Société d'Automatique,
de Génie Industriel & de Productique

Implémenter et optimiser des réseaux de neurones pour assurer une perception temps-réel des véhicules autonomes

Nicolas Muller[1], Vasile Giurgi[1], Thomas Josso-Laurain[1], Maxime Devanne[1], Jean-Philippe Lauffenburger[1]
[1] IRIMAS, Université de Haute-Alsace, Mulhouse, Frances
nom.prenom@uha.fr

Mots-clés *: Véhicule autonome, réseaux de neurones, implémentation, optimisation*

1 Introduction

Les véhicules autonomes sont désormais un sujet scientifique et industriel, voire sociétal et politique. La bonne navigation d'un véhicule automatisé repose en grande partie sur la qualité de sa perception, i.e. sa capacité à analyser l'environnement qui l'entoure. Pour se faire, il repose sur ses capteurs embarqués et sur les récentes avancées en traitement de données par réseaux de neurones pour extraire des informations utiles à sa circulation.

Ce résumé présente un système de perception basé sur un réseau de neurones (DeepLabV3+) effectuant une segmentation de la route à partir d'une image caméra RGB. Différents datasets sont considérés pour entraîner le réseau, ainsi que différentes méthodes d'optimisation afin de garantir des performances temps-réel. Les résultats expérimentaux montrent l'efficacité de l'architecture à partir des données collectées sur le prototype du laboratoire IRIMAS.

2 Présentation des équipements matériels et logiciels

L'institut IRIMAS est équipé d'un prototype de véhicule autonome ARTEMIPS. Il s'agit d'un Renault Scénic instrumenté pour la perception et la commande des véhicules autonomes. Il dispose notamment d'une caméra MANTA pour l'acquisition des données, et d'un ordinateur DELL AlienWare M15 R6 avec une GPU NVIDIA RTX 3080 pour le traitement temps-réel. Le logiciel de temps-réel RTMaps d'Intempora[1] permet de combiner acquisition et traitement sur la même interface. L'implémentation de réseaux de neurones peut se faire via le composant PythonBridge.

Plusieurs jeux de données de conduite de véhicule sont utilisés et comparés pour l'entraînement et la validation du réseau, parmi lesquels le jeu de données du KIT, KITTI [1] ; une base de données synthétique, CityScape [2] ; et un jeu de données issues de la caméra embarquée sur ARTEMIPS circulant dans l'agglomération de Mulhouse (appelé UHA par la suite).

[1] https://intempora.com/products/rtmaps/

3 Système de perception caméra – DeepLab V3+

Les images de la caméra RGB sont utilisée pour détecter l'espace navigable (caractérisé par la route). Un réseau de neurones DeepLabV3+ [3] permet la prédiction. Le modèle est entraîné suivant différentes configurations de datasets (KITTI, UHA, Cityscapes + les différentes combinaisons deux-à-deux + les trois datasets mélangés). Les images de test sont toutes issues du jeu de données de l'UHA.

Les différents modèles obtenus sont comparés par rapport à la vérité terrain, elle aussi issue des datasets. Les métriques de comparaison sont celles utilisées habituellement en vision par ordinateur : Accuracy, Precision, Recall et F1 Score [4]. Le Tableau (1) illustre les performances obtenues :

entraîné sur	**F1 Score**	**Accuracy**	**Precision**	**Recall**
KITTI	0.8485	0.8589	0.8613	0.8589
UHA	0.9335	0.9309	0.9485	0.9308
CityScapes	0.9426	0.9424	0.9455	0.9424
UHA+CityScapes	0.9667	0.9658	0.9699	0.9658
UHA+CityScapes+KITTI	0.9722	0.9712	0.9747	0.9712

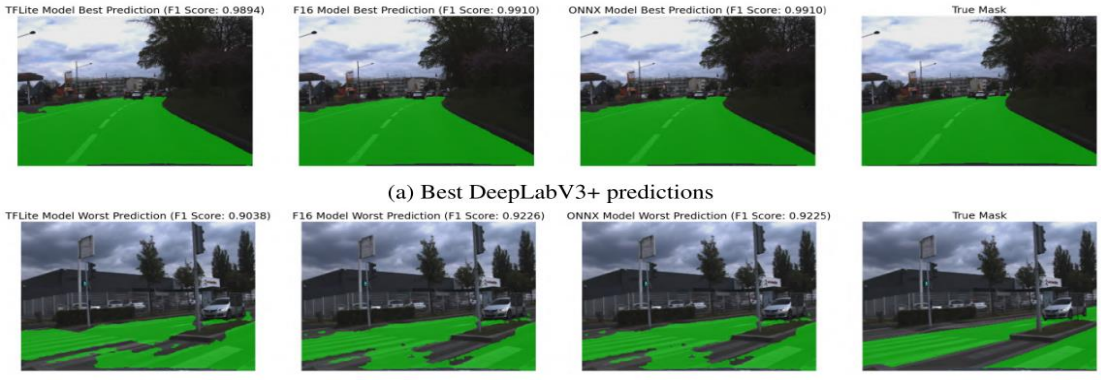
TAB. 1 – Performances des différents modèles obtenus

Les meilleures performances sont atteintes par le DeepLab entraîné sur le dataset composé des trois jeux de données ; cependant, certains cas de figure de la base KITTI nuisent à la prédiction. Le choix se porte donc sur UHA+CityScapes dans une configuration hiérarchique : le réseau est entraîné sur l'ensemble de CityScapes, puis un fine-tuning est réalisé avec la base de données de l'UHA.

4 Implémentation temps-réel de la perception d'un véhicule autonome

Le modèle DeepLabV3+ ainsi obtenu est implémenté dans le véhicule autonome ARTEMIPS sous le logiciel RTMaps en utilisant la librairie TensorFlow et OpenCV. Les premiers résultats qualitatifs sont très performants (0.9788 pour le F1 Score moyen) mais permettent un traitement en 16FPS, ce qui ne permet pas le temps-réel dans toutes les situations (en fonction de la vitesse du véhicule et de la nature de l'environnement proche).

Afin d'améliorer la rapidité de prédiction du réseau, différentes méthodes d'optimisation sont comparées : F16 (troncature des nombres flottantes 32bits => 16bits), TFLite (méthode de quantification 32bits => int8) et ONNX (Open Neural Network Exchange : conversion de framework). La Figure (1) montre les résultats de prédiction obtenus en embarquant les modèles DeepLabV3+ optimisés selon les trois méthodes.



(a) Best DeepLabV3+ predictions

(b) Worst DeepLabV3+ predictions

FIG. 1 – Prédictions du DeepLabV3+ optimisé selon TFLite ; F16 ; ONNX et la vérité-terrain associée

Les performances en terme de F1 Score sont assez identiques entre les modèles optimisés par F16 et ONNX, légèrement plus faibles avec TFLite (car on réduit la précision en réduisant la taille du modèle). On peut alors s'intéresser à la rapidité des prédictions en calculant les FPS des trois approches.

DeepLabV3+ optimisé avec	**FPS**	**Size (Kb)**	**Hardware**
sans optimisation	15.75	1998	GPU
TFLite	0.25	574	CPU
F16	17	806	GPU
ONNX	150	1574	GPU

TAB. 2 – Comparaison des méthodes d'optimisation pour augmenter les FPS

Si TFLite était prometteur concernant la rapidité de prédiction, ce n'est malheureusement pas le cas en pratique : la méthode étant dédiée aux systèmes embarqués, ceux-ci n'intègrent usuellement pas de GPU. TFLite force donc l'ensemble du système de perception à travailler sur la CPU, ce qui dégrade considérablement le temps de calcul. La conversion de modèles dans différents frameworks réalisé par ONNX montre donc des performances équivalentes pour un gain en FPS très important, avec 150FPS pour traiter les données. Cette rapidité de prédiction n'est bien sûr pas atteignable dans le véhicule d'essai, la cadence d'acquisition de la caméra MANTA ne permettant pas une perception à ce rythme.

5 Conclusions et perspectives

Ce résumé présente l'implémentation temps-réel de systèmes de perception pour véhicules autonomes basés sur des réseaux de neurones. A travers l'application de détection de route, deux points d'importance sont soulignés concernant le passage au temps-réel d'algorithmes d'IA type DeepLabV3+ : la composition du jeu de données considéré (que ce soit en quantité ou en qualité d'images) et la nécessité d'utiliser une méthode d'optimisation. Dans ce résumé, l'architecture proposée est la suivante : DeepLabV3+ entraîné sur CityScapes, fine-tuné sur le dataset propre à l'application dans l'agglomération de Mulhouse, optimisé grâce à la librairie ONNX et implémenté dans le véhicule autonome ARTEMIPS via le logiciel RTMaps. Les résultats expérimentaux montrent des performances de prédiction de la route entre 92% et 99% pour un temps de prédiction théorique de 150FPS.

Références

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013, doi: 10.1177/0278364913491297.

[2] M. Cordts *et al.*, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 3213–3223. doi: 10.1109/CVPR.2016.350.

[3] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," presented at the Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 801–818. Accessed: Mar. 14, 2025. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/html/Liang-Chieh_Chen_Encoder-Decoder_with_Atrous_ECCV_2018_paper.html

[4] D.-V. Giurgi, T. Josso-Laurain, M. Devanne, and J.-P. Lauffenburger, "Real-time road detection implementation of UNet architecture for autonomous driving," in *2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, Jun. 2022, pp. 1–5. doi: 10.1109/IVMSP54334.2022.9816237.