

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
جامعة عمار ثلجي بالأغواط
UNIVERSITE AMAR TELIDJI LAGHOUAT



كلية العلوم
FACULTE DES SCIENCES
DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE

Mémoire de Master

Domaine : Mathématiques et Informatique
Filière : Informatique
Option : Systèmes et Réseaux Informatiques

Par :
GUEDDOUDA Bachir BOUKERMOUCHE Yacine
Thème

Application des méthodes d'intelligence artificielle pour la détection du comportement d'un conducteur.

Proposé par Mme : BENKOUIDER Sara

Année Universitaire 2020/2021

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

DÉDICACES

*Je dédie ce modeste travail aux deux être qui n'ont cessé de se sacrifier pour que je réussisse.
À ma mère, à mon père en témoignage de mon affection et de ma profonde gratitude pour leur soutien moral et financier et leurs encouragements, vous êtes le tout pour moi qu'Allah vous protège.*

À ma chère grande mère qui m'a soutenu et qui m'a donné la force pour continuer, qu'Allah la benisse.

À ma chère soeur qui m'a beaucoup aidé durant cet cursus modeste. À mon chère frère Anis mon bras droit, qu'Allah les protège pour moi.

À toute ma famille proche soit elle ou lointaine et à tous ce qui m'ont enseigné au long de ma vie scolaire.

À tous ceux que j'aime et ceux qui m'aiment.

Bachir

DÉDICACES

Je dédie ce mémoire à mes chers parents, qui m'ont soutenu et encouragé durant ces années d'études. Qu'ils trouvent ici le témoignage de ma profonde reconnaissance.

À mes chers frères et soeurs, ceux qui ont partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail. Ils m'ont chaleureusement supporté et encouragé tout au long de mon parcours.

À ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité.

À tous mes amis pour leurs aide et leurs soutien moral durant l'élaboration de ce travail.

Merci à tous.

Yacine

REMERCIEMENT

Tout d'abord, nous remercions Allah, notre créateur de nos avoir donné la force, la volonté et le courage afin d'accomplir ce modeste travail

Nous tenons à remercier notre encadreur Mme BENKOUIDER Sara pour son encadrement, sa disponibilité et ses précieux conseils durant l'élaboration de ce travail.

Nos remerciements également aux membres du jury qui ont accepté de juger et d'assister à la soutenance de ce travail.

Nous voudrions aussi exprimer notre gratitude à tous les enseignants du département d'informatique pour leurs qualités scientifiques et pédagogiques.

Nous tenons à remercier chaleureusement tous mes proches et tous ceux qui de près ou de loin, nous ont apporté leurs sollicitudes pour accomplir ce travail.

Résumé

La conduite de véhicules est une tâche importante qui nécessite une attention spéciale du conducteur pour prévenir les accidents de la route. La fatigue et la distraction au volant est l'un des principaux facteurs à l'origine des accidents de la circulation. La surveillance de l'état des yeux, de la position de la tête et le suivi de la bouche du conducteur présentent des critères efficaces pour détecter la fatigue et la distraction. Certains progrès ont été réalisés pour la détection de la fatigue du conducteur à l'aide des méthodes basée sur le contact qui utilise des pièces du véhicule (telles que l'angle de braquage et la pression sur la pédale) et des signaux physiologiques (électrocardiogramme et électromyogramme). Cependant, un système sans contact est plus puissant pour les conditions du monde réel. L'objectif principal de ce travail est de proposer un système robuste et performant basé sur les techniques d'intelligence artificielle pour une détection efficace de la fatigue et de la distraction du conducteur. Afin d'atteindre ces objectifs nous avons utilisé trois modèles différents, le premier modèle utilise un réseau de neurones pour détecter et analyser les yeux, le deuxième pour détecter et suivre le mouvement de la tête et le dernier modèle est utilisé pour analyser les mouvements de la bouche. L'utilisation de ces modèles nous a permis d'avoir de bons résultats et cela dans des scénarios différents.

Mots clés : Détection du comportement, Fatigue, Distraction, Intelligence artificielle, Réseau de neurones.

Abstract

Driving is an important task that requires special attention from the driver to prevent traffic accidents. Fatigue and distracted driving is one of the main factors causing traffic accidents. Monitoring eye condition, head position and driver mouth tracking present effective criteria for detecting fatigue and distraction. Some progress has been made to detect driver fatigue using contact-based methods that use vehicle parts (such as steering angle and pedal pressure) and physiological signals. However, a contactless system is more powerful for real world conditions. The main objective of this work is to propose a robust and efficient system based on artificial intelligence techniques for efficient detection of driver fatigue and distraction. In order to achieve these objectives we have used three different models, the first model uses a neural network to detect and analyze the eyes, the second to detect and follow the movement of the head and the last model is used to analyze the movements of the mouth. The use of these models allows us to have good results and that in different scenarios.

Keywords : Behavior detection, Fatigue, Distraction, Artificial intelligence, Neural network.

ملخص

القيادة مهمة تتطلب اهتماما خاصا من السائق لمنع وقوع الحوادث المرورية . يعتبر الإرهاق والقيادة المشتتة من العوامل الرئيسية المسببة للحوادث المرورية . تقدم مراقبة حالة العين ووضع الرأس وتتبع فم السائق معايير فعالة للكشف عن التعب والإلهاء . تم إحراز بعض (التقدم في اكتشاف إجهاد السائق باستخدام طرق قائمة على التلامس تستخدم أجزاء السيارة مثل زاوية التوجيه وضغط الدواسة) والإشارات الفسيولوجية . ومع ذلك ، فإن نظام عدم التلامس هو أكثر قوة لظروف العالم الحقيقي . الهدف الرئيسي من هذا العمل هو اقتراح نظام قوي وفعال يعتمد على تقنيات الذكاء الاصطناعي للكشف الفعال عن إجهاد السائق وإلهائه من أجل تحقيق هذه الأهداف استخدمنا ثلاثة نماذج مختلفة ، النموذج الأول يستخدم الشبكة العصبية لكشف وتحليل العينين ، والثاني لاكتشاف ومتابعة حركة الرأس ، والنموذج الأخير لتحليل حركات الفم . يتيح لنا استخدام هذه النماذج الحصول على نتائج جيدة وذلك في سيناريوهات مختلفة

الكلمات المفتاحية : كشف السلوك ، التعب ، الهاء ، الذكاء الاصطناعي ، الشبكة العصبية

TABLE DES MATIÈRES

Dédicaces	ii
Table des matières	vii
Liste des Figures	xi
Liste des Tableaux	xii
Liste des abréviations	xiii
1 Introduction générale	1
2 La détection de comportement du conducteur	4
2.1 Introduction	4
2.2 Les principaux causes des accidents	4
2.2.1 Causes sans rapport avec l'erreur humaine	5
2.2.2 Causes dus à une erreur du conducteur	5
2.3 La fatigue	6
2.3.1 Définition	6
2.3.2 Les causes de la fatigue	7
2.3.3 Les effets et les conséquences de la fatigue du conducteur	7
2.3.4 Techniques générales de détection de la fatigue	7
2.4 Les systèmes de détection de la fatigue	8
2.4.1 Système intrusif	10
2.4.2 Système non intrusif	13
2.5 Conclusion	15

3	Techniques de détection de la fatigue	16
3.1	Introduction	16
3.2	Processus de détection de la fatigue	16
3.2.1	Capture vidéo	17
3.2.2	Détection de visage	17
3.2.3	Extraction des caractéristiques	18
3.2.4	Analyse des caractéristiques	18
3.2.5	Classification	19
3.2.6	Analyse des travaux	20
3.3	Travaux connexes basés sur les réseaux de neurones CNN	21
3.3.1	Système de détection de fatigue en temps réel basé sur CNN multi-tâche	21
3.3.2	Détection de bâillement du conducteur basée sur l'apprentissage neuronal convolutionnel profond et le suivi du nez	23
3.3.3	Classification de la direction du regard oculaire en temps réel à l'aide d'un réseau neuronal convolutionnel	27
3.3.4	Analyse de l'état de l'œil basée sur le P-FDCN pour la détection de la fatigue	30
3.3.5	Comparaison entre les travaux précédents	33
3.3.6	Analyse et critiques	34
3.4	Conclusion	34
4	Approche proposée	35
4.1	Introduction	35
4.2	Les outils utilisés	35
4.2.1	Outils logiciels	35
4.2.2	Outils matériels	37
4.3	Notre approche	37
4.3.1	Préparation du dataset	38
4.3.2	Phase d'apprentissage (training our model)	39
4.3.3	Détection de la fatigue et de la distraction	44
4.4	Analyse et tests	47
4.4.1	Scénario 1 : Le conducteur dans l'état naturel (il ne porte rien)	48
4.4.2	Deuxième scénario : Le conducteur porte des lunettes	50
4.4.3	Troisième scénario : le conducteur porte un masque chirurgical	53
4.5	Conclusion	55

5 Conclusion générale et perspectives

57

TABLE DES FIGURES

2.1	Causes des accidents relevés en 2017	5
2.2	Classification des systèmes de détection de fatigue du conducteur (DFD)	9
2.3	Capteurs EEG [49]	11
2.4	Capteurs ECG [49]	11
2.5	Des capteurs connectés au dos sieges et volant [49]	12
2.6	L'écart-type de la position latérale (SDLP) [51]	13
2.7	Analyse du visage en utilisant le deep learning [50]	14
3.1	Processus de détection de la la fatigue du conducteur.	17
3.2	Modèle proposé de système de détection de la fatigue du conducteur [25].	21
3.3	Détecter les repères du visage [25].	22
3.4	Travail de la file d'attente de fréquences [25].	23
3.5	Présentation du système de détection de bâillement proposé [30].	24
3.6	Exemple de suivi échoué en raison du changement d'éclairage sous un pont [30].	25
3.7	<i>NN_YO</i> ,un réseau de neurones pour la détection des bâillements NN [30].	26
3.8	Caractéristiques de dégradé autour des coins de la bouche dans des états normaux et bâillements [30].	26
3.9	Schéma du système global [33].	28
3.10	Localisation de la région des yeux à l'aide d'une approche géométrique (ROI) [33].	28
3.11	Architecture du réseau utilisé [33].	29
3.12	Cadre de détection de fatigue [38].	30
3.13	Projection des noyaux des yeux [38].	31
3.14	Structure du modèle P-FDCN [38].	31
4.1	Schéma global.	37

4.2	250 images des yeux ouverts.	38
4.3	300 images des yeux fermés.	39
4.4	Training algorithm.	40
4.5	Shuffle And Normalisation algorithm.	41
4.6	Architecture de notre modèle.	41
4.7	Pseudocode de la création de notre modèle.	42
4.8	Notre modèle pendant l'entraînement.	43
4.9	La précision de notre modèle.	43
4.10	La perte de notre modèle.	43
4.11	Repères faciaux créés par Dlib.	45
4.12	Les points repères de la bouche créés par Dlib.	46
4.13	La détection de la distraction.	47
4.14	La précision de notre système dans un état naturel et normal	48
4.15	La précision de notre système dans un état naturel et fatigué	49
4.16	La précision de notre système dans un état naturel et distrait ou dans un état normale et entrain de bailler	50
4.17	La précision de notre système si le conducteur porte des lunettes et il est dans un état normale.	51
4.18	La précision de notre système si le conducteur porte des lunettes et il est fatigué.	51
4.19	La précision de notre système si le conducteur porte des lunettes et il est distrait.	52
4.20	La précision de notre système si le conducteur porte un masque et il est dans un état normale.	53
4.21	La précision de notre système si le conducteur porte un masque et il est fatigué.	54
4.22	La précision de notre système si le conducteur porte des lunettes et il est distrait.	55

LISTE DES TABLEAUX

2.1	Bande de fréquence des Signaux ECG et EEG [49]	11
3.1	Technique CNN sur la détection de la fatigue du conducteur.	20
3.2	Comparaison entre les travaux connexes.	33

LISTE DES ABRÉVIATIONS

Centre National de Prévention et de Sécurité routières	CNPSR
Advanced Driver Assistance Systems	ADAS
Convolutional Neural Network	CNN
Intelligent Transport System	STI
Blood Alcohol Concentration	BAC
Driver Fatigue Detection	DFD
Electrocardiography	ECG
Electro-encéphalogramme	EEG
Electro-oculogramme	EOG
Heart Rate	HR
Heart Rate Variability	HRV
Low frequency	LF
High frequency	HF
Standard Deviation of Lane Position	SDLP
Steering Wheel Movements	SWM
The Standard Deviation of Lateral Position	SDLP
Local Binary Patterns	LBP
Histograms of Oriented Gradients	HOG

L'industrie automobile a fait une amélioration radicale au cours de la dernière décennie qui a rendu nos véhicules plus puissants, plus faciles à conduire, écoénergétiques et respectueux de l'environnement. L'efficacité des conducteurs est un problème majeur entraînant des milliers d'accidents de la route chaque année. Environ 20 % des accidents mortels sont liés à la fatigue du conducteur. Près de 50 % de ces accidents entraînent la mort ou des blessures graves en raison de ces types d'accidents. L'estimation de la National Highway Traffic Safety Administration montre que près de 100,000 accidents chaque année sont liés à la fatigue des conducteurs. Cela cause chaque année près de 2,000 décès, 80,000 blessures et 13 milliards de dollars de pertes économiques [11].

La réduction des accidents et la sécurité publique sont les principaux objectifs du système de transport intelligent (ITS). La majorité des accidents de la route qui se produisent chaque année sont dus au conducteur fatigué. La fatigue réduit la capacité de réflexion et la capacité de prise de décision du conducteur. Le conducteur se sent somnolent après le déjeuner, si le sommeil n'est pas suffisant, une longue conduite entraîne certains des symptômes. Les rapports indiquent que les pourcentages d'accidents dus à la fatigue du conducteur sont différents selon les pays. La recherche montre qu'environ 10 à 20% des accidents de la route peuvent être évités en surveillant le visage du conducteur.

La fatigue du conducteur est une menace alarmante pour la sécurité routière et la situation est encore pire pour les gens qui travaillent pendant 24 heures. Les recherches montrent qu'un conducteur qui n'a pas dormi pendant 17 heures est exposé au même risque d'accident lorsqu'une personne dont le taux d'alcoolémie (taux d'alcool dans le sang) est de 0,05 g / 100 ml. Par conséquent, un conducteur fatigué court un risque deux fois supérieur à celui d'une personne sans alcoolémie. Le chauffeur qui n'a pas dormi pendant 24 heures a des performances de conduite similaires à celles d'une personne 0,1 g / 100 ml de BAC (Blood Alcohol Concentration)[12].

Les technologies avancées en électronique, capteurs, caméra, algorithmes de vision industrielle, données des dispositifs d'acquisition permettent de développer des systèmes de détection de comportement (SDC) en temps réel capable de surveiller le conducteur pendant son activité de conduite et qui offrent une option sécuritaire afin d'empêcher le pire. Les systèmes de détection de comportement ont comme principales fonctions de détecter tout signe de fatigue, de distraction ou de somnolence pouvant être décelé chez le conducteur, comme les yeux qui se ferment sur une période prolongée, une tête qui hoche vers l'avant ou tout autre mouvement erratique. Certains systèmes prennent en compte l'ensemble du visage du conducteur en analysant une multitude de points précis dans sa figure.

De nombreuses recherches ont été menées au cours de la dernière décennie sur la surveillance en temps réel du conducteur à l'aide d'un système de détection de comportement. Le système a beaucoup de contraintes en termes de cout, d'efficacité, de temps réel, de rapidité, taille et consommation d'énergie. Cette recherche est une contribution au développement de temps système de détection de comportement du conducteur pour relever les défis mentionnés ci-dessus.

L'objectif principal de ce travail est de concevoir et de développer un système de détection de comportement en utilisant les techniques basée sur la vision par ordinateur et sur les méthodes de l'intelligence artificielle. Notre système vise a détecté la fatigue et la distraction du conducteur en temps réel en analysant les yeux, la position de la tête ainsi que le bâillement. L'analyse et les tests effectués ont montré l'efficacité de notre système en terme de précision et cela dans différents scénarios et conditions. Voici les objectifs à atteindre lors du développement de notre système de détection du comportement :

- * Le système doit surveiller l'œil du conducteur, la bouche ainsi que le mouvement de sa tête.
- * Le système vise à surveiller le comportement du conducteur en temps réel sous différents conditions environnementales.
- * Les performances du système doivent être stables à tout moment.
- * Le système doit être précis, rapide et fiable.

La structure de ce memoire est résumée comme suit :

- Dans le **Chapitre II**, nous présentons tout d'abord les principaux causes des accidents. Puis, nous détaillons les deux facteurs principaux qui mettent le conducteur en risque. Par la suite, nous présentons les systèmes de détection de fatigue et ces deux types intrusif et non intrusif.
- Le **Chapitre III** est consacré à une revue de l'état de l'art des méthodes utilisées par les chercheurs pour détecter le comportement du conducteur et l'avertir dans les situations dangereuses, ainsi que la présentation et la comparaison entre ces travaux faits à base de

réseaux de neurones convolutionnels.

- Dans le **Chapitre IV**, nous présentons notre approche proposée dans ce travail, nous montrerons l'implémentation et les tests qui ont été effectués, ainsi que les langages et les outils utilisés dans cette partie. Enfin, nous terminons par une conclusion et quelques améliorations futures.

CHAPITRE 2

LA DÉTECTION DE COMPORTEMENT DU CONDUCTEUR

2.1 Introduction

Ces dernières années, l'évolution de la technologie a pris un essor important dans le domaine de l'industrie automobile, sans prendre en compte la sécurité des usagers. Tous les usagers empruntant les routes peuvent être victimes des différents dangers inhérents à la conduite. Cependant suite à des comportements inadaptés ils peuvent être victimes ou responsables des accidents. C'est pour cette raison que différents organismes, officiels comme associatifs, réalisent à travers le territoire national des actions de sensibilisation à la prévention routière, afin d'assurer la sécurité des différents usagers. Tels que les systèmes de détection de comportement du conducteur. Dans ce chapitre, nous passerons en revue les causes des accidents et en particulier la fatigue, les causes et les conséquences de cette dernière sur le comportement du conducteur, et finalement nous discuterons sur les systèmes de détection de comportement et leurs différents types.

2.2 Les principales causes des accidents

Malgré les nouvelles avancées de la technologie des véhicules, telles que les caméras de recul, les phares adaptatifs et les systèmes de collision avant, les accidents de voiture se produisent toujours. Les accidents peuvent changer radicalement la vie d'une personne. Ils peuvent entraîner une perte de salaire, de la douleur et de la souffrance ou des invalidités permanentes. La représentation graphique (Figure 2.1) faite par le Centre National de Prévention et de Sécurité Routières (CNPSR) montre les personnes tuées en Algérie sur les routes en 2017, Le graphique montre que 96,05 % des accidents sont causés par le facteur humain.

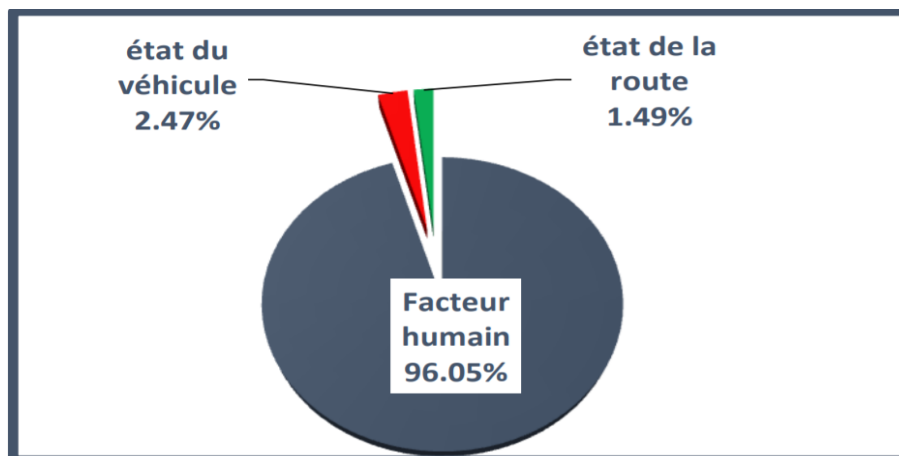


FIGURE 2.1 – Causes des accidents relevés en 2017

D'après la (Figure 2.1) on peut classer les causes des accidents en 2 catégories :

2.2.1 Causes sans rapport avec l'erreur humaine

Parfois, les conducteurs les plus sûrs sont victimes d'accidents de voiture, mais pas par leur propre erreur. D'autres facteurs peuvent avoir une grande influence.

- Conditions météorologiques** Les routes glissantes dues à la pluie, à la neige ou à la glace affectent la traction dans les pneus, ce qui entraîne la glissade des véhicules hors de la route ou heurte d'autres objets ou véhicules. Le brouillard, les niveaux d'eau et la vitesse du vent peuvent également contribuer aux accidents liés aux conditions météorologiques.
- Conditions routières mal entretenues** les routes augmentent le risque de crevaison d'un pneu et influencent la capacité du conducteur à freiner adéquatement.
- Défauts de véhicule** Les voitures se composent de centaines et de centaines de pièces qui les font fonctionner en toute sécurité et en douceur. Mais avec autant de pièces, il n'est pas rare qu'elles se détruisent et fonctionnent mal.

2.2.2 Causes dus à une erreur du conducteur

La plupart des accidents de la route n'ont pas nécessairement à voir avec le fait de savoir comment conduire un véhicule, mais plus encore avec les décisions que le conducteur prend au volant. En fait, environ 94 % de tous les accidents de la route sont causés par une erreur humaine. À un moment donné, un certain nombre de conducteurs se trouveront coupables d'avoir commis certaines des causes les plus courantes d'accidents de la route. Cependant, une sensibilisation accrue à ces causes peut inciter les conducteurs à être plus prudents sur la route.

- a. **Distraction au volant** L'une des principales causes d'accidents de voiture peut être attribuée à la distraction au volant. Malgré l'opinion populaire, le cerveau est incapable de se concentrer sur plus d'une tâche à la fois. Cela signifie que lorsqu'un conducteur en fait des textos, parle au téléphone, lit ou même mange en conduisant, le cerveau bascule entre les tâches et est incapable de se concentrer constamment sur ce qui s'en vient.
- b. **La conduite sous l'état d'ivresse** La conduite en état d'ébriété est responsable d'un grand nombre de décès par jour. Mais grâce à de récents efforts d'éducation, les décès liés à la conduite en état d'ébriété ont diminué.
- c. **L'excès de vitesse** aller tard au travail, suivre la circulation ou simplement conduire involontairement au-delà de la limite de vitesse sont parmi les principales causes d'accidents liés à la vitesse.
- d. **Le non respect du code de la route** Environ 40 % des accidents de voiture se produisent aux intersections, la cause étant l'exécution d'un feu rouge ou d'un panneau d'arrêt. Il ne vaut pas la peine d'éviter une à deux minutes (ou moins) d'attente pour risquer de vous blesser ou même de vous tuer ou de tuer d'autres personnes.
- e. **Conduite imprudente** Changer de voie trop rapidement, négliger ouvertement la sécurité et/ou les biens d'autrui et omettre délibérément de respecter les mesures de prudence ne sont que quelques-unes des caractéristiques de la conduite imprudente.
- f. **Conduite agressive** La conduite agressive comprend toute combinaison d'infractions au code de la route qui mettent en danger d'autres conducteurs ou biens, y compris les excès de vitesse.
- g. **Fatigue** La somnolence ou l'épuisement pur et simple peut nuire à la capacité d'un conducteur à réagir ou à discerner avec compétence les conditions sur la route.

Dans ce chapitre nous avons pris en considération un facteur principal qui est la fatigue et qui influe sur le style de conduite de l'automobiliste et qui peut engendrer des accidents de la route.

2.3 La fatigue

2.3.1 Définition

La fatigue est une état physiologique consécutif à un effort prolongé, à un travail physique ou intellectuel intense et se traduisant par une difficulté à continuer cet effort ou ce travail. Elle est considérée comme un symptôme plutôt qu'un signe car il s'agit d'un sentiment subjectif. La fatigue des conducteurs est l'une des causes des accidents de la route. Un conducteur fatigué

ne parvient souvent pas à arrêter le véhicule au bon moment, ce qui entraîne un accident. Les statistiques montrent que près de 20 % des accidents de la route sont dus à la fatigue des conducteurs. La plupart des collisions dues à la fatigue des conducteurs surviennent pendant les heures de matin.

2.3.2 Les causes de la fatigue

La fatigue est causée par un sommeil inadéquat sur une période de temps. Un minimum de sept à huit heures de sommeil sain est nécessaire pour une personne en une journée. Les conducteurs dont le sommeil est insuffisant prennent de mauvaises décisions au volant, ce qui augmente les risques d'accident. Les principales causes de fatigue du conducteur sont les suivantes :

- Manque de sommeil
- Conduire pendant la nuit
- Les troubles du sommeil
- Rythmes circadiens
- Temps consacré à la tâche de conduite
- La Forme physique du conducteur / condition physique
- Mauvaise alimentation
- Motivation personnelle et soucis
- La circulation

2.3.3 Les effets et les conséquences de la fatigue du conducteur

Les conséquences de la fatigue du conducteur peuvent être désastreuses parce qu'il affecte les capacités de prise de décision du conducteur qui conduit à traverser la voie et entrer en collision avec d'autres objets. Cet accident entraîne non seulement des dommages au véhicule, mais le conducteur qui conduit le véhicule et d'autres personnes assises à l'intérieur sera en danger de vie en dehors des dommages causés au véhicule et aux biens. Environ 50 % de risques de décès ou de blessures graves dans ce type d'accident lorsqu'un conducteur ne parvient pas à arrêter le véhicule. La fatigue réduit le temps de réaction, réduit la vigilance et la concentration et les capacités de prise de décision. Dans la plupart des cas, les conducteurs savent quand ils se sentent somnolents et peuvent prendre une décision quant à savoir s'ils doivent continuer à conduire ou à se reposer, mais ils choisissent généralement d'ignorer les risques.

2.3.4 Techniques générales de détection de la fatigue

Même s'il existe de nombreuses méthodes disponibles pour détecter la fatigue, les méthodes les plus utilisées sont classées en trois catégories : le modèle de conduite, mesures biologiques et

mesures des véhicules.

- a. Les modèles de conduite sont basés sur le principe du traitement d'image. Le modèle de conduite comprend les postures du corps, l'ouverture des yeux, le clignement des yeux, la détection des mouvements du regard de la tête et la détection des bâillements. Grâce aux techniques de traitement d'image, il est facile de surveiller le visage du conducteur et de rechercher les mouvements oculaires qui indiquent que le conducteur n'est plus en état de conduire. Dans de tels systèmes, les images capturées par la caméra sont analysées pour détecter la variation de conduite chez le conducteur. Le système de traitement d'image analyse le taux de fermeture et d'ouverture de la paupière ou le mouvement de la tête ou le clignement de la paupière ou le taux de bâillement.
- b. Les mesures biologiques utilisent des signaux médicaux tels que l'électrocardiogramme (ECG), l'électro-oculogramme (EOG), l'électroencéphalogramme (EEG) et la température. Dans de tels systèmes, l'activité physiologique est surveillée en permanence et signalée au conducteurs.
- c. Les mesures basées sur le véhicule impliquent des facteurs tels que le calcul de l'écart de voix, la mesure de la pression du véhicule lors de l'application des freins et la déviation de l'angle de braquage.

2.4 Les systèmes de détection de la fatigue

La fatigue au volant du conducteur est l'une des principales raisons des accidents de la route. Le problème est grave pour les conducteurs de véhicules commerciaux. Un accident typique de fatigue se produit en raison de la dérive hors de la chaussée sans freinage. Ces accidents se produisent généralement tôt le matin, c'est-à-dire entre 3h00 et 5h00. L'estimation de la fatigue du conducteur et l'alerte du conducteur sont une tâche difficile à laquelle sont confrontés de nombreux chercheurs. La fatigue du conducteur réduit les capacités de prise de décision du conducteur et en une fraction de seconde l'accident peut se produire, de sorte que le système de détection de la fatigue doit être plus précis, efficace et assez rapide pour alerter le conducteur le plus tôt possible.

Le développement d'un système de détection de la fatigue du conducteur est une tâche difficile parce que le système doit être placé dans le véhicule et doit prendre en compte la taille, la vitesse, le coût et la compatibilité et en outre il ne devrait pas distraire et ennuyer le conducteur au volant. Les hypothèses de base et les exigences d'un système de détection de fatigue (DFD) du conducteur sont les suivantes :

- * Le système DFD ne devrait pas favoriser l'utilisation d'une « alarme snooze ».
- * DFD devrait fournir une notification en temps opportun sur la fatigue.

* Le système d'alarme utilisé dans le système DFD ne doit pas produire d'alarme de nuisance.

Le système de surveillance de la fatigue du conducteur devrait gérer la structure et favoriser la compréhension de la fatigue au volant. Les points communs pour traiter la fatigue du conducteur sont les rythmes circadiens et une quantité suffisante de sommeil. Les niveaux d'alerte et de performance sont optimisés si le conducteur dort suffisamment en une journée. Dans le cas contraire, les niveaux de performance et les capacités décisionnelles du conducteur seront réduits. De nombreuses structures de gestion ont été proposées pour gérer la sécurité et la responsabilité du conducteur. Depuis, la fatigue n'est pas bien définie, ce qui rend ces structures de gestion qualitatives. Cela a confondu les décideurs avec moins de faits et plus d'options. Le chercheur a conçu de nombreuses façons de détecter et de surveiller la fatigue du conducteur à un stade précoce en fonction de différentes approches. Par conséquent, les systèmes de détection de la fatigue du conducteur (DFD) peuvent être classés en deux catégories (voir la Figure 2.2).

1. Système intrusif
2. Système non intrusif

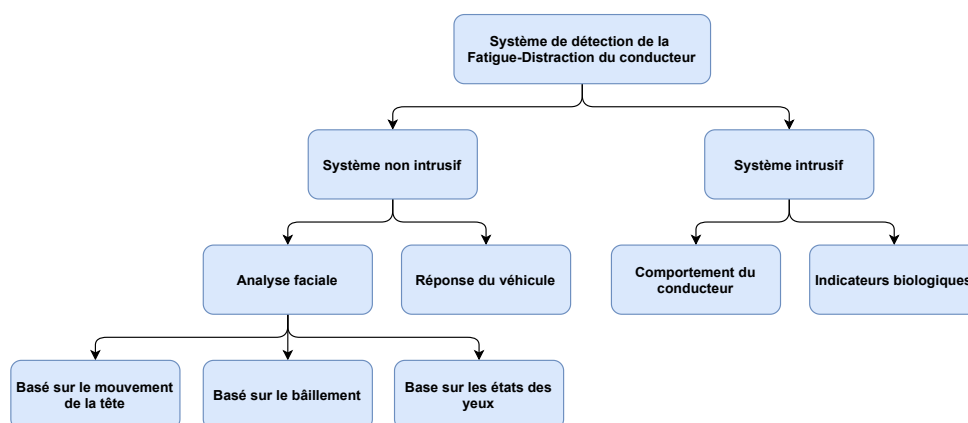


FIGURE 2.2 – Classification des systèmes de détection de fatigue du conducteur (DFD)

2.4.1 Système intrusif

Dans un système intrusif, le conducteur entre en contact physique avec des capteurs attachés à son corps, ce qui cause toujours de l'agacement et des dérangements. Les systèmes intrusifs sont classés en deux types :

- a. **Système de surveillance des caractéristiques physiologiques :** Dans cette approche, les chercheurs ont considéré les Signaux Électroencéphalogramme (EEG), Electro-Cardiogramme (ECG) pour mesurer la fatigue du conducteur . La fréquence cardiaque (HR) peut être facilement déterminée à l'aide de l'ECG est utilisée pour déterminer la fatigue du conducteur. La variabilité de la fréquence cardiaque (HRV) est un autre paramètre mesuré à l'aide du signal ECG a deux plages de fréquences. La basse fréquence (LF) tombe dans la plage de 0,04 à 0,15 Hz et la haute fréquence (HF) est comprise entre 0,14 et 0,4 Hz [3]. Les variations des intervalles R-R (battement à battement) de la fréquence cardiaque peuvent être mesurées à l'aide de VRC. Le rapport de LF à HF est mesuré en continu par des signaux ECG en connectant des sondes au conducteur, si les entraînements progressent vers le niveau de fatigue à partir d'un état éveillé, le rapport diminue.

Le signal physiologique le plus fréquemment utilisé pour mesurer la fatigue du conducteur est l'ElectroEncephaloGram (EEG). Les changements dans la bande de fréquence alpha (diminutions) et la bande de fréquence thêta (les augmentations) du signal EEG indiquent une fatigue. La Figure 2.3 et la Figure 2.4 montrent la connexion de l'EEG et Capteurs ECG au conducteur pour une surveillance continue des caractéristiques physiologiques.

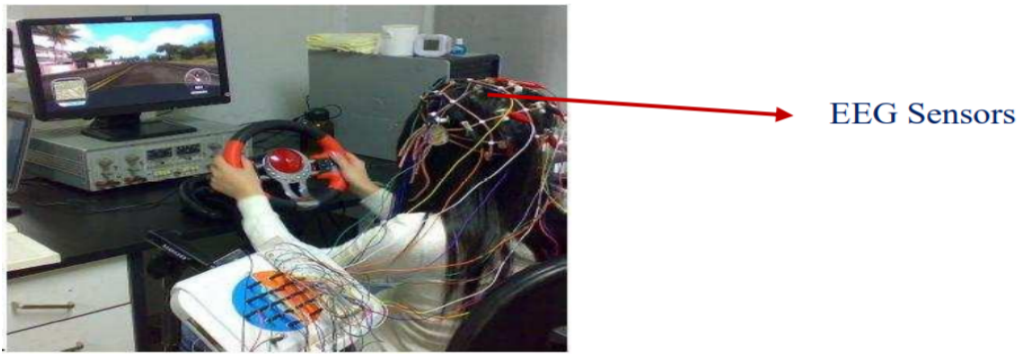


FIGURE 2.3 – Capteurs EEG [49]

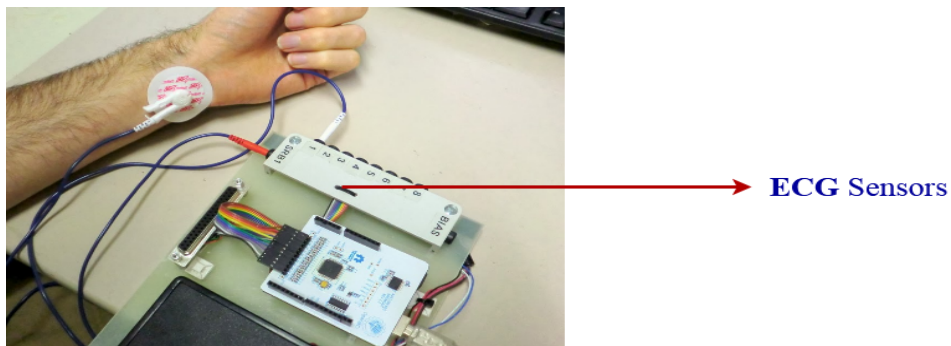


FIGURE 2.4 – Capteurs ECG [49]

Les différents niveaux de fréquence des signaux EEG sont de 0,5 à 4 Hz (bande delta), ce qui correspond à activité du sommeil, 4-8 Hz (bande thêta), associée à la fatigue, 8-13 Hz (bande alpha), qui correspond à la relaxation et à la créativité, et 13-25 Hz (bande bêta), qui se rapporte à vigilance [1] comme indiqué dans le Tableau 2.1.

TABLE 2.1 – Bande de fréquence des Signaux ECG et EEG [49]

ECG Signal		EEG Signal	
Gamme de fréquences(Hz)	Usage	Gamme de fréquences(Hz)	Activité
0.04 - 0.15	LF (basse fréquence)	0.5 - 4 (delta)	Dormir
0.14 - 0.4	HF (haute fréquence)	4 - 8 (theta)	fatigue
–	–	8 - 13 (alpha)	Relaxation

l'inconvénient majeur des systèmes de surveillance est d'attacher des capteurs directement au conducteur ce qui n'est pas réaliste et qui provoque des perturbations et des ennuis.

b. **Système de détection du fonctionnement du conducteur** : Dans cette approche, les changements physiques tels que l'affaissement de la posture sont mesurés en plaçant des capteurs sur la banquette arrière et la fréquence cardiaque (FC) est mesurée en plaçant des capteurs sur volant. Des capteurs de mouvement du volant (SWM) et un capteur de pouls ont été utilisés par Toyota pour enregistrer la fréquence cardiaque. La Figure 2.5 montre l'image des capteurs connectés au dos sièges et au volant.

L'inconvénient de cette approche, est qu'elle ne parvient pas à lire les signaux si le conducteur porte des gants et pose une seule main sur le volant. La performance des capteurs placés sur la banquette arrière réduit au contact de l'environnement et après une longue utilisation.

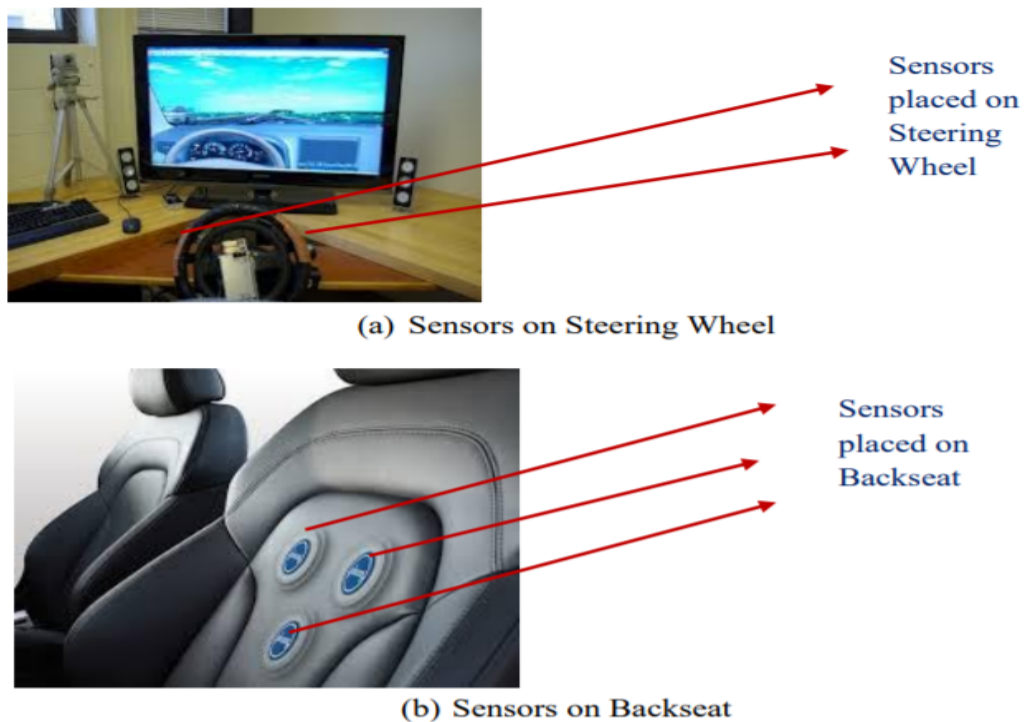


FIGURE 2.5 – Des capteurs connectés au dos sieges et volant [49]

Les systèmes discutés ci-dessus sont intrusifs parce que certains perturbent, distrayant le conducteur. Voici quelques limitations du système intrusif :

- Le matériel est complexe et extrêmement invasif
- Ne peut pas être placé facilement
- Mauvaise performance
- Perturbation
- Non fiable
- Bruit
- Énervant
- Impossible de mesurer l'ECG, la surveillance du signal EEG à long terme
- La connexion des sondes ECG et EEG prend plus de temps

2.4.2 Système non intrusif

Les limitations mentionnées ci-dessus peuvent être traitées en utilisant un système non intrusif. Dans ce système, le conducteur n'entre jamais en contact avec les électrodes, ne dérangera pas et n'ennuiera donc pas le conducteur. Le système non intrusif traite l'analyse faciale du conducteur.

- a. **Système de détection de la réponse du véhicule :** Dans cette approche, la réponse du véhicule est mesurée à l'aide des capteurs placés sur les véhicules. L'angle du volant est mesuré en plaçant un capteur d'angle, l'écart-type de la position latérale (SDLP) suit la position de la voie via caméra. Le mouvement de la voiture sur la voie de la route est surveillé par une caméra externe et un nombre de capteurs. Le prototype est illustré à la Figure 2.6. Caractéristiques géométriques, conditions de route et entraînement requis pour le conducteur sont les inconvénients du système.



FIGURE 2.6 – L'écart-type de la position latérale (SDLP) [51]

- b. **Système d'analyse du visage :** L'analyse d'image du visage est le domaine de recherche le plus courant pour les applications comme le système de sécurité humaine, la détection de visage, etc. La détection de visage est l'étape de base du système de reconnaissance qui a plusieurs applications telles que la détection et le suivi du visage, la reconnaissance du visage, d'expression, classification de genre, systèmes biométriques, systèmes de sécurité.

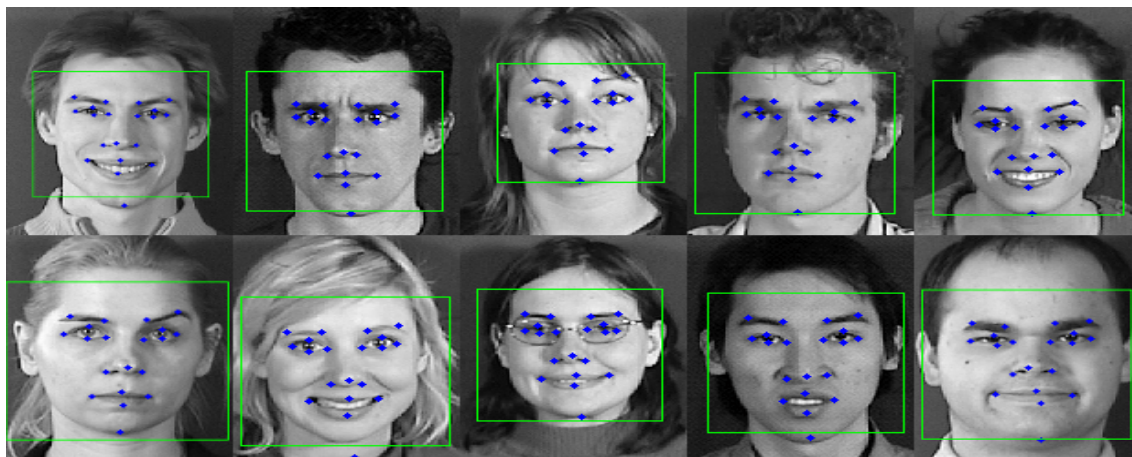


FIGURE 2.7 – Analyse du visage en utilisant le deep learning [50]

L'extension des algorithmes de reconnaissance faciale peut être appliquée pour identifier divers objets tels que les véhicules, les humains et les signes, etc[2]. Ces algorithmes peuvent être utilisés pour reconnaître et différencier divers traits du visage comme les sourcils, les yeux, la bouche, etc. (voir la Figure 2.7). Les émotions sont bien comprises grâce aux caractéristiques acoustiques. Pour améliorer le taux de détection, les caractéristiques visuelles et acoustiques sont combinées. Le développement d'une telle technique efficace est une tâche difficile pour de nombreux chercheurs. La reconnaissance faciale est la première et majeure étape du développement du système de détection de fatigue du conducteur. Dans le système de détection de fatigue non intrusif, la fatigue apparaît dans les yeux. Pour éviter les accidents de la route, on pense qu'en surveillant les yeux du conducteur, les symptômes de fatigue du conducteur peuvent être estimés. La détection de fatigue du conducteur implique la lecture de cadres continus et l'observation du visage, les yeux et les clignements des yeux. Non seulement les yeux, mais les bâillements, les hochements de tête et autres expressions faciales sont prises en compte dans la détection de la fatigue. Mais la recherche montre que la détection de la fatigue basée sur le pourcentage de la fermeture des yeux est le moyen le plus efficace pour détecter la fatigue à un stade précoce. Voici les exigences du système de détection de fatigue du conducteur :

- Le système doit être non intrusif, ce qui ne distraira pas le conducteur.

- Le système doit fonctionner dans toutes les conditions en temps réel avec une grande précision.

2.5 Conclusion

Dans ce chapitre, nous avons parlé des différentes causes des accidents qui sont peut être dus à une erreur humaine où qui sont dus à d'autres circonstances sans rapport avec le dernier. Nous avons détaillé sur la cause la plus dangereuse qui est la fatigue, ainsi qu'on a discuté les avantages et les inconvénients des différents systèmes de détection de la fatigue. Dans le chapitre suivant, on présentera certains travaux connexes qui ont essayé de résoudre le problème de détection de la fatigue.

CHAPITRE 3

TECHNIQUES DE DÉTECTION DE LA FATIGUE

3.1 Introduction

La fatigue au volant est considérée comme l'une des causes les plus graves d'accidents de la route mortels, en particulier pour les conducteurs de longue distance qui luttent considérablement dans des conditions de conduite monotones. En outre, divers types de systèmes d'avertissement ont été proposés parmi d'autres mesures préventives contre la fatigue. Ces systèmes émettent un avertissement lors de la détection d'un comportement dangereux du conducteur tel qu'un temps de fermeture oculaire prolongé, une augmentation du taux de bâillement ...etc. Ce chapitre donne un aperçu des procédures de mesure et des différentes méthodes de détection de la fatigue du conducteur et les organise par catégorie. En conséquence, un certain nombre d'accidents de la route peuvent être évités si le conducteur a été alerté à l'avance.

3.2 Processus de détection de la fatigue

Les méthodes comportementales mesurent les niveaux de la fatigue grâce à l'utilisation de caméras montées dans la voiture pour observer les traits du visage tels que l'état des yeux, le mouvement de la tête, le taux de clignotement et le bâillement. La plupart des chercheurs suivent un processus général pour extraire les traits du visage du flux de la caméra. Après l'obtention de ces fonctionnalités, un traitement supplémentaire est appliqué pour déterminer le niveau de la fatigue, généralement en appliquant des techniques d'apprentissage automatique telles que les machines à vecteurs de support (SVM), les réseaux neuronaux convolutionnels (CNN) ou les modèles de Markov cachés (HMM). Ces techniques sont entraînées à l'aide de fonctionnalités et de sorties étiquetées pour créer des modèles qui peuvent être utilisés pour la prédiction de la

fatigue. La partie la plus difficile de ce processus consiste à trouver un grand ensemble de données qui couvre la variabilité attendue entre les races et les différents pigments de la peau. Il s'agit d'un défi particulier en raison des problèmes de sécurité et de confidentialité qui surviennent lors de la publication d'ensembles de données à des fins académiques et commerciales. La série d'étapes suivies sur la Figure 3.1 suggère un processus commun de détection de la fatigue. Ces étapes sont :

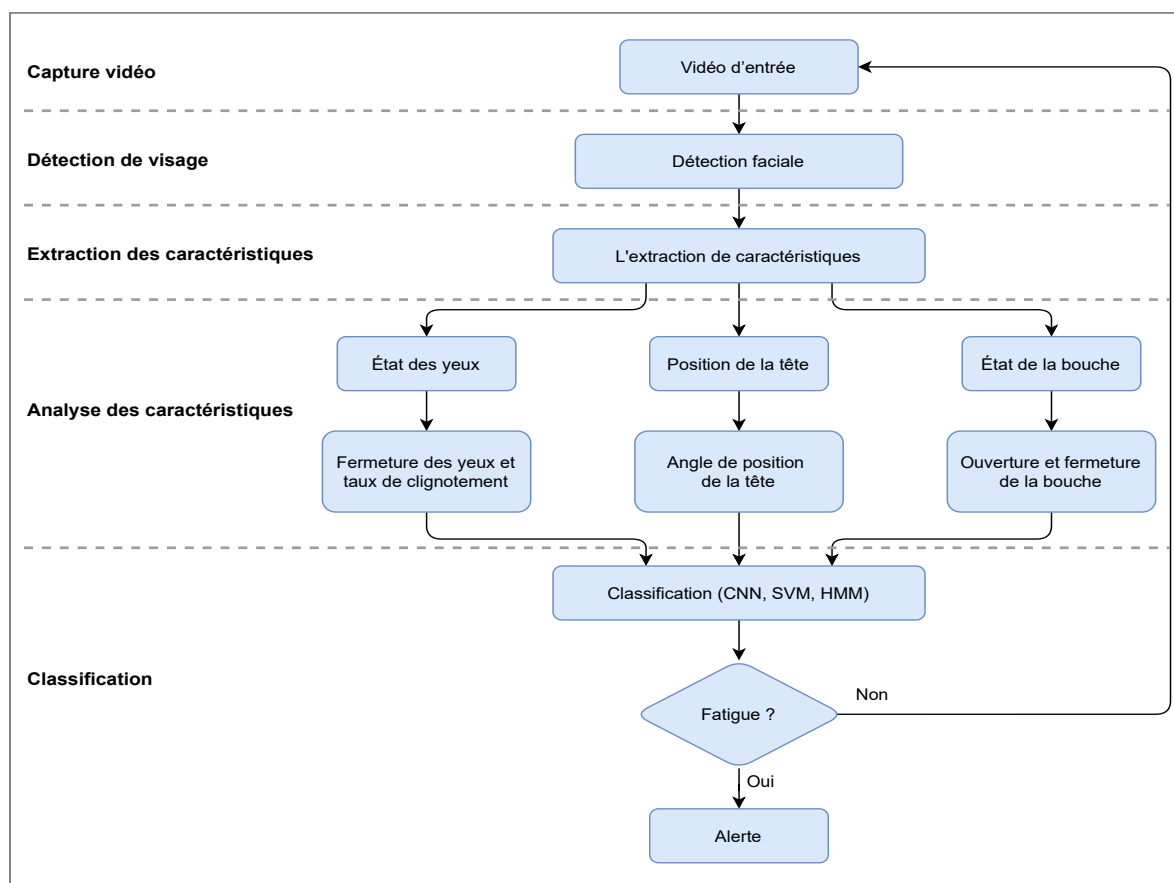


FIGURE 3.1 – Processus de détection de la la fatigue du conducteur.

3.2.1 Capture vidéo

C'est l'étape où les images vidéo d'une caméra fixe ou d'un smartphone sont décomposées en une série d'images. Les images vidéo sont prises de manière à ce que seul le visage du conducteur soit capturé.

3.2.2 Détection de visage

La deuxième étape vise généralement à détecter le visage dans les cadres d'image. Viola et Jones est l'algorithme le plus utilisé pour détecter le visage du conducteur à partir de l'image

[22]. Cependant, lorsque des CNN sont utilisés, l'image entière est généralement transmise à un réseau doté de plusieurs filtres et les fonctionnalités sont automatiquement extraites. Les CNN combinent les deux étapes de détection du visage et d'extraction des caractéristiques.

3.2.3 Extraction des caractéristiques

Si la détection de visage est appliquée, les caractéristiques sont généralement extraites à l'aide de différentes méthodes telles que la localisation des points de repère, l'histogramme des gradients orientés (HOG) et les modèles binaires locaux (LBP).

3.2.4 Analyse des caractéristiques

Les traits tracés peuvent ensuite être traités plus avant, comme c'est le cas pour PERCLOS ou EAR pour l'analyse de l'œil ou les méthodes basées sur la bouche pour la détection du bâillement. Dans ce qui suit, nous montrons les traits du visage qui sont généralement pris sur le visage du conducteur pendant la conduite :

- a. **L'analyse de la fermeture de yeux** : L'état des yeux est une caractéristique importante qui est largement utilisée pour déterminer la fatigue chez le conducteur. Les méthodes utilisées pour mesurer le niveau de fatigue comprennent le pourcentage de fermeture des yeux (PERCLOS) [13] et le rapport d'aspect des yeux (EAR). EAR est le rapport entre la hauteur et la largeur de l'œil et a été introduit par Soukupova et Cech en 2016 [14]. En revanche, perclos est le pourcentage de fermeture des yeux sur une période de temps. La principale différence entre les deux est que l'EAR classe le rapport de l'œil à mesure qu'il diminue tandis que perclos classe si l'œil est ouvert ou fermé.
- b. **Taux de clignement des yeux** : Les méthodes qui mesurent le taux de clignement utilisent la fréquence des clignements des yeux pour mesurer la fatigue. Lorsque le conducteur est somnolent, le taux de clignotement diminue.
- c. **Analyses de bâillement** : Le bâillement est un processus principalement involontaire d'ouverture de la bouche et d'inspiration profonde, remplissant les poumons d'air. C'est une réponse très naturelle à la fatigue. En fait, le bâillement est généralement déclenché par la somnolence ou la fatigue. Certains bâillements sont courts et d'autres durent plusieurs secondes avant d'expirer la bouche ouverte. Des yeux larmoyants, des étirements ou des soupirs audibles peuvent accompagner le bâillement [52].
- d. **Analyse des expressions faciales** : L'analyse de l'expression faciale fait référence aux systèmes informatiques qui tentent d'analyser et de reconnaître automatiquement les mouvements du visage et les modifications des caractéristiques du visage à partir d'informations visuelles.

Parfois, l'analyse des expressions faciales est confondue avec l'analyse des émotions dans le domaine de la vision par ordinateur [53].

3.2.5 Classification

L'étape de classification se compose de classificateurs qui sont utilisés pour la prise de décision sur le niveau de la fatigue d'un conducteur. Si le classificateur détecte des traits de la fatigue sur la base des paramètres pondérés, alors une alarme sera activée suggérant qu'un conducteur fait une pause.

Diverses mesures sont utilisées dans différentes études pour détecter un visage et extraire des caractéristiques du flux vidéo. Malheureusement, la plupart de ces études utilisent des ensembles de données différents qui peuvent favoriser leurs propres algorithmes. Cela est dû au manque d'ensembles de données standardisés pouvant servir de référence. En conséquence, il est difficile de comparer les approches en évaluant simplement les exactitudes signalées. Les techniques d'apprentissage automatique pour classer différents niveaux de la fatigue sont maintenant discutées, ainsi qu'un examen des mesures qui forment un système de détection de la fatigue du conducteur.

- a. **Support Vector Machines (SVM)** Les machines à vecteurs de support ou séparateurs à vaste marge (en anglais support vector machine, SVM) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression. Les SVM sont une généralisation des classifieurs linéaires [54]. Ils tentent de trouver un hyperplan qui sépare les données d'entraînement en classes prédéfinies. Dans le domaine de la fatigue du conducteur, les SVM sont principalement utilisées pour apprendre à classer différents états du conducteur à partir de données étiquetées. De nombreux travaux ont tenté d'utiliser les capacités des SVM dans la détection de la fatigue. Un certain nombre de mesures ont été utilisées comme fonctionnalités pour déterminer le niveau de la fatigue d'un conducteur à l'aide de SVM. Mais un inconvénient potentiel de cette approche est la faible fréquence d'images pouvant être atteinte, ce qui pourrait entraîner des expressions faciales manquées.
- b. **Hidden Markov Model (HMM)** Les HMM sont un modèle statistique utilisé pour faire des prédictions sur les états cachés en fonction des états observés définis par des probabilités. Les HMM ont été développées à la fin des années 1960 et au début des années 1970 par Leonard Baum et ses collègues [18]. Aujourd'hui, les HMM sont largement utilisés dans des applications telles que la reconnaissance d'expression faciale, l'annotation de gènes, la modélisation des erreurs de séquence d'ADN et la classification des virus informatiques [19] [20].
- c. **Réseaux de neurones Convolutionnels (CNN)** Les CNN sont similaires à un réseau neuronal ordinaire qui est également composé de neurones constitués de poids apprenables [21]. Les

CNN utilisent des couches de convolutions spatiales bien adaptées aux images, qui présentent de fortes corrélations spatiales. Les CNN ont fait leurs preuves dans des domaines tels que la reconnaissance d'images, l'analyse vidéo et la classification [22] Yann Le Cun et Yoshua Bengio ont été les premiers à appliquer les CNN en vision par ordinateur [23], mais l'excellente performance des CNN en vision par ordinateur n'est apparue qu'en 2012, lorsque les réseaux de neurones à convolution profonde ont montré d'excellents résultats en reconnaissance d'objets [24].

Quelques travaux à base des trois techniques précédentes SVM, HNN, CNN sont mentionnés dans le tableau 2.1 avec leurs degré de précision .

TABLE 3.1 – Technique CNN sur la détection de la fatigue du conducteur.

Auteur	Année	Mesure	Classificateurs /méthodes	Précision %
A. George and A. Routray	2016	Le regard des yeux	CNN (Viola-Jones algorithme)	98.32
F. Zhang et al	2017	état des yeux	CNN (AdaBoost LBF and PERCLOSE)	95.18
Y. Sun et al	2013	Le clignement des yeux	HMM, SVM	90.99
Pan et al.	2007	Le clignement des yeux	HMM	95.7
L.Pauly and D. Sankar	2015	état des yeux	HOG, SVM	91.6
A. Punitha et al	2014	état des yeux	SVM	93.5

3.2.6 Analyse des travaux

Bien que de nombreux travaux aient été menés à ce jour, il reste encore beaucoup à faire pour améliorer les systèmes de détection de la fatigue des conducteurs. Le principal défi identifié au cours de cet examen est que chacun des systèmes examinés utilisait des ensembles de données différents pour atteindre ses objectifs et, par conséquent, ne sont pas faciles à comparer. De plus, les ensembles de données utilisés pour tester ces systèmes ont tendance à être limités et sont généralement capturés dans des environnements contrôlés, ce qui peut entraîner des défaillances dans des situations réelles. Dans une tentative de fournir une comparaison équitable. On a collecté des articles qui utilisent principalement la précision de la classification pour comparer les performances des systèmes. L'estimation des performances a révélé que les CNN produisaient des résultats plus précis que les SVM et les HMM.

3.3 Travaux connexes basés sur les réseaux de neurones CNN

3.3.1 Système de détection de fatigue en temps réel basé sur CNN multi-tâche

B. K. Savas and Y. Becerikli [25] ont proposé un modèle de réseau neuronal convolutionnel Multi-tâches ConCNN, ce modèle est utilisé pour détecter la fatigue du conducteur en temps réel. L'algorithme Dlib est utilisé pour identifier avec précision les informations sur l'œil et la bouche du conducteur. Ensuite, le système est formé avec des modèles ConNN multi-tâches pour la détermination des paramètres de fatigue. La plage de fréquences à utiliser ici est maintenue constante dans l'étude et le nombre d'images est fixé. Enfin, selon les paramètres de fatigue, la fatigue est évaluée comme étant « très fatiguée, moins fatiguée et pas fatiguée ». Ces situations sont également testées et codées dynamiquement à certaines périodes qui maintiennent leur continuité. Les performances de précision du système testé en temps réel sont très robustes. Dans cette étude, la détection de la fatigue du conducteur est effectuée avec ConNN multi-tâches à l'aide de données brutes obtenues à partir de différents ensembles de données dans la littérature. Le modèle du système de détection de fatigue du conducteur est présenté sur la Figure 3.2

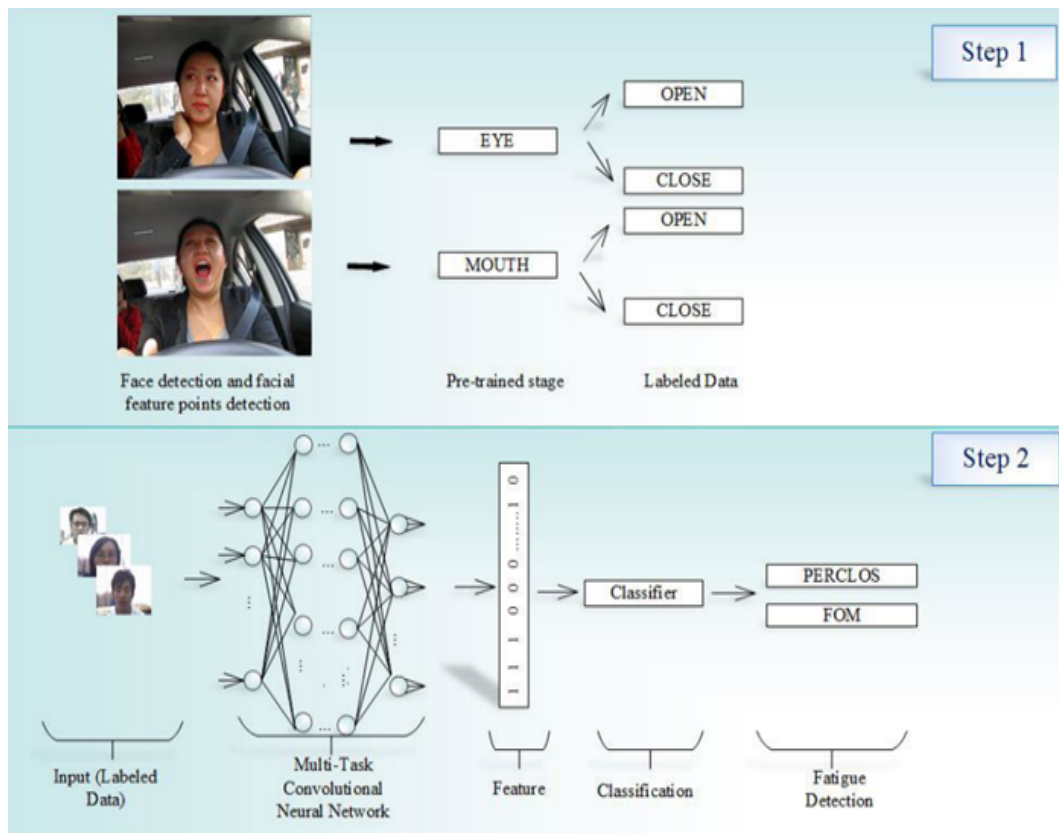


FIGURE 3.2 – Modèle proposé de système de détection de la fatigue du conducteur [25].

Dlib, qui est une bibliothèque et une plate-forme de programmation indépendante créée avec C++ , contient un ensemble de données de pointeur de visage créé par Sagonas et al. Le détecteur de repère facial pré-entraîné de la Bibliothèque Dlib est utilisé pour prédire l'emplacement de 68 coordonnées (x,y) qui cartographient les repères faciaux dans la zone faciale [26]. La détection des repères faciaux est un sujet critique en termes d'estimation des formes de zone faciale. Dans cette étude, la bibliothèque Dlib est utilisée pour détecter et suivre le visage des conducteur dans des vidéos en temps réel. Par conséquent, plusieurs structures faciales sont détectées sur la zone du visage à l'aide de méthodes. La détection des points de repère du visage est illustrée à la Figure 3.3



FIGURE 3.3 – Détecter les repères du visage [25].

Deux ensembles de données différents (YawDD et NthuDDD) sont utilisés dans l'étude pour l'apprentissage et les tests. Avec l'algorithme Dlib, l'emplacement de 68 points dans la région du visage est capturé sur la détermination de la plage de fréquences. Par conséquent, PERCLOS (pourcentage de fermeture des yeux) et FOM (fréquence de bâillement) sont calculés dans la vidéo pour déterminer les régions de l'œil et de la bouche du conducteur. Ainsi, les informations sur l'état de la bouche et de l'œil sont étiquetées. Les données étiquetées sont classées par ConNN multitâche. Afin de calculer avec précision les paramètres de fatigue, une certaine plage de fréquences est nécessaire (nombre total d'images : N). La plage de fréquences à utiliser ici est maintenue constante dans l'étude et le nombre d'images est fixé. Chaque nouveau frame est supprimé et le dernier frame est aussi supprimé. Ainsi, la plage de fréquences reste constante.

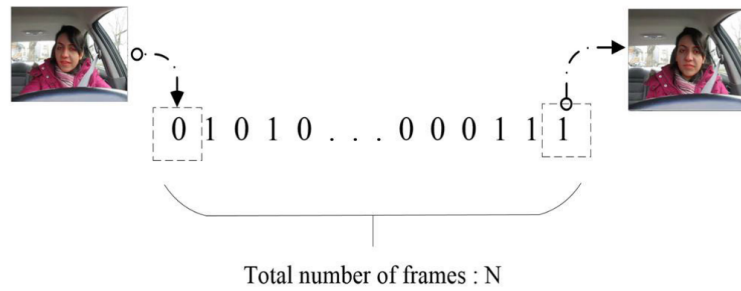


FIGURE 3.4 – Travail de la file d'attente de fréquences [25].

Les algorithmes d'optimisation Stochastic Gradient Descent (SGD) [29] sont utilisés dans l'apprentissage du modèle proposé. Lorsque les conducteurs sont fatigués, ils auront une série de réactions comportementales telles que les yeux fermés ou bâillements. Par conséquent, il peut être accepté, la fatigue plus sèche peut être calibrée en calculant le paramètre PERCLOS et FOM. Selon la littérature [27], [28], l'état de fatigue du conducteur a été déterminé lorsque le seuil perclos $F_{perclos} > 0,24$. En outre, lorsque $F_{fom} > 0,16$, il est déterminé que les symptômes d'étirement indiquent la fatigue. L'état de fatigue du conducteur est divisé en trois niveaux. Le premier type est un niveau très fatiguant et cette évaluation se compose de deux étapes. Le premier est seulement lorsque le temps de fermeture des yeux dépasse 5 sec ($n_{close} > 150$ images). Dans ce cas, le système donne un avertissement indépendamment du bâillement. L'autre contrôle l'état oculaire et l'état de bâillement ensemble. Si l'heure de fermeture des yeux du conducteur $F_{perclos} > 0,24 \parallel F_{fom} > 0,16$, le conducteur est extrêmement fatigué. Dans le deuxième type de fatigue, le conducteur ferme les yeux $F_{perclos} = 0,15 \sim 0,24 \parallel$ Si $f_{fom} > 0,16$, on observe que le conducteur est moins fatigué. Le troisième type est lorsque le chauffeur est dans l'état normal. Ici, le conducteur n'a aucun signe de fatigue et peut conduire normalement. Dans cette étude, les valeurs seuils PERCLOS et FOM sont testées sur des vidéos de test, le système proposé peut modéliser la relation interactive entre l'œil, la bouche et les sous-états.

3.3.2 Détection de bâillement du conducteur basée sur l'apprentissage neuronal convolutionnel profond et le suivi du nez

Zhang et al [30] présentent un système de détection de bâillement qui se compose d'un détecteur de visage, d'un détecteur de nez, d'un traqueur de nez et d'un détecteur de bâillement. Des algorithmes d'apprentissage en profondeur sont développés pour détecter la zone du visage du conducteur et l'emplacement du nez. Un algorithme de suivi du nez qui combine le filtre de Kalman avec un tracker TLD (Track-Learning-Detection) open-source dédié et développé pour générer des résultats de suivi robustes dans des conditions de conduite dynamiques. Enfin, un réseau neuronal est développé pour la détection de bâillement sur la base des fonctionnalités

telles que la valeur de confiance du suivi du nez, les fonctionnalités de gradient autour des coins de la bouche et les fonctionnalités de mouvement du visage.

Le modèle de calcul représenté sur la Figure 3.5 est développé pour une détection robuste de bâillement du conducteur dans diverses conditions d'éclairage de la conduite. Le système contient trois principaux composants, un détecteur de visage, un détecteur de nez, un traqueur de nez et un détecteur de bâillement. Le détecteur de visage utilise un réseau neuronal à convolution profonde (CNN) pour localiser la zone du visage dans une image du conducteur, et un autre CNN est développé pour détecter le nez du conducteur dans la zone du visage détectée. En tant que première étape du système de détection de bâillement, les détecteurs de visage et de nez jouent une importance vitale sur les performances du système, car dans le système de détection de bâillement à plusieurs étapes, les performances de chaque étape dépendent des résultats générés par les étapes précédentes. Les résultats générés par le détecteur de nez sont utilisés comme entrée dans le traqueur de nez basé sur TLD pour estimer le mouvement dans les images suivantes, suivre le changement d'apparence dynamique de l'expression du visage du conducteur dans l'imagerie vidéo, identifier la zone de délimitation du nez et générer un suivi du nez dans chaque image. La zone de recherche dans cet algorithme est limitée à une petite région basée sur la position du nez dans l'image précédente générée par le filtrage de Kalman pour obtenir des résultats efficaces. Pour la détection du bâillement, des fonctions de gradient autour des coins de la bouche ont été utilisées, et la valeur du gradient a subi des changements significatifs pendant le bâillement. Étant donné qu'un conducteur tourne souvent la tête, les valeurs de gradient de bord seules ne seraient pas suffisantes. Par conséquent, le détecteur de bâillement (Yawning Detector), un réseau neuronal, prend également la valeur de confiance du suivi du nez et l'orientation du mouvement du visage comme entrée pour une détection de bâillement robuste.

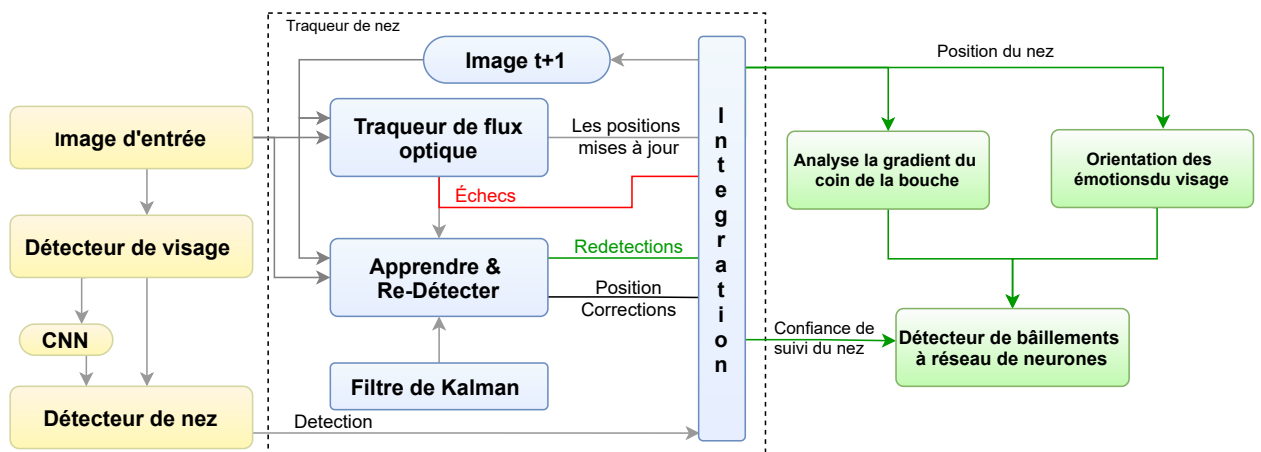


FIGURE 3.5 – Présentation du système de détection de bâillement proposé [30].

TLD est un algorithme de vision par ordinateur proposé par Kalal, Mikolajczyk et Jiri Matas

[31] pour suivre un objet unique dans des environnements sans contraintes et il est utilisé pour le suivi d'objets à long terme en temps réel. Cependant, le suivi du nez du conducteur dans la conduite réelle est confronté à de nombreux défis. Le principal défi vient des changements soudains d'éclairage pendant la conduite tels que passer sous un pont, entrer dans un tunnel, etc (voir la Figure 3.6). Pour résoudre ces problèmes l'algorithme TLD a été proposé pour suivre le nez et développer un schéma de re-détection du nez pour obtenir un nouvel emplacement en cas d'échec du suivi. Etant donné que le visage du conducteur a une plage de déplacement relativement petite, il est possible de re-détecter le nez dans une zone limitée de la trame actuelle autour de l'emplacement du nez détecté dans la trame précédente. Une méthode de filtrage de Kalman est utilisée pour estimer la zone de recherche en fonction de l'emplacement du nez détecté précédemment.



FIGURE 3.6 – Exemple de suivi échoué en raison du changement d'éclairage sous un pont [30].

La figure suivante Figure 3.7 illustre le réseau neuronal développé, *NN_YD*, pour détecter l'état de bâillement.

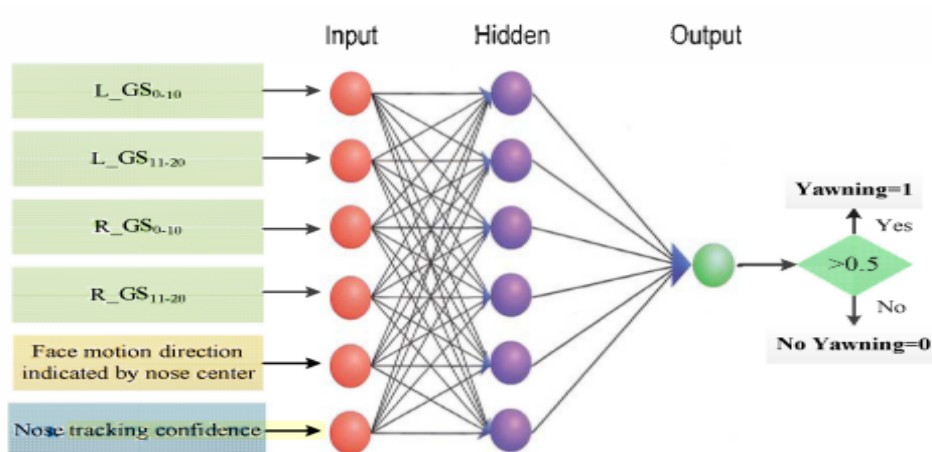


FIGURE 3.7 – *NN_YO*, un réseau de neurones pour la détection des bâillements NN [30].

Pour la détection de bâillement, la région de la bouche est divisée en deux zones, les zones de gauche et de droite (voir Figure 3.8). À partir de la moitié gauche de la région de la bouche, ils ont calculés les caractéristiques L_GS (0-10), qui est la somme des intensités de gradient des pixels dans la moitié gauche de la région de la bouche qui sont comprises entre 0 et 10 degrés, et L_GS (11-20), qui est défini de manière similaire comme L_GS (0-10), sauf que les forces sont dans la plage de 11 à 20 degrés. À partir de la moitié droite de la région de l'embouchure, les caractéristiques R_GS (0-10) et R_GS (11-20) ont été calculées, similaires à L_GS (0-10) et L_GS (11-20). Ces caractéristiques sont illustrées sur la Figure 3.8 L'image du côté gauche montre la forme de la bouche à l'état normal et celle du côté droit l'état de bâillement. Lors du bâillement, le nombre d'arêtes verticales autour des coins de la bouche augmente considérablement, ce qui implique que davantage de pixels ont des inclinaisons de gradient comprises entre 0 et 20 degrés. Par conséquent, L_GS et R_GS sont des fonctionnalités puissantes pour détecter l'état de bâillement.

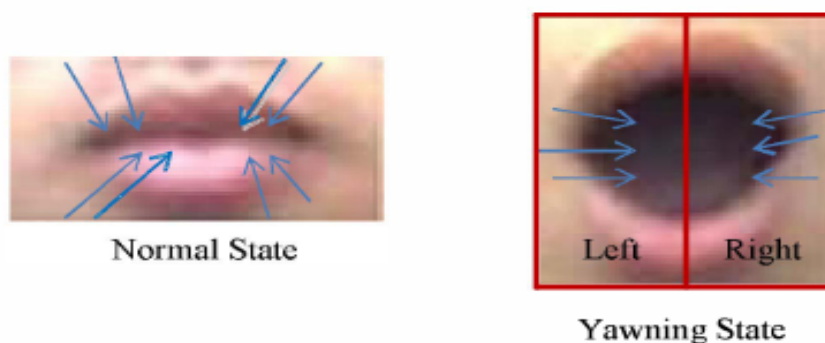


FIGURE 3.8 – Caractéristiques de dégradé autour des coins de la bouche dans des états normaux et bâillements [30].

Les résultats expérimentaux utilisant le jeu de données YawDD [32] montrent que le réseau convolutif profond peut générer un résultat de classification satisfaisant pour détecter le visage et le nez du conducteur par rapport à d'autres méthodes de classification, et le système de détection de bâillement est efficace dans la détection en temps réel du bâillement du conducteur.

3.3.3 Classification de la direction du regard oculaire en temps réel à l'aide d'un réseau neuronal convolutif

George et al. [33] ont proposé un système en temps réel pour la classification de la direction du regard et l'estimation des indices d'accès oculaire. Dans la première étape, l'algorithme détecte le visage à l'aide d'une version modifiée de l'algorithme de Viola-Jones. Une région oculaire rugueuse est obtenue à l'aide de relations géométriques et de repères faciaux. La région de l'œil obtenue à l'étape suivante est nécessaire pour classer la direction du regard oculaire à l'aide du réseau neuronal convolutif. Ce modèle traite la classification de la direction du regard comme un problème de classification multiclasse, évitant ainsi la nécessité d'un étalonnage. Les directions oculaires obtenues peuvent être utilisées pour trouver l'IAO et ainsi déduire le processus cognitif de l'utilisateur. Les modèles dans lesquels les yeux se déplacent lorsque les humains accèdent leurs souvenirs sont connus sous le nom d'indices d'accès oculaire (IAO). Les informations obtenues peuvent être utiles pour l'analyse de vidéos d'interrogation, l'interaction homme-machine, la recherche d'informations, etc. Les faits caractéristiques de ce modèle sont présentés ci-dessous :

- Propose un système en temps réel pour la classification de la direction du regard.
- Utilisation d'un réseau neuronal convolutif basé sur le classificateur de direction du regard, qui est robuste contre les erreurs de localisation oculaire.
- L'approche proposée surpasse les algorithmes de pointe dans la classification de la direction du regard.
- L'algorithme atteint une fréquence d'images moyenne de 24 fps dans un environnement de bureau.

Le schéma global proposé est illustré à la Figure 3.9, différentes étapes de l'algorithme sont décrites ci-dessous.

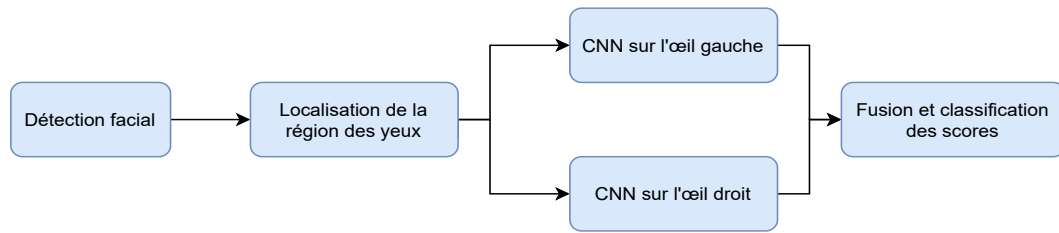


FIGURE 3.9 – Schéma du système global [33].

La première étape de l'algorithme est la détection des visages. Une version modifiée de la méthode viola-jones [34] a été utilisée, l'étape suivante consiste à obtenir la région de l'œil. Ils ont utilisé deux méthodes différentes pour obtenir la région de l'œil. Dans la première méthode, la région de l'œil pour la classification est obtenue géométriquement à partir du cadre de sélection du visage renvoyée par le détecteur de visage (ROI). La dimension de la région de l'œil est affichée sur une image du jeu de données HPEG [35], dans la Figure 3.10 les régions oculaires obtenues sont re-mises à l'échelle à une résolution de 42 x 50 pour les étapes suivantes (ROI). Dans la deuxième méthode, un détecteur de repère facial est utilisé pour trouver les coins des yeux et d'autres points facial.

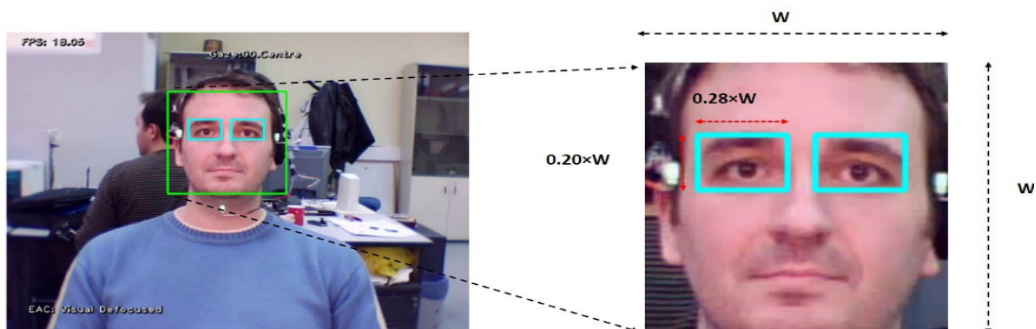


FIGURE 3.10 – Localisation de la région des yeux à l'aide d'une approche géométrique (ROI) [33].

Pour la classification de la direction du regard oculaire, la région de l'œil obtenue à partir de l'étape précédente est utilisée dans un cadre de classification multiclasse pour prédire les classes EAC. Le réseau neuronal convolutionnel (CNN) est utilisé pour la classification avec trois couches de convolution. L'étape d'entrée est constitué d'images de dimension 42 x 50. Pour la première couche convolutionnelle, 24 filtres de dimension 7 x 7 sont utilisés. Cette étape a été suivie d'une unité linéaire redresseur (ReLU). La couche ReLU introduit une non-linéarité aux activations. La fonction de non-linéarité peut être représentée comme suit :

$$f(x) = \max(0, x)$$

où, x est l'entrée et $f(x)$ la sortie après l'unité ReLU. Une couche de regroupement max est ajoutée après l'étape ReLU. La couche de regroupement max effectue un sous-échantillonnage spatial de chaque image en sortie. Ils ont utilisé des couches de mise en commun maximale de 2×2 qui réduisent la résolution spatiale à la moitié. Deux étages similaires avec des dimensions de filtre 5×5 et 3×3 sont également ajoutés. Après les couches convolutionnelle, ReLU et max-pooling dans la troisième couche convolutionnelle, les sorties de toutes les activations sont jointes dans une couche entièrement connectée. Le nombre de neurones de sortie correspond au nombre de classes dans l'application particulière. La structure du réseau est illustrée à la Figure 3.11. La perte softmax est utilisée sur les classes comme mesure d'erreur.

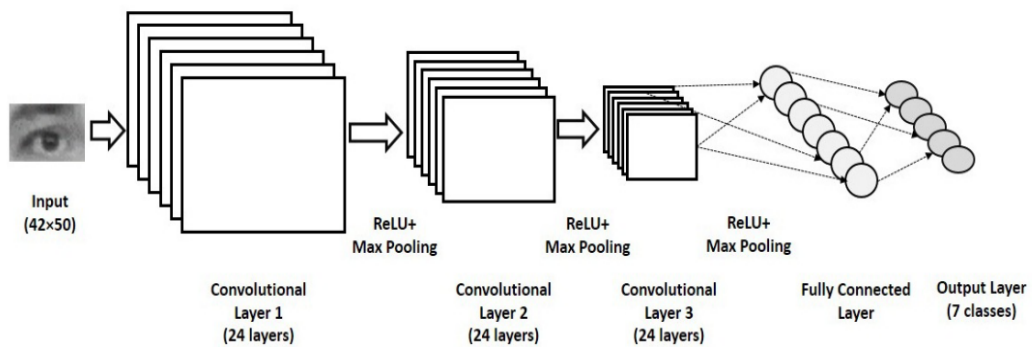


FIGURE 3.11 – Architecture du réseau utilisé [33].

Deux réseaux CNN sont formés indépendamment pour les yeux gauche et droit. Les scores des deux réseaux sont utilisés pour obtenir les étiquettes de classe.

$$\text{score} = (\text{scoreL} + \text{scoreR}) / 2$$

où scoreL et scoreR désignent les scores obtenus à partir de CNN gauche et droite respectivement. Les expériences qui ont été menées dans la base de données Eye Chimera [36], [37] qui contient les sept classes IAO. Ce jeu de données contient des images de 40 sujets. Pour chaque sujet, des images avec des directions de regard différentes (pour différentes classes de IAO) sont disponibles. Le nombre total d'images dans l'ensemble de données est de 1170. Enfin, un cadre de classification en temps réel de la direction du regard est présenté. La direction estimée du regard oculaire est utilisée pour déduire l'accès oculaire aux indices, donnant des informations sur les états cognitifs. La complexité de calcul est très moindre. L'algorithme proposé fonctionne même avec des rotations dans le plan de la face. La direction du regard obtenue peut également être utilisée pour des applications d'interaction homme-ordinateur. La complexité de calcul de l'algorithme en phase de test est moindre, ce qui le rend adapté aux appareils intelligents avec des caméras basse résolution utilisant des modèles pré entraînés.

3.3.4 Analyse de l'état de l'œil basée sur le P-FDCN pour la détection de la fatigue

Huang et coll. [38] ont proposé une nouvelle approche de détection de la fatigue fondée sur l'analyse de l'état des yeux. Plus précisément, ils ont d'abord construit un réseau convolutionnel de détection de fatigue (FDCN) basé sur un réseau neuronal convolutionnel commun (CNN). Ensuite, ils intègrent des cœurs de projection dans FDCN pour construire P-FDCN qui non seulement améliorent la résistance des fonctionnalités apprises aux changements d'échelle, mais renforcent l'apprentissage des informations de texture. Leurs résultats expérimentaux démontrent que l'approche proposée permet d'atteindre le taux de reconnaissance de 94,9 %, ce qui donne une promotion de 1,2 % du taux de précision moyen par rapport aux résultats obtenus en utilisant le CNN seul sur la base de données Closed Eyes in the Wild (CEW). En outre, leur approche a une amélioration de la précision de 1,0 % sur la base de données ZJU, ainsi que de surpasser considérablement le modèle RCNN plus rapide et la méthode de projection traditionnelle. Le cadre général du système de détection de la fatigue proposé comprend deux étapes, comme le montre la Figure 3.12. La première étape est la détection du visage et la position des points de repère. Dans ce processus, le réseau convolutionnel en cascade multitâche (MTCNN) [39] est choisi car il est connu comme l'un des détecteurs de visage les plus rapides et les plus précis. Le modèle MTCNN calcule la taille du visage détecté, puis recadre l'œil en fonction de la distribution de position des traits du visage. Plus précisément, ils sélectionnent une tranche d'une hauteur de $1/5$ de visage et d'une largeur de $9/25$ comme zone des yeux centrée sur le point de repère de l'œil. La segmentation oculaire à l'aide de la méthode adaptative peut améliorer la robustesse du système. La deuxième étape est le modèle de détection de fatigue. FDCN prend la sortie de la première étape (tranche oculaire) comme entrée. Les étapes suivantes expliqueront en détail le fonctionnement des deux modèles FDCN et P-FDCN.

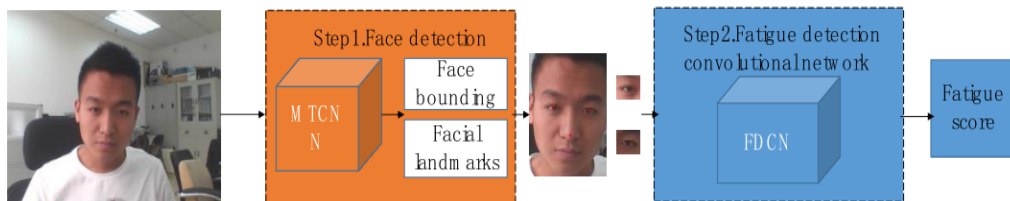


FIGURE 3.12 – Cadre de détection de fatigue [38].

Le modèle FDCN prend la tranche de l'œil gauche et la tranche de l'œil droit comme entrée du réseau. La taille d'origine de la tranche d'entrée est liée à la taille du visage, puis elle redimensionne les tranches dans une taille de $80 * 80$ uniformément. Ce réseau se compose de quatre couches de convolution et de deux couches entièrement connectées. Chaque couche est convolutionnelle suivie d'une couche de regroupement. Pour réduire le nombre de neurones et

les poids, la convolution par champs réceptifs locaux, le partage de paramètres et la mise en commun des réseaux neuronaux sont utilisés pour optimiser la structure. D'autre part, la méthode de projection (modèle P-FDCN) est simple et efficace à utiliser. La projection mappe les entités de l'espace de haute dimension à l'espace simple, ce qui rend les entités plus compactes. La diffusion d'une image peut également mieux apprendre ses propriétés de texture.

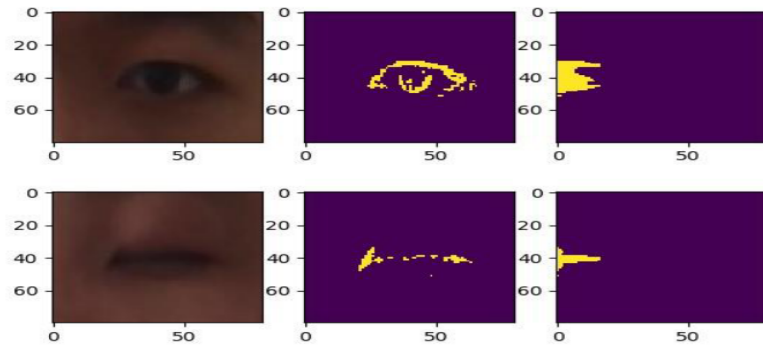


FIGURE 3.13 – Projection des noyaux des yeux [38].

Ainsi, le P-FDCN intègre la projection dans les réseaux neuronaux pour ajouter une modulation artificielle. Par rapport au FDCN, le P-FDCN ajoute une méthode de calcul hors ligne pour effectuer un traitement de projection verticale sur les tranches oculaires d'entrée en tant que prétraitement. Plus précisément, il utilise la projection de zone sensible locale qui sélectionne uniquement un pixel dont le voisinage ressemble le plus à la pupille oculaire. Le résultat de la projection hors connexion est illustré à la Figure 3.13. La première colonne est des images d'origine, la deuxième colonne est des images binaires et la dernière colonne est des images projetées sensibles locales. La première ligne comporte un test oculaire ouvert et la deuxième ligne comporte un test oculaire fermé.

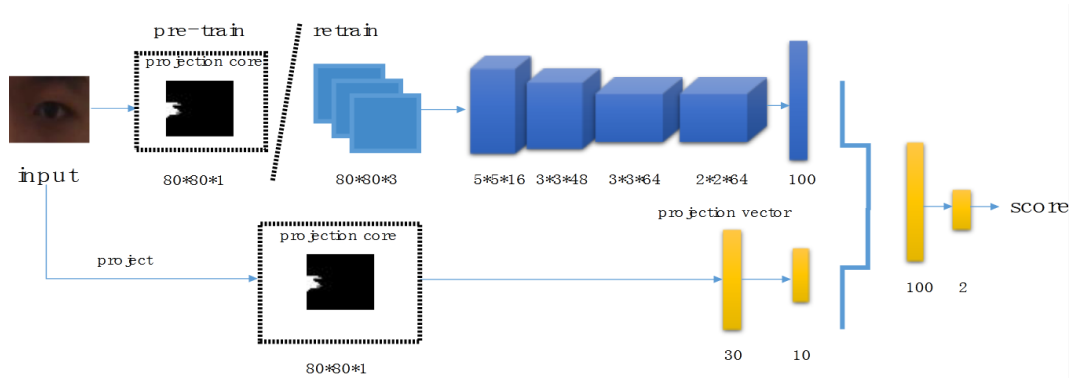


FIGURE 3.14 – Structure du modèle P-FDCN [38].

Après avoir construit les cartes de projection et les avoir appliquées au pré-entraînement du

FDCN, les images d'origine sont utilisées pour reformer le réseau comme illustré à la Figure 3.14. Dans ce modèle, le réseau à double flux est utilisé pour l'apprentissage. Le deuxième flux prend le noyau de projection comme entrée. Puis il vectorise les caractéristiques de projection dans le sens verticale pour obtenir un vecteur propre de dimension 30. Les deux couches entièrement connectées sont connectées aux deux dernières couches.

3.3.5 Comparaison entre les travaux précédents

TABLE 3.2 – Comparaison entre les travaux connexes.

Méthode	Auteurs / Année	Dataset	Methodes de détection	Measures
Système de détection de fatigue en temps réel basé sur ConNN multitâche	B. K. Savas and Y. Becerikli / 2020	YawDD NthuDDD	ConCNN en tant que classificateur. L'algorithme Dlib comme un détecteur facial.	Détection des yeux (PERCLOS) et de la bouche (FOM)
Détection de bâillement du conducteur basée sur l'apprentissage neuronal convolutionnel profond et le suivi du nez	W. Zhang, Y. L. Murphey, T. Wang, and Q. Xu / 2015	YawDD	Détecteur de visage. Détecteur de nez. Traqueur de nez. Détecteur de bâillement. Deep CNN.	Détection du nez et de la bouche
Classification de la direction du regard oculaire en temps réel à l'aide d'un réseau neuronal convolutionnel	A. George and A. Routray / 2016	Eye Chimera dataset	Système en temps réel pour la classification de la direction du regard et l'estimation du EAC. L'algorithme de Viola-Jones pour la Détection faciale.	Détection des yeux (direction du regard)
Analyse l'état de l'œil basée sur le P-FDCN pour la détection de la fatigue	R. Huang, Y. Wang, and L. Guo / 2018	CEW ZJU	Une nouvelle approche de détection de fatigue basée sur l'analyse de l'état des yeux. MTCNN est utilisé comme un détecteur de visage. La méthode P-FDCN est utilisée comme un classificateur.	Détection des yeux (PERCLOS)

3.3.6 Analyse et critiques

Après analyse du Tableau 3.2 nous avons constaté que la plupart des auteurs ont obtenu des résultats satisfaisants sur la détection et la reconnaissance des visages. En effet, les auteurs avec la détection de la fatigue et de la somnolence du conducteur, ont pu réduire le nombre d'accidents et cela grâce aux algorithmes proposés. Cependant, malgré les résultats encourageants, nous constatons certains manquements : -dans les travaux de (B. K. Savas and Y. Becerikli, 2020), malgré que les résultats de leur méthode prouvent que cette dernière peut maintenir une performance en temps réel et qu'elle est rapide en termes de temps, le modèle nécessite beaucoup de temps d'apprentissage. -Dans les travaux de (W. Zhang, Y. L. Murphey, T. Wang, and Q. Xu, 2015) le système de détection de bâillement proposé ne répond pas à l'exigence en temps réel qui a une grande complexité de calcul et ne répond pas aux défis tels que l'obstruction faciale (Masque de santé pendant la période de COVID). -Pour ce qui est des travaux de (A. George and A. Routray, 2016), et en dépit de leur bonne détection d'expression faciale le système prend en considération seulement la détection des yeux, lorsque le conducteur porte des lunettes les performances du modèle proposé seront dégradées. -Dans les travaux de (R. Huang, Y. Wang, and L. Guo, 2018), Considérant la perte de temps causée par le traitement des cœurs de projection dans cette méthode, la variation de luminance et le fait de se couvrir les yeux avec des lunettes pourraient encore être un problème.

3.4 Conclusion

Dans ce chapitre, nous avons présenté une revue de la littérature des principales contributions dans le domaine de la détection de mouvement et de la fatigue du conducteur, ces travaux ont été choisis par rapport à notre étude. Une synthèse de ces travaux a été présentée pour cerner la problématique et montrer l'impact de notre contribution face à leurs lacunes. Dans le chapitre suivant nous proposons notre approche qui consiste à former un système permettant la détection de la fatigue et de la distraction des conducteurs de transport public.

4.1 Introduction

Dans ce chapitre, nous présentons notre technique de détection de la fatigue et de la distraction des conducteurs en se basant sur le suivi des yeux, la position de tête ainsi que le mouvement de la bouche. L'objectif de ce travail est de construire un système robuste et performant pour une détection efficace de la fatigue et de la distraction. Afin d'atteindre ces objectifs, nous avons utilisé trois modèles, le premier pour détecter et analyser les yeux, le deuxième pour détecter et suivre le mouvement de la tête et le dernier modèle est utilisé pour analyser les mouvements de la bouche. L'utilisation de ces modèles nous a permis d'avoir de très bons résultats et cela dans des scénarios différents.

4.2 Les outils utilisés

4.2.1 Outils logiciels

A. Environnement

- **Pycharm** : est l'IDE le plus populaire pour Python, et comprend d'excellentes fonctionnalités telles qu'une excellente complétion et inspection du code avec un débogueur avancé et une prise en charge de la programmation Web et de divers cadres. Il est créé par la société tchèque JetBrains qui se concentre sur la création d'un environnement de développement intégré pour divers langages de développement Web [40].
- **Anaconda** : est un gestionnaire de packages, une distribution gratuite et open-source des langages de programmation Python et R pour le calcul scientifique qui vise à simplifier le package gestion et déploiement [41].

- **Jupyter-notebook** : Le bloc-notes Jupyter est un environnement informatique interactif qui permet aux utilisateurs de créer des documents de bloc-notes comprenant : du code en direct, des widgets interactifs, des tracés, du texte narratif, des équations, des images, de la vidéo[42].

B. Framework

- **Tensorflow** : Il fut créé par l'équipe Google Brain pour mener des recherches sur le ML et le Deep Learning. Il est considéré comme une version moderne de Theano [43]
- **Keras** : Le Framework le plus haut niveau, le plus convivial de la liste. Il permet aux utilisateurs de choisir si les modèles qu'ils construisent sont exécutés sur Theano ou TensorFlow. Il est écrit et entretenu par Francis Chollet, un autre membre de l'équipe Google Brain [43].

C. Bibliothèques utilisées

- **OpenCV (Open Source Computer Vision Library)** : est une bibliothèque de logiciels open source de vision par ordinateur et d'apprentissage automatique, elle possède plus de 2500 algorithmes optimisés, qui comprennent un ensemble complet d'algorithmes de vision par ordinateur et d'apprentissage machine à la fois classique et à la pointe de la technologie[43]
- **Dlib** : est une boîte à outils C++ moderne contenant des algorithmes d'apprentissage automatique et des outils pour créer des logiciels complexes en C++ pour résoudre des problèmes du monde réel. Il est utilisé à la fois dans l'industrie et dans les universités dans un large éventail de domaines, notamment la robotique, les appareils embarqués, les téléphones mobiles et les grands environnements informatiques haute performance. La licence open source de Dlib vous permet de l'utiliser dans n'importe quelle application, gratuitement [44].
- **Numpy** : est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion facilitée des tableaux de nombres, des fonctions sophistiquées (diffusion), on peut aussi l'intégrer le code C / C ++ et Fortran [45].
- **Matplotlib** : est une bibliothèque de traçage pour le langage de programmation Python et son extension mathématique numérique NumPy. Il fournit une API orientée objet permettant d'incorporer des graphiques dans les applications [46].

D. Langages de programmation

- **Python 3** : est le langage de programmation le plus utilisé dans le domaine du Machine Learning, du Big Data et de la Data Science [47].

4.2.2 Outils matériels

- **Caméra de surveillance** : en ce qui concerne nos tests, nous avons utilisé les caméras des pc portables qui fournissent des photos d'une qualité d'image acceptable .

4.3 Notre approche

Afin de mettre en œuvre notre système de détection de la fatigue et de la distraction, nous avons proposé l'architecture illustrée dans la Figure 4.1

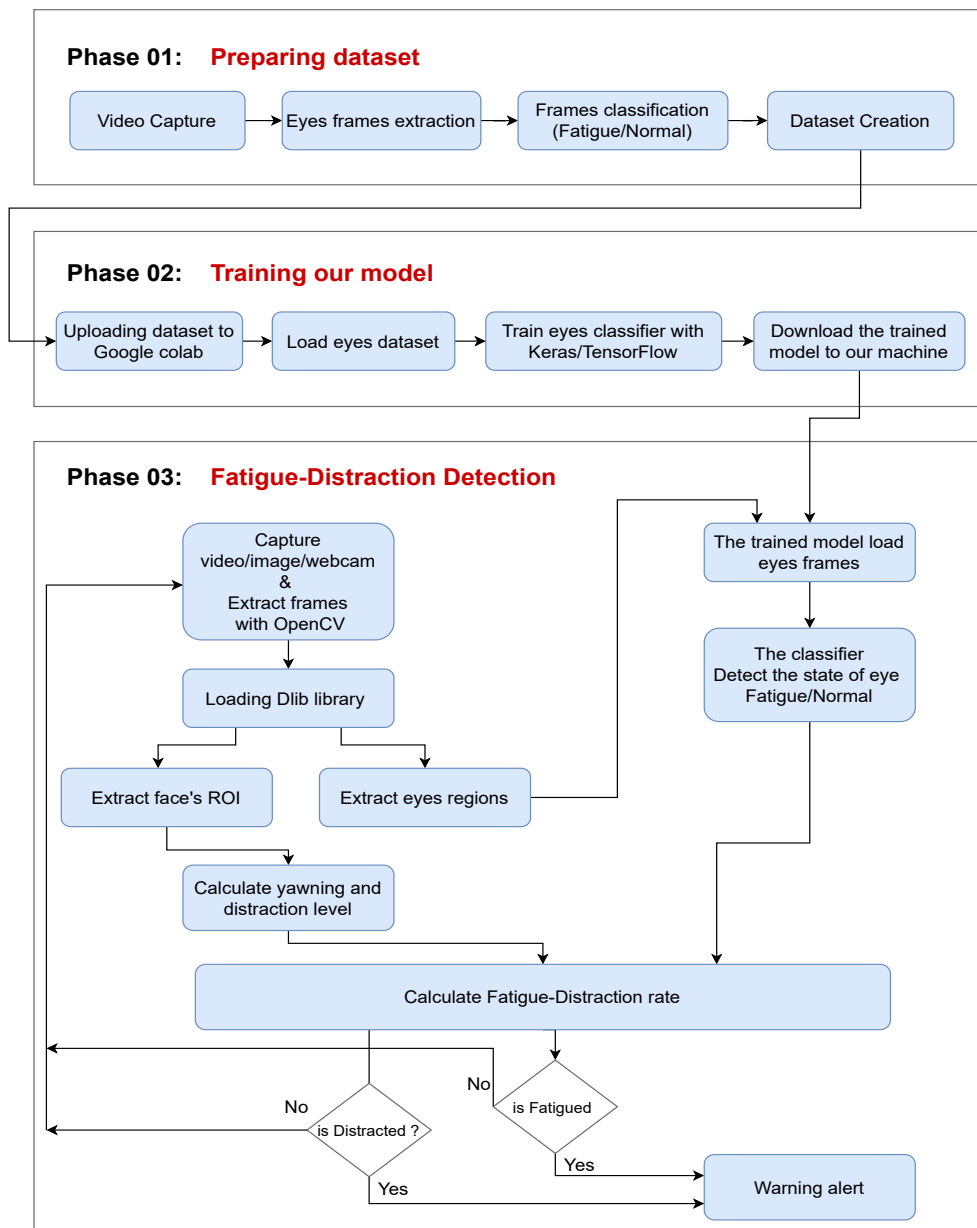


FIGURE 4.1 – Schéma global.

4.3.1 Préparation du dataset

Le dataset utilisé pour notre modèle est créé par nous-mêmes. Pour créer l'ensemble de données, nous avons écrit un script qui capture les images des yeux à partir d'une caméra en utilisant la bibliothèque openCV et les stocke sur le disque local. Après cela, nous les avons classés et séparés en deux catégories : « Normal » et « Fatigue », avec le nettoyage et la suppression des images indésirables qui n'étaient pas nécessaires à la construction du modèle. Notre dataset comprend environ 550 images, 300 images des yeux fermés et 250 images des yeux ouverts sous différentes directions de regard (voir la Figure 4.2).

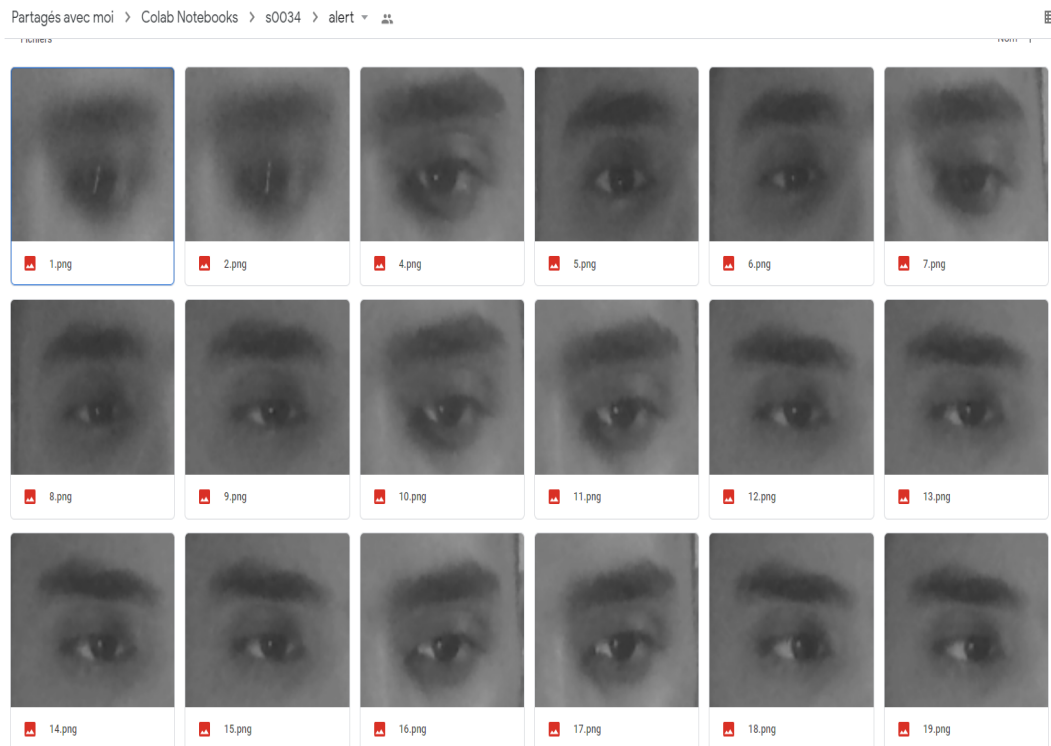


FIGURE 4.2 – 250 images des yeux ouverts.

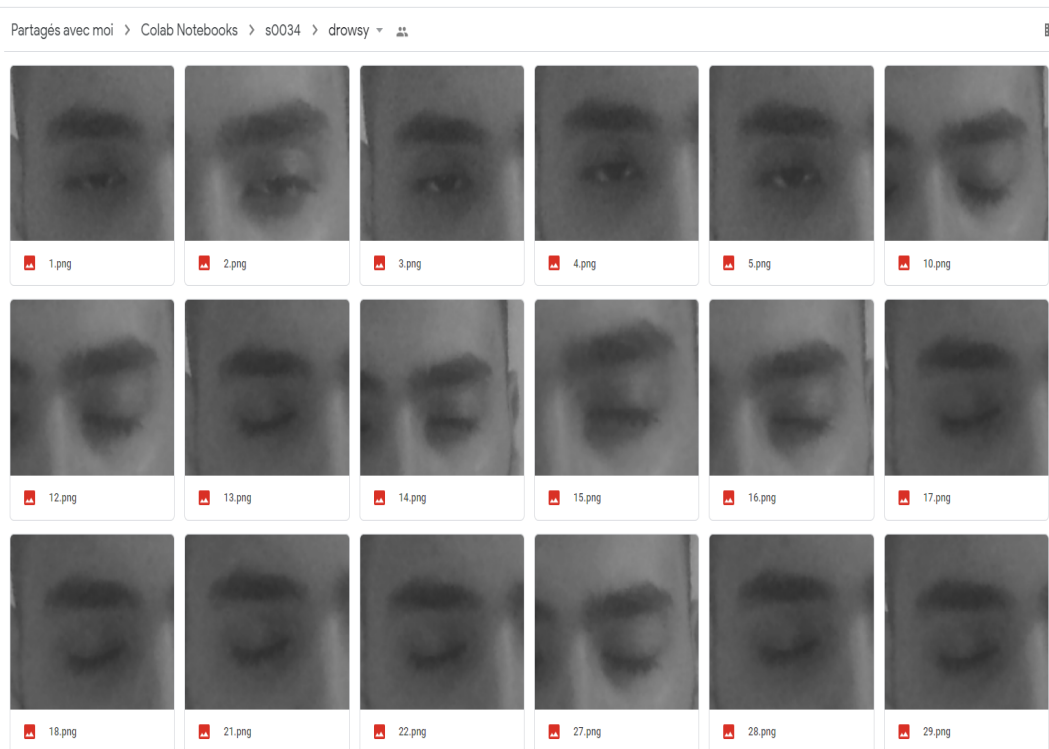


FIGURE 4.3 – 300 images des yeux fermés.

4.3.2 Phase d'apprentissage (training our model)

Après avoir capturé et classé notre ensemble de données, nous devons le télécharger sur Google Drive, puis avec google colab, nous exécuterons l'ensemble de code python à l'aide des deux bibliothèques tensorflow et keras pour entraîner notre modèle. Les étapes suivantes expliquent le fonctionnement de l'entraînement.

A. **Préparation** Dans cette partie, nous avons créé deux catégories qui se réfèrent aux deux fichiers des images des yeux fermés et ouverts. Nous avons redimensionné et converti chaque image à 50x50 pixels et en couleur noir et blanc pour optimiser leurs tailles. Le pseudo-code suivant montre plus de détails (voir la Figure 4.4)

```
Algorithm 1: TrainingAlgorithm
Import      numpy, matplotlib.pyplot, os, cv2
Input :     Path, Dataset_images
Output :    TrainingData[]

Begin
  Path = getPath()
  TrainingData = []
  Categories = [ "Fatigue" , "Normal" ]
  For i in Categories do
    Files = getFiles(Path+Categories[i])
    For file in Files do
      Image = ReadImg(file)
      Img = ResizImg(image)
      Grey = Color_RGB2Grey(Img)
      TrainingData.append(Grey)
    End
  End
End
```

FIGURE 4.4 – Training algorithm.

B. **La normalisation** Dans cette étape, nous avons utilisé deux structures de données Y : représente les étiquettes (Labels) correspondantes (1 pour l'œil fermé, 0 pour l'œil ouvert). X : est une liste de matrices où chaque d'eux représente les valeurs des pixels d'une image (entre 0 et 255). Convertissez l'image en un vecteur afin qu'elle puisse être entrée dans le modèle. Mettez également à l'échelle les données en les divisant par 255 (voir la Figure 03). puis nous divisons chaque valeur par 255 pour les rendre entre 0 et 1 (normalisation), voir la figure 4.5

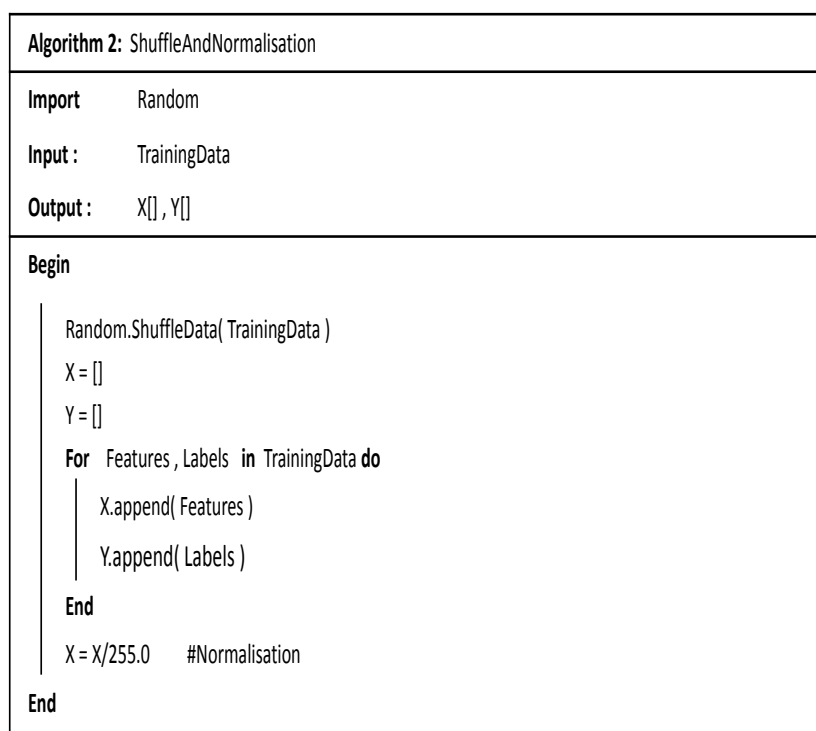


FIGURE 4.5 – Shuffle And Normalisation algorithm.

C. **La création de notre modèle :** Le modèle que nous avons utilisé est construit avec Keras en utilisant un réseau de neurones convolutif (CNN). Un CNN est un type spécial de réseau de neurones profonds qui fonctionne extrêmement bien à des fins de classification d'images. Il se compose essentiellement d'une couche d'entrée , d'une couche de sortie et des couches cachées (Hidden layers) qui peut avoir plusieurs nombres de couches ,voir la figure 4.6. Une opération de convolution est effectuée sur ces couches à l'aide d'un filtre qui effectue une multiplication matricielle 2d entre la couche et le filtre.

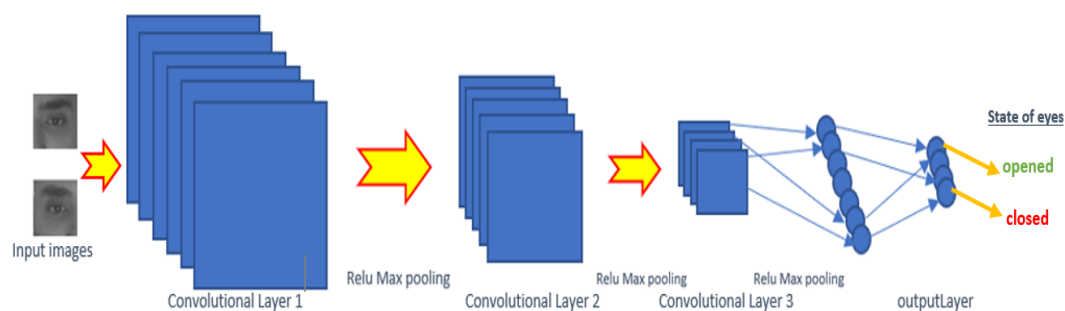


FIGURE 4.6 – Architecture de notre modèle.

Nous avons utilisé les paramètres suivants lors de l'implémentation de notre modèle CNN :

- Categorical-cross entropy : permet la réduction de l'erreur tout en utilisant l'algorithme

de la descente de gradient. 24 Expérimentation

- Epoch : désigne le nombre d'itération dans notre base de données. Optimiser : permet de réduire le poids des erreurs.
- Loss : désigne le taux d'erreur.
- Accuracy : désigne le taux de précision lors de la validation .
- Val Accuracy : désigne le taux de précision dans la phase d'apprentissage.
- History : elle permet de donner au modèle toutes les images dont elle aura besoin en fonction du nombre d'itération (epochs) défini, dans le but de réduire le taux d'erreur.

Notre modèle CNN se compose des couches suivantes :

- * Couche convolutive ; 256 nœuds, taille de noyau 3
- * Couche convolutive ; 256 nœuds, taille de noyau 3
- * Couche dense ; 64 nœuds.

Dans toutes les couches, une fonction d'activation 'Relu' est utilisée sauf la couche de sortie dans laquelle nous avons utilisé Softmax. Le pseudo-code suivant (la Figure 4.7) montre comment affecter ces couches dans notre modèle.

Algorithm 3: CreatingModel	
Import	TensorFlow, Keras
Input :	X[], Y[]
Output :	Model
Begin	
Model = Sequential()	
Model.add(Conv2D_Layer(256, (3,3)))	<i>#Conv2D layer, 256 nœuds , filtre 3x3</i>
Model.add(Activation_Layer('relu'))	
Model.add(MaxPooling2D_Layer((2,2))	
Model.add(Flatten_Layer())	
Model.add(Dense_Layer())	
Model.Compilation(loss='binary_crossentropy', optimizer='adam', metrics='accuracy')	
Model.fit(X, Y, batch_size=32, epochs=24, validation_split=0.3)	
Model.save(Path)	
End	

FIGURE 4.7 – Pseudocode de la création de notre modèle.

Après 24 périodes d'entraînement (epochs), nous avons obtenu une précision de 0,9859, la Figure 4.8 présent le résumé de notre modèle. Et enfin, nous avons enregistré les résultats de l'exécution (le modèle entraîné) dans Google Drive pour le télécharger et l'utiliser plus tard.

```

Epoch 22/24
20/20 [=====] - 19s 974ms/step - loss: 0.0500 - accuracy: 0.9844 - val_loss: 0.0271 - val_accuracy: 0.9964
Epoch 23/24
20/20 [=====] - 19s 980ms/step - loss: 0.0380 - accuracy: 0.9875 - val_loss: 0.0161 - val_accuracy: 0.9964
Epoch 24/24
20/20 [=====] - 19s 979ms/step - loss: 0.0319 - accuracy: 0.9859 - val_loss: 0.0208 - val accuracy: 0.9964

```

FIGURE 4.8 – Notre modèle pendant l’entraînement.

D. **Les résultats de l’entraînement et de la validation**D’après la figure 4.9, nous remarquons que les précisions de l’entraînement et de la validation augmentent progressivement jusqu’à ce qu’elles atteignent un certain seuil, indiquant qu’aucune autre amélioration n’est apportée et entraînant la fin de l’entraînement. De même, dans la figure 4.10 nous observons que la perte de validation et de formation diminue au fur et à mesure que le modèle apprend, jusqu’à ce qu’il se termine finalement lorsque le modèle commence à se sur-ajuster.

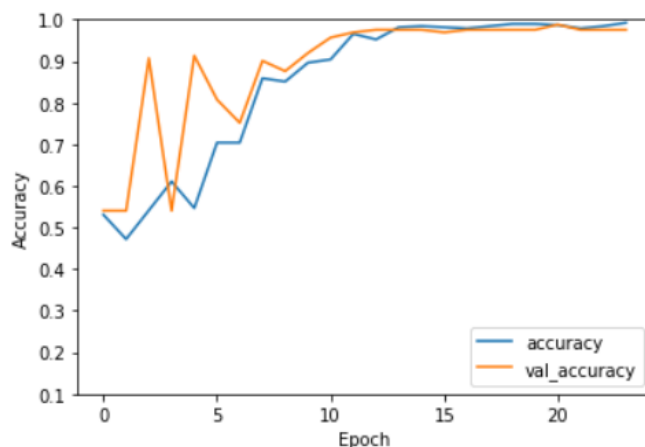


FIGURE 4.9 – La précision de notre modèle.

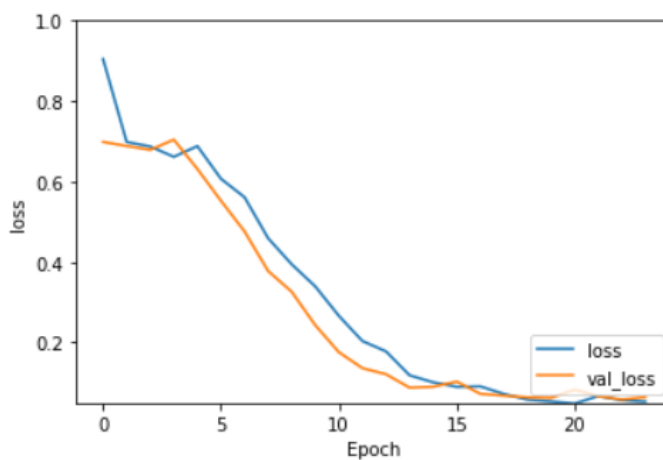


FIGURE 4.10 – La perte de notre modèle.

4.3.3 Détection de la fatigue et de la distraction

Après avoir créé et entraîné notre modèle CNN et le téléchargé sur notre machine locale, il est maintenant nécessaire de réaliser notre application complète.

A. L'utilisation d'OpenCV pour accéder à la webcam

Nous avons déjà utilisé cette librairie pour de l'extraction des yeux précédemment, et maintenant nous l'utilisons pour capturer une vidéo en direct à partir de la webcam. Alors pour accéder à la webcam, nous avons fait une boucle infinie qui va capturer chaque image. Nous utilisons la méthode fournie par OpenCV, `cv2.VideoCapture(0)` pour accéder à la caméra et définir l'objet de capture (`cap`). `cap.read()` lira chaque frame et nous stockons l'image dans une variable `frame`.

B. Détecter le visage dans l'image et créer une région d'intérêt (ROI)

Pour détecter le visage dans l'image, nous devons d'abord convertir l'image en niveaux de gris par l'algorithme OpenCV pour que la détection d'objet prenne des images grises en entrée. Nous n'avons pas besoin d'informations de couleur pour détecter les objets. Nous utiliserons la bibliothèque DLib pour détecter les visages. Ces deux lignes de code sont utilisées pour définir notre classificateur :

```
hog_face_detector = dlib.get_frontal_face_detector()
dlib_facelandmark = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
```

Ensuite, nous effectuons la détection en utilisant :

```
faces = hog_face_detector (gris)
```

Il renvoie un vecteur de 68 repères faciaux (landmarks) autour du visage, des yeux, du nez et de la bouche (voir la Figure 4.11) avec des coordonnées (x,y), que nous pouvons extraire à l'aide du code suivant :

```
x = face_landmarks.part(n).x
y = face_landmarks.part(n).y
```

Maintenant, nous pouvons parcourir les visages et dessiner des zones de contour pour chaque visage. Et ainsi nous devons extraire les images des yeux pour les tester par notre modèle pré-entraîné.

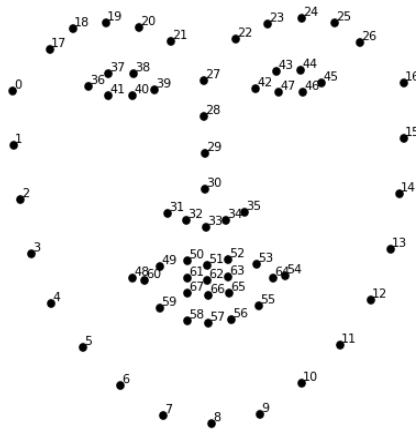


FIGURE 4.11 – Repères faciaux créés par Dlib.

C. La détection/classification de l'état des yeux avec notre modèle CNN

Tout d'abord, nous chargeons notre modèle en utilisant la bibliothèque Keras avec le code

```
model = keras.models.load_model('model')
```

Ensuite, nous devons extraire uniquement les données de l'œil de l'image complète. Cela peut être réalisé en extrayant les points de repère autour de la région des yeux, puis nous pouvons extraire l'image de l'œil de l'image d'origine avec ce code.

```
eye = gray [y :y + h , x : x + w ]
```

La variable *eye* ne contient que les données d'image de l'œil. Cela sera introduit dans notre modèle de classificateur CNN qui prédit si les yeux sont ouverts ou fermés. Nous devons effectuer certaines opérations car le modèle a besoin des bonnes dimensions pour commencer. Pour cela. Le code suivant redimensionne les images des yeux extraites précédemment, puis, on va normaliser nos données pour une meilleure convergence (Toutes les valeurs seront comprises entre 0-1), pour qu'elles soient prêtes pour la prédiction.

```
new_array = cv2.resize(eye,(50,50))
X_input = np.array(new_array).reshape(-1,50,50,1)
X_input.shape
X_input = X_input/255.0
ynew = model.predict_classes(X_input)
```

Maintenant, nous prédisons l'état des yeux avec notre modèle

```
prediction = model.predict(X_input)
```

Si la valeur de *prediction = 1*, cela indique que les yeux sont ouverts et le conducteur est dans un état normal, si la valeur de *prediction = 0* alors, cela indique que le conducteur est un état fatigué.

D. La détection de l'ouverture de la bouche (POM)

Afin de faire la distinction entre le bâillement et la parole, nous avons estimé une valeur seuil qu'une bouche peut atteindre dans le cas du bâillement, qui peut être utilisée pour évaluer l'état de la bouche du conducteur. Selon les points caractéristiques de la bouche définis par Dlib voir la Figure 4.12 :

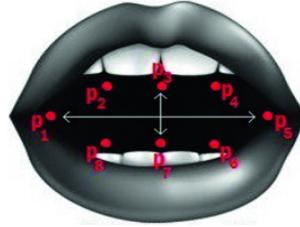


FIGURE 4.12 – Les points repères de la bouche créés par Dlib.

On a défini un terme appelé POM (Percentage opening mouth), lorsque la bouche du conducteur est dans des états différents, on peut remarquer le changement de variation entre les points (P3 et P7) et entre (P1 et P5) . De ce point de vue, on peut calculer cette variation avec l'équation suivante :

$$\begin{aligned}
 POM &= \text{Dist}(p3,p7)*100/\text{Dist}(p1,p5) \\
 p3 &= \text{face_landmarks.part}(62).y \\
 p7 &= \text{face_landmarks.part}(66).y \\
 p1 &= \text{face_landmarks.part}(60).x \\
 p5 &= \text{face_landmarks.part}(64).x \\
 yawning_rate &= \text{int}((y66-y62)*100/(x64-x60))
 \end{aligned}$$

Les résultats montrent que le bâillement commence si la valeur *yawning_rate* est supérieure à 50%.

E. La détection de mouvement de la tête (Distraction)

Dans cette partie, nous avons calculé la distraction du conducteur en suivant les étapes suivantes :

$$\begin{aligned}
 x5 &= \text{face_landmarks.part}(0).x \\
 x6 &= \text{face_landmarks.part}(16).x \\
 xx &= \text{int}((x5 + x6)/2)
 \end{aligned}$$

La valeur xx présente la ligne rouge verticale dans la Figure 4.13, cet ligne rouge est situé dans le centre de rectangle

$$\begin{aligned}
 x2 &= \text{face_landmarks.part}(39).x \\
 x3 &= \text{face_landmarks.part}(42).x
 \end{aligned}$$

Les valeurs x_2 et x_3 présentent les lignes vertes verticales gauche et droit respectivement illustrés dans la Figure 4.13. Le système classe le conducteur comme distrait si la ligne rouge ne se pose pas entre les 2 lignes vertes, c'est-à-dire que le conducteur n'est pas en train de surveiller la route ensuite lorsque le conducteur est distrait le système augmente le score de distraction qu'on l'avait nommé distraction-meter et lorsqu'il atteint la valeur 60 (environ 8s) l'image est encadré en rouge et un son d'alarme se déclenche pour avertir le conducteur ,voir la figure suivante.

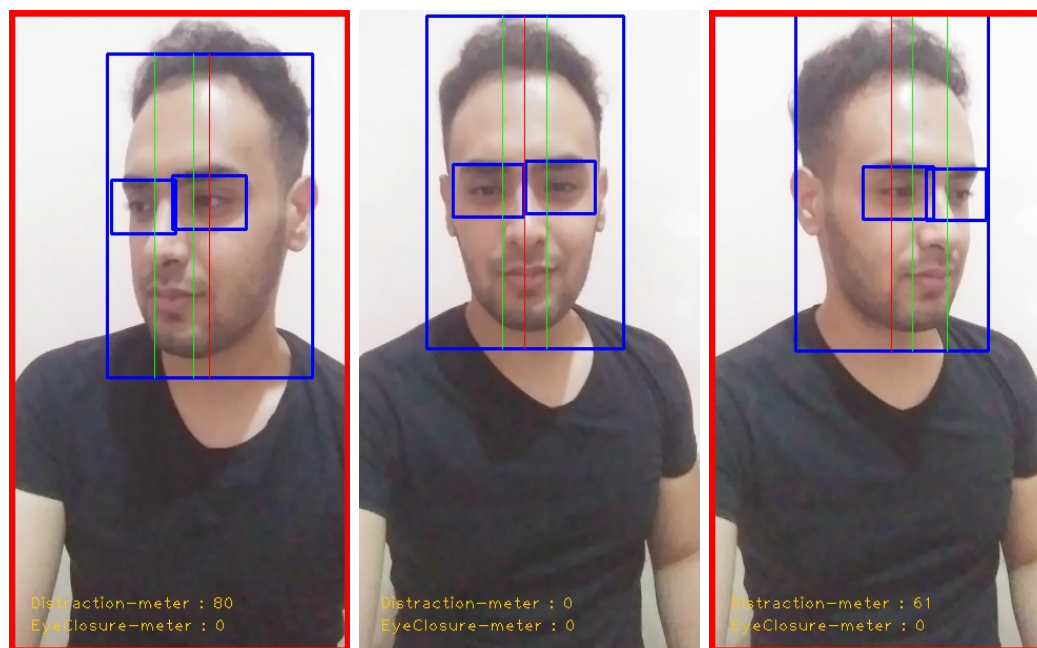


FIGURE 4.13 – La détection de la distraction.

4.4 Analyse et tests

On a effectué l'exécution de notre système sur 3 scénarios différents pour étudier son efficacité, on a vu le cas où le chauffeur est dans l'état naturel c'est à dire lorsque qu'il porte rien, le deuxième scénario c'est lorsqu'il porte des lunettes, et le dernier cas c'est lorsque le conducteur porte un masque chirurgical, et chaque scénario est divisé en trois partie car on doit simuler le conducteur dans l'état normal, lorsqu'il est fatigué et lorsqu'il est distrait. Notre contribution vise également à détecter la fatigue et la distraction dans la conjoncture actuelle lors de la propagation du covid 19, où notre système peut fonctionner même lorsque le conducteur porte une bavette.

4.4.1 Scénario 1 : Le conducteur dans l'état naturel (il ne porte rien)

A. Etat normal

La Figure 4.14) représente la variation de la précision de détection de la vigilance du conducteur en fonction de nombre d'images étudié , c'est à dire lorsque le conducteur est dans l'état normal, non fatigué et non distrait la détection faciale est très précise et stable lorsque le nombre de frames s'approche de la valeur 1000 la précision atteint 96

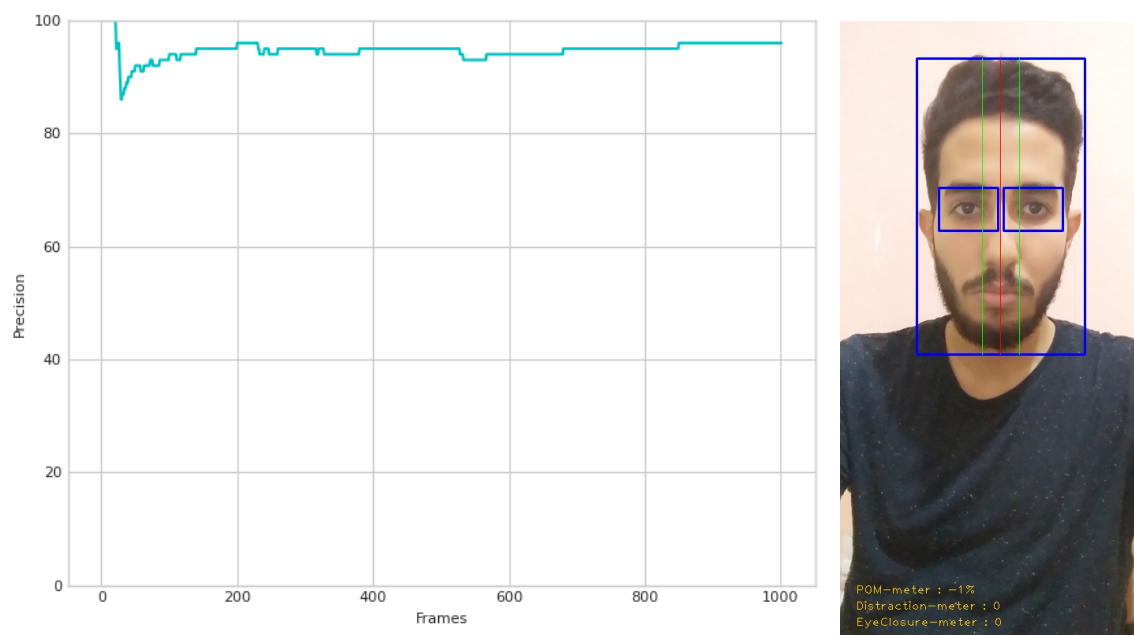


FIGURE 4.14 – La précision de notre système dans un état naturel et normal

B. Etat fatigué

La Figure 4.15 représente la variation de la précision de détection de la Fatigue du conducteur en fonction du nombre d'images étudiées. On a simulé dans cette étape le cas où le conducteur est fatigué. La précision dans ce cas est déséquilibrée jusqu'à l'atteinte de la valeur de 500 frames de test elle se stabilise vers la valeur de 86% .

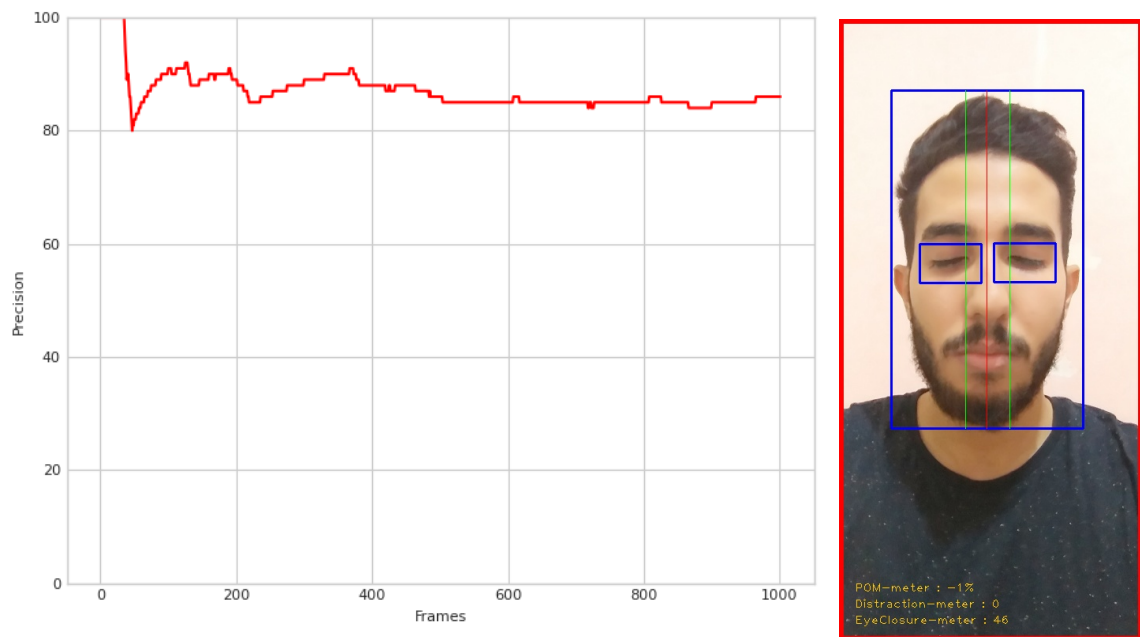


FIGURE 4.15 – La précision de notre système dans un état naturel et fatigué

C. Etat distrait où entrain de bailler

Le troisième graphe (figure 5) représente la variation de la précision de détection de la distraction et du bâillement conducteur en fonction du nombre d'images étudiées, dans cette phase on simule le cas où le conducteur est distrait où il est entrain de bailler (voir la Figure 4.16). La précision dans ce cas donne de très bon résultat avec n'importe quel nombre d'images.

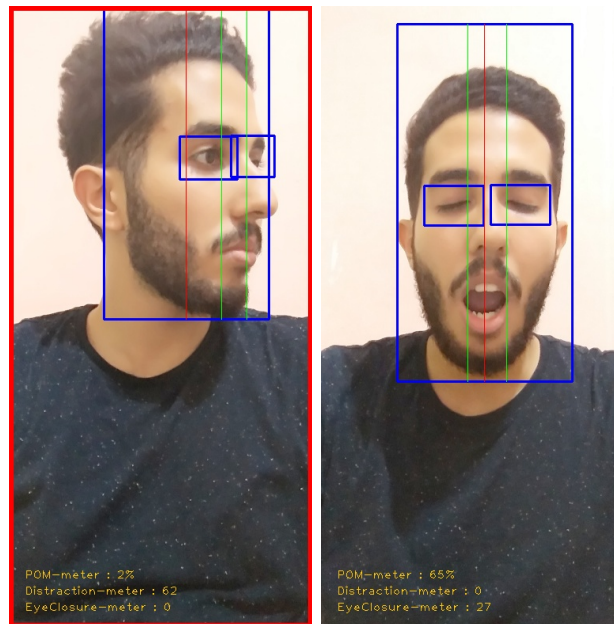
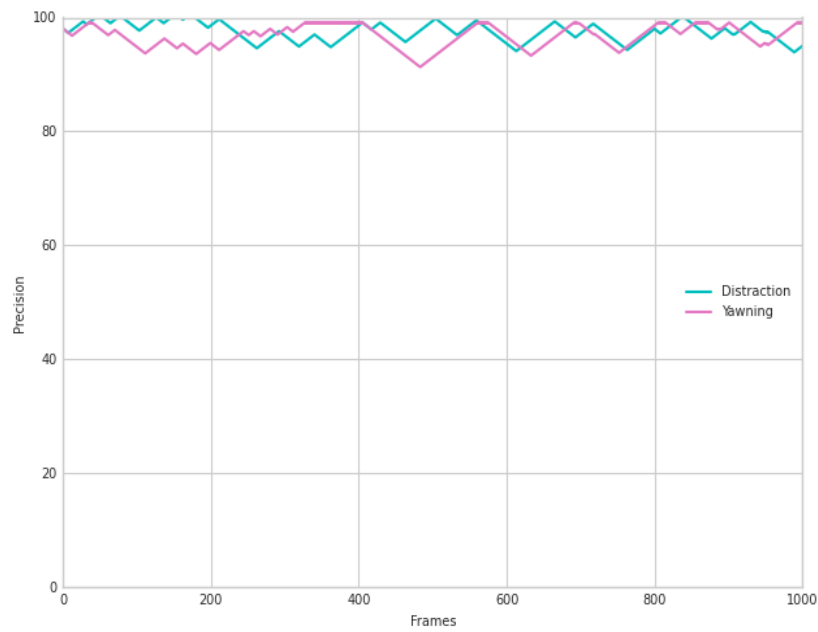


FIGURE 4.16 – La précision de notre système dans un état naturel et distrait ou dans un état normale et entrain de bailler

4.4.2 Deuxième scénario : Le conducteur porte des lunettes

A. État normal

Lorsque le conducteur porte des lunettes, le taux de Précision lorsqu'il est dans l'état normal va être diminué en le comparant au cas où il ne porte rien. Par contre, il augmente en fonction du nombre d'images étudiées et se stabilise vers la valeur de 80% lorsqu'on atteint 1000 images et c'est un très bon résultat en le comparant aux autres méthodes de l'état de

l'art. Le résultat d'exécution en temps réel est montré dans la Figure 4.17.

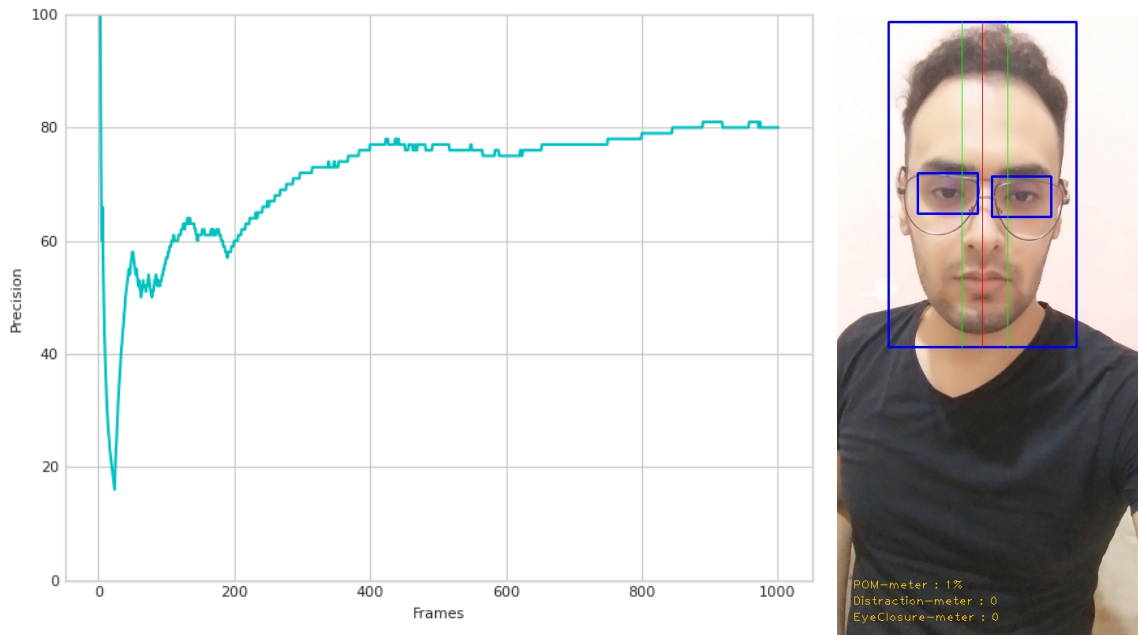


FIGURE 4.17 – La précision de notre système si le conducteur porte des lunettes et il est dans un état normale.

B. État fatigué

Le deuxième graphe (la Figure 4.18) représente la variation de la précision de détection de la Fatigue qui va être stable même en augmentant le nombre de frames.

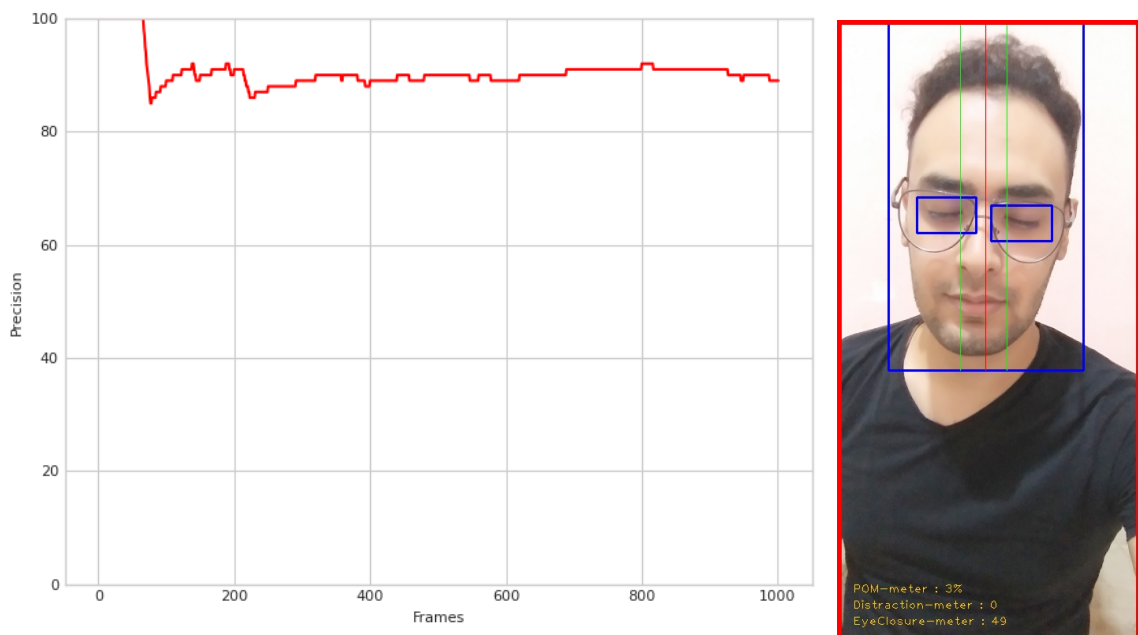


FIGURE 4.18 – La précision de notre système si le conducteur porte des lunettes et il est fatigué.

C. État distrait où entrain de bailler

Le taux de précision dans le dernier graphe (la Figure 4.19) va être relativement élevé car la bibliothèque Dlib utilisée dans notre système est très performante en termes de détection faciale même si le conducteur porte des lunettes , le résultat d'exécution est montré dans la figure

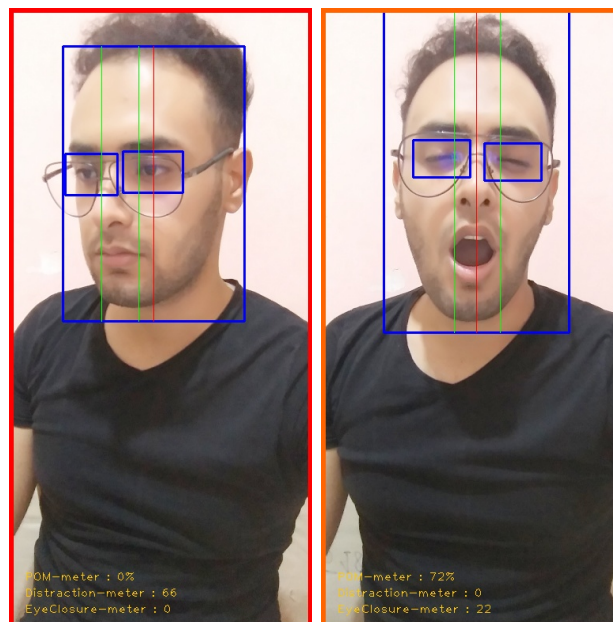
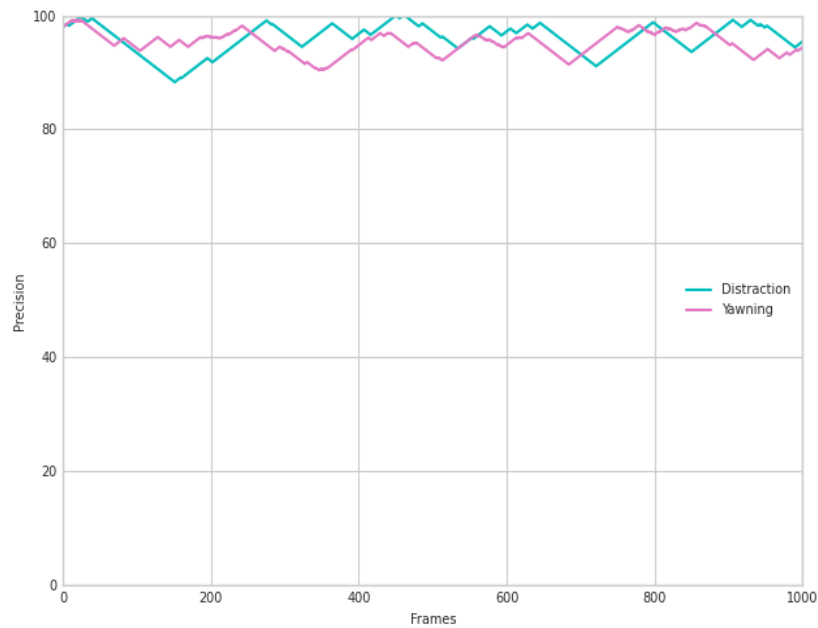


FIGURE 4.19 – La précision de notre système si le conducteur porte des lunettes et il est distrait.

4.4.3 Troisième scénario : le conducteur porte un masque chirurgical

Lorsque le conducteur porte un masque chirurgical, le taux de précision sera déséquilibré et dépend énormément de la façon dont on porte le masque , si la bavette est bien acérée sur le visage le système détecte les caractéristiques faciales facilement sinon il ne détectera rien.

A. État normal

Dans le graphe montré dans la Figure 4.20 On remarque que la précision est dégradée lorsque le nombre d'images testé et relativement petit mais elle s'améliore lorsque le nombre de frames s'augmente.

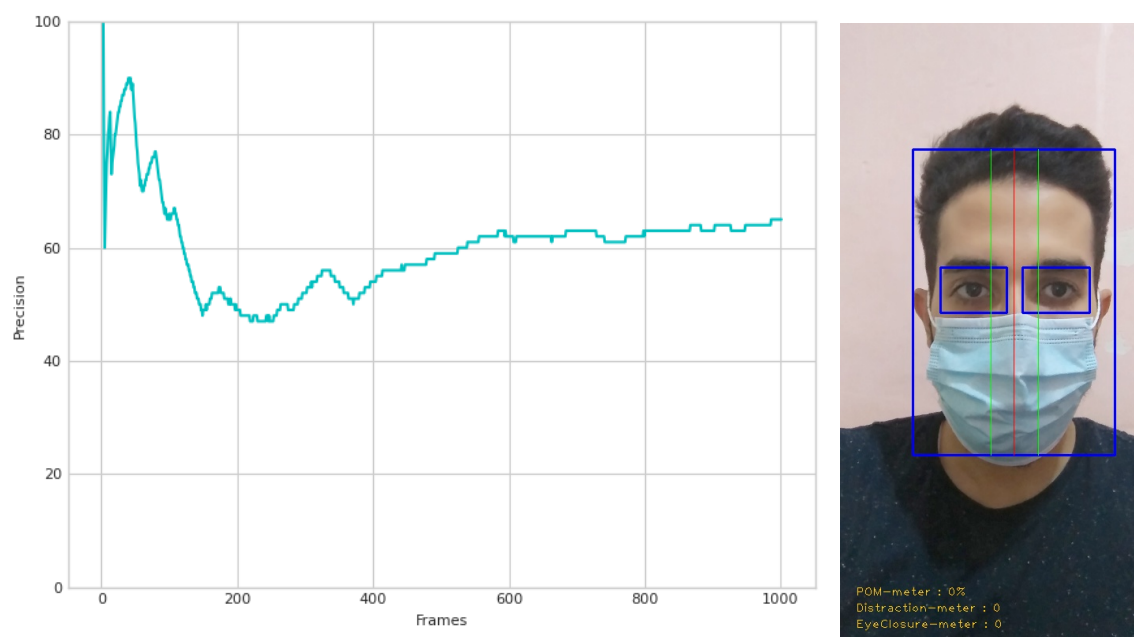


FIGURE 4.20 – La précision de notre système si le conducteur porte un masque et il est dans un état normale.

B. État fatigué

Le deuxième graphe (la Figure 4.21) représente la variation de la précision de détection de la Fatigue qui va se stabiliser vers la valeur de 70%, lorsqu'on augmente le nombre de frames.

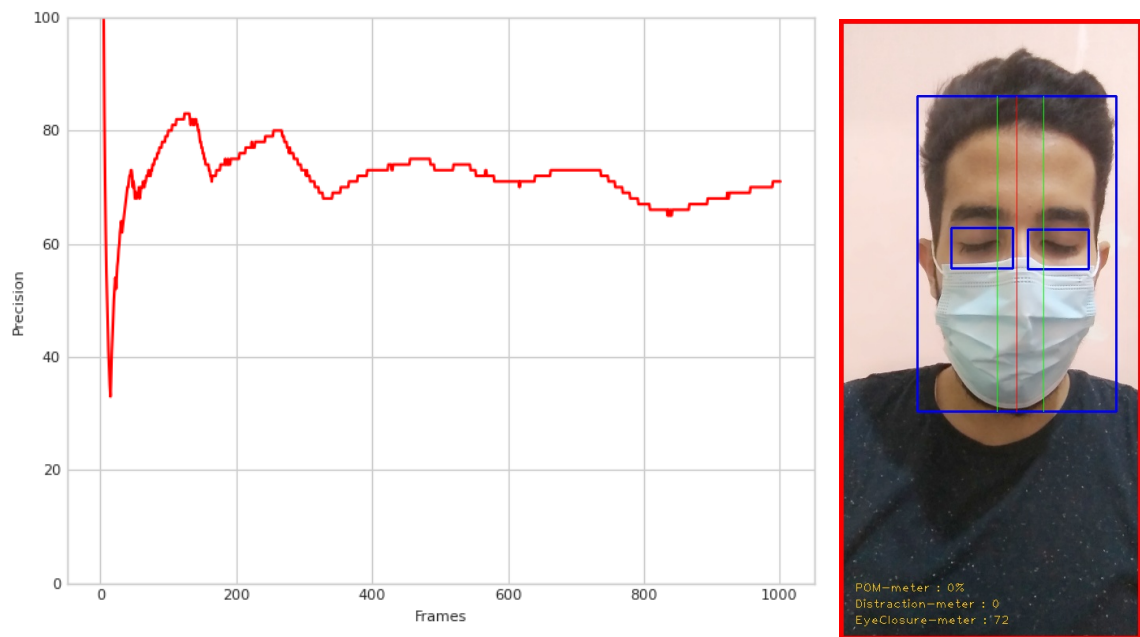


FIGURE 4.21 – La précision de notre système si le conducteur porte un masque et il est fatigué.

C. État distrait où entrain de bâiller

Le taux de précision de la distraction dans le dernier graphe va être relativement élevé car il prend en compte seulement la position de la tête qui ne va pas être influencée par le port du masque, par contre le taux de précision du bâillement va être nul car la bouche du conducteur est totalement couverte avec la bavette voir figure (voir la Figure) .

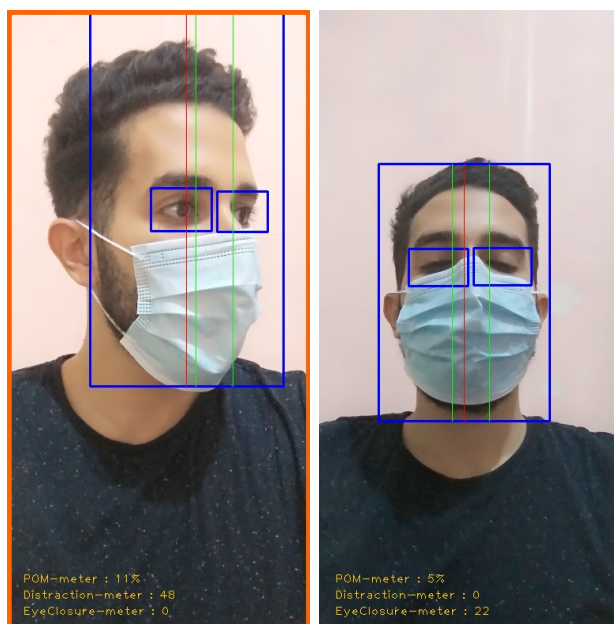
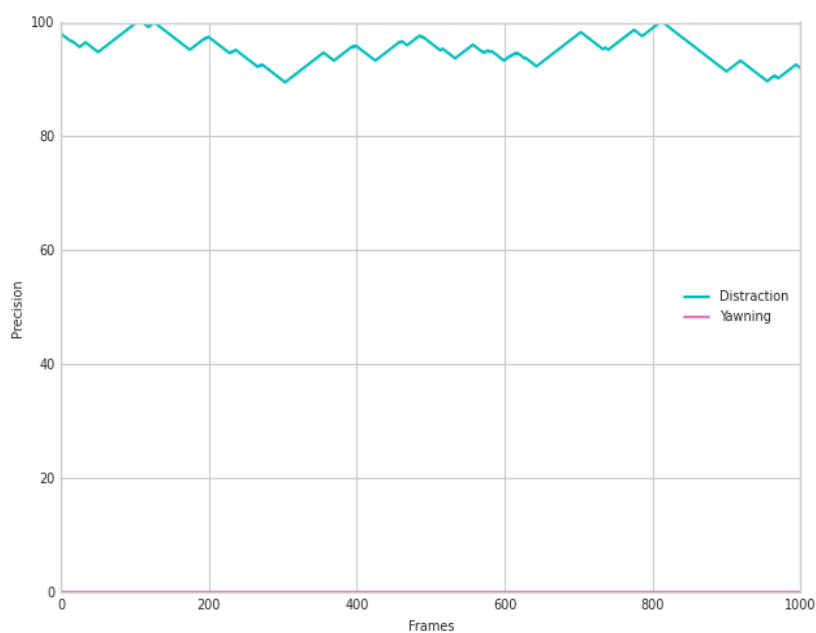


FIGURE 4.22 – La précision de notre système si le conducteur porte des lunettes et il est distrait.

4.5 Conclusion

La fatigue et la distraction au volant est l'un des principaux facteurs à l'origine des accidents de la circulation. Nos visages contiennent beaucoup d'informations utiles, la surveillance de l'état des yeux, de la position de tête et du bâillement du conducteur présentent des critères efficaces pour détecter la fatigue et la distraction. Dans ce contexte et dans ce chapitre, on a présenté notre protocole de détection. Cette approche a permis de détecter la fatigue et la distraction en temps réel et a donné des résultats efficaces en termes de précision. L'approche ne présente qu'un

premier pas pour d'autres travaux de détection de comportement général du conducteur qui se basent sur les techniques de l'intelligence artificielle. Dans le chapitre suivant on va clôturer notre mémoire avec une conclusion générale de notre projet .

CHAPITRE 5

CONCLUSION GÉNÉRALE ET PERSPECTIVES

De nos jours, le transport fait partie intégrante de notre vie quotidienne. Nous l'utilisons pour répondre à nos besoins, mais malgré toutes les facilités offertes, les accidents de la route restent l'un de ses plus gros inconvénients, car il occupe la cinquième place dans les causes de décès. C'est pourquoi de nombreux chercheurs et développeurs se sont dirigés vers la recherche des causes de ces accidents et de trouver des moyens pour les réduire. Ils ont conclu que la principale cause de ces accidents est la fatigue et la distraction pendant la conduite. Ce qui nous a motivé à étudier ce sujet intéressant. Notre objectif est en fait de proposer un nouveau modèle de détection de comportement du conducteur de transports public en temps réel afin de diminuer le nombre d'accidents et de décès. Pour cela, nous avons utilisé un ensemble d'outils algorithmiques et logiciels, ainsi que des techniques d'intelligence artificielle. Notre système de détection de la fatigue et de la distraction du conducteur est basé sur un réseau de neurones pour la détection des différents mouvements de yeux, et sur des techniques pour le suivi de mouvement et de de la tête et de la bouche. Le système proposé peut être exécuté en temps réel, exploité facilement dans les moyens de transports et il est simple et facile à mettre en œuvre. Les résultats de simulation et des tests effectués ont montré les bonnes performances de notre système et cela dans différentes conditions et pour différents scénarios.

Ce mémoire de fin d'études nous a permis de :

- Approfondir nos connaissances théoriques et pratiques déjà acquises.
- Maîtriser les nouvelles techniques et compléter notre initiale pour atteindre ainsi un niveau de perfection supérieur et pouvoir apprendre d'autres nouveautés.
- Construire des savoirs et des savoir-faire dans le domaine de vision par ordinateur plus précisément la détection de mouvements qui est un thème d'actualité.
- Comprendre les différentes étapes pour créer l'algorithme de Deep Learning en python .

Comme perspectives à notre travail, nous proposons de :

- Améliorer notre système de détection de la distraction et de la fatigue quand le conducteur porte un masque médical ou/et des lunettes.
- Utiliser d'autres méthodes d'intelligence artificielle pour la détection du comportement.
- Comparer les performances de notre méthode avec d'autres systèmes.
- Utiliser d'autres paramètres qui peuvent améliorer la précision de notre système tels que l'angle de braquage du volant, le taux d'accélération et de freinage.

- [1] Akin M., Kurt M., Sezgin N., Bayram M., “Estimating vigilance level by using EEG and EMG signals”, *Neural Comput. Applications*, 17, pp.227–236,(2009)
- [2] Yang,M.H., Ahuja,N., “Gaussian Mixture model for human skin colour and its application in image and video databases”, *Proceedings of SPIE : Conference on Storage and Retrieval for Image and Video Databases*, vol. 3656, pp. 458– 466,(1999).
- [3] Karamjeet Singh, Rupinder Kaur,”Physical and Physiological Drowsiness Detection Methods”, *International Journal of IT, Engineering and Applied Sciences Research*, pp.35-43, Volume 2, (2013).
- [4] A. Doshi and M.M. Trivedi. On the roles of eye gaze and head dynamics in predicting driver’s intent to change lanes. *Intelligent Transportation Systems, IEEE Transactions on*, 10(3) :453–462, Sept 2009.
- [5] Jane C. Stutts, Donald W. Reinfurt, Loren Staplin, and Eric A. Rodgman. *The role of driver distraction in traffic crashes. AAA Foundation for Traffic Safety*, 2001.
- [6] S. Kaplan, M. A. Guvensan, A. G. Yavuz, and Y. Karalurt, “Driver behavior analysis for safe driving : A survey,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3017–3032, 2015.
- [7] E. Wahlstrom, O. Masoud, and N. Pap a nikolopoulos. Vision-based methods for driver monitoring. In *Proceedings of the IEEE Intelligent Transportation Systems*, volume 2, pages 903–908, Shanghai, China, October 2003.
- [8] T. Kato, T. Fujii, and M. Tanimoto. Detection of driver’s posture in the car by using a far infrared camera. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 339–344, Parma, Italy, June 2004.

-
- [9] Xuetao Zhang, Nanning Zheng, Fan Mu, and Yongjian He. Head pose estimation using isophote features for driver assistance systems. In Proceedings of the IEEE Intelligent Vehicles Symposium, pages 568–572, Xi'an, China, June 2009.
- [10] P. Watta, S. Lakshmanan, and Yulin Hou. Nonparametric approaches for estimating driver pose. *Vehicular Technology, IEEE Transactions on*, 56(4) :2028–2041, July 2007.
- [11] Ji, Q., Zhu, Z. and Lan, P. : Real-time Nonintrusive Monitoring and Prediction of Driver Fatigue. *Vehicular Technology, IEEE Transactions on*, 53(4), pp. 1052–1068 (2004).
- [12] Unknown, “Fatigue statistics,” Gov.au. [Online]. Available : <http://www.tac.vic.gov.au/road-safety/statistics/summaries/fatigue-statistics>. [Accessed : 27-Jun-2021].
- [13] R. Knippling and P. Rau, “PERCLOS : A Valid Psychophysiological Measure of Alertness As Assessed by Psychomotor Vigilance,” vol. 31, no. 5, pp. 1237–1252, 1998.
- [14] J. Cech and T. Soukupova, “Real-Time Eye Blink Detection using Facial Landmarks,” 21st *Comput. Vis. Winter Work.*, 2016.
- [15] A. Ben-Hur and J. Weston, “A user’s guide to support vector machines,” *Methods Mol. Biol.*, vol. 609, pp. 223–239, 2010.
- [16] N. Vladimir, *The nature of statistical learning theory.*, vol. 8, no. 6. 1997.
- [17] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A Training Algorithm for Optimal Margin Classifiers,” *Proc. fifth Annu. Work. Comput. Learn. theory*, pp. 144–152, 1992.
- [18] L. E. Baum and T. Petrie, “Statistical Inference for Probabilistic Functions of Finite State Markov Chains,” *Statistics (Ber)*., pp. 50– 60, 1966.
- [19] T. S. Prasad and N. R. Kisore, “Application of Hidden Markov Model for classifying metamorphic virus,” *Souvenir 2015 IEEE Int. Adv. Comput. Conf. IACC 2015*, pp. 1201–1206, 2015.
- [20] B.-J. Yoon, “Hidden Markov Models and their Applications in Biological Sequence Analysis,” *Curr. Genomics*, vol. 10, no. 6, pp. 402–415, 2009.
- [21] “CS231n Convolutional Neural Networks for Visual Recognition.” [Online]. Available : <http://cs231n.github.io/convolutionalnetworks/>.
- [22] U. Karn, “An Intuitive Explanation of Convolutional Neural Networks,” 2016. [Online]. Available : <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>.
- [23] Y. LeCun and Y. Bengio, “The Handbook of Brain Theory and Neural Networks,” no. 1, M. A. Arbib, Ed. MIT Press, 1998, pp. 1–5.

-
- [24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2012.
- [25] B. K. Savas and Y. Becerikli, "Real time driver fatigue detection system based on multi-task ConNN," *IEEE Access*, vol. 8, pp. 12491–12498, 2020
- [26] Dlib C++ Library, Accessed : Sep. 25, 2019. [Online]. Available : <http://dlib.net/>
- [27] W. Zhang, Y. L. Murphey, T. Wang, and Q. Xu, "Driver yawning detection based on deep convolutional neural learning and robust nose tracking," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.
- [28] F. Zhang, J. Su, L. Geng, and Z. Xiao, "Driver fatigue detection based on eye state recognition," in *Proc. Int. Conf. Mach. Vis. Inf. Technol. (CMVIT)*, Feb. 2017, pp. 105–110
- [29] I. J. Goodfellow, Y. Bengio, and A. C. Courville, *Deep Learning (Adaptive Computation and Machine Learning)*. Cambridge, MA, USA : MIT Press, 2016. [Online]. Available : <http://www.deeplearningbook.org/>
- [30] W. Zhang, Y. L. Murphey, T. Wang, and Q. Xu, "Driver yawning detection based on deep convolutional neural learning and robust nose tracking," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015.
- [31] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. *Tracking-Learning-Detection*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010, Vol.6, No.1, pp : 1-13.
- [32] Shabnam Abtahi, Mona Omidyeganeh, Shervin Shirmohammadi, and Behnoosh Hariri. *YawDD : A Yawning Detection Dataset*, *Proceedings of the 5th ACM Multimedia Systems Conference*, 2014, pp : 24-28. New York : ACM Press.
- [33] A. George and A. Routray, "Real-time eye gaze direction classification using convolutional neural network," in *2016 International Conference on Signal Processing and Communications (SPCOM)*, 2016
- [34] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001.CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1.IEEE, 2001, pp. I–511.
- [35] S. Asteriadis, D. Soufleros, K. Karpouzis, and S. Kollias, "A natural head pose and eye gaze dataset," in *Proceedings of the International Workshop on Affective-Aware Virtual Agents and Social Robots*. ACM, 2009, p. 1.
- [36] L. Florea, C. Florea, R. Vr^anceanu, and C. Vertan, "Can your eyes tell me how you think ? a gaze directed estimation of the mental activity," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2013, pp. 60–1.

-
- [37] R. Vranceanu, C. Florea, L. Florea, and C. Vertan, "Nlp eac recognition by component separation in the eye region," in *Computer Analysis of Images and Patterns*. Springer, 2013, pp. 225–232.
- [38] R. Huang, Y. Wang, and L. Guo, "P-FDCN based eye state analysis for fatigue detection," in *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, 2018.
- [39] Zhang K, Zhang Z, Li Z, et al. Joint face detection and alignment using multitask cascaded convolutional networks[J]. *IEEE Signal Processing Letters*, 2016, 23(10) : 1499-1503.
- [40] "Télécharger : Graphique, traitement d'images et vision," *Developpez.com*. [Online]. Available : <https://cpp.developpez.com/telecharger/index/categorie/456/Graphique-traitement-d-images-et-vision>. [Accessed : 04-Jul-2021].
- [41] Wikipedia contributors, "Anaconda (distribution Python)," *Wikipedia, The Free Encyclopedia*. [Online]. Available : [https://fr.wikipedia.org/w/index.php?title=Anaconda_\(distribution_Python\)&oldid=181729622](https://fr.wikipedia.org/w/index.php?title=Anaconda_(distribution_Python)&oldid=181729622). [Accessed : 04-Jul-2021].
- [42] L. Fontaine, "Jupyter - Guide rapide," *Hebergementwebs.com*, 16-Nov-2020. [Online]. Available : <https://www.hebergementwebs.com/tutoriel-jupyter/jupyter-guide-rapide>. [Accessed : 04-Jul-2021].
- [43] Wikipedia contributors, "Google Brain," *Wikipedia, The Free Encyclopedia*. [Online]. Available : https://fr.wikipedia.org/w/index.php?title=Google_Brain&oldid=179167477. [Accessed : 04-Jul-2021].
- [44] Wikipedia contributors, "Keras," *Wikipedia, The Free Encyclopedia*, 13-Jun-2021. [Online]. Available : <https://en.wikipedia.org/w/index.php?title=Keras&oldid=1028356610>. [Accessed : 04-Jul-2021].
- [45] "dlib C++ Library," *Dlib.net*. [Online]. Available : <http://dlib.net/>. [Accessed : 04-Jul-2021].
- [46] *Univ-toulouse.fr*. [Online]. Available : <https://perso.math.univ-toulouse.fr/fdelebec/files/2018/03/python-bibliotheques.pdf>. [Accessed : 04-Jul-2021].
- [47] Wikipedia contributors, "Matplotlib," *Wikipedia, The Free Encyclopedia*, 03-Jul-2021. [Online]. Available : <https://en.wikipedia.org/w/index.php?title=Matplotlib&oldid=1031751031>. [Accessed : 04-Jul-2021].
- [48] F. Sangare and *TIC Magazine BF*, "Python : tout savoir sur le principal langage Big Data et Machine Learning," *Ticmagazine.bf*, 14-Sep-2018. [Online]. Available : <https://ticmagazine.bf/python-tout-savoir-sur-le-principal-langage-big-data-et-machine-learning/>. [Accessed : 04-Jul-2021].

- [49] Vijayalaxmi, Elizabeth, Rani D, “High performance and low power driver fatigue detection using image processing technique” , 2015. Department of Electronics and Communication Engineering . GITAM University. p. 13.
- [50] “Visual Anlaysis.” [Online]. Available : <https://ibug.doc.ic.ac.uk/research/face-analysis/>.
- [51] “Automatic emergency braking” [Online]. Available : <https://www.bosch-mobility-solutions.com/en/solutions/>.
- [52] “What Causes Excessive Yawning and How to Treat It.” [Online]. Available : <https://www.healthline.com/health/yawning-excessive>.
- [53] “facial expression anlysis.” [Online]. Available : <https://www.cs.cmu.edu/cga/behavior/>.
- [54] [online]. <https://fr.wikipedia.org/wiki/>.