

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the word 'Sécurité'.

Sécurité

Cahier de laboratoire : détection d'obstacles en milieu urbain

*Génération d'un plan de navigation à partir d'une
scène #PFE23-R-198*

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

Auteurs : Alexis MARIE, David MARCHES, Luca BANKOFSKI, Martial BROSTIN, Théo HELLER, Thomas MABILLE

Objectif de l'expérience

Développer et valider un système utilisant le deep learning pour détecter en temps réel les obstacles et ralentissements afin d'améliorer la sécurité des véhicules. L'objectif est d'atteindre une précision de détection supérieure à 60% avec un temps de traitement inférieur à 1 seconde (le temps de réaction d'un humain).

Sommaire

Objectif de l'expérience	1
Sommaire	1
Expérience 1 : Reconnaissance d'image avec YOLO v8n	2
Expérience 2 : Détection de la dangerosité avec RNN*	5
Références :	6

Expérience 1 : Reconnaissance d'image avec YOLO v8n

Protocole expérimental :

- Matériels :
 - MacBook Air M2 ;
 - YOLO v8n ;
 - DFG Traffic Sign dataset.
- Méthodes :
 - Entraînement du modèle YOLO avec le dataset de panneaux de signalisation après conversion au format adéquat ;
 - Évaluation des performances en termes de précision et temps d'inférence.

Observations :

- 10/01/2024 : Les données initialement au format COCO doivent être transférées à un autre format appelé YOLO pour pouvoir entraîner le modèle sur ces données. Il a fallu faire un parsing des données COCO au format JSON pour les transformer au format .txt (un fichier par image) recensant chaque rectangle normalisé contenant un objet et la classe de l'objet contenu dans le rectangle.

Exemple du contenu d'un fichier .txt au format YOLO :



Les étiquettes de ce format doivent être exportées au format YOLO avec une *.txt par image. S'il n'y a pas d'objets dans une image, aucune *.txt est nécessaire. Le fichier *.txt doit être formaté avec une ligne par objet en class x_center y_center width height format. Les coordonnées de la boîte doivent être en xywh normalisées (de 0 à 1). Si les cases sont en pixels, il faut les diviser x_center et width par la largeur de l'image, et y_center et height par la hauteur de l'image. Les numéros de classe doivent être indexés à zéro (commencer par 0).

Le fichier d'étiquettes correspondant à l'image ci-dessus contient 2 personnes (classe 0) et une égalité (classe 27) :

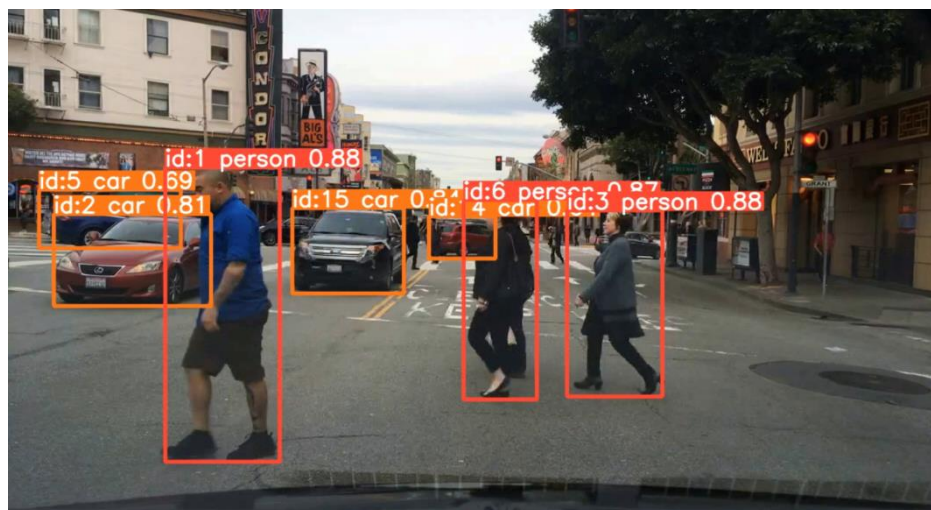
```
zidane.txt
0 0.481719 0.634028 0.690625 0.713278
0 0.741094 0.524306 0.314750 0.933389
27 0.364844 0.795833 0.078125 0.400000
```

Pour plus d'information, se référer à la documentation YOLO
(<https://docs.ultralytics.com/fr/datasets/detect/#ultralytics-yolo-format>)

- 17/01/2024 : Test du modèle Yolo v8 sur des vidéos sans fine tuning. On cherche à connaître l'efficacité du modèle pour la reconnaissance de voitures et de piétons en analysant 50 vidéos du dataset de vidéo car Crash et en vérifiant pour chaque frame la bonne détection des objets voulu. On comptera les vidéos une fois pour les objets proches et une autre fois pour la totalité de l'image.

- 22/01/2024 : Les données ainsi que le code pour l'entraînement sont fonctionnels. Cependant les ressources informatiques à disposition ne sont pas suffisantes pour nous permettre d'entraîner le modèle.

Données :



Reconnaissance des personnes:

	Détection	Certitude
Globale	21,3 %	69,2 %
Proche	83,0 %	84,1 %

Reconnaissance des voitures :

	Détection	Certitude
Globale	56,7 %	73,2 %
Proche	92,1 %	82,9 %

Temps de traitement : 111ms

Taux de vision globale : 47,4%

Taux de certitude globale : 72,6%

Taux d'erreur : 2,6%

Analyses préliminaires :

Les résultats obtenus sont satisfaisants pour le champ proche, cependant pour les objets se situant plus loin, les résultats ne sont pas aussi satisfaisants.

La qualité de l'image ainsi que l'entraînement du modèle peuvent être des paramètres qui expliquent cette différence de performance entre le champ proche et le champ global (impacté par la faible performance sur le champ lointain).

Le taux de certitude représente la certitude qu'a le modèle qu'un objet appartienne bien à la classe qu'il lui a assigné. Le taux de certitude est bon.

Le taux d'erreur représente le pourcentage d'erreurs recensées. Il est important de noter que certaines erreurs viennent de la confusion de deux classes, souvent entre une voiture et un camion ou un bus. Ce taux d'erreur est donc très satisfaisant.

Le temps de traitement pour la partie détection d'objet est bien inférieur au temps de traitement de l'œil humain, ce résultat est satisfaisant.

Conclusion :

- Le modèle YOLO sans fine tuning est très efficace pour détecter les voitures, les personnes et même certains signaux de signalisation. Il faudrait cependant plus de ressources pour réussir à détecter avec plus de précision le sens des panneaux détectés.

Expérience 2 : Détection de la dangerosité avec RNN*

Protocole expérimental :

- Matériels :
 - MacBook Air M2 ;
 - LSTM RNN ;
 - Car Crash dataset.

- Méthodes :
 - Extraction des caractéristiques visuelles avec le modèle YOLO pré-entraîné ;
 - Suivi de la détection de situations dangereuses avec un RNN.

Observations :

Notes sur le traitement des vidéos et la préparation des données pour le RNN.

Commentaires sur les performances du RNN en termes de détection des situations dangereuses.

Données :

Enregistrement des sorties du RNN, y compris les faux positifs et les faux négatifs.

Graphiques illustrant les performances du RNN sur le temps.

Analyses préliminaires :

Analyse des tendances dans la détection des situations dangereuses et des possibles améliorations du modèle.

Conclusion :

Évaluation de la capacité du RNN à déterminer la dangerosité des situations. Recommandations pour des ajustements futurs et optimisation du modèle.

Références :

[1] DFG Traffic Sign dataset

[2] Car Crash dataset