

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE DE
LA MAÎTRISE EN MATHÉMATIQUES ET
INFORMATIQUE APPLIQUÉES

PAR
NDARAW SECK

DETECTION AUTOMATIQUE DE DISTRACTION ET DE SOMNOLENCE AU
VOLANT PAR RESEAUX DE NEURONES CONVOLUTIFS (CNN)

JANVIER 2026

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

REMERCIEMENTS

Je remercie le bon Dieu, tout puissant de m'avoir donné la force de mener à terme ce travail.

Je tiens à remercier grandement Mr François Meunier d'avoir accepté d'être mon directeur de recherche, son engagement, sa disponibilité ainsi que son soutien pour la réussite de ce projet. Et il a surtout développé en moi une capacité de recherche et d'engagement.

Je souhaiterais manifester ma reconnaissance particulièrement à la secrétaire, Madame Guimond Chantal pour tout le travail remarquable qu'elle a eu à faire pour les étudiants durant ces deux années de formation.

J'adresse également mes sincères remerciements à tous les membres du département de Mathématiques et Informatique de l'Université du Québec à Trois-Rivières (UQTR) qui ont toujours été disponibles pour m'aider à réaliser mon cursus universitaire au programme de maîtrise en mathématiques et informatique appliquées.

Résumé

La somnolence et la distraction au volant constituent des défis majeurs pour la sécurité routière, entraînant une réduction significative des réflexes des conducteurs, augmentant les distractions et risquant de causer des accidents graves, voire mortels. Parmi les diverses causes de somnolence figurent le manque de sommeil, des horaires de travail irréguliers, des troubles du sommeil comme l'apnée, ainsi que la consommation de certains médicaments ou substances [\[3\]](#).

L'utilisation des téléphones portables au volant représente un facteur aggravant, établissant la distraction comme la principale cause d'accidents de la route. Face à cette problématique, le développement d'un système de détection et d'avertissement de la somnolence et de la distraction devient crucial pour améliorer la sécurité automobile et prévenir les accidents.

Dans ce projet, nous concevons un tel système à partir d'une caméra, et ce en exploitant la détection des yeux, la reconnaissance des signes de somnolence et le suivi du schéma de clignement des yeux en même temps inclure la détection de l'usage du téléphone au volant grâce à des techniques avancées de vision par ordinateur.

L'efficacité de notre modèle est évaluée grâce à un ensemble de données composé d'images de conducteurs en situation de conduite. Ainsi, les résultats obtenus mettent en évidence la pertinence de notre système pour identifier la somnolence et la distraction, contribuant ainsi à la sécurité routière.

Les mots-clés sont : vision par ordinateur, détection de somnolence, distraction au volant, apprentissage profond, MobileNetV2

TABLE DES MATIERES

REMERCIEMENTS	II
Résumé	III
LISTE DES FIGURES	VI
LISTE DES TABLEAUX	IX
Chapitre 1	11
Introduction générale	11
Chapitre 2	15
Techniques de détection de la somnolence et la distraction chez les conducteurs	15
2.1 Introduction.....	15
2.2 Relation entre la somnolence, distraction et la conduite.....	15
2.3 Système de détection de la somnolence chez le conducteur	16
2.3.1 Introduction aux expressions faciales	17
2.3.2 Définition de visage	18
2.3.3 Les étapes préliminaires à la reconnaissance faciale.....	18
2.3.3.1 Définition de la détection faciale.....	19
2.3.3.2 Détection de visage	20
2.3.3.3 Extraction des caractéristiques du visage	21
2.3.4 Techniques de détection de la somnolence des conducteurs	21
2.3.4.1 Techniques basées sur le traitement d'images.....	24
2.3.4.1.1 Techniques de correspondance des modèles.....	25
2.3.4.1.2 Technique basée sur le clignement des yeux	25
2.3.4.1.3 Technique basée sur le bâillement.....	27
2.3.4.2 Technique basée sur le signal EEG.....	30
2.3.4.3 Techniques basées sur les réseaux de neurones artificiels.....	33
2.4 Conclusion	34
Chapitre 3	35
Principes de l'Apprentissage Automatique et Profond : Réseaux de Neurones et Architectures Avancées	35
3.1 Introduction.....	35
3.2 L'apprentissage automatique	36
3.2.1 Les phases de l'apprentissage automatique.....	36
3.2.2 Types d'apprentissage automatique	37
3.2.2.1 Apprentissage supervisé (Supervised learning)	38
3.2.2.2 Apprentissage non supervisé (Unsupervised learning).....	39
3.2.2.3 Apprentissage par renforcement (Reinforcement learning).....	40
3.3 L'apprentissage profond.....	40
3.3.1 Classification des méthodes d'apprentissage profond.....	41
3.3.1.1 Réseaux de neurones récurrents	41
3.3.1.2 Réseaux de neurones artificiels	42
3.3.1.3 Réseaux de neurones convolutifs	44
3.3.1.3.1 L'architecture réseaux de neurones convolutifs	45

3.3.1.3.2 Architectures CNN populaires	50
3.3.1.3.3 L'architecture MobileNetV2	55
3.4 Travaux connexes à la détection de la somnolence	57
3.5 Conclusion	58
Chapitre 4	59
Conception de l'architecture pour la détection de la somnolence et la distraction	59
4.1 Introduction.....	59
4.2 Objectif	59
4.3 L'architecture générale	60
4.3.1 L'architecture détaillée du système de détection de la somnolence et distractions	61
4.3.2 Préparation des données	62
4.3.2.1 MobileNetV2	63
4.3.2.2 Eye_aspect_ratio(eye)	67
4.3.3 Fonctionnalité eye_aspect_ratio(eye).....	68
4.4 Détection de la somnolence et la distraction.....	70
4.4.1 Localisation des yeux	72
4.4.2 Détection de distraction.....	73
4.4.3 Détection de la somnolence par ratio aspectuel des yeux (EAR)	73
4.5 Enregistrement des événements et captures	74
4.6 Reconnaissance faciale	75
4.7 Conclusion	75
Chapitre 5	77
Implémentation et résultats	77
5.1 Introduction.....	77
5.2 Les outils	77
5.2.1 Le matériel	77
5.2.2 Environnements et outils de développement.....	78
5.2.2.1 Python.....	78
5.3 Instructions d'utilisation.....	80
5.4 Conclusion	99
Chapitre 6	100
Conclusion générale	100
Références.....	102

LISTE DES FIGURES

Figure 2.1 : Muscles faciaux et leur contrôle nerveux [4].	19
Figure 2.2 : Architecture d'un système de reconnaissance des expressions faciales.	20
Figure 2.3 : Représentation en 68 points de repères faciaux (facial Landmark) utiliser pour la détection et l'analyse des expressions faciales [59].	21
Figure 2.4 : Illustration du processus de détection du visage dans une image numérique [76].	22
Figure 2.5 : Les différentes méthodes de détection de somnolence utilisées.	23
Figure 2.6 : Illustration de la détection et de l'analyse des repères faciaux par traitement d'images [82].	24
Figure 2.7 : Représentation conceptuelle d'un réseau de neurones artificiels appliqué à l'analyse faciales [83].	25
Figure 2.8 : Illustration des capteurs utilisés dans les méthodes axées sur la performance d'un conducteur [84].	26
Figure 2.9 : Surveillance du visage du conducteur par techniques de traitement d'images [97].	26
Figure 2.10 : Exemple d'analyse du visage par correspondance des modèles [93].	27
Figure 2.11 : Détection de la somnolence par l'analyse du clignement des yeux [85].	28
Figure 2.12 : Détection du bâillement comme indicateur de somnolence d'un conducteur [87].	30
Figure 2.13 : Illustration des mouvements faciaux associés à un bâillement [96].	31
Figure 2.14 : Illustration des contours supérieurs et inférieurs et la structure du bâillement [64].	32
Figure 2.15 : Casque à électrodes EEG [12].	33
Figure 2.16 : Les cinq bandes de fréquences [14].	34
Figure 2.17 : Réseau de neurones artificiels [88].	35
Figure 3.1 : Les principaux types d'apprentissage automatique [24].	39
Figure 3.2 : Processus de classification et de régression [27].	40
Figure 3.3 : Création de clusters à partir des données [28].	41
Figure 3.4 : Illustration de l'architecture de Deep Learning [19].	42
Figure 3.5 : Architecture des réseaux de neurones récurrents [21].	43
Figure 3.6 : L'architecture d'un réseau de neurones artificiels [88].	44
Figure 3.7 : L'architecture des réseaux neuronaux artificiels avec ses composantes [33].	45
Figure 3.8 : Architecture et composition d'un réseau de neurones convolutifs [91].	47

Figure 3.9 : Processus de convolution [37].	48
Figure 3.10 : Autre application d'un processus de convolution [37].	48
Figure 3.11 : Représentation graphique de la fonction d'activation ReLu [39].	50
Figure 3.12 : Types de fonctions pooling [41].	51
Figure 3.13 : Schéma fonctionnel du CNN LeNet [31].	52
Figure 3.14 : Schéma fonctionnel de l'architecture du réseau de neurone convolutif Alexnet [45].	53
Figure 3.15 : Schéma fonctionnel de l'architecture du réseau de neurone convolutif ZFNet [44].	53
Figure 3.16 : Schéma fonctionnel de l'architecture du réseau de neurones convolutifs GoogleNet avec toutes ses caractéristiques [46].	54
Figure 3.17 : L'architecture du réseau de neurone convolutif VGGNet [47].	55
Figure 3.18 : Architecture du module Inception [48].	56
Figure 3.19 : L'architecture du réseau résiduel ResNet [49].	57
Figure 3.20 : Architecture détaillée de MobileNetV2 [57].	58
Figure 3.21 : L'architecture de détecteur d'objets [58].	58
Figure 4.1 : L'architecture générale du système de détection de la somnolence et la distraction.	62
Figure 4.2 : Description des étapes de la détection de la somnolence et de la distraction.	64
Figure 4.3 : Image illustre un exemple de conducteur en état d'éveil, servant de référence pour la comparaison avec les états de somnolence et de distraction.	66
Figure 4.4 : Image illustrant une conductrice présentant des signes de somnolence, caractérisés par la fermeture prolongée des paupières et une diminution de l'ouverture oculaire.	66
Figure 4.5 : Image illustre une conductrice servant de référence en état d'éveil normal, caractérisés par une ouverture oculaire complète et une expression faciale stable.	67
Figure 4.6 : Image illustrant une conductrice en état de somnolence effectuant un bâillement.	67
Figure 4.7 : Image illustrant un exemple de conducteur servant de référence dont le regard est orienté vers l'avant, correspondant à une posture normale de conduite et à un état d'attention visuelle adéquat.	68
Figure 4.8 : Image montrant un conducteur dont le regard est détourné de l'axe frontal (état de distraction) de la route, orienté vers la gauche en direction de la caméra.	68
Figure 4.9 : Architectures de MobileNetV2 couplée à l'analyse du Ratio Aspectuel des Yeux (EAR) pour la détection de la somnolence et la distraction du conducteur.	69
Figure 4.10 : Points de repères de référence extraits avec la fonctionnalité eye_aspect_ratio(eye) [94].	70
Figure 4.11 : Les points de repères autour des yeux [95].	70

Figure 4.13 : Localisation du visage et de la région oculaire du conducteur.....	74
Figure 4.14 : Situation d'ouverture et de fermeture des yeux.	75
Figure 4.15 : Captures d'évènements enregistrés.	76
Figure 4.16 : Conducteurs en somnolence	77
Figure 5.1 : Importation des bibliothèques.	83
Figure 5.2 : Définition des variables et paramètres.	84
Figure 5.3 : Fonction de prétraitement.	84
Figure 5.4 : Fonction eye_aspect_ratio ()......	85
Figure 5.5 : Fonction d'identification du conducteur.	86
Figure 5.6 : Fonction detect_distraction ().	86
Figure 5.7 : Boucle principale d'exécution.	87
Figure 5.8 : Détection et alerte de somnolence.	88
Figure 5.9 : Alerte de distraction.	88
Figure 5.10 : Identification de conducteur.	88
Figure 5.11 : Image montrant le conducteur dans un état d'éveil normal avec contour.	90
Figure 5.12 : Images montrant le conducteur dans un état d'éveil normal sans contour.	91
Figure 5.13 : Images montrant le conducteur en état de somnolence avec contour.	92
Figure 5.14 : Images montrant le conducteur en état de somnolence sans contour.	93
Figure 5.15 : Images montrant le conducteur effectuant un bâillement.	94
Figure 5.16 : Images montrant le conducteur dans un état d'éveil.	95
Figure 5.17 : Images montrant un cas de distraction détectée automatiquement par le système lors des expérimentations.	95
Figure 5.18 : Image montrant un conducteur distrait par l'utilisation d'un téléphone portable.	96
Figure 5.19 : Image montrant un conducteur distrait lors d'un appel téléphonique au volant.	97

LISTE DES TABLEAUX

Tableau 5.1 : Comparaison des performances du système de détection de la somnolence et de la distraction.....	96
Tableau 5.2 : Matrice de confusion globale du système de détection de la somnolence et de la distraction obtenus pour les quatre conducteurs.....	96
Tableau 5.3 : Évaluation quantitative des performances du système de détection de la somnolence et de la distraction.....	97
Tableau 5.4 : Indicateurs globaux du système de détection de la somnolence et de la distraction. ..	97

Chapitre 1

Introduction générale

La somnolence est un état qui précède le sommeil ou encore un fort désir de sommeil. La somnolence peut être dangereuse lorsqu'un certain niveau de concentration est nécessaire tel que lors de la conduite de véhicules motorisés. La somnolence n'est pas seulement ressentie lorsqu'il y a de la fatigue, mais elle peut l'être lorsque la personne est sous l'influence de drogues, de médicaments ou d'alcool.

Une étude menée par la Fondation pour la sécurité routière de l'Association américaine des automobilistes a mis en lumière que la fatigue du conducteur contribue de 16 % à 21 % des accidents de la route [1]. Par ailleurs, la distraction au volant a été identifiée par la National Highway Traffic Safety Administration (US, 2010) comme la cause principale des accidents de la route, représentant 80 % des collisions. L'utilisation des téléphones portables accentue ce phénomène, étant impliquée dans 26 % des collisions [1].

Cette situation montre l'importance cruciale de détecter les signes de fatigue pour prévenir les conséquences graves de la somnolence au volant, telles que le ralentissement du temps de réaction, la baisse de l'attention et la diminution des capacités d'esprit. Dans ce contexte, la technologie de reconnaissance faciale présente un potentiel significatif. Elle pourrait évaluer les caractéristiques du visage pour détecter efficacement la fatigue, donc donner une solution innovante pour lutter contre la somnolence au volant, notamment dans les cas où la distraction est liée à l'utilisation d'un téléphone portable.

La détection de la somnolence repose sur l'observation du visage, l'identification de la position des yeux et l'analyse de la fréquence des clignements oculaires. Partant d'une séquence d'images d'un conducteur, un modèle paramétré de type réseau de neurones peut être entraîné à classifier l'activité d'un conducteur en plusieurs catégories de distraction.

Il existe plusieurs méthodes de détection de la fatigue d'un conducteur [2] :

- Méthodes basées sur l'état de conduite du véhicule [50] [51] : Certains auteurs dont Ramesh et al. suggère l'utilisation d'un capteur pour détecter en temps réel le mouvement du volant afin de déterminer le degré de fatigue d'un conducteur [52].
- Méthodes basées sur la base d'indicateurs physiologiques [53] [56] : Rohit et al. ont analysé les caractéristiques de l'électroencéphalogramme (EEG) à l'aide d'une analyse

discriminante linéaire et d'une machine à vecteurs de support pour détecter la fatigue d'un conducteur en temps réel [65].

- Méthodes sur la base de la fusion d'informations : Wang Fei et al. ont utilisé la fusion d'informations pour intégrer les indications physiologiques aux conditions de conduite afin d'identifier la fatigue d'un conducteur en rassemblant le signal EEG du sujet et les données associées à la manipulation du volant [2].
- Méthode sur la base de la vision par ordinateur [66] [67]: Grace a mesuré la taille et la position de la pupille à l'aide d'une lumière infrarouge de différentes longueurs d'onde [68]. Yan et al. ont utilisé la vision artificielle pour extraire la forme géométrique de la bouche [69].

Compte tenu des risques importants liés à la somnolence et à la distraction, la détection de ces phénomènes chez les conducteurs est cruciale pour la sécurité routière. Notre approche se concentre sur l'utilisation d'algorithmes d'apprentissage profond plus précisément MobileNetV2 entraîné avec les mesures extraites de la fonction `eye_aspect_ratio(eye)` pour analyser les données de reconnaissance faciale et détecter les signes de somnolence et de distraction d'un conducteur.

L'une des branches de l'intelligence artificielle la plus importante est la vision par ordinateur. Cette dernière est le domaine de l'informatique qui se concentre sur la reproduction des parties de la complexité du système de vision humain et qui permet aux ordinateurs de reconnaître et de traiter les objets dans les vidéos et les images de la même façon que l'humain le fait. Jusqu'à maintenant, la vision par ordinateur ne fonctionnait que de façon réduite. Grâce aux recherches sur l'intelligence artificielle et aux innovations en matière d'apprentissage profond, le domaine a connu un grand bond vers l'avant dernièrement et a pu dépasser de loin l'humain dans beaucoup de tâches dans le domaine de la détection et de la reconnaissance des signes de somnolence et de distraction chez un conducteur [70].

En exploitant les avancées de l'apprentissage profond, notre méthode vise à identifier les indicateurs de somnolence et de distraction, tels que les yeux fermés ou les bâillements, à partir des images du visage des conducteurs et de l'utilisation du téléphone portable. Notre modèle de réseaux de neurones convolutifs extrait les caractéristiques discriminantes et réalise la détection en temps réel. Cette approche permet d'alerter les conducteurs en cas de signes de somnolence et de distraction, contribuant ainsi à réduire les risques d'accidents liés à la fatigue et à la distraction au volant.

Les autres parties du mémoire sont organisées de la manière suivante :

2^{er} chapitre

La première section se concentre sur l'impact de la somnolence et de la distraction des conducteurs sur la sécurité routière. Elle examine les facteurs contribuant à la somnolence et à la distraction au volant dans le but de sensibiliser aux problèmes de ces comportements chez les conducteurs.

La deuxième section est de présenter les fondements de la technique de vision par ordinateur utilisée pour la détection de la somnolence et la distraction des conducteurs. Elle étudie les différentes étapes et les principaux concepts, tels que la détection des visages et l'extraction des caractéristiques faciales, spécifiquement dans le contexte de la reconnaissance faciale.

3^{ieme} chapitre

Ce chapitre se concentre sur les concepts fondamentaux de l'apprentissage automatique et de l'apprentissage profond, en mettant en avant les principes des réseaux de neurones artificiels. Les concepts de rétropropagation, l'apprentissage supervisé et non supervisé, ainsi que les architectures couramment utilisées, telles que les réseaux de neurones convolutifs (CNN), les réseaux de neurones récurrents (RNN) et les réseaux neurones artificiels (ANN) sont illustrés par quelques exemples. On y trouve également des exemples d'architectures populaires de CNN.

4^{ieme} chapitre

Ce chapitre porte sur la conception du système que nous proposons pour la détection de la somnolence et la distraction. Il présente l'architecture générale du système ainsi que ses différentes composantes fonctionnelles, en mettant en évidence les interactions entre les modules de vision par ordinateur et d'apprentissage profond. Les choix méthodologiques et techniques retenus sont présentés et justifiés, notamment en ce qui concerne la détection du visage, l'extraction des caractéristiques faciales, la sélection du modèle de classification et l'intégration des différents modules au sein d'un système cohérent et temps réel.

5^{ieme} chapitre

Dans ce chapitre, nous examinerons l'implémentation de l'architecture en décrivant les différentes logicielles et matérielles ainsi que les outils utilisés dans ce projet, et nous expliquerons comment mettre en œuvre le système d'apprentissage en profondeur. Nous

présenterons également l'évaluation empirique du système à travers plusieurs scénarios expérimentaux et les résultats obtenus afin d'apprécier ses performances et sa robustesse.

6^{ème} chapitre

Ce chapitre correspondant à la conclusion générale du mémoire, revient en résumé sur le travail effectué, des perspectives futures et l'originalité de notre approche. Nous mettrons également en évidence les avantages et les faiblesses du système développé.

Chapitre 2

Techniques de détection de la somnolence et la distraction chez les conducteurs

2.1 Introduction

La distraction et la somnolence au volant constituent deux facteurs majeurs d'accidents mortels. Selon Actua-Formation en 2022, en France, la distraction au volant représentait près de 25 % des accidents mortels, tandis que la somnolence provoquait $\frac{1}{3}$ des accidents mortels sur l'autoroute. Et cela, sans compter le nombre de blessés graves impliqués dans un accident de la route [3].

La somnolence au volant est un état de fatigue avancé (sans préavis contrairement à la fatigue que l'on peut anticiper) qui peut affecter la vigilance et la capacité à conduire en toute sécurité. Elle survient généralement sur des trajets moyens à long, de plus de 2 heures. Elle touche les conducteurs qui roulent sur des routes souvent en ligne droite, sur des trajets monotones : autoroutes, nationales, départementales [3].

Les distractions au volant peuvent survenir sur tout type de trajet, qu'ils soient courts ou longs, connus ou habituels. Elles sont souvent provoquées par le comportement du conducteur : téléphone, musique trop forte, manger et boire au volant, etc.

À la différence des causes provoquant la distraction au volant, la somnolence, elle, est complètement subie par le conducteur. Mais heureusement, il existe des leviers et des solutions à mettre en place pour lutter contre l'endormissement au volant [3].

Dans ce chapitre, nous aborderons de problème de la somnolence et de la distraction lors de la conduite, ainsi que les techniques de détection de la somnolence et de la distraction chez les conducteurs. L'objectif de ce travail est de sensibiliser les conducteurs aux dangers de la somnolence et la distraction au volant et de les aider à prendre des mesures pour éviter les accidents liés à cet état.

2.2 Relation entre la somnolence, distraction et la conduite automobile

Les signes de la somnolence au volant incluent des bâillements répétés, des paupières lourdes, fermetures prolongées de yeux, des difficultés à se concentrer et des périodes d'inattention.

Dans les cas les plus graves, la somnolence peut entraîner une perte de conscience momentanée ou des micro-sommeils, c'est-à-dire de courtes périodes d'endormissement involontaire.

La fatigue et la somnolence peuvent entraîner des temps de réaction plus lents, une diminution de la vigilance, une baisse de la coordination œil-main, ainsi qu'une diminution de la capacité à maintenir une attention soutenue [3].

La distraction au volant désigne une situation où le conducteur n'est plus attentif à sa conduite parce qu'il se concentre sur autre chose. Les distractions au volant peuvent survenir sur tout type de trajets, qu'ils soient courts ou longs, connus ou habituels. Elles sont souvent provoquées par le comportement du conducteur par exemple écrire des messages textes, parler au téléphone ou aux passagers.

Les passagers peuvent distraire le conducteur en engageant une conversation, en étant bruyants ou turbulents, en jouant de la musique trop fort, en utilisant leur téléphone portable ou en se chamaillant entre eux [3].

Ces effets peuvent rendre le conducteur moins capable de réagir rapidement aux situations imprévues sur la route, telles que des changements de voie, des freinages brusques ou des obstacles sur la route. Cela peut augmenter considérablement le risque d'accident.

2.3 Système de détection de la somnolence chez le conducteur

Pour détecter la somnolence au volant il existe une technologie innovante conçue justement pour lutter contre ce genre de problématique. Afin de renforcer la sécurité des conducteurs et de réduire le nombre d'accidents de la route, les constructeurs automobiles ont intégré des alarmes et des alertes destinées à prévenir les signes de fatigue. Une conduite sécurisée exige une attention totale de la part du conducteur, lui permettant ainsi d'anticiper les dangers et d'éviter les accidents [71].

L'un des systèmes d'aide à la conduite étudiés dans ce projet permet la détection de la somnolence et la distraction. Ce dispositif utilise la caméra d'un ordinateur pour capturer des images en temps réel et alerter le conducteur en cas de signes de somnolence ou de distraction ou à partir d'une séquence de vidéo.

2.3.1 Introduction aux expressions faciales

L'expression faciale joue un rôle essentiel dans la communication et le comportement non verbal. Une expression faciale peut alors exprimer certains états d'une personne, tels que la fatigue, la somnolence. Celle-ci résulte de l'activation, volontaire ou non, de l'un ou de plusieurs des 44 muscles faciaux, permettant ainsi la production d'environ 250 000 expressions différentes [74]. Déjà étudiée au XVIIIe siècle par Darwin et Duchenne de Boulogne, l'expression faciale a été au cœur des recherches sur les émotions, notamment à travers les travaux de Sylvan Tomkins dans les années 1960 [73]. Ses disciples, Paul Ekman et Carroll Izard, ont soutenu l'idée selon laquelle un nombre limité d'émotions de base sont associées à des expressions faciales automatiques, universelles et innées [72].

Dans la seconde moitié du dix-huitième siècle, le neurologue Duchenne de Boulogne a mené une série d'expériences sur l'expression faciale. Il a utilisé la photographie ainsi que la stimulation électrique des muscles du visage pour accentuer les mouvements liés à l'expression des émotions. Il a notamment observé que le sourire reflétant un véritable bonheur se distingue du sourire volontaire par la contraction spécifique du muscle orbiculaire autour des yeux [4]. La figure 2.1 montre les muscles faciaux et leur contrôle nerveux.

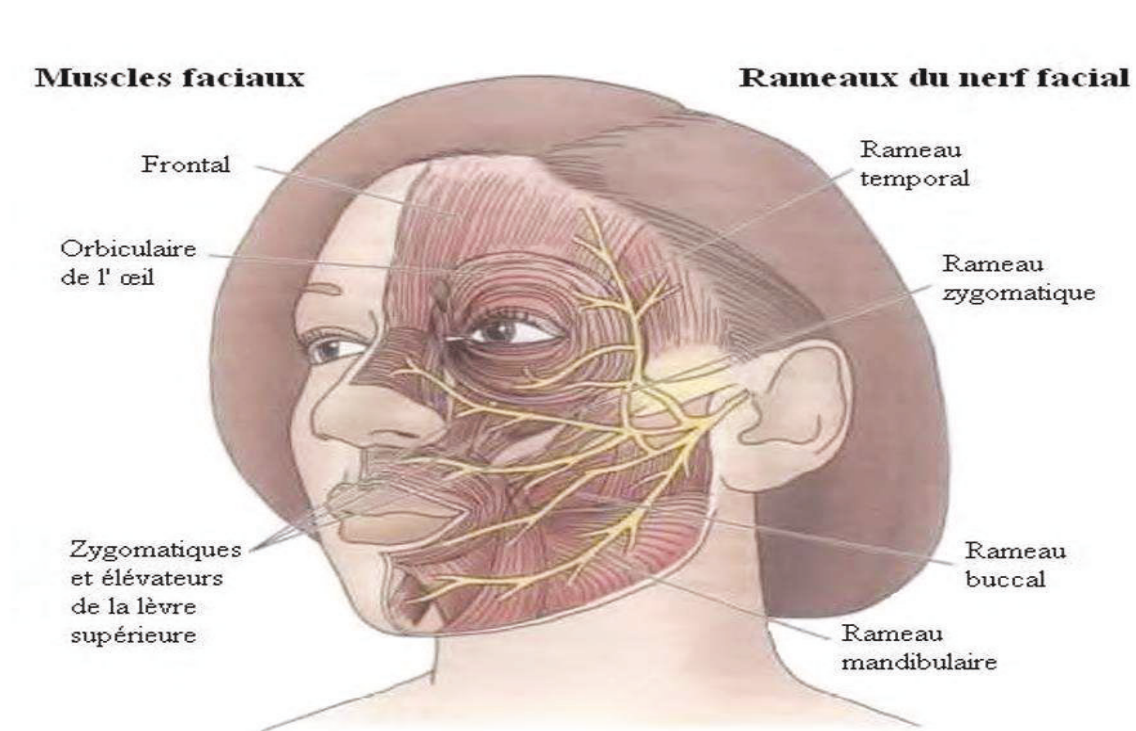


Figure 2.1 : Muscles faciaux et leur contrôle nerveux [4].

2.3.2 Définition de visage

Le visage peut être décrit comme une structure tridimensionnelle comprenant deux configurations distinctes. La configuration "externe" correspond au contour du visage, façonné par les reliefs osseux et encadré par les cheveux. À l'intérieur de cette structure s'intègre la configuration "interne", définie par l'ensemble des traits faciaux [75]. En outre, certains éléments spécifiques, tels que la chevelure, les lunettes ou encore la couleur de la peau, contribuent à l'unicité de chaque visage.

2.3.3 Les étapes préliminaires à la reconnaissance faciale

L'extraction des caractéristiques faciales constitue une étape essentielle d'un système automatique de détection de somnolence. Elle vise à identifier les points clés permettant de décrire les expressions du visage. Toutefois, trois paramètres doivent être pris en compte en tant qu'hypothèses indispensables pour l'application développée dans la présente recherche [4] :

- Le visage du sujet doit être positionné face à la caméra pour garantir une analyse précise.

- Lors de la phase d'initialisation, le sujet doit adopter une expression faciale neutre afin d'établir une référence pour l'analyse des variations ultérieures.

- Les conditions d'éclairage doivent rester stables afin d'assurer une détection fiable et précise des expressions faciales.

Toute variation de l'un de ces paramètres peut impacter la précision et la fiabilité des résultats de notre système.

La figure 2.2 montre l'architecture d'un système de reconnaissance des expressions faciales.

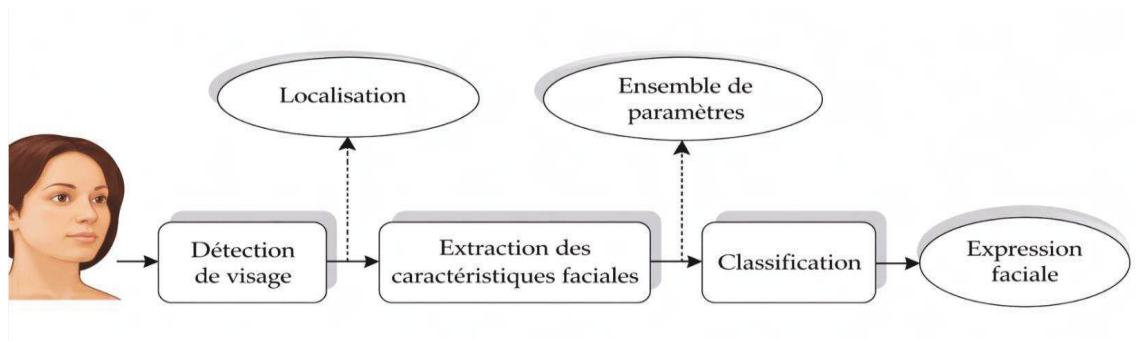


Figure 2.2 : Architecture d'un système de reconnaissance des expressions faciales.

2.3.3.1 Définition de la détection faciale

La détection de visage vise à localiser les visages dans une image ou une vidéo en se basant sur des traits distinctifs tels que les yeux, le nez, la bouche ou la couleur de la peau. Bien que cette tâche soit intuitive pour l'être humain, un ordinateur la perçoit sous forme d'images bidimensionnelles, en niveaux de gris ou en couleurs (RGB).

Pour surmonter cette difficulté, plusieurs techniques ont été mises au point, permettant à la machine d'identifier plus efficacement les traits faciaux essentiels, comme le détecteur de points de repères à 68 points (facial Landmark), utilisé par certains auteurs dont Amato et al. [59].

La figure 2.3 montre la position des 68 points (facial Landmark) de repères qui sont localisés dans une image de visage pour faciliter ensuite des processus comme l'identification d'une personne ou la détection des expressions faciales.

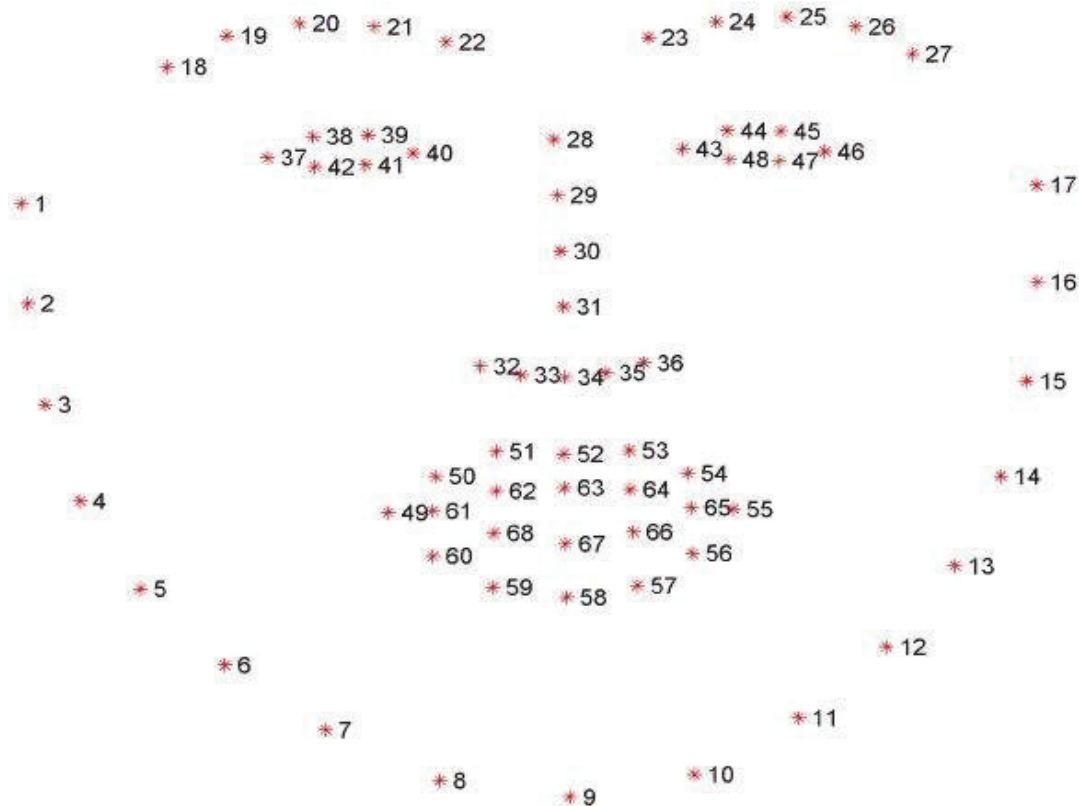


Figure 2.3 : Représentation en 68 points de repères faciaux (facial Landmark) utilisés pour la détection et l'analyse des expressions faciales [59].

2.3.3.2 Détection de visage

À cette étape, l'objectif est d'extraire le visage à partir d'une image. Cette détection peut être réalisée grâce à diverses approches, telles que l'analyse de la couleur de la peau, la reconnaissance de la forme de la tête ou l'identification des caractéristiques faciales. Cependant, la présence de plusieurs visages dans l'image ou d'un arrière-plan irrégulier peut compliquer la segmentation et affecter sa précision. Par conséquent, le succès de cette phase dépend en grande partie de la qualité des images obtenues [4].

La figure 2.4 illustre l'étape de détection et localisation du visage dans une image. À gauche, identification des points caractéristiques faciaux (landmarks). À droite, la représentation du maillage facial obtenu à partir de ces points.

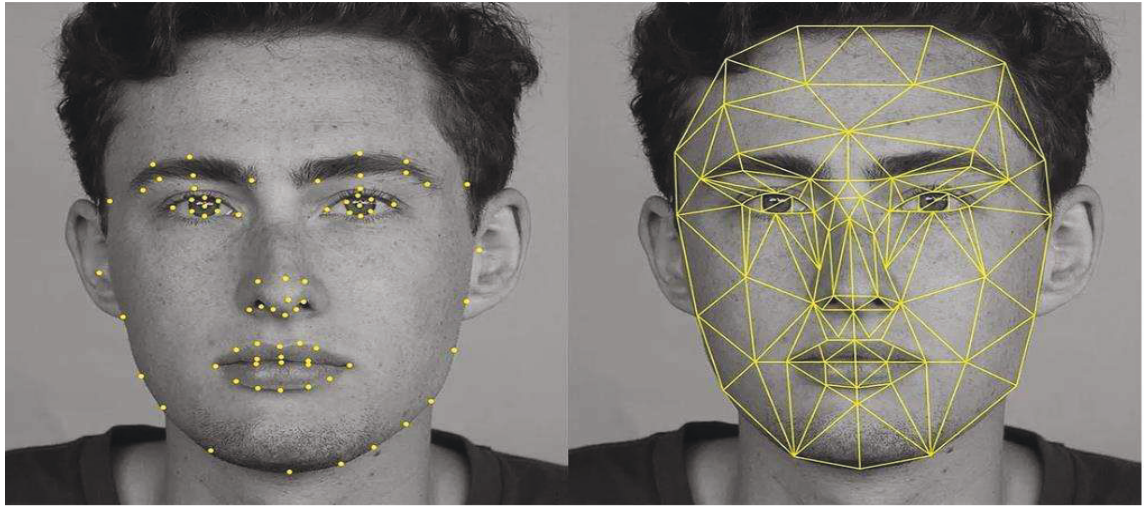


Figure 2.4 : Illustration du processus de détection du visage dans une image numérique [76].

Différentes approches sont suggérées dans la littérature [77] :

- Méthodes basées sur les connaissances : Ces méthodes reposent sur des règles visant à modéliser les connaissances relatives aux caractéristiques d'un visage. Par exemple, dans une image, un visage est généralement identifié par la présence de deux yeux symétriques, d'un nez et d'une bouche. Ces règles établissent les relations entre les différentes composantes faciales afin de faciliter leur détection et leur reconnaissance. On peut citer comme exemple la méthode de G. Yang et T.S. Huang basée sur une stratégie de multi-résolution [78].
- Méthodes basées sur les caractéristiques invariantes : Ces méthodes s'appuient sur des caractéristiques structurelles telles que les traits du visage, la texture et la couleur de la peau, qui demeurent perceptibles malgré les variations de pose, d'angle de vue ou de conditions

d'éclairage. Citons comme exemple les travaux de S. McKenna, S. Gong, et Y. Raja qui se basent sur la détection de la couleur de la peau [79].

- Méthodes basées sur l'apprentissage machine : Ces méthodes permettent de créer des modèles comme des réseaux de neurones qui sont entraînés à l'aide d'un ensemble d'images d'apprentissage, permettant de capturer la diversité des apparences faciales. Ces méthodes exploitent des techniques telles que l'apprentissage automatique et l'analyse statistique afin d'identifier les caractéristiques distinctives des images de visages et de non-visages. Un exemple de méthode appartenant à cette catégorie est celle développée par H. Rowley, S. Baluja et T. Kanade basée sur des réseaux de neurones [80].

2.3.3.3 Extraction des caractéristiques du visage

Il existe deux approches distinctes pour le processus d'extraction des caractéristiques du visage. La première consiste à extraire l'ensemble du visage, elle est souvent implémentée avec une approche globale de reconnaissance de visage. La seconde, appelée reconnaissance locale, se focalise sur l'extraction de points spécifiques dans les zones caractéristiques du visage, comme la bouche et le nez, afin d'identifier les différences distinctives [5]. Elles sont utilisées avec une méthode locale de reconnaissance et aussi pour l'estimation de la pose du visage [81].

2.3.4 Techniques de détection de la somnolence des conducteurs

Etant donné que certaines caractéristiques d'un visage peuvent être utilisées pour détecter des situations de somnolence. Il devient dans ce contexte essentiel de développer des approches efficaces pour détecter la somnolence au volant afin d'améliorer la sécurité routière [98].

La Figure 2.5 illustre les différentes approches de détection de somnolence utilisées.



Figure 2.5 : Les différentes méthodes de détection de somnolence utilisées.

Explorons trois principales méthodes pour y parvenir.

- Techniques de traitement d'images : approches qui analysent les expressions faciales et les mouvements des yeux d'un conducteur à l'aide de caméras et d'algorithmes de vision par ordinateur [82].

La figure 2.6 illustre les techniques de détection de somnolence basées sur le traitement d'images. Le système analyse les expressions faciales, l'ouverture des yeux, la direction du regard ainsi que la position et l'orientation de la tête du conducteur afin d'évaluer son niveau de vigilance.



Figure 2.6 : Illustration de la détection et de l'analyse des repères faciaux par traitement d'images [82].

- Techniques basées sur les réseaux de neurones artificiels : ces approches exploitent l'intelligence artificielle pour reconnaître des schémas caractéristiques de la somnolence en s'appuyant sur des données visuelles (images du visage, expressions faciales, mouvement des yeux) et comportementales [83].

Grâce à leur capacité d'apprentissage, les réseaux de neurones artificiels permettent d'extraire des caractéristiques pertinentes à partir de grandes quantités de données et d'améliorer progressivement la précision de la détection, même dans des conditions complexes ou variables.

La figure 2.7 illustre principe de fonctionnement des réseaux de neurones artificiels pour l'analyse visuelle et comportementale. Les données issues des images du visage et des comportements observés sont traitées par des réseaux neuronaux afin d'identifier des schémas associés à la somnolence.



Figure 2.7 : Représentation conceptuelle d'un réseau de neurones artificiels appliqué à l'analyse faciales [83].

- Techniques d'EEG (électroencéphalographie) : approches qui mesurent l'activité cérébrale d'un conducteur à l'aide de capteurs afin de détecter les signaux (EEG) indiquant un état de fatigue ou de somnolence.

En complément certaines approches utilisent également l'électro-oculographie EOG afin de mesurer les mouvements oculaires horizontaux et verticaux. L'analyse conjointe des signaux EEG (mesure les signaux du cerveau) et EOG (mesure les contractions musculaires des yeux) permet une détection plus précise des micro-sommeils et changements cognitifs liés à la fatigue [84]. Les signaux EOG verticaux et horizontaux permettent de mesurer les contractions musculaires associées aux mouvements des yeux. L'EOG horizontal capte principalement les déplacements latéraux du regard, tandis que l'EOG vertical mesure les mouvements de montée et de descente des yeux, notamment lors des clignements ou des fermetures prolongées des paupières.

La figure 2.8 montre les capteurs d'EEG et EOG utilisés pour la détection de la somnolence d'un conducteur.

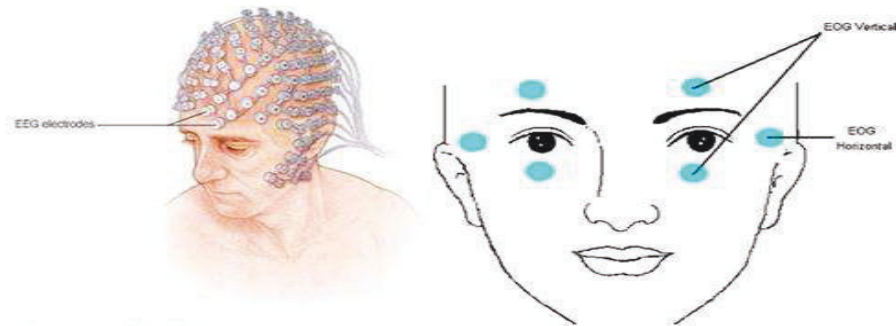


Figure 2.8 : Illustration des capteurs utilisés dans les méthodes axées sur la performance d'un conducteur [84].

Ces méthodes, employées individuellement ou combinées, contribuent à la mise en place de systèmes avancés pour prévenir les accidents liés à la fatigue au volant.

2.3.4.1 Techniques basées sur le traitement d'images

Les systèmes de surveillance du visage du conducteur permettent d'identifier des signes révélateurs tels que la fatigue, la somnolence et la distraction. Ces signes, observables visuellement, constituent des indicateurs essentiels pour évaluer l'état de vigilance du conducteur. Ces indicateurs peuvent être classés en trois catégories [97].

La figure 2.9 illustre un exemple de détection de comportements associés à la somnolence à l'aide de techniques basées sur le traitement d'images.



Figure 2.9 : Surveillance du visage du conducteur par techniques de traitement d'images [97].

2.3.4.1.1 Techniques de correspondance des modèles

Cette technique a été développée à l'aide de la technologie Augment c'est-à-dire une technologie d'amélioration, permettant au système d'intégrer plusieurs modules fonctionnant simultanément. L'œil étant la zone la plus révélatrice des signes de fatigue, son état peut être analysé pour détecter l'endormissement d'un conducteur. Ainsi, si les yeux restent fermés pendant une durée prolongée, une alarme est automatiquement déclenchée. De plus, le système peut être entraîné à reconnaître et différencier les modèles de yeux ouverts et fermés du conducteur afin d'améliorer la précision de la détection [6].

La Figure 2.10 illustre un exemple de mise en œuvre de la technique de correspondance des modèles pour l'analyse du comportement d'un conducteur.



Figure 2.10 : Exemple d'analyse du visage par correspondance des modèles [93].

G1 : caméra gauche n°1 – généralement orientée vers le visage du conducteur.

G2 : caméra gauche n°2 – autre vue du visage ou d'un angle complémentaire.

D3 : caméra droite n°3 – souvent orientée vers le poste de conduite.

D4 : caméra droite n°4 – vue secondaire (tableau de bord, environnement intérieur, posture).

2.3.4.1.2 Technique basée sur le clignement des yeux

Cette technique repose sur l'analyse des signes de somnolence et des symptômes comportementaux du micro-sommeil, visibles sur le visage, notamment autour des yeux. Une fermeture prolongée des paupières, des modifications du rythme de clignement (réduction des clignements spontanés, clignements plus longs et plus marqués) sont mesurées afin de détecter

l'état de fatigue d'un conducteur. Lorsqu'il commence à somnoler, la fréquence et l'apparence de ses clignements diffèrent des conditions normales, rendant ainsi la détection de la somnolence plus efficace. En conditions normales, un clignement dure 0,2 à 0,4 seconde. En état de somnolence, il peut durer plus d'une seconde, voire plusieurs frames consécutifs sur la séquence vidéo [7] [8].

La Figure 2.11 illustre un exemple d'application de la technique basée sur le clignement des yeux. La fermeture prolongée des paupières et l'asymétrie de l'ouverture oculaire constituent des indicateurs visuels de la somnolence du conducteur.



Figure 2.11 : Détection de la somnolence par l'analyse du clignement des yeux [85].

Ce système repose sur un modèle de couleur de peau dans l'espace colorimétrique HSI pour détecter les visages. Il utilise l'analyse des contours pour localiser les yeux, ainsi qu'un suivi dynamique pour en assurer la reconnaissance en temps réel. L'état des yeux permet alors d'évaluer celui du conducteur. Si les yeux restent fermés durant cinq images consécutives, le conducteur est considéré comme étant en état de fatigue. Cela repose sur le fait qu'un clignement normal dure de 100 à 400 ms, tandis qu'un clignement prolongé (>1 s) indique souvent une somnolence imminente [15].

Les systèmes de détection de la somnolence d'un conducteur reposent généralement sur l'identification et la localisation du visage et des yeux. Ils évaluent l'état de vigilance en calculant le rapport d'aspect des yeux (Eye Aspect Ratio - EAR), qui mesure l'espacement des paupières afin de déterminer si les yeux sont ouverts ou fermés [16].

L'une des mesures les plus couramment utilisées pour détecter la somnolence est le PERCLOS (Percentage of Eye Closure). Développé pour analyser les variations visibles du mouvement des paupières, ce paramètre évalue le pourcentage de temps pendant lequel les yeux restent partiellement ou totalement fermés sur une période donnée. Contrairement aux clignements rapides, le PERCLOS met en évidence les fermetures lentes des paupières, indicatrices d'un état de fatigue croissant [17] [18].

Finalement on détecte l'état de vigilance d'un conducteur avec le calcul de moyenne de PERCLOS des deux yeux.

$$\text{PERCLOS} = \frac{T_{\text{yeux fermés}}}{T_{\text{total d'observations}}} \times 100$$

$T_{\text{yeux fermés}}$: durée totale (en secondes ou nombre de frames) pendant laquelle les yeux sont fermés.

$T_{\text{total d'observations}}$: durée totale du segment vidéo observé (souvent 1 minute).

2.3.4.1.3 Technique basée sur le bâillement

Cette technique permet une détection non invasive et en temps réel de la somnolence d'un conducteur à travers l'analyse du bâillement. Le système fonctionne en plusieurs étapes, incluant la détection et le suivi du visage. Le bâillement est supposé être caractérisé par une large ouverture verticale de la bouche, qui peut être identifiée en analysant le taux d'ouverture et les variations de la zone du contour buccal [86].

La Figure 2.12 illustre la technique de détection de somnolence basée sur l'analyse du bâillement. Une ouverture verticale importante de la bouche, observée sur une durée significative, constitue un indicateur visuel de l'état de fatigue du conducteur.



Figure 2.12 : Détection du bâillement comme indicateur de somnolence d'un conducteur [87].

Le détecteur de contours proposé repose sur un modèle original basé sur la morphologie de la bouche lors d'un bâillement. Dans ce cas, la bouche apparaît sous forme d'une large zone sombre de forme pseudo-circulaire. Cette zone est encadrée par des parties de peau correspondant aux lèvres supérieure et inférieure, ainsi que, potentiellement, par des dents visibles [9].

Ainsi, l'importante différence d'intensité entre la zone sombre du bâillement et les régions environnantes, telles que les lèvres ou les dents, est exploitée pour concevoir un détecteur précis du contour de la grande ouverture de la bouche, de manière similaire au détecteur de contour de l'iris [9].

Certains auteurs comme Neha Chaudhari [96] analysent uniquement les pixels x de l'image de la bouche susceptibles de correspondre à sa grande ouverture. Pour chaque pixel x , nous identifions un voisinage composé de n pixels situés au-dessus de la bouche et de n pixels situés en dessous de la bouche, puis nous calculons la différence d'intensité entre le pixel x et ces pixels environnants. La valeur de n est définie comme un cinquième du nombre total de

colonnes dans l'image de la bouche, ce qui permet d'adapter l'analyse à la taille de la région observée.

La Figure 2.13 illustre une séquence d'image correspondant au processus de bâillement.

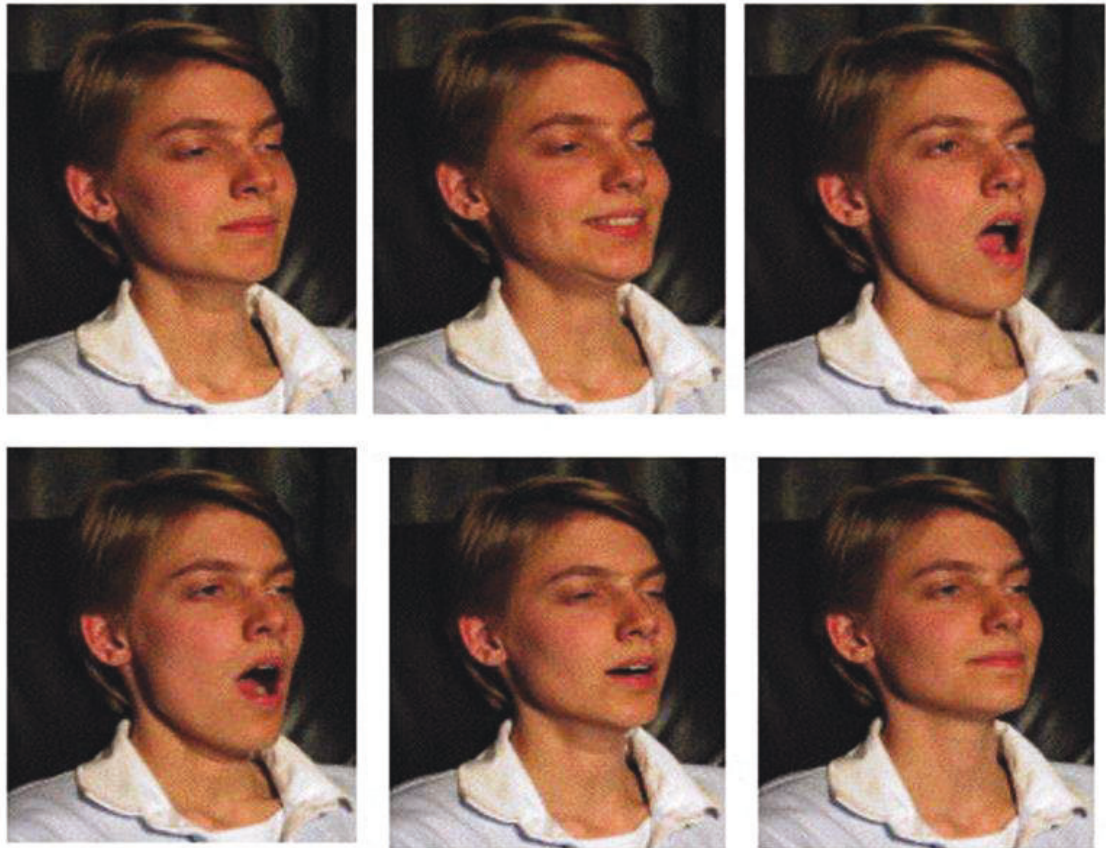


Figure 2.13 : *Illustration des mouvements faciaux associés à un bâillement* [96].

Détection du contour supérieur :

Si au moins $(n-1)$ pixels donnent une différence supérieure au seuil THS_{UP} et si au moins $(n-1)$ pixels produisent une différence inférieure à un seuil $THIN_{F}$, alors on conclut que le pixel x appartient au contour supérieur de l'ouverture de la bouche et fixons alors le pixel x à 1.

Détection du contour inférieur :

Si nous avons au moins $(n-1)$ pixels qui donnent une différence inférieure à $THIN_{F}$ et d'au moins $(n-1)$ pixels qui donnent une différence supérieure à THS_{UP} , nous concluons que

le pixel x appartient au contour inférieur de l'ouverture de la bouche et le fixons alors le pixel x à 1.

La différence d'intensité est calculée en comparant un pixel candidat avec celle de pixels situés au-dessus et au-dessous de celui-ci, puis en utilisant des seuils (THINF, THSUP) pour déterminer s'il appartient au contour supérieur ou inférieur de la bouche lors d'un bâillement.

La figure 2.14 illustre les contours supérieur et inférieur de la bouche pendant le bâillement.

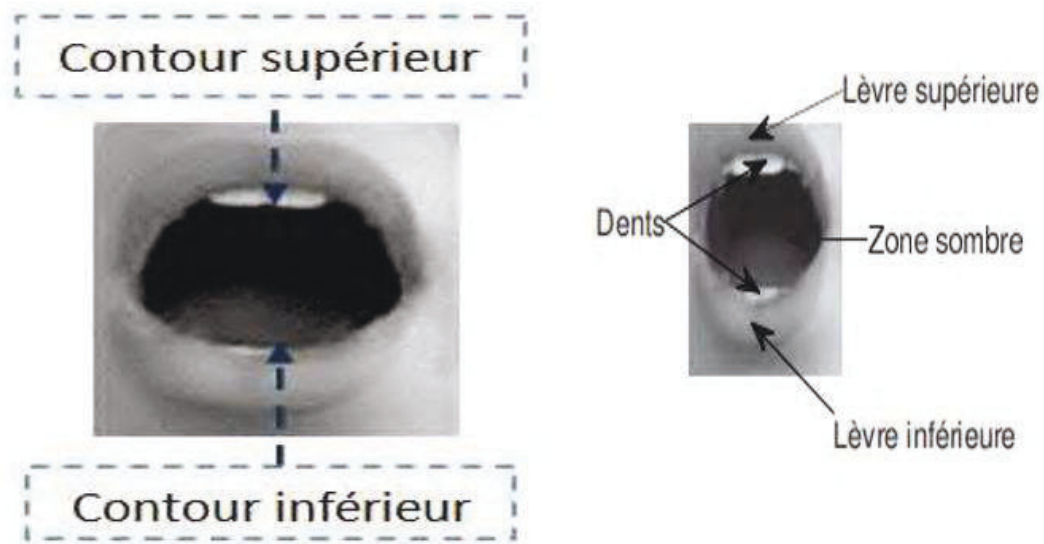


Figure 2.14 : Illustration des contours supérieurs et inférieurs et la structure du bâillement [64].

2.3.4.2 Technique basée sur le signal EEG

Cette technique permet de mesurer l'activité électrique du cerveau à l'aide d'électrodes placées sur le cuir chevelu. Ces électrodes sont intégrées dans un casque (voir figure 2.15) que le conducteur doit porter pendant la conduite. La méthode repose sur l'analyse du spectre de puissance et l'utilisation de l'algorithme FastICA pour extraire les signaux cérébraux pertinents et détecter les signes de somnolence. On peut aussi capter d'autres signaux tels que EOG (mouvement des yeux) et ECG (rythme cardiaque) [10] [11].

FastICA permet d'extraire les signaux significatifs indépendants de chaque électrode.

La Figure 2.15 illustre un casque à électrodes EEG.



Figure 2.15 : Casque à électrodes EEG [12].

L'électroencéphalogramme (EEG) est l'enregistrement de différences de potentiels électrique effectués sur la surface de la tête. Il mesure l'activité électrique du cerveau par des électrodes placées sur le cuir chevelu souvent représenté sous la forme d'un tracé appelé électroencéphalogramme. L'EEG est un examen indolore et non invasif qui renseigne sur l'activité neurophysiologie du cerveau au cours du temps et en particulier du cortex cérébral soit dans un diagnostic en neurologie. Il a une très faible amplitude, de l'ordre du microvolt. Il se présente différemment selon l'âge, selon l'état de vigilance [13].

L'EEG est l'un des signaux les plus fiables pour analyser l'état d'un conducteur. Il permet de mesurer l'activité cérébrale à travers différentes bandes de fréquences, chacune étant associée à un état spécifique.

L'analyse de ces fréquences permet de détecter les signes de fatigue et de somnolence chez le conducteur, contribuant ainsi à améliorer la sécurité routière.

En effet, le cerveau humain est composé de milliards de neurones qui interagissent entre eux par le biais de signaux électriques. La communication simultanée de ces neurones génère une activité électrique importante, qui peut être détectée à l'aide d'un EEG. Comme ces

impulsions se produisent à intervalles réguliers, sous forme d'ondes, elles sont appelées « ondes cérébrales ».

La Figure 2.16 permet d'observer les 5 principales bandes de fréquences cérébrales.
Bandes de fréquences-ondes cérébrales [14].

Les premières bandes de fréquences les plus connues sont celles de l'activité alpha, qui augmentent lorsque les yeux sont fermés et peuvent être observées dans le lobe occipital associé au système visuel humain lors d'un état de veille détendu.

D'autres bandes de fréquences comprennent : delta, thêta, bêta, low gamma et high gamma. Cette dernière appartient aux rythmes rapides et est associée au traitement cognitif.

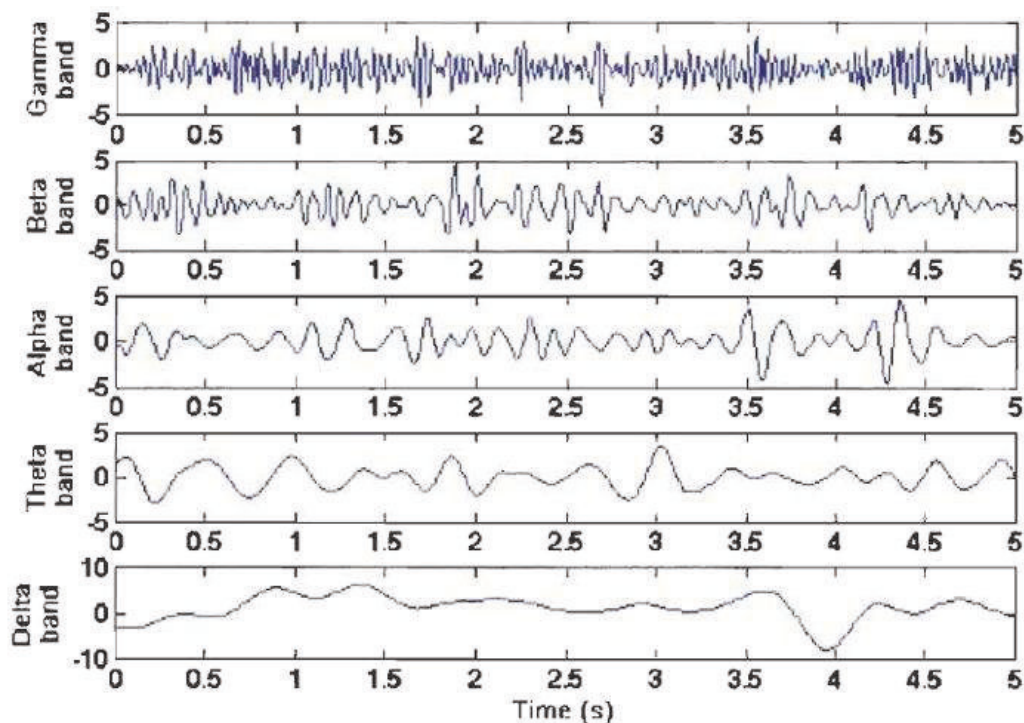


Figure 2.16 : Les cinq bandes de fréquences [14].

L'EEG (électroencéphalographie) est l'un des signaux les plus fiables pour analyser l'état d'un conducteur. Il permet de mesurer l'activité cérébrale à travers différentes bandes de fréquences, chacune étant associée à un état spécifique :

- Bande delta (0,5 - 4 Hz) : Correspond à l'activité liée au sommeil profond.

- Bande thêta (4 - 8 Hz) : Associée à la somnolence et à la transition entre l'éveil et le sommeil.
- Bande alpha (8 - 13 Hz) : Indique un état de relaxation, souvent observé dans un état de repos les yeux fermés.
- Bande bêta (13 - 25 Hz) : Représente un état de vigilance et de concentration accrue.
- Bande Gamma (31 Hz et plus) : Reflète le mécanisme de la conscience.

2.3.4.3 Techniques basées sur les réseaux de neurones artificiels

Cette technologie utilise des réseaux de neurones pour détecter la somnolence du conducteur. De nombreux chercheurs comme M.Vats et A.Garg [20] ont menés des recherches dans le domaine de l'optimisation de la détection de la somnolence d'un conducteur en utilisant des réseaux de neurones. Les personnes fatiguées manifestent des changements visibles dans leur comportement, notamment à travers des modifications des caractéristiques faciales telles que les yeux, la position de la tête et l'expression du visage. Pour utiliser ces signaux visuels, ces chercheurs ont créé un réseau de neurone artificiel pour détecter la somnolence [20].

La méthode proposée repose sur la conception d'un réseau de neurones artificiels capable d'analyser ces signaux visuels présentés comme entrées du réseaux, afin d'identifier les signes de somnolence et d'alerter le conducteur en conséquence (sortie du réseau) [20].

La figure 2.17 montre l'architecture d'un réseau de neurones artificiels.

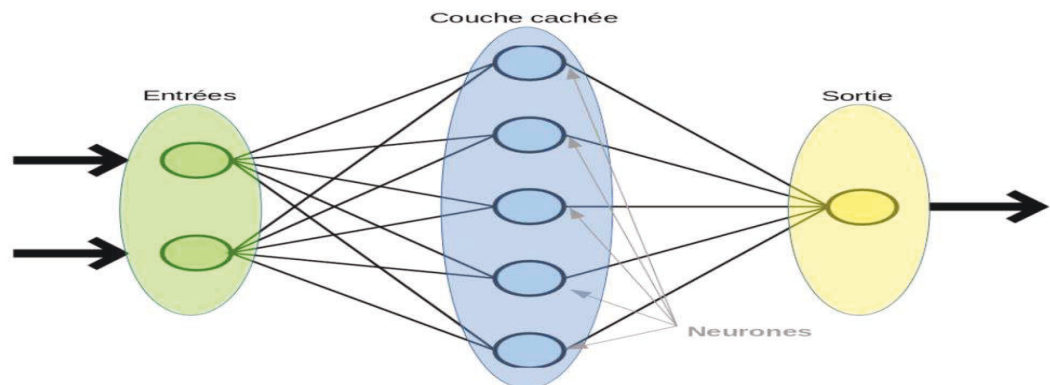


Figure 2.17 : Réseau de neurones artificiels [88].

Plus récemment, des approches basées sur l'apprentissage en profondeur (deep learning), telles que l'utilisation des réseaux de neurones convolutifs (CNN), ont émergé. Ces techniques représentent une avancée majeure, notamment pour les applications de vision par

ordinateur, telles que la classification d'images, la détection d'objets, la reconnaissance des émotions ou encore la segmentation de scènes.

2.4 Conclusion

La somnolence au volant représente un enjeu majeur de sécurité routière, pouvant entraîner de graves accidents. Les conducteurs qui somnolent manifestent divers signes, comme des bâillements fréquents et des difficultés de concentration. Il est donc crucial de mettre en place des solutions permettant de détecter la somnolence en temps réel afin de prévenir ces dangers.

L'évolution des technologies joue un rôle clé dans pour la détection de ce phénomène. Les méthodes d'analyse des expressions faciales, en particulier celles basées sur les réseaux de neurones convolutifs (CNN), se développent et offrent des solutions prometteuses pour surveiller en temps réel l'état de vigilance des conducteurs.

Cependant, bien que ces techniques progressent, des améliorations sont encore nécessaires pour renforcer leur précision et leur fiabilité. Par ailleurs, la sensibilisation des conducteurs aux risques liés à la somnolence est tout aussi essentielle. Des mesures préventives, telles que faire des pauses régulières et s'assurer d'un repos suffisant avant de conduire, doivent être encouragées.

En définitive, la sécurité routière est une responsabilité collective, et la prévention de la somnolence au volant constitue un élément fondamental pour réduire le nombre d'accidents et protéger la vie des usagers de la route.

Dans le but de détecter les épisodes de somnolence, il existe des approches de détection de somnolence basées sur les réseaux de neurones. Le chapitre qui suit permet de présenter les principes de l'apprentissage machine pouvant être appliqués à la détection de la somnolence.

Chapitre 3

Principes de l'Apprentissage Automatique et Profond : Réseaux de Neurones et Architectures Avancées

3.1 Introduction

La détection de la somnolence chez les conducteurs est un enjeu majeur en matière de sécurité routière. Les accidents liés à la fatigue ou à la somnolence au volant peuvent entraîner des conséquences dramatiques. Toutefois, identifier ces signes chez les conducteurs reste un défi, rendant la situation encore plus dangereuse.

C'est dans ce contexte que l'intelligence artificielle (IA) joue un rôle clé. Des chercheurs ont exploré l'utilisation de l'IA pour détecter la somnolence des conducteurs de manière plus efficace. Parmi les domaines de l'IA, l'apprentissage automatique et l'apprentissage profond ont récemment suscité un vif intérêt [\[89\]](#).

L'apprentissage automatique permet aux ordinateurs de développer des modèles prédictifs à partir de vastes ensembles de données, sans nécessiter de programmation explicite. Quant à l'apprentissage profond, il représente une approche avancée de l'apprentissage automatique qui a connu un essor considérable ces dernières années. Ces techniques d'IA ont démontré un potentiel significatif dans divers domaines, y compris la surveillance de l'état des conducteurs, contribuant ainsi à renforcer la sécurité sur les routes [\[89\]](#).

Ce chapitre présente un aperçu des méthodes d'apprentissage profond que nous utiliserons pour mener à bien ce présent travail de recherche. Nous mettrons principalement l'accent sur l'architecture des réseaux de neurones convolutifs (CNN), en commençant par une introduction à l'apprentissage automatique et aux différents types d'algorithmes qui y sont associés. Par la suite, nous explorerons les algorithmes d'apprentissage profond et leur fonctionnement. Enfin, nous présenterons un aperçu des architectures CNN les plus couramment utilisées à partir desquelles nous pourrions nous inspirer pour implémenter notre modèle de détection de la somnolence.

3.2 L'apprentissage automatique

L'apprentissage automatique (Machine Learning) est une approche de programmation informatique qui exploite des modèles statistiques et des probabilités pour permettre aux ordinateurs d'apprendre de manière autonome, sans nécessiter une programmation explicite. Son objectif principal est d'enseigner aux machines à analyser, agir et réagir de manière similaire aux humains, en affinant progressivement leur méthode d'apprentissage et en enrichissant leurs connaissances au fil du temps [89].

L'apprentissage automatique est particulièrement utile lorsque les données sont abondantes, mais que les connaissances sont limitées ou insuffisamment développées. Il permet non seulement aux machines d'apprendre, mais aussi d'aider les humains à mieux comprendre les phénomènes sous-jacents. Grâce aux algorithmes d'apprentissage, des modèles sont créés pour identifier l'importance relative de certaines informations et analyser leurs interactions, facilitant ainsi la résolution de problèmes spécifiques [22].

3.2.1 Les phases de l'apprentissage automatique

De manière générale, les algorithmes d'apprentissage automatique suivent plusieurs phases essentielles, qui sont les suivantes [90] :

1. Collecte et préparation des données : Cette étape consiste à rassembler les données essentielles pour entraîner le modèle d'apprentissage automatique et les préparer en vue de leur analyse.
2. Sélection de modèles d'apprentissage automatique : Il existe plusieurs types de modèles d'apprentissage automatique, chacun ayant ses propres avantages et limitations, en fonction de la nature des données et de la tâche à accomplir.
3. Entraînement du modèle : Le modèle est entraîné sur les données d'apprentissage afin d'identifier les motifs et les relations entre les différentes variables, lui permettant ainsi de générer des prédictions ou de prendre des décisions plus précises.
4. Évaluation du modèle : Une fois le modèle entraîné, il doit être évalué sur des données de test afin de mesurer sa précision et sa capacité à généraliser de nouvelles données. Cette étape permet de vérifier la performance du modèle et d'identifier d'éventuels ajustements nécessaires pour améliorer sa fiabilité.

5. Optimisation et ajustements : L'amélioration du modèle passe par l'ajustement de ses paramètres (c'est-à-dire entre autres le taux d'apprentissage, le nombre de neurones, etc.) et l'application de techniques telles que la validation croisée afin d'optimiser sa performance et sa capacité de généralisation.
6. Déploiement et mise en production : Le modèle est intégré dans un environnement opérationnel afin d'être utilisé en conditions réelles, garantissant ainsi son application pratique et son efficacité dans des situations concrètes.
7. Surveillance et mise à jour : Le suivi des performances du modèle est essentiel pour garantir son efficacité. Il inclut une mise à jour régulière en fonction de nouvelles données ou de l'évolution du problème, afin d'assurer une adaptation continue et une amélioration de sa précision.

3.2.2 Types d'apprentissage automatique

L'apprentissage automatique est un domaine assez vaste. La figure 3.1 montre les types d'apprentissage automatique.

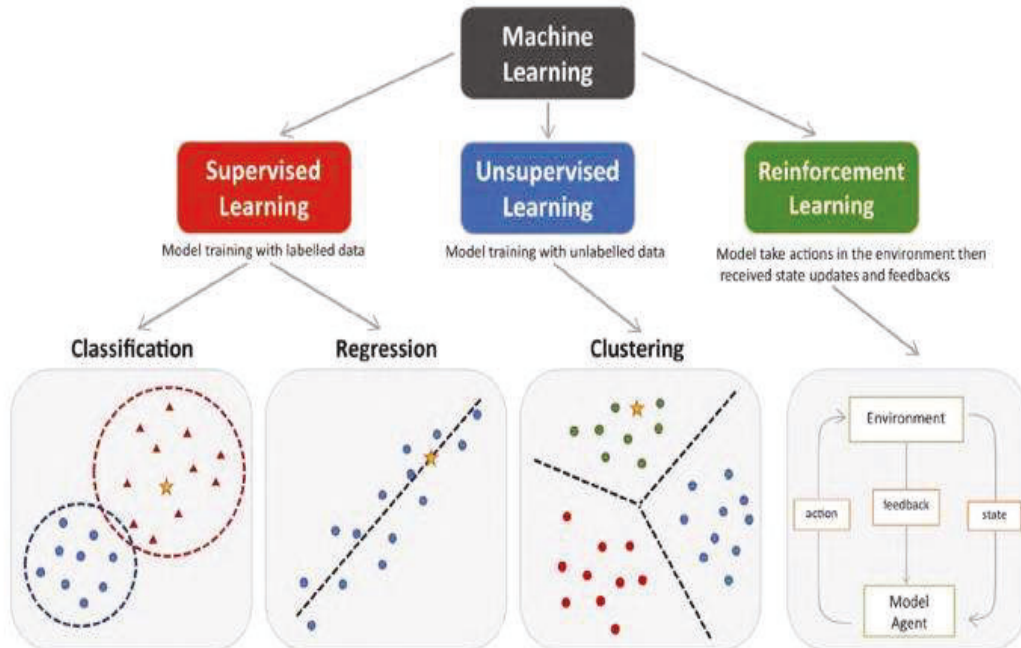


Figure 3.1 : Les principaux types d'apprentissage automatique [24].

3.2.2.1 Apprentissage supervisé (Supervised learning)

L'apprentissage supervisé est une méthode d'intelligence artificielle qui repose sur la conception d'un algorithme capable d'apprendre et à faire des prédictions grâce à un processus d'entraînement. Elle se divise généralement en deux catégories principales : la classification et la régression [24].

Le processus de création d'un modèle visant à organiser les données en groupes distincts, à simplifier leur structure et à effectuer des prédictions relève de la catégorisation. Cette approche repose sur la classification, une méthode permettant de résumer les ensembles de données tout en identifiant les relations clés et leur structure globale. Lorsqu'il est possible de définir des classes distinctes d'éléments, celles-ci peuvent être identifiées, leurs caractéristiques énumérées et les informations mieux structurées pour une récupération plus efficace. Cela facilite également l'intégration de nouveaux objets dans le système [25].

La régression est quant à elle une technique de modélisation prédictive qui vise à établir une relation significative entre une variable dépendante et une ou plusieurs variables indépendantes, également appelées variables explicatives. Il existe plusieurs types de régression, notamment la régression linéaire, logistique, polynomiale, Ridge, Lasso, ainsi que d'autres variantes adaptées à différents types de données et de problématiques [26].

La figure 3.2 permet de schématiser le processus de classification et de régression.

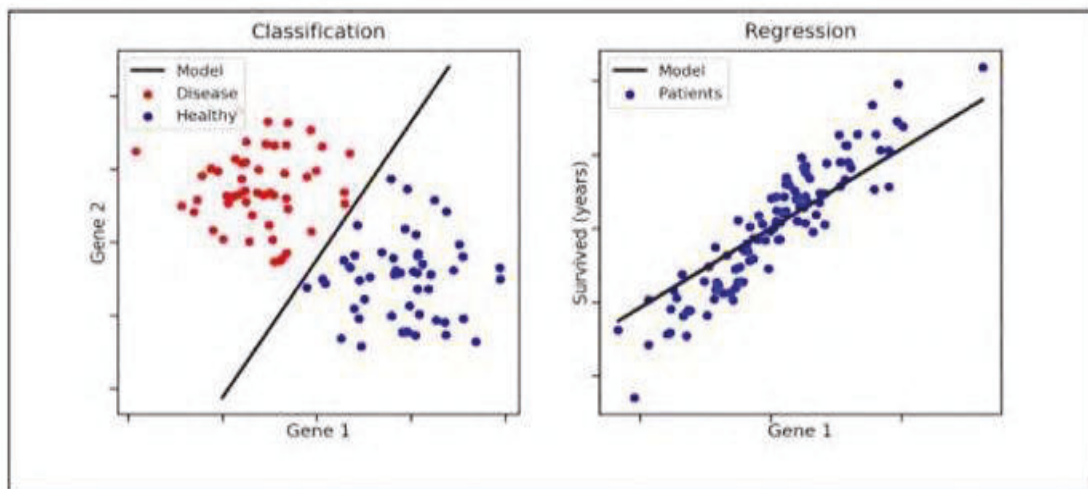


Figure 3.2 : Processus de classification et de régression [27].

3.2.2.2 Apprentissage non supervisé (Unsupervised learning)

L'apprentissage non supervisé est une approche utilisée lorsque les données ne sont pas à priori étiquetées. Cela signifie que seules les données d'entrée sont disponibles, sans connaissance préalable des sorties associées. L'objectif du système est alors d'identifier des caractéristiques communes au sein des données d'entraînement. Cette méthode repose principalement sur des algorithmes de regroupement (clustering), qui permettent de détecter des structures et des schémas sous-jacents dans les données [28].

L'analyse de cluster est une technique couramment employée en extraction de données et en apprentissage automatique pour regrouper des éléments similaires en clusters. Parmi les méthodes les plus utilisées, le K-means clustering vise à diviser un ensemble d'objets en K clusters, en minimisant la somme des distances au carrées entre chaque objet et la moyenne du cluster auquel il est attribué. Cette approche permet d'identifier des structures sous-jacentes dans les données et d'optimiser leur organisation en fonction de similarités [28].

La figure 3.3 illustre le processus de formation des clusters à partir d'un ensemble de données.

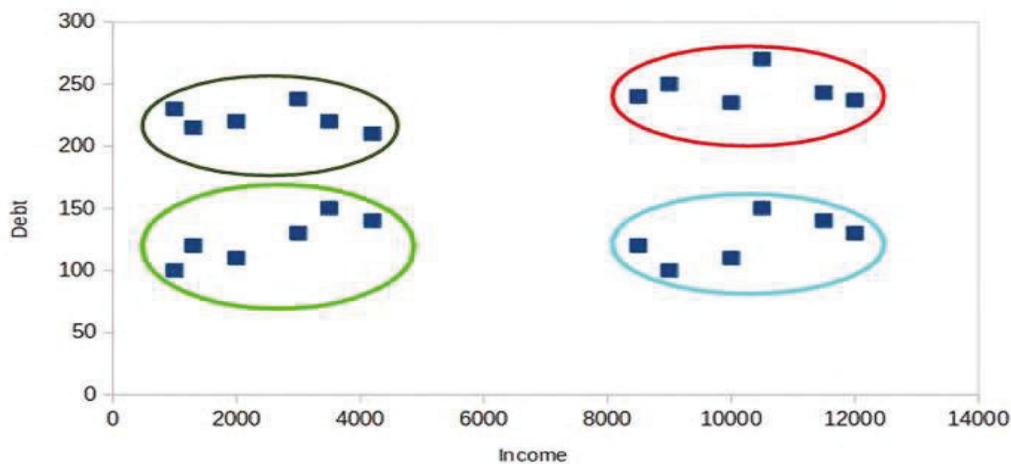


Figure 3.3 : Création de clusters à partir des données [28].

Le clustering hiérarchique et le clustering K-means sont deux approches courantes en apprentissage non supervisé pour regrouper des points de données en catégories distinctes. Le K-means segmente les données en un nombre prédéfini de clusters, en minimisant la variance au sein de chaque groupe. En revanche, le clustering hiérarchique construit une structure arborescente qui illustre les relations entre les groupes, permettant ainsi une analyse plus flexible des niveaux de similarité entre les éléments [28].

3.2.2.3 Apprentissage par renforcement (Reinforcement learning)

L'apprentissage par renforcement se concentre sur la manière dont un agent autonome interagit avec son environnement afin d'apprendre à adopter des comportements optimaux pour atteindre ses objectifs. Il repose sur un système de récompenses qui évalue les actions de l'agent en fonction de leurs conséquences, favorisant ainsi l'amélioration progressive de ses décisions. Cette approche trouve de nombreuses applications, notamment dans les jeux stratégiques, comme un ordinateur jouant aux échecs contre un humain, la reconnaissance de la parole ou encore la classification de nouvelles formations astronomiques [29].

3.3 L'apprentissage profond

L'apprentissage profond (Deep Learning) est une branche avancée de l'apprentissage automatique qui repose sur l'utilisation de réseaux de neurones artificiels à plusieurs couches, imitant le fonctionnement du cerveau humain. Il est particulièrement utilisé dans des domaines informatiques tels que la reconnaissance visuelle (panneaux routiers pour véhicules autonomes), la reconnaissance vocale, la robotique, la bio-informatique, la sécurité, la santé ou encore l'éducation assistée par ordinateur [42].

La figure 3.4 illustre l'architecture d'un réseau à plusieurs couches.

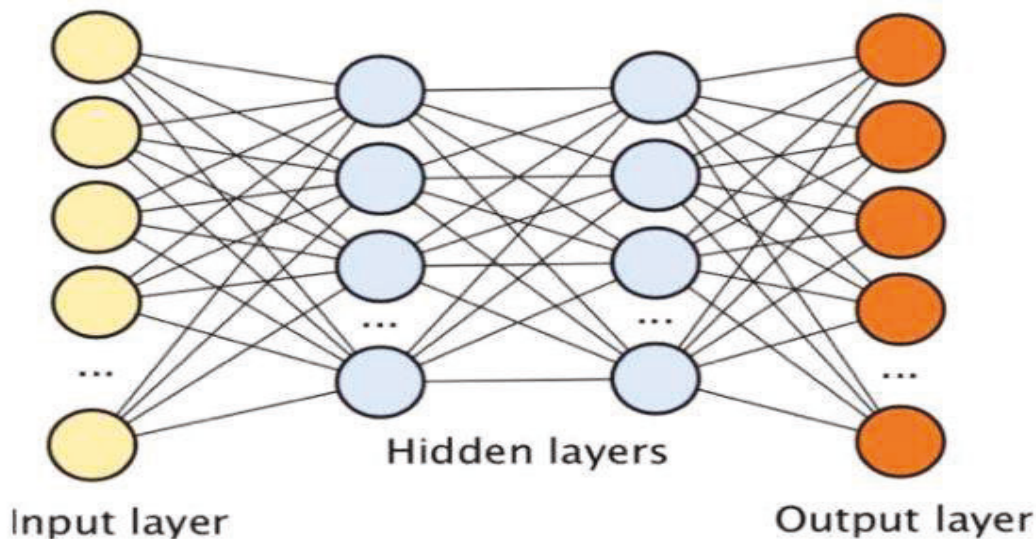


Figure 3.4 : Illustration de l'architecture de Deep Learning [19].

Contrairement au Machine Learning, qui s'appuie souvent sur des données structurées et réalise principalement des tâches de routine, le Deep Learning traite des données non structurées et permet d'exécuter des tâches complexes. Grâce à ses architectures neuronales profondes (Voir figure 3.4), il est capable d'analyser des volumes massifs d'informations, rendant possible des avancées majeures en intelligence artificielle [22] [42].

Une IA à apprentissage profond (Deep Learning) a la capacité d'apprendre de manière autonome, en découvrant par elle-même des modèles et des caractéristiques dans les données. En revanche, une IA à apprentissage automatique (Machine Learning) nécessite souvent une intervention humaine, notamment pour la sélection des caractéristiques pertinentes (feature engineering) et l'ajustement des modèles. Grâce à ses réseaux de neurones multicouches, le Deep Learning peut extraire automatiquement des informations complexes sans nécessiter une supervision humaine constante [22] [42].

3.3.1 Classification des méthodes d'apprentissage profond

3.3.1.1 Réseaux de neurones récurrents

Les réseaux neuronaux récurrents (RNN) sont une catégorie de réseaux de neurones largement utilisés en apprentissage profond (Deep Learning). Leur particularité réside dans leur capacité à réutiliser tel que schématisé à la figure 3.5, leurs sorties précédentes comme nouvelles entrées, ce qui les rend particulièrement adaptés au traitement des données séquentielles [23]. Ils peuvent être déployés et entraînés progressivement à l'aide de la rétropropagation classique ou d'une variante spécifique appelée rétropropagation dans le temps (BPTT).

La figure 3.5 illustre l'architecture d'un réseau de neurones récurrents.

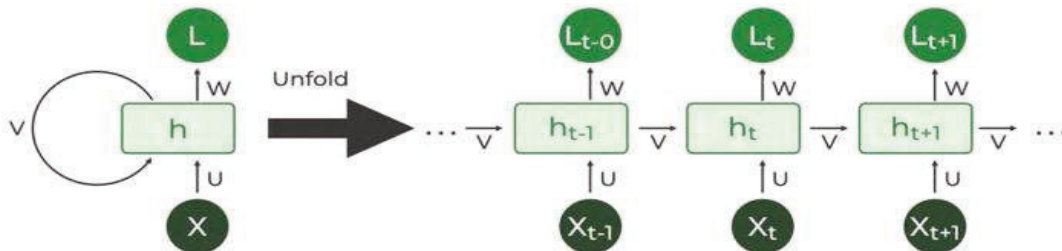


Figure 3.5 : Architecture des réseaux de neurones récurrents [21].

3.3.1.2 Réseaux de neurones artificiels

Les réseaux neuronaux artificiels s'inspirent des réseaux neuronaux biologiques qui structurent le cerveau humain. À l'image de ce dernier, ils se composent de neurones organisés en plusieurs couches et interconnectés entre eux.

Ces neurones, appelés nœuds, possèdent des équivalents biologiques : dans un réseau neuronal artificiel (ANN), les entrées correspondent aux dendrites, les nœuds aux noyaux cellulaires, les poids aux synapses et la sortie aux axones.

Les ANN sont des modèles statistiques non linéaires capables d'établir des relations complexes entre les entrées et les sorties afin d'identifier de nouveaux modèles. Ils sont largement utilisés dans divers domaines, notamment la reconnaissance d'images, la reconnaissance vocale, la traduction automatique et le diagnostic médical [33].

Le principal avantage des réseaux neuronaux artificiels (ANN) réside dans leur capacité à apprendre à partir d'échantillons de données. L'une de leurs applications les plus courantes est l'approximation de fonctions aléatoires. Grâce à ces technologies, il est possible d'obtenir des solutions optimisant la distribution de manière efficace et rentable [33].

De plus, les ANN peuvent générer un résultat de sortie à partir d'un simple échantillon de données, sans nécessiter l'ensemble complet. Leur forte capacité de prédiction en fait un outil précieux pour améliorer les méthodes d'analyse de données existantes.

La figure 3.6 illustre l'architecture d'un réseau de neurones artificiels.

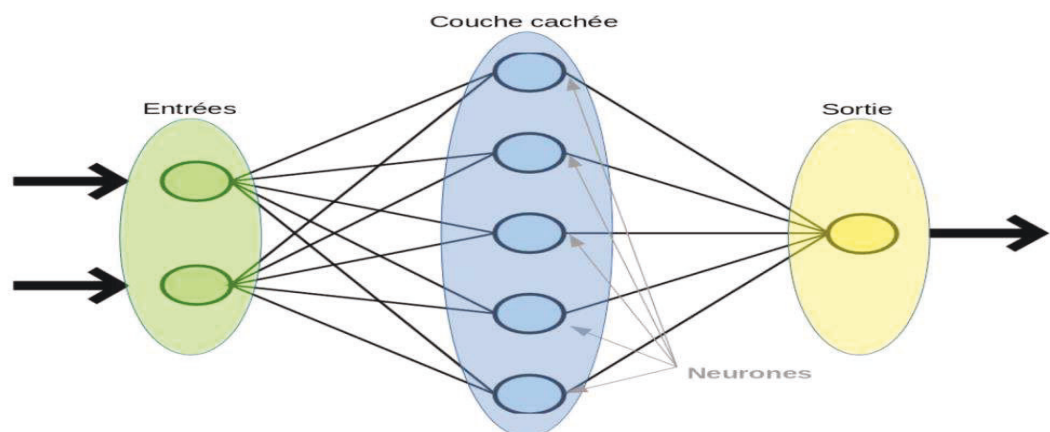


Figure 3.6 : L'architecture d'un réseau de neurones artificiels [88].

La figure 3.7 illustre l'architecture des réseaux neuronaux artificiels avec la mise en évidence des composantes d'un neurones artificiels soient les fonctions de transfert et d'activation (exemple sigmoid, tanh).

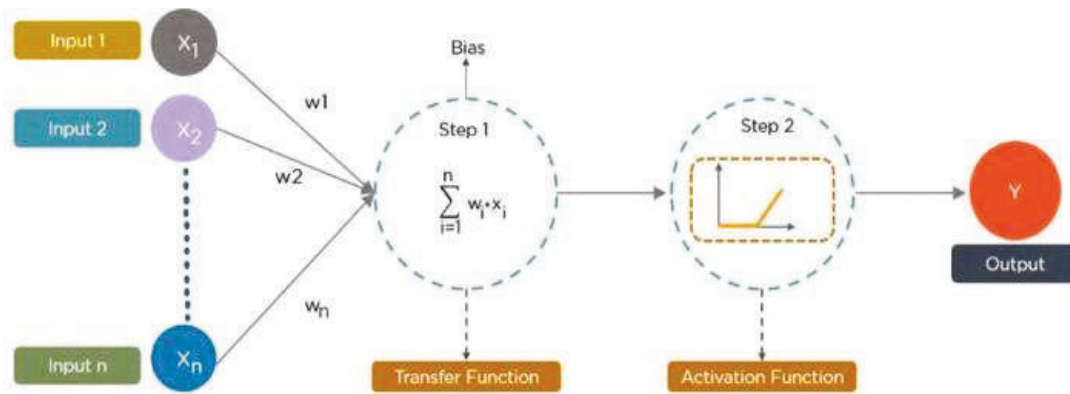


Figure 3.7 : L'architecture des réseaux neuronaux artificiels avec ses composantes [33].

Comme observé à la figure 3.7 la sortie d'un neurone peut être constituée d'une fonction ReLU (Rectified Linear Unit) qui est un élément clé de l'évolution de l'apprentissage profond. Simple à utiliser, elle surpasse néanmoins des fonctions d'activation traditionnelles comme Sigmoid ou Tanh, offrant ainsi de meilleures performances et une convergence plus rapide dans les réseaux neuronaux profonds.

La formule de ReLU est : $f(x) = \max(0, x)$

La fonction ReLU (Rectified Linear Unit) et sa dérivée sont toujours monotones. Lorsqu'elle reçoit une entrée négative, elle renvoie 0, tandis que pour une valeur positive, elle restitue cette valeur directement. Ainsi, sa sortie varie entre 0 et l'infini [34].

Le Leaky ReLU est une version modifiée de la fonction ReLU. Contrairement à la ReLU classique qui renvoie 0 pour les valeurs négatives en entrée, Leaky ReLU introduit une petite composante linéaire, permettant ainsi de conserver une faible valeur pour les entrées négatives, évitant ainsi le problème des neurones inactifs. Mathématiquement, on peut l'écrire comme [35].

$f(x) = 0.01$ pour tout $x < 0$

$f(x) = x$ pour tout $x \geq 0$

La fonction sigmoïde est quant à elle l'une des fonctions d'activation la plus couramment utilisée, car elle est non linéaire. La fonction sigmoïde convertit les valeurs en un intervalle compris entre 0 et 1. Elle peut être définie comme : $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$

La fonction sigmoïde est quant à elle une fonction en forme de S et continuellement différentiable. Le dérivé de la fonction est : $\text{sigmoid}'(x) = \text{sigmoid}(x) \cdot (1 - \text{sigmoid}(x))$

De plus, la fonction sigmoïde n'est pas symétrique par rapport à zéro, ce qui signifie que les sorties de tous les neurones auront le même signe. Cette limitation peut être atténuée en utilisant une variante de la fonction sigmoïde [35].

La fonction softmax est une opération mathématique qui convertit un vecteur de nombres réels en un vecteur de probabilités, dont la somme est égale à 1. Chaque valeur du vecteur initial est transformée en un nombre compris entre 0 et 1. La formule de la fonction softmax est de la forme :

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (3.1)$$

Comme mentionné précédemment, la fonction softmax peut être appliquée à un vecteur z de longueur K . Pour chaque élément de z , elle calcule l'exponentielle de la valeur, puis la divise par la somme des exponentielles de toutes les valeurs du vecteur z [36].

Dans le contexte des réseaux de neurones la fonction softmax est appliquée sur la sortie Y (voir figure 3.7) permettant ainsi de produire des sorties normalisées dans l'intervalle $[0,1]$.

3.3.1.3 Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (ConvNets ou CNN) sont une catégorie de réseaux de neurones particulièrement efficaces dans des domaines tels que la reconnaissance et la classification d'images.

Inspirés du perceptron multicouche et du fonctionnement biologique du cerveau, les CNN ont été appliqués avec succès à l'identification d'objets et de visages. L'un des premiers réseaux de neurones convolutifs, LeNet, a été proposé par Yann LeCun en 1988. Il était principalement utilisé pour la reconnaissance de caractères, notamment la lecture de codes postaux, de chiffres, et d'autres symboles [31].

En 2012, Alex Krizhevsky et Geoffrey Hinton ont remporté la compétition ImageNet en utilisant un réseau de neurones convolutifs (CNN), marquant ainsi le début d'une nouvelle ère pour l'apprentissage profond [45].

3.3.1.3.1 L'architecture réseaux de neurones convolutifs

Généralement, un réseau de neurones convolutifs (CNN) est composée de : couches de convolution, couches d'activation, couches de pooling et couches entièrement connectées.

La figure 3.8 illustre l'architecture et composition d'un réseau de neurones convolutifs.

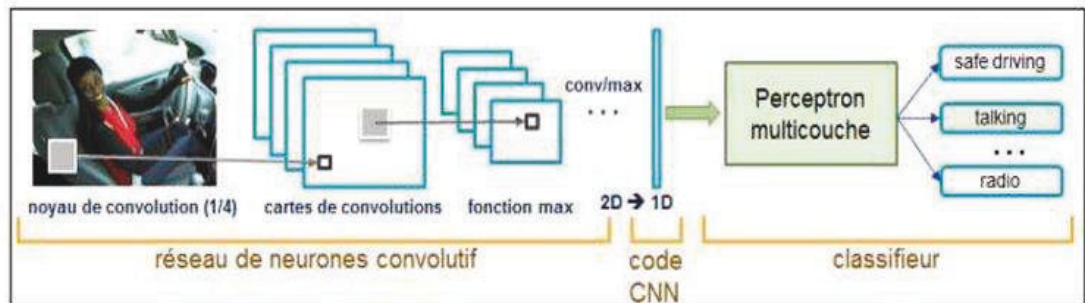


Figure 3.8 : Architecture et composition d'un réseau de neurones convolutifs [91].

Dans un réseau de neurones convolutifs (CNN), les couches de convolution réalisent les calculs nécessaires pour extraire des informations pertinentes à partir des données de l'image. Chaque couche de convolution est composée de noyaux, qui sont des filtres dont les pondérateurs peuvent être déterminées par un processus d'apprentissage. Une image en couleur peut être représentée comme une grille tridimensionnelle de pixels, tandis qu'un filtre est une matrice bidimensionnelle de poids appliquée aux différentes zones de l'image pour détecter des caractéristiques spécifiques [92].

Un filtre est généralement une matrice de taille 3x3, bien que ses dimensions puissent varier. Il effectue un produit scalaire entre les valeurs des pixels de l'image et ses propres poids. Ensuite, il est déplacé de n pixels, un mouvement appelé stride (souvent fixe à 1), avant d'appliquer à nouveau le calcul sur une autre zone de l'image. Ce processus, connu sous le nom de convolution, se répète jusqu'à couvrir l'ensemble de la matrice [92].

Les résultats obtenus à chaque position du filtre forment une grille appelée carte d'activation ou carte de caractéristiques. Cette carte contient les informations extraites par le filtre, mettant en évidence les éléments détectés dans l'image d'origine [37].

La figure 3.9 illustre le processus de convolution appliquée dans une couche convolutive d'un réseau de neurones convolutifs (CNN) :

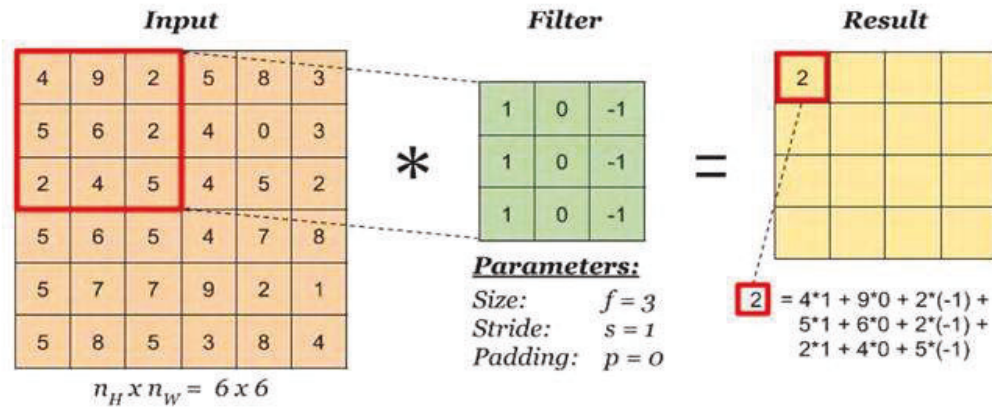


Figure 3.9 : Processus de convolution [37].

Comme illustré dans les figures 3.9 et 3.10, le noyau ou filtre possède une dimension de 3x3, et une sous-matrice de même taille est extraite de l'image. Le produit des matrices est ensuite stocké dans la matrice de sortie. Le filtre se déplace ensuite d'un pixel vers l'avant, conformément au pas de 1 utilisé dans cet exemple (voir figure 3.10), et l'opération de calcul du produit scalaire est répétée [37].

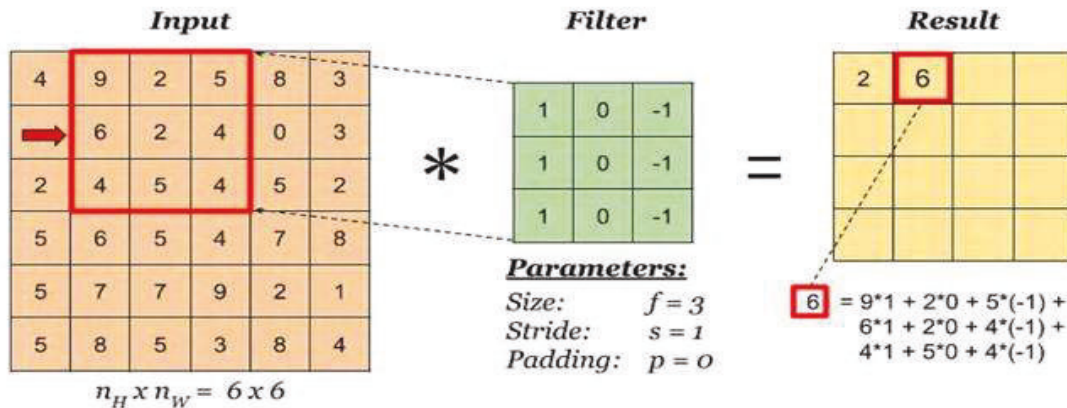


Figure 3.10 : Autre application d'un processus de convolution [37].

Le processus de convolution poursuit son déplacement sur la grille de la même façon, générant une carte de caractéristiques. Le pas peut être défini par n'importe quel entier, sachant qu'une valeur plus élevée réduit la taille de la matrice de sortie.

L'efficacité du traitement peut être améliorée en insérant, entre les couches de traitement, une couche d'activation qui applique une fonction mathématique aux signaux de sortie. La fonction d'activation Relu contraint les neurones à produire uniquement des valeurs positives.

La couche ReLU ne modifie pas la taille de l'entrée, ce qui signifie que les dimensions x et y restent identiques. En réalité, l'Unité Linéaire Rectifiée (ReLU) peut être vue comme une troncature appliquée individuellement à chaque élément de l'entrée [39].

$$y_{i,j,d} = \max\{0, x_{i,j,d}^l\} \quad (3.2)$$

Avec $0 \leq i < H_{l+1}, 0 \leq j < W_{l+1}, 0 \leq d < D_{l+1}$

H_l , W_l et D_l représentent respectivement la hauteur, la largeur et la profondeur les dimensions du tenseur d'entrée (ou de sortie) pour une certaine couche d'un réseau de neurones convolutifs (CNN).

Il n'y a pas de paramètre à l'intérieur d'une couche ReLU, donc pas besoin de paramètre d'apprentissage dans cette couche.

$$\frac{dy_{i,j,d}}{dx_{i,j,d}^l} = \llbracket x_{i,j,d}^l > 0 \rrbracket, \quad (3.3)$$

Cette formule valide dans le processus d'entraînement (rétropropagation) est la fonction indicatrice, étant 1 si son argument est vrai, et 0 sinon. Par conséquent, nous avons :

$$\left[\frac{\partial z}{\partial x^l} \right]_{i,j,d} = \begin{cases} \left[\frac{\partial z}{\partial y} \right]_{i,j,d}, & \text{si } x_{i,j,d}^l > 0 \\ 0, & \text{sinon} \end{cases} \quad (3.4)$$

Autrement dit, lorsque $x_{i,j,d}^l > 0$, le gradient est rétropropagé sans modification vers la couche précédente. En revanche, lorsque $x_{i,j,d}^l \leq 0$, le gradient est annulé, ce qui empêche toute mise à jour associée à ce neurone.

L'objectif principal de la fonction ReLU est d'accroître la non-linéarité du réseau de neurones convolutif (CNN). En effet, l'information sémantique contenue dans une image (par exemple, une personne et un Husky assis côte à côte sur un banc dans un jardin) est une représentation hautement non linéaire des valeurs de pixels de l'entrée. Par conséquent, nous souhaitons que la transformation de l'entrée vers la sortie du CNN soit également fortement non linéaire [39].

Bien que simple, la fonction ReLU est intrinsèquement non linéaire, ce qui la rend particulièrement adaptée à cet objectif.

Si l'on considère $x_{i,j,d}^l$ comme l'une des caractéristiques extraites par les couches convolutives au niveau l , appartenant à un tenseur de dimension $H_l \times W_l \times D_l$, issu de la couche précédente $l - 1$. Cette valeur peut être positive, nulle ou négative. Par exemple, $x_{i,j,d}^l$ peut être positif si une région spécifique de l'image d'entrée contient certains motifs distinctifs, comme une tête de chien, une tête de chat ou d'autres formes similaires.

En revanche, cette valeur sera nulle ou négative lorsque les motifs correspondants sont absents. L'application de la fonction d'activation ReLU (Rectified Linear Unit) transformera toutes les valeurs négatives en zéro, ce qui signifie que $x_{i,j,d}^l$ ne sera activé que lorsque ces motifs seront présents dans cette région particulière de l'image. Intuitivement, cette propriété est essentielle pour la détection de motifs et la reconnaissance d'objets complexes [39].

La figure 3.11 illustre la représentation graphique de la fonction d'activation ReLU.

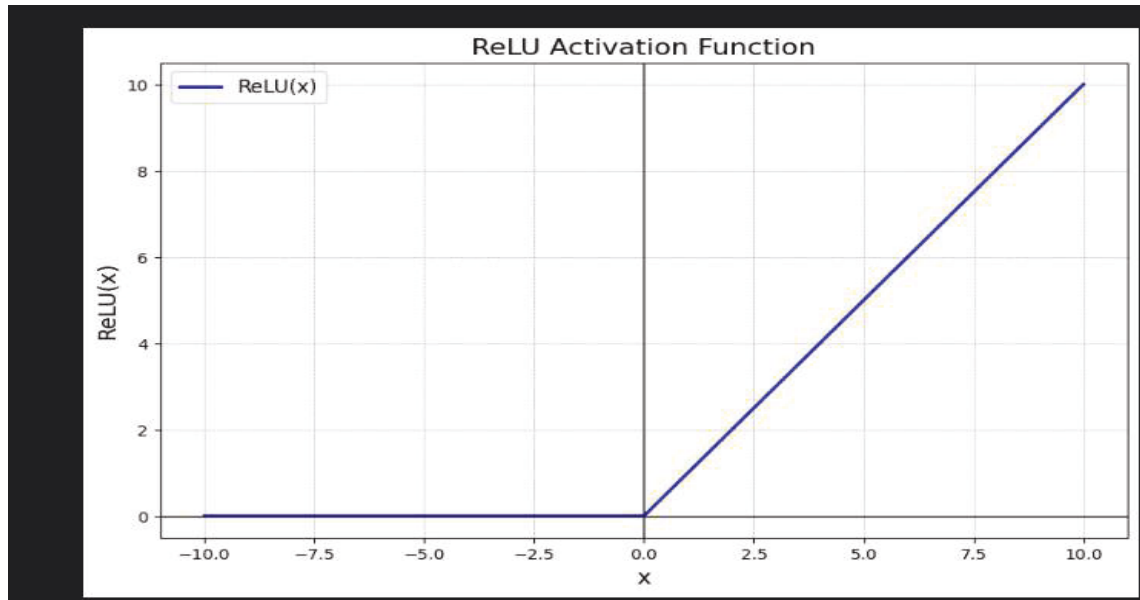


Figure 3.11 : Représentation graphique de la fonction d'activation ReLU [39].

Cette fonction est définie par $\text{ReLU}(x) = \max(0, x)$, ce qui signifie que toutes les valeurs négatives de l'entrée sont annulées, tandis que les valeurs positives sont conservées sans modification.

La couche de pooling est généralement insérée entre deux couches de convolution. Son rôle est d'effectuer une opération de pooling, un processus de sous-échantillonnage visant à réduire la taille des images tout en conservant leurs principales caractéristiques. Pour ce faire, l'image est divisée en cellules régulières de 2×2 pixels sans chevauchement ou en cellules de 3×3 pixels avec un pas de 2 pixels. Le pooling est ensuite appliqué, produisant des sorties de dimensions réduites par rapport aux entrées, mais contenant toujours les informations essentielles. Cette réduction du nombre de paramètres et des calculs améliore les performances du réseau tout en limitant le risque de surapprentissage [38].

Il existe deux types courants de couches de pooling comme le montre la figure 3.12 : le max pooling et l'average pooling. Le premier consiste à sélectionner la valeur maximale, tandis que le second calcule la valeur moyenne. Le max pooling est généralement privilégié par rapport à l'average pooling. De plus, la couche de pooling ne contient aucun paramètre à apprendre. L'intuition derrière le max pooling est qu'une valeur élevée indique la présence d'une caractéristique détectable [40].

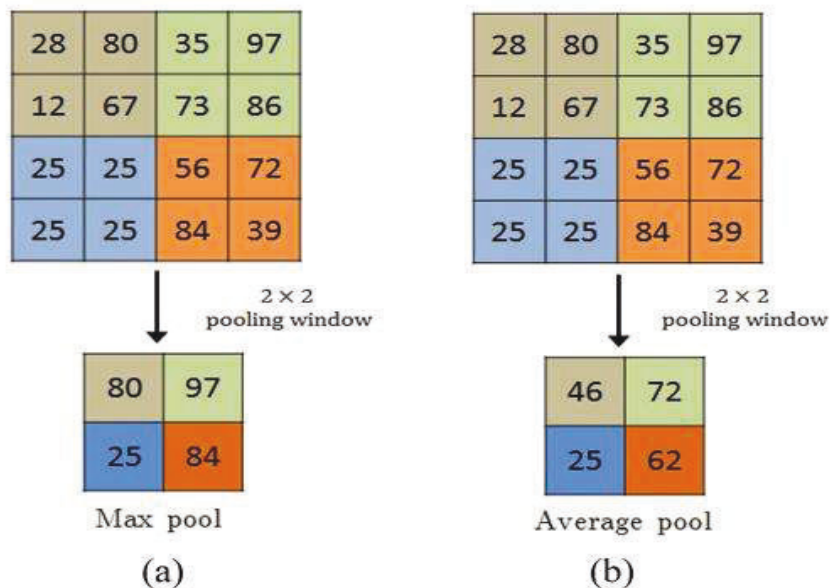


Figure 3.12 : Types de fonctions pooling [41].

La couche fully-connected (couches classifications) est systématiquement la dernière couche d'un réseau neuronal, qu'il soit convolutif ou non, et n'est donc pas spécifique aux CNN. Son rôle est de recevoir un vecteur en entrée et de générer un nouveau vecteur en sortie [30].

La couche de classification transforme les activations finales en prédictions, souvent sous forme de probabilités de classes.

3.3.1.3.2 Architectures CNN populaires

Les architectures de réseaux de neurones convolutifs (CNN) les plus populaires sont décrites ci-dessous [43] :

LeNet est un réseau neuronal convolutif emblématique conçu par Yann LeCun et ses collaborateurs en 1990, puis optimisé en 1998. L'architecture la plus performante et la plus renommée de LeNet a été largement utilisée pour la reconnaissance des codes postaux, des chiffres et d'autres caractères [31]. L'architecture de LeNet repose sur une succession structurée de couches de convolution, de couches de sous-échantillonnage (pooling) et de couches entièrement connectées, permettant une extraction hiérarchique des caractéristiques visuelles.

La figure 3.13 montre le schéma fonctionnel d'un réseaux de neurones convolutifs LeNet.

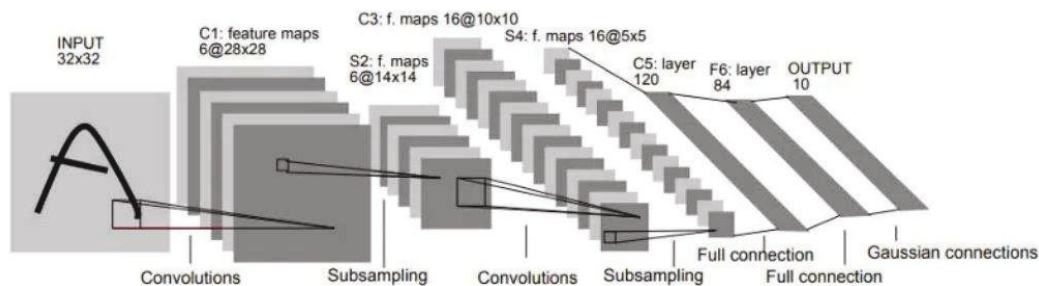


Figure 3.13 : Schéma fonctionnel du CNN LeNet [31].

La première architecture de réseau neuronal convolutif largement reconnue est AlexNet, qui a contribué à populariser l'utilisation des CNN en vision par ordinateur. Développée par Alex Krizhevsky et ses collaborateurs, elle a été présentée en 2012 lors du défi ILSVRC (ImageNet Large Scale Visual Recognition Challenge). AlexNet a surclassé ses concurrents en obtenant un taux d'erreur de 16 % dans le Top-5, contre 26 % pour le deuxième finaliste, marquant ainsi une avancée majeure dans le domaine [45].

La figure 3.14 présente le schéma fonctionnel de l'architecture AlexNet, illustrant la succession des différentes couches depuis l'image d'entrée jusqu'à la sortie de classification.

Figure 3.14 : Schéma fonctionnel de l'architecture du réseau de neurone convolutif Alexnet [45].

Après AlexNet, le réseau neuronal convolutif développé par Matthew Zeiler et Rob Fergus a remporté l'ILSVRC 2013. Nommé ZFNet (Zeiler & Fergus Net), celui-ci a apporté des améliorations à AlexNet en optimisant les hyperparamètres de l'architecture. Ces améliorations incluaient principalement l'augmentation des dimensions des couches de convolution centrales ainsi que la réduction de la taille des foulées et des filtres dans la couche primaire, permettant ainsi une meilleure extraction des caractéristiques [44].

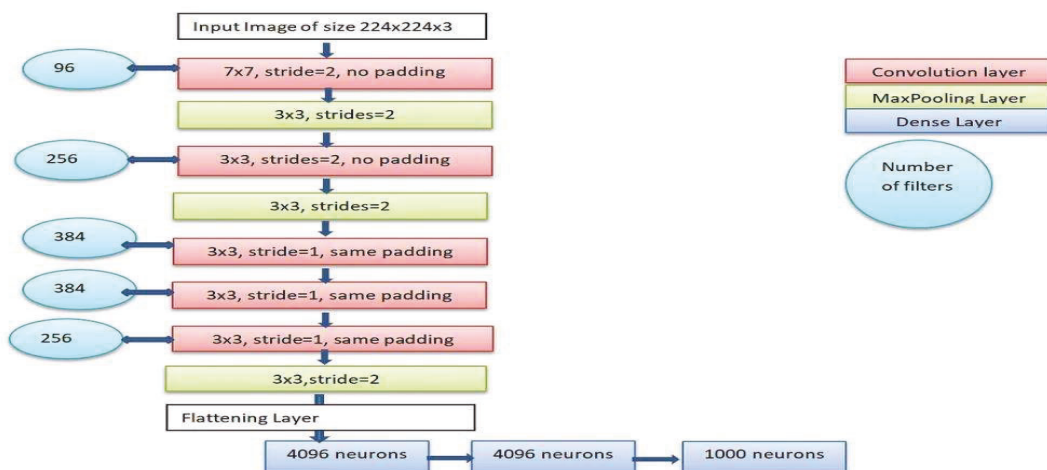


Figure 3.15 : Schéma fonctionnel de l'architecture du réseau de neurones convolutifs ZFNet [44].

GoogLeNet est une architecture, développée par Szegedy et al. chez Google, a remporté l'ILSVRC 2014. Elle a atteint un taux d'erreur de 6,67 % dans le Top-5, se rapprochant ainsi des performances humaines. GoogLeNet se distinguait par ses 22 couches, tout en étant plus efficace qu'AlexNet en termes d'hyperparamètres : il n'en comptait que 4 millions, contre 60 millions pour AlexNet, ce qui optimisait son efficacité et sa consommation de ressources [46].

La figure 3.16 illustre l'architecture complète de GoogLeNet, mettant en évidence l'enchaînement des couches convolutives, des modules Inception et des couches de classification.

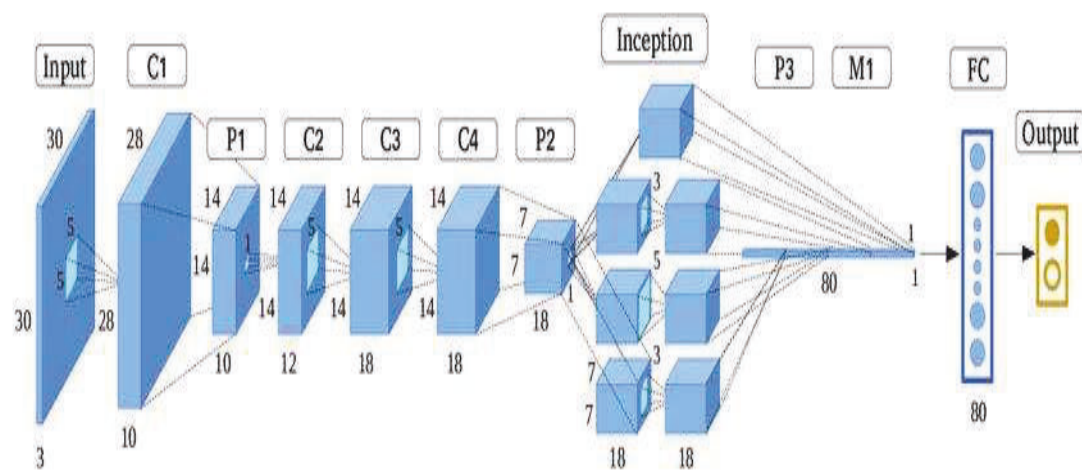


Figure 3.16 : Schéma fonctionnel de l'architecture du réseau de neurones convolutifs GoogLeNet avec toutes ses caractéristiques [46].

Comparée à LeNet, AlexNet et ZFNet, l'architecture GoogLeNet se distingue par sa profondeur élevée, son faible nombre de paramètres et l'introduction des modules Inception, permettant une analyse multi-échelle efficace des caractéristiques visuelles. Alors que LeNet reste adaptée à des tâches simples, alors que AlexNet et ZFNet nécessitent quant à eux des ressources importantes.

VGGNet est une architecture de réseau neuronal convolutif développée par Karen Simonyan, Andrew Zisserman et al. à l'Université d'Oxford. Ce modèle comprend 16 couches et possède jusqu'à 95 millions de paramètres, ayant été entraîné sur plus d'un milliard d'images réparties en 1000 classes. Il prend en charge des images d'entrée de 224 x 224 pixels et extrait 4096 caractéristiques convolutives [47].

VGGNet est particulièrement apprécié pour son efficacité computationnelle et constitue une base robuste pour de nombreuses applications en vision par ordinateur, notamment la détection d'objets, grâce à sa flexibilité et sa capacité à être transféré vers diverses tâches. Les architectures CNN ont évolué de LeNet, adaptée à des tâches simples, vers des modèles plus profonds comme AlexNet et ZFNet, offrant de meilleures performances mais avec un coût computationnel élevé. GoogLeNet a amélioré l'efficacité grâce aux modules Inception, tandis que VGGNet se distingue par sa précision et sa simplicité architecturale, au prix d'une forte consommation de ressources.

La figure 3.17 illustre l'architecture générale du réseau convolutif VGGNet, mettant en évidence l'enchaînement des couches de convolution, de pooling, des couches entièrement connectées et la fonction d'activation softmax.

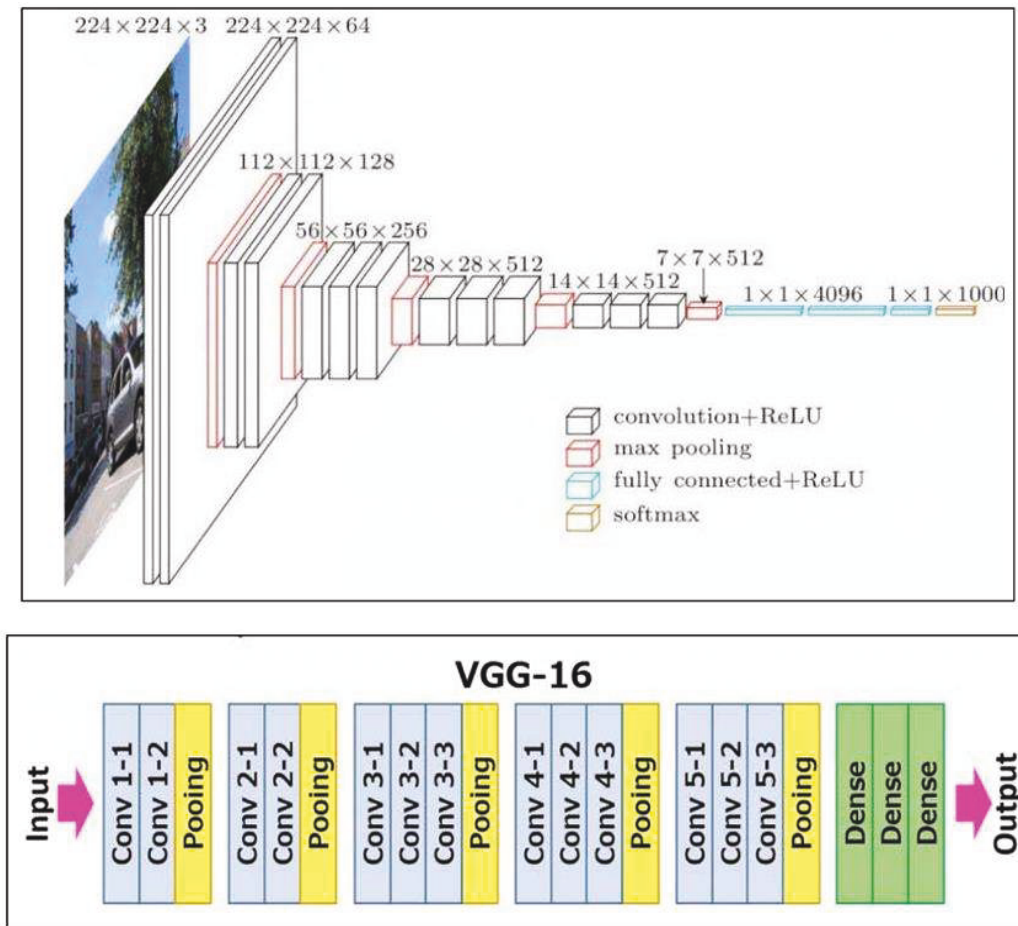


Figure 3.17 : L'architecture du réseau de neurone convolutif VGGNet [47].

L'architecture Inception est constituée d'un certains nombres de modules. Ces modules intègrent des filtres de différentes tailles (1x1, 3x3 et 5x5) permettant de capturer diverses informations spatiales. Ensuite, un filtre de concaténation est utilisé pour regrouper les résultats obtenus. De plus, afin de limiter la densité des connexions, une couche de pooling moyen global est employée à la place d'une couche entièrement connectée. Ces ajustements ont permis une réduction significative du nombre de paramètres [48].

La structure générale d'un module Inception est illustrée à la figure 3.18.

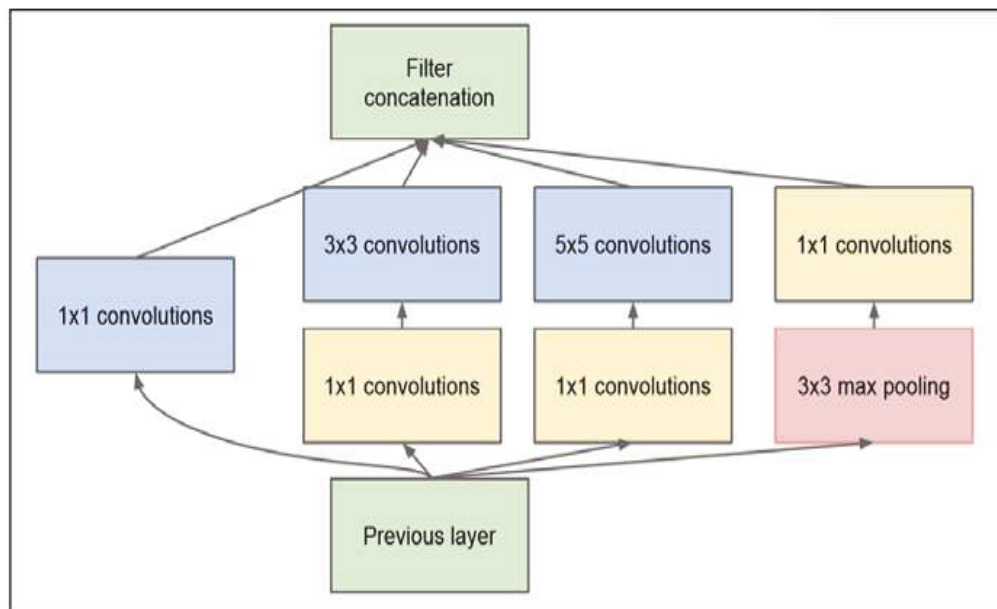


Figure 3.18 : Architecture du module Inception [48].

Kaiming He et al, [49] ont développé le réseau résiduel (ResNet), qui est un modèle de réseau de neurones convolutifs qui se distingue par ses connexions résiduelles (skip connections) et l'utilisation de la normalisation par lots (Batch Normalization). Contrairement à d'autres architectures, ResNet ne possède pas de couches entièrement connectées en fin de réseau, ce qui le rend plus efficace en termes de calcul [49].

Cependant, son principal inconvénient réside dans la complexité de son évaluation, en raison du grand nombre de paramètres qu'il contient. Malgré cela, ResNet (voir figure 3.19) reste un modèle de référence en vision par ordinateur et constitue souvent le choix privilégié pour l'implémentation de réseaux convolutifs profonds dans les applications pratiques [49].

La figure 3.19 illustre l'architecture du réseau résiduel ResNet.

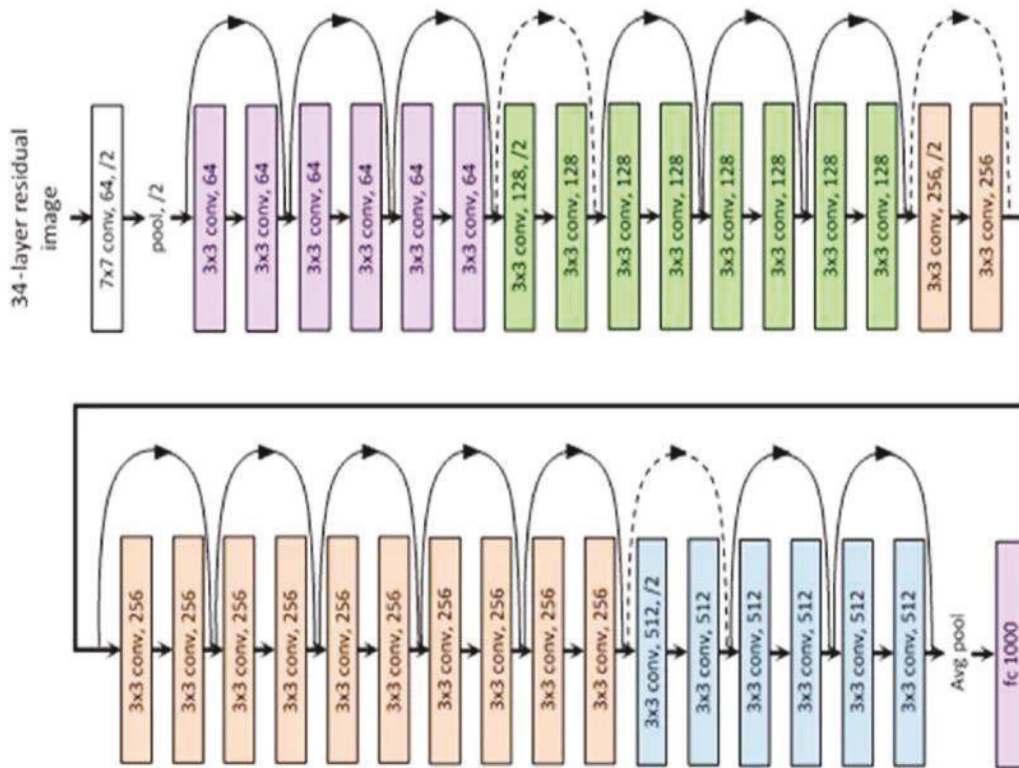


Figure 3.19 : L'architecture du réseau résiduel ResNet [49].

3.3.1.3.3 L'architecture MobileNetV2

MobileNetV2 est une architecture de réseau neuronal convolutif conçu de façon à bien fonctionner sur les appareils mobiles. Il est basé sur une structure résiduelle inversée où les connexions résiduelles se trouvent entre les couches de goulot d'étranglement. La couche d'expansion intermédiaire utilise des convolutions légères dans le sens de la profondeur pour filtrer les caractéristiques en tant que source de non-linéarité. Dans l'ensemble, l'architecture de MobileNetV2 contient la couche initiale entièrement convolution avec 32 filtres, suivie de 19 couches de goulot d'étranglement résiduelles [57].

La figure 3.20 illustre l'architecture détaillée de MobileNetV2.

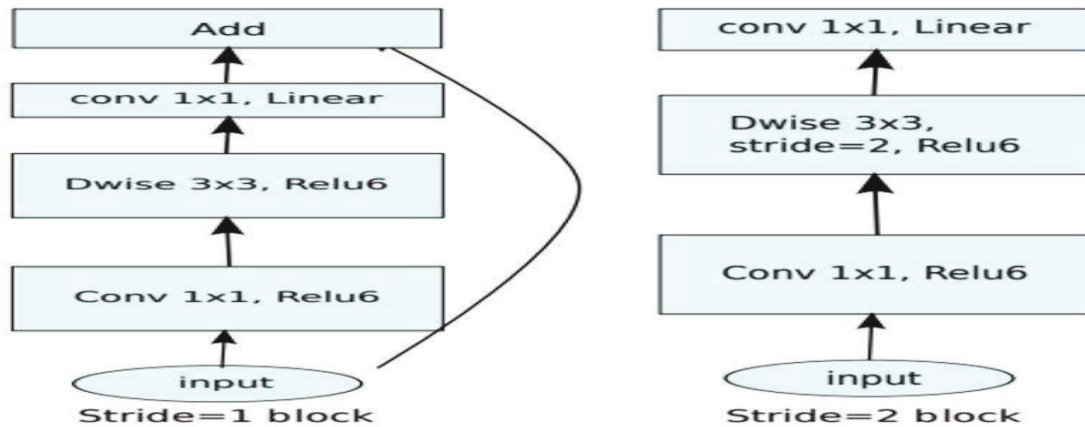


Figure 3.20 : Architecture détaillée de MobileNetV2 [57].

- **Composants d'un détecteur d'objets**

Un détecteur d'objets est généralement composé généralement de trois parties.

- Backbone (Dorsale)
- Neck
- Head (Tête)

La figure 3.21 illustre l'architecture du détecteur d'objet.

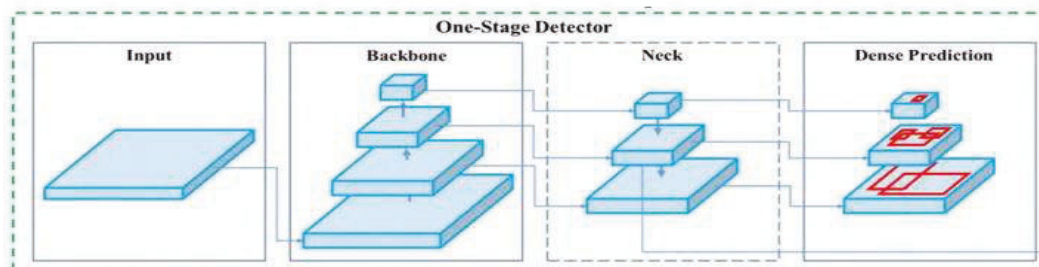


Figure 3.21 : L'architecture de détecteur d'objets [58].

- ❖ **Backbone (Dorsale)**

Un backbone est utilisé comme extracteur de caractéristiques préentraîné à l'aide de tâches de classification sur le jeu de données ImageNet. Les détecteurs d'objets entraînés et testés sur GPU s'appuient généralement sur des backbones plus lourds, tels que VGG, ResNet, DenseNet ou Darknet-19/53. En revanche, pour une utilisation sur CPU, on privilégie des backbones plus légers comme SqueezeNet ou MobileNet. Il existe toujours un compromis entre

vitesse et précision : les backbones légers offrent une exécution plus rapide, mais au prix d'une précision inférieure comparée à celle des modèles plus lourds [58].

Dans la détection de la somnolence, le backbone pourrait jouer un rôle central, car c'est lui qui ferait l'extraction des caractéristiques visuelles pertinentes à partir des images du visage d'un conducteur avant la classification finale.

❖ Neck

Le Neck de MobileNetV2 combine et fusionne les représentations des couches ConvNet avant de les transmettre à la tête de prédiction (Head). Il fait partie intégrante du réseau et joue un rôle clé en collectant des cartes de caractéristiques (features) issues de différentes étapes de l'épine dorsale (backbone). En termes simples, il s'agit d'un agrégateur de caractéristiques. Le Neck du pipeline de détection d'objets sera abordé plus en détail dans les sections suivantes [58].

Il permet de mieux exploiter les caractéristiques extraites par le backbone pour renforcer la précision et la stabilité des prédictions dans la détection de la somnolence.

❖ Head (Tête)

La Head est utilisée pour prédire les coordonnées des boîtes englobantes ainsi que les étiquettes de classe des objets.

C'est elle qui transforme les informations extraites et fusionnées par le backbone et le neck en décisions finales interprétables [58].

3.4 Travaux connexes à la détection de la somnolence

D'autres travaux de recherche ont été consacrés à la détection de la somnolence chez les conducteurs. Plusieurs approches et applications ont été proposées dans ce domaine, illustrant la diversité des méthodes et technologies mises en œuvre pour améliorer la sécurité routière.

- Mallah et al. [54] ont mis au point un système robuste aux variations de la luminosité pour la détection d'objets. Ils ont exploité l'algorithme Haar, associé au classificateur de visage intégré aux bibliothèques OpenCV. La délimitation des zones oculaires est effectuée à partir de la région faciale en s'appuyant sur des facteurs anthropométriques (hauteur, largeur, distances relatives entre les yeux, le nez et la bouche). Par la suite, la détection des paupières permet d'évaluer le degré de fermeture des yeux.

- Tian et Qin [55] ont développé un système innovant visant à détecter l'état des yeux d'un conducteur, un facteur clé pour évaluer son niveau d'attention visuelle. Ce système fonctionne en temps réel et repose sur une combinaison de plusieurs techniques avancées de traitement d'image, offrant ainsi des performances supérieures à celles des approches arithmétiques traditionnelles.

Le système utilise différentes méthodes pour détecter l'état des yeux, adaptées selon les conditions de luminosité (jour et nuit).

- De jour, il combine la correspondance de la couleur de la peau et la projection verticale pour détecter et suivre la région du visage, validée ensuite par l'analyse de la symétrie géométrique du visage.
- De nuit, il applique une analyse basée sur l'intensité de la lumière associée à la projection verticale pour accomplir la même tâche.

Ensuite, la position des yeux est détectée à l'aide de la projection horizontale, puis validée par certaines caractéristiques faciales. Enfin, une nouvelle fonction de complexité est introduite afin de quantifier de manière objective l'état des yeux du conducteur à partir des variations observées dans la région oculaire au cours du temps. Cette fonction reposant sur l'analyse des changements d'intensité et de texture dans la zone des yeux entre des images consécutives de la séquence vidéo est utilisée pour évaluer l'état des yeux, permettant d'enregistrer la fréquence de clignements et de fermetures des paupières [32].

3.5 Conclusion

Dans ce chapitre, nous avons abordé les principes fondamentaux de l'apprentissage automatique ainsi que les bases de l'apprentissage profond qui sont largement exploités dans la détection de la somnolence d'un conducteur. Nous nous sommes ensuite penchés sur les réseaux de neurones convolutifs (CNN) en analysant leurs structures et architectures respectives. Nous avons également examiné en détail plusieurs modèles de CNN bien connus, tels que GoogleNet, ResNet et VGGNet. Enfin, nous avons présenté un aperçu des avancées récentes dans la détection de la somnolence par les travaux de Tian et Qin et Mallah et al [54] [55]. Le prochain chapitre sera consacré à la conception de l'architecture du système de détection de la somnolence et la distraction implémenté dans le cadre de ce travail de recherche.

Chapitre 4

Conception de l'architecture pour la détection de la somnolence et la distraction

4.1 Introduction

Le processus de détection des objets dans une image repose sur deux étapes essentielles : la localisation des objets et leur classification. Parmi les approches les plus couramment utilisées, les réseaux de neurones convolutifs (CNN) jouent un rôle clé dans la reconnaissance et la localisation des objets.

Les algorithmes de détection basés sur les CNN se distinguent par leur efficacité et leur robustesse, ce qui explique leur large adoption dans le développement de systèmes de détection de la somnolence et de la distraction chez un conducteur.

Dans ce chapitre, nous présenterons l'architecture du réseau de neurones implémentée dans cette recherche conçu pour la détection de la somnolence et de la distraction à l'aide d'un algorithme basé sur les CNN. Nous commencerons par une vue d'ensemble de la structure du modèle avant d'examiner en détail le rôle et le fonctionnement de chacun de ses composants.

4.2 Objectif

L'objectif de la détection de la somnolence et de la distraction est d'identifier les signes de fatigue chez les conducteurs afin de prévenir les risques d'accidents de la route. Ce système permet une surveillance en temps réel de l'état d'éveil et de vigilance des conducteurs, en particulier lorsqu'ils parcourent de longues distances ou évoluent dans des conditions propices à la somnolence.

Une détection précoce de la somnolence et de la distraction permet la mise en place de mesures préventives adaptées, telles que des rappels pour effectuer des pauses régulières, des alertes sonores ou visuelles, voire des recommandations explicites de repos. En réduisant les risques liés à la fatigue au volant, cette technologie contribue à renforcer la sécurité routière et à protéger la vie des usagers.

4.3 L'architecture générale

Notre système suit plusieurs étapes afin de construire un modèle d'apprentissage profond dédié à la détection de la somnolence et de la distraction.

Dans un premier temps, le système collecte un ensemble de données et applique un prétraitement adapté comprenant la détection du visage, la normalisation des images et l'extraction des zones d'intérêt (yeux et bouche). Ces données prétraitées sont ensuite introduites dans le modèle MobileNetV2 qui assure la détection et l'extraction automatique des caractéristiques visuelles, permettant d'obtenir un modèle à la fois léger, précis et performant.

Par la suite, le processus de détection est renforcé par l'intégration du Ratio Aspectuel des Yeux (EAR), un indicateur clé permettant d'évaluer le niveau de fatigue du conducteur à partir de la durée et de la fréquence de fermeture des paupières. Cette approche complémentaire facilite une détection plus efficace et robuste des signes de somnolence et de distraction.

Enfin, la fusion des résultats issus de MobileNetV2 et de l'analyse du Ratio Aspectuel des Yeux (EAR) permet d'accroître la précision globale du système, contribuant ainsi au renforcement de la sécurité routière et à la prévention des accidents liés à la fatigue au volant.

La figure 4.1 montre l'architecture générale du système de détection de la somnolence et de la distraction implémentée dans cette recherche, décrivent ainsi les principales étapes du processus de détection.

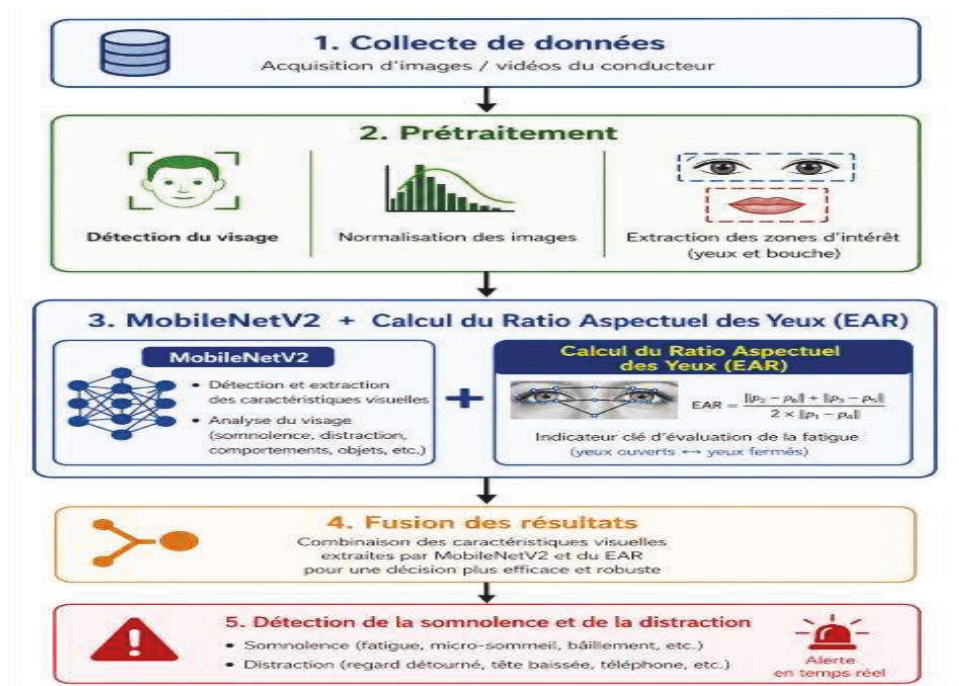


Figure 4.1 : L'architecture générale du système de détection de la somnolence et la distraction.

4.3.1 L'architecture détaillée du système de détection de la somnolence et distractions

Notre système de détection de la somnolence et de distractions repose sur l'apprentissage profond et suit un processus structuré en plusieurs étapes pour construire un modèle performant.

Dans un premier temps, le système extrait les données visuelles à partir d'images, d'enregistrements vidéo ou de flux en temps réel capturés par une caméra embarquée. Ces données sont ensuite traitées par le modèle MobileNetV2, après avoir subi un prétraitement adapté comprenant la détection du visage, la normalisation des images et l'extraction des régions d'intérêt (yeux et bouche).

Grâce au modèle MobileNetV2, le système est capable de déterminer l'état du visage, des yeux et de la bouche du conducteur, en identifiant si ces derniers sont ouverts ou fermés. Sur cette base, une évaluation du niveau de somnolence est réalisée et classée en trois catégories : faible, moyenne ou élevée.

Enfin, la combinaison de MobileNetV2 avec l'analyse du Ratio Aspectuel des Yeux (EAR) permet d'améliorer la précision globale du système. MobileNetV2 se charge de l'analyse visuelle des yeux et de la bouche, tandis que l'EAR fournit une mesure quantitative du degré de fatigue à partir des variations de clignement et de fermeture des paupières. Cette approche conjointe assure une détection plus fiable et plus réactive de la somnolence et de la distraction, contribuant ainsi à renforcer la sécurité routière.

Les étapes de traitement du flux d'images du visage d'un conducteur permettant la détection automatique de la somnolence ou distraction sont présentées dans la figure 4.2.

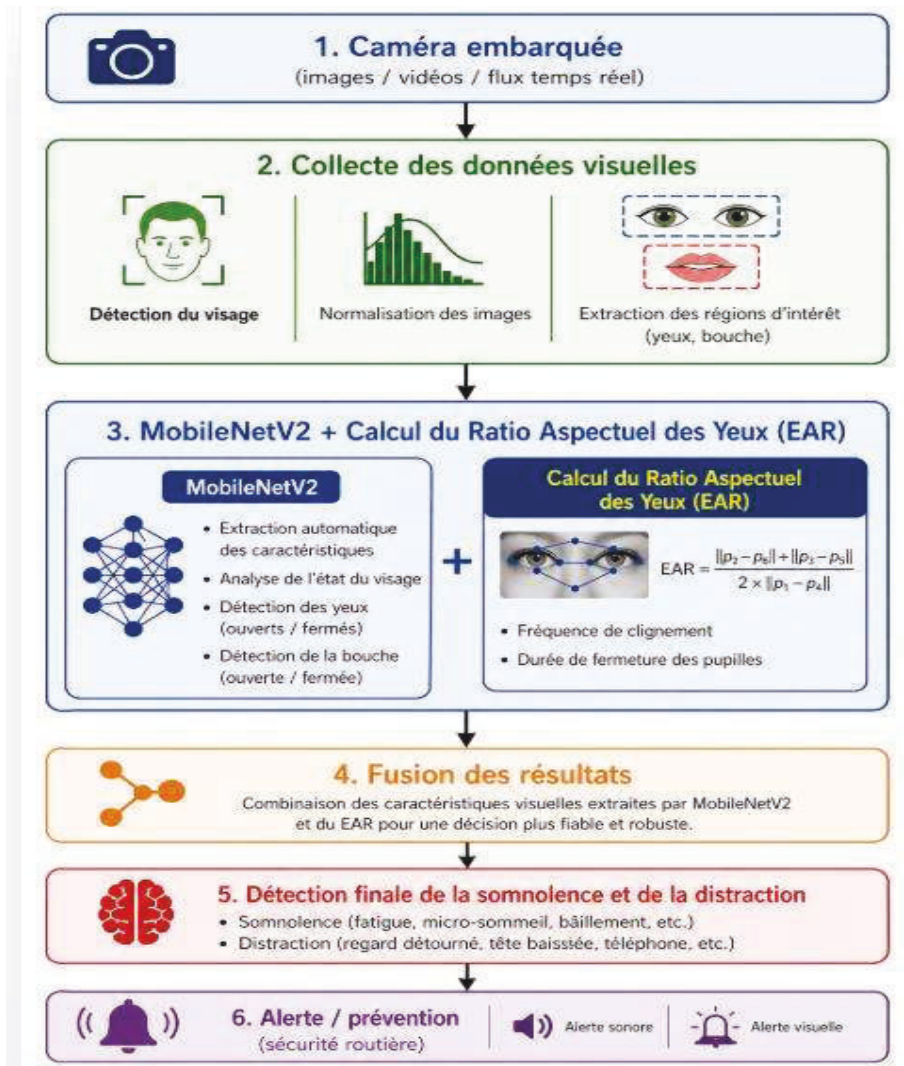


Figure 4.2 : Description des étapes de la détection de la somnolence et de la distraction.

4.3.2 Préparation des données

Avant d'exploiter les données dans les modèles d'apprentissage profond, une préparation rigoureuse s'avère indispensable. Ce processus comprend plusieurs étapes essentielles, à savoir : la collecte des données brutes (images, vidéos ou flux en temps réel), leur nettoyage afin d'éliminer les incohérences, les doublons ou les artefacts, puis le prétraitement, qui englobe la normalisation, la mise à l'échelle et l'augmentation des données qui consiste à générer artificiellement de nouvelles images à partir des échantillons existants. L'augmentation étant réalisée à l'aide de transformations géométriques et photométriques, telles que : la rotation (modification de l'angle du visage), le recadrage (cropping) ou le changement d'échelle.

Ces différentes étapes visent à garantir que le modèle dispose de données homogènes et de haute qualité, optimisant ainsi sa capacité d'apprentissage et améliorant la précision de la détection de la somnolence et de la distraction d'un conducteur.

4.3.2.1 MobileNetV2

Le programme a pour objectif principal d'alerter le conducteur dès l'apparition de signes de somnolence ou de distraction. Pour ce faire, il utilise la caméra embarquée afin de capturer des images en temps réel ou d'analyser des vidéos pour extraire des images pertinentes du visage du conducteur.

Afin d'améliorer la précision du modèle, des vidéos spécifiques sur la somnolence et la distraction ont été intégrées, permettant ainsi d'enrichir son apprentissage préalable et d'optimiser sa capacité à détecter ces états de somnolence et de distraction. Ces vidéos permettent d'analyser différents états de vigilance, notamment l'état d'éveil, la somnolence et la distraction, à travers l'observation des comportements faciaux et oculaires.

Les figures 4.3 à 4.8 sont des images tirées de séquences de vidéos de conducteurs en situation de conduite en simulateur.

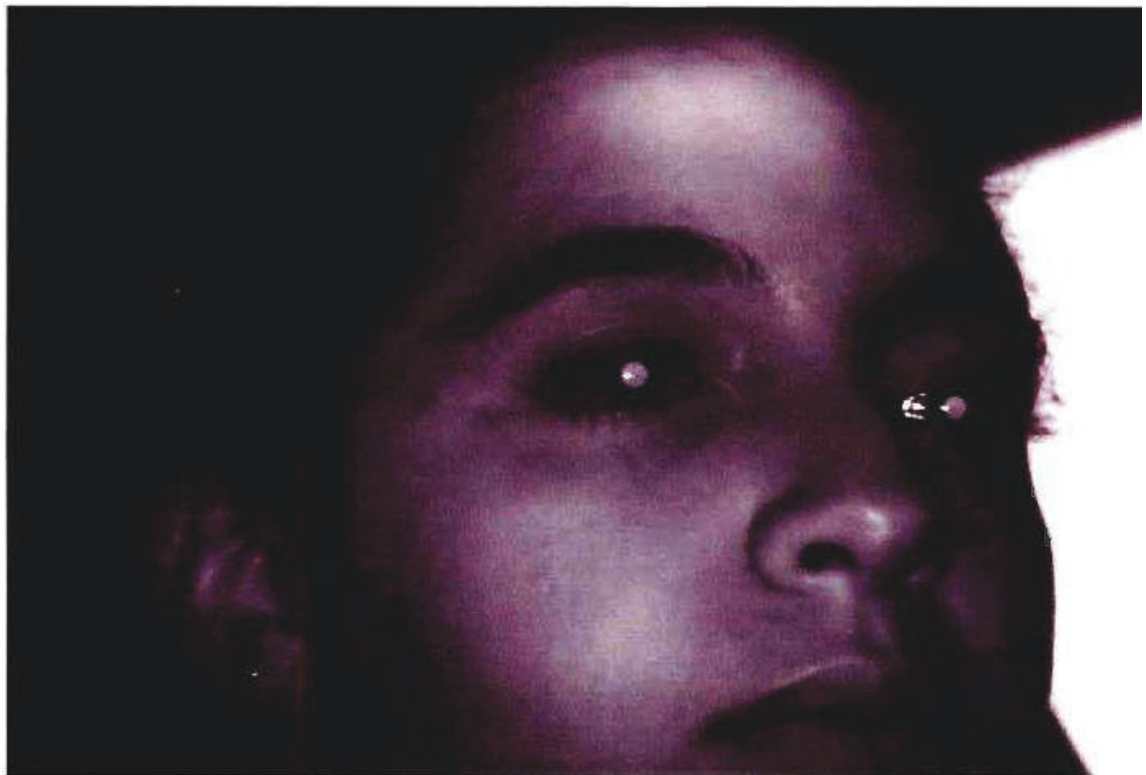


Figure 4.3 : Image illustre un exemple de conducteur en état d'éveil, servant de référence pour la comparaison avec les états de somnolence et de distraction.



Figure 4.4 : Image illustrant une conductrice présentant des signes de somnolence, caractérisés par la fermeture prolongée des paupières et une diminution de l'ouverture oculaire.



Figure 4.5 : Image illustrant une conductrice servant de référence en état d'éveil normal, caractérisés par une ouverture oculaire complète et une expression faciale stable.



Figure 4.6 : Image illustrant une conductrice en état de somnolence effectuant un bâillement.



Figure 4.7 : Image illustrant un exemple de conducteur servant de référence dont le regard est orienté vers l'avant, correspondant à une posture normale de conduite et à un état d'attention visuelle adéquat.



Figure 4.8 : Image montrant un conducteur dont le regard est détourné de l'axe frontal (état de distraction) de la route, orienté vers la gauche en direction de la caméra.

Pour assurer la détection de la somnolence et la distraction du conducteur, le système repose sur MobileNetV2, un modèle de réseau de neurones convolutifs reconnu pour son efficacité, utilisé dans cette étude comme extracteur automatique de caractéristiques visuelles (yeux, bouche et l'orientation de la tête). Ce modèle permet d'analyser efficacement les images d'un conducteur et d'estimer son niveau d'attention à partir des informations visuelles extraites. Avant leur exploitation par le réseau, les images sont redimensionnées et prétraitées grâce à la fonction `preprocess_image()`. Cette étape garantit l'homogénéité des données d'entrée et optimise les performances du modèle en assurant une compatibilité avec l'architecture de MobileNetV2.

En complément, le programme exploite MobileNetV2 pour identifier des objets susceptibles de causer des distractions, comme un téléphone portable. Si un objet de distraction est détecté avec un niveau de confiance élevé ($>50\%$), une alerte sonore est déclenchée et une capture d'image est réalisée pour informer le conducteur.

4.3.2.2 Eye_aspect_ratio(eye)

La fonction `eye_aspect_ratio(eye)` calcule le Ratio Aspectuel des Yeux (EAR), un indicateur clé permettant de détecter la somnolence à partir de l'ouverture des paupières. Une baisse soutenue de l'EAR indique une fermeture des paupières.

Si la valeur de EAR descend en dessous du seuil défini ($\text{thresh} = 0.25$ qui représente le seuil critique du Ratio Aspectuel des Yeux (EAR) en dessous duquel on considère que les yeux sont fermés) pendant plus de $\text{frame_check} = 20$ images consécutives signifient que le système vérifie la fermeture des yeux sur 20 images consécutives avant de déclencher une alerte sonore afin d'avertir le conducteur.

La figure 4.9 illustre l'architecture combinée de MobileNetV2 et du Ratio Aspectuel des Yeux (EAR), utilisée comme extracteur automatique de caractéristiques pour la détection de la somnolence et la distraction du conducteur.

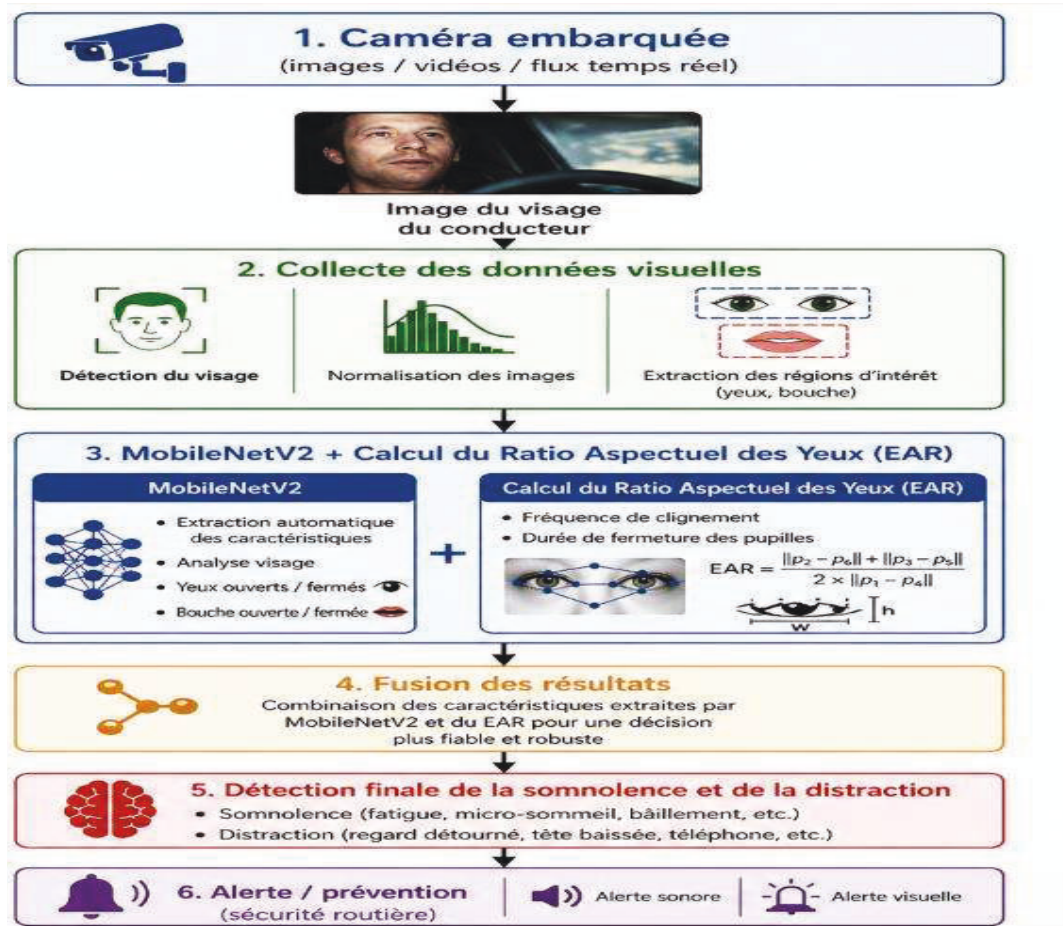


Figure 4.9 : Architectures de MobileNetV2 couplée à l'analyse du Ratio Aspectuel des Yeux (EAR) pour la détection de la somnolence et la distraction du conducteur.

Les résultats issus de MobileNetV2 et de l'analyse EAR sont ensuite fusionnés afin d'améliorer la précision et la robustesse de la détection. Cette combinaison conduit à une décision finale concernant l'état de somnolence ou de distraction du conducteur. En cas de détection d'un état critique, le système déclenche une alerte sonore et/ou visuelle, tout en sauvegardant les captures d'images correspondant à l'état de somnolence dans le fichier driver_data.csv, contribuant ainsi à la prévention des accidents et au renforcement de la sécurité routière.

4.3.3 Fonctionnalité eye_aspect_ratio(eye)

L'Eye Aspect Ratio (EAR) est un indicateur largement utilisé en vision par ordinateur pour mesurer le degré d'ouverture des yeux. Cette métrique joue un rôle essentiel dans les systèmes de surveillance de la vigilance, notamment dans les applications de détection de la somnolence au volant ou de suivi de l'attention [95].

Les figures 4.10 et 4.11 illustrent le positionnement des points de repères de référence extrait avec la fonctionnalité eye_aspect_ratio(eye).

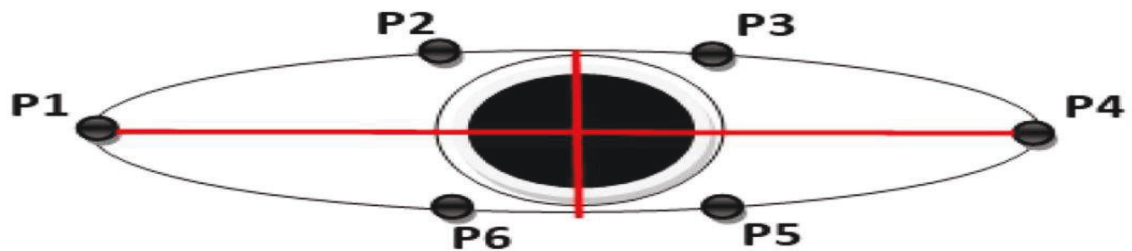
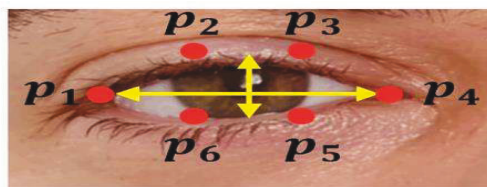
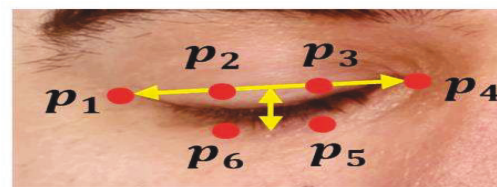


Figure 4.10 : Points de repères de référence extraits avec la fonctionnalité eye_aspect_ratio(eye) [94].



Open eye will have more EAR

(a)



Closed eye will have less EAR

(b)

Figure 4.11 : Les points de repères autour des yeux [95].

(a) Œil ouvert.

(b) Œil fermé.

Le calcul de l'EAR repose sur la position de six points de repère faciaux localisés autour de chaque œil, généralement extraits à l'aide de détecteurs de points clés faciaux. Comme illustré dans les figures 4.10 et 4.11, ces points sont notés P_1 à P_6 et correspondent aux contours supérieur, inférieur et latéraux de l'œil.

L'EAR est ensuite obtenu à partir du rapport entre ces distances, selon une formule normalisée [95] :

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

$||P_2 - P_6||$ et $||P_3 - P_5||$ représentent les distances verticales entre les paupières,

$||P_1 - P_4||$ correspond à la distance horizontale entre les coins de l'œil.

Si l'EAR est élevé, cela signifie que l'œil est ouvert, donc les distances $||P_2 - P_6||$ et $||P_3 - P_5||$ sont élevées.

Si l'EAR diminue fortement (en dessous d'un certain seuil), cela indique que l'œil est fermé ou en train de se fermer, donc les distances $||P_2 - P_6||$ et $||P_3 - P_5||$ sont faibles.

Dans notre système, un seuil critique est défini ($EAR < 0,25$) afin de caractériser l'état de fermeture des yeux. Si cette condition est maintenue sur un nombre consécutif d'images (par exemple 20 images), le conducteur est considéré comme somnolent et une alerte est déclenchée. Cette approche permet de distinguer un clignement normal d'un état de somnolence prolongé.

Utilisation des fonctions d'Alerte

- Somnolence : Si le programme détecte une fermeture prolongée des yeux, une alerte sonore est déclenchée.

- Distraction : Lorsqu'un objet de distraction est détecté, le programme affiche un message "Distraction détecté" sur l'écran, joue une alerte sonore et enregistre l'image.

- Visage Non Détecté : Si le visage du conducteur n'est pas visible ou détecté pendant une période prolongée par exemple en raison d'un mouvement brusque, d'une inclinaison

excessive de la tête ou d'une obstruction de la caméra, le programme émet également une alerte sonore.

Ajouter des conducteurs

Pour ajouter un nouveau conducteur, utilisez la fonction `add_driver(image_path, name)`
: Python

```
add_driver('r'./chemin_vers_image.png', 'Nom Conducteur')
```

Cela permet d'enregistrer l'encodage facial de ce conducteur pour l'identifier durant les sessions à partir d'une image de référence. L'extraction de l'encodage facial se fait à l'aide de la bibliothèque `face_recognition`.

Une fois l'encodage enregistré, lors des exécutions ultérieures, le programme compare en temps réel les visages capturés par la caméra avec les encodages présents dans la base. Le conducteur est ainsi reconnu automatiquement, ce qui permet d'assurer un suivi personnalisé de son état de vigilance (sommolence et distraction) et d'adapter les alertes en fonction de l'utilisateur détecté.

Gestion des données de capture

Toutes les captures d'écran sont stockées dans le dossier `captures`, tandis que les données d'événements sont enregistrées dans `driver_data.csv` pour une analyse future.

4.4 Détection de la somnolence et la distraction

La détection de la somnolence et la distraction repose sur la combinaison complémentaire du modèle `MobileNetV2` et de la fonction `eye_aspect_ratio((eye))` le paramètre `eye` représente les coordonnées des points caractéristiques (landmarks) du contour de l'œil) intégrée dans un cadre d'apprentissage profond basés sur les réseaux neurones convolutifs (CNN).

`MobileNetV2` un réseau de neurones convolutifs (CNN) léger et optimisé pour l'inférence en temps réel est utilisé pour identifier les situations de distraction, notamment en détectant la présence d'objets tels qu'un téléphone portable ou un regard détourné du conducteur (voir figure 4.8).

De son côté, la fonction `eye_aspect_ratio(EAR)` permet d'évaluer l'état de vigilance d'un conducteur en mesurant le degré d'ouverture des paupières. Une diminution significative

et prolongée de cette valeur indique une fermeture des yeux, ce qui constitue un indicateur fiable de somnolence.

Ainsi, la combinaison d'un CNN (MobileNetV2) pour l'analyse spatiale des images de visage et du EAR pour mesurer directement l'ouverture des yeux permet une détection robuste, simultanée et en temps réel de la somnolence et de la distraction, améliorant significativement la sécurité d'un conducteur.

La figure 4.12 illustre le schéma global du système de détection de la somnolence et de la distraction d'un conducteur basé sur MobileNetV2 et du Ratio Aspectuel des Yeux (EAR).

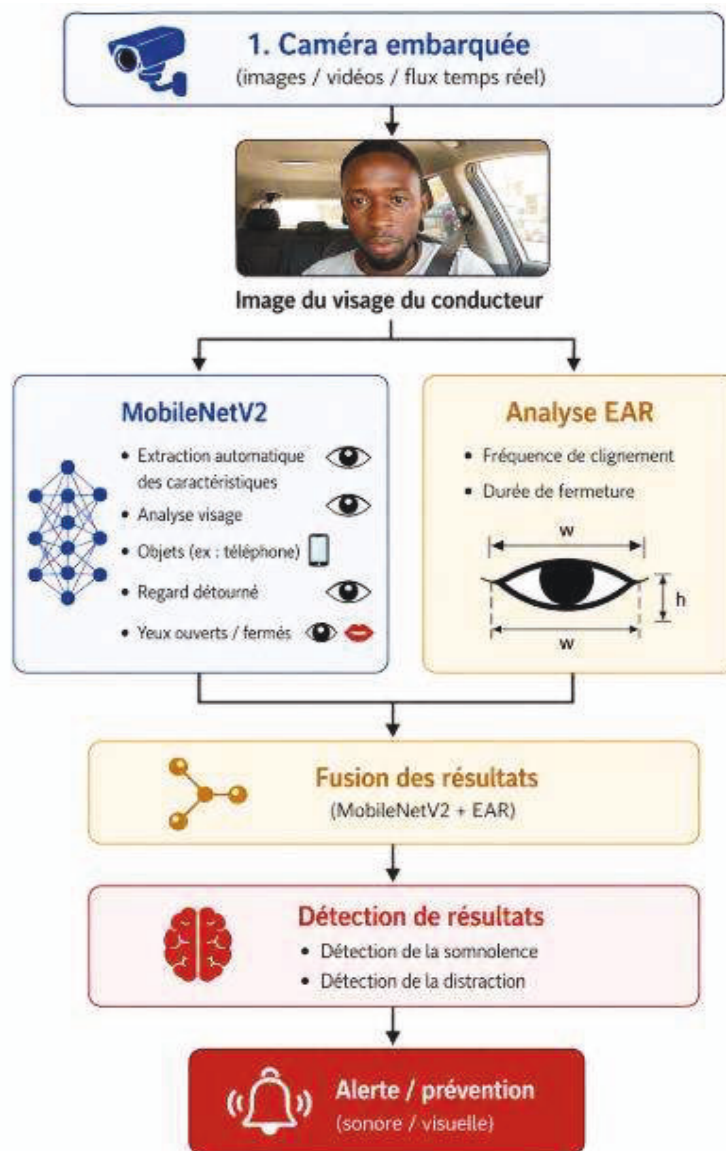


Figure 4.12 : Architecture globale du système de détection de la somnolence et de la distraction du conducteur.

4.4.1 Localisation des yeux

Dans ce présent projet, nous nous concentrons sur la détection de la somnolence et de la distraction d'un conducteur en nous basant principalement sur l'analyse de l'état des yeux. Les informations extraites de la région oculaire constituent le point de départ du pipeline : dès que les yeux sont correctement détectés, le script poursuit l'évaluation du niveau de vigilance à partir de leurs variations morphologiques.

La précision des techniques modernes de repères faciaux (facial landmarks) joue un rôle essentiel dans ce processus [59]. Grâce aux avancées de la vision par ordinateur, il est possible d'identifier avec une très grande exactitude la zone oculaire et d'en extraire les coordonnées nécessaires au calcul du Eye Aspect Ratio (EAR). Chaque œil est ainsi représenté par six points clés, permettant de mesurer sa forme, son ouverture et son évolution au cours du temps.

Cette représentation géométrique fine constitue une base fiable pour détecter : a fermeture progressive des yeux, la fréquence des clignements, les micro-fermetures caractéristiques des phases de somnolence, ou encore les mouvements oculaires associés à la distraction [95].

En utilisant ces repères faciaux, l'application bénéficie d'une estimation robuste et continue du comportement visuel d'un conducteur, ce qui permet de déclencher efficacement les étapes ultérieures de détection de la somnolence ou de la distraction.

La figure 4.13 illustre la localisation du visage et de la région oculaire d'un conducteur.

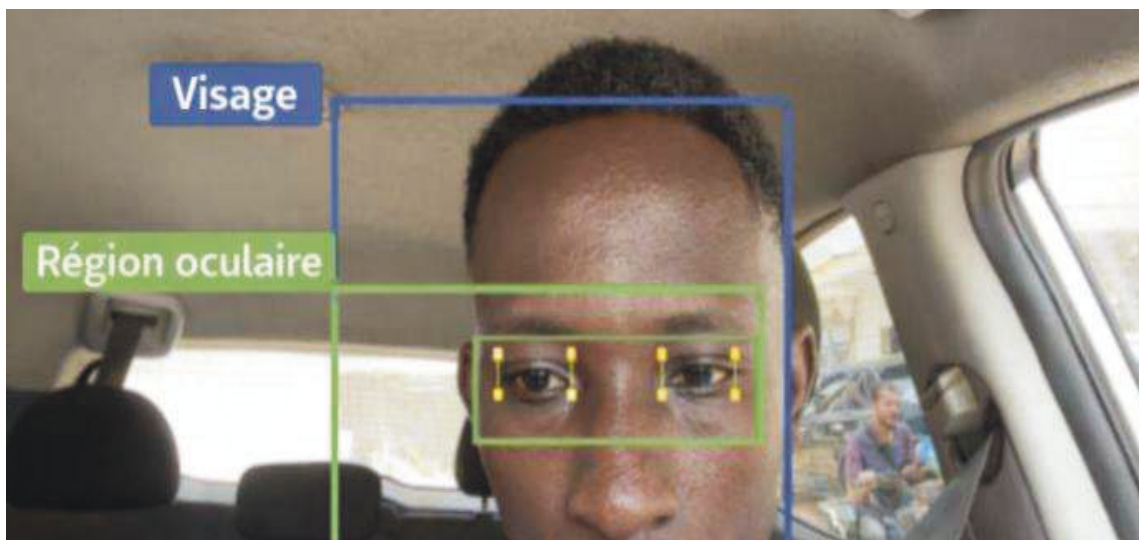


Figure 4.13 : Localisation du visage et de la région oculaire d'un conducteur.

4.4.2 Détection de la distraction

Le programme s'appuie sur MobileNetV2, un réseau de neurones convolutifs (CNN) léger et optimisé pour l'inférence en temps réel, afin d'identifier les objets associés aux comportements de distraction, tels que l'utilisation d'un téléphone portable ou la manipulation d'objets non liés à la conduite. Lorsque MobileNetV2 détecte un objet de distraction avec un niveau de confiance supérieur à 50 %, le système déclenche immédiatement une alerte sonore pour avertir le conducteur. En parallèle, une capture de l'image est enregistrée afin de conserver une trace visuelle de la situation de distraction détectée.

4.4.3 Détection de la somnolence par ratio aspectuel des yeux (EAR)

La fonction `eye_aspect_ratio(eye)` calcule le ratio aspectuel de l'œil (EAR), un paramètre clé pour évaluer l'état de somnolence d'un conducteur. Le EAR mesure l'ouverture de l'œil à partir des six points oculaires identifiés par les repères faciaux. Lorsque les yeux commencent à se fermer, la distance verticale entre les paupières diminue, ce qui fait chuter la valeur du EAR.

Dans notre système, un seuil fixe ($\text{thresh} = 0.25$) est utilisé pour déterminer si l'œil est considéré comme fermé. Si le EAR reste inférieur à ce seuil pendant plus de 20 images consécutives ($\text{frame_check} = 20$), cela indique une fermeture prolongée des yeux, caractéristique d'un épisode de somnolence ou d'un micro-sommeil. Dans ce cas, une alerte sonore est immédiatement déclenchée afin de prévenir le conducteur.

La figure 4.14 illustre le processus de détection de la somnolence par analyse de l'états des yeux.



Figure 4.14 : Situation d'ouverture et de fermeture des yeux.

Personnalisation des paramètres

- Seuil de somnolence (*thresh*) : ce paramètre permet d'ajuster la sensibilité du système à la fermeture des yeux. En modifiant la valeur de *thresh*, il est possible de rendre la détection plus stricte (valeur plus élevée) ou plus tolérante (valeur plus basse). Un seuil bien calibré permet de distinguer efficacement un simple clignement d'un signe réel de somnolence.
- Nombre de frames (*frame_check*) : ce paramètre détermine la durée pendant laquelle les yeux doivent rester fermés avant que le système ne déclenche une alerte. En augmentant *frame_check*, on réduit les faux positifs, tandis qu'en le diminuant, on rend le système plus réactif. Ce réglage contrôle donc la rapidité et la robustesse du déclenchement de l'alerte sonore.

4.5 Enregistrement des événements et captures

Les événements liés à la somnolence, à la distraction ou à l'absence de détection du visage sont automatiquement consignés dans un fichier CSV (*driver_data.csv*). Chaque enregistrement inclut un horodatage précis, permettant de retracer la chronologie des incidents et d'analyser le comportement du conducteur au cours du temps.

Parallèlement, le système sauvegarde également des captures d'écran des situations détectées dans le dossier captures. Ces images constituent une preuve visuelle des événements, facilitant l'analyse et la validation des alertes générées par le système.

La figure 4.15 illustre les captures d'événements enregistrés.

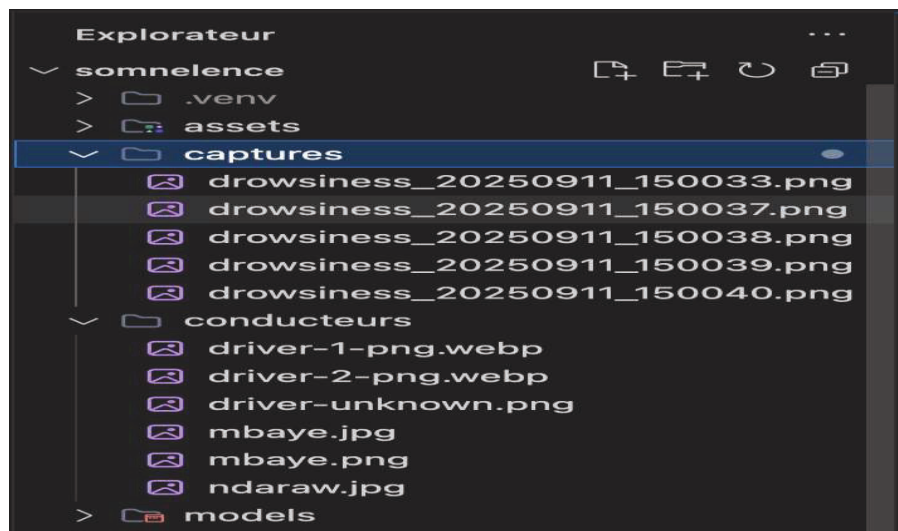


Figure 4.15 : Captures d'événements enregistrés.

4.6 Reconnaissance faciale

Le programme utilise la bibliothèque `face_recognition` pour identifier automatiquement le conducteur. Lors de l'ajout d'un nouveau conducteur, une image de référence de son visage est importée, puis un encodage facial est généré à partir des caractéristiques biométriques extraites. Cet encodage est ensuite stocké et associé à l'identité du conducteur.

Pendant l'exécution, la fonction `identify_driver()` analyse en continu les visages détectés dans le flux vidéo. Elle compare chaque encodage facial issu de la caméra aux encodages des visages connus à l'aide d'un algorithme de distance (basé sur un seuil). Si une correspondance est trouvée, le conducteur est identifié ; dans le cas contraire, le système signale un visage inconnu ou une absence de correspondance.

La figure 4.16 illustre une liste des visages des conducteurs en somnolence détectés.



```
Nombre de visages detectes : Mbaye DIOP -> 0
Aucun visage trouve dans l'image.
Nombre de visages detectes : Ndaraw Seck -> 1
Nombre de visages detectes : Driver 2 -> 0
Aucun visage trouve dans l'image.
Nombre de visages detectes : Driver 3 -> 0
Aucun visage trouve dans l'image.
```

Figure 4.16 : *Conducteurs en somnolence*

4.7 Conclusion

Dans ce chapitre, nous avons présenté en détail la conception de l'architecture de notre système de détection de la somnolence et de la distraction au volant. Cette architecture repose sur une combinaison complémentaire de techniques classiques de vision par ordinateur et de modèles avancés de deep learning, permettant d'exploiter à la fois l'information spatiale et temporelle du comportement facial d'un conducteur.

D'une part, l'analyse des images faciales est assurée par MobileNetV2, un réseau de neurones convolutifs (CNN) optimisé pour une exécution en temps réel. Ce modèle extrait des

caractéristiques discriminantes du visage, nécessaires à la détection de comportements de distraction, tels que la manipulation d'un téléphone portable ou l'orientation du regard hors de la route.

D'autre part, la mesure du ratio aspectuel des yeux (EAR) constitue un outil robuste pour évaluer la fermeture des paupières. Cette approche classique de vision par ordinateur, fondée sur les repères faciaux, permet de détecter les baisses de vigilance liées à une fermeture prolongée des yeux, indicateur direct de somnolence.

Cette combinaison complémentaire de MobileNetV2 et du EAR confère au système une capacité d'analyse complète. L'intégration de ces approches garantit une détection fiable, précise et réactive des états de somnolence et de distraction, même dans des conditions d'éclairage ou de mouvement variées.

Dans le chapitre suivant, nous présenterons l'implémentation pratique du système de détection de somnolence et distraction, en détaillant les outils logiciels, les bibliothèques utilisées (face_recognition, TensorFlow, dlib, OpenCV, etc.), ainsi que les étapes de développement ayant permis de transformer cette architecture théorique en une solution entièrement fonctionnelle et d'en évaluer les performances.

Chapitre 5

Implémentation et résultats

5.1 Introduction

Dans ce chapitre, nous présentons les différentes étapes de la mise en œuvre de notre système de détection de la somnolence et de la distraction au volant. L'objectif principal est de traduire l'architecture théorique développée au chapitre précédent en une solution logicielle fonctionnelle, capable d'opérer en temps réel et dans des conditions proches de la conduite réelle. Nous commençons par décrire le langage de programmation choisi, ainsi que les outils, bibliothèques et environnements utilisés pour le développement de l'application.

Nous détaillons ensuite les algorithmes intégrés, notamment ceux issus de la vision par ordinateur et de l'apprentissage profond. En particulier, les réseaux de neurones convolutifs (CNN), tels que MobileNetV2 sont exploités pour l'analyse spatiale des images et de la détection des comportements de distraction, tandis que le ratio aspectuel des yeux (EAR) est utilisé pour analyser les variations temporelles de l'ouverture des paupières et évaluer l'états de vigilance du conducteur.

Enfin, nous analysons les résultats obtenus dans divers contextes d'utilisation afin d'évaluer la robustesse, la précision et les limites de notre solution. Cette analyse permet d'apprécier la performance globale du système et d'identifier les pistes d'améliorations possibles.

5.2 Les outils

Dans cette section, nous présentons et décrivons les différents outils utilisés pour la mise en œuvre de notre système de détection de la somnolence et de la distraction au volant. Ces outils incluent à la fois les environnements logiciels, les bibliothèques de vision par ordinateur, ainsi que les frameworks de deep learning nécessaires au traitement des images et à l'analyse des séquences temporelles.

5.2.1 Le matériel

Nous avons généré l'ensemble de nos résultats sur une machine Asus disposant des caractéristiques matérielles suivantes :

- Modèle : VivoBook_ASUSLaptop TP412FAC_TP412FA
- Processeur : Intel(R) Core(TM) i3-10110U CPU @ 2.10GHz (4 CPUs), ~2.6GHz
- Mémoire : 8 Go
- Disque dur : 256 Go (SSD)
- Système d'exploitation : Windows11 64 bits

5.2.2 Environnements et outils de développement

Nous avons utilisé le langage de programmation python, nous avons également utilisé différentes bibliothèques et des outils pour permettre de réaliser notre programme.

5.2.2.1 Python

Python est un langage de programmation interprété de haut niveau, polyvalent et facile à apprendre. Il prend en charge les structures de données de haut niveau, une approche simple mais efficace de la programmation orientée objet, et une syntaxe élégante qui permet aux programmeurs de s'exprimer de manière claire et concise [\[96\]](#).

Pour un projet aussi complexe que la détection de la somnolence et la distraction, où il est nécessaire d'interagir avec des périphériques matériels comme un ordinateur portable Asus, Python offre une interface claire et efficace pour la gestion des données et des processus.

➤ Les bibliothèques utilisées :

Un des principaux avantages de Python est la disponibilité de bibliothèques puissantes adaptées à divers domaines, notamment le traitement d'images, l'intelligence artificielle et la gestion des données.

• OpenCV

OpenCV (Open Source Computer Vision Library) est une bibliothèque logicielle open source de vision par ordinateur et d'apprentissage automatique. OpenCV a été conçu pour fournir une infrastructure commune aux applications de vision par ordinateur et pour accélérer l'utilisation de la perception automatique dans les produits commerciaux. Dans le cadre de notre projet, OpenCV occupe une place centrale. Elle est utilisée pour : l'acquisition d'images à partir de la caméra en temps réel, le prétraitement des images (conversion, redimensionnement,

filtrage), la détection et l'extraction des régions d'intérêt, notamment la zone du visage et des yeux, l'intégration avec d'autres modules de deep learning (CNN, RNN) via les interfaces de traitement des images [\[60\]](#).

- **dlib**

Utilisée principalement pour la détection des points de repère du visage (facial landmarks), la bibliothèque dlib joue un rôle essentiel dans notre système. Grâce à son modèle de détection basé sur des algorithmes d'apprentissage statistique, dlib permet d'identifier avec précision les points clés du visage, notamment autour des yeux, du nez, de la bouche et de la mâchoire [\[62\]](#).

Cette précision est cruciale pour l'analyse des expressions faciales, du clignement des yeux et des mouvements de la tête, éléments indispensables à la détection de la somnolence et de la distraction. Les coordonnées des repères faciaux extraits par dlib permettent ainsi de calculer le Eye Aspect Ratio (EAR) et de suivre les variations géométriques du visage au fil du temps.

- **Numpy**

Pour gérer et manipuler les matrices et tableaux d'images, Numpy a été utilisé. Il permet de traiter les données efficacement, notamment dans les étapes de prétraitement des images capturées avant l'analyse par les algorithmes de détection [\[61\]](#).

Dans notre système, NumPy intervient principalement lors des étapes de prétraitement des images, telles que : la normalisation et la mise à l'échelle des pixels, la transformation des images en matrices exploitables par les algorithmes de détection, le traitement rapide des flux d'images en temps réel, l'intégration fluide avec OpenCV et les modèles de deep learning.

Cette efficacité computationnelle est indispensable pour maintenir les performances, notamment lorsque les images doivent être traitées avant d'être analysées par les modèles de détection basés sur des CNN.

- **TensorFlow**

TensorFlow est un framework de programmation pour le calcul numérique qui a été rendu Open Source par Google en novembre 2015.

Depuis sa publication, TensorFlow n'a cessé de gagner en popularité, pour devenir très rapidement l'un des Framework les plus utilisés dans le domaine de l'apprentissage profond. Son nom est inspiré du fait que les opérations courantes sur des réseaux de neurones sont faites

sur des tableaux multi-dimensionnelles, appelées Tenseurs. Un tenseur à deux dimensions est l'équivalent d'une matrice. Aujourd'hui, les principaux produits de Google utilisent TensorFlow : Gmail, Google Photos et reconnaissance de la parole [63].

Dans notre projet, TensorFlow joue un rôle central pour : l'entraînement et le déploiement du modèle MobileNetV2 (CNN).

➤ **Application sur un ordinateur portable Asus :**

Python est parfaitement compatible avec les ordinateurs tels que Asus. Nous avons pu exécuter notre système grâce à l'efficacité de Python et de ses bibliothèques. La gestion des périphériques a été facilitée par des modules Python, qui permettent une interaction directe avec le matériel.

5.3 Instructions d'utilisation

Le programme utilise la caméra de l'ordinateur pour capturer des images en temps réel et alerter le conducteur en cas de signes de somnolence ou de distraction.

L'implémentation des composantes du système de détection de la somnolence et la distraction au volant

Dans cette section, nous allons décrire les deux parties de notre système, à savoir la partie de la détection de la somnolence et la partie de la détection de la distraction au volant. Ces deux parties reposent sur des approches complémentaires combinant vision par ordinateur et modèles avancés de deep learning. La première exploite principalement l'analyse géométrique des yeux et l'évolution temporelle de leur ouverture, tandis que la seconde s'appuie sur un réseau de neurones convolutifs (CNN) pour identifier les objets ou comportements indicateurs de distraction.

La création du modèle MobileNetV2 et Ratio Aspectuel des Yeux (EAR)

Comme nous avons expliqué dans le chapitre 4, nous devons obtenir un modèle qui permet la détection de la somnolence et de la distraction au volant. En utilisant le langage Python avec l'environnement Tensorflow et des bibliothèques bien connues dans le domaine du Deep learning et vision par ordinateur.

Entraînement préalable

Dans notre approche, nous nous appuyons sur des modèles ayant bénéficié d'un entraînement préalable. Le modèle MobileNetV2 peut, par exemple, être pré-entraîné sur la

base de données ImageNet, qui contient environ 1,2 million d'images réparties en 1 000 classes. Ce pré-entraînement permet au réseau d'apprendre des caractéristiques visuelles générales, telles que les contours, les formes, les textures et certains motifs visuels.

Dans le cadre de notre système, la couche de sortie initiale de MobileNetV2 est adaptée afin de répondre à notre problématique spécifique. Au lieu de prédire les 1 000 classes d'ImageNet, le modèle est ajusté pour classer les états du conducteur selon les classes étudiées, notamment l'état normal, la somnolence et la distraction. Ainsi, la couche de sortie peut comporter trois neurones correspondant respectivement aux classes : normal, somnolence et distraction.

Cette adaptation permet de réutiliser les caractéristiques apprises lors de l'entraînement préalable tout en spécialisant le modèle pour la détection des comportements à risque au volant. Dans ce contexte, la somnolence est associée à des signes tels que la fermeture prolongée des yeux, le micro-sommeil ou le bâillement, tandis que la distraction est liée à des comportements comme le regard détourné, la tête baissée ou l'utilisation du téléphone portable.

Structure générale du programme

L'application est structurée en plusieurs sections principales :

1. Importation des bibliothèques et des modules
2. Configuration des paramètres de détection
3. Définition des fonctions principales
4. Boucle principale d'exécution
5. Enregistrement et gestion des événements

Importation des bibliothèques et modules

La première partie du code consiste en l'importation des bibliothèques essentielles au fonctionnement du programme comme le montre la figure 5.1.


```

import cv2
import dlib
import numpy as np
import pygame
import mediapipe as mp
import tensorflow as tf
from scipy.spatial import distance as dist
from imutils import face_utils
from datetime import datetime
import csv
import os
import face_recognition

```

Figure 5.1 : Importation des bibliothèques.

- cv2 : Utilisé pour la manipulation d'images et de vidéos.
- dlib : Utilisé pour la détection des points de repère du visage (ex. yeux, bouche).
- media pipe et tensorflow : Utilisés pour la détection d'objets (ex. téléphone).
- pygame : Utilisé pour jouer les sons d'alerte.
- face_recognition : Pour l'identification du conducteur en utilisant la reconnaissance faciale.

Initialisation des paramètres de détection

Des variables globales et paramètres de configuration sont définis au début du code pour contrôler les seuils et les constantes comme le montre la figure 5.2.

```

thresh = 0.25
frame_check = 20
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

```

Figure 5.2 : Définition des variables et paramètres.

- thresh = 0.25 : seuil utilisé pour déterminer lorsque le EAR indique une fermeture des yeux.
- frame_check = 20 : nombre d'images consécutives pendant lesquelles le EAR doit rester sous le seuil pour déclencher une alerte.
- detector : Détecteur de visage de dlib.
- predictor : Prédicteur pour obtenir les points de repère des yeux et de la bouche.

Fonctions Principales

Fonction de Prétraitement

Cette fonction redimensionne et prétraite l'image pour la passer au modèle MobileNetV2, utilisé pour la détection d'objets (distractions). La figure 5.3 présente le code permettant le prétraitement des images.

```
def preprocess_image(image):  
    image = tf.image.resize(image, (224, 224))  
    image = tf.keras.applications.mobilenet_v2.preprocess_input(image)  
    return image
```

Figure 5.3 : Fonction de prétraitement.

Redimensionnement de l'image à 224×224 pixels : MobileNetV2 impose une taille d'entrée fixe. Le redimensionnement garantit que chaque image respecte le format attendu par le réseau de neurones convolutif (CNN).

Détection de somnolence : Fonction `eye_aspect_ratio ()`

La fonction calcule le Eye Aspect Ratio (EAR). La figure 5.4 présente le code permettant de calculer le EAR.

```
def eye_aspect_ratio(eye):  
    A = dist.euclidean(eye[1], eye[5])  
    B = dist.euclidean(eye[2], eye[4])  
    C = dist.euclidean(eye[0], eye[3])  
    ear = (A + B) / (2.0 * C)  
    return ear
```

Figure 5.4 : Fonction `eye_aspect_ratio ()`.

- A et B représentent les distances verticales entre les paires de points de l'œil.
- C correspond à la distance horizontale entre les coins externes de l'œil.
- Le EAR est ensuite calculé comme la moyenne des distances verticales normalisée par la distance horizontale.
- Si ce ratio est inférieur au seuil prédéfini (0.25) et maintenue sur plusieurs images consécutives, le programme détecte une fermeture prolongée des yeux, signe de somnolence.

- Les points 0, 1, 2, 3, 4 et 5 représentent les 6 points de repère (landmarks) autour de l'œil :

0 : coin externe de l'œil.

3 : coin interne de l'œil (ou inverse selon œil gauche/droit).

1 & 2 : points de la paupière supérieure.

4 & 5 : points de la paupière inférieure.

Fonction d'identification du conducteur

Pour permettre au système de reconnaître automatiquement le conducteur, nous utilisons la bibliothèque `face_recognition`, basée sur des modèles avancés d'apprentissage profond (CNN) pour l'extraction d'encodages faciaux.

La figure 5.5 montre le code utilisé.

```
known_encodings = []
known_names = []

def add_driver(image_path, name):
    image = face_recognition.load_image_file(image_path)
    encoding = face_recognition.face_encodings(image)[0]
    known_encodings.append(encoding)
    known_names.append(name)
```

Figure 5.5 : *Fonction d'identification du conducteur.*

- Cette fonction ajoute un conducteur en chargeant une image et en extrayant un encodage facial.
- Chaque conducteur est identifié par son encodage et un nom pour faciliter le suivi des alertes spécifiques.

Ajout de conducteurs

Pour ajouter un nouveau conducteur, utilisez la fonction `add_driver (image_path, name)` :

Python

```
add_driver(r'./chemin_vers_image.png', 'Nom Conducteur')
```

Cela permet d'enregistrer l'encodage facial à l'aide de la bibliothèque `face_recognition`, puis enregistre cet encodage ainsi que le nom associé dans la liste des conducteurs connus.

Détection de distraction : Fonction `detect_distraction()`

La figure 5.6 montre le code utilisé.

```
def detect_distraction(frame):  
    input_tensor = tf.convert_to_tensor(frame)  
    input_tensor = tf.expand_dims(input_tensor, 0)  
    detections = object_detection_model(input_tensor)  
    return detections
```

Figure 5.6 : Fonction `detect_distraction()`.

- TensorFlow et MobileNetV2 sont utilisés pour détecter les objets potentiellement distrayants, comme le téléphone, Interactions avec d'autres passagers, Manipuler un objet sur le siège passager.
- Si un objet de distraction comme le téléphone, se retourner vers l'arrière est détecté avec une probabilité élevée, une alerte est déclenchée.

Boucle principale d'exécution

La boucle principale s'exécute en continu pour capturer les images de la webcam, traiter les données et déclencher les alertes en fonction de l'état détecté. La figure 5.7 montre le code utilisé.

```
while True:  
    ret, frame = cap.read()  
    if not ret:  
        break  
  
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)  
    rects = detector(gray, 0)  
  
    for rect in rects:  
        shape = predictor(gray, rect)  
        shape = face_utils.shape_to_np(shape)  
  
        leftEye = shape[lStart:lEnd]  
        rightEye = shape[rStart:rEnd]  
  
        leftEAR = eye_aspect_ratio(leftEye)  
        rightEAR = eye_aspect_ratio(rightEye)  
  
        ear = (leftEAR + rightEAR) / 2.0
```

Figure 5.7 : Boucle principale d'exécution.

- Capture d'images : Capture continue des frames de la caméra.
- Conversion en niveaux de gris : Améliore la précision de la détection de visage.
- Détection de visage et des yeux : Analyse chaque visage détecté et calcule le EAR des yeux.

Détection et alerte de somnolence

La figure 5.8 montre le code utilisé.

```
if ear < thresh:
    COUNTER += 1
    if COUNTER >= frame_check:
        cv2.putText(frame, "SOMNOLENCE!", ...)
        play_alert_sound()
else:
    COUNTER = 0
```

Figure 5.8 : Détection et alerte de somnolence.

- Si le EAR est en dessous du seuil pour frame_check images consécutives comme 20 images consécutives, une alerte de somnolence est émise.

Alerte de distraction

La figure 5.9 montre le code utilisé.

```
detections = detect_distraction(frame)
for detection in detections:
    if detection['score'] > 0.5 and detection['label'] == "phone":
        cv2.putText(frame, "DISTRACTION!", ...)
        play_alert_sound()
        save_capture(frame)
```

Figure 5.9 : Alerte de distraction.

- Si un objet de distraction est détecté, une alerte visuelle et sonore est émise, et l'image est enregistrée.

Identification de conducteur et gestion des données

Identification de conducteur

La figure 5.10 montre le code utilisé.

```
rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
face_encodings = face_recognition.face_encodings(rgb_frame)

for face_encoding in face_encodings:
    matches = face_recognition.compare_faces(known_encodings, face_encoding)
    name = "Unknown"

    if True in matches:
        first_match_index = matches.index(True)
        name = known_names[first_match_index]

    record_event(name, event_type, frame)
```

Figure 5.10 : Identification de conducteur.

- Le conducteur est identifié dans chaque frame. Si son visage est inconnu, une alerte est émise.
- Les événements sont enregistrés avec un horodatage dans un fichier CSV.

Gestion des données de capture

Fichier CSV

Tous les événements sont enregistrés dans `driver_data.csv` avec des informations comme la date, l'heure, le type d'événement et le conducteur.

Captures d'écran

Les captures de distraction sont enregistrées dans le dossier captures avec des noms de fichiers basés sur la date et l'heure.

Exécution du programme

Pour lancer le programme, exécutez la commande suivante dans un terminal : `bash`

Copier le code : `python nom_du_programme.py`

Résultats obtenus à la détection de la somnolence et la distraction

Les captures obtenues nous ont permis d'extraire plusieurs indicateurs essentiels liés à l'état de vigilance du conducteur. En particulier, le Eye Aspect Ratio (EAR) a permis de mesurer avec précision la durée de fermeture des yeux (eye closure) et d'identifier les fermetures prolongées caractéristiques d'un épisode de somnolence.

Par ailleurs, l'analyse de l'ouverture de la bouche (mouth open) a permis de détecter les bâillements (yawn detected), souvent associés à un état de fatigue avancé. L'ensemble de ces informations EAR, eye closure, mouth open et bâillements constitue un ensemble de signaux précieux pour évaluer le degré de vigilance du conducteur et déclencher, lorsque nécessaire, des mesures d'alerte afin d'assurer sa sécurité.

Présentation des performances obtenues pour la somnolence

On distingue deux situations, elles sont décrites dans ce qui suit.

Sujet en état d'éveil

En se basant sur les expérimentations qui ont été faites, les résultats obtenus avec une séquence d'images nous montrent un sujet qui est dans un état d'éveil normal sous deux formes :

Sujet avec contour

Capture de l'image d'un sujet sur laquelle nous avons appliqué des contours de détection pour le visage, la bouche et les yeux. Le conducteur est dans un état d'éveil normal regardant vers l'avant lors de la conduite.

La figure 5.11 montre un conducteur qui est dans un état d'éveil normal.

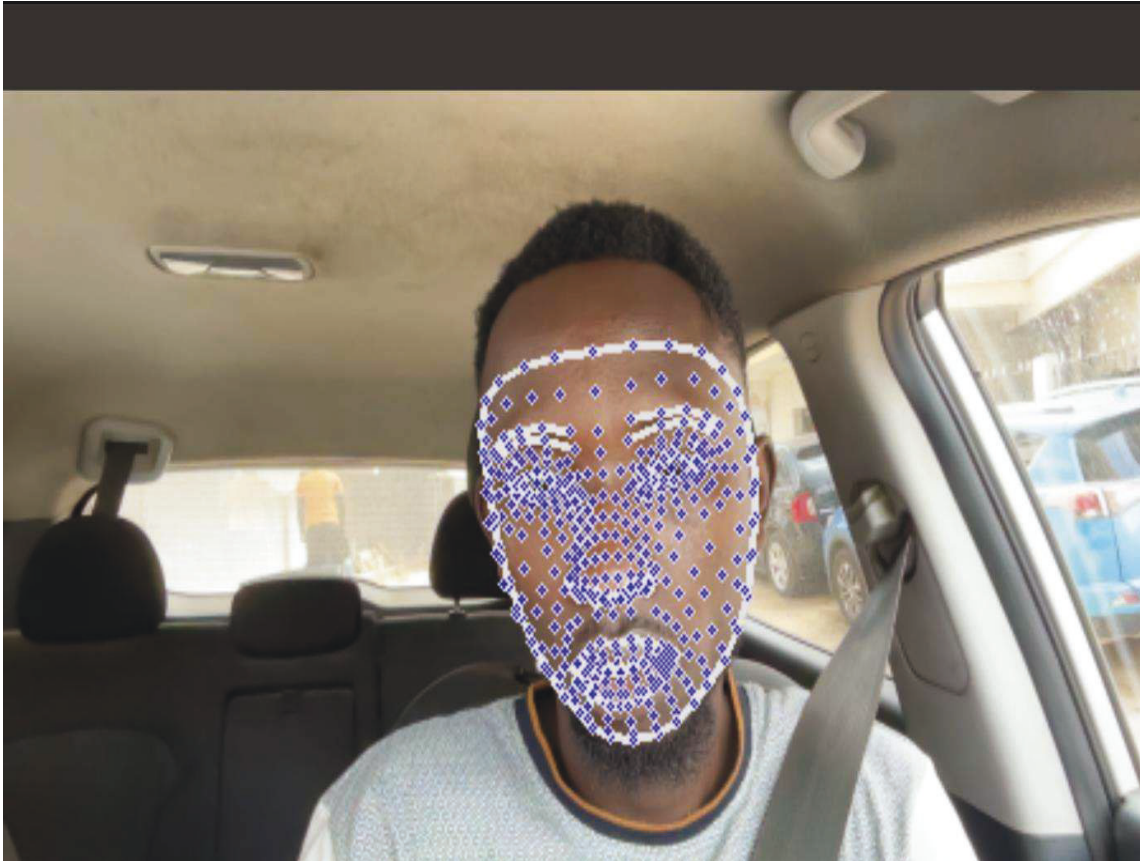


Figure 5.11 : Image montrant le conducteur dans un état d'éveil normal avec contour.

Dans cette première forme, l'algorithme détecte correctement l'ensemble des points de repère du visage (*facial landmarks*), comme illustré dans l'image 5.11.

Ces points, répartis sur les yeux, les sourcils, le nez, la bouche et les contours du visage, permettent : de suivre précisément les micro-variations faciales, de calculer le EAR (Eye Aspect Ratio) permettant d'évaluer l'ouverture des yeux, d'analyser l'expression générale du visage, et de surveiller l'orientation de la tête.

Le nombre élevé de points détectés et leur stabilité indiquent que le sujet est bien éveillé, avec une ouverture oculaire normale et une absence de signes de somnolence (pas de baisse du EAR, pas de fermeture progressive des yeux).

Sujet sans contour

Capture de l'image d'un sujet sur laquelle aucun contour de détection pour le visage, la bouche et les yeux n'est appliqué. Le conducteur est dans un état d'éveil normal regardant vers l'avant lors de la conduite.

La figure 5.12 montre le conducteur dans un état d'éveil normal (sans contour).

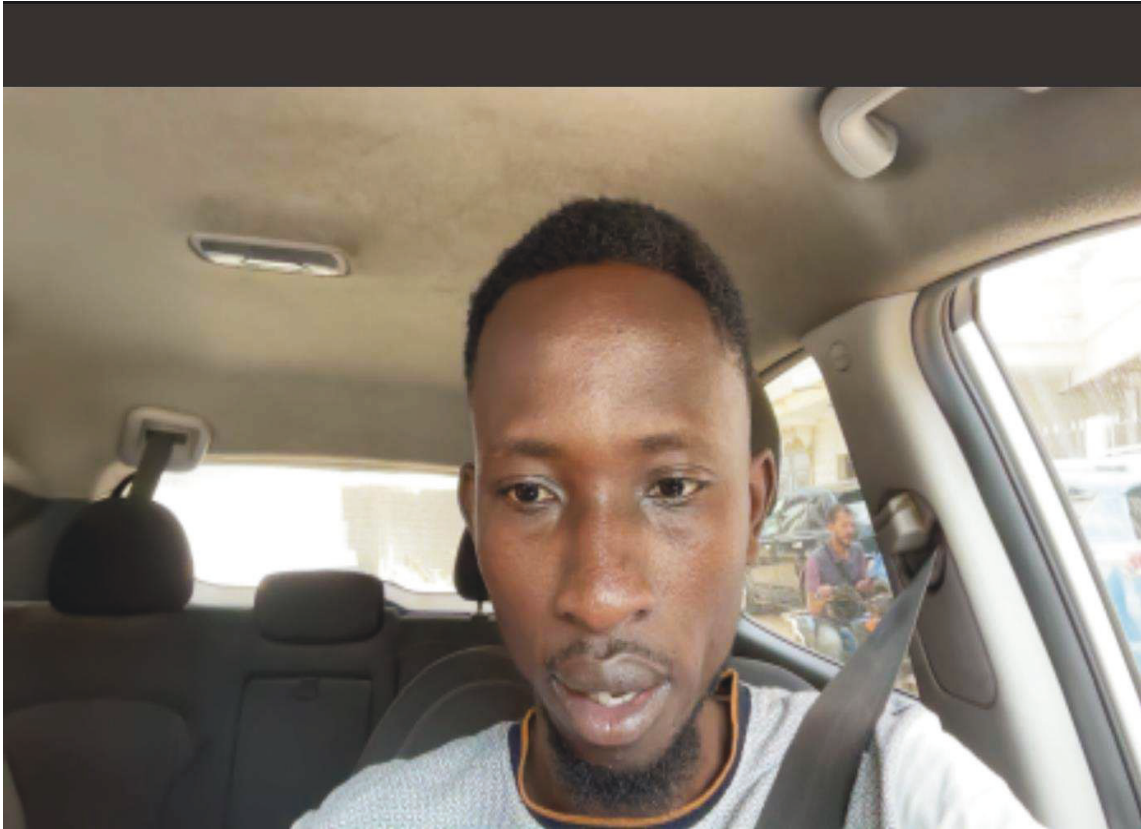


Figure 5.12 : Images montrant le conducteur dans un état d'éveil normal sans contour.

Contrairement au cas précédent où les landmarks sont affichés pour illustrer la détection du visage, ici l'image présente uniquement le visage naturel du sujet, sans annotation.

Cette capture illustre un cas où le conducteur est pleinement éveillé, et où le système détecte correctement la situation sans afficher les contours ou repères faciaux.

Sujet en état de somnolence

L'objectif de cette section est de présenter les résultats obtenus à partir du test de notre modèle sur un sujet placé dans une situation simulée de somnolence.

Les observations ont été réalisées selon deux approches complémentaires :

Image montrant un candidat en état de somnolence avec contour

La figure 5.13 illustre un conducteur en état de somnolence, détecté par le système, avec l’affichage des contours faciaux. Cette représentation met clairement en évidence une fermeture complète des yeux.

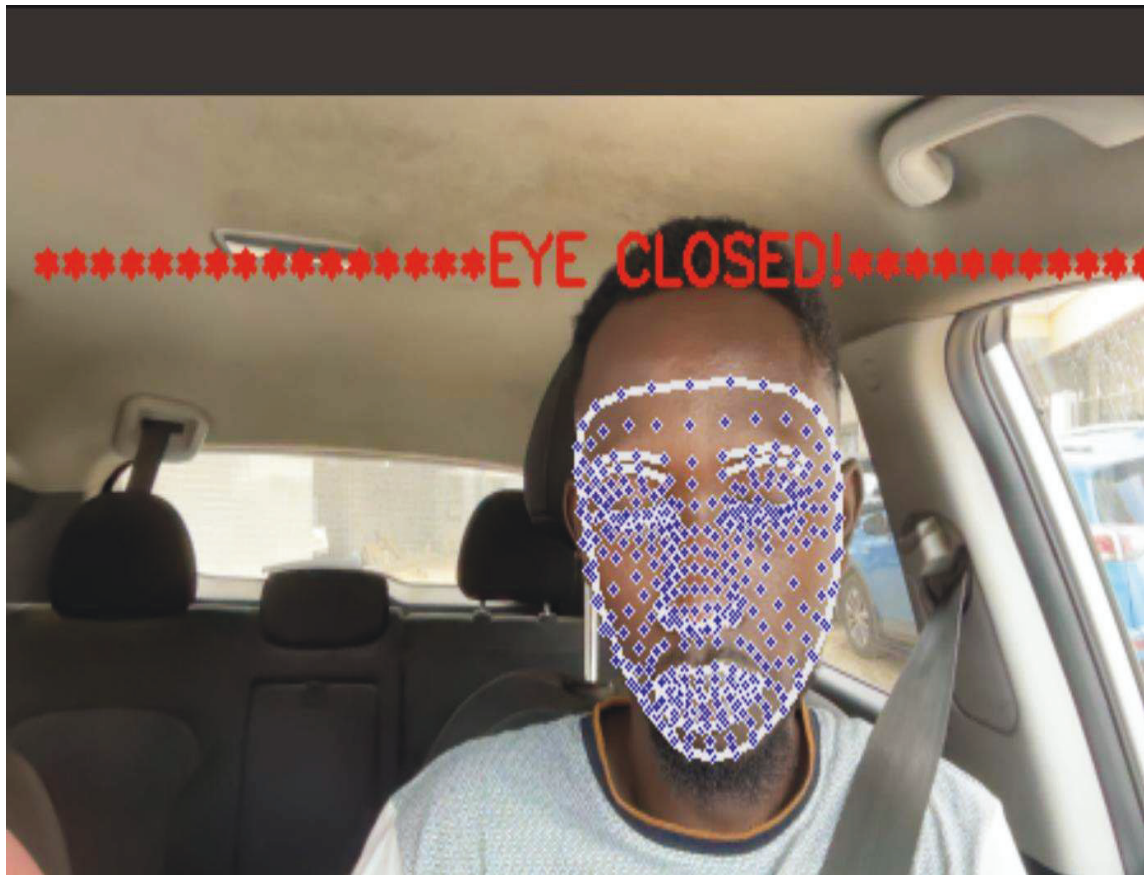


Figure 5.13 : Images montrant le conducteur en état de somnolence avec contour.

Cette image nous permet d’observer que le conducteur a fermé les yeux (eye closure) avec contour grâce à l’analyse des repères faciaux (*facial landmarks*) donc on peut en conclure que le conducteur est en état de somnolence car $>$ au seuil prédéfini (0.25) pendant 20 frames de fermeture.

Le message “*EYE CLOSED!*” affiché par le système indique que le seuil de somnolence a été atteint.

Image montrant un candidat en état de somnolence sans contour

La figure 5.14 illustre un conducteur en état de somnolence, détecté par le système, sans affichage des contours faciaux. Cette représentation met clairement en évidence une fermeture complète des yeux.

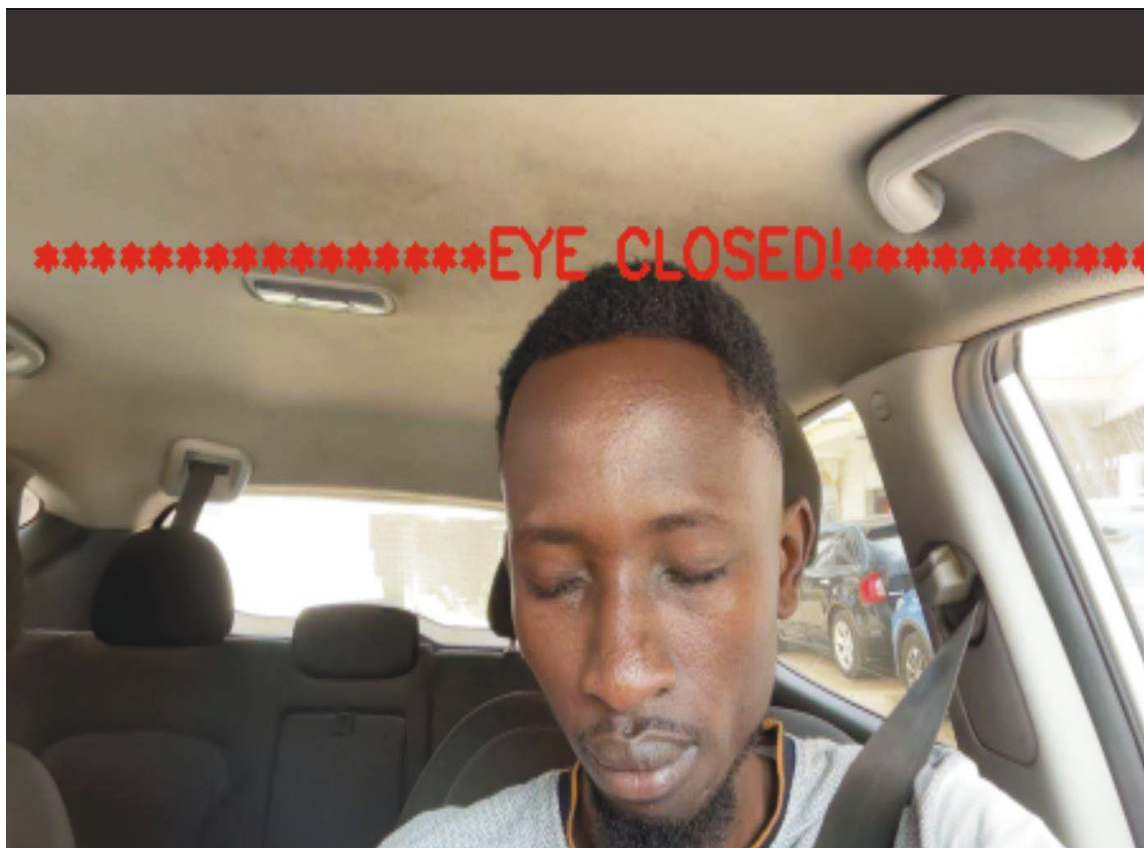


Figure 5.14 : Images montrant le conducteur en état de somnolence sans contour.

Cette image nous permet d’observer que le conducteur a fermé les yeux (eye closure) sans affichage des repères faciaux donc on peut en conclure que le conducteur est en état de somnolence car $>$ au seuil prédéfini (0.25) pendant 20 frames de fermeture.

Malgré l’absence de contours visibles, le système a correctement détecté la fermeture prolongée des yeux, comme indiqué par le message “*EYE CLOSED!*”.

L’absence d’annotation graphique permet de visualiser le comportement réel du conducteur tel qu’il apparaît dans une situation normale, tout en confirmant que la détection reste fiable même sans afficher les repères faciaux.

Image montrant un candidat effectuant un bâillement

La figure 5.15 illustre un conducteur en train d’effectuer un bâillement au cours de la conduite. L’image est extraite des captures générées par notre système et enregistrées dans le fichier driver_data.csv lors de l’analyse des différents états d’un conducteur.



Figure 5.15 : Images montrant le conducteur effectuant un bâillement.

On observe une ouverture importante de la bouche, accompagnée d'une fermeture complète des yeux, deux signes généralement associés à une baisse de vigilance.

Ce comportement, visible lors des phases de fatigue, est pris en compte par notre système.

Présentation des performances obtenues pour la distraction

Ici on présente les différents résultats obtenus pour la distraction sur un conducteur.

Sujet en état d'éveil

La figure 5.16 montre un sujet en état d'éveil, image utilisée comme référence pour l'analyse des comportements de distraction.



Figure 5.16 : Image montrant le conducteur dans un état d'éveil.

Cette image montre un conducteur attentif, avec les yeux ouverts et une posture normale, sans signe particulier de somnolence ni d'orientation détournée de la tête souvent associée à des distractions

Image montrant un candidat distrait détecté par notre application

La figure 5.17 montre un cas de distraction détectée automatiquement par notre système.

Cette image est extraite des captures générées par notre système lors de l'analyse des différents états du conducteur.

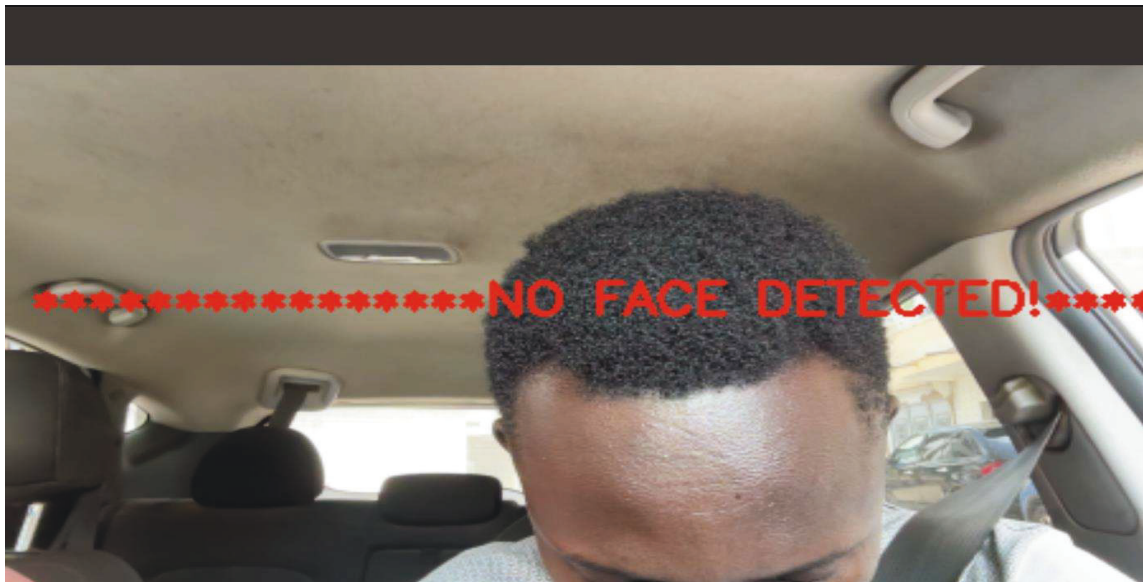


Figure 5.17 : Images montrant un cas de distraction avec comme message "NO FACE DETECTED!" détectée automatiquement par le système lors des expérimentations.

Ici, le conducteur adopte une posture qui empêche la caméra de capturer correctement son visage : la tête est inclinée vers le bas, entraînant une absence totale de détection faciale.

Le message “NO FACE DETECTED!” affiché à l’écran indique que : aucun visage n’a été reconnu dans le champ de la caméra, les repères faciaux (landmarks) nécessaires au calcul du EAR et au suivi du regard ne peuvent pas être extraits, et par conséquent, le système considère cet état comme une situation de distraction potentielle.

En pratique, ce type de posture peut être associé à des actions telles que : l’utilisation d’un téléphone portable, l’utilisation d’un téléphone portable, la consultation d’un objet placé sur les genoux,

Cette situation montre la capacité de notre modèle à détecter non seulement les distractions liées à des objets par MobileNetV2, mais aussi celles liées à la façon de se tenir ou à la perte du champ de vue.

Image montrant un candidat distrait par son téléphone

La figure 5.18 montre un cas de distraction au volant causé par l’utilisation d’un téléphone portable. L’image est extraite des captures générées par notre système et enregistrées dans le fichier driver_data.csv lors de l’analyse des différents états d’un conducteur.



Figure 5.18 : Image montrant un conducteur distrait par l'utilisation d'un téléphone portable.

On y observe un conducteur dont l'attention est entièrement dirigée vers l'écran du téléphone, entraînant un détournement du regard et une perte de vigilance visuelle.

Ce type de comportement constitue l'une des formes de distraction les plus critiques, car il combine : distraction visuelle, distraction cognitive, et distraction manuelle.

Image montrant un candidat faisant un appel

La figure 5.19 montre un cas de distraction au volant lié à l'utilisation du téléphone pour effectuer un appel. L'image est extraite des captures générées par notre système et enregistrée dans le fichier driver_data.csv lors de l'analyse des différents états d'un conducteur.

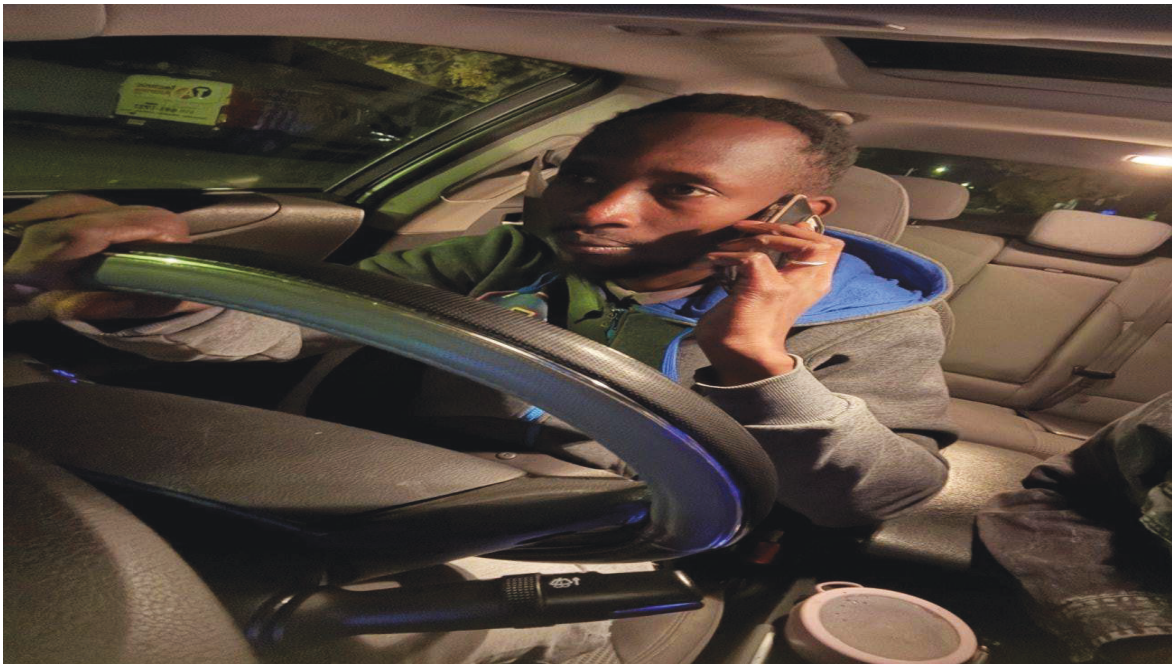


Figure 5.19 : Image montrant un conducteur distrait lors d'un appel téléphonique au volant.

Le conducteur tient le volant d'une main tandis que l'autre main est utilisée pour maintenir le téléphone à l'oreille. Ce comportement représente une forme de distraction particulièrement dangereuse aussi appelée attention divisée.

À la suite d'un ensemble d'expériences menées avec notre système de détection de la somnolence et de la distraction, plusieurs constats importants ont été mis en évidence. Tout d'abord, la détection des yeux constitue une étape fondamentale, car cette région du visage contient des informations essentielles permettant d'interpréter l'état d'un conducteur (éveil, fatigue, somnolence). Ainsi, pour garantir une interprétation fiable du comportement du conducteur, il est indispensable que la méthode utilisée pour la détection des yeux soit à la fois précise, stable et rapide. Le conducteur doit être bien éclairé pour une reconnaissance faciale

précise. La caméra doit être placée de manière à bien capter le visage du conducteur, sans obstacles. Le fichier music.wav est nécessaire pour les alertes sonores.

Ces mêmes conditions s'appliquent également pour la détection de la distraction, puisque celle-ci repose non seulement sur la reconnaissance faciale mais aussi sur l'identification d'objets distrayeurs (téléphone, regard détourné, tête baissée, etc.).

L'efficacité de notre système de détection de la somnolence et de la distraction a été évaluée à travers plusieurs scénarios réels et simulés, réalisés sur quatre conducteurs différents plaçant les conducteurs dans différents états (éveil, fatigue, bâillement, regard détourné, utilisation du téléphone, tête baissée).

L'accuracy permet d'évaluer la performance globale du système, tandis que la précision mesure la fiabilité des alertes déclenchées. Le rappel quantifie la capacité du système à détecter l'ensemble des situations dangereuses, et le F1-score fournit une mesure équilibrée combinant précision et rappel. Ces métriques offrent ainsi une évaluation complète et pertinente des performances du système de détection.

Dans ce travail les états des conducteurs ont été regroupés de la manière suivante :

Classe positives (situation dangereuse) : somnolence, micro-sommeil, fermeture prolongée des yeux, regard détourné, tête baissée, utilisation du téléphone.

Classe négative (situation normale) : état d'éveil, clignement normal

Le système discrimine les différents états du conducteur en s'appuyant sur une combinaison d'indicateurs visuels. La somnolence est identifiée à partir de l'analyse du ratio d'ouverture des yeux (EAR), notamment en cas de fermeture prolongée, tandis que la distraction est détectée grâce au modèle MobileNetV2, qui permet d'identifier des comportements tels que le regard détourné, la tête baissée ou l'utilisation du téléphone. En l'absence de ces signaux, l'état du conducteur est considéré comme normal.

Les tableaux 5.1, 5.2, 5.3 et 5.4 montrent la comparaison des performances de notre système.

Scénario testé	Méthode utilisée	Résultat observé	Performance	Commentaire
État normal (éveil + clignement)	MobileNetV2 + EAR	Aucune alerte	Très bonne	Bonne stabilité, absence de faux positifs
Somnolence (fermeture des yeux + micro-sommeil)	EAR + compteur temporel	Alerte déclenchée	Très bonne	Détection fiable avec prise en compte du temps
Signes de fatigue (bâillement)	Landmarks (bouche)	Détecté	Bonne	Indicateur complémentaire de fatigue
Distraction (regard détourné + tête baissée)	MobileNetV2	Détection de distraction	Bonne	Sensible à l'angle de la caméra
Utilisation du téléphone	MobileNetV2 (détection objet)	Alerte déclenchée	Très bonne	Détection robuste avec score élevé
Conditions dégradées (occlusion + faible éclairage)	CNN + landmarks / système global	Détection instable	Moyenne	Sensible aux obstacles et à la luminosité
Visage non détecté	CNN + dlib	Message d'erreur	Bonne	Évite les décisions erronées

Tableau 5.1 : Comparaison des performances du système de détection de la somnolence et de la distraction.

L'analyse du tableau met en évidence que le système présente d'excellentes performances dans les scénarios standards, notamment pour la détection de la somnolence basée sur l'EAR et la détection de la distraction via MobileNetV2. La combinaison des approches permet d'améliorer la robustesse globale. Toutefois, certaines limites apparaissent dans des conditions difficiles telles que les occlusions partielles du visage ou les environnements faiblement éclairés, ce qui constitue un axe d'amélioration futur.

	Prédit : Dangereux	Prédit : Normal	Total réel
Réel : Dangereux	5 706 (TP)	294 (FN)	6 000
Réel : Normal	280 (FP)	3 720 (TN)	4 000
Total prédit	5 986	4 014	10 000

Tableau 5.2 : Matrice de confusion globale du système de détection de la somnolence et de la distraction obtenus pour les conducteurs.

Dans cette matrice de confusion, les états de somnolence et de distraction sont regroupés dans une seule classe “dangereux”.

Scénario	Méthode	Accuracy (%)	Précision (%)	Rappel (%)	F1-score (%)
État normal (éveil + clignement)	MobileNetV2 + EAR	96.1	95.0	96.8	95.9
Somnolence (fermeture des yeux + micro-sommeil)	EAR + temporel	97.2	96.3	98.2	97.2
Signes de fatigue (bâillement)	Landmarks bouche	93.5	92.1	94.3	93.2
Distraction (regard détourné + tête baissée)	MobileNetV2	93.5	92.5	94.0	93.2
Utilisation du téléphone	MobileNetV2 (détection objet)	98.1	97.6	98.8	98.2
Conditions dégradées (occlusion + faible éclairage)	CNN + landmarks / système	87.8	86.2	88.7	87.5

Tableau 5.3 : Évaluation quantitative des performances du système de détection de la somnolence et de la distraction.

Le tableau met en évidence des performances globalement élevées du système, avec une détection particulièrement efficace des situations critiques, bien que certaines limites subsistent dans des conditions visuelles dégradées.

Métrique globale	Valeur (%)
Accuracy moyenne	94.1
Précision moyenne	93.0
Rappel moyen	95.1
F1-score moyen	94.0
Latence moyenne	≈ 35 ms / frame
Fréquence de traitement	≈ 28–30 FPS

Tableau 5.4 : Indicateurs globaux du système de détection de la somnolence et de la distraction.

Les métriques globales montrent que le système est à la fois précis, fiable et capable de fonctionner en temps réel, ce qui le rend adapté aux applications de surveillance du conducteur.

Ces résultats confirment la pertinence de la combinaison des approches CNN et vision artificielle pour une application en temps réel.

5.4 Conclusion

Dans ce chapitre, nous avons présenté en détail le système de détection de la somnolence et de distraction développé dans le cadre de ce projet. Ce système repose principalement sur l'analyse de l'état des yeux (ouverts ou fermés) grâce au calcul du EAR (Eye Aspect Ratio) afin d'évaluer le niveau de somnolence d'un conducteur (clignement prolongés des yeux, hochement de la tête) ainsi que sur la détection de comportements de distraction comme l'usage du téléphone cellulaire ou mauvaise orientation de la tête, identifiés grâce à des techniques de vision par ordinateur et à l'exploitation d'un réseau de neurones convolutifs (CNN).

Il existe plusieurs méthodes de détection de somnolence et distraction. L'utilisation de ces méthodes dépend du problème à résoudre.

Par ailleurs, nous avons décrit les outils, bibliothèques et environnements de développement essentiels à la mise en œuvre du système proposé. Enfin, nous avons présenté et analysé les résultats expérimentaux obtenus, lesquels démontrent l'efficacité et la robustesse du système dans la détection simultanée de la somnolence et des sources de distraction. Grâce à la combinaison d'approches de vision par ordinateur, de mesures géométriques telles que le EAR, et de modèles de deep learning comme MobileNetV2, le système parvient à identifier de manière fiable les comportements à risque en temps réel.

Toutefois, comme tout système embarqué, certaines limites et perspectives d'amélioration subsistent et méritent d'être abordées.

Chapitre 6

Conclusion générale

L'intelligence artificielle (IA) à ouverte de nouvelles vues dans beaucoup de domaines, surtout dans la détection de la somnolence et de la distraction au volant. Avec les progrès de l'apprentissage profond (deep learning), l'IA prend aujourd'hui une place clé dans le changement des pratiques liées à la sécurité sur route. En utilisant la force de ces méthodes, cette étude visait à améliorer la précision des façons de détecter la somnolence et la distraction, dans le but d'empêcher les accidents liés à la fatigue et au manque d'attention des conducteurs. Ce projet nous a permis d'explorer et de comparer plusieurs méthodes de détection, issues de différents paradigmes, permettant d'analyser et de surveiller en temps réel l'état d'un conducteur pendant la conduite.

L'intelligence artificielle a été très utile pour rendre plus efficaces les systèmes qui détectent la somnolence, surtout grâce à l'ajout du design MobileNetV2 (MobileNetV2 qui est utilisée pour identifier les signes de distraction) et aussi de la mesure eye_aspect_ratio (EAR qui permet d'évaluer l'état de somnolence du conducteur).

Donc notre approche est basée dans la combinaison complémentaire de MobileNetV2 qui excelle dans la reconnaissance d'objets tels que le téléphone portable, tandis que le EAR est un indicateur fiable pour détecter les fermetures prolongées des yeux, les clignements anormaux ou les bâillements.

Les résultats expérimentaux obtenus démontrent clairement l'efficacité et la robustesse du modèle proposé pour la détection de la somnolence et de la distraction. Le système réagit rapidement aux changements du comportement visuel et détecte de manière fiable les objets distrayeurs grâce au modèle MobileNetV2.

En conclusion, les résultats montrent l'effet important de l'intelligence artificielle et de l'apprentissage approfondi dans le domaine de la sécurité routière. La création et la mise en place d'un système qui utilise MobileNetV2 et eye_aspect_ratio (eye) constituent une contribution pertinente à l'amélioration des technologies de détection de la somnolence et de la distraction au volant contrairement aux systèmes classiques qui se limitent soit à la détection oculaire soit à la reconnaissance d'objets. En plus De nombreux travaux utilisent des modèles lourds comme VGG, ResNet ou YOLOv3 pour détecter la distraction. Ces architectures, bien que performantes, sont moins adaptées à une exécution en temps réel dans un véhicule. Le choix

de MobileNetV2, optimisé pour les environnements embarqués, constitue un avantage important, car il garantit : une vitesse d'exécution supérieure et une faible consommation de ressources. Notre travail se distingue en testant : différents scénarios de distraction, des conditions de lumière variées et plusieurs comportements réalistes (téléphone, regard détourné, tête baissée, bâillement, fermeture progressive des yeux).

Certaines parties du projet nécessitent des développements complémentaires, notamment l'analyse de la direction du regard et la détection de l'orientation de la tête. Ces deux éléments sont étroitement liés au niveau d'attention du conducteur. Leur étude approfondie permettrait d'améliorer la précision du système dans la détection de la fatigue et de la somnolence. Le traitement de ces problématiques relève directement du champ de la vision par ordinateur appliquée à la sécurité routière, et chacune d'entre elles pourrait constituer un sujet de recherche à part entière.

Le présent travail a mis en évidence la complexité du phénomène de somnolence et distraction au volant, qui fait intervenir des variables multiples et parfois interdépendantes.

Alors, il semble que notre sujet de recherche recèle encore de nombreux aspects à explorer, certains n'ayant pas été pensées au départ, mais offrant un potentiel considérable pour de futures études comme l'analyse de la direction du regard, la détection de l'état de la tête, où la direction du regard et l'état de la tête varient selon l'attention du conducteur. L'intégration de détecteurs d'objets plus récents comme YOLO, EfficientNet, MobileNetV3 et de modèles temporels plus avancés (comme les LSTM) pour analyser l'évolution du comportement sur de longues séquences, ainsi que l'ajout d'autres signaux complémentaires (fréquence cardiaque, capteurs embarqués) pourraient augmenter la précision dans la détection des distractions liées aux objets (téléphone, boisson, etc.). La diversité des visages, des conditions d'éclairage, des lieux de conduite et des angles de vue pourrait être élargie afin d'améliorer la généralisation du modèle et sa robustesse en conditions réelles.

En somme, les perspectives présentées ouvrent la voie à des recherches multidimensionnelles visant à perfectionner la détection de la somnolence et de la distraction. L'intégration de nouvelles techniques d'intelligence artificielle et de vision par ordinateur pourrait permettre, à terme, la conception de systèmes embarqués intelligents contribuant activement à la réduction des accidents de la route.

Références

- [1] Tefft, B.C. (2010). The Prevalence and Impact of Drowsy Driving (Technical Report). Washington, D.C: AAA Foundation for Traffic Safety. November 2010.
- [2] Zuopeng Zhao, Nana Zhou, Lan Zhang, Hualin Yan, Yi Xu, Zhongxin Zhang, et al., Driver fatigue detection based on convolutional neural networks using em-cnn. Computational intelligence and neuroscience, 2020.
- [3] Actua-formation, Distractions-et-somnolence-au-volant-les-risques-encourus 2022
- [4] Faiza Abdat. Reconnaissance automatique des émotions par données multimodales : expressions faciales et signaux physiologiques. Université de Metz, France, 2010.
- [5] W. Hizem et B. Dorizzi. Capteur intelligence pour la reconnaissance de visage, Thèse présentée pour l'obtention du diplôme de Docteur De l'Institut National Des Télécommunications : Electronique Informatique, Université Pierre et Marie Curie-Paris 6,2009.
- [6] D. Jayanthi et M. Bommy. Conducteur en temps réel basé sur la vision système de détection de la fatigue pour un contrôle efficace des véhicules. International Journal of Engineering and Advanced Technology (IJEAT), 2 :2249 – 8958, October 2012.
- [7] A. Lenskiy et J. Lee. Driver's eye blinking detection using novel color and texture segmentation algorithms. International Journal of Control, Automation, and Systems, pages 1598–6446, 2012.
- [8] A. M. Malla, P. R. Davidson, P. J. Bones, R. Green, et R. D. Jones. Automated video-based measurement of eye closure for detecting behavioral microsleep. In conférence internationale annuelle Conférence de l'IEEE EMBS Buenos Aires, Argentine, 31 août – 4 septembre 2010.
- [9] Nawal Alioua. Extraction et analyse des caractéristiques faciales : Application à l'hypovigilance chez le conducteur. PhD thesis, INSA de Rouen ; Université Mohammed V-Agdal (Rabat, Maroc). Faculté des sciences, 2015.
- [10] A. Lenskiy et J. Lee. Driver's eye blinking detection using novel color and texture segmentation algorithms. International Journal of Control, Automation, and Systems, pages 1598–6446, 2012.

- [11] A. M. Malla, P. R. Davidson, P. J. Bones, R. Green, et R. D. Jones. Automated video-based measurement of eye closure for detecting behavioral microsleep, In conférence internationale annuelle Conférence de l'IEEE EMBS Buenos Aires, Argentine, 31 août – 4 septembre 2010.
- [12] Christophe Olry, Des électrodes sur le cerveau pour traduire la pensée en mouvement, 24 juillet 2006.
- [13] Tzyy-Ping Jung, Colin Humphries, Te-Won Lee, Scott Makeig, Martin McKeown, Vicente Iragui et Terrence J Sejnowski. Extended ica removes artifacts from electroencephalographic recordings. *Advances in neural information processing systems*, 10, 1997.
- [14] Centre intégré en neuroimagerie et neurostimulation de Québec. Eeg. <http://www.cinq.ulaval.ca/EEG>, (Consulté le 12 janvier 2025).
- [15] Wen-Bing Horng, Chih-Yuan Chen, Yi Chang, et Chun-Hai Fan. Driver fatigue detection based on eye tracking and dynamic template matching. In *IEEE International Conference on Networking, Sensing and Control*, 2004, volume 1, pages 7–12. IEEE, 2004.
- [16] Wenhui Dong et Xiaojuan Wu. Fatigue detection based on the distance of eyelid. In *Proceedings of 2005 IEEE International Workshop on VLSI Design and Video Technology*, 2005., pages 365–368. IEEE, 2005.
- [17] David F Dinges et Richard Grace. Perclos: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance. US Department of Transportation, Federal Highway Administration, Publication Number FHWA-MCRT-98-006, 1998.
- [18] WW Wierwille. Overview of research on driver drowsiness definition and driver drowsiness detection. In *Proceedings: International Technical Conference on the Enhanced Safety of Vehicles*, volume 1995, pages 462–468. National Highway Traffic Safety Administration, 1995. 7
- [19] Kukkala, Vipin Kumar, et al. «Advanced Driver-Assistance Systems: A Path Toward Autonomous Vehicles. ». 25 septembre 2018. *IEEE Consumer Electronics Magazine*, vol. 7, no. 5, pp. 18-26.
- [20] M. Vats et A. Garg. Detection and security system for drowsy driver by using artificial neural network technique. *International Journal of Applied Science and Advance Technology*, 2012.

- [21] Pedro Borges. Deep Learning : réseaux de neurones récurrents. 18 octobre 2018
- [22] Everton Gomedes, Ph. D, L'apprentissage automatique : Révolutionnant notre monde grâce à l'intelligence artificielle, publié le 27 mai 2023.
- [23] Samaya Madhavan, M. Tim Jones Architectures de Deep Learning 2017.
- [24] Junjie Peng, Elizabeth C Jury, Pierre Dönnès, et Coziana Ciurtin. Machine learning techniques for personalised medicine approaches in immune-mediated chronic inflammatory diseases: applications and challenges. *Frontiers in pharmacology*, 12: 720694, 2021.
- [25] Allan David Gordon. Classification. CRC Press, 1999.
- [26] Priyal Agarwal. Regression analysis, 2021.
- [27] Daniel Vieira et Joao Paixao. Vector field neural networks. arXiv preprint arXiv :1905.07033, 2019.
- [28] Sharma Pulkit. The most comprehensive guide k-means clustering. Analytics Vidhya, 2019.
- [29] S Gangadhar et R Shanta. Chapter 8-machine learning. *Handbook of Statistics*, 38 :197–228, 2018.
- [30] Kimia Nadjahi Pascal Monasse. Découvrez les différentes couches d'un cnn, 2022.
- [31] Siddhesh Bangar, L'architecture de LeNet 5 juin 22, 2022
- [32] Zhichao Tian and Huabiao Qin. Real-time driver's eye state detection. In *IEEE International Conference on Vehicular Electronics and Safety*, 2005., pages 285–289. IEEE, 2005.
- [33] Debasish Kalita. An overview and applications of artificial neural network, 2022.
- [34] Rectified linear unit (relu), 2019.
- [35] Sagar Sharma, Simone Sharma, et Anidhya Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12) :310–316, 2017.
- [36] Talha Ashar. How to implement the softmax function in python.
- [37] Projectpro. Introduction to convolutional neural networks architecture, 2023.
- [38] Kimia Nadjahi et Pascal Monasse. Découvrez les différentes couches d'un CNN, 2022.

- [39] Sagar Sharma, Simone Sharma, et Anidhya Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12) :310–316, 2017.
- [40] Nebojša Bačanić Džakula et al. Convolutional neural network layers and architectures. In *Sinteza 2019-International Scientific Conference on Information Technology and Data Related Research*, pages 445–451. Singidunum University, 2019.
- [41] Pratik Ahamed, Soumyadeep Kundu, Tauseef Khan, Vikrant Bhateja, Ram Sarkar et Ayatullah Mollah. Handwritten arabic numerals recognition using convolutional neural network. *Journal of Ambient Intelligence and Humanized Computing*, 11, 11 2020.
- [42] Lindgren Walter, et al. « Requirements for the Design of Advanced Driver Assistance Systems - The Differences between Swedish and Chinese Drivers ». 2008. *International Journal of Design*. 2. pp. 41-54.
- [43] Zeiler, Matthew D. et Rob Fergus. "Visualiser et comprendre les réseaux convolutifs." Dans *Conférence européenne sur la vision par ordinateur*, 2014, pp. 818-833.
- [44] Azeem - J Un bref aperçu de l'architecture ZFNet 9 novembre 2023.
- [45] Siddhesh Bangar L'architecture d'AlexNet 24 juin 2022.
- [46] Siddhesh Bangar GoogLeNet : une plongée dans la technologie des réseaux neuronaux de Google 14 mai 2024.
- [47] Siddhesh Bangar L'architecture VGG-Net expliquée 28 juin 2022.
- [48] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Apprentissage résiduel profond pour la reconnaissance d'images, 10 déc. 2015
- [50] Thumani A., Soua R, Karray F, et Kamel MS, Tendances récentes dans les systèmes de surveillance de la sécurité du conducteur: état de l'art et défis, *IEEE Transactions on Vehicular Technology*. (2017) 66, no. 6, 4550 – 4563.
- [51] Soua MK et Bundele MM, Conception et analyse de l'algorithme k-means pour la détection de la fatigue cognitive chez le conducteur de véhicule à l'aide du signal de pouls

d'oxymétrie, Actes de la conférence internationale IEEE sur la communication et le contrôle informatique (IC4), septembre 2015, Indore, Inde.

[52] Ramesh MV, Nair AK, et Kunnathu A., Thekkeyil: réseau intelligent de capteurs de volant pour la surveillance et la détection en temps réel de la somnolence du conducteur, International Journal of Computer Science and Security (IJCSS) (2011).

[53] Balandong RP, Ahmad RF, Mohamad Saad MN, et Malik AS, Une revue des systèmes de détection automatique de la somnolence basés sur l'EEG pour les conducteurs, Ieee Access. (2018) 6, 22908 – 22919.

[54] Amol M Malla, Paul R Davidson, Philip J Bones, Richard Green et Richard D Jones. Automated video-based measurement of eye closure for detecting behavioral microsleep. In 2010 annual international conference of the IEEE engineering in medicine and biology, pages 6741–6744. IEEE, 2010.

[55] Zhichao Tian et Huabiao Qin. Real-time driver's eye state detection. In IEEE International Conference on Vehicular Electronics and Safety, 2005., pages 285 289. IEEE, 2005.

[56] Zhao X., Xu S., Rong J. et al., Seuil de discrimination de la fatigue de conduite basé sur l'entropie d'échantillon d'électroencéphalographie par analyse de courbe caractéristique de fonctionnement du récepteur, Journal de l'Université Jiaotong du Sud-Ouest. (2013) 48, no. 1, 178 – 183.

[57] Mark Sandler Andrew Howard Menglong Zhu Andrey Zhmoginov Liang-Chieh Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks 21 mars 2019.

[58] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, MobileNetV2 : résidus inversés et goulets d'étranglement linéaires 21 mars 2019.

[59] Giuseppe Amato, Fabrizio Falchi, Claudio Gennaro, Claudio Vairo, A Comparison of Face Verification with Facial Landmarks and Deep Features, April 2018.

[60] <https://opencv.org/>(Consulté le 12 Avril 2025).

[61] <http://www.numpy.org/> (Consulté le 12 Avril 2025).

[62] <https://www.dlib.org/> (Consulté le 12 Avril 2025).

[63] <https://www.tensorflow.org/> (Consulté le 12 Avril 2025).

[64] Dr David Picovski, Définition anatomique des lèvres.

- [65] Rohit F., Kulathumani V., Kavi R., Elwarfalli I., Kecojevic V., et Nimbarte A., Détection de la somnolence en temps réel à l'aide de bandeaux de détection cérébrale portables et légers, IET Intelligent Transport Systems . (2017) 11, no. 5, 255 – 263.
- [66] Anitha C., Venkatesha BS, et Adiga BS, Un système expert à deux volets pour la détection des bâillements, Procedia Computer Science. (2016) 92, 63 – 71.
- [67] Karchani M., Mazloumi A., Saraji GN et al., Présentation d'un modèle pour les changements dynamiques d'expression faciale dans la détection de la somnolence des conducteurs, Electronic Physician . (2015) 7, no. 2, 1073 – 1077.
- [68] Boon-Leng L, Dae-Seok L, et Boon-Giin L, Détection de la fatigue du conducteur par GSR et EMG à l'aide d'un dispositif portable mobile, Actes de la conférence TENCON 2015-2015 IEEE Region 10, novembre 2015, Macao, Chine.
- [69] Yan J.-J, Kuo H.-H., Lin Y.-F, et Liao T.-L, Système de détection de la somnolence du conducteur en temps réel basé sur PERCLOS et le traitement d'images en niveaux de gris, Actes du Symposium international sur l'informatique, le consommateur et le contrôle (IS3C) 2016, juillet 2016, Xi'an, Chine, 243 – 246.
- [70] Ibm, Qu'est-ce que la vision par ordinateur ?
- [71] Le Guide de l'auto, Comment une voiture détecte-t-elle la fatigue?, 12 juin 2023.
- [72] P. Ekman, « Darwin, Deception, and Facial Expression », Annals of the New York Academy of Sciences, vol. 1000, 2003, p. 205-22.
- [73] Silvan S. Tomkins, Affect Imagery Consciousness, vol. I, The Positive Affects, Londres, Tavistock, 1962.
- [74] Charles Darwin, The Expression of the Emotions in Man and Animals [archive], Londres, John Murray, 1872.
- [75] Patrick Verstichel, La reconnaissance des visages et ses anomalies, 01 octobre 2001 POUR LA SCIENCE N° 288.
- [76] Akshay L Chandra, Tutoriel : Filtres selfie utilisant l'apprentissage profond et OpenCV (détection des points de repère faciaux), 21 juillet 2018.
- [77] S.Guerfi. Authentification d'individus par reconnaissance de caractéristique biométriques liées aux visages 2D/3D, thèse doctorat, université d'Evry-val d'Essonne, France, 2008.

- [78] Appstud/lab-appstud-reconnaissance-de-visages-grace-aumachine-learning-85a612b368db 19.06.2022.
- [79] S. McKenna, S. Gong et Y. Raja : Modelling Facial Colour and Identity with Gaussian Mixtures. Pattern Recognition, 31(12):1883–1892, 1998.
- [80] H. Rowley, S. Baluja et T. Kanade: Neural Network-based Face Detection. IEEE Trans. Pattern Analysis and Machine Intelligence, 20(1):23–38, 1998.
- [81] V. Bruce. Recognizing faces. Lawrence Erlbaum Associates, London, U.K., 1988.
- [82] TOSALL, Système de surveillance des conducteurs, 3 avril 2020.
- [83] Découverte des Réseaux de Neurones Artificiels: Le Guide pour les novices, 29 juin 2023.
- [84] S. Hu and G. Zheng, "Driver Drowsiness Detection with Eyelid related Parameters by Support Vector Machine," Expert Systems with Applications, vol. 36, pp. 7651-7658, 2009.
- [85] Langis Michaud, Cligner des yeux souvent, un signe de bonne santé, 13 décembre 2021.
- [86] B. Hariri, S. Abtahi, S. Shirmohammadi, et L. Martel. Une méthode de mesure bâillante pour détecter Somnolence du conducteur. PhD thesis, Université d'Ottawa, Ottawa, Canada.
- [87] Alamyimages, le bâillement de l'homme au volant d'un véhicule, Publié le 09 août 2017.
- [88] Réseau de neurones artificiels, Wikipédia (Consulté le 15 Juin 2025).
- [89] Campus RÉCIT, 3.1 Les principes fondamentaux de l'apprentissage automatique et profonde (Consulté le 22 Mars 2025).
- [90] Campus RÉCIT, les phases d'apprentissage d'une IA (Consulté le 22 Mars 2025).
- [91] Kang S., Byun H., Lee SW. « Real-Time Pedestrian Detection Using Support Vector Machines. ». 2002. vol 2388.
- [92] <https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html>.
- [93] Jean-Jacques Speyer, Point sur les détecteurs d'hypovigilance et d'incapacité des pilotes de lignes, page 18.
- [94] Navid Shahsavari, Ehsan Zare Bidaki, Alexandre Wong, Paul J Murphy, Améliorations apportées à l'enregistrement, à la segmentation et à la suppression des artefacts d'images dans le système d'imagerie ThermOcular, Avril 2025.

[95] Dhruv Pandey, Détecteur de rapport d'aspect des yeux (EAR) et de somnolence utilisant dlib, 21 avril 2021.

[96] Neha Chaudhari, Détection des clignements d'yeux à l'aide de Python 12 juil. 2020.

[97] Samsara, Détection de la fatigue de nouvelle génération basée sur le réseau Samsara.

[98] <https://www.scitepress.org/Papers/2024/123860/123860.pdf>