

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : Sciences de L'ingénierat

Département : Electronique

Domaine : Sciences et Techniques

Filière : Electronique

Spécialité : Instrumentations

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

**SYSTEME DE DETECTION DE SOMNOLENCE POUR
CONDUCTEURS UTILISANT RASPBERRY-PI ET OPEN-CV**

Présenté par : *HACHEMANE Abdelhalim*

Encadrant : *FEZARI Mohamed* Professeur Université Badji Mokhtar Annaba

Jury de Soutenance :

BENSAKER Bachir	Professeur	Université Badji Mokhtar Annaba	Président
FEZARI Mohamed	Professeur	Université Badji Mokhtar Annaba	Encadrant
BENOUARET Mohamed	professeur	Université Badji Mokhtar Annaba	Examineur

Année Universitaire : 2020/2021

REMERCIEMENTS

Je remercie avant tous dieu qui m'as donné le courage et la volonté pour achever ce travail.
Je remercie profondément mon encadreur M. FEZARI Mohamed de m'avoir guidé avec patience, et pour tous ses conseils, orientations et corrections.

Je remercie chaleureusement et avec gratitude mes enseignants.

Mes remerciements sont adressés aussi aux membres du jury qui ont bien voulu nous honorer par leur présence.

Je remercie mon frère et mon bras droit Riad Hachemane pour être toujours a mes cotés.

Mes vifs remerciements à ma familles de m'avoir soutenu pendant mon cursus universitaire.

Enfin, un grand merci à tous mes amis qui m'ont apporté de l'aide de près ou de loin sans exception.

DEDICACES

Moi HACHEMANE Abdelhalim,

À mon père

À ma mère

À mon frère

À ma sœur

À toute ma famille

À tous mes amis

À tous ceux qui me sont chers

Je dédie ce travail.

RESUME

La conception et la réalisation d'un système de détection de somnolence (État intermédiaire entre la veille et le sommeil) pour les conducteurs fait l'objet de ce mémoire.

L'objectif de ce système est de détecter la somnolence des conducteurs et de le signaler via une alerte vocale et une sirène. Le système envoie ensuite des informations concernant l'état du conducteur au Cloud ce qui permet de les consulter et de suivre l'état des conducteurs en temps réel depuis n'importe quel endroit dans le monde.

Le système repose sur une carte Raspberry Pi (Ordinateur) qui contrôle tout le système et qui reçoit l'information de la camera puis fait un traitement d'image et détecte la somnolence grâce à l'intelligence artificielle pour ensuite activer l'alarme et alerter le conducteur via un hautparleur.

Dans le premier chapitre on présente l'avancée des recherches scientifiques dans ce domaine et quelques travaux réalisés par des chercheurs des quatre coins du globe, puis l'étude théorique de la détection du visage et des yeux par traitement d'image, et enfin on présente les objectifs de notre projet.

Dans le second chapitre on présente un synoptique du système que nous avons réalisé et une description des ressources hardware et software utilisées, ensuite une description de l'algorithme que nous avons développé et l'application web utilisée pour afficher les résultats.

Enfin, le dernier chapitre présente les tests effectués, leurs résultats et leurs discussion, on fini par une conclusion sur ces résultats et on donne aussi des perspectives pour l'avenir de ce projet.

Le mémoire se termine par une conclusion générale qui fait la synthèse des résultats obtenus.

Mots-clés : somnolence des conducteurs, système de détection, système d'alerte.

ABSTRACT

The design and construction of a drowsiness detection system for drivers (Intermediate state between wakefulness and sleep) is the subject of this thesis.

The purpose of this system is to detect driver sleep and report it via a voice alert and a siren. The system then sends the informations of driver status to Cloud, allowing it to be viewed and tracked in real time from anywhere in the world.

The system is based on a Raspberry Pi card (Computer) which controls the entire system, the card receives information from the camera, performs image processing and detects drowsiness using artificial intelligence, then activates the alarm that alerts the driver via a loudspeaker.

The first chapter presents the progress of scientific research in this field, some work carried out by researchers from all over the world, then the theoretical study of detection of the face and eyes by image processing and finally the objectives of our project.

In the second chapter we present a synoptic of the system that we have produced and a description of the hardware and software resources used, then we introduce the algorithm that we developed and the web application used to display the results.

The last chapter presents the effectuated tests, their results and their discussion, we end up with a conclusion related to these results and some perspectives for the future of this project.

The thesis ends with a general conclusion which summarizes the results obtained.

Keywords: drivers drowsiness, detection system, alert system.

ملخص

موضوع هذه الأطروحة هو تصميم وإنجاز نظام لأمن الطرقات و هو كشف نعاس السائقين (و هو حالة بين اليقظة و النوم).

الهدف من هذا النظام هو كشف نعاس السائق و الإبلاغ عنه عبر تنبيه صوتي و صفارة إنذار. النظام يرسل بعدها معلومات حول حالة السائق إلى السحابة (Cloud) مما يسمح بمتابعة حالة السائقين على المباشر و من أي مكان في العالم.

يعتمد النظام على لوحة Raspberry Pi (حاسوب) التي تتحكم بالنظام بأكمله، تتلقى هذه اللوحة معلومات من الكاميرا ثم تقوم بمعالجة الصور و كشف النوم باستخدام الذكاء الاصطناعي لتفعيل التنبيه الصوتي عبر مكبر الصوت.

الفصل الأول يستعرض تقدم البحث العلمي في هذا المجال و بعض الأعمال التي قام بها باحثون من شتى أنحاء العالم، ثم الدراسة النظرية لاكتشاف حالة الوجه و العينين و معالجة الصور و في آخره أهداف مشروعنا.

الفصل الثاني يقدم موجزا شاملا للنظام الذي قمنا بتصميمه و وصفا لموارد الأجهزة و البرامج المستخدمة، ثم وصفا للخوارزمية التي قمنا بتطويرها و تطبيق الويب المستخدم لعرض النتائج.

الفصل الأخير يعرض الاختبارات التي أجريناها و نتائجها و كذا نقاشها، و ينتهي بخاتمة حول هذه النتائج و وجهات نظر لمستقبل هذا المشروع.

تنتهي الأطروحة بخاتمة عامة تلخص النتائج التي حصلنا عليها.

LISTE DES ABRÉVIATIONS

DNSR: Délégation Nationale de la Sécurité Routière

GSM: Global System for Mobile Communications

SWM: Steering Wheel Movement

SDLP: Standard Deviation of Lane Position

3D: Three Dimension

SQL: Structured Query Language

SBC: Single-Board Computer

RAM: Random Access Memory

GB: Giga Byte

LPDDR: Low-Power Double Data Rate

USB: Universal Serial Bus

HDMI: High-Definition Multimedia Interface

Wifi: Wireless Fidelity

IEEE: Institute of Electrical and Electronics Engineers

OpenCV: Open Source Computer Vision

API: Application Programming Interface

Dlib: Digital Library

Pytsx3: Python Text To Speech Version 3

MB: Mega Byte

LED: Light Emitting Diode

IR: Infrared

GPS: Global Positioning System

UART: Universal Asynchronous Receiver Transmitter

AGPS: Assisted Global Positioning System

SMS: Short Message Service

ML: Machine Learning

LISTE DES FIGURES

Figure 1: Diverses techniques pour détecter la somnolence	3
Figure 2: Prototype de l'œuvre proposée	4
Figure 3: Système en cours d'exécution.....	4
Figure 4: Détection du visage par la camera du Smartphone	5
Figure 5: Illustration du fonctionnement du système	5
Figure 6: Détecteur de ligne sur véhicule	6
Figure 7: Exemples de détection de visages	7
Figure 8: Illustration des différentes boîtes englobantes pour la détection des yeux	8
Figure 9: La Méthode de détection de l'œil.....	9
Figure 10: Synoptique du système.....	10
Figure 11: Raspberry Pi 4 model B.....	11
Figure 12: module camera pour Rspberry Pi	12
Figure 13: Haut parleur.....	12
Figure 14: Modem 4G - WIFI.....	13
Figure 15: chargeur voiture.....	13
Figure 16: Exemple de détection de visage avec une cascade de classifieur.....	15
Figure 17: Caractéristique des yeux haar-cascade	16
Figure 18: Caractéristique de Haar	16
Figure 19: exemples de détections avec OpenCV et Dlib	17
Figure 20: Python text to speech.....	17
Figure 21: Détection du visage et des yeux	18
Figure 22: Diagramme d'activité de l'algorithme réalisé	19
Figure 23: Synoptique de l'application web.....	22
Figure 24: Les forfaits Heroku.....	23
Figure 25: Page d'accueil	23
Figure 26: Page d'informations sur l'état du conducteur	24
Figure 28: Les différentes conditions de tests.....	25
Figure 29: Tests jour et nuit	25
Figure 27: Dispositif en marche.....	26
Figure 30: Module camera infrarouge	27
Figure 31: module GPS.....	28

LISTE DES TABLEAUX

Tableau 1: Tests dans un environnement bien éclairé	25
Tableau 2: Tests dans un véhicule en marche.....	26

TABLE DES MATIERES

INTRODUCTION GENERALE	1
CHAPITRE 1	2
I- Introduction	2
II- Etat de l'art.....	3
1- Approche physiologique	4
2- Etude du comportement	5
3- Approche basée sur le véhicule.....	6
III- Problématique et solution proposé.....	7
IV- Etude théorique de la détection du visage et des yeux par traitement d'image.....	7
1- Détection du visage	7
2- Détection des yeux	8
V- Objectifs du projet.....	9
CHAPITRE 2	10
I- Synoptique du système	10
II- Description Hardware	11
1- Raspberry Pi	11
2- Module Camera	12
3- Haut-parleur	12
4- Modem 4G – WIFI.....	13
5- Chargeur voiture.....	13
III- Description Software	14
1- Python.....	14
2- Open CV.....	14
2-1- Description d'OpenCV.....	14
2-2- La Détection du visage avec le détecteur d'OpenCV	15
3- Haar-cascade	16
4- Dlib.....	17
5- Pyttsx3.....	17
IV- Description du programme développé.....	18
1- Fonctionnement du programme	18
2- les principales fonctions et instructions utilise.....	20
2-1- Initialisation du system.....	20

2-2- Capture Vidéo	20
2-3- Traitement de l'image	20
2-4- Alerte audio	20
2-5- Envoyer des informations vers le serveur:	21
V- Application Web	22
1- Création du l'application web	22
2- fonctionnement de l'application web	23
CHAPITRE 3	25
I- Tests et Résultats	25
1- Test dans un environnement éclairé	25
2- Test dans un véhicule en marche	25
3- Discussions des résultats obtenus	26
4- Conclusion	26
II- Perspectives	27
1- Utilisation d'une carte Raspberry plus performante	27
2- Utilisation d'une meilleur camera	27
3- ajout d'un module GPS	28
4- Optimisation de l'application web	28
CONCLUSION GENERALE	29

INTRODUCTION GENERALE

Les accidents de la circulation sont considérés comme les problèmes les plus graves pour la société et la sécurité dans la majorité des pays en ces temps modernes sans exception.

Dans notre pays le nombre de victimes des accidents de la route a considérablement augmenter ces dernières années et c'est du à plusieurs facteurs.

Dans ce travail de recherche nous proposons un système embarqué à base d'une carte Raspberry Pi et d'une camera, pour venir en aide aux conducteurs de bus ou de camions de transport des entreprises. Le système permet de détecter la fatigue et la somnolence des conducteurs par la saisie d'image et son traitement.

Le système embarqué permet de détecter le visage et les yeux. Selon l'état des yeux le système envoie un signal sonore et un message oral pour éveiller le conducteur.

Le système envoie aussi les données sur l'état du conducteur dans une page web sécurisé que le responsable de l'entreprise de transport peut consulter à tout moment.

CHAPITRE 1

I- Introduction

Selon les statistiques fournis par la délégation nationale de la Sécurité routière en Algérie, en 2019^[01] le nombre d'accidents de la route a atteint un chiffre effrayant qui dépasse les 22,507 cas. Avec 3,270 morts et 31,010 blaisés. En 2020^[02] le nombre d'accidents a baissé, 18,949 cas conduisant à 2,844 morts et 25,836 blaisés. La wilaya de Msila vient en tête du classement avec 798 accidents, suivis par la capitale Alger avec 779 sinistres.

D'après la DNSR les mercredis et jeudis sont les journées qui enregistrent le plus d'accidents, car précédant le week-end (31,49% du nombre global des accidents), alors que la journée du vendredi est celle qui enregistre le moins d'accidents vu le manque de déplacements.

L'excès de vitesse vient en tête des principales causes des accidents de la route. Il y a aussi d'autres facteurs très importants comme la fatigue, les malaises des personnes souffrants de maladie chroniques, la longueur du trajet, le non accompagnement du chauffeur et le sommeil au volant qui engendre des accidents mortels surtout pour les conducteurs de bus et de camions qui transportent les passagers et les marchandises pour des très longues distances et dans plusieurs cas sans avoir le repos nécessaire pour accomplir cette tâche qui demande beaucoup de concentration et de vigilance.

En effet, la principale raison de 20 % des accidents et de 30 % des accidents mortels dans le monde est causée par la somnolence des conducteurs et leur manque de concentration.^[03]

II- Etat de l'art

L'augmentation des accidents de la route dans le monde a motivée les chercheurs à trouver des solutions pour les réduire.

De nombreuses solutions différentes ont été proposées pour ce problème, certaines d'entre elles se contentent d'avertir le conducteur lors du sommeil, d'autres arrêtent automatiquement le véhicule et envoient des informations au Cloud. Cependant, la plupart des systèmes de détection de somnolence existants souffrent encore de certains inconvénients.

La plupart des systèmes ne peuvent détecter la somnolence d'une façon précise et émettent des faux avertissements aux conducteurs, tout en leur permettant d'ignorer les avertissements et continuer à conduire en état de somnolence. Dans ce cas, des accidents de la route peuvent encore se produire et les menaces imposées par ces conducteurs aux autres usagers de la route sont encore élevées. Dans la figure ci-dessous on présente les différentes techniques utilisées pour la détection du sommeil.

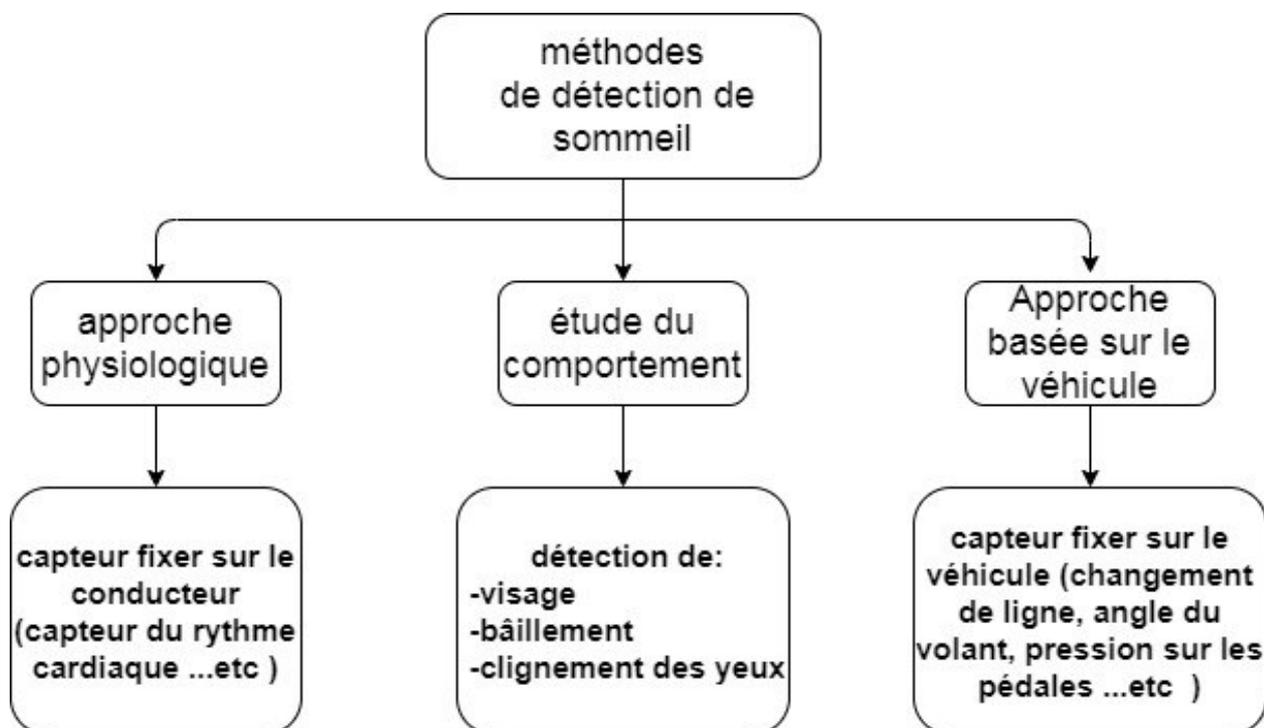


Figure 1: Diverses techniques pour détecter la somnolence

1- Approche physiologique

L'approche physiologique fait référence à la méthode de détection de la somnolence en utilisant des données recueillies par des capteurs fixés sur le corps du conducteur, tel que le capteur de rythme cardiaque.

Selon M. Arunasalam¹, N. Yaakob, A. Amir, M. Elshaikh, N. F. Azahar, 2019^[4], la somnolence peut être détectée en utilisant les données collectées par un capteur de rythme cardiaque et un capteur de clignement des yeux. Ces données sont traitées, ensuite, si la somnolence est détectée, la vitesse du véhicule sera réduite jusqu'à son arrêt complet, et l'état actuel et l'emplacement du conducteur sont à la fois suivis et envoyés au Cloud via le module GSM.

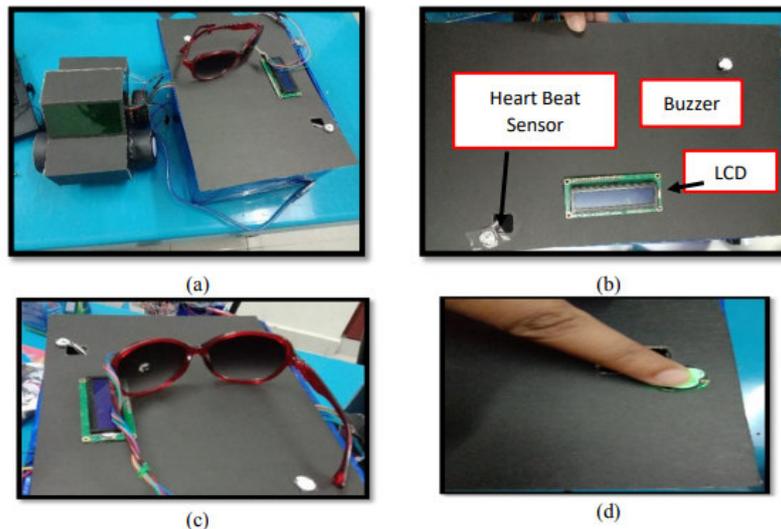


Figure 2: Prototype de l'œuvre proposée

E. Rogado, J.L. García, R. Barea, L.M. Bergasa, 2009^[5], les auteurs de cet article ont présenté une méthode pour détecter les premiers signes de somnolence du conducteur en analysant différentes variables biologiques et environnementales, tel que la fréquence cardiaque, la pression exercée sur le volant et la différence de température entre l'intérieur et l'extérieur du véhicule. La collecte et le traitement de ces variables leur ont permis d'estimer le niveau de fatigue du conducteur de manière indirecte.



Figure 3: Système en cours d'exécution

2- Etude du comportement

L'étude comportementale fait référence aux changements dans les réactions du visage du conducteur qui pourraient généralement être détecté en surveillant l'état des yeux et de la bouche.

Maysoon Abulkhair, Arwa H.Alsahli, Kawther M.Taleb, Atheer M.Bahran, Fatimah M.Alzahrani, Hend A.Alzahrani, Lamiaa Fattou hIbrahim, 2015^[6], Dans cet article, les auteurs ont pris un système de détection de fatigue qui a été proposé dans un travail récent est l'ont modifié pour fonctionner dans un Smartphone au lieu d'un ordinateur qui est très difficile à fixer sur un véhicule, le téléphone mobile est fixé dans la voiture par un support en face du conducteur. Les informations sur l'état du visage et des yeux sont obtenues grâce à la camera du Smartphone ensuite un algorithme fait le traitement d'image. Lorsque la somnolence est détectée, le téléphone déclenche une alarme pour alerter le conducteur.



Figure 4: Détection du visage par la camera du Smartphone

Jie Yi Wong et Phooi Yee Lau, 2019^[7], ont utilisé le Raspberry Pi 3 model B dans leur travail qui proposé un système pour alerter le conducteur en temps réel basé sur la surveillance de l'état du conducteur. Grace à une camera Raspberry il acquis les images et les traites pour détecter si le conducteur a les yeux fermés, basé sur le rapport hauteur/largeur des yeux, si les yeux du conducteur sont fermés pendant 2 secondes il y a somnolence ou inattention du conducteur, un avertisseur émet un bip fort pour l'alerter.



Figure 5: Illustration du fonctionnement du système

3- Approche basée sur le véhicule

Une autre méthode pour mesurer la somnolence du conducteur grâce à des capteurs placée sur divers composants du véhicule, y compris le volant et la pédale d'accélération, les signaux envoyés par les capteurs sont analysés pour déterminer le niveau de somnolence.

D'après Arun Sahayadhas, Kenneth Sundaraj et Murugappan Murugappan, 2012^[8], Certains chercheurs ont découvert que la privation de sommeil peut entraîner une plus grande variabilité dans la vitesse de conduite. Cependant, les deux facteurs les plus couramment utilisés sont le mouvement du volant et l'écart type de la position de la voie.

Le mouvement du volant (SWM) est mesuré à l'aide d'un capteur d'angle monté sur la colonne de direction, le comportement de conduite du conducteur est mesuré. En cas de somnolence, le nombre des micro-corrrections sur le volant diminuent par rapport à la conduite normale. Les conducteurs privés de sommeil effectuaient moins de retournements de volant que les conducteurs normaux.

L'écart type de la position sur la voie (SDLP) est une autre mesure par laquelle le niveau de la somnolence du conducteur peut être évalué. La position de la voie est suivie à l'aide d'une caméra externe. Mais la SDLP dépende des facteurs externes comme le marquage routier, les conditions climatiques et d'éclairage. En résumé, de nombreuses études ont déterminé que les mesures basées sur les véhicules sont un mauvais prédicateur de somnolence car le risque d'erreurs est grand. De plus, la SDLP peut également être causé par tout type de conduite avec facultés affaiblies, y compris la conduite sous l'influence de l'alcool ou d'autres médicaments, en particulier les déprimeurs.



Figure 6: Détecteur de ligne sur véhicule

III- Problématique et solution proposé

Vue le nombre effrayant d'accidents de la route du à la fatigue et la somnolence des conducteurs lors des longs trajets et durant la nuit, une solution embarquée au moindre cout et non invasive est nécessaire, Pour venir en aide au conducteurs.

La solution propose avec la disponibilité des SBC à moindre cout, La carte Raspberry Pi 4 avec 2GB de RAM nous permet d'implémenter quelques algorithmes de détection du visage et classification de l'état des yeux (ouvert, fermé).

IV- Etude théorique de la détection du visage et des yeux par traitement d'image

1- Détection du visage

La localisation de visage est réalisée à l'aide de Haar Cascade basée sur les travaux de Viola & Jones^[09]. Ces détecteurs utilisent une cascade de filtres chargés d'extraire des indices caractéristiques du visage. Cette cascade de filtres est optimisée par une phase d'apprentissage par méthode dite "boostée"^[10].

La détection de visage par l'algorithme de type Viola & Jones est caractériser par sa rapidité d'exécution. Cette méthode est efficace lorsque le visage fait face à la caméra, les performances baissent de façon significative lorsque le visage s'incline de plus de 20° dans le plan de la caméra ou effectue une rotation 3D de plus de 25°. De plus, la localisation du visage reste grossière, puisque d'une image à la suivante, la boîte englobante donnée par le détecteur peut subir un changement de taille de l'ordre de 20% et un décalage spatial de l'ordre de 10% de la taille du visage. Cela est peu gênant dans le cadre de détection des conducteurs filmés de face.

De plus, les bonnes performances globales ainsi que la rapidité d'exécution sont des avantages dans le but de réaliser un algorithme fonctionnant en temps réel.

La figure 2 présente des exemples de détection de visage réalisés à l'aide de Haar Cascade.

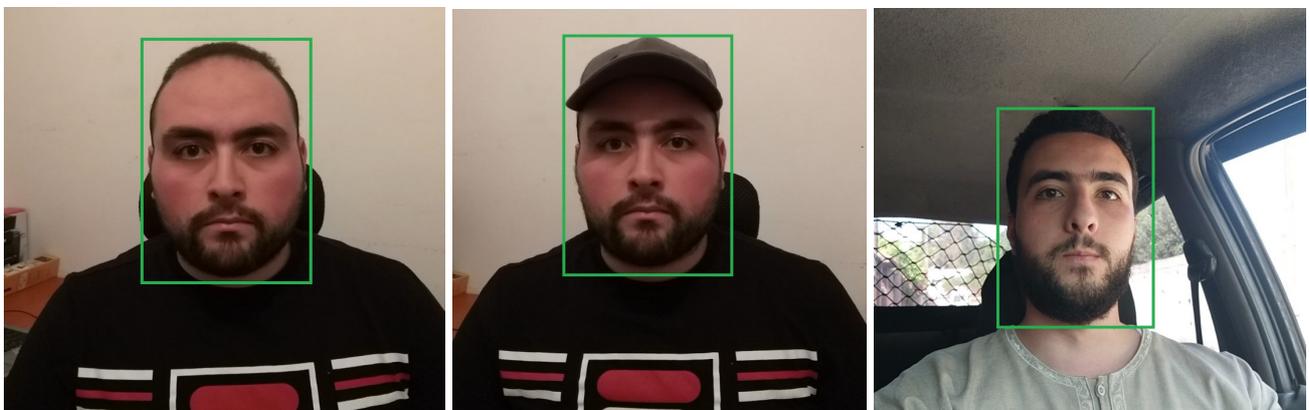


Figure 7: Exemples de détection de visages

2- Détection des yeux

Pour détecter les yeux, il faut d'abord détecter le visage, puis détecter automatiquement la boîte englobante de chaque œil. C'est sur ces boîtes que l'analyse automatique de clignement est menée. On suppose que chaque œil se trouve dans le quart supérieur du visage (droit ou gauche). La conséquence immédiate de cette hypothèse est que le visage doit être vertical dans le plan de la caméra. Ainsi le système peut tolérer des rotations jusqu'à 45° dans le plan de la caméra. Chacune des yeux est recherchée de manière indépendante.

On cherche donc à définir une boîte englobante autour de l'œil à partir de la zone de recherche prédéfinie. On choisit de prendre une boîte englobante carrée représentant $1/16^{\text{ème}}$ de la taille de la boîte englobant le visage, comme présenté sur la figure 2. Ce choix est conforme aux travaux de Hammal ^[11].

Sur la figure 2, la boîte englobante du visage est représentée en pointillés vert, la zone de recherche de l'œil est en rouge et la boîte englobante de l'œil est en jaune.

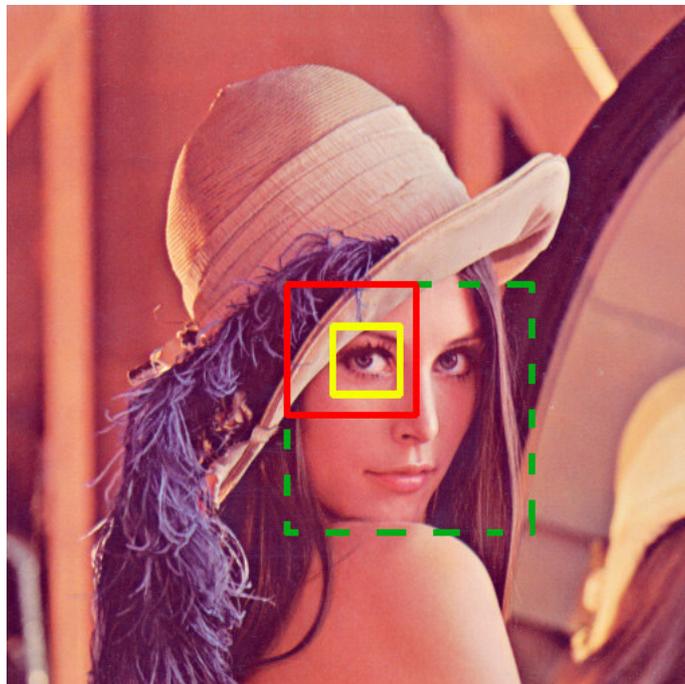


Figure 8: Illustration des différentes boîtes englobantes pour la détection des yeux

Dans le visage, l'œil ouvert peut se caractériser par la richesse de ses contours. En effet, des contours horizontaux sont présents entre l'œil et la paupière ainsi que des contours verticaux au niveau de la frontière entre l'iris et la cornée (blanc de l'œil). Des méthodes dégradées peuvent ainsi permettre de détecter l'emplacement de l'œil ^[12]. Cependant, l'œil doit être détecté qu'il soit ouvert ou fermé. Lorsque l'œil est fermé, seuls les contours horizontaux persistent. La paupière présente alors les mêmes caractéristiques que le sourcil qui présente essentiellement des contours horizontaux.

La méthode la plus utilisée dans la détection des yeux est présentée en figure 3. Un rehaussement des contours est réalisé sur la zone de recherche de l'œil à l'aide du filtre OPL. Le gradient horizontal de cette image est alors calculé puis fusionné à cette image par multiplication de manière à mettre en avant les jonctions horizontales. On réalise alors les projections horizontale et verticale de l'image de manière à

détecter la position de l'œil ^[13, 14]. L'abscisse de l'œil correspond au maximum global de la projection horizontale. Pour l'ordonnée de l'œil, on s'intéresse aux maxima locaux de la projection verticale.

On choisira alors le maximum correspondant à la partie inférieure de la zone de recherche. En utilisant cette méthode, nous faisons en fait l'hypothèse que s'il existe deux maxima locaux, celui de la partie supérieure de l'image correspond au sourcil et celui de la partie inférieure à l'iris.

Une fois le centre de l'œil trouvé, une boîte englobante peut alors être définie autour de l'œil. L'originalité de cette méthode est qu'on se sert des paupières pour situer les yeux. Il est ainsi possible de détecter les yeux même s'ils sont fermés, ce qui n'est pas le cas des méthodes qui se basent uniquement sur les caractéristiques de l'iris pour détecter les yeux.

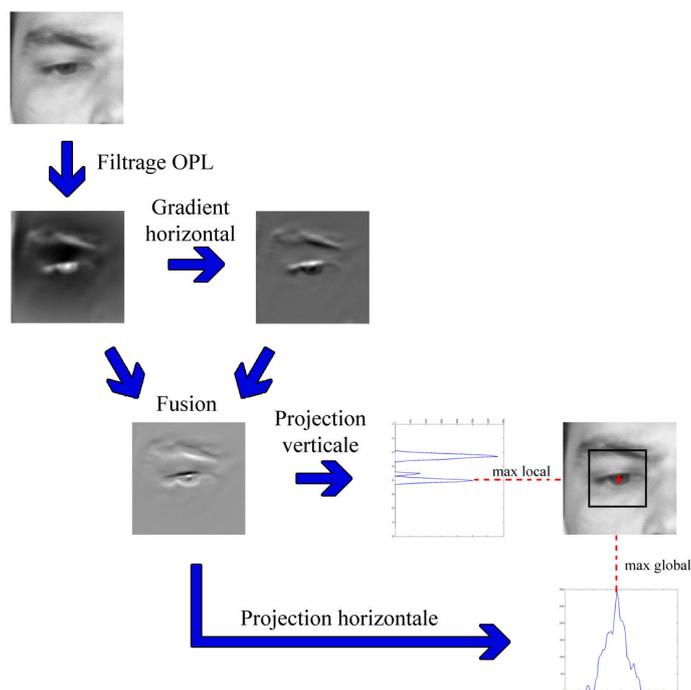


Figure 9: La Méthode de détection de l'œil

V- Objectifs du projet

Dans ce projet on a pour but d'offrir une solution pour les entreprises de transport de passagers et de marchandise ou les individus qui font des longs trajets souvent.

Cette solution consiste à proposer un système capable de détecter la somnolence des conducteurs durant le trajet, et de lancer des signaux d'alerte à plusieurs niveaux afin que le conducteur prend conscience de son état et s'arrête pour se reposer avant de continuer le trajet en toute sécurité.

On propose aussi une interface web qui permet à la compagnie de suivre en temps réel l'état de plusieurs conducteurs en même temps via des informations envoyées par des systèmes de détection de somnolence installés sur les véhicules.

CHAPITRE 2

I- Synoptique du système

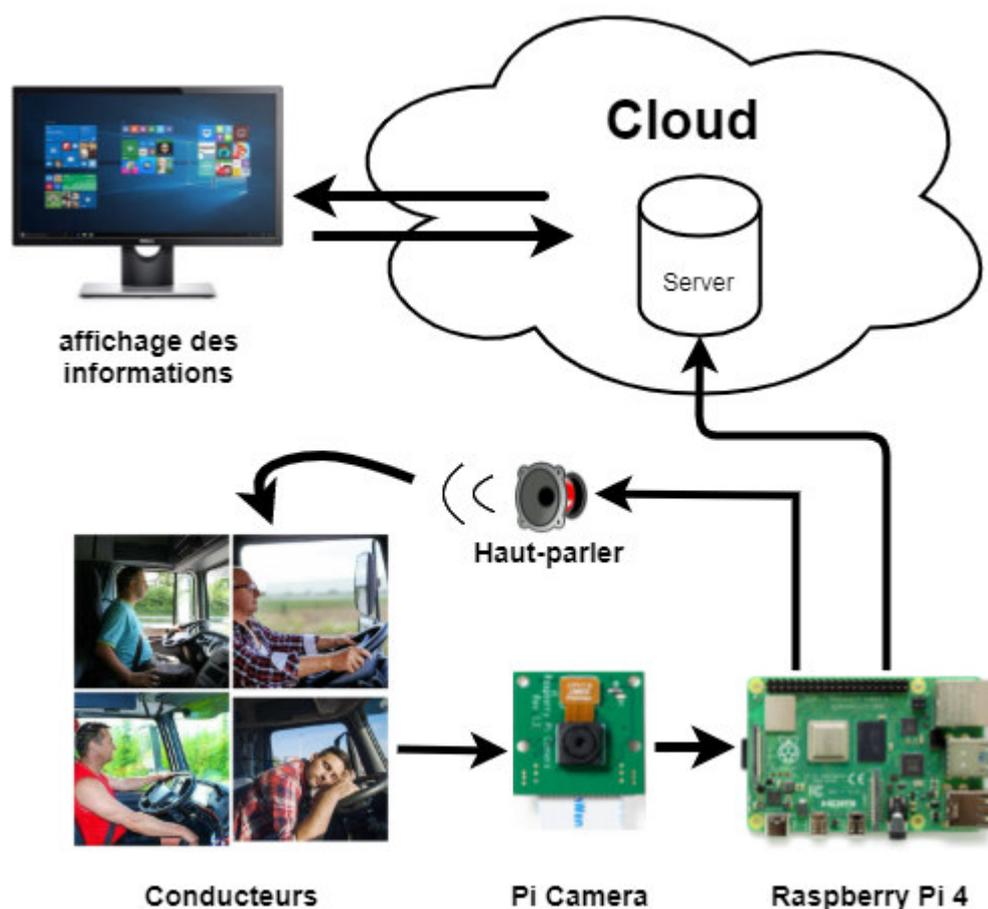


Figure 10: Synoptique du système

Ce projet est constitué d'une Raspberry Pi ainsi qu'une camera et un haut-parleur.

Dans un premier lieu on utilise une camera qui est placée en face du conducteur pour obtenir des images en temps réel. Ensuite, on procède au traitement d'image par un algorithme qui détecte le visage du conducteur puis ces yeux afin de décider si ce dernier a les yeux ouverts ou bien fermer.

Si le conducteur a les yeux ouverts ou bien cligne des yeux, le système le considère comme éveillé et continue à suivre son état.

S'il ferme les yeux pour plus de 3 secondes, le système lance une alerte grâce à la synthèse vocale qui répète la phrase « Attention! Wake up ». Ensuite, si le conducteur ouvre les yeux l'alerte vocale s'arrête. Et s'il continue à fermer les yeux pour plus de 7 secondes, le système lance un son de sirène très fort pour obliger le conducteur à ce réveillé. Pendant tous ce temps le système envoie périodiquement l'état du conducteur chaque 10 secondes au Cloud. En cas d'alerte il envoie une notification accompagné d'une image du conducteur en question en temps réel.

Les informations envoyées au serveur sont enregistrées dans une base de données SQL, lorsqu'on accède au site web, ces informations sont récupérées et organisées, puis présenté dans une interface graphique, pour permettre d'observer les états des conducteurs et de conserver un historique. Cette interface web est crée avec la technologie Django et héberger dans le Cloud (Heroku).

II- Description Hardware

La détection de somnolence dans ce projet est effectuée avec du traitement d'image, alors il est nécessaire d'utiliser un matériel adéquat pour cette tâche rigoureuse en ressource et calcul, la carte Raspberry Pi est alors choisit pour ses performances et sa petite taille.

Le Hardware utilisé dans ce projet est constitué d'une carte Raspberry Pi 4 model B, un module camera, un haut-parleur, un modem Wifi et un chargeur voiture.

1- Raspberry Pi

La carte Raspberry Pi est un nano ordinateur de la taille d'une carte de crédit que l'on peut brancher à un écran et utilisé comme un ordinateur standard. Sa petite taille, et son prix intéressant fait d'elle un produit idéal pour tester différentes choses, et notamment la création d'un serveur Web chez soi. Évidemment, pour sa taille il ne faut pas s'attendre à des performances incroyables, mais pour mettre en ligne des projets à montrer au client ou expérimenter avec linux c'est largement suffisant.^[15]

Ces SBC ont connu une évolution depuis la toute première carte Raspberry Pi 1 modèle B sortie en 2012 suivi du modèle A en 2013, Pi 2 B et Pi Zéro en 2015 , pi 3 B en 2016, pi 3 B + en 2018, enfin la dernière version disponible a ce jour la Pi 4 B, mise sur le marché le 24 juin 2019 au prix allant de 35 a 75 dollars selon la capacité de ça RAM 2GB, 4GB et 8GB.

Dans ce projet on a utilisé la carte Raspberry Pi 4 model B qui possède : une RAM (2GB, LPDDR4), un processeur (Broadcom BCM2711 Quad-core 1.5GHz), 4 ports USB, 2 ports micro HDMI ce qui la rend capable de faire marcher 2 écrans en résolution 4K, wifi (IEEE 802.11ac Wireless), Bluetooth 5.0, et plein d'autres caractéristiques qui la rende idéal pour notre projet^[16].

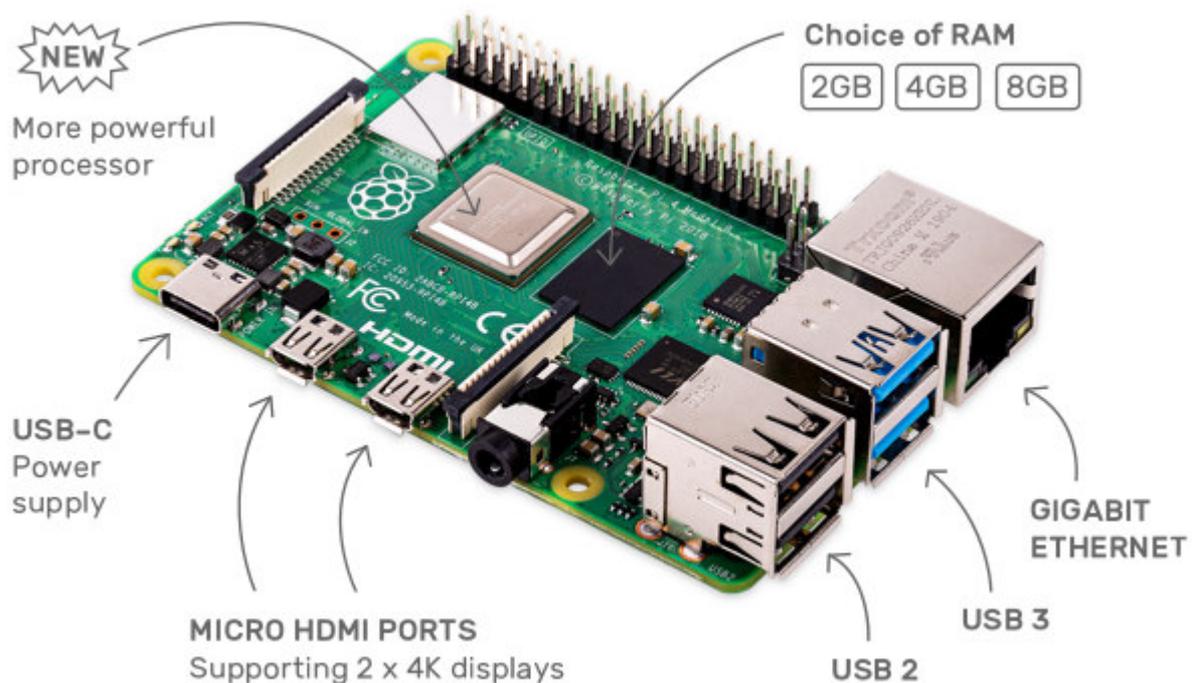


Figure 11: Raspberry Pi 4 model B

2- Module Camera

Le module camera Raspberry Pi v2 est équipé d'un capteur Sony IMX219 8-megapixel, il supporte les modes vidéo: 1080p30, 720p60 et VGA90, compatible avec les versions 1, 2,3 et 4 de la carte Raspberry.

Ce module est utilisé pour capturer des vidéos du conducteur pour être traité par l'algorithme dans la Raspberry Pi en temps réel^[17].

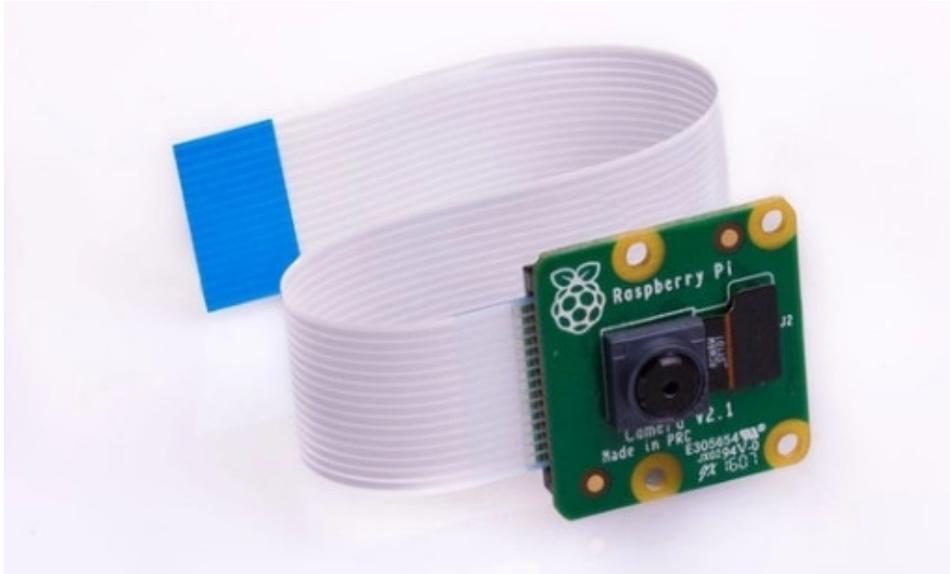


Figure 12: module camera pour Raspberry Pi

3- Haut-parleur

Un haut-parleur est un transducteur électroacoustique destiné à produire des sons à partir d'un signal électrique, Il est en cela l'inverse du microphone^[18]. Attachée à la Raspberry Pi il représente la sortie audio pour avertir le conducteur avec une synthèse vocale ou bien un son de buzzer.



Figure 13: Haut parleur

4- Modem 4G – WIFI

Le modem dispose d'une carte Sim qui lui permet de se connecter au réseau mobile 4G et d'accéder a internet, il partage la confection via un réseau Wifi ce qui permet aux appareils autour de lui, comme la carte Rspberry Pi, de se connecter à internet.



Figure 14: Modem 4G - WIFI

5- Chargeur voiture

C'est un adaptateur 12V À 5V pour alimenter la carte Raspberry à partir de la prise 12V (Allume cigare) présente sur le véhicule.



Figure 15: chargeur voiture

III- Description Software

Ce système de détection de sommeil pour conducteur est conçu principalement avec Python et autres bibliothèques utiles dans cette réalisation :

1- Python

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages. Il est accessible pour les débutants avec une prise en main rapide.

Les principales utilisations de Python par les développeurs sont :

- La programmation d'applications.
- La création de services web.
- La génération de code.
- La méta-programmation.

Techniquement, ce langage servira surtout pour le Scripting et l'automatisation (interaction avec les navigateurs web).^[19]

2- Open CV

2-1- Description d'OpenCV

OpenCV (Open Source Computer Vision) est une bibliothèque open source de traitement et d'analyse d'images et de vidéos avec des interfaces pour les principaux langages de programmation: C, C++, Java, C#, Python... Python sera le langage de programmation que l'on utilise pour implémenter le programme de détection de somnolence.

OpenCV fournit une API de bas et haut niveau, elle fournit aussi une interface pour la programmation en parallèle. OpenCV est utilisée dans les laboratoires de recherche et aussi dans l'industrie.

Les principales fonctions d'OpenCV :

- Manipulation d'image (chargement, sauvegarde, copie, conversion...).
- Manipulation et acquisition de vidéos.
- Manipulation de matrices et algèbre linéaire.
- Structures de données utilitaires variées (listes, files, ensembles, graphes...).
- Traitement d'images (filtrage, détection de discontinuités, morphologie mathématique...).
- Analyse d'images (composantes connexes, ajustement de primitives, transformée en distance...).
- Vision (calibrage de caméra, stéréovision...).
- Reconnaissance de forme (modèle de Markov, ACP...).
- Interface graphique (affichage d'images, de vidéos, gestion des événements...)^[20]

2-2- La Détection du visage avec le détecteur d'OpenCV

Comme mentionné plus haut, la première étape de la reconnaissance de visages est leur détection. La bibliothèque OpenCV rend assez facile la détection de visage de face dans une image en utilisant son détecteur "Haar Cascade" (également connu sous le nom de la méthode Viola-Jones).

L'algorithme de Viola et Jones, qui est une méthode de détection d'objet (détection de visage, détection de personne, détection d'avion, détection de voiture) dans une image numérique, est implémenté dans OpenCV.

OpenCV contient plusieurs classificateurs pour la détection de visages dont les classificateurs pour: les poses frontales, les visages de profil, la détection des yeux, la détection du nez, la détection de la bouche...

Pour la détection de visage de face, nous pouvons utiliser l'un des classificateurs Haar Cascade qui viennent avec OpenCV, chacun de ces classificateurs va donner des résultats légèrement différents en fonction de l'environnement où l'on les utilise.^[20]

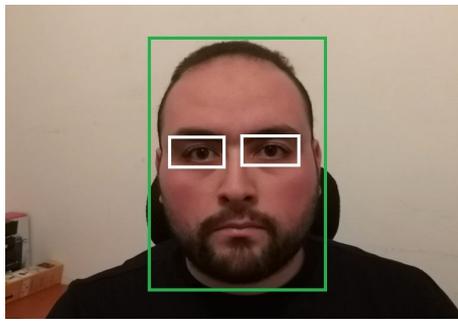


Figure 16: Exemple de détection de visage avec une cascade de classifieur

3- Haar-cascade

La détection d'objets à l'aide de classificateurs en cascade basés sur les caractéristiques de Haar est une méthode de détection d'objets efficace proposée par Paul Viola et Michael Jones. C'est une approche basée sur le Machine Learning, où La fonction cascade est entraînée à partir de nombreuses images positives et négatives. Elle est ensuite utilisée pour détecter des objets dans d'autres images. OpenCV fournit une méthode d'entraînement ou des modèles pré-entraînés, qui peuvent être utilisés directement. Les modèles pré-entraînés se trouvent dans le dossier de données de l'installation OpenCV. [21]



Figure 17: Caractéristique des yeux haar-cascade

La première étape consiste à collecter les caractéristiques Haar. Une caractéristique de Haar est essentiellement des calculs effectués sur des régions rectangulaires adjacentes à un emplacement spécifique dans une fenêtre de détection. Le calcul consiste à additionner les intensités de pixels dans chaque région et à calculer les différences entre les sommes.

Voici quelques exemples de caractéristiques Haar ci-dessous. [22]

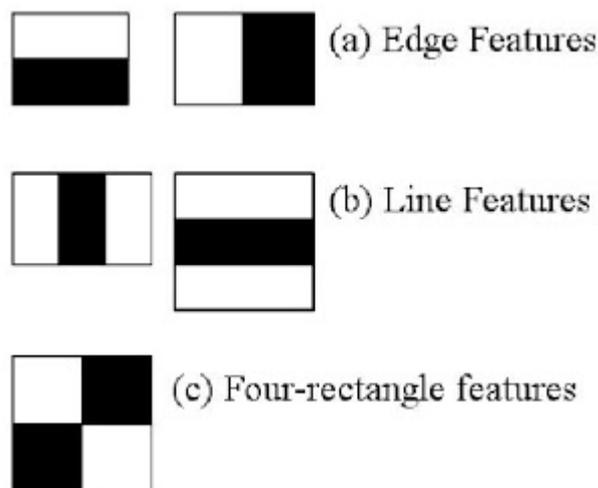


Figure 18: Caractéristique de Haar

Dans ce projet on a utilisé 2 classificateurs Haar cascade, un pour détecter le visage et l'autre pour détecter les yeux.

4- Dlib

Dlib est un ensemble d'outils moderne contenant des algorithmes de Machine Learning, et des outils permettant de créer des logiciels complexes pour résoudre des problèmes du monde réel. Il est utilisé à la fois dans l'industrie et dans l'éducation et dans plein d'autres domaines, notamment la robotique, les appareils embarqués, les téléphones mobiles et les grands environnements informatiques haute performance.

La licence open source de Dlib permet de l'utiliser dans n'importe quelle application, gratuitement.^[23]

Dlib a une bibliothèque graphique qui nous permet de faire du traitement d'image. Le résultat donné par Dlib est mieux qu'Open-CV et Haar-cascade. Comme on le voit dans la figure si dessous :

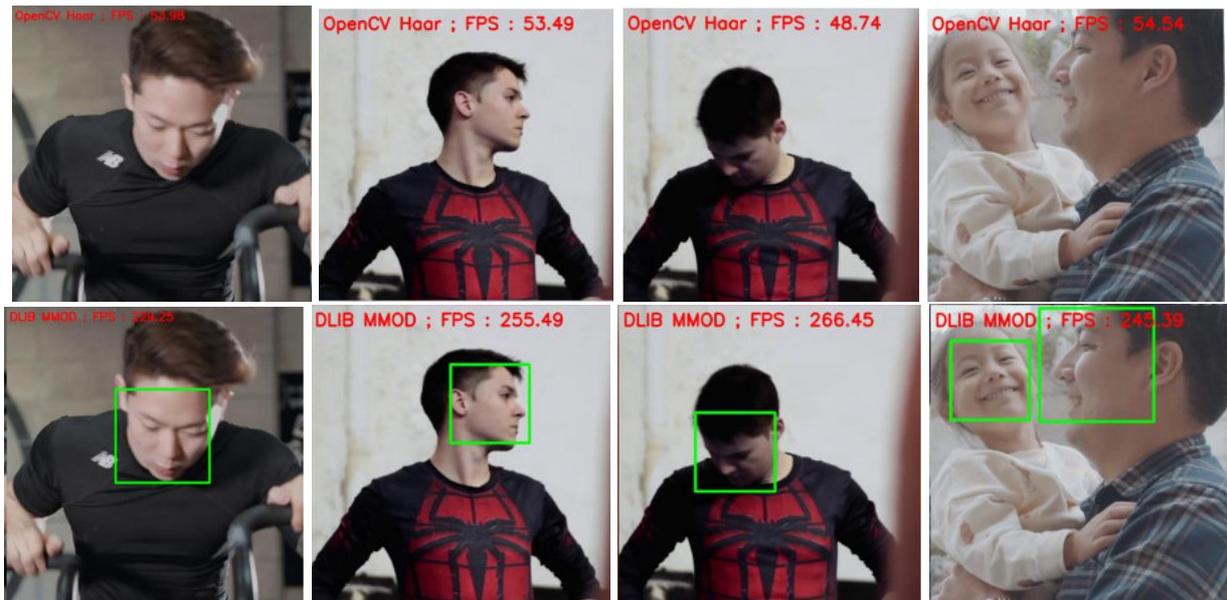


Figure 19: exemples de détections avec OpenCV et Dlib

Mais Dlib est très couteux en termes de ressources et calculs, ses performances avec la Raspberry Pi ne sont pas satisfaisantes.

5- Pyttsx3

Pyttsx3 (Python text to speech V3) est une bibliothèque de conversion texte-parole en Python.

Contrairement aux bibliothèques alternatives, il fonctionne hors ligne, c'est un outil très facile à utiliser qui convertit le texte saisi en parole, à l'alerte cette bibliothèque est utiliser pour générer un son audio à partir d'un texte comme "Attention! Réveillez vous!" pour avertir le conducteur à travers un haut-parleur.

Text to Speech using Python

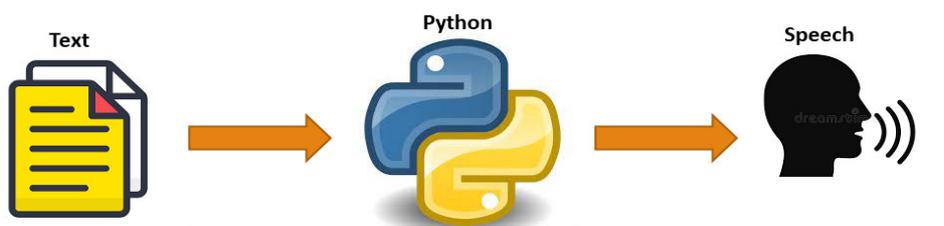


Figure 20: Python text to speech

IV- Description du programme développé

1- Fonctionnement du programme

L'état du conducteur est détecté à partir des réactions du visage, précisément les yeux, les yeux du conducteur sont suivis et on teste s'ils sont fermés ou pas.

Il y a trois états principaux pour l'algorithme, qui sont:

- visage non détecté.
- visage détecté et yeux détecté (état normal).
- visage détecté et yeux non détecté (état de somnolence).

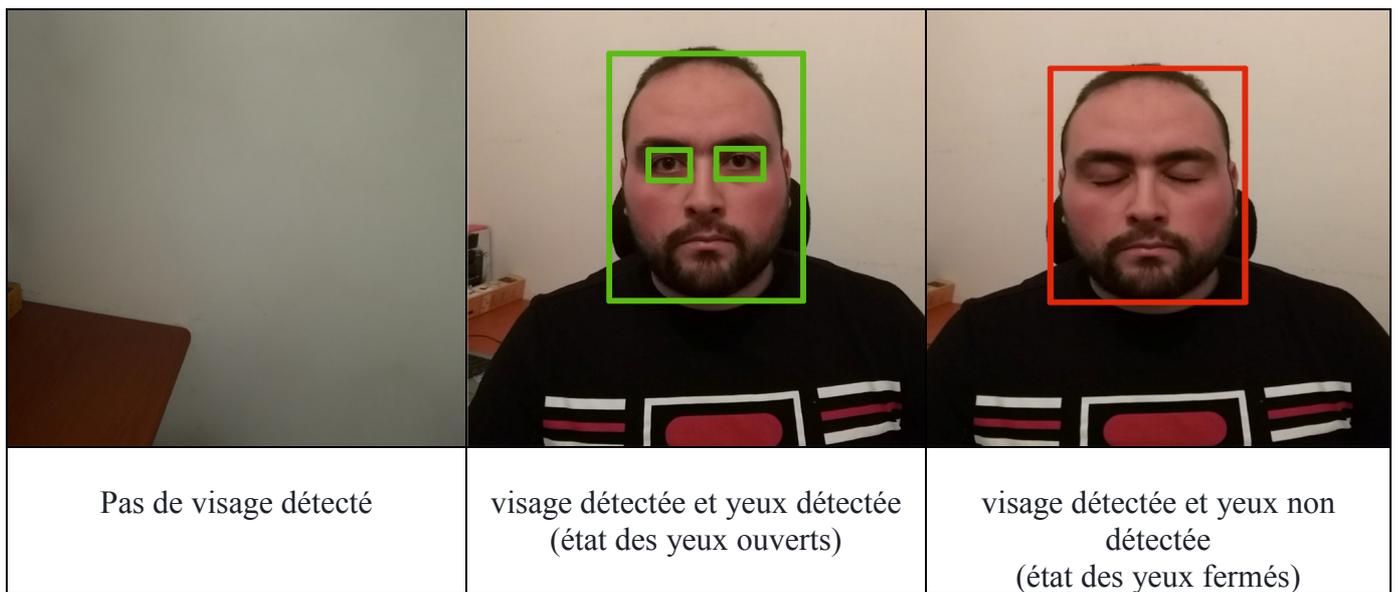


Figure 21: Détection du visage et des yeux

Dans l'état où aucun visage n'est détecté, le programme ne fait rien et reste en attente.

Lorsqu'un visage est détecté, on teste pour voir si des yeux sont détectés dans ce visage.

Et si ils sont ouverts alors on est dans l'état normal.

Sinon si ils ne sont pas ouverts c'est l'état d'alerte.

Le programme lance une alerte audio et envoie les informations concernant l'état du conducteur au Cloud.

La figure ci dessous représente le diagramme d'activité du programme réalisé :

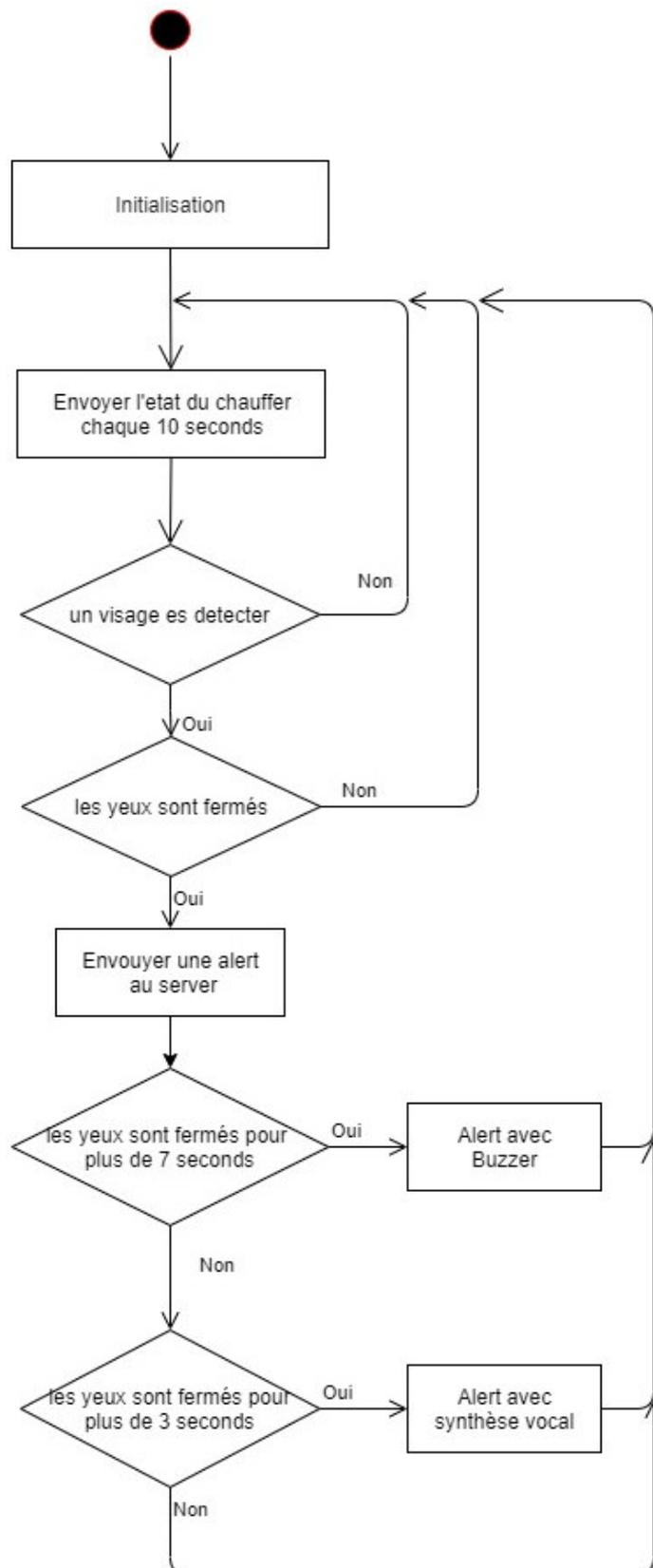


Figure 22: Diagramme d'activité de l'algorithme réalisé

2- les principales fonctions et instructions utilise

2-1- Initialisation du system

```
import requests # pour envoyer les information au server
import cv2      # openCV, pour détecter le visage et les yeux
import pyttsx3  # Synthèse vocal
import vlc      # Pour la lecture audio
import threading
import base64
import datetime
import time
```

Dans cette partie on fait appelle aux différentes bibliothèques qu'on va utiliser dans notre projet.

2-2- Capture Vidéo

```
cap = cv2.VideoCapture(0) # pour capturé des images depuis le module camera
ret, img = cap.read()     # lecture de l'image actuelle
img = cv2.flip(img, 0)    # rotation de l'image
```

Ces instructions nous permettent de capturer une séquence d'images, puis de sélectionner l'image actuelle pour la traiter, on fait tourner l'image dans le sens correcte.

2-3- Traitement de l'image

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # convertir l'image en noir et blanc
gray = cv2.bilateralFilter(gray, 5, 1, 1) # filtrer pour réduire le bruit
```

Ces instructions nous permettent de simplifier l'image en la rendant noir et blanc, puis appliquer un filtre de réduction de bruit. Cette procédure améliore les performances du traitement.

```
faces = face_cascade.detectMultiScale(gray, 1.3, 5, minSize=(50, 50))
eyes = eye_cascade.detectMultiScale(face_box, 1.3, 5, minSize=(5, 5))
```

La première institution sert a détecté le visage dan l'image actuelle, et la seconde nous permet de détecté les yeux dans ce visage

2-4- Alerte audio

```
def alert(diff):
    if diff >= 3 and diff < 6:
        engine.say('Attention Wake up!')
        engine.runAndWait()

    elif diff >= 6:
        vlc_instance = vlc.Instance('--aout=alsa')
        buzzer = vlc_instance.media_player_new()
        buzzer.set_media(
            vlc_instance.media_new('src/media/buzzer.wav')
```

```

)
vlc.libvlc_audio_set_volume(buzzer, 100)

buzzer.play()
time.sleep(3)
buzzer.pause()

alert_triggered = False

```

Cette fonction sert à donner des alertes audio en fonction du temps écoulé depuis la fermeture des yeux, si se temps dépasse les 3 secondes, la fonction lance une alerte audio via une synthèse vocal, et si il est supérieur a 6 secondes, la fonction lance une alerte audio plus forte (buzzer).

2-5- Envoyer des informations vers le serveur:

```

def server_send(current_img, current_diff=None):
    if(alert_triggered):
        if current_diff != None :
            diff = format(current_diff, '.2f')
        elif closed_time != None:
            diff = format(time.time() - closed_time, '.2f')
        else:
            diff = '0'

        date = datetime.datetime.now().isoformat()
        img_url = upload_image(current_img)
        data = {'state': 'Drowsy', 'closed_time': diff, 'image': img_url,
              'date': date, 'driver': identifier}

    else:
        data = {'state': 'Good', 'date': datetime.datetime.now().isoformat(),
              'driver': identifier}

    res = requests.post(url=f'{server_url}/add-state/', data=data)

```

Cette fonction sert à actualiser l'état du conducteur en envoient une requêtes au server avec les nouvelles informations.

V- Application Web

1- Création du l'application web

Notre application web est développée avec la Framework Django, qui est une Framework open source construite avec le langage de programmation Python, elle nous faciliter la création des applications web et on trouve plein de modèles ouvert à utiliser et à modifier. Elle à une base de données SQLite, ce server reçoit des informations envoyée par la carte raspberry pi, il les sauvegarde dans la base de données, pour ensuite les afficher dans une interface graphique qui est accessible via le lien si dessous: "<https://drowsiness-detection-ubma.herokuapp.com>".

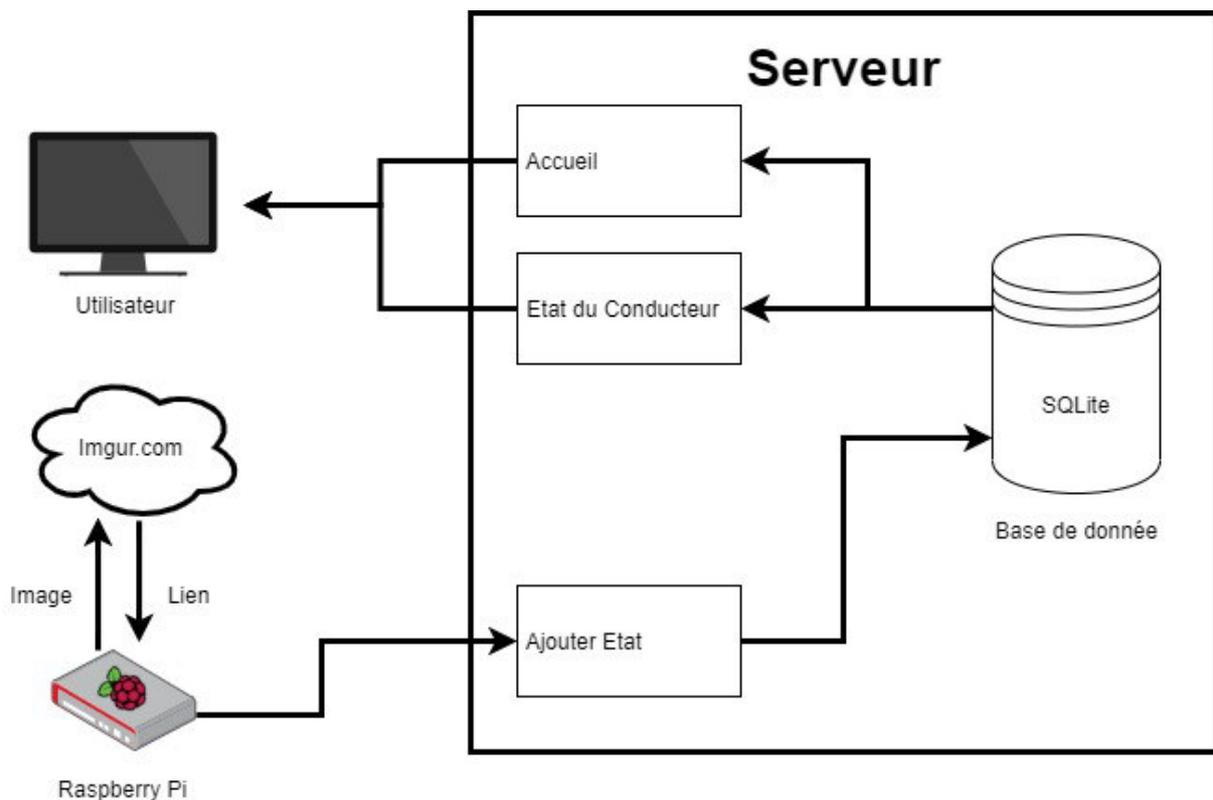


Figure 23: Synoptique de l'application web

Notre serveur est hébergé sur le Cloud (Heroku), nous avons opté pour se Cloud parce qu'il offre plusieurs forfaits payant et gratuit.



Figure 24: Les forfaits Heroku

L'abonnement gratuit que nous avons choisi est très généreux, il offre un serveur avec 512MB de RAM et un stockage un peu limité.

Il permet aussi de déployer notre code source depuis Github (une plateforme d'hébergement de code pour contrôle de version et collaboration), dans laquelle on héberge notre code source de l'application.

A cause des restrictions de stockage de l'abonnement gratuit sur Heroku, on a du passer par un site intermédiaire (imgur.com) pour stocker nos images, en s'inscrivant sur ce site il nous permet d'uploader les images directement en lui envoyant l'image et il nous rend en retour le lien de l'image uploadée, on sauvegarde ce lien dans la base de donnée du serveur pour l'afficher après dans l'interface graphique web.

2- fonctionnement de l'application web

Notre application présente au utilisateur une interface graphique simple et facile à utiliser, elle est constituée de deux pages principales, la première est la page d'accueil, elle présente les conducteurs avec leur informations, comme le nom, prénom, le type de véhicule et si le conducteur est active ou pas.

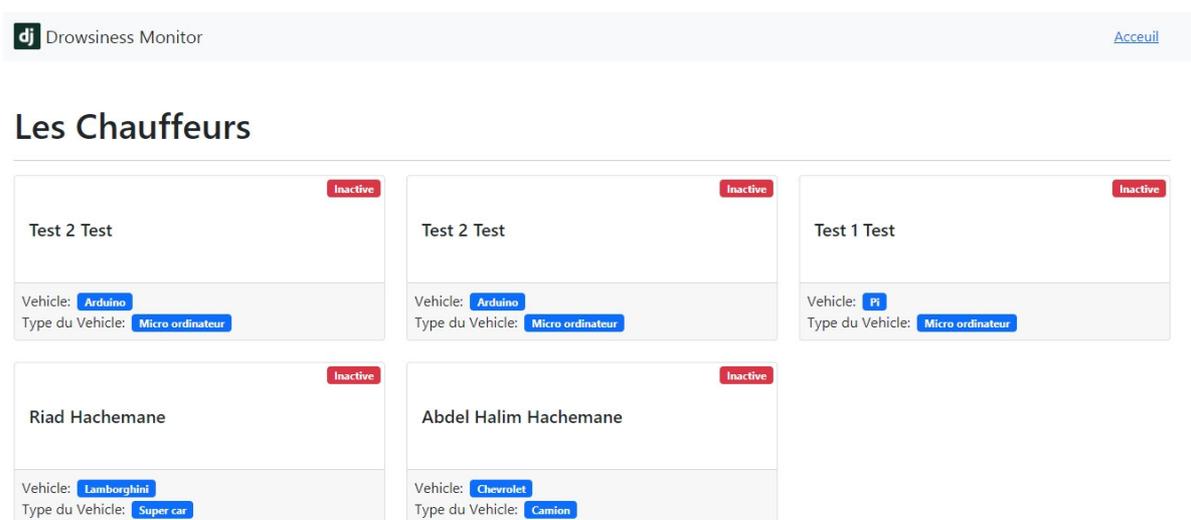


Figure 25: Page d'accueil

Quand on clique sur un conducteur, on est redirigés à la deuxième page qui affiche l'historique des états de ce conducteur, Il est présenté sous forme de liste qui contient l'état du conducteur et la date de cet état, on remarque que si le conducteur est en état de somnolence une photo est afficher aussi avec ces informations.

dj Drowsiness Monitor [Acceuil](#)

Abdel Halim Hachemane

Éveiller le chauffeur est bien éveillé	2021-05-30 00:54:13
Éveiller le chauffeur est bien éveillé	2021-05-30 00:44:42
Endormi Yeux fermé depuis: 10.67 seconds	2021-05-30 00:44:33
	
Endormi Yeux fermé depuis: 9.29 seconds	2021-05-30 00:44:32
	

Figure 26: Page d'informations sur l'état du conducteur

CHAPITRE 3

I- Tests et Résultats

1- Test dans un environnement éclairé

On a effectué une série de tests sur 5 personnes de sexe masculin et féminin, d'âge entre 21ans et 62ans, et en ayant le visage nu, puis en portant des lunettes, et enfin avec une bavette. Ces tests on était fait dans une chambre bien éclairé, afin d'obtenir les meilleurs résultats possibles.



Figure 27: Les différentes conditions de tests

On a obtenu les résultats présentés dans le tableau si dessous :

Type de test	Nombre de tests	Test réussit	Fausse alerte	Pas de détection
Visage nu	15 tests (5 personnes)	15	0	0
Avec bavette	15 tests (5 personnes)	15	0	0
Avec lunettes	22 tests (5 personnes)	15	2	5

Tableau 1: Tests dans un environnement bien éclairé

2- Test dans un véhicule en marche

On a effectué une série de tests sur 2 personnes de sexe masculin, dans un véhicule en marche et dans plusieurs conditions pour tester l'efficacité de notre dispositif dans la vie réelle.

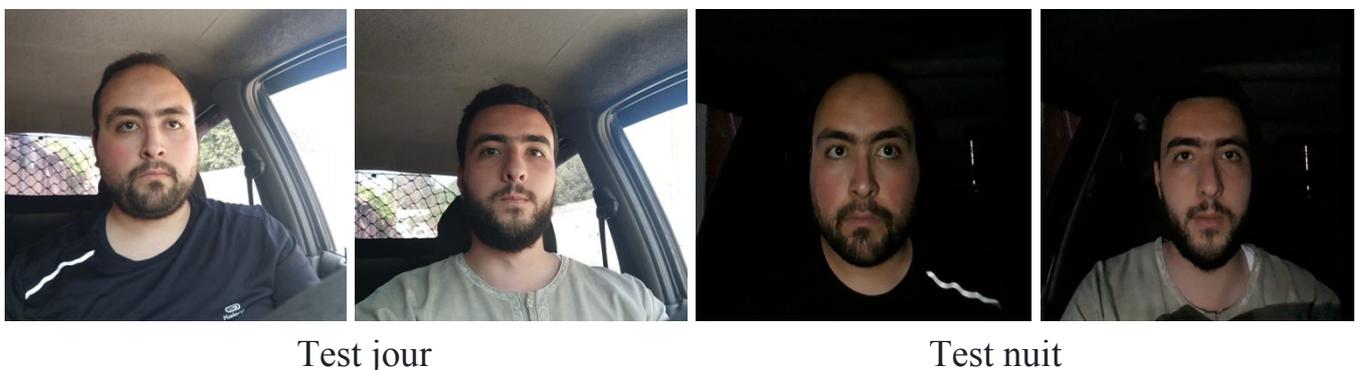


Figure 28: Tests jour et nuit

Les résultats obtenus de ces tests sont présentés dans le tableau ci-dessous :

Type de test	Nombre de tests	Test réussi	Fausse alerte	Pas de détection
jour	10 (2 personnes)	10	0	0
Avec bavette	12 (2 personnes)	9	1	2
Avec lunette	7 (2 personnes)	1	1	5
nuit	6 (2 personnes)	0	0	6

Tableau 2: Tests dans un véhicule en marche

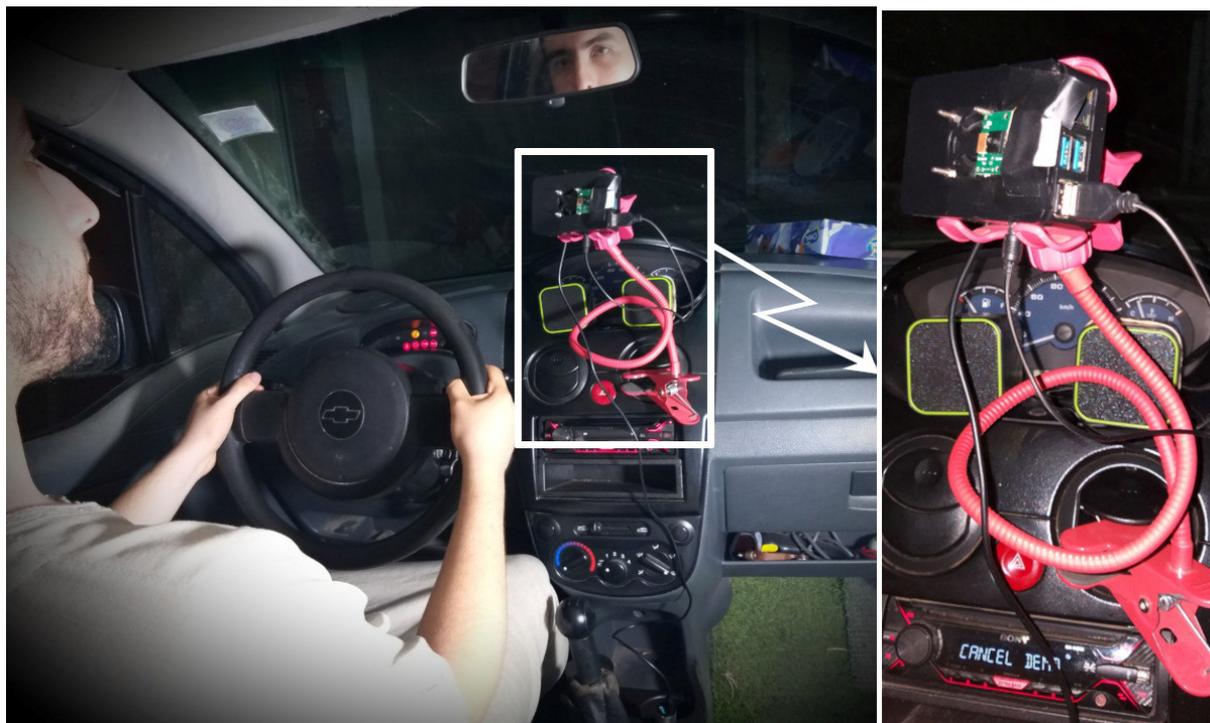


Figure 29: Dispositif en marche

3- Discussions des résultats obtenus

Les résultats obtenus dans un environnement bien éclairé sont très satisfaisants, quelque soit les conditions du visage: nu, porte des lunettes, porte une bavette.

Les résultats obtenus dans un véhicule en marche le jour sont assez satisfaisants, la détection est meilleure dans l'ombre. Avec des lunettes le reflet ne nous permet pas d'avoir une bonne détection et le port de masque n'as pas un grand impacte sur la détection des yeux.

Dans la nuit il est impossible de détecter le visage et les yeux du conducteur, ce qui rend notre dispositif carrément inutilisable.

4- Conclusion

Les résultats obtenus dans la journée sont acceptable. Dans la nuit, la ou le problème de somnolence arrive, il faut éclairer la cabine du conducteur par une lumière qui ne dérange pas le conducteur.

II- Perspectives

Les perspectives envisageables en prolongement direct de cette thèse concernent 4 objectifs importants: l'utilisation d'une Raspberry plus performante, l'utilisation d'une meilleure camera, l'ajout d'un module GPS, et enfin l'optimisation de l'application web.

1- Utilisation d'une carte Raspberry plus performante

Utilisation d'une version plus performante de Raspberry pi 4 qui procède 8GB de RAM, ce qui nous permettra d'utiliser Dlib, qui est plus précis et donne de meilleurs résultats dans la détection du visage et des yeux, mais qui demande beaucoup de ressources Hardware.

2- Utilisation d'un meilleur camera

Le module caméra infrarouge pour Raspberry Pi nous permettra de conférer à notre projet une vision nocturne et une prise de vue avec focale réglable. Nous pouvons y brancher des LED IR pour la prise de vue nocturne, ou des LED de fill-in pour renforcer la lumière naturelle avec une autre source d'éclairage. Ce module est doté d'un Capteur OV5647 de 5 mégapixels, et d'une résolution maximale de capture vidéo de 1080 p. ^[24]



Figure 30: Module camera infrarouge

3- ajout d'un module GPS

Ce module est compatible avec Raspberry Pi A, B, A+, B+, Zero, 2, 3. Avec sa puce GPS L80-39 il communique les données satellitaires via UART ou USB CP2102 comme pont USB à puce UART, il est stable et plus rapide, 66 canaux de recherche et 22 canaux de suivi simultanés, Technologie AGPS avancée EASY™ sans mémoire externe^[25]. Ce module nous permettra de suivre la localisation des véhicules en temps réel ainsi que leurs vitesses, et on connaîtra la position exacte des véhicules en cas d'accident.



Figure 31: module GPS

4- Optimisation de l'application web

Améliorer l'interface graphique et la présentation des informations dans la page web et ajouter une fonctionnalité qui nous permettra d'envoyer des notifications en forme d'email et de SMS aux personnes concernées. Aussi, utiliser un hébergement (Heroku) payant ce qui nous donnera plus d'espace mémoire, et un nombre de visiteurs illimité, ainsi que la possibilité de télécharger les images directement vers le serveur sans avoir à passer par un site d'hébergement d'images (imgur.com).

CONCLUSION GENERALE

La réalisation de notre projet comme système de détection de somnolence pour les conducteurs a démontré que c'était une tâche faisable et applicable, avec un budget raisonnable et un matériel facile pour se procurer.

Les systèmes de détection de somnolence pour les conducteurs seront bientôt indispensables dans les véhicules, et ca deviendra à la porter de tous.

L'expérimentation qu'on a menée était un succès et a prouvée que le système est capable de jouer son rôle en détectant la somnolence du conducteur et en le prévenant.

Dans ce travail, nous avons fait l'étude et la réalisation d'un système à base d'un SBC de nouvelle génération, nous avons aussi implémenté des procédures d'intelligence artificielle avec traitement d'image en temps réel en utilisant des outils qui facilitent l'opération. À travers les libraires open CV et les filtres de classification Haar Cascade (ML) dans la détection de l'état des yeux.

Nous avons aussi testé DLib qui est une procédure ML plus puissante, c'est une bibliothèque graphique utilisé dans le traitement d'image mais qui nécessite une RAM plus grande que celle de la carte Raspberry Pi.

Nous avons développé une interface web, pour que le responsable de l'entreprise puisse consulter et suivre l'état physique des conducteurs.

Finalement, nous avons proposé des perspectives qui peuvent améliorer la qualité du projet.

REFERENCES BIBLIOGRAPHIQUES

- [01] Atlas Magasine. « Baisse des accidents de la route en 2019 », 2020. Récupéré de <https://www.atlas-mag.net/article/baisse-des-accidents-de-la-route-en-2019>.
- [02] Algérie presse service. «Accidents-Bilan 2020», 2021. Récupéré de <https://www.aps.dz/societe/117350-accidents-bilan-2020-le-moins-tragique-depuis-trois-decennies>
- [03] Sigari M H, Fathy M and Soryani M. «A driver face monitoring system for fatigue and distraction detection», 2013. Récupéré de <https://www.semanticscholar.org/paper/77206f3bf18340c0dfe295266672aa16370f43cd>
- [04] M Arunasalam, N Yaakob, A. Amir, M Elshaikh et N F Azahar. «Real-Time Drowsiness Detection System for Driver Monitoring», 2019. Récupéré de <https://iopscience.iop.org/article/10.1088/1757-899X/767/1/012066>
- [05] E. Rogado, J.L. García, R. Barea, L.M. Bergasa, et E. López. «Driver Fatigue Detection System», 2009. Récupéré de https://www.researchgate.net/publication/224441521_Driver_Fatigue_Detection_System
- [06] Maysoon Abulhair, Arwa H.Alsahli, Kawther M.Taleb, Atheer M.Bahran, Fatimah M.Alzahrani, Hend A.Alzahrani et Lamiaa Fattou hIbrahim. «Mobile Platform Detect and Alerts System for Driver Fatigue», 2015. Récupéré de <https://www.sciencedirect.com/science/article/pii/S1877050915026666>
- [07] Jie Yi Wong et Phooi Yee Lau. «Real-Time Driver Alert System Using Raspberry Pi», 2019. Récupéré de <https://ph02.tci-thaijo.org/index.php/ECTI-EEC/article/view/215488>
- [08] Arun Sahayadhas, Kenneth Sundaraj et Murugappan Murugappan. «Detecting Driver Drowsiness Based on Sensors», 2012. Récupéré de <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3571819/>
- [09] P. Viola and M. Jones. «Robust real-time object detection». *International Journal of Computer Vision*, 57(2) :137–154, 2002.
- [10] M. Bartlett, G. Littlewort, I. Fasel, and J. Movellan. «Real time face detection and facial expression recognition: Development and applications to human computer interaction». In *Proc. of the Computer Vision and Pattern Recognition Workshop, Madison, USA, 2003*.
- [11] Z. Hammal. *Segmentation des traits du visages, «analyse et reconnaissance d'expressions faciales par le modèle de croyance transférable»*. PhD thesis, Université J. Fourier, Grenoble, 2006.
- [12] R. Kothari and J. L. Mitchell. «Detection of eye locations in unconstrained visual images». In *Proc of the IEEE ICIP conference, Lausanne, Switzerland, 1996*.
- [13] Z.-H. Zhou and X. Geng. «Projection functions for eye detection». *Pattern Recognition*, 37(5) :1049–1056, 2004.

[14] K. Peng, L. Chen, S. Ruan, and G. Kukharev. «A robust algorithm for eye detection on gray intensity face without spectacles». *Journal of Computer Science and Technology*, 5(3) :127–132, 2005.

[15] Récupéré de <https://grafikart.fr/blog/raspberry-pi-utilisation>

[16] Récupéré de <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/>

[17] Récupéré de <https://www.raspberrypi.org/products/camera-module-v2/>

[18] Récupéré de <https://fr.wikipedia.org/wiki/Haut-parleur>

[19] Récupéré de <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/>

[20] Manitra Tsilavina RAZAFIMANDIMBY. «Détection et reconnaissance de visages», 2016. Récupéré de http://biblio.univ-antananarivo.mg/pdfs/razafimandimbyManitraT_MP_MAST2_16.pdf

[21] Récupéré de https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

[22] Aditya Mittal. «Haar Cascades Explained», 2020. Récupéré de <https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d>

[23] Récupéré de <http://dlib.net/>

[24] Récupéré de <https://www.generationrobots.com/fr/403499-module-camera-infrarouge-pour-raspberry-pi-et-nvidia-jetson-nano.html>

[25] Récupéré de <https://www.kubii.fr/cartes-extension-cameras-raspberry-pi/1922-module-gps-pour-raspberry-pi-kubii-3272496007888.html>