmes embarqués de surveillance du conducteur, notamment pour distinguer l'état somnolent de l'état éveillé, ou pour classer les expressions faciales en différentes catégories d'alerte. Dans notre travail, la détection de la somnolence est formulée comme une classification binaire : le modèle MobileNetV2 prédit si le conducteur est somnolent ou non somnolent à partir des images capturées en temps réel.

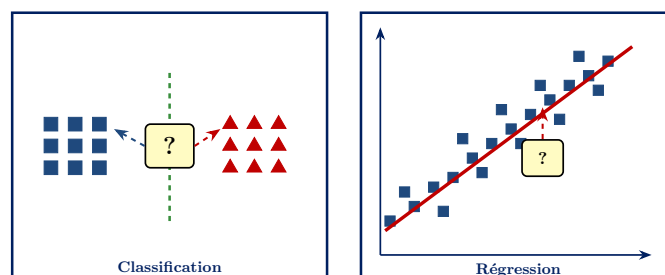


FIGURE 2.5 – Classification vs régression

2.5.2 Apprentissage supervisé et non supervisé

L'apprentissage profond se divise généralement en deux catégories principales selon la nature des données utilisées. Dans l'apprentissage supervisé, le modèle est entraîné à partir de données étiquetées, c'est-à-dire que chaque image est associée à une classe connue. Par exemple, dans les systèmes de détection de somnolence, les images du conducteur sont annotées comme somnolent ou non somnolent, et dans les systèmes de détection de collision, les images sont éti-

quetées selon la présence ou l'absence d'obstacle. Dans l'apprentissage non supervisé, le modèle apprend à partir de données non étiquetées en identifiant des structures ou des regroupements au sein des données. La Figure 2.6 illustre ce principe. Dans le domaine de la sécurité routière, cette approche est utilisée par exemple pour regrouper automatiquement des comportements de conduite similaires ou pour détecter des anomalies dans le flux vidéo sans annotation préalable.

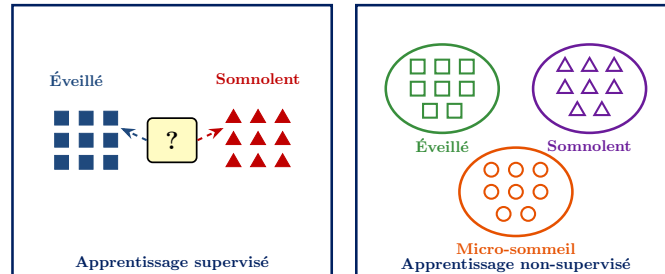


FIGURE 2.6 – Apprentissage supervisé vs non-supervisé

2.5.3 Transfer Learning

Le transfer learning est une technique d'apprentissage profond qui consiste à réutiliser un modèle préentraîné sur un large dataset, tel qu'ImageNet, et à l'adapter à une nouvelle tâche spécifique, Comme illustré dans la Figure 2.7.

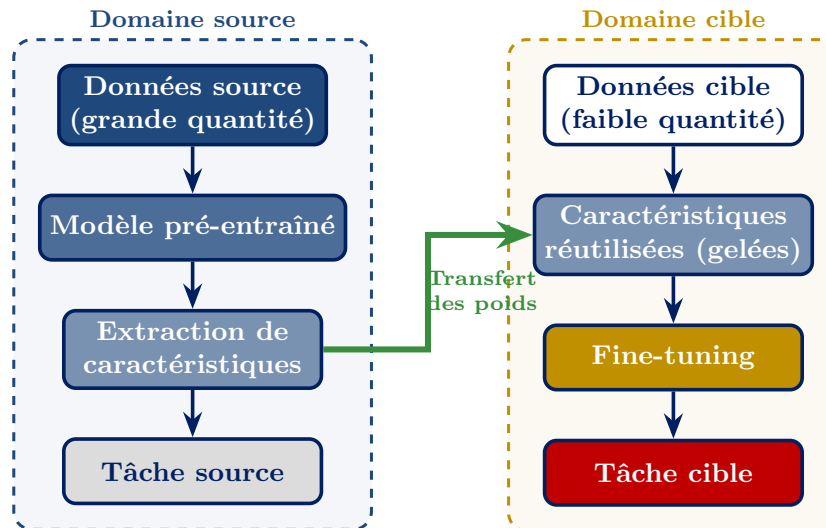


FIGURE 2.7 – Principe du transfert d'apprentissage

Plutôt que d'entraîner un réseau depuis zéro, ce qui nécessite une grande quantité de données et un coût computationnel élevé, le transfer learning permet de bénéficier des représentations déjà apprises par le modèle et de les affiner sur le dataset cible. Dans le domaine de la sécurité routière et de la détection de la somnolence, cette approche est particulièrement adaptée car les datasets disponibles sont souvent de taille limitée. Par exemple, des modèles préentraînés comme VGG16 ont été utilisés pour la détection des émotions du conducteur, ResNet50 pour

la classification des comportements de conduite dangereux, et MobileNetV2 pour la détection de la somnolence sur des systèmes embarqués à ressources limitées.

2.5.4 Réseaux de neurones convolutifs (CNN)

Les réseaux de neurones profonds constituent aujourd'hui des outils principaux utilisés dans divers domaines scientifiques. Le réseau de neurones convolutif (CNN) est un type particulier de réseau de neurones profonds, composé de plusieurs couches (couche d'entrée, couches cachées, couche de sortie), utilisées pour le traitement des données, chacune suivie d'une fonction d'activation, puis d'une couche de pooling.

La Figure 2.8 illustre l'architecture des réseaux de neurones convolutifs (CNN). Une image est introduite dans le modèle, puis elle passe à travers plusieurs couches de convolution afin d'extraire les caractéristiques importantes. Ensuite, ces caractéristiques sont traitées par d'autres couches, et enfin, l'image est classifiée dans une catégorie donnée.

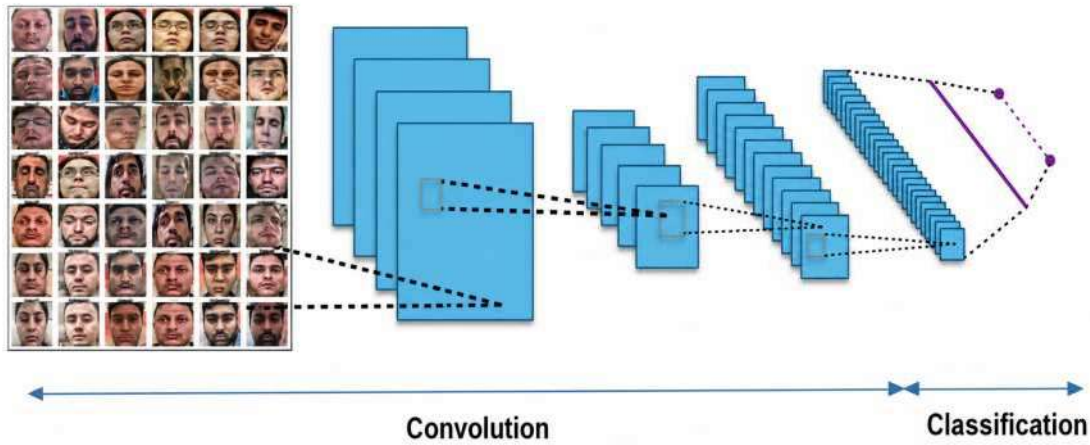


FIGURE 2.8 – Architecture des réseaux de neurones convolutifs (CNN).

2.5.4.1 Architecture de base

Les réseaux de neurones artificiels s'inspirent directement du fonctionnement biologique du cerveau humain. Un neurone formel est modélisé comme une fonction mathématique à plusieurs variables réelles, où les connexions d'entrée simulent les dendrites et la sortie unique représente l'axone. Le modèle le plus élémentaire de cette catégorie est le **perceptron**, introduit en 1957 par F. Rosenblatt (voir Figure 2.9), constitué des éléments suivants :

- **Des entrées** ($x_1, x_2, \dots, x_n$) : qui représentent les signaux ou les caractéristiques provenant des données initiales.
- **Des poids associés** ($w_{j1}, w_{j2}, \dots, w_{jn}$) : qui pondèrent l'importance de chaque entrée lors de la transmission.
- **Un biais** (b_j) : un paramètre ajustable permettant de réguler le seuil d'activation indépendamment des entrées.
- **Une fonction d'activation** $\Phi(\cdot)$: qui introduit la non-linéarité dans le modèle.
- **Une sortie** (y_j) : le résultat final calculé par le neurone.

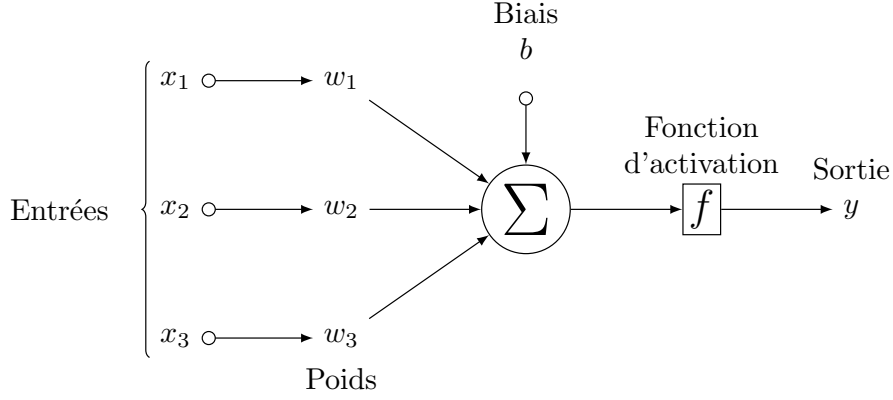


FIGURE 2.9 – Modèle mathématique d'un neurone artificiel (perceptron de Rosenblatt) avec ses poids et son biais.

Comme l'illustre la Figure 2.9, le traitement de l'information s'effectue de manière séquentielle. Chaque signal d'entrée x_i est d'abord multiplié par son poids synaptique correspondant w_{ji} . Le système calcule ensuite la somme pondérée de ces entrées, enrichie par le terme de biais b_j . Enfin, cette somme globale, appelée combinaison linéaire, est soumise à la fonction d'activation non linéaire $\Phi(\cdot)$ pour générer la sortie finale y_j .

2.5.4.2 Couches de convolution, pooling et activation

Couches de convolution : Ces réseaux reposent sur des filtres de convolution (matrices numériques) appliqués aux données d'entrée avant leur transmission aux neurones (voir la Figure 2.10). Le masque de filtrage K glisse sur l'ensemble de l'image d'entrée I et estime le produit scalaire entre les poids des filtres et la valeur de l'image, produisant une carte de caractéristiques F donnée par :

$$F(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n) \quad (2.3)$$

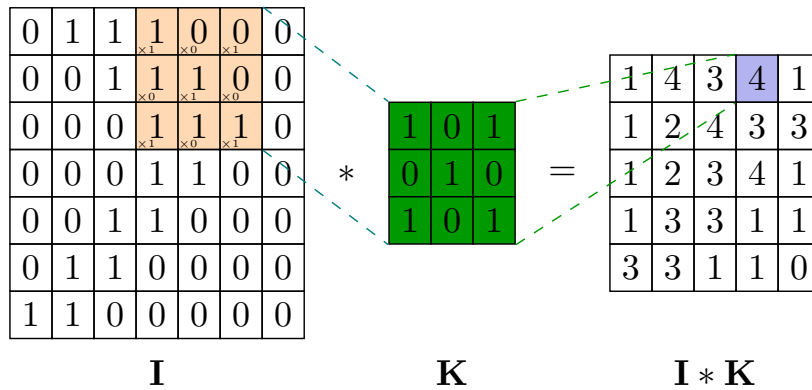


FIGURE 2.10 – Opération de convolution bidimensionnelle entre la matrice d'entrée I et le noyau de filtrage K

Pooling : La couche de pooling réduit la taille des données pour faciliter le calcul et éviter que le réseau ne mémorise trop les détails inutiles (surapprentissage). Elle conduit à une

perte maîtrisée d'informations qui aide le réseau à réduire à la fois l'overfitting et la charge de calcul. Deux méthodes sont principalement utilisées. La méthode du Max Pooling, qui retient la valeur maximale dans chaque fenêtre glissante, est la plus couramment utilisée en raison de ses excellentes performances (voir la Figure 2.11). La méthode Average Pooling, quant à elle, calcule la moyenne de toutes les valeurs contenues dans la fenêtre glissante (voir la Figure 2.12).

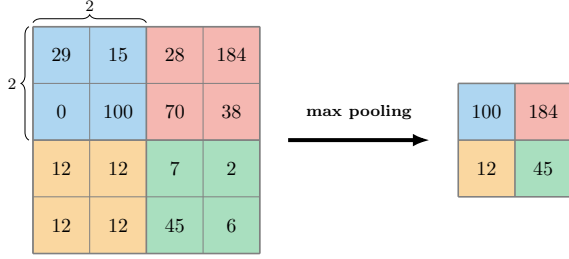


FIGURE 2.11 — Principe du sous-échantillonnage par la méthode du Max Pooling (filtre 2×2)

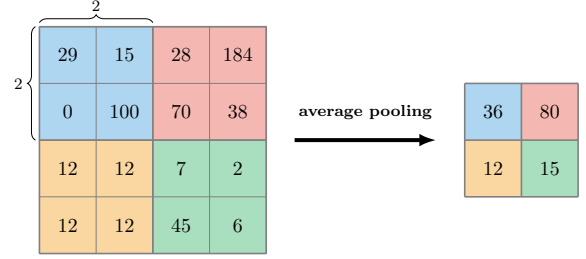


FIGURE 2.12 — Principe du sous-échantillonnage par la méthode de l'Average Pooling (filtre 2×2)

Fonctions d'activation : Elles permettent au réseau d'apprendre des relations complexes dans les données, en lui conférant une capacité d'expression non linéaire. Les fonctions d'activation les plus utilisées sont les suivantes (voir la Figure 2.13) :

- **Sigmoid** : Réduit la valeur d'entrée entre 0 et 1 :

$$F(x) = \frac{1}{1 + e^{-x}} \quad (2.4)$$

- **Tanh** : Produit un résultat compris entre -1 et 1 :

$$F(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.5)$$

- **ReLU (Rectified Linear Unit)** : Fonction la plus couramment utilisée en apprentissage profond :

$$F(x) = \max(0, x) \quad (2.6)$$

- **Softmax** : Utilisée en dernière couche pour la classification multi-classes ; elle convertit un vecteur de scores en probabilités :

$$F(x_i) = \frac{e^{x_i}}{\sum_p e^{x_p}} \quad (2.7)$$

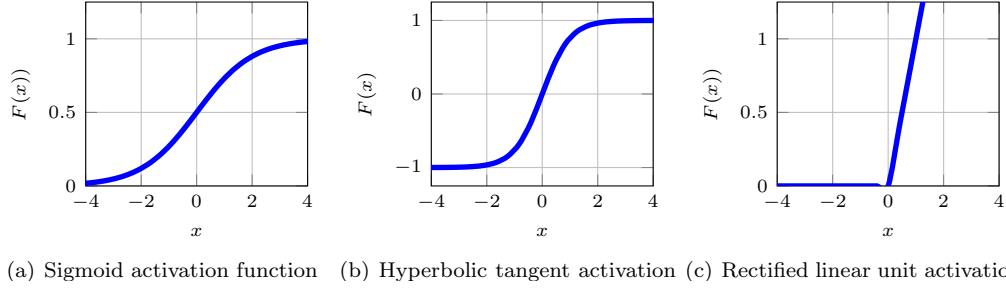


FIGURE 2.13 – Représentation graphique des principales fonctions d'activation non linéaires

Dropout : est une technique de régularisation visant à réduire le surapprentissage dans les réseaux neuronaux artificiels en empêchant les co-adaptations complexes sur les données d'apprentissage (voir la Figure 2.14). Cette technique consiste à mettre aléatoirement à zéro les sorties de certains neurones pendant le processus d'apprentissage, réduisant ainsi la taille effective du modèle et améliorant sa capacité de généralisation. Elle n'est pas appliquée lors de la phase d'inférence.

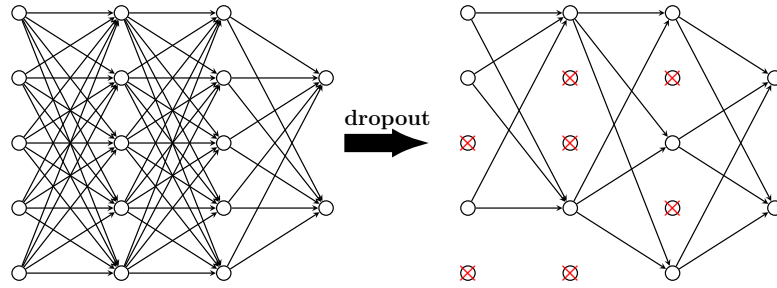


FIGURE 2.14 – Concept de la régularisation par Dropout pour la réduction du surapprentissage (Overfitting)

2.5.4.3 Processus d'apprentissage

Le processus d'ajustement de la valeur des poids est défini comme l'apprentissage (*training*) du réseau neuronal. Le CNN démarre avec des poids aléatoires et est alimenté par un ensemble de données étiquetées. L'apprentissage s'effectue en deux étapes :

- **Forward pass** : détermination d'une prédiction et calcul de l'erreur par rapport à la valeur de référence.
- **Backward pass** : mise à jour des poids des neurones en fonction de l'erreur, via l'algorithme d'optimisation choisi.

Différentes fonctions de coût sont utilisées, notamment l'erreur quadratique moyenne :

$$L = \frac{1}{n} \sum_{i=1}^n (y^i - Y^i)^2 \quad (2.8)$$

et l'entropie croisée :

$$L = \frac{1}{n} \sum_{i=1}^n [y^i \log(Y^i) + (1 - y^i) \log(1 - Y^i)] \quad (2.9)$$

où Y est la prédiction du neurone, y est la valeur de référence et n est le nombre de prédictions.

L'optimisation repose sur la descente de gradient stochastique (SGD) et ses variantes telles qu'Adam et RMSprop. Les hyperparamètres clés du processus d'apprentissage sont le nombre d'époques (nombre de passages complets sur le dataset) et la taille des lots (*batch size*).

2.5.4.4 Architectures célèbres

VGG16 : VGG16 a été développée par Simonyan et Zisserman et a obtenu la deuxième place dans la compétition ILSVRC 2014. Elle se compose de 16 couches convolutionnelles avec une structure uniforme et utilise des convolutions 3×3 pour extraire les caractéristiques des images. Elle repose sur des opérations de convolution suivies de couches de max pooling et de couches entièrement connectées jusqu'à la couche de sortie. Toutes les couches utilisent un stride égal à 1, sans padding. Elle contient environ 138 millions de paramètres, ce qui la rend volumineuse et peut entraîner un surapprentissage [57].

ResNet : ResNet a été introduite par Kaiming He et al. en 2015. Elle repose sur des connexions résiduelles (skip connections) et permet de construire des réseaux très profonds allant jusqu'à 152 couches avec une complexité réduite par rapport à VGGNet. Elle a obtenu de très bonnes performances et est utilisée pour améliorer la vitesse de convergence et les performances des modèles [58].

MobileNet : L'architecture de MobileNet repose sur des convolutions séparables en profondeur (depthwise separable convolutions). Elle utilise deux hyperparamètres principaux, le multiplicateur de largeur (width multiplier) et le multiplicateur de résolution (resolution multiplier), afin de réduire la taille du modèle [59,60].

Le Tableau 2.1 résume et compare ces trois architectures selon leur nombre de paramètres, leur profondeur et leur principal domaine d'application.

TABLEAU 2.1 – Comparaison des architectures CNN de référence.

Architecture	Paramètres	Profondeur	Caractéristiques principales
VGG16	138 M	16 couches	Convolutions 3×3 uniformes, stride = 1 ; volumineuse et sujette au sur-apprentissage.
ResNet-50	25,6 M	50 couches	Connexions résiduelles (skip connections), blocs bottleneck 1×1 - 3×3 - 1×1 , bonne convergence.
MobileNetV2	3,4 M	53 couches	Résidus inversés avec goulots linéaires, 19 blocs bottleneck ; plus léger et plus précis que V1.

2.5.5 Métriques d'évaluation (Accuracy, Precision, Recall, F1-Score)

Après la construction d'un modèle de classification, plusieurs métriques sont utilisées afin de vérifier la qualité des résultats et la performance de l'algorithme. Ces métriques reposent

sur la matrice de confusion (voir Figure 2.15), qui résume les prédictions du modèle en quatre catégories. Le TP (True Positives) représente les instances correctement classées positives, TN (True Negatives) représente les instances correctement classées négatives, FP (False Positives) représente les instances négatives classées à tort comme positives, et FN (False Negatives) représente les instances positives classées à tort comme négatives.

Classe prédite	
Classe réelle	Positive
	Négative
Positive	TP
Négative	TN

FIGURE 2.15 – Matrice de confusion pour un problème de classification binaire.

La précision globale (Accuracy) est une mesure utilisée pour évaluer les performances des modèles de classification. Elle représente la proportion des prédictions correctement réalisées par le modèle. Elle est définie comme suit :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.10)$$

Le rappel (Recall) est défini comme la proportion des instances positives correctement détectées parmi l'ensemble des instances positives réelles :

$$Recall = \frac{TP}{TP + FN} \quad (2.11)$$

La précision (Precision) correspond à la proportion des instances positives correctement prédites parmi l'ensemble des instances prédites positives :

$$Precision = \frac{TP}{TP + FP} \quad (2.12)$$

Enfin, le F1-score est une mesure qui combine la précision et le rappel en calculant leur moyenne harmonique :

$$F1 = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (2.13)$$

2.6 Conclusion

Dans ce chapitre, nous avons présenté les concepts théoriques essentiels à la compréhension du système développé. Nous avons d'abord introduit les notions fondamentales de l'intelligence artificielle et du Machine Learning, avant d'expliquer le rôle de la vision par ordinateur et de

l'apprentissage profond dans l'analyse d'images et la détection de la somnolence. Nous avons également décrit le framework MediaPipe, notamment son architecture BlazeFace pour la détection du visage et Face Mesh pour l'extraction des 468 points de repère faciaux, ainsi que les indicateurs géométriques EAR et MAR utilisés pour évaluer l'état de vigilance du conducteur. Enfin, nous avons détaillé l'architecture des réseaux de neurones convolutifs, leurs composantes clés (convolution, pooling, activation, dropout) ainsi que les métriques d'évaluation employées.

Ces fondements théoriques constituent la base nécessaire pour comprendre et mettre en œuvre les méthodes de détection faciale et les techniques de deep learning appliquées dans ce travail. Ils permettront au chapitre suivant de détailler la conception et l'implémentation du système.

Chapitre 3

Conception du système de détection

3.1	Introduction	38
3.2	Architecture globale du système	38
3.2.1	Vue d'ensemble comparative des deux approches	38
3.2.2	Pipeline Méthode 1 : MobileNetV2	38
3.2.3	Pipeline Méthode 2 : MediaPipe	39
3.3	Méthode 1 : approche Deep Learning (MobileNetV2)	39
3.3.1	Dataset : Driver Drowsiness Dataset (DDD)	39
3.3.2	Prétraitement des images	41
3.3.3	Transfer Learning et architecture MobileNetV2	41
3.3.4	Entraînement et résultats	42
3.3.5	Optimisation : Quantification TensorFlow Lite	44
3.4	Méthode 2 : approche Landmarks (MediaPipe)	46
3.4.1	Pipeline de détection faciale	46
3.4.2	Logique de décision	47
3.4.3	Génération des alertes	47
3.4.4	Tests en conditions réelles (<i>In the Wild</i>)	48
3.5	Comparaison des deux méthodes	49
3.6	Conclusion	50

3.1 Introduction

Dans ce chapitre, nous présentons les différentes étapes suivies pour le développement de notre système de détection de la somnolence au volant. Pour cela, nous nous sommes appuyés sur deux approches principales : MobileNetV2 et MediaPipe. MobileNetV2 a été utilisé comme modèle de classification basé sur l'apprentissage profond et le transfert d'apprentissage afin de détecter l'état du conducteur à partir des images. Quant à MediaPipe, il a été employé pour la détection du visage et l'extraction des points de repère faciaux (*Face Landmarks*) en temps réel, permettant ainsi l'analyse de l'état des yeux et des expressions du visage. Nous aborderons également les différentes étapes de prétraitement des données, l'entraînement et l'évaluation des modèles, ainsi que les techniques d'optimisation mises en œuvre pour obtenir un système performant et adapté à une utilisation en temps réel.

3.2 Architecture globale du système

3.2.1 Vue d'ensemble comparative des deux approches

Afin de garantir une détection fiable et robuste de la somnolence au volant, nous avons conçu et évalué deux approches complémentaires. La première, basée sur l'architecture MobileNetV2, repose sur un modèle de Deep Learning entraîné sur un large jeu de données d'images faciales de conducteurs. La seconde, basée sur le framework MediaPipe, exploite une analyse géométrique du visage en calculant des métriques morphologiques telles que l'EAR et le MAR sans nécessiter de phase d'entraînement. Le Tableau 3.1 résume les caractéristiques principales de chacune.

TABLEAU 3.1 – Vue d'ensemble comparative des deux approches.

Critère	MobileNetV2	MediaPipe
Type	Deep Learning	Règles géométriques
Entrée	Images labellisées	Flux vidéo
Entraînement	Oui (Kaggle)	Non
Décision	Classificateur CNN	Seuil EAR/MAR
Optimisation	TensorFlow Lite	Fenêtre glissante
Déploiement	Systèmes embarqués	Raspberry Pi / PC

3.2.2 Pipeline Méthode 1 : MobileNetV2

Le pipeline de la première méthode s'articule autour de quatre étapes séquentielles :

1. **Capture vidéo** : Acquisition continue des images du conducteur, image par image, à l'aide de la caméra embarquée.
2. **Prétraitement** : Redimensionnement à 224×224 pixels et normalisation des images dans l'intervalle $[-1, 1]$.
3. **Classification** : Transmission de l'image du visage au modèle MobileNetV2 pour analyser les traits faciaux et classer l'état du conducteur : Non-Drowsy (vigilant) ou Drowsy

(somnolent).

4. **Alerte** : Si l'état de somnolence est détecté de manière répétée sur plusieurs frames consécutives, le système déclenche automatiquement une **alarme sonore**.

3.2.3 Pipeline Méthode 2 : MediaPipe

Le pipeline de la seconde méthode repose sur une analyse géométrique en temps réel :

1. **Capture vidéo** : Acquisition continue du flux vidéo du conducteur.
2. **Détection faciale** : Localisation du visage via l'architecture BlazeFace intégrée à MediaPipe.
3. **Extraction des landmarks** : Détection de 468 points de repère faciaux pour identifier précisément les régions des yeux et de la bouche.
4. **Calcul des métriques** : Calcul des ratios EAR et MAR à partir des coordonnées des landmarks.
5. **Décision et alerte** : Application d'un seuil sur les valeurs EAR/MAR combiné à un filtrage temporel sur une fenêtre glissante de 13 frames pour déclencher l'alerte.

3.3 Méthode 1 : approche Deep Learning (MobileNetV2)

3.3.1 Dataset : Driver Drowsiness Dataset (DDD)

Le Driver Drowsiness Dataset (DDD) est constitué d'images de visages de conducteurs extraites du Real-Life Drowsiness Dataset (voir Figure 3.1) [61]. Les vidéos ont été converties en images à l'aide du logiciel VLC, puis les visages ont été détectés grâce à l'algorithme de Viola-Jones afin d'extraire la région d'intérêt. Ce dataset a été utilisé pour entraîner et tester des modèles de Deep Learning destinés à la détection de la somnolence du conducteur.



FIGURE 3.1 – Exemples d’images de visages de conducteurs extraites du Real-Life Drowsiness Dataset (DDD).

Le DDD est composé d’images faciales de conducteurs réparties en deux catégories principales : les conducteurs somnolents (Drowsy) et les conducteurs vigilants (Non-Drowsy) [61]. Les images sont enregistrées au format RGB avec une résolution originale de 227×227 pixels.

Le dataset contient plus de 41 790 images pour une taille totale d’environ 2,32 Go, ce qui permet d’assurer un apprentissage robuste et une bonne capacité de généralisation du modèle de Deep Learning. Afin d’évaluer efficacement les performances du système, le dataset a été divisé en trois sous-ensembles :

- **Ensemble d’entraînement (Train)** : 29 255 images.
- **Ensemble de validation (Validation)** : 6 269 images.
- **Ensemble de test (Test)** : 6 269 images.

Cette répartition permet d’assurer un entraînement optimal du modèle ainsi qu’une évaluation fiable de ses performances sur des données non vues durant l’apprentissage.

Comme illustré dans la Figure 3.2, la base de données présente une répartition équilibrée entre les deux classes. La classe Drowsy représente 53,5% des images, tandis que la classe Non-Drowsy en représente 46,5%. Cette parité est cruciale pour garantir la robustesse du modèle MobileNetV2 et éviter tout biais algorithmique lors de l’entraînement.



FIGURE 3.2 – Répartition des classes dans le Driver Drowsiness Dataset.

3.3.2 Prétraitement des images

Avant l’entraînement du modèle, les images ont subi plusieurs étapes de prétraitement afin d’optimiser la qualité des données et d’adapter le modèle aux contraintes du temps réel :

- **Redimensionnement** : Bien que l’architecture MobileNetV2 soit initialement conçue pour des images d’entrée de 227×227 pixels, nous avons choisi de redimensionner nos images en 224×224 pixels. Ce choix stratégique permet de réduire considérablement la taille des données, d’alléger la charge de calcul et d’accélérer la vitesse d’inférence, ce qui est idéal pour un déploiement efficace sur des systèmes embarqués aux ressources limitées.
- **Normalisation** : Les images ont été normalisées à l’aide de la méthode de prétraitement recommandée pour l’architecture MobileNet. Cette normalisation est particulièrement adaptée à ce modèle, car MobileNet a été conçu et entraîné à l’origine avec des valeurs de pixels ramenées dans l’intervalle $[-1, 1]$. Cela permet d’améliorer la stabilité de l’apprentissage, de favoriser une convergence plus rapide et d’assurer une meilleure compatibilité avec le réseau utilisé. L’équation de normalisation appliquée est :

$$x_{norm} = \frac{x - 127.5}{127.5} \quad (3.1)$$

où x représente la valeur originale du pixel et x_{norm} la valeur normalisée.

3.3.3 Transfer Learning et architecture MobileNetV2

L’apprentissage par transfert (*Transfer Learning*) est devenu une stratégie incontournable en Deep Learning. Au lieu de concevoir et d’entraîner un modèle à partir de zéro, ce qui s’avère extrêmement lourd en termes de calculs et de temps, cette approche permet de réutiliser les connaissances déjà acquises par un réseau pré-entraîné sur une base de données massive.

Le concept est simple : les caractéristiques visuelles globales extraites lors d’une première tâche sont transférées pour résoudre un nouveau problème similaire. Dans notre cas, cela évite d’avoir à collecter et labelliser des millions d’images, une tâche qui serait fastidieuse et chronophage.

En pratique, lors de l'introduction de nos propres données, les poids du modèle initial sont légèrement réajustés (fine-tuning). Cela permet d'obtenir un modèle final hautement performant et précis, tout en économisant les ressources matérielles et le temps de convergence.

L'architecture MobileNetV2 a été retenue pour sa légèreté et son efficacité. Grâce aux poids pré-entraînés sur ImageNet, elle offre une meilleure généralisation qu'un CNN classique. Pour l'adapter à notre classification binaire (Drowsy/Non-Drowsy), nous avons effectué les modifications suivantes :

- **Extraction de caractéristiques (Backbone)** : Utilisation de `include_top = False` pour supprimer la classification originale tout en conservant les couches convolutives robustes.
- **Normalisation (Rescaling)** : Ajout d'une mise à l'échelle des pixels dans l'intervalle $[-1, 1]$ pour accélérer la convergence.
- **Prévention du surapprentissage (Dropout)** : Insertion d'une couche Dropout (taux de 0,4) pour améliorer la robustesse du modèle.
- **Tête de classification (Classification Head)** : Ajout d'une couche dense avec activation Sigmoid, spécifiquement adaptée à nos deux classes.

L'implémentation de l'apprentissage par transfert s'est concrétisée en gelant les couches convolutives initiales du modèle MobileNetV2, initialement pré-entraîné sur la base de données de référence ImageNet. En conservant ces poids fixes, ces couches agissent comme un extracteur de caractéristiques hautement performant, capable de capturer les détails faciaux essentiels à la détection de la fatigue (tels que l'état des yeux et l'expression de la bouche).

Pour adapter cette architecture à notre problématique spécifique, le bloc de classification d'origine a été remplacé par de nouvelles couches fully-connected (denses). Ces couches finales ont été configurées et entraînées pour interpréter les vecteurs de caractéristiques extraits et réaliser la classification binaire terminale entre nos deux classes cibles : *Drowsy* (en état de somnolence) et *Non-Drowsy* (éveillé).

3.3.4 Entraînement et résultats

L'entraînement complet du modèle a été réalisé sur la plateforme cloud Kaggle. Les résultats expérimentaux démontrent que, malgré sa structure légère, le modèle MobileNetV2 réalise des performances exceptionnelles.

Comme le montre la Figure 3.3, les courbes d'exactitude (accuracy) progressent de manière stable pour atteindre un score final de 99,3% pour l'entraînement (Train) et dépassant 99,8% pour la validation (Val). En contrepartie, la perte (Loss) décroît continuellement pour atteindre des valeurs minimales proches de 0,027 pour le Train et 0,005 pour la Validation. Le fait que les performances de validation restent excellentes et alignées avec celles de l'entraînement écarte tout risque de surapprentissage et témoigne de la haute capacité de généralisation du modèle.



FIGURE 3.3 – Courbes d'apprentissage (Train vs Validation) du modèle

L'analyse de la matrice de confusion (Figure 3.4) montre une classification hautement performante sur la base de test, avec seulement 11 erreurs de prédiction sur un total de 6 269 images. Le modèle a correctement classifié 3 346 images comme Non-Drowsy (Vrais Positifs) et 2 912 images comme Drowsy (Vrais Négatifs). On enregistre uniquement 5 faux positifs et 6 faux négatifs. Enfin, la précision globale du modèle s'établit à 99.82%, soit un taux d'erreur inférieur à 0.2% sur l'ensemble du jeu de test, ce qui confirme la robustesse et la fiabilité du système de détection développé.

		Classe prédite	
		Drowsy	Non-Drowsy
Classe réelle	Drowsy	2912	5
	Non-Drowsy	6	3346

FIGURE 3.4 – Matrice de confusion du modèle

Le Tableau 3.2 présente le rapport de performance global du modèle MobileNetV2 sur l'ensemble du jeu de test, composé de 6269 images. Les résultats confirment l'excellente capacité de discrimination du modèle, avec une exactitude (accuracy) de 99.82%, traduisant seulement 11 erreurs de classification sur les 6269 images testées. La précision globale atteint 99.79%, le rappel 99.83% et le F1-score 99.81%, ce qui témoigne d'un équilibre quasi-parfait entre les faux positifs et les faux négatifs sur l'ensemble de la base de test.

TABLEAU 3.2 – Rapport de performance global du modèle MobileNetV2

Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Support
99.82	99.79	99.83	99.81	6269

La Figure 3.5 présente un échantillon de dix images de test, accompagnées de leur étiquette réelle (*True*) et de la prédiction fournie par le modèle MobileNetV2 (*Pred*). Sur l'ensemble de cet échantillon, comprenant cinq cas « Drowsy » et cinq cas « Non Drowsy », toutes les prédictions du modèle correspondent exactement à l'étiquette réelle, illustrant la grande capacité du modèle à distinguer correctement les deux états du conducteur sur des visages variés (différences d'angle, de luminosité et de morphologie).



FIGURE 3.5 – Échantillon de prédictions du modèle MobileNetV2 sur des images de test, avec comparaison entre l'étiquette réelle (*True*) et la prédiction du modèle (*Pred*)

3.3.5 Optimisation : Quantification TensorFlow Lite

3.3.5.1 Principe de la quantification

La quantification est une étape cruciale dans le déploiement des modèles de Deep Learning sur des plateformes matérielles limitées en ressources. Cette technique consiste à réduire la précision numérique des poids (weights) et des activations du modèle, en convertissant les valeurs initiales de la virgule flottante 32 bits (Float32) vers un format d'entiers codés sur 8 bits (INT8).

Cette transition permet non seulement une réduction drastique de l'empreinte mémoire du modèle, mais elle est également fortement recommandée pour l'optimisation des noyaux et l'accélération matérielle. En effet, la majorité des processeurs des systèmes embarqués exécutent les calculs sur des entiers (INT8) de manière beaucoup plus rapide et économe en énergie que les calculs en virgule flottante.

3.3.5.2 Modèle Float32

Par défaut, lors de la phase d'entraînement sur Kaggle, le modèle MobileNetV2 utilise une représentation standard en virgule flottante sur 32 bits (Float32). Ce format est indispensable

durant l'apprentissage car il offre la précision numérique nécessaire pour converger efficacement et ajuster les poids du réseau avec finesse.

Cependant, en phase d'inférence, ce format Float32 engendre une consommation mémoire élevée et des temps de calcul prolongés, ce qui contredit les exigences de fluidité d'une application de sécurité routière. C'est pourquoi ce modèle initial sert uniquement de base de référence pour évaluer les gains de performance après optimisation.

3.3.5.3 Quantification post-entraînement

Afin d'optimiser notre modèle MobileNetV2 pour un déploiement efficace en temps réel, nous avons exploré trois approches principales de quantification post-entraînement, résumées dans le Tableau 3.3.

1. **Quantification INT8 dynamique (Post-Training Dynamic Range)** : C'est notre approche initiale. Elle cible les poids des connexions en les compressant sur 8 bits, tandis que les activations sont quantifiées dynamiquement durant l'inférence. Elle offre un excellent compromis pour notre système : le fichier `.tflite` est réduit jusqu'à 4 fois, facilitant son intégration sur smartphones ou microcontrôleurs, tout en maintenant une précision de classification quasi identique au modèle original en Float32.
2. **Quantification Float16** : Cette méthode convertit les poids du modèle du format 32 bits vers le format 16 bits. Elle permet de réduire la taille du modèle de moitié sans nécessiter de calibration complexe. Bien que moins agressive que l'INT8, elle préserve une fidélité quasi totale à la précision du modèle original, ce qui en fait une option de secours intéressante si une perte de précision est observée lors de la quantification en entiers.
3. **Quantification « Full Integer » (entière complète)** : Il s'agit de la stratégie la plus aboutie pour l'optimisation sur le matériel embarqué. Contrairement à la version dynamique, elle quantifie à la fois les poids et les activations en 8 bits, nécessitant l'utilisation d'un jeu de données de calibration représentatif. Cette approche permet de s'affranchir totalement des calculs en virgule flottante, maximisant ainsi l'accélération matérielle sur les DSP ou les unités de traitement neuronal (NPU), et garantissant une efficacité énergétique optimale pour notre système de détection en continu.

Ces techniques permettent de réduire l'empreinte mémoire et d'accélérer l'inférence tout en préservant la précision.

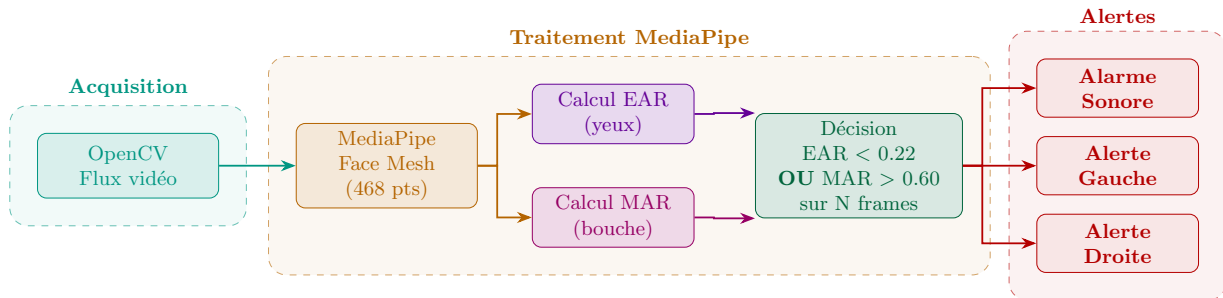
TABLEAU 3.3 – Comparaison des modèles TensorFlow Lite quantifiés.

Métrique (%)	Original (Keras)	Dynamic Range	Float16	Full Integer
Accuracy	99,82	88,43	94,00	66,00
Precision	99,79	88,40	94,70	76,70
Recall	99,83	88,10	94,00	66,00
F1-Score	99,81	88,05	94,00	64,10
Taille (KB)	8 826,88	2 445,45	4 358,71	2 643,55
Temps d'inférence (s)	4,207	0,172	0,082	0,124

3.4 Méthode 2 : approche Landmarks (MediaPipe)

3.4.1 Pipeline de détection faciale

Le pipeline de détection faciale de notre système repose sur l'architecture BlazeFace, intégrée au framework MediaPipe, choisie pour son efficacité exceptionnelle dans le traitement de flux vidéo en temps réel. Comme l'illustre la Figure 3.6, le processus s'articule autour de trois grandes phases : l'acquisition, le traitement MediaPipe et la génération des alertes.

**FIGURE 3.6** – Pipeline complet du système de détection par landmarks : acquisition, traitement MediaPipe et alertes.

- **Acquisition** : Le flux vidéo est capturé en continu via OpenCV, fournissant les images brutes au module de traitement.
- **Extraction de repères faciaux** : Le système traite les images vidéo pour extraire 468 points de repère faciaux, permettant une compréhension précise de la géométrie du visage (yeux, bouche et traits du visage), même dans des conditions d'éclairage variées.
- **Calcul parallèle des métriques EAR et MAR** : À partir des 468 points extraits, le système calcule simultanément le ratio d'ouverture des yeux (EAR, Équation 2.1) et le ratio d'ouverture de la bouche (MAR, Équation 2.2), comme détaillé précédemment.
- **Optimisation de la latence** : L'algorithme minimise la latence grâce à l'extraction directe des caractéristiques par convolution, permettant une analyse continue des trames sans retard perceptible.
- **Définition de la zone d'intérêt (ROI)** : Le système définit la zone du visage comme

une région d'intérêt, ce qui réduit le volume de données traitées et concentre les ressources informatiques uniquement sur les points critiques nécessaires à la prise de décision, tels que le suivi des yeux.

- **Flexibilité et performance** : Bien que conçue initialement pour les appareils mobiles, l'architecture démontre une excellente stabilité lorsqu'elle est utilisée avec une webcam sur des systèmes informatiques classiques, garantissant une grande précision dans le suivi des mouvements du visage pendant la conduite.

3.4.2 Logique de décision

Pour garantir la fiabilité de notre système, il est indispensable de distinguer un clignement d'œil naturel des variations accidentelles du ratio EAR, Équation 2.1). Le système ne se base pas sur l'analyse d'une seule image, mais sur une logique temporelle robuste permettant de réduire les erreurs de détection.

- **Analyse par fenêtre glissante** : Plutôt que de surveiller l'ouverture de l'œil instantanément, nous utilisons une fenêtre glissante regroupant les valeurs EAR sur une série de N images consécutives. Cette méthode permet de reconnaître la dynamique réelle du clignement et d'éliminer les erreurs causées par les mouvements soudains ou les variations de luminosité.
- **Contraintes temporelles de validation** : Pour valider la fermeture de l'œil, le système impose des contraintes de durée. Un clignement est considéré comme valide uniquement si la fermeture dure entre 60 ms et 700 ms. Toute variation en dehors de cet intervalle est ignorée, ce qui permet de filtrer les bruits et les événements non physiologiques.
- **Critère de décision combiné** : Comme illustré dans la Figure 3.6, l'état de somnolence est déclenché dès que l'une des deux conditions suivantes est vérifiée sur la fenêtre de N images : $EAR < 0,22$ (Équation 2.1) **ou** $MAR > 0,60$ (Équation 2.2). Cette logique disjonctive permet de capturer aussi bien les signes de fermeture prolongée des yeux que les épisodes de bâillement.
- **Détection de l'orientation de la tête (gauche / droite)** : En complément des métriques EAR et MAR, le système exploite la position relative des landmarks du nez et des yeux pour estimer l'orientation horizontale de la tête. Un déplacement significatif du point central du nez par rapport à l'axe médian du visage permet de détecter une rotation vers la gauche ou vers la droite, révélatrice d'une perte d'attention ou d'un assoupissement latéral.

3.4.3 Génération des alertes

Une fois le critère de décision validé, le système déclenche une réponse adaptée à la situation détectée, comme le montre la dernière étape de la Figure 3.6 :

- **Alarme sonore** : Signal principal déclenché systématiquement en cas de détection de somnolence (EAR ou MAR hors seuil), destiné à réveiller immédiatement l'attention du conducteur.
- **Alerte tête tournée à gauche** : Déclenchée lorsque l'orientation horizontale de la tête

dépasse le seuil de rotation vers la gauche, signalant une perte de concentration sur la route.

- **Alerte tête tournée à droite** : Déclenchée de manière symétrique lorsque la rotation de la tête dépasse le seuil vers la droite, indiquant une distraction ou un relâchement de l'attention.

3.4.4 Tests en conditions réelles (*In the Wild*)

Afin de garantir l'efficacité du système dans un environnement de conduite réel, nous avons évalué l'algorithme dans des conditions dites *in the wild*. Cet environnement présente des défis techniques complexes, différents de ceux rencontrés dans un cadre expérimental contrôlé (voir le Figure 3.7).

- **Gestion des variations de luminosité** : Grâce aux techniques de normalisation et à l'architecture flexible de MediaPipe, le système de mesure du ratio d'ouverture des yeux (EAR) demeure stable même face aux variations d'intensité lumineuse à l'intérieur de l'habitacle.
- **Adaptation aux mouvements de la tête** : L'utilisation de la technologie Face Mesh, basée sur le suivi de 468 points faciaux, permet de maintenir une zone d'intérêt précise autour des yeux malgré les changements d'orientation de la tête pendant la conduite.
- **Efficacité des performances et des ressources** : Le framework MediaPipe permet d'atteindre un équilibre optimal entre la précision du traitement et la consommation des ressources grâce à un traitement efficace des trames vidéo, assurant ainsi la fluidité du système sur des plateformes à ressources limitées.
- **Réduction des fausses alertes** : La combinaison du filtrage temporel et des mécanismes de synchronisation permet de réduire significativement les fausses alertes (*False Positives*) provoquées par les mouvements aléatoires de la tête ou les incohérences temporaires des données.



FIGURE 3.7 – Échantillons testés en temps réel.

3.5 Comparaison des deux méthodes

Le Tableau 3.4 synthétise les performances comparées des deux approches développées, qui présentent des avantages complémentaires.

TABLEAU 3.4 – Comparaison détaillée des deux approches de détection de somnolence.

Critère	MobileNetV2	MediaPipe
Type d'approche	Deep Learning	Règles géométriques
Entraînement	Oui (41 790 images)	Non
Précision (Accuracy)	99,8%	Dépend du seuil EAR
FPS	15-18 (Float16)	Temps réel 21-28
Taille du modèle	4 358 KB (Float16)	Bibliothèque légère
Robustesse luminosité	Moyenne	Bonne
Robustesse pose	Bonne	Très bonne
Déploiement embarqué	TFLite / Raspberry Pi	Raspberry Pi / PC
Indicateurs	Drowsy / Non-Drowsy	EAR, MAR

L'approche MobileNetV2 offre une précision remarquable grâce à l'apprentissage sur un large jeu de données, atteignant une exactitude de 99,8% sur le jeu de test. Elle est particulièrement adaptée aux scénarios où la qualité d'image est maîtrisée et où une classification binaire robuste est requise. Toutefois, comme le montre le Tableau 3.4, sa robustesse face aux varia-

tions de luminosité reste modérée, ce qui peut limiter ses performances en conditions réelles non contrôlées (in the wild).

L'approche MediaPipe, quant à elle, se distingue par sa légèreté et son absence de phase d'entraînement. Elle repose sur des métriques géométriques interprétables (EAR, MAR) et offre une grande adaptabilité aux conditions réelles de conduite grâce au filtrage temporel, avec une robustesse supérieure à la fois à la luminosité et aux variations de pose. Cette stabilité en environnement non contrôlé en fait une solution particulièrement pertinente pour les systèmes embarqués à ressources très limitées.

En pratique, une architecture hybride combinant les deux approches permettrait de bénéficier simultanément de la haute précision du modèle Deep Learning et de la réactivité temps réel de l'approche géométrique, constituant ainsi une solution optimale pour la sécurité routière.

3.6 Conclusion

Ce chapitre a été consacré à la conception technique et à la réalisation concrète de notre système de détection de la somnolence selon deux approches complémentaires. La première, basée sur MobileNetV2 et la quantification TensorFlow Lite, offre une précision quasi-parfaite de 99,8% tout en restant optimisée pour les systèmes embarqués. La seconde, basée sur MediaPipe et l'analyse géométrique EAR/MAR, garantit une détection légère et réactive en temps réel sans nécessiter de phase d'entraînement. Les tests effectués démontrent l'efficacité des deux approches, confirmant leur complémentarité pour une intégration robuste dans un système de sécurité routière embarqué.

Deuxième partie

Partie Pratique

Chapitre 4

Environnement de Développement

4.1	Introduction	53
4.2	Environnement matériel	53
4.2.1	Présentation du Raspberry Pi	53
4.2.2	Écran TFT LCD 7 pouces	56
4.2.3	La caméra Pi	56
4.2.4	Carte Micro SD	57
4.3	Configuration matérielle	58
4.3.1	Téléchargement de l'image de distribution	58
4.3.2	Enregistrement du contenu de l'image sur la carte SD	58
4.3.3	Configuration initiale	58
4.4	Environnement de travail	60
4.4.1	Langage Python	61
4.4.2	L'IDE Thonny	61
4.4.3	Bibliothèques utilisées	62
4.4.4	Environnement de développement	63
4.4.5	Environnement virtuel : drowsy	64
4.5	Conclusion	64

4.1 Introduction

La mise en place d'un système de détection de somnolence au volant repose non seulement sur des algorithmes performants, mais également sur un environnement matériel et logiciel bien adapté aux contraintes du temps réel et du déploiement embarqué. Après avoir présenté les fondements théoriques et les approches de détection dans les chapitres précédents, ce chapitre est consacré à la description détaillée de l'environnement de développement utilisé pour la réalisation concrète du système.

Dans un premier temps, nous présentons les composants matériels retenus, à savoir le Raspberry Pi 4 comme plateforme embarquée principale, la caméra Pi pour l'acquisition vidéo, l'écran TFT LCD pour l'affichage, ainsi que la carte Micro SD pour le stockage. Dans un second temps, nous décrivons les étapes de configuration du système d'exploitation et des interfaces de communication. Enfin, nous présentons l'environnement logiciel, notamment le langage Python, l'IDE Thonny et les différentes bibliothèques utilisées pour le traitement d'images, l'apprentissage profond et la détection en temps réel.

4.2 Environnement matériel

4.2.1 Présentation du Raspberry Pi

Le Raspberry Pi est un nano-ordinateur offrant une solution économique et performante pour de nombreuses applications informatiques. Grâce à son architecture compacte et à sa faible consommation d'énergie, il est largement utilisé dans les systèmes embarqués, l'automatisation et la vision par ordinateur. Il permet également d'effectuer différentes tâches telles que le traitement des données, la surveillance et le contrôle de dispositifs externes.

4.2.1.1 Historique

Le Raspberry Pi est un ordinateur monocarte (Single Board Computer) développé par la fondation Raspberry Pi. Depuis la commercialisation du premier modèle en 2012, plusieurs versions ont été proposées, notamment les modèles A, B, Zero ainsi que le Compute Module destiné aux applications industrielles.

Ces cartes embarquent un processeur (CPU), un processeur graphique (GPU) et une mémoire, tout en fonctionnant avec une alimentation de 5 V. Elles disposent également d'un connecteur dédié à la caméra et de broches GPIO permettant l'interfaçage avec différents capteurs et composants électroniques.

Afin d'étendre les possibilités de la carte, des modules complémentaires appelés HAT (Hardware Attached on Top) peuvent être ajoutés. Le Raspberry Pi intègre également différentes interfaces de communication, notamment Ethernet et les connexions sans fil, ce qui lui permet de s'adapter à de nombreux types de projets.

Il peut être utilisé comme un ordinateur classique sous Linux ou fonctionner en mode « headless », sans écran ni clavier, pour exécuter des applications de manière autonome. Contrairement aux microcontrôleurs tels que l'Arduino ou le Raspberry Pi Pico, il est capable de faire fonctionner un système d'exploitation complet et de gérer plusieurs processus simultanément.



FIGURE 4.1 – Évolution du Raspberry Pi.

Après cet aperçu historique de l'évolution du Raspberry Pi, nous présentons dans ce qui suit les composants matériels du modèle retenu pour notre projet, à savoir le Raspberry Pi 4 Model B.

4.2.1.2 Composants de base

Le Raspberry Pi 4 Model B est le plus récent ajout à la gamme populaire d'ordinateurs Raspberry Pi. Il représente une avancée significative par rapport à la génération précédente grâce à des améliorations majeures dans la vitesse du processeur, les performances multimédia, la mémoire et la connectivité. De plus, il conserve une compatibilité ascendante avec les modèles précédents et une consommation d'énergie similaire. En raison de ces améliorations, le Raspberry Pi 4 Model B offre des performances de bureau comparables à celles des PC x86 d'entrée de gamme.

Afin de mieux cerner les capacités techniques de cette carte, nous détaillons ci-après ses principales spécifications.

4.2.1.3 Spécifications

- **Processeur** : Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) SoC 64 bits @ 1,5 GHz.
- **Mémoire** : 8 Go de mémoire SDRAM LPDDR4-3200.
- **Connectivité** :
 1. Wi-Fi : IEEE 802.11ac 2,4 GHz et 5,0 GHz sans fil.
 2. Bluetooth : Bluetooth 5.0, BLE.
 3. Ethernet : Gigabit Ethernet.
 4. USB : 2 ports USB 3.0 et 2 ports USB 2.0.
 5. GPIO : connecteur GPIO standard à 40 broches, entièrement rétrocompatible avec les cartes précédentes.
- **Vidéo et son** :
 1. HDMI : 2 ports micro-HDMI (jusqu'à 4Kp60 pris en charge).

4.2.2 Écran TFT LCD 7 pouces

L'écran TFT LCD (Thin Film Transistor Liquid Crystal Display) est une technologie d'affichage basée sur les cristaux liquides et les transistors à couche mince. Cette technologie permet d'améliorer la qualité de l'image, notamment en termes de contraste et de précision d'affichage. Contrairement aux écrans LCD à matrice passive ou à commande directe, l'écran TFT LCD fonctionne selon le principe de la matrice active, ce qui garantit de meilleures performances visuelles.

Les écrans TFT LCD sont largement utilisés dans de nombreux équipements électroniques tels que les téléviseurs, les moniteurs d'ordinateur, les téléphones mobiles, les appareils portables, les consoles de jeux vidéo, les assistants numériques personnels, les systèmes de navigation, les projecteurs ainsi que les tableaux de bord automobiles (voir Figure 4.3).



FIGURE 4.3 – Écran TFT LCD 7 pouces pour Raspberry Pi.

L'écran étant défini, nous présentons à présent le module caméra, composant essentiel de notre système puisqu'il assure l'acquisition du flux vidéo du conducteur en temps réel.

4.2.3 La caméra Pi

Le module caméra haute résolution, compatible avec l'ensemble des modèles Raspberry Pi, permet la capture d'images avec une grande sensibilité et un faible niveau de bruit tout en conservant un format compact et léger. Il se connecte à la carte Raspberry Pi par l'intermédiaire d'une interface série pour caméra (CSI), spécialement conçue pour prendre en charge des débits de données élevés et transmettre directement les données des pixels au processeur (voir Figure 4.4).

4.2.3.1 Spécifications

- Résolution : 8 mégapixels.
- Résolution des images fixes : 3280×2464 pixels.
- Taux maximal de transfert d'images :
 - 1080p : 30 images par seconde (encodage et décodage).
 - 720p : 60 images par seconde.
- Connexion au Raspberry Pi : câble plat 15 broches relié à l'interface série dédiée MIPI à 15 broches pour caméra.

- Balance des blancs automatique.
- Filtre de bande automatique.
- Détection automatique de la luminance 50/60 Hz.
- Calibrage automatique du niveau de noir.
- Température de fonctionnement : de $-20\text{ }^{\circ}\text{C}$ à $60\text{ }^{\circ}\text{C}$.
- Stabilité de l'image : de $-20\text{ }^{\circ}\text{C}$ à $60\text{ }^{\circ}\text{C}$.
- Taille de l'objectif : 1/4 pouce.
- Dimensions : $23,86 \times 25 \times 9\text{ mm}$.
- Poids : 3 g.



FIGURE 4.4 – Raspberry Pi Camera Module 2.

Une fois la caméra présentée, nous décrivons le dernier composant matériel de notre système, à savoir la carte Micro SD qui assure le stockage du système d'exploitation et des données du projet.

4.2.4 Carte Micro SD

La carte SD (Secure Digital) est un support de stockage amovible de petite taille destiné à l'enregistrement des données numériques. Elle est largement utilisée dans différents équipements électroniques tels que les appareils photo numériques, les smartphones, les tablettes et les ordinateurs portables.

Dans le domaine des systèmes embarqués, les cartes SD servent généralement au stockage du système d'exploitation ainsi que des programmes exécutés sur des plateformes telles que le Raspberry Pi et l'Arduino. Leur utilisation est particulièrement appréciée grâce à leur format compact, leur caractère amovible et la facilité avec laquelle elles peuvent être transférées d'un appareil à un autre (voir Figure 4.5).

Il est essentiel d'utiliser une carte SD de bonne qualité afin d'assurer des performances satisfaisantes et une utilisation fiable sur le long terme. Les cartes SD sont disponibles avec différentes capacités de stockage, allant de quelques gigaoctets à plusieurs téraoctets. Le choix de la capacité dépend des besoins et des exigences du projet.



FIGURE 4.5 – Carte Micro SD.

4.3 Configuration matérielle

Avant de démarrer le développement, il est indispensable de configurer correctement le Raspberry Pi. Cette section décrit les étapes successives permettant d'installer le système d'exploitation, de configurer les interfaces de communication et d'activer les périphériques nécessaires au bon fonctionnement du système.

4.3.1 Téléchargement de l'image de distribution

La première étape consiste à télécharger gratuitement l'image du système d'exploitation Raspberry Pi OS depuis le site officiel de Raspberry Pi vers notre ordinateur. Pour notre projet, nous avons choisi la version Raspberry Pi OS (64 bits) avec bureau.

Une fois l'image téléchargée, l'étape suivante consiste à la transférer sur la carte Micro SD afin de préparer le support de démarrage du Raspberry Pi.

4.3.2 Enregistrement du contenu de l'image sur la carte SD

Pour transférer l'image du système d'exploitation sur la carte SD, nous avons utilisé un logiciel dédié à cette opération. L'application Raspberry Pi Imager a été retenue afin de garantir une compatibilité optimale.

Après avoir lancé le logiciel, nous sélectionnons l'image du système d'exploitation à l'aide du bouton « CHOOSE OS » en indiquant le fichier correspondant. Ensuite, le bouton « CHOOSE STORAGE » permet de choisir la carte SD sur laquelle l'image sera enregistrée. Une fois ces paramètres définis, nous cliquons sur « WRITE » pour démarrer le processus d'écriture.

À la fin de l'opération, un message de confirmation apparaît pour signaler la réussite de l'enregistrement. La carte SD peut alors être retirée de l'ordinateur puis insérée dans l'emplacement prévu sur le Raspberry Pi.

La carte SD étant prête, nous procédons à la configuration initiale du système, notamment la modification du nom d'hôte et l'activation des interfaces SSH, VNC et caméra.

4.3.3 Configuration initiale

Par défaut, le Raspberry Pi possède le nom d'hôte « raspberrypi », visible sur le réseau. La première étape de configuration consiste à modifier ce nom selon la procédure suivante :

1. Se connecter au terminal.

2. Exécuter la commande `sudo raspi-config`.
3. Sélectionner l'option « System Options ».
4. Choisir « Hostname » puis appuyer sur Entrée. Si un message d'avertissement concernant l'utilisation de caractères spéciaux apparaît, cliquer simplement sur « Ok ».
5. Saisir le nouveau nom d'hôte puis valider avec « Ok ».

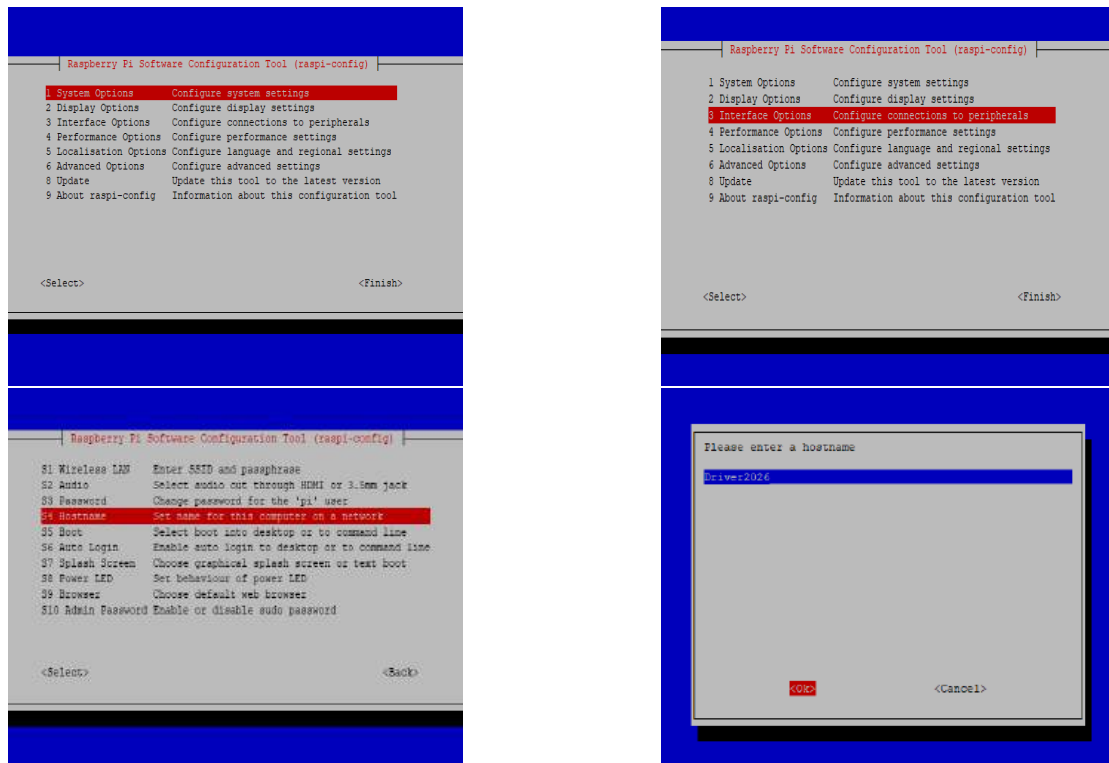


FIGURE 4.6 – Modification du nom d'hôte.

Pour des raisons de sécurité, le service SSH est désactivé par défaut sur le Raspberry Pi, même s'il est déjà installé. La deuxième étape de configuration consiste donc à l'activer selon les étapes suivantes :

1. Se connecter au terminal.
2. Exécuter la commande `sudo raspi-config`.
3. Sélectionner l'option « Interface Options ».
4. Choisir « SSH » dans le menu.
5. Sélectionner « Enable » pour activer le service.
6. Confirmer le message à l'aide du bouton « Ok ».
7. Cliquer sur « Finish » pour terminer la configuration.



FIGURE 4.7 – Activation du SSH.

Après l'activation de SSH sur le Raspberry Pi (voir Figure 4.7), il est nécessaire d'installer un client SSH sur l'ordinateur. Dans notre cas, l'ordinateur fonctionne sous Windows ; nous avons donc utilisé le logiciel PuTTY, un programme open source développé et maintenu par une communauté d'utilisateurs.

Le protocole VNC permet de contrôler un ordinateur à distance tout en affichant son bureau. Cette fonctionnalité nous permet d'accéder au Raspberry Pi et d'observer son interface graphique sans avoir besoin d'un écran connecté directement à la carte. Comme le montre la Figure 4.8, l'activation de VNC s'effectue de la même manière que celle de SSH, en sélectionnant simplement l'option « VNC » dans le menu « Interface Options ».

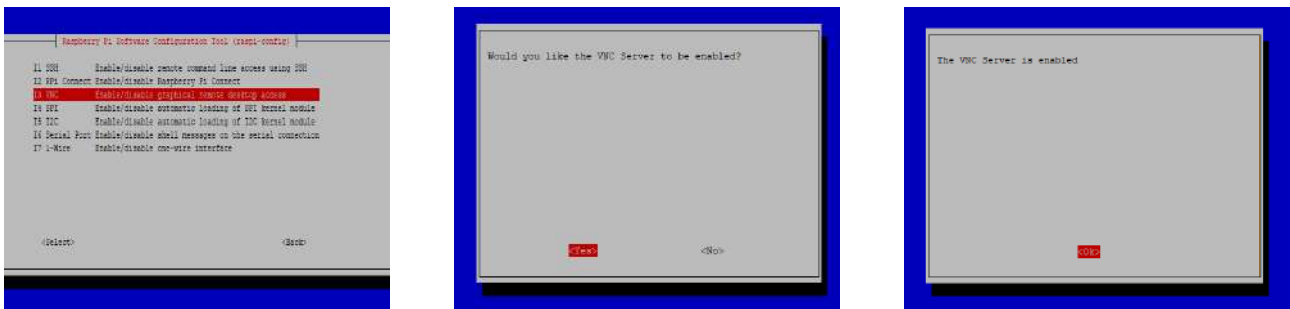


FIGURE 4.8 – Activation du VNC.

Il est à noter que sur les versions récentes de Raspberry Pi OS, la caméra est activée par défaut sans nécessiter l'exécution de la commande `sudo raspi-config`. Il suffit de connecter le module caméra au port CSI de la carte et de redémarrer le système pour qu'elle soit immédiatement reconnue et opérationnelle.

Il est important de maintenir le système à jour afin d'assurer le bon fonctionnement des pilotes et des différents composants. Pour cela, il suffit d'ouvrir le terminal et d'exécuter la commande `sudo apt-get update`. Le Raspberry Pi se connecte alors au dépôt de paquets de la distribution Raspbian afin de récupérer les versions les plus récentes des logiciels installés.

4.4 Environnement de travail

Le système de détection a été entièrement développé en Python, en s'appuyant sur un ensemble de bibliothèques spécialisées. Cette section présente le langage de programmation retenu, l'environnement de développement utilisé ainsi que les principales bibliothèques intégrées dans notre projet.

4.4.1 Langage Python

Python est un langage de programmation particulièrement adapté aux débutants grâce à sa syntaxe simple et claire. Il est facile à apprendre, à lire et à maintenir. Il dispose également d'une bibliothèque standard riche et compatible avec plusieurs systèmes d'exploitation tels que Windows, UNIX et Mac. Python permet de développer de nombreuses applications, allant du traitement de texte à la programmation de jeux.



FIGURE 4.9 – Illustration du bureau du Raspberry Pi.

Pour faciliter l'écriture et le débogage des scripts Python sur le Raspberry Pi, nous avons utilisé un environnement de développement intégré adapté à cette plateforme.

4.4.2 L'IDE Thonny

Thonny est un environnement de développement intégré (IDE) conçu pour l'apprentissage et l'enseignement du langage Python. Il facilite la compréhension du fonctionnement des programmes grâce à plusieurs outils de visualisation, tels que l'exécution pas à pas, l'évaluation des expressions et la représentation de la pile d'appels.

Cet IDE permet d'écrire, d'exécuter et d'enregistrer des scripts Python. Il offre également une fenêtre d'exécution permettant d'afficher directement les résultats du programme. Thonny est installé par défaut sur le système Raspbian et peut être lancé à partir du menu « Programmation » (voir Figure 4.10).



FIGURE 4.10 – L'IDE Thonny.

En complément de l'IDE, notre projet repose sur plusieurs bibliothèques Python couvrant des domaines variés tels que le traitement d'images, l'apprentissage profond, la détection faciale et la gestion des alertes sonores. Ces bibliothèques sont présentées ci-après.

4.4.3 Bibliothèques utilisées

Pandas est une bibliothèque Python utilisée pour la manipulation et l'analyse des données structurées. Elle permet de traiter facilement des données organisées sous forme de tableaux grâce à une structure appelée `DataFrame`, composée de lignes et de colonnes. Cette bibliothèque est souvent utilisée pour le filtrage, le tri et l'organisation des données. Dans ce projet, Pandas a été utilisée pour la gestion et le traitement des données issues des expérimentations.

NumPy (Numerical Python) est une bibliothèque open source largement utilisée pour le calcul scientifique en Python. Elle fournit des structures de données performantes, notamment les tableaux multidimensionnels et les matrices. De nombreuses bibliothèques Python s'appuient sur NumPy pour manipuler les données et effectuer des opérations mathématiques et statistiques de manière efficace avec un code simplifié.

Matplotlib est une bibliothèque Python dédiée à la création de graphiques et de représentations visuelles en deux dimensions. Elle permet de générer différents types de figures, tels que les courbes, les histogrammes et les diagrammes, facilitant ainsi l'analyse et l'interprétation des résultats obtenus.

OpenCV (Open Source Computer Vision Library) est une bibliothèque open source spécialisée dans la vision par ordinateur et le traitement d'images. Développée en C et C++, elle est compatible avec plusieurs systèmes d'exploitation tels que Linux, Windows et Mac OS X, tout en proposant des interfaces pour différents langages de programmation, notamment Python. Conçue pour les applications en temps réel, OpenCV met à disposition un large ensemble de fonctions permettant de développer rapidement des applications de vision artificielle dans divers domaines tels que la sécurité, l'imagerie médicale, la robotique et l'étalonnage des caméras.

TensorFlow est une bibliothèque open source dédiée au calcul numérique et à l'apprentissage automatique. Elle offre un environnement flexible permettant de concevoir, d'entraîner et de déployer efficacement des modèles d'intelligence artificielle, notamment les réseaux de neurones profonds. Grâce à ses nombreux outils, TensorFlow facilite le développement d'applications basées sur l'apprentissage profond.

SciPy est une bibliothèque Python destinée au calcul scientifique et à la résolution de problèmes numériques. Elle regroupe plusieurs modules spécialisés permettant de réaliser différentes opérations, notamment l'optimisation, l'algèbre linéaire, le traitement du signal et les statistiques. SciPy est souvent utilisée en complément de NumPy afin de fournir des outils plus avancés pour l'analyse et le calcul scientifique. Dans ce projet, elle a été utilisée pour effectuer certains calculs liés au traitement des données et des distances.

Parmi les bibliothèques dédiées à la vision par ordinateur et à la détection faciale, MediaPipe occupe une place centrale dans notre système.

MediaPipe est une bibliothèque open source développée par Google, dédiée au traitement multimédia en temps réel et à la vision par ordinateur. Elle propose des solutions prêtes à l'emploi pour la détection et le suivi d'objets (visage, mains, posture), ce qui facilite l'intégration de modèles d'intelligence artificielle dans des applications interactives ou des systèmes embarqués.

Enfin, afin de compléter le système par un mécanisme d'alerte sonore, la bibliothèque Pygame a été intégrée.

Pygame est une bibliothèque open source pour Python basée sur la bibliothèque SDL (Simple DirectMedia Layer). Elle fournit des modules permettant de gérer les éléments multimédias, notamment le son et les images. Dans ce projet, elle a été utilisée pour la lecture des alertes sonores.

L'ensemble de ces bibliothèques ayant été présenté, nous décrivons à présent l'environnement de développement global mis en place pour garantir la reproductibilité et la stabilité du projet sur les différentes plateformes.

4.4.4 Environnement de développement

Le système proposé a été développé à l'aide du langage de programmation Python, version 3.11.9, en raison de sa compatibilité avec les principales bibliothèques d'intelligence artificielle et de vision par ordinateur, notamment MediaPipe et TensorFlow.

Dans un premier temps, un environnement de développement a été mis en place sur un ordinateur afin de garantir la stabilité des dépendances et de faciliter l'entraînement et le test du modèle. Par la suite, afin de permettre son déploiement sur une plateforme embarquée, une tentative de mise à niveau et d'adaptation du projet vers Python 3.13 a été effectuée pour qu'il fonctionne sur un Raspberry Pi, à l'aide du fichier `requirements.txt` (voir Listing 4.1).

```
1 numpy==1.26.4
2 opencv-python==4.9.0.80
3 mediapipe==0.10.14
4 scipy==1.11.4
5 pygame==2.5.2
```

Listing 4.1 – Contenu du fichier `requirements.txt`.

Le fichier `requirements.txt` est un fichier texte utilisé dans les projets Python pour répertorier toutes les bibliothèques et dépendances nécessaires au bon fonctionnement de l'application. Ce fichier permet l'installation automatique des paquets requis, avec leurs versions appropriées, à l'aide d'une seule commande.

L'importance de ce fichier réside dans sa capacité à garantir la reproductibilité du projet dans différents environnements. Il garantit que tous les appareils sur lesquels le programme est exécuté utilisent les mêmes versions des bibliothèques, ce qui évite les problèmes de compatibilité et facilite le déploiement du système, en particulier sur des plateformes embarquées telles que le Raspberry Pi.

4.4.5 Environnement virtuel : drowsy

Afin de garantir la stabilité et l'isolation des dépendances du projet, nous avons créé un environnement virtuel Python dédié, nommé **drowsy**. Un environnement virtuel permet d'installer les bibliothèques nécessaires au projet de manière indépendante du système, évitant ainsi tout conflit de versions avec d'autres projets Python installés sur la même machine.

La création et l'activation de cet environnement s'effectuent à l'aide des commandes suivantes :

```
1 # Création de l'environnement virtuel
2 python -m venv drowsy
3
4 # Activation sous Linux / Raspberry Pi OS
5 source drowsy/bin/activate
6
7 # Exécution du programme principal
8 python3 main.py
```

Listing 4.2 – Création et activation de l'environnement virtuel **drowsy**.

Une fois l'environnement activé, toutes les bibliothèques listées dans le fichier **requirements.txt** sont installées à l'intérieur de cet environnement grâce à la commande suivante :

```
1 pip install -r requirements.txt
```

Listing 4.3 – Installation des dépendances dans l'environnement virtuel.

Cette approche garantit que le système de détection de somnolence fonctionne de manière identique sur l'ordinateur de développement et sur le Raspberry Pi, en s'assurant que les mêmes versions des bibliothèques sont utilisées sur les deux plateformes.

4.5 Conclusion

Dans ce chapitre, nous avons présenté l'environnement matériel et logiciel mis en place pour le développement et le déploiement du système de détection de somnolence. Sur le plan matériel, le Raspberry Pi 4 constitue la plateforme embarquée centrale, associé à la caméra Pi pour l'acquisition vidéo, à l'écran TFT LCD pour l'affichage et à la carte Micro SD pour le stockage. Sur le plan logiciel, le système repose sur le langage Python, l'IDE Thonny et un ensemble de bibliothèques spécialisées, installées dans un environnement virtuel dédié nommé **drowsy** afin de garantir la reproductibilité et la stabilité du projet. L'ensemble de cet environnement constitue une base solide pour le développement et le déploiement efficace du système de détection de somnolence présenté dans le chapitre suivant.

Chapitre 5

Évaluation Comparative et Discussio

5.1	Introduction	66
5.2	Protocole expérimental	66
5.3	Résultats	68
5.3.1	Performances sur Raspberry Pi	73
5.3.2	Journalisation des données (log CSV)	75
5.4	Discussion	75
5.5	Conclusion	76

5.1 Introduction

Ce chapitre est consacré à l'évaluation comparative rigoureuse des deux approches développées dans le cadre de ce travail : le modèle MobileNetV2 (optimisé par quantification via TensorFlow Lite) et le système basé sur MediaPipe. L'objectif est d'analyser le comportement de chaque architecture face aux contraintes du temps réel. À travers une série de tests effectués sur le matériel cible, nous déterminerons le système qui allie précision et rapidité d'exécution, afin de garantir l'efficacité du système embarqué dans le véhicule pour la sécurité du conducteur.

5.2 Protocole expérimental

Le système de détection de la somnolence repose sur une série d'étapes successives d'analyse et de classification des images, comme l'illustre la figure 5.1, qui a été développée à partir d'une carte Raspberry Pi 4. Ces étapes sont les suivantes :

1. **Acquisition des images** Dans un premier temps, la caméra 8 mégapixels du Raspberry Pi capture en continu des images du conducteur.
2. **Prétraitement (Prétraitement)** Les images capturées sont transmises à la bibliothèque OpenCV, qui se charge de lire et de traiter les flux vidéo.
3. **Détection du visage (Détection du visage)** Ensuite, MediaPipe Face Mesh extrait 468 points caractéristiques du visage, ce qui permet de localiser avec précision la position des yeux et de la bouche.
4. **Calcul des indicateurs** À partir de ces points, le système calcule l'indicateur EAR pour mesurer le degré d'ouverture des yeux et l'indicateur MAR pour analyser le degré d'ouverture de la bouche.
5. **Prise de décision** La logique de prise de décision est appliquée : si la valeur de l'EAR devient inférieure au seuil défini ou si la valeur du MAR dépasse le seuil autorisé, le système considère qu'un état de somnolence a été détecté.
6. **Génération de l'alerte** Dans ce cas, une alerte sonore est déclenchée par un buzzer afin d'attirer l'attention du conducteur.
7. **Exécution embarquée**

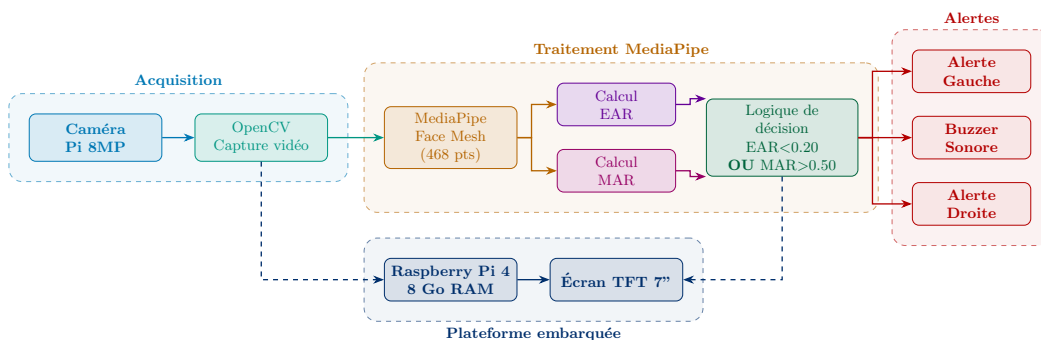


FIGURE 5.1 – Architecture générale du système embarqué basé sur Raspberry Pi.

Cette architecture permet une détection fiable de la somnolence en temps réel, tout en respectant les exigences des systèmes embarqués destinés à la sécurité routière. L'ensemble de la chaîne de traitement est exécuté sur une carte Raspberry Pi 4 équipée d'un écran TFT de 7 pouces, offrant ainsi un système autonome capable d'assurer une surveillance continue et de fonctionner en temps réel au sein d'un environnement embarqué dédié aux applications de sécurité routière.



FIGURE 5.2 – Dispositif matériel du système embarqué : Raspberry Pi 4, écran TFT 7 pouces et module caméra monté sur support, exécutant l'interface de détection en temps réel.

La figure 5.2 présente le montage matériel complet utilisé pour les tests expérimentaux, conforme à l'architecture décrite en figure 5.1. Le système repose sur une carte Raspberry Pi 4, connectée à un écran TFT tactile de 7 pouces affichant en temps réel l'interface de détection ainsi que les indicateurs EAR, MAR et le statut de vigilance du conducteur. Le module caméra, fixé sur un support orientable, est positionné face au sujet afin d'assurer une acquisition continue du flux vidéo. L'ensemble du traitement — détection du visage, calcul des indicateurs et prise de décision — est exécuté localement sur la carte, sans recours à une connexion externe.

L'évaluation des deux modèles (MobileNetV2 et MediaPipe) a été menée selon deux environnements de test — une station de travail (PC) et la carte Raspberry Pi 4 — et repose sur deux critères communs : le calcul du nombre d'images traitées par seconde (FPS) et une évaluation visuelle qualitative des résultats de détection.

Évaluation sur PC : Sur la station de travail, chaque modèle a été testé sur un ensemble d'images représentant différents états du conducteur (normal, somnolence, bâillement). Le FPS a été mesuré afin d'estimer la vitesse de traitement de chaque modèle dans un environnement disposant de ressources de calcul importantes. Une évaluation visuelle a ensuite été réalisée en comparant, pour chaque image, l'état détecté par le modèle (ainsi que les indicateurs EAR et MAR dans le cas de MediaPipe) à l'état réel observé, afin de vérifier la cohérence des résultats.

Évaluation sur Raspberry Pi 4 : Le même protocole a été appliqué sur la carte Raspberry Pi 4, dans des conditions matérielles plus contraintes et représentatives du système embarqué final. Le FPS a de nouveau été mesuré pour chaque modèle, permettant de comparer leur capacité à fonctionner en temps réel sur ce matériel limité. L'évaluation visuelle a également été reconduite afin de vérifier que la qualité de détection restait cohérente malgré la réduction des ressources disponibles par rapport au PC.

Cette double évaluation, croisant les deux modèles et les deux environnements de test, permet de comparer non seulement la rapidité de traitement (FPS) de MobileNetV2 et de MediaPipe, mais aussi la stabilité de leurs résultats lors du passage d'une station de travail à un système embarqué aux ressources limitées.

5.3 Résultats

L'interface du modèle MobileNetV2 a été conçue pour offrir un affichage simple et rapide des résultats de classification. Comme le montre la figure 5.3, le système affiche l'état du conducteur en temps réel sous la forme de deux catégories : « Somnolent » (Drowsy) ou « Normal » (Non Drowsy). Le niveau de confiance est également affiché afin d'évaluer la fiabilité de la prédiction fournie par le modèle. Lorsque le conducteur présente des signes de somnolence, la couleur de l'interface vire au rouge et un message d'alerte s'affiche. Le système intègre également un compteur de détection qui comptabilise le nombre d'images consécutives classées comme « somnolence » avant le déclenchement de l'alerte. Cette interface permet une prise de décision rapide tout en restant facile à comprendre pour l'utilisateur.

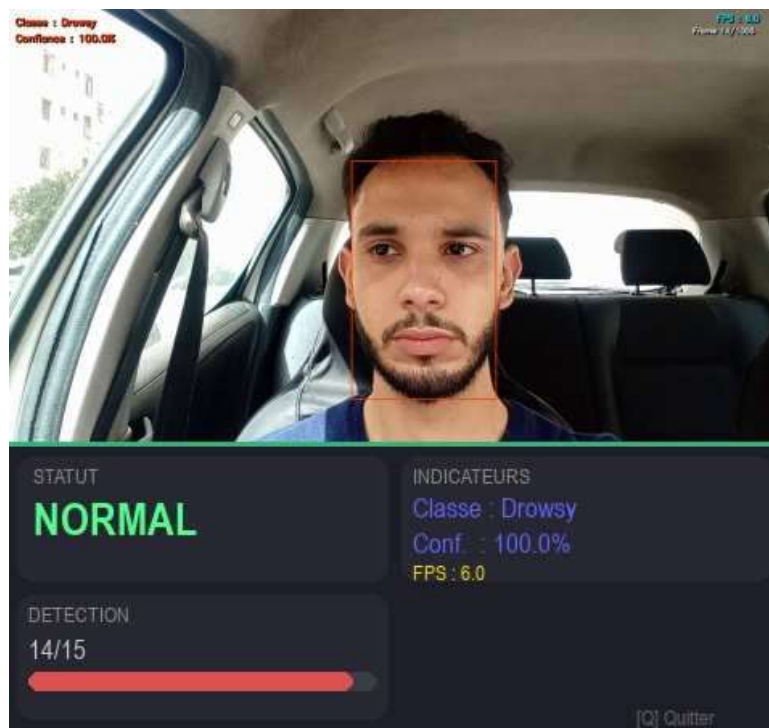


FIGURE 5.3 – Interface de model MobileNetV2

MobileNetV2 a été évalué sur une station de travail. Les résultats démontrent une grande précision de classification grâce à son architecture profonde, permettant de distinguer efficacement les états « *Drowsy* » et « *Non Drowsy* » en se basant sur le contenu global de l'image.

Comme le montrent ces tests de la figure 5.4 dans certains cas, il s'est avéré que le modèle MobileNetV2 a émis une fausse alerte et a classé un conducteur et, dans certains cas, il n'a pas réussi à détecter correctement un état de somnolence. Et bien que MobileNetV2 ait obtenu de bons résultats en matière de classification, il a commis certaines erreurs dans des conditions réelles. C'est pourquoi on a eu recours au modèle basé sur MediaPipe.

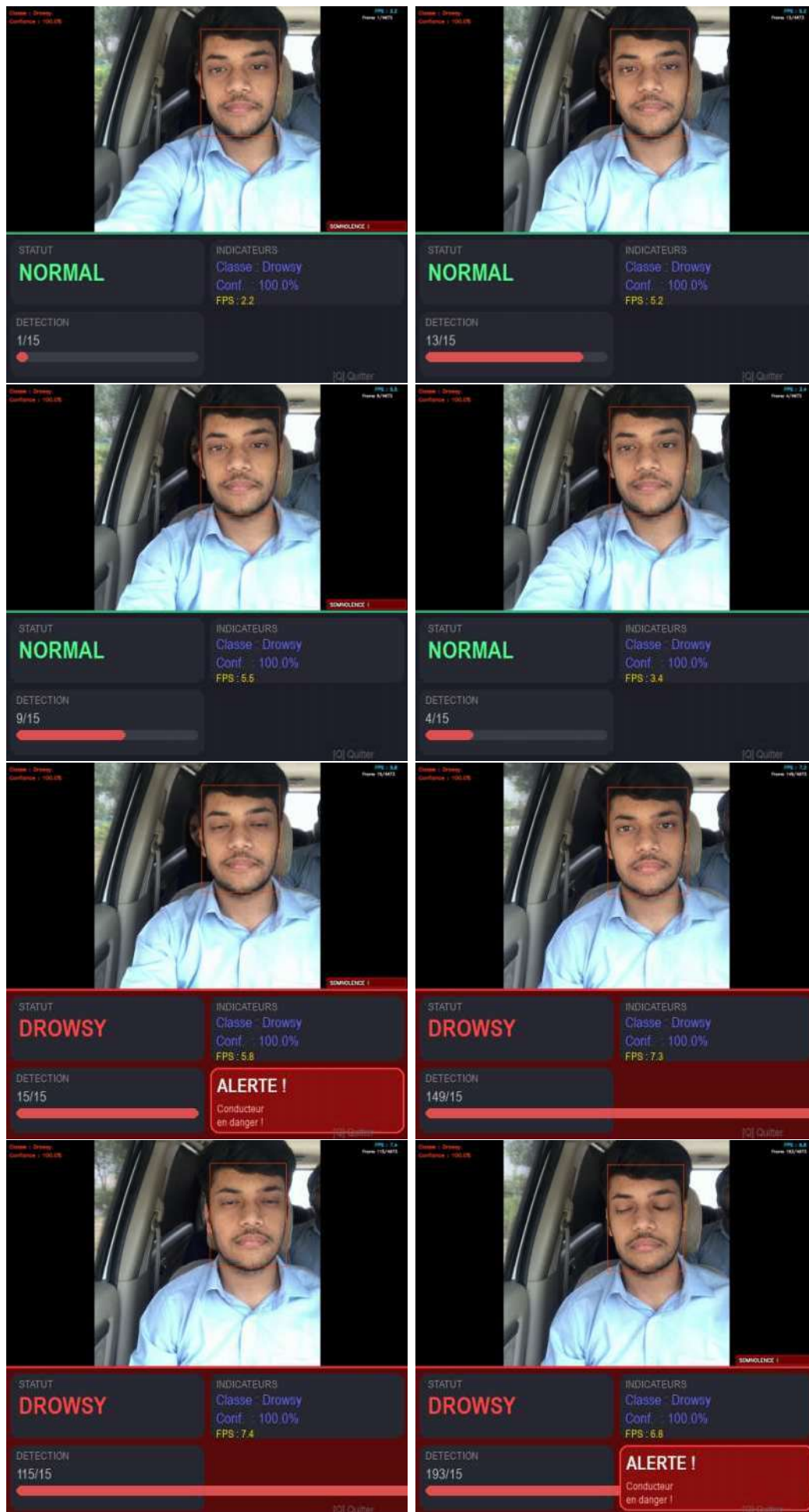


FIGURE 5.4 – Validation expérimentale du système MobileNetV2

MediaPipe a été évalué par une analyse géométrique précise des points de repère faciaux. Cette approche nous a fourni des indicateurs mathématiques stables (EAR et MAR), rendant la détection basée sur des critères géométriques précis et observables en temps réel.

L'interface basée sur MediaPipe fournit des informations plus détaillées sur le processus de détection. Comme le montre la Figure 5.5, le système affiche en temps réel les repères faciaux sur le visage du conducteur, ce qui permet de suivre avec précision les mouvements des yeux et de la bouche, ainsi que la position de la tête. L'interface affiche également un ensemble d'indicateurs utilisés pour évaluer le niveau de vigilance, dont les plus importants sont l'EAR (Eye Aspect Ratio) pour mesurer le degré d'ouverture des yeux et le MAR (Mouth Aspect Ratio) pour détecter le taux de bâillement. De plus, des informations sur l'orientation de la tête, telles que le pitch, le yaw et la pose, sont affichées afin d'améliorer la précision et la fiabilité du processus de détection. Lorsque les valeurs mesurées dépassent les seuils prédéfinis, l'état du conducteur est classé comme « somnolent » (Drowsy) et un message d'alerte s'affiche, accompagné du déclenchement de l'alarme. Cette interface permet non seulement de détecter la somnolence, mais aussi de comprendre les indicateurs et les caractéristiques qui ont conduit à cette décision.

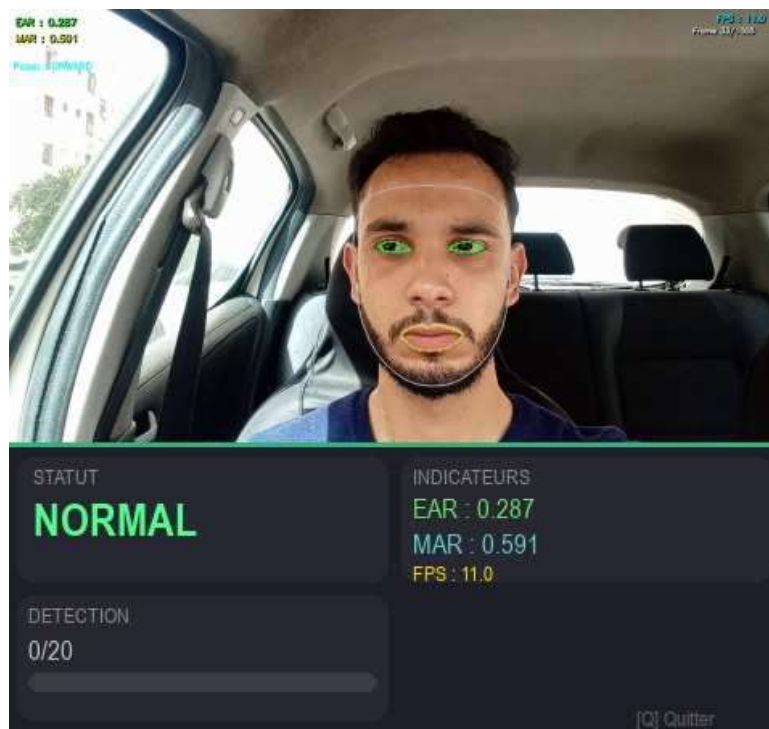


FIGURE 5.5 – Interface de model Mediapipe

Comme le montrent les images expérimentales de la figure 5.6, MediaPaPipe s'est avéré capable de reconnaître correctement les différents états du conducteur. Le système s'est appuyé sur une extraction précise des points faciaux, ce qui a permis de calculer les indices EAR et MAR de manière fiable. Lorsque les valeurs de ces indices se situent dans les limites normales ($EAR > 0,22$ et $MAR < 0,6$), l'état du conducteur est considéré comme normal et aucune alerte n'est déclenchée. En revanche, en cas de fermeture des yeux pendant un certain temps ou lors d'un bâillement, lorsque les valeurs EAR et MAR dépassent les seuils définis, le système déclenche

l'alerte sonore avec succès. Ces résultats confirment l'efficacité et la fiabilité du système dans la détection des signes de somnolence, ainsi que sa fiabilité dans des conditions de conduite réelles.

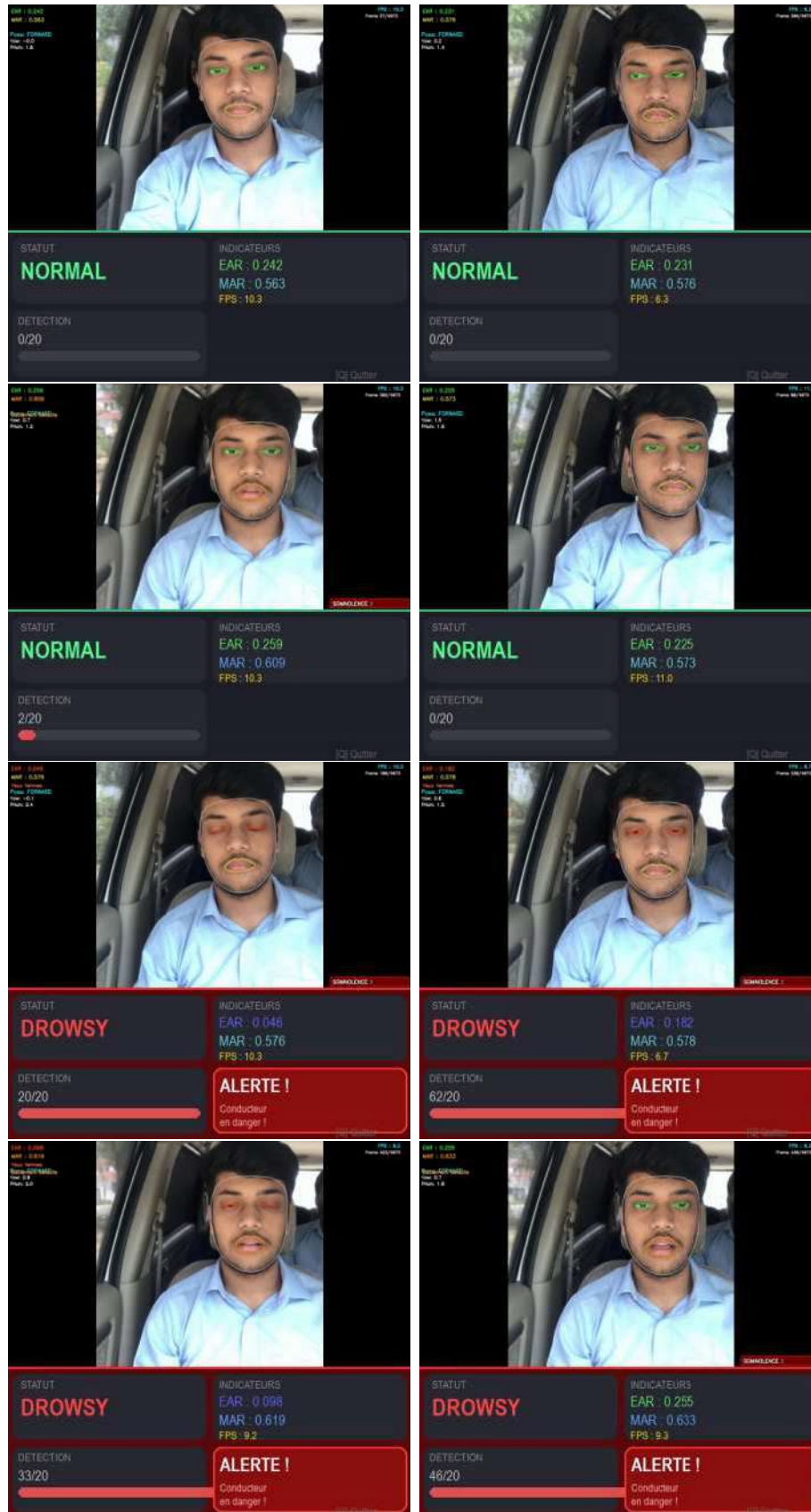


FIGURE 5.6 – Validation expérimentale du système Mediapipe

5.3.1 Performances sur Raspberry Pi

Lors de la phase de déploiement sur le matériel embarqué Raspberry Pi 4, nous nous sommes concentrés sur le modèle le plus adapté aux contraintes du temps réel au vu de ses exigences computationnelles élevées, confirmées par nos tests préliminaires sur station de travail, nous avons décidé de ne pas le déployer sur le Raspberry Pi 4 afin d'éviter tout problème de latence et d'instabilité de performance à l'intérieur du véhicule.

Concernant MediaPipe : Il a été retenu comme solution finale pour le déploiement sur Raspberry Pi 4. Les tests expérimentaux en conditions réelles à bord du véhicule ont prouvé que ce modèle est très léger, assurant un traitement vidéo d'une grande fluidité. Cette performance nous a permis d'intégrer un buzzer, générant une alerte sonore instantanée dès la détection de la somnolence. Cette réactivité en temps réel a fait de MediaPipe le choix technique optimal pour ce projet.

À ce stade, nous avons évalué la capacité du système MediaPipe à détecter correctement les signes de somnolence (fermeture prolongée des yeux et bâillements), à travers deux séries de tests réalisées dans des conditions différentes : avec et sans port de lunettes de soleil.

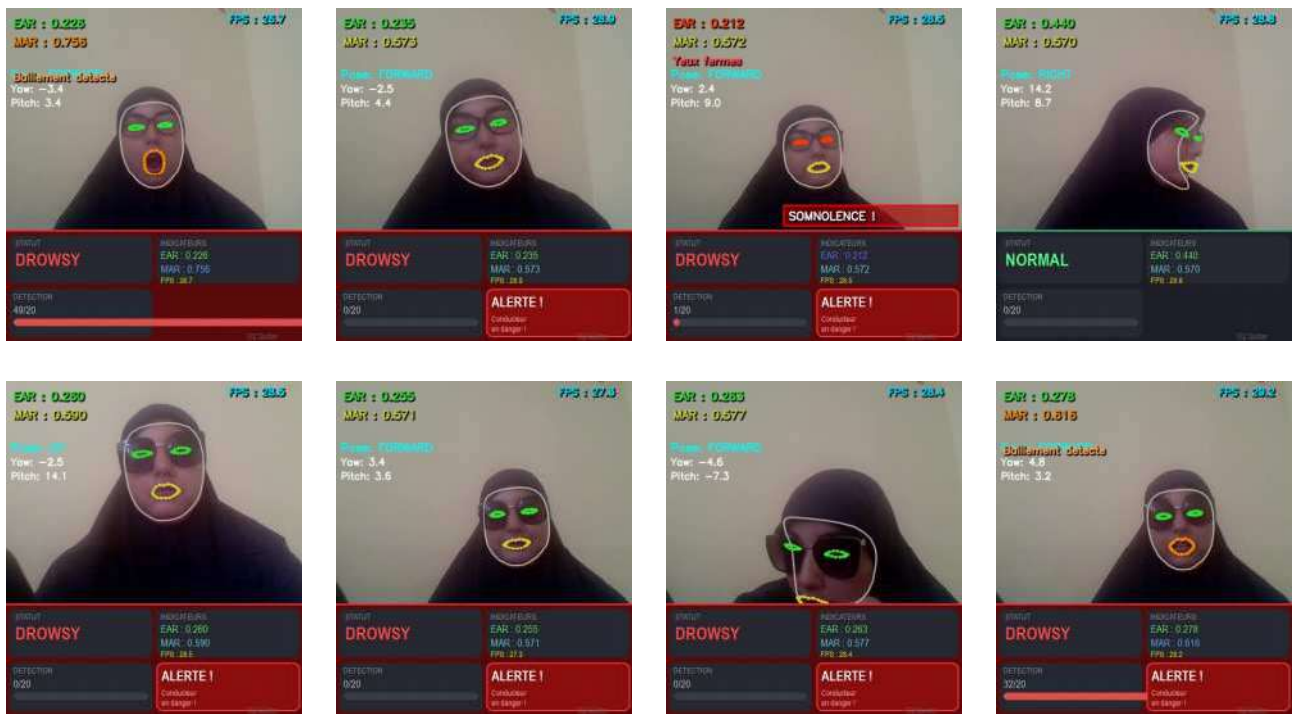


FIGURE 5.7 – Tests expérimentaux du système MediaPipe sur un sujet portant des lunettes de soleil

L'analyse de ces résultats permet de dégager deux constats.

Détections cohérentes. Dans deux des tests, une valeur de MAR supérieure au seuil (0,756 et 0,616, pour un seuil de 0,60) entraîne correctement la détection d'un bâillement. Dans un autre, une valeur d'EAR de 0,212 (inférieure au seuil de 0,22) déclenche correctement le message « SOMNOLENCE! » correspondant à une fermeture des yeux. Enfin, un test pris en vue de

profil (pose *RIGHT*, yaw = 14,2) est correctement classé comme « *NORMAL* », avec des valeurs d'EAR et de MAR toutes deux dans la plage normale.

En revanche, quatre des tests affichent un statut *DROWSY* accompagné du déclenchement de l'alerte, alors que les valeurs d'EAR (entre 0,235 et 0,263) et de MAR (entre 0,571 et 0,590) affichées restent dans la plage considérée comme normale, et que le compteur de détection reste à 0/20. Ce comportement traduit une incohérence entre les indicateurs numériques affichés et la décision finale du système. Le point commun entre ces quatre cas est le port de lunettes de soleil, qui semble perturber l'estimation interne des points de repère oculaires sans que cela soit visible sur la valeur globale de l'EAR affichée à l'écran.

Sur le plan des performances temps réel, le FPS reste stable et élevé sur l'ensemble des huit tests (entre 26,7 et 29,2 images par seconde), quels que soient l'état détecté ou l'orientation du visage, ce qui confirme la légèreté du modèle MediaPipe sur Raspberry Pi 4.

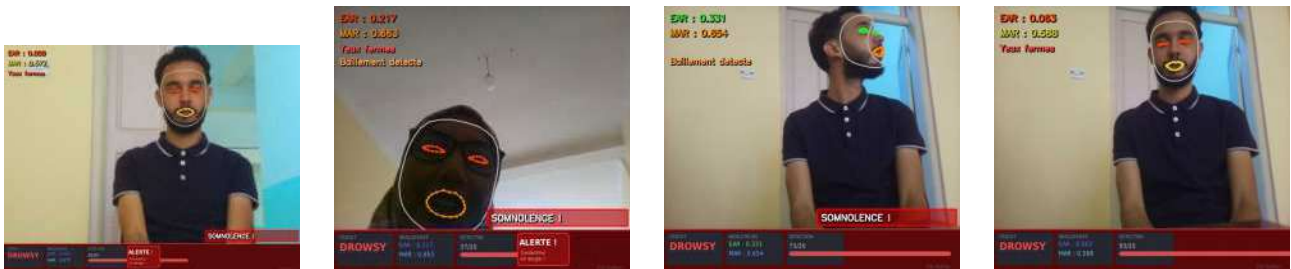


FIGURE 5.8 – Tests expérimentaux du système MediaPipe sans port de lunettes : fermeture prolongée des yeux et bâillement, sous différents angles de tête

La figure 5.8 illustre le comportement du système en l'absence de lunettes, dans des conditions de pose plus extrêmes que les tests précédents. Les résultats obtenus se révèlent cette fois pleinement cohérents avec la logique des seuils définis.

Dans le premier test, une fermeture prolongée des yeux entraîne une valeur d'EAR très faible (0,058), largement inférieure au seuil de 0,22, ce qui déclenche correctement le message « *SOMNOLENCE!* » ainsi que l'alerte sonore, avec un compteur de détection atteignant 68/20.

Le deuxième test, réalisé avec une orientation de tête extrême (regard vers le haut), confirme la robustesse du système : malgré cet angle inhabituel, le visage reste correctement détecté, et la combinaison d'un EAR de 0,217 (yeux fermés) et d'un MAR de 0,663 (bâillement) déclenche simultanément les deux indicateurs, avec un compteur de détection de 37/20.

Le troisième test, en vue de profil avec la tête inclinée vers l'arrière, montre également une détection correcte du bâillement (MAR = 0,654), confirmant que le système reste fonctionnel même dans des poses non frontales.

Enfin, le quatrième test reproduit une fermeture prolongée des yeux (EAR = 0,063) avec un compteur de détection atteignant 93/20, la valeur la plus élevée observée parmi l'ensemble des tests, traduisant une fermeture des yeux maintenue sur une longue durée.

Contrairement aux tests réalisés avec lunettes de soleil (figure 5.7), où des faux positifs apparaissaient malgré des indicateurs numériques normaux, l'ensemble de ces quatre tests présente une parfaite cohérence entre les valeurs d'EAR/MAR affichées et le statut final attribué par le système, y compris sous des angles de tête prononcés. Ceci confirme que la principale source

de dégradation des performances de MediaPipe Face Mesh provient bien du port de lunettes, et non de l'orientation de la tête.

5.3.2 Journalisation des données (log CSV)

Le système enregistre en continu les indicateurs EAR, MAR, l'orientation de la tête et le statut de vigilance dans un fichier `drowsy_log.csv`, selon le format suivant :

```
datetime,ear,mar,head_pose,status
2026-06-20 16:11:22,0.231,0.567,FORWARD,Normal
```

TABLEAU 5.1 – Extrait du fichier de log (`drowsy_log.csv`).

Datetime	EAR	MAR	Head pose	Status
2026-06-20 16 :11 :22	0,231	0,567	FORWARD	Normal
2026-06-20 16 :11 :22	0,237	0,561	FORWARD	Normal
2026-06-20 16 :11 :22	0,228	0,566	FORWARD	Normal
2026-06-20 16 :11 :23	0,136	0,571	FORWARD	Normal
2026-06-20 16 :11 :23	0,116	0,565	FORWARD	Normal
2026-06-20 16 :11 :23	0,192	0,563	FORWARD	Normal

La journalisation continue des données (`datetime`, `EAR`, `MAR`, `head_pose`, `status`) dans un fichier CSV présente plusieurs intérêts pour le système :

- **Traçabilité** : conserver un historique complet des indicateurs de vigilance permet de reconstituer le déroulement d'un trajet et de vérifier le comportement du système a posteriori.
- **Débogage et validation** : les logs facilitent l'analyse des cas de faux positifs ou faux négatifs en confrontant les valeurs EAR/MAR enregistrées au statut détecté.
- **Ajustement des seuils** : l'étude statistique des valeurs loguées (EAR, MAR) sur des trajets réels permet d'affiner les seuils de détection (0,22 / 0,60) pour réduire les fausses alertes.
- **Constitution de données d'entraînement** : les séquences enregistrées peuvent enrichir un futur jeu de données pour ré-entraîner ou améliorer le modèle.
- **Preuve et analyse a posteriori** : en cas d'incident, le journal constitue une trace exploitable de l'état de vigilance du conducteur au moment des faits.

5.4 Discussion

La comparaison entre les deux approches révèle des profils de performance complémentaires. MediapipeV2 excelle par sa précision de classification pure, grâce à son architecture de réseau profond, ce qui en fait un outil puissant pour l'analyse de données complexes. Toutefois, sa lourdeur computationnelle le rend difficile à exploiter directement sur Raspberry Pi 4 pour une application critique en temps réel. À l'inverse, MediaPipe se révèle être la solution la plus

équilibrée. Bien qu'il repose sur une approche différente de la classification classique, il offre une réactivité quasi instantanée sur le matériel embarqué. Pour un système de sécurité devant alerter le conducteur sans aucun retard, la fluidité de MediaPipe constitue un atout décisif. Afin de finaliser ce système d'alerte, un buzzer a été intégré au Raspberry Pi ; ce dernier est activé automatiquement dès que le système détecte des signes de somnolence, garantissant ainsi une réponse sonore immédiate et efficace.

Bien que les résultats obtenus soient probants dans des conditions standards, notre système présente certaines limites liées à l'environnement réel :

Contraintes environnementales : L'efficacité de la détection est fortement dépendante de la qualité de l'éclairage. En effet, lors de la conduite nocturne ou dans une voiture mal éclairée, le système perd en précision.

Facteurs humains : Le port de lunettes (médicales ou de soleil) peut gêner le suivi précis des points de repère faciaux. De même, les mouvements brusques du conducteur peuvent induire des erreurs de détection temporaires. Pour répondre à ces défis, nous proposons les pistes d'amélioration suivantes :

Approche hybride : Développer un système combinant la robustesse de MediapipeV2 pour la précision et la rapidité de MediaPipe pour le traitement en temps réel.

Adaptation matérielle : Intégrer une caméra infrarouge afin de garantir une surveillance continue et précise, même dans l'obscurité totale, palliant ainsi les limites de l'éclairage naturel.

5.5 Conclusion

En conclusion, ce chapitre a permis de valider deux approches distinctes. Bien que MediaPipeV2 offre de bonnes performances en termes de classification, il n'a pas donné de résultats satisfaisants en temps réel, avec une exécution instable et une performance limitée par rapport à MediaPipe.

À l'inverse, MediaPipe s'est distingué par sa fluidité et son efficacité sur Raspberry Pi 4, grâce à sa faible consommation de ressources et à son traitement en temps réel basé sur les landmarks du visage. Ces caractéristiques en font la solution la plus adaptée à notre système embarqué.

Conclusion Générale

Au terme de ce travail, nous avons conçu et réalisé un système intelligent de détection de la somnolence au volant basé sur les techniques de vision par ordinateur et l'intelligence artificielle. L'objectif principal était de développer une solution embarquée capable de surveiller l'état de vigilance du conducteur en temps réel afin de contribuer à la prévention des accidents liés à la fatigue.

Dans un premier temps, une étude théorique a permis de mieux comprendre les causes et les conséquences de la somnolence au volant, ainsi que les différentes approches utilisées dans la littérature pour sa détection. Cette analyse a mis en évidence l'intérêt des méthodes basées sur l'observation du comportement du conducteur à travers les caractéristiques faciales.

Par la suite, deux approches ont été développées et évaluées. La première repose sur un modèle de classification d'images basé sur l'architecture MobileNetV2, tandis que la seconde exploite MediaPipe pour l'extraction et l'analyse des points de repère faciaux. Afin de permettre une exécution efficace sur une plateforme embarquée à ressources limitées, le modèle MobileNetV2 a été optimisé à l'aide d'une quantification en INT8, réduisant ainsi sa taille et son temps d'inférence tout en conservant des performances satisfaisantes.

L'implémentation pratique du système a été réalisée sur une carte Raspberry Pi 4 associée à une caméra et à un dispositif d'alerte sonore. Les expérimentations menées ont démontré la faisabilité de la solution proposée ainsi que sa capacité à détecter l'état de somnolence du conducteur en temps réel. Les résultats obtenus montrent également que l'utilisation de modèles légers de Deep Learning et d'outils de vision par ordinateur constitue une approche pertinente pour les systèmes embarqués dédiés à la sécurité routière.

Enfin, ce travail ouvre plusieurs perspectives d'amélioration. Parmi celles-ci figurent l'intégration d'approches hybrides combinant les avantages de MobileNetV2 et de MediaPipe, l'amélioration de la robustesse du système face aux variations d'éclairage et aux mouvements du conducteur, ainsi que l'évaluation de la solution sur un ensemble plus large d'utilisateurs et dans des conditions de conduite plus diversifiées. Ces perspectives permettront de renforcer davantage la fiabilité et l'efficacité du système dans des applications réelles de surveillance du conducteur.

Bibliographie

- [1] Association Prévention Routière. Somnolence au volant, July 2024. Article mis à jour le 23 Juil 2024.
- [2] Muhammad Kamran, Malik Muhammad Naeem Mannan, and Myung Jeong. Drowsiness, fatigue and poor sleep's causes and detection : A comprehensive study. *IEEE Access*, PP :1–1, 11 2019.
- [3] Arun Sahayadhas, Kenneth Sundaraj, and Murugappan Murugappan. Detecting driver drowsiness based on sensors : a review. *Sensors*, 12(12) :16937–16953, 2012.
- [4] Cleveland Clinic. Somnolence (drowsiness) : What it is, causes treatment, 2025. [en ligne ; consulté le 27 mars 2026].
- [5] A. Righi. Détection de la fatigue du conducteur avec jetson nano. Mémoire de master, Université 08 Mai 1945, Guelma, Algérie, 2024.
- [6] Willem B Verwey and David M Zaidel. Predicting drowsiness accidents from personal attributes, eye blinks and ongoing driving behaviour. *Personality and individual differences*, 28(1) :123–142, 2000.
- [7] Rachel Korson and Christian Guilleminault. Obstructive sleep apnea syndrome. In *Sleep Disorders Medicine : Basic Science, Technical Considerations and Clinical Aspects*, pages 567–596. Springer, 2017.
- [8] Assurance Prévention. Conduite sur autoroute : comment prévenir les risques ?, 2023. [en ligne ; consulté le 27 mars 2026].
- [9] Terapiz. Sommeil léger : causes, symptômes et solutions, 2024. Consulté le 27 mars 2026.
- [10] Gemma Pastor Cerezuela, Pilar Tejero, Mariano Chóliz, Mauricio Chisvert, and M José Monteagudo. Wertheim's hypothesis on 'highway hypnosis' : empirical evidence from a study on motorway and conventional road driving. *Accident Analysis & Prevention*, 36(6) :1045–1054, 2004.
- [11] Bruno Claustrat, Jocelyne Brun, and Guy Chazot. The basic physiology and pathophysiology of melatonin. *Sleep medicine reviews*, 9(1) :11–24, 2005.
- [12] Stage de sensibilisation. La somnolence au volant : signes, risques et prévention, 2024. Consulté le 27 mars 2026.
- [13] Tecnic. Fatigue au volant : rester concentré, 2024. Consulté le 27 mars 2026.

- [14] Walter W Wierwille, SS Wreggit, CL Kirn, LA Ellsworth, and RJ Fairbanks. Research on vehicle-based driver status/performance monitoring ; development, validation, and refinement of algorithms for detection of driver drowsiness. final report. Technical report, 1994.
- [15] Jim Horne and Louise Reyner. Vehicle accidents related to sleep : a review. *Occupational and environmental medicine*, 56(5) :289–294, 1999.
- [16] World Health Organization. Global road safety report 2024. Technical report, World Health Organization, 2024. Global status report on road safety.
- [17] World Health Organization. Sleepiness and road traffic accidents : A systematic review. Technical report, World Health Organization, 2023. Systematic review on sleepiness and road safety.
- [18] World Health Organization EMRO. Risk assessment of road traffic accidents related to sleepiness during driving. Technical report, World Health Organization Regional Office for the Eastern Mediterranean, 2024. Regional risk assessment report.
- [19] National Highway Traffic Safety Administration. Drowsy driving statistics. Technical report, NHTSA, 2017. Crash data and statistics related to drowsy driving.
- [20] AAA Foundation for Traffic Safety. 2024 traffic safety report. Technical report, AAA Foundation for Traffic Safety, 2024. Annual report on traffic safety issues including drowsy driving.
- [21] Arrive Alive Road Safety. Facts about drowsy driving, 2024. Overview and statistics on drowsy driving.
- [22] Délégation nationale à la sécurité routière (DNSR). Road safety statistics 2024, 2024. Statistiques de sécurité routière 2024, cité par Atlas Magazine.
- [23] National Road Safety Delegation. Road safety statistics in algeria. Technical report, Ministry of Interior, Algeria, 2024. Official Algerian road safety statistics.
- [24] Délégation Nationale à la Sécurité Routière. Statistiques des accidents de la route en algérie 2024. Technical report, Ministère de l’Intérieur, Algérie, 2024. Rapport officiel sur la sécurité routière en Algérie.
- [25] Gendarmerie Nationale. Bilan des accidents de la circulation en algérie. Technical report, Ministère de la Défense Nationale, 2024. Rapports statistiques sur les accidents de la route.
- [26] Direction Générale de la Sûreté Nationale. Statistiques des accidents de la circulation en algérie. Technical report, DGSN, 2024. Données officielles sur les accidents de la route.
- [27] European Commission. General safety regulation (eu) 2019/2144, 2019. Regulation requiring driver monitoring systems in new vehicles from 2024.
- [28] Ministry of Land, Infrastructure, Transport and Tourism. Road transport vehicle act, 2023. Japanese regulations including driver monitoring requirements.
- [29] Standardization Administration of China. Driver monitoring system technical requirements (gb/t 40429-2021), 2021. Chinese national standard for driver monitoring systems.

-
- [30] European Commission. Road safety in the european union. Technical report, 2023. Driver fatigue involved in 20-25% of accidents.
 - [31] VIAS Institute. Road safety barometer belgium. Technical report, 2023. Survey on driver behaviour including fatigue.
 - [32] Fondation VINCI Autoroutes. Baromètre de la conduite responsable. Technical report, 2024. Study on driver drowsiness in France.
 - [33] Y. Dong, Z. Hu, K. Uchimura, and N. Murayama. Driver inattention monitoring system for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 12(2) :596–614, 2011.
 - [34] J. Horne and L. Reyner. Sleep-related vehicle accidents. *BMJ*, 310 :565–567, 1996.
 - [35] M. Ingre, T. Akerstedt, B. Peters, A. Anund, and G. Kecklund. Subjective sleepiness and accidents related to shifts and time of day. *Sleep*, 29 :1518–1524, 2006.
 - [36] K. Nakayama and et al. Driver drowsiness detection based on eye and head movement. *IEICE TRANSACTIONS on Information and Systems*, E82-D(10) :2030–2037, 1999.
 - [37] S. H. Fairclough and R. Graham. Prediction of driver drowsiness using psychophysiological measures. *Biological Psychology*, 50 :139–148, 1999.
 - [38] Sultan Alhazmi. *Towards Context-based Fatigue Detection System in Vehicular Area Network*. PhD thesis, University of Ottawa, 2013. ProQuest Dissertations & Theses, MR95555.
 - [39] G. Perrotte. *Apport des indicateurs physiologiques et posturaux pour la détection de la somnolence au volant en conduite partiellement autonome*. PhD thesis, Aix-Marseille Université, 2024.
 - [40] M Arunasalam, N Yaakob, A Amir, M Elshaikh, and NF Azahar. Real-time drowsiness detection system for driver monitoring. In *IOP Conference Series : Materials Science and Engineering*, volume 767, page 012066. IOP Publishing, 2020.
 - [41] Reza Ghoddoosian, Marnim Galib, and Vassilis Athitsos. A realistic dataset and baseline temporal model for early drowsiness detection. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.
 - [42] A. Williamson, D. A. Lombardi, and S. Folkard. The link between fatigue and safety. *Accident Analysis & Prevention*, 43 :498–515, 2011.
 - [43] Yanchao Dong, Zhencheng Hu, Keiichi Uchimura, and Nobuki Murayama. Driver inattention monitoring system for intelligent vehicles : A review. *IEEE transactions on intelligent transportation systems*, 12(2) :596–614, 2010.
 - [44] Shabnam Abtahi, Mona Omidyeganeh, Shervin Shirmohammadi, and Behnoosh Hariri. Yawdd : A yawning detection dataset. In *Proceedings of the 5th ACM multimedia systems conference*, pages 24–28, 2014.
 - [45] Areg Mikael Sarvazyan, José Ángel González, Marc Franco-Salvador, Francisco Rangel, Berta Chulvi, and Paolo Rosso. Overview of autextification at iberlef 2023 : Detection and attribution of machine-generated text in multiple domains. *arXiv preprint arXiv :2309.11285*, 2023.

- [46] Niall O'Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. Deep learning vs. traditional computer vision. In *Science and information conference*, pages 128–144. Springer, 2019.
- [47] Henri Parinaud. *La vision ; étude physiologique*. Doin, 1898.
- [48] Siham Essahraoui, Ismail Lamaakal, Ikhlas El Hamly, Yassine Maleh, Ibrahim Ouahbi, Khalid El Makkaoui, Mouncef Filali Bouami, Paweł Pławiak, Osama Alfarraj, and Ahmed A Abd El-Latif. Real-time driver drowsiness detection using facial analysis and machine learning techniques. *sensors*, 25(3) :812, 2025.
- [49] Reza Ghoddoosian, Marnim Galib, and Vassilis Athitsos. A realistic dataset and baseline temporal model for early drowsiness detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.
- [50] Ank Zaman, Prosenjit Chatterjee, and Rajat Sharma. Real-time drivers' drowsiness detection : A deep learning approach. In *2024 2nd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*, pages 1–5. IEEE, 2024.
- [51] Manuel JCS Reis, Carlos Serôdio, and Frederico Branco. Edge-visionguard : A lightweight signal-processing and ai framework for driver state and low-visibility hazard detection. *Applied Sciences*, 16(2) :1037, 2026.
- [52] Lie Yang, Henglai Wei, Zhongxu Hu, and Chen Lv. A domain generalization method for deploying driver distraction detection models to practical application scenarios. *Engineering Applications of Artificial Intelligence*, 152 :110844, 2025.
- [53] Quentin Massoz, Thomas Langohr, Clémentine François, and Jacques G Verly. The ulg multimodality drowsiness database (called drozy) and examples of use. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–7. IEEE, 2016.
- [54] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd : Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.
- [55] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once : Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [56] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe : A framework for building perception pipelines. *arXiv preprint arXiv :1906.08172*, 2019.
- [57] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*, 2014.
- [58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- [59] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets : Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv :1704.04861*, 2017.
- [60] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2 : Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- [61] Ismail Nasri. Driver drowsiness dataset (DDD). Kaggle, 2021. Consulté le 14 Mars 2026.