



Mémoire de fin d'étude

**Pour l'obtention du diplôme de Master**

Filière : Automatique

Spécialité : Automatique

**Présenté par :**

SEKKIL Hicham Mohamed

MEBROUKI Mahmoud

Thème

**Etude comparative entre les différentes architectures des réseaux de neurones convolutifs (CNNs) pour la détection de la fatigue du conducteur**

Soutenu publiquement, le 26 / 09 / 2021, devant le jury composé de :

M. Ghouti ABDELLAOUI	Maitre de conférences B	ESSA. Tlemcen	Président
Melle. Imane NEDAJR	Maitre de conférences B	ESSA. Tlemcen	Directeur de mémoire
M. Hicham MEGNAFI	Maitre de conférences A	ESSA. Tlemcen	Co-Directeur de mémoire
M. Mustapha Anwar BRAHAMI	Maitre de conférences B	ESSA. Tlemcen	Examineur 1
M. Zohir KARAOUZANE	Maitre assistant A	ESSA. Tlemcen	Examineur 2

Année universitaire : 2020 / 2021



Mémoire de fin d'étude  
**Pour l'obtention du diplôme de Master**

Filière : Automatique

Spécialité : Automatique

**Présenté par :**

SEKKIL Hicham Mohamed

MEBROUKI Mahmoud

Thème

**Etude comparative entre les différentes  
architectures des réseaux de neurones  
convolutifs (CNNs) pour la détection de  
la fatigue du conducteur**

Soutenu publiquement, le 26 / 09 / 2021, devant le jury composé de :

M. Ghouti ABDELLAOUI	Maître de conférences B	ESSA. Tlemcen	Président
Melle. Imane NEDAJR	Maître de conférences B	ESSA. Tlemcen	Directeur de mémoire
M. Hicham MEGNAFI	Maître de conférences A	ESSA. Tlemcen	Co-Directeur de mémoire
M. Mustapha Anwar BRAHAMI	Maître de conférences B	ESSA. Tlemcen	Examineur 1
M. Zohir KARAOUZANE	Maître assistant A	ESSA. Tlemcen	Examineur 2

Année universitaire : 2020 / 2021

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## **Remerciement**

*Tout d'abord, nous remercions le bon Dieu tout-puissant, de nous avoir donnés la force et l'audace de dépasser toutes les difficultés, et de mener à bien ce travail.*

*Au nom d'ALLAH le clément et le miséricordieux*

*Nous présentons notre immense gratitude et nos remerciements, les plus sincères à la directrice de ce mémoire, mademoiselle **Imane NEDJAR**, de nous avoir fait confiance et accepter de diriger ce travail, sa disponibilité, ses orientations pertinentes et avisées, sa patience, ses qualités scientifiques et surtout humaines ont constitué un apport considérable, sans lequel, ce travail n'aurait pas vu le jour. Merci pour toute l'aide que tu nous as apportée, pour ton encouragement et pour ton enthousiasme permanent.*

*Nous tenons à remercier monsieur **Hicham MEGNAFI**, codirecteur de ce mémoire, pour son soutien tout au long de nos travaux et pour ses conseils scientifiques et techniques. Nous sommes particulièrement reconnaissants à la confiance qu'il nous a toujours témoignée.*

*De même, nous souhaitons remercier les différents membres constituant le jury de mémoire pour avoir accepté de donner de leur temps pour évaluer notre travail.*

*Enfin, nous voudrions remercier nos familles et nos proches pour leurs soutiens sans failles qu'ils nous ont apportés depuis toujours. Nous remercions également tous nos amis et tous ceux qui ont été impliqués d'une manière ou d'une autre dans la réussite de ce travail.*

*Dédicace*

*À tous ceux qui m'ont appris une lettre.*

*Hicham.*

## ***Dédicace***

*Du fond du cœur je dédie ce travail à*

*Mes chers parents que nulle dédicace ne puisse exprimer mes sincères sentiments  
et mon éternelle gratitude, pour leur patience illimitée, leur encouragement continu,  
leur aide, en témoignage de mon profond amour et respect pour leurs grands sacrifices.*

*Mes frères et ma belle-sœur, pour leur amour et leur soutien*

*Mes chers grands parents et toute ma famille.*

*Tous mes amis et tous ce qui m'ont aidé de près ou de loin dans la réalisation de ce travail.*

*Mahmoud.*

## Résumé

La fatigue du conducteur diminue sa vigilance ce qui peut mener à des conséquences graves comme les accidents routiers.

Dans ce mémoire nous avons proposé une méthode non intrusive de la détection de la fatigue du conducteur basé sur l'intelligence artificielle et la vision par ordinateur.

En effet, nous avons réalisé une étude comparative entre les différentes architectures du réseau neurone convolutive pour la détection de la fatigue au volant en analysant les caractéristiques faciales du conducteur (l'état des yeux et le bâillement). Par la suite, l'architecture sélectionnée a été utilisée pour calculer les deux mesures PERCLOS et FOM (PERcentage of eye CLOsure and yawning Frequency Of Mouth) afin de prédire la somnolence du conducteur. La méthode a obtenu un taux de détection égale à 81,25 %.

**Mots clés :** Fatigue, Analyse des Caractéristiques Faciales, Apprentissage Profond, Réseau de Neurones Convolutives, PERCLOS, FOM.

---

## Abstract:

### Comparative study between different architectures of convolutional neural networks (CNNs) for driver fatigue detection

Driver fatigue reduces vigilance which can lead to significant consequences such as road traffic crashes.

In this thesis, we have proposed a non-intrusive method of driver fatigue detection based on artificial intelligence and computer vision.

Thus, we carried out a comparative study between the different architectures of the convolutional neuron network for driver fatigue detection, by analyzing the facial characteristics of the driver (eye condition and yawning).

Subsequently, the selected architecture was used to calculate the two measurements PERCLOS and FOM (PERcentage of eye CLOsure and yawning Frequency Of Mouth) to predict driver drowsiness. The method obtained a detection rate equal to 81.25%.

**Keywords:** Fatigue, Facial Feature Analysis, Deep Learning, Convolutional Neural Network, PERCLOS, FOM.

---

## ملخص:

يقلل إجهاد السائق من يقظته مما قد يؤدي إلى عواقب وخيمة مثل حوادث الطرق. في هذه المذكرة اقترحنا طريقة غير تدخلية لاكتشاف إجهاد السائق تعتمد على الذكاء الاصطناعي ورؤية الحاسوب.

في البداية، أجرينا دراسة مقارنة بين مختلف الشبكات العصبية التلافيفية للكشف عن إرهاق السائق من خلال تحليل خصائص وجه السائق (حالة العين والتثاؤب). بعد ذلك، استخدمنا الشبكة المختارة لحساب قياسين PERCLOS و FOM (PERcentage of eye CLOsure and Frequency Of Mouth) من أجل التنبؤ بنعاس السائق. حصلت الطريقة على معدل كشف يساوي 81.25%.

الكلمات المفتاحية: التعب، تحليل خصائص الوجه، التعلم العميق، شبكة الخلايا العصبية التلافيفية، PERCLOS، FOM.

# Table des matières

Liste des acronymes .....	viii
Table des figures .....	x
Liste des tableaux.....	xiii
Introduction générale .....	1
Chapitre 1 : Systèmes de détection de la fatigue.....	3
1.1 Introduction.....	3
1.2 L'impact de la fatigue sur les accidents routières .....	4
1.3 Relation entre la somnolence et la conduite.....	4
1.4 Les indicateurs scientifiques de la fatigue.....	5
1.4.1 Les indicateurs physiologiques .....	6
1.4.2 Analyse de la conduite .....	6
1.4.3 Analyse faciale.....	7
1.5 Etat de l'art sur les approches de détection de fatigue .....	7
1.5.1 Approches basées sur les performances du conducteur .....	8
1.5.2 Approches basées sur les signaux physiologique.....	10
1.5.3 Approches basées sur l'analyse faciale.....	12
1.6 Comparaison entre les méthodes de détection de fatigue .....	15
1.7 Les Technologies Utilisé Actuellement dans l'Industrie Automobile.....	16
1.7.1 Contrôle d'alerte du conducteur Volvo (Volvo Driver Alert Control) .....	17
1.7.2 Aide À L'attention Mercedes-Benz (Mercedes-Benz Attention Assist) .....	17
1.7.3 Alerte du conducteur Volkswagen (Volkswagen Driver Alert) .....	18
1.7.4 Alerte Attention Conducteur Peugeot.....	19
1.7.5 La Bague Arrêtez de dormir (StopSleep).....	19
1.7.6 Boîtier MiniTOUCANGO de Innov+ .....	20

1.8 Conclusion .....	20
<b>Chapitre 2 : L'intelligence artificielle et l'apprentissage profond .....</b>	<b>22</b>
2.1 Introduction.....	22
2.1.1 Définition.....	23
2.1.2 Historique.....	23
2.1.3 Sous-Domains de l'Intelligence Artificielle.....	24
2.1.4 Les avantages de DL et pourquoi nous l'avons choisi plutôt que ML.....	25
2.2 L'Apprentissage Profond .....	26
2.2.1 Définition.....	26
2.2.2 Historique.....	26
2.2.3 Des réseaux neuronaux aux réseaux neuronaux profonds .....	29
2.2.4 Principe de fonctionnement de l'apprentissage profond.....	31
2.2.5 Les types d'apprentissage profond.....	32
2.2.6 Les types des réseaux de neurones profonds.....	34
2.2.7 Les couches des réseaux neurones profond .....	37
2.3 Réseaux de Neurones Convolutifs (ConvNets/CNNs) .....	37
2.3.1 Le principe de Réseaux de neurones convolutifs.....	38
2.3.2 Types de couches dans le réseau neuronal convolutif .....	39
2.3.3 Les Architectures de CNN .....	42
2.4 Applications de l'apprentissage profond.....	48
2.4.1 Reconnaissance de caractères.....	48
2.4.2 Reconnaissance vocale .....	49
2.4.3 Les jeux vidéo .....	49
2.4.4 Diagnostic de la maladie .....	49
2.4.5 Les voitures à conduite autonome.....	50

2.5 Conclusion ..... 50

**Chapitre 3 : La méthode proposée pour la détection de la fatigue du conducteur ..... 51**

3.1. Introduction ..... 51

3.2. Mise en place de l'environnement de programmation ..... 51

3.3. L'approche Proposée ..... 53

    3.3.1 Étape 1 : Préparation des données ..... 55

    3.3.2 Étape 2 : Comparaison entre les différentes architectures ..... 65

    3.3.3 Étape 3 : Détection de fatigue ..... 79

3.4 Analyse des résultats ..... 82

3.5 Limitations du travail ..... 85

3.6 Conclusion ..... 85

**Conclusion Générale ..... 87**

**Annexe A ..... 88**

**Annexe B ..... 91**

**Références ..... 100**

# Liste des acronymes

<b>NHTSA</b>	National Highway Traffic Safety Administration
<b>CNN</b>	Convolutional Neural Networks
<b>ASIRT</b>	International Safety Record for Road Travel
<b>DNSR</b>	Délégation Nationale à la Sécurité Routière
<b>INSV</b>	Institut National du Sommeil et de la Vigilance
<b>SVM</b>	Support Vector Machine
<b>KNN</b>	K-Nearest Neighbors
<b>SDLP</b>	Standard Deviation of Lateral Position
<b>EEG</b>	ElectroEncéphaloGraphie
<b>ECG</b>	ElectroCardioGraphie
<b>EOG</b>	Electro-OculoGraphie
<b>MLP</b>	Multi-Layer Perceptron
<b>PNN</b>	Prototype Nearest Neighbor
<b>VFC</b>	Variabilité de la Fréquence Cardiaque
<b>PERCLOS</b>	PERcentage of eye CLOSsure
<b>FOM</b>	Frequency Of Mouth
<b>IA / AI</b>	Intelligence Artificielle / Artificial Intelligence
<b>ML</b>	Machine Learning
<b>DL</b>	Deep Learning
<b>ANN</b>	Artificial Neural Networks
<b>DNN</b>	Deep Neural Networks
<b>FF</b>	Feed Forward
<b>DFE</b>	Deep Feed Forward
<b>RNN</b>	Recurrent Neural Networks
<b>LSTM</b>	Long-Shor Term Memory
<b>ReLU</b>	Rectified Linear Unit
<b>ILSVRC</b>	ImageNet Large Scale Visual Recognition Challenge
<b>OCR</b>	Optical Character Recognition
<b>EAR</b>	Eye Aspect Ratio
<b>MAR</b>	Mouth Aspect Ratio

<b>YawDD</b>	Yawning Détection Dataset
<b>FN</b>	Faux Négatif
<b>FP</b>	Faux Positif
<b>VN</b>	Vrai Négatif
<b>VP</b>	Vrai Positif
<b>GPU</b>	Graphics Processing Unit

# Table des figures

<b>Figure 1-1</b> : Les principaux indicateurs scientifiques de la fatigue [11].	6
<b>Figure 1-2</b> : Organigramme des différentes approches de la détection de fatigue [17].	8
<b>Figure 1-3</b> : Les différentes bandes de fréquences de EEG [31].	12
<b>Figure 1-4</b> : Le système de détection de la fatigue de Volvo dans la voiture CX90 [43].	17
<b>Figure 1-5</b> : L'alerte du système de détection de la fatigue de Mercedes-Benz (ATTENTION ASSIST) [45].	18
<b>Figure 1-6</b> : L'alerte du système de détection de la fatigue de Volkswagen (Driver Alert) [46].	18
<b>Figure 1-7</b> : La bague StopSleep montrant ses 8 capteurs de détection de la fatigue [50].	19
<b>Figure 1-8</b> : La camera MiniTOUCANGO et ses 5 niveaux d'alerte lumineux [51].	20
<b>Figure 2-1</b> : Une ligne du temps montrant l'histoire de l'IA moderne et ses principaux événements depuis 1943 jusqu'en 2015 [56].	<b>Error! Bookmark not defined.</b>
<b>Figure 2-2</b> : Un diagramme de Venn montrant l'IA et ses sous-domaines inclus les uns dans les autres, chacun avec un exemple de technologie d'IA [57].	24
<b>Figure 2-3</b> : Un neurone biologique en comparaison avec un neurone artificiel (un perceptron) [66].	28
<b>Figure 2-4</b> : Un réseau neuronal simple avec une couche d'entrées et une couche de sorties neuronale [70].	30
<b>Figure 2-5</b> : Un réseau neuronal comportant deux couches cachées ou plus (réseau neuronal profond).	31
<b>Figure 2-6</b> : Schéma d'un neurone ouvert [70].	31
<b>Figure 2-7</b> : Schéma des réseaux de neurones convolutifs [78].	39
<b>Figure 2-8</b> : Exemple de fonctionnement de Max pooling et Average pooling [80].	41
<b>Figure 2-9</b> : Histoire évolutive des CNNs montrant les innovations architecturales [66].	43
<b>Figure 3-1</b> : Logos des outils utilisés dans cette étude.	53
<b>Figure 3-2</b> : La première étape de la méthode, génération de la base de données.	54
<b>Figure 3-3</b> : La deuxième étape de la méthode, étude comparative et choix de l'architecture optimale.	55
<b>Figure 3-4</b> : La troisième étape de la méthode, test de l'architecture et prédiction de la fatigue.	55
<b>Figure 3-5</b> : Exemples de données dans YawDD.	56
<b>Figure 3-6</b> : Redimensionnement de l'image.	57
<b>Figure 3-7</b> : Mise à l'échelle de l'image en niveaux de gris.	57

<b>Figure 3-8</b> : Les 68 point de repères du visage défini dans la bibliothèque Dlib. ....	58
<b>Figure 3-9</b> : Les régions d'intérêt prédit par la fonction de prédiction de Dlib.....	58
<b>Figure 3-10</b> : Arbre des dossiers des ensembles de données générés.....	60
<b>Figure 3-11</b> : Le repère facial de la région des yeux. ....	61
<b>Figure 3-12</b> : Les valeurs de EAR et MAR obtenu a partir des vidéos d' un homme et une femme.....	62
<b>Figure 3-13</b> : La détection de l'état des yeux et de la bouche .....	63
<b>Figure 3-14</b> : Dessin des contours sur les images de données pour améliorer les résultats de l'apprentissage (avant et après la séparation des données).....	64
<b>Figure 3-15</b> : Les masques des images (avant et après la séparation des données) .....	65
<b>Figure 3-16</b> : Les courbes de précisent et l'erreur du modèle VGG16 pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement. ....	67
<b>Figure 3-17</b> : Les courbes de précisent et l'erreur du modèle VGG19 pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement. ....	68
<b>Figure 3-18</b> : Les courbes de précisent et l'erreur du modèle EfficientNetB7 pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement. ....	69
<b>Figure 3-19</b> : Les courbes de précisent et l'erreur du modèle DenseNet pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement. ....	70
<b>Figure 3-20</b> : Les courbes de précisent et l'erreur du modèle InceptionV3 pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement. ....	71
<b>Figure 3-21</b> : Les courbes de précisent et l'erreur du modèle ResNet50V2 pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement. ....	72
<b>Figure 3-22</b> : Les courbes de précisent et l'erreur du modèle InceptionResNetV2 pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement. ....	73
<b>Figure 3-23</b> : Les courbes de précisent et l'erreur du modèle Xception pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement. ....	74
<b>Figure 3-24</b> : Les courbes de précisent et l'erreur du modèle NasNetMobile pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement. ....	75
<b>Figure 3-25</b> : Les courbes de précisent et l'erreur du modèle MobileNetV2 pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement. ....	76
<b>Figure 3-26</b> : Un exemple des images classifiées correctement avec le model MobileNetV2.....	79

**Figure 3-27** : Les courbes des valeurs de PERCLOS en fonction de l'état du conducteur avec le modèle MobileNetV2..... 81

**Figure 3-28** : Les courbes des valeurs de FOM en fonction de l'état du conducteur avec le modèle MobileNetV2..... 81

**Figure 3-29** : L'organigramme de décision de la détection de la fatigue. .... 82

# Liste des tableaux

<b>Tableau 1-1</b> : Comparaison entre les approches de détection de fatigue .....	15
<b>Tableau 1-2</b> : Comparaison entre les méthodes basée sur l'analyse faciale d'image .....	16
<b>Tableau 3-1</b> : La distribution équilibré d'ensemble de donné sur les trois étapes d'entraînement.....	64
<b>Tableau 3-2</b> : Le résultat obtenu dans la dernière 10ème époque par chaque architecture testée dans la classification de l'état des yeux.....	77
<b>Tableau 3-3</b> : Le résultat obtenu dans la dernière 10ème époque par chaque architecture testée dans la classification de l'état de la bouche.....	78
<b>Tableau 3-4</b> : Une liste des vidéos séparées selon l'état avec le PERCLOS et le FOM calculés et l'état prédit correspondant avec la vraie prédiction marquée en vert et la fausse en rouge.....	83
<b>Tableau 3-5</b> : Matrice de confusion de l'évaluation de nombre de vraies et fausses prédictions du modèle..	84
<b>Tableau 3-6</b> : Une liste de vidéos assemblées de différents états avec les PERCLOS et FOM calculés et l'état prédit correspondant avec la vraie prédiction marquée en vert et la fausse en rouge.....	84

# Introduction générale

L'industrie automobile a connu de grands progrès ces dernières années. Malgré cela, les accidents de la route font encore de nombreuses victimes chaque année. Selon une étude menée par l'Organisation mondiale de la santé [1], Chaque année, plus de 1,35 million de personnes meurent des suites d'accidents de la circulation. De 20 à 50 millions de personnes supplémentaires souffrent de blessures non mortelles et nombre d'entre elles deviennent handicapées.

L'inattention du conducteur est l'une des principales causes de ces accidents, Une étude publiée par la National Highway Traffic Safety Administration (NHTSA), en 2015 [2], a indiqué que 94% des accidents de la route sont causés par des erreurs dues au conducteur. Plus précisément, la distraction et la fatigue sont les causes les plus fréquentes, représentant 41 % des blessures.

Ce qui explique l'intérêt des pays d'étudier les accidents routiers dans la tentative de les comprendre et d'y trouver des solutions préventives [3]. Pour ces raisons, il est nécessaire de concevoir un système de surveillance du conducteur afin de l'aider à éviter les accidents.

Les techniques de détection de fatigue des conducteurs ont de larges perspectives de contribution dans la prévention des accidents de la route. En effet, elles attirent de plus en plus l'attention des chercheurs de l'industrie automobile.

L'objectif principal de notre travail est d'utiliser et de tester les méthodes de l'intelligence artificielle pour concevoir un système de détection de la fatigue du conducteur.

Ce mémoire est organisé en trois chapitres comme suit :

- Dans le premier chapitre, on va introduire les systèmes de détection de la fatigue, et on présentera en détail les approches existant dans la littérature.
- Dans le deuxième chapitre nous présenterons le concept de l'intelligence artificielle et l'apprentissage en profondeur. Puis nous introduirons les réseaux de neurones convolutifs (CNNs) et leurs différentes architectures.
- Dans le dernier chapitre, on présentera l'approche que nous avons proposée pour la détection de la fatigue du conducteur, et aussi les résultats expérimentaux obtenus.

# Chapitre 1

## Systèmes de détection de la fatigue

### 1.1 Introduction

La conduite est une activité complexe qui consiste à effectuer plusieurs tâches simultanément : suivre la route, surveiller la vitesse, éviter les obstacles, contrôler le véhicule, etc. Il est donc clair que cette activité nécessite une très grande vigilance pour éviter les accidents.

Selon les dernières recherches menées par Association pour le record international de sécurité pour les déplacements routiers (the Association for the International Safety Record for Road Travel (ASIRT) [4], il a été constaté que les accidents de la route sont l'une des principales causes de décès dans le monde avec environ 1,2 million de personnes qui meurent chaque année, ce qui équivaut à une moyenne de 3 287 personnes par jour. Alors que le nombre de blessures non mortelles a été estimé entre 20 millions et 50 millions de personnes. En plus de nombreuses pertes financières estimées à 518 milliards de dollars américains sont enregistrées annuellement, ce qui est équivaut à 1,4 milliard de dollars chaque jour.

La situation en Algérie n'est pas différente de celle d'autres pays. En effet, selon les statistiques établies par le Délégation nationale à la sécurité routière (DNSR), l'année 2020 a enregistré 2.844 morts et 25.836 blessés à cause de 18.949 accidents de la circulation au niveau national [5].

## **1.2 L'impact de la fatigue sur les accidents routières**

La fatigue entraîne une perte d'attention, donc l'analyse des situations est moins efficace et le temps de réaction est beaucoup plus long que dans les cas normaux. Par conséquent, la prise de décision face à un danger est moins claire, ce qui augmente le risque d'accident.

La fatigue a un impact direct sur les accidents routiers, car elle est considérée comme l'une des principales causes d'accidents mortels. Des études ont confirmé que la fatigue est à l'origine d'environ 15 à 20 % des accidents mortels [6].

L'administration nationale de la sécurité routière aux États-Unis a indiqué que plus de 100 000 accidents de la route sont une conséquence directe de la somnolence au volant, entraînant 1550 décès, 71000 blessures et 12,5 milliards de dollars de pertes financières [7].

Une étude menée par le Professeur Damien Léger, président du Conseil scientifique de l'Institut national du sommeil et de la vigilance (INSV) a montré qu'aux premiers signes de somnolence, le conducteur doit s'arrêter parce que les risques d'avoir un accident dans la demi-heure qui suit sont multipliés par 3 ou 4. Ses réflexes sont altérés et plus il roule vite et plus les conséquences sont graves en cas d'accident [8].

## **1.3 Relation entre la somnolence et la conduite**

Notre corps a besoin de dormir suffisamment chaque jour. Plus nous manquons d'heures de sommeil, plus il est difficile de penser et d'agir comme nous souhaitons. Le manque de sommeil peut nous rendre moins alerte

et affecte notre coordination, notre jugement et notre temps de réaction au volant. C'est ce qu'on appelle une déficience cognitive.

Au volant, l'impact de la somnolence est similaire à celui de l'alcool. Des études ont montré que conduire pendant plus de 24 heures sans dormir est identique à la conduite avec une concentration d'alcool dans le sang de 0,10%<sup>1</sup> [9].

La somnolence est souvent le résultat d'une fatigue physique ou même mentale. D'autre part, toute activité qui se poursuit assez longtemps, causera à une difficulté à maintenir des performances qualifiées, y compris la conduite.

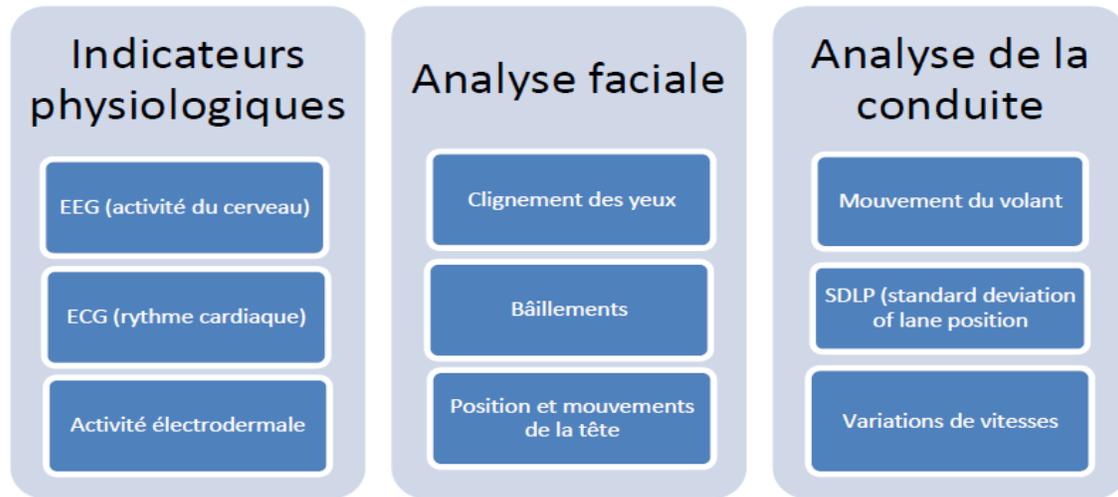
Une étude de Schmidt et al. [10] a montré qu'en cas de conduite prolongée, due à la fatigue, les conducteurs ne perçoivent pas leur propre niveau de somnolence et ne peuvent donc pas prendre la décision de se reposer. Par conséquent, afin de réduire ce phénomène, la somnolence du conducteur doit être surveillée en permanence.

#### **1.4 Les indicateurs scientifiques de la fatigue**

Il existe cependant un certain nombre d'indicateurs scientifiques permettant à un appareil de déterminer l'état de vigilance ou d'éveil du conducteur. Les principaux indicateurs, regroupés dans trois grandes catégories : les indicateurs physiologiques, l'analyse faciale et l'analyse de la conduite comme montre la figure suivante :

---

<sup>1</sup> La limite en Algérie est de 0,02%



**Figure 1-1** : Les principaux indicateurs scientifiques de la fatigue [11].

### 1.4.1 Les indicateurs physiologiques

Il a été établi que tout changement du niveau de vigilance, peut se traduire par des changements d'états psychophysiologiques, et selon Wolfram et al, nous pouvons mettre en évidence une diminution du niveau de vigilance associée à la fatigue grâce à des mesures de l'activité du système nerveux central [12].

Les signaux physiologiques les plus couramment utilisés pour mesurer l'état fonctionnel du conducteur sont notamment les niveaux d'activité : cérébrale, cardiaque, musculaire, respiratoire et électrodermale.

Ces analyses physiologiques peuvent nécessiter l'utilisation de certains outils spécifiques comme l'électroencéphalogramme, l'électrocardiogramme, l'électro-oculogramme, l'électromyogramme, etc.

### 1.4.2 Analyse de la conduite

Les performances globales du comportement du véhicule, peuvent être utilisées pour estimer le niveau de somnolence des conducteurs [13].

Alors qu'une étude menée par Arnedt et al a prouvé que des changements soudains et fréquents de la trajectoire ou de la vitesse de la voiture peuvent être considérés comme un indicateur de somnolence au volant [14].

Cela se fait en utilisant un capteur de direction sur le volant pour examiner la force exercée pendant la rotation et le changement (Steering Wheel Movements) [15], et la stabilité du contrôle de la trajectoire (Standard Deviation of Lane Position) qui nécessite l'usage d'une caméra frontale pour mesurer la position latérale du véhicule, ainsi que la variation de vitesse.

### **1.4.3 Analyse faciale**

Le comportement des conducteurs change avec l'arrivée de la somnolence. Ils font moins de mouvements de tête et font plus de maniérismes (se toucher la tête, se gratter etc.). Ils penchent la tête d'un côté, ils baillent plus mais, surtout, l'activité des paupières se modifie.

Les clignements des yeux ont été répertoriés comme indicateurs de fatigue, se produisent spontanément et représentent donc un phénomène naturel, simple et facilement accessible qui reflète l'activation du système nerveux central. Dans la littérature, il est constaté que le temps de fermeture des paupières et la fréquence des clignements augmentent avec la somnolence [16].

## **1.5 Etat de l'art sur les approches de détection de fatigue**

Différentes approches sont adoptées pour réaliser le système de détection de fatigue. Elles peuvent être divisées en trois catégories principales :

Les approches basées sur les performances du conducteur, les approches basées sur les signaux physiologiques du conducteur et les approches basées sur l'analyse faciale du conducteur, présentées la figure suivante :

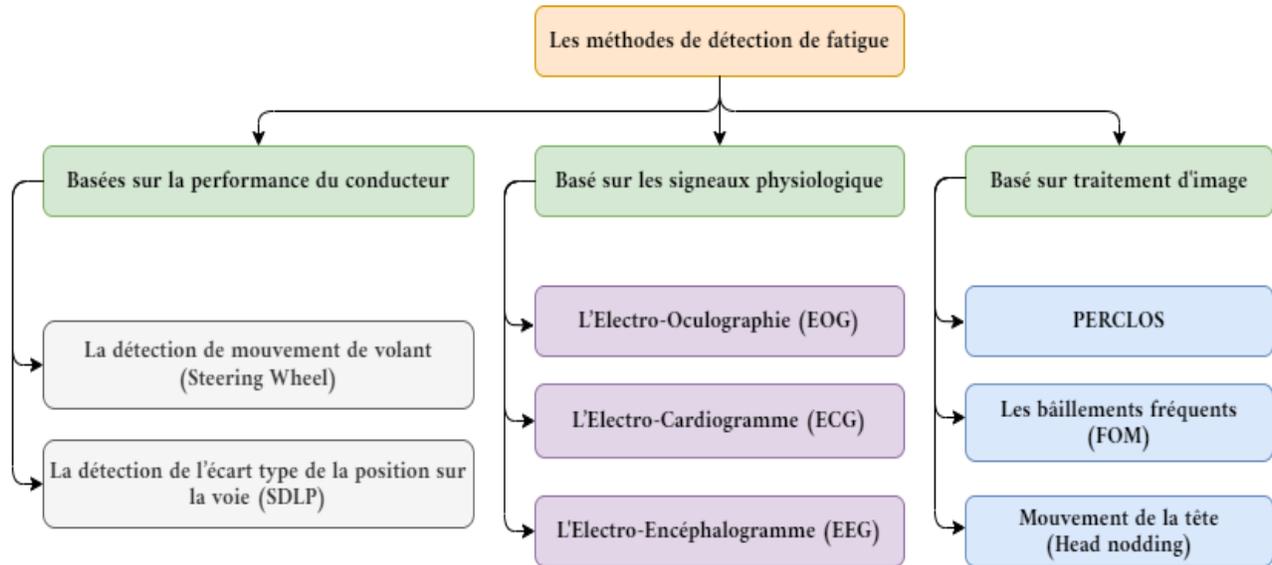


Figure 1-2 : Organigramme des différentes approches de la détection de fatigue [17].

### 1.5.1 Approches basées sur les performances du conducteur

Généralement, la conduite est influencée par plusieurs facteurs comme les tâches de conduire (le changement de vitesses et l'accélération, courbure et largeur de la voie etc.) et l'expérience de conducteur. En basant sur ces facteurs, on peut détecter l'état du conducteur.

Dans la plupart des cas, on utilise des captures pour surveiller les facteurs précédents par le placement de ces captures dans les différents composants de voiture comme le volant et la pédale d'accélération, Dans ces méthodes, plusieurs recherches étaient proposées, cependant la détection de mouvement de volant et la détection de l'écart type de la position sont les mesures les plus couramment utilisées.

#### A. La détection de mouvement de volant (Steering Wheel)

Krajewski et al [18] ont créé un système basé sur l'extraction des caractéristiques du mouvement de volant, qui a une relation avec le temps, la fréquence et l'environnement. Puis ils ont utilisé les algorithmes d'apprentissage de machine à vecteur supports SVM (Support Vector Machine) [19] et K-plus-proches-voisins

KNN (K-Nearest Neighbors) [20] pour classifier ses caractéristiques. Ce système atteint une précision de 86.1%.

## **B. La déviation standard de la position latérale (SDLP)**

La mesure de la déviation standard de la position latérale (*Standard Deviation of Lateral Position - SDLP*) est utilisée comme indicateur comportemental de la somnolence. Ce test restitue la façon dont le conducteur maintient la position de son véhicule sur la chaussée. Elle fournit des indications sur ses écarts de trajectoire et sa capacité à contrôler le mouvement latéral du véhicule. Ce test possède une haute reproductibilité et est significativement affecté par la somnolence et la diminution de l'attention au volant.

La SDLP est mesurée en centimètres, à l'aide d'un dispositif électro-optique monté à l'arrière du véhicule qui enregistre en continu la position latérale du véhicule par rapport à la ligne de délimitation de la voie (ligne blanche).

Le SDLP représente la variation de la position latérale. C'est un indicateur de stabilité du contrôle de la trajectoire. Son calcul nécessite l'usage d'une caméra frontale pour mesurer la position latérale du véhicule. Cette mesure permet d'évaluer les niveaux de vigilance du conducteur [21].

Il s'agit d'un indicateur simple à calculer, mais sensible aux caractéristiques de la route. La plupart des études de SDLP [22] [23] sont effectuées sur autoroute. Les auteurs notent que la fatigue du conducteur contribuerait à la détérioration du contrôle de trajectoire latérale (SDLP augmenté), Alors que la vigilance du conducteur contribuerait à de bonnes performances de contrôle des voies latérales (SDLP réduit).

## **1.5.2 Approches basées sur les signaux physiologique**

En cas de fatigue et de somnolence, une personne subite de nombreux changements dans son corps, notamment des changements dans la vitesse du rythme cardiaque, l'état cérébral et l'état musculaire. Sur la base de ces changements, de nombreuses études ont été proposées permettant de détecter la fatigue et la somnolence. Dans ces techniques en général les capteurs portés par le conducteur sont utilisés, ces capteurs permettent la capture de plusieurs types de signaux, L'électroencéphalographie (EEG) pour identifier « l'état du cerveau », l'électro-oculographie (EOG) pour prédire « le mouvement des yeux » et l'électrocardiographie (ECG) pour « la fréquence cardiaque ».

### **A. L'Electro-Oculographie (EOG)**

Les signaux EOG ont été utilisé pour surveiller l'état du conducteur pendant la conduite. Dans la littérature on trouve les méthodes qui ont utilisé les signaux EOG pour surveiller l'état du conducteur pendant la conduite. Et Ceci est fait en classifier les mouvements des yeux à l'aide des signaux vertical et horizontal de l'EOG obtenu à partir des capteurs. La classification a été faite par SVM dans [24] et par PNN (Prototype Nearest Neighbor) et MLP (perception multicouche) [25].

### **B. L'Electro-Cardiographie (ECG)**

D'autres mesures liées à l'activité du système nerveux périphérique ont été employées [26]. Lorsque le conducteur devient somnolent, le système nerveux autonome est affecté, ce qui peut être identifié par une variation de la fréquence cardiaque. Par conséquent, la variabilité de la fréquence cardiaque (VFC) peut être adoptée comme un indicateur pour détecter la somnolence.

La variabilité de la fréquence cardiaque (VRC) fournit un moyen passif de quantifier physiologiquement la somnolence. Il est défini comme la mesure de la variation des battements cardiaques et est calculé en analysant la série chronologique d'intervalles battement à battement [27]. La VRC a déjà été utilisée pour examiner la charge de travail mentale, le stress, et la fatigue du conducteur. En ce qui concerne la fatigue du conducteur, le VRC peut fournir des données utiles sur le moment où la fatigue devient un problème pendant la conduite.

Une augmentation de la VRC est une indication d'une diminution de la charge de travail mentale, qui peut survenir chez les conducteurs somnolents au cours d'une conduite monotone prolongée [28]. Mais une charge de travail plus faible peut également être liée à une vigilance moindre, ce qui peut avoir un impact négatif sur les performances du conducteur.

### **C. L'Electro-Encéphalographie (EEG)**

Cette technique est basée sur l'enregistrement de l'activité du cerveau à partir d'un EEG, Les enregistrements EEG se font à l'aide d'électrodes qui décèlent les modifications de la vigilance dans certaines zones du cerveau.

D'autres travaux ont montré que les signaux électroencéphalographiques (EEG) Contient des informations susceptibles de renseigner sur la charge mentale du conducteur [29].

Arun et al. Montre dans une étude que L'EEG contient différentes bandes de fréquences associées à différents états de vigilance du conducteur [30]. Où la plage delta entre 0,5 et 4 Hz correspond à l'activité du sommeil profond, la bande thêta entre 4 et 8 Hz correspond à la somnolence, la plage alpha entre 8 et 13 Hz correspond à la relaxation et la plage bêta entre 13 et 25 Hz correspond à l'excitation.

La figure 1.3 montre les différentes bandes de fréquences de EEG :

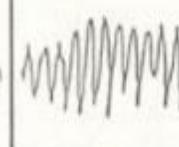
DELTA	THETA	ALPHA	BETA	GAMMA
Inférieure à 4Hz	4 à 8 Hz	8 à 13 Hz	13 à 35 Hz	Supérieure à 35Hz
Sommeil profond	Somnolence	Relaxation	Une attention à son maximum	Excitation
				

Figure 1-3 : Les différentes bandes de fréquences de EEG [31].

Tel que :  $\alpha$  : relaxation légère ;  $\beta$  : éveil et activité ;  $\gamma$  : gestion des informations

$\delta$  : sommeil et méditation profondes ;  $\theta$  : relaxation profonde

### 1.5.3 Approches basées sur l'analyse faciale

Les approches précédentes sont classées comme des approches intrusives car elles influencent l'attention de conducteur. Les recherches sont orientées vers les méthodes basées sur les traitements des images où on utilise une caméra pour extraire les symptômes de somnolence du conducteur comme les mouvements des yeux, les bâillements fréquents, les mouvements de tête [32].

La plupart des études publiées sur l'utilisation des traitements des images se concentrent sur les mouvements des yeux et plus précisément sur la mesure PERCLOS car plusieurs études prouvent que cette mesure est une mesure fiable pour prédire la somnolence. [33]

#### A. Pourcentage de Fermeture des Yeux (PERCLOS)

La mesure PERCLOS consiste à calculer le pourcentage de fermeture des paupières sur la pupille (où la paupière couvre 80 %). Il mesure ce phénomène par intervalles de 1 à 3 minutes et en déduit un indice de fatigue [34].

Le PERCLOS peut être obtenu par l'équation :

$$f_{PERCLOS} = \frac{n_{fermer}}{N_{fermer\&ouvert}} \times 100\% \quad (1.1)$$

$n_{fermer}$  : représente le nombre d'images d'œil fermés à un moment donné.

$N_{fermer\&ouvert}$  : représente le nombre total d'ouvertures et de fermetures d'œil dans une période.

Garcia et al. [35] Présentent un système de prédiction de l'état du conducteur. Les auteurs utilisent l'algorithme de Viola et Jones dans l'étape de détection du visage et les yeux du conducteur, Puis ils estiment l'état des yeux avec la projection intégrale, et la dernière étape est prédire l'état du conducteur avec la mesure PERCLOS.

W. Zhang et al. Ont proposé un autre système [36], ont proposé un système de détection de la somnolence du conducteur basé sur PERCLOS et aussi cinq autres mesures qui sont : durée maximale de fermeture, fréquence de clignement, niveau moyen d'ouverture des yeux, vitesse de fermeture des yeux et vitesse d'ouverture des yeux. Et pour classer ces mesures, W. Zhang et al ont utilisé la méthode discriminante linéaire de Fisher.

Bien que des études confirment que PERCLOS est l'un des meilleurs moyens de détecter la fatigue, certains experts s'inquiètent de l'effet du comportement oculaire sans rapport avec les niveaux de fatigue sur la précision des mesures. Ils ont constaté que la poussière, un changement soudain d'éclairage, l'éblouissement et les changements d'humidité étaient des facteurs qui n'étaient pas associés à la fatigue, mais pouvaient affecter le comportement des yeux du conducteur [37]. Par conséquent, ce système peut être soumis à des taux de fausses alarmes plus élevés. Trutschel et al. Ont effectué une étude détaillée et ont prouvé sur huit sessions de test, chacune durant une heure, que le PERCLOS a un taux d'erreur de 35%. [33]

## **B. Les bâillements fréquents (Yawning)**

Dans presque tous les systèmes, où le bâillement est détecté comme un symptôme de fatigue, les méthodes de reconnaissance sont établies de manière similaire. La caméra enregistre une image du visage. La bouche est

alors identifiée et son état (fermé ou ouvert) est détecté par analyse d'image. Si la bouche est ouverte, nous devons décider s'il bâille.

Dariusz Sawicki [38] propose un système qui peut distinguer entre les bâillements fréquents et l'ouverture de la bouche quand on parle, on chante où on sourira. Ce système est composé de trois étapes principales la première est la distinction des lèvres, ceci se fait avec l'algorithme de détection des repères faciaux et la deuxième étape est l'estimation de l'état de ces lèvres après le calcul de rapport de la hauteur de la bouche à leur largeur (Mouth Aspect Ratio MAR). Dans la dernière étape, la distinction entre la parole, le rire et le bâillement se fait à l'aide de la mesure FOM.

La fréquence de la bouche (FOM) montre le ratio de bouche ouverte en fonction du nombre de bouche ouverte et fermée.

Le calcul de FOM est similaire au calcul de PERCLOS et ça valeur peut être calculée comme suit :

$$f_{FOM} = \frac{n_{ouvert}}{N_{fermer\&ouvert}} \times 100\% \quad (1.2)$$

$n_{ouvert}$  : représente le nombre d'images de la bouche fermés à un moment donné.

$N_{fermer\&ouvert}$  : représente le nombre total d'ouvertures et de fermetures de la bouche dans une période.

### **C. Mouvement de la tête (Head nodding)**

La relation entre le mouvement de la tête du conducteur et la fatigue du conducteur a été largement étudiée.

Bergasa et al. [39] Considère que le hochement de tête est caractérisé par le fait que la tête s'écarte excessivement de sa position et de son orientation nominale pendant une période de temps prolongée ou trop fréquemment. Sur la base de sa conclusion, la fréquence de hochement de tête a été choisie pour évaluer la fatigue du conducteur. Kaplan considère que le hochement de tête se caractérise soit par un léger balancement,

soit par une secousse sporadique de la tête, soit par un hochement de tête involontaire et la relation entre le hochement de tête et la fatigue du conducteur est discutée par l'auto-évaluation du conducteur [40].

Ji et al. [41] Utilise la caméra à objectif grand angle se concentrant sur le visage pour suivre le mouvement de la tête du conducteur, et constate que les inclinaisons fréquentes de la tête indiquent généralement le début de la fatigue du conducteur.

## 1.6 Comparaison entre les méthodes de détection de fatigue

Le tableau suivant contient une simple comparaison entre les méthodes de détection de somnolence au volant du conducteur :

Approches	Avantages	Inconvénients
<ul style="list-style-type: none"><li>• La performance du conducteur</li></ul>	<ul style="list-style-type: none"><li>• Non intrusive</li></ul>	<ul style="list-style-type: none"><li>• Dépend des conditions extérieures</li></ul>
<ul style="list-style-type: none"><li>• L'analyse faciale du conducteur</li></ul>	<ul style="list-style-type: none"><li>• Non intrusive</li></ul>	<ul style="list-style-type: none"><li>• Affectés par les conditions d'éclairage de l'arrière-plan</li></ul>
<ul style="list-style-type: none"><li>• Les signaux physiologiques du conducteur</li></ul>	<ul style="list-style-type: none"><li>• Détection anticipée</li></ul>	<ul style="list-style-type: none"><li>• Intrusive</li><li>• Faible qualité de signal dans les solutions non intrusives</li></ul>

**Tableau 1-1** : Comparaison entre les approches de détection de fatigue [17].

Les méthodes basées sur l'analyse faciale du conducteur sont devenues de plus en plus populaires dans l'industrie automobile aujourd'hui [42]. Ces méthodes ont été approuvées pour leur grande précision dans la prédiction de la somnolence du conducteur en temps réel. Ainsi qu'elles ne sont pas intrusives et qui ne dérangent pas le conducteur.

Étant donné les nombreuses méthodes dans l'approche basée sur l'analyse faciale du conducteur on donne une simple comparaison entre les 3 méthodes : Etat des yeux, Bâillements fréquents, Mouvements de tête.

Méthodes	Avantages	Inconvénients
<ul style="list-style-type: none"><li>• <b>Etat des yeux (PERCLOS)</b></li></ul>	<ul style="list-style-type: none"><li>• Contient des informations qui décrit l'attention du conducteur</li><li>• Le grand danger réside dans la conduite avec les yeux fermés</li></ul>	<ul style="list-style-type: none"><li>• Influencé par les lunettes et l'éclairage.</li></ul>
<ul style="list-style-type: none"><li>• <b>Bâillements fréquents (Yawning)</b></li></ul>	<ul style="list-style-type: none"><li>• Détection du mouvement de la bouche facile</li></ul>	<ul style="list-style-type: none"><li>• Ne prédire pas toujours la somnolence</li></ul>
<ul style="list-style-type: none"><li>• <b>Mouvements de tête (Head nodding)</b></li></ul>	<ul style="list-style-type: none"><li>• La combinaison avec d'autres symptômes donne des bons résultats</li></ul>	<ul style="list-style-type: none"><li>• Ne prédire pas toujours la somnolence</li><li>• Besoin de vision stéréoscopique (caméra 3d ou logiciel 3D).</li></ul>

Tableau 1-2 : Comparaison entre les méthodes basée sur l'analyse faciale d'image [17].

## 1.7 Les Technologies Utilisé Actuellement dans l'Industrie Automobile

Les systèmes de détection de fatigue du conducteur sont utilisés dans l'industrie depuis un certain temps. En raison de leur importance et de la concurrence pour offrir aux clients une sécurité maximale, de nombreux leaders automobiles ont adopté ces systèmes dans leurs voitures.

### 1.7.1 Contrôle d'alerte du conducteur Volvo (Volvo Driver Alert Control)

Volvo a introduit son système de détection de la fatigue depuis une décennie maintenant, il se concentre sur le véhicule et non sur le conducteur [43]. Il est conçu principalement pour les autoroutes et non pour les zones urbaines, car il est activé lorsque la vitesse atteint 65 km/h et reste supérieur à 60 km/h. Le système est équipé d'une caméra fixée à l'arrière du rétroviseur intérieur pour détecter les marquages routiers et les compare aux mouvements du volant. Si des zigzags se produisent souvent, le conducteur est alerté par un message et un symbole sur l'écran du conducteur. Si les performances ne sont pas améliorées, l'alerte se répète en un instant [44].



Figure 1-4 : Le système de détection de la fatigue de Volvo dans la voiture CX90 [43].

### 1.7.2 Aide À L'attention Mercedes-Benz (Mercedes-Benz Attention Assist)

Le système surveille le comportement du conducteur pendant les 20 premières minutes de conduite. Lorsqu'il est vraiment en état d'alerte, il commence alors à comparer ses performances actuelles avec les performances capturées au début pour comprendre le comportement du conducteur et prendre une décision, par exemple pour éviter les nids-de-poule. L'angle du volant puisqu'il est équipé d'un capteur de direction mesurant la force exercée lors de la rotation, il vérifie également les déviations de trajectoire et les zigzags, en plus de prendre en considération les autres capteurs de la voiture comme le système de freinage, la stabilité de conduite, les paramètres du moteur et les limites de visibilité, etc [15].

Le système vérifie près de 90 indices pour déterminer si le conducteur est somnolent ou non. S'il détecte des erreurs liées à la fatigue du conducteur, le système l'avertit par un signal sonore et visuel qu'il doit faire une pause, et l'alerte est répétée toutes les 15 minutes s'il est ignoré [45].



**Figure 1-5 :** L'alerte du système de détection de la fatigue de Mercedes-Benz (ATTENTION ASSIST) [45].

### **1.7.3 Alerte du conducteur Volkswagen (Volkswagen Driver Alert)**

Le système recherche tout manque de concentration en surveillant en permanence et de manière intelligente les performances du conducteur sur la route, comme l'utilisation des pédales, le maniement du volant et les éventuels écarts de trajectoire [46]. Il évalue également la réaction du conducteur aux signaux routiers lorsque la vitesse est supérieure à 64 km/h [47].

Lorsque le système détecte une dégradation de la conduite indiquant une somnolence, il alerte le conducteur par un affichage visuel sur le tableau de bord et un signal sonore [48].



**Figure 1-6 :** L'alerte du système de détection de la fatigue de Volkswagen (Driver Alert) [46].

#### 1.7.4 Alerte Attention Conducteur Peugeot

Celui-ci est très similaire à celui de Volvo puisqu'il dispose également d'une caméra placée au sommet du pare-brise qui évalue la vigilance du conducteur en surveillant les déviations de trajectoire et les zigzags par rapport aux marquages au sol.

Le système dispose d'une alerte de temps de conduite lorsque le véhicule a roulé pendant plus de deux heures consécutives à plus de 65 km/h et commence alors à suggérer de se reposer [49].

#### 1.7.5 La Bague Arrêtez de dormir (StopSleep)

Un appareil en forme d'anneau à deux anneaux, contenant 8 capteurs, dédié à la mesure de l'activité des glandes sudoripares reflétant l'activité électrique de la peau. Ces modifications électriques de la peau représentent l'activité cérébrale à la surface de la peau et entraînent une augmentation ou une diminution de la conductance cutanée. L'anneau a trois états, le premier est l'état actif. Le second est le micro état de sommeil, qui commence lorsque le capteur détecte une diminution de l'activité électrique de la peau (indiquée par l'inattention et le manque de conscience), puis l'anneau commence à vibrer. Le dernier état est l'état de sommeil lorsque les symptômes de fatigue sont graves, puis la cloche vibrera et émettra un bip fort [50].



**Figure 1-7** : La bague StopSleep montrant ses 8 capteurs de détection de la fatigue [50].

### 1.7.6 Boîtier MiniTOUCANGO de Innov+

Un boîtier en forme de tête de robot, monté sur le tableau de bord, se compose d'une caméra infrarouge à haute vitesse de 60 images par seconde qui surveille le visage du conducteur dans toutes les conditions d'éclairage. La caméra utilise un capteur optique NIR (*Near-Infrared*) pour capter le visage du conducteur en se basant sur un model 68 points de repère sur le visage pour détecter les signaux de somnolence tels que la fermeture des paupières, les bâillements et les mouvements de la tête. Le boîtier capte également d'autres signaux provenant de l'environnement de conduite, comme la vitesse, l'emplacement, le type de route, la durée du trajet et l'heure de la journée, et combine toutes ces informations pour fournir un système de prédiction de la fatigue très précis.

MiniTOUCANGO offre 5 niveaux d'alerte de somnolence, chacun avec un son unique et des couleurs progressives, et il peut également détecter les distractions comme le fait de regarder son téléphone portable ou de regarder en dehors de la route avec 2 autres niveaux d'alerte de distraction [51].



Figure 1-8 : La camera MiniTOUCANGO et ses 5 niveaux d'alerte lumineux [51].

## 1.8 Conclusion

Dans ce chapitre nous avons présenté l'état de l'art des approches utilisées dans la détection de la fatigue du conducteur. Au terme de cet état de l'art, il est à souligner que la littérature présente des solutions assez

disparates quant au problème de détection de la fatigue du conducteur. Nous avons pris soin de les structurer, en distinguant les approches basées sur le comportement du conducteur, Les approches basé sur les signaux physiologique et Les approches basé sur l'analyse faciale.

Les approches basées sur le comportement du conducteur sont prometteuses. Elles se basent sur les actions du conducteur sur les organes de commande (angle volant, couple conducteur, freinage etc.) et de la position du véhicule par rapport à la voie (écart latéral, angle de cap, etc.). C'est une approche plus facile à mettre en œuvre et bénéfique dans le contexte de production en masse.

Les approches qui se basent sur les signaux physiologiques ne sont pas les plus simples à mettre en œuvre. Leur implémentation en masse dans les véhicules peut se révéler coûteuse [52], à cause des coûts de certains équipements supplémentaires requis. Elles sont aussi des méthodes intrusives, Ce qui peut gêner le conducteur. D'autre part elles sont fortement influencés par plusieurs paramètres comme le type de voiture, la géométrie de la route et l'expérience du conducteur [53].

Les approches qui se basent sur l'analyse faciale sont les plus étudiées à l'heure actuelle. Elles ne sont pas les plus simples à mettre en œuvre. Mais elles ne sont pas intrusives ceci les rend plus utiles dans les systèmes de détection de la fatigue.

## Chapitre 2

# L'intelligence artificielle et l'apprentissage profond

### 2.1 Introduction

Tout au long de l'histoire, l'homme a cherché une invention qui pourrait simuler l'esprit humain dans son modèle de pensée, et pour cela, des scientifiques, des découvreurs et des écrivains ont essayé de trouver une explication logique du concept d'intelligence artificielle.

Après des années de recherche, l'IA est sortie des laboratoires de recherche pour devenir une partie intégrante de notre vie quotidienne, à commencer par nous aider à naviguer dans les villes et à éviter les embouteillages, jusqu'à l'utilisation d'assistants virtuels pour nous aider à effectuer diverses tâches.

### **2.1.1 Définition**

L'intelligence artificielle (IA) est la science et l'ingénierie de la création de machines intelligentes qui peuvent penser par elles-mêmes et prendre leurs propres décisions afin d'imiter le comportement (intelligence) humain [54].

### **2.1.2 Historique**

L'origine de l'IA remonte à l'Antiquité, aux tentatives des philosophes classiques de systématiser la pensée humaine [55], et aux premiers romans de science-fiction (Frankenstein de Mary Shelley, 1818, ou R.U.R de Karel Čapek, 1920).

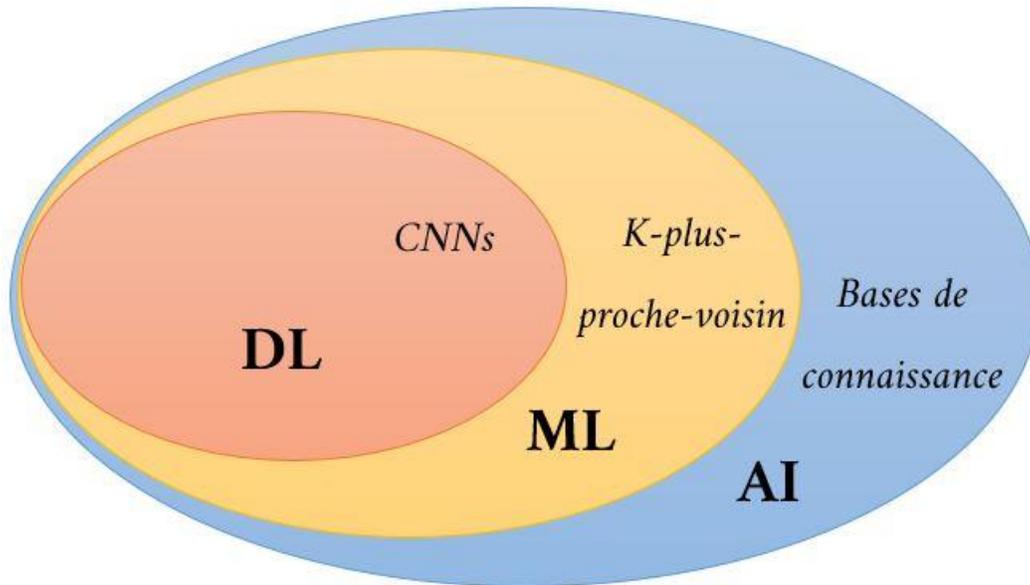
Mais les véritables recherches sur l'IA n'ont commencé qu'après la deuxième Guerre mondiale, lorsque certains scientifiques ont commencé à travailler sur des machines intelligentes.

La première conférence sur l'IA a été donnée par Alan Turing en 1947, dans laquelle il a décidé qu'il était préférable d'utiliser des programmes informatiques plutôt que de construire des machines dédiées. En 1950, il a publié un article intitulé "Computing Machinery and Intelligence" dans lequel il proposait un test qui a été baptisé d'après lui " le test de Turing " et qui devait déterminer si une machine pouvait être considérée comme intelligente ou non.

En revanche, de nombreux experts en IA estiment aujourd'hui que le test de Turing n'est pas une bonne mesure de l'intelligence artificielle. Ils pensent qu'il s'agit d'un test très médiocre, car il ne prend en compte que le comportement extérieur [55]. Mais pour l'instant, tous les autres substituts sont encore en phase de développement, ce qui laisse le jeu de l'imitation reste toujours le premier test de l'IA.

### 2.1.3 Sous-Domains de l'Intelligence Artificielle

Le domaine de l'IA contient deux sous-domaines majeurs. L'apprentissage machine (*Machine Learning ML*) et l'apprentissage profond (*Deep Learning DL*) comme montre la figure 2.2 :



**Figure 2-1** : Un diagramme de Venn montrant l'IA et ses sous-domaines inclus les uns dans les autres, chacun avec un exemple de technologie d'IA [56].

#### A. Apprentissage Machine (Machine Learning)

Une spécialisation de l'intelligence artificielle qui consiste en un ensemble d'algorithmes (généralement appelés *modèle* [57]) destinés à construire des applications intelligentes qui apprennent de l'expérience pour effectuer des prédictions intelligentes et s'améliorer au cours du temps sur la base d'un ensemble de données [58]. Cet ensemble de données définit également le type d'algorithme que nous choisissons pour le modèle.

La construction d'une application ML repose sur 4 étapes qui sont la sélection et la préparation d'un ensemble de données, le choix d'un algorithme à exécuter sur les données, l'entraînement de l'algorithme pour créer le modèle et enfin l'utilisation et l'amélioration du modèle [58].

Ce qui suit est une liste de certains des algorithmes ML couramment utilisés qui peuvent être appliqués à presque tout type de problème de données : Régression linéaire, Régression logistique, Arbre de décision, SVM, Naive Bayes, kNN, K-Means, Random Forest... etc. [59].

## **B. Apprentissage Profond (Deep Learning)**

Un sous-ensemble de l'apprentissage machine qui se compose de réseaux neuronaux pour effectuer le processus d'apprentissage sans intervention humaine.

Ce chapitre se concentre sur l'apprentissage profond et nous l'aborderons en détail dans les sections suivantes.

### **2.1.4 Les avantages de DL et pourquoi nous l'avons choisi plutôt que ML**

L'apprentissage profond est capable d'extraire, de traiter et d'apprendre des représentations de données dans leur forme originale, non modifiées, non étiquetées. Cette méthode est très limitée dans l'apprentissage machine et pourrait ne pas être possible, elle nécessite de sélectionner soigneusement les caractéristiques des données avec une grande compétence en ingénierie pour les représentations. L'avantage de DL sur ML est qu'il est capable de comprendre les caractéristiques des données brutes sans les extraire manuellement comme dans ML. Cela permet d'économiser du temps et des efforts, en particulier pour les données très volumineuses [60]. En outre, les résultats obtenus par l'apprentissage profond sont meilleurs, plus précis et plus perfectionnés que ceux de l'apprentissage machine [61].

## 2.2 L'Apprentissage Profond

### 2.2.1 Définition

L'apprentissage profond ou *Deep Learning* (DL) est un type d'intelligence artificielle dérivé de l'apprentissage machine où la machine est capable d'apprendre par elle-même, sans supervision, sans intervention humaine, en simulant la mécanique du cerveau humain pour exécuter avec précision des applications complexes basées sur des réseaux neuronaux artificiels [62].

### 2.2.2 Historique

Certains des premiers algorithmes d'apprentissage que nous connaissons aujourd'hui sont inspirés de l'apprentissage biologique, certains d'entre eux imitent la fonction du cerveau humain, et l'apprentissage profond est la version artificielle des neurones biologiques, appelés aussi réseaux neuronaux artificiels (ANN).

Les recherches sur les ANN remontent aux premières apparitions de l'IA. En 1943, le neuroscientifique Warren McCulloch et le logicien Walter Pitts ont publié l'article "*A logical calculus of the ideas immanent in nervous activity*", dans lequel ils proposaient le premier neurone artificiel permettant de calculer des fonctions arithmétiques et logiques [63].

Le raisonnement intelligent du cerveau humain est considéré comme la principale raison pour laquelle l'apprentissage profond adopte les neurones comme base. En inversant le mécanisme d'apprentissage du cerveau pour reproduire son comportement, il serait possible d'obtenir un modèle intelligent [56].

Nous allons donc examiner le neurone biologique et le neurone artificiel, en comparant les deux, en expliquant leur fonctionnement et les similitudes entre ces différents types de neurones.

## **A. Neurone Biologique**

Un neurone est une cellule vivante composée de trois parties principales : le corps cellulaire (soma), les dendrites et l'axone (plus l'axone terminal ou synapse).

Le corps cellulaire (soma), peut prendre des formes variables (pyramidale, sphérique, étoilée), sa forme est définie par une membrane qui sépare l'intérieur du neurone de l'extérieur. Un neurone est constitué de prolongements qui lui permettent d'établir des connexions avec d'autres cellules (neurones), ces prolongements qui reçoivent des signaux d'autres cellules sont appelés "dendrites" (entrées). L'unique prolongement qui diffuse le signal du neurone vers d'autres neurones est appelé "axone" (sortie) [64]. Ainsi, les dendrites et les axones établissent les connexions entre les cellules, en d'autres termes, l'axone reçoit l'impulsion du soma, sous la forme d'un signal électrique, et la transfère aux dendrites des neurones connectés via les "synapses" (axones terminaux). A la réception d'un signal, un neurone peut être activé ou non (opération booléenne) en fonction de la puissance du signal pour le déclencher. Si c'est le cas, le neurone est activé et le signal est transmis le long du cerveau en transmettant le message et en déclenchant le processus d'apprentissage [65]. (Voir figure 2.3).

## **B. Neurone artificiel**

Un neurone artificiel est une représentation mathématique et informatique du biologique qui est capable de traiter les entrées reçues d'un ensemble de connexions caractérisées par des poids synaptiques. Ces poids indiquent l'importance de la connexion [66]. L'entrée reçue est traitée à l'intérieur d'un nœud (neurone), puis transmise au nœud suivant en tant que sortie du premier neurone.

Le nœud reçoit le signal d'entrée comme catalyseur d'autres nœuds, il calcule une fonction de seuil linéaire simple et transmet le signal de sortie à un autre nœud uniquement lorsque le signal est plus fort que le seuil limite du nœud [67], le signal résultant est étiqueté par un poids permettant de préciser dans quelle mesure le

signal est amplifié ( $w > 0$ ) ou diminué ( $w < 0$ ). Cette opération entraîne la mise à jour des poids synaptiques et c'est ainsi que l'apprentissage se produit [65].

En 1958, Frank Rosenblatt a mis en œuvre les premiers modèles [56] dans son article intitulé "A Probabilistic Model for Information Storage and Organization in the Brain" (Un modèle probabiliste pour le stockage et l'organisation de l'information dans le cerveau), ce qui a donné lieu à un exemple plus flexible appelé le perceptron, capable de mettre à jour automatiquement les poids de classification d'une entrée [68].

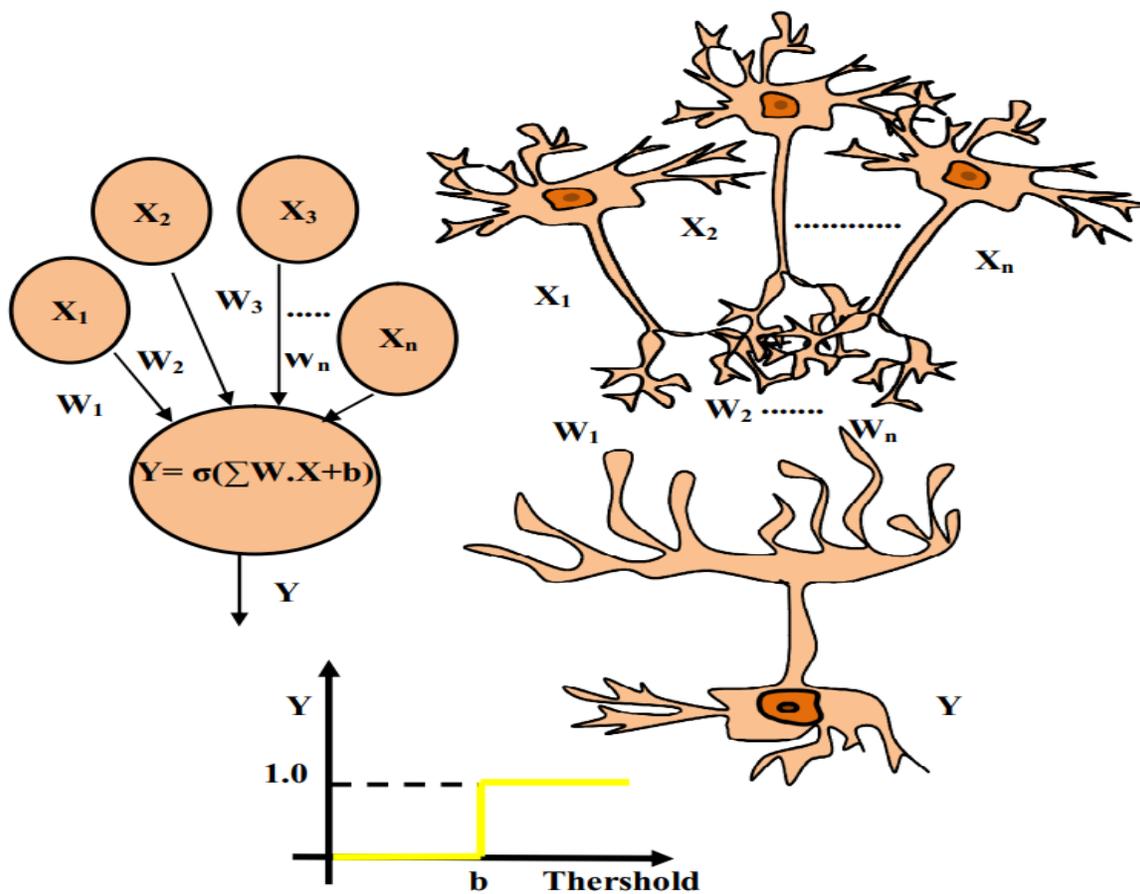


Figure 2-2 : Un neurone biologique en comparaison avec un neurone artificiel (un perceptron) [65].

### **C. L'Analogie Entre le Neurone Biologique et le Neurone Artificiel**

La similitude entre le neurone biologique et le neurone artificiel est que les dendrites sont les entrées et l'axone est la sortie et qu'ils représentent tous deux les connexions avec d'autres neurones pour les neurones biologiques et artificiels respectivement.

Les synapses sont les poids des connexions et l'activité dans le soma ressemble aux calculs internes des nœuds.

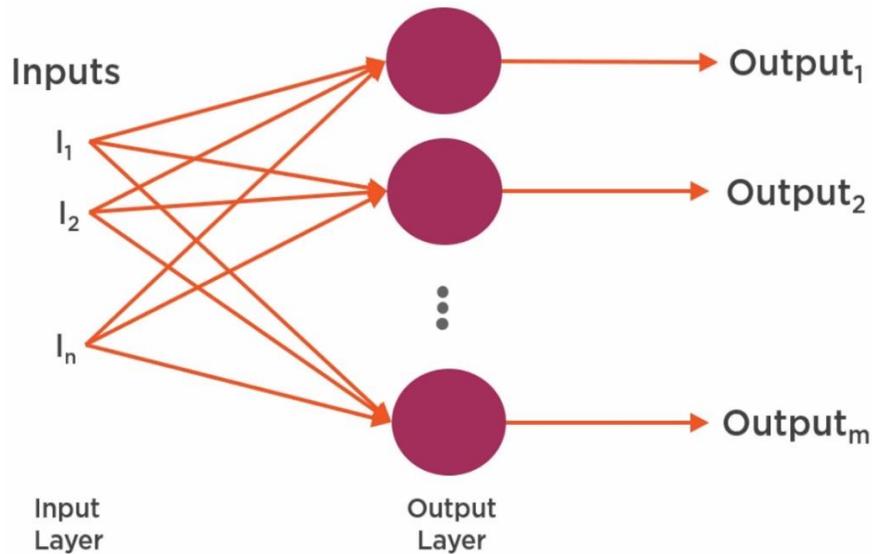
Une autre analogie est que le seuil limite pour le passage du signal ressemble à la puissance du signal électrique nécessaire pour déclencher l'activité dans les cellules connectées.

On note que les neurones artificiels ne sont généralement pas conçus pour être des modèles réalistes de la fonction biologique.

#### **2.2.3 Des réseaux neuronaux aux réseaux neuronaux profonds**

La structure d'un réseau neuronal simple est présentée dans la figure 2.4. Pour une meilleure compréhension, nous allons faire le parcours de l'entrée à la sortie et passer en revue les structures et la terminologie.

À gauche, nous avons la couche d'entrée, qui injecte des données dans le réseau. Ces données peuvent être des valeurs provenant d'une table de données, des images d'une caméra, des sons d'un enregistrement ou la sortie d'un capteur. La couche d'entrée ne modifie pas les données, elle les transmet simplement pour qu'elles soient traitées par les autres couches. Les données de la couche d'entrée sont transmises à une autre couche de neurones. Ces couches peuvent être de différents types et être chargées d'effectuer des transformations spécifiques sur les données, selon les besoins de la solution. Dans un réseau simple, la couche d'entrée peut être directement connectée à une couche de neurones de sortie, qui fournit les sorties finales.



**Figure 2-3 :** Un réseau neuronal simple avec une couche d'entrées et une couche de sorties neuronale [69].

Cependant, dans la plupart des réseaux, la couche d'entrée n'est pas directement connectée à la couche de sortie, mais à des couches cachées qui sont définies comme n'étant ni des couches d'entrée ni des couches de sortie. Les couches cachées que nous définissons pour le réseau sont désignées pour effectuer de nombreuses transformations intéressantes et nécessaires pour répondre aux exigences de la solution pour le problème donné, et faire ce que nous définissons pour elles.

*Mais comment peut-on appeler un réseau neuronal un réseau neuronal profond et d'où vient cet apprentissage profond ?*

Le réseau peut avoir de nombreuses couches cachées et s'il y a deux couches cachées ou plus, nous appelons le réseau un réseau neuronal profond. Dans cette étude, nous nous concentrons sur ces réseaux neuronaux profonds et sur leur processus d'apprentissage, connu sous le nom d'apprentissage profond.

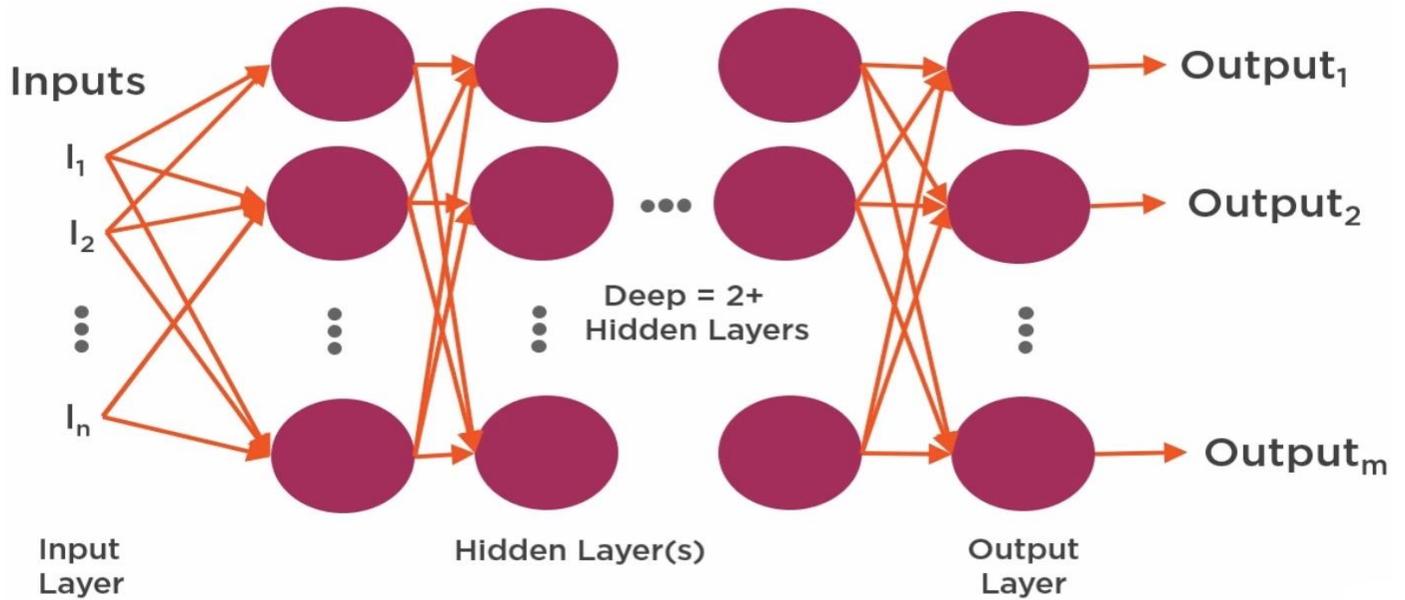


Figure 2-4 : Un réseau neuronal comportant deux couches cachées ou plus (réseau neuronal profond).

### 2.2.4 Principe de fonctionnement de l'apprentissage profond

Jusqu'à présent, nous n'avons vu que l'architecture globale des réseaux neuronaux, présentée à la figure 2.5, mais pour comprendre le fonctionnement de ces réseaux et le processus d'apprentissage, nous devons d'abord comprendre le fonctionnement interne d'un seul neurone, présenté à la figure 2.6.

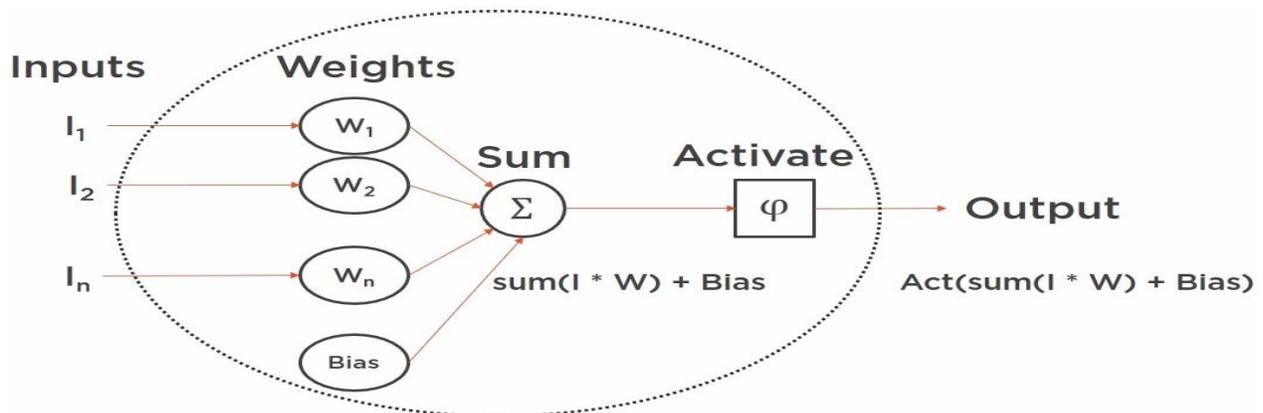


Figure 2-5 : Schéma d'un neurone ouvert [69].

Le neurone effectue une simple sommation mathématique des poids multipliés par la valeur d'entrée et ajoute un biais. Le produit de ces opérations passe par une fonction d'activation non linéaire et la sortie de la fonction d'activation est la sortie du neurone.

$$y = f(\sum_{i=1}^n (x_i * w_i) + b_i) \quad (2.1)$$

Il convient de noter que le modèle de perceptron est uniquement capable d'apprendre et de résoudre des problèmes linéairement séparables, ce qui est rarement le cas dans les problèmes du monde réel et certainement pas dans notre étude qui nécessite des réseaux neuronaux profonds.

Par conséquent, ce modèle ne peut comprendre aucune interaction entre deux variables d'entrée. Ainsi, afin d'étendre sa capacité à résoudre des fonctions non linéaires de  $x$ , nous pouvons appliquer le modèle linéaire non pas à  $x$  lui-même mais à une entrée transformée  $\varphi(x)$ , où  $\varphi$  est une transformation non linéaire. Apprendre cette transformation non linéaire est la stratégie de l'apprentissage profond [56].

### **2.2.5 Les types d'apprentissage profond**

L'une des principales caractéristiques du réseau neuronal est sa capacité à utiliser les données d'entrée pour former les poids et les biais afin que le signal sortant du neurone change en fonction des données d'entrée. Pour effectuer cette formation, nous exposons le réseau à des données, avec chaque ensemble de données, un algorithme est utilisé pour ajuster les poids et les biais afin de minimiser l'erreur du réseau dans la prédiction des valeurs des données. Cela se fait à travers des processus appelés propagation avant et propagation arrière. Lorsque ces processus sont terminés, on dit que le réseau est formé et que les poids et les biais de tous les neurones ont été ajustés pour donner les meilleurs résultats sur les données d'entraînement. Cette opération de formation ou d'apprentissage peut prendre différentes formes selon le type de données dont nous disposons pour alimenter le réseau, elle peut être supervisée, non supervisée ou semi-supervisée (hybride).

## **A. Apprentissage supervisé**

Dans le cas de l'apprentissage supervisé, nous avons à notre disposition un ensemble de données étiquetées. Les ensembles de données sont dits étiquetés lorsqu'ils comportent à la fois des variables d'entrée ( $X$ ) et une variable de sortie ( $Y$ ) ; les données ont donc déjà été associées à la bonne réponse et nous utilisons un algorithme pour apprendre la fonction de correspondance entre l'entrée et la sortie [70].

Cet ensemble établit la base de l'apprentissage et l'algorithme qui apprend à partir de cet ensemble peut être considéré comme un expert supervisant le processus, d'où le nom d'apprentissage supervisé. L'objectif est d'approcher efficacement la fonction de mise en correspondance afin que, chaque fois qu'une nouvelle donnée ( $X$ ) est entrée, elle puisse prédire le paramètre correspondant ( $Y$ ) en effectuant des prédictions répétées sur les données d'apprentissage tandis que l'enseignant corrige à chaque fois jusqu'à ce que l'algorithme atteigne un bon taux de performance, puis l'apprentissage s'arrête.

Les problèmes de ce type d'apprentissage peuvent être regroupés en problèmes de classification (lorsque le paramètre de sortie est une catégorie, comme rouge ou bleu) et de régression (lorsque le paramètre de sortie est une valeur réelle, comme les poids et les âges) et d'autres qui sont basés sur ces deux types de problèmes comme la recommandation pour le premier et la prédiction de séries temporelles pour le second [71].

## **B. Apprentissage non supervisé**

L'apprentissage non supervisé est l'opposé de l'apprentissage supervisé. Dans ce cas, nous ne fournissons pas de données d'apprentissage au réseau, mais simplement des données d'entrée ( $X$ ) sans sortie associée, et le réseau doit conclure de lui-même à la structure des données pour obtenir davantage d'informations. On parle d'apprentissage non supervisé car il n'y a pas de données étiquetées à partir desquelles les neurones peuvent

apprendre. Il est donc un peu plus difficile à mettre en œuvre car ils sont laissés seuls pour trouver des distributions intéressantes dans les données.

Les problèmes de ce type peuvent être regroupés en deux catégories : le regroupement (lorsque nous devons trouver les combinaisons essentielles dans les données, comme le regroupement des clients par comportement d'achat) et l'association (lorsque nous devons induire des règles décrivant de grandes parties de nos données, comme les personnes qui achètent A ont également tendance à acheter B) [71].

### **C. Apprentissage semi-supervisé (hybride)**

Ce type d'apprentissage est un peu les deux types mentionnés plus haut, où nous disposons d'une immense quantité de données d'entrée ( $X$ ), mais dont seule une partie est étiquetée. L'obtention de données étiquetées peut être trop coûteuse ou prendre trop de temps, voire les deux, alors que les données non étiquetées sont abordables et faciles à collecter et à sauvegarder, ce qui explique que de nombreux problèmes du monde réel utilisent l'apprentissage hybride [71].

Maintenant que nous avons vu la forme générale de l'apprentissage profond et que nous avons compris comment l'apprentissage se déroule, nous allons passer directement à certaines de ses types de réseaux de neurones et ensuite les composants de ces réseaux.

#### **2.2.6 Les types des réseaux de neurones profonds**

Les architectures ANN sont trop nombreuses, nous n'en mentionnerons donc que quelques-unes qui sont liées à notre travail [72].

## **A. Perceptron (P)**

Le type le plus élémentaire de réseaux neuronaux, et le plus ancien. Capable de résoudre uniquement les problèmes linéairement séparés. Composé d'un nœud unique qui prend les entrées multipliées par leurs poids, ajoute un biais et les fait passer par une fonction d'activation pour obtenir la sortie. Exemple : fonction **ET**.

## **B. Réseau Feed Forward (FF)**

L'une des plus anciennes approches de DL qui remonte aux années 1950, un réseau souvent formé par rétropropagation avec une couche cachée et tous les nœuds entièrement connectés, et l'unité<sup>2</sup> circule de l'entrée à la sortie sans boucles arrière. Exemple : fonction **XOR**.

## **C. Réseau Deep Feed Forward (DFF)**

Une copie de FF avec plus d'une couche cachée qui a déclenché la deuxième vague historique de développement des ANN<sup>3</sup> entre 1980 et 1995 [56]. Il utilise également la rétropropagation pour l'apprentissage, mais en raison de l'empilement élevé de couches denses, les temps d'apprentissage augmentent considérablement. Jusqu'aux années 2000, lorsque la puissance de calcul a augmenté, il est utilisé comme architecture de base pour tous les autres réseaux. Exemple : **Sort of automatic feature engineering**.

---

<sup>2</sup> Une unité fait souvent référence à la fonction d'activation dans une couche par laquelle les entrées sont transformées via une fonction d'activation non linéaire (par exemple par la fonction sigmoïde logistique).

<sup>3</sup> La deuxième vague d'apprentissage profond qui a commencé dans les années 1980 jusqu'en 1995 avec la rétropropagation pour entraîner un réseau avec une ou deux couches cachées. Elle était appelée "conectionnisme" ou "approche conectionniste" [56].

## **D. Réseau neuronal convolutif (CNN)**

Le type de réseaux neuronaux le plus utilisé, généralement pour la reconnaissance d'images et la vision par ordinateur. Il introduit des nouveaux nœuds : kernel, convolution et pool. Le kernel représente les filtres à former pour la reconnaissance, les cellules de convolution effectuent une opération de convolution sur les pixels de l'image et les kernels pour trouver des caractéristiques dans une image et la couche de pool réduit la dimensionnalité de la taille des données. Enfin, il est rattaché à un réseau DFF pour un traitement ultérieur.

Exemple : **Image Recognition.**

## **E. Réseau neuronal récurrent (RNN)**

Précédemment appelés "réseaux de Jordan", ils ressemblent aux réseaux FF normaux mais révèlent un nouveau type de nœuds - des nœuds récurrents dans toutes les couches cachées. Ces nœuds bouclent pour recevoir leur propre sortie avec un retard défini - en plus de la transmettre. Utilisés lorsque le contexte est important, lorsque les décisions des époques précédentes sont nécessaires pour prendre de nouvelles décisions. Exemple : **Natural Language Processing.**

## **F. Mémoire à long et court terme (LSTM)**

Utilisée lorsque les données du problème présentent des décalages ou des écarts de temps qui nécessitent le stockage d'une partie des données. Même principe que les RNN mais avec une nouvelle cellule. Cellules de mémoire capables de mémoriser les données de nombreuses itérations précédentes, de décider de la quantité d'informations à conserver des itérations passées, de réguler la quantité d'informations transmises à la couche suivante et la quantité à détruire. Exemple : **Writing and Speech Recognition.**

### **2.2.7 Les couches des réseaux neurones profond**

Une couche est la composante de base des réseaux neuronaux, composée d'un nombre précis et bien choisi de neurones uniformes ayant la même fonction d'activation. La couche agit comme un conteneur qui transforme l'entrée reçue à l'aide de ces fonctions, généralement non linéaires, et la transmet à la couche suivante [73].

La première couche d'un réseau est la couche d'entrée, la dernière est la couche de sortie et les couches intermédiaires sont appelées couches cachées.

#### **Types de couches fréquemment utilisées**

- **Dense** : Une couche qui connecte chacun de ses neurones avec tous les neurones de la couche précédente. La transformation qu'elle effectue est simple, il suffit de calculer la sortie de la fonction d'activation pour l'entrée afin de la transmettre à la couche suivante.
- **Dropout** : Une couche qui met aléatoirement la fonction d'activation de l'entrée à 0 avec une probabilité d'acceptation donnée. Utilisé pour traiter la tendance du réseau à sur-apprendre (overfitting).
- **Flatten** : Convertit l'entrée (la sortie des couches précédentes) en un vecteur 1D que nous pouvons classer, sans affecter la taille du lot.

### **2.3 Réseaux de Neurones Convolutifs (ConvNets/CNNs)**

Un réseau neuronal convolutif (CNN) est un algorithme d'apprentissage profond qui peut prendre une image d'entrée, attribuer de l'importance (poids et biais apprenables) à divers aspects / objets de l'image et être capable de différencier l'un de l'autre. Ces dernières années, les CNNs sont devenus un élément central de toutes les méthodes de vision par ordinateur.

En 1980, Kunihiko Fukushima a proposé un réseau neuronal de reconnaissance d'image très basique dit le Néocognitron. [74]

Les ConvNets ont été introduits pour la première fois dans les années 1980 par Yann LeCun, chercheur postdoctoral en informatique. LeCun s'était appuyé sur le travail de Kunihiko Fukushima.

La première version de l'architecture ConvNet [75], appelée LeNet améliorée par LeCuN et al. [76], pouvait reconnaître les chiffres manuscrits. Les CNN ont trouvé un marché de niche dans les services bancaires, postaux et bancaires, où ils lisent les codes postaux sur les enveloppes et les chiffres sur les chèques.

### **2.3.1 Le principe de Réseaux de neurones convolutifs**

Le réseau de neurone convolutif (CNN) est un sous-type de réseaux de neurones artificiels (ANN).

L'innovation des réseaux de neurones convolutifs est la possibilité d'apprendre automatiquement un grand nombre de filtres en parallèle spécifiques à un jeu de données d'apprentissage sous les contraintes d'un problème de modélisation prédictive spécifique, comme la classification d'images.

Ils sont constitués de neurones avec des poids et des biais apprenables. Chaque neurone spécifique reçoit de nombreuses entrées et prend ensuite une somme pondérée sur eux, où il le fait passer par une fonction d'activation et répond en retour avec une sortie.

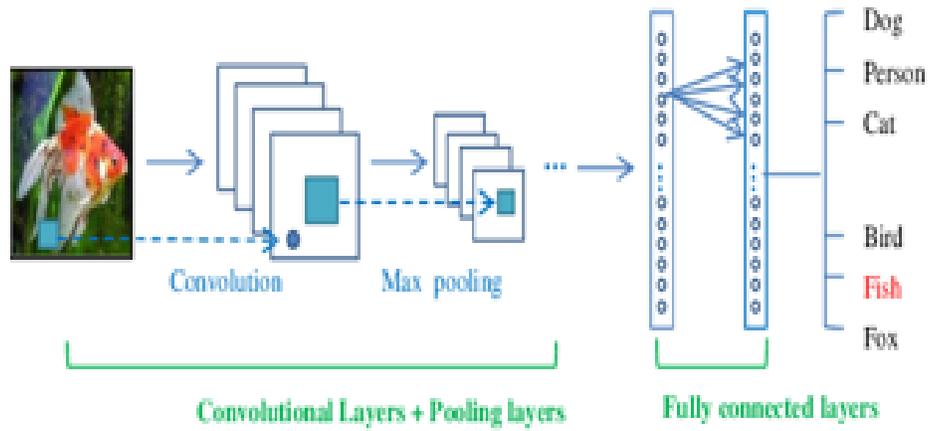


Figure 2-6 : Schéma des réseaux de neurones convolutifs [77].

### 2.3.2 Types de couches dans le réseau neuronal convolutif

L'architecture CNN comprend plusieurs types de couches. Une architecture typique consiste en des répétitions d'un empilement de plusieurs couches de convolution et d'une couche de regroupement, suivies d'une ou plusieurs couches entièrement connectées. Bien que les opérations de convolution et de regroupement décrites dans cette section soient pour 2D-CNN, en particulier, contrairement au réseau neuronal régulier, les couches d'un ConvNet ont des neurones disposés en 3 dimensions : largeur, hauteur, profondeur.

Comme nous l'avons décrit ci-dessus, un simple ConvNet est une séquence de couches, et chaque couche d'un ConvNet transforme un volume d'activations en un autre grâce à une fonction différentiable. Nous utilisons trois principaux types de couches pour créer des architectures ConvNet : la couche convolutive (Convolutional layer), la couche de regroupement (Pooling layer) et la couche entièrement connectée (Fully-connected layer) (exactement comme dans les réseaux de neurones classiques). Nous empilerons ces couches pour former une architecture ConvNet complète.

Les types de couches les plus utilisés pour créer des CNN sont : CONV – ReLU – POOL – FC – BN – DO.

## **A. Couche convolutive (*Convolutional layer CONV*)**

La couche de convolution a des noyaux (filtres) et chaque noyau a une largeur, une profondeur et une hauteur. Cette couche produit les cartes de caractéristiques à la suite du calcul du produit scalaire entre les noyaux et les régions locales de l'image.

Leur travail consiste à extraire les informations pertinentes de l'image (caractéristiques) grâce à une opération de convolution. Cette opération fait glisser un ensemble de filtres sur une image. Le poids de ces filtres est mis à jour lors de l'apprentissage et c'est grâce à eux que le réseau réussit à reconnaître les images par la suite.

## **B. Couche d'unité linéaire rectifiée (*Rectified Linear Unit layer ReLU*)**

La couche d'unité linéaire rectifiée (en anglais rectified linear unit layer) (ReLU) est une fonction d'activation qui est utilisée sur tous les éléments du volume permet d'éliminer toutes les valeurs négatives et conserver les valeurs positives. Elle a pour but d'introduire des complexités non-linéaires au réseau.

## **C. Couche de Pooling**

L'étape de pooling est une technique de sous-échantillonnage. Généralement, une couche de pooling est insérée régulièrement entre les couches de correction et de convolution. En réduisant la taille des cartes de caractéristiques, donc le nombre de paramètres du réseau, cela accélère le temps de calcul et diminue le risque de sur-apprentissage.

Il y a de nombreuses opérations dans cette couche, par exemple Max pooling et Average pooling.

- **Mise en commun maximale (*Max-pooling*)** : Il s'agit d'une opération de pooling qui ne prélèvera que le maximum d'un pool. Cela se fait en fait avec l'utilisation de filtres glissant à travers l'entrée ; et

à chaque foulée, le paramètre maximum est retiré et le reste est abandonné. Cela sous-échantillonne en fait le réseau.

- **Mise en commun moyenne (Average pooling)** : Est une opération de regroupement qui calcule la valeur moyenne d'un pool et l'utilise pour créer un sous-échantillonnage (regroupé), elle extrait les fonctionnalités plus facilement que Max Pooling.

La figure 2.8 montre un exemple d'opération de max pooling et Average pooling avec une taille de filtre 2x2 pixels à partir d'une entrée de pixels 4x4.

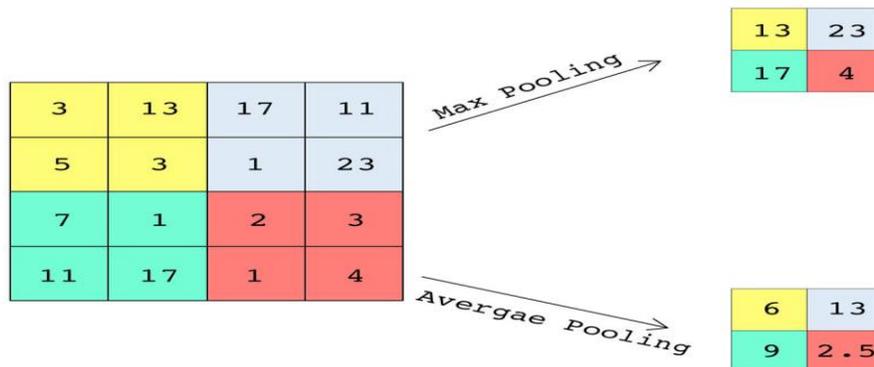


Figure 2-7 : Exemple de fonctionnement de Max pooling et Average pooling [78].

#### D. Couche entièrement connectée (Fully-connected layer FC)

Cette couche est à la fin du réseau. Elle permet la classification de l'image à partir des caractéristiques extraites par la succession de bloc de traitement. Elle est entièrement connectée, car toutes les entrées de la couche sont connectées aux neurones de sorties de celle-ci. Ils ont accès à la totalité des informations d'entrée. Chaque neurone attribue à l'image une valeur de probabilité d'appartenance à la classe  $i$  parmi les  $C$  classes possibles. Chaque probabilité est calculée à l'aide de la fonction « softmax » dans le cas où les classes sont exclusivement mutuelles.

### **E. Couche de normalisation par lots (*Batch normalization layer BN*)**

La normalisation par lots est une technique d'apprentissage de réseaux de neurones profonds pour améliorer la vitesse, la performance et la stabilité des réseaux de neurones artificiels. Elle est utilisée pour normaliser le volume d'entrée avant de le transmettre à la couche suivante du réseau.

### **F. Couche d'abandon (*Dropout layer DO*)**

Est une méthode pour réduire le surajustement et améliorer l'erreur de généralisation dans les réseaux neuraux profonds dont il déconnecte aléatoirement les entrées de la couche précédente avec une probabilité  $p$ , ce qui permet d'éviter le sur-apprentissage.

En particulier, les couches CONV, FC et BN effectuent des transformations qui sont fonction non seulement des activations dans le volume d'entrée, mais aussi des paramètres (les poids et biais des neurones), d'autre part, Les couches RELU/POOL implémenteront une fonction fixe.

Notez que certaines couches contiennent des paramètres et d'autres non.

### **2.3.3 Les Architectures de CNN**

De nos jours, les CNN sont considérés comme les algorithmes les plus largement utilisés parmi les inspirés des techniques d'Intelligence Artificielle (IA). L'histoire de CNN commence par le neurobiologique expériences menées par Hubel et Wiesel (1959, 1962) [79]. Leur travail a fourni une plate-forme pour de nombreux modèles cognitifs, et CNN a remplacé presque tous ceux-ci. Au fil des décennies, différents efforts ont été menés pour améliorer les performances des CNN.

Il existe de nombreuses architectures CNN réputées. Les architectures CNN les plus populaires sont données dans la figure 2.9.

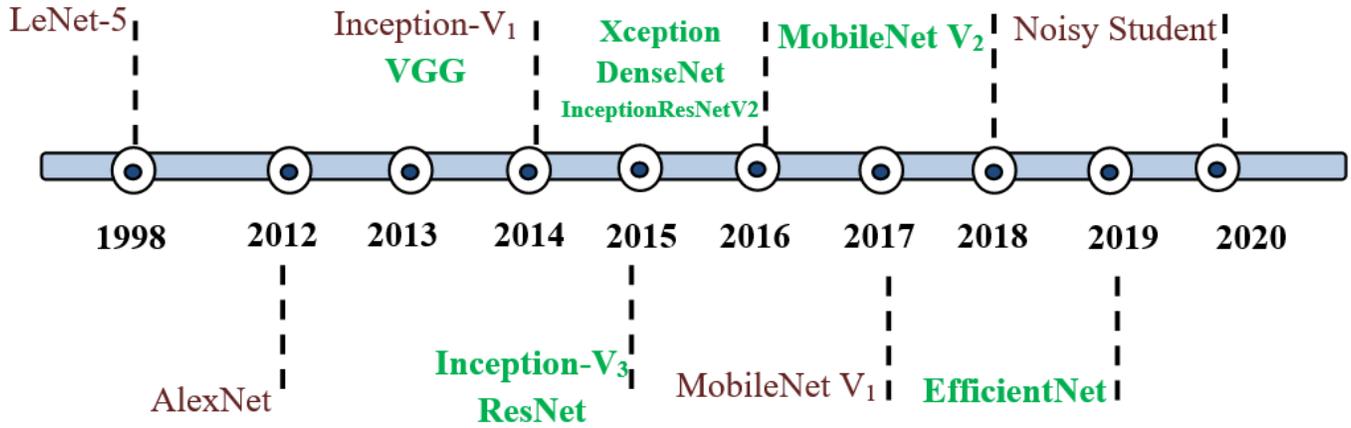


Figure 2-8 : Histoire évolutive des CNNs montrant les innovations architecturales [65].

### A. LeNet-5

LeNet était le réseau neuronal convolutif le plus archétype développé par Yann LeCun en 1990 [80] et amélioré plus tard en 1998 [81]. L'architecture LeNet la plus efficace et la plus connue et elle a été utilisée pour lire les codes postaux, les chiffres, etc.

Cette architecture contient 4 couches convolutive (CONV) et mise en commun (Pooling) alternées, suivies de 3 couches entièrement connectées (fully-connected). LeNet était la première architecture CNN, qui non seulement réduit le nombre de paramètres mais a pu apprendre les caractéristiques de pixels bruts automatiquement.

### B. AlexNet

La première architecture CNN célèbre est AlexNet, qui popularise le réseau de neurones convolutifs en vision par ordinateur (computer vision), développé par Alex Krizhevsky, Ilya Sutskever et Geoff Hinton [82]. Plus tard, en 2012, AlexNet a été présenté au défi ImageNet ILSVRC et il a considérablement dépassé les performances du deuxième finaliste.

AlexNet contient 5 couches convolutives avec des unités linéaires rectifiées (ReLU) comme fonctions d'activation, 3 couches Max Pooling et 3 couches entièrement connectées (FC).

### **C. VGG**

L'utilisation réussie des CNN dans les tâches de reconnaissance d'image a accéléré la recherche en conception architecturale. À cet égard, Simonyan et al ont proposé un principe de conception simple et efficace pour les architectures CNN. Leur architecture, nommé comme le Groupe visuel Géométrie (VGG) de l'université d'Oxford [83], Sa principale réalisation a été de remplacer les filtres 11x11 et 5x5 avec une pile de 3x3 couche de filtres. L'utilisation de filtres de petite taille offre un avantage supplémentaire de faible complexité de calcul en réduisant le nombre de paramètres.

### **D. GoogleNet**

GoogleNet a été le gagnant du concours 2014-ILSVRC, connu également comme Inception-V1, Il a été développé par une équipe de Google (Christian Szegedy et al). [84]

C'est un type de réseau de neurones convolutif basé des modules Inception, ces blocs encapsulent des filtres de différentes tailles (1x1, 3x3 et 5x5) pour capturer des informations spatiales à différentes échelles, suivi du filtre Concat qui permet de concaténer les résultats des filtres. En outre, la densité de la connexion a été réduite en utilisant la mise en commun moyenne globale à la dernière couche, au lieu d'utiliser une couche entièrement connectée. Ces réglages de paramètres ont provoqué une diminution significative du nombre de paramètres de 60 millions à 4 millions de paramètres.

## **E. Inception-v3**

Inception-V3 est une version améliorée d'Inception-V1 et V2 proposée par Christian Szegedy et al. [85]. L'idée d'Inception-V3 était de réduire le coût de calcul des réseaux profonds (Deep networks) sans affecter la généralisation. À cette fin, Szegedy et al. Remplacement des filtres de grande taille (5x5 et 7x7) par des filtres petits et asymétriques (1x7 et 1x5) et la convolution  $5 \times 5$  est transformée en deux opérations de convolution  $3 \times 3$ .

## **F. ResNet**

Kaiming He et al. [86] ont développé un réseau résiduel (ResNet). Cette architecture CNN présente des connexions de saut uniques et une utilisation essentielle de la normalisation par lots (Batch Normalization). Encore une fois, l'architecture n'a pas de couches entièrement connectées à la fin du réseau. Le principal inconvénient de ce réseau est qu'il est très coûteux à évaluer en raison de la vaste gamme de paramètres. Cependant, jusqu'à présent, ResNet est considéré comme un modèle de réseau neuronal convolutif à la pointe de la technologie et constitue l'option par défaut pour l'utilisation des ConvNets dans la pratique. Il avait été le gagnant de l'ILSVRC 2015.

## **G. FaceNet**

FaceNet est un système de reconnaissance faciale développé en 2015 par Florian Schroff et al. chez Google [87].

C'est un système qui, étant donné une image d'un visage, extraira des caractéristiques de haute qualité du visage et prédira une représentation vectorielle de 128 éléments de ces caractéristiques, appelée incrustation de visage, qui peut ensuite être utilisée pour former un système d'identification de visage.

Le système FaceNet peut être largement utilisé grâce à de multiples implémentations open source tierces du modèle et à la disponibilité de modèles préformés.

## **H. Xception**

François Chollet de Google [88] a proposé l'architecture extrême Inception nommée Xception, où les modules Inception ont été remplacés par des modules modifiés "depthwise separable convolutions." qui ont été proposé par Laurent Siphre en 2014 [89].

## **I. DenseNet**

Le réseau convolutive dense (DenseNet) introduit par Huang et al. [90], est une architecture de réseau où chaque couche est directement connectée à toutes les autres couches de manière anticipée (dans chaque bloc dense). Ce type de connexion est appelé connectivité dense. Pour chaque couche, les cartes d'entités de toutes les couches précédentes sont traitées comme des entrées distinctes tandis que ses propres cartes d'entités sont transmises en tant qu'entrées à toutes les couches suivantes.

## **J. MobileNet-v1**

MobileNet est une architecture légère développée par Howard et al. [91] de Google, le modèle MobileNet est conçu pour être utilisé dans des applications mobiles, cette architecture utilise des convolutions séparables en profondeur. Il réduit considérablement le nombre de paramètres par rapport au réseau avec des convolutions régulières avec la même profondeur dans les filets. Il en résulte des réseaux de neurones profonds légers. Howard et al introduisent deux hyperparamètres globaux simples qui font un compromis efficace entre latence et précision. Ces hyperparamètres permettent au constructeur de modèles de choisir le modèle de la bonne taille pour leur application sur les contraintes du problème.

Ces paramètres sont le multiplicateur de largeur «  $\alpha$  » et le multiplicateur de résolution «  $\rho$  ».

## **K. MobileNet-v2**

La deuxième version de MobileNet proposée par Sandler et al [92]. Il est basé sur une structure résiduelle inversée où les connexions résiduelles sont entre les couches de goulot d'étranglement (bottleneck layer). La couche d'expansion intermédiaire utilise des convolutions légères en profondeur pour filtrer les entités en tant que source de non-linéarité.

## **L. EfficientNet**

Les réseaux de neurones convolutifs (ConvNets) sont couramment développés avec un budget de ressources fixe, puis mis à l'échelle pour une meilleure précision si plus de ressources sont disponibles.

Tan et Le ont travaillé pour concevoir un nouveau réseau de base et le mettre à l'échelle pour obtenir une famille de modèles [93] En 2019, appelé EfficientNet.

Tan et Le proposent une nouvelle méthode de mise à l'échelle composée (EfficientNet), qui utilise un composé  $\phi$  pour mettre à l'échelle uniformément la profondeur, la largeur et la résolution du réseau (profondeur :  $d = \alpha^\phi$  largeur :  $w = \beta^\phi$ , résolution :  $r = \gamma^\phi$ ).  $\alpha$ ,  $\beta$  et  $\gamma$  sont des constantes déterminées par une petite recherche de grille avec la contrainte que :  $\alpha \beta^2 \cdot \gamma^2 \approx 2$  et  $\alpha \geq 1, \beta \geq 1, \gamma \geq 1$ .

## **M. Noisy Student**

Noisy Student a été introduit par l'équipe de recherche de Google (Xie et al) en 2020. [94] Noisy Student Training est une approche d'apprentissage semi-supervisée. Il étend l'idée d'auto-formation et de distillation avec l'utilisation de modèles d'élèves égaux ou plus grands et de bruit ajouté à l'élève pendant l'apprentissage. Noisy Student Training cherche à améliorer l'auto-formation et la distillation de deux manières. Premièrement,

cela rend l'élève plus grand ou au moins égal à l'enseignant afin que l'élève puisse mieux apprendre à partir d'un plus grand ensemble de données. Deuxièmement, cela ajoute du bruit à l'élève, de sorte que l'élève bruyant est obligé d'apprendre plus durement à partir des pseudo étiquettes. Pour brouter l'élève, il utilise le bruit d'entrée tel que l'augmentation des données, et le bruit du modèle tel que la profondeur stochastique pendant l'entraînement.

## **2.4 Applications de l'apprentissage profond**

Depuis le début de l'ère de l'apprentissage profond en 2010, l'apprentissage profond a été étudié, développé et utilisé pour accomplir plusieurs tâches. De nombreuses grandes entreprises technologiques telles que Facebook, Google, Microsoft, Amazon, Apple et bien d'autres sont entrées dans la compétition pour devenir des entreprises axées sur l'IA, notamment en utilisant l'apprentissage profond, qui était - et est toujours - le sommet de l'intelligence artificielle. Depuis lors, il a été largement utilisé pour résoudre des problèmes du monde réel dans divers domaines allant de la classification d'images, la reconnaissance de l'écriture manuscrite et de la parole, la finance et les prédictions de faillite, l'industrie des jeux vidéo, la robotique, le diagnostic médical et des maladies, jusqu'à la vision par ordinateur et les véhicules à conduite autonome. Cette section mentionne certaines des applications les plus demandées.

### **2.4.1 Reconnaissance de caractères**

Trouver des mots-clés dans des morceaux de textes peut être difficile pour les yeux humains, surtout pour ceux qui ont une faible capacité visuelle. L'apparition des bureaux sans papier a rendu nécessaire l'analyse automatisée des documents. Ces deux tâches, ainsi que d'autres qui exigent que les machines reconnaissent les caractères, peuvent être facilement réalisées grâce à la "*Optical Character Recognition (OCR)*", qui constitue un élément de base. La technique de pointe utilisée par les nombreux chercheurs qui tentent de construire des systèmes OCR

de haute précision avec un taux d'erreur minimal pour différentes langues et différents types de polices (numériques ou manuscrites, séparées ou attachées) est l'apprentissage profond pour sa supériorité en matière de reconnaissance d'images [95].

### **2.4.2 Reconnaissance vocale**

Tandis que l'apprentissage profond peut être bon dans la reconnaissance d'image comme mentionné précédemment, il peut également donner une performance remarquable dans la reconnaissance de la parole pour sa grande capacité de prédire les phrases et leur signification à partir du contexte de la conversation et aussi la flexibilité que l'apprentissage profond fournit pour inclure toutes sortes de langues ainsi que les différents accents [96].

### **2.4.3 Les jeux vidéo**

L'IA a été introduite pour la première fois dans les jeux vidéo sur la console Atari 2600, dans le jeu "Computer Space", en 1970 [97]. Depuis lors, l'émergence de l'apprentissage profond dans cette industrie a permis aux développeurs de rendre les ennemis générés par l'ordinateur plus humains en termes de comportement et de prise de décisions par eux-mêmes, rendant les jeux plus difficiles et plus exigeants. Une étude de J. Hruska [98] montre que les joueurs apprécient toujours un bon système d'IA. Il est beaucoup plus efficace de tester une voiture dans l'univers d'un jeu numérique plutôt que de risquer d'endommager le matériel coûteux ou même de blesser des humains lors de tests réels.

### **2.4.4 Diagnostic de la maladie**

L'une des principales causes de décès chez l'homme est due à un certain nombre de maladies graves allant de toutes sortes de cancers aux maladies neuronales et respiratoires et bien d'autres encore. Une solution pourrait être un diagnostic précoce dans des hôpitaux hautement sophistiqués et équipés qui peuvent être rarement

disponibles ou à un coût élevé. D'un autre côté, le diagnostic des maladies par classification, dans lequel les réseaux neuronaux profonds sont excellents, peut fournir une solution moins chère, plus simple, sans risque et non invasive pour un tel problème, uniquement en employant une structure de réseau et des méthodes d'apprentissage appropriées aux ensembles de données médicales qui alimentent les classificateurs intelligents basés sur les DNN pour générer des décisions de diagnostic [99].

#### **2.4.5 Les voitures à conduite autonome**

Le rêve de produire des voitures entièrement autonomes ou "sans conducteur" n'est plus un rêve car de nombreux géants de l'industrie tels que Tesla, Mercedes, BMW et bien d'autres s'efforcent de réaliser cet objectif. Une technologie à fort potentiel pour mener le développement est l'apprentissage profond qui a pris le contrôle des principaux sous-domaines de la conduite autonome qui sont représentés dans "perception, localisation, planification et contrôle" [99].

### **2.5 Conclusion**

Dans ce chapitre nous avons introduit les concepts de base de l'intelligence artificielles (IA). Et nous avons ensuite détaillé l'apprentissage profond qui représente l'un des concepts le plus important dans IA.

Dans la section de l'apprentissage profond nous avons donné un aperçu sur les réseaux de neurones artificiels, et le passage de ces réseaux aux réseaux neuronaux profonds. Nous avons aussi décrit les types des réseaux neuronaux profonds et à la fin nous avons présenté en détail les réseaux de neurones convolutifs qu'on va les utiliser dans notre méthode présentée dans le chapitre suivant.

## **Chapitre 3**

# **La méthode proposée pour la détection de la fatigue du conducteur**

### **3.1. Introduction**

Dans le premier chapitre nous avons présenté les trois indicateurs utilisés pour la détection de la fatigue du conducteur : indicateur physiologique, analyse faciale et analyse de la conduite.

Notre approche proposée utilise l'analyse faciale du conducteur, basé sur la détection des clignements des yeux et les bâillements.

Dans ce chapitre, on va présenter l'approche proposé pour la détection de la fatigue du conducteur basé sur les méthodes de l'intelligence artificielle. Cette approche est composée en 3 étapes chacune sera détaillée séparément.

### **3.2. Mise en place de l'environnement de programmation**

Dans cette section on va présenter les outils et les bibliothèques utilisés dans le développement de notre système.

**Visual Studio Code** : Visual Studio Code (*VSCode*) est un éditeur de code source léger et puissant développé par Microsoft. Il est doté d'une interface interactive très agréable et possède plusieurs fonctions d'aide et d'extensions. Il supporte un large éventail de langages de programmation, y compris celui que nous allons utiliser, Python.

**Python** : Python est un langage de programmation de haut niveau, interprété, à objets orientés, utilisé pour un très large ensemble d'applications et de domaines tels que le développement de jeux, la création d'applications de bureau ainsi que le développement d'applications. Il est très convivial pour les débutants et facile à apprendre et à utiliser. Nous avons utilisé la version 3.9.5.

**OpenCV** : Une bibliothèque de vision par ordinateur à code source ouvert qui a été créé pour regrouper les applications de vision par ordinateur. Elle dispose de tonnes d'algorithmes optimisés (plus de 2500) à cet égard. Elle supporte une variété de systèmes d'exploitation, de langages de programmation, de matériels et de logiciels.

**Dlib** : une boîte à outils moderne en C++ (qui a un très bon support pour python, le langage que nous avons utilisé) contenant des algorithmes d'apprentissage automatique et des outils pour plusieurs applications allant des dispositifs embarqués jusqu'aux environnements de calcul à haute performance.

**TensorFlow** : est une bibliothèque logicielle de bout en bout, gratuite et à code source ouvert développé par Google, utilisée pour construire et déployer des modèles d'apprentissage machine, aussi bien pour les débutants que pour les professionnels.

**Keras** : est une API d'apprentissage profond de haut niveau, simple, flexible et puissante, dont le backend est TensorFlow. Elle vise à faciliter et à minimiser les actions requises pour résoudre les problèmes, ce qui en fait le Framework de Deep Learning le plus utilisé.

L'entraînement des modèles d'apprentissage profond avec des millions de neurones sur un ordinateur ordinaire prendrait beaucoup de temps et nous avons 10 architectures à tester, donc cela prendrait un temps énorme, sans compter les essais, il fallait alors trouver un moyen de le faire. « Google Colaboratory » était ce que nous recherchions.

**Google Colaboratory** : ou « Colab » est un service cloud qui donne un accès à des ressources computationnelles, notamment des GPU, ce qui le rend très adapté aux applications d'apprentissage profond comme les nôtres, permettant à quiconque d'écrire et d'exécuter du code python directement sur le navigateur. Il est fourni gratuitement par Google.

La figure ci-dessous montre les logos de ces logiciels, bibliothèques et APIs.



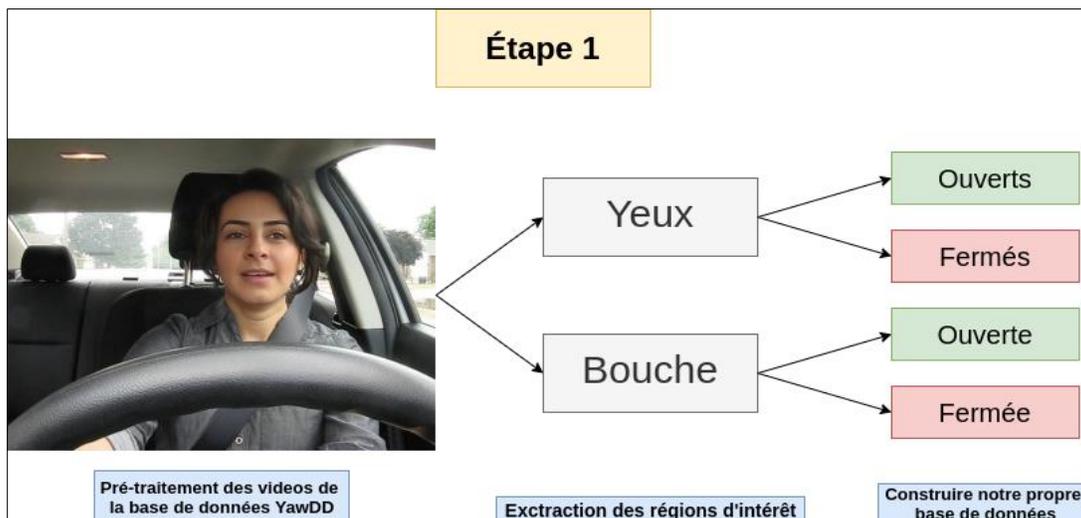
**Figure 3-1** : Logos des outils utilisés dans cette étude.

### 3.3. L'approche Proposée

L'approche proposée reçoit la vidéo du conducteur, par la suite elle identifie l'état du conducteur (fatigué ou actif). Elle est composée de 3 étapes, dans la première étape on prépare les ensembles d'apprentissages et des

tests pour la classification de l'état des yeux et la bouche (ouvert ou fermé). Dans la deuxième étape, on teste 10 architectures CNNs et on sélectionne la meilleure architecture. Dans la dernière étape, on détecte l'état du conducteur (fatigué ou actif) en utilisant le CNN pour la classification de l'état des yeux et la bouche, après on calcule deux paramètres qui indiquent l'état du conducteur. Ces paramètres sont le pourcentage de fermeture des yeux (PERcentage of eye CLOSure: PERCLOS) et la fréquence de fermeture de la bouche (Frequency Of Mouth: FOM).

- **1<sup>ère</sup> Étape :** Dans la première étape on détecte le visage dans chaque image. Après, on localise les yeux (gauche et droit) et la bouche. Par la suite, on distingue leur état avec le calcul des deux rapport hauteur/largeur de l'œil (*Eye Aspect Ratio : EAR*) et de la bouche (*Mouth Aspect Ratio : MAR*) puis nous stockons ces images sous forme de données étiquetées, comme montre la figure 3-2 :



**Figure 3-2 :** La première étape de la méthode, génération de la base de données.

- **2<sup>ème</sup> Étape :** Dans cette étape, nous utilisons les ensembles de données générés précédemment pour tester les différentes architectures de CNN mentionnées au chapitre 2. L'architecture donnant les meilleurs résultats sera utilisée dans la suite du développement, illustrée dans la figure 3-3.

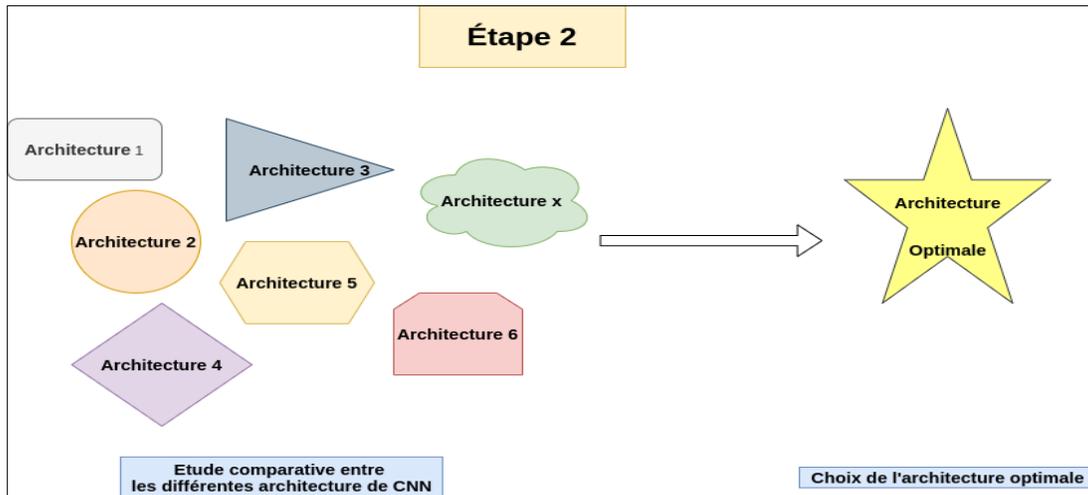


Figure 3-3 : La deuxième étape de la méthode, étude comparative et choix de l’architecture optimale.

- **3<sup>ème</sup>Étape** : l’architecture choisi sera utilisée pour classifier les images puis nous calculons PERCLOS et FOM afin de déterminer l’état général du conducteur (actif, fatigué).

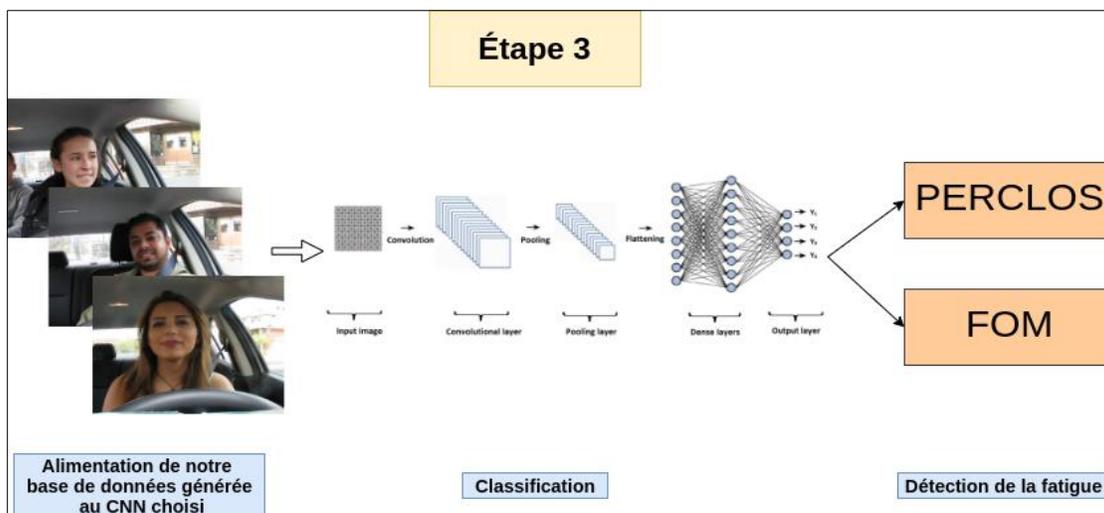


Figure 3-4 : La troisième étape de la méthode, test de l’architecture et prédiction de la fatigue.

### 3.3.1 Étape 1 : Préparation des données

#### A. La base de données utilisée

La base de données utilisée dans cette expérimentation est la base de détection de bâillement (*Yawning Detection Dataset: YawDD* ). Elle se compose d’une séquence des vidéos acquise à partir d’une caméra Canon avec une résolution de 640\*480 pixels à 30 images par seconde (30 FPS) au format AVI sans audio [100].

**YawDD** : est un ensemble des données qui contient deux sous-ensembles de données vidéo de conducteurs présentant diverses caractéristiques faciales, elle est utilisée pour tester des algorithmes et des modèles destinés principalement à la détection des bâillements, mais également à la reconnaissance et au suivi du visage. Les vidéos sont prises dans des conditions d'illumination réelles et variables.

Deux emplacements différents pour l'installation de la caméra ont été pris en compte dans la collection de base de données. Dans le premier scénario, la caméra était installée sous le rétroviseur avant et dans le deuxième scénario, la caméra était installée sur le tableau de bord du conducteur.

Les participants ont été invités à s'asseoir dans le siège du conducteur et à attacher leur ceinture de sécurité pour rendre le scénario plus réaliste.

Chaque participant a trois / quatre vidéos. Chaque vidéo a différentes conditions buccales telles que la parole normale, le chant et les étirements. Dans cet ensemble de données, il y a 322 vidéos de différents groupes ethniques de conducteurs masculins et féminins avec ou sans lunettes et lunettes de soleil [101].

Quelques exemples de données dans YawDD sont présentés dans la figure 3.5 :



**Figure 3-5** : Exemples de données dans YawDD.

## B. Le pré-traitement

Dans le pré-traitement, nous avons divisée les vidéos en séquences d'images. Par la suite, ces images ont été redimensionnées de la résolution de 680\*480 pixels à 320\*240 pixels et converties de 3 canaux de couleurs (Rouge, Vert, Bleu - RGB) en un niveau de gris, afin de réduire le nombre de pixels et donc les calculs du réseau neuronal, ces opérations ont été faites à l'aide de la bibliothèque **OpenCV**.

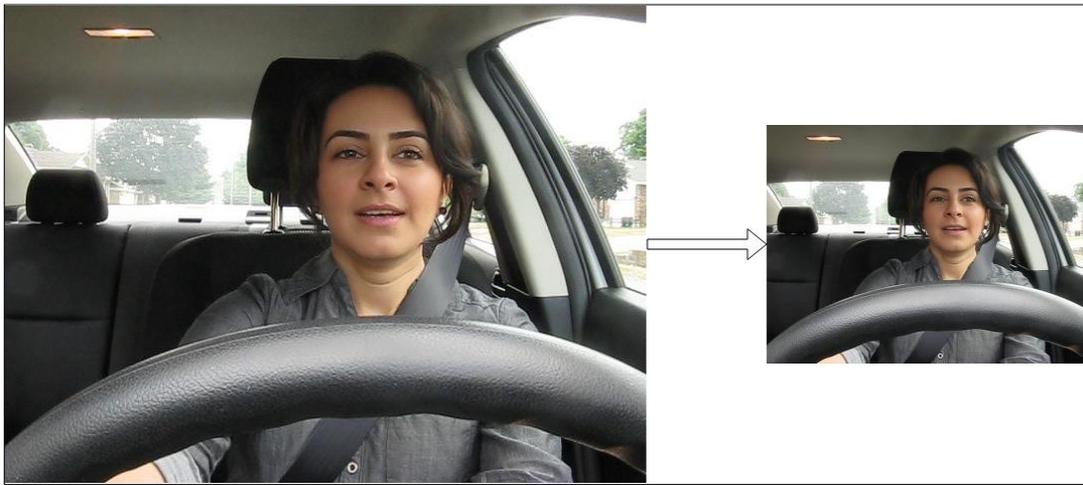


Figure 3-6 : Redimensionnement de l'image.



Figure 3-7 : Mise à l'échelle de l'image en niveaux de gris.

### C. Détection des yeux et la bouche

Tout d'abord, la zone de visage dans chaque image est détectée avec la fonction de la bibliothèque **Dlib** "visage frontal" (*dlib.get\_frontal\_face\_detector*). A partir les points de visage obtenu on prédire alors les régions d'intérêts (*Region Of Interest : ROI*) des yeux et de la bouche avec la fonction de prédiction (*dlib.shape\_predictor*).

Le visage est décrit par 68 points de repère en coordonnées x-y, comme le montre la figure ci-dessous :

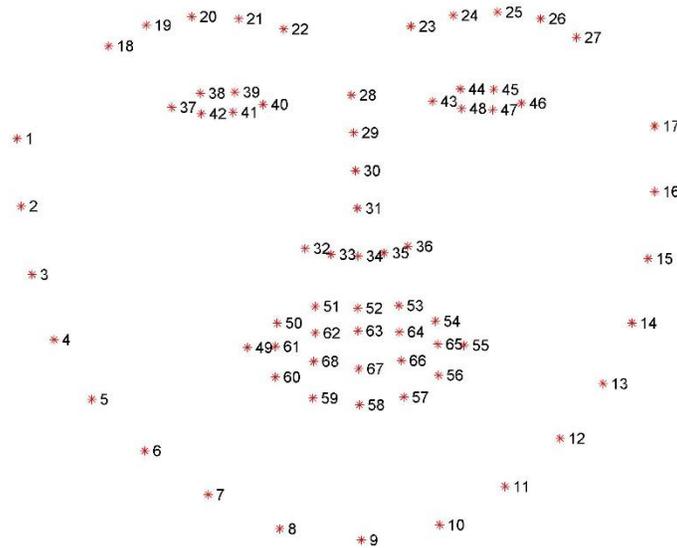


Figure 3-8 : Les 68 point de repères du visage défini dans la bibliothèque Dlib.



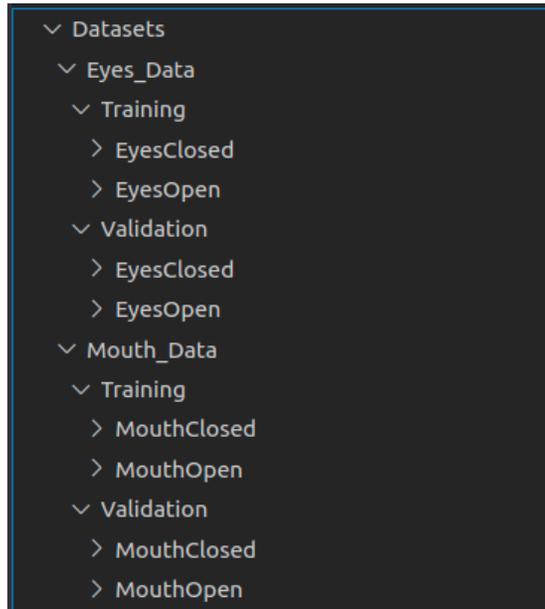
Figure 3-9 : Les régions d'intérêt prédit par la fonction de prédiction de Dlib

## **D. Génération de notre base de données**

L'étape la plus importante dans l'apprentissage des réseaux de neurone profond est de bien préparer l'ensemble d'apprentissage et de validation.

Après avoir traité les séquences des images de chaque vidéo, nous les avons collectés pour générer la base. Vu que les fonctions de la bibliothèque Dlib ne détecte pas le visage des personnes qui ne sont pas devant la caméra et aussi ne détecte pas les yeux quand le conducteur porte des lunettes. Nous avons utilisé uniquement les vidéos où la caméra est placée sur le tableau de bord et à partir de celles-ci, nous avons utilisé que les vidéos des personnes qui ne portent pas de lunettes. En Total 14 vidéos de YawDD ont été utilisées (8 de femmes et 6 d'hommes conducteurs).

Les images sont classées dans des dossiers « état ouvert ou fermé » en fonction de l'état des yeux et de la bouche. Au début, nous avons construit un ensemble de données mixte entre les yeux et la bouche, mais après plusieurs tests et expérimentations, nous avons constaté que pour obtenir des bons résultats, il été préférable de les séparer et d'entraîner deux modèles. Notre base de données est donc composée de deux dossiers principaux, Yeux et Bouche, à l'intérieur de chaque dossier il y a deux dossiers, un pour l'entraînement et l'autre pour la validation du modèle (pour les données de teste nous avons utilisé celles de la validation). 9 vidéos ont été choisies pour le dossier d'entraînement et les 5 autres pour la validation. La classification des images dans cette étape est faite en calculant le rapport d'aspect des yeux (EAR) et de bouche (MAR), suivant le seuil obtenu l'image est enregistrée dans le dossier approprié.



**Figure 3-10** : Arbre des dossiers des ensembles de données générés.

- **Rapport d'aspect des yeux (EAR) et de bouche (MAR)**

Le rapport d'aspect est un attribut de projection d'image qui décrit la relation proportionnelle entre la largeur et la hauteur d'une image, dans le cas d'œil. Ce rapport est généralement constant lorsque l'œil est ouvert et commence à tendre vers zéro lors de la fermeture de l'œil. Étant donné que le clignement des yeux est effectué par les deux yeux de manière synchrone.

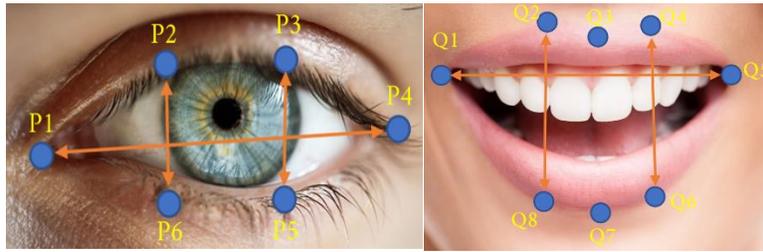
Le rapport d'aspect des yeux est calculé à l'aide de l'équation :

$$EAR = \frac{\|p2-p6\| + \|p3-p5\|}{2 \times \|p1-p4\|} \quad (3.1)$$

De même, pour déterminer le paramètre de bâillement, le rapport d'aspect de la bouche (MAR) est calculé en utilisant

$$MAR = \frac{\|q2-q8\| + \|q4-q6\|}{2 \times \|q1-q5\|} \quad (3.2)$$

Où  $p_1, p_2, \dots, p_6$  sont les 6 repères des yeux et  $q_1, q_2, \dots, q_8$  sont les repères de bouche, les deux repères sont illustrés dans **la figure 3.11**



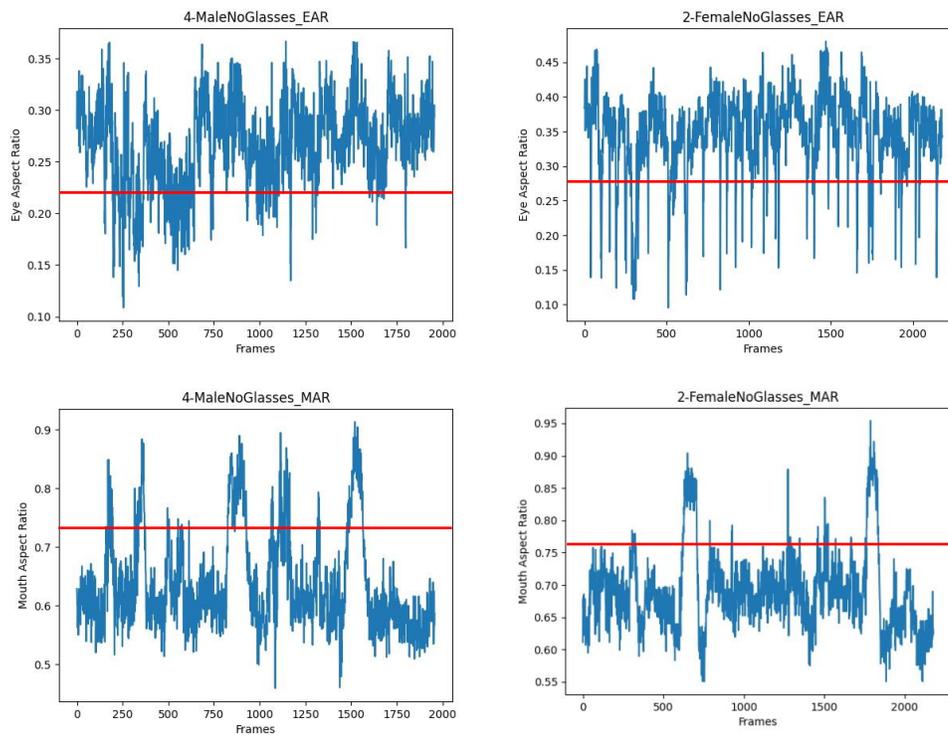
**Figure 3-11** : Le repère facial de la région des yeux.

- **Sélection du seuil EAR / MAR**

Pour classer les images dans leur dossier, nous avons besoin d'un seuil de EAR et MAR. Dans un premier temps, nous avons utilisé un nombre direct trouvé dans la littérature 0,3 pour EAR et 0,64 pour MAR. En observant les images générées dans chaque classe, nous avons conclu que ces paramètres ne sont pas valides et donnent une erreur de classification élevée. Les premières tentatives de changement ont consisté à jouer sur ces nombres en les augmentant ou en les diminuant un peu, mais toutes les tentatives ont abouti aux mêmes résultats.

Comme chaque personne a une taille de yeux et de bouche différente (certaines sont plus grandes que d'autres), fixer un certain seuil pour toutes les personnes ne serait pas un choix équitable pour fournir des résultats valides. La solution proposée est de calculer le rapport EAR, MAR dans toutes les images de chaque vidéo (personne) et sélectionner le rapport maximum atteint. Le seuil pour EAR est alors fixé au 60% du maximum et pour MAR de 80% du maximum. Les 60% et 80% ont été choisis en comparant les valeurs EAR, MAR dans chaque vidéo (représentées dans les figures ci-dessous).

(Pour MAR, nous avons pris un seuil élevé de 80% pour que l'état de parole ne soit pas classé comme une bouche ouverte).



**Figure 3-12** : Les valeurs de EAR et MAR obtenu à partir des vidéos d' un homme et une femme.

Sur la figure ci-dessus, nous pouvons voir que tout ce qui est en dessous de la ligne rouge, qui représente le seuil, est considéré comme état fermé et tout ce qui est au-dessus est considéré comme état ouvert.

Nous pouvons également voir la différence entre les seuils de chaque personne, c'est pourquoi nous avons utilisé cette méthode, car prendre un seuil commun ne serait pas pratique.

La figure suivante montre les différents états obtenus des yeux et de la bouche.



Figure 3-13 : La détection de l'état des yeux et de la bouche

## E. Équilibrage de la base de données

La division des vidéos en séquence d'images dans ces conditions a donné lieu à des ensembles de données non équilibrés. Les ensembles de données non équilibrés sont un gros problème pour les modèles d'apprentissage profond que nous avons rencontrés nous-mêmes lors des premiers essais d'entraînement qui ont donné à des mauvais résultats.

Pour résoudre ce problème, nous avons deux choix : soit augmenter le nombre d'images de la classe minoritaire en utilisant le générateur de données d'images fourni par la bibliothèque d'apprentissage profond KERAS, soit de limiter le nombre d'images dans la classe majoritaire.

Nous avons donc fini par choisir la dernière pour deux raisons :

- Premièrement, parce qu'il en résultera un ensemble de données relativement petit, mais néanmoins suffisant, permet aussi de réduire le temps nécessaire à l'entraînement du modèle.
- Deuxièmement, l'utilisation des fonctions du générateur pour augmenter le nombre des images dans la classe minoritaire, peut entraîner une transformation des images et un décalage de leurs informations.

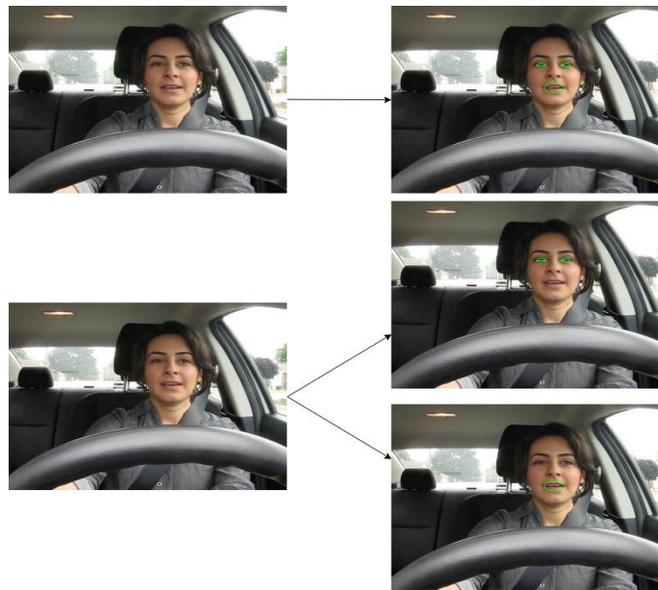
	Base de données des yeux	Base de données de la bouche
<i>Ensemble d'entraînement</i>	2000	2000
<i>Ensemble de validation</i>	400	400
<i>Ensemble de test</i>	400	400
<i>Totale</i>	2800	2800

**Tableau 3-1** : La distribution équilibré d'ensemble de donné sur les trois étapes d'entraînement.

## F. Masquage des données :

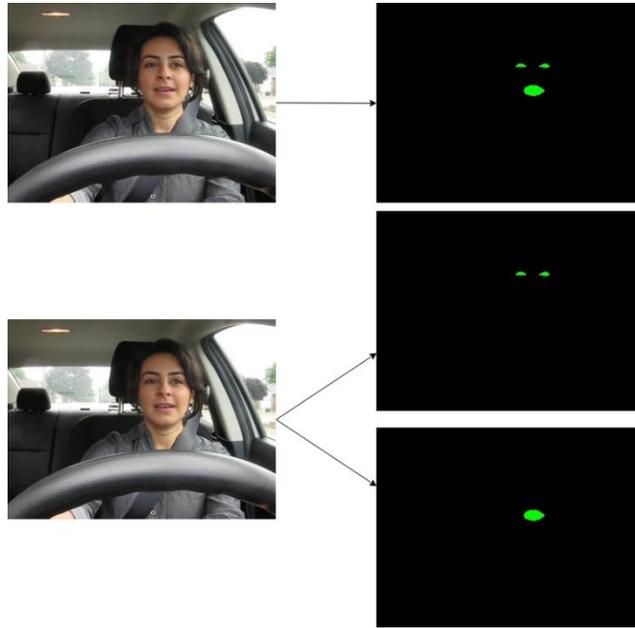
Comme nous intéressons à la détection de la fatigue du conducteur suivant l'état de ses yeux et la bouche. Notre région d'intérêt est seulement les yeux et la bouche, mais les images avaient beaucoup d'information comme l'environnement, la voiture, le volant, et d'autres. Ce qui a rendu la classification directe de ces images une tâche difficile pour le CNN.

La première solution montrée dans la figure 3.14 proposée était d'ajouter les contours les yeux et la bouche dans les images, mais ceci n'a donné lieu à aucune amélioration :



**Figure 3-14** : Dessin des contours sur les images de données pour améliorer les résultats de l'apprentissage (avant et après la séparation des données)

La deuxième solution proposée est d'extraire la région d'intérêt (les yeux et la bouche), cela été fait par l'introduction d'un masque sur les images en cachant toutes les informations non nécessaires et en ne laissant que ce qui est nécessaire. Un masque noir a été placé sur toutes les images et les contours du ROIs ont été remplis de couleur verte, laissant ainsi le CNN voir uniquement les ROIs. Cette procédure a augmenté la précision du CNN.



**Figure 3-15 :** Les masques des images (avant et après la séparation des données).

Ainsi, après la création d'une base de données fiable, il est maintenant temps de passer à l'étape 2, qui consiste à entraîner les modèles et à faire une étude comparative entre les architectures CNNs mentionnées au chapitre 2 pour voir laquelle est la mieux adaptée à notre problématique de détection de la fatigue du conducteur.

### **3.3.2 Étape 2 : Comparaison entre les différentes architectures**

Dans le chapitre 2, nous avons présenté les architectures CNNs les plus récentes et décrites dans le site officiel de KERAS et TensorFlow, ces architectures seront utilisées dans notre études expérimentale.

Nous devons noter que toutes ces architectures sont entraînées sur la base de données ImageNet.

- **ImageNet** : est une base de données qui vise à améliorer la recherche sur la vision par ordinateur et l'apprentissage profond, elle compte plus de 14 millions d'images hiérarchisées [102].

L'entraînement des CNNs sur des grands ensembles de données peut prendre des jours, voire des semaines, même sur des ordinateurs multi-gpu très performants. L'apprentissage par transfert est donc un moyen de raccourcir cette opération chronophage.

- **L'apprentissage par transfert** : consiste à prendre des modèles prés-entraînés sur une base de données de référence standard de vision par ordinateur (tel que : ImageNet, COCO, FMNIST, etc.) congeler les couches inférieures pour ne pas être entraînées à nouveaux et à réutiliser ses poids en plus d'ajouter quelques couches en haut.

## A. Les métriques utilisées

Le choix de la meilleure architecture était basé sur 4 métriques que nous allons expliquer pour mieux comprendre la décision prise.

- **Précision** : est la métrique utilisée pour évaluer les modèles de classification (comme le nôtre), elle représente le nombre de prédictions vraies obtenues par le modèle [103]. Elle est décrite par :

$$Précision = \frac{Nombre\_de\_prédiction\_correctes}{Nombre\_total\_de\_prédiction} \quad (3.3)$$

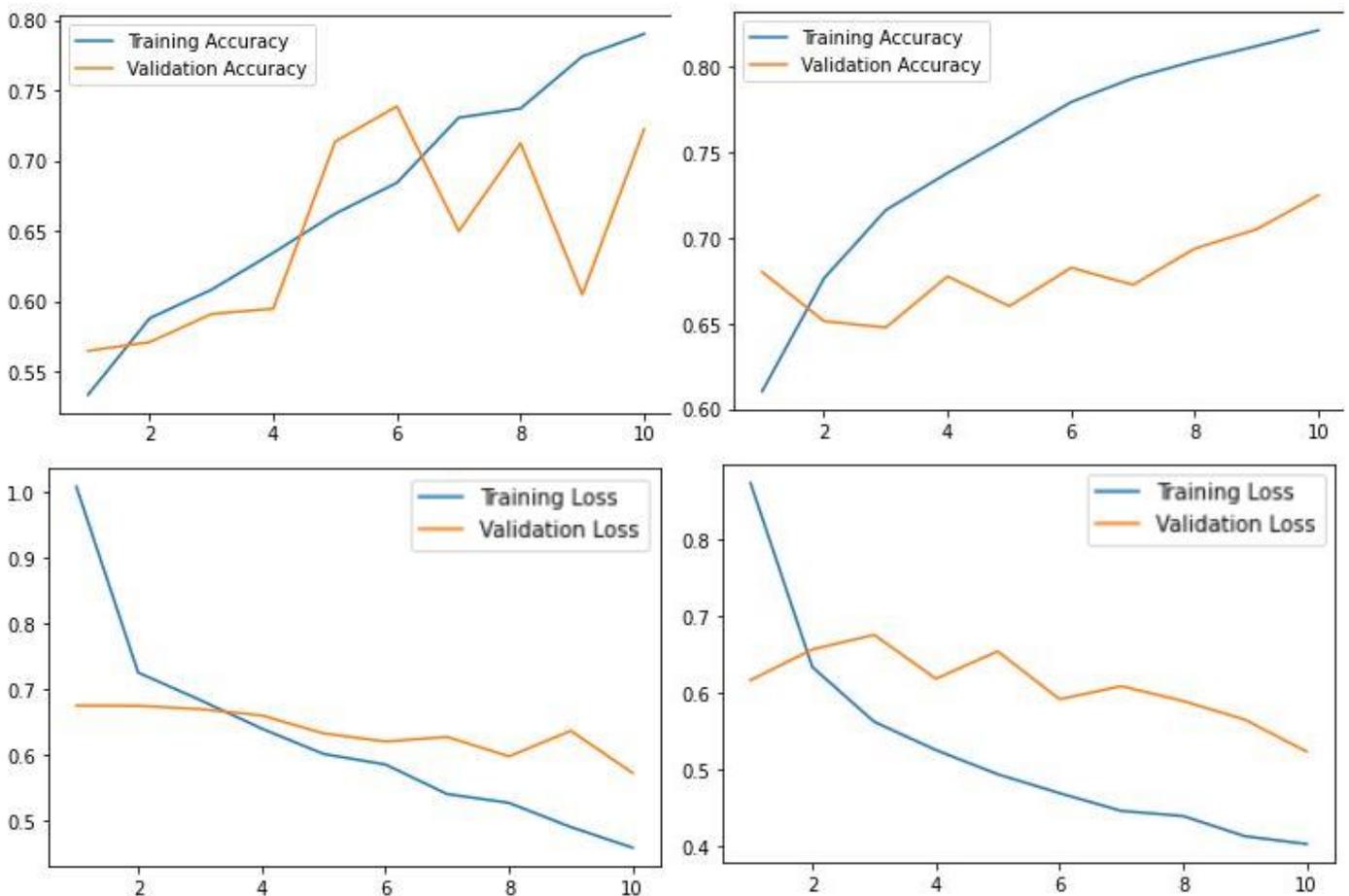
- **L'erreur** : La valeur de l'erreur est la pénalité pour les mauvaises prédictions, décrivant à quel point la prédiction est mauvaise ou éloignée de sa valeur réelle [103]. Pour notre problème, nous avons utilisé la fonction d'entropie croisée catégorielle qui calcule cette somme comme pertes :

$$Perte = - \sum_{i=1}^{TailleDeSortie} y_i * \log \hat{y}_i \quad (3.4)$$

- **Précision de la validation** : est la précision calculée pendant l'entraînement mais sur des données qui ne sont pas utilisées pour l'entraînement. Elle est utilisée pour valider le degré de généralisation de notre modèle sur des données non vues.
- **Erreur de validation** : Calculée de la même manière que la perte d'apprentissage mais n'est pas utilisée pour mettre à jour les poids, elle sert uniquement à voir les pertes généralisées.

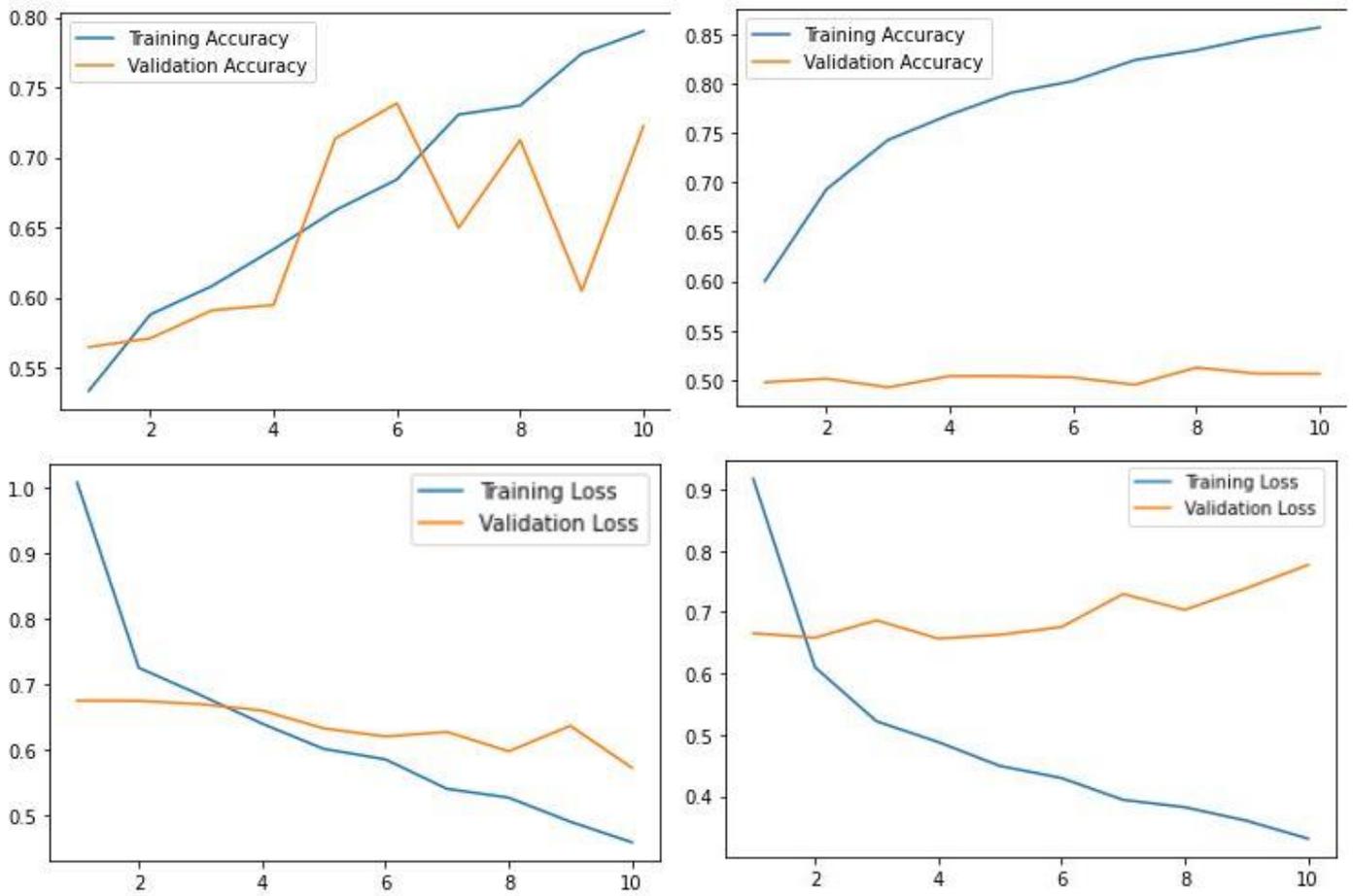
## B. Résultats des tests des architectures

- **VGG16**



**Figure 3-16** : Les courbes de précision et l'erreur du modèle VGG16 pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement.

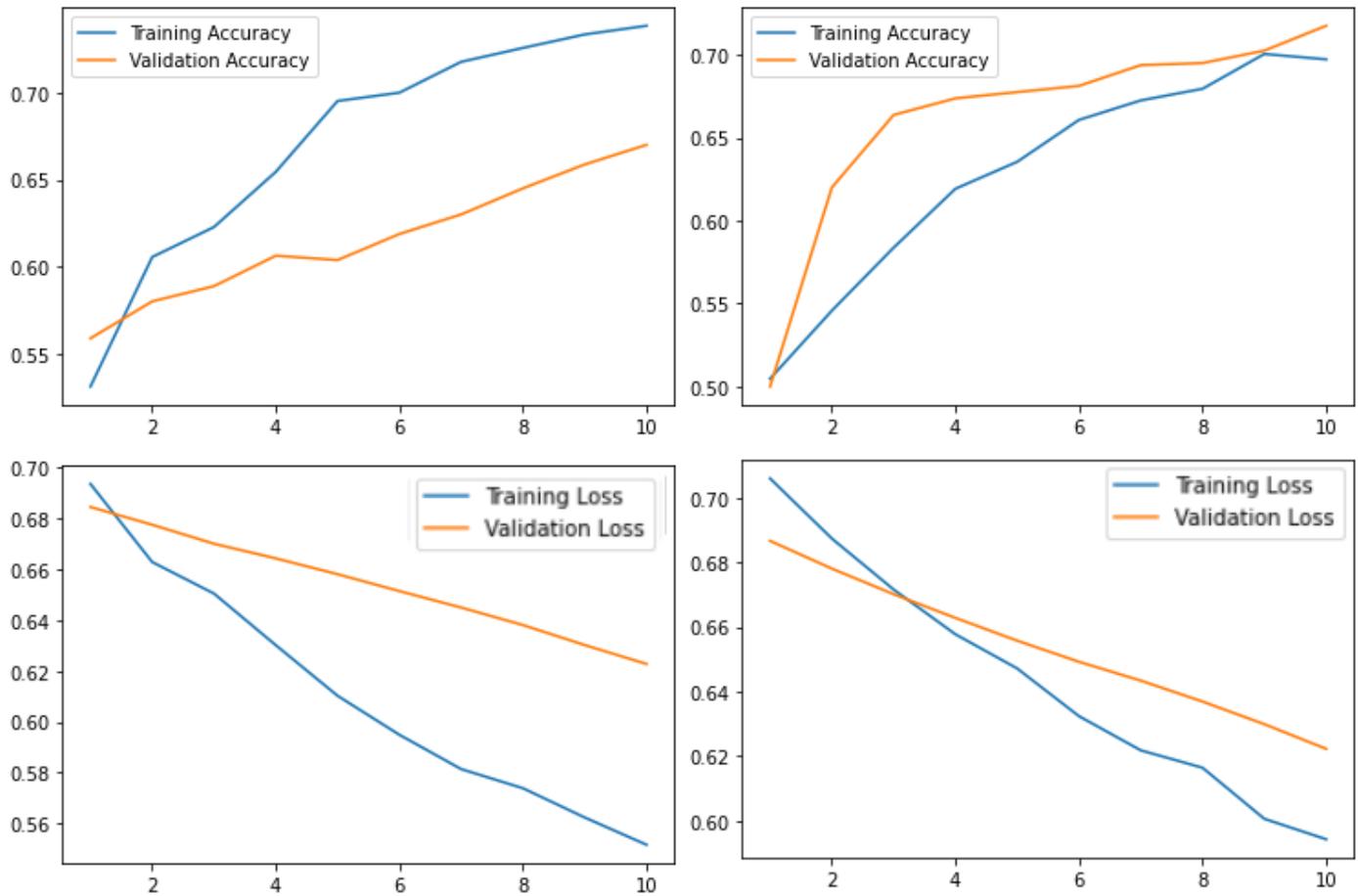
- VGG19



**Figure 3-17 :** Les courbes de précision et l'erreur du modèle VGG19 pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement.

Nous pouvons voir que les modèles VGG16 et VGG19 ont obtenu des résultats médiocres dans ce problème de classification.

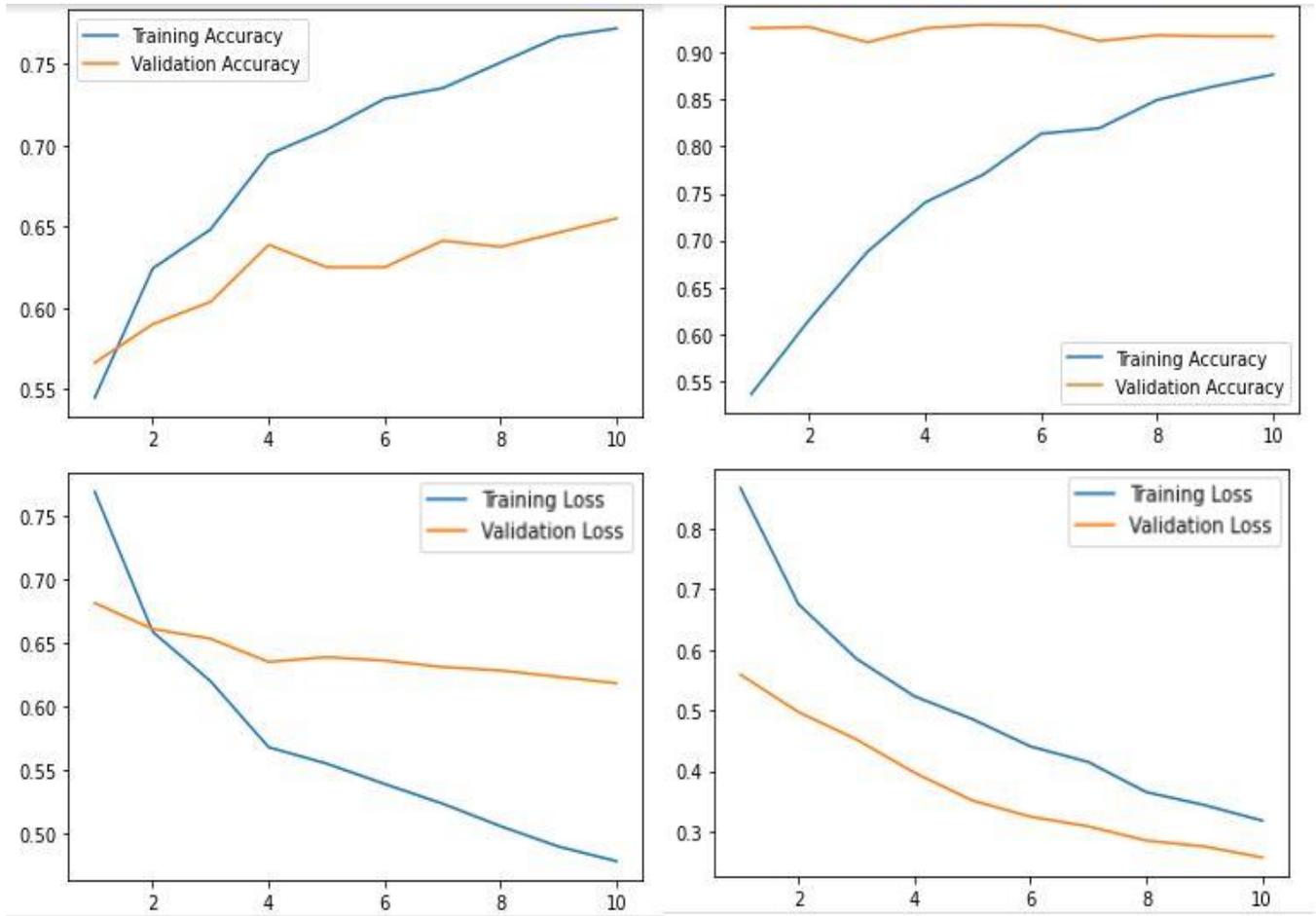
- **EfficientNetB7**



**Figure 3-18** : Les courbes de précision et l'erreur du modèle EfficientNetB7 pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement.

Cette architecture s'est bien développée pendant les époques de l'apprentissage, en particulier le modèle de bouche, mais la perte est si grande et la précision n'est pas si élevée.

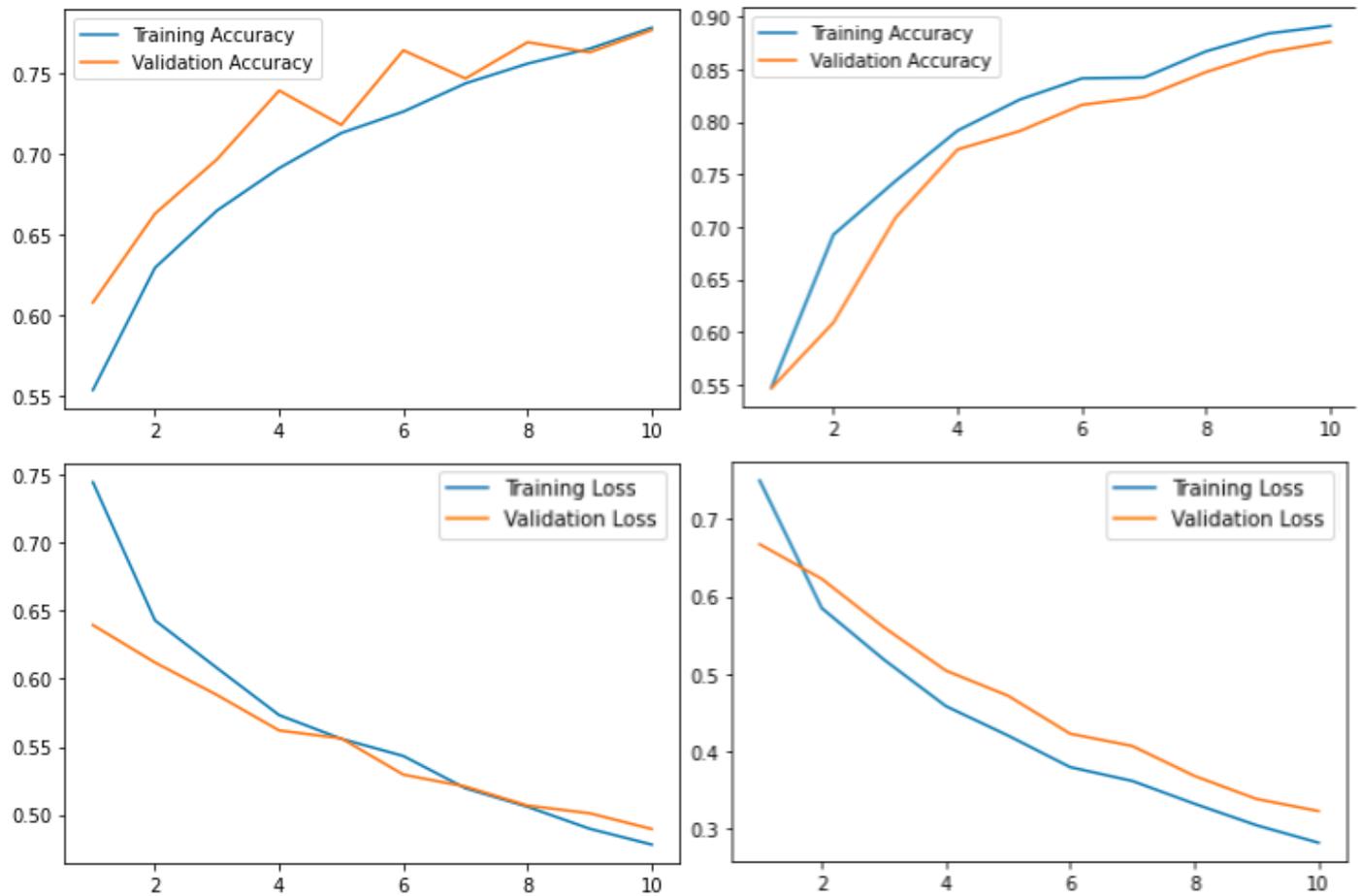
- DenseNet



**Figure 3-19** : Les courbes de précision et l'erreur du modèle DenseNet pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement.

Le modèle des yeux de DenseNet a atteint la plus haute précision de validation.

- InceptionV3

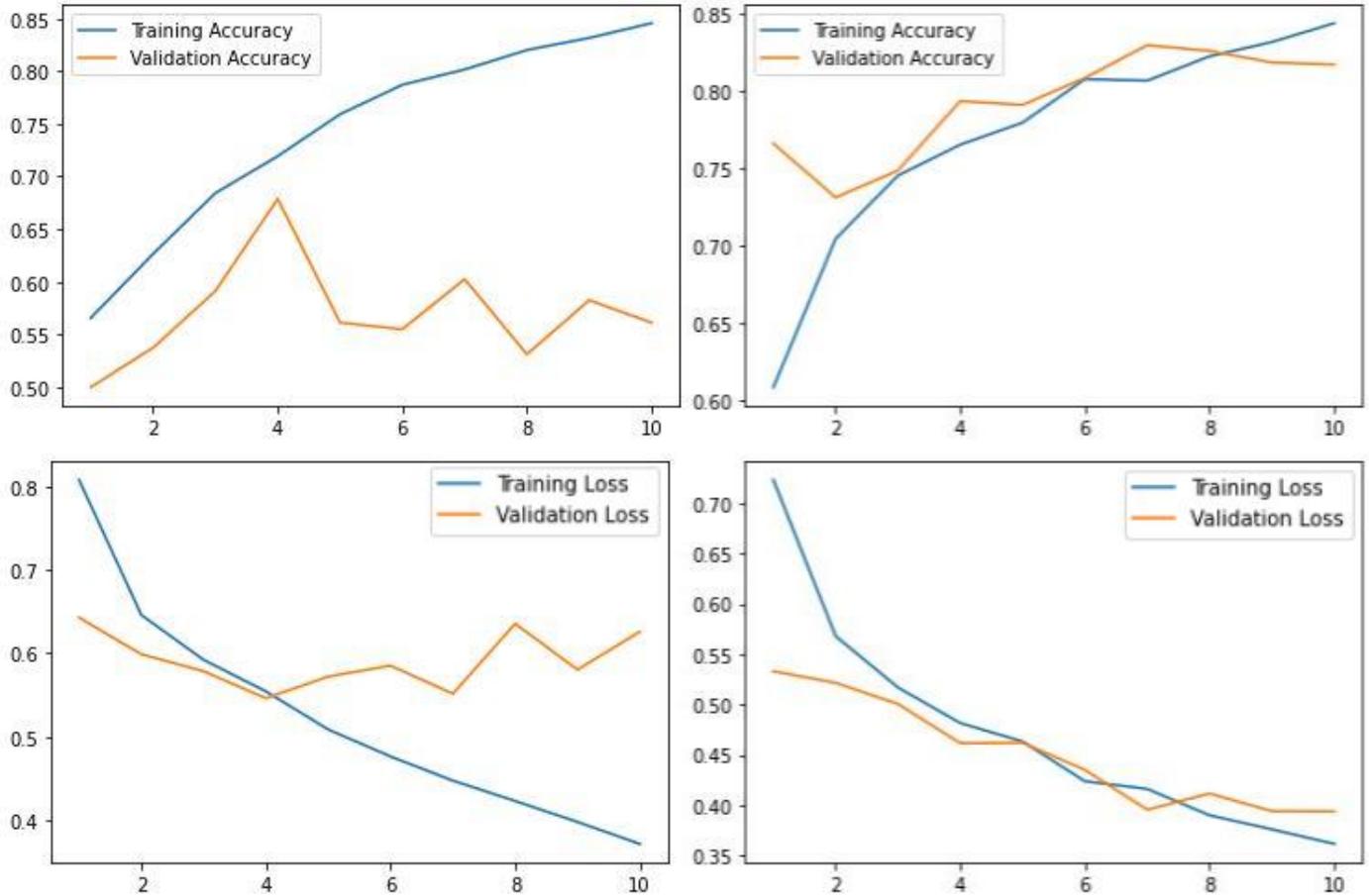


**Figure 3-20** : Les courbes de précision et l'erreur du modèle InceptionV3 pendant 10 époques

d'apprentissage pour yeux et de bouche à gauche et à droite respectivement.

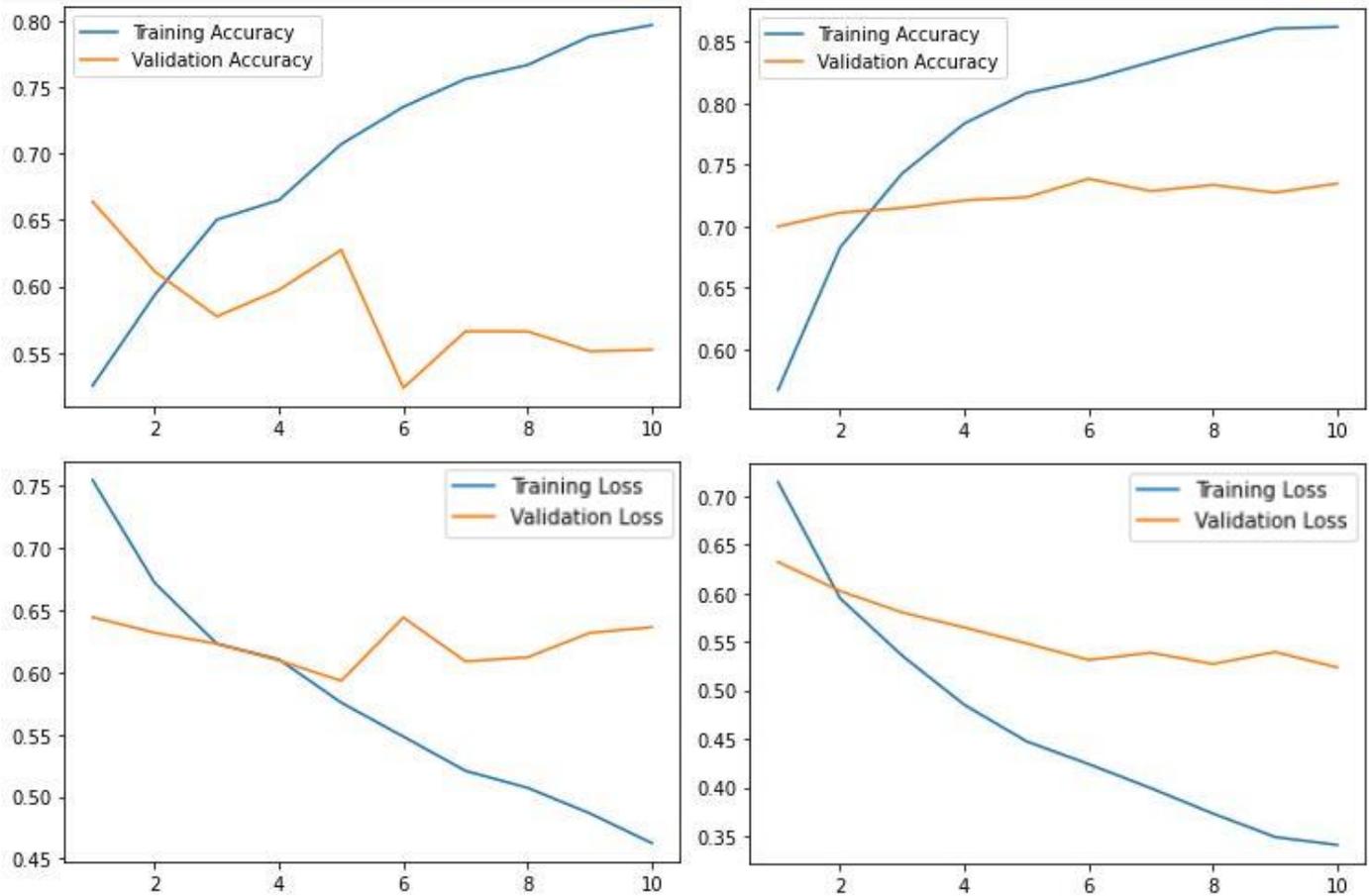
InceptionV3 est une architecture qui fonctionne très bien dans de nombreux problèmes, ce que nous pouvons voir clairement dans le nôtre ici, car elle a atteint une précision décente dans les deux modèles, en particulier celui de la bouche, avec une bonne généralisation.

- ResNet50V2



**Figure 3-21 :** Les courbes de précision et l'erreur du modèle ResNet50V2 pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement.

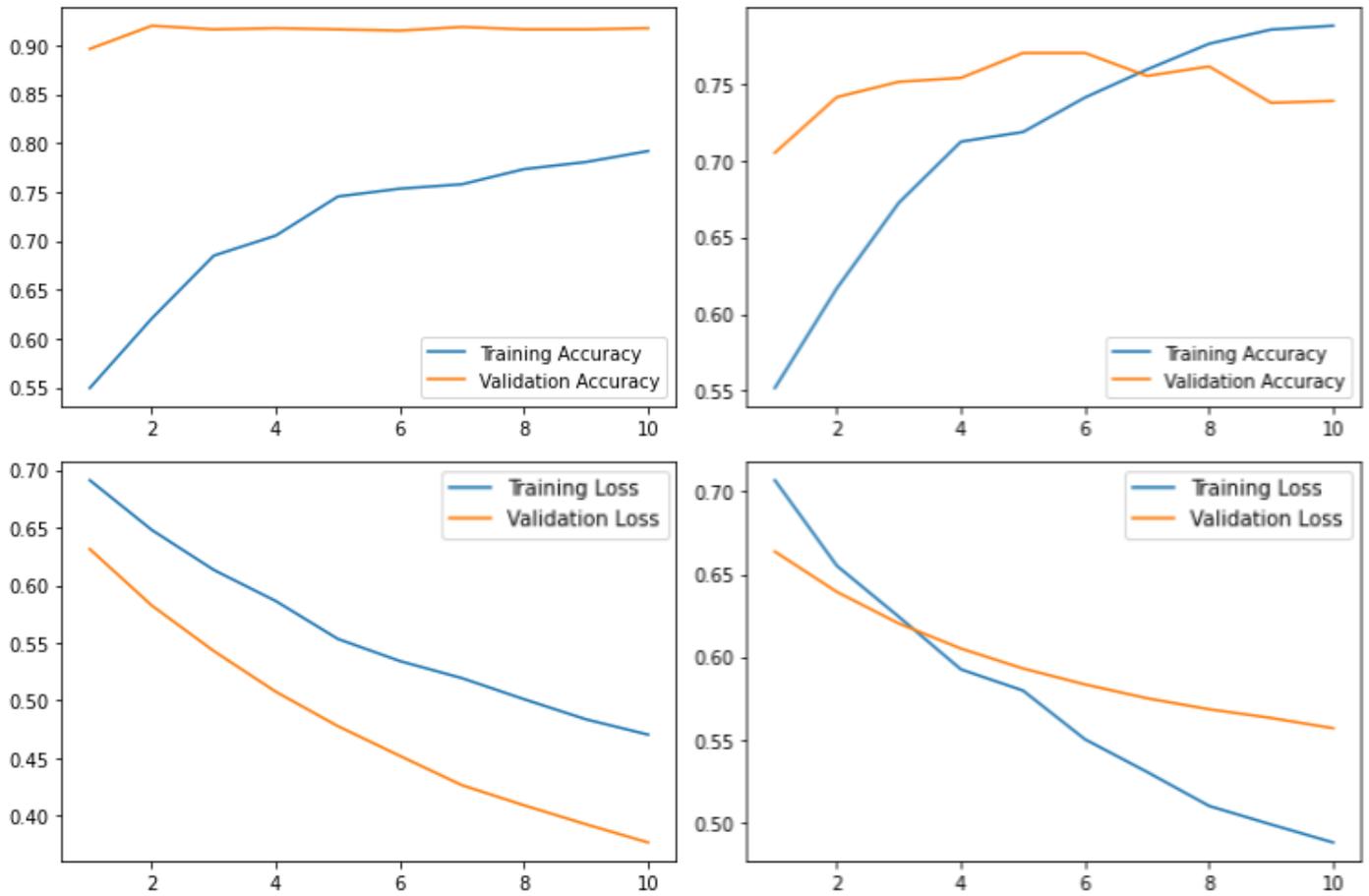
- InceptionResNetV2



**Figure 3-22 :** Les courbes de précision et l'erreur du modèle InceptionResNetV2 pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement.

Pour les deux architectures précédentes (ResNet et InceptionResNet), le modèle des yeux a été sûr ajusté et n'a pas bien appris, mais le modèle de la bouche a obtenu des résultats acceptables.

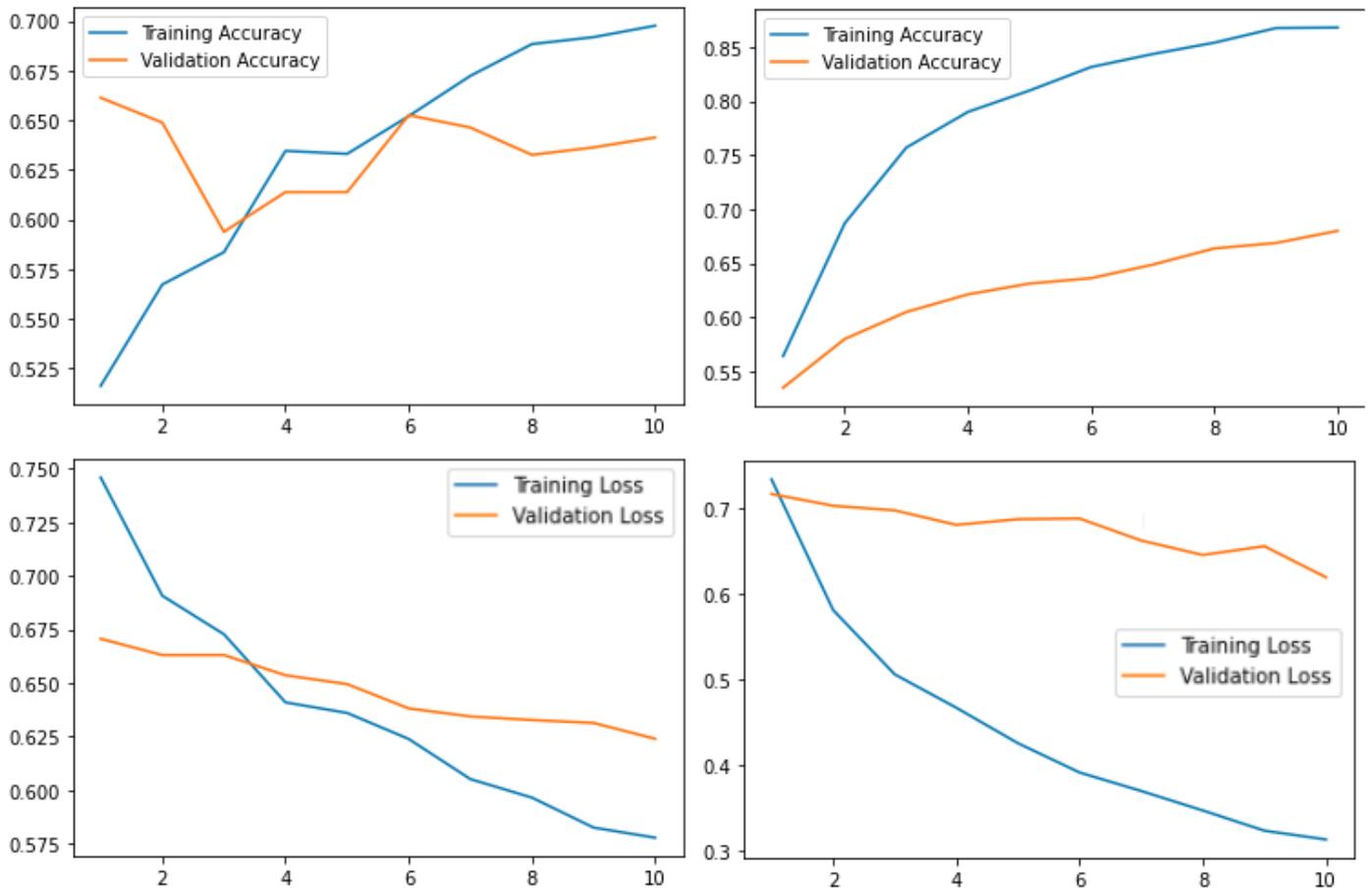
- Xception



**Figure 3-23 :** Les courbes de précision et l'erreur du modèle Xception pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement.

Le modèle des yeux de Xception a atteint la plus haute précision de validation, ce qui est le plus important dans les modèles afin de classer des données non vues.

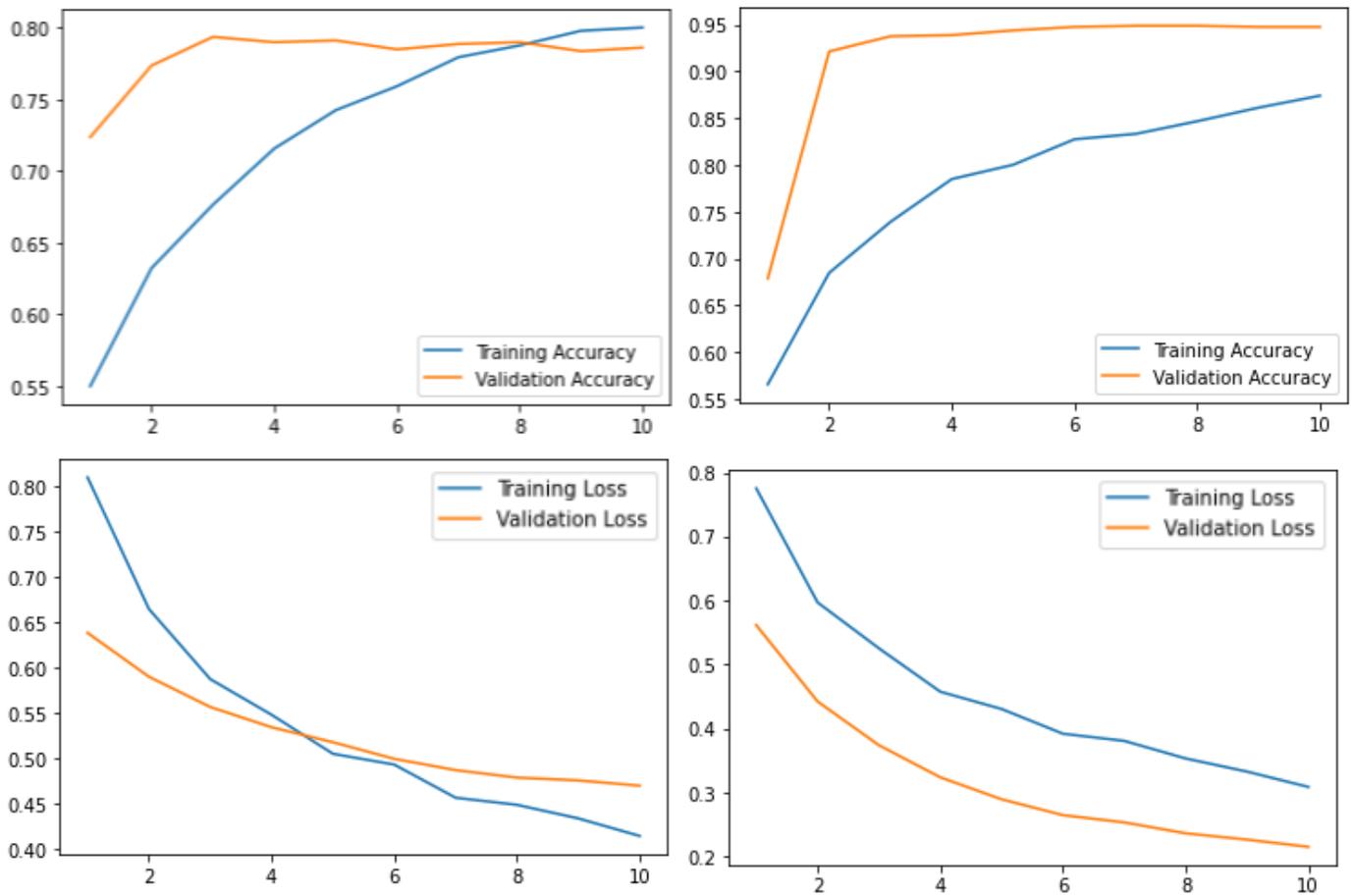
- NasNetMobile



**Figure 3-24 :** Les courbes de précision et l'erreur du modèle NasNetMobile pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement.

Cette l'architecture NasNetMobile n'a pas donné un bon résultat pour la classification des yeux et la bouche.

- **MobilenetV2**



**Figure 3-25 :** Les courbes de précision et l'erreur du modèle MobileNetV2 pendant 10 époques d'apprentissage pour yeux et de bouche à gauche et à droite respectivement.

MobileNet a bien classifié les données avec de bons résultats de validation, en plus elle est la plus légère et la plus rapide.

Les résultats obtenus pour chaque architecture testée sont présentés dans les deux tableaux qui suivent :

**Tableau des résultats des modèles des yeux**

	<i>L'erreur</i>	<i>Précision</i>	<i>L'erreur de validation</i>	<i>Précision de la validation</i>
VGG16	0.3096	0.8658	1.0220	0.5475
VGG19	0.4589	0.7903	0.5729	0.7225
EfficientNetB7	0.5516	0.7385	0.6228	0.6700
DenseNet	0.4785	0.7717	0.6181	0.6550
InceptionV3	0.4780	0.7778	0.4895	0.7763
ResNet50V2	0.3721	0.8455	0.6258	0.5612
InceptionResnetV2	0.4627	0.7965	0.6363	0.5525
Xception	0.4704	0.7918	0.3769	0.9175
NasNetMobile	0.5779	0.6975	0.6240	0.6413
MobileNetV2	0.4148	0.8002	0.4703	0.7862

**Tableau 3-2 :** Le résultat obtenu dans la dernière 10ème époque par chaque architecture testée dans la classification de l'état des yeux.

### Tableau des résultats des modèles de la bouche

	<i>L'erreur</i>	<i>Précision</i>	<i>L'erreur de validation</i>	<i>Précision de la validation</i>
VGG16	0.4028	0.8215	0.5232	0.7250
VGG19	0.3308	0.8568	0.7770	0.5063
EfficientNetB7	0.5942	0.6973	0.6222	0.7175
DenseNet	0.3177	0.8767	0.2568	0.9175
InceptionV3	0.2820	0.8915	0.3228	0.8763
ResNet50V2	0.3617	0.8443	0.3939	0.8175
InceptionResnetV2	0.3407	0.8622	0.5234	0.7350
Xception	0.4877	0.7878	0.5568	0.7387
NasNetMobile	0.3130	0.8680	0.6198	0.6800
MobileNetV2	0.3088	0.8740	0.2153	0.9475

**Tableau 3-3 :** Le résultat obtenu dans la dernière 10ème époque par chaque architecture testée dans la classification de l'état de la bouche.

En analysant les résultats des tests de toutes les architectures, qui sont présentés dans les figures et les tableaux ci-dessus, nous avons fini par choisir **MobileNetV2**.

**MobileNetV2 :** Cette architecture a obtenu un très bon score pour la classification de l'état des yeux et la bouche, De plus, il s'agit d'une architecture légère qui convient parfaitement au système intégré Raspberry Pi dans notre projet d'ingénierie.



Figure 3-26 : Un exemple des images classifiées correctement avec le model MobileNetV2.

### 3.3.3 Étape 3 : Détection de fatigue

Comme la durée des vidéos des conducteurs de la base de données YawDD était sur 1 minute à 1:50 minute contenant les 3 états : normal, parlant et bâillant.

Dans la première étape, nous avons découpé les vidéos de l'ensemble de données YawDD en plus petites vidéos en tenant compte de l'état visualisé du conducteur (normal, parlant ou bâillant) et nous les avons stockées dans différents dossiers.

Dans la deuxième étape, l'état des yeux et de la bouche ont été classés en utilisant MobileNet. Ensuite, pour détecter l'état fatigue ou actif, nous avons calculé les deux paramètres : le pourcentage de fermeture des yeux PERCLOS et la fréquence des mouvements de bouche dont nous avons détaillé au chapitre 1.

On rappelle que :

- **Pourcentage de fermeture des yeux (PERCLOS)**

PERCLOS représente le ratio des yeux fermés en fonction du nombre des yeux ouverts et fermés [104].

Cette valeur est calculée comme suit :

$$f_{PERCLOS} = \frac{n_{fermés}}{N_{fermés} + n_{ouverts}} \times 100\% \quad (3.5)$$

$n_{fermer}$  : représente le nombre d'images d'œil fermés à un moment donné.

$N_{fermer\&ouvert}$  : représente le nombre total d'ouvertures et de fermetures d'œil dans une période.

La recherche a prouvé qu'une personne cligne des yeux environ 10 fois par minute dans des conditions normales. PERCLOS haut ou bas est une méthode utilisée pour détecter la fatigue oculaire [105].

- **Fréquence de la bouche (FOM)**

La fréquence de la bouche (FOM) est le ratio de bouche ouverte en fonction du nombre de bouche ouverte et fermée.

Le calcul de FOM est similaire au calcul de PERCLOS et ça valeur est calculée comme suit :

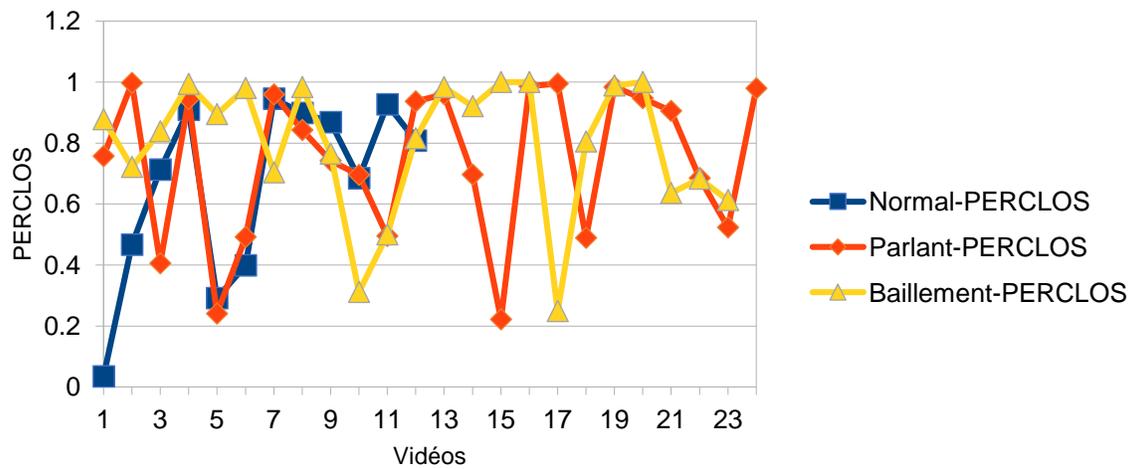
$$f_{FOM} = \frac{n_{ouvert}}{N_{fermer\&ouvert}} \times 100\% \quad (3.6)$$

$n_{ouvert}$  : représente le nombre d'images de la bouche ouverte à un moment donné.

$N_{fermer\&ouvert}$  : représente le nombre total d'ouvertures et de fermetures de la bouche dans une période.

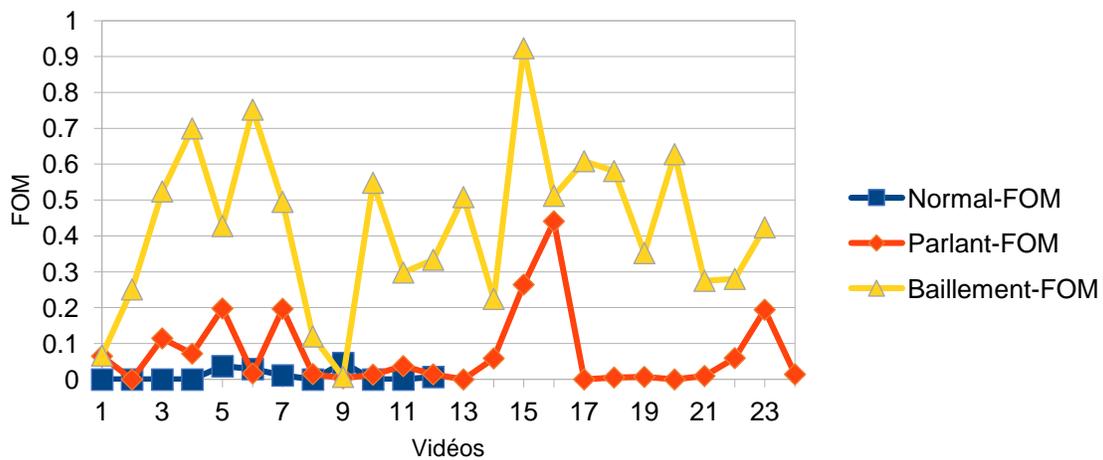
Nous avons établi un code python qui utilise le modèle entraîné pour classer les séquences d'images des vidéos coupées de chaque dossier, calculer ces deux paramètres pour chaque vidéo. Nous avons ensuite tracé les résultats obtenus (PERCLOS des trois états ensemble et ainsi pour FOM).

**Comparaison entre les valeurs de PERCLOS calculés avec MobileNetV2 des trois états**



**Figure 3-27 :** Les courbes des valeurs de PERCLOS en fonction de l'état du conducteur avec le modèle MobileNetV2.

**Comparaison entre les valeurs de FOM calculés avec MobileNetV2 des trois états**



**Figure 3-28 :** Les courbes des valeurs de FOM en fonction de l'état du conducteur avec le modèle MobileNetV2.

A partir de ces graphes nous avons établi un arbre de décision pour déterminer si le conducteur est fatigué ou non. Illustré dans la figure ci-dessous :

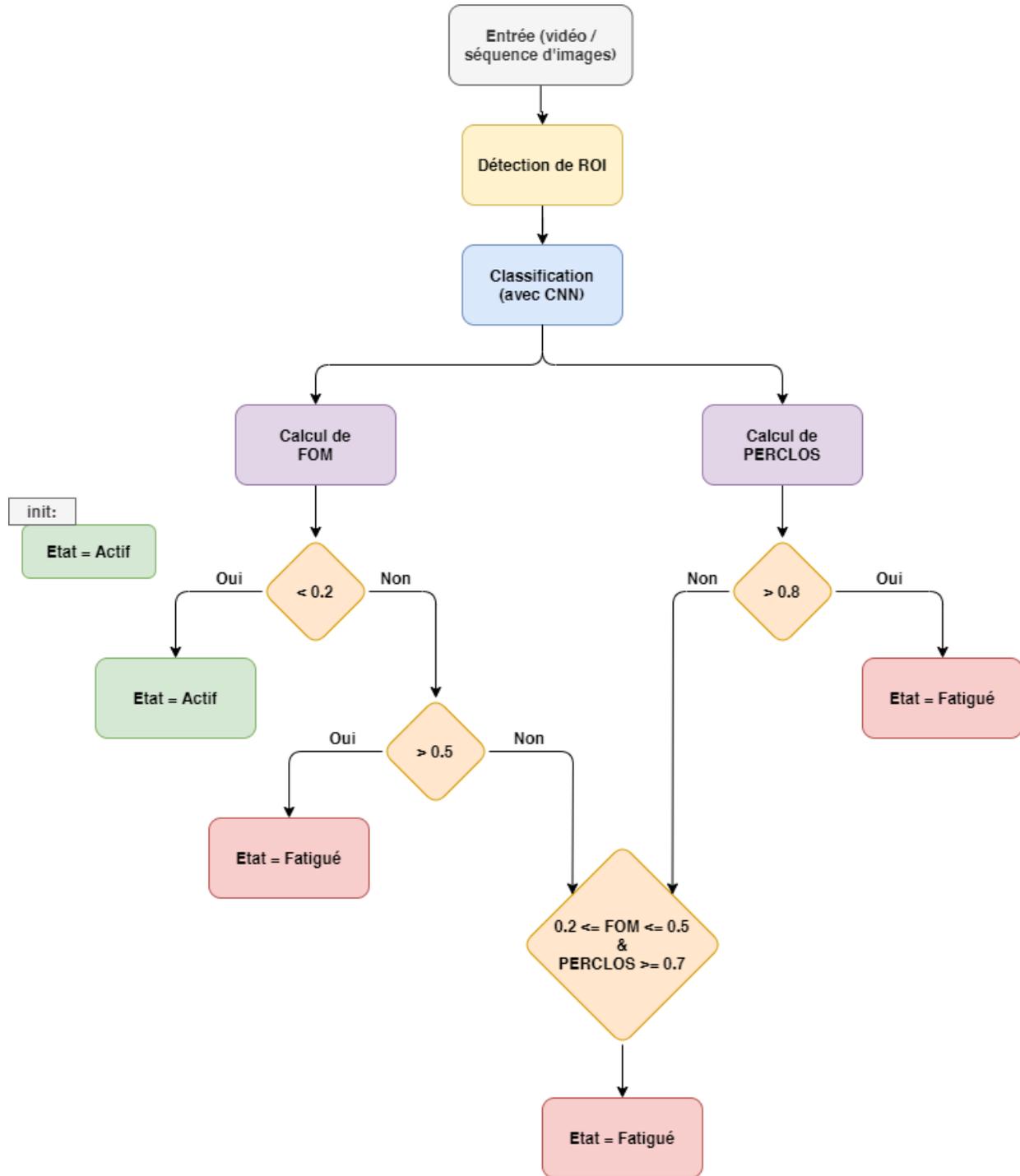


Figure 3-29 : L'organigramme de décision de la détection de la fatigue.

### 3.4 Analyse des résultats

Pour valider et tester la méthode proposée pour la détection de fatigue du conducteur, 20% des vidéos ont été utilisé pour l'évaluation.

Et la détection de la fatigue a été prédite comme suit :

Filename	PERCLOS	FOM	Predicted State	Real (Visualised) State
Fichier 1	0.46179401993356	0.00332225913621	active	active
Fichier 2	0.67878787878788	0.01212121212121	active	active
Fichier 3	0.69061876247505	0.04590818363274	active	drowsy
Fichier 4	0.3035343035343	0.05197505197505	active	active
Fichier 5	0.99252934898613	0.00960512273212	drowsy	active
Fichier 6	0.77777777777778	0.03174603174603	active	active
Fichier 7	0.94270833333333	0.02604166666667	drowsy	drowsy
Fichier 8	0.78744939271255	0.0668016194332	active	drowsy
Fichier 9	0.79514824797844	0.02964959568733	active	active
Fichier 10	1	0.04667863554758	drowsy	drowsy
Fichier 11	0.78838174273859	0.66390041493776	drowsy	drowsy
Fichier 12	0.79318734793187	0.58394160583942	drowsy	drowsy
Fichier 13	1	0.51700680272109	drowsy	drowsy
Fichier 14	0.42342342342342	0.36036036036036	drowsy	drowsy
Fichier 15	0.94372294372294	0.26839826839827	drowsy	drowsy
Fichier 16	0.99486301369863	0.25	drowsy	drowsy

**Tableau 3-4 :** Une liste des vidéos séparées selon l'état avec le PERCLOS et le FOM calculés et l'état prédit correspondant avec la vraie prédiction marquée en vert et la fausse en rouge.

Les résultats présentés dans la table ci-dessus, prouvent que l'application de détection de la fatigue que nous avons proposée est performante puisqu'elle a obtenu 13 prédictions vraies sur 16. La majorité des prédictions incorrectes (2 sur 3) concernaient l'état de parole car il s'agit de l'état le plus déroutant pour le modèle lorsque la bouche est considérée comme ouverte alors que le conducteur ne bâille pas vraiment mais parle seulement, c'est donc là que les erreurs se produisent.

On calcule maintenant la précision de la classification de ces 16 fichiers.

<p><b>Vrai Positif (VP) :</b></p> <ul style="list-style-type: none"> <li>• Réalité : Somnolent</li> <li>• Modèle prédit : Somnolent</li> <li>• Nombre de résultats VP : 8</li> </ul>	<p><b>Faux Positif (FP) :</b></p> <ul style="list-style-type: none"> <li>• Réalité : Actif</li> <li>• Modèle prédit : Somnolent</li> <li>• Nombre de résultats FP : 1</li> </ul>
<p><b>Faux Négatif (FN) :</b></p> <ul style="list-style-type: none"> <li>• Réalité : Somnolent</li> <li>• Modèle prédit : Actif</li> <li>• Nombre de résultats FN : 2</li> </ul>	<p><b>Vrai Négatif (VN) :</b></p> <ul style="list-style-type: none"> <li>• Réalité : Actif</li> <li>• Modèle prédit : Actif</li> <li>• Nombre de résultats VN : 5</li> </ul>

**Tableau 3-5 :** Matrice de confusion de l'évaluation de nombre de vraies et fausses prédictions du modèle.

Le taux des prédictions correctes est calculé par l'équation (3.7) comme suit :

$$Accuracy = \frac{VP + VN}{VP + FP + FN + VN} = \frac{5 + 8}{5 + 2 + 1 + 8} = 0.8125 \quad (3.7)$$

Donc la performance globale du modèle était très bonne avec une efficacité de 81,25%.

Nous avons fait un autre teste pour contourner la contrainte du temps qui existe dans les vidéos de la base *YawDD*. En effet, nous avons assemblé quelques vidéos de différents états mais avec un état dominant pour voir si l'application peut le détecter l'état du conducteur. Les résultats étaient les suivants :

Filename	PERCLOS	FOM	Predicted State	Real (Visualised) State
1-NNTNYN	0.30634842342342	0.26527921204804	active	active
2-NYTYT	0.90189921372294	0.69684159201776	drowsy	drowsy
3-TYTNTT	0.64956735182127	0.51700680272109	drowsy	active
4-YTNNTN	0.45666450584424	0.40512648940541	active	active
5-YYNTYY	0.97469420692186	0.70303716687121	drowsy	drowsy

**Tableau 3-6 :** Une liste de vidéos assemblées de différents états avec les PERCLOS et FOM calculés et l'état prédit correspondant avec la vraie prédiction marquée en vert et la fausse en rouge.

Les combinaisons de vidéos assemblées sont visibles dans le nom du fichier où chaque lettre fait référence à un état donné (N : Normal, T : Parlant, Y : Baillant).

En utilisant cette méthode, la méthode a obtenu un taux de prédiction égale 80%.

### **3.5 Limitations du travail**

Nous avons traversé une longue route dans cette étude pour comparer les performances de différentes architectures CNNs dans la détection de la fatigue des conducteurs. Nous avons été confrontés à de nombreux problèmes, mais la plupart d'entre eux ont fini par être surmontés, mais il en reste d'autres qui posent encore des limites à notre travail, nous allons donc mentionner quelques-uns.

- Le CNNs ont besoin d'un ensemble très élevé de données pour faire l'apprentissage. Dans cette étude nous avons utilisé uniquement une seule base de données, Nous envisageons d'utiliser NTHU-DDD pour valider les résultats obtenus.
- La bibliothèque Dlib ne détecte pas le visage si la personne n'est pas en face la caméra. Une utilisation de CNNs est possible afin de surmonter ce problème.
- Les temps d'entraînement des CNNs est très élevé. Colab, la plateforme gratuite utilisée pour l'entraînement de nos modèles, a une limite d'utilisation car les ressources qu'elle fournit ne sont pas infinies. Nous n'avons pas les ressources de calcul nécessaire pour effectuer plus de tests d'entraînement ou même des époques plus élevées pour chaque modèle.

### **3.6 Conclusion**

La détection de la fatigue à l'aide des méthodes de vision par ordinateur est une tâche très défiante, et elle est en cours de développement même dans les grandes entreprises technologiques.

Dans ce chapitre, nous avons utilisé les réseaux neuronaux convolutifs pour faire la détection de la fatigue du conducteur, en expliquant le processus, étape par étape, du tout début de la mise en place de l'environnement de programmation à la construction d'une base de données appropriée pour la détection de la fatigue, puis aux tests de plusieurs architectures CNN et à l'obtention du meilleur modèle pour notre application, nous avons aussi analysé les résultats obtenus pour classer l'état des yeux ouverts / fermés et la bouche ouverte / fermées afin de calculer les paramètres de fatigue PERCLOS et FOM que nous avons utilisés pour déterminer l'état du conducteur.

Nous avons aussi validé nos résultats avec deux différentes méthodes et nous avons fini par mentionner certaines limites de notre travail.

# Conclusion Générale

La fatigue et la somnolence provoquant un retard d'une réaction physique et mentale ont toujours été des facteurs de risque pour les conducteurs, en raison du risque élevé d'accidents qu'elles entraînent.

Les statistiques effrayantes de l'impact de la somnolence sur la route, ont été notre motivation pour le développement d'un système de détection de la fatigue du conducteur. Ce système basé sur l'intelligence artificielle, aide à prévenir les accidents mortels en alertant le conducteur s'il montre des signes de somnolence.

Dans ce travail nous avons choisi d'utiliser les mesures comportementales PERCLOS et FOM, extrait à partir de l'analyse faciale du conducteur.

La première étape de notre méthode consistée à préparer la base de données, en appliquant les fonctions de prétraitement aux vidéos.

Ensuite, nous avons comparé les architectures CNNs les plus récentes afin de choisir la plus adéquate à notre problème.

À la fin, nous avons utilisé le CNN sélectionné pour calculer PERCLOS et FOM, et suivant un arbre de décision pour avons conclu l'état du conducteur.

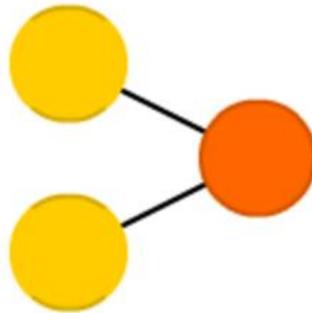
Comme perspective, nous voulons réaliser ce travail sur le terrain, en utilisant des systèmes embarqués installés dans des voitures réelles.

# Annexe A

Dans cette annexe, nous présenterons les différentes architectures des réseaux neuronaux définies dans la section 1.2.8.

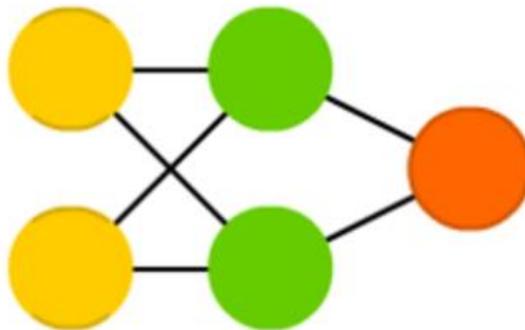
Nous rappelons que les architectures sont : P, FF, DFF, CNN, RNN et LSTM.

## Perceptron (P)



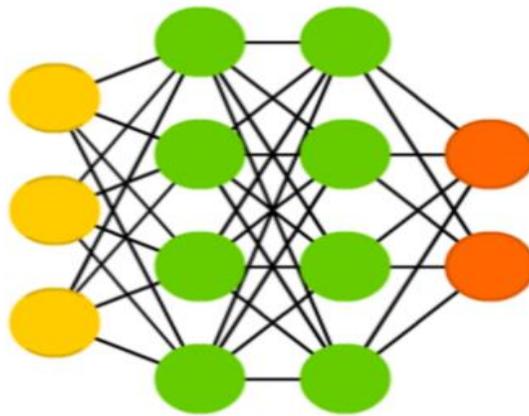
**Figure A.1 :** Architecture de Perceptron.

## Feed Forward (FF)



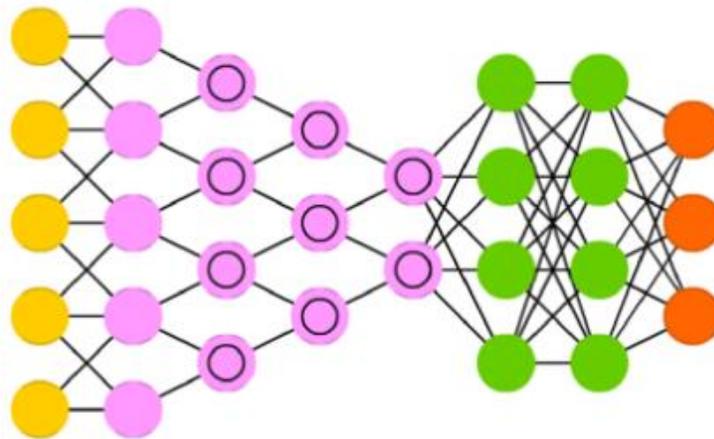
**Figure A.2:** Architecture de Feed Forward Network.

### Deep Feed Forward (DFF)



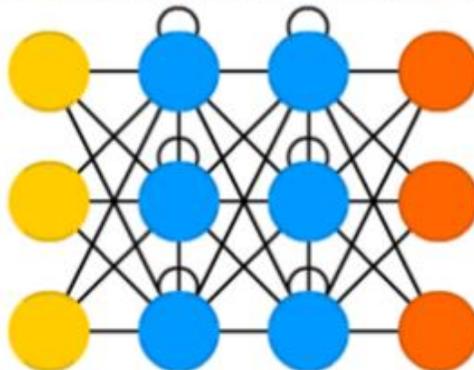
**Figure A.3:** Architecture de Deep Feed Forward Network.

### Deep Convolutional Network (DCN)



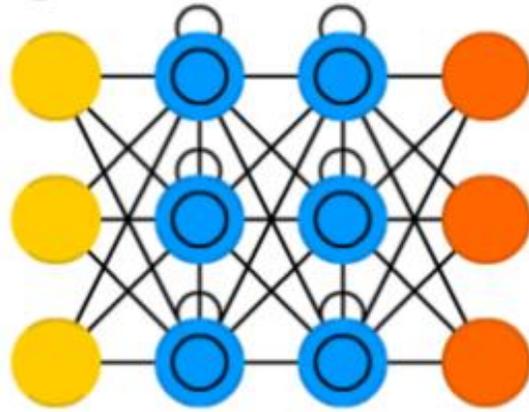
**Figure A.4 :** Architecture de Convolution Neural Network.

### Recurrent Neural Network (RNN)



**Figure A.5 :** Architecture de Ruccurent Neural Network.

## Long / Short Term Memory (LSTM)



**Figure A.6:** Architecture de Long / Short Term Memory.

# Annexe B

Dans cette annexe, nous présenterons les différentes architectures CNNs définies dans la section 1.3.3.

Nous rappelons que les architectures sont : LeNet-5, AlexNet, Inception-v1, Inception-v3, Resnet, FaceNet, Xception, DenseNet, MobileNet-v1, MobileNet-v2, EfficientNet, Noisy Student.

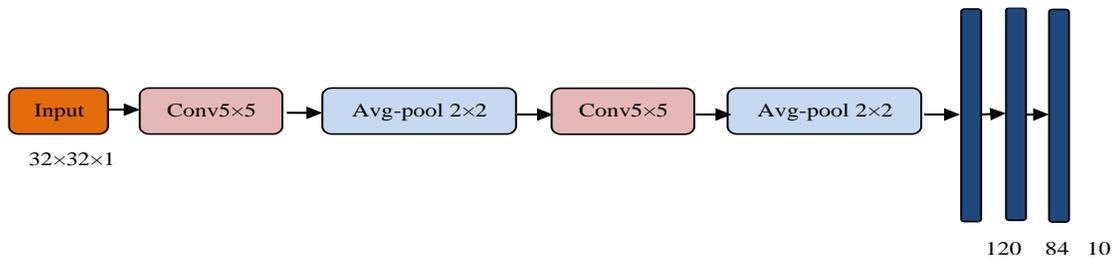


Figure B.1 : Architecture LeNet-5.

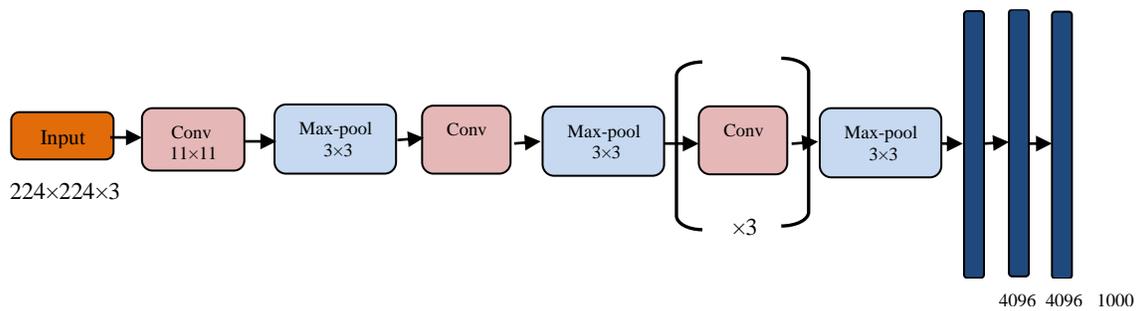


Figure B.2 : Architecture AlexNet.

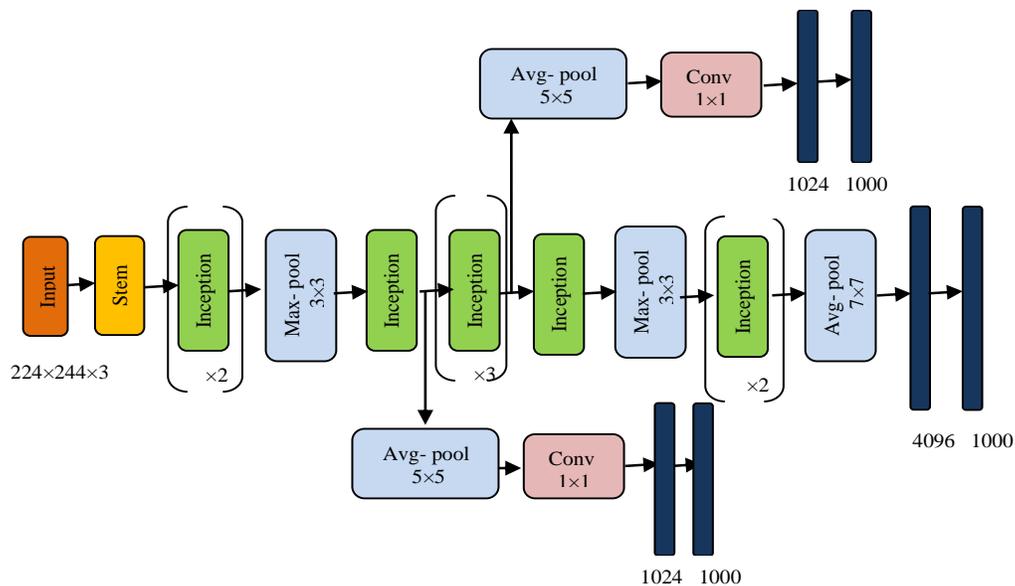
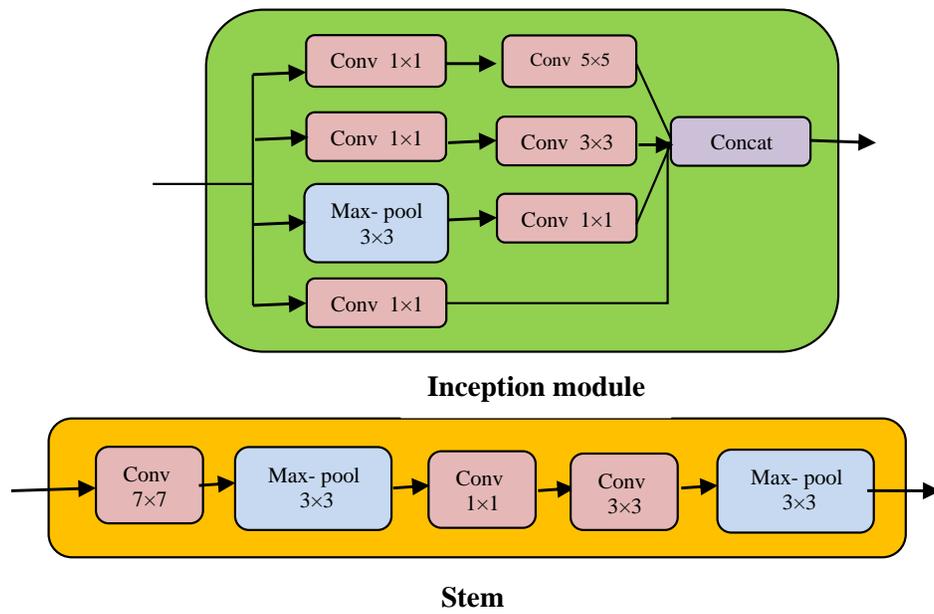
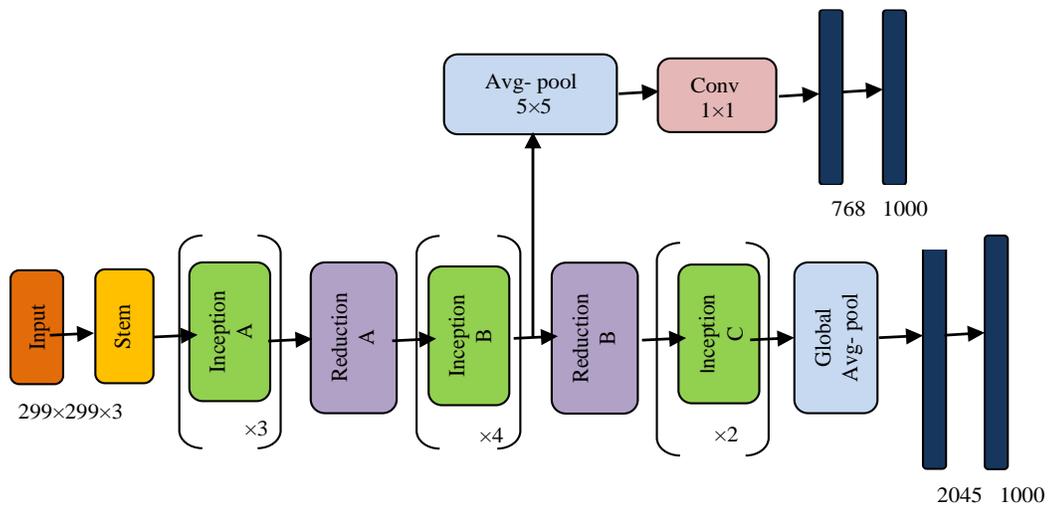
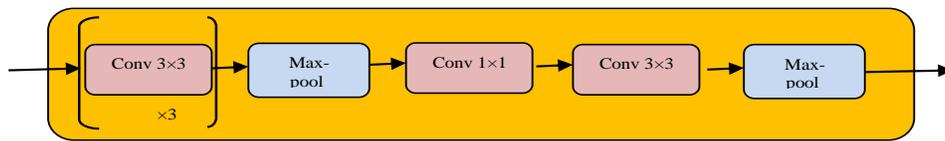


Figure B.3: Architecture Inception-v1.

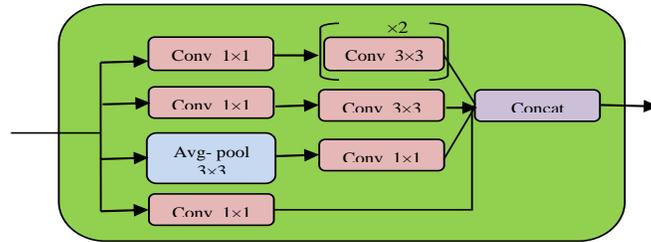


**Figure B.4:** Module Inception et Architecture Stem.

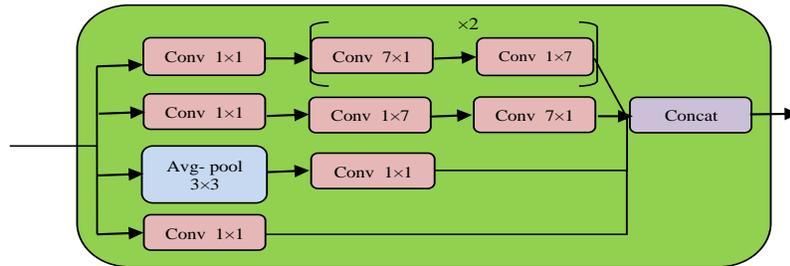




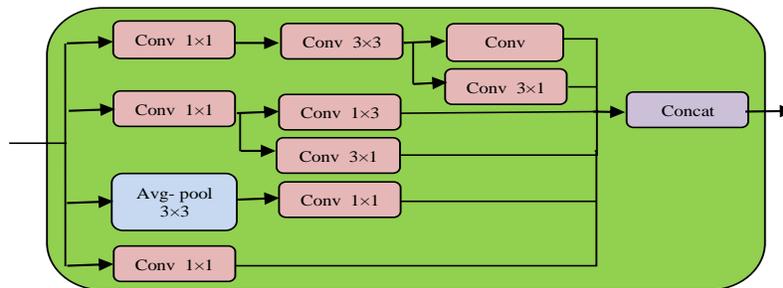
**Stem**



**Inception-A**

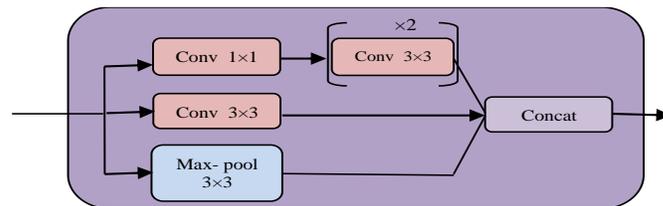


**Inception-B**

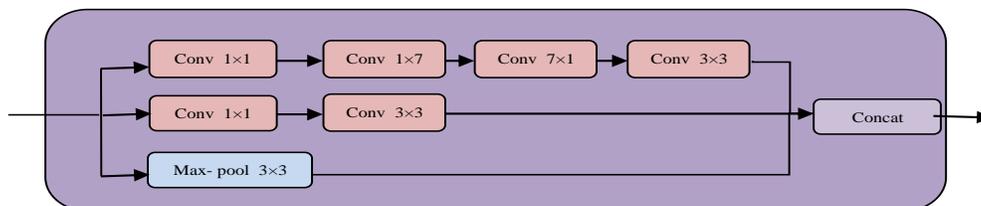


**Inception-C**

**Figure B.6: Stem et Module InceptionV3 A-B-C.**

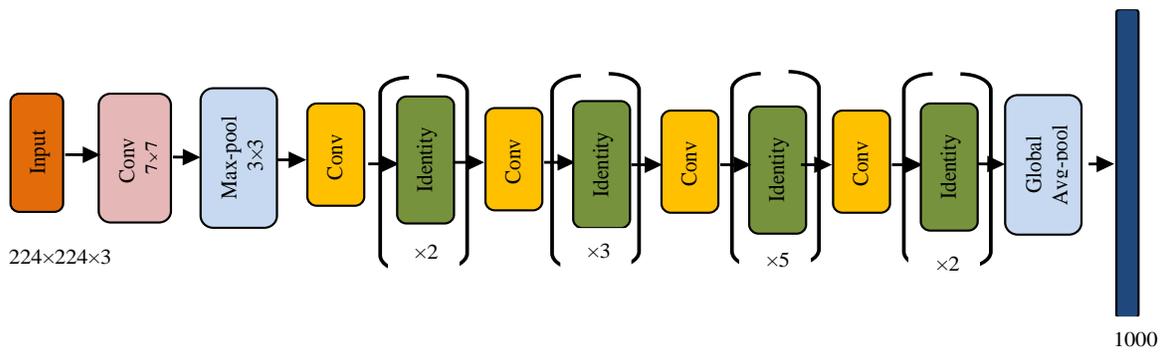


**Reduction-A**

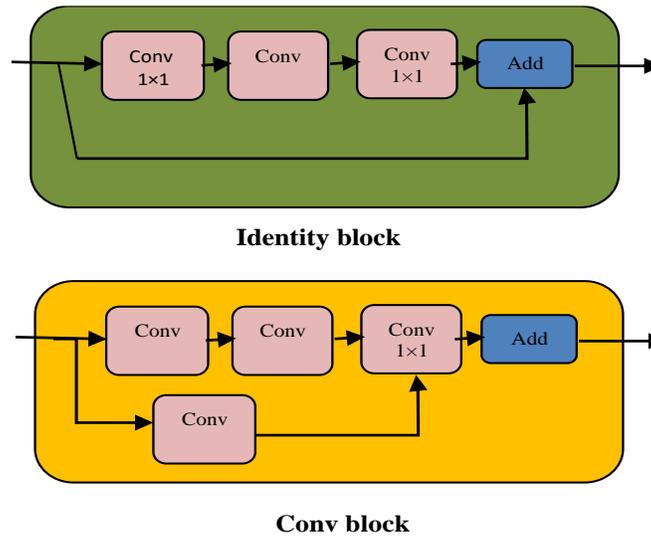


**Reduction-B**

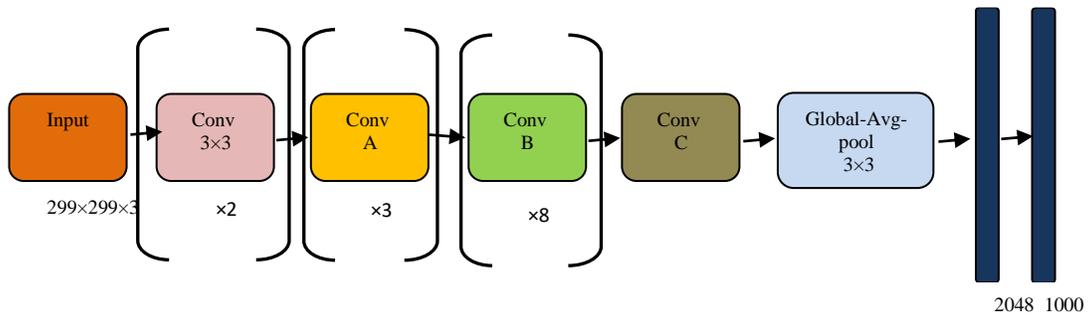
**Figure B.7: Reduction A-B of InceptionV3 architecture.**



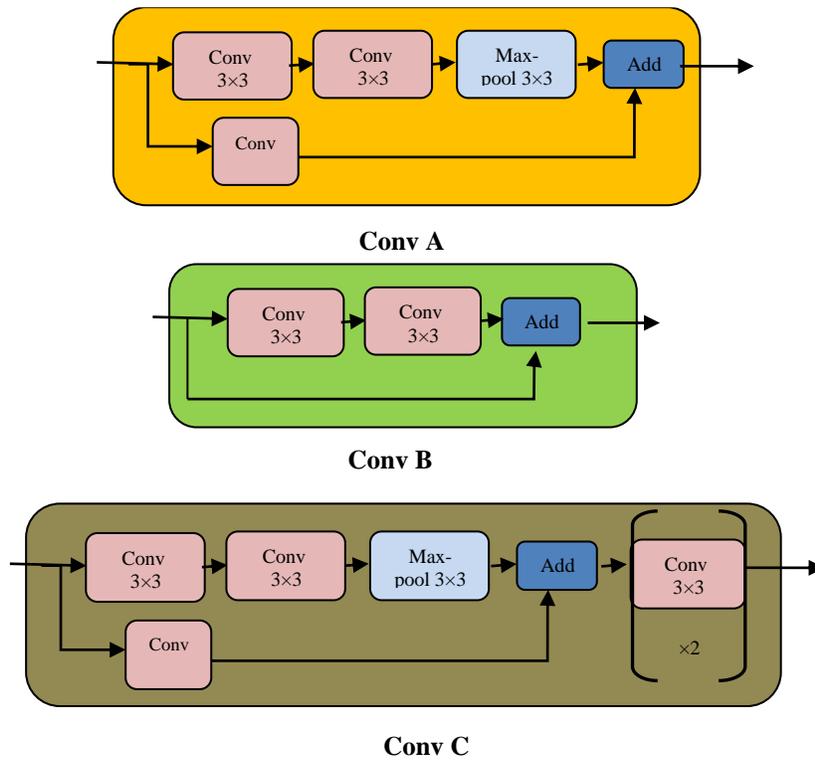
**Figure B.8:** Architecture ResNet.



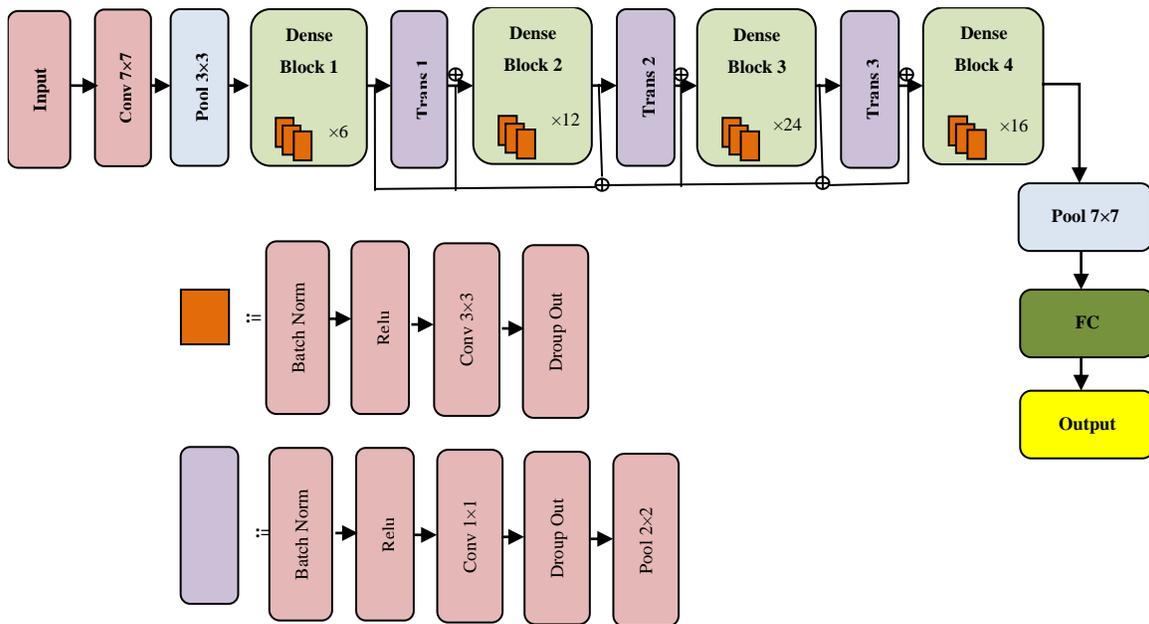
**Figure B.9 :** Block Identity et convolution de ResNet.



**Figure B.10:** Architecture Xception.



**Figure B.11:** Conv A, B, C de Xception.



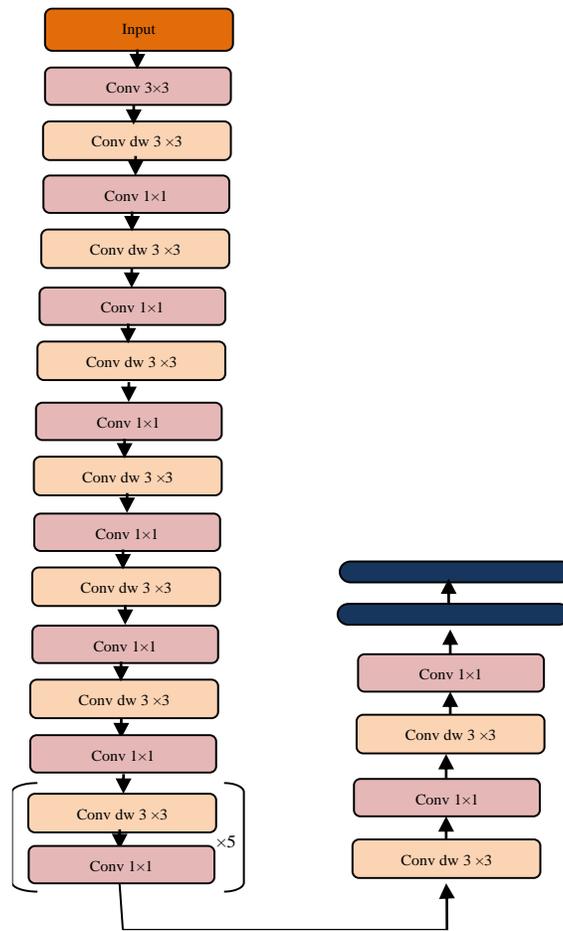


Figure B.13 : Architecture MobileNet-v1.

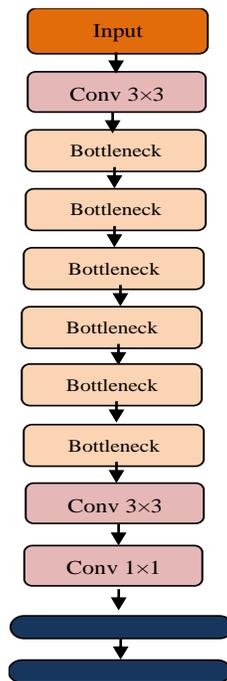
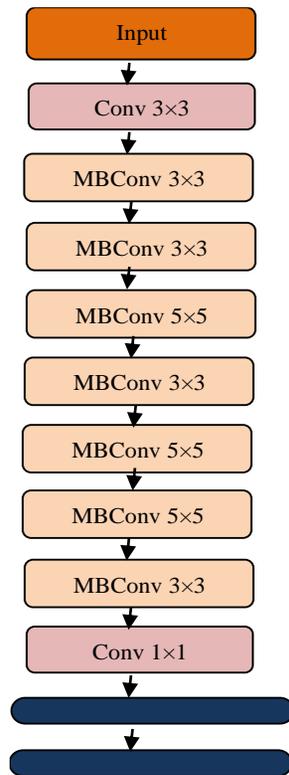
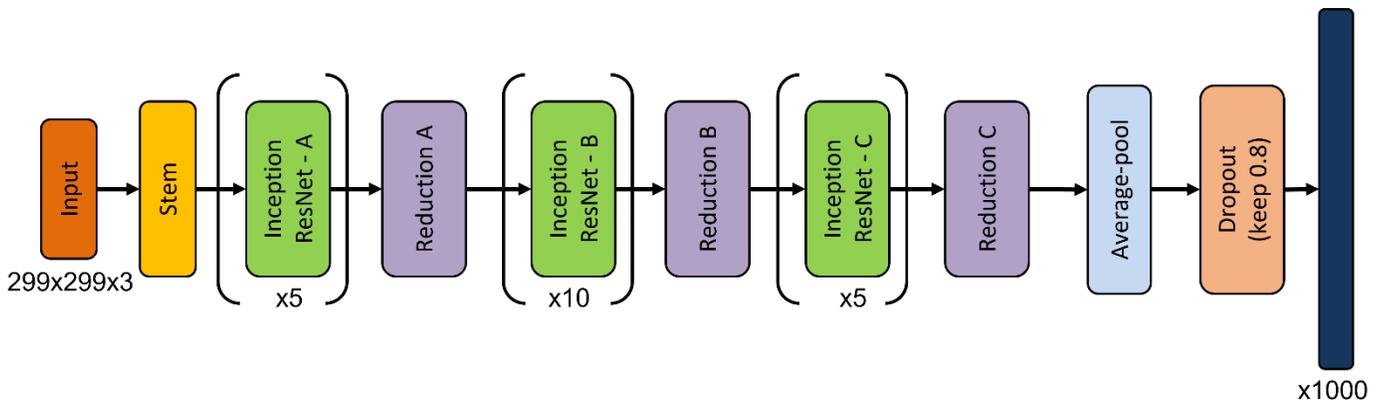


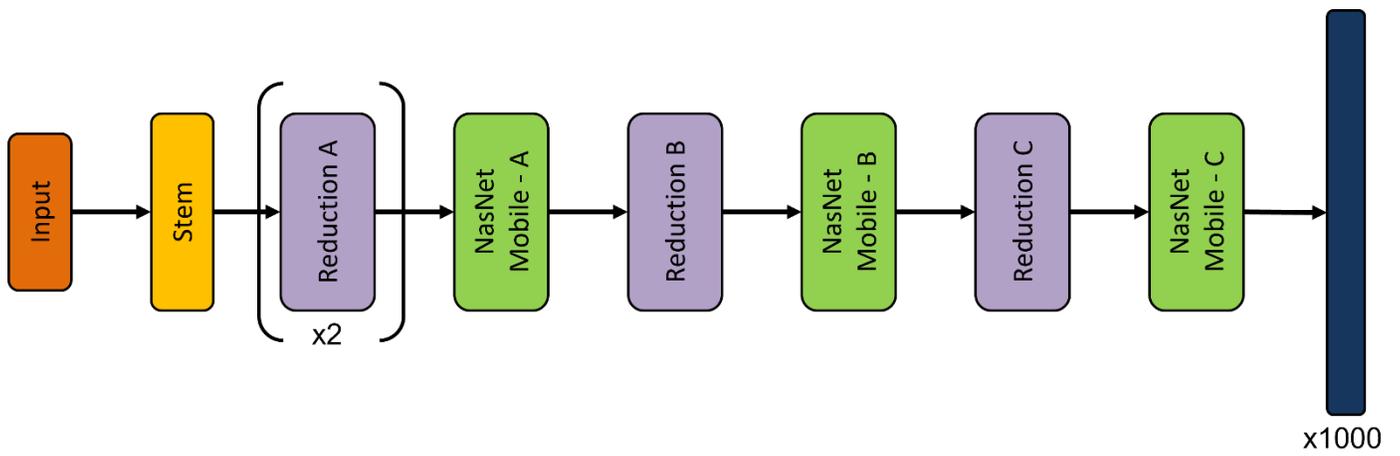
Figure B.14 : Architecture MobileNet-v2.



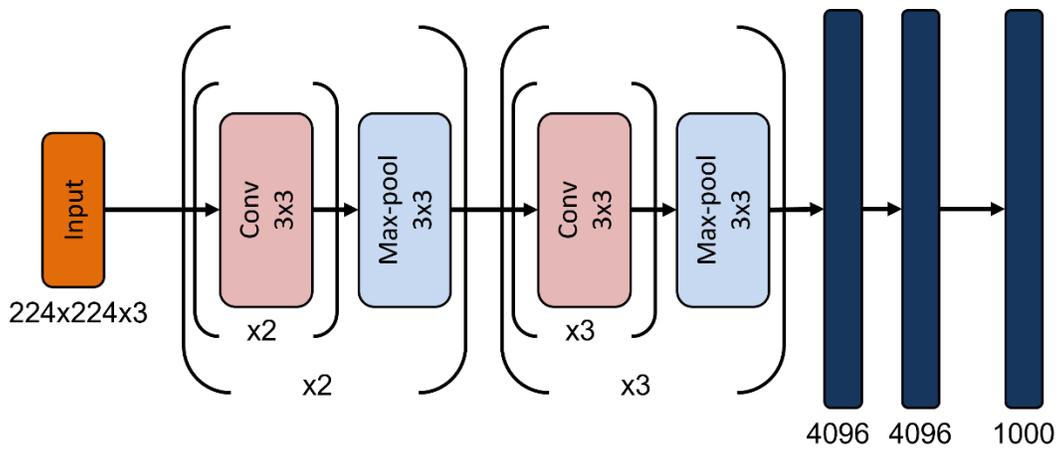
**Figure B.15 :** Architecture EfficientNet.



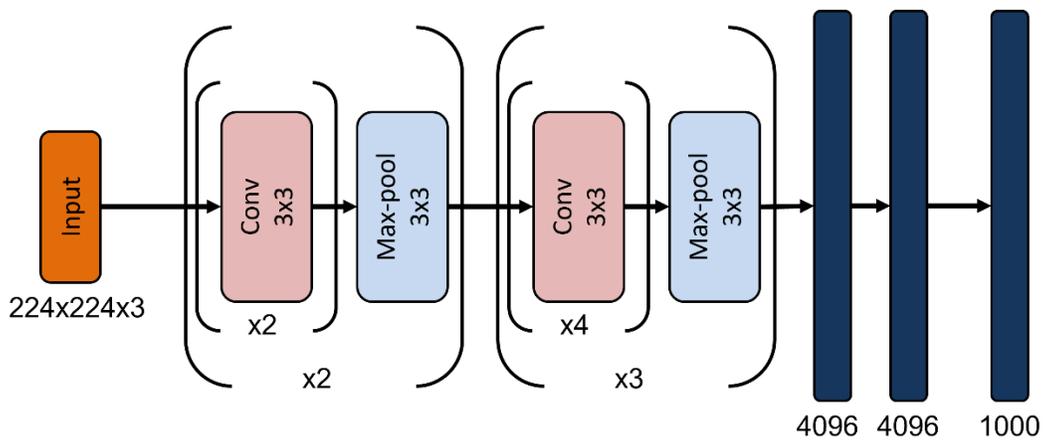
**Figure B.16 :** Architecture InceptionResNetV2.



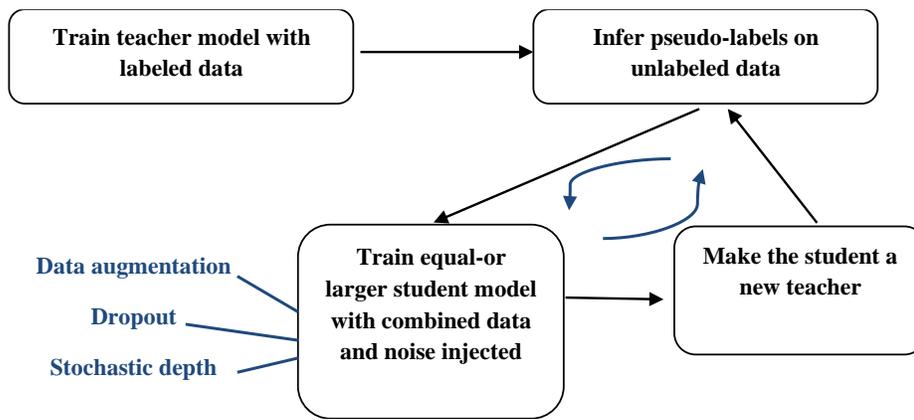
**Figure B.17** : Architecture NasNet Mobile.



**Figure B.17** : Architecture VGG16.



**Figure B.17** : Architecture VGG19.



**Figure B.20:** Architecture Noisy Student.

# Références

- [1] «Organization mondiale de la santé,» 07 12 2018. [En ligne]. Available: <https://www.who.int/fr/news-room/fact-sheets/detail/road-traffic-injuries>. [Accès le 03 07 2021].
- [2] NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION, et al., «Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey,» *Washington, DC: US Department of Transportation*, vol. 2, pp. pp. 1-2, 2015.
- [3] WILLIAMSON, Ann, LOMBARDI, A.David, FOLKARD et Simon, «The link between fatigue and safety,» *Accident Analysis & Prevention*, vol. 43, n° %12, pp. pp. 498-515., 2011,.
- [4] «"ROAD SAFETY FACTS," ASIRT,» 2018. [En ligne]. Available: <https://www.asirt.org/safe-travel/road-safety-facts/>. [Accès le 13 06 2021].
- [5] «Algérie Presse Service,» 10 02 2021. [En ligne]. Available: <https://www.aps.dz/societe/117350-accidents-bilan-2020-le-moins-tragique-depuis-trois-decennies>. [Accès le 2021 08 15].
- [6] M. Schrauf, M. Simon, E. Schmidt et W. Kincses, «Assessing Drivers' Fatigue State Under Real Traffic Conditions Using EEG Alpha Spindles,» chez *Proceedings of the 6th International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design : driving assessment 2011*, 2011.
- [7] «Facts and Stats - Drowsy Driving - Stay Alert, Arrive Alive,» May 2018. [En ligne]. Available: <https://drowsydriving.org/about/facts-and-stats/>. [Accès le 25 April 2021].
- [8] «securite-routiere.gouv,» [En ligne]. Available: <https://www.securite-routiere.gouv.fr/dangers-de-la-route/la-fatigue-et-la-conduite>. [Accès le 2021 06 13].
- [9] DAWSON, Drew, REID et Kathryn., «Fatigue, alcohol and performance impairment. *Nature*,» vol. 388, n° %16639, pp. 235-235, 1997.
- [10] A. E. Schmidt, M. Schrauf, M. Simon, M. Fritzsche, A. Buchner et W. E. Kincses, «Drivers' misjudgement of vigilance state during prolonged monotonous daytime driving,» *Accident Analysis & Prevention*, vol. 41, p. 1087–1093, 9 2009.
- [11] F. Vandemeulebroek, «Réaction des conducteurs face aux avertissements d'un détecteur «portatif » de somnolence,» l'institut Vias, Bruxelles, Belgique, 2017.

- [12] W. Boucsein, A. Haarmann et F. Schaefer, «Combining Skin Conductance and Heart Rate Variability for Adaptive Automation During Simulated IFR Flight,» p. 639–647, 2007.
- [13] M. García-García, A. Caplier et M. Rombaut, «Sleep Deprivation Detection for Real-Time Driver Monitoring Using Deep Learning,» chez *Lecture Notes in Computer Science*, Springer International Publishing, 2018, p. 435–442.
- [14] ARNETT, J.Todd, WILDE, JS.Gerald, MUNT, W.Peter et al., «How do prolonged wakefulness and alcohol compare in the decrements they produce on a simulated driving task? Accident Analysis & Prevention,» *Accident Analysis & Prevention*, vol. 33, n° 13, p. 337–344, 2001.
- [15] Nikolskaia, Kseniia, Bessonov, Vladislav, Starkov, Artem, Minbaleev et Aleksey, «Prototype of Driver Fatigue Detection System Using Convolutional Neural Network,» chez *2019 International Conference "Quality Management, Transport and Information Security, Information Technologies" (IT&QM&IS)*, 2019.
- [16] CAFFIER, P.Philipp, ERDMANN, Udo, ULLSPERGER et Peter, «Experimental evaluation of eye-blink parameters as a drowsiness measure,» *European journal of applied physiology*, vol. 89, n° 13, pp. 319-325, 2003.
- [17] L. A. Kaddour Khaled, «Développement d'une Application de Détection Somnolence au Volant du Conducteur Pour Eviter les Accidents de la Route,» 2019.
- [18] J. Krajewski, D. Sommer, U. Trutschel, D. Edwards et M. Golz, «Steering Wheel Behavior Based Estimation of Fatigue,» *Driving assessment conference*, 2009.
- [19] F. Lauer et G. Bloch, «Incorporating Prior Knowledge in Support Vector,» *Neurocomputing, Elsevier*, vol. 71, pp. 1578-1594, 2007.
- [20] Z.Zhang, «Introduction to machine learning: K-nearest neighbors,» *Annals of translational Medicine*, vol. 4, p. 218, 2016.
- [21] Ablamvi. AMEYOE, «Estimation de la distraction fondée sur un modèle dynamique de conducteur: principes et algorithmes. 2016,» Thèse de doctorat. Ecole des Mines, Nantes, 2016.
- [22] Y. Liang et J. D. Lee, «Combining Cognitive and Visual Distraction: Less Than the Sum of Its Parts,» *Accident Analysis & Prevention*, vol. 42, n° 13, pp. 881-890, 2010.
- [23] H. Zhang and M. Smith, «"A Literature Review of Visual Distraction Research,"» *SAVE-IT: Safety Vehicles using adaptive Interface Technology (Task7)*, Delphi Electronics & Safety, 2004.

- [24] G. Hu.Shuyan, «Driver Drowsiness Detection with Eyelid related Parameters by Support Vector Machine,» *Expert Systems with Applications*, vol. 36, pp. 7651-7658, 2009.
- [25] P.Antoine, C.Sylvie et A.Caplier, «EOG-based drowsiness detection Comparison between a fuzzy system and two supervised learning classifiers,» *The International Federation of Automatic Control*, 2011.
- [26] CHUA, Chern-Pin, MCDARBY, Gary, HENEGHAN et Conor., «Combined electrocardiogram and photoplethysmogram measurements as an indicator of objective sleepiness,» *Physiological measurement*, vol. 29, n° %18, p. 857, 2008.
- [27] «Institute of HeartMath,» Science of heart: Exploring the role of the heart in human performance, 2007. [En ligne]. Available: <https://www.heartmath.org/resources/downloads/science-of-the-heart/>. [Accès le 2021 07 12].
- [28] HORNE, A.Jim, A. Louise et REYNER., «Sleep related vehicle accidents,» *BMJ*, vol. 310, n° %16979, pp. 565-567., 1995.
- [29] C. Berka, D. J. Levendowski, M. N. Lumicao, A. Yau, G. Davis, V. T. Zivkovic, R. E. Olmstead, P. Tremoulet et P. L. Craven., «EEG Correlates of Task Engagement and Mental Workload in Vigilance, Learning, and Memory Tasks,» *Aviation, space, and environmental medicine*, vol. 78, n° %15, pp. 231-244, 2007..
- [30] S.Arun, S.Kenneth et M.Murugappan, «Detecting Driver Drowsiness Based on Sensors: A Review,» *Sensors (Basel, Switzerland)*doi: 10.3390/s121216937, vol. 12, pp. 16937-16953, 2012.
- [31] Grandchamp et Romain, «Electroencéphalographie et Interfaces Cerveau-Machine,» *Neurosciences. Université Paul Sabatier - Toulouse III*, 2012.
- [32] C.Ahlstrom, T.Victor, C.Wege et E.Steinmetz, «Processing of eye/head-tracking data in largescale naturalistic driving data sets,» *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, n° %12, pp. 553-564, 2012.
- [33] TRUTSCHEL, Udo, SIROIS, Bill, SOMMER, David et al, «PERCLOS: An alertness measure of the past,» *In : Proceedings of the Sixth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, pp. 172-179, 2011.
- [34] DINGES, F.David, GRACE et Richard., «PERCLOS: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance,» *US Department of Transportation, Federal Highway Administration, Publication Number FHWA-MCRT-98-006*, 1998.
- [35] I. García, S. Bronte, L. M. Bergasa, J. Almazán et J. Yebes, «Vision-based drowsiness detector for real driving conditions,» *IEEE Intelligent Vehicles Symposium*, pp. 618-623,, 2012.

- [36] W.Zhang, B.Cheng et Y.Lin, «Driver drowsiness recognition based on computer vision technology,» *Tsinghua Science and Technology*, vol. 17, pp. 354-362, 2012.
- [37] J. D. NAUROIS et Charlotte, «Modèles de détection et de prédiction de la somnolence au volant pour des systèmes personnalisés d'aide à la conduite,» Thèse de doctorat, Aix-Marseille., 2018.
- [38] M. Ochocki et D. Sawicki, «Yawning Recognition based on Dynamic Analysis and Simple Measure,» in *Proceedings of the International Conference on Computer-Human Interaction Research and Applications*, pp. 111-117, 2017.
- [39] L. M. BERGASA, J. NUEVO, SOTELO, M. A. et al, «Real-time system for monitoring driver vigilance,» *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, n° %11, pp. 63-77, 2006.
- [40] A.Katherine.KAPLAN, Anna.ITOI et C.William.DEMENT, «Awareness of sleepiness and ability to predict sleep onset: can drivers avoid falling asleep at the wheel,» *Sleep Medicine*, vol. 9, n° %11, pp. pp. 71-79., 2007.
- [41] JI, Qiang, ZHU, Zhiwei, LAN et Peilin., «Real-time nonintrusive monitoring and prediction of driver fatigue,» *IEEE transactions on vehicular technology*, vol. 53, n° %14, pp. 1052-1068, 2004.
- [42] MAY, F.Jennifer et L. 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, n° %13, 2009.
- [43] Taub et A.Eric, «Sleepy Behind the Wheel? Some Cars Can Tell - The New York Times,» 16 March 2017. [En ligne]. Available: [www.nytimes.com/2017/03/16/automobiles/wheels/drowsy-driving-technology.html](http://www.nytimes.com/2017/03/16/automobiles/wheels/drowsy-driving-technology.html). [Accès le 05 May 2021].
- [44] «Driver Alert Control | Driver Alert Control | Assistance à la conduite | XC90 2020 Late | Assistance Volvo,» 07 November 2020. [En ligne]. Available: <https://www.volvocars.com/ma/support/manuals/xc90/2019w46/assistance-a-la-conduite/driver-alert-control/driver-alert-control>. [Accès le 05 May 2021].
- [45] «How Does ATTENTION ASSIST® Work?,» [En ligne]. Available: <https://www.waltersmercedesbenzofriverside.com/how-does-attention-assist-work/>. [Accès le 05 May 2021].
- [46] «Driver Fatigue Detection | Miles Continental,» [En ligne]. Available: <https://www.milescontinental.co.nz/news/features/driver-fatigue-detection/>. [Accès le 05 May 2021].

- [47] «VW Driver Assistance | Lane Assist & Driver Alert System | Volkswagen Malaysia,» [En ligne]. Available: <https://www.volkswagen.com.my/technology/driver-assistance>. [Accès le 05 May 2021].
- [48] «Driver Alert System | Volkswagen Newsroom,» [En ligne]. Available: <https://www.volkswagen-newsroom.com/en/driver-alert-system-3932>. [Accès le 05 May 2021].
- [49] «Alerte Attention Conducteur - Forum-Peugeot.com,» [En ligne]. Available: <https://www.forum-peugeot.com/technologies/alerte-attention-conducteur/>. [Accès le 05 May 2021].
- [50] «StopSleep : the best solution against drowsiness,» [En ligne]. Available: <https://www.stopsleep.co.uk/>. [Accès le 05 May 2021].
- [51] «MiniTOUCANGO, vigilant driving | Toucango,» [En ligne]. Available: <https://toucango.com/en/minitoucango-product/>. [Accès le 05 May 2021].
- [52] C. Fors, C. Ahlström, P. Sörner, J. Kovaceva, E. Hasselberg, M. Krantz, J. F. Grönvall, K. Kircher et A. Anund, «Camera based sleepiness detection,» *Final report of the project SleepEYE*, 2011.
- [53] K. Kircher, C. Ahlstrom et A. Kircher, «Comparison of two eye-gaze based real-time driver distraction detection algorithms in a small-scale field operational test,» in *Proc. 5th Int. Driving Symp Human Factors Driver Assessment, Train. Veh. Des*, pp. 16-23, 2009.
- [54] J. McCarthy, «WHAT IS ARTIFICIAL INTELLIGENCE?,» *Stanford University*, pp. 2-3, 1998.
- [55] T. Lewis, «A Brief History of Artificial Intelligence | Live Science,» Live Science, 04 December 2014. [En ligne]. Available: <https://www.livescience.com/49007-history-of-artificial-intelligence.html>. [Accès le 07 April 2021].
- [56] I. Goodfellow, Y. Bengio et A. Courville, *Deep Learning*, MIT Press, 2016.
- [57] J. A. Nichols, H. W. Herbert Chan and M. A. B. Baker, "Machine learning: applications of artificial intelligence to imaging and diagnosis,," *Biophysical reviews*, vol. 11, no. 1, pp. 111-118, 2 2019.
- [58] «What is Machine Learning? | IBM,» IBM, 15 July 2020. [En ligne]. Available: <https://www.ibm.com/cloud/learn/machine-learning#toc-deep-learn-nOh7s5Rf>. [Accès le 21 April 2021].
- [59] S. RAY, «Commonly used Machine Learning Algorithms | Data Science,» 09 September 2017. [En ligne]. Available: <https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/>. [Accès le 21 April 2021].

- [60] N. K. Chauhan et K. Singh, «A Review on Conventional Machine Learning vs Deep Learning,» chez 2018 *International Conference on Computing, Power and Communication Technologies (GUCON)*, 2018.
- [61] Sulis.Setiowati, Zulfanahri, Eka.Franita et Igi.Ardiyanto, «A review of optimization method in face recognition: Comparison deep learning and non-deep learning methods,» 2017.
- [62] M. HARGRAVE, «Deep Learning Definition,» 06 April 2021. [En ligne]. Available: <https://www.investopedia.com/terms/d/deep-learning.asp>. [Accès le 23 April 2021].
- [63] W. S. McCulloch et W. Pitts, «A logical calculus of the ideas immanent in nervous activity,» *The Bulletin of Mathematical Biophysics*, vol. 5, p. 115–133, 12 1943.
- [64] I. Stoianov, «Connectionist lexical processing,» 9 2001.
- [65] Imane.Nedjar, «Medical images indexation and annotation [Thèse de doctorat, Université de Tlemcen],» Tlemcen, 2015.
- [66] L. Tucci, «What is artificial neuron? - Definition from WhatIs.com,» May 2018. [En ligne]. Available: <https://searchcio.techtarget.com/definition/artificial-neuron>. [Accès le 21 April 2021].
- [67] M. Robins, «The Difference Between Artificial Intelligence, Machine Learning and Deep Learning,» Intel, 27 May 2020. [En ligne]. Available: <https://www.intel.com/content/www/us/en/artificial-intelligence/posts/difference-between-ai-machine-learning-deep-learning.html>. [Accès le 22 April 2021].
- [68] F. Rosenblatt, «The perceptron: A probabilistic model for information storage and organization in the brain,» *Psychological Review*, vol. 65, p. 386–408, 1958.
- [69] J. Kurata, «Deep Learning with Keras | Pluralsight,» 26 March 2018. [En ligne]. Available: <https://www.pluralsight.com/courses/keras-deep-learning>. [Accès le 09 May 2021].
- [70] B. Mohammed et B. Brahim, «L`apprentissage profond (Deep Learning) pour la classification et la recherche d'images par le contenu,» 2017.
- [71] J. Brownlee, *Master Machine Learning Algorithms: Discover How They Work and Implement Them From Scratch, Machine Learning Mastery*, 2016.
- [72] A. Tch, «The mostly complete chart of Neural Networks, explained | by Andrew Tch | Towards Data Science,» 04 August 2017. [En ligne]. Available: <https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464>. [Accès le 22 April 2021].

- [73] T. Dettmers, «Deep Learning in a Nutshell: Core Concepts | NVIDIA Developer Blog,» NVIDIA, 03 November 2015. [En ligne]. Available: <https://developer.nvidia.com/blog/deep-learning-nutshell-core-concepts/>. [Accès le 22 April 2021].
- [74] K. Fukushima, S. Miyake et T. Ito, «Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,» *Biological Cybernetics*, vol. 36, pp. 93-202, 1980.
- [75] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard et L. D. Jackel, «Backpropagation applied to handwritten zip code recogniti,» *Neural computation*, vol. 1, n° %14, pp. 541-551, 1989.
- [76] Y. LeCun, L. Bottou, Y. Bengio et P. Haffner, «Gradient-based learning applied to document recognition,» *Proceedings of the IEEE*, vol. 86, n° %111, pp. 2278-2324, 1998.
- [77] V.Rajan, «Towards Efficient Intrusion Detection using Deep Learning Techniques: A Review,» *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 6, 2017.
- [78] Sohaib, Asif et K.Amjad, «Automatic COVID-19 Detection from chest radiographic images using,» *medRxiv preprint doi: <https://doi.org/10.1101/2020.11.08.20228080>*, 2020.
- [79] Wiesel et Hubel, «Receptive fields of single neurones in the cat's striate cortex,» *The Journal of Physiology*, vol. 148, n° %13, p. 574–591, 1959.
- [80] Y. LeCun et al, «Handwritten digit recognition with a backpropagation network,» in *Advances in neural information processing systems*, pp. 396-404, 1990.
- [81] Y. LeCun et al, «Gradient-based learning applied to document recognition,» *Proceedings of the IEEE*, vol. 86, pp. 2278-2324, 1998.
- [82] KRIZHEVSKY, Alex, SUTSKEVER, Ilya, HINTON et E.Geoffrey, «Imagenet classification with deep convolutional neural networks,» in *Advances in neural information processing systems*, pp. 1097-1105, 2012.
- [83] K.Simonyan et A.Zisserman, «Very deep convolutional networks for large-scale image recognition,» *arXiv preprint arXiv :1409.1556*, 2014.
- [84] C.Szegedy, W.Liu, Y.Jia, P.Sermane, S.Reed, D.Anguelov, D.Erhan, V.Vanhoucke et A.Rabinovich, «Going deeper with convolutions,» in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1-9, 2015.
- [85] C.Szegedy, V.Vanhoucke, S.Ioffe, J.Shlens et Z.Wojna, «Rethinking the inception architecture for computer vision,» in *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 2818–2826, 2016.

- [86] HE, Kaiming, ZHANG, Xiangyu, REN, Shaoqing et al., «Deep residual learning for image recognition,» in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2016.
- [87] SCHROFF, Florian, KALENICHENKO, Dmitry, PHILBIN et James, «FaceNet : A Unified Embedding for Face Recognition and Clustering,» in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *arXiv preprint arXiv :1503.03832*, 2015.
- [88] CHOLLET et François., «Xception: Deep learning with depthwise separable convolutions,» in *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 1251–1258, 2017.
- [89] L.Sifre et S.Mallat, «Rigid-motion scattering for image classification,» *arXiv* 2014. *arXiv preprint arXiv:1403.1687.*, 2014.
- [90] G. Huang, Z. Liu, L. V. D. Maaten et K. Q. Weinberger, «Densely connected convolutional networks,» in *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 4700–4708, 2017.
- [91] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto et H. Adam, «Mobilenets: Efficient convolutional neural networks for mobile vision applications,» *arXiv preprint arXiv:1704.04861*, 2017.
- [92] M.Sandler, A.Howard, M.Zhu, A.Zhmogino et L.Chen, «Mobilenetv2 : Inverted residuals and linear bottlenecks,» in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 4510–4520, 2018.
- [93] M.Tan et Q. V.Le, «Efficientnet: Rethinking model scaling for convolutional neural networks,» *arXiv preprint arXiv:1905.11946*, 2019.
- [94] XIE, Qizhe, LUONG, Minh-Thang, HOVY, Eduard et al, «Self-training with noisy student improves imagenet classification,» *arXiv preprint arXiv :1911.04252*, 2019.
- [95] Pundlik, Shrinivas, Singh, Anikait, Baghel, Gautam, Baliutaviciute, Vilte, Luo et Gang, «A Mobile Application for Keyword Search in Real-World Scenes,» *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 7, p. 1–10, 2019.
- [96] «Deep Learning for Speech Recognition | By ODSC - Open Data Science | Medium,» Open Data Science, 12 August 2019. [En ligne]. Available: <https://medium.com/@ODSC/deep-learning-for-speech-recognition-cbbebab15f0d>. [Accès le 23 April 2021].
- [97] G.Skinner et T.Walmsley, «Artificial Intelligence and Deep Learning in Video Games A Brief Review,» chez *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, 2019.

- [98] J.Hruska, «The Quest to Improve Video Game AI,» PC Magazine, 2016.
- [99] V. E. BALAS, S. S. ROY, D. SHARMA et al, Handbook of Deep Learning Applications, V. E. Balas, S. S. Roy, D. Sharma et P. Samui, Éds., Springer International Publishing, 2019.
- [100] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi et B. Hariri, *YawDD: Yawning Detection Dataset*, IEEE DataPort, 2020.
- [101] Y. BECKERL• et B. SAVAS, «Real Time Driver Fatigue Detection System Based on Multi-Task ConNN,» *Computer Engineering Department, Kocaeli University, 41380 Izmit, Turkey*, vol. 8, pp. 12491-12498, 2020.
- [102] «ImageNet,» 11 Mars 2021. [En ligne]. Available: <https://image-net.org/>. [Accès le 16 July 2021].
- [103] «Classification: Accuracy | Machine Learning Crash Course,» [En ligne]. Available: <https://developers.google.com/machine-learning/crash-course>. [Accès le 16 July 2021].
- [104] P.Jones et M. J.Viola, «Robust real-time object detection,» *International Journal of Computer Vision*, vol. 4, pp. pp 34-47, Jul. 2001.
- [105] Y. Freund et R. E. Schapire, «Experiments with a new boosting algorithm,» *Proc. ICML*, vol. 96, pp. 148-156, Jul. 1996.
- [106] «Java Point,» [En ligne]. Available: <https://www.javatpoint.com/history-of-artificial-intelligence>.
- [107] WILLIAMSON, Ann, LOMBARDI, A.David, FOLKARD et Simon, «The link between fatigue and safety,» *Accident Analysis & Prevention*, vol. 43, n° %12, pp. 498-515, 2011.

## Résumé

La fatigue du conducteur diminue sa vigilance ce qui peut mener à des conséquences graves comme les accidents routiers.

Dans ce mémoire nous avons proposé une méthode non intrusive de la détection de la fatigue du conducteur basé sur l'intelligence artificielle et la vision par ordinateur.

En effet, nous avons réalisé une étude comparative entre les différentes architectures du réseau neurone convolutive pour la détection de la fatigue au volant en analysant les caractéristiques faciales du conducteur (l'état des yeux et le bâillement). Par la suite, l'architecture sélectionnée a été utilisée pour calculer les deux mesures PERCLOS et FOM (PERcentage of eye CLOsure and yawning Frequency Of Mouth) afin de prédire la somnolence du conducteur. La méthode a obtenu un taux de détection égale à 81,25 %.

**Mots clés :** Fatigue, Analyse des Caractéristiques Faciales, Apprentissage Profond, Réseau de Neurones Convolutive, PERCLOS, FOM.

---

## Abstract:

### Comparative study between different architectures of convolutional neural networks (CNNs) for driver fatigue detection

Driver fatigue reduces vigilance which can lead to significant consequences such as road traffic crashes.

In this thesis, we have proposed a non-intrusive method of driver fatigue detection based on artificial intelligence and computer vision.

Thus, we carried out a comparative study between the different architectures of the convolutional neuron network for driver fatigue detection, by analyzing the facial characteristics of the driver (eye condition and yawning).

Subsequently, the selected architecture was used to calculate the two measurements PERCLOS and FOM (PERcentage of eye CLOsure and yawning Frequency Of Mouth) to predict driver drowsiness. The method obtained a detection rate equal to 81.25%.

**Keywords:** Fatigue, Facial Feature Analysis, Deep Learning, Convolutional Neural Network, PERCLOS, FOM.

---

## ملخص:

يقلل إجهاد السائق من يقظته مما قد يؤدي إلى عواقب وخيمة مثل حوادث الطرق. في هذه المذكرة اقترحنا طريقة غير تدخلية لاكتشاف إجهاد السائق تعتمد على الذكاء الاصطناعي ورؤية الحاسوب.

في البداية، أجرينا دراسة مقارنة بين مختلف الشبكات العصبية التلافيفية للكشف عن إرهاق السائق من خلال تحليل خصائص وجه السائق (حالة العين والتأؤب). بعد ذلك، استخدمنا الشبكة المختارة لحساب قياسين PERCLOS و FOM (PERcentage of eye CLOsure and Frequency Of Mouth) من أجل التنبؤ بنعاس السائق. حصلت الطريقة على معدل كشف يساوي 81.25%.

الكلمات المفتاحية: التعب، تحليل خصائص الوجه، التعلم العميق، شبكة الخلايا العصبية التلافيفية، PERCLOS، FOM.