



## Rapport de projet

### “Détection de somnolence au volant”

**Réalisé par :** Messar Cylia et Tagzirt Elissa.

**Encadré par :** Mm Daoud Siham.

**Groupe :** 2 CS SIQ1

Ecole nationale Supérieure d'Informatique  
2023-2024, ESI, Oued Smar

**Introduction :**

La détection de la fatigue ou de la somnolence chez les conducteurs est un enjeu majeur pour la sécurité routière. Les accidents causés par la somnolence au volant sont malheureusement fréquents et souvent graves. Dans ce contexte, la mise en place de systèmes de surveillance en temps réel pour détecter les signes de fatigue chez les conducteurs peut considérablement réduire le nombre d'accidents et sauver des vies.

Ce projet vise à développer un système de détection de la somnolence qui capture les images du conducteur et permet d'analyser des caractéristiques spécifiques, telles que le clignement des yeux ou d'autres indicateurs physiologiques, afin de déterminer l'état de vigilance du conducteur.

Pour atteindre cet objectif, le travail demandé comprend les étapes suivantes :

- 1) Téléchargement ou création de vidéos : Collecter des vidéos du conducteur en situation de conduite.
- 2) Détection des caractéristiques somnolentes : Identifier les traits physiologiques, comme les yeux ou d'autres caractéristiques, qui indiquent la somnolence.
- 3) Utiliser une méthode machine Learning ou autre qui permet de classer le conducteur (somnolent ou pas).

Ce projet, encadré par Mme Daoud Siham, est réalisé dans le cadre de la spécialité Systèmes Informatiques à l'École nationale supérieure d'informatique d'Alger pour l'année académique 2023-2024. Il est le fruit du travail conjoint de Messar Cylia et Tagzirt Elissa.

### **Étude de l'existant :**

- 1) **Marché automobile :**

Sur le marché automobile, on remarque déjà des systèmes intégrés pour assurer la détection de somnolence ainsi que la sécurité routière. Par exemple :

- Mercedes Attention Assist
  - BMW Driver Attention Camera
- 2) **Techniques de détection :**
- Suivi des yeux (Eye-tracking).
  - Analyse faciale.
  - Détection des signes physiologiques.
- 3) **Méthode de machine learning :**
- Modèles supervisés.
  - Modèles non supervisés.
  - Apprentissage profond (Deep Learning).

### **Méthode mise en oeuvre :**

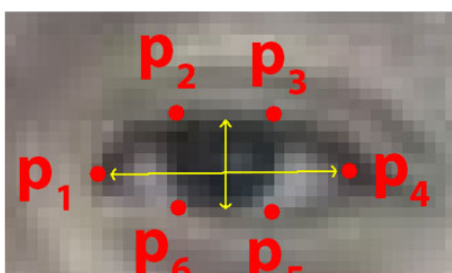
1) **Basée sur le rapport d'aspect des yeux (EAR) :**

Outil utilisé : [shape\\_predictor\\_68\\_face\\_landmarks.dat](#)

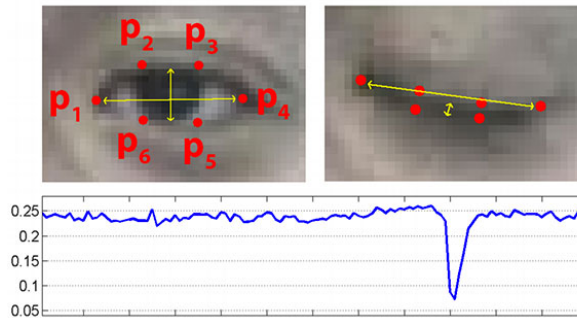
C'est quoi ce fichier et à quoi il sert?

- Un modèle qui identifie avec précision 68 points de repère faciaux, y compris les contours des yeux, des sourcils, du nez, de la bouche et de la mâchoire. Ces points sont cruciaux pour une analyse détaillée des caractéristiques faciales. Ce fichier de données contient un modèle de prédiction des formes basé sur un algorithme de machine learning, souvent entraîné avec la bibliothèque dlib.
- Il procède ainsi :
  - i) **Localisation** de visage dans l'image. Ceci se fait à l'aide d'un détecteur de visage basé sur des Histogrammes de Gradients Orientés (HOG) ou un modèle de réseau neuronal convolutif (CNN).
  - ii) **Initialisation** : Une fois que le visage est détecté, le modèle initialise les positions des points de repère à une position moyenne ou prédéfinie à l'intérieur de la boîte englobante du visage.
  - iii) **Affinement** : Le modèle affine progressivement ces positions initiales à travers plusieurs étapes d'arbre de régression. À chaque étape, un arbre de régression est utilisé pour prédire les ajustements nécessaires pour rapprocher les points de leur position finale.

Détection en utilisant **le rapport d'aspect des yeux(EAR) :**



$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$



**Seuil** est généralement initialisé à **0.25**.

**Avantages :**

- Simple est efficace.
- Robustesse aux Variations de Visage(robuste face aux variations de taille et de forme des yeux entre différents individus).
- Précision.
- Facile à intégrer.

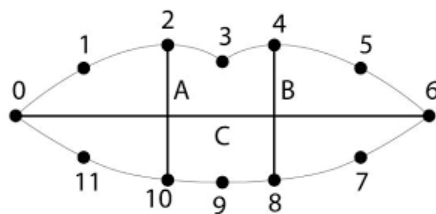
**Inconvénients :**

- Sensibilité aux conditions d'éclairage.
- Occlusions.
- Dépendance à l'initialisation.

## 2) Basée sur le rapport d'aspect de la bouche en complément au EAR :

Outil utilisé : le même fichier à été utilisé "*shape\_predictor\_68\_face\_landmarks.dat*".

Détection en utilisant **le rapport d'aspect de la bouche(MAR)** :



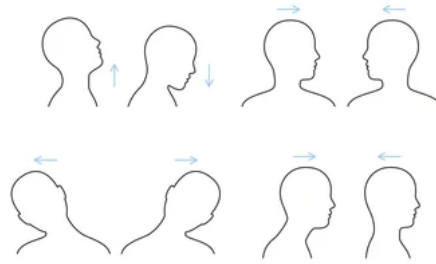
$$MAR = A + B / 2 C$$

**Seuil** est généralement initialisé à **0.79**.

**Avantages :**

- Complémentarité avec l'EAR.

En plus : **Calcul d'inclinaison de la tête.**



**Comment?**

- Détection des Points de Repère sur le Visage.
- **Définir les Points du Modèle 3D** : Pré Définissez les positions en 3D de ces points clés sur un modèle de tête. Par exemple :
  - Pointe du nez (0, 0, 0)
  - Menton (0, -330, -65)
  - Coins des yeux et de la bouche avec des coordonnées fixes.
- **Correspondance 3D-2D** : Faites correspondre les points détectés sur le visage (2D) avec les points du modèle 3D prédéfini.
- **Calcul des Vecteurs de Rotation et de Translation** : Utilisez une méthode appelée solvePnP pour estimer la position et l'orientation de la tête. Cela donne un vecteur de rotation (qui indique comment la tête est tournée) et un vecteur de translation (qui indique où se trouve la tête par rapport à la caméra).
- **Conversion du Vecteur de Rotation en Matrice de Rotation** : la matrice de rotation est une représentation plus facile à utiliser pour les calculs.
- **Obtention des Angles d'Euler** : Convertissez la matrice de rotation en angles d'Euler. Les angles d'Euler sont des rotations autour des axes X, Y et Z.

Par exemple, une rotation autour de l'axe X peut indiquer une inclinaison de la tête vers l'avant ou l'arrière.

**Seuil et paramètres :**

- **Légère inclinaison (0° à 10°)** : Peut être une position normale ou une légère somnolence.
- **Inclinaison modérée (10° à 20°)** : Peut indiquer un état de relaxation avancé ou un début de somnolence.
- **Forte inclinaison (20° à 30° et plus)** : Peut fortement suggérer un état de somnolence ou de sommeil imminent.

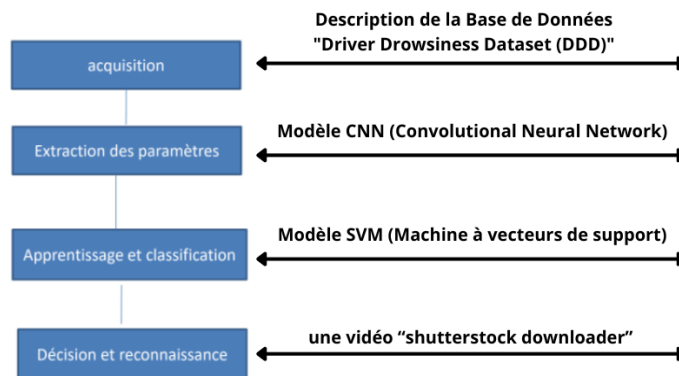
### 3) **Modèle supervisé (CNN et SVM) :**

Dans cette partie, nous détaillons la méthode mise en œuvre pour détecter la somnolence au volant, utilisant un modèle supervisé combinant les réseaux de

neurones convolutionnels (CNN) et les machines à vecteurs de support (SVM). Les étapes de construction de notre modèle de reconnaissance, telles que vues en cours, sont expliquées ci-dessous :

**1. Acquisition :**

Nous avons collecté des données à partir de la base de données "Driver Drowsiness Dataset (DDD)", qui contient des images de conducteurs somnolents et non somnolents.



**2. Extraction des paramètres :**

Nous avons utilisé un modèle CNN pour extraire des caractéristiques visuelles à partir des images de visages, permettant de détecter des motifs spécifiques associés à la somnolence.

**3. Apprentissage et classification :**

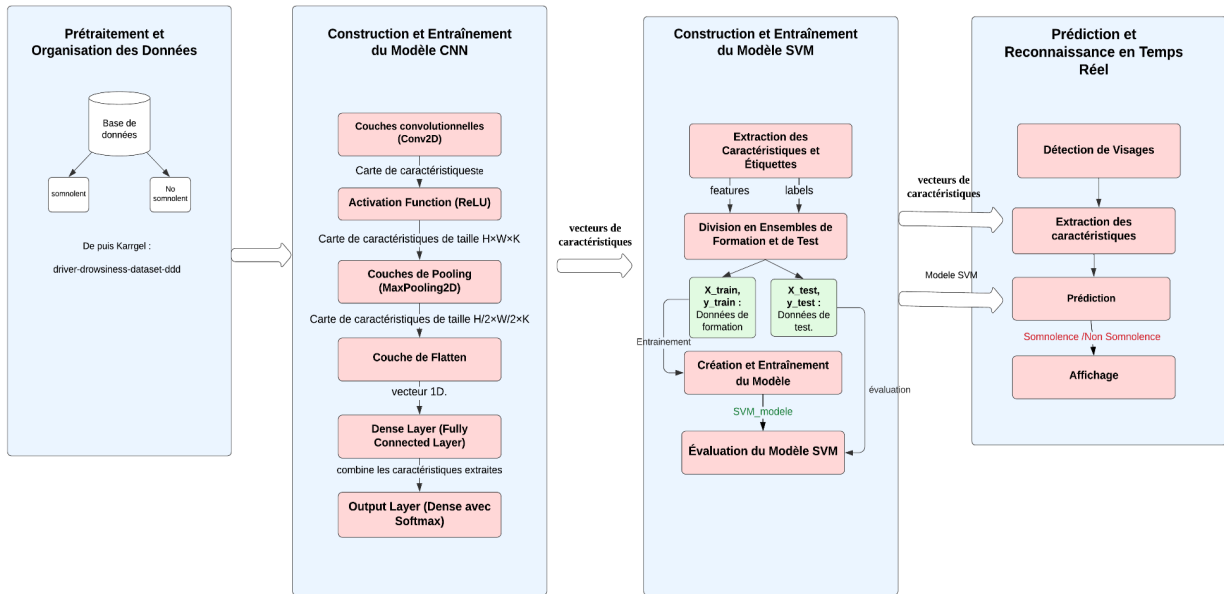
Nous avons entraîné un modèle SVM à l'aide des vecteurs de caractéristiques extraits par le CNN pour classifier les images en fonction de l'état de vigilance du conducteur (somnolent ou non somnolent).

**4. Décision et reconnaissance :**

Nous avons appliqué le modèle SVM sur de nouvelles images pour prédire et reconnaître en temps réel les cas de somnolence, permettant ainsi de prendre des mesures appropriées pour assurer la sécurité du conducteur.

**Le schémas globale :**

## L'architecture de détection de somnolence



### 1) Préparation de la Base de Données :

Dans cette première étape, nous avons préparé la base de données "**Driver Drowsiness Dataset (DDD)**" en organisant les images selon leurs classes respectives. Cette base de données est destinée à la détection de la somnolence chez les conducteurs et contient les données suivantes :

- **Somnolent** : Images de conducteurs montrant des signes de somnolence (yeux fermés, bâillements).
  - **Non Somnolent** : Images de conducteurs éveillés et alertes (yeux ouverts, expressions vigilantes).
- Source des Données : Téléchargeable depuis Kaggle sous le nom "driver-drowsiness-dataset-ddd".

### 2) Modèle CNN (Convolutional Neural Network) :

Nous avons utilisé le modèle CNN (Convolutional Neural Network) pour la détection de la somnolence. Le modèle CNN est conçu pour extraire des caractéristiques visuelles à partir des images de visages et apprendre à détecter des motifs spécifiques pouvant indiquer la somnolence. Voici une description détaillée des différentes parties du modèle :

**Entrée** : Images de visages prétraitées.

#### 1. Couches convolutionnelles (Conv2D) :

Les couches convolutionnelles appliquent des filtres de convolution pour extraire des caractéristiques locales comme les bords, les textures et les motifs dans les images.

- **Formule :**

$$Y = f(W * X + b)$$

$$Y[i, j, k] = f \left( \sum_m \sum_n \sum_c X[i + m, j + n, c] \cdot K[m, n, c, k] + b_k \right)$$

où X est l'image d'entrée, W est le filtre (noyau), b est le biais, et f est la fonction d'activation (ReLU dans ce cas).

Ces couches prennent une image d'entrée et appliquent plusieurs filtres pour produire des cartes de caractéristiques (feature maps). Chaque filtre extrait un type spécifique de caractéristique de l'image. Par exemple, une couche Conv2D avec 32 filtres de taille 3x3 détectera 32 motifs différents dans l'image d'entrée.

## 2. Activation Function (ReLU) :

La fonction d'activation ReLU (Rectified Linear Unit) introduit la non-linéarité dans le modèle pour permettre l'apprentissage de motifs complexes.

- **Formule :**

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

La fonction ReLU transforme toutes les valeurs négatives en zéro et laisse les valeurs positives inchangées. Cela aide le modèle à apprendre des motifs non linéaires, essentiels pour comprendre des structures complexes dans les images.

## 3. Couches de pooling (MaxPooling2D) :

Les couches de pooling réduisent la dimension des cartes de caractéristiques tout en conservant les informations les plus importantes.

- **Formule :**

$$Y = \max(X[i : i + 2, j : j + 2])$$

où X est la carte de caractéristiques d'entrée et Y est la carte de caractéristiques après pooling.

Ces couches prennent la valeur maximale dans chaque fenêtre de pooling de taille 2x2. Cela réduit la taille de l'image de moitié dans chaque dimension, diminuant le nombre de paramètres et de calculs nécessaires tout en conservant les informations



essentielles. Dans notre modèle, nous avons utilisé plusieurs couches de pooling après chaque couche de convolution pour successivement réduire la dimension des cartes de caractéristiques.

#### 4. Couche de flatten

La couche de flatten convertit les matrices 2D en un vecteur 1D. En effet, cette couche prépare les données pour les couches denses entièrement connectées en aplatissant les cartes de caractéristiques en un vecteur unique. Cela permet de passer des cartes de caractéristiques de dimension (hauteur, largeur, profondeur) à une seule dimension (vecteur), qui peut ensuite être traitée par des couches fully connected.

Si l'entrée est une matrice 2D de dimensions (h,w,d), où h est la hauteur, w est la largeur et d est la profondeur (nombre de filtres), la sortie sera un vecteur 1D de longueur  $h \times w \times d$ .

#### 5. Couches denses (Fully Connected Layer) :

Les couches denses appliquent une transformation linéaire suivie d'une activation.

- Formule : 
$$Y = W \cdot X + b$$

où W est la matrice de poids, X est le vecteur d'entrée, et b est le biais.

Ces couches fully connected combinent les caractéristiques extraites par les couches convolutionnelles et les transforment en un ensemble de sorties. Chaque neurone de la couche dense est connecté à tous les neurones de la couche précédente, permettant une intégration complète des caractéristiques.

#### 6. Couche de sortie (Dense avec Softmax) :

Cette couche a pour objectif de produire des probabilités pour chaque classe (Somnolence, Non Somnolence).

- Formule : 
$$P(y_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

où  $P(y_i)$  est la probabilité de la classe i et  $z_i$  est le score brut pour la classe i.

Cette couche prend les scores des classes et les convertit en probabilités qui somment à 1. La fonction softmax assure que la somme des probabilités est égale à 1, facilitant l'interprétation des résultats du modèle.

### 3) Modèle SVM (Machine à vecteurs de support):

Le modèle SVM est utilisé pour classifier les images en fonction de l'état de vigilance du conducteur (somnolent ou non somnolent). Voici les étapes détaillées du processus :

### 1. Extraction des Caractéristiques et Étiquettes :

Cette étape consiste à préparer les données d'entrée pour le modèle SVM en extrayant les caractéristiques pertinentes à partir des images à l'aide du CNN. Les étiquettes associées (somnolent ou non somnolent) sont également collectées.

**Entrée :**

- **Vecteurs de caractéristiques (features)** extraits par le CNN pour chaque image.
- **Étiquettes correspondantes (labels)** indiquant si l'image représente une personne somnolente ou non.

**Sortie :**

- **features** : Un tableau contenant les vecteurs de caractéristiques de toutes les images.
- **labels** : Un tableau contenant les étiquettes correspondantes.

### 2. Division en Ensembles de Formation et de Test :

Les données sont divisées en ensembles de formation et de test. L'ensemble de formation est utilisé pour entraîner le modèle, tandis que l'ensemble de test est utilisé pour évaluer les performances du modèle sur des données qu'il n'a pas vues pendant l'entraînement.

**Entrée :**

- **features** : Le tableau des vecteurs de caractéristiques.
- **labels** : Le tableau des étiquettes.

**Sortie :**

- **X\_train, y\_train** : Données de formation.
- **X\_test, y\_test** : Données de test.

### 3. Entraînement du SVM :

Le modèle SVM est entraîné à l'aide des données de formation pour trouver l'hyperplan optimal qui sépare les deux classes avec la plus grande marge. La fonction de coût Hinge Loss est utilisée pour ajuster les paramètres du modèle afin de minimiser les erreurs de classification.

#### a. Création et Entraînement du Modèle

Entrée :

- **X\_train** : Vecteurs de caractéristiques pour l'entraînement.
- **y\_train** : Étiquettes pour l'entraînement.

Sortie :

- **svm\_model** : Le modèle SVM entraîné.

#### b. Formule de l'Hyperplan de Décision :

$$w \cdot x + b = 0$$

Le SVM trouve l'hyperplan qui sépare les deux classes (somnolent et non somnolent) avec la plus grande marge possible.

#### c. Fonction de Coût (Hinge Loss) :

$$\text{Loss}(x_i, y_i) = \max(0, 1 - y_i \cdot (w \cdot x_i + b))$$

Cette fonction de coût pénalise les points mal classés ou proches de l'hyperplan. Si  $y \cdot f(x)$  est inférieur à 1, la perte est positive, sinon elle est nulle.

### 4. Classification avec le Modèle SVM :

Une fois le modèle SVM entraîné, il est utilisé pour faire des prédictions sur l'ensemble de test. Chaque vecteur de caractéristiques est classé en fonction de sa position par rapport à l'hyperplan.

#### Prédiction sur les Données de Test

Entrée :

- **X\_test** : Vecteurs de caractéristiques pour le test.

Sortie :

- **y\_pred** : Prédictions du modèle SVM.

#### Formule de Prédiction :

$$y = \text{sign}(w \cdot x + b)$$

Le modèle SVM utilise l'hyperplan appris pour classer chaque vecteur de caractéristiques xxx comme somnolent (  $y=1$ ) ou non somnolent (  $y=-1$ ).

## 5. Évaluation du Modèle SVM et calcul de la Précision :

La précision du modèle est calculée en comparant les prédictions du modèle SVM avec les étiquettes réelles des données de test. La précision représente le pourcentage de prédictions correctes.

Entrée :

- **y\_test** : Étiquettes réelles des données de test.
- **y\_pred** : Prédictions du modèle SVM.

Sortie :

- **accuracy** : Précision du modèle SVM.

Formule

:

$$\text{Accuracy} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}}$$

Notre précision : 0.756

## 6. Sauvegarde du Modèle :

Une fois le modèle entraîné et évalué, il est sauvegardé pour une utilisation ultérieure. Cela permet de déployer le modèle en production ou de le recharger pour d'autres analyses.

Entrée :

- **svm\_model** : Le modèle SVM entraîné.

Sortie :

- Fichier contenant le modèle SVM sauvegardé.

## 4) Prédiction et Reconnaissance en Temps Réel :

### 1. Détection de Visages :

Nous avons utilisé la bibliothèque dlib pour détecter les visages dans chaque frame de la vidéo. dlib fournit des outils de détection de visage précis et rapides, permettant d'analyser chaque frame pour localiser les visages des conducteurs.

### 2. Extraction des Caractéristiques :

Nous avons utilisé le modèle CNN pour extraire les caractéristiques des visages détectés. Le CNN, entraîné auparavant, génère des vecteurs de caractéristiques à partir des images des visages.

### 3. Prédiction :

Nous avons utilisé le modèle SVM pour prédire si la personne est somnolente ou non à partir des caractéristiques extraites par le CNN. Le SVM classe chaque image en tant que "somnolente" ou "non somnolente".

#### **4. Affichage :**

Nous avons annoté les frames de la vidéo avec les résultats de la prédiction. Les résultats sont affichés en temps réel, avec des cadres autour des visages changeant de couleur en fonction de l'état de somnolence détecté.

#### **Conclusion :**

La combinaison des réseaux de neurones convolutionnels (CNN) et des machines à vecteurs de support (SVM) s'est révélée être une approche efficace pour la détection de la somnolence au volant. En utilisant le CNN, nous avons pu extraire des caractéristiques visuelles complexes à partir des images de visages, identifiant des motifs spécifiques indicateurs de somnolence. Ces caractéristiques ont ensuite été classées par le SVM, qui a appris à distinguer avec précision entre les états de somnolence et de vigilance.

Les étapes de ce processus comprennent la préparation des données, l'extraction des caractéristiques, l'entraînement du modèle et l'application du modèle pour la classification en temps réel. Cette méthodologie, basée sur une combinaison de techniques avancées de deep learning et de machine learning, a permis de développer un système robuste capable de détecter la somnolence avec une précision satisfaisante. Ce système peut être utilisé pour améliorer la sécurité routière en fournissant des alertes en temps réel aux conducteurs somnolents.

## **Conclusion Globale**

Ce projet de détection de la somnolence au volant, réalisé par Messar Cylia et Tagzirt Elissa sous la supervision de Mme Daoud Siham, a démontré l'efficacité des techniques de machine learning pour améliorer la sécurité routière. En combinant les capacités des réseaux de neurones convolutionnels (CNN) et des machines à vecteurs de support (SVM), nous avons développé un système capable de détecter les signes de somnolence chez les conducteurs en temps réel.

Le projet a impliqué une série d'étapes cruciales, de la collecte des données à la mise en œuvre des modèles de classification, en passant par l'entraînement et l'évaluation des performances. Grâce à une approche méthodique et à l'utilisation de technologies de pointe, nous avons pu atteindre une précision de détection satisfaisante et créer un système pratique et efficace pour la détection de la somnolence. Ce travail, réalisé dans le cadre de la spécialité Systèmes Informatiques à l'École Nationale Supérieure d'Informatique d'Alger pour l'année académique 2023-2024, illustre le potentiel des solutions de machine learning pour répondre à des enjeux réels et critiques de la société.

Ce projet ouvre la voie à des améliorations futures et à une intégration possible dans des systèmes de sécurité automobile pour prévenir les accidents causés par la fatigue au volant, contribuant ainsi à sauver des vies et à rendre les routes plus sûres.