الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدة د. مولاي الطاهر

كلية الرياضيات و الإعلام الآلي و الاتصالات السلكية و اللاسلكية

قسم: الإعلام الآلي

Mémoire de Master en informatique

Spécialité : Intelligence Artificielle : Principes et Applications

T h è m e

Détection de la somnolence des conducteurs à l'aide de l'Apprentissage Profond

- **Présenté par :**

  Kouadri Ahmed

  Kaid Hamid

- **Dirigé par :**

  Mr. Khiati Nadri

Année universitaire 2024-2025

Remerciements

En préambule à ce mémoire, nous remercions ALLAH qui nous a aidé et donné la patience et le courage durant cette longue année d'étude.

Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Nos remerciements vont aussi au corps professoral et administratif de département d'informatique de l'université MOULAY TAHAR de SAIDA pour la richesse et la qualité de ses enseignements et qui déploie des grands efforts pour assurer à leurs étudiants une formation de qualité.

*Nous remercions aussi notre encadreur Dr.**Nadri Khiati**, pour le temps qu'il nous a consacrés et leurs précieux conseils.*

Nos remerciements iront également vers tous ceux qui ont accepté avec bienveillance de participer au jury de ce mémoire.

On n'oublie pas nos parents pour leur contribution, leur soutien et leur patience.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragées au cours de la réalisation de ce mémoire.

Merci à tous et à toutes.

Dédicaces

Je dédie ce modeste travail

À mes parents qui depuis mon plus jeune âge ont toujours fait leur maximum, en consacrant temps et argent, pour m'éveiller et m'encourager dans mes passions. C'est grâce à vous et pour vous que j'ai fait mon mémoire. Aucun mot sur cette page ne saurait exprimer ce que je vous dois, ni combien je vous aime. Qu'Allah vous bénisse, vous assiste, vous vienne en aide ;

Pour ma femme, mon pilier, et mes chers enfants

A mon binôme KOUADRI AHMED

A mes chers frères, sœurs et mes neveux

A mon encadreur KHIATI NADRI

A tous mes collègues, plus particulièrement AISSA, FAHIM, et toute la section Master2 IA en témoignage de mon amitié sincère

A tous ceux qui m'ont soutenu, qu'ils trouvent ici l'expression de mon amour et ma profonde.

Gratitude

HAMID

Dédicaces

Je dédie ce travail à :

Ma Mère et mon défunt Père pour leurs amours inestimables, leurs confiances, leurs soutiens, leurs sacrifices et toutes les valeurs qu'ils ont su m'inculquer.

A ma femme

A mes fils Khalil, Abd Allah, Mohamed, Acheraf

A mes frères et mes sœurs

A toute mes familles : Kouadri &Sersar

A mon encadreur KHIATI NADRI

A mon binôme KAID HAMID

A tous mes collègues M2 IA, plus particulièrement AISSA Mimoune

Ahmed

ملخص

يركز هذا المشروع على تصميم وتنفيذ نظام الكشف عن نعاس السائق، باستخدام التعلم العميق. والهدف منه هو منع الحوادث المرورية الناجمة عن نعاس السائق من خلال تطوير نظام قادر على مراقبة علامات التعب. تم تجهيز النظام بكاميرا لالتقاط الصور الآنية، وأجهزة لإصدار تنبيهات صوتية عند اكتشاف علامات النعاس، وباستخدام خوارزمية Facial Landmark (68) يتم تحديد مكان الوجه والعين، في حين يسمح نموذج الشبكة العصبية التلافيفية CNN باكتشاف حالة العين، في حالة إغلاق أو فتح الجفون، وهو مؤشر رئيسي للنعاس. يقترح هذا المشروع حلاً فعالاً لتحسين السلامة المرورية، مع إمكانيات التحسين والدمج في أنظمة المركبات الذكية.

الكلمات المفتاحية: إرهاق القيادة، التعلم العميق، التعرف التلقائي على تعابير الوجه، الشبكة العصبية التلافيفية.

Abstract

Your summary in English

Our final year project focuses on the design and implementation of a driver drowsiness detection system using deep learning. The goal is to prevent road accidents caused by driver drowsiness by developing a system capable of monitoring signs of fatigue.

The system is equipped with a camera for real-time image acquisition and speakers to emit audible alerts when signs of drowsiness are detected. The Facial Landmark (68) facial detection algorithm is used for face and eye detection, while the CNN model detects the state of the eyes, particularly to identify eyelid closure or opening, which is a key indicator of drowsiness.

This project proposes an effective solution to improve road safety, with potential for improvement and integration into intelligent vehicle systems.

Keywords: Driver drowsiness, Deep learning, Automatic recognition of facial expressions, CNN.

Résumé

Votre résumé en français

Notre projet de fin d'études se concentre sur la conception et la réalisation d'un système détection de somnolence au volant, utilisant l'apprentissage profond. L'objectif est de prévenir les accidents de la route causés par la somnolence du conducteur en développant système capable de surveiller les signes de fatigue.

Le système est équipé d'une caméra pour l'acquisition d'images en temps réel, et des baffles pour émettre des alertes sonores lorsque des signes de somnolence sont détectés, l'algorithmes de la détection faciale Facial Landmark (68) est utilisée pour la détection du visage et des yeux, tandis que le model CNN permet de détecter l'état des yeux, en particulier pour identifier la fermeture ou l'ouverture des paupières, ce qui est un indicateur clé de la somnolence.

Ce projet propose une solution efficace pour améliorer la sécurité routière, avec des possibilités d'amélioration et d'intégration dans des systèmes intelligents de véhicules.

Mots clés : somnolence au volant, l'apprentissage profond, Reconnaissance automatique des expressions faciales, CNN.

TABLE DES MATIÈRES

Remerciements I
Résumé.....IV
Liste des figures V
Liste des Tableaux VI
Introduction générale 1
I. L'intelligence artificielle et l'apprentissage profond 3
1. Introduction 4
2. L'apprentissage automatique.....5
2.1 Les phase de l'apprentissage automatique.....5
2.2 Types d'apprentissage automatique 5
2.2.1 Apprentissage supervisé (Supervised Learning).....6
2.2.2 Apprentissage non supervisé (Unsupervised learning)……...7
2.2.3 Apprentissage par renforcement (Reinforcement learning)……8
3. L'apprentissage profond.....9
3.1 Réseaux de neurones artificiels.....9
3.2 Les fonctions d'activation.....10
3.2.1 ReLu.....10
3.2.2 Sigmoid.....11
3.2.3 Softmax.....11
3.3 Réseaux de neurones récurrents.....12
3.3.1 Réseaux de mémoire à long terme et à court terme (LSTM).....13
3.3.2 LSTM et RNN.....15
3.4 Réseaux de neurones convolutionnels 16
3.4.1 L'architecture réseaux de neurones convolutionnels.....16
3.4.2 Architectures CNN populaires 18
4. Conclusion :.....20
II. Reconnaissance automatiques des expressions faciales 21
2.1 Introduction : 22
2.2 La Vision par Ordinateur :.....22
2.3 Fonctionnement des systèmes de vision par ordinateur : 22
2.4 Le traitement d'image 23
2.4.1 Étapes fondamentales du traitement d'image.....24

2.4.2 Applications du traitement d'images : 25
2.5 Les expressions faciales 26
2.5.1 Les six expressions faciales universelles : 26
2.5.2 La reconnaissance automatique des expressions faciales 26
2.5.3 Historique de reconnaissance des expressions faciales....27
2.5.4 Les étapes de reconnaissance des expressions faciales....27
2.5.5 Le fonctionnement d'un système d'analyse des expressions faciales....30
2.5.6 Difficultés de la reconnaissance des visages : 33
2.5.7 Différentes méthodes développées pour la reconnaissance faciale....33
2.5.8 Les algorithmes de la détection faciale : 35
2.6 Conclusion :....38
III. Etat de l'art sur les approches de détection de la somnolence 39
3.1 Introduction : 40
3.2. Relation entre la somnolence et la conduite : 40
3.3. Approches de détection de somnolence 41
2.3.1. Approches axées sur la performance du conducteur 42
2.3.2. Approches axées sur les signaux physiologiques....42
2.3.3. Approches axées sur le traitement d'image 43
2.4. Conclusion....48
IV. CONCEPTION ET IMPLEMENTATION 49
4.1 Introduction....50
4.2 Conception 50
4.2.1 Présentation du système par l'architecture de CNN....51
4.2.2 Collecte et exploitation de la base des données 52
4.2.3 Conception de modèle CNN 53
4.2.4 La détection des points caractéristiques dans un visage....55
4.2.5 Détection des yeux....56
4.3 Implémentation : 57
4.3.1 Environnement de développement du matériel 57

4.3.2 Mise en place et la configuration de l'environnement 57
4.3.3 Les bibliothèques utilisées :.....59
4.3.4 Fonctionnement de programme principal :.....59
4.4. Résultats expérimentaux :.....62
4.4.1 Entrainement du modèle.....62
4.2 Vérification de la prédiction du modèle :.....65
4.4.3 Résultats de simulation :.....67
5. Conclusion 68
Conclusion générale.....69
Bibliographiés.....71

Liste des figures

Figure 1.1 : Les principaux types d'apprentissage automatique ……...…6

Figure 1.2 : La classification et la régression ……...…7

Figure 1.3 : créer clusters à partir des données…….……8

Figure 1.4 : Architecture des réseaux neuronaux artificiels.……...…10

Figure 1.5 : Réseaux de neurones récurrents…….....……..12

Figure 1.6 : A. Représentation par couche convolutive……...……....13

Figure 1.7 : B.Représentation par couche convolutive…….…….. …..…….14

Figure 1.8 : Types pool……..…….15

Figure 1.9 : L'architecture VGGNet……..……..……..16

Figure 1.10 : L'architecture ResNet…….……...……16

Figure 1.11 : Architecture d'une cellule LSTM……..……..……..17

Figure 1.12 : Les composants d'une cellule LSTM…….……...…….18

Figure 1.13 : Le modèle LSTM……..……...……..19

Figure 1.14 : LSTM et RNN……..……...……..……19

Figure 2.1: les domaines d'application de vision par ordinateur……..……23

Figure 2.2: Technique de détection de visage……..……28

Figure 2.3: Système d'analyse des expressions faciales…….…...…….30

Figure 2.4: Méthodes de reconnaissance faciale……...……..…33

Figure 2.5: 68 Facial Landmark……..……35

Figure 3.1 présente les méthodes de détection de somnolence……..…..……..…40

Figure 3.2: capteurs utilisés dans les méthodes axées sur la performance du onducteur……43

Figure 3.3 : Architecteur utilisé dans l'approche de Bhargava Reddy et al……...……..…46

Figure 4.1 : Système de détection de somnolence……...……...51

Figure 4.2 : Base de données utilisée……..……..…….52

Figure 4.3: Exemples d'annotations d'images de l'ensemble de base de données…..…...…..…53

Figure 4.4 : Architecture de notre modèle……...……...54

Figure 4.5 : Les couches de l'architecture de modèle CNN……..…...……..55

Figure 4.6 : Shape predictor land marks 68……...…….56

Figure 4.7: Région d'un oeil représentée par les points caractéristiques……..…….....56

Figure 4.8: Image illustrative de la détection des yeux……....…….57

Figure 4.9 : Le Logo de python……..…….……..….…….58

Figure 4.10 : Espace de travail de Visual studio…….……..…….59

Figure 4.11: Organigramme d'algorithme réalisé..……..…….……62

Figure 4.12: Le début de l'entrainement……..…….….……63

Figure 4.13 Fin de l'entrainement……..……63

Figure 4.14 : Présentation de perte de l'entrainement……...……...…..64

Figure 4.15 : Présentation de l'exactitude de l'entrainement……..……..…….…64

Figure 4.16 : Code de prédiction de notre modèle……..…….…….66

Figure 4.17 : L'image testé par notre model CNN …….…...……...…….66

Figure 4.18 : Résultat de la prédiction de l'image **(**s0001_00009_0_0_0_0_0_01)…...…...….67

Figure 4.19 : Résultat des yeux ouvert……..…….…67

Figure 4.20 : Résultat des yeux fermés……..……..…….…68

Liste des Tableaux

Tableaux 3.1 : Synthèse des travaux basé sur la traitement d'image ……....…48

Introduction générale

La conduite automobile est l'une des activités les plus courantes et essentielles de la vie quotidienne. cependant, la somnolence au volant constitue un danger majeur pour la sécurité routière, mettant en péril la vie des conducteurs, des passagers et des autres usagers de la route, souligne l'importance de détecter les signes de fatigue, ce que la reconnaissance faciale pourrait contribuer à accomplir en évaluant les caractéristiques du visage, afin de prévenir les conséquences graves de la somnolence au volant telles que le ralentissement du temps de réaction, la diminution de l'attention et la baisse des capacités cognitives.

Compte tenu des risques significatifs associés à la somnolence, sa détection chez les conducteurs est primordiale pour la sécurité routière. Notre approche s'appuie sur l'utilisation d'algorithmes d'apprentissage profond, notamment CNN (Les réseaux de neurones convolutionnels), afin d'exploiter les données de reconnaissance faciale pour identifier les signes de somnolence du conducteur.

En tirant parti des avancées de l'apprentissage profond, notre méthode vise à détecter les indicateurs de somnolence, tels que les yeux fermés, à partir des images faciales des conducteurs. Nos modèles de réseaux de neurones convolutifs sont utilisés pour extraire les caractéristiques discriminantes et permettre une détection en temps réel. Cette approche a pour but d'alerter les conducteurs en cas de signes de somnolence, contribuant ainsi à réduire les risques d'accidents liés à la fatigue au volant.

Les autres sections de ce mémoire sont structurées comme suit :

Le premier chapitre se concentre sur les concepts fondamentaux de l'apprentissage automatique et de l'apprentissage profond, en mettant en lumière les principes des réseaux de neurones artificiels. Il couvre les concepts de rétropropagation, l'apprentissage supervisé et

non supervisé, ainsi que les architectures couramment utilisées, telles que les réseaux de neurones convolutifs (CNN), les réseaux de neurones récurrents (RNN) et les réseaux de mémoire à long terme et à court terme (LSTM), en fournissant des exemples d'architectures.

Le deuxième chapitre est divisé en deux parties distinctes :

- La première section est dédiée à vision par ordinateur. Elle explore les différentes étapes et les concepts clés, tels que la détection des visages et l'extraction des caractéristiques faciales. L'objectif est de présenter les fondements de la technique de vision par ordinateur employée pour la détection de la somnolence.
- La deuxième section aborde spécifiquement la reconnaissance faciale pour la détection de la somnolence des conducteurs, en utilisant la vision par ordinateur et les différentes méthodes développées pour la reconnaissance faciale.

Le troisième chapitre est un état de l'art qui donne un aperçu général sur deux étapes de détection : une étape de détection du visage accompagné de ses différentes méthodes basées sur les techniques de la comparaison des modèles de visage, les caractéristiques invariantes et les apparences, au niveau de cette dernière on a illustré la méthode de Viola et Jones. Et une étape de détection des yeux avec les méthodes basées sur l'apprentissage du modèle statistique d'apparence afin d'analyser leurs états.

Le quatrième chapitre est divisé en deux parties distinctes :

- La première section est dédiée à la conception du système que nous proposons pour la détection de la somnolence, en présentant ses architectures générales et détaillées.
- La deuxième section abordera l'implémentation de l'architecture et les outils utilisés dans ce projet, ainsi que la mise en œuvre du système d'apprentissage profond. L'évaluation empirique et les résultats seront également présentés dans ce chapitre.

CHAPITRE I

L'INTELLIGENCE ARTIFICIELLE ET L'APPRENTISSAGE PROFOND

1. Introduction

Le machine learning et le deep learning, considérés comme des sous-domaines de l'IA, ont récemment suscité un grand intérêt. Grâce à l'apprentissage automatique, les ordinateurs peuvent élaborer eux-mêmes des modèles d'apprentissage, sans nécessiter de programmation préalable, en se basant sur d'importants ensembles de données. L'apprentissage profond, une des nombreuses variantes de l'apprentissage automatique qui a prospéré ces dernières années, occupe la couche directement inférieure. Ces méthodes d'IA ont montré un potentiel considérable dans divers domaines.

Ce chapitre présente un aperçu des techniques d'apprentissage profond que nous allons employer pour l'évolution de notre projet. Nous allons principalement nous focaliser sur l'architecture des réseaux de neurones convolutifs (CNN), en commençant par le sujet de l'apprentissage automatique et les diverses sortes d'algorithmes qu'il emploie. Par la suite, nous traiterons les algorithmes d'apprentissage profond. Nous proposons aussi un aperçu des architectures CNN les plus fréquentes.

2. L'apprentissage automatique

Le ML est un domaine de l'intelligence artificielle qui a été défini en 1959 par son pionnier Arthur Samuel comme « le domaine d'étude qui permet aux ordinateurs d'acquérir la capacité d'apprendre sans être explicitement programmés à apprendre » [7], ayant pour but de permettre aux ordinateurs d'acquérir des connaissances. Ce n'est pas un ordinateur intelligent au sens traditionnel, mais il permet aux ordinateurs d'exécuter des tâches en apprenant à partir de données plutôt qu'en suivant des instructions explicites. Contrairement à la programmation traditionnelle, où les règles sont définies manuellement par les programmeurs. Le machine Learning aborde des questions complexes dans lesquelles la

programmation traditionnelle présente des limites.

Le principe fondamental du machine learning consiste à donner à un ordinateur un ensemble de données d'entraînement (ou training set), puis à lui donner la possibilité d'analyser ces données afin de déterminer des modèles et des relations. Après avoir mis en place le modèle, il peut servir à faire des prédictions ou des classifications sur de nouvelles données.

2.1 Les phases de l'apprentissage automatique

En règle générale, les phases suivantes constituent la séparation des algorithmes d'apprentissage automatique :

1. **Collecte et préparation des données :** il s'agit de collecter les données nécessaires pour entraîner le modèle d'apprentissage automatique et le préparer pour l'analyse.
2. **Sélection de modèles d'apprentissage automatique :** Il existe différents types de modèles d'apprentissage automatique, chacun présentant des avantages et des inconvénients en fonction du type de données et de la tâche.
3. **Entraînement du modèle :** le modèle est entraîné sur les données d'apprentissage pour voir les motifs et les relations entre les variables.
4. **Évaluation du modèle :** Une fois le modèle formé, il doit être évalué sur des données de test pour mesurer sa précision et sa capacité à généraliser à de nouvelles données.
5. **Utilisation du modèle :** Une fois que le modèle a été évalué et validé, il peut être utilisé pour faire des prévisions ou des prédictions sur de nouvelles données.

2.2 Types d'apprentissage automatique

L'apprentissage automatique est un domaine assez vaste. La figure 1.1 ci-dessous illustre les différents types d'apprentissage automatique.

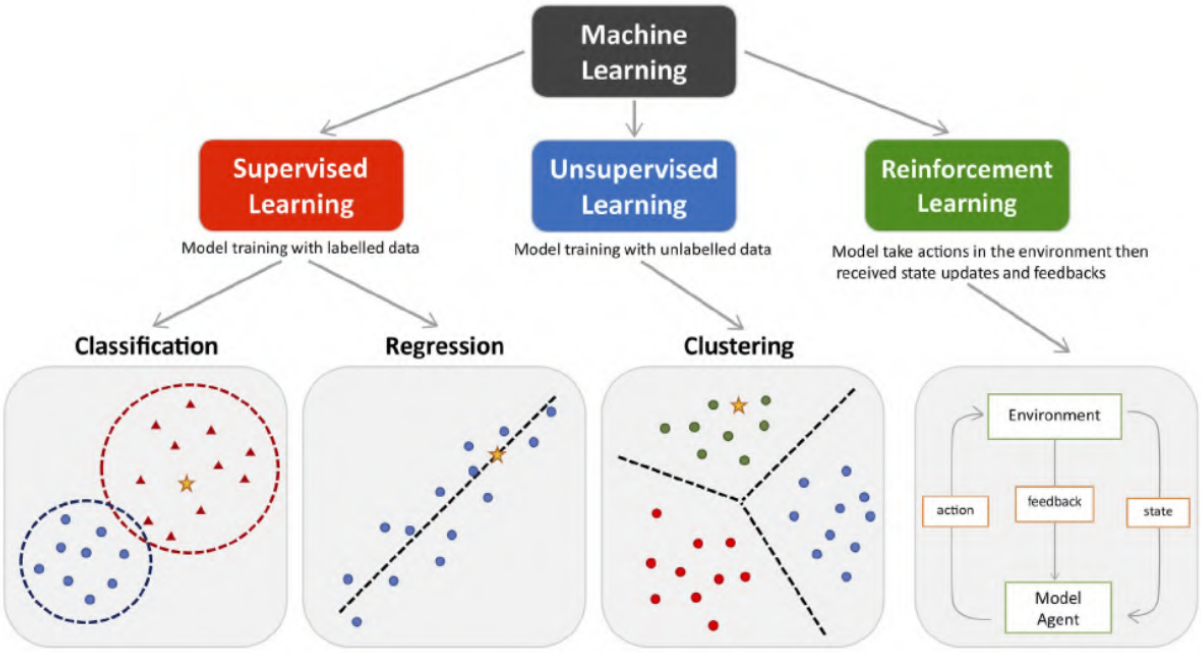


Figure 1.1 : Les principaux types d'apprentissage automatique. [1]

2.2.1 Apprentissage supervisé (Supervised Learning)

L'apprentissage supervisé est une approche d'apprentissage automatique qui consiste à créer un algorithme capable d'apprendre à faire des prédictions. Cet apprentissage est réalisé par apprentissage. L'apprentissage supervisé se divise généralement en deux grandes catégories : la classification et la régression.

a) Classification

Le processus de création d'un modèle permettant de diviser les données en catégories distinctes, de les simplifier et de réaliser des prédictions relève de la catégorisation. Ces ensembles de données peuvent être synthétisés à l'aide d'une méthode de classification qui permet également d'identifier les relations clés et la structure de l'ensemble de données. Si l'on détermine l'existence de classes distinctes d'éléments, celles-ci peuvent être identifiées, leurs caractéristiques calculées et les informations structurées et extraites plus efficacement. Cela facilite également l'affectation de nouveaux objets au système.[2]

b) Régression

La régression est une technique de modélisation prédictive qui vise à identifier une relation significative entre une variable dépendante et une ou plusieurs variables indépendantes, également appelées variables cibles. Il existe différents types de techniques de régression : linéaire, logistique, polynomiale, Ridge, Lasso et bien d'autres.[3]

La figure ci-dessous Figure 1.2 montre la classification et la régression.

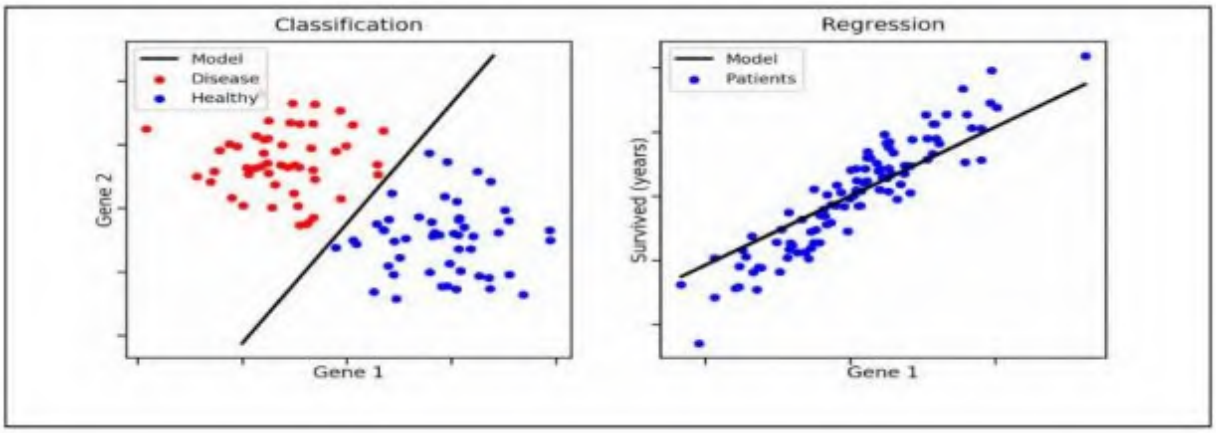


Figure 1.2 : La classification et la régression.

2.2.2 Apprentissage non supervisé (Unsupervised learning)

L'apprentissage non supervisé se produit lorsque les données ne sont pas étiquetées. Nous disposons donc de données d'entrée dont nous ignorons les résultats. L'ensemble de données et l'objectif du système sont d'identifier les caractéristiques communes des données d'apprentissage. L'apprentissage non supervisé repose principalement sur des algorithmes de clustering.

Clustering

L'analyse de cluster constitue une méthode employée dans l'extraction de données et l'apprentissage automatiquepour regroupé des éléments similaires en clusters. La méthode de clustering K-means est souvent adoptée pour l'analyse de cluster, avec pour objectif de segmenter un ensemble d'objets en K clusters de manière à minimiser la somme des distances au carré entre

les objets et leur moyenne de cluster respective[4]. La figure ci-dessous 1.3 illustre la formation des clusters à partir des données.

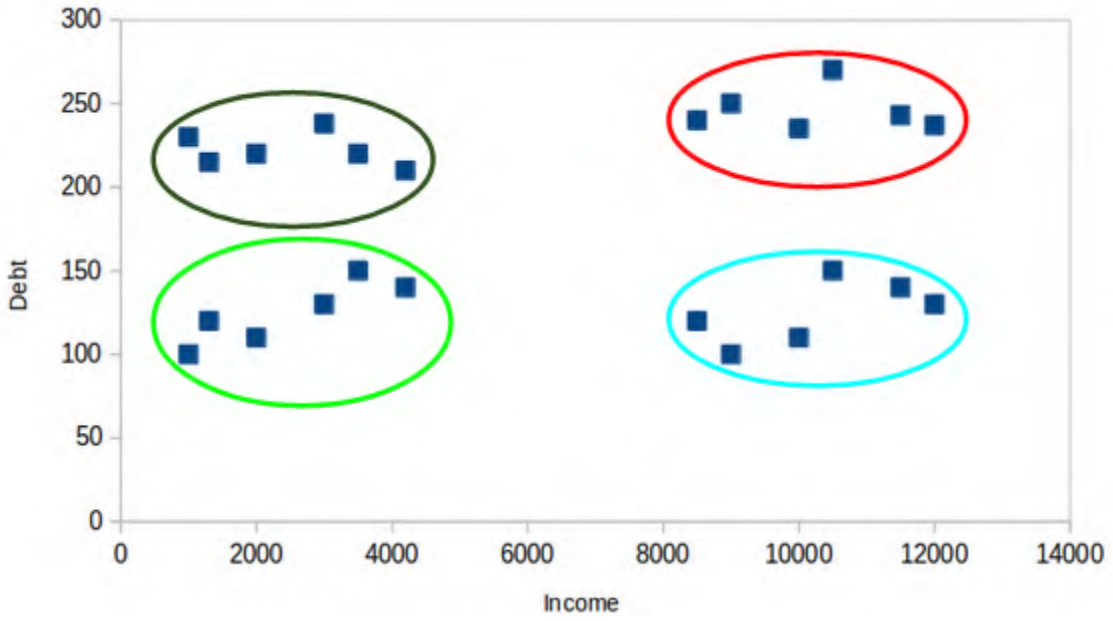


Figure 1.3 : créer clusters à partir des données.

Le clustering hiérarchique et le clustering k-means constituent deux méthodes fréquemment utilisées dans le champ de l'apprentissage non supervisé pour classer les points de données en groupes distincts. Tandis que le clustering k-means partitionne les données en un nombre déterminé de groupes, le clustering hiérarchique établit une structure en arborescence pour illustrer les relations existant entre les catégories. [4]

2.2.3 Apprentissage par renforcement (Reinforcement learning)

L'apprentissage par renforcement met l'accent sur la manière dont un agent autonome interagit avec son environnement pour apprendre à choisir les comportements les plus appropriés afin d'atteindre ses objectifs. Il s'appuie sur les actions effectuées par l'agent dans l'environnement pour évaluer et récompenser son comportement, considérant les résultats de ses actions. L'apprentissage par renforcement a diverses applications, notamment un ordinateur qui joue aux échecs contre un humain, l'apprentissage de la reconnaissance de la parole et la classification de nouvelles formations astronomiques.

3. L'apprentissage profond

Le Deep Learning est largement utilisé dans de multiples domaines [17], incluant la reconnaissance d'images, la traduction automatique, les véhicules autonomes, le diagnostic médical avancé, les recommandations personnalisées, la modération automatique des réseaux sociaux, ainsi que la prédiction financière et le trading automatisé. Il est également appliqué dans l'identification de défauts sur des pièces, la détection de malwares et de fraudes, la création de chatbots pour des interactions conversationnelles, l'exploration spatiale, ainsi que dans le développement de robots intelligents. Ces applications témoignent de la capacité du Deep Learning à transformer divers secteurs grâce à ses capacités d'analyse et de prise de décision basées sur les données.

3.1 Réseaux de neurones artificiels

Le réseau de neurones artificiels s'inspire des réseaux neuronaux biologiques qui illustrent la configuration du cerveau humain. Tout comme le cerveau, ces réseaux artificiels sont composés de neurones organisés en différentes couches interconnectées. Ces neurones sont désignés sous le terme de nœuds. Dans un ANN, les dendrites des réseaux neuronaux biologiques symbolisent les entrées, les corps cellulaires sont représentés par les nœuds, les synapses correspondent aux poids, tandis que les axones sont liés aux sorties.

Les ANN agissent en tant que modèles statistiques non linéaires, capables d'illustrer une relation complexe entre les entrées et les sorties pour établir un nouveau modèle. Les réseaux neuronaux artificiels trouvent des applications variées, telles que la reconnaissance d'images, la réception vocale, la traduction automatique, ainsi que dans le domaine du diagnostic médical.

Le fait qu'un réseau de neurones artificiels (ANN) s'améliore en se basant sur des exemples de données représente un atout significatif. L'utilisation la plus courante des ANN consiste en

l'approximation de fonctions aléatoires. Avec ces technologies, il est possible d'atteindre des solutions qui désignent la distribution de manière économique. Un ANN peut également produire un résultat basé sur un exemple de données au lieu d'exploiter l'intégralité du jeu de données. Les ANN peuvent servir à perfectionner les méthodes d'analyse de données déjà en place grâce à leur forte capacité prédictive. La figure 1.4 ci-dessous montre l'architecture des réseaux neuronaux artificiels.

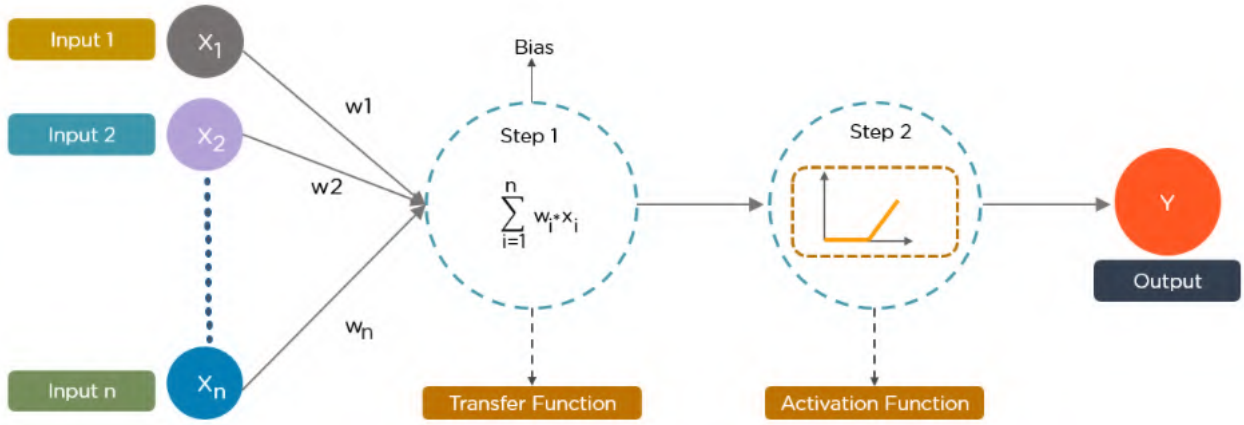


Figure 1.4 : Architecture des réseaux neuronaux artificiels. [5]

3.2 Les fonctions d'activation

3.2.1 ReLu

Le ReLU, ou fonction d'activation linéaire rectifiée, figure parmi les quelques indicateurs de changement dans le domaine de l'apprentissage profond. Sa simplicité d'utilisation le rend bien plus efficace que des fonctions d'activation comme sigmoid ou tanh. La formule du ReLU est :

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

La fonction ReLU et sa dérivée conservent toujours une monotonie. Lorsque la fonction reçoit une valeur négative, elle produit 0 ; en revanche, si la valeur positive x est introduite, elle restitue cette valeur. Par conséquent, la sortie varie de 0 à l'infini. [1] Leaky ReLU constitue une

variante ajustée de la fonction ReLU. Plutôt que d'assigner la valeur zéro à la fonction ReLU lorsque x s'avère négatif, elle lui attribue une composante linéaire d'une taille très réduite, à savoir la Composante linéaire de x. Sur le plan mathématique, cela peut être formulé comme [6].

$$f(x) = 0.01x, x < 0$$

$$f(x) = x, x >= 0$$

3.2.2 Sigmoid

Il s'agit de la fonction d'activation la plus couramment utilisée en raison de sa nature non linéaire. La fonction sigmoïde convertit les valeurs dans une plage allant de 0 à 1. Elle peut être définie comme suit :

$$f(x) = 1/e^{(-x)}$$

La fonction sigmoïde est continuellement différentiable et présente une forme en S. Sa dérivée est la suivante :

$$f'(x) = 1 - sigmoid(x)$$

De surcroît, la fonction sigmoïde ne présente pas de symétrie par rapport à zéro, ce qui signifie que les signes des valeurs de sortie de tous les neurones seront identiques. Cette condition peut être optimisée par l'emploi de la fonction sigmoïde.[6]

3.2.3 Softmax

La fonction softmax représente une composante mathématique qui convertit un vecteur composé de nombres réels en un vecteur de probabilités dont la somme est égale à 1. Chaque Valeur du vecteur d'origine est transformée en un chiffre variant entre 0 et 1. La formule de la fonction softmax est indiquée ci-dessous :

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

Comme mentionné précédemment, la fonction softmax peut traiter un vecteur z comportant K éléments. Pour chaque élément dans z, la fonction softmax applique la fonction exponentielle standard à cette valeur. Elle procède ensuite en divisant par la somme des exponentielles de toutes les valeurs présentes dans z.

3.3 Réseaux de neurones récurrents (RNN) :

Les réseaux neuronaux récurrents (RNN) Figure 1.5 représentent une catégorie de réseaux neuronaux fréquemment employés dans l'apprentissage profond. Les RNN intègrent leurs sorties précédentes comme nouvelles entrées, ce qui les rend adaptés au traitement de données séquentielles. Les RNN peuvent être utilisés et entraînés au fil du temps grâce à la rétropropagation classique ou à une variation de rétropropagation connue sous le nom de rétropropagation dans le temps. [7]

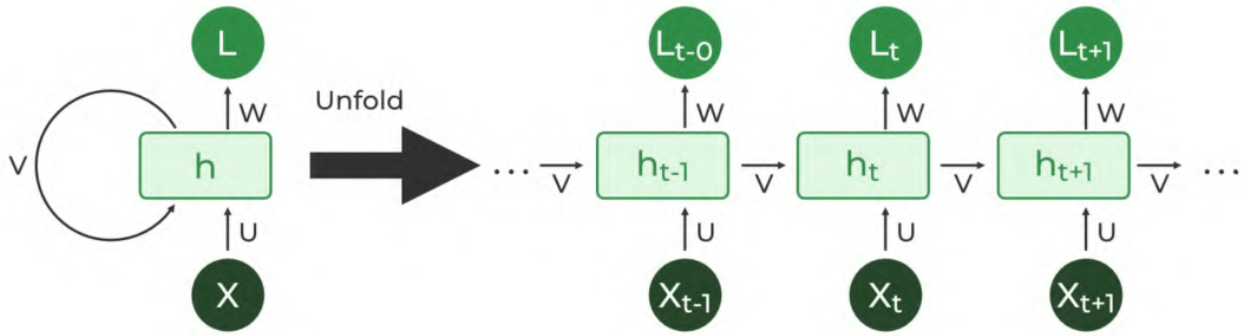


Figure 1.5 : Réseaux de neurones récurrents.

3.3.1 Réseaux de mémoire à long terme et à court terme (LSTM)

Les réseaux à mémoire à long terme et à court terme (LSTM), comme illustré à la figure 1.11 ci-dessous, constituent une architecture de réseau neuronal récurrent capable d'apprendre les dépendances à long terme. Ce modèle a été conçu pour résoudre les problèmes de gradient évanescent et est considéré comme une architecture de réseau neuronal profond au fil du temps. Le composant principal de la couche de mémoire à long terme est la cellule de mémoire [15].

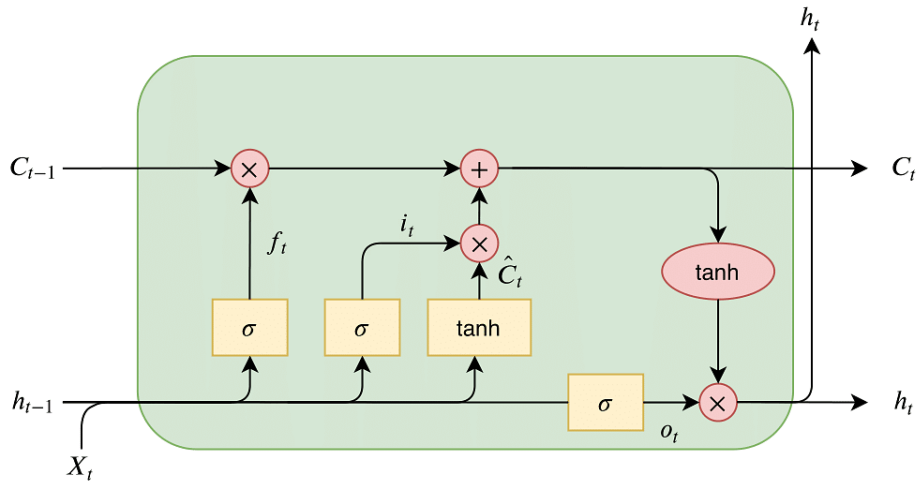


Figure 1.11 : Architecture d'une cellule LSTM.

Dans la figure 2.12 ci-dessus, chaque ligne connecte la sortie d'un nœud aux entrées d'autres nœuds. Les cercles roses représentent des opérations ponctuelles, telles que l'addition de vecteurs, tandis que les boîtes jaunes désignent les niveaux appris du réseau neuronal. Lorsque deux lignes convergent, elles fusionnent en une seule. Lorsqu'une ligne se divise, ses informations sont copiées et les copies sont dirigées vers différentes destinations [16].

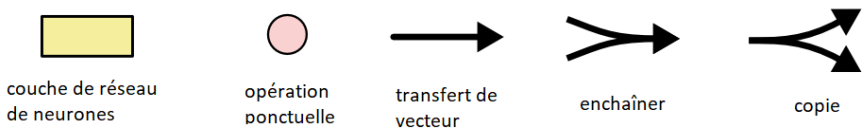


Figure 1.12 : Les composants d'une cellule LSTM.

Une cellule de mémoire est composée de quatre éléments principaux : une porte d'entrée, un neurone avec reconnexion, une porte d'oubli et une porte de sortie. Les équations suivantes décrivent le fonctionnement pas à pas d'une couche de cellules de mémoire pour des séries temporelles d'entrée sous la forme X = (x1, x2, x3, ..., xn), et des cellules de mémoire cachées H = (h1, h2, h3, ..., hn) [15].

$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i\right)$$
$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f\right)$$
$$o_t = \sigma\left(x_t U^o + h_{t-1} W^o\right)$$
$$\tilde{C}_t = \tanh\left(x_t U^g + h_{t-1} W^g\right)$$
$$C_t = \sigma\left(f_t * C_{t-1} + i_t * \tilde{C}_t\right)$$
$$h_t = \tanh\left(C_t\right) * o_t$$

Dans ce calcul, le signe représente une multiplication par éléments. En ignorant les termes de biais, il est possible de démontrer comment la couche cachée est calculée à l'instant ht. Dans les calculs susmentionnés :

— i, f, o désignent respectivement les portes d'entrée, d'oubli et de sortie.

— Wi, Wf, Wo sont les poids reliant la couche de récurrence à l'instant t-1 à la couche cachée à l'instant t.

— Ui, Uf, Uo représentent les poids qui connectent la couche cachée à l'instant t-1 à la couche récursive à l'instant t.

Dans le modèle illustré à la figure 1.13 ci-dessus, la porte de sortie constitue la dernière composante d'une cellule LSTM. Cependant, elle ne représente qu'une partie du processus global. D'autres éléments doivent être pris en considération avant que le réseau LSTM ne puisse générer les prédictions escomptées [16].

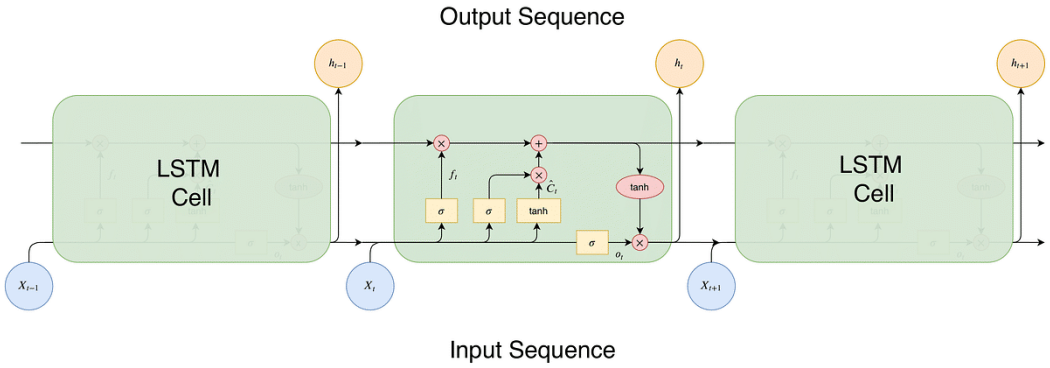


Figure 1.13 : Le modèle LSTM.

3.3.2 LSTM et RNN

Bien que les réseaux de neurones récurrents (RNN) se soient montrés efficaces pour la prédiction de séquences, l'apprentissage des dépendances à long terme peut s'avérer complexe. Cette difficulté est principalement attribuable aux problèmes d'explosion et de disparition du gradient, qui surviennent lorsque le gradient se propage à travers de nombreuses couches d'un réseau récurrent [16]. La figure 1.14 ci-dessous illustre les architectures LSTM et RNN.

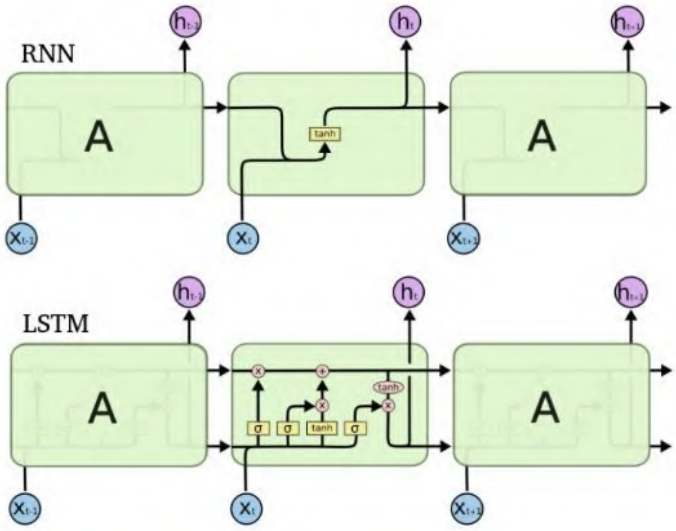


Figure 1.14 : LSTM et RNN.

3.4 Réseaux de Neurones Convolutionnels (CNN) :

Les réseaux de neurones convolutifs (CNN) représentent une sous-catégorie des réseaux de neurones, spécifiquement conçus pour l'analyse d'images. Ils sont appliqués dans divers domaines tels que la reconnaissance, la classification, la détection et la prédiction. Un avantage majeur des CNN réside dans leur capacité à apprendre de manière autonome un ensemble vaste et abstrait de caractéristiques lors de l'entraînement simultané sur un ensemble de données pour un problème de modélisation prédictive spécifique. [8]

3.4.1 L'architecture réseaux de neurones convolutionnels

Le réseau de neurones convolutif (CNN) est constitué de trois types de couches principales : la couche de **convolution**, la couche de **pooling** et la couche **full connected**.

a) La couche de convolution :

Cette section du CNN est responsable des calculs visant à extraire des informations pertinentes des données d'image. Chaque couche convolutive est constituée de noyaux, qui sont des filtres apprenables. Une image couleur peut être conceptualisée comme une grille tridimensionnelle de pixels. Un filtre, en revanche, est un réseau bidimensionnel de poids qui se déplace sur les champs réceptifs de l'image afin de détecter la présence ou l'absence d'une caractéristique. Un filtre est généralement appliqué à une image sous la forme d'une matrice 3x3, bien que sa taille puisse varier. Le produit scalaire des poids des pixels originaux et des poids du filtre est ensuite calculé. Le filtre est ensuite décalé de 'n' pixels, ce qui est appelé le 'stride', et le produit scalaire des pixels d'entrée et des poids du filtre est déterminé. Ce processus, également connu sous le nom de 'convolution', se poursuit jusqu'à la fin de la matrice lue. Le résultat de chaque produit scalaire est une grille appelée carte d'activation ou carte de caractéristiques. Le filtre a été utilisé pour extraire les informations sur les caractéristiques de l'image et les placer sur

la carte des caractéristiques [9]. Les figures 1.6 et 1.7 illustrent la représentation par couche convolutive dans les réseaux de neurones convolutifs (CNN).

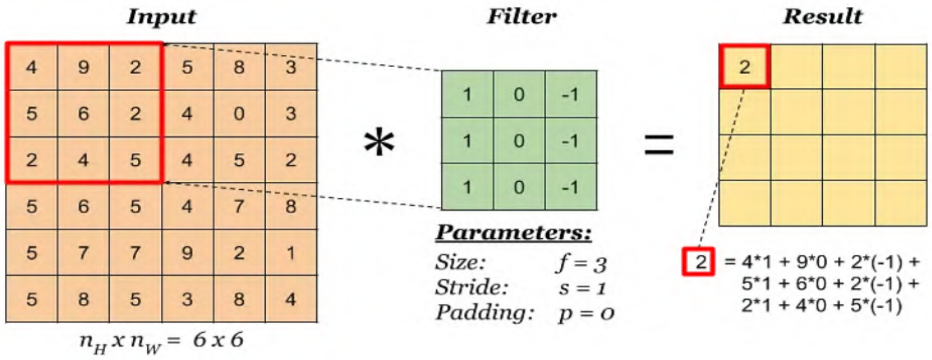


Figure 1.6 : Représentation par couche convolutive.

Comme illustré dans l'image ci-dessus, le noyau ou filtre possède une dimension de 3x3, et une sous-matrice de taille équivalente est extraite de l'image. Le produit matriciel est enregistré dans la matrice de sortie. Le filtre se déplace ensuite d'un pixel, étant donné que le pas est de 1 dans l'exemple précédent, et le calcul du produit scalaire est répété.

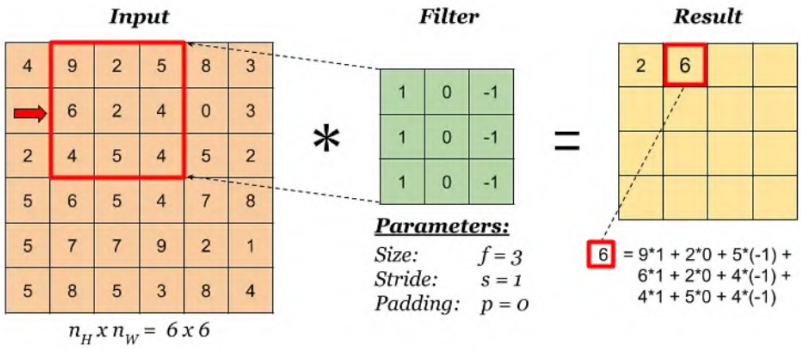


Figure 1.7 : Représentation par couche convolutive

b) La couche de pooling :

La couche de pooling est généralement insérée entre deux couches de convolution. Elle réalise une opération de sous-échantillonnage, réduisant ainsi la taille des images tout en conservant leurs caractéristiques essentielles. Pour ce faire, l'image est divisée en cellules régulières de 2 × 2 pixels non chevauchantes, ou en cellules de 3 × 3 pixels séparées par un pas de 2 pixels. Ensuite, le processus de pooling est appliqué. Les résultats obtenus sont similaires

aux entrées, mais de taille réduite. La couche de pooling diminue le nombre de paramètres et de calculs au sein du réseau, ce qui améliore ses performances et prévient le surapprentissage [10].

Comme illustré à la figure 1.8, deux types courants de couches de pooling sont le pooling max et le pooling average, qui sélectionnent respectivement la valeur maximale et la valeur moyenne. Le pooling max est plus fréquemment utilisé que le pooling average. La couche de pooling ne contient aucun paramètre à apprendre. L'idée derrière le pooling max est qu'une valeur élevée indique la détection d'une caractéristique [11].

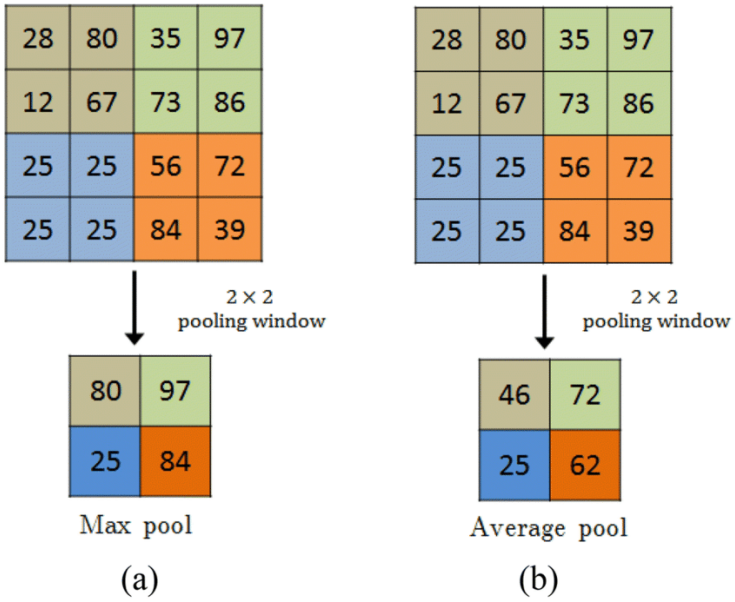


Figure 1.8 : Types pool.

c) La couche de full connected :

La couche fully-connectedest systématiquement la dernière couche d'un réseau neuronal, qu'il soit convolutif ou non, et n'est donc pas une caractéristique exclusive des CNN. Ce type de couche prend un vecteur en entrée et génère un nouveau vecteur en sortie [10].

3.4.2 Architectures CNN populaires

a) L'architecture VGGNet

VGGNet est une architecture de réseau de neurones convolutifs (CNN) développée par Karen Simonyan, Andrew Zisserman et leurs collaborateurs à l'Université d'Oxford. Comme

illustré à la figure 1.9, VGGNet est un CNN à 16 couches, comportant jusqu'à 95 millions de paramètres, et a été entraîné sur plus d'un milliard d'images (1000 classes). Il est capable de traiter de grandes images d'entrée de 224 x 224 pixels, pour lesquelles il extrait 4096 caractéristiques convolutives. Le modèle VGG CNN est efficace en termes de calcul et constitue une base solide pour de nombreuses applications en vision par ordinateur, grâce à son adaptabilité à diverses tâches, y compris la détection d'objets [12].

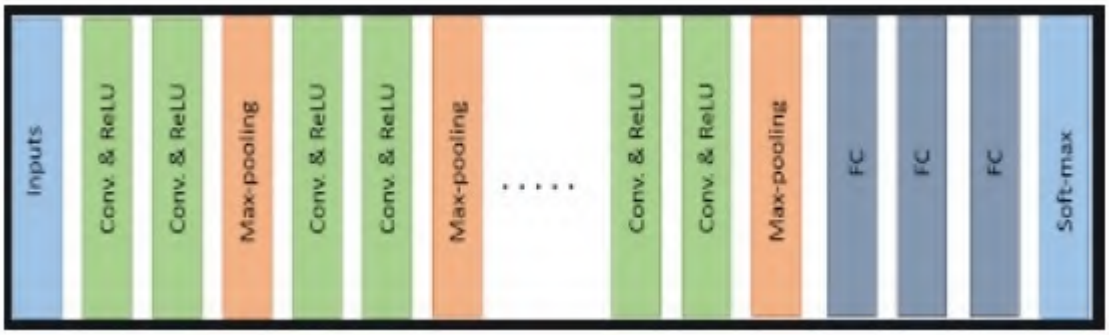


Figure 1.9 : L'architecture VGGNet.

b) L'architecture ResNet :

Kaiming He et al. ont développé ResNet, un modèle de réseau de neurones convolutifs caractérisé par ses connexions résiduelles et l'intégration de la normalisation par lots (Batch Normalization). Contrairement à d'autres architectures, ResNet ne comporte pas de couches entièrement connectées en fin de réseau. Cependant, son principal inconvénient réside dans la complexité de son évaluation due à son grand nombre de paramètres. ResNet est actuellement considéré comme un modèle de réseau CNN de pointe et est fréquemment privilégié pour l'utilisation des ConvNets en pratique [13]. La figure 1.10 ci-dessous illustre l'architecture ResNet.

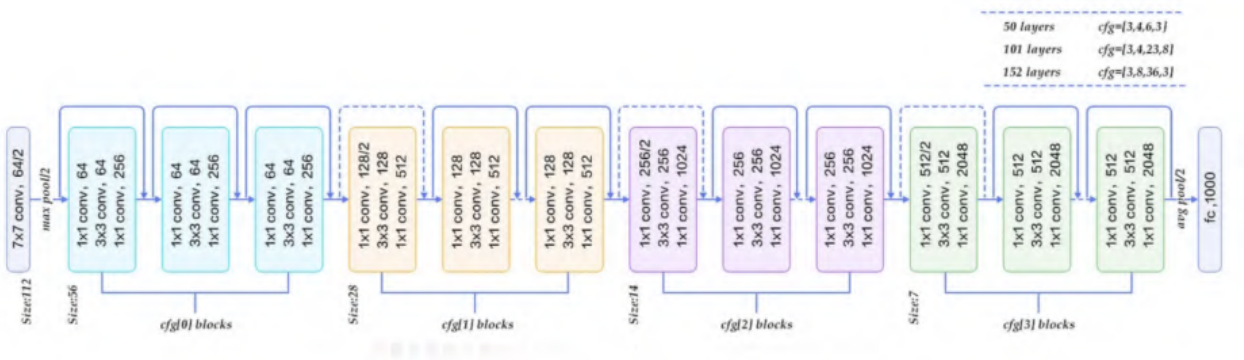


Figure 1.10 : L'architecture ResNet.

c) Inception :

Il s'agit d'une architecture qui exploite les modules Inception. Ce module intègre des filtres de différentes tailles (1x1, 3x3 et 5x5) afin de capturer des informations spatiales. Ensuite, un filtre de concaténation permet d'assembler les résultats de ces filtres. De plus, pour réduire la densité de connexion, un pooling moyen global est appliqué dans la dernière couche, remplaçant ainsi une couche entièrement connectée. Ces ajustements de paramètres ont permis une réduction significative du nombre de paramètres [14].

4. Conclusion :

Ce chapitre a abordé les principes fondamentaux des méthodologies d'apprentissage automatique et les bases de l'apprentissage profond. Nous avons ensuite approfondi les réseaux de neurones récurrents (RNN), les réseaux neuronaux convolutifs (CNN) et les réseaux à mémoire à long terme et à court terme (LSTM), en analysant leurs architectures et leurs couches distinctes. Des exemples notables de CNN, tels que ResNet, VGGNet et Inception, ont également été examinés. En conclusion, un aperçu exhaustif des avancées récentes dans le domaine a été présenté. Le chapitre suivant se concentrera sur notre système de détection de la somnolence, en détaillant ses composants essentiels et en évaluant ses performances de manière approfondie.

CHAPITRE II

RECONNAISSANCE AUTOMATIQUE DES EXPRESSIONS FACIALES

2.1 Introduction :

L'analyse et la reconnaissance automatique des expressions faciales ont suscité un intérêt considérable parmi les chercheurs en vision par ordinateur. Ce domaine, en pleine expansion au sein de l'intelligence artificielle, cherche à reproduire la capacité humaine à identifier et interpréter les expressions faciales. Ce chapitre offre une exploration approfondie des concepts, techniques et défis liés à la reconnaissance automatique des expressions faciales, fournissant ainsi une base solide pour appréhender cette technologie complexe et prometteuse.

2.2 La Vision par Ordinateur :

La vision par ordinateur regroupe un ensemble de méthodes conçues pour l'analyse automatisée d'images. Dès 1964, les experts en intelligence artificielle ont commencé à s'intéresser à la reproduction des capacités sensorielles humaines. La vision étant le sens le plus crucial pour l'être humain, de nombreux efforts ont été consacrés à ce domaine. L'objectif commun de ces travaux est de conférer un sens aux images ou d'extraire des informations sémantiques à partir de données visuelles stockées dans des bases de données [17]. La vision par ordinateur vise à imiter la capacité du cerveau humain à comprendre une scène en mouvement à partir de signaux visuels, notamment en identifiant rapidement des objets et en prévoyant leurs déplacements.

2.3 Fonctionnement des systèmes de vision par ordinateur :

La vision par ordinateur est une branche de l'intelligence artificielle et de la perception artificielle axée sur le canal visuel. Un système de vision par ordinateur a pour fonction de fournir des informations sur l'environnement au système, en utilisant des capteurs tels que des caméras vidéo. À l'interface entre le système et son environnement, un capteur détecte les changements d'état du monde et les communique au système. Une caméra, agissant comme

une rétine humaine artificielle, convertit le flux lumineux capturé par son objectif optique en une séquence d'images organisées chronologiquement (le flux vidéo). Chaque image est représentée par une matrice de pixels bidimensionnelle. Un système de vision par ordinateur analyse les données de bas niveau du flux vidéo pour en extraire des informations de plus haut niveau [18].reconnaissance faciale, les véhicules autonomes, la surveillance de sécurité et l'analyse médicale.

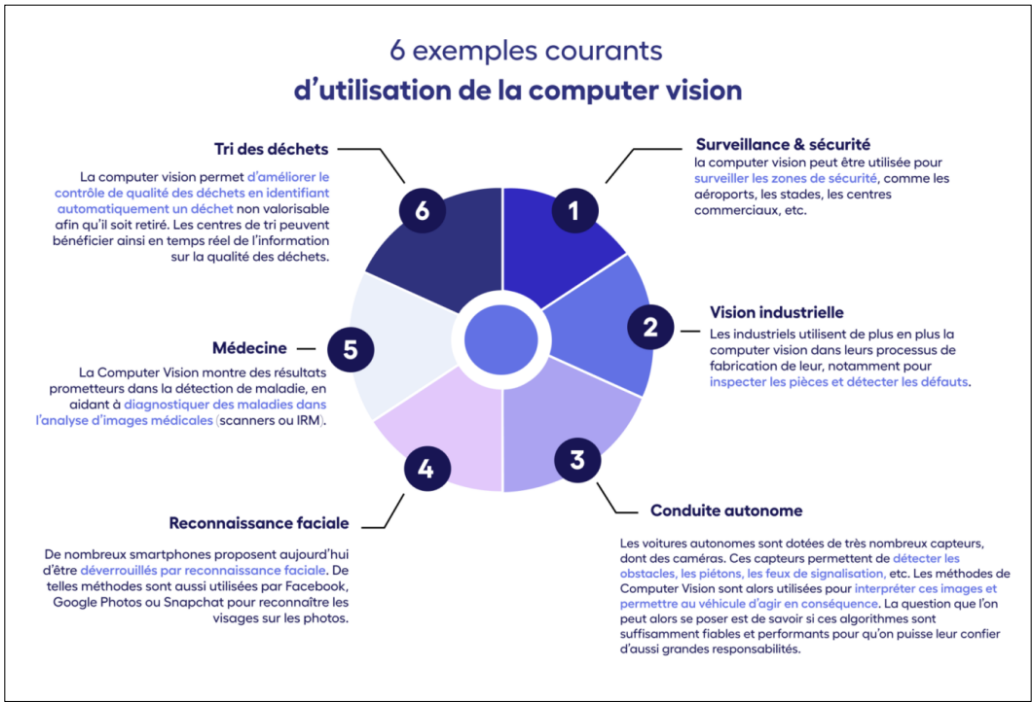


Figure 2.1: les domaines d'application de vision par ordinateur

2.4 Le traitement d'image

Le traitement d'images est un élément fondamental de la vision par ordinateur, combinant l'informatique et les mathématiques appliquées. Son but est de manipuler des images à l'aide de diverses techniques, ce qui aboutit à une nouvelle image, une version modifiée, ou simplement un paramètre lié à l'image originale. Ces résultats peuvent ensuite être utilisés pour des analyses complémentaires ou pour éclairer des processus décisionnels.

Ce domaine est d'une importance capitale dans de nombreuses applications pratiques, telles que la robotique, les véhicules autonomes et la reconnaissance d'objets. En permettant la transformation et la manipulation simultanée de multiples images, le traitement d'images facilite l'extraction d'informations précieuses à partir de ces ensembles de données visuelles[19].

2.4.1 Étapes fondamentales du traitement d'image

Les étapes clés du traitement d'image impliquent plusieurs processus qui transforment une image brute en une version exploitable pour une analyse ou une application spécifique. Voici les principales étapes [20] :

a. Acquisition: L'acquisition d'images est la première étape indispensable du traitement d'images, car aucun traitement n'est possible sans elles. Elle peut être définie comme l'opération de récupération d'une image provenant d'une source, généralement matérielle, pour la transmettre aux processus subséquents.

b. Prétraitement: Le prétraitement est une phase cruciale du traitement d'images, car il prépare l'image brute pour les étapes d'analyse et d'interprétation ultérieures. L'objectif est de nettoyer et d'améliorer l'image afin de faciliter la segmentation, l'extraction de caractéristiques et la reconnaissance. Cela peut inclure la correction de l'illumination, le redimensionnement et la réduction du bruit.

c. Segmentation: La segmentation d'images est une étape essentielle dans le traitement d'images et la vision par ordinateur. Elle consiste à diviser une image en différents ensembles de régions ou segments homogènes selon certains critères de similarité. La segmentation vise principalement à rendre la représentation d'une image plus claire et plus accessible à l'analyse en simplifiant ou en modifiant sa représentation.

d. Analyse et interprétation: Une fois la segmentation d'une image effectuée, chaque région est représentée et décrite de manière appropriée pour un traitement informatique ultérieur. L'objectif de l'interprétation et de l'analyse d'image est d'identifier et d'évaluer diverses cibles présentes dans une image afin de collecter des informations pertinentes. Une cible est définie comme toute structure ou objet visible dans une image.

2.4.2 Applications du traitement d'images :

Le traitement d'images est devenu un outil puissant pour améliorer la qualité visuelle, extraire des données importantes ou des informations significatives, et automatiser des processus complexes dans divers domaines. Voici quelques-uns des principaux avantages du traitement d'images, quel que soit le secteur d'activité [21] :

- Il permet de disposer de l'image numérique dans le format souhaité (image améliorée, radiographie, négatif photo, IRM, etc.), ce qui facilite l'amélioration des images pour une meilleure interprétation humaine.
- Le traitement et l'extraction d'images peuvent permettre à la machine d'interpréter les informations.
- Il offre la possibilité de manipuler les pixels de l'image pour obtenir la densité et le contraste désirés.
- Les images peuvent être facilement stockées et récupérées.
- Il facilite la diffusion électronique des images vers des fournisseurs externes.
- Il permet de détecter et de suivre automatiquement des objets d'intérêt en temps réel.

2.5 Les expressions faciales

La reconnaissance des expressions faciales (REF) se concentre sur la classification des mouvements du visage et des déformations des traits faciaux à l'aide d'informations visuelles.

L'expression faciale est cruciale dans la communication et les comportements non verbaux, jouant un rôle essentiel en permettant de visualiser un changement sur le visage par l'activation d'un ou plusieurs des 44 muscles faciaux (offrant 250 000 expressions possibles) [22].

2.5.1 Les six expressions faciales universelles :

Au XXe siècle, les chercheurs Ekman et Friesen [23] ont identifié six émotions fondamentales à travers une étude culturelle et psychologique. Ils ont démontré que certaines émotions sont universellement ressenties par les individus, indépendamment de leur culture. Les tentatives de description des émotions humaines reposent sur l'idée que les êtres humains expriment de manière universelle un ensemble d'émotions primaires distinctes, telles que la tristesse, la colère, la peur, le dégoût, le bonheur et la surprise.

Cette hypothèse d'universalité a été largement adoptée en informatique affective en raison de sa simplicité et de sa prétention à l'universalité [24].

2.5.2 La reconnaissance automatique des expressions faciales

Ce domaine de recherche en intelligence artificielle, plus précisément en vision par ordinateur, vise à identifier et analyser les émotions exprimées par les personnes à travers leurs expressions faciales. Cette technologie trouve diverses applications dans des domaines tels que la sécurité, la santé, l'éducation et l'interaction homme-machine. La reconnaissance automatique des expressions faciales reste un défi complexe et captivant en vision par

ordinateur, car les individus peuvent exprimer leurs émotions et leurs réactions involontaires de manière très diverse [25].

2.5.3 Historique de reconnaissance des expressions faciales

L'étude de l'expression faciale, initiée dès le XVIIIe siècle par des chercheurs comme Darwin et Duchenne de Boulogne, a joué un rôle clé dans l'exploration des émotions. Les travaux de Sylvan Tomkins et de ses élèves Caroll Izard et Paul Ekman dans les années 1960 ont avancé l'idée qu'il existe un nombre limité d'émotions de base universelles et innées, associées à des expressions faciales automatiques et congénitales.

Au cours du XVIIIe siècle, le neurologue Duchenne de Boulogne a mené des expériences sur l'expression faciale des émotions. La photographie et la stimulation électrique des muscles du visage ont été utilisées pour mettre en évidence les mouvements liés à l'expression des émotions [22].

2.5.4 Les étapes de reconnaissance des expressions faciales

Un système d'analyse d'expression faciale comprend généralement trois étapes principales. La détection du visage est la première étape, qui permet de définir la zone de recherche. Ensuite, il est nécessaire d'extraire les informations requises pour mieux décrire l'expression. Enfin, à partir de ces données, l'image est attribuée à une catégorie d'expressions à l'aide d'un classificateur [22] :

2.5.4.1 Détection de visage

Cette étape consiste à localiser le visage dans une image. La détection du visage peut être réalisée à l'aide de techniques basées sur la couleur de la peau, la forme de la tête ou la détection de différentes caractéristiques du visage. Cependant, cette étape devient plus complexe lorsque l'image contient plusieurs éléments faciaux ou un fond non homogène, ce

qui peut perturber la segmentation précise du visage. Le succès de cette étape dépend de la qualité des images obtenues [26].

Les techniques de détection de visages :

Plusieurs techniques de détection de visages ont été développées ces dernières années, classées en quatre catégories :

1.Techniques basées sur les connaissances acquises ;

2.Techniques basées sur les caractéristiques invariantes ;

3.Techniques basées sur la mise en correspondance ;

4.Techniques basées sur l'apprentissage.

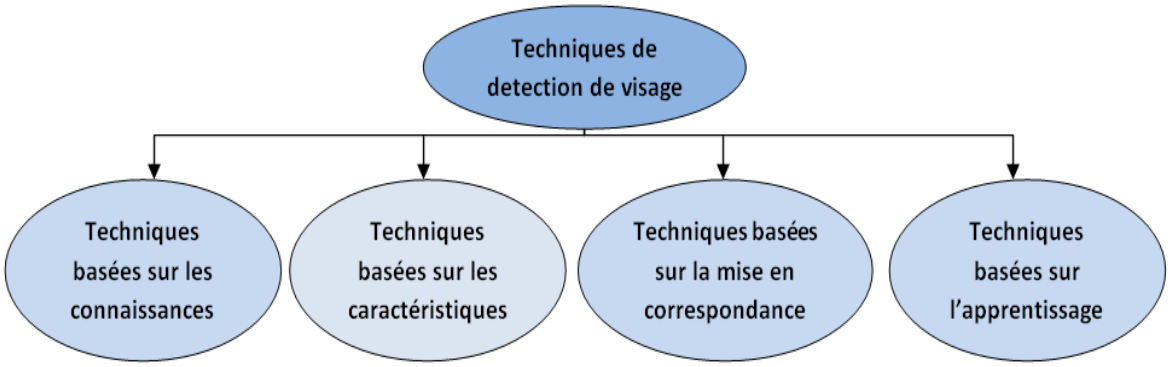


Figure 2.2: Technique de détection de visage

a. Techniques basées sur les connaissances acquises :

Ces approches reposent sur la compréhension des caractéristiques du visage et des relations entre ses différentes parties. Voici quelques exemples qui peuvent servir à tester la présence d'un visage sur une image :

- Le centre du visage présente des valeurs d'intensité lumineuse uniformes ;

- Un visage est souvent représenté par deux yeux symétriques, un nez et une bouche ;
- Les liens entre les diverses parties du visage peuvent être illustrés par leurs distances et leurs positions respectives.

La difficulté de ces méthodes réside dans la complexité de traduire la connaissance des caractéristiques du visage humain en règles précises. De plus, il est difficile d'étendre cette méthode pour détecter des visages dans diverses orientations, car il n'est pas possible d'énumérer tous les cas. Cependant, ces techniques sont efficaces pour détecter les visages de face [22].

b. Techniques basées sur les caractéristiques invariantes:

Les algorithmes de cette catégorie se basent sur le fait que les traits du visage ne sont pas modifiés, même en cas de changement d'orientation du visage, d'angle de prise de vue ou de conditions d'éclairage. Cette catégorie inclut des méthodes basées sur la détection de la couleur de la peau, la détection de diverses caractéristiques faciales, etc.

Ces méthodes sont inefficaces lorsque les images d'entrée sont de faible qualité ou lorsque certaines parties du visage sont dissimulées [22].

c. Techniques basées sur la mise en correspondance :

Les méthodes utilisées ici sont basées sur des motifs de visage pré-établis et soigneusement sélectionnés. Cette approche consiste à évaluer la relation entre des zones de l'image d'entrée et les motifs de visages.

Ces techniques ont l'avantage d'être faciles à mettre en œuvre [22].

d. Techniques basées sur l'apprentissage :

Le problème de la détection de visages est abordé par les méthodes de cette catégorie comme un problème de classification. Des techniques d'apprentissage automatique supervisé sont utilisées pour déterminer si une image appartient à la classe des visages ou à celle des non-visages. Ainsi, un ensemble de données d'apprentissage est constitué d'images représentant des visages et d'images ne représentant pas des visages, ce qui permet de définir une règle de séparation entre les deux classes.

Actuellement, ce sont les méthodes les plus performantes et les plus adaptées pour la détection de visages en temps réel [22].

2.5.4.2 Extraction des caractéristiques du visage :

L'extraction des caractéristiques est une étape essentielle de notre système. Son objectif est d'identifier les éléments pertinents qui décrivent les expressions faciales. Cela peut être réalisé de deux manières différentes : La première méthode consiste à extraire des parties complètes du visage, généralement à l'aide d'une méthode de reconnaissance universelle. La seconde approche, appelée reconnaissance locale, identifie quelques éléments distinctifs dans des zones spécifiques du visage, comme la bouche et le nez [27].

2.5.5 Le fonctionnement d'un système d'analyse des expressions faciales

Dans un système de détection des expressions faciales, une image suit un processus bien défini dès son entrée pour déterminer les expressions faciales d'un sujet. Ce processus comprend différentes étapes [28], comme le montre le schéma ci-dessous :

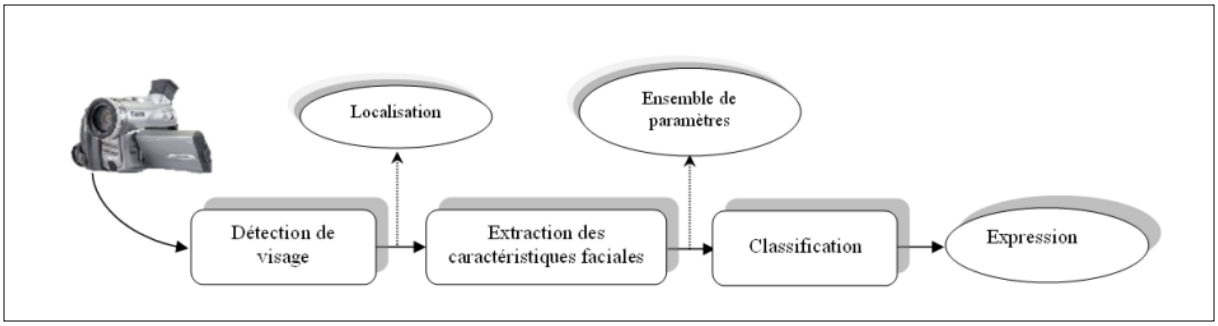


Figure 2.3: Système d'analyse des expressions faciales [22]

1.Le monde physique :

Il s'agit de l'environnement extérieur au système avant l'acquisition de l'image. Généralement, trois paramètres sont pris en compte à cette étape : l'éclairage, la variation de posture et l'échelle.

2. L'acquisition de l'image:

À cette étape, il est nécessaire d'extraire l'image du monde extérieur, que ce soit de manière statique avec un appareil photo ou dynamique avec une caméra. Cela présente un risque de bruit et entraîne une représentation 2D (une structure de niveaux de gris) pour le visage, qui est une structure 3D (RGB).

3. Les prétraitements

Cette étape vise à éliminer le bruit et les interférences, souvent causés par la qualité des équipements optiques ou électroniques lors de l'acquisition d'images. Elle utilise des méthodes de traitement et de restauration d'images pour ne conserver que les données essentielles, préparant ainsi l'image pour les étapes ultérieures. Cette phase est cruciale, car il est impossible d'obtenir une image totalement exempte de bruit en raison de facteurs environnementaux comme le fond et la lumière, qui sont généralement inconnus. Divers traitements et améliorations de la qualité de l'image peuvent être appliqués, tels que la

normalisation (ajustement de la taille), l'égalisation et le filtre médian. La détection et la localisation du visage dans une image peuvent également être intégrées à cette étape, permettant d'isoler et de recadrer la région pertinente.

4. L'extraction des paramètres

L'extraction des paramètres est une étape fondamentale du système de reconnaissance. Elle consiste à transformer l'image dans un environnement de travail plus simple, ce qui facilite l'exploitation des données et permet de se concentrer uniquement sur les informations pertinentes.

5. La classification

La procédure de classification utilise les caractéristiques extraites pour déterminer l'expression faciale. Des modèles statistiques comme les machines à vecteurs de support (SVM), les modèles de mélange gaussien (GMM) ou les arbres de décision peuvent être employés. Cependant, les réseaux de neurones, en particulier les réseaux de neurones convolutifs (CNN) et récurrents (RNN), sont souvent privilégiés en raison de leur capacité à apprendre des caractéristiques complexes et à produire des résultats plus précis.

6. L'apprentissage

Pour garantir l'efficacité des modèles de classification, il est impératif de les entraîner sur des ensembles de données étiquetés. Les bases de données utilisées dans l'apprentissage supervisé contiennent des images de visages annotées avec les expressions correspondantes. En entraînant les modèles sur ces données, ils peuvent identifier les motifs et les caractéristiques associés à chaque expression.

7. La décision

La dernière étape est la décision finale, où les résultats de la classification sont interprétés et utilisés pour prendre des mesures appropriées. Généralement, cette étape implique l'analyse des informations issues de la classification des expressions faciales pour prendre des décisions éclairées.

2.5.6 Difficultés de la reconnaissance des visages :

- La variation de la position : La meilleure condition pour détecter les visages est que l'image soit frontale.
- Le blocage d'éléments : La présence de choses telles que des barbes, des lunettes ou des chapeaux apporte une grande diversité. Il peut aussi y avoir une partie des visages couverte par des objets ou d'autres visages.
- Les expressions faciales : Les expressions du visage présentent également une grande diversité en raison des divers gestes réalisés.
- Les exigences de l'imagerie : Les différentes caméras et les conditions, car les conditions d'éclairage et le type de caméra ont un impact sur la qualité d'une image, ce qui a un impact sur l'apparence d'un visage.

2.5.7 Différentes méthodes développées pour la reconnaissance faciale

Il existe diverses méthodes qui ont été élaborées dans ce domaine. On peut distinguer deux grandes catégories : les méthodes locales et les méthodes globales (ou holistiques).

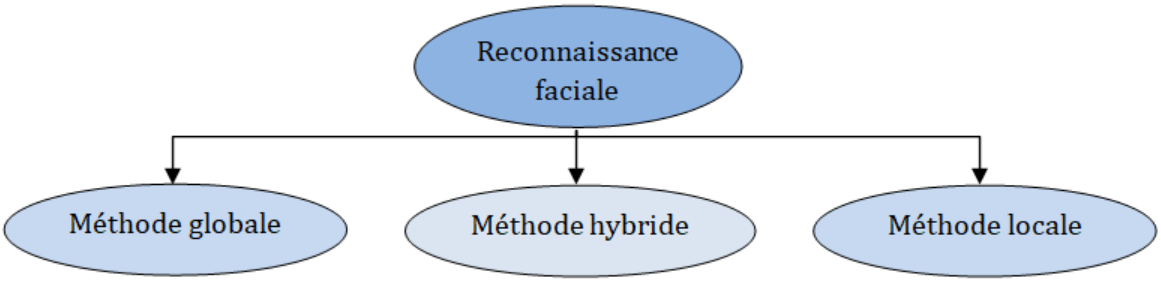


Figure 2.4: Méthodes de reconnaissance faciale

1. Méthodes globales:

Ces méthodes reposent sur des méthodes d'analyse statistique bien établies. Pour normaliser les images, il n'est pas nécessaire d'identifier certains points spécifiques du visage (centres des yeux, narines, centre de la bouche, etc.). Les images de visage (qui peuvent être considérées comme des matrices de valeurs de pixels) sont traitées dans ces approches de manière globale et sont généralement converties en vecteurs, c'est à dire exploitent l'ensemble du visage comme point d'entrée de l'algorithme de reconnaissance. Elles sont plus simples à manipuler. Le principal avantage des méthodes globales réside dans leur rapidité de mise en place et leur complexité moyenne dans les calculs de base. Par contre, elles sont extrêmement vulnérables aux fluctuations de l'éclairage, de la pose et de l'expression faciale [29].

2. Méthodes locales:

Ces approches reposent sur des modèles, s'appuyant sur des connaissances préalables que l'on possède sur la morphologie du visage et se fondent généralement sur des aspects spécifiques de celui-ci. Ce modèle correspond à la perception du visage par l'homme, c'est-à-dire à nos représentations de traits de visage et de parties telles que les yeux, le nez, la bouche, etc. [29]

Elles représentent une autre façon de prendre en considération la non-linéarité en créant un espace de caractéristiques locales et en utilisant des filtres d'images adaptés, afin de réduire les variations des distributions des visages.

Par rapport aux méthodes globales, ces techniques présentent l'avantage de permettre une modélisation plus aisée des variations de dépose, d'éclairage et d'expression. Cependant, leur utilisation est plus difficile, car il est souvent nécessaire de placer manuellement un certain nombre de points sur le visage, tandis que les méthodes globales ne requièrent que la position des yeux pour normaliser les images, ce qui peut être automatiquement et de manière assez fiable réalisé par un algorithme de détection des yeux.

3. Méthodes hybrides:

Il s'agit d'une association des méthodes globales et locales qui combinent la détection de caractéristiques géométriques (ou structurales) globales avec l'extraction de caractéristiques d'apparence locales. [29]

2.5.8 Les algorithmes de la détection faciale :

a) 68 Facial Landmark :

Les points de repère faciaux (Facial landmarks) sont des points clés le long de la forme du visage détecté.Ils peuvent être utilisés comme caractéristiques du visage pour effectuer plusieurs tâches telles que l'amélioration de la reconnaissance faciale, l'alignement des images faciales, la distinction entre les hommes et les femmes, l'estimation de la pose de la tête et, bientôt, l'analyse de la position du visage.

Afin d'effectuer la détection des visages et d'extraire les repères faciaux d'une image, il est nécessaire d'utiliser la bibliothèque dlib. Le détecteur de Facial landmarks est une implémentation de l'approche présentée par Kazemi et al [30]. Il renvoie un tableau de 68

points sous forme de coordonnées (x,y) qui correspondent à des structures faciales du visage, comme le montre la figure ci-dessous (figure 17).

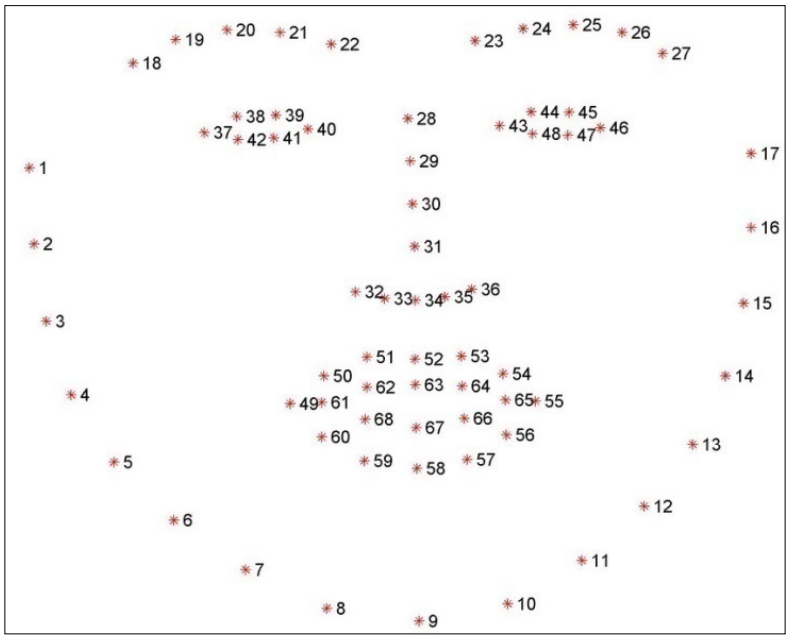


Figure 2.5 : 68 Facial Landmark

b) Viola & Jones :

L'algorithme Viola & Jones est nommé en l'honneur de deux chercheurs en vision par ordinateur, Paul Viola et Michael Jones, qui ont présenté la méthode dans le journal scientifique intitulé International Journal of Computer Vision (IJCV) qui a été publié en 2001. Initialement conçue pour repérer des visages, la méthode de Viola & Jones peut également être employée pour repérer d'autres objets en temps réel avec une vitesse remarquable.

Dans cet algorithme, Viola et Jones ont employé des caractéristiques de type Haar afin de repérer les visages, et il fonctionne sur une image en niveaux de gris.

• Les fonctionnalités de type Haar-cascade:

Haar cascades est une méthode populaire de détection d'objets en temps réel, souvent utilisée pour la détection de visages dans des images ou des flux vidéo. La méthode a été

introduite par Paul Viola et Michael Jones en 2001, dans leur célèbre article intitulé "Rapid Object Detection using a Boosted Cascade of Simple Features"[31].

Cette approche utilise les caractéristiques Haar, qui sont des masques simples permettent de prendre en compte les variations de contraste dans une image, ainsi qu'une série de classificateurs entraînés pour repérer ces caractéristiques. Les régions non pertinentes de l'image sont rapidement rejetées grâce au principe de la cascade, ce qui rend la méthode particulièrement rapide et efficace pour des applications telles que la détection de visages.

Cette méthode basée sur les différences de sommes de pixels calculées dans des zones rectangulaires.

Dans notre projet nous avons utilisé deux classificateurs Haar, le premier pour détecter le visage et le deuxième pour les yeux.

c) HOG & Linear SVM :

Les SVM sont des modèles d'apprentissage automatique supervisés populaires en vision par ordinateur, ils ont efficaces pour séparer et classer les données. On peut définir le modèle SVM comme un espace à haute dimension de points où plusieurs classes sont séparées grâce à l'utilisation d’un hyperplan optimal.

Un histogramme de gradients orientés (HOG) est une technique de description d'images utilisée pour la reconnaissance d'objets dans le domaine de vision par ordinateur. HOG analyse la distribution des gradients (les variations de luminosité) et leurs orientations dans une région localisée de l'image. En règle générale, il divise l'image en petites régions appelées cellules et compare les pixels de cette zone les uns aux autres, cherche à mesurer la variation de l'obscurité, puis à déterminer la direction où se produit la plus importante modification.

2.6 Conclusion :

En conclusion, la reconnaissance faciale est une technologie en constante évolution, promettant des applications vastes et diverses. Dans la partie précédente, nous avons discuté les concepts fondamentaux de la reconnaissance faciale ainsi que l'importance de la vision par ordinateur dans la reconnaissance des expressions faciales.

La détection de la somnolence est un domaine particulièrement prometteur où la reconnaissance faciale joue un rôle essentiel. Cette technologie peut détecter les signes de fatigue chez les conducteurs grâce à ses compétences en interprétation des expressions faciales et en suivi des mouvements des yeux. En associant ces données à des modèles d'apprentissage automatique, les systèmes peuvent informer les conducteurs en temps réel afin de prévenir les accidents causés par la fatigue au volant, ce qui permet de sauver des vies et d'améliorer considérablement la sécurité routière.

Le chapitre suivant approfondira cette application spécifique en mettant en lumière comment la reconnaissance faciale contribue activement à la sécurité routière en détectant la somnolence.

CHAPITRE III

ÉTAT DE L'ART SUR LES APPROCHES DE DETECTION DE LA SOMNOLENCE

3.1 Introduction :

D'après les dernières données publiées par l'*Association for Safe International Road Travel* (ASIRT) [33], les accidents de la circulation constituent l'une des principales causes de mortalité à l'échelle mondiale, engendrant des pertes humaines et économiques considérables. En effet, ces accidents provoquent environ 1,2 million de décès par an et des pertes financières estimées à 518 milliards de dollars annuellement, soit une moyenne quotidienne de 3 287 victimes et 1,4 milliard de dollars.

Une étude complémentaire menée par *Mariam B. et Laurent C.* [35] confirme une augmentation progressive du nombre de tués et de blessés entre 1970 et 2015, soulignant ainsi l'urgence d'une meilleure politique de sécurité routière.

Le contexte algérien ne déroge pas à cette tendance mondiale. Selon les statistiques du *Centre National de Prévention et de Sécurité Routières* (CNPSR) [34], 4 812 personnes ont perdu la vie dans des accidents de la route en 2014.

3.2. Relation entre la somnolence et la conduite :

La somnolence se caractérise par une baisse du niveau de vigilance, se manifestant par des difficultés à maintenir un état d'éveil, bien que la personne puisse être temporairement réactivée par des stimuli externes. Ses origines sont multifactorielles :

- **Manque de sommeil** (privations ou troubles du rythme circadien).
- **Effets secondaires de médicaments** (sédatifs, antihistaminiques, etc.).
- **Consommation de substances** (alcool, drogues).
- **Pathologies neurologiques** (apnée du sommeil, narcolepsie).

Principalement induite par la fatigue mentale ou physique, la somnolence altère significativement les capacités cognitives et motrices.

Impact sur la conduite automobile

Toute activité prolongée – notamment la conduite – entraîne une dégradation progressive des performances, même chez les conducteurs expérimentés. La somnolence au volant se traduit par :

- Un ralentissement des temps de réaction.
- Une diminution de la concentration.
- Des erreurs de jugement (distances, trajectoires).

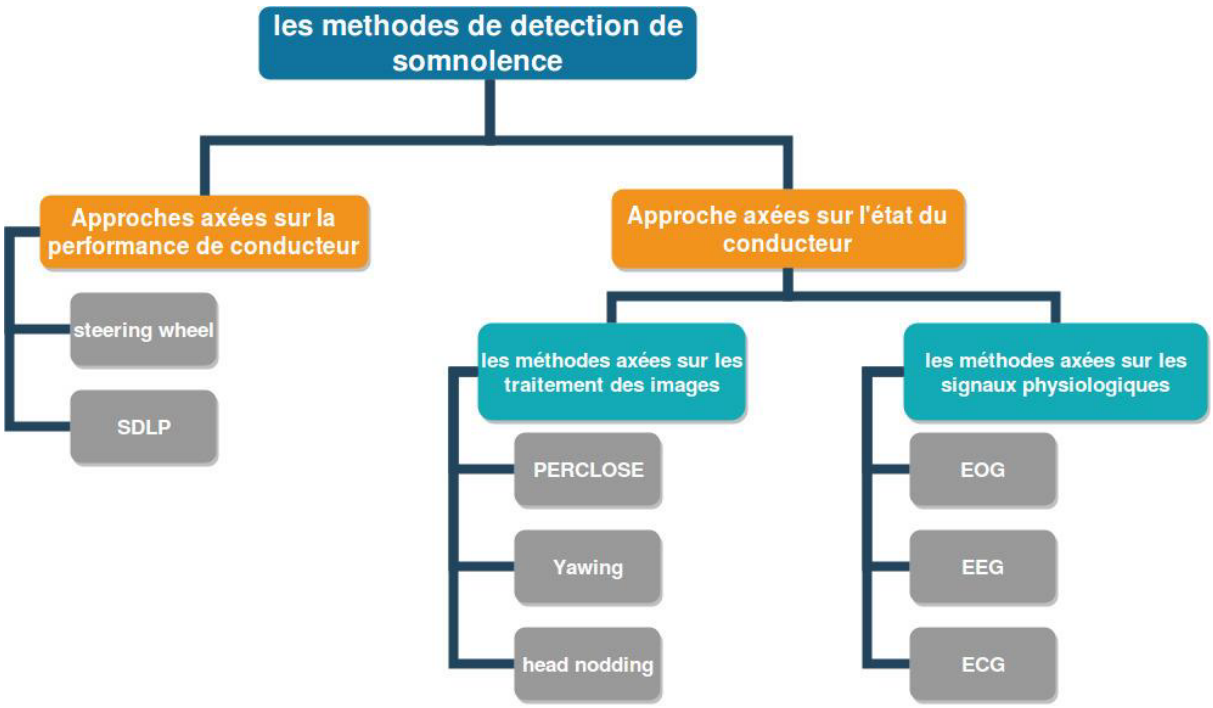
Nécessité d'une surveillance continue :

Pour prévenir les accidents, une détection en temps réel de l'état de vigilance du conducteur s'avère essentielle. Les systèmes actuels visent à identifier les signes précurseurs de somnolence (clignements des paupières, mouvements erratiques du volant) afin d'alerter le conducteur avant que la situation ne devienne critique.

3.3. Approches de détection de somnolence

Les approches technologiques pour identifier la somnolence se divisent en deux catégories principales :

1. **Méthodes basées sur les performances du conducteur** :
    - Analyse du comportement de conduite (trajectoire, freinage, etc.).
2. **Méthodes basées sur l'état physiologique du conducteur** :
    - **Signaux physiologiques** (rythme cardiaque, activité cérébrale via EEG).
    - **Traitement d'images** (suivi des paupières, expressions faciales, mouvements oculaires).



La Figure 3.1 présente les méthodes de détection de somnolence.

3.3.1. Approches axées sur la performance du conducteur

Les méthodes basées sur la performance du conducteur reposent sur l'analyse des modifications du comportement de conduite. Ces approches considèrent deux catégories de facteurs : les paramètres liés à la tâche de conduite (changement de vitesse, accélération, trajectoire dans les virages) et les caractéristiques individuelles du conducteur (expérience, habitudes). Pour surveiller ces indicateurs, des capteurs sont installés sur différents composants du véhicule comme le volant et les pédales.

- Parmi les mesures les plus utilisées, l'analyse des mouvements du volant a fait l'objet de plusieurs recherches. Krajewski et al. [37] ont développé un système qui extrait les caractéristiques temporelles, fréquentielles et contextuelles des mouvements du volant. Leur approche utilise des algorithmes d'apprentissage machine comme les SVM (Support Vector Machine) [38] et KNN (K-Nearest Neighbors) [39] pour classer ces caractéristiques, atteignant une précision de 86,1%.
- Une autre mesure couramment employée est l'écart-type de la position latérale (SDLP). Cette métrique évalue les déviations du véhicule par rapport à sa trajectoire idéale. M. Inger et al. [40] ont étudié la relation entre le SDLP et l'échelle de somnolence de Karolinska (KSS) [41], démontrant une corrélation directe entre ces deux indicateurs. Leurs travaux montrent qu'une augmentation du SDLP correspond systématiquement à une élévation du niveau de fatigue mesuré par le KSS.

Ces approches utilisent généralement des caméras pour surveiller la voie ou des capteurs intégrés dans les composants du véhicule. Elles permettent une évaluation indirecte de l'état de vigilance du conducteur en analysant son comportement au volant.

3.3.2. Approches axées sur les signaux physiologiques

Les méthodes basées sur les signaux physiologiques exploitent les modifications corporelles induites par la somnolence, en particulier au niveau cardiaque (ECG), oculaire (EOG) et cérébral (EEG). Ces approches nécessitent généralement le port de capteurs spécifiques par le conducteur. L'électro-oculographie (EOG) permet de détecter les mouvements oculaires caractéristiques de la fatigue à l'aide d'algorithmes de classification comme les SVM [42], PNN et MLP [43]. L'électrocardiographie (ECG) analyse la variabilité du rythme cardiaque, où une augmentation anormale constitue un indicateur fiable de diminution de vigilance [44]. Quant à l'électroencéphalographie (EEG), elle fournit une mesure directe de l'état de vigilance grâce à

l'analyse des bandes spectrales : delta (0.5-4 Hz, sommeil profond), thêta (4-8 Hz, somnolence), alpha (8-13 Hz, relaxation) et bêta (13-25 Hz, veille active) [45]. Bien que ces méthodes offrent une évaluation précise de l'état de fatigue, leur application en conditions réelles de conduite reste limitée par le caractère invasif des dispositifs de mesure.

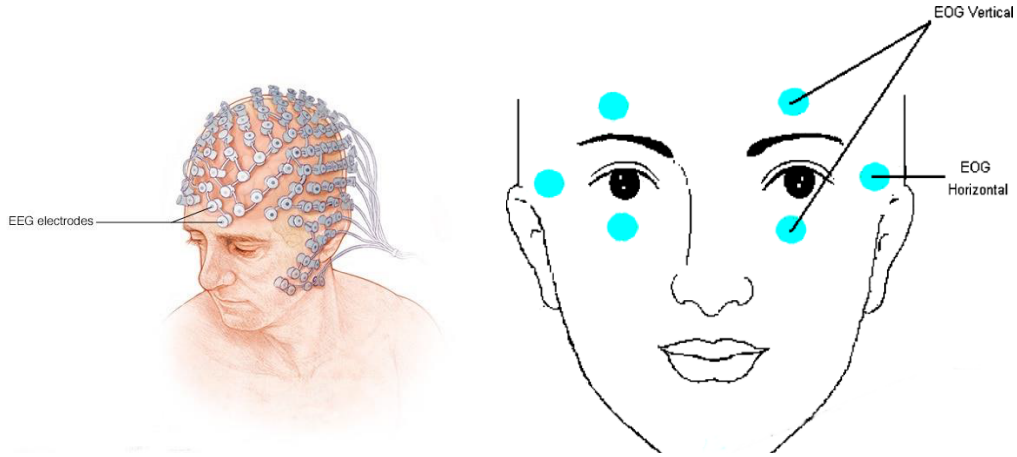


Figure 3.2:capteurs utilisés dans les méthodes axées sur la performance du conducteur **[42]**.

3.3.3. Approches axées sur le traitement d'image

Contrairement aux méthodes intrusives nécessitant le port de capteurs, les approches basées sur le traitement d'image représentent une solution non invasive pour détecter la somnolence au volant. Ces techniques exploitent des caméras embarquées pour analyser en temps réel les manifestations physiques de la fatigue chez le conducteur, notamment :

- Les mouvements oculaires (clignements, fermeture des paupières)
- Les bâillements fréquents
- Les mouvements caractéristiques de la tête

Parmi les différents indicateurs visuels, la mesure PERCLOS (Percentage of Eye Closure) [46] s'est imposée comme la métrique la plus fiable pour évaluer l'état de somnolence. Cette mesure quantitative, calculant le pourcentage de temps où les paupières couvrent plus de 80% de la pupille sur une période donnée, a été validée par de nombreuses études comme corrélée de manière significative avec le niveau de fatigue.

Les systèmes modernes combinent généralement :

1) Un module de détection faciale pour localiser les régions d'intérêt (yeux, bouche)
2) Des algorithmes de suivi oculaire précis

3) Une analyse temporelle des paramètres PERCLOS [46]

4) Des classifieurs pour interpréter les signaux visuels

Cette approche présente l'avantage majeur de ne pas perturber le conducteur tout en offrant une bonne précision de détection, ce qui explique son adoption croissante dans les véhicules haut de gamme.

- Garcia et al. [47] ont développé un système de détection de somnolence en trois phases principales. Dans un premier temps, le système utilise l'algorithme de Viola-Jones [32] pour détecter le visage du conducteur et localiser ses yeux lors de la phase de prétraitement. Ensuite, la phase d'analyse affine la détection en localisant précisément les pupilles et en estimant le degré d'ouverture des yeux grâce à une technique de projection intégrale. Enfin, le système détermine l'état de vigilance du conducteur en calculant la mesure PERCLOS (Percentage of Eye Closure), qui évalue le niveau de fatigue en fonction du taux de fermeture des paupières.
- W. Zhang et al. [48] ont développé un système de détection de somnolence utilisant six paramètres oculaires : PERCLOS, durée et fréquence de clignement, ouverture moyenne des yeux, ainsi que les vitesses d'ouverture et de fermeture des paupières. Ces mesures sont analysées par un Discriminant Linéaire de Fisher pour différencier efficacement les états de vigilance et de fatigue. Cette approche multi-critères permet une détection plus précise que les méthodes utilisant un seul indicateur.
- Adrian [49] a développé une méthode de détection de fatigue utilisant l'Eye Aspect Ratio (EAR). Après détection faciale et localisation des yeux, le système calcule en continu l'EAR pour évaluer leur ouverture. Une alarme se déclenche si les yeux restent fermés pendant 48 images consécutives (environ 2 secondes), offrant une solution simple, robuste aux variations lumineuses et peu coûteuse en calcul.
- Reza Ghoddoosian et al. [50] ont conçu un système simplifié analysant trois indicateurs oculaires: fréquence de clignement, ouverture moyenne et vitesse d'ouverture des yeux. Ces paramètres temporels sont traités par un réseau LSTM, optimisé pour détecter les schémas de fatigue dans des séquences vidéo. Cette solution allège la complexité computationnelle tout en maintenant une bonne précision de détection.
- Dariusz Sawicki [51] a développé une approche innovante pour la détection des bâillements, symptôme important de fatigue souvent négligé. Son système utilise d'abord des algorithmes de détection des repères faciaux pour localiser précisément les lèvres, puis calcule le rapport

hauteur/largeur de l'ouverture buccale. Cette méthode permet de distinguer efficacement les bâillements (indicateurs de fatigue) des simples mouvements de bouche liés à la parole ou aux expressions faciales. L'approche complète ainsi les systèmes de détection basés sur les mouvements oculaires, offrant une analyse plus globale de l'état de vigilance. Son principal avantage réside dans sa capacité à réduire les faux positifs tout en maintenant une complexité algorithmique modérée, ce qui en fait une solution pratique pour les systèmes embarqués.

- M. Omidyeganeh et al. [52] ont développé une approche alternative pour l'analyse des bâillements en combinant plusieurs techniques de vision par ordinateur. Leur système repose sur deux phases principales : une détection initiale du visage et de la bouche utilisant l'algorithme de Viola-Jones, suivie d'une analyse dynamique des mouvements buccaux via la théorie de back-projection. Cette méthode permet de quantifier simultanément la fréquence et l'amplitude des changements buccaux, offrant ainsi une évaluation plus nuancée de l'état de fatigue du conducteur. L'approche se distingue par sa capacité à capturer à la fois les aspects temporels (taux d'ouverture/fermeture) et spatiaux (ampleur des mouvements) des bâillements, fournissant ainsi des indicateurs complémentaires aux méthodes purement morphologiques. Cette solution illustre la diversité des techniques disponibles pour l'analyse du même symptôme de fatigue, avec des compromis différents en termes de complexité et de précision.
- Les mouvements de tête représentent également un indicateur pertinent pour évaluer l'état de vigilance du conducteur. Lam Thanh Hien et Do Nang Toan [53] ont développé une méthode basée sur l'analyse géométrique de points faciaux clés. Leur approche commence par détecter le visage puis extrait cinq repères caractéristiques : les deux coins externes des yeux, deux points marquant les extrémités de la bouche et la pointe du nez. À partir de ces points, le système calcule trois types de paramètres : les distances entre points, les distances entre points et contours du visage, ainsi que la surface du triangle formé par les coins de la bouche et l'extrémité nasale. L'évaluation finale de la somnolence s'effectue en comparant ces mesures à des seuils prédéterminés, permettant ainsi de détecter les mouvements caractéristiques de la fatigue comme l'affaissement ou les oscillations anormales de la tête.

Plusieurs études récentes proposent des systèmes combinant plusieurs indicateurs physiologiques pour une détection plus fiable. M. Omidyeganeh et al. [54] associent les bâillements et le PERCLOS, tandis qu'Esra Vural et al. [55] intègrent trois paramètres : bâillements, mouvements de tête et clignements oculaires. Ces approches multimodales utilisent les mêmes techniques d'extraction de caractéristiques que les méthodes unimodales, mais améliorent significativement la précision en croisant plusieurs signes de fatigue. Leur principal avantage réside dans la réduction

des faux positifs grâce à une analyse plus complète du comportement du conducteur.

Plus récemment, de nouvelles approches basées sur l'apprentissage en profondeur, notamment l'utilisation des réseaux de neurones convolutifs (CNN) et récurrents (RNN), ont vu le jour. Ces techniques représentent une avancée majeure, en particulier dans le domaine de la vision par ordinateur, où elles sont largement utilisées pour des tâches telles que la classification d'images, la détection d'objets, la reconnaissance des émotions ou encore la segmentation de scènes. En s'appuyant sur ces concepts, plusieurs méthodes innovantes ont été développées pour la détection de la somnolence, tirant parti de la puissance des CNN et RNN pour extraire automatiquement des caractéristiques pertinentes à partir de données visuelles complexes.

Bhargava Reddy et al. [56] ont proposé un système de détection de la somnolence basé sur les réseaux de neurones convolutifs (CNN). Ce système se compose de deux étapes principales. Dans un premier temps, ils utilisent le réseau MTCNN (Multi-Task Cascaded Convolutional Neural Networks) pour détecter les éléments clés du visage tels que les yeux, le nez, les lèvres gauche et droite, entre autres. Les informations extraites lors de cette première étape servent ensuite d'entrée à un second réseau CNN, chargé d'analyser ces caractéristiques faciales pour en extraire l'état et, finalement, prédire l'état de vigilance du conducteur.

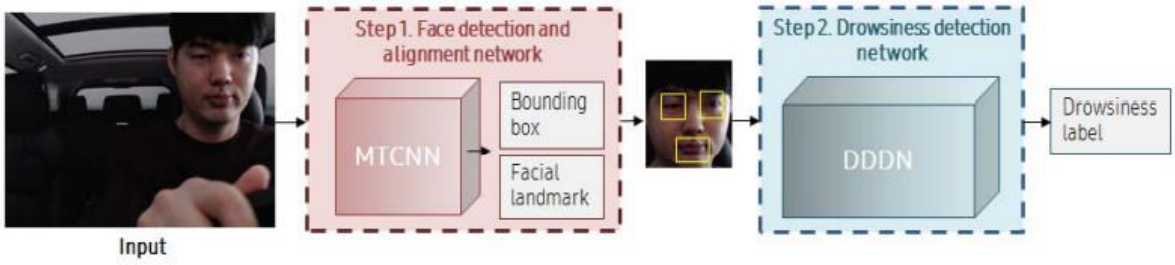


Figure 3.3: Architecteur utilisé dans l'approche de Bhargava Reddy et al **[56]**.

Un autre système, développé par Nisha Gandhi et al. [57], combine deux types d'apprentissage en profondeur pour améliorer la détection de la somnolence. Dans cette approche, un réseau de neurones convolutifs (CNN) est utilisé pour déterminer l'état des yeux, qu'ils soient ouverts ou fermés. Par la suite, un réseau de neurones récurrent (RNN) est employé pour analyser la séquence temporelle de ces états et en déduire l'état général du conducteur. Cette combinaison permet de mieux capturer les dynamiques temporelles liées à la somnolence, offrant ainsi une prédiction plus fiable.

Le tableau 3.1 synthétique les travaux mentionnés, mettant en évidence les auteurs, méthodes utilisées, paramètres analysés, avantages, et limites éventuelles :

Auteur(s)	Méthode utilisée	Indicateurs analysés	Avantages	Limites
Garcia et al. [47]	Détection faciale (Viola-Jones), projection intégrale, PERCLOS	PERCLOS (fermeture des yeux)	Bonne précision, méthode validée cliniquement	Basée sur un seul paramètre
W. Zhang et al. [48]	Analyse multi-paramètres + Discriminant Linéaire de Fisher	PERCLOS, clignements (durée/fréquence), ouverture/fermeture paupières	Détection plus précise, robuste	Complexité de traitement plus élevée
Adrian [49]	Eye Aspect Ratio (EAR)	EAR, détection fermeture prolongée des yeux	Simplicité, faible coût, robuste à la lumière	Seulement basé sur les yeux
Reza Ghoddoosian et al. [50]	RNN (LSTM), analyse temporelle	Fréquence de clignement, ouverture moyenne, vitesse d'ouverture	Moins complexe, efficace dans les vidéos	Paramètres limités, dépend du suivi oculaire précis
Dariusz Sawicki [51]	Analyse géométrique des lèvres	Rapport hauteur/largeur buccale (bâillement)	Réduction des faux positifs, complémentaire aux systèmes oculaires	Nécessite détection buccale précise
M. Omidyeganeh et al. [52]	Viola-Jones + back-projection	Bâillements (fréquence et amplitude)	Analyse spatiale et temporelle, plus nuancé	Complexité modérée

Lam Thanh Hien & Do Nang Toan [53]	Analyse géométrique des repères faciaux	Mouvements de tête (distances, surfaces)	Capte des signes négligés, robuste	Moins précis seul, dépend de la stabilité faciale
M. Omidyeganeh et al. [54]	Fusion d'indicateurs	Bâillements + PERCLOS	Réduction des faux positifs	Augmentation de la complexité
Esra Vural et al. [55]	Fusion d'indicateurs	Bâillements, clignements, mouvements de tête	Haute précision, multimodal	Besoin de synchronisation des signaux
Bhargava Reddy et al. [56]	CNN + MTCNN	Détection faciale (yeux, nez, lèvres) + CNN prédictif	Automatisation, détection robuste	Besoin de ressources matérielles importantes
Nisha Gandhi et al. [57]	CNN + RNN	État des yeux (CNN) + séquence temporelle (RNN)	Capture des dynamiques temporelles	Plus coûteux en calcul, dépendant de la qualité vidéo

Tableux 3.1 : synthèse des travaux basé sur la traitement d'image.

2.4. Conclusion

Ce chapitre a présenté les principaux types de systèmes de détection automatique de la somnolence chez le conducteur, en les classant en trois catégories distinctes. Les méthodes basées sur les performances du conducteur sont non intrusives, mais leur fiabilité peut être fortement influencée par des facteurs externes tels que le type de véhicule, les conditions routières ou encore l'expérience du conducteur. Les méthodes fondées sur les signaux physiologiques offrent de bons résultats en termes de précision, toutefois, leur nature intrusive, nécessitant le port de capteurs, peut perturber le confort du conducteur. Enfin, les approches basées sur le traitement d'image, bien que techniquement plus complexes, se distinguent par leur caractère non intrusif et leur capacité à fournir des résultats fiables. Ces dernières représentent une solution prometteuse, notamment grâce aux avancées récentes en vision par ordinateur et en apprentissage profond.

CHAPITRE IV

CONCEPTION ET IMPLEMENTATION

4.1 Introduction

Ce chapitre pratique explore les étapes nécessaires pour réaliser un système de détection de somnolence au volant, en intégrant des techniques avancées de traitement d'images et de détection en temps réel.

En plus dans ce chapitre, nous détaillerons les différentes phases de développement du système, y compris la configuration des outils nécessaires, la création des algorithmes de détection, et l'intégration des composants matériels et logiciels pour aboutir à une solution fonctionnelle.

4.2 Conception

La somnolence est une perte de conscience qui se manifeste par une somnolence et des difficultés à rester éveillé, mais la personne peut être réveillée par des stimuli faciles. La plupart des gens se sentent somnolents parce qu'ils sont fatigués, ce qui peut être à la fois mental et physique. D'autre part, si vous faites quelque chose pendant longtemps, comme conduire, il sera difficile de maintenir un niveau de performance élevé. Aussi, pour arrêter ou réduire le nombre d'accidents, le niveau de sommeil d'un conducteur doit toujours être vérifié et pour résoudre ce problème on a proposé un système de la détection de somnolence des conducteurs avec l'utilisation de l'apprentissage profond

Le but général de notre projet est du construire un système de la détection de somnolence basé sur l'utilisation des réseaux de neurone approfondi, nous avons utilisé Les réseaux de neurones convolutionnels CNN pour détecter le statut de conducteur (somnolence ou Vigilant). L'entrer de notre system est une séquence des trames (vidéo) qui a acquis à partir d'une caméra. Puis le système analyse chaque image séparément. Ce processus se fait dans 4 étapes principales : Premièrement, on détecte le visage dans chaque image. Après, on détecte les deux yeux (gauche et

droit) du visage. Par la suite, on détecte l'état des yeux. Finalement on détecte l'état du conducteur. Dans ce chapitre nous avons résumé notre système et testé le résultat de simulation.

Le schéma ci-dessous (Figure 4.1) illustre les différentes étapes suivies dans notre système pour la détection de la somnolence au volant, allant de la détection de visage à l'analyse de l'état des yeux.

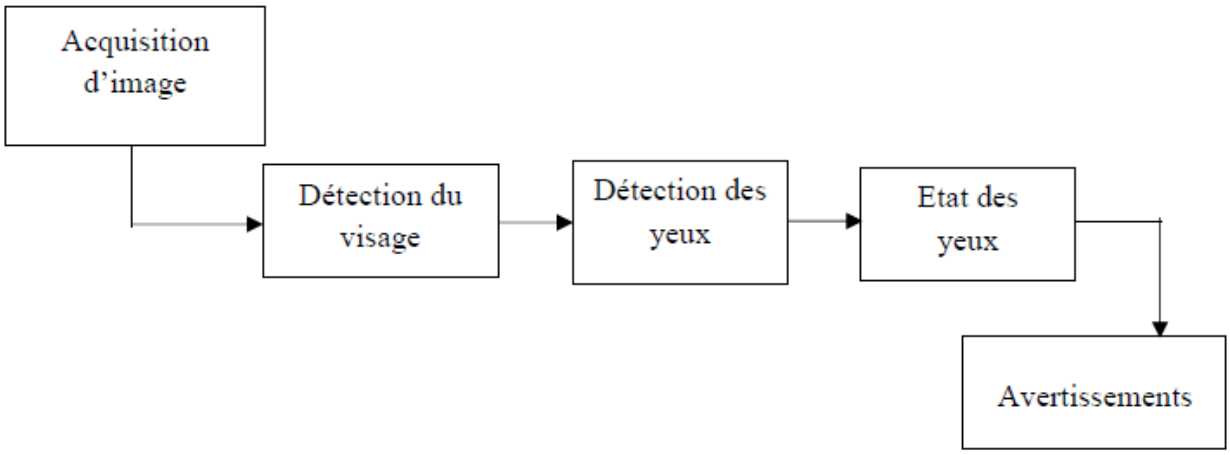


Figure 4.1 : Système de détection de somnolence.

4.2.1 Présentation du système par l'architecture de CNN :

Nous avons choisi l'architecture CNN pour le développement du système de détection de somnolence, car elle présente plusieurs avantages par rapport aux autres méthodes. Tout d'abord, l'architecture CNN est capable d'extraire des caractéristiques pertinentes des images du visage sans avoir besoin d'un prétraitement complexe. Ensuite, l'architecture CNN est robuste aux variations de luminosité, de pose et d'expression faciale, ce qui est essentiel pour un système de détection de somnolence en temps réel. Enfin, l'architecture CNN peut être facilement entraînée et optimisée avec des techniques de deep learning, ce qui permet d'améliorer les performances du système.

4.2.2 Collecte et exploitation de la base des données

Pour analyser la détection de la somnolence, nous avons exploité une base de données de 84 898 images sous le nom de « **MRL-Eye-Dataset**». Nous l'avons séparée en deux parties : une partie pour l'entraînement et une partie pour le test. La partie entraînement comprend 81 674 images, réparties en quatre catégories : 40837 images d'yeux fermés, 40837 images d'yeux ouverts. La partie test comprend 3 224 images, également réparties en quatre catégories : 1612 images d'yeux fermés, 1612 images d'yeux ouverts. Nous avons exploité les fichiers train et tests des yeux ouverts et fermés pour entraîner et évaluer notre modèle de détection de la somnolence (figure 4.3).

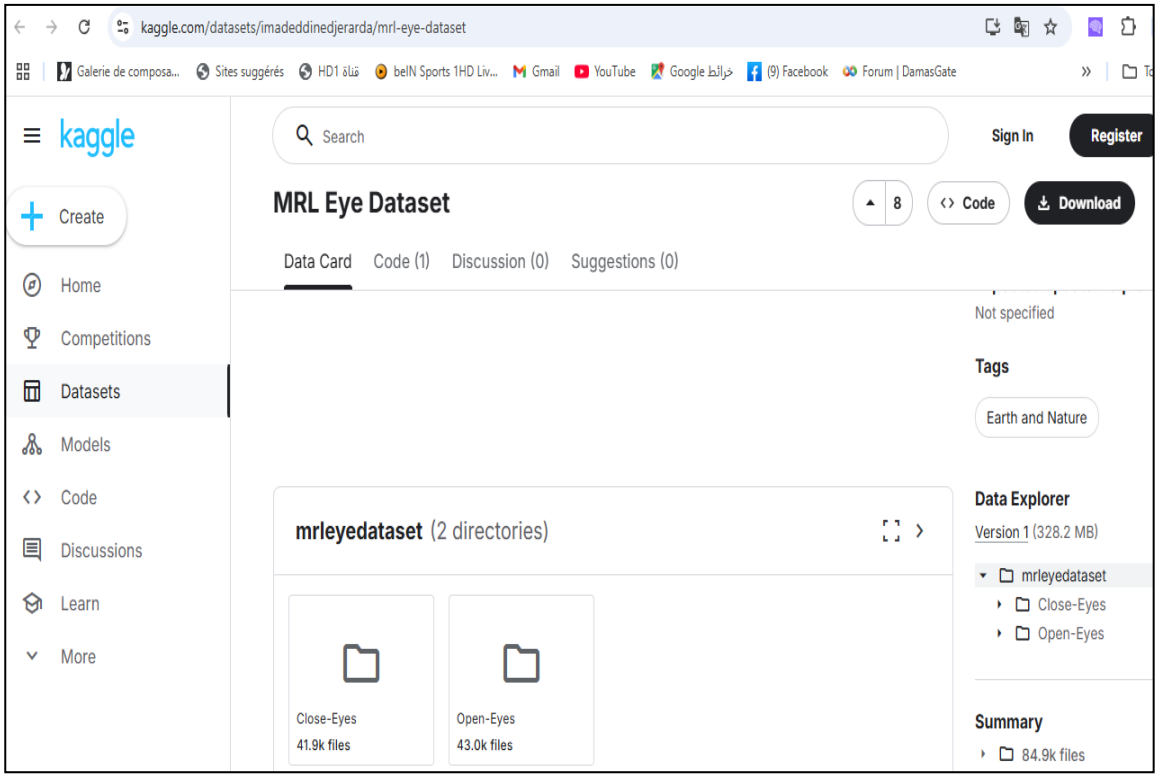


Figure 4.2 : Base de données utilisée

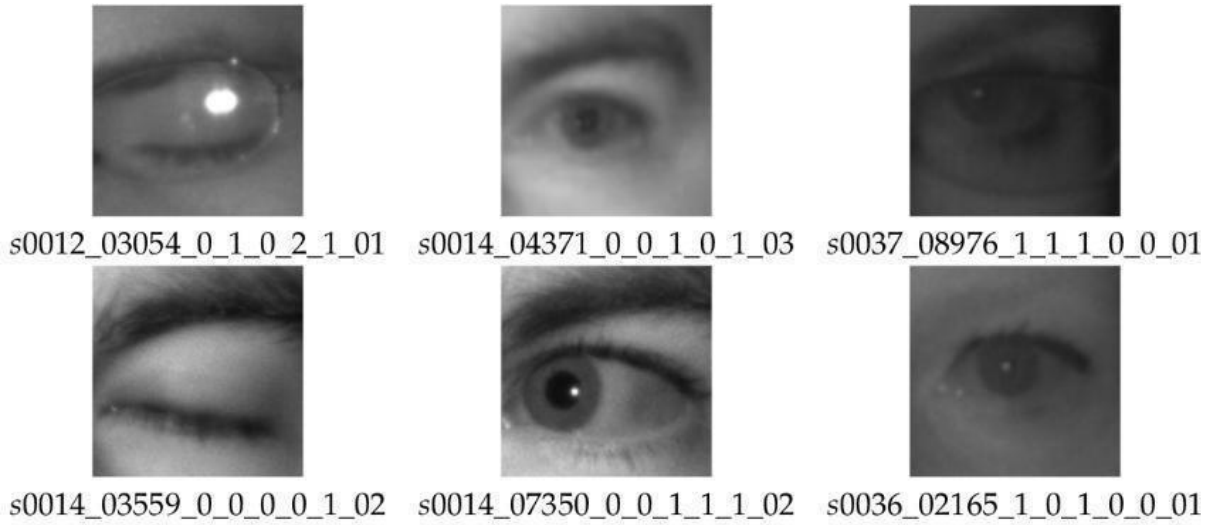


Figure 4.3: Exemples d'annotations d'images de l'ensemble de base de données

4.2.3 Conception de modèle CNN :

L'organigramme ci-dessus (figure 4.4) présente architecture de notre modèle CNN pour la détection de somnolence.

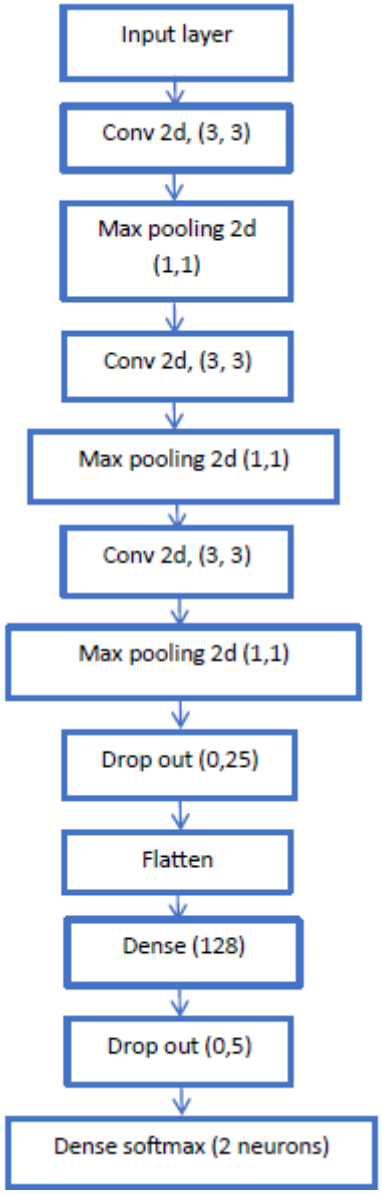


Figure 4.4 : Architecture de notre modèle.

- Il prend en entrée des images en niveaux de gris de taille 64x64 pixels et renvoie une probabilité pour chacune des deux classes possibles.

- Il se compose de trois couches de convolution avec des filtres de taille 3x3 et des fonctions d'activation Relu, suivies chaque une d'une couche de max-pooling avec une taille de pool de 2x2.

- Il utilise ensuite une couche de flatten pour aplatir les données en un vecteur.

- Il ajoute ensuite une couche dense avec 128 neurones et une fonction d'activation ReLU, suivie d'une autre couche de dropout avec un taux de 0.5 pour réduire le surapprentissage.

- Enfin, il termine par une couche dense avec deux neurones et une fonction d'activation softmax pour produire les probabilités des classes.

- Il compile le modèle avec l'optimiseur Adam, la fonction de perte entropie croisée catégorielle et la métrique accuracy.

- Il entraîne le modèle avec les données générées par la fonction generator, en utilisant une taille de batch de 32, un nombre d'époques de 20 époques et une validation croisée.

- Il utilise également un objet History pour enregistrer l'historique de l'apprentissage du modèle.

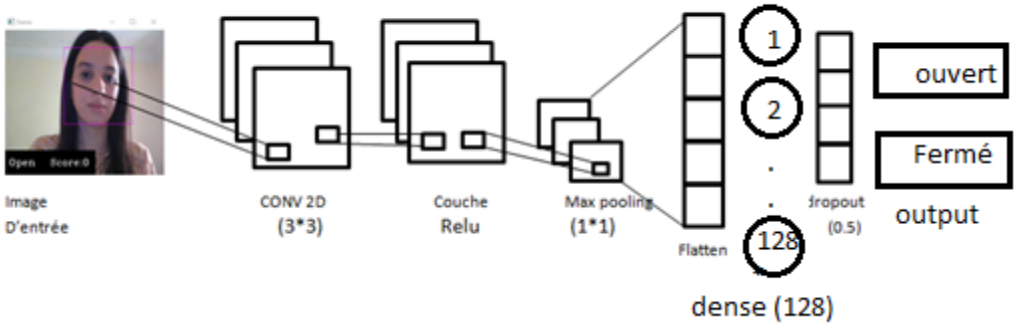


Figure 4.5 : Les couches de l'architecture de modèle CNN.

4.2.4 La détection des points caractéristiques dans un visage :

L'étape de caractérisation consiste à dégager les caractéristiques pertinentes de l'expression faciale étudiée et d'éliminer les informations redondantes. Puisque nous intéressons à la fatigue, nous cherchons à extraire des points caractéristiques spécifiques dans le visage : les yeux droit et gauche. La détection de ces points est implémentée dans la librairie Dlib utilisé sous le langage Python. Elle permet de produit 68 point 2D de coordonnées (x, y) qui cartographient des structures faciales spécifiques. Ces points sont stockés dans un tableau indexé. Voici donc les indices de chaque point parmi les 68 points (Figure 4.6).

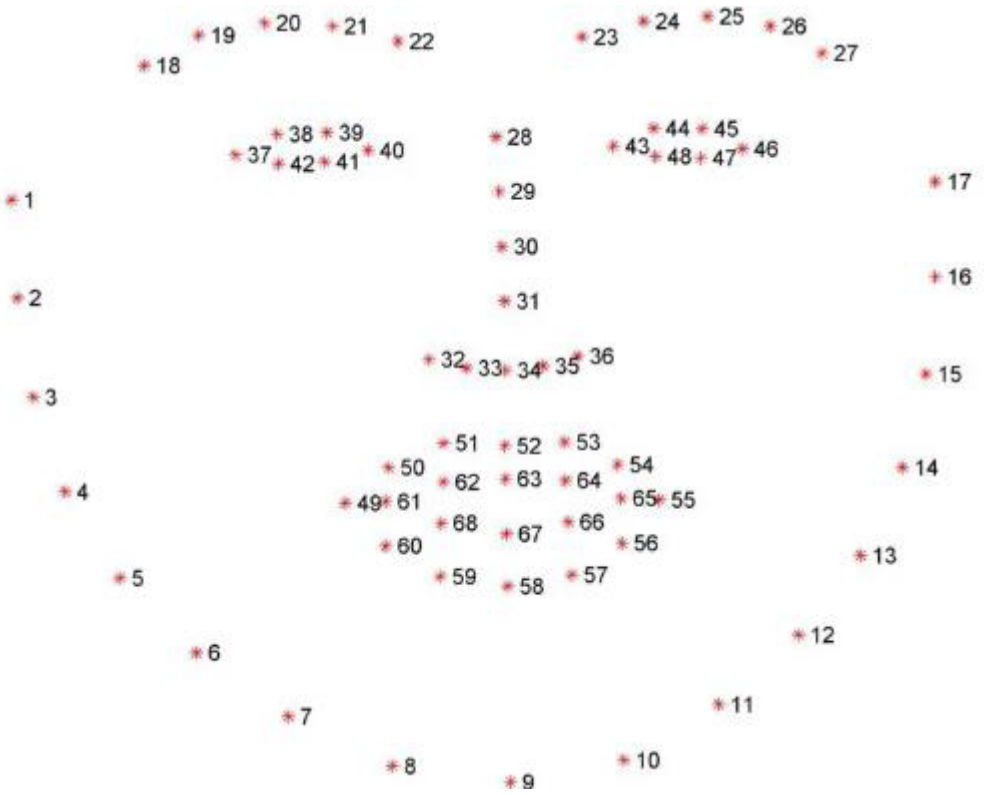


Figure 4.6: Shape predictor land marks 68.

4.2.5 Détection des yeux

L'ordre des repères faciaux est toujours le même quoique ce soit la taille de l'image ou du visage, le contour du visage est détecté par les points ordonnés de 1 à 17. Les points de 37 à 42 représentent la détection de l'oeil gauche et 43 à 48 représentent l'oeil droit.

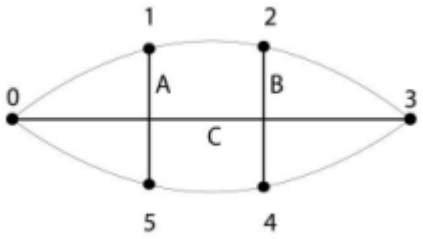


Figure 4.7: Région d'un oeil représentée par les points caractéristiques.

Après la détection des yeux, des rectangles sont dessinés autour de chaque oeil détecté, permettant de visualiser leur emplacement exact dans l'image (Figure 4.8).

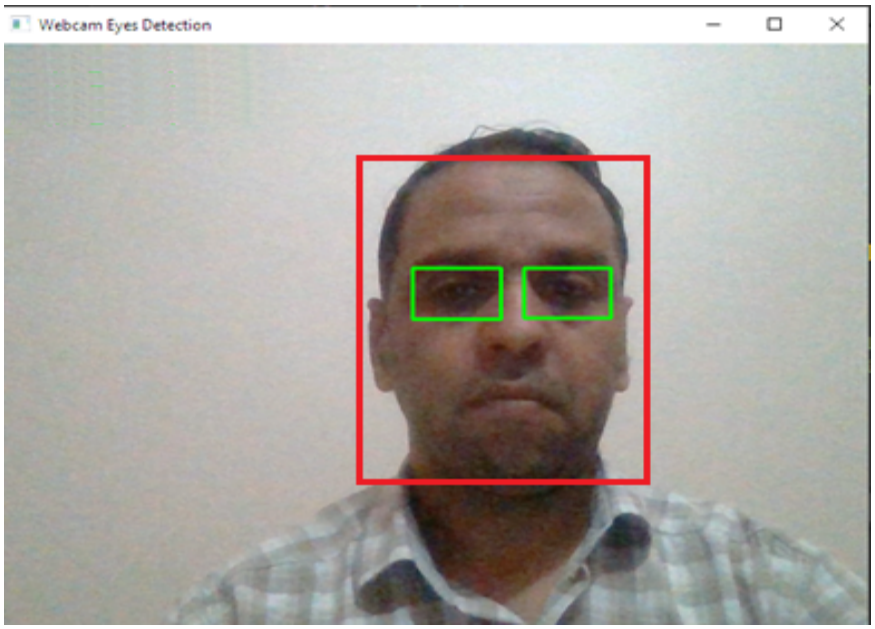


Figure 4.8: Image illustrative de la détection des yeux.

4.3. Implémentation :

4.3.1 Environnement de développement du matériel

Nous avons généré nos résultats sur une machine qui possèdent les caractéristiques suivantes :

— Type de processeur : Intel(R) Celeron(R) N4020 CPU @ 1.10GHz

— Capacité de la RAM : 4 Go.

— Disque dur : 256 GO (SSD)

— Système d'exploitation : Windows10 Professionnel 64 bits

4.3.2 Mise en place et la configuration de l'environnement

Dans notre projet, nous avons choisi d'utiliser le langage de programmation Python pour ses fonctionnalités avancées en matière de deep learning, sa flexibilité, sa compatibilité avec une grande variété de plateformes, sa grande communauté de développeurs et ses outils de visualisation de données avancés.

1. Langage python (3.8.5):

Python est un langage de programmation de haut niveau utilisé pour la programmation générale. Créé par Guidovan Rossum et Sortien 1991, Python a une philosophie de conception qui met l'accent sur la lisibilité du code, notamment en utilisant des espaces importants. Il fournit des constructions qui permettent une programmation claire à petite et à grande échelle. Python dispose d'un système de type dynamique et d'une gestion automatique de la mémoire. Il prend en charge de multiples paradigmes de programmation, y compris orienté objet, impératifs, fonctionnels et procéduraux, et dispose d'une bibliothèque standard vaste et complète. Les interpréteurs de Python sont disponibles pour de nombreux systèmes d'exploitation.

Python est un langage très utilisé pour implémenter des CNN, car il offre de nombreuses bibliothèques et modules spécialisés dans le domaine du deep learning, tels que TensorFlow, PyTorch, Keras, etc. Ces outils permettent de créer facilement des modèles de CNN performants et personnalisables, en utilisant des réseaux pré-entraînés ou en créant ses propres architectures.

Figure 4.9 : Le Logo de python.

2. Visual Studio Code :

Nous avons choisi d'utiliser VS Code comme environnement de développement pour notre projet de CNN pour sa popularité, sa compatibilité avec de nombreux langages de programmation, ses fonctionnalités avancées pour le développement de logiciels, sa grande communauté de développeurs et son intégration avec des outils de développement externes.

"Visual Studio Code" est un environnement de développement intégré (IDE) développé gratuitement et de manière ouverte par Microsoft. Il est utilisé principalement pour le développement de logiciels, et prend en charge une grande variété de langages de programmation

tels que Python, C++, Java, JavaScript, et bien d'autres. Ce programme vise à fournir une interface utilisateur simple et intuitive, tout en offrant des fonctionnalités avancées telles que la coloration syntaxique, l'intégration Git, la détection d'erreurs et le débogage. VS Code est très populaire auprès des développeurs grâce à son extensibilité, ce qui permet aux utilisateurs d'ajouter des fonctionnalités supplémentaires en installant des extensions, en faisant de Visual studio code un environnement de développement très personnalisable selon les besoins de chaque utilisateur.

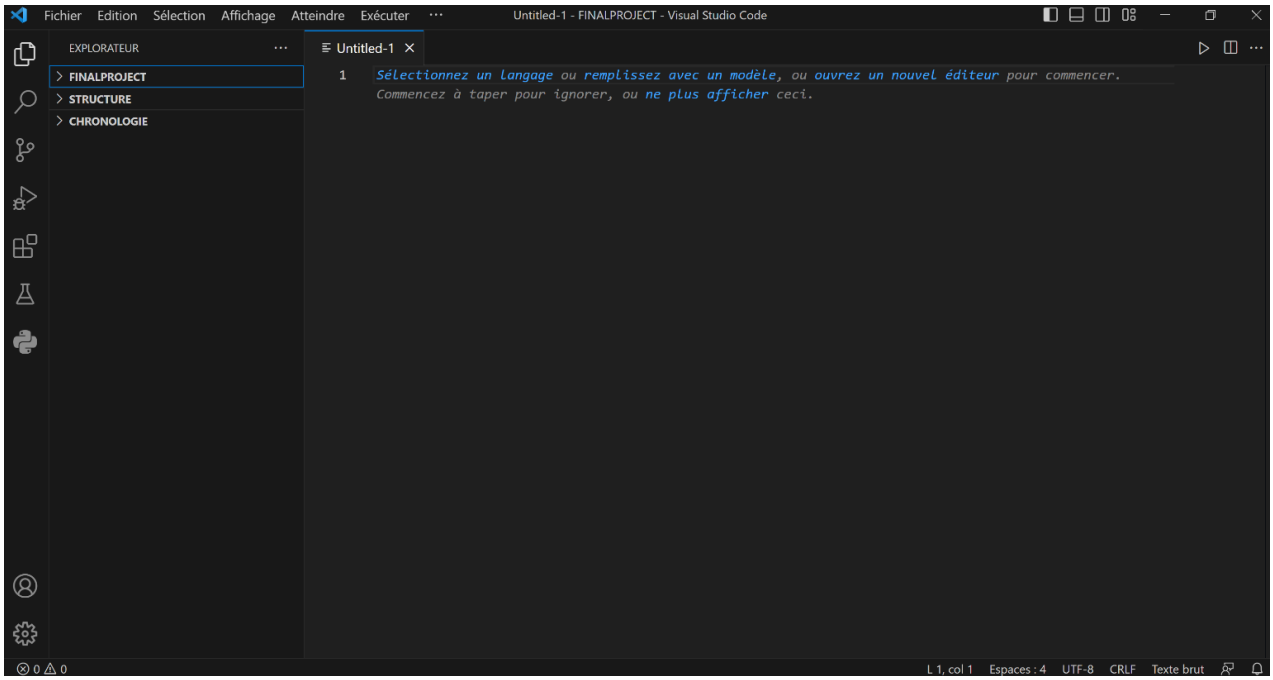


Figure 4.10 : Espace de travail de Visual studio.

4.3.3 Les bibliothèques utilisées :

Dans ce programme en a utilisé plusieurs bibliothèques python pour la détection des yeux et de visage D'une personne dans une image capturée en temps réel depuis la caméra. Voici une brève explication de chaque bibliothèque utilisée dans le programme.

1. Dlib :

Dlib est une bibliothèque open source multiplateforme (c'est-à-dire peut être exécutée directement sur n'importe quelle plate-forme sans préparation spéciale) écrite en langage C ++, qui

fournit une documentation complète et précise ainsi que de puissants outils de débogage. Dlib vise à fournir un environnement tout aussi riche pour le développement de logiciels d'apprentissage automatique, y compris la classification, la régression, le clustering, Réseaux bayésiens. Et prenant en charge des fonctionnalités, notamment le threading, la mise en réseau, les algorithmes numériques, le traitement d'image et plein d'autres. Les fonctionnalités qui nous intéressent sont celles de traitement d'image, et spécifiquement les outils de détection d'objets dans les images, tel que la détection de face frontale, qui est basé sur l'approche de détection des repères faciaux.

2. TensorFlow :

TensorFlow est une bibliothèque de machine learning open source, développée par Google, qui permet de créer et d'exécuter des applications de machine learning et de deep learning. TensorFlow offre une plate-forme de bout en bout pour le machine learning, avec des outils pour préparer les données, créer des modèles, les déployer et les maintenir. TensorFlow propose également des modèles pré-entraînés ou personnalisables, ainsi que des bibliothèques et des extensions spécialisées pour enrichir TensorFlow de nouvelles fonctionnalités. TensorFlow est utilisé pour résoudre des problèmes concrets avec le machine learning, tels que la détection et reconnaissance d'images, le traitement du langage naturel, la génération de texte, la recommandation ou le renforcement.

3. Keras :

Keras est une bibliothèque de haut niveau pour le deep learning, qui permet de créer et d'entraîner des modèles de réseaux neuronaux de manière simple et intuitive. Elle est basée sur TensorFlow, le Framework de calcul numérique développé par Google. Keras offre plusieurs avantages par rapport aux autres bibliothèques de deep learning :

- Elle est conviviale : elle propose une interface claire et cohérente, optimisée pour les cas d'utilisation courants. Elle fournit également des messages d'erreur clairs et explicites.

- Elle est modulaire et composable : les modèles Keras sont construits en assemblant des blocs configurables, avec peu de contraintes.

Il est possible de créer des couches, des métriques et des fonctions de perte personnalisées. Keras peut être utilisée pour différents types de problèmes de deep learning, tels que la classification d'images, la génération de texte, le traitement du langage naturel ou la recommandation. Elle dispose d'une documentation complète et de nombreux tutoriels pour apprendre à l'utiliser efficacement,

4. NumPy

NumPy est un module Python permettant de manipuler les tableaux. Il contient des outils pour manipuler un objet tableau multidimensionnel de haute performance. C'est le module Python le plus important pour le calcul scientifique.

5. Winsound

Le module winsound en Python permet d'interagir avec les fonctionnalités de lecture de son de Windows. Il offre des fonctions pour émettre des bips, lire des fichiers sonores et interagir avec les sons système. Il est principalement utilisé pour créer des alertes sonores simples ou pour intégrer des sons dans des applications Windows.

4.3.4 Fonctionnement de programme principal :

Une fois que le modèle a été exécuté, nous avons sauvegardé le modèle entraîné sous forme d'un fichier .h5. Ensuite on l'a téléchargé dans le code principal, où nous avons effectué des prédictions sur la somnolence déclencher une alarme, (Figure 4.11).

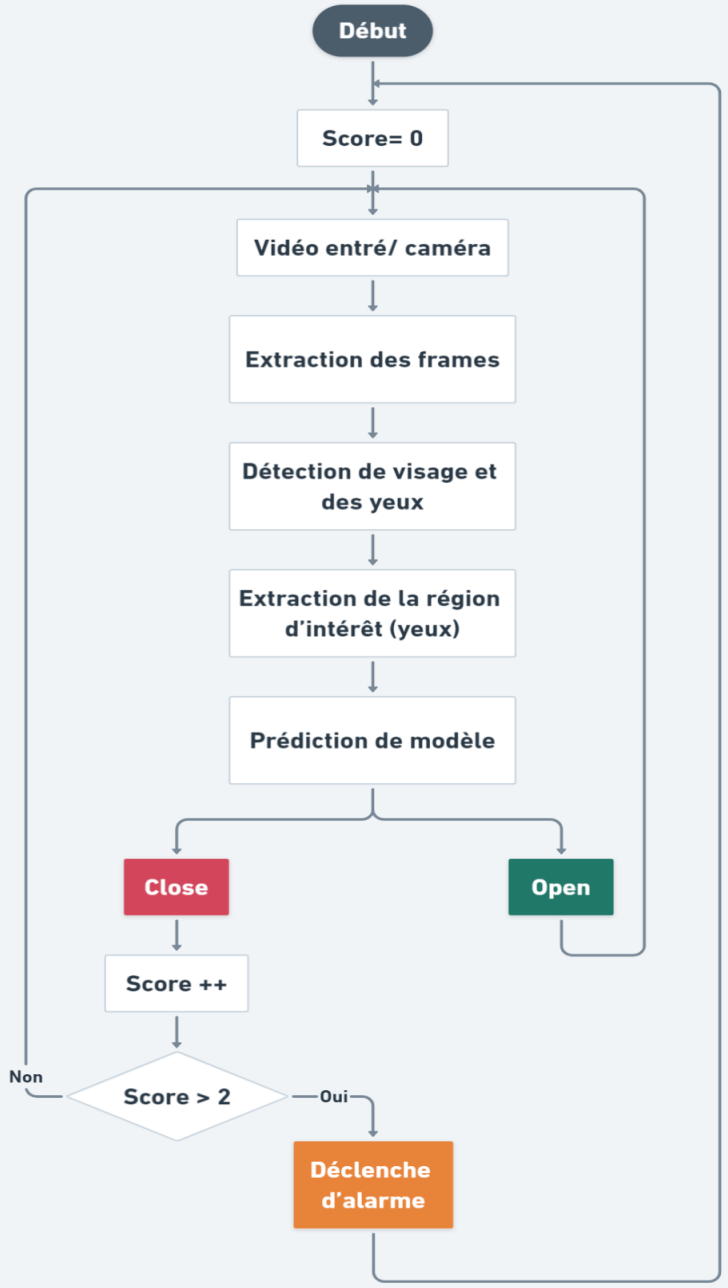


Figure 4.11: Organigramme d'algorithme réalisé.

4.4 Résultats expérimentaux :

4.4.1 Entrainement du modèle

Au cours de l'entraînement, la fonction de perte et la précision sont calculées et affichées pour chaque lot (batch) de données. À la fin de chaque époque, la fonction de perte et la précision sont calculées pour l'ensemble des données de validation (Figure 4.12). L'objectif de l'entraînement est

de minimiser la fonction de perte et d'augmenter la précision pour obtenir un modèle performant et précis.

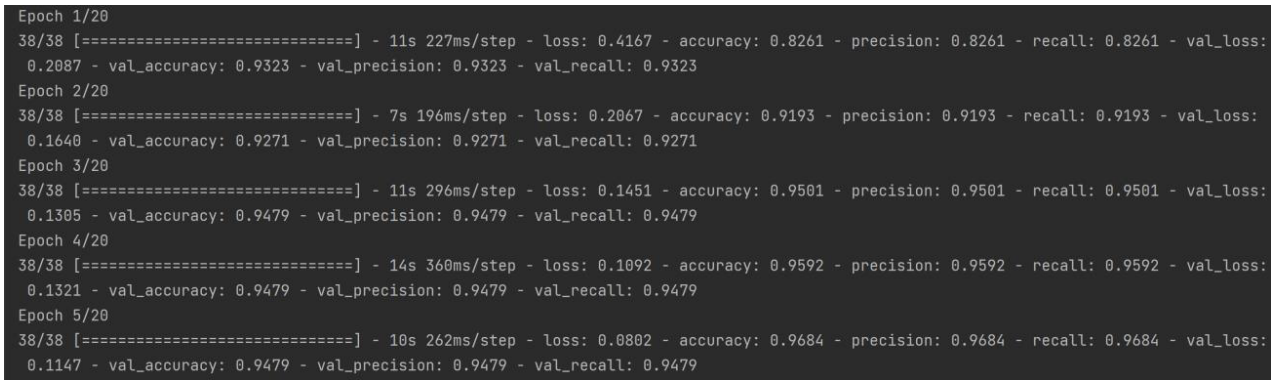


Figure 4.12: Le début de l'entrainement**.**

Les résultats de l'exécution de notre modèle montrent que nous l'avons entraîné sur un total de 20 époques. À chaque époque, le modèle a été alimenté avec un lot de données d'entraînement et a ajusté ses paramètres internes pour minimiser la fonction de perte, qui mesure la disparité entre les valeurs prédites et réelles.

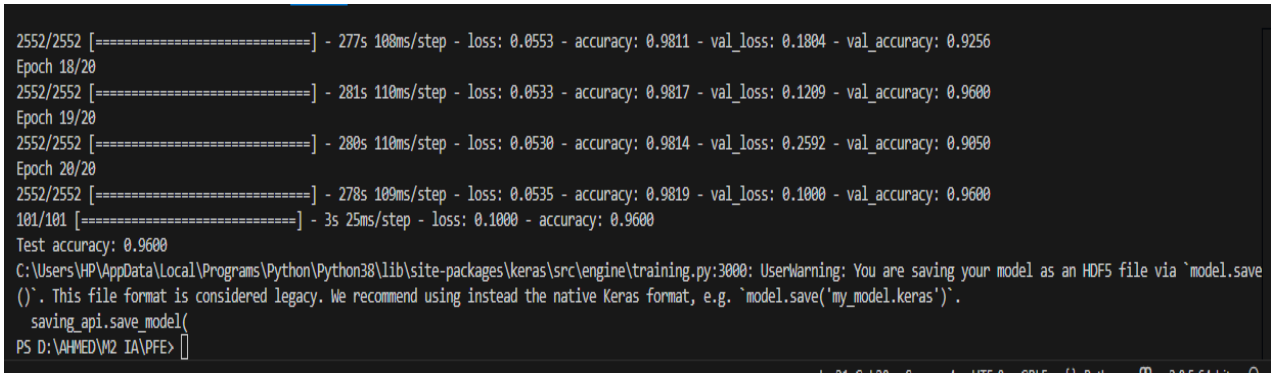


Figure 4.13 Fin de l'entrainement

Nous avons représenté les résultats d'exécution de notre modèle dans des graphes affichant l'évolution de l'exactitude (accuracy) et la fonction de perte (loss). (Figure 4.14, 4.15)

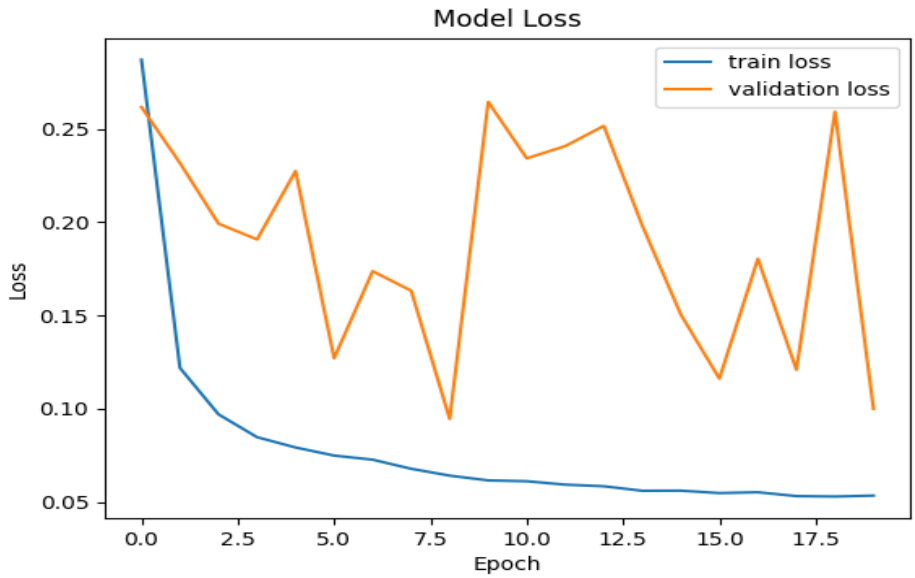


Figure 4.14 : Présentation de perte de l'entrainement**.**

En ce qui concerne la métrique de perte, le modèle a commencé avec une perte d'entraînement de 0,7219 et une perte de validation de 0,2187 lors de la première époque. Tout au long de l'entraînement, la perte a diminué progressivement, indiquant la capacité du modèle à optimiser ses paramètres et à effectuer de meilleures prédictions. À la fin de l'entraînement, le modèle a atteint une perte d'entraînement de 0.0535 et une perte de validation de 0,1000.

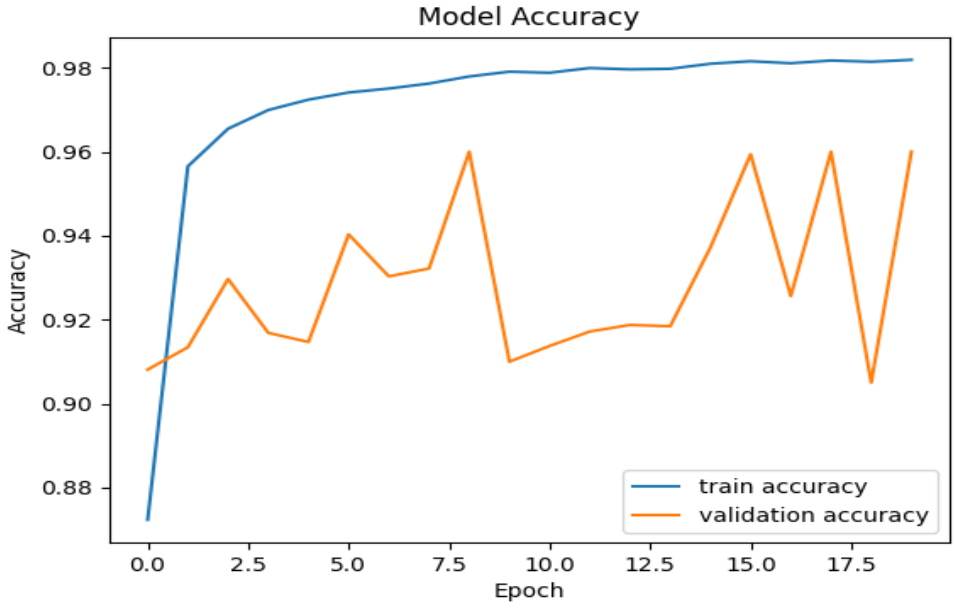


Figure 4.15 : Présentation de l'exactitude de l'entrainement**.**

L'exactitude, qui mesure la proportion de prédictions correctes, augmente également progressivement, atteignant une valeur de 98,19% à la dernière époque. Et une exactitude légèrement inférieure, mais toujours impressionnante de 0,9600 sur les données de validation. Ce qui signifie que le modèle a correctement prédit l'état de somnolence du conducteur pour 96% des instances dans l'ensemble de validation.

Ces résultats démontrent que le modèle a efficacement appris à partir des données d'entraînement et s'est bien généralisé aux données de validation. Avec une précision de 96% sur l'ensemble de validation, le modèle a montré sa capacité à effectuer des prédictions précises sur des données inconnues. Les valeurs décroissantes de perte indiquent que le modèle a réussi à converger vers des valeurs de paramètres optimales lors de l'entraînement.

4.4.2 Vérification de la prédiction du modèle :

Pour mieux tester l'efficacité du modèle développé dans ce projet, nous avons créé un fichier python « prediction.py » (Figure 4.16) pour faire des prédictions sur des images. Les résultats obtenus sont illustrés dans les figures ci-dessous :

```python
# Fonction pour prédire l'état des yeux (Open ou Closed)
def predict_eye_state(img_path, model, img_size):

    # Charger l'image en niveaux de gris
    img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)

    if img is None:
        print("Erreur : Impossible de charger l'image. Vérifiez le chemin.")
        return
   # Redimensionner l'image à la taille attendue par le modèle
    img = cv2.resize(img, img_size)

    # Normaliser l'image (valeurs de pixels entre 0 et 1)
    img = img / 255.0

    # Ajouter des dimensions pour correspondre à l'entrée du modèle
    img = img[np.newaxis, ..., np.newaxis]

    # Faire la prédiction
    prediction = model.predict(img)
    predicted_class = np.argmax(prediction)  # Récupérer la classe prédite (0 ou 1)

    # Afficher le résultat
    if predicted_class == 0:
        print("Prédiction : Open ")
    else:
        print("Prédiction : Closed ")
```

Figure 4.16 : Code de prédiction de notre modèle.

Et maintenant, en va tester si l'oeil est fermé ou ouvert, le variable predicted_class prendre les valeurs 0 ou 1, si l'oeil est détecté comme fermé, alors la valeur de predicted_class est égale à un "1", et si l'oeil est détecté comme ouvert, alors la valeur de predicted_class est égale à zéro "0".

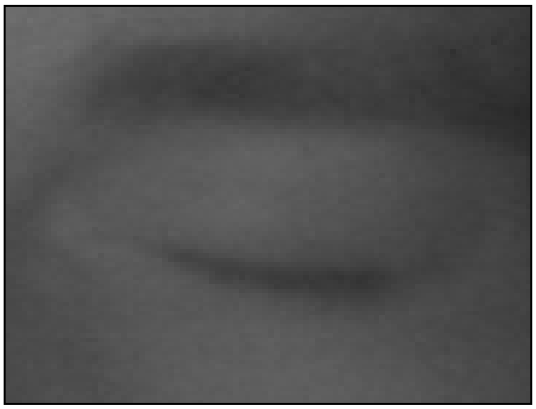


Figure 4.17 : l'image testé par notre model CNN.

Puisque l'oeil est fermé, nous avons obtenu la valeur de classe égale à 0, alors notre programme de prédiction est retenu la valeur Closed.

```
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 495ms/step
Pr♦diction : Closed
Merci d'avoir utilis♦ le programme. Au revoir !
```

Figure 4.18 : résultat de la prédiction de l'image **(**s0001_00009_0_0_0_0_0_01)

4.4.3 Résultats de simulation :

L'objectif de cette section est de présenter les résultats obtenus à partir des tests de notre méthode proposée. Les résultats obtenus sont les suivants :

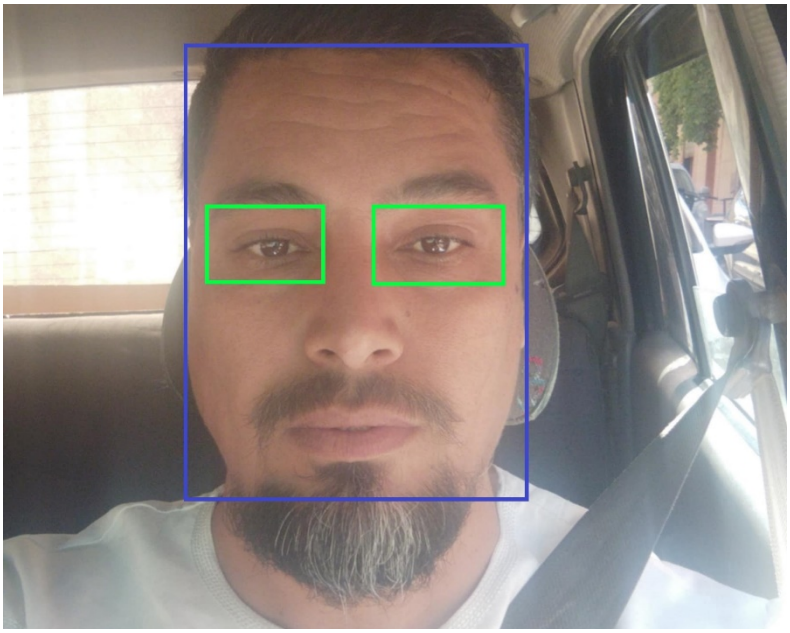


Figure 4.19 : Résultat des yeux ouvert.

Dans cette situation, les yeux ont été détectés dans un état ouvert avec un score de 0. Lorsque les yeux se ferment, Le score incrémente, lorsqu'il atteint 2 ou plus, ce qui correspond à une durée de deux secondes. Dans ce cas, l'alarme est déclenchée (Figure 4.19).

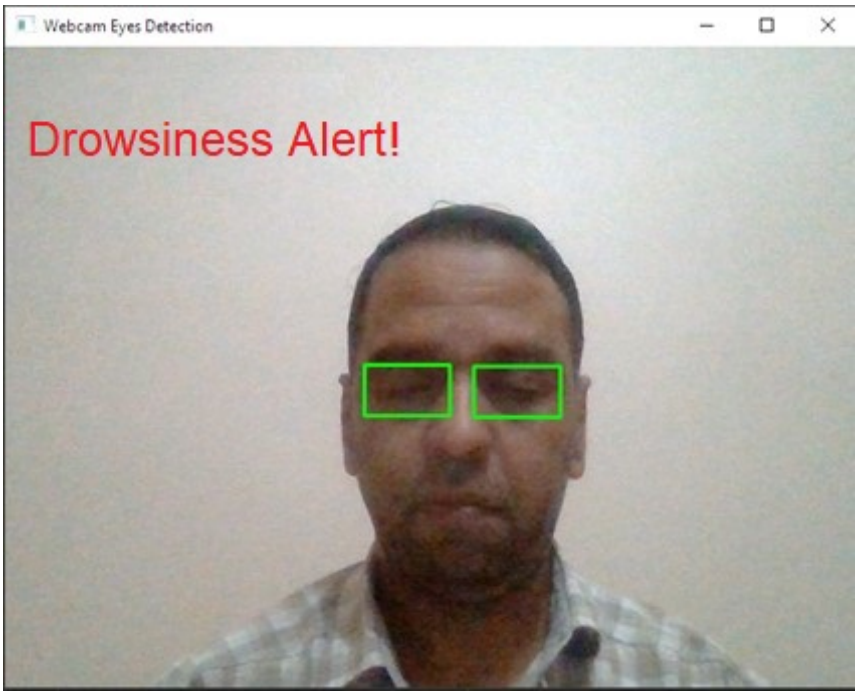


Figure 4.20 : Résultat des yeux fermés.

2.6 Conclusion

Dans ce chapitre, nous avons abordé différents aspects de la détection de la somnolence, tels que la collecte et le prétraitement des données, la conception et l'entraînement du modèle CNN, ainsi que l'évaluation des performances de ce dernier. Nous avons commencé par d'écrire la problématique, les défis de notre système de détection de somnolence ainsi que les objectifs et les contraintes de ce système. Après cela, nous avons présenté les étapes pour collecter des données pertinentes, pour les rendre adaptées à l'entraînement de notre modèle CNN, et concevoir une architecture de réseau adaptée à la détection de la somnolence.

Enfin, nous avons évalué les performances du modèle de détection de la somnolence et discuté les résultats obtenus, et nous sommes arrivées à la conclusion suivante : le modèle CNN développé s'est avéré très satisfaisant lors de la simulation vue qu'il a été entrainé de manière efficace. Nous avons conclu dans ce chapitre que les résultats de notre simulation est impressionnante et que notre modèle a été entrainé de manière efficace.

Conclusion générale

L’intelligence artificielle (IA) a ouvert de nouvelles perspectives dans divers domaines, y compris la détection de la somnolence. Grâce à l’apprentissage profond, l’IA joue un rôle central dans la révolution des pratiques liées à la détection de la somnolence. En exploitant la puissance de l’IA, cette étude vise à améliorer l’exactitude des méthodes de détection de la somnolence et obtenir de meilleurs résultats dans la prévention des incidents liés à la somnolence.

Ce projet nous a permis d'explorer plusieurs méthodes de détection de somnolence au volant du conducteur pendant la conduite. Nous avons choisi une méthode pour développer un système de détection de somnolence basé sur l'apprentissage profond, en particulier l'architecture CNN.

Cette architecture combine les avantages de l’algorithmes de la détection faciale Facial landmark (68) pour la détection du visage, et des yeux, et ceux de CNN pour la prédiction de la fatigue basée sur la fermeture des yeux, notre système est capable de localiser et de suivre les traits du visage, y compris les yeux, ce qui permet de détecter les signes de somnolence tels que les clignements des yeux fréquents, le système qui sont des indicateurs importants de la fatigue.

Les résultats obtenus ont clairement démontré l’efficacité de notre modèle pour la détection la somnolence. Nous avons obtenu un score de précision 98,19%.

En conclusion, notre résultat met en évidence l’impact remarquable de l’IA et de l’apprentissage profond dans la détection de la somnolence. Grâce à la conception et à la mise en œuvre d’un système basé sur Les réseaux de neurones convolutionnels CNN.

Cependant, cette recherche pourrait être améliorée sur certains points que nous considérons comme des perspectives pour nos travaux :

- L’utilisation d’autres méthodes de reconnaissance, et la combinaison avec d’autres pour la conception d’un système hybride.
- Fiabiliser le système en diminuant la sensibilité aux contraintes d’éclairage, de pose et d’occultation par l’utilisation de nouvelles techniques de normalisation.
- Accroitre le taux de précision en entrainant le modèle sur des bases des données plus volumineuses et en augmentant le nombre d’époques.
- Etendre notre système pour l’acquisition des images 3D et inclure plus de classes d’expressions.

Bibliographie

[1] Junjie Peng, Elizabeth C Jury, Pierre Dönnes, and Coziana Ciurtin. Machine learning techniques for personalised medicine approaches in immune-mediated chronic inflammatory diseases : applications and challenges. Frontiers in pharmacology, 12 : 720694, 2021.

[2] Allan David Gordon. Classification. CRC Press, 1999.

[3] Priyal Agarwal. Regression analysis, 2021. URL https://www.analyticsvidhya.com/blog/2021/05/regression-analysis-real-time-portugal-2019-election-results/.

[4] Sharma Pulkit. The most comprehensive guide k-means clustering. Analytics Vidhya,2019.

[5] Debasish Kalita. An overview and applications of artificial neural networks, 2022. URL https://www.analyticsvidhya.com/blog/2022/03/an-overview-and-applications-of-artificial-neural-networks-ann/.

[6] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. Towards Data Sci, 6(12) :310–316, 2017. in python. URL https://www.educative.io/answers/how-to-implement-the-softmax-function-in-python.

[7] Tim Jones Samaya Madhavan. Deep learning architectures,2017. URL https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures/.

[8] Mehdi Mohammadi, Ala Al-Fuqaha, Sameh Sorour, and Mohsen Guizani. Deep learning for iot big data and streaming analytics : A survey. IEEE Communications Surveys & Tutorials, 20(4) :2923–2960, 2018

[9] projectpro. Introduction to convolutional neural networks architecture,2023. URL https://www.projectpro.io/article/introduction-to-convolutional-neural-networks-algorithm-architecture/560#mcetoc_1fs0joo1hg.

[10] Kimia Nadjahi Pascal Monasse. Découvrez les différentes couches d'un cnn, 2022. URL https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5083336-decouvrez-les-differentes-couches-dun-cnn.

[11] Nebojša Bačanin Džakula et al. Convolutional neural network layers and architectures.In Sinteza 2019-International Scientific Conference on Information Technology and Data Related Research, pages 445–451. Singidunum University, 2019.

[12] Ajitesh Kumar. Different types of cnn architectures explained : Examples, 2023. URL https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015.

[15] Mehrdad Farahani, Marzieh Farahani, Mohammad Manthouri, and Okyay Kaynak.Short-term traffic flow prediction using variational lstm networks. arXiv preprint arXiv :2002.07922, 2020.

[16] Shivanshu Aggarwal. The ultimate guide to building your own lstm models, 2023.

URL https://www.projectpro.io/article/lstm-model/832.

[17] Audebert, N. (2018). *Classification de données massives de télédétection* (Doctoral dissertation, Université Bretagne Sud).

[18] Bérard, F. (1999). *Vision par ordinateur pour l'interaction homme-machine fortement couplée* (Doctoral dissertation, Université Joseph-Fourier-Grenoble I).

[19] Robert, J. (2023). Image Processing : principes fondamentaux et usages pratiques. Disponible : https://datascientest.com/image-processing-tout-savoir

[20] Belsare, A. D., & Mushrif, M. M. (2012). Histopathological image analysis using image processing techniques: An overview. *Signal & Image Processing*, *3*(4), 23.

[22] Aichour, Y. (2007). *Détection automatique des expressions faciales à partir de séquences d'images pour l'évaluation de l'état des facultés d'une personne* (Doctoral dissertation, Université du Québec à Trois-Rivières).

[23] Abdat, F. (2010). Reconnaissance automatique des émotions par données multimodales: expressions faciales et signaux physiologiques. *Université de Metz, France*.

[24] Ekman, P., & Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of personality and social psychology*, *17*(2), 124.

[25] Corneanu, C. A., Simón, M. O., Cohn, J. F., & Guerrero, S. E. (2016). Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications.

[26] Boughida, Adel & Kouahla, Med Nadjib & Lafifi, Yacine. (2018). *État de l'art sur les systèmes de reconnaissance des expressions faciales.*

[27] OUAMANE, H. (2012). *Identification de reconnaissance faciale avec des expressions* (Master's thesis).

[28] Siouane Mouhssin & Saada Fawzi. (2020). *Reconnaissance des expressions faciales en temps réel* (Master's thesis, Université Blida).

[29]Laskri, M. T., & Chefrour, D. (2002). Who_Is: système d'identification des visages humains. *Revue Africaine de Recherche en Informatique et Mathématiques Appliquées*, *1*.

[30] OUAMANE, H. (2012). *Identification de reconnaissance faciale avec des expressions* (Master's thesis).

[31] Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees.

[32] Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features.

[33] "ROAD SAFETY FACTS," ASIRT, 2018. [Online]. Available:https://www.asirt.org/safe-travel/road-safety-facts/. [Accessed 1 5 2019].

[34] 2014 " إحصاءات حوادث المرور المسجلة خلال سنة ", CNPSR, alger, 2014.

[35] M. Bougueroua and L. Carnis, "Insécurité routière et développement économique : analyse du cas algérien," *RTS - Recherche Transports Sécurité,* vol. 2018, p. 13p, Apr 2018.

[36] National Center for Statistics and Analysis, "Drowsy Driving 2015," Washington, 2017.

[37] J. Krajewski, D. Sommer, U. Trutschel, D. Edwards and M. Golz, "Steering Wheel Behavior Based Estimation of Fatigue," *Driving assessment conference,* 2009.

[38] P. Winston, Director, *Learning: Support Vector Machines.* [Film]. USA: Massachusetts Institute of Technology, 2010.

[39] z. Zhang, "Introduction to machin learning: K-nearest neighbors," *Annals of translational Medcine,* vol. 4, p. 218, 2016.

[40] M. Inger, T. Åkerstedt, B. Peters, A. Anund and G. Kecklund, "Subjective sleepiness, simulated driving performance and blink duration: Examining individual differences," *Journal of sleep research,* vol. 15, pp. {47-53, 2006.

[41] M. Gillberg, G. Kecklund and T. Åkerstedt, " Relations between performance and subjective ratings of sleepiness during a night awake," *Sleep: Journal of Sleep Research & Sleep Medicine,* vol. 17, pp. 236-241, 1994.

[42] S. Hu and G. Zheng, "Driver Drowsiness Detection with Eyelid related Parameters by Support Vector Machine," *Expert Systems with Applications,* vol. 36, pp. 7651-7658, 2009.

[43] P. Antoine, C. Sylvie and A. Caplier, "EOG-based drowsiness detection:Comparison between a fuzzy system and two supervised learning classifiers," *The International Federation of Automatic Control,* 2011.

[44] M. Patel, S. K. Lal, D. Kavanagh and P. Rossiter, "Applying neural network analysis on heart rate variability data to assess driver fatigue," *Expert Systems with Applications,* vol. 38, p. 7235–7242, 2011.

[45] S. Arun, S. Kenneth and M. Murugappan, "Detecting Driver Drowsiness Based on Sensors: A Review," *{Sensors (Basel, Switzerland,* vol. 12, pp. 16937-16953, 2012.

[46] J. May and C. Baldwin, "Driver fatigue: The importance of identifying causal factors of fatigue when considering detection and countermeasure technologies," *Transportation Research Part F: Traffic Psychology and Behaviour,* vol. 12, pp. 218-224, 2009.

[47] I. García, S. Bronte, L. M. Bergasa, J. Almazán and J. Yebes, "Vision-based drowsiness detector for real driving conditions," *IEEE Intelligent Vehicles Symposium,* pp. 618-623, 2012.

[48] W. Zhang, B. Cheng and Y. Lin, "Driver drowsiness recognition based on computer vision technology," *Tsinghua Science and Technology,* vol. 17, pp. 354-362, 2012.

[49] A. Rosebrock, "Drowsiness detection with OpenCV," 08 May 2018. [Online]. Available: https://www.pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/. [Accessed 29 May 2019].

[50] G. Reza, G. Marnim and A. Vassilis, "A Realistic Dataset and Baseline Temporal Model for Early Drowsiness," *CoRR,* vol. abs/1904.07312, 2019.

[51] M. Ochocki and D. Sawicki, "Yawning Recognition based on Dynamic Analysis and Simple Measure," in *Proceedings of the International Conference on Computer-Human Interaction Research and Applications*, in Funchal, Madeira, Portugal, pp. 111-117, 2017.

[52] M. Omidyeganeh, S. Shirmohammadi, S. Abtahi, A. Khurshid, M. Farhan, J. Scharcanski, B. Hariri, D. Laroche and L. Martel, "Yawning Detection Using Embedded Smart Cameras," *IEEE Transactions on Instrumentation and Measurement,* vol. 65, pp. 1-13, 2016.

[53] T. H. Lam and N. T. Do, "Algorithm to detect driver's drowsiness based on nodding behavior," *International Journal of Soft Computing, Mathematics and Control (IJSCMC),* vol. 5, 2016.

[54] M. Omidyeganeh, A. Javadtalab and S. Shirmohammadi, "Intelligent driver drowsiness detection through fusion of yawning and eye closure," *IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems Proceedings,* pp. 1-6, 2011.

[55] V. Esra, C. Mujdat, E. Aytul, L. Gwen, B. Marian and M. Javier, "Automated Drowsiness Detection For Improved Driving Safety," *International Conference on Automotive Technologies,* 2008.

[56] B. Reddy, Y. Kim, S. Yun, C. Seo and J. Jang, "Real-Time Driver Drowsiness Detection for Embedded System Using Model Compression of Deep Neural Networks," 2017.

[57] G. Nisha, N. Tejas and Y. Aditya, "Drowsy Driver Detection in Video Sequences using LSTM with CNN Features," 31 Des 2017. [Online]. Available: https://github.com/nishagandhi/DrowsyDriverDetection. [Accessed 21 july 2019].