Défauts ferroviaires : vers une détection visuelle embarquée

Saša Radosavljevic, Kevin Hoarau, Sergio Alberto Rodriguez Florez, Abdelhafid El Ouardi, Alain Rivero

Défauts ferroviaires : vers une détection visuelle embarquée

Sasa RADOSAVLJEVIC [1,2], Kevin HOARAU [1], Sergio RODRÍGUEZ FLÓREZ [1],
Abdelhafid EL OUARDI [1], Alain RIVERO [2]

[1] Université Paris-Saclay, ENS Paris-Saclay, CNRS, SATIE, 91190, Gif-sur-Yvette, France
[2] SNCF Réseau

sasa.radosavljevic@ens-paris-saclay.fr

Résumé

Les infrastructures ferroviaires nécessitent une surveillance continue pour prévenir les défauts pouvant affecter la sécurité et l'efficacité du réseau. Cet article propose une évaluation expérimentale d'un système de détection de défauts visuels sur rails, basé sur le détecteur YOLOv8 et déployé sur des architectures CPU-GPU embarquées. L'étude analyse l'atteinte des objectifs de détection sous contraintes temps-réel et évalue la capacité du système embarqué à maintenir une précision de détection satisfaisante tout en respectant les exigences industrielles en matière de traitement embarqué. Les résultats obtenus avec des données issues d'acquisitions réelles montrent que l'approche retenue représente un compromis viable entre précision de détection, temps de traitement et contraintes matérielles.

Mots-clés

Détection de défauts visuels sur rails YOLOv8, Systèmes embarqués, Traitement temps-réel.

Abstract

Railway infrastructures require continuous monitoring to prevent defects that could impact the safety and efficiency of the network. This paper presents the detection of visual rail defects, an overview of the state of the art, and deployment results of a method based on the YoloV8 detector on embedded CPU-GPU architectures. Evaluations with real-world data show that, thanks to an algorithmic complexity study, the proposed system represents a compromise between defect detection accuracy and computation time.

Keywords

Railway visual defect detection, YOLOv8, Embedded systems, Real-time processing.

1 Introduction

Le secteur ferroviaire est confronté à des problèmes d'une grande complexité technique. L'expansion technologique de ce secteur (trains autonomes), la variété des métiers abordés, les contraintes environnementales et sociétales contribuent à accroître cette complexité. Face à ces enjeux, les entreprises ferroviaires se tournent de plus en plus vers des outils performants, frugaux en énergie et embarquables dans les trains commerciaux, avec pour objectif d'apporter au client final une meilleure qualité de service tout en optimisant les coûts. Ces dernières années, les systèmes développés ont cherché à répondre à cette complexité en combinant différentes approches, dont l'Intelligence Artificielle (IA) et l'information multimodale. La transformation des modes de transport, l'anticipation des besoins des clients, et la nécessité de prévenir les incidents, font que le secteur ferroviaire est naturellement concerné par les avancées de cette technologie. Cette nouvelle problématique de recherche rassemble de nombreuses équipes de chercheurs mutualisant leurs savoir-faire et générant des solutions alternatives, permettant de passer de la maintenance corrective à la maintenance prédictive et de sécuriser le trafic.
Les réseaux de neurones artificiels (RNA) et en particulier les réseaux convolutifs profonds (CNN) ont trouvé de nombreuses applications dans la maintenance des infrastructures et du matériel roulant. Dans ces domaines techniques, bien que les progrès réalisés par l'IA soient exponentiels, nous sommes encore loin d'exploiter son plein potentiel [1]. Son succès et sa diffusion dans notre domaine dépendent de nombreux paramètres : l'acceptation de ces techniques par les principaux acteurs métiers, et la mise à disposition de données annotées et de leurs qualité [2]. Cependant, et malgré de multiples avantages, l'utilisation des RNA ou des CNN souffre de nombreux défauts, notamment de la nécessité d'une base de données d'apprentissage cohérente et de la capacité à les annoter [3]. Cet article explore une approche de détection embarquée et temps réel des défauts ferroviaires à l'aide de réseaux de neurones optimisés. La section 2 donne un bref aperçu de l'état de l'art et de l'avancement dans la détection visuelle des défauts ferroviaires. La section 3 décrit le système embarqué proposé, tandis que la section 4 détaille la méthodologie d'évaluation de la détection de défauts par vision. La section 5 présente les résultats expérimentaux, analysant la précision de détection et les performances de temps de traitement en inférence. Enfin, la section 6 présente une discussion sur les perspectives d'amélioration.

2 Aperçu de l'état de l'art

Les recherches dans le traitement d'image ont été bousculées par l'émergence du machine learning et des avan-

cées dans les réseaux de neurones. La vision par ordinateur a significativement amélioré la détection automatique des défauts ferroviaires. Les caméras de haute résolution associées à des algorithmes d'intelligence artificielle (IA) permettent la détection d'une diversité de défauts tels que les fissures, les ondulations et l'usure. Les travaux de Kou et al. (2022) ont encouragé l'utilisation de réseaux de neurones convolutifs (CNN) pour la détection de défauts de surface des rails pour améliorer la détectabilité et les temps de détection [4]. De la même manière, les contributions de Lei et Jia (2019) ont démontré la possibilité d'utiliser des machines à vecteurs de support (SVM) et l'analyse en composantes principales (PCA) pour le traitement des ensembles de données visuelles pour la caractérisation des défauts [5].

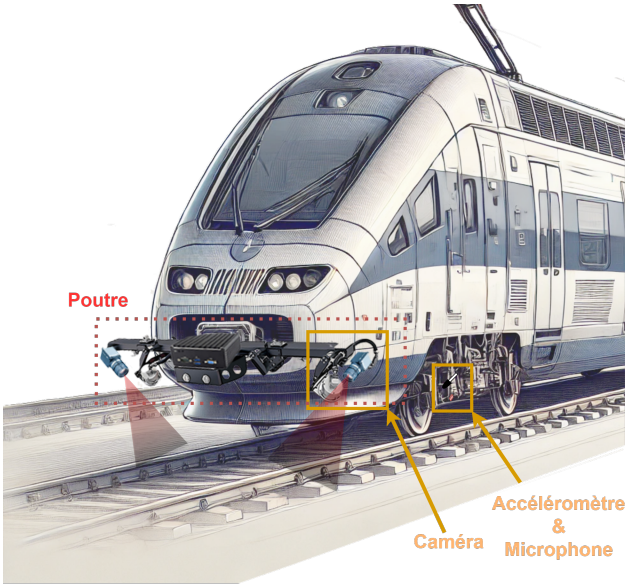

FIGURE 1 – Croquis du système embarqué train

Des études plus récentes, telles que la revue de Kumar et Harsha (2024) [6], ont approfondi l'utilisation des réseaux de neurones profonds, mettant en avant leur capacité à traiter des défauts complexes avec une grande précision. Par ailleurs, l'utilisation de systèmes d'imagerie embarqués sur drones a été explorée par Xiong et al. (2023) [7], soulignant leur potentiel pour accéder à des zones difficiles d'accès et garantir une qualité de données constante. Malgré ces avancées, les techniques de vision par ordinateur restent fortement influencées par des conditions externes telles que l'éclairage et les conditions météorologiques [8, 2]. Plus récemment encore, les architectures basées sur les transformeurs ont également été étudiées pour la détection des défauts ferroviaires. RailFormer, un système innovant basé sur les réseaux du type Transformer, a démontré des performances supérieures pour la détection des défauts en surface des rails en combinant des mécanismes d'attention Criss-Cross pour une extraction plus efficace des features [9]. De même, RailTrack-DaViT, une approche basée sur les transformeurs visuels, a surpassé les réseaux CNN traditionnels comme ResNet et EfficientNet sur plusieurs jeux d'entraînement, atteignant une précision inégalée pour l'inspection automatique des infrastructures ferroviaires [10].

Les techniques traditionnelles de contrôle non destructif (CND), bien que très précises, manquent de scalabilité. Les systèmes de vision par ordinateur, quant à eux, bien qu'efficaces, sont sensibles aux variations de l'environnement et aux contraintes liées à l'efficacité énergétique, au stockage des données et à l'encombrement des équipements. Ces limitations soulignent la nécessité d'une approche plus globale et adaptable pour la détection des défauts ferroviaires. Dans ce contexte, l'objectif de cet article est de démontrer la faisabilité du déploiement de YOLOv8n sur des architectures embarquées en conditions réelles. Pour cela, nous posons deux hypothèses de travail : (i) l'émulation d'un flux d'images à une vitesse équivalente à 160 km/h permet de simuler fidèlement un environnement embarqué ; (ii) la plateforme Jetson Orin Nano est capable de soutenir une fréquence de traitement suffisante ($\geq$ 30 FPS) tout en maintenant des performances de détection satisfaisantes. Bien que l'application visée puisse être assimilée à de la classification binaire (présence de défaut ou non), l'utilisation d'un modèle de détection d'objets se justifie par la présence de plusieurs éléments sur l'image d'une part, et la connaissance de la position de l'objet pour la corrélation à d'autres modalités d'autre part.

3 Modèle du système proposé

Afin de soulager les contraintes d'inspections sans ajouter de nouveaux engins dédiés à la surveillance de l'infrastructure, nous proposons l'intégration d'un système multimodal sur une poutre fixée à l'avant ou à l'arrière des trains (Fig. 1). De cette manière, l'analyse à la volée doit être réalisée avec la contrainte de la vitesse maximale du train.

3.1 Capteurs embarqués

Le système repose sur une combinaison de caméras linéaires, d'accéléromètres et de microphones, permettant une surveillance multimodale adaptée aux contraintes ferroviaires. Les caméras linéaires exploitent le déplacement rectiligne du train pour capturer des images haute vitesse avec une grande précision. L'acquisition est synchronisée avec l'odométrie du train, garantissant que chaque déclenchement d'acquisition correspond précisément à une avance mesurée du train. Les caméras sont placées de manière à observer l'intégralité des rails, que ce soit les éléments de l'âme du rail ou de son champignon (voir Fig. 2).

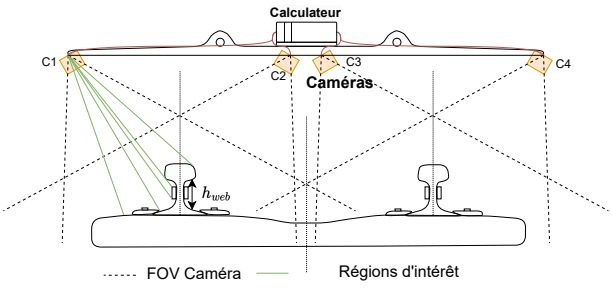

FIGURE 2 – Placement des caméras linéaires.

L'ajout des capteurs d'accélération et sonores permet d'élargir la détection aux défauts non visuels, en complétant l'analyse des caméras par des informations physiques supplémentaires sur l'interaction roue-rail. En effet, celle-ci ne peut avoir de sensation quant à ce qu'il se passe au niveau de l'interaction train/rail. Un accéléromètre est placé sur l'axe de la roue afin de mesurer la réponse dynamique du contact rail-roue lors du passage sur des défauts de surface, des anomalies structurelles ou des jonctions de rails. Un microphone complète l'analyse en apportant une redondance avec les mesures de l'accéléromètre et des caméras. Il permet également la détection de nouveaux défauts, tels que le grippage de frein ou des anomalies acoustiques au niveau des bogies. Toutefois, la combinaison de ces capteurs présentée dans cette section et les nouveaux défauts dont elle permet la détection ne sont pas explorés dans cet article, qui se limitera à l'exploitation des données visuelles.

3.2 Description des défauts

L'inspection automatisée des rails vise à identifier différents types de défauts (Fig. 3) pouvant impacter la sécurité et la maintenance de l'infrastructure du réseau ferroviaire. On peut les classer en plusieurs catégories selon leur zone d'apparition et leur moyen de détection. Outre les ruptures, la Table 1 regroupe les types de défauts et leurs descriptions selon deux catégories principales, les défauts de surface représentant les défauts d'usure du rail et les défauts structurels représentant les défauts des éléments constituant les rails. L'ajout des capteurs d'accélérométrie et de son permet d'identifier deux catégories supplémentaires difficilement détectables par les caméras, à savoir l'instabilité du ballast et les anomalies autour des bogies. A cela s'ajoute la détection d'éléments contextuels pouvant apporter des informations supplémentaires quant à la présence d'un défaut. Par exemple, la présence de cés de serrage enlève la nécessité de présence de tous les boulons. La présence d'une éclisse affirme un joint de rail à la place d'une rupture (si le modèle s'est trompé).

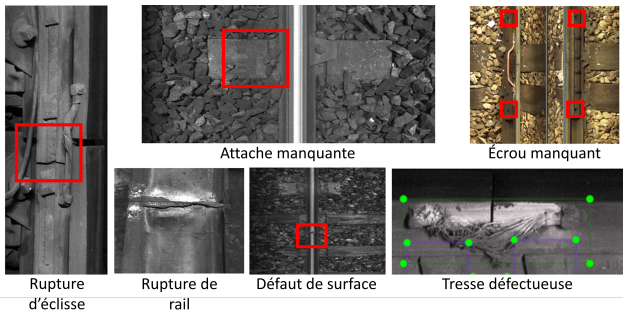

FIGURE 3 – Exemples de défauts.

3.3 Description de la chaîne de traitement

Le prototypage du système repose sur une architecture ROS2 (Robot Operating System 2) [13], permettant une communication efficace entre les différentes composantes logicielles et une gestion optimisée des flux de données en temps réel. Son paradigme de communication publish/subscribe (Fig. 4) assure une gestion efficace des flux de données provenant de capteurs hétérogènes, en maintenant une faible latence. Les données issues de la caméra sont simulées à l'aide d'un rosbag permettant de cadencer un flux de données.

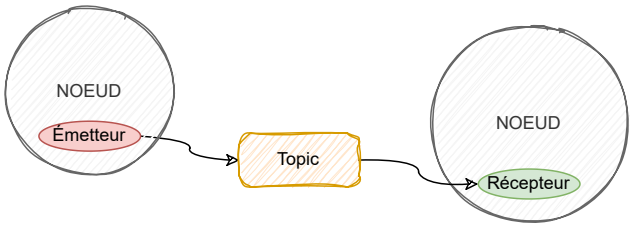

FIGURE 4 – Communication entre noeuds ROS2.

Dans notre cas, il n'y aura qu'un type de capteur (noeud caméra), un algorithme de traitement (noeud Yolov8n) et un élément de stockage des résultats (noeud bag). Les données sont alors transmises sous forme de topic aux subscribers concernés. La chaîne monomodale composée d'une caméra est présentée dans Fig. 6

L'architecture de réseau de neurones convolutifs (CNN) utilisée pour la détection des défauts est YOLO (You Only Look Once [14]). Ce modèle est conçu pour détecter les objets en une seule passe en générant des zones d'intérêt autour des éléments identifiés. Il associe simultanément à chaque région une classe prédite, permettant une classification efficace en temps réel.

YOLOv5 [15] a été l'un des premiers détecteurs d'objets à intégrer la prédiction des boîtes englobantes et des étiquettes de classe dans un réseau différentiable de bout en bout. Dans cette architecture, la partie convolutive du réseau est prédéfinie et figée, tandis qu'une partie des couches entièrement connectées reste ajustable pour adapter le modèle à une résolution spécifique.

Dans cette étude, nous utilisons YOLOv8 [16], qui apporte plusieurs améliorations significatives par rapport à YOLOv5. Les versions plus récentes ne sont pas étudiées car leurs améliorations n'impactent que peu les datasets à faible nombre de classes. Tout d'abord, l'optimisation des performances se traduit par une meilleure gestion de la quantification des poids (FP16, INT8), ce qui permet une exécution plus rapide sur des plateformes embarquées comme les Jetson Orin Nano/Xavier, tout en réduisant la consommation des ressources matérielles. Ensuite, l'architecture a été repensée avec un backbone amélioré, basé sur CSPDarknet, et l'intégration de mécanismes d'attention qui affinent la détection en concentrant les ressources de calcul sur les zones les plus pertinentes de l'image.

4 Méthodologie d'évaluation

Afin de simuler les contraintes réelles sur le système de calcul, un montage Hardware in the Loop (HiL, fig .5) a été proposé pour à la fois émuler l'acquisition des images et leur transfert vers l'élément de calcul, et avoir un environnement communicant permettant d'évaluer les temps de

TABLE 1 – Classification des défauts ferroviaires et leurs caractéristiques selon documents normatifs [11, 12]

Catégorie	Type de défaut	Description	Taille (mm)
Défauts de surface	Fissures	Microfissures ou fissures sur le rail	2 - 50
	Usure excessive	Érosion anormale du champignon du rail	> 10
	Corrugation	Ondulations périodiques sur le rail	10 - 1000
Défauts structurels	Rupture de rail	Fracture complète ou partielle du rail	50 - 12 000
	Boulons manquants	Fixations absentes	-
	Rupture d'éclisse	Défaillance aux joints de rails	50 - 150
Défauts contextuels*	Objets sur la voie	Corps étrangers pouvant perturber le trafic	Variable
	Niveau de ballast	Déformation sous les traverses	-
Interaction rail-roue*	Défauts de contact	Anomalies dans l'interface rail-roue	-
	Vibrations excessives	Dégradations mécaniques révélées par l'accéléromètre	-
	Bruits anormaux	Indicateur d'un problème de freinage ou bogie	-

*Non présenté dans l'article ou pas encore implémenté.

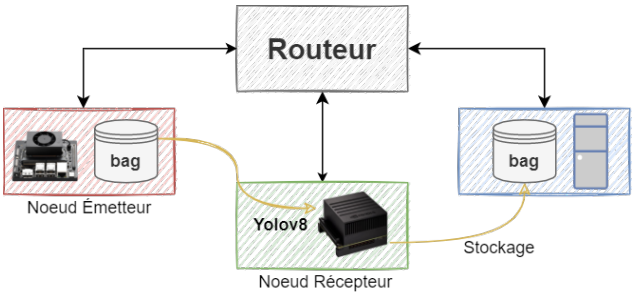

FIGURE 5 – Schéma du montage.

transfert et le comportement entre les différents éléments.

4.1 Entraînement du modèle

Dans une première étape, le modèle a été entraîné à partir d'un jeu de données issu d'images provenant de l'engin de surveillance SIM (Switch Inspection and Measurement) opéré par Eurailscoot. Ces données ont été manuellement annotées par un expert ferroviaire afin d'assurer une classification précise des défauts détectés. Cependant, la subjectivité de l'expert peut entraîner des variations dans la caractérisation des défauts, notamment pour ceux liés aux irrégularités de surface. Les données sont constituées de deux types de vues distinctes : une vue de dessus (RC - Rail centric), composée d'images en niveaux de gris de 1504×1500 pixels, et une vue latérale (SV - Side View), composée d'images en couleur RGB de 768×1508 pixels.
L'ensemble de données contextuelles [C] (Table 3) contient 1327 images RC et 9585 images SV pour l'entraînement, 167 images RC et 1296 images SV pour la validation, ainsi que 173 images RC et 1304 images SV pour le test. De son côté, l'ensemble des données des défauts [D] (Table 2) contient 3456 images RC et 5779 images SV pour l'entraînement, 282 images TV et 414 images SV pour la validation, ainsi que 263 images RC et 409 images SV pour le test.
Les jeux de données présentent un déséquilibre des données qui est dû à la fréquence d'apparition des éléments recherchés sur le réseau SNCF. Ces images étant annotées à

TABLE 2 – Jeu de données des défauts [**D**]

Classe	Composition					
	Train	***%***	***Val***	***%***	***Test***	***%***
Attache	8019	64.3	593	67.3	590	68.1
Surface	2949	23.6	195	22.1	183	21.1
Boulon	1502	12.1	93	10.6	93	10.8
Total	12470	87.7	881	6.2	866	6.1

Train. images : 9235, Val. images : 696, Test images : 672

TABLE 3 – Jeu de données des éléments de contexte [**C**]

Classe	Composition					
	Train	***%***	***Val***	***%***	***Test***	***%***
Tresse	3786	15.9	522	16.5	524	15.7
Cés de serrage	578	2.4	56	1.8	60	1.8
Éclisse	6045	25.5	795	25.2	872	26.2
Joint	6661	28.1	868	27.6	935	28.1
Soudure	2818	11.9	374	11.9	373	11.2
Marque	3828	16.2	533	17	563	17
Total	23716	78.5	3148	10.5	3327	11

Train. images : 10912, Val. images : 1463, Test images : 1477

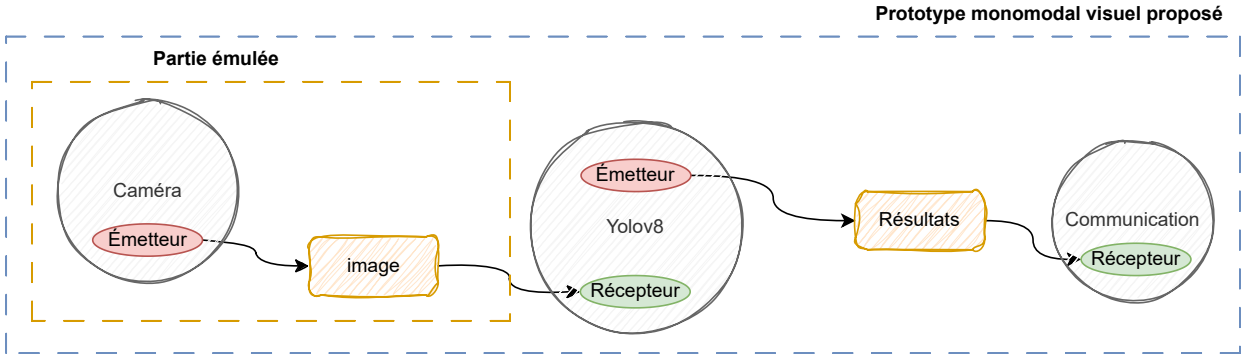

FIGURE 6 – Chaîne de traitement de l'architecture logicielle du prototype monomodal.

partir de plusieurs tournées, les objets ont une quantité proportionnelle aux défauts présents sur les voies, expliquant l'absence de ruptures dans les jeux de données. Afin de quantifier les performances du modèle de détection, nous définissons plusieurs métriques basées sur les notions de Vrai/Faux Positif/Négatif (Table 4).

TABLE 4 – Notions utiles pour la mesure d'efficacité d'un modèle

	Appartient à la classe	**N'appartient pas à la classe**
Attribué à la classe	Vrai Positif (VP)	Faux Positif (FP)
Non attribué à la classe	Faux Négatif (FN)	Vrai Négatif (VN)

Avec ces éléments, nous pouvons définir deux grandeurs qui seront par la suite analysées pour déterminer le point de fonctionnement d'un modèle pour une classe donnée.

$$\text{Précision} = \frac{VP}{VP + FP} \tag{1}$$

La Précision représente la capacité du modèle à minimiser les erreurs d'attribution d'un objet à une classe donnée.

$$\text{Rappel} = \frac{VP}{VP + FN} \tag{2}$$

Le Rappel évalue la capacité du modèle à ne pas omettre des objets appartenant à une classe donnée.

$$AP = \int_0^1 P(r)dr, \tag{3}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{4}$$

La précision moyenne (4) résume la courbe Précision-Rappel (PR Curve) à un recouvrement de boîtes de détections donné (IoU, Intersection over Union = 0,5 veut dire que l'aire de l'intersection de la détection vaut la moitié de l'union de celle-ci et la vérité terrain). Elle permet d'indiquer la précision du modèle selon la rigueur appliquée à la localisation des objets détectés. Une valeur élevée indique que le modèle repère la majorité des objets en faisant peu d'erreurs de classification. Dans notre cas, un IoU de 0,5 est suffisant pour avoir une bonne idée de l'emplacement de l'objet sans détériorer le taux de détection.

4.2 Évaluation de la chaîne sur cible embarquée

L'évaluation du système en mode HiL a été réalisée sur une plateforme GPU embarquée de type Nvidia Jetson, la Jetson Xavier (Version 32 Go, GPU Nvidia Volta à 512 cœurs dont 64 cœurs Tensor, cadencé à 1377 MHz) en mode MAXN permettant d'utiliser le plein potentiel de la carte. Afin de simuler un environnement de production et d'évaluer les performances en conditions réelles, nous avons utilisé ROS2 et son outil "rosbag", permettant d'enregistrer et de rejouer des flux d'images à une cadence définie, imitant ainsi le déplacement du train.

La Jetson Xavier transmet les images à une fréquence définie (correspondant à la vitesse du train simulé). La Jetson Orin Nano exécute le noeud qui traite les images par réseaux de neurones. Les images sont publiées sur le Topic « /topic_image ». Les résultats des traitements sont ensuite enregistrés dans un rosbag, permettant d'analyser la latence et l'efficacité du pipeline sur les Topics « /Results_topic » et « /CnnTime_topic », respectivement le résultat de l'inférence et le tableau contenant les temps de calcul des différentes étapes de traitement.

4.3 Performances recherchées

Afin d'améliorer la réactivité de la détection des défauts et de limiter les contraintes de stockage, le passage d'un traitement "hors ligne" vers un traitement "en ligne" (à la volée) devient un objectif naturel et technologique. Pour cela, le système doit être temps-réel et répondre à une contrainte temporelle de traitement des données capteurs. L'objectif étant d'atteindre une vitesse de diagnostic confondue avec la vitesse de déplacement du train pour éviter l'accumulation de données dans le système. La vitesse de train visée est celle d'un TER pouvant circuler jusqu'à 160 km/h. La Table 1 indique une taille minimale d'un défaut visible de 2 mm imposant une définition minimale de la caméra de

TABLE 5 – Performances de détection des défauts en FP32 et FP16

Classe	FP32		FP16	
	mAP@0.5	**mAP@0.5-0.95**	**mAP@0.5**	**mAP@0.5-0.95**
Attache défectueuse	0.975	0.667	0.972	0.661
Défaut de surface	0.715	0.4	0.723	0.341
Boulon manquant	0.971	0.449	0.952	0.439
Toutes classes	0.887	0.487	0.882	0.480

TABLE 6 – Performances de détection du contexte en FP32 et FP16

Classe	FP32		FP16	
	mAP@0.5	**mAP@0.5-0.95**	**mAP@0.5**	**mAP@0.5-0.95**
Tresse	0.962	0.557	0.962	0.552
Cés de serrage	0.991	0.570	0.994	0.554
Éclisse	0.992	0.676	0.994	0.673
Joint	0.866	0.360	0.864	0.352
Soudure	0.953	0.501	0.950	0.499
Marque	0.937	0.528	0.938	0.523
Toutes classes	0.950	0.532	0.950	0.526

1 millimètre par pixel. Ainsi, l'objectif recherché est un système qui répond à un traitement temps-réel (30 FPS) avec des images de résolutions 1504x1504 pixels et une vitesse maximale de 160 km/h (éq. 44 445 pixels par seconde). Dans le même objectif, nous cherchons à atteindre une précision de détection de l'ordre de 95% avant compromis. Afin de réduire la charge de calcul tout en maintenant une précision acceptable, une sous-résolution des images à 2 mm/pixel peut être envisagée. Ce compromis permet de réduire les besoins en puissance de calcul sans altérer significativement la qualité de la détection. La pertinence de cette sous-résolution sera justifiée dans la section 5, où nous analyserons son impact sur les performances de classification et de reconnaissance des défauts.

5 Résultats et analyse

L'unité de calcul a été configurée de manière à exploiter au maximum la fréquence d'horloge et les capacités de calcul du système. En complément des temps de traitement du réseau de neurones, nous mesurons également le temps nécessaire à la désérialisation de l'image lors de la réception du message encodé par le noeud d'inférence. Cette approche permet ainsi d'évaluer le temps de traitement global de l'ensemble de la chaîne, en prenant en compte l'intégralité des étapes, de l'acquisition à l'inférence.

5.1 Performances de détection

La perte en précision relative au gain apporté en temps d'inférence pour une sous-résolution de 2,35 (Table 7) est très négligeable dans le cas des objets de contexte et est un compromis acceptable pour les éléments de défauts, notamment car la perte vient des défauts de surface que l'on vise à améliorer par l'ajout de modalités au système. De plus, on constate que le rapport de résolution et le rapport de temps d'inférence n'est pas quadratique comme on pourrait l'imaginer. Ce qui nous laisse imaginer pouvoir gagner encore un

TABLE 7 – Résultats de mAP pour différentes résolutions de pixel

Résolution (px)	**Contexte**		**Défauts**	
	640	**1504**	**640**	**1508**
mAP@0.5	0.950	0.953	0.887	0.904
mAP@0.5-0.95	0.523	0.538	0.487	0.497
Temps d'inférence (rapport)	x1	x1,8	x1	x1,7

ratio en faisant l'inférence sur plusieurs images côte à côte à condition d'avoir les capacités mémoire intégrées dans le système.

La première évaluation porte sur la capacité du modèle à détecter les objets recherchés. Un premier rosbag contenant la partie test (évaluation) du modèle est présenté au nœud d'inférence. Ce bag est décrit dans la colonne test des Tables 2 et 3. Les résultats de détection sont présentés Table 5 et 6. L'étude de la quantization des modèles permet de montrer que les pertes apportées par le passage de flottants sur 32 bits (FP32) vers 16 bits (FP16) sont très faibles [17], notamment pour les éléments de contexte. Un écart plus important est présent entre les défauts de surface et les boulons manquants, qui, combinés, ne provoquent pas une perte importante.

5.2 Résultats en temps d'inférence

Finalement, pour mesurer les capacités des architectures embarquées à traiter les images en temps réel, un rosbag contenant une tournée de l'engin SIM sur une voie parisienne composé de 4416 images avec ou sans objets à détecter est envoyé sur le noeud de traitement pour obtenir une mesure statistique des différents temps de traitement. L'analyse a été portée, d'une part, sur la mesure des temps de traitement autour de l'inférence tel que la désérialisation des messages ROS, le pré-traitement ainsi que

TABLE 8 – Résultats pour le temps de traitement sur la Jetson Xavier avec coefficients sur 32 bits et 16 bits.

Format	Désérialisation (ms)	Prétraitement (ms)	Inférence (ms)	Post-traitement (ms)	Temps total (ms)	Fréquence (FPS)
FP32						
Moyenne	1.25	4.07	9.06	3.12	17.50	57.1
Écart-type	0.23	0.75	0.06	0.10	1.14	3.56
Médiane	1.16	3.80	9.06	3.12	17.14	58.34
Minimum	1.07	3.64	9.01	1.73	15.45	64.72
Maximum	2.05	7.65	9.16	3.59	22.45	44.54
FP16						
Moyenne	1.22	4.01	5.36	3.10	13.69	73.0
Écart-type	0.21	0.05	0.76	0.10	1.12	5.48
Médiane	1.15	3.75	5.36	3.09	13.35	74.9
Minimum	1.05	3.58	5.31	2.85	12.79	78.18
Maximum	2.01	7.50	5.46	3.79	18.76	53.3

le post-traitement, et d'autre part sur l'inférence du réseau sur le GPU de la Jetson AGX Xavier. Les 1% des temps d'inférence les plus bas et les 1% les plus élevés sont exclus des statistiques afin d'éliminer les valeurs aberrantes. L'inférence est réalisée à partir d'un modèle TensorRT (engine) exporté depuis les poids du modèle PyTorch (pt). Ce dernier ayant en moyenne 50% en plus en temps d'inférence sur l'architecture Xavier, ce qui n'est pas envisageable pour un déploiement sur cible embarquée. Afin d'optimiser les performances d'inférence du modèle sur GPU embarqué, une étude de quantification a été réalisée. Cette opération consiste à convertir les poids du modèle initial, stockés sur 32 bits flottants (FP32), vers un format réduit à 16 bits flottants (FP16). L'objectif est de diminuer les besoins en mémoire et d'accélérer les calculs en bénéficiant du support matériel des GPUs modernes pour la précision FP16. Les mesures réalisées comparent les temps de traitement moyens des étapes de la chaîne d'inférence (désérialisation, prétraitement, inférence, post-traitement) entre les deux formats. Les résultats (voir Table 8) montrent que la quantification FP16 permet de réduire significativement le temps total de traitement par image, passant de 17,5 ms à 13,69 ms en moyenne, soit un gain d'environ 21% pour l'ensemble de la chaîne et de 40% pour l'inférence. Cela se traduit par une augmentation de la fréquence d'images traitées de 57 FPS à 73 FPS, ouvrant la possibilité de traiter deux flux caméra simultanément avec une seule unité de calcul. Ce gain est obtenu sans dégradation notable de la qualité de détection.

6 Perspectives et conclusion

Les résultats des expériences réalisées montrent la possibilité de l'utilisation du modèle YOLOv8 nano pour la détection de défauts ferroviaires en temps réel sur des architectures embarquées. Les compromis et les optimisations réalisés permettent d'obtenir des résultats en termes de détection et de temps d'inférence viables pour le déploiement de tel modèle sur un système embarqué.

L'un des premiers axes d'amélioration concerne la robustesse du modèle face aux conditions réelles d'exploitation. Bien que les performances soient évaluées en laboratoire avec des données issues de relevés réels, l'impact des variations environnementales, telles que les changements de luminosité, les conditions météorologiques et les vibrations du train, reste à explorer en profondeur. Il est essentiel de valider l'efficacité du modèle dans ces conditions réelles pour garantir une détection fiable sur l'ensemble du réseau ferroviaire.

Par ailleurs, l'optimisation des modèles pour l'embarqué reste un enjeu crucial. L'étude a montré que la quantification en FP16 permet une exécution plus rapide sans perte significative de précision, mais d'autres méthodes d'optimisation, comme l'usage de modèles quantifiés en INT8, le pruning de réseaux ou des accélérateurs spécialisés dans les inférences d'IA pourraient encore améliorer les performances, notamment à exécuter plusieurs flux vidéo en parallèle et amener à un système encore plus frugal et compact. Enfin, pour garantir une adoption généralisée, il serait pertinent d'évaluer la généralisation du modèle sur d'autres réseaux ferroviaires à l'international. Chaque réseau possède des spécificités techniques et environnementales qui peuvent influencer la performance des algorithmes de détection. Un réapprentissage spécifique à chaque réseau, basé sur des données issues de leurs infrastructures ferroviaires, permettrait d'adapter le modèle à diverses configurations et d'améliorer sa robustesse face à des scénarios variés.

En conclusion, cette étude a permis de démontrer l'intérêt des méthodes de détection basées sur l'IA pour la maintenance ferroviaire en temps réel. L'intégration du modèle YOLOv8 sur des architectures embarquées a prouvé sa capacité à détecter efficacement les défauts tout en respectant les contraintes de calcul inhérentes aux systèmes embarqués. Toutefois, plusieurs axes de recherche restent à explorer pour améliorer la robustesse, l'évolutivité et l'intégration du système dans une architecture globale de maintenance prédictive des infrastructures ferroviaires. Ces perspectives ouvrent la voie à une nouvelle génération de solutions intelligentes pour la surveillance et l'entretien des voies ferrées, contribuant ainsi à améliorer la sécurité et l'efficacité du réseau ferroviaire.

Références

[1] Wassamon Phusakulkajorn, Alfredo Núñez, Hongrui Wang, Ali Jamshidi, Arjen Zoeteman, Burchard Ripke, Rolf Dollevoet, Bart De Schutter, and Zili Li. Artificial intelligence in railway infrastructure : current research, challenges, and future opportunities. *Intelligent Transportation Infrastructure*, 2023. https://doi.org/10.1093/iti/liad016.

[2] Guoqing Jing, Xuanyang Qin, Haoyu Wang, and Chengcheng Deng. Developments, challenges, and perspectives of railway inspection robots. *Automation in Construction*, 2022. https://doi.org/10.1016/j.autcon.2022.104242.

[3] Andrei Popescu-Belis. Managing multimodal data, metadata and annotations : Challenges and solutions. *Multimodal signal processing : Theory and applications for human-computer interaction*, page 207, 2009. https://doi.org/10.1016/B978-0-12-374825-6.00013-7.

[4] Lei Kou. A Review of Research on Detection and Evaluation of the Rail Surface Defects. *Acta Polytechnica Hungarica*, 2022. https://doi.org/10.12700/APH.19.3.2022.3.14.

[5] Lei Jia, Ming Zhu, Ryan Sherman, Jeewoong Park, Yingtao Jiang, and Hualiang Teng. Rail defect detection technology : A review of the current methods and an acoustic based method proposed for high-speed-rail. 11 2019. https://doi.org/10.1177/0361198105191600110.

[6] Ankit Kumar and S. P. Harsha. A systematic literature review of defect detection in railways using machine vision-based inspection methods. *International Journal of Transportation Science and Technology*, 2024. https://doi.org/10.1016/j.ijtst.2024.06.006.

[7] Longhui Xiong, Guoqing Jing, Jingru Wang, Xiubo Liu, and Yuhua Zhang. Detection of Rail Defects Using NDT Methods. *Sensors*, 2023. https://doi.org/10.3390/s23104627.

[8] Milica Mićić, Ljiljana Brajović, Luka Lazarević, and Zdenka Popović. Inspection of RCF rail defects – Review of NDT methods. *Mechanical Systems and Signal Processing*, 2023. https://doi.org/10.1016/j.ymssp.2022.109568.

[9] Feng Guo, Jian Liu, Yu Qian, and Quanyi Xie. Rail surface defect detection using a transformer-based network. *Journal of Industrial Information Integration*, 2024. https://doi.org/10.1016/j.jii.2024.100584.

[10] Aniwat Phaphuangwittayakul, Napat Harnpornchai, Fangli Ying, and Jinming Zhang. RailTrack-DaViT : A Vision Transformer-Based Approach for Automated Railway Track Defect Detection. *Journal of Imaging*, 2024. https://doi.org/10.3390/jimaging10080192.

[11] Federal Railroad Administration. Manual of rail defects. *FRA Technical Documents*, 2015.

[12] International Union of Railways. Rail defect manual - uic7-2500-1. 2007.

[13] Steve Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot Operating System 2 : Design, Architecture, and Uses In The Wild. *Science Robotics*, 2022. https://doi.org/10.48550/arXiv.2211.07752.

[14] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once : Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. https://doi.org/10.48550/arXiv.1506.02640.

[15] Glenn Jocher. Ultralytics yolov5, 2020. https://github.com/ultralytics/yolov5.

[16] Q. Wang, W. Feng, H. Zhao, B. Liu, and S. Lyu. Yolov8 : A novel object detection algorithm with enhanced performance and robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, 2024. https://doi.org/10.1109/adics58448.2024.10533619.

[17] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. Pruning and Quantization for Deep Neural Network Acceleration : A Survey, June 2021. https://doi.org/10.48550/arXiv.2101.09671.