

Détection Automatique des Traînées Astronomiques avec YOLO – Une Approche Exploratoire pour la Connaissance du Domaine Spatial

Loshan RASAN¹, Sonimith HANG¹, Xhesika LACI¹, Binbin XU²

¹Department of Computer Science & Artificial Intelligence, IMT Mines Ales

²EuroMov Digital Health in Motion, Univ. Montpellier, IMT Mines Ales

loshan.rasan@mines-ales.org, sonimith.hang@mines-ales.org, xhesika.laci@mines-ales.org,
binbin.xu@mines-ales.fr

Résumé

L'identification des traînées transitoires est cruciale pour la Connaissance du Domaine Spatial, mais les méthodes classiques sont coûteuses et inadaptées au temps réel. Nous explorons YOLOv8m pour cette tâche, en l'entraînant sur le StreaksYoloDataset (2 388 images annotées, télescope Stelina). L'approche prend en compte les variations de bruit, d'éclairement et de champ de vision. Les résultats montrent un bon équilibre entre rapidité et précision (mAP@50-95 : 0,90), bien que des améliorations soient nécessaires pour limiter les faux positifs et détecter les traînées faibles.

Mots-clés

détection de traînées astronomiques, YOLO, Connaissance du Domaine Spatial, Veille Spatiale, suivi des satellites, détection d'objets

1 Introduction

L'intensification des activités spatiales accentue le besoin de Connaissance du Domaine Spatial (Veille Spatiale), car les satellites, les débris spatiaux et les rayons cosmiques perturbent les observations astronomiques. Une détection efficace de ces traînées transitoires est donc essentielle pour la surveillance spatiale et l'astronomie.

Les approches classiques de détection des traînées astronomiques reposent principalement sur des méthodes telles que le seuillage [1], la transformée de Hough [2], la transformée de Radon [3], la détection de contours [4, 5], la soustraction d'images [6] etc. Bien que ces techniques soient efficaces dans des cas spécifiques, elles présentent plusieurs limitations. Le seuillage, par exemple, est rapide mais très sensible au bruit et aux variations d'intensité, rendant la détection imprécise. La transformée de Hough est performante pour les traînées linéaires mais souffre d'un coût computationnel élevé, limitant son application en temps réel. De même, les méthodes de détection de contours dépendent fortement des paramètres choisis et sont inefficaces pour les traînées diffuses ou à faible contraste. La soustraction d'images, bien qu'efficace pour détecter des objets transitoires, nécessite une image de référence propre et peut être

perturbée par des variations instrumentales.

Toutefois, ces méthodes traditionnelles atteignent rapidement leurs limites face à la complexité croissante des observations astronomiques. Cette contrainte a conduit au développement de nouvelles approches basées sur l'apprentissage profond et automatique, notamment les réseaux de neurones, qui ont démontré une performance supérieure dans la détection automatique des traînées. Varela et al. (2019) ont montré la supériorité des réseaux de neurones convolutionnels (CNN) sur les méthodes classiques pour la détection de traînées [7]. Pöntinen et al. (2020) ont évalué StreakDet, démontrant son efficacité pour l'extraction automatique de traînées dans des images simulées d'Euclid [8].

Ces modèles restent néanmoins coûteux en calcul et peu adaptés aux contraintes du temps réel. Dans cette étude, nous proposons une approche basée sur YOLOv8m [9], un détecteur rapide et optimisé pour l'identification automatique des traînées astronomiques, offrant une solution performante pour l'astronomie et la surveillance spatiale.

L'objectif de cette étude est de développer un modèle automatisé de détection des traînées astronomiques avec une haute précision, tout en garantissant un compromis optimal entre rapidité et fiabilité. Pour cela, nous évaluons et validons notre approche sur le StreaksYoloDataset [10], un ensemble d'images annotées permettant de tester la robustesse du modèle. Nous analysons également ses limites, notamment en ce qui concerne les sous-détections et les faux positifs, afin d'identifier des axes d'amélioration. Enfin, cette étude vise à fournir un outil évolutif, capable de répondre aux besoins croissants de la communauté astronomique en matière de surveillance et d'analyse des traînées transitoires.

2 Méthodologie

2.1 Dataset

L'efficacité d'un modèle d'apprentissage profond dépend fortement de la qualité et de la diversité des données utilisées pour son entraînement. Dans cette étude, nous utilisons le StreaksYoloDataset [10], un ensemble d'images astrono-

miques brutes capturées entre mars 2022 et février 2023 avec des télescopes intelligents Stellina, dans la région du Luxembourg. Ce dataset contient des images avec une résolution de 640 x 640 pixels, prises à l'aide d'un télescope Stellina, qui utilise un doublet à faible dispersion (ouverture de 80 mm, focale de 400 mm, ratio focal de f/5) et un capteur CMOS Sony IMX178 d'une résolution de 6,4 millions de pixels. Les images sont annotées avec la position des traînées, qui représentent des satellites, des débris spatiaux ou des rayons cosmiques.

Le dataset est structuré selon le format YOLO, avec des fichiers séparés pour les images en format JPEG (compression minimale) et des fichiers texte contenant les annotations des positions des traînées. Ces fichiers sont compatibles avec des outils d'entraînement de pointe et des logiciels graphiques comme MakeSense [11]. Le *StreaksYoloDataset* permet d'entraîner des modèles de détection pour la surveillance du domaine spatial (SDA), et offre ainsi une solution adaptée aux équipements astronomiques accessibles au public.

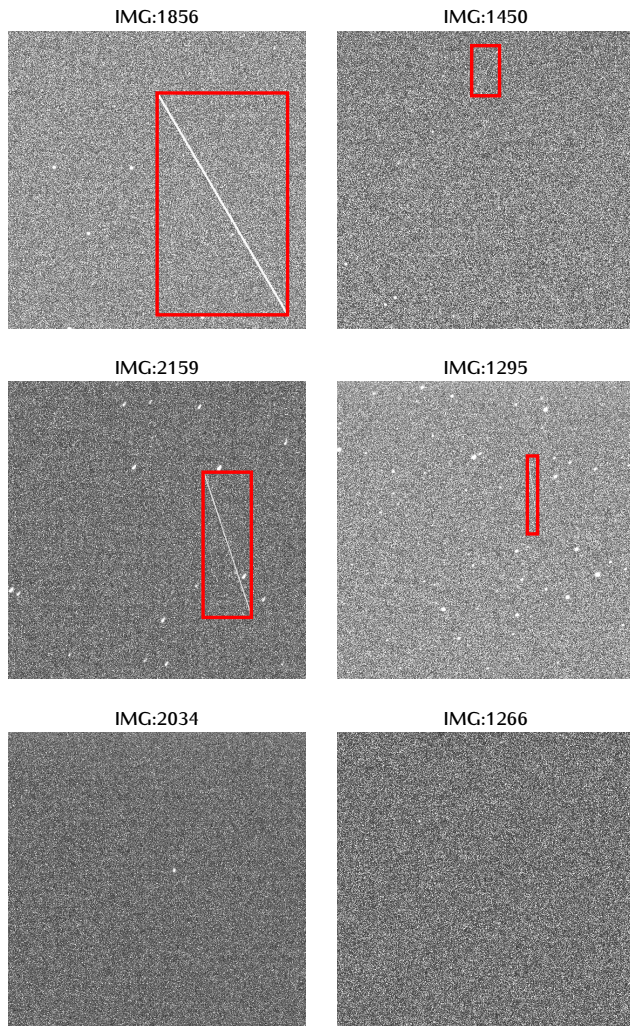


FIGURE 1 – Exemples d'images dans le StreaksYoloDataset. Les "streaks" sont annotés en rectangle rouge. Un "streak" pourrait être absent dans certaines images.

Pour améliorer la robustesse du modèle, ce dataset couvre une large variabilité des conditions d'observation : ajustement du temps d'exposition en fonction de la luminosité, diversité des niveaux de bruit selon les conditions atmosphériques, prise en compte de l'éclairement variable, et capture sur des champs de vision variés, allant des zones denses en étoiles aux régions plus vides.

2.2 Modèle – YOLOv8

Les traînées (streaks) observés dans les images astronomiques résultent de passage des objets dans l'espace / l'atmosphère très bruité. Cela rend leur identification particulièrement complexe. Ce problème s'inscrit dans le cadre plus large de la détection d'objets en apprentissage automatique avec les réseaux de neurones profonds (DNN). Parmi les architectures courantes, on y retrouve notamment les R-CNN (Region-based Convolutional Neural Network) [12] et ses variants qui offrent une haute performance mais sont très coûteux en calcul ; ainsi que les SSD (Single Shot MultiBox Detector) [13] qui sont plus rapides mais souvent moins performants sur la détection des objets de petite taille.

Les images astronomiques sont particulièrement volumineuses : les images de très haute résolution sont très rarement compressées pour éviter toute perte d'informations. Dans cette étude, nous optons pour YOLO (You Only Look Once), une famille de détecteurs en état de l'art réputés pour leur compromis entre la vitesse et la précision. YOLOv8m publié en janvier 2023 [9], en particulier, se distingue par son architecture optimisée et son efficacité. YOLOv8 surpasse d'autres modèles de la même famille (YOLOv5, YOLOv6, YOLOv7), en offrant une meilleure performance tout en réduisant la taille du modèle et la latence (Fig. 2).

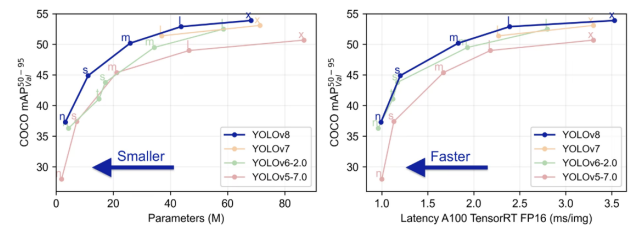


FIGURE 2 – Comparaison de YOLOv8 avec les versions précédentes de YOLO (YOLOv5, YOLOv6, YOLOv7) en termes de précision (mAP) par rapport à la taille du modèle (gauche) et de vitesse d'inférence (droite). Source [14]

Un avantage clé de YOLOv8m réside dans sa facilité de fine-tuning, grâce aux bibliothèques bien documentées d'Ultralytics, permettant une adaptation rapide à des ensembles de données spécifiques. Les principales caractéristiques de YOLOv8m, la variante utilisée dans cette étude, sont les suivantes :

- 25,9 millions de paramètres, répartis sur 295 couches.
- Coût computationnel : 79,1 GFLOPs.
- Optimisé pour l'inférence, offrant une précision élevée tout en restant léger.

2.3 Configuration de l'Entraînement

Bien que les images astronomiques diffèrent des images naturelles utilisées pour l'entraînement des modèles YOLO, la structure des traînées (streaks) présente des similarités avec des formes simples telles que des lignes ou des rectangles très étroites sur un fond complexe. Le fine-tuning d'un modèle pré-entraîné sur un large corpus d'images permet ainsi de tirer parti des représentations apprises tout en adaptant le modèle aux spécificités des traînées astronomiques. Toutefois, l'un des inconvénients de cette approche est que le modèle conserve une architecture relativement lourde. Une réduction du nombre de paramètres, en sélectionnant un modèle plus compact, pourrait améliorer son efficacité sur des dispositifs à ressources limitées.

Hyperparamètres et Protocole d'Entraînement

L'entraînement du modèle a été effectué sur 75 époques avec un batch-size de 32 et une résolution d'image de 640×640 pixels. L'optimisation repose sur SGD (Stochastic Gradient Descent) avec un taux d'apprentissage initial de 0.001, un momentum de 0.9 et une pénalisation L2 (weight decay) de 0.0005 pour éviter le sur-apprentissage.

Pour améliorer la généralisation, une technique de label smoothing (0.05) a été appliquée pour atténuer la sur-confiance du modèle. De plus, l'entraînement a été accéléré grâce à l'Automatic Mixed Precision (AMP), permettant une réduction significative de l'utilisation de mémoire. Les poids du modèle ont été sauvegardés dans un répertoire projeté pour assurer un suivi efficace des expériences.

Cette configuration vise à maximiser l'apprentissage du modèle tout en équilibrant précision et efficacité computationnelle.

Validation du Modèle et Seuil de Confiance

L'impact du seuil de confiance sur la précision, le rappel et le F1-score a été analysé sur l'ensemble de validation, avec un seuil IoU de 0.70 assurant une correspondance rigoureuse entre les prédictions et les annotations (Fig. 3). Un seuil de 0.4 a été retenu pour réduire les faux positifs tout en maintenant un bon rappel. Ce paramètre reste ajustable en fonction des exigences spécifiques de l'application.

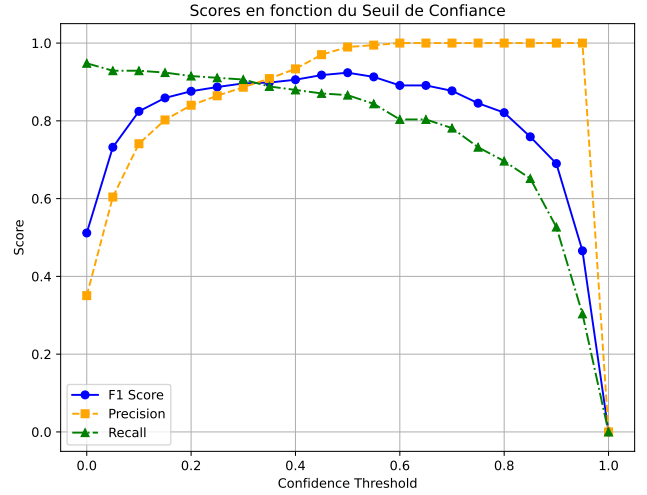


FIGURE 3 – Impact du seuil de confiance sur la précision, le rappel et le F1-score (validation, IoU = 0.70).

Le modèle a ensuite été évalué sur l'ensemble de test.

2.4 Évaluation des Performances

Nous évaluons le modèle à l'aide de plusieurs métriques, notamment la mAP@50-95, la précision, le rappel et le F1-score. La mAP@50-95 mesure la précision moyenne sur plusieurs seuils IoU (0.50 à 0.95), équilibrant précision et rappel pour une évaluation fiable [15, 16].

La précision (precision) représente la proportion de détections correctes parmi l'ensemble des prédictions, tandis que le rappel (recall) quantifie la capacité du modèle à détecter toutes les traînées existantes. Le F1-score est la moyenne harmonique de la précision et du rappel, offrant une mesure globale de la performance [16]. Enfin, la spécificité reflète la capacité du modèle à éviter les fausses détections.

Les métriques utilisées suivent les standards établis dans le domaine de la détection d'objets, notamment ceux définis dans les travaux du PASCAL VOC Challenge [15]. De plus, des analyses comparatives approfondies ont été menées sur ces métriques pour évaluer leur pertinence dans différents contextes de détection [16, 17].

2.4.1 Métrique et Seuils IoU

L'Intersection over Union (IoU) mesure le chevauchement entre la boîte de prédiction et la vérité terrain [18] :

$$\text{IoU} = \frac{|B_p \cap B_g|}{|B_p \cup B_g|} \quad (1)$$

où B_p et B_g sont respectivement la boîte de prédiction et la boîte de vérité terrain.

Un seuil IoU optimal de 0.7 a été retenu pour équilibrer précision et rappel.

Seuil IoU	Effet
0.50	Plus de détections, risque de faux positifs
0.75	Bon compromis précision-rappel
0.95	Haute précision, risque de faux négatifs

TABLE 1 – Impact des seuils IoU sur la détection.

3 Résultats et Discussion

3.1 Courbe de Perte

Pour évaluer l'évolution de l'apprentissage du modèle YOLOv8m, la Figure 4 présente la variation des pertes d'entraînement et de validation, ainsi que l'évolution du mAP50-95, au fil des époques.

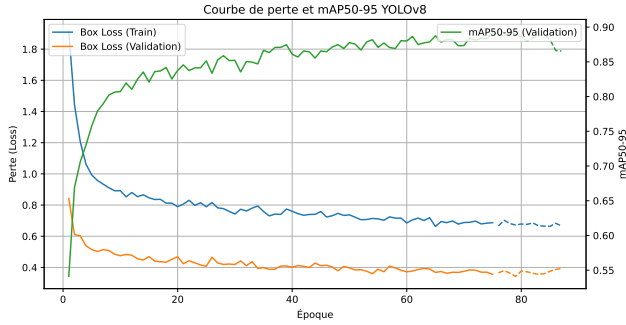


FIGURE 4 – Évolution de la perte (Loss) et du mAP50-95 pour l'entraînement et la validation de YOLOv8m.

La courbe de perte de validation reste inférieure à celle de l'entraînement et commence à converger de manière significative sur l'ensemble d'entraînement à partir d'environ 60 époques. La perte d'entraînement diminue rapidement avant de se stabiliser, tandis que la perte de validation suit une évolution similaire, témoignant d'un bon apprentissage du modèle. Parallèlement, le mAP50-95 progresse rapidement avant d'atteindre un plateau, indiquant une amélioration notable de la précision du modèle.

En vue des résultats, 75 époques est un bon compromis.

3.2 Performance du Modèle

Le modèle YOLOv8m a été évalué sur un ensemble de test de 333 images, atteignant des performances élevées tout en maintenant une faible complexité computationnelle. Les principales métriques sont résumées dans le Tableau 2.

Métrique	Score
Précision	0.93
Rappel	0.88
F1-Score	0.90
Spécificité	0.88
mAP@50-95	0.90

TABLE 2 – Performance du modèle YOLOv8m sur le StreaksYoloDataset.

Le modèle atteint une précision de 0,93 et le rappel 0,88 avec un seuil IoU fixé à 0,70. Cette performance se traduit par un F1-score de 0,90 qui indique un bon équilibre entre la capacité du modèle à détecter les traînées et à limiter les fausses détections. La spécificité de 0,88 montre également que le modèle YOLOv8m parvient à éviter un excès de fausse-positive, un aspect essentiel pour des applications astronomiques où la fiabilité des détections est pri-

mordiale. La mAP@50-95 de 0,90 confirme la stabilité du modèle sur une large plage de seuils IoU, ce qui démontre ainsi sa cohérence en termes de détection.

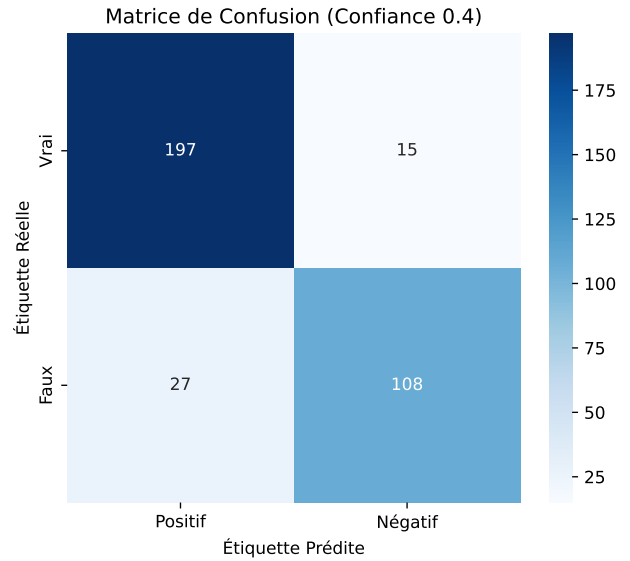


FIGURE 5 – Matrice de confusion du modèle avec un seuil de confiance de 0.4 sur le dataset de test

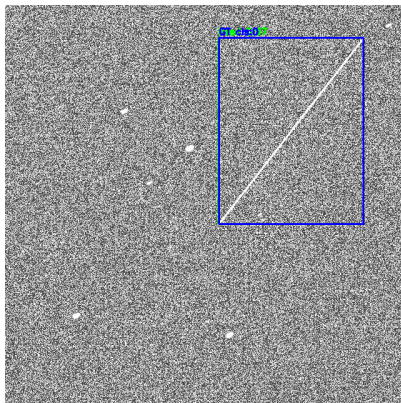
3.3 Analyse des Faux Positifs et Faux Négatifs

Une analyse détaillée des erreurs a été réalisée à l'aide d'une matrice de confusion (Figure 5) avec un seuil de confiance de 0.4 (Fig. 3), sélectionné à partir du dataset de validation pour optimiser la précision et le rappel.

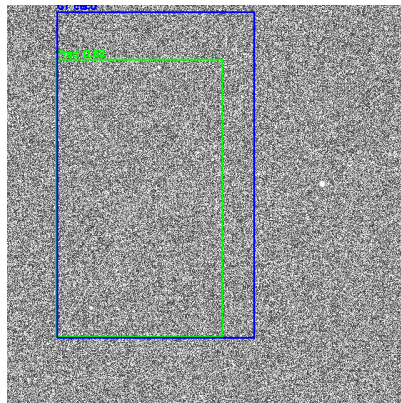
Sur les données de test, 42 sur les 333 images ne sont pas classées correctement :

- Faux Positifs (FP) : 27 faux positifs ont été détectés, réduisant légèrement la précision. Ces erreurs sont principalement dues à :
 - La confusion avec des étoiles brillantes ou des rayons cosmiques.
 - Des artefacts optiques ou du bruit du capteur.
 - Une tendance du modèle à détecter des traînées dans des zones à fort contraste.
- Faux Négatifs (FN) : Le modèle a omis 15 traînées réelles, affectant légèrement le rappel. Ces omissions sont attribuées à :
 - Des traînées de faible intensité, difficilement discernables.
 - Des occultations partielles causées par d'autres objets célestes.
 - Une variabilité en épaisseur et en longueur des traînées rendant leur détection plus complexe

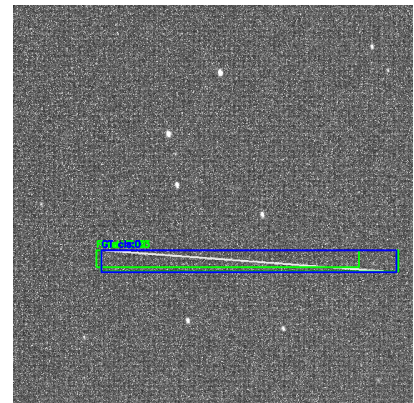
Pour mieux analyser les performances du modèle, nous présentons six cas illustrant différents scénarios de détection des traînées. Ces exemples permettent d'identifier les points forts du modèle ainsi que ses principales limitations :



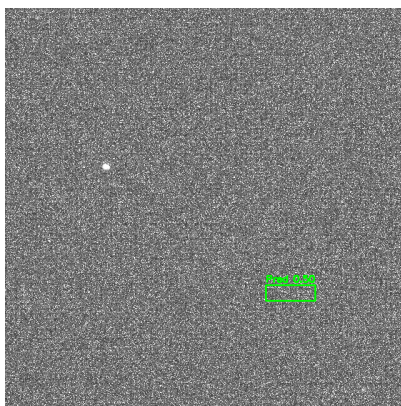
(a) Exemple de détection précise où la boîte prédite (vert) est alignée avec la vérité terrain (bleu).



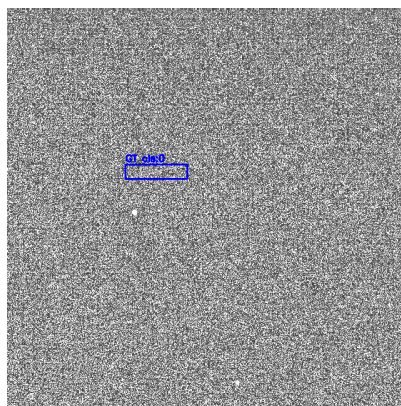
(b) Exemple de sous-détection où la boîte prédite couvre une zone inférieure à la traînée réelle.



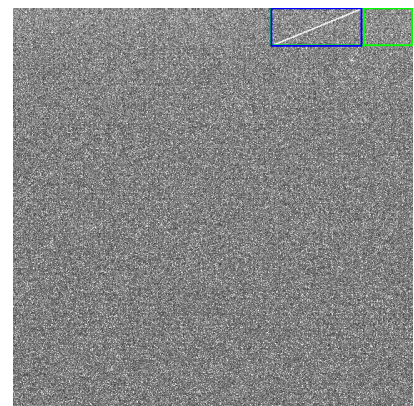
(c) Exemple de mauvais alignement entre la boîte prédite (vert) et la vérité terrain (bleu).



(d) Exemple de faux positif où le bruit de fond est mal classifié comme une traînée.



(e) Exemple de détection manquée où la traînée n'est pas reconnue par le modèle.



(f) Exemple de mauvaise classification où un objet céleste est détecté comme une traînée.

FIGURE 6 – Exemples de détection de traînées

• Cas 1 : Détection Précise et Alignement Optimal

Dans certains cas, la boîte englobante prédite correspond étroitement à la vérité terrain, illustrant la capacité du modèle à localiser précisément les traînées. Un exemple est présenté dans la Figure 6a.

• Cas 2 : Sous-Détection avec Boîte Réduite

Le modèle peut parfois sous-estimer la région de la traînée, générant une boîte plus petite que la vérité terrain (Figure 6b). Cela peut être due à des variations de luminosité ou de longueur des traînées, rendant la détection plus difficile.

• Cas 3 : Mauvais Alignement de la Détection

Dans certains cas, la boîte englobante prédite ne recouvre pas entièrement la traînée réelle (Figure 6c), ce qui entraîne une détection partielle ou imprécise.

• Cas 4 : Détection d'Artefacts de Bruit

Le modèle identifie parfois des artefacts de bruit de fond comme des traînées, générant des faux positifs (Figure 6d). Une meilleure prétraitement des images ou un ajustement

du seuil de confiance pourrait limiter ce phénomène.

• Cas 5 : Détection Manquée

Certaines traînées ne sont pas détectées, notamment lorsqu'elles présentent un faible contraste avec le fond (Figure 6e).

• Cas 6 : Mauvaise Classification d'Objets

Dans de rares cas, le modèle confond un objet céleste brillant avec une traînée, classifiant incorrectement une structure lumineuse comme un objet linéaire (Figure 6f).

3.4 Évolution des métriques selon la taille du jeu d'entraînement

Afin d'évaluer la robustesse de notre méthode de fine-tuning ainsi que de dégager des recommandations pratiques quant à l'usage optimal du cadre proposé, nous avons étudié l'impact de la taille du jeu d'entraînement sur les performances du modèle. Pour cela, nous avons divisé l'ensemble des données d'entraînement en cinq sous-ensembles égaux à l'aide d'une validation croisée (K-Fold avec $K = 5$), en maintenant un jeu de validation constant.

Le modèle YOLOv8 a été entraîné cinq fois (avec 20 epochs seulement), avec une quantité croissante de données d'entraînement : $\frac{1}{5}$, $\frac{2}{5}$, ..., jusqu'à l'intégralité des données disponibles ($\frac{5}{5}$). À chaque étape, le modèle est réinitialisé afin de garantir l'indépendance des résultats. Les performances sont ensuite évaluées sur le même jeu de validation, selon plusieurs métriques standard : précision, rappel, F1-score, spécificité, et mAP@50 :95.

L'ensemble de notre dataset contient 1 722 images. Les résultats expérimentaux montrent une nette amélioration des performances lorsque la taille du jeu d'entraînement passe de 344 images (soit 20%) à 690 images (soit 40%). Au-delà de 690 images, les courbes de performances tendent à se stabiliser, suggérant que les bénéfices d'un apport supplémentaire de données deviennent marginaux. Cela indique qu'un compromis raisonnable peut être trouvé entre le coût d'annotation et les performances du modèle.

Il est à noter que les performances pourraient encore être améliorées en adaptant le nombre d'epochs pour chaque volume de données, afin d'optimiser pleinement l'entraînement. Toutefois, nous avons conservé un nombre fixe d'epochs (20) pour toutes les expériences, afin de maintenir un cadre de comparaison cohérent.

Cette analyse permet de mieux comprendre la sensibilité du modèle à la quantité de données disponibles, et constitue une base utile pour orienter les futures campagnes d'acquisition et d'annotation.

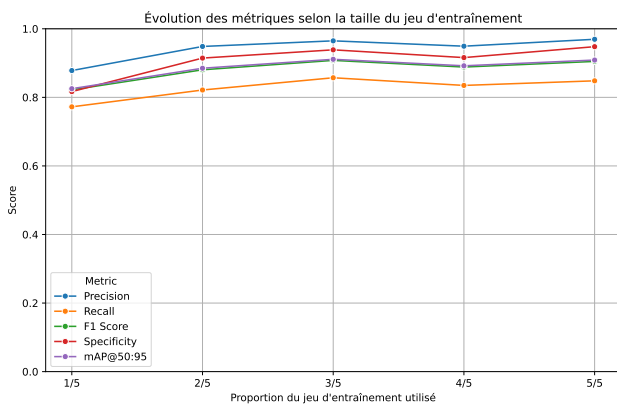


FIGURE 7 – Évolution des métriques globales (précision, rappel, F1-score, spécificité et mAP@50 :95) en fonction de la taille du jeu d'entraînement

3.5 Perspectives et Améliorations Futures

Bien que performant, le modèle peut être optimisé pour une meilleure adaptation aux contraintes astronomiques. Plusieurs axes d'amélioration sont envisageables :

- **Diversification des données** : Intégrer des traînées faibles, occultées et de morphologies variées pour améliorer la généralisation.
- **Ajustement dynamique du seuil de confiance** : Adapter dynamiquement le seuil pour réduire les faux positifs sans altérer le rappel.

- **Optimisation du post-traitement** : Améliorer la suppression non maximale (NMS) pour affiner les prédictions et limiter les détections redondantes.
- **Exploitation temporelle** : Intégrer une analyse multi-image pour renforcer la robustesse face aux artefacts transitoires.
- **Exploration et comparaison d'architectures de détection** : L'approche actuelle repose sur YOLOv8m en raison de son bon compromis entre rapidité et précision. Toutefois, d'autres architectures telles que *SSD* (Single Shot MultiBox Detector), *RetinaNet* ou *DETR* (Detection Transformer) pourraient également être adaptées à la détection de traînées astronomiques. Une comparaison systématique avec ces modèles constitue une perspective pertinente pour affiner l'évaluation. Par ailleurs, l'étude d'architectures plus récentes, comme *YOLOv11*, intégrant des mécanismes d'attention ou une fusion multi-échelle des caractéristiques, pourrait contribuer à une meilleure détection des traînées subtiles ou complexes, tout en maintenant une efficacité computationnelle.
- **Collaboration interdisciplinaire** : Renforcer les partenariats avec des astronomes pour intégrer des connaissances métier spécifiques (par exemple, les contraintes physiques des traînées) dans la conception du modèle, afin d'aligner les prédictions avec les attentes du domaine.
- **Modèles plus légers** : Entraîner un modèle compact sans recourir aux poids pré-entraînés sur le jeu de données COCO (Common Objects in Context), afin d'éviter le transfert de caractéristiques inadaptées issues de domaines visuels génériques (tels que les objets du quotidien). Cette approche vise à réduire la complexité du modèle, à éliminer les biais hors domaine et à optimiser les performances en inférence, notamment dans des environnements contraints en ressources computationnelles.

Ces améliorations renforceront la précision et l'efficacité du modèle, tout en garantissant une adaptation efficace aux contraintes opérationnelles des missions astronomiques, qu'elles soient professionnelles ou amateurs.

Accessibilité et Impact sur la Recherche Astronomique

L'inférence rapide sur des GPU de moyenne gamme facilite l'adoption de cette approche, aussi bien par les observatoires professionnels que par les projets de science participative. Cette accessibilité favorise une collaboration internationale pour la surveillance du ciel et contribue à la préservation des données astrophysiques.

L'optimisation du modèle repose sur l'utilisation de GPU comme le Tesla T4 (15 Go de VRAM) et l'Automatic Mixed Precision (AMP), permettant un traitement rapide et efficace tout en réduisant la charge mémoire. Le modèle atteint un temps d'inférence de 11,6 ms par image, rendant l'approche adaptée aux applications en temps réel de détection des traînées. Ces optimisations la rendent viable sur

des plateformes accessibles, assurant une grande réactivité pour la surveillance spatiale.

Pour le finetuning, il est possible d'utiliser Google Colab, offrant un environnement cloud avec GPU pour les utilisateurs disposant de ressources limitées, et facilitant ainsi l'accès à la technologie de détection.

Cette approche améliore l'accessibilité à la surveillance spatiale et contribue à la préservation des données astrophysiques.

En automatisant la détection et la filtration des traînées, notre modèle réduit les interférences artificielles, permettant aux chercheurs d'optimiser leur temps d'analyse et d'améliorer la qualité des observations. Cette approche s'inscrit ainsi dans une démarche de veille spatiale, essentielle pour la protection des infrastructures spatiales et la recherche astronomique.

3.6 Perspectives et Défis Ouverts

Une détection multi-classes en temps réel permettra-t-elle une surveillance totalement autonome du ciel? Comment la fusion de capteurs et le traitement avancé des formes pourraient-ils améliorer la détection des traînées faibles ou courbes? Ces questions illustrent le potentiel d'innovation de cette approche, tant pour la protection des satellites que pour l'exploration du cosmos.

En combinant apprentissage profond et amélioration des méthodes d'observation, ce pipeline démontre que la détection des traînées peut être rapide, précise et accessible, ouvrant ainsi la voie à une surveillance plus agile et efficace du ciel nocturne.

4 Conclusion

Cette étude a exploré la détection automatique des traînées transitoires dans un ciel de plus en plus encombré. En combinant le StreaksYoloDataset et l'architecture YOLOv8m, nous avons développé une approche rapide et précise, adaptée aux contraintes du temps réel.

Nos résultats montrent que le modèle offre un bon équilibre entre précision et rappel, mais montrent également des marges d'amélioration, notamment dans la détection des traînées faibles et la réduction des faux positifs. Des améliorations, telles que l'optimisation du post-traitement, l'intégration d'informations temporelles et l'exploitation de nouvelles architectures pourraient renforcer la robustesse du modèle.

Ces travaux ouvrent ainsi la voie à des méthodes plus efficaces pour la surveillance du ciel, contribuant à la recherche astronomique et à la *Veille Spatiale*.

Références

- [1] E. Bertin and S. Arnouts, "SExtractor : Software for source extraction," *Astronomy and Astrophysics Supplement Series*, vol. 117, p. 393–404, June 1996.
- [2] P. Rautiainen and A. M. Mel'nik, "N-body simulations in reconstruction of the kinematics of young stars in the galaxy," *Astronomy and Astrophysics*, vol. 519, p. A70, Sept. 2010.
- [3] G. Nir, B. Zackay, and E. O. Ofek, "Optimal and efficient streak detection in astronomical images," *arXiv preprint arXiv :1806.04204*, 2018.
- [4] C. Hollitt and M. Johnston-Hollitt, "Feature detection in radio astronomy using the circle hough transform," *Publications of the Astronomical Society of Australia*, vol. 29, no. 3, p. 309–317, 2012.
- [5] D. e. a. Kim, "Astride : Automated streak detection for astronomical images." Astrophysics Source Code Library, 2016.
- [6] B. Zackay, E. O. Ofek, and A. Gal-Yam, "Proper image subtraction—optimal transient detection, photometry, and hypothesis testing," *The Astrophysical Journal*, vol. 830, p. 27, Oct. 2016.
- [7] A. e. a. Varela, "Application of convolutional neural networks for streak detection in wide-field images," in *Proceedings of the Asteroid, Comets, and Meteors Conference*, p. E89, 2019.
- [8] M. e. a. Pöntinen, "Detection of solar system objects using streakdet in euclid simulated images," *Astronomy & Astrophysics*, vol. XXX, p. XXX, 2020.
- [9] G. Jocher, J. Qiu, and A. Chaurasia, "Ultralytics YOLO," Jan. 2023.
- [10] O. Parisot, "Streaksyolodataset : Labeled raw astronomical images for streaks detection," *Zenodo*, 2023.
- [11] P. Skalski, "make-sense." Online Documentation, 2019.
- [12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.
- [13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, *SSD : Single Shot MultiBox Detector*, p. 21–37. Springer International Publishing, 2016.
- [14] Ultralytics, "Yolov8 : Real-time object detection." Online Documentation, 2023.
- [15] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [16] R. Padilla, S. Netto, and E. Da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," in *2020 International Conference on Pattern Recognition (ICPR)*, pp. 2911–2918, 2020.
- [17] S. Kashyap, "Mean average precision (map) in object detection." Learn OpenCV Blog, 2018.
- [18] B. J. Tatiana Merkulova, "Evaluation framework for image segmentation algorithms," April, 2025.