

Reconnaissance des panneaux de signalisation

ce projet est un mini projet de intelligence artificielle du premier simestre de l'ingénieur 4 info GL . Vous pouvez trouver tout le code lié à ce projet sur [github](#).

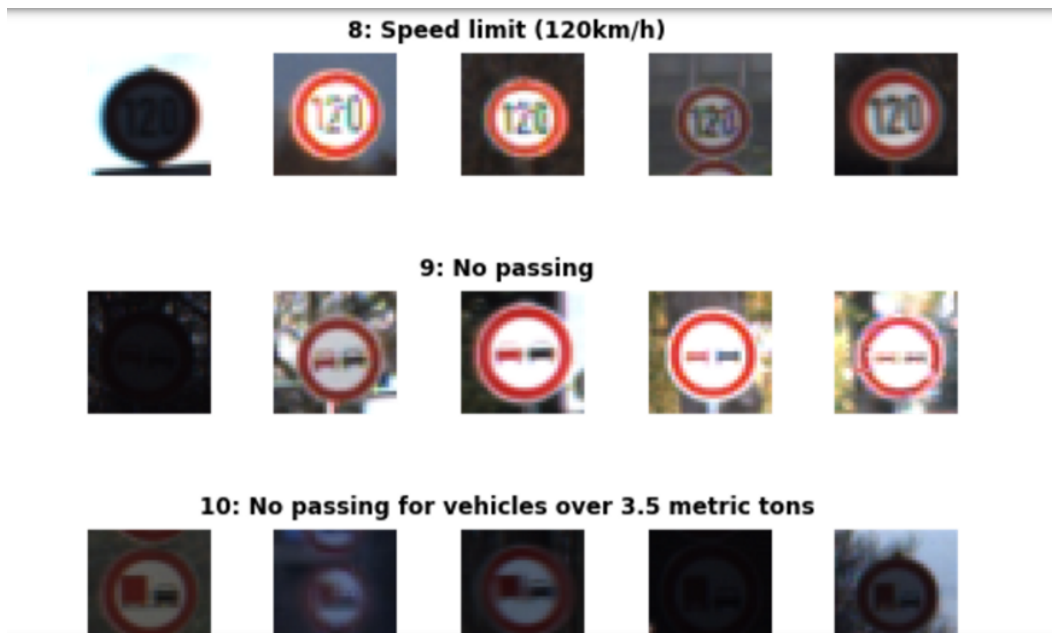
Introduction

Les panneaux de signalisation font partie intégrante de notre infrastructure routière. Ils fournissent des informations critiques, parfois des recommandations convaincantes, aux usagers de la route, ce qui les oblige à leur tour à ajuster leur comportement de conduite pour s'assurer qu'ils respectent la réglementation routière actuellement en vigueur. Sans ces panneaux utiles, nous serions très probablement confrontés à plus d'accidents, car les conducteurs ne recevraient pas de commentaires critiques sur la vitesse à laquelle ils pourraient rouler en toute sécurité, ni informés des travaux routiers, des virages serrés ou des passages à niveau scolaire à venir. À notre époque moderne, environ 1,3 million de personnes meurent chaque année sur les routes. Ce nombre serait beaucoup plus élevé sans nos panneaux de signalisation. Naturellement, les véhicules autonomes doivent également respecter la législation routière et donc reconnaître et comprendre les panneaux de signalisation.

Traditionnellement, des méthodes standard de vision par ordinateur étaient utilisées pour détecter et classer les panneaux de signalisation, mais celles-ci nécessitaient un travail manuel considérable et fastidieux pour créer à la main des caractéristiques importantes dans les images. Au lieu de cela, en appliquant le Deep Learning à ce problème, nous créons un modèle qui classe de manière fiable les panneaux de signalisation, apprenant à identifier les caractéristiques les plus appropriées pour ce problème par lui-même. Dans cet article, je montre comment créer une architecture Deep Learning capable d'identifier les panneaux de signalisation avec une précision de près de 95 % sur l'ensemble de test.

Images et distribution

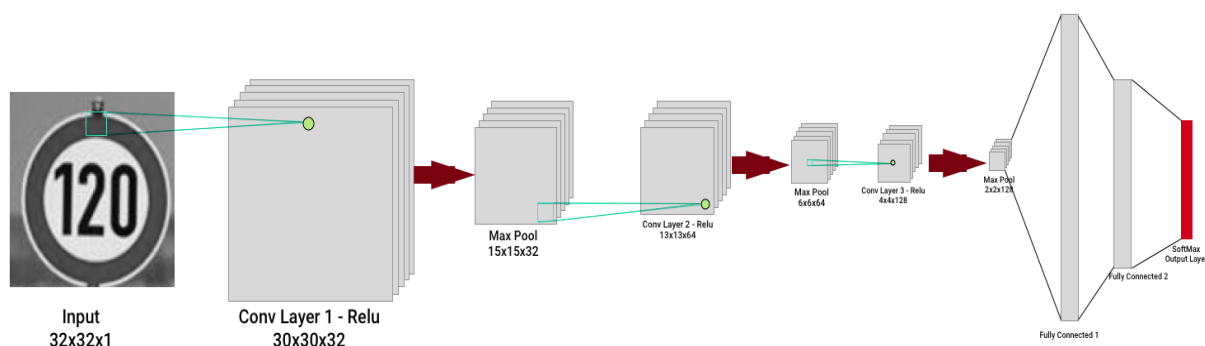
Vous pouvez voir ci-dessous un échantillon des images de l'ensemble de données, avec des étiquettes affichées au-dessus de la rangée d'images correspondantes.



Architecture du modèle

Nous avons ajouté quelques ajustements et créé une base de code modulaire qui nous permet d'essayer différentes tailles de filtre, profondeur et nombre de couches de convolution, ainsi que les dimensions des couches entièrement connectées

Nous avons principalement essayé des tailles de filtres 5x5 et 3x3, et commençons avec une profondeur de 32 pour notre première couche convolutive. l'architecture est présentée ci-dessous :



Le réseau est composé de plusieurs couches convolutives - la taille du noyau est de 3x3, avec un doublement de la profondeur au niveau de la couche suivante - utilisant ReLU comme fonction d'activation, chacune suivie d'une opération de pooling max 2x2. Les 3 dernières couches sont entièrement connectées, la dernière couche produisant 43 résultats (le nombre total d'étiquettes possibles) calculés à l'aide de la fonction d'activation SoftMax.

Le réseau est formé à l'aide de la categorical_crossentropy avec l'optimiseur Adam

Afin d'améliorer la fiabilité du modèle, nous nous sommes tournés vers le dropout, qui est une forme de régularisation où les poids sont conservés avec une probabilité. les poids non conservés sont ainsi droppés. Cela empêche le modèle de overtraining.

Nous essayons des différentes configurations et trouvons les meilleurs résultats, avec une précision de test de 95.45 %, sur le modèle en utilisant les valeurs de dropout 0,25.

Test sur de nouvelles images

Nous avons également décidé de tester notre modèle sur de nouvelles images, pour nous assurer qu'il est bien généralisé à plus que les panneaux de signalisation de notre ensemble de données d'origine. Nous avons donc téléchargé des nouvelles images et les avons soumises à notre modèle pour des prédictions.



Les images ont été choisies pour les raisons suivantes :

- . Ils représentent différents panneaux de signalisation que nous classons actuellement
- . Ils varient en forme et en couleur
- . Ils sont dans des conditions d'éclairage différentes (le 4ème a une réflexion du soleil)
- . Ils sont sous des orientations différentes (la 3ème est inclinée)
- . Ils ont des antécédents différents

- . La dernière image est en fait un dessin, pas une image réelle, et nous voulions tester le modèle par rapport à elle
- . Certains d'entre eux sont dans des classes sous-représentées

Nous atteignons une précision parfaite de 100% sur les nouvelles images. 🎉🎉🎉

Conclusion

Nous avons expliqué comment l'apprentissage en profondeur peut être utilisé pour classer les panneaux de signalisation avec une grande précision, en utilisant diverses techniques de prétraitement et de régularisation et en essayant différentes architectures de modèle. Nous avons construit un code hautement configurable et développé un moyen flexible d'évaluer plusieurs architectures. Notre modèle a atteint près de 96% de précision sur l'ensemble de test.

Personnellement, j'ai vraiment apprécié ce projet et j'ai acquis une expérience pratique en utilisant Tensorflow, matplotlib et en étudiant les architectures de réseaux de neurones artificiels. De plus, j'ai exploré quelques articles fondateurs dans ce domaine, ce qui a renforcé ma compréhension et, plus important encore, affiné mon intuition sur le deep learning.