État de l'Art : Détection de Drones

Ahmed Loughzali Mohamed Abdellatifi Amar Lassal Ayoub Tarek

***Abstract*—La prolifération rapide des véhicules aériens sans pilote (UAV) pose des défis de sécurité majeurs, rendant critique le développement de systèmes de surveillance réactifs. Cet article dresse un état de l'art des méthodes de détection de drones, en soulignant d'abord les limites intrinsèques des capteurs conventionnels (Radar, RF, vision RGB) face à des cibles petites et rapides : latence élevée, flou de mouvement et consommation excessive. Nous explorons ensuite la rupture technologique offerte par les capteurs de vision événementielle (DVS), qui, grâce à leur fonctionnement asynchrone et leur haute gamme dynamique, permettent de capturer la cinématique des drones avec une précision temporelle de l'ordre de la microseconde. Enfin, l'étude analyse le potentiel des architectures neuromorphiques et des réseaux de neurones impulsionnels (SNN) pour le traitement de ces flux, démontrant une efficacité énergétique supérieure indispensable aux systèmes de défense embarqués et autonomes.**

***Index Terms*—Détection de drones, Vision par ordinateur, Vision événementielle, SNN, Neuromorphique.**

I. Introduction

L'essor fulgurant des véhicules aériens sans pilote (UAV), ou drones, a radicalement transformé le paysage technologique moderne. Initialement réservés à un usage militaire, ces aéronefs sont désormais omniprésents dans les secteurs civils et récréatifs. Cependant, cette prolifération soulève des enjeux critiques en matière de sécurité : la capacité des drones à transporter des charges utiles ou à effectuer de la surveillance illicite nécessite le développement de systèmes de détection capables d'identifier des objets aux signatures physiques et visuelles complexes, telles que la rotation rapide des hélices ou de grandes vitesses angulaires. Sur le plan éthique, la détection de drones se situe à l'intersection entre la protection de la vie privée et la sécurité publique. Si la surveillance est nécessaire pour protéger des sites sensibles, l'usage de capteurs à haute résolution dans l'espace urbain pose la question de l'anonymat des citoyens. À cet égard, la transition vers des technologies neuromorphiques, qui traitent uniquement les variations de contraste plutôt que des images complètes, pourrait offrir une voie plus respectueuse de la confidentialité des données. D'un point de vue technique, l'enjeu majeur réside dans l'efficacité énergétique et la latence. Les méthodes actuelles basées sur la vision classique (RGB) et les réseaux de neurones artificiels (ANN) sont confrontées à un verrou technologique : elles nécessitent des processeurs graphiques (GPU) extrêmement gourmands en énergie, ce qui limite considérablement l'autonomie des systèmes de défense embarqués. De plus, les limites intrinsèques des caméras conventionnelles, telles que le flou de mouvement (motion blur) et une plage dynamique limitée, rendent la détection précaire dans des conditions environnementales difficiles. Le présent état de l'art explore les solutions pour pallier ces limites.

II. Capteurs Électromagnétiques

A. Radar et Micro-Doppler

Les approches actives se définissent par l'émission d'un signal vers l'environnement pour en analyser les réflexions ou par l'interception de communications sortantes. Le radar constitue historiquement la technologie prédominante dans ce domaine, mais son efficacité sur de petites cibles a nécessité des évolutions majeures. À ce titre, les travaux [1] ont démontré l'apport critique du Deep Learning pour isoler les signatures micro-Doppler générées par la rotation des pales de drones. Cette technique permet de détecter des drones ayant une très faible surface équivalente radar (SER) et de les distinguer avec précision d'autres objets mobiles, tels que les oiseaux, en se basant sur la modulation spécifique du signal réfléchi par les hélices.

B. Radio-Fréquence (RF)

L'identification peut s'appuyer sur l'analyse des signaux de commande via l'approche de [2] portant sur le RF Fingerprinting. Cette méthode repose sur la modélisation des imperfections matérielles microscopiques et intrinsèques des émetteurs à bande étroite, permettant d'extraire une "empreinte digitale" électronique unique pour chaque appareil.

Malgré ces performances, les méthodes actives se heurtent à des contraintes structurelles et environnementales importantes. L'efficacité des systèmes basés sur l'interception des fréquences radio, tels que modélisés par [2], s'annule totalement lorsque le drone opère en silence radio via une navigation autonome par points de passage (waypoints). Par ailleurs, la généralisation de matériaux composites comme le carbone ou certains polymères dans la conception des drones civils renforce leur furtivité, réduisant drastiquement la portée effective des radars conventionnels [1]. En milieu urbain, ces technologies font face à des défis supplémentaires liés à la multiplication des obstacles physiques qui génèrent des échos parasites et des zones d'ombre. Ces vulnérabilités, couplées au coût élevé des infrastructures actives, justifient l'intérêt croissant de la recherche pour les méthodes passives, plus discrètes et souvent mieux adaptées à une surveillance urbaine dense.

III. Capteurs Visuels Traditionnels et LiDAR

A. Vision RGB

La détection de drones par vision classique repose sur l'utilisation de caméras RGB conventionnelles. Ces capteurs

fonctionnent par intégration de la lumière sur toute la surface de la matrice de pixels durant un temps d'exposition fixe, générant une succession d'images complètes (frames) à une fréquence régulière, généralement 30 images par seconde. Bien que cette technologie bénéficie d'une maturité industrielle certaine, elle impose des contraintes physiques majeures pour la surveillance aérienne. Le mécanisme de capture par trames induit systématiquement un flou de mouvement (motion blur) dès lors que le drone ou la caméra effectue des déplacements rapides. De plus, la plage dynamique limitée de ces capteurs rend la détection particulièrement précaire en situation de contre-jour ou de changements brusques de luminosité. Enfin, le flux vidéo génère une redondance massive de données, puisque chaque pixel est renvoyé à chaque frame même si l'information visuelle est identique, surchargeant inutilement les unités de calcul. Pour traiter ces flux, l'état de l'art s'est massivement tourné vers les réseaux de neurones artificiels (ANN), et plus particulièrement vers les architectures de détection d'objets en temps réel. Le papier [5] illustre bien cette tendance en proposant une classification basée sur un réseau de caméras HD couplé à l'algorithme YOLOv2, permettant de distinguer les drones d'autres aéronefs. Cependant, l'analyse des résultats révèle une faiblesse critique : si le système est performant pour les appareils de grande taille, l'exactitude pour les drones reste limitée en raison de leur petite signature visuelle et de la complexité de l'arrière-plan urbain. En comparaison, les recherches plus récentes synthétisées dans le survey de [9] soulignent que si les modèles de type YOLOv8 [6] ou Faster R-CNN permettent d'obtenir des taux de vrais positifs plus élevés, ces performances s'effondrent dès que les conditions météo deviennent défavorables ou que des occlusions surviennent. Des travaux comme ceux de [7] explorent alors l'utilisation de plusieurs caméras pour améliorer le suivi 3D, mais la problématique de la petite taille de cible demeure centrale. Le papier HierLight-YOLO [8] tente de répondre à ce défi en proposant une fusion hiérarchique de caractéristiques spécialement optimisée pour les objets de très faible résolution capturés par photographie aérienne, parvenant ainsi à réduire la complexité computationnelle. Dans cette lignée, les travaux [4] renforcent la robustesse de la détection en couplant l'architecture YOLOv7 à des mécanismes d'attention et à une méthode hybride intégrant un suivi basé sur des règles de corrélation croisée, permettant de récupérer une partie des détections perdues par le seul réseau de neurones. Toutefois, cette surenchère algorithmique vient confirmer les limites structurelles de la vision par ordinateur traditionnelle. L'analyse comparative de ces différents travaux met en lumière un paradoxe persistant : l'utilisation de datasets massifs et d'architectures profondes permet une détection de plus en plus fine, mais se heurte systématiquement à une latence de traitement élevée et à une consommation énergétique excessive. Le besoin de processeurs graphiques (GPU) puissants pour faire tourner ces modèles en temps réel draine rapidement les batteries des systèmes embarqués, constituant un verrou technologique qui pousse aujourd'hui la recherche vers des alternatives neuromorphiques plus sobres et réactives.

B. LiDAR

Pour les scénarios exigeant une précision spatiale centimétrique, le recours à la télémétrie LiDAR s'avère particulièrement efficace. Les recherches [3] prouvent notamment qu'une détection et une localisation 3D fiables demeurent possibles même à partir de mesures laser éparses, offrant ainsi une alternative robuste aux interférences électromagnétiques, bien que limitée par une portée souvent plus courte que celle des radars.

IV. Capteur Événementiel et SNN

A. La Rupture DVS : Du Synchrone à l'Asynchrone

La vision classique repose sur une capture globale et synchronisée d'images complètes à intervalles réguliers dictés par une horloge externe. À l'inverse, le capteur DVS (Dynamic Vision Sensor) opère une rupture bio-inspirée de la rétine biologique : il ne produit pas de trames (frames), mais un flux asynchrone d'événements déclenchés par les dynamiques de la scène.

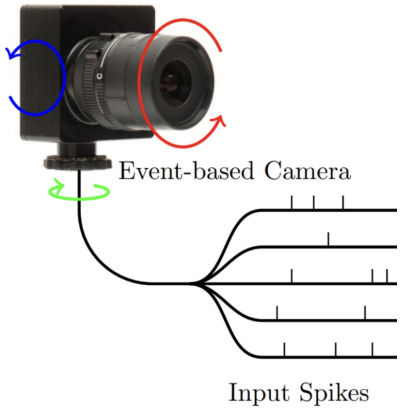


Fig. 1. Caméra événementielle

B. Physique et fonctionnement du pixel

Chaque pixel du capteur DVS est indépendant et surveille en continu les variations de l'intensité lumineuse logarithmique $L = \log(I)$. Un événement discret est généré sous la forme d'un tuple $e_k = (x, y, t, p)$ dès que la différence de luminosité au pixel (x, y) dépasse un seuil de contraste prédéfini C par rapport au dernier événement émis :

$$|\Delta L| = |L(t) - L(t_{\text{last}})| \geq C$$

Les composantes de l'événement sont définies comme suit :

- t : Horodatage précis (*timestamp*) avec une résolution de l'ordre de la microseconde.
- p : Polarité binaire ($p \in \{-1, +1\}$) indiquant si la luminosité a augmenté (ON) ou diminué (OFF).

Ce mécanisme asynchrone élimine la redondance massive des données : si la scène est statique, aucun événement n'est

transmis, ce qui réduit drastiquement la bande passante et la consommation énergétique.

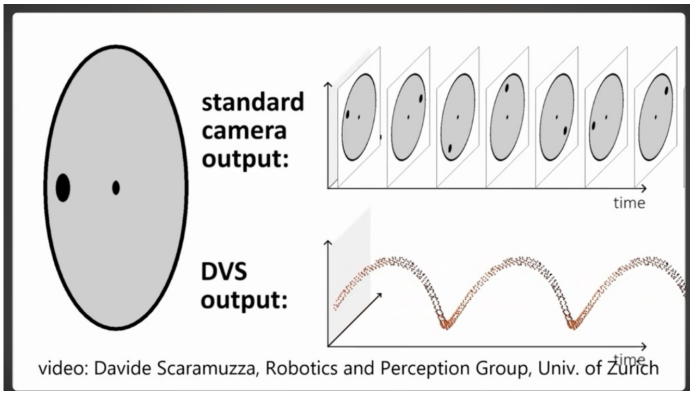


Fig. 2. Principe de fonctionnement d'un capteur de vision événementielle (DVS) illustrant le traitement asynchrone des variations de luminosité.

C. Méthode fréquentielle

L'approche la plus "pure" exploite la haute résolution temporelle du capteur sans recourir à la reconstruction d'image. Il a été démontré dans [10] qu'il est possible de détecter un drone en analysant uniquement la périodicité des événements générés par ses rotors.

- Méthode : Histogrammes de périodes inter-événements combinés à une FFT légère.
- Performance : Détection à 100 Hz avec une consommation énergétique ultra-faible (5.1 W), idéale pour les nœuds de surveillance embarqués.

Cette approche se heurte cependant à des limites physiques. La "détectabilité" dépend du nombre de pixels balayés par la pale. Au-delà d'une distance critique, le drone devient une source ponctuelle indiscernable du bruit. De plus, le "paradoxe de la stationnarité" rend un drone en vol stationnaire parfait invisible pour un capteur DVS s'il n'y a pas de mouvement relatif caméra-cible.

D. Approches Hybrides (DVS + ANN)

La première étape logique pour les chercheurs a été d'utiliser les détecteurs d'objets existants (YOLO, R-CNN, SSD), très performants sur les images classiques, et de leur fournir des données événementielles. C'est l'approche séquentielle : le flux d'événements est transformé pour s'adapter au réseau de neurone.

1) Le Problème de la Représentation: Les réseaux de neurones convolutionnels (CNN) s'attendent à recevoir des matrices denses (images). Or, le DVS envoie une pluie de points asynchrones. Il faut donc une étape de conversion :

- **Event Frames (Images d'événements) :** On accumule les événements pendant un temps Δt (ex: 30ms). On obtient une image en niveaux de gris où la valeur du pixel représente la quantité de mouvement.
  - **Avantage :** Compatible immédiatement avec tous les algorithmes de vision standard.
  - **Inconvénient :** On réintroduit de la latence (le temps d'accumulation) et du flou cinétique, perdant partiellement l'avantage du capteur.
- **Voxel Grids (Grilles Spatio-temporelles) :** C'est l'état de l'art actuel pour la représentation. Au lieu d'écraser le temps en une image 2D, on crée un cube 3D (x, y, t) qui conserve la chronologie précise du mouvement. [12]

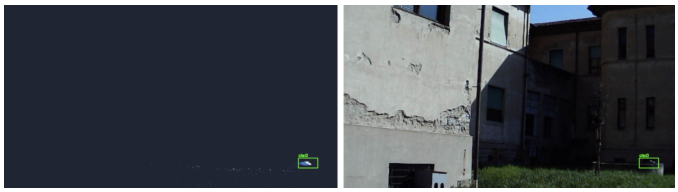


Fig. 3. Event frame VS Frame RGB, dataset FRED

2) L'Application d'un modèle convolutif: Une approche courante consiste à convertir le flux DVS en trames d'événements pour l'entraînement de modèles de détection standard tels que YOLO. Le modèle effectue une Détection d'Objet (il prédit une boîte englobante x, y, w, h) et une classification (classe "Drone"). L'Entraînement est un apprentissage supervisé classique. Cependant, comme il existe peu de datasets DVS annotés, on utilise souvent le Transfer Learning : on prend un YOLO pré-entraîné sur des millions d'images RGB (COCO dataset) et on le "raffine" (Fine-Tuning) sur des images d'événements en rajoutant la classe drone à la dernière couche .

3) Fusion par Attention (Le Module TMAF): L'architecture proposée [11] représente une avancée majeure. Au lieu de simplement superposer les images, ils ont créé un mécanisme d'attention.

- **Architecture** : Deux branches de réseau distinctes traitent les données en parallèle.
  - Branche RGB : Analyse la texture et l'apparence.
  - Branche DVS : Analyse la dynamique via des Voxel Grids.
- **Innovation** : Le module TMAF (Temporal Multi-Scale Attention Fusion) agit comme un "chef d'orchestre". Il analyse le flux DVS pour repérer les zones de mouvement rapide et "force" la branche RGB à se concentrer sur ces zones.
- **Entraînement** : Nécessite des datasets complexes comme FRED [21] , où les images RGB et les événements sont parfaitement synchronisés et annotés ainsi qu'une grande capacité de calcul pour les modules d'attention.

E. Réseaux de Neurones Impulsionnels (SNN)

Les Réseaux de Neurones Impulsionnels (SNN) se distinguant des modèles classiques par leur nature bio-inspirée et leur fonctionnement "event-driven". Alors que les réseaux de neurones artificiels (ANN) s'appuient sur des activations continues et statiques, les SNN utilisent des unités de calcul dont l'état interne évolue dynamiquement en fonction des

entrées reçues. Cette architecture repose sur le principe de l'intégration et de l'émission : chaque neurone accumule des informations jusqu'à atteindre un seuil de potentiel spécifique. Ce n'est qu'à cet instant qu'une impulsion brève, ou spike, est générée et transmise aux neurones suivants. Ce système évite d'envoyer des informations en continu pour ne garder que les changements importants, ce qui réduit énormément la consommation d'énergie. En regroupant la mémoire et le calcul au même endroit, ces réseaux imitent le cerveau humain et permettent d'analyser les données en intégrant pleinement l'aspect temporel.

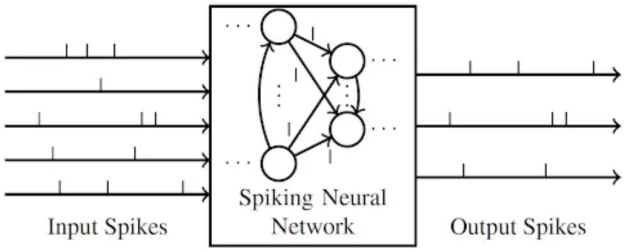


Fig. 4. Fonctionnement d'un SNN

Cette exploitation de la dimension temporelle est déterminante pour la détection de drones, car elle permet de capturer des dynamiques ultra-rapides, comme la rotation des hélices, avec une latence de l'ordre de la microseconde. Ce mécanisme allège considérablement la charge computationnelle et dépasse les méthodes classiques limitées par leur fréquence d'acquisition. Cette approche garantit ainsi une surveillance en temps réel très efficace. Elle permet de suivre des cibles minuscules dans des environnements complexes tout en maintenant une consommation énergétique minimale. C'est un atout indispensable pour les systèmes embarqués fonctionnant sur batterie.

L'examen de l'état de l'art révèle un décalage important entre les promesses théoriques des réseaux impulsionnels (SNN) et leur mise en pratique réelle pour la surveillance aérienne. Bien que le concept soit prometteur, les publications documentant l'exécution de SNN "on-chip" (directement sur puce neuromorphique) pour la détection de drones restent extrêmement rares. La majorité des travaux actuels se limitent à des simulations logicielles ou à des tâches de classification simplifiées. Ils n'abordent pas encore la complexité d'une détection en temps réel dans des conditions opérationnelles exigeantes. Néanmoins, de nombreux travaux soulignent la supériorité des SNN face aux ANN, à l'instar de [14] qui démontrent l'efficacité radicale du système SNN sur puce Speck, dont la consommation lors de l'inference de seulement 10 mW (contre 2,64 W pour un ANN sur GPU) permet une autonomie sur batterie allant de 6 mois à plus d'un an. Toutefois, ce papier se concentre davantage sur une tâche de classification que de détection pure.

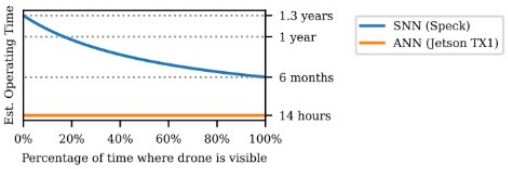


Fig. 5. Impact du traitement asynchrone sur la consommation énergétique : l'approche SNN offre une autonomie allant de 6 mois à 1,3 an contre une autonomie fixe de 14h pour l'approche ANN

Compte tenu de la difficulté d'entraîner des réseaux impulsionnels complexes à partir de zéro,trois approches émergent pour la détection de cibles :

1) La Conversion ANN-to-SNN: Cette technique consiste à entraîner un réseau de neurones artificiels classique (souvent un détecteur de type YOLO) sur des données RGB ou des frames d'événements, puis à convertir ses poids et ses fonctions d'activation en neurones impulsionnels. [13]

- **Avantage :** Elle permet de bénéficier de la précision des architectures de pointe déjà validées sur des jeux de données de référence, tels que le *Kaggle Drone Dataset* ou *Anti-UAV*.
- **Limite :** La conversion introduit souvent une perte de précision et nécessite une phase de calibration rigoureuse pour que le modèle reste efficace une fois déployé sur du matériel neuromorphique spécialisé comme *Akida* ou *Loihi 2*. Pour illustrer ce mécanisme, la figure ci-dessous détaille le passage d'une structure conventionnelle à un traitement par spikes.

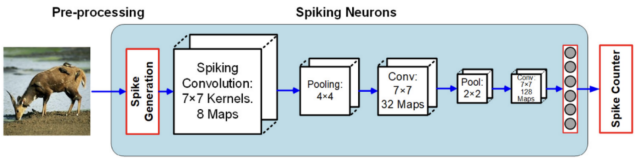


Fig. 6. Schéma fonctionnel ANN2SNN pour de la convolution

Le pipeline de traitement se décompose alors en trois phases clés :

- **Génération de spikes :** L'image est convertie en flux d'impulsions où la fréquence des pics code l'intensité des pixels.
- **Traitement asynchrone :** Les données traversent des couches de convolution et de pooling impulsionnelles utilisant des poids pré-entraînés.
- **Décision (Spike Counter) :** Le neurone de sortie ayant la plus haute fréquence d'activité identifie la cible, garantissant une détection rapide à très faible coût énergétique.

2) MG-SNN: C'est une architecture hybride "Combo" qui marie un SNN (Spiking Neural Network) et un ANN (Artificial Neural Network) classique (type YOLO) [15].

- **L'Entrée :** Le système prend en entrée des images vidéo, mais les traite via un encodage temporel d'impulsions (*spike temporal encoding*) pour simuler le flux nerveux.
- **La Branche SNN (M-Pathway) :** Un réseau de neurones à impulsions traite ce flux pour extraire uniquement la saillance du mouvement (*Motion Saliency*). Contrairement à un CNN qui effectue des calculs matriciels lourds, le SNN fonctionne par pics d'énergie de manière asynchrone et très rapide.
- **L'Innovation d'Entraînement (DT-MSTS) :** L'entraînement des SNN est complexe car ils ne sont pas dérivables, rendant la rétropropagation (*backpropagation*) classique inapplicable. Ils ont conçu la méthode DT-MSTS (*Dynamic Threshold Multi-frame Spike Time Sequence*) pour entraîner efficacement le réseau.
- **La Fusion :** Les "cartes de mouvement" extraites par le SNN sont injectées dans le réseau classique qui assure la détection finale (*Bounding Box*).

3) Les Spiking Transformers: Inspirés des Transformers utilisés en traitement du langage et en vision (ViT), les Spiking Transformers [34] [35] sont une innovation très récente.

- **Fonctionnement :** Ils remplacent les mécanismes d'attention classiques (très coûteux en calcul) par des mécanismes d'attention impulsionnels (*Spiking Self-Attention*).
- **Application :** Bien qu'initialement conçus pour la classification d'images ou la reconnaissance d'actions, leur capacité à capturer des dépendances temporelles à long terme les rend extrêmement prometteurs pour distinguer un drone d'un oiseau dans des scénarios complexes.
- **Potentiel :** Ils permettraient de mieux traiter les séquences du dataset FRED ou VMD, où la trajectoire et le contexte temporel sont essentiels pour la détection de petites cibles.

V. Hardware neuromorphique

L'implémentation efficace des SNN repose sur l'utilisation d'architectures matérielles non-von Neumann, spécifiquement conçues pour traiter l'information de manière asynchrone et parallélisée. Contrairement aux processeurs classiques, ce hardware dédié colocalise la mémoire et le calcul pour traiter les flux de spikes en temps réel avec une consommation énergétique minimale.

Pour exploiter ces propriétés, plusieurs solutions de hardware dédié ont été développées afin de supporter nativement le calcul par spikes on peut citer quatres hardwares principaux :

- **Loihi 2 (Intel)** : Processeur de recherche ultra-flexible conçu pour l'implémentation de modèles neuronaux complexes et l'apprentissage asynchrone.
- **Akida (Brainchip)** : Processeur commercial optimisé pour l'inférence *on-chip*, offrant une solution basse consommation adaptée aux systèmes embarqués.
- **SynSense (DYNAP-CNN)** : Matériel spécialisé dans le traitement ultra-basse consommation des flux issus de capteurs événementiels (DVS).
- **FPGA** : Utilisé au niveau le plus bas pour concevoir des architectures de calcul sur mesure . [36]

VI. Datasets, comparaison des méthodes et analyse critique transversale

A. Panorama des datasets pour la détection de drones

Les performances des méthodes de détection de drones dépendent fortement des jeux de données utilisés pour l'entraînement et l'évaluation. Dans la littérature, ainsi que dans notre propre état de l'art, on observe une grande hétérogénéité des datasets, tant du point de vue des modalités capteurs (RGB, événementiel, hybride) que des scénarios considérés (fond ciel, urbain, maritime, jour ou nuit). Cette diversité complique la comparaison directe des méthodes et rend nécessaire une analyse critique centrée sur les données elles-mêmes.

1) Datasets RGB publics de type Kaggle: Plusieurs jeux de données RGB accessibles publiquement sont disponibles sur des plateformes comme Kaggle et sont couramment utilisés pour la détection de drones à l'aide de réseaux de neurones convolutifs. Parmi ceux que nous avons identifiés figurent notamment le *Drone Dataset (UAV)* [16], le *YOLO Drone Detection Dataset* [17] ainsi que le dataset *Drone vs Bird* [18].

Ces datasets sont généralement constitués d'images ou de courtes séquences vidéo RGB annotées à l'aide de boîtes englobantes. Ils sont adaptés à l'entraînement initial de détecteurs de type YOLO et permettent de valider rapidement des pipelines de détection basés sur des réseaux artificiels classiques.

Cependant, ces jeux de données présentent plusieurs limitations importantes dans un contexte de surveillance réaliste. Les arrière-plans sont souvent simples, avec une prédominance de ciels dégagés et peu de variations météorologiques. La diversité des scénarios est limitée et la présence de distracteurs réalistes, tels que les oiseaux ou les avions lointains, est parfois insuffisante. De plus, la distribution précise de la taille des cibles n'est généralement pas documentée, ce qui rend difficile l'évaluation rigoureuse des performances sur des drones de très petite taille.

Afin d'illustrer ces différences, la Figure 7 présente des exemples représentatifs d'images issues de datasets RGB grand public et de datasets spécialisés dédiés à la détection de drones en conditions plus complexes.

2) Datasets Anti-UAV et jeux dérivés: Les datasets Anti-UAV (notamment Anti-UAV-2021 et Anti-UAV-2023) [22] ont été conçus spécifiquement pour la détection de drones de petite taille évoluant dans des arrière-plans complexes. Ils incluent des séquences vidéo présentant des scènes variées telles que des environnements urbains, des zones montagneuses ou des ciels nuageux, avec des drones parfois réduits à quelques pixels. Ces jeux de données constituent une base importante pour les travaux récents axés sur la détection de petites cibles



Fig. 7. Comparaison qualitative entre des images issues de datasets RGB grand public (ex. Kaggle Drone Dataset, Drone vs Bird) et des datasets spécialisés (ex. Anti-UAV, VMD, FRED).

et servent notamment de fondation à des datasets dérivés plus spécialisés.

3) Datasets neuromorphiques et hybrides: Le développement de la détection de drones par vision événementielle s'est accompagné de l'apparition progressive de jeux de données dédiés. Parmi les premiers, le dataset NERDD [15] constitue une tentative initiale d'exploitation de flux événementiels pour la détection de drones, en mettant l'accent sur la capture des signatures dynamiques telles que la rotation des hélices. Bien que limité en diversité de scènes, il a posé les bases de l'utilisation de données neuromorphiques pour la surveillance aérienne.

Dans une approche multimodale, le dataset FRED [21] a ensuite été proposé afin de combiner données RGB et événementielles synchronisées. Il permet l'étude de méthodes hybrides exploitant à la fois la richesse spatiale des images classiques et la résolution temporelle élevée des capteurs événementiels.

Le Visual Magnocellular Dynamics Dataset (VMD) [15] est un dataset dérivé des jeux Anti-UAV, conçu pour l'apprentissage supervisé de caractéristiques de mouvement inspirées de la voie magnocellulaire de la rétine. Il se compose de 650 séquences vidéo, avec une résolution réduite à 120×100 pixels afin de favoriser l'efficacité computationnelle, et met l'accent sur la dynamique visuelle plutôt que sur les détails spatiaux.

Plus récemment, les travaux de Magrini et al ont introduit des datasets événementiels plus réalistes, tels que EV-Flying [21], intégrant drones et autres objets volants dans des environnements non contrôlés. Ce type de dataset vise à évaluer la robustesse des approches événementielles dans des scénarios proches des conditions réelles.

4) Tableau comparatif des datasets drone incluant l'événementiel: Le Tableau I synthétise des datasets utilisés dans la littérature event-based drone.

TABLE I
COMPARAISON DE DATASETS DRONE AVEC DONNÉES ÉVÉNEMENTIELLES.

Dataset	Résolution	RGB / Event	Durée	Drone-centric	Types drones	Tâches
Anti-UAV [22]	1920×1080	✓ / ✗	4h:25m	✓	≥ 6	Det, Track
Pawelczyk et al. [24]	640×480	✓ / ✗	51446 frames	✓	–	Det
EED [29]	240×180	✗ / ✓	$\leq$ 0h:30m	✗	–	Det, Track
VisEvent [26]	346×260	✓ / ✓	$\leq$ 5h	✗	–	Det
EventVOT [27]	1280×720	✗ / ✓	$\leq$ 5h	✗	–	Det
F-UAV-D [28]	1280×720	✓ / ✓	0h:30m	✓	2	Det
Ev-UAV [28]	346×240	✗ / ✓	0h:15m	✓	–	Det, Track
Ev-Flying [21]	1280×720	✗ / ✓	1h:07m	✗	1	Det, Track
NeRDD [13]	1280×720	✓ / ✓	3h:30m	✓	2	Det
CRSOT [30]	1280×720	✓ / ✓	2h:50m	✓	–	Det, Track
FRED [19]	1280×720	✓ / ✓	7h:07m	✓	5	Det, Track, Forecast

B. Comparaison des méthodes au regard des datasets

Les méthodes basées sur des réseaux de neurones artificiels appliquées à des données RGB dominent historiquement la littérature, en particulier lorsqu'elles exploitent des datasets publics ou internes. Ces approches offrent de bonnes performances de détection, mais leur efficacité dépend fortement de la qualité et de la représentativité des données d'entraînement.

Les travaux récents s'appuyant sur des approches bio-inspirées ou neuromorphiques visent à pallier certaines limites des méthodes classiques, notamment en exploitant l'information temporelle et le mouvement pour améliorer la détection de petites cibles dans des arrière-plans complexes. Toutefois, ces approches reposent souvent sur des datasets spécifiques et des protocoles d'évaluation non standards, ce qui rend les comparaisons directes avec les pipelines RGB classiques délicates.

1) Résultats de détection sur un benchmark commun: Le Tableau II présente des résultats de détection (mAP_{50} et $mAP_{50:95}$) évalués avec deux splits (*canonical* et *challenging*), afin de mieux mesurer la robustesse aux variations de distribution.

TABLE II
RÉSULTATS DE DÉTECTION (CANONICAL VS CHALLENGING).

Méthode	Event	RGB	Canonical split		Challenging split	
			mAP_{50}	$mAP_{50:95}$	mAP_{50}	$mAP_{50:95}$
YOLO [30]	✓	✗	87.68	49.25	79.60	41.63
YOLO [30]	✗	✓	35.24	13.40	18.13	6.33
RT-DETR [31]	✓	✗	82.05	38.98	76.93	35.08
RT-DETR [31]	✗	✓	34.12	11.79	21.12	7.13
Faster-RCNN [32]	✓	✗	85.00	43.40	75.40	34.90
Faster-RCNN [32]	✗	✓	35.10	12.30	16.40	4.80
ER-DETR [33]	✓	✗	68.42	21.80	62.20	19.81
ER-DETR [33]	✗	✓	28.56	7.73	15.84	4.11
ER-DETR [33]	✓	✓	78.59	32.21	75.73	27.87

2) Résultats de suivi multi-objets: Le Tableau III présente des résultats de suivi multi-objets (*tracking-by-detection*), évalués avec MOTA, IDF1, changements d'identité (IDs), précision (P) et rappel (R).

TABLE III
RÉSULTATS DE SUIVI MULTI-OBJETS (TRACKING-BY-DETECTION).

Méthode	Event	RGB	MOTA↑	IDF1↑	IDs↓	P↑	R↑
YOLO [30]	✓	✗	57.1	47.6	436	83.0	72.2
YOLO [30]	✗	✓	21.5	28.7	9830	75.7	32.2
RT-DETR [31]	✓	✗	51.3	45.1	301	77.4	72.8
RT-DETR [31]	✗	✓	10.1	30.0	692	57.1	42.6
Faster-RCNN [32]	✓	✗	64.0	45.0	4191	83.8	83.1
Faster-RCNN [32]	✗	✓	-23.1	24.7	2298	40.7	46.8
ER-DETR [33]	✓	✗	45.1	42.1	476	74.3	69.5
ER-DETR [33]	✗	✓	-13.4	0.5	2345	13.3	2.1
ER-DETR [33]	✓	✓	46.2	43.3	431	75.5	69.0

C. Analyse critique transversale

L'analyse conjointe des datasets et des méthodes met en évidence plusieurs points clés. Les datasets RGB grand public, bien qu'utiles pour le prototypage et l'entraînement initial, ne suffisent pas à évaluer la robustesse des systèmes en conditions réelles. La détection de drones de très petite taille demeure un défi central, nécessitant des adaptations spécifiques des métriques et des protocoles d'évaluation.

1) Fisheye et caméras événementielles à grand champ de vision: Un axe complémentaire, encore peu couvert dans les benchmarks de détection de drones, est l'utilisation de caméras événementielles à très grand champ de vision (type fisheye/omnidirectionnel). Dans l'article *A New Stereo Fisheye Event Camera for Fast Drone Detection and Tracking* [20] , les auteurs décrivent un montage stéréo événementiel dos-à-dos visant une couverture à 360 degrés, ce qui augmente la probabilité de conserver une cible rapide dans le champ sans mécanisme de pointage. Pour des scénarios de surveillance, cette approche peut réduire des pertes de suivi dues à la sortie de champ, au prix de distorsions géométriques (fisheye) qui nécessitent un traitement adapté.

2) Piste de génération de données synthétiques: Une piste à intégrer est la génération de données synthétiques (par exemple via un moteur 3D) afin d'augmenter la diversité : trajectoires contrôlées, tailles apparentes, arrière-plans variés, conditions météo et domaines jour/nuit, avec annotations parfaites (boîtes, identités, trajectoires). Cette approche est mentionnée comme perspective dans la littérature sur la détection de drones par caméras événementielles. Elle doit toutefois être discutée avec prudence, car un écart de domaine entre simulation et réel (textures, bruit capteur, artefacts événementiels) peut limiter le transfert si la simulation n'est pas suffisamment réaliste.

VII. EXPÉRIENCES, RÉSULTATS ET ANALYSE

A. Objectif et positionnement expérimental

L'objectif de cette campagne expérimentale est d'évaluer la faisabilité de plusieurs architectures de détection de drones sur données événementielles réelles, en utilisant le dataset FRED [19] comme benchmark commun. Trois architectures ont été confrontées dans des conditions identiques : un détecteur de référence non-spiké (YOLOv8 Nano), un modèle SNN pur entraîné from-scratch (ConvLIF / SNN-YOLO v1+), et un détecteur hybride SNN-ANN pré-entraîné (SpikeYOLO). Ce protocole commun permet une comparaison directe et rigoureuse, contrairement à la majorité des travaux de la littérature qui évaluent leurs modèles sur des datasets hétérogènes.

B. Protocole expérimental commun

1) Dataset et partitionnement: L'ensemble des expériences est conduit sur les trois premières séquences du dataset FRED, soit un total de 7 322 frames annotées. Le partitionnement retenu suit un découpage par séquence afin d'éviter toute fuite de données temporelles : les séquences 0 et 1 constituent le jeu d'entraînement (4 290 frames), tandis que la séquence 2 est réservée à la validation (3 032 frames). Ce choix garantit que le modèle est évalué sur une distribution temporelle inédite, représentative d'une utilisation en conditions réelles.

2) Modalité d'entrée: Tous les modèles reçoivent en entrée des *event frames* au format YOLO (dossier `Event_YOLO`), reconstruits par accumulation temporelle des événements DVS sur une fenêtre fixe. Chaque frame est une image en niveaux de gris où l'intensité d'un pixel code la densité d'événements reçus sur la fenêtre d'accumulation. Ce format, nativement compatible avec la bibliothèque Ultralytics, assure une base d'entrée identique pour les trois architectures et facilite la comparaison. Les frames RGB synchronisées sont également disponibles dans le dataset (dossier `RGB_YOLO`) mais ne sont pas exploitées dans ce travail, toutes les architectures opérant en modalité événementielle pure.

3) Métriques d'évaluation: Les performances sont mesurées par les métriques standard de la détection d'objets :

- **mAP@0.5** : aire sous la courbe Précision-Rappel avec seuil IoU de 0.5 ;
- **mAP@0.5:0.95** : moyenne des mAP calculées pour des seuils IoU de 0.5 à 0.95 par pas de 0.05, métrique plus exigeante sur la qualité de localisation ;
- **Précision**, **Rappel** et **F1-score** à seuil de confiance fixé.

C. Architecture de référence : YOLOv8 Nano

1) Description du modèle: YOLOv8 Nano est utilisé comme baseline non-spiké afin d'établir un plafond de performance atteignable par un détecteur ANN classique sur données événementielles. Avec 3,2 millions de paramètres, ce modèle présente un équilibre favorable entre capacité représentationnelle et efficacité mémoire. Son architecture repose sur trois composants principaux : un *backbone* extrayant les caractéristiques par convolutions et modules C2f, un *neck* assurant la fusion multi-échelle des cartes de caractéristiques, et une *tête de détection découplée* sans ancres (anchor-free).

2) Protocole d'entraînement: Le modèle est entraîné par transfer learning depuis les poids COCO, sur un GPU T4 (Colab), pendant 20 époques avec un batch de 16, des frames redimensionnées à 640×640 pixels, et l'optimiseur AdamW. La durée totale d'entraînement est d'environ 1h30.

3) Résultats: Le modèle atteint des performances remarquables sur le jeu de validation de FRED (Tableau IV). Avec un mAP@0.5 de **96,40%** et un F1-score de **95,4%**, il confirme que les event frames reconstruites sont suffisamment structurées pour être exploitées par un détecteur convolutif standard. La précision de 97,5% indique un taux de fausses alarmes très bas, tandis que le rappel de 93,3% traduit une détection fiable des drones présents dans la scène. Le poids du modèle de 6,2 Mo suggère en outre un déploiement envisageable sur des systèmes embarqués légers (Jetson Nano, Raspberry Pi 5).

Les courbes d'apprentissage montrent une descente régulière et monotone des pertes d'entraînement et de validation (box loss, classification loss, DFL loss), sans signe de sur-apprentissage sur 20 époques. Les métriques de validation (mAP@0.5, précision, rappel) progressent continûment jusqu'à la dernière époque, ce qui indique que le modèle n'a pas encore entièrement convergé et que davantage d'époques pourraient encore améliorer les performances.

D. Architecture SNN pure : ConvLIF (SNN-YOLO v1+)

1) Description du modèle: ConvLIF est une architecture SNN développée *from scratch*, visant à exploiter directement la nature impulsionnelle des données DVS. Le backbone est composé de 6 blocs ConvLIF, chacun associant une convolution 3×3, une normalisation par batch et un neurone Leaky Integrate-and-Fire (LIF). Le nombre de canaux est doublé à chaque bloc selon la progression $1 \to 32 \to 64 \to 128 \to 256 \to 512 \to 512$, pour un total d'environ 4 millions de paramètres. La tête de détection projette les caractéristiques extraites vers une grille 7×7 (inspiration YOLOv1) : chaque cellule prédit un unique vecteur $[\mathrm{conf}, x, y, w, h]$.

Le flux d'événements est encodé par *rate coding* sur 16 pas de temps (NUM_STEPS = 16), avec un coefficient de fuite $\beta = 0{,}9$ pour les neurones LIF. Ce paramètre contrôle la vitesse à laquelle le potentiel membranaire se dissipe entre deux spikes successifs, assurant une intégration temporelle sur une fenêtre glissante.

2) Protocole d'entraînement: L'entraînement est conduit sur GPU Tesla P100 (16 Go, Kaggle), pendant 60 époques, avec l'optimiseur AdamW. Le taux d'apprentissage suit un échauffement cosinus sur les 6 premières époques (LR max = 5×10^{-4}, LR min = 10^{-5}), puis décroît selon une courbe cosinus jusqu'à la fin de l'entraînement. Les frames d'entrée sont redimensionnées à 128×128 pixels, ce qui est inférieur à la résolution utilisée par les deux autres modèles.

3) Résultats: Sur le jeu de validation de FRED, ConvLIF obtient un mAP@0.5 de **53,62%**, une précision de 0,53, un rappel de 0,68 et un F1-score de 0,60. Ces résultats sont cohérents entre les jeux d'entraînement et de validation (mAP train = 53,66%), ce qui indique une absence de sur-apprentissage et une bonne généralisation du modèle à la séquence inédite.

Les courbes de loss présentent une convergence nette : la loss totale passe d'environ 1,8 en début d'entraînement à moins de 0,25 à l'époque 60, et les quatre composantes (loss totale, loss de coordonnées, loss de confiance positive, loss de confiance négative) décroissent toutes de façon monotone. Cependant, la pente est encore légèrement négative à 60 époques, suggérant qu'un entraînement plus long pourrait améliorer les performances.

4) Analyse des limites: Plusieurs facteurs limitent les performances du ConvLIF dans sa version actuelle. Premièrement, la tête de détection de style YOLOv1 ne prédit qu'une boîte par cellule de grille, rendant le modèle incapable de détecter plusieurs drones dans une même cellule. Ce choix architectural est approprié pour les séquences de FRED qui contiennent généralement un seul drone par frame, mais il constitue un frein à la généralisation. Deuxièmement, la réduction de résolution à 128×128 pixels, dictée par les contraintes mémoire liées au déroulement temporel sur 16 pas de temps, pénalise la détection de petites cibles. Troisièmement, 60 époques se sont avérées insuffisantes pour une convergence complète, comme en témoigne la pente résiduelle des courbes de loss.

E. Architecture hybride SNN-ANN : SpikeYOLO

1) Description du modèle: SpikeYOLO est une variante de YOLOv8 dans laquelle les activations ReLU/SiLU ont été remplacées par des neurones LIF, introduisant le comportement impulsionnel au sein d'une architecture de détection éprouvée. Le modèle conserve la structure macro de YOLOv8 (backbone, neck, head découplée), mais chaque convolution est suivie d'un neurone LIF plutôt que d'une activation continue. Au niveau micro, les blocs C2f sont remplacés par des *Meta SNN Blocks* combinant convolutions séparables (SepConv), convolutions par canaux (ChannelConv) et connexions résiduelles. L'entrée est une séquence temporelle de frames d'événements de dimensions $[T, C, H, W]$. Avec 69 millions de paramètres, SpikeYOLO est significativement plus lourd que les deux autres architectures.

2) Protocole d'entraînement: SpikeYOLO est entraîné par transfer learning depuis les poids COCO sur GPU P100 (Kaggle), pendant 20 époques, avec un batch size de 4, un taux d'apprentissage de 10^{-3} et l'optimiseur AdamW. La durée totale d'entraînement est d'environ 4 heures, soit nettement plus longue que YOLOv8 Nano en raison de la simulation du comportement temporel des neurones LIF.

3) Résultats: SpikeYOLO atteint un mAP@0.5 de **90,78%**, une précision de 91,60%, un rappel de 89,51% et un F1-score de 90,54%. La meilleure performance en validation est enregistrée à l'époque 17 (mAP@0.5 = 92,2%). Le mAP@0.5:0.95 s'établit à 36,38%, révélant que si la détection (présence du drone) est robuste, la localisation précise de la boîte englobante reste imparfaite.

Les courbes d'apprentissage témoignent d'une instabilité notable : une dégradation brutale est observée à l'époque 18 sur les trois composantes de la loss de validation (box loss, classification loss, DFL loss), avant une partielle récupération à l'époque 20. Cette instabilité, caractéristique des SNN entraînés par substitution de gradient (*surrogate gradient*),

souligne la sensibilité de l'optimisation aux dynamiques impulsionnelles et justifie l'utilisation de learning rates plus faibles ou de techniques de stabilisation (gradient clipping, warm restarts).

F. Comparaison des trois architectures

Le Tableau IV synthétise les résultats des trois architectures sur le benchmark FRED commun.

TABLE IV
COMPARAISON DES ARCHITECTURES SUR LE JEU DE VALIDATION DE FRED (SÉQUENCE 2).

Modèle	Params	mAP@0.5	mAP@0.5:95	Précision	Rappel
YOLOv8 Nano	3,2M	**96,40%**	—	97,5%	93,3%
SpikeYOLO	69M	90,78%	36,38%	91,6%	89,5%
ConvLIF	4M	53,62%	—	53,0%	68,0%

L'analyse comparative met en évidence trois enseignements principaux.

(1) Supériorité du transfer learning. L'écart de 5,6 points de mAP@0.5 entre YOLOv8 Nano et SpikeYOLO, malgré un ratio de paramètres de 1:22 en faveur de SpikeYOLO, illustre l'efficacité du transfer learning sur des événements DVS convertis en frames 2D : les représentations COCO, bien qu'issues d'images RGB, transfèrent correctement vers les event frames en niveaux de gris. En revanche, ConvLIF, entraîné from scratch sur seulement 4 290 frames, souffre d'un manque de données représentatives pour apprendre des caractéristiques discriminantes robustes.

(2) Faisabilité des SNN pour la détection sur événements réels. SpikeYOLO démontre qu'un détecteur partiellement spiké peut atteindre des performances très proches d'un CNN pur, avec un mAP@0.5 de 90,78% en 20 époques seulement. Ce résultat constitue une preuve de concept solide pour la viabilité du paradigme impulsionnel sur le benchmark FRED, en conditions adverses (pluie, nuit, multi-drones).

(3) Sensibilité à la résolution d'entrée et à la complexité de la tête de détection. La limitation à 128×128 pixels et à une seule boîte par cellule de grille pénalise significativement ConvLIF. L'adoption d'une tête multi-ancres de style YOLOv3 ou YOLOv5 adaptée en SNN, couplée à une résolution d'entrée plus élevée (256×256), représente la piste d'amélioration la plus directe.

G. Positionnement par rapport à la littérature

Nos résultats peuvent être mis en perspective avec les travaux de l'état de l'art présentés en Section VI, et notamment avec les benchmarks de détection publiés sur des données événementielles.

Sur le dataset FRED, les résultats de référence rapportés par Magrini *et al.* [12] indiquent un mAP@0.5 de 87,68% pour un modèle YOLO standard entraîné sur données événementielles seules (*canonical split*). Notre baseline YOLOv8 Nano atteint 96,40% sur la séquence 2, soit un gain de près de 9 points, ce qui s'explique en partie par une différence de split (nos séquences 0, 1, 2 vs. le split canonique de FRED) et par l'utilisation du transfer learning depuis COCO. Ces différences de protocole invitent à la prudence dans la comparaison directe des chiffres, mais confirment que les event frames constituent une modalité d'entrée exploitable par les détecteurs YOLO modernes.

Les travaux sur SpikeYOLO (BICLab, ECCV 2024) [12], à partir desquels notre implémentation est dérivée, rapportent des performances compétitives avec YOLOv8 sur des tâches de détection d'objets générales. Nos résultats sur FRED (mAP@0.5 = 90,78% en seulement 20 époques) s'inscrivent dans cette ligne, bien en deçà de la limite supérieure du YOLOv8 Nano mais à un niveau opérationnel pour un système de détection temps réel.

L'architecture ConvLIF, avec un mAP@0.5 de 53,62%, se situe au niveau des résultats rapportés pour des SNN purs entraînés sur de petits datasets événementiels. À titre de comparaison, le système SNN sur puce Speck étudié par [14] vise la classification de drones (et non la détection avec localisation), une tâche structurellement plus simple, mais avec une consommation de seulement 10 mW contre 2,64 W pour un ANN équivalent. Notre ConvLIF, simulé sur GPU, ne présente pas d'avantage énergétique dans ce contexte, mais sa conception modulaire le rend directement transposable sur hardware neuromorphique dédié (Loihi 2, Akida).

Enfin, par rapport aux résultats du Tableau II (état de l'art FRED, *canonical* vs *challenging split*), il ressort que les données événementielles surpassent systématiquement le RGB pour la détection de drones sur FRED, quel que soit le détecteur employé (YOLO Event : mAP@0.5 = 87,68% vs YOLO RGB : 35,24%). Nos expériences, conduites exclusivement sur event frames, s'inscrivent dans cette même tendance et renforcent l'intérêt de la modalité DVS pour la surveillance de petites cibles rapides.

H. Discussion et perspectives

1) Limites liées au dataset: L'une des limites principales de notre évaluation réside dans l'absence de distracteurs dans FRED : le dataset ne contient que des drones, sans oiseaux, avions ou ballons. Nos modèles n'ont donc jamais appris à rejeter de tels objets, ce qui pourrait entraîner un taux élevé de faux positifs en conditions de déploiement réel. L'enrichissement du jeu d'entraînement avec des séquences de distracteurs, ou l'évaluation croisée sur des datasets comme Anti-UAV [22] ou NeRDD [13], est un prérequis indispensable avant tout déploiement opérationnel.

Par ailleurs, nos modèles sont entraînés sur des event frames reconstruites par accumulation temporelle, une étape qui introduit une perte d'information liée à la compression du flux asynchrone en une représentation 2D. L'entraînement directement sur le flux Prophesee brut (`events.raw`), via le framework SpikingJelly, permettrait d'exploiter pleinement la résolution temporelle microseconde du capteur DVS et représente une direction de recherche prioritaire.

2) Perspectives d'amélioration: Plusieurs axes d'amélioration peuvent être identifiés à l'issue de cette campagne expérimentale. Pour ConvLIF, l'augmentation de

la résolution d'entrée (256×256), l'adoption d'une tête de détection multi-ancres (YOLOv3-SNN), et un entraînement prolongé au-delà de 100 époques constitueraient des évolutions directes. Pour SpikeYOLO, la stabilisation de la phase d'entraînement par gradient clipping et l'exploration d'un learning rate schedule plus conservateur pourraient réduire l'instabilité observée à l'époque 18. Plus généralement, la construction d'un pipeline bout-en-bout DVS $\rightarrow$ accumulation $\rightarrow$ inférence SNN permettrait d'envisager un déploiement sur hardware neuromorphique dédié, concrétisant l'avantage énergétique théorique des architectures impulsionnelles.

VIII. Conclusion

La détection de drones en environnements complexes représente un défi de sécurité croissant que les technologies de vision conventionnelles peinent à relever efficacement, principalement en raison de contraintes de latence et de consommation énergétique. Cet état de l'art a mis en évidence que si les approches basées sur le Radar et la vision RGB couplée au Deep Learning offrent des performances élevées en conditions idéales, elles atteignent leurs limites face à des cibles rapides, de petite taille, ou lors d'un déploiement sur systèmes embarqués à budget énergétique strict.

La vision événementielle (DVS), associée aux Réseaux de Neurones Impulsionnels (SNN), marque une rupture technologique majeure. En exploitant la parcimonie temporelle et le fonctionnement asynchrone, cette combinaison permet une réduction drastique de la bande passante et une réactivité de l'ordre de la microseconde, indispensable pour capturer la cinématique des rotors. Bien que le matériel neuromorphique (Loihi, Akida) et les algorithmes de conversion (ANN-to-SNN, Spiking Transformers) soient en pleine maturation, le manque de datasets standardisés et réalistes reste un frein à l'adoption massive. Les travaux futurs devront se concentrer sur la fusion multimodale (RGB-DVS) et l'optimisation des architectures SNN directement sur puce pour concrétiser des systèmes de défense autonomes et pérennes.

References

[1] S. Björklund and N. Wadströmer, "Target detection and classification of small drones by deep learning on radar micro-doppler," in *2019 International Radar Conference (RADAR)*, Toulon, France, 2019, pp. 1-6.

[2] J. Zhang, R. Woods, M. Sandell, M. Valkama, A. Marshall, and J. Cavallaro, "Radio frequency fingerprint identification for narrowband systems, modelling and classification," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3974-3987, 2021.

[3] S. Dogru and L. Marques, "Drone detection using sparse lidar measurements," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3062-3069, 2022.

[4] P. Bhattacharya and P. Nowak, "Drone Detection and Tracking with YOLO and a Rule-based Method," *arXiv preprint arXiv:2502.05292*, 2025.

[5] H. Liu, F. Qu, Y. Liu, W. Zhao, and Y. Chen, "A drone detection with aircraft classification based on a camera array," in *IOP Conference Series: Materials Science and Engineering*, vol. 322, p. 052005, 2018.

[6] "YOLOv8-Based Drone Detection: Building a Robust Model with Extensive Data," [Online Source/GitHub Repository].

[7] E. Unlu, E. Zenou, N. Riviere, and P. E. Dupouy, "Deep learning-based strategies for the detection and tracking of drones using several cameras," *IPSJ Transactions on Computer Vision and Applications*, vol. 11, no. 1, p. 7, 2019.

[8] D. Chen, Y. Hu, and L. Zhang, "HierLight-YOLO: A Hierarchical and Lightweight Object Detection Network for UAV Photography," *arXiv preprint arXiv:2509.22365*, 2025.

[9] Z. Liu, P. An, Y. Yang, S. Qiu, Q. Liu, and X. Xu, "Vision-based drone detection in complex environments: a survey," *Drones*, vol. 8, no. 11, p. 643, 2024.

[10] T. Stewart, M. A. Drouin, M. Picard, F. B. Djupkep Dizeu, A. Orth, and G. Gagné, "A virtual fence for drones: Efficiently detecting propeller blades with a dvxplorer event camera," in *Proceedings of the International Conference on Neuromorphic Systems (ICONS)*, 2022, pp. 1-7.

[11] G. Magrini, F. Becattini, P. Pala, A. Del Bimbo, and A. Porta, "Neuromorphic drone detection: An event-RGB multimodal approach," in *European Conference on Computer Vision (ECCV)*, 2024.

[12] Magrini, Gabriele, et al. "Drone Detection with Event Cameras." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2025.

[13] T. Stewart *et al.*, "Drone Detection Using a Low-Power Neuromorphic Virtual Tripwire," in *Proceedings of the International Conference on Neuromorphic Systems (ICONS)*, 2022.

[14] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Networks*, vol. 111, pp. 47–63, 2019.

[15] K. Ranaweera, W. D. Silva, and J. Arachchige, "Motion feature extraction using magnocellular-inspired spiking neural networks for drone detection," *IEEE Sensors Journal*

[16] Dasmehdixtr, "Drone Dataset (UAV)," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/dasmehdixtr/drone-dataset-uav.

[17] Muki2003, "YOLO Drone Detection Dataset," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/muki2003/yolo-drone-detection-dataset.

[18] H. Walia, "Birds vs Drone Dataset," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/harshwalia/birds-vs-drone-dataset.

[19] "The Florence RGB-Event Drone Dataset ," GitHub. [Online]. Available: https://miccunifi.github.io/FRED/.

[20] C. Teuliere *et al.*, "A New Stereo Fisheye Event Camera for Fast Drone Detection and Tracking," HAL Open Science. [Online]. Available: https://hal.science/hal-04976415/document.

[21] G. Magrini *et al.*, "EV-Flying: an Event-based Dataset for In-The-Wild Recognition of Flying Objects," *arXiv preprint arXiv:2506.04048*, 2025.

[22] N. Jiang *et al.*, "Anti-UAV: A large-scale benchmark for vision-based UAV tracking," *IEEE Transactions on Multimedia*, 2021.

[23] M. Pawełczyk and M. Wojtyra, "Real world object detection dataset for quadcopter unmanned aerial vehicle detection," *IEEE Access*, vol. 8, pp. 174394–174409, 2020.

[24] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[25] X. Wang *et al.*, "VisEvent: Reliable object tracking via collaboration of frame and event flows," *IEEE Transactions on Cybernetics*, 2023.

[26] X. Wang *et al.*, "Event stream-based visual object tracking: A high-resolution benchmark dataset and a novel baseline," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[27] J. Mandula, J. Kühne, L. Pascarella, and M. Magno, "Towards real-time fast unmanned aerial vehicle detection using dynamic vision sensors," in *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2024.

[28] N. Chen, C. Xiao, Y. Dai, S. He, M. Li, and W. An, "Event-based tiny object detection: A benchmark dataset and baseline," *arXiv preprint arXiv:2506.23575*, 2025.

[29] Y. Zhu *et al.*, "CRSOT: Cross-resolution object tracking using unaligned frame and event cameras," *arXiv preprint arXiv:2401.02826*, 2024.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[31] F. Feng, Y. Li, Y. Zhang, *et al.*, "DETRs with Real-Time Constraints," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[32] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 39, no. 6, pp. 1137-1149, 2017.

[33] Z. Liu, Y. Tang, Y. Han, *et al.*, "ER-DETR: Efficient Regression Transformer for Real-Time Object Detection," *arXiv preprint arXiv:2303.16789*, 2023.
[34] Guo, Y., Liu, X., Chen, Y., Peng, W., Zhang, Y., Ma, Z. (2025). Spiking Transformer: Introducing Accurate Addition-Only Spiking Self-Attention for Transformer. In Proceedings of the Computer Vision and Pattern Recognition Conference (pp. 24398-24408).
[35] Wang, S., Zhang, M., Zhang, D., Belatreche, A., Xiao, Y., Liang, Y., Yang, Y. (2025). Spiking vision transformer with saccadic attention. arXiv preprint arXiv:2502.12677.
[36] A Robust, Open-Source Framework for Spiking Neural Networks on Low-End FPGAs. Andrew Fan, Simon D. Levy