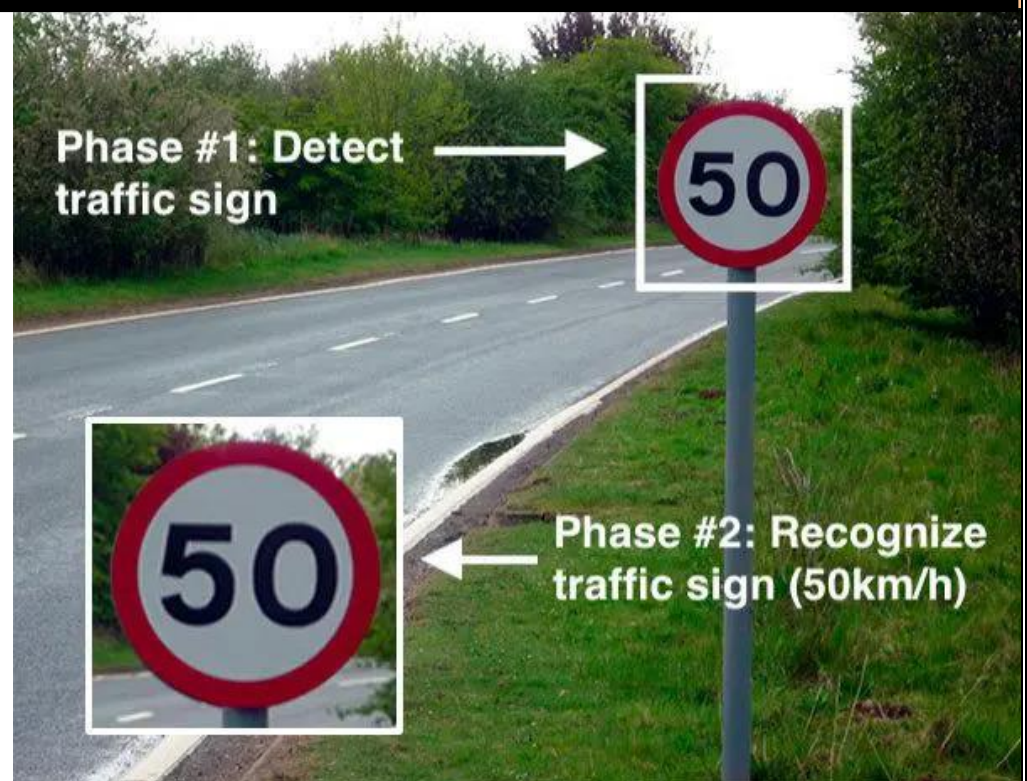


Descriptif sur l'élaboration du modèle informatique de Traffic Sign Recognition Use Case



CONTENUE DU DESCRIPTIF

INTRODUCTION	2
Présentation du modèle de reconnaissance de signalisation.....	2
La DATASET utilisée	2
Le choix de la méthodologie CNN pour l'apprentissage automatique	3
L'utilisation de Tensorflow et Keras.....	3
Pré-traitement des données (Data pre-processing).....	4
Le module de "Traffic Sign Recognition"	5
Résultats Expérimentales	5
Résultats préliminaires en temps réel.....	7
Conclusion.....	9

INTRODUCTION

Présentation du modèle de reconnaissance de signalisation

Notre projet de reconnaissance de panneaux de signalisation à l'aide de l'apprentissage profond avec une convolutional neural network (CNN) a été réalisé dans le cadre de la compétition Smart Mobility. Cette compétition vise à encourager le développement de solutions innovantes pour améliorer la sécurité et l'efficacité des systèmes de transport, en utilisant les technologies de l'information et de la communication. Notre objectif était donc de développer un modèle de machine learning capable de détecter efficacement les différents types de panneaux de signalisation présents sur les routes, afin d'améliorer la sécurité des usagers de la route et de faciliter la gestion du trafic.

La DATASET utilisée

Pour cela, nous avons utilisé la dataset "The German Traffic Sign Detection Benchmark", qui est l'une des plus grandes datasets disponibles pour la reconnaissance de panneaux de signalisation. Cette dataset comprend des milliers d'images de panneaux de signalisation capturées dans des conditions réelles de circulation, avec une grande variété de tailles, de formes, de couleurs et de conditions d'éclairage. Nous avons utilisé cette dataset pour entraîner et tester notre modèle de CNN de reconnaissance de panneaux de signalisation.

Lien du DATASET : https://benchmark.ini.rub.de/gtsdb_dataset.html



Le choix de la méthodologie CNN pour l'apprentissage automatique

L'utilisation de l'apprentissage profond avec une CNN présente de nombreux avantages pour la reconnaissance de panneaux de signalisation. Contrairement aux méthodes traditionnelles de traitement d'images, qui nécessitent une sélection manuelle de caractéristiques pertinentes, l'apprentissage profond permet d'extraire automatiquement les caractéristiques discriminantes des images, ce qui améliore la précision de la détection. De plus, les CNN sont capables de traiter efficacement les images en haute résolution, ce qui permet de détecter les panneaux de signalisation à une grande distance et dans des conditions de faible éclairage. Enfin, la méthode de l'apprentissage profond est très flexible et peut être adaptée à différents types de panneaux de signalisation, en ajoutant simplement de nouvelles données d'entraînement au modèle.

L'utilisation de Tensorflow et Keras

L'utilisation de TensorFlow et Keras pour l'élaboration de notre modèle de reconnaissance de panneaux de signalisation présente plusieurs avantages. Tout d'abord, TensorFlow est particulièrement adapté à la conception de modèles de CNN, car elle fournit des outils pour la gestion des tenseurs et des opérations de convolution, ainsi que pour la définition de graphes de calculs. En outre, TensorFlow permet l'utilisation de plusieurs processeurs et de cartes graphiques pour accélérer les calculs et l'entraînement du modèle, ce qui peut être particulièrement utile pour les jeux de données volumineux.



Simple. Flexible. Powerful.

Keras, quant à elle, facilite grandement la définition et l'entraînement de modèles de CNN. Elle fournit des couches pré-définies pour les réseaux de neurones, ainsi que des outils pour la compilation et l'optimisation des modèles. En outre, Keras est hautement configurable, ce qui permet de personnaliser facilement les modèles et d'expérimenter différentes architectures pour améliorer les performances.

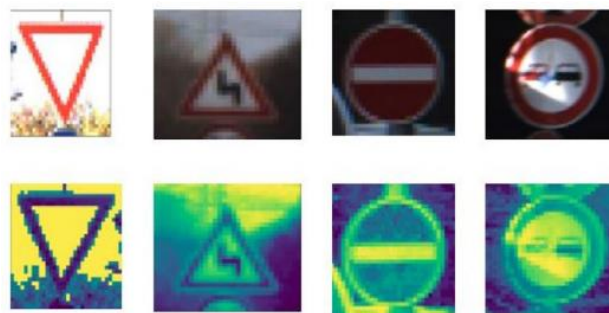
En résumé, l'utilisation de TensorFlow et Keras pour la reconnaissance de panneaux de signalisation à l'aide de CNN permet de tirer parti des avantages de ces deux bibliothèques pour obtenir un modèle de haute qualité et performant. TensorFlow fournit les outils nécessaires pour concevoir et optimiser des modèles de CNN, tandis que Keras facilite la définition et l'entraînement de ces modèles en permettant une configuration rapide et une personnalisation facile.

Pré-traitement des données (Data pre-processing)

Dans la phase de détection, les annotations ROI de l'image ont été converties au format Pascal VOC à partir des annotations au format CSV et la même chose a également été faite pour le modèle de classification. Dans la phase de classification, les images ont été redimensionnées à 32x32, puis converties en niveaux de gris, puis les images ont été normalisées. Le redimensionnement des images était dû au fait que la taille de l'image de l'ensemble de données variait de 15x15 à 250x250 pixels. De plus, étant donné que notre CNN ne reçoit que des images d'entrée de même taille, toutes les images doivent être redimensionnées à une taille spécifique avant d'être transmises au réseau. Ensuite, la conversion des images en niveaux de gris était due au fait que les couleurs des images ne sont pas un facteur déterminant très important pour la classification de l'image. Et cela réduit également la complexité du traitement du CNN. Ensuite, la normalisation des images a été faite en divisant chaque pixel par le pixel maximum. Cela garantit que les pixels d'entrée ont une distribution de données analogue. Cela accélère également la convergence lors de la formation du réseau.

Des techniques d'augmentation d'image telles que les décalages, la luminosité et le zoom ont été utilisées pour uniformiser la distribution du nombre d'images par classe afin d'améliorer la précision de la classification et de réduire les biais. Le réseau de détection en deux étapes est plus précis dans la détection des boîtes englobantes et des objets de classe.

Les résultats du pre-processing sont présentés dans la figure suivante.



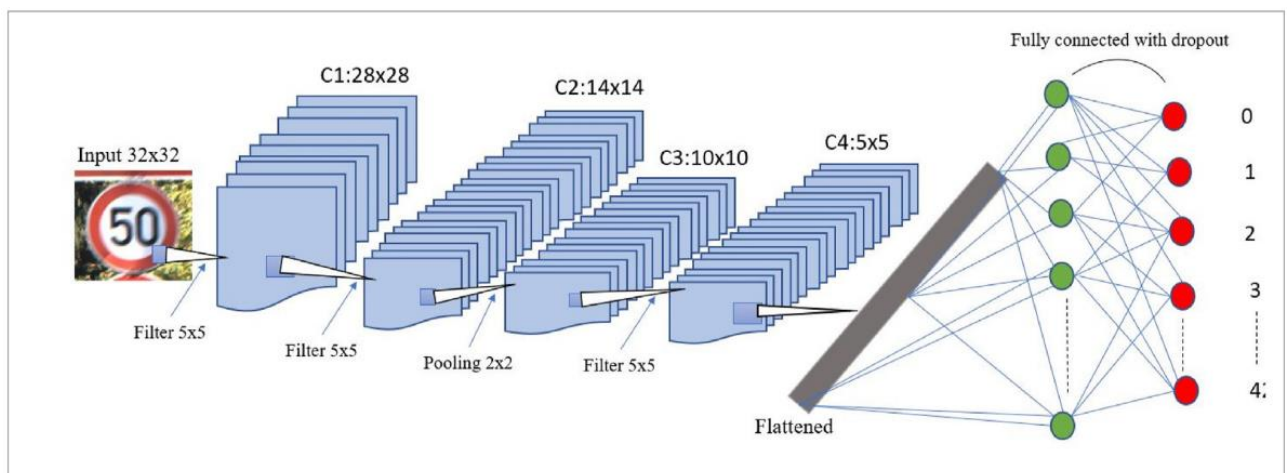
La répartition des images dans les 43 classes est illustrée dans la figure ci-dessous :



Le module de “Traffic Sign Recognition”

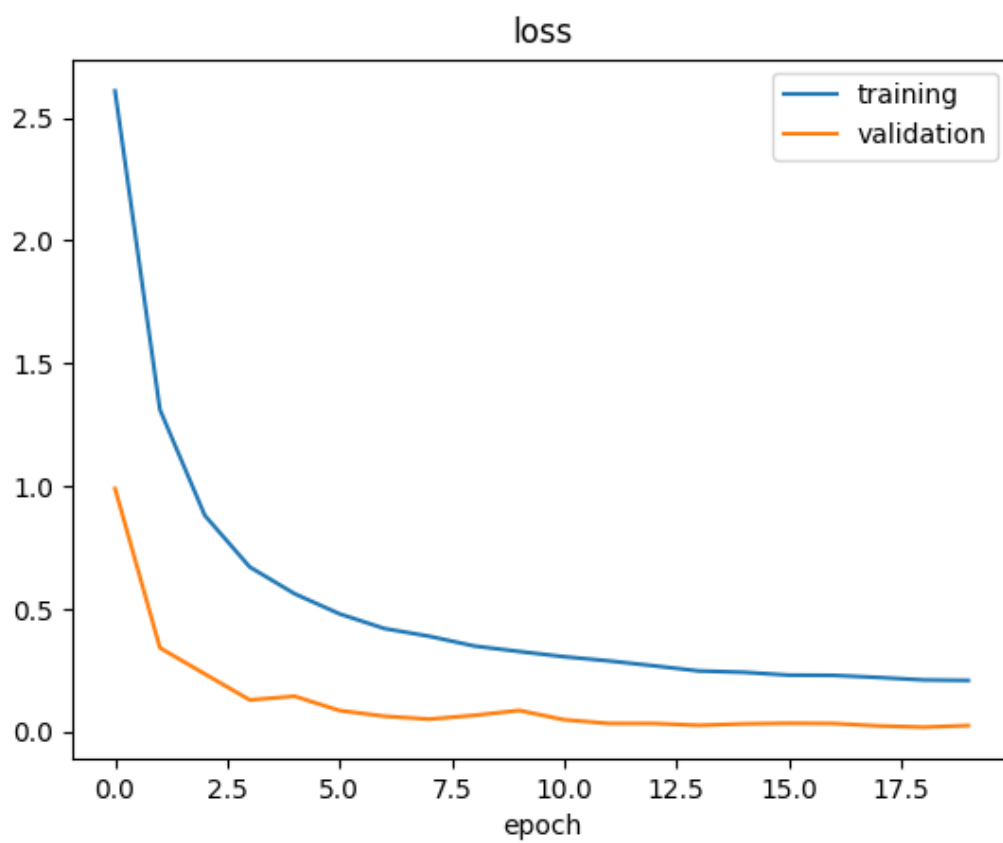
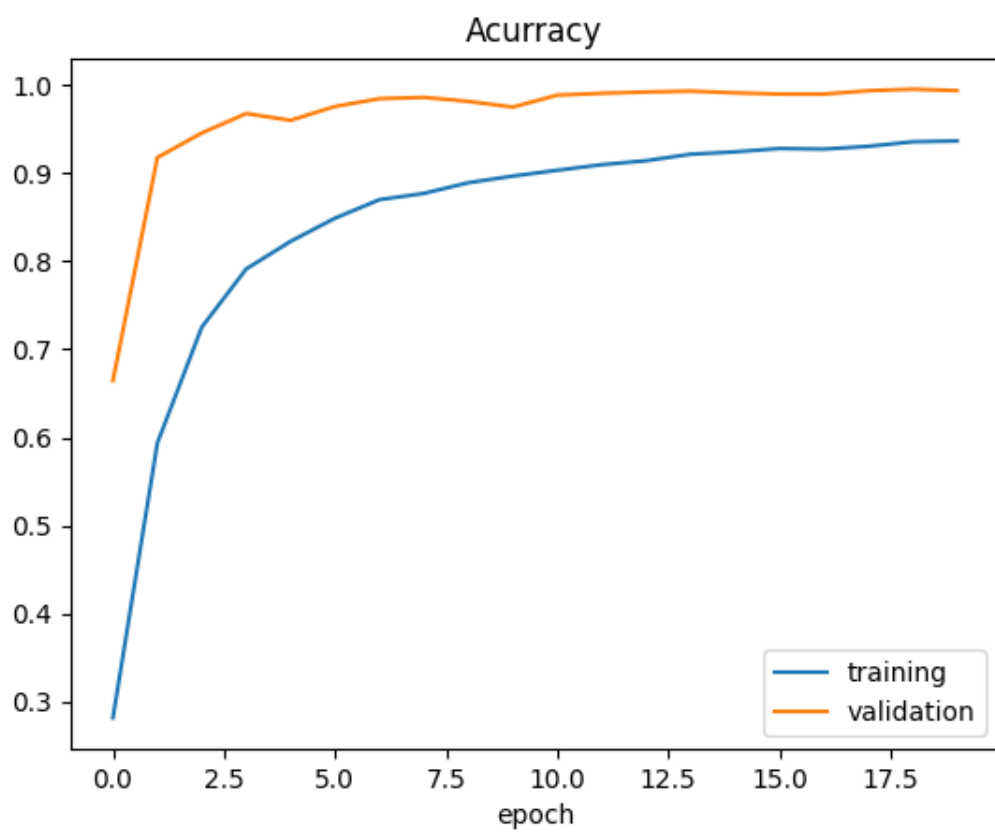
Pour le module de classification, la GTSRB a été utilisée. L'ensemble de données a été divisé selon un ratio de 20 % pour les tests, 20 % pour la validation et 60 % pour la formation. Le nombre d'images dans chaque classe n'est pas réparti uniformément, par conséquent, une technique d'augmentation a été utilisée pour augmenter le nombre d'ensembles d'apprentissage. La classe « Limite de vitesse 20 km/h », représentée par 0 sur la figure ci-dessus, compte 210 images, tandis que la classe « Limite de vitesse 50 km/h », représentée par 2, compte 2250 images. En raison de ces écarts, le modèle peut devenir biaisé en faveur de la classe avec plus d'images. Alors les différents paramètres d'augmentation comprenaient la rotation aléatoire, l'étirement et les retournements sont appliqués. Ces paramètres d'augmentation sont utilisés pour équilibrer l'ensemble de données afin de réduire les biais. A partir du jeu de données, les images se succèdent pour la même classe, du fait de cette présence, un brassage aléatoire a été effectué sur le jeu de données pour éviter la fluctuation des fonctions d'entraînement et de perte. Un réseau neuronal CNN a été construit pour effectuer l'extraction et la classification des caractéristiques sur l'ensemble d'apprentissage.

La figure ci-dessous représente la méthodologie CNN utilisée pour la classification :



Résultats Expérimentales

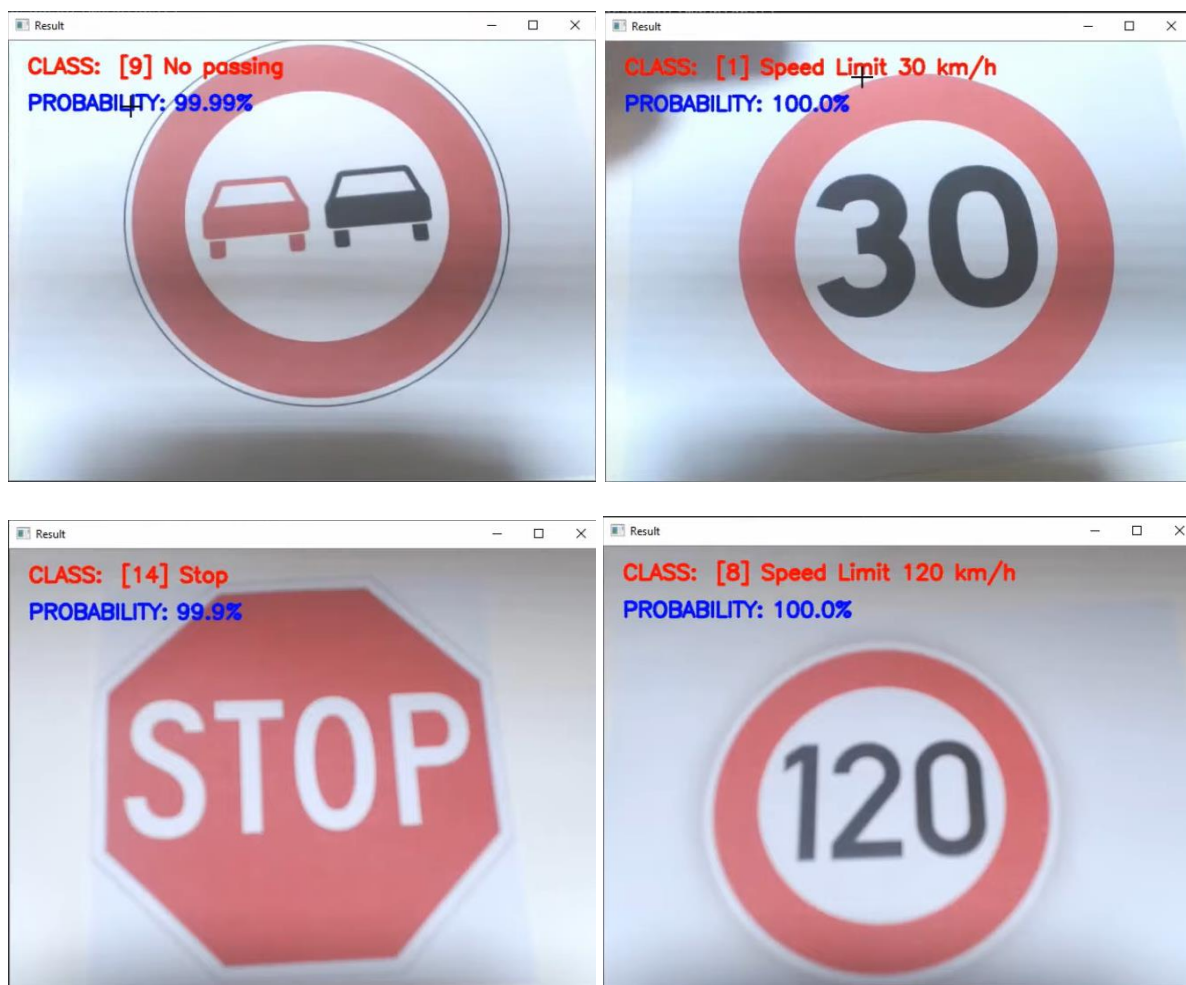
Vers la fin de notre classification, le CNN développé a atteint une bonne précision de 99,20 % sur l'ensemble de données de test GTSRB. Les graphiques de précision (accuracy) et de dispersion (Loss) pour la formation et la validation du modèle sont illustrés sur les courbes ci-après.



Notre modèle a pris environ 6,63 s pour classer toutes les images de l'ensemble de test. L'ensemble de test se compose de 12360 images ; par conséquent, le modèle prend 0,14 ms pour classer une seule image. Il est considérablement rapide pour les applications qui ne sont pas en temps réel. La comparaison entre notre CNN et d'autres classifications d'images CNN de pointe est donnée dans le tableau suivant :

Model name	Time	Loss	Accuracy	No. of Parameters
Our CNN	6.631 s	0.031	99.20%	0.8 M
Enet-V1 [12]	7.794 s	0.064	98.69%	0.9 M
Enet-V2 [12]	3.090 s	0.2642	96.78%	0.31 M
MCDNN [24]	11.4	0.024	99.46%	38.5 M
Co. CNNs [16]	-	-	99.35%	5.22 M

Résultats préliminaires en temps réel





Conclusion

Dans ce descriptif, une nouvelle architecture CNN a été conçue pour élaborer un bon modèle pour notre Use Case de la reconnaissance des panneaux de signalisation, et a été testée sur l'ensemble de la DATASET de GTSRB et a atteint une très grande précision de 99,20 % avec une dispersion minimale. Bien que ce modèle s'agit d'un modèle léger qui peut être utilisé dans des ordinateurs avec de petites ressources pour la reconnaissance des panneaux de signalisation, et bien évidemment pour notre cas de voiture autonome équipée d'une caméra USB et une unité de commande et de processing comme Raspberry Pi 4. Cette haute précision et robustesse peut être justifiée par les techniques de pré-traitement qui ont été utilisées avant la formation du CNN. La DATASET GTSRB a été conçu à l'origine pour la reconnaissance de 3 super classes, à savoir les panneaux d'interdiction, d'obligation et de danger. L'ensemble de données se compose de 600 images pour la formation et de 300 images pour les tests. Il y a en moyenne 200 images par catégorie en formation. Mais notre objectif n'était pas seulement de détecter les 3 super classes mais les 43 classes du panneau de signalisation allemand en utilisant le même ensemble de données.