

# Détection de Voies à Faible Consommation avec SNN sur SPECK

Crescenzo Edoardo MAURIELLO<sup>1</sup> Hugo BULZOMI<sup>1</sup> Jean MARTINET<sup>2</sup> Yuta NAKANO<sup>1</sup>

<sup>1</sup>Imra Europe S.a.S., 220 Rue Albert Caquot, 06904 Sophia Antipolis, France

<sup>2</sup>Laboratoire d'Informatique, Signaux et Systèmes de Sophia-Antipolis (i3S) UMR 7271 UCA CNRS Les Algorithmes - bât. Euclide B 2000 route des Lucioles 06900 Sophia Antipolis

**Résumé** – La détection des voies de circulation est une tâche cruciale dans l'industrie automobile, constituant un élément fondamental et critique pour la sécurité des systèmes de conduite autonome. Au fil des ans, de nombreuses approches ont été proposées, la plupart reposant sur des réseaux neuronaux convolutionnels (CNN) traitant des images RVB. Bien que ces méthodes permettent d'obtenir une précision et une robustesse élevées, elles se heurtent à des exigences importantes en matière de calcul et d'énergie, ce qui les rend inadaptées à un fonctionnement en temps réel sur des systèmes embarqués à faible consommation d'énergie. Dans ce travail, nous explorons une approche alternative légère, frugale qui s'appuie sur le matériel neuromorphique et les réseaux convolutionnel neuronaux Spiking (SCNN) optimisé pour le déploiement sur la puce neuromorphique SPECK. Nous démontrons que la détection de voies peut être effectuée avec une consommation d'énergie de seulement 12mW soit deux ordres de grandeur de moins que les modèles d'apprentissage profond conventionnels. Au-delà des performances expérimentales, nous discutons également de l'intégration de notre approche dans l'écosystème neuromorphique actuel. Nous présentons les limitations rencontrées avec la carte SPECK, notamment en termes de contraintes de mémoire, de latence d'inférence et de support logiciel. Cette analyse met en lumière les défis et opportunités liés à l'usage de matériel neuromorphique pour des tâches de perception embarquée, et souligne son potentiel pour les véhicules autonomes de prochaine génération.

**Abstract** – Lane detection is a crucial task in the automotive industry, forming a fundamental and safety-critical component of autonomous driving systems. Over the years, numerous approaches have been proposed, with the majority relying on convolutional neural networks (CNNs) processing RGB images. While these methods achieve high accuracy and robustness, they come at the cost of significant computational and energy demands, making them unsuitable for real-time operation on power-constrained embedded systems. In this work, we explore a lightweight, energy-efficient alternative approach leveraging neuromorphic hardware and spiking convolutional neural networks (SCNNs) optimized for deployment on the SPECK neuromorphic chip. We demonstrate that lane detection can be performed with a power consumption of just 12mW—two orders of magnitude lower than conventional deep learning models. Beyond the experimental results, we also discuss how our approach fits within the broader neuromorphic computing landscape. We highlight the limitations we encountered with the SPECK chip, particularly in terms of memory constraints, inference latency, and software support. This analysis sheds light on both the challenges and opportunities of using neuromorphic hardware for embedded perception tasks, and emphasizes its potential for next-generation autonomous vehicles.

## 1 Introduction

La détection des voies est essentielle pour la sécurité des systèmes de conduite autonome. Les CNN, bien que performants, requièrent des architectures lourdes comme U-Net[1], inadaptées aux solutions embarquées. Les SNN, inspirés des neurones biologiques[4], offrent une alternative énergétiquement efficace, surtout sur du matériel neuromorphique comme SPECK[10]. Cet article propose une approche de détection de voies ultra-basse consommation basée sur un SNN entraîné sur un dataset d'images événementielles et testé sur SPECK via Sinabs[7]. Ceci est la continuation du travail de Mauriello et al. [5].

### 1.1 Approches pour la détection des voies

La détection de voies peut être formulée comme une tâche de segmentation sémantique, où chaque pixel est classé comme voie ou arrière-plan. La plupart des méthodes reposent sur des CNN, comme les FCN[3] ou U-Net[6], qui offrent une bonne précision mais sont coûteux en calcul et peu adaptés au matériel neuromorphique comme SPECK.

Les SNN, associés aux caméras événementielles, offrent

une alternative prometteuse pour la vision embarquée à faible consommation. Contrairement aux caméras classiques, les capteurs événementiels capturent des données de manière asynchrone avec une grande précision temporelle. Si les CNN ont prouvé leur efficacité pour cette tâche, l'utilisation des SNN reste encore peu explorée. LaneSNN[8], déployé sur Loihi[2], en est une première contribution : il combine un encodeur convolutif à pointes avec un décodeur dense, mais reste gourmand en calcul. Un décodeur spiking plus léger, couplé à un SoC économe, pourrait améliorer son efficacité.

#### 1.1.1 SPECK

La carte SPECK de Sinabs[10][7], basée sur un SoC neuromorphique, intègre 9 cœurs DYNAP-CNN, chacun avec une mémoire dédiée aux poids et aux neurones. Elle impose plusieurs contraintes : absence d'upsampling, skip connections déconseillées, et taille d'entrée limitée à 128×128 pixels.

Malgré ces limites, SPECK reste l'une des plateformes les plus économes en énergie, idéale pour des applications embarquées à très faible consommation. Nous présentons dans ce travail les éléments fonctionnels de notre implémentation, ainsi que les limitations rencontrées, en les confrontant à d'autres

architectures pour orienter de futurs choix matériels.

## 1.2 Modèle de neurone

Le modèle Leaky Integrate-and-Fire (LIF) est couramment utilisé en neurosciences computationnelles pour simuler l'activité neuronale de manière efficace. Il intègre les pointes entrantes dans un potentiel membranaire qui décroît selon une fonction de fuite :

$$\tau \dot{v} = -v(t) + R(I_{syn} + I_{bias}) \quad (1)$$

où  $\tau$  régit la dynamique temporelle,  $I_{syn} = Wx$  est la somme des entrées synaptiques pondérées,  $I_{bias}$  un biais constant, et  $v(t)$  représente la fuite.

Ce travail utilise une variante simplifiée, le modèle Integrate-and-Fire (IAF), où la fuite  $v(t)$  est constante, facilitant le calcul et améliorant l'efficacité du matériel.

Pour l'apprentissage, nous employons la rétropropagation spatio-temporelle (STBP)[9], une variante de la rétropropagation à travers le temps (BPTT) adaptée aux SNN. STBP apprend les dépendances temporelles en propageant le gradient à travers le temps et les couches, indispensable pour gérer les trames temporellement espacées en apprentissage et en inférence.

## 1.3 Évaluation

L'évaluation repose sur l'Intersection over Union (IoU), une métrique de segmentation sémantique :

$$IoU = \frac{\text{prédit} \cap \text{cible}}{\text{prédit} \cup \text{cible}} = \frac{VP}{FP + FN + VP} \quad (2)$$

où VP = Vrais Positifs, FP = Faux Positifs, FN = Faux Négatifs.

## 2 Dataset

Le dataset DET [3] est un jeu de données haute résolution basé sur les événements, conçu pour la détection des voies. Il contient des images d'événements et leurs étiquettes associées, avec une résolution de  $1280 \times 800$  pixels. Chaque image est générée en accumulant jusqu'à 255 événements par pixel sur 30 ms. Les étiquettes sont multi-classes, mais la segmentation d'instances ne sera pas exploitée dans cette étude.



FIGURE 1 : Exemple d'une image contenu dans DET.

### 2.1 Prétraitement

Nous avons appliqué un prétraitement aux images du jeu de données. En raison de l'angle de la caméra de l'événement, nous supprimons les 240px du haut, ce qui représente 30% de la taille d'origine, obtenant ainsi une forme de  $1280 \times 560$  et 160px du bas, ce qui représente 20% supplémentaires de la taille originale, ce qui donne  $400 \times 800$ ;

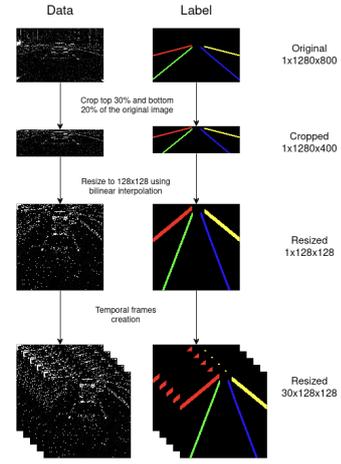


FIGURE 2 : Étapes de prétraitement des images et des label

Comme l'ensemble de données contient des images statiques, deux approches différentes pour construire les cadres ont été testées.

- Probabiliste, qui consiste à utiliser chaque valeur de pixel (entre 0 et 255) comme une probabilité et à générer des images sur cette base. Nous générons donc la probabilité  $\mathbb{P}$  du cadre en fonction de la valeur présente dans le pixel  $valeur(x, y)$  divisée par la valeur maximale parmi l'ensemble des données de 255 comme le formule 3. Ce masque est ensuite utilisé pour générer la valeur  $valeur'(x, y)$  en générant une valeur aléatoire entre 0 et 1  $rand(0, 1)$  et en fixant la valeur  $valeur'(x, y)$  à 1 si la valeur aléatoire est inférieure au masque pour ce pixel, sinon à 0 comme montré par la formule 4.
- Copie, clonage simple de l'image plusieurs fois.

$$\mathbb{P}(x, y) = valeur(x, y)/255 \quad (3)$$

$$valeur'(x, y) = \begin{cases} 1 & \text{if } rand(0, 1) \leq \mathbb{P}(x, y) \\ 0 & \text{if } rand(0, 1) > \mathbb{P}(x, y) \end{cases} \quad (4)$$

Les deux méthodes génèrent  $\#frames$  à partir de l'image monocanal, avec  $\#frames = 30$ , conformément aux 30 ms définies par le dataset et efficaces pour l'entraînement. Pour améliorer la généralisation, une augmentation de données a été appliquée : 50% des images sont retournées horizontalement, ce qui renforce la robustesse du modèle en détection de voies.

## 3 Réseau Neuronal

Nous utilisons un réseau convolutionnel dont l'architecture décrite dans la table 1. Le réseau peut être divisé en deux structures principales : l'encodeur et le décodeur, dont les couches sont suivies d'activations impulsives IAF (Integrate And Fire). Ces dernières imitent la dynamique temporelle des neurones biologiques. Pour améliorer la généralisation et la stabilité, des couches de batch normalization sont insérées entre les convolutions et une couche de dropout est incluse avant la

couche convolutive finale. Le nombre final de paramètres pouvant être appris dans le réseau est de 6365. Afin d’obtenir une prédiction, le tenseur de sortie est moyenné sur la dimension temporelle pour générer une matrice dont la résolution finale est de  $32 \times 32 = 1024$  pixels (même taille que nos labels). Cette structure permet au réseau d’extraire efficacement les marquages de voies à partir de données événementielles tout en respectant les contraintes de notre matériel neuromorphique.

TABLE 1 : table des couches convolutives. Couche; Forme (Input, Output); Filtres (Input, Output); Surface de traitement (K), Pas (S), Marge (P); Paramètres.

Co	Fo I, O	Fi. I, O	K,S,P	Par
<i>E1</i>	128, 64	1, 4	3,2,1	40
<i>E2</i>	64, 64	4, 4	3,1,1	148
<i>E3</i>	64, 32	4, 8	3,2,1	296
<i>E4</i>	32, 32	8, 16	3,1,1	1178
<i>E5</i>	32, 32	16, 32	3,1,1	4640
<i>D<sub>FCNN</sub></i>	32, 32	32, 1	1,1,0	33

## 4 Entraînement du réseau

Le réseau a été entraîné avec une carte graphique Nvidia GeForce RTX 4070, Python, Pytorch, Sinabs, Samna.

Les hyperparamètres ont été optimisés pour la performance. Un batch size de 32 a été choisi pour un entraînement efficace. Chaque séquence a été traitée sur 30 pas de temps pour capturer la dynamique temporelle. Le taux d’apprentissage était de  $10^{-3}$ , assurant un bon équilibre entre stabilité et vitesse de convergence.

Le modèle, basé sur 2 classes (voie / non-voie), a été entraîné avec la fonction de perte MSE et l’optimiseur Adam, sur 200 époques.

Le dataset a été divisé en 80% pour l’entraînement, 10% pour la validation et 10% pour le test, garantissant une évaluation sur des données inédites.

## 5 Résultats

Les résultats, résumés dans la table 2, montrent que notre modèle, FCNN\_NB est moins complexe et plus économe en énergie par rapport aux modèles neuromorphiques de l’état de l’art.

TABLE 2 : Comparaison avec le SOTA

Classificateur	Puiss.	Params	Lat	IoU
CNNLaneSNN	1W	1.39M	7ms	0.6, 0.3
C600 LaneSNN	1W	1.2M	7ms	0.6, 0.6
<b>FCNN_NB</b>	12mW	6k	19ms	0.3, 0.2

L’inférence sur 30 frames a affiché une latence de 19,26 ms, confirmant l’efficacité du matériel neuromorphique pour des applications à faible latence. Cependant, l’IoU a chuté à 0,2, suggérant un compromis entre contraintes matérielles et

précision, potentiellement dû à la quantification, aux limites de précision ou aux ressources réduites de SPECK.

Côté consommation, le réseau a atteint 12,1 mW, répondant aux exigences des systèmes embarqués. Ce résultat souligne le potentiel de SPECK pour la détection de voies en temps réel dans des environnements contraints en énergie. Des optimisations futures pourraient améliorer la précision tout en conservant cette efficacité énergétique, notamment via des ajustements d’architecture ou de nouveaux schémas d’encodage.

## 6 Mise en Perspective

Au-delà des résultats expérimentaux présentés, il nous paraît intéressant de discuter de la place de ce travail dans la sphère neuromorphique plus globale. Ce projet illustre une évolution significative de l’écosystème neuromorphique : il est désormais possible, avec une équipe restreinte et des moyens limités, de développer un prototype matériel complet, fonctionnel, et énergétiquement efficace pour une tâche de vision embarquée. Ce constat marque une rupture importante avec la situation d’il y a seulement quelques années, où la mise en œuvre d’un pipeline neuromorphique complet nécessitait souvent des ressources de recherche considérables, tant matérielles qu’humaines.

Grâce à des plateformes comme SPECK, le développement d’architectures spiking pour des applications concrètes devient accessible et réaliste, tant en termes de temps, de coût que de maintenance logicielle. Sa très faible consommation énergétique (de l’ordre du milliwatt) et sa compacité matérielle ouvrent la voie à des usages embarqués dans des environnements contraints, comme l’assistance à la conduite, la robotique mobile ou l’IoT événementiel. Le fait que ce travail s’appuie sur une plateforme matériellement opérationnelle, et non sur une simple simulation, marque une étape importante : la neuromorphie franchit le cap de la recherche exploratoire pour répondre à des besoins industriels. Dans ce contexte, SPECK s’est révélé un outil doublement formateur. Ses limitations (profondeur réduite, absence de multiplications inter-couches, pas de skip-connections, contraintes mémoire) exigent un design de réseau sobre et astucieux, tout en éclairant les compromis entre expressivité algorithmique et efficacité énergétique — un enjeu clé du calcul neuromorphique.

En comparaison avec des plateformes plus puissantes comme Loihi 2, SPECK n’offre ni la complexité topologique ni la flexibilité synaptique de certaines de ses aînées. Cependant, son accessibilité, sa simplicité de déploiement, et son efficacité énergétique font de cette plateforme un excellent tremplin pour le prototypage rapide et le transfert vers des solutions embarquées réelles.

### 6.1 Défis Actuels et Futur pour le Neuromorphique

Malgré ses promesses technologiques — efficacité énergétique, traitement asynchrone, parallélisme massif, compatibilité naturelle avec les signaux événementiels — l’intelligence artificielle neuromorphique peine encore à s’imposer dans des applications commerciales à grande échelle. Il est intéressant de rappeler que le concept même de calcul spiking remonte

aux travaux de Wolfgang Maass dans les années 1990, qui ont formalisé le cadre computationnel des réseaux de neurones impulsifs (SNNs) comme une alternative potentiellement plus puissante et plus biologique que les réseaux classiques. Pourtant, plus de deux décennies plus tard, force est de constater que le neuromorphique reste largement cantonné au monde académique, malgré un engouement constant dans les cercles de recherche.

Une des raisons majeures de cette stagnation est probablement l'absence d'une application phare, d'un cas d'usage commercial fort — une killer app — capable de justifier un basculement industriel vers ce paradigme computationnel. Contrairement au deep learning, qui a trouvé en la vision artificielle, la reconnaissance vocale et la traduction automatique des catalyseurs immédiats pour sa démocratisation, les réseaux SNNs n'ont pas encore rencontré leur équivalent transformateur.

Dans ce contexte, des plateformes comme SPECK jouent un rôle potentiellement déterminant par leur simplicité, leur faible coût, leur faible consommation énergétique, et leur capacité à exécuter des SNNs en temps réel sur un support matériel concret. SPECK ne prétend pas offrir la même puissance ou la même complexité que ces plateformes de recherche, mais elle répond à une autre exigence tout aussi cruciale : permettre à une communauté plus large — ingénieurs, industriels, développeurs embarqués — de construire, tester, et déployer rapidement des solutions neuromorphiques réalistes.

Dans cette optique, ce travail contribue à poser les bases de ce futur : en démontrant qu'il est possible de concevoir et de déployer une solution SNN complète pour un problème concret — la détection de lignes de trafic à partir d'un flux événementiel — avec des ressources limitées, il participe à réduire la distance entre le potentiel théorique du neuromorphique et sa réalité industrielle. Dans cette continuité, et au regard des limitations rencontrées lors de l'implémentation de notre prototype sur la plateforme SPECK, nous proposons à titre prospectif et subjectif une liste non exhaustive de fonctionnalités qui, selon nous, pourraient rapprocher l'écosystème neuromorphique de l'émergence d'une véritable « killer app ». Il ne s'agit pas d'une critique technique de SPECK, mais d'une projection fondée sur notre expérience pratique du déploiement.

- **Support des skip-connections.** L'impossibilité d'exprimer des connexions résiduelles limite fortement l'utilisation de topologies modernes, pourtant essentielles à la stabilité et à la profondeur des réseaux. La prise en charge, même partielle, de telles structures pourrait ouvrir la voie à des architectures SNN plus robustes et performantes.
- **Support partiel de la self-attention.** La possibilité d'implémenter des blocs attentionnels, même de manière approximative ou échantillonnée, serait une avancée majeure vers l'adaptation de SNNs à des tâches complexes de traitement séquentiel, de perception multimodale ou de prise de décision contextuelle.
- **Maintien du traitement asynchrone.** Contrairement à certaines plateformes comme Akida, qui imposent un traitement synchronisé de l'information par le réseau, nous pensons qu'il est crucial de conserver la *philosophie événementielle et asynchrone* du calcul neuromorphique, qui constitue l'un de ses avantages fonde-

mentaux — tant en termes de pertinence biologique que d'efficacité énergétique.

Ces axes d'amélioration reflètent notre conviction qu'il est désormais temps de construire des plateformes neuromorphiques à la fois expressives, compactes et orientées produit, qui conservent l'essence événementielle du paradigme tout en intégrant certains éléments-clés du deep learning moderne. Une telle convergence pourrait bien représenter le socle technologique d'une future application commerciale structurante.

## 7 Conclusion

Avec notre approche, nous avons obtenu un réseau à très faible complexité et à consommation énergétique ultra-réduite, capable de fonctionner avec dix fois moins d'énergie que Loihi. Cette efficacité se fait toutefois au prix d'une latence environ triplée et d'un IoU réduit de moitié par rapport à l'implémentation de Viale et al. [8].

## Références

- [1] Sun-Woo et al. BAEK : Real-time lane detection based on deep learning. *Journal of electrical engineering & technology*, 17(1):655–664, 2022.
- [2] Mike et al. DAVIES : Loihi : A neuromorphic manycore processor with on-chip learning. *IEEE MICRO*, 38(1):82–99, 2018.
- [3] Jonathan LONG, Evan SHELHAMER et Trevor DARRELL : Fully convolutional networks for semantic segmentation, 2015.
- [4] Wolfgang MAASS : Networks of spiking neurons : The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.
- [5] Crescenzo Edoardo MAURIELLO, Hugo BULZOMI, Jean MARTINET et Yuta NAKANO : End-to-end neuromorphic lane detection. In *Proceedings of 2025 10th International Workshop on Advances in Sensors and Interfaces (IWASI)*, volume 10, pages 161–165, 2025.
- [6] Olaf RONNEBERGER, Philipp FISCHER et Thomas BROX : U-net : Convolutional networks for biomedical image segmentation, 2015.
- [7] SINABS : Sinabs devkit documentation.
- [8] Alberto VIALE, Alberto MARCHISIO, Maurizio MARTINA, Guido MASERA et Muhammad SHAFIQUE : Lanensns : Spiking neural networks for lane detection on the loihi neuromorphic processor. 2022.
- [9] Yujie WU, Lei DENG, Guoqi LI, Jun ZHU et Luping SHI : Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, 12, mai 2018.
- [10] Man et al. YAO : Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip. *Nature communications*, 15(1):4464–4464, 2024.