

<b>Mini Projet Intelligence Artificielle Avancée</b>
<b>Classe 3 IDL</b>
<b>Mr. Arafet ZOUARI</b>
<b>Année universitaire 2024 - 2025</b>

## Objectif

Construire un modèle d'apprentissage pour détecter les infractions (violations) au code de la route ciblant spécifiquement les feux de circulations et les panneaux (par exemple feux rouges, panneaux stop ect). Nous aurons une **interface graphique** à partir de laquelle nous pourrons tester différentes images pour savoir la pertinence du système.

### Base de données

#### [DataSet](#)

**Nom des classes:** Green Light, Red Light, Speed Limit 10, Speed Limit 100, Speed Limit 110, Speed Limit 120, Speed Limit 20, Speed Limit 30, Speed Limit 40, Speed Limit 50, Speed Limit 60, Speed Limit 70, Speed Limit 80, Speed Limit 90, Stop

### Étape 1: Définition de l'architecture du modèle CNN:

Nous devons définir à quoi ressemblera notre modèle et cela nécessite de répondre à des questions telles que :

- Combien de couches convolutives voulons-nous ?
- Quelle doit être la fonction d'activation de chaque couche ?
- Combien d'unités cachées doit avoir chaque couche.

## **Étape 2 : Entraînement du modèle:**

La base de données est composée d'images d'entraînement accompagnées de leurs étiquettes réelles correspondantes, d'images de test et de validation, ainsi que de leurs étiquettes réelles correspondantes. Nous définissons également le nombre d'époques à utiliser lors de l'entraînement. Pour commencer, nous allons exécuter le modèle pendant 10 époques (ce nombre d'époques peut être modifié par la suite) .

NB: les étiquettes sont en **format Darknet**.

## **Etape 3 : Prédiction des images:**

Nous chargeons les données de test (images) et prédisons ensuite les classes pour ces images

## **Etape 4 : Estimation des performances du modèle**

Enfin, vous générez un matrice de confusion ainsi qu'afficher l'accuracy de votre modèle

## **Environnement :**

Vous pouvez utiliser l'environnement et le langage de programmation de votre choix.