

DÉPARTEMENT DE GÉOMATIQUE APPLIQUÉE
FACULTÉ DES LETTRES ET SCIENCES HUMAINES
UNIVERSITÉ DE SHERBROOKE

Développement d'un outil de détection des ponceaux par intelligence artificielle

*Essai présenté pour l'obtention du grade de Maître en sciences (M. Sc.) en
géomatique appliquée et télédétection.*

Ogma Marcellin, DANKOMA

Sherbrooke

Mai 2023

Remerciements

Je souhaite manifester mon profond respect et ma sincère gratitude à tous ceux qui ont contribué de manière significative à la réalisation de mon projet d'étude.

Premièrement, un grand merci au professeur Yacine Bouroubi pour avoir accepté la tâche importante d'évaluer ce travail. Votre implication est grandement valorisée.

Mon profond respect et mon infinie reconnaissance vont au professeur Mickaël Germain. Plus qu'un directeur d'essai, il a été un véritable modèle d'humanité pour moi. Sa disponibilité inébranlable et son écoute attentive ont été une source d'inspiration continue.

Je tiens à témoigner ma gratitude au professeur Samuel Foucher pour avoir accepté d'être le co-encadrant de cet essai. Sa précieuse contribution a été essentielle et j'en suis véritablement reconnaissant.

Je tiens à exprimer une gratitude particulière à monsieur Guillaume Simard. Non seulement vous avez été mon superviseur de stage et encadrant de ce projet, mais vous avez également été un mentor précieux pour moi. Vous avez patiemment guidé mes premiers pas dans le monde professionnel de la géomatique, m'enseignant avec sagesse et détermination.

Une dédicace spéciale à ma mère, Madame DANKOMA Mazimahiya, pour ses prières incessantes et son soutien indéfectible. Chacune de ses prières m'a donné la force de persévérer et j'espère que ce travail sera à la mesure de ses sacrifices.

Je pense à mon père, feu DANKOMA Baku Lemou, dont l'absence est profondément ressentie en ce moment significatif. J'aurais tant souhaité qu'il soit là pour partager cette étape avec moi.

Un merci chaleureux à ma grande sœur, DANKOMA Bifédama. Sa résilience et son engagement indéfectible envers notre famille, particulièrement après le départ de notre père, méritent toute mon admiration. Son soutien constant tout au long de mon parcours universitaire a été une véritable source de force.

Je remercie mon frère, DANKOMA Saguenda, commissaire de police, pour sa présence inébranlable à mes côtés, dans les moments d'allégresse comme dans les épreuves. Ma gratitude envers lui est immense.

À Dr DJALNA Kouyadega, je tiens à exprimer toute ma reconnaissance. Depuis 2014, vous avez été bien plus qu'un beau-frère pour moi; vous avez été un second père. Merci pour les valeurs que vous nous avez inculquées. Je vous suis éternellement redevable.

À DANKOMA Kpeta, mon guide initial dans la vie. Merci d'avoir cru en moi dès mes années de lycée. Sachez que si j'écris ces mots aujourd'hui, c'est grâce à votre foi en moi.

À ma bien-aimée, Madame PIYAKI Pidenam, ta présence a illuminé mon existence et tes paroles encourageantes ont nourri mon âme. Je te suis infiniment reconnaissant pour toutes ces nuits que tu as passées éveillée, dans l'attente et la prière pour moi. Merci.

A mes frères et sœurs DANKOMA Babomta, DANKOMA Kotamba, DANKOMA Déyanne Blaise, DANKOMA Ma'ana Charles et DANKOMA Dogénsaga Grégoire. Votre soutien et votre amour, en dépit de mon absence, ont été un véritable rempart pour moi. Vous êtes toujours dans mon cœur et je vous aime.

Je ne peux passer sous silence l'importance des amis qui ont jalonné mon parcours. Tout d'abord, Ablaye SARR, notre rencontre au Canada a été plus qu'une simple coïncidence, elle a marqué le début d'une fraternité inébranlable. Tu es plus qu'un ami pour moi, tu es un frère. Ta présence a grandement enrichi mon expérience et je te suis profondément reconnaissant. Ensuite, ma toute ma reconnaissance à M. Taha Mazouz et Omar Bembello Harouna pour votre soutien de toute forme depuis que nos chemins se sont croisés.

Enfin, à toute la cohorte d'automne 2021, merci d'avoir rendu ce parcours enrichissant et mémorable. La force de notre cohésion a été un moteur essentiel tout au long de cette formation. Chacun de vous a apporté une pierre unique à l'édifice de cette expérience, et pour cela, je vous suis infiniment reconnaissant. Ainsi, cet essai est non seulement le fruit de mon travail, mais également le reflet de toutes ces rencontres et expériences partagées.

Je dédie cet essai à toute la famille DANKOMA. Votre soutien et votre amour ont été les fondations sur lesquelles j'ai construit cette réussite. Merci.

Table des matières

Liste des figures	iii
Liste des tableaux.....	v
Liste des abréviations.....	vi
1 Introduction.....	1
1.1 Contexte de l'étude	1
1.2 Problématique	2
1.3 Objectifs de l'étude.....	2
2 Cadre théorique.....	3
2.1 L'intelligence artificielle.....	3
2.2 Les réseaux de neurones artificiels : Architecture et leurs fonctionnements.....	5
2.2.1 Architecture.....	5
2.2.2 Fonctionnement des réseaux de neurones artificiels.....	7
2.3 Classification par imagerie.....	9
2.4 Détection des objets	10
2.5 L'apprentissage profond en géomatique.....	10
2.6 Autres méthodes de détection des objets en dehors de l'IA	12
2.7 Augmentation des données	13
2.8 Quelques bibliothèques d'apprentissage profond	14
3. Matériel et méthodes.....	16
3.1 Site d'étude	16
3.2 Les données de l'étude.....	17
3.2.1 Les données.....	17
3.2.2 Sorties de terrain	18
3.2.3 Les données d'entraînement et de validation.....	20

3.3 Méthodologie	20
3.4 Prétraitements des données d'entraînement.....	23
3.4.1 Augmentations des données d'entraînement et de validation.....	25
3.4.2 Données de prédiction.....	26
3.4.4 Architecture des modèles Faster R-CNN et YOLOv8.....	29
4. Résultats et discussions.....	38
4.1 Résultats.....	38
4.1.1 Entraînement du modèle	38
4.1.2 Exemples de résultats de détection des ponceaux.....	47
4.2 Discussion.....	53
4.2.1 Performances des modèles.....	53
4.2.3 Discussion des résultats de la détection des ponceaux	56
4.2.4 Limites du projet et recommandations.....	59
5. Conclusion	60
6 Références.....	61

Liste des figures

Figure 1 : Intelligence artificielle, apprentissage automatique et apprentissage profond traduit de chollet(2021).....	4
Figure 2 : Composition d'un réseau de neurones artificiels à une couche cachée.	6
Figure 3: Structure d'un réseau de neurones artificiels(perceptron).....	8
Figure 4: Transformations appliquées sur les imagettes des images d'entraînement.....	26
Figure 5: Algorithme de suppression du non maximum. Cet algorithme est tiré de (Terven et Cordova-Esparza, 2023).....	30
Figure 6: Suppression du non maximum (NMS) (Terven et Cordova-Esparza, 2023); a) objets avec plusieurs boîtes englobantes; b) objets après application de NMS.....	30
Figure 7: Diagramme de l'architecture complète de Faster RCNN (Ghoury et al., 2019)	32
Figure 8: : Comparaison de YOLOv8 aux autres modèles YOLO, image tirée de (« YOLOv8 Ultralytics: State-of-the-Art YOLO Models », 2023).	34
Figure 9: Diagramme de l'architecture complète de YOLOv8 tiré de (OpenMMLab, 2023)	35
Figure 10: Évolution du f1_score pour 100 époques sur les données de validation.....	39
Figure 11: Évolution de la perte d'entraînement pour 100 époques sur les données d'entraînement	40
Figure 12: Évolution du f1_score pour 300 époques sur les données de validation.....	41
Figure 13: Évolution de la perte d'entraînement pour 300 époques sur les données d'entraînement	42
Figure 14: Les différentes étapes de mise en œuvre d'un modèle YOLOv8 sur roboflow	45
Figure 15: Matrice de confusion issue de la validation du modèle YOLOv8	46
Figure 16: Résultats de l'entraînement du modèle YOLOv8.....	47
Figure 17: Extrait des résultats de détection des ponceaux du modèle Faster R-CNN pour 100 époques	48
Figure 18: Exemples de fausse détection du modèle Faster R-CNN pour 100 époques ..	48
Figure 19: Extrait des résultats de détection des ponceaux du modèle Faster R-CNN pour 300 époques	49
Figure 20: Extrait des résultats de détection des ponceaux du modèle yoloV8.....	50
Figure 21: Inférence sur les données de base. Détection des ponceaux sur les imagettes	51

Figure 22: Résultat de l'inférence sur les imageries	51
Figure 23: Image reconstituée à partir de l'inférence sur les imageries	52
Figure 24: Deuxième inférence sur une image teste d'une résolution de 5472 x 3648 ...	53
Figure 25: Courbe de Précision x Rappel du modèle Faster R-CNN pour les 100 et 300 époques sur l'ensemble des données d'entraînement et de validation. a) données d'entraînement (100 époques); b) données de validation (100 époques); c) données d'entraînement (300 époques); b) données de validation (300 époques).	55
Figure 26: Fausses détections des ponceaux avec le modèle YOLOv8.....	57

Liste des tableaux

Tableau 1 : Données utilisées	18
Tableau 2 : Caractéristiques des jeux de données utilisés	25
Tableau 3 : Comparaisons des bibliothèques d'apprentissage profond libres et ouvertes d'après les études faites par (Elshawi et al., 2021; Rao, 2023b; YAPICI et Topaloğlu, 2021)	28
Tableau 4 : Architectures YOLO et leurs dates de sortie	33
Tableau 5 : Comparaison entre de différents plans de google colab.	36
Tableau 6 : Métriques obtenus après l'entraînement du modèle en fonction du nombre d'époques (architecture Faster R-CNN)	38
Tableau 7 : Métriques obtenues après l'entraînement du modèle en fonction du nombre d'époques (architecture YOLOv8).	46
Tableau 8 : résumé des métriques de l'entraînement des modèles Fast R-CNN et YOLOv8	58

Liste des abréviations

CNN :	Réseau de Neurones Connectés (« Convolutional Neural Network »).
Fast R-CNN :	Réseau neuronal convolutif rapide basé sur les régions (« Fast Region-based Convolutional Neural Network »)
GPU :	Graphics Processing Unit
IA :	Intelligence Artificielle
LiDAR :	Light Detection and Ranging
MAE :	Erreur Absolue Moyenne
MAMH :	Ministère des Affaires municipales et de l'Habitation
MERN :	Ministère de l'Énergie et des Ressources Naturelles
MFFP :	Ministère des Forêts, de la Faune et des Parcs
MGA :	Maxxum Gestion d'Actifs
MNS :	Non-Maximum Suppression
MTQ :	Ministère des Transports du Québec
PIIRL :	Plan d'Intervention d'Infrastructures Routier
R-CNN :	Réseau neuronal convolutif basé sur les régions (« Region-based Convolutional Neural Network»)
ResNet :	Réseau neuronal résiduel (« residual neural network»)
RMSE :	Erreur Quadratique Moyenne
RNA :	Réseau de neuronal artificiel
SSD :	Single Shot Detector
SVM :	Machine à vecteurs de support (« Support Vector Machine »)
YOLO :	You Only Look Once

1 Introduction

1.1 Contexte de l'étude

Les ponceaux sont définis comme des structures de drainage en béton, en tôle ondulée ou en matières plastiques. Ceux-ci enjambent des routes, des voies ferrées et des autoroutes, avec une portée de trois mètres ou moins (Najafi et Bhattachar, 2011). Ils constituent un conduit placé sous une route pour transporter l'eau de surface d'un côté à l'autre. Généralement, les usagers de la route ignorent l'existence de ces infrastructures, car elles se retrouvent sous la route et elles sont plus ou moins visible pour l'usager de la route.

En effet, au Canada, le réseau routier, dont le ministère des transports du Québec (MTQ) est responsable, est constitué de ponceaux, un bien public majeur qui doit être inspecté et entretenu adéquatement et régulièrement. Comme la majorité des infrastructures, l'état des ponceaux commence à se dégrader dès leurs installations. L'intensité du trafic, les charges supérieures aux limites permises, la circulation de l'eau, la période du gel / dégel, les polluants et les sels de déglacage sont autant de facteurs qui peuvent entraîner des dommages structuraux avec le temps. Les conséquences de la détérioration des ponceaux peuvent aller du simple inconvénient pour les utilisateurs à la destruction des structures et des chaussées (MTQ, 2019).

De ce fait, l'inspection minutieuse d'un ponceau est essentielle pour assurer la qualité des infrastructures et la sécurité des usagers. Un suivi de l'état de chacun des ponceaux doit être géré efficacement.

Pour favoriser les bonnes pratiques de gestion de cet actif, le MTQ a développé une méthode de gestion des ponceaux. Les grandes étapes de cette méthode sont l'inventaire des ponceaux, la mise en place d'un programme d'inspection ainsi que d'une stratégie d'intervention (MTQ, 2019; Paquette et Association québécoise des transports, 2015).

La première étape de l'inventaire des ponceaux est leur identification. C'est une étape qui consiste à identifier tous les ponceaux qui se trouvent sur le réseau routier ou hydraulique. Cette opération peut être réalisée en examinant les plans et les cartes existants soit en

effectuant une inspection visuelle sur le terrain ou en utilisant des outils de surveillance tels que des drones.

Les ponceaux sont souvent situés dans des endroits difficiles d'accès et leur identification peut nécessiter du temps et des ressources considérables. Donc, dans ce cas-ci, l'utilisation de l'intelligence artificielle pour la détection des ponceaux peut offrir une solution prometteuse à ce défi. Les algorithmes d'apprentissage automatique peuvent être formés sur des données de lidar ou de photographies aériennes pour identifier les ponceaux

1.2 Problématique

L'un des ouvrages hydrauliques les plus souvent négligés est le ponceau. Un entretien et des inspections régulières sont nécessaires pour assurer une utilisation efficace et sécuritaire de ces structures. Cependant, les inspections de ponceaux requièrent beaucoup de temps et de ressources. Il est donc difficile de les effectuer sur une base régulière (Lessard et al., 2022; Mohammadi et al., 2023).

Au niveau des municipalités, la gestion des ponceaux est de la responsabilité des gestionnaires de la voirie. Cependant, l'inventaire complet des ponceaux n'est pas toujours disponible en raison de l'éloignement de certaines zones rurales et la non-disponibilité historique des données du MTQ pour ce type d'infrastructure (Larocque, 2020; Paradis-Lacombe, 2018). Il est donc nécessaire de développer et d'évaluer des méthodes automatisées pour localiser les positions des ponceaux (Lessard et al., 2022).

Afin de détecter et de répertorier les ponceaux isolés pour une zone d'étude, cet essai propose de développer un outil de recherche automatisé de détection des ponceaux à partir d'images de drone à très haute résolution spatiale.

1.3 Objectifs de l'étude

L'objectif principal de cette étude est de développer un outil de détection automatique des ponceaux à partir des images de drone.

Afin d'atteindre cet objectif, deux objectifs secondaires ont été retenus:

- Réaliser un inventaire et acquérir des images drones des ponceaux existants;
- Développer un modèle de détection des objets par les méthodes de réseaux de neurones convolutionnels - CNN (*Convolutional Neural Networks*).

2 Cadre théorique

2.1 L'intelligence artificielle

L'intelligence artificielle (IA) fait référence à un système ou à une machine qui imite l'intelligence humaine pour effectuer des tâches. Il peut s'améliorer de manière itérative en fonction des informations collectées (Kiyindou, 2021).

À l'époque de la révolution industrielle, les développements majeurs de l'innovation technologique ont réussi à transformer de nombreuses tâches et processus manuels qui persistaient depuis des décennies alors que les humains avaient atteint les limites de leurs capacités physiques (Dwivedi *et al.*, 2021). L'IA offre le même potentiel de transformation pour augmenter et remplacer les tâches ainsi que les activités humaines dans un large éventail d'applications industrielles, intellectuelles et sociales (Russell, 2010).

Le concept d'IA a vu le jour dans les années 1950 lorsque Alan Turing, considéré comme père de l'IA, s'est intéressé à la notion de machine consciente. En 1956, le terme d'IA est utilisé officiellement lors d'un atelier à Dartmouth par des pionniers de l'IA (Alan, 1950 ; Cooper et Van Leeuwen, 2013; Dortier, 2014).

Bien que l'IA ait démontré une efficacité pour résoudre des problèmes logiques bien définis, tels que les échecs, il s'est avéré impossible de trouver des règles explicites pour résoudre des problèmes plus complexes et flous, tels que la classification des images, la reconnaissance vocale et la traduction de langues (Chollet, 2021; Rougier, 2021).

Depuis l'arrivée de cette technologie, l'IA a connu de multiples développements. En 1980, l'informaticien japonais Fukushima (1980) a publié le premier réseau de neurones artificiels à convolution profonde démontrant de hautes performances dans la reconnaissance d'images, en particulier dans le cas d'utilisation de la reconnaissance de caractères manuscrits.

En 2006, Geoffrey Hinton et son équipe ont publié un nouveau modèle d'algorithmes d'apprentissage en profondeur qui surpasse les performances des modèles précédents et leur permet de s'entraîner plus rapidement et plus efficacement (Dong *et al.*, 2021; Hinton *et al.*, 2006).

L'IA est un domaine général qui englobe l'apprentissage automatique et l'apprentissage profond (fig.1), mais cela inclut également de nombreuses autres approches qui n'impliquent aucun apprentissage. Les premiers programmes d'échecs, par exemple, n'impliquaient que des règles codées en dur élaborées par des programmeurs et n'étaient pas qualifiés d'apprentissage automatique (Chollet, 2021).

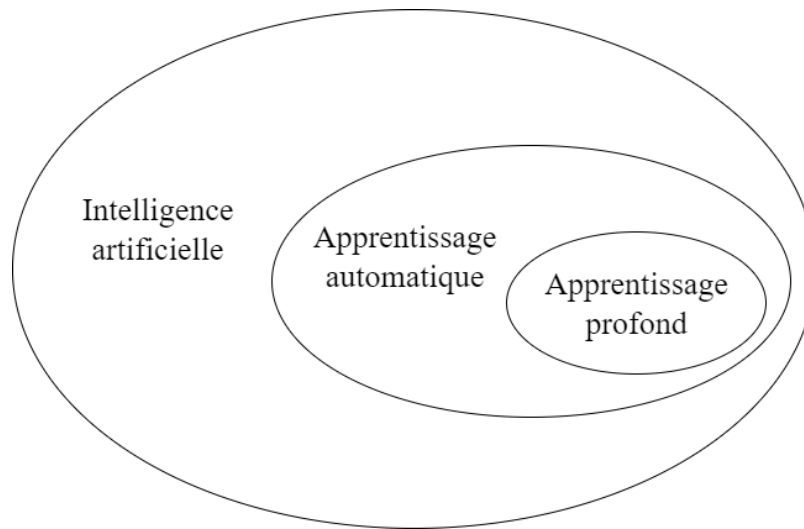


Figure 1 : Intelligence artificielle, apprentissage automatique et apprentissage profond traduit de chollet(2021)

L'apprentissage automatique décrit la capacité d'un système à apprendre, à partir de données spécifiques à un problème, pour automatiser le processus de création de modèles analytiques et de résolution de tâches associées (Janiesch *et al.*, 2021). En termes simples, il vise à automatiser la tâche de construction de modèles analytiques pour effectuer des tâches cognitives telles que la détection d'objets et la traduction en langage naturel. Ceci est possible grâce aux algorithmes qui apprennent de manière itérative, à partir des données de formation spécifique à un problème (Janiesch *et al.*, 2021; Jordan et Mitchell, 2015).

L'apprentissage profond est un sous-ensemble de l'apprentissage automatique basé sur des réseaux de neurones artificiels. Pour de nombreuses applications, les modèles d'apprentissage profond surpassent les modèles d'apprentissage automatique et les approches d'analyse de données traditionnelles (Janiesch *et al.*, 2021). Il s'agit d'un réseau de neurones avec de nombreuses couches et paramètres.

2.2 Les réseaux de neurones artificiels : Architecture et leurs fonctionnements

Au cours de la dernière décennie, l'utilisation des réseaux de neurones artificiels (RNA) s'est développée dans de nombreuses disciplines telles que l'économie, l'écologie et l'environnement, la biologie et la médecine, etc. Ils sont particulièrement adaptés à la résolution de problèmes tels que la classification, la prédiction, l'optimisation, la reconnaissance de formes et la mémoire associative (Drew et Monson, 2000; Schmitt *et al.*, 2001). Le RNA est une technique de modélisation inspirée du cerveau humain qui permet d'apprendre par l'exemple à partir de données représentatives décrivant des phénomènes physiques ou des processus de prise de décision (Sadiq *et al.*, 2019).

2.2.1 Architecture

Les réseaux de neurones artificiels sont des modèles mathématiques, de type boîte noire, formés par un ensemble d'éléments de calcul disposés dans une structure spécifique, chacun doté d'une fonction de transfert. Chaque paramètre du réseau est représenté par des valeurs appelées des poids, associés aux connexions des éléments de calcul (Dan *et al.*, 2002).

Un réseau de neurones est composé d'au minimum trois couches (fig. 2) : la couche d'entrée, la couche cachée et la couche de sortie. La couche d'entrée contient des neurones d'entrée qui envoient des informations à la couche cachée. La couche cachée, quant à elle, contient des neurones qui effectuent des calculs sur les données d'entrée et transmettent le résultat à la couche de sortie.

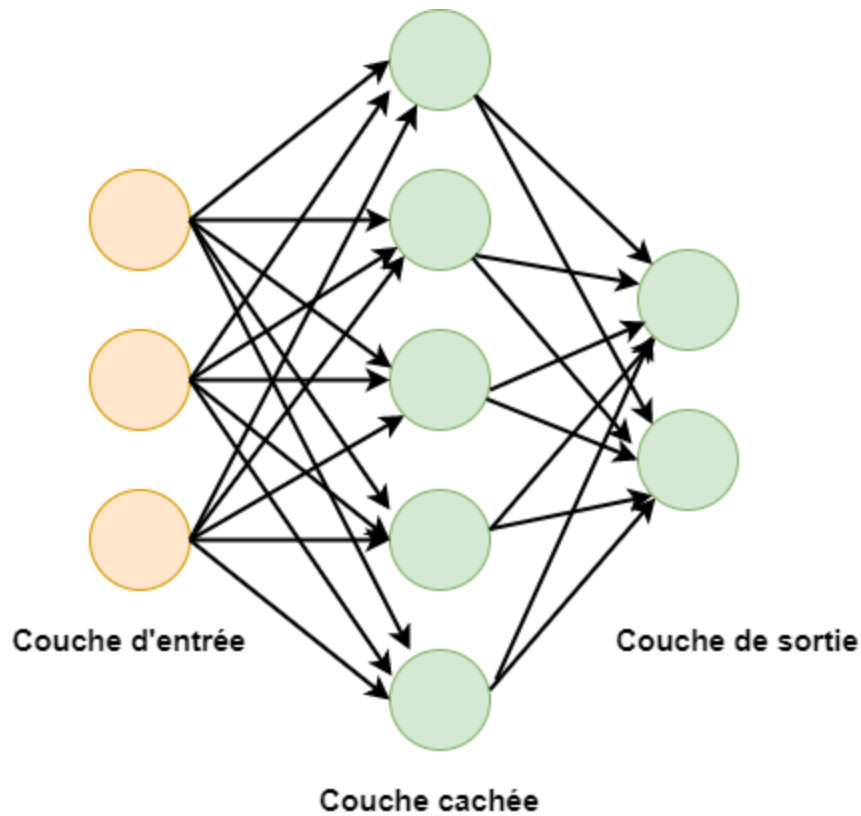


Figure 2 : Composition d'un réseau de neurones artificiels à une couche cachée.

La couche d'entrée contient des pondérations, des fonctions d'activation et des fonctions de coût. Les connexions entre les neurones sont appelées des poids qui correspondent à des valeurs numériques. Les poids entre les neurones déterminent la capacité d'apprentissage du réseau de neurones. Lors de la formation d'un réseau de neurones artificiels, les poids entre les neurones sont modifiés.

Il existe plusieurs types d'architectures de RNA dont en voici les principales : :

- **Réseau de neurones à alimentation directe (*feedforward neural network*)** : il s'agit du réseau de neurones le plus simple, dans lequel les neurones sont organisés en couches et les signaux ne se déplacent que dans une direction, soit de l'entrée vers la sortie.
- **Réseau de neurones classique à rétroaction** : c'est un type de RNA qui comprend des connexions de rétroaction permettant à la sortie du réseau d'influencer sa propre entrée. Dans ces réseaux, la sortie d'un ou plusieurs neurones est renvoyée en tant

qu'entrée à un ou plusieurs neurones antérieurs du réseau, permettant au réseau d'apprendre et de s'adapter en fonction de ses sorties précédentes. Les connexions de rétroaction dans ces réseaux créent des connexions récurrentes, ce qui permet au réseau de traiter des séquences de données, telles que des données temporelles.

- **Réseau de neurones récurrents (*recurrent neural network*)** : dans ce type de réseau, les neurones sont organisés en couches, mais les signaux peuvent circuler dans toutes les directions, ce qui permet aux réseaux récurrents de modéliser des séquences et des données temporelles. Parmi les réseaux de neurones récurrents, on peut citer les réseaux de neurones récurrents à mémoire à court terme (*long short-term memory network*). Ceux-ci représentent une variante des réseaux récurrents, qui permet de modéliser des séquences plus longues en mémorisant les informations importantes et en oubliant les informations obsolètes.
- **Réseau de neurones adversaires génératifs (*generative adversarial network*)** : ce type de réseau est conçu pour générer des données à partir de distributions de probabilité en utilisant un générateur et un discriminateur pour améliorer la qualité des données générées.
- **Réseau de neurones convolutionnels (*convolutional neural network*)** : ce type de réseau est souvent utilisé pour la vision par ordinateur et l'analyse de données spatiales. Il est conçu pour détecter les caractéristiques locales des données d'entrée en utilisant des filtres convolutifs (Naïma Humblot, 2021; Hourrane, 2018; Wira, 2009).

2.2.2 Fonctionnement des réseaux de neurones artificiels

Le fonctionnement des RNA est basé sur l'interconnexion de plusieurs neurones artificiels. Chaque neurone est une unité de traitement qui reçoit des entrées, les traite et génère une sortie. Les entrées sont généralement des données numériques qui sont pondérées et sommées avant d'être passées à travers une fonction d'activation qui détermine la sortie du neurone (Drew et Monson, 2000).

En d'autres termes, les informations sont toutes d'abord introduites dans la couche d'entrée (X_1, X_2, \dots, X_m) pour être transmises à la couche cachée. Ensuite, l'interconnexion entre ces deux couches attribue de manière aléatoire des poids (W_1, W_2, \dots, W_m) pour chaque entrée

à un poids initial. Un biais (b) est ensuite ajouté à chaque neurone d'entrée, puis la somme des poids et biais combinés passe par la fonction d'activation (f) (fig.3).

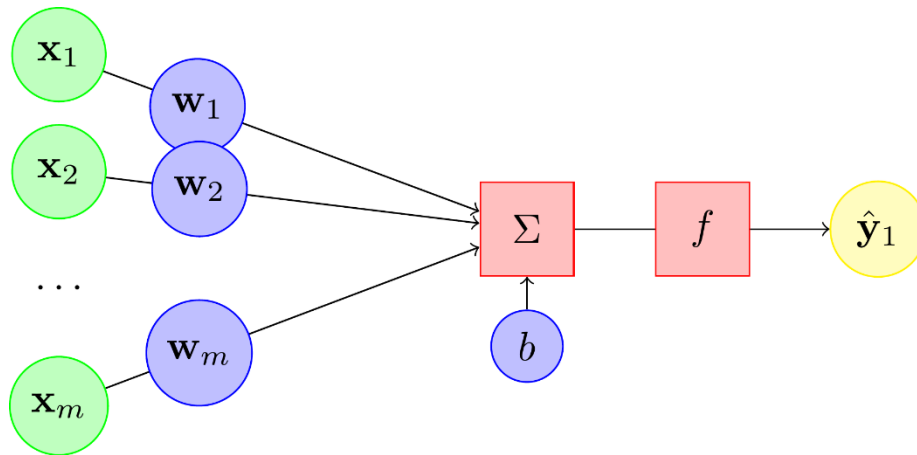


Figure 3: Structure d'un réseau de neurones artificiels(perceptron)

Le processus de fonctionnement d'un RNA se déroule en plusieurs étapes :

- i. **La propagation avant (*feedforward*)** : les entrées sont introduites dans le réseau et sont propagées à travers les différentes couches de neurones jusqu'à la couche de sortie. À chaque couche, les neurones effectuent des calculs en utilisant les poids qui leur sont assignés pour produire une sortie.
- ii. **La fonction d'activation** : une fois que la sortie de chaque neurone est calculée, elle est passée à une fonction d'activation non linéaire, qui introduit de la non-linéarité dans le modèle. Cette non-linéarité est essentielle pour permettre au réseau de modéliser des relations complexes entre les entrées et les sorties.
- iii. **La rétropropagation de l'erreur (*backpropagation*)** : une fois que la sortie de la couche de sortie est calculée, l'erreur entre cette sortie et la sortie attendue est calculée. Cette erreur est ensuite propagée en sens inverse à travers le réseau, couche par couche, en utilisant la dérivée de la fonction d'activation et les poids de chaque connexion pour calculer l'erreur à attribuer à chaque neurone.
- iv. **L'ajustement des poids** : une fois que l'erreur est propagée à travers le réseau, les poids sont ajustés en utilisant un algorithme d'optimisation tel que la descente de gradient. L'objectif de cet ajustement est de minimiser l'erreur globale du réseau en trouvant les valeurs optimales pour les poids.

- v. **Répétition des étapes *i* à *iv*** : les étapes de propagation avant, de fonction d'activation, de rétropropagation de l'erreur et d'ajustement des poids sont répétées plusieurs fois (appelées "époques " ") jusqu'à ce que l'erreur du réseau soit suffisamment faible ou que le réseau ait atteint une certaine précision.
- vi. **Évaluation du modèle** : une fois que le réseau a été entraîné, il peut être utilisé pour prédire des sorties pour de nouvelles entrées. Ces prédictions peuvent ensuite être évaluées par rapport à des sorties réelles pour évaluer la qualité du modèle.

La fonction d'activation est responsable du déclenchement du nœud pour effectuer l'extraction de caractéristiques et, finalement, pour calculer la sortie (Drew et Monson, 2000).

La fonction d'activation est le facteur le plus important dans un réseau de neurones, car elle décide si, oui ou non, un neurone sera activé et si son information sera transférée à la couche suivante. Cela signifie simplement qu'elle décidera si l'entrée du neurone dans le réseau est pertinente, ou non, dans le processus de prédiction. Pour cette raison, on l'appelle aussi seuil ou transformation pour les neurones qui peuvent faire converger le réseau.

2.3 Classification par imagerie

La classification basée sur les "*imagerie*" est une technique d'apprentissage automatique qui consiste à découper une image en petites régions appelées "*imagerie*" et à classer chaque imagerie individuellement. Cette approche est souvent utilisée pour classer des images à haute résolution spatiale, telles que des images satellites ou médicales, où les caractéristiques d'intérêt peuvent être très petites par rapport à la taille de l'image (Hou *et al.*, 2016).

La classification des imagerie est généralement effectuée à l'aide d'un réseau neuronal convolutif. Ces imagerie sont introduites dans un CNN, qui apprend à classer chaque imagerie individuellement en fonction de ses caractéristiques visuelles. Une fois que chaque imagerie a été classée, les prédictions sont généralement agrégées pour produire une prédiction globale pour l'image entière.

Cette méthode permet d'obtenir de meilleurs résultats de classification en utilisant des modèles plus simples que ceux qui seraient nécessaires pour classer l'image entière en une

seule fois. Cependant, cela peut nécessiter plus de temps de calcul, car cela implique de traiter chaque image individuellement.

2.4 Détection des objets

La détection d'objets est un domaine de l'apprentissage profond qui consiste à identifier la présence et la localisation d'objets spécifiques dans une image ou une vidéo (Mahendru et Dubey, 2021). Il existe plusieurs approches pour la détection d'objets en utilisant l'apprentissage en profondeur, notamment :

- i. **R-CNN** : les régions proposées par R-CNN sont d'abord générées, puis chaque région est classée et localisée. C'est une méthode assez lente, mais précise (Girshick *et al.*, 2014).
- ii. **YOLO** : *You Only Look Once* (YOLO) est une méthode plus rapide que R-CNN, car elle effectue une seule inférence sur l'ensemble de l'image en une seule fois (Redmon *et al.*, 2016).
- iii. **SSD** : *Single Shot Detector* (SSD) est une méthode de détection d'objet rapide et précise qui utilise un réseau de neurones à convolution pour détecter les objets dans une image (Liu *et al.*, 2016).
- iv. **Faster R-CNN** : Faster R-CNN utilise une méthode de génération de propositions plus rapide pour accélérer la détection d'objets (Ren *et al.*, 2015a).
- v. **Mask R-CNN** : *Mask R-CNN* est une variante de *Faster R-CNN* qui permet également de segmenter l'image pour identifier les zones d'objets spécifiques (He *et al.*, 2017).

En général, la détection d'objets en utilisant l'apprentissage profond est une tâche complexe qui nécessite une grande quantité de données d'entraînement et de ressources de calcul. Cependant, ces techniques sont de plus en plus utilisées dans de nombreuses applications, telles que la reconnaissance de plaques d'immatriculation, la surveillance vidéo, la détection de défauts de qualité et la reconnaissance faciale.

2.5 L'apprentissage profond en géomatique

En géomatique, les modèles et méthodologies basés sur l'IA sont couramment utilisés pour augmenter les performances des technologies (Chen *et al.*, 2023). L'utilisation de

l'apprentissage profond en géomatique présente plusieurs avantages, notamment une plus grande précision dans l'analyse des données spatiales. Cela peut conduire à une meilleure prise de décision dans des domaines tels que la gestion des ressources naturelles, l'urbanisme, la surveillance environnementale, la planification d'urgence, etc.

L'apprentissage profond peut être appliqué à la géomatique de différentes manières. Par exemple, il peut être utilisé pour l'analyse et la classification d'images satellites, la prédiction des changements environnementaux à partir de données géospatiales ou encore, la modélisation de phénomènes spatiaux complexes (Akhatar, 2023; Karas, 2023; Kpegouni *et al.*, 2023; Lin-Kwong-Chon *et al.*, 2023; Marques, 2023).

Les méthodes d'apprentissage profond ont été appliquées dans de nombreuses études sur les technologies de télédétection en raison de leurs performances exceptionnelles dans l'extraction de caractéristiques, au niveau de la classification et l'interprétation des données. Ces dernières années, on note plusieurs applications de l'apprentissage profond en géomatique. Par exemple, dans l'article de Yadav *et al.* (2023), on observe une méthode pour détecter les plants de coton volontaires dans un champ de maïs à l'aide de l'apprentissage en profondeur sur des images de télédétection capturées par des drones. Les auteurs ont utilisé un algorithme de détection d'objets basé sur un CNN pour extraire des caractéristiques des images et localiser les plants de coton volontaires dans un champ de maïs. Les résultats ont montré que l'algorithme a réussi à détecter les plants de coton volontaires avec une précision élevée, même dans des conditions difficiles telles que la présence de plantes de maïs et de mauvaises herbes qui pourraient interférer avec la détection. Une autre application très intéressante est la détection automatique des bâtiments à partir des images LiDAR et aériennes à très haute résolution en utilisant un réseau de neurones profond (Ojogbane *et al.*, 2021). Dans cette étude, ils ont utilisé un modèle CNN qui a été entraîné sur un grand ensemble de données comprenant des images LiDAR et aériennes de différents types de bâtiments. Le modèle a ensuite été utilisé pour détecter les bâtiments dans de nouvelles images LiDAR et aériennes. Les résultats ont montré que la méthode proposée est efficace pour détecter les bâtiments avec des taux de détection élevés et de faibles taux d'erreur dans l'identification des objets.

Au département de géomatique appliquée de l'université de Sherbrooke, ces dernières années, plusieurs projets de recherche se sont intéressés à l'apprentissage profond. On peut d'ailleurs citer Kpegouni *et al.*(2023) qui ont proposé une méthode basée sur l'apprentissage profond pour aider au recensement des populations en Afrique à partir d'images satellites à très haute résolution. D'autres auteurs tels que THRSabo *et al.* (2022) ont développé une méthode d'adaptation des architectures ADDA et semi-ADDA pour la détection d'objets par apprentissage profond sur les images satellites THR. Diaby *et al.*, (2021) ont développé un système intelligent de reconnaissance automatisée pour la caractérisation des états de surface de la chaussée en temps réel par une approche multi capteurs. Enfin, Clabaut *et al.* (2020) ont utilisé une approche d'apprentissage en profondeur pour détecter des chapeaux de fer dans l'Arctique canadien.

2.6 Autres méthodes de détection des objets en dehors de l'IA

Il existe d'autres méthodes de détection des images en dehors de l'IA. Parmi ces méthodes, on peut citer :

- i. **la segmentation des images** : la segmentation d'image est une tâche de base dans l'analyse d'image dans laquelle l'image est partitionnée en régions significatives dont les points ont presque les mêmes propriétés, par exemple, les niveaux de gris, les valeurs moyennes ou les propriétés de texture (Bins *et al.*, 1996). La segmentation est utilisée dans le traitement des images de télédétection depuis l'avènement du satellite Landsat-1. Cependant, après le lancement du satellite haute résolution IKONOS en 1999, le paradigme de l'analyse d'images est passé d'une approche basée sur les pixels à une approche basée sur les objets. Par conséquent, le but de la segmentation est passé de l'aide à l'étiquetage des pixels à la reconnaissance d'objets (Hossain et Chen, 2019).
- ii. **Techniques bayésiennes classiques de détection** : la théorie bayésienne peut être appliquée à l'analyse d'images satellites optiques pour résoudre des problèmes tels que la classification, la détection d'objets et l'estimation de paramètres. Dans cette approche, la probabilité conditionnelle de la classe ou de l'objet donné de l'image est calculée en utilisant des connaissances à priori sur les propriétés des classes ou des objets, ainsi que des informations sur l'image elle-même (Proia et Pagé, 2009).

iii. **Détection des objets basée sur les filtres de Gabor** : la détection basée sur les filtres de Gabor est une technique de traitement d'image qui utilise des filtres de Gabor pour extraire des caractéristiques de texture à partir d'images.

Le filtre Gabor, développé à l'origine par Dennis Gabor, est un filtre linéaire principalement utilisé pour la détection des contours, l'évaluation de surface, l'extraction de caractéristiques, la reconnaissance d'objets et de nombreuses autres applications (Munawar *et al.*, 2021).

La détection basée sur les filtres de Gabor peut être utilisée pour la détection d'objets dans des images satellites, notamment la détection de bâtiments, de routes, de voitures et d'autres objets de forme régulière ou répétitive. Les caractéristiques de texture extraites par les filtres de Gabor peuvent être combinées avec d'autres caractéristiques, telles que la forme et la couleur, pour améliorer la précision de la détection (Huang *et al.*, 1979; Lee et Wang, 1999; Munawar *et al.*, 2021; Twum *et al.*, 2022).

iv. **Détection basée sur les caractéristiques géométriques** : cette méthode implique la détection d'objets en se basant sur leurs caractéristiques géométriques, telles que leur forme, leur taille et leur orientation. Les objets peuvent être détectés en comparant ces caractéristiques avec des modèles préalablement définis.

2.7 Augmentation des données

Les modèles d'apprentissage profond pour la détection des objets dépendent d'une grande quantité de données pour des résultats cohérents. La disponibilité d'une quantité suffisante de données n'est pas toujours possible.

« L'augmentation des données est un ensemble de techniques qui peuvent être utilisées pour étendre la taille de l'ensemble de données et améliorer la qualité des images dans l'ensemble de données d'une quantité requise » (Kaur et al., 2021).

Les méthodes d'augmentation des données peuvent être regroupées en quatre catégories à savoir : les méthodes basées sur des transformations géométriques, des transformations de couleur, des occlusions aléatoires, enfin, les méthodes basées sur l'apprentissage en profond (Kaur *et al.*, 2021). En effet, lors des transformations, il existe des risques de pertes

de l'information (de géométries par exemple). Les méthodes basées sur l'apprentissage profond, (style neuronal, l'apprentissage additif et les réseaux additifs génératifs) permettent de s'affranchir dans la majorité des cas, du risque de cette perte d'information (Chen et Cao, 2019).

Les transformations géométriques sont des techniques qui modifient la position des pixels de l'image dans l'ensemble de données d'apprentissage (la rotation, la mise à l'échelle, le retournement, le recadrage, le remplissage, la rotation, la transformation affine, etc). Ces techniques ont été utilisées dans le cadre de cet essai étant donné que la quantité des données d'entraînement disponibles est très insuffisante (fig.4).



Figure 4: Quelques techniques de transformation des images. L'image a été tirée de (Solawetz, 2020)

2.8 Quelques bibliothèques d'apprentissage profond

Nous avons exploré quatre bibliothèques d'apprentissage profond dans le cadre de cet essai :

- *TensorFlow* est une plateforme d'apprentissage profond, populaire et mature, développée par Google. Elle offre une grande flexibilité pour la création de modèles d'apprentissage profond et prend en charge une large gamme de tâches d'apprentissage automatique. *TensorFlow* dispose également d'une communauté active et d'une documentation complète (YAPICI et Topaloğlu, 2021).
- *PyTorch* est un *framework* d'apprentissage profond développé par Facebook, qui se distingue par sa simplicité et sa facilité d'utilisation. Il offre un style de

programmation dynamique, ce qui le rend très approprié pour la recherche en intelligence artificielle et l'expérimentation. PyTorch est également bien intégré avec Python et offre une communauté active et une documentation complète (Rao, 2023a).

- Keras est une API de haut niveau pour les réseaux de neurones, qui peut être utilisée avec des backends tels que *TensorFlow*, Theano ou CNTK. Elle est connue pour sa simplicité et sa convivialité, ce qui la rend appropriée pour les débutants en apprentissage profond. Keras offre également une large gamme de fonctionnalités pour la création de modèles d'apprentissage profond, avec une documentation complète et une communauté active (Elshawi *et al.*, 2021; Rao, 2023a).
- MXNet est un *framework* d'apprentissage profond développé par Apache, qui offre une grande flexibilité et une efficacité élevée pour la création de modèles d'apprentissage profond. Il prend en charge plusieurs langages de programmation, dont Python, et offre également des fonctionnalités pour le déploiement sur des appareils embarqués et des systèmes distribués (Kim *et al.*, 2022; Rao, 2023a).

3. Matériel et méthodes

3.1 Site d'étude

Le site d'étude regroupe deux municipalités de la MRC des Sources : Saint-Camille et Wotton (fig.4). La MRC des Sources se trouve à la limite Nord-Ouest de la région administrative de l'Estrie. D'après le ministère des Affaires municipales et de l'Habitation (MAMH)(2023), la MRC des Sources a une superficie totale de 785.70 km² avec une population de 14 787 habitants. Elle est composée de sept municipalités à savoir : la municipalité de Danville, Ham-Sud, Sain-Adrien, Saint-George-de-Windsor-Val-des-Sources, Saint-Camille et Wotton (MAMH, 2023).

D'après le MAMH (2023), Saint-Camille est la troisième municipalité la moins peuplée de la MRC avec une population de 572 habitants après celle de Ham-Sud (245 habitants) et Saint Adrien 566 habitants. Elle est située en Estrie, à 35 km environ au nord-est de Sherbrooke et 25 km au sud-est de Val-des-Sources. Avec une superficie de 82.650 km², elle est couverte à 60 % par la forêt.

La municipalité de Wotton est la troisième municipalité la plus peuplée de la MRC des Sources avec une population de 1 438 habitants. Elle a une superficie de 143,78 km². Tout comme la municipalité de Saint-Camille, la municipalité de Wotton fait partie de la région administrative de l'Estrie. Elle est limitée au nord par Saint-Rémi-de-Tingwick, à l'est par Saint-Joseph-de-Ham-Sud, à l'ouest par Danville, au sud-ouest par Saint-Georges-de-Windsor et Saint-Camille au Sud-Est. D'ailleurs, la municipalité de Wotton est essentiellement agricole.

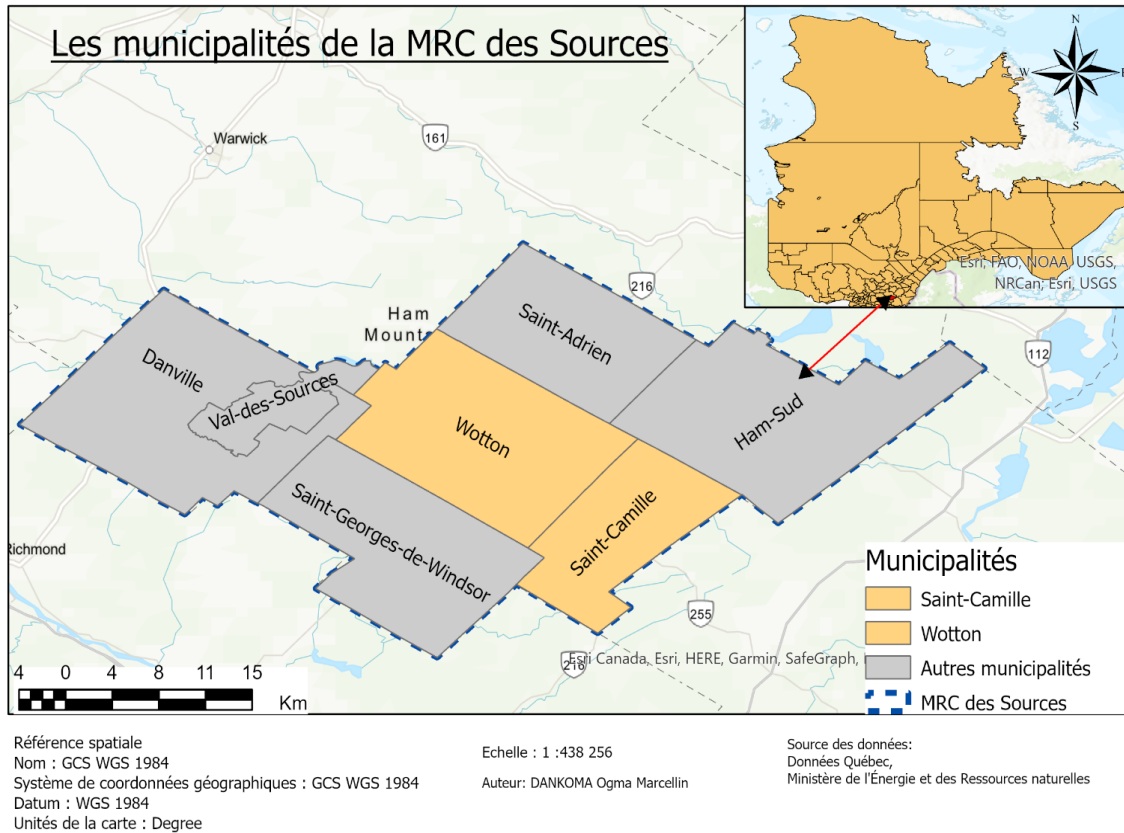


Figure 5: Carte de la MRC des sources et localisation de la zone d'étude (Wotton et Saint-Camille)

3.2 Les données de l'étude

3.2.1 Les données

Les données utilisées proviennent de plusieurs sources, dont de MGA, du ministère des Forêts, de la Faune et des Parcs (MFFP) et du ministère de l'énergie et des ressources naturelles (MERN). Dans un premier temps, MGA a mis à disposition les photos des inspections précédentes. Ces images ont été acquises par des inspecteurs engagés par MGA dans le cadre de la mise en place du plan d'intervention d'infrastructures routier (PIIRL) de la MRC des Sources. Les photos ont été prises lors de la campagne d'été 2022 à l'aide d'un téléphone intelligent. Les images ont une dimension de 3264 par 2448 pixels. Ensuite,

une sélection des ponceaux potentiels a été effectuée, en tenant compte de la diversité des ponceaux et en s’assurant que ceux-ci étaient visibles à l’œil nu.

Les autres données, soit la couche des limites administratives, la couche du réseau hydrographique et des lits d’écoulement, ont été obtenues sur le site de Données Québec. L’ensemble des données, leurs formats et leurs sources sont présentés dans le tableau 1.

Tableau 1 : Données utilisées

Données	Format	Références
Localisation des ponceaux déjà existants	Vectorel(.shp)	MGA
Routes locales 1 et 2	Vectorel(.shp)	MGA
Photos des ponceaux déjà inspectés	Matriciel(.jpg)	MGA
Lits d’écoulement	Vectorel(.shp)	MFFP
Réseau hydrographique	Vectorel (.gdb)	MERN
Découpage administratif de la municipalité des Sources	vectorelles (.shp)	MERN
Images drone des ponceaux	Matriciel (.jpg)	Université de Sherbrooke

3.2.2 Sorties de terrain

Deux sorties de terrain ont été planifiées durant l’automne 2022. Les dates des sorties de terrain ont été choisies en fonction de la température et de la luminosité du jour. En effet, pour planifier la sortie de terrain, il a été nécessaire de tenir compte des conditions météorologiques : l’absence de neige, la présence de soleil, de faibles vents, etc. Ainsi, la première sortie de terrain a eu lieu le 11 novembre 2022 à Wotton et la deuxième sortie, le 3 décembre 2022 à Saint-Camille.

Les couches obtenues de MGA (la localisation des ponceaux, les segments de routes locales 1 et 2) ont permis de développer une application (fig. 6) de collecte des données en utilisant l'outil *Field Map* dans ArcGIS Online.

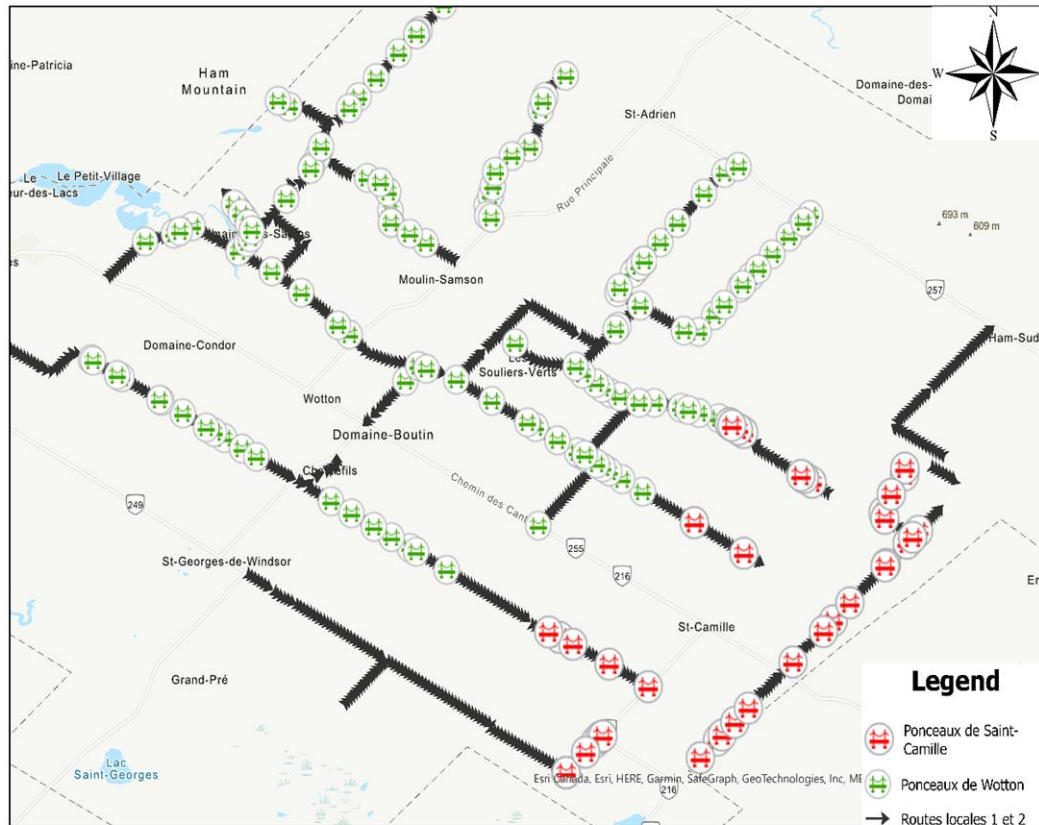


Figure 6: Présentation de la carte Field Maps pour planifier la sortie de terrain

« ArcGIS Field Maps est une application tout-en-un qui utilise des cartes basées sur les données et des formulaires mobiles pour aider les travailleurs à effectuer la capture et la mise à jour des données, à trouver des actifs et des informations et à signaler leurs emplacements en temps réel. » (« ArcGIS Field Maps », 2022).

L'application *Field Maps* qui est développée a été installée sur les téléphones intelligents. Cette application permettait d'obtenir des informations sur la position en temps réel et pouvait être utilisé en mode « hors ligne ». Cette dernière fonctionnalité s'est avérée d'une grande utilité, car à Wotton par exemple, la connexion n'était pas toujours disponible.

3.2.3 Les données d'entraînement et de validation

Les données d'entraînements ont été obtenues sur le site d'étude grâce à la sortie de terrain. Ce sont des images de drones. Elles ont été prises par le drone DJI mini 2 du département de géomatique appliquée et télédétection de l'université de Sherbrooke. Le DJI mini 2 est un drone qui pèse moins de 250 grammes et donc, est exempté d'une autorisation préalable pour voler (Transports Canada, 2023).

La dimension des images prises par drones est de 5472 par 3648 pixels avec une hauteur de vol de 20 mètres. La hauteur des images est déterminée en fonction des lignes électriques. Par exemple, pour des raisons de sécurité, il n'est pas recommandé de faire voler un drone plus bas ou plus proche des obstacles (par exemple les lignes électriques ou les poteaux électriques). Aussi, il faut tenir compte, de la hauteur maximale à laquelle l'œil nu est capable de distinguer le ponceau dans une image.

Les données connexes telles les routes locales 1 et 2 et les lits d'écoulement, ont permis de valider les résultats obtenus et d'améliorer la précision du modèle.

3.3 Méthodologie

La méthodologie de cet essai est présentée à la figure 7. La première partie de cette étape consiste en une revue de littérature, permettant ainsi de se familiariser et de choisir la méthode d'inventaire automatisé des ponceaux. Cette étape a joué un rôle clé dans la réflexion sur la librairie d'apprentissage profond, facilitant ainsi son choix et la compréhension de ses interactions. De plus, elle a permis de déterminer le format des données, le fonctionnement des algorithmes de détection des objets, ainsi que la méthode de détection des objets.

La deuxième partie est la collecte des données terrain. C'est à ce stade que le choix de l'échantillon des ponceaux a été fait. Cette étape a également permis de planifier la sortie de terrain en développant une application de collecte des données avec *ArcGIS Field Maps*. Parallèlement, à cette étape, une sélection finale de la librairie a été effectuée.

La troisième étape, après la collecte des données, a permis de choisir les données d'entraînement, de validation et de prédiction. Les données d'entraînement ont été ensuite

annotées. Parallèlement aux données d'entraînement, les données de validation ont été également annotées.

La quatrième étape fut la mise en place et l'entraînement d'apprentissage profond. C'est dans cette étape que le modèle a été développé. Plusieurs époques et tests ont été réalisés afin de déterminer le modèle le plus optimal à conserver. Cette partie est suivie de celle de l'évaluation du modèle. L'évaluation du modèle consiste à faire passer aux modèles des données tests, entièrement distincts des données utilisées pour faire l'entraînement. Dans cette étape, les métriques d'évaluation sont utilisées pour mesurer la performance du modèle. Le choix des métriques d'évaluation dépend de l'objectif du modèle. Par exemple, pour un problème de classification, les métriques couramment utilisées peuvent inclure la précision, le rappel, la F-mesure, l'aire sous la courbe ROC, etc. Pour un problème de régression, on peut utiliser des métriques comme l'erreur quadratique moyenne (RMSE), l'erreur absolue moyenne (MAE), etc.

Une fois que le modèle a été évalué, si la précision après l'étape d'évaluation est faible, le modèle est réentraîné. Sinon, on passe à l'étape suivante qui est le post-traitement. Dans cet essai, le post-traitement consiste à exporter les boîtes et à calculer les centroïdes. Une fois le centroïde calculé, on peut déterminer les coordonnées du ponceau à partir d'un simple script sur les images.

La dernière étape consiste à présenter les résultats, à mener une discussion, à formuler des recommandations et à conclure

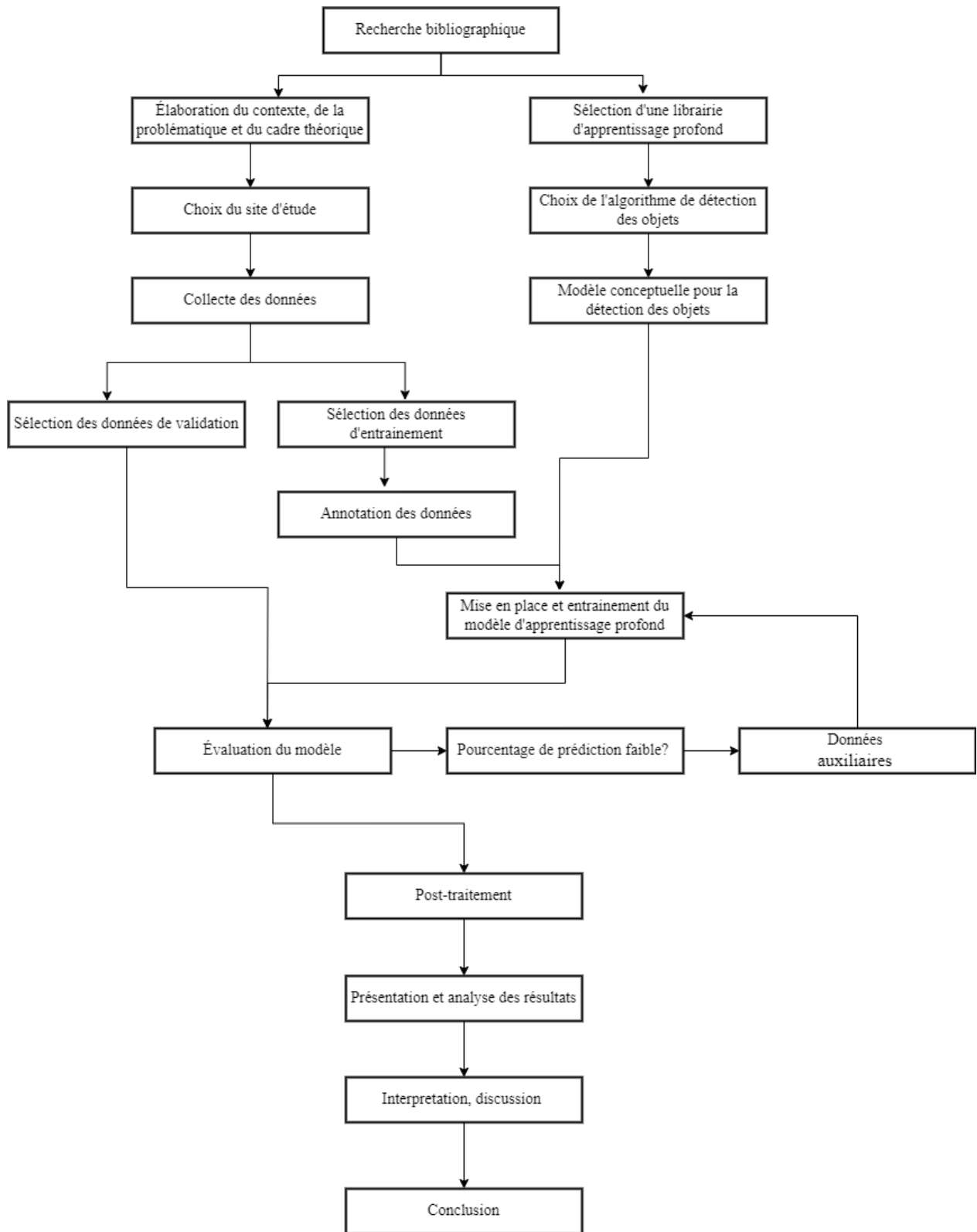


Figure 7: Organigramme Méthodologique de l'essai

3.4 Prétraitements des données d'entraînement

Le prétraitement des données a commencé par la création des *imassettes*. Les dimensions des images étant grandes, 5472 par 3648 pixels, elles ont été décomposées en *imassettes*. La raison principale de ce prétraitement est de réduire la taille des images afin de pouvoir les charger dans la mémoire du GPU et d'offrir une taille uniforme des images pour former des *batches* (lots de données). Il est également possible d'annoter ces images avant de créer des *imassettes* lors de l'entraînement du modèle. Les deux méthodes (la création des *imassettes* avant et après l'annotation) ont été testées et ont donné le même résultat.

Une *imasette* est un carré de taille fixe et centré sur un pixel. C'est une décomposition de l'image en quelques dizaines de pixels de chaque côté (Tabti, 2016). Dans le cas de cet essai, la dimension commune la plus élevée pour les jeux de données est de 512 par 512 pixels. Cette décomposition de l'image a été faite avec le langage Python (fig. 8).



Figure 8: Extrait des *imassettes* utilisés pour la formation du modèle

Les *imassettes* obtenues ont ensuite été annotées afin de créer les jeux de données d'entraînement. Les jeux de données d'entraînement ont été créés via la plateforme d'annotation des images *makesense.ai* (« *Make Sense* », 2023).

Makesense.ai est une plateforme en ligne qui est gratuite pour annoter les images destinées généralement pour l'apprentissage automatique. C'est un outil qui ne nécessite pas d'installation préalable, et qui est multiplateforme. Elle a été développée en format *TypeScript* et est basé sur *React/Redux*, deux bibliothèques gratuites de JavaScript. Cet outil, selon Skalski (2019), est parfait pour la préparation des jeux de données pour de petits projets d'apprentissage profond de vision par ordinateur. Les étiquettes préparées dans « *makesense* » peuvent être téléchargées dans de nombreux formats dépendamment de l'objectif du projet. Pour la détection des objets, les formats prises en charge sont : COCO au format JSON, YOLO et XML, le format CSV et le format JSON. Il existe d'autres outils tels que Labels Studio, diffgram, labeling, cvat, imageTagger, labelMe, VIA, Daturks, etc, (Simon, 2022).

Une seule classe a été choisie dans l'annotation des images, c'est la classe ponceau. L'annotation a consisté à créer des « *Bounding Box* » autour des images. C'est une sorte de carrée entourant le ponceau dans notre image.

Le tableau 2 suivant présente les caractéristiques des jeux de données que nous avons utilisés.

Tableau 2 : Caractéristiques des jeux de données utilisés

	Sortie de terrain	Imagettes
Nombre d'images total	150	1200
Nombre d'images d'entraînement	130	1040
Nombre d'images de test	20	160
Nombre de classes	-	1
Nom des classes	-	Ponceau
Dimension des tuiles	5472 par 3648 pixels	512 par 512 pixels
Bandes	RGB	RGB
Zone de couverture	MRC des Sources	
Taille sur le disque	1.3 Go	96 Mo

Un quart (1/4) des images annotées ont été réservées pour la validation. Et quelques images issues de la sortie de terrain sont réservées pour des tests.

3.4.1 Augmentations des données d'entraînement et de validation

Une augmentation des données a été faite. Les transformations utilisées dans le cadre de cet essai sont :

- **La rotation** : c'est une technique qui consiste à soumettre l'image d'entrée à une rotation aléatoire dans le sens des aiguilles d'une montre ou dans le sens inverse d'un certain nombre de degrés, ce qui modifie la position de l'objet dans l'image. Cette technique a été appliquée pour augmenter les données en changeant l'angle de vue du ponceau.
- **Le retournement**: c'est une autre transformation géométrique qui inverse l'image horizontalement ou verticalement. Le retournement horizontal est

choix de la librairie d'apprentissage profond est une étape importante du projet. Donc, dans ce travail, l'ensemble des quatre librairies sélectionnées (2.8) étaient libres et ouvertes étant donné que l'objectif du projet est de réduire le coût de l'inventaire des pinceaux. D'autres critères incluent la facilité de prise en main de la librairie, la disponibilité de la documentation, la présence d'une communauté active, la performance et la flexibilité de la librairie.

- La facilité de prise en main se réfère à la facilité avec laquelle on peut se familiariser avec la librairie.
- La documentation concerne la qualité et la disponibilité de l'ensemble des tutoriels, des exemples, des forums d'aide.
- La flexibilité représente les opérations possibles que nous pouvons faire avec la librairie, y compris les différents types d'architectures supportées (Cornioley, 2018).
- La communauté constitue l'ensemble des utilisateurs, des développeurs, des chercheurs et d'autres parties prenantes qui utilisent, contribuent et soutiennent activement cette librairie dans leur travail et leurs projets liés à la librairie.
- La performance d'une librairie d'apprentissage profond se réfère à sa capacité à fournir des résultats précis, efficaces et rapides dans la mise en œuvre d'algorithmes d'apprentissage profond.

Le tableau 3 présente la comparaison de librairies d'apprentissage profond, gratuites et libres inspiré de Cornioley (2018). Chaque librairie a été évaluée sur une échelle de 1 à 3, où 1 représente un niveau bas et 3 un niveau élevé.

Tableau 3 : Comparaisons des bibliothèques d'apprentissage profond libres et ouvertes d'après les études faites par (Elshawi et al., 2021; Rao, 2023b; YAPICI et Topaloğlu, 2021)

	TensorFlow	Pytorch	Keras	MXNet
Prise en main	+*	+++	+++	++
Documentation	+++	+++	+++	+
Communauté	+++	+++	++	++
Flexibilité	+++	+++	++	++
Performance	+++	+++	++	++

*+ représente un niveau de 1 tandis que +++ représente un niveau 3.

En fonction de ces critères, il ressort que *TensorFlow*, *PyTorch* et *Keras* sont les bibliothèques d'apprentissage profond les mieux évaluées. Ils offrent une prise en main relativement facile, une documentation complète, une communauté active, une flexibilité élevée et de bonnes performances. *MXNet* présente également des avantages, bien qu'elle soit légèrement moins évaluée dans certains critères tels que la documentation et la communauté.

TensorFlow est bien évaluée en termes de documentation, de communauté, de flexibilité et de performance, mais elle peut être légèrement plus complexe pour ce qui est de la prise en main.

PyTorch obtient de bons scores en termes de prise en main, de communauté, de flexibilité et de performance, mais sa documentation peut être légèrement moins complète que celle de *TensorFlow*.

Keras est également une bonne option en termes de prise en main, de documentation et de performance, mais sa communauté peut être légèrement moins active que celles de *TensorFlow* et *PyTorch*, et sa flexibilité est également légèrement inférieure.

En fonction de ces critères, *PyTorch* et *TensorFlow* sont généralement considérées comme les deux principales options pour l'apprentissage profond en raison de leur popularité, de

leur large adoption par la communauté de recherche et de leur rôle majeur dans le domaine. Keras, quant à elle, est souvent utilisée comme une interface haut niveau pour TensorFlow ou PyTorch, offrant une prise en main plus simple pour les débutants en apprentissage profond (Rao, 2023b).

Le choix s'est finalement porté sur *Pytorch*, bien que sa documentation soit moins complète, elle demeure cependant en constante amélioration

3.4.4 Architecture des modèles Faster R-CNN et YOLOv8

Deux architectures d'apprentissages profonds ont été utilisées dans ce projet : *Faster R-CNN* et *YOLOv8*.

i. Faster R-CNN

Faster R-CNN est une architecture dérivées des architectures R-CNN et Fast R-CNN. Ces deux dernières sont donc les précurseurs de Faster R-CNN.

En effet, le traitement avec l'architecture R-CNN comprend trois étapes principales: la première étape subdivise les images en plusieurs régions où les objets peuvent être localisés. La deuxième étape consiste à générer un vecteur de caractéristique pour chacune de ses régions grâce un CNN. Enfin, pour faire la classification, les vecteurs caractéristiques sont donnés à un ensemble de machines à vecteur support (SVM). Chaque SVM aide à déterminer la classe des objets (S. K *et al.*, 2021). Pendant la classification, un seul objet peut être détecté plusieurs fois. Pour éliminer ces boîtes supplémentaires, l'algorithme de non-maximum suppression (NMS) est utilisé (fig. 10).

Require: Set of predicted bounding boxes B , confidence scores S , IoU threshold τ , confidence threshold T

Ensure: Set of filtered bounding boxes F

```
1:  $F \leftarrow \emptyset$ 
2: Filter the boxes:  $B \leftarrow \{b \in B \mid S(b) \geq T\}$ 
3: Sort the boxes  $B$  by their confidence scores in descending order
4: while  $B \neq \emptyset$  do
5:   Select the box  $b$  with the highest confidence score
6:   Add  $b$  to the set of final boxes  $F$ :  $F \leftarrow F \cup \{b\}$ 
7:   Remove  $b$  from the set of boxes  $B$ :  $B \leftarrow B - \{b\}$ 
8:   for all remaining boxes  $r$  in  $B$  do
9:     Calculate the IoU between  $b$  and  $r$ :  $iou \leftarrow IoU(b, r)$ 
10:    if  $iou \geq \tau$  then
11:      Remove  $r$  from the set of boxes  $B$ :  $B \leftarrow B - \{r\}$ 
12:    end if
13:  end for
14: end while
```

Figure 5: Algorithme de suppression du non maximum. Cet algorithme est tiré de (Terven et Cordova-Esparza, 2023).

L'algorithme NMS est une technique utilisée pour réduire les boîtes englobantes superposées et améliorer la qualité de la détection dans les algorithmes de détection d'objets. Elle permet de filtrer les boîtes englobantes redondantes et non pertinentes pour ne garder que les plus précises. La figure 11 montre la différence entre la sortie avant et après NMS.

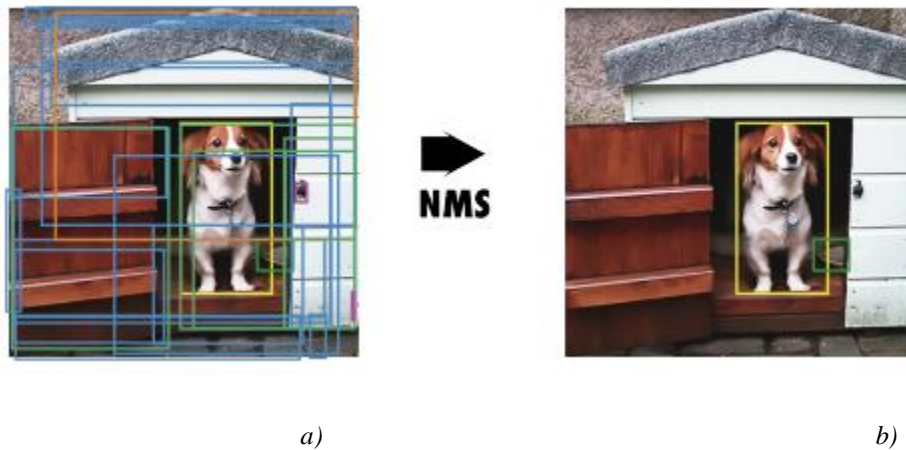


Figure 6: Suppression du non maximum (NMS) (Terven et Cordova-Esparza, 2023); a) objets avec plusieurs boîtes englobantes; b) objets après application de NMS

La figure 11 a) montre la sortie typique d'un modèle de détection d'objets contenant plusieurs boîtes englobantes superposées avant l'application de la technique de suppression du NMS. En, b) la même sortie est présentée après l'application de NMS.

Bien que l'architecture R-CNN ait permis d'obtenir une grande précision dans la détection des objets, plusieurs inconvénients ont été observés. Parmi ces inconvénients, on peut citer le coût élevé en termes de mémoires, de temps de traitement et d'apprentissage. Ces conséquences ont conduit à la modification de l'architecture, donnant ainsi naissance à l'architecture Fast R-CNN. Dans cette nouvelle architecture, c'est l'image entière qui est envoyée à l'entrée d'un CNN. L'ensemble des SVM assurant la fonction de classification est remplacé par une seule couche d'activation de softmax (Babayán *et al.*, 2019). Cela permet au réseau de neurones d'être utilisé une seule fois pour l'ensemble de l'image au lieu de traiter plusieurs régions. Aussi, une seule couche softmax est utilisée au lieu d'avoir à entraîner de nombreux SVM. Cela a permis de réduire considérablement le temps de traitement et d'apprentissage. Aussi, cette modification a permis de réduire le coût en termes de mémoire.

Malgré la robustesse et les avantages significatifs de l'architecture Fast R-CNN contrairement à l'architecture R-CNN, l'algorithme de sélection des propositions de région constitue un inconvénient. Cela a conduit à la modification de cette étape donnant naissance à l'architecture Faster-RCNN.

Faster RCNN est l'une des architectures les plus rapides des réseaux de neurones profonds. Elle a été introduite pour la première fois en 2015 par Shaoqing Ren, Kaiming He, Ross Girshick et Jian Sun. Il est utilisé pour résoudre les problèmes de détection des objets (Ren *et al.*, 2015b). Il est combiné avec les réseaux de proposition par région (RPN). Un RPN est un réseau de neurones convolutif qui prédit les limites d'objets et de scores d'objectivité à chaque proposition (Ren *et al.*, 2015b). Le RPN génère des propositions d'objets, qui sont ensuite utilisées par le détecteur Faster RCNN pour classer et affiner les limites des objets (S. K *et al.*, 2021).

Le diagramme de la figure 12 présente l'architecture des modèles Faster R-CNN.

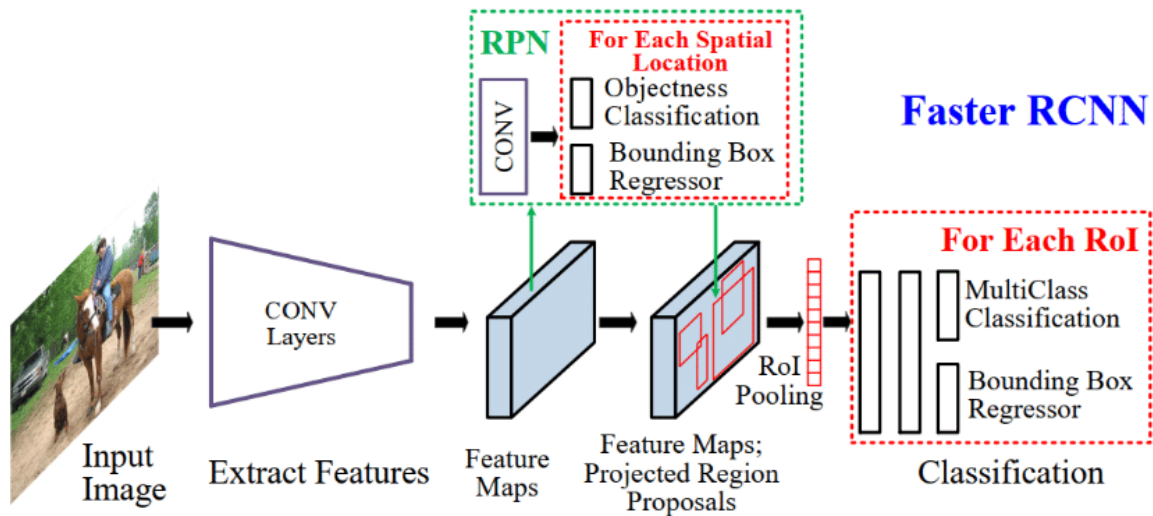


Figure 7: Diagramme de l'architecture complète de Faster RCNN (Ghoury et al., 2019)

ii. YOLOv8

YOLO est un système de détection d'objets en temps réel qui a été présenté pour la première fois en 2016 par Joseph Redmon, Santosh Divvala, Ross Girshick et Ali Farhadi de l'Université de Washington. L'algorithme YOLO traite les images en temps réel et peut détecter plusieurs objets dans une seule image (Redmon *et al.*, 2016).

Dans cette architecture, l'image d'entrée est subdivisée en cellules grâce à une grille. La taille de la grille dépend de la taille d'entrée de l'image et de la taille de la dernière couche convolutionnelle du réseau. Chaque cellule passe par un CNN. Ce CNN a été entraîné préalablement sur un vaste ensemble de données d'images pour acquérir des caractéristiques exploitables dans la détection d'objets. Ce CNN prédit les boîtes de délimitation et les probabilités de classe directement à partir d'images complètes en une seule évaluation. La probabilité de présence d'un objet est estimée pour chaque cellule de la grille (Babayán *et al.*, 2019).

Dans l'algorithme YOLO, les résultats de la détection sont présentés sous la forme d'un tenseur $7 \times 7 \times 1024$. La probabilité estimée de la présence d'un objet de chaque classe dans le rectangle de délimitation actuel est le produit de la probabilité estimée de la présence d'un objet dans une cellule et de la probabilité estimée pour une classe particulière.

La famille de modèles YOLO n'a cessé d'évoluer depuis la sortie initiale en 2016 (tableau 4). Plusieurs mises à jour ont été faites donnant naissance à plusieurs versions de YOLO (Terven et Cordova-Esparza, 2023) :

Tableau 4 : Architectures YOLO et leurs dates de sortie

Architecture	Date de sortie
YOLOv1	2016
YOLOv2	2017
YOLOv3	2018
YOLOv4	2020
YOLOv5	2020
YOLOv6	2022
YOLOv7	2022
YOLOv8	2023

YOLOv8 est la dernière version de YOLO. Cette version a de nouvelles fonctionnalités et améliorations pour une performance, une flexibilité et une efficacité accrues (fig.13) (« Ultralytics YOLOv8 Docs », 2015).

YOLOv8 prend en charge une gamme complète de tâches d'IA visionnaire, y compris la détection, la segmentation, l'estimation de la pose, le suivi et la classification des objets. Cette polyvalence permet aux utilisateurs d'exploiter les capacités de YOLOv8 dans diverses applications et domaines (« Ultralytics YOLOv8 Docs », 2015).

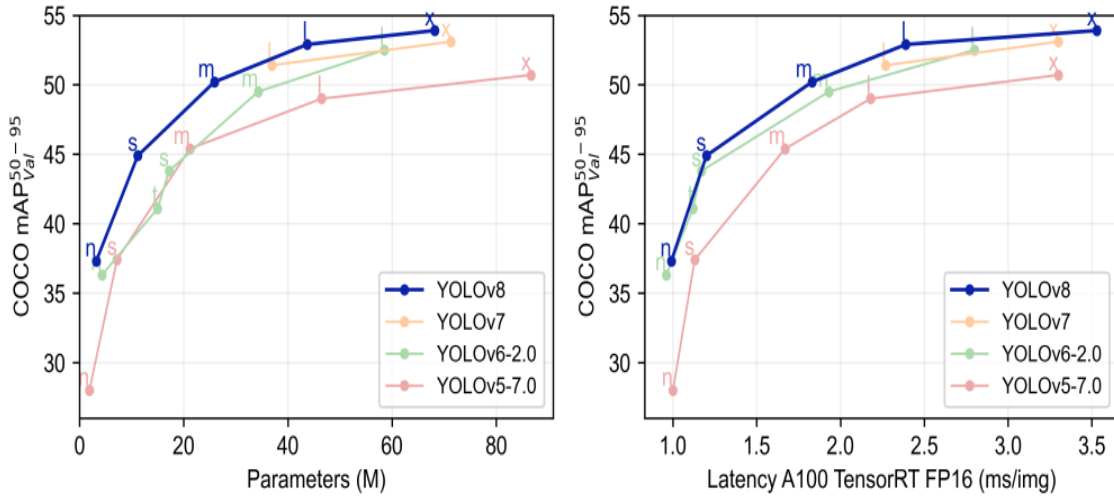


Figure 8: : Comparaison de YOLOv8 aux autres modèles YOLO, image tirée de (« YOLOv8 Ultralytics: State-of-the-Art YOLO Models », 2023).

YOLOv8 est une architecture conçue pour être rapide, précise et facile à utiliser, ce qui en fait un excellent choix pour ce projet de recherche. Le diagramme suivant (fig. 14) présente l'architecture détaillée de YOLOv8 (OpenMMLab, 2023).

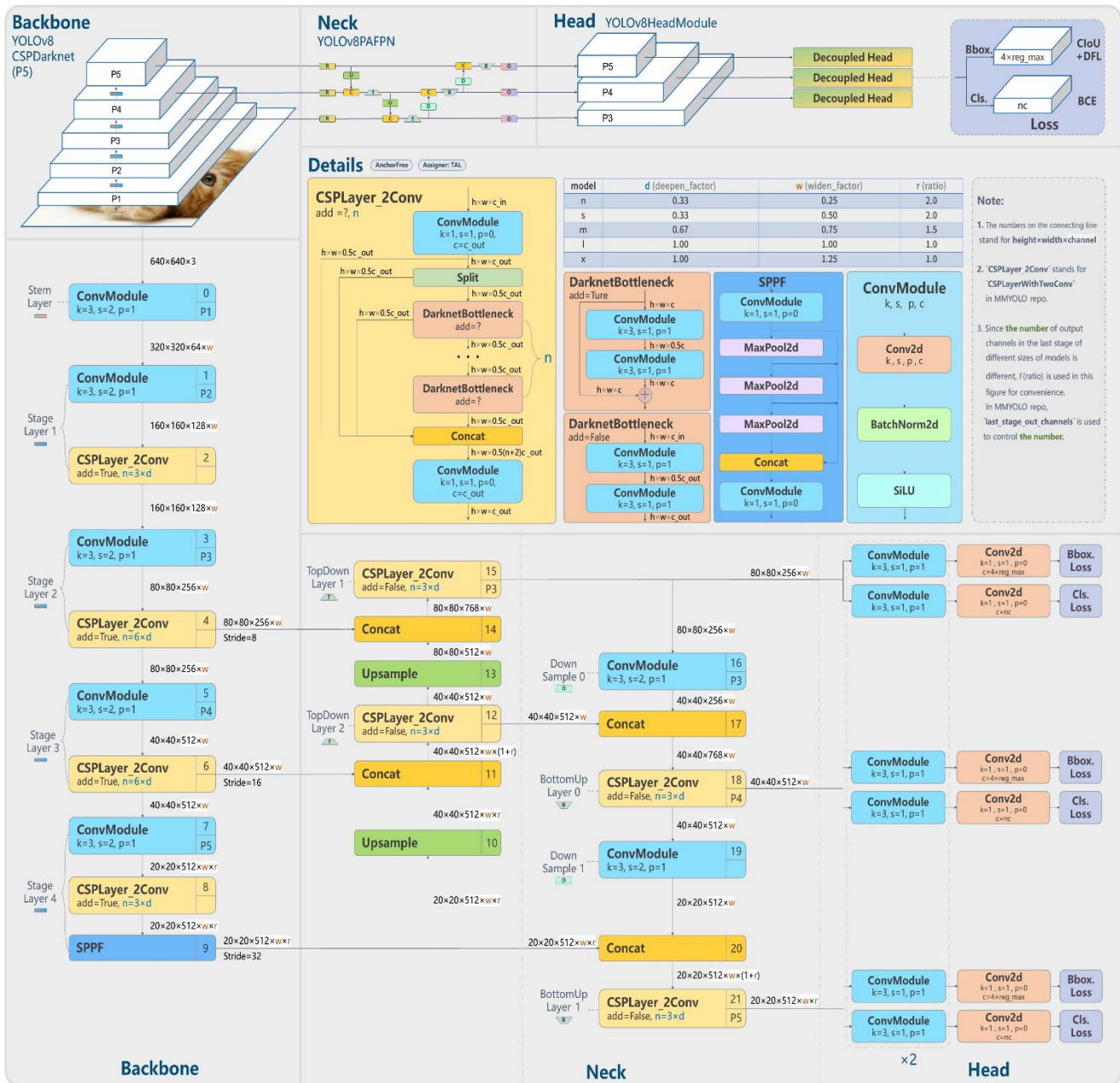


Figure 9: Diagramme de l'architecture complète de YOLOv8 tiré de (OpenMMLab, 2023)

3.4.4.1 Environnement d'entraînement des modèles

Les deux modèles ont été entraînés sur *Google Colaboratory*. Il s'agit d'un service populaire de bloc-notes Python hébergés. Sa version de base est totalement gratuite. Il suffit de créer un compte pour commencer à l'utiliser. Bien que la version gratuite offre une disponibilité de ressources et une vitesse acceptable, elle présente quelques inconvénients,

notamment en ce qui concerne les limitations de la version gratuite, la disponibilité du GPU, le nombre des sessions pour un même compte Google ainsi que la durée d'une session inactive. De plus les versions gratuites ont une quantité limitée de mémoire vive, ce qui est problématique lorsqu'il s'agit de travailler avec des ensembles de données volumineux. *Google Colaboratory* offre actuellement trois plans : le plan Pay As You Go, le plan Colab Pro et le plan Colab Pro+ (« Google Colab », 2019).

Tableau 5 : Comparaison entre de différents plans de google colab.

Pay As You Go (Payment à l' utilisation)	Colab Pro (13.99 \$ CA)	Colab Pro+ (67.20\$ CA)
<ul style="list-style-type: none"> • 100 unités pour 13.99 \$ CA ou 500 unités pour 67.20 \$ CA • Aucun abonnement requis. • GPU plus rapides 	<ul style="list-style-type: none"> • 100 unités de calcul par mois • GPU plus rapides • Plus de mémoire • Terminal 	<ul style="list-style-type: none"> • 500 unités de calcul par mois • GPU plus rapides • Plus de mémoire • Exécution en arrière-plan • Terminal

Le plan Colab Pro est suffisant pour ce projet. Après une première inscription et quelques entraînements, les 100 unités se sont épuisées. Il a fallu faire une nouvelle souscription. Finalement, le choix a été fait pour le plan Google Colab Pro+, ce plan permet d'exécuter en arrière-plan, d'avoir 500 unités et surtout, même si les 500 unités sont épuisées, il est possible d'avoir un accès prioritaire aux ressources et ceux, pour un mois.

3.4.4.2 Évaluation des modèles

Les métriques utilisées pour évaluer le modèle sont, la précision moyenne (**AP**), la **mAP**, le **recall** et le **F1-score** :

- **La précision** mesure le degré d'exactitude des prédictions. Autrement dit, c'est le pourcentage des prédictions qui sont correctes.

C'est le rapport entre les vrais positifs et l'ensemble de la prédiction :

$$precision = \frac{VP}{VP + FP}$$

Où VP représente les vrais positifs (prédiction correcte) et FP les faux positifs (prédits comme positifs, mais incorrects).

- **La mAP (précision moyenne pondérée)** mesure la précision d'un modèle de détection d'objet en calculant moyenne des précisions (AP). C'est une mesure d'évaluation qui est utilisée pour la détection d'objets c'est-à-dire la localisation et la classification.
- **IoU (Intersection over Union)** est une métrique utilisée dans la détection d'objets pour quantifier l'exactitude des boîtes englobantes prédites par le modèle par rapport aux boîtes englobantes réelles. IoU est également souvent utilisé pour déterminer si une prédiction est un vrai positif, un faux positif ou un faux négatif dans le contexte de l'évaluation de la précision et du rappel.
- **Le Recall (rappel)** est une mesure de l'exhaustivité de l'algorithme ou du modèle de récupération et est défini comme la proportion d'éléments pertinents qui sont récupérés par l'algorithme, par rapport à tous les éléments pertinents possibles. Le rappel permet de trouver si la proportion de résultats positifs réels a été identifiée correctement.

Le rappel se calcule de cette façon :

$$Recall = \frac{VP}{VP + FN}$$

Avec FN les faux négatifs (prédits comme négatifs, mais vraiment corrects).

Le rappel prend des valeurs entre 0 et 1, où une valeur de 1 indique une capacité du modèle à identifier toutes les occurrences positives de la classe donnée, et une valeur de 0 indique que le modèle n'a pas réussi à identifier correctement toutes les occurrences positives.

Un rappel élevé signifie que l'algorithme est capable de récupérer une grande proportion d'éléments pertinents.

- **Le F1-score** évalue la capacité d'un modèle de classification à prédire efficacement les individus positifs, en faisant un compromis entre la précision et le *Recall*. Il est utilisé pour évaluer la performance globale du modèle de classification binaire. Il est calculé comme suit :

$$F1 - score = 2 * \frac{(Precision * Recall)}{(Précision + Recall)}$$

Le F1-score varie de 0 à 1, où une valeur de 1 indique une performance parfaite et une valeur de 0 indique une très mauvaise performance.

4. Résultats et discussions

4.1 Résultats

4.1.1 Entraînement du modèle

i. Faster R-CNN

Le modèle Faster R-CNN a été entièrement entraîné dans *Google colab*. Plusieurs tests ont été faits afin d'optimiser la configuration des hyperparamètres (*learning rate, batch size, backbone*, et augmentation de données). Il a été entraîné sur plusieurs époques. Le tableau 6 présente les résultats pour deux époques : 100 et 300.

Tableau 6 : Métriques obtenus après l'entraînement du modèle en fonction du nombre d'époques (architecture Faster R-CNN)

Nombre d'époques	Précision	F1-score	Rcall	Train_loss	Durée d'entraînement (secondes)
100	0.68	0.65	0.62	0.11	5470
300	0.99	0.96	0.93	0.07	16807

Les figures suivantes sont les valeurs des métriques pour les deux expériences :

- 100 époques

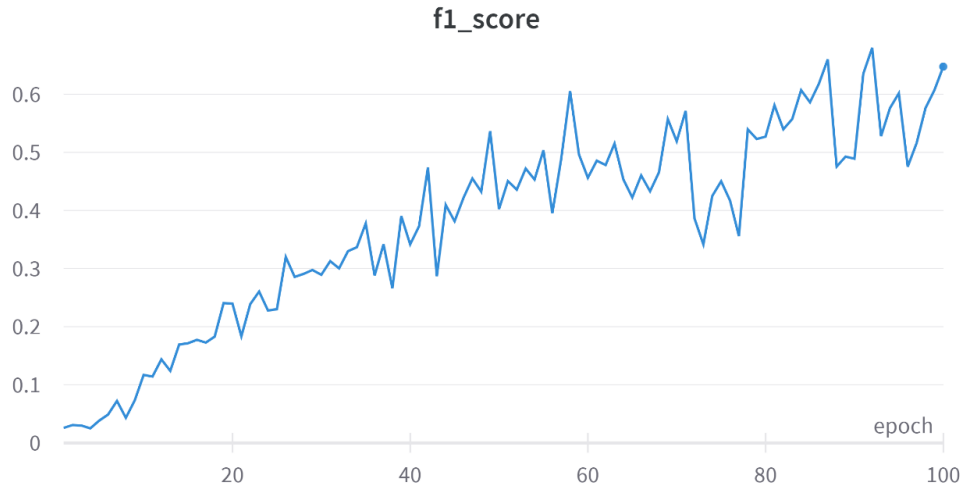


Figure 10: Évolution du $f1_score$ pour 100 époques sur les données de validation

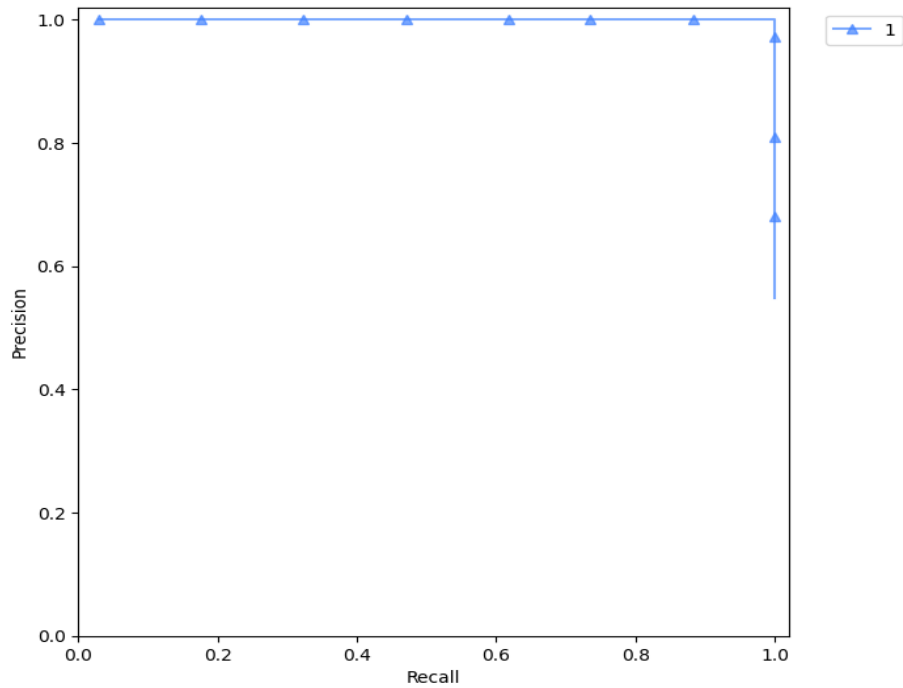


Figure 15-1 : Courbe de précision x Rappel du modèle Faster R-CNN pour 100 epochs sur l'ensemble des données de validation



Figure 11: Évolution de la perte d'entraînement pour 100 époques sur les données d'entraînement

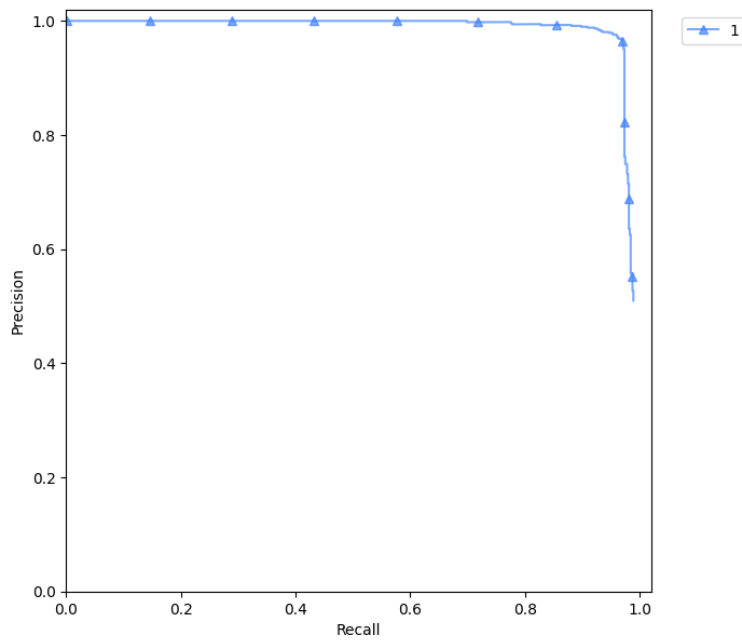


Figure 16-1 : Courbe de Précision x Rappel du modèle Faster R-CNN pour 100 epochs sur l'ensemble des données d'entraînement

- 300 époques

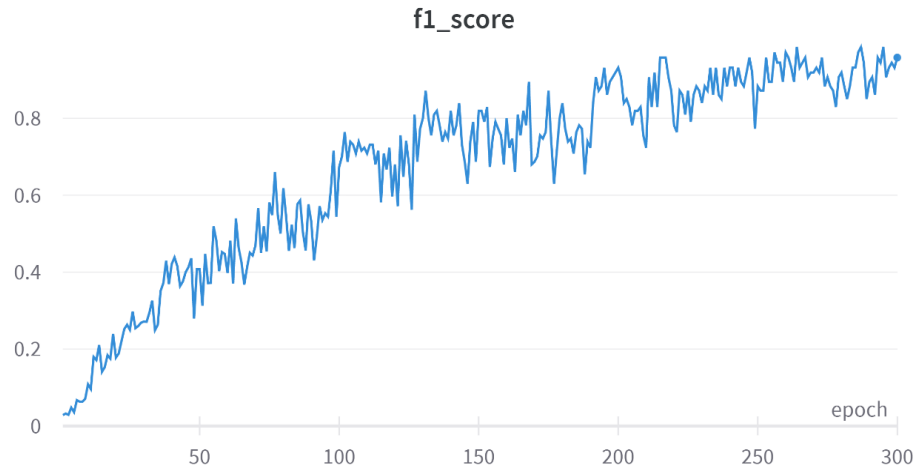


Figure 12: Évolution du $f1_score$ pour 300 époques sur les données de validation

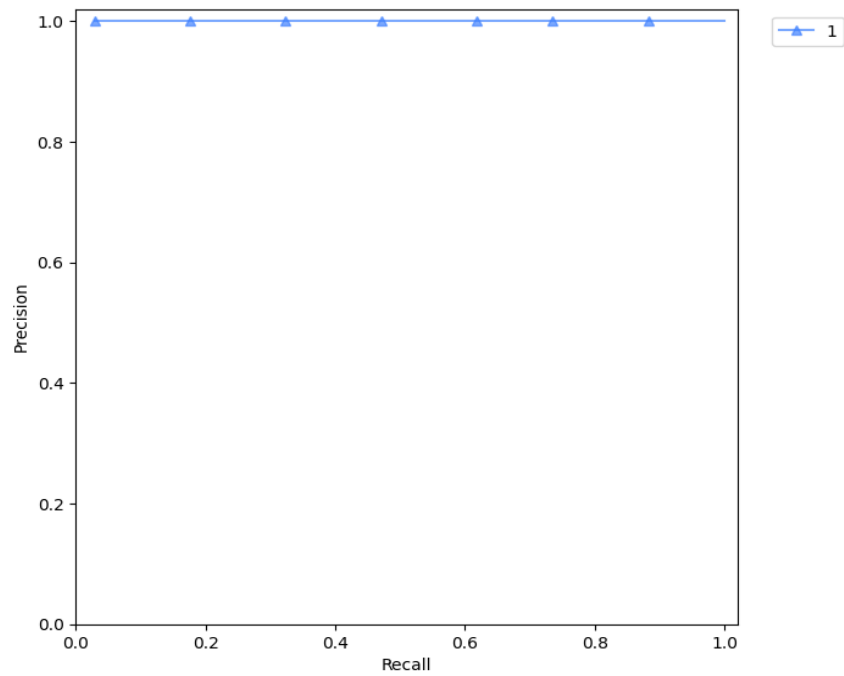


Figure 17-1 Courbe de Précision \times Rappel du modèle Faster R-CNN pour 300 epochs sur l'ensemble des données de validation

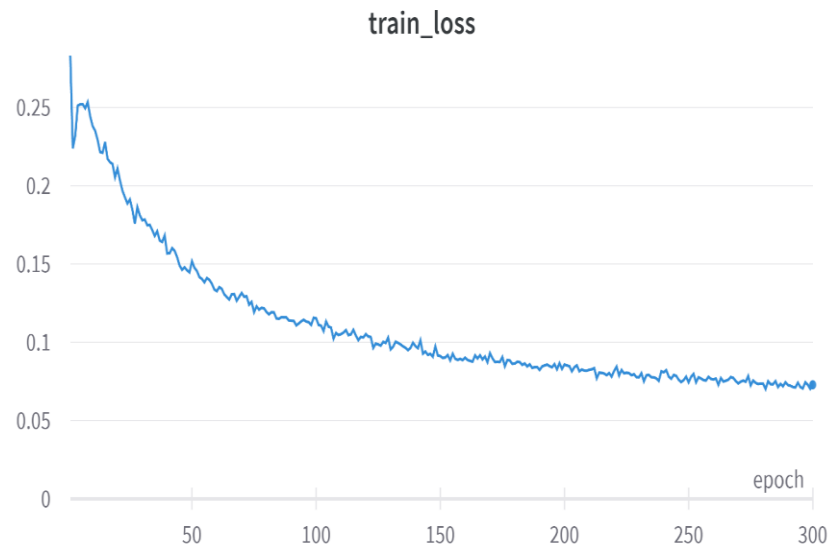


Figure 13: Évolution de la perte d'entraînement pour 300 époques sur les données d'entraînement

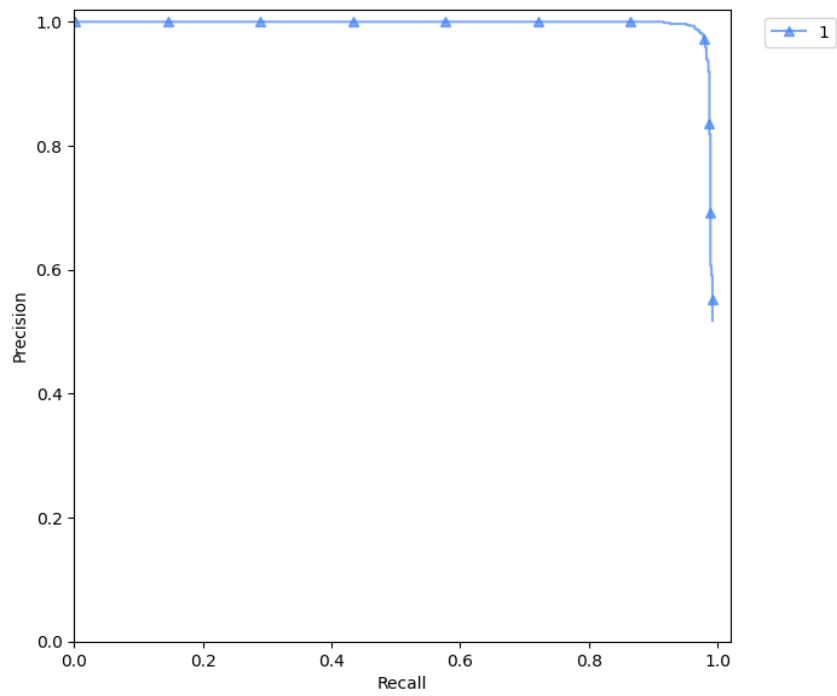


Figure 18-1 Courbe de Précision x Rappel du modèle Faster R-CNN pour 300 epochs sur l'ensemble des données d'entraînement

ii.YOLOv8

L'entraînement de YOLOv8 a été réalisé en deux étapes : la première sur la plateforme *roboflow* et la seconde dans *Google Colaboratory*

- **La première étape : *roboflow***

Elle consiste à utiliser la plateforme *roboflow* (Dwyer et Nelson, 2022). *Roboflow* est une plateforme qui permet aux développeurs de créer leurs propres modèles ou applications de vision par ordinateur, quels que soient leurs niveaux de compétences. Pour la détection des objets, nous avons utilisé les mêmes données (images et annotations) lors de l'entraînement du modèle Faster R-CNN pour garder une certaine logique dans la formation des modèles. Cela permet aussi de mieux comparer les deux architectures.

Pour utiliser efficacement *roboflow*, cinq étapes sont nécessaires : la première sous étape (*Source Images*) consiste à importer les images et les annotations, ce sont des données sources. Ces données sources dans la deuxième sous étape, ont été divisées en données d'entraînement et de test (*Train/Test Split*). La troisième sous étape consiste à faire un prétraitement des données d'entraînement (*Preprocessing*), elle permet d'appliquer les transformations (auto-orientation et redimensionnement) d'images à l'ensemble des jeux de données afin de réduire le temps d'entraînement tout en augmentant la performance. La quatrième sous étape consiste à faire une augmentation des données (*augmentation*). Les transformations faites sont également les mêmes que l'architecture Faster R-CNN (la rotation, le retournement, la normalisation, le lissage ...). La dernière sous étape consiste à générer les jeux de données d'entraînement avec l'ensemble des transformations faites précédemment.

Une fois ces sous-étapes terminées, une version est créée, suivie d'un entraînement du modèle. L'entraînement du modèle prend plusieurs minutes dépendamment du volume des jeux de données. Lorsque l'entraînement est terminé, une notification est envoyée au mail de l'utilisateur afin de l'informer. Lorsque l'entraînement du modèle est fait, il est possible de faire une inférence sur les jeux de données tests.

Le modèle crée peut-être déployer sur plusieurs à savoir *NVIDIA Jetson, Luxonis OAK* voir sur les téléphones intelligents iOS.

- **Deuxième étape : *Google Colaboratory***

La deuxième étape consiste à effectuer des inférences sur les images en local. De nouveaux jeux de données personnalisés peuvent être utilisés à ce niveau afin de faire un entraînement sur le modèle préentraîné (*fine-tuning*). Il suffit d'installer YOLOv8, de définir le nombre d'époques et d'utiliser le modèle préentraîné en spécifiant sa version dans *roboflow*.

Le diagramme suivant (fig.19) résume les différentes étapes de mise en œuvre et pour l'entraînement du modèle dans roboflow.

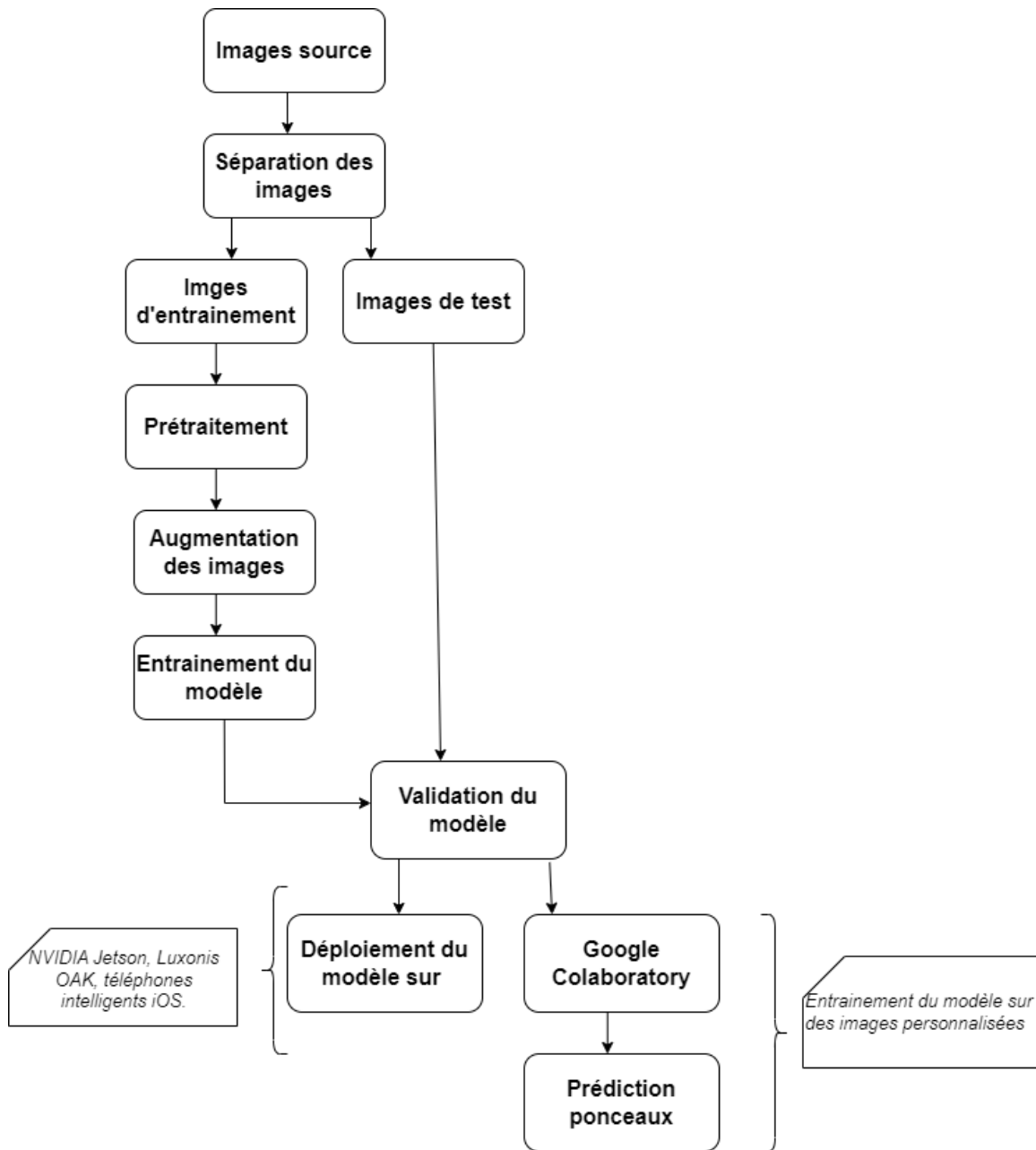


Figure 14: Les différentes étapes de mise en œuvre d'un modèle YOLOv8 sur roboflow

Le nombre d'époques sur *roboflow* est déterminé en fonction de la précision obtenue. Contrairement à l'entraînement dans *Google colaboratory* où l'utilisateur doit spécifier le nombre d'époques.

Le tableau 7 présente les précisions issues de l'entraînement sur *roboflow* et sur *Google colaboratory*.

Tableau 7 : Métriques obtenues après l'entraînement du modèle en fonction du nombre d'époques (architecture YOLOv8).

Plateforme	Nombre d'époques	mAP	Précision	F1-Score	recall
Roboflow	320	0.97	1	0.98	0.98
Google colab (fine-tuning)	125	0.98	1	0.97	0.96

La matrice de confusion issue du *fine-tuning* du modèle YOLOv8 est le suivant :

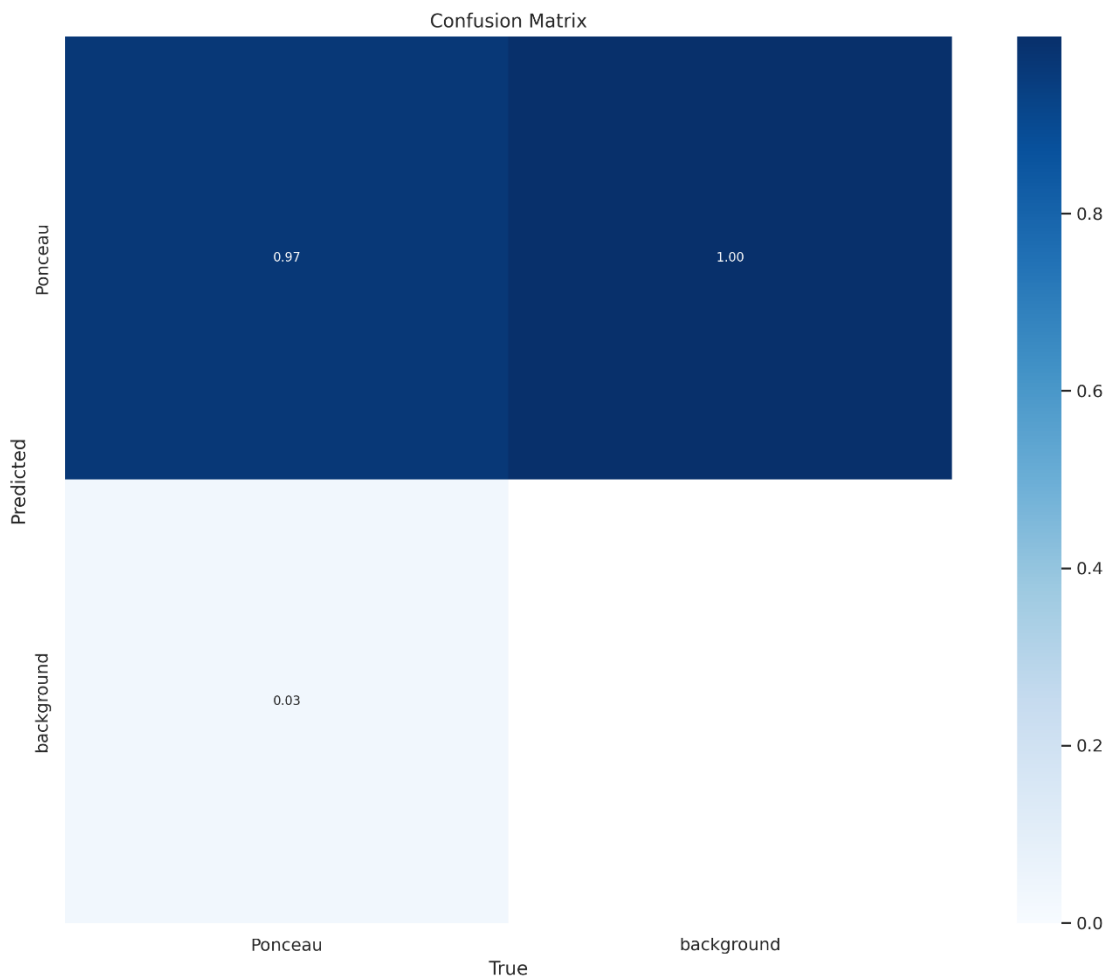


Figure 15: Matrice de confusion issue de la validation du modèle YOLOv8

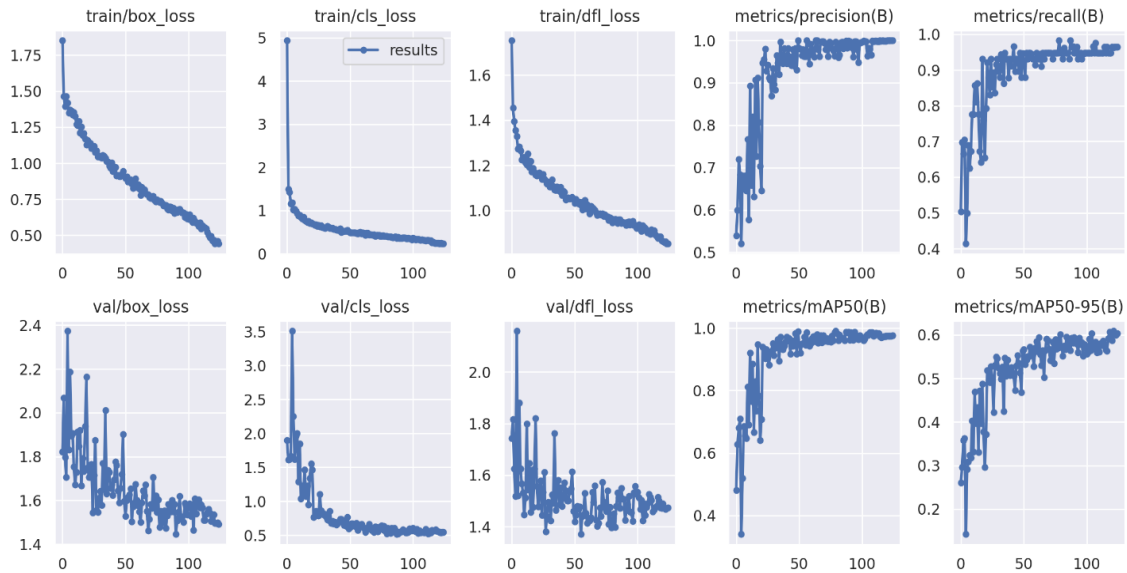


Figure 16: Résultats de l'entraînement du modèle YOLOv8.

4.1.2 Exemples de résultats de détection des ponceaux

L'entraînement des modèles a été fait sur les *imassettes* dont la résolution est de 512 x 512. Il est cependant impossible de faire une inférence sur les images d'origines (dont la résolution est de 5472 x 3648).

Un script Python a été développé pour reconstituer l'image après la détection à partir des imassettes. Les figures 22, 23, 24 montrent les résultats de la prédiction sur les imassettes et le résultat de l'inférence sur les images originales pour les deux modèles.

i. Faster R-CNN

Les résultats de Faster R-CNN sont présentés en fonction des époques. Le choix des images pour le modèle Faster R-CNN est aléatoire. Les *imassettes* qui ont été générés peuvent dans certains cas être différents (surtout à chaque entraînement).

Il faut noter que les *bounding box* de couleurs vertes sont la vérité de terrain, en rouge, c'est la prédiction avec la plus grande précision. Toutes les autres prédictions sont en bleu.

- 100 époques

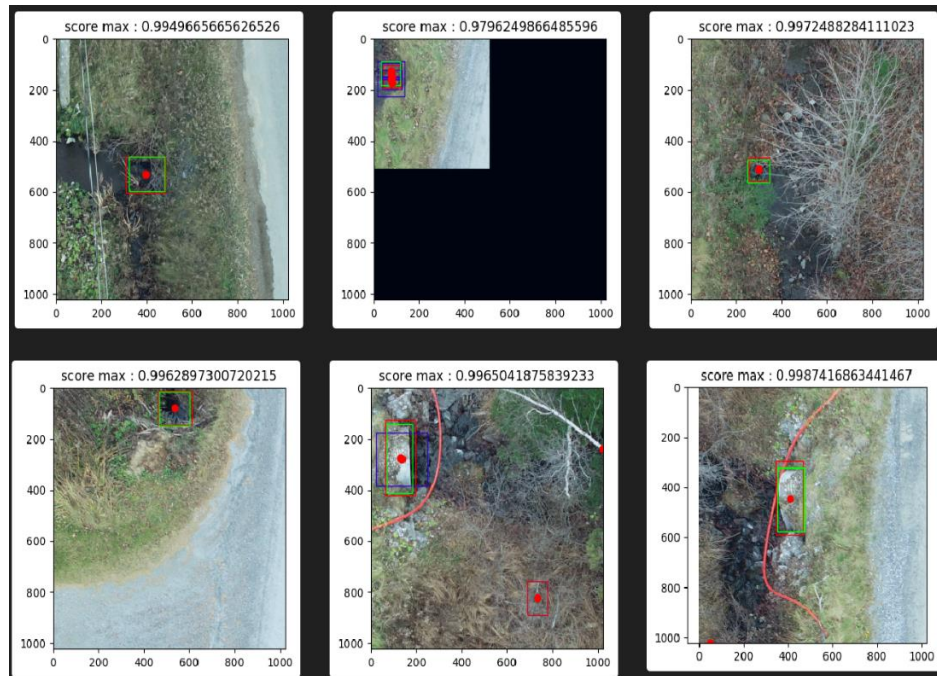


Figure 17: Extrait des résultats de détection des ponceaux du modèle Faster R-CNN pour 100 époques

Sur la figure 23, on peut remarquer que la plupart des ponceaux ont été très bien détectés. On note une fausse détection sur un des *imagettes* (deuxième ligne et deuxième colonne).

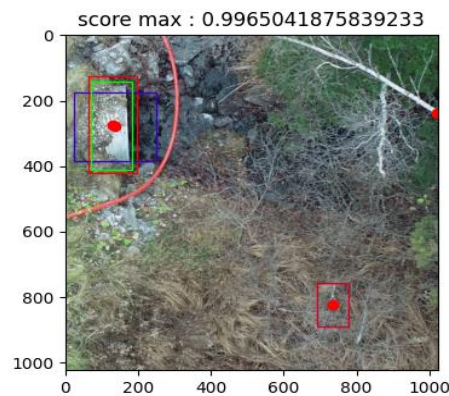


Figure 18: Exemples de fausse détection du modèle Faster R-CNN pour 100 époques

Bien que le ponceau soit très bien détecté, une fausse alarme a été détectée avec un haut score.

On peut remarquer sur la figure 24 que tous les ponceaux ont également été très bien détectés. Il existe également une fausse alarme de la figure 23 sur la figure 24 (première ligne et première colonne).

- 300 époques

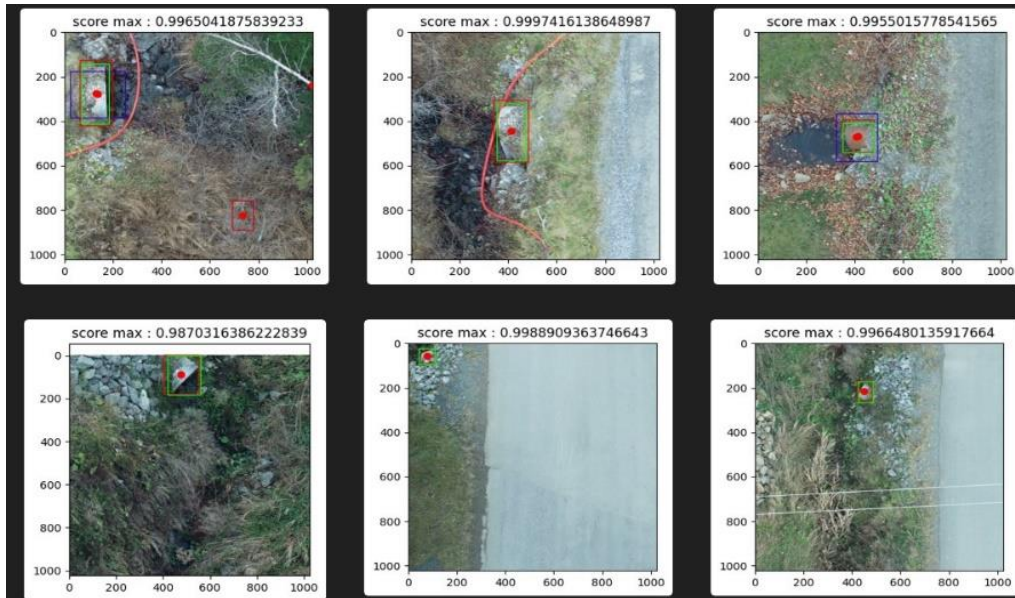


Figure 19: Extrait des résultats de détection des ponceaux du modèle Faster R-CNN pour 300 époques

Tout comme pour les deux autres époques, la détection des ponceaux a été bien faite pour les données de prédiction. Il existe également une fausse alarme pour la même imagerie que les précédentes époques.

ii. YOLOv8

La figure 25 montre le résultat de la détection des ponceaux avec le modèle YOLOv8. La détection a été correctement faite avec de très bons scores (80 % et 90 %). Bien que dans certaines situations le ponceau soit moins visible, la prédiction a été exacte. Par exemple, la première imagerie (première ligne et première colonne de la figure 25) n'a pas de ponceaux. C'est une imagerie qui a été ajoutée dans les données de test afin de mieux évaluer la prédiction.



Figure 20: Extrait des résultats de détection des ponceaux du modèle yoloV8

Une inférence a été effectuée sur deux images originales afin de s'assurer que le modèle fonctionne très bien sur nos données de base avec une résolution de 5472 x 3648.

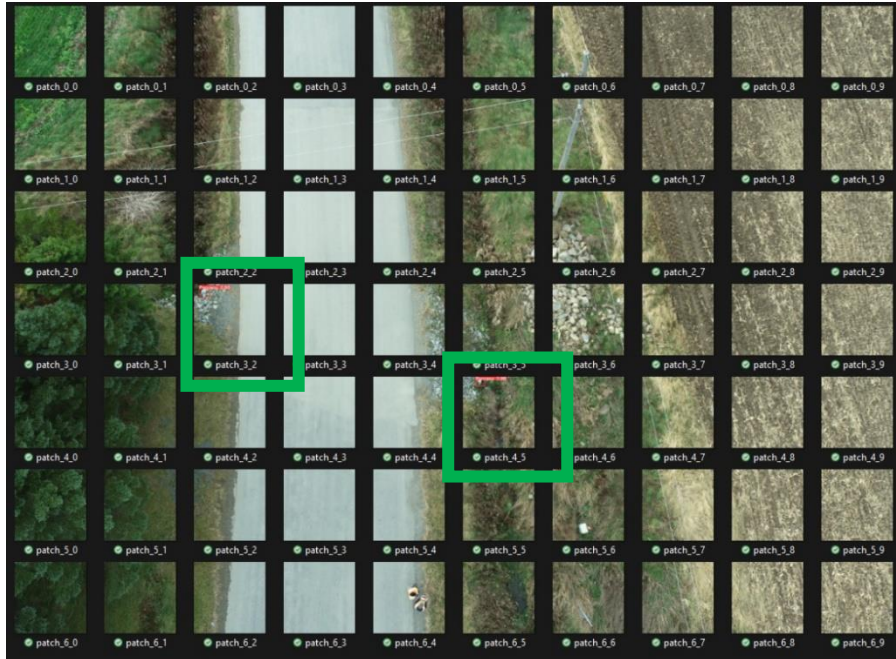


Figure 21: Inférence sur les données de base. Détection des ponceaux sur les imagettes

La détection a été exacte sur les imagettes comme on peut le voir sur la figure 27. Le score des deux ponceaux est de 0.86 pour le premier ponceau et 0.88 pour le second.

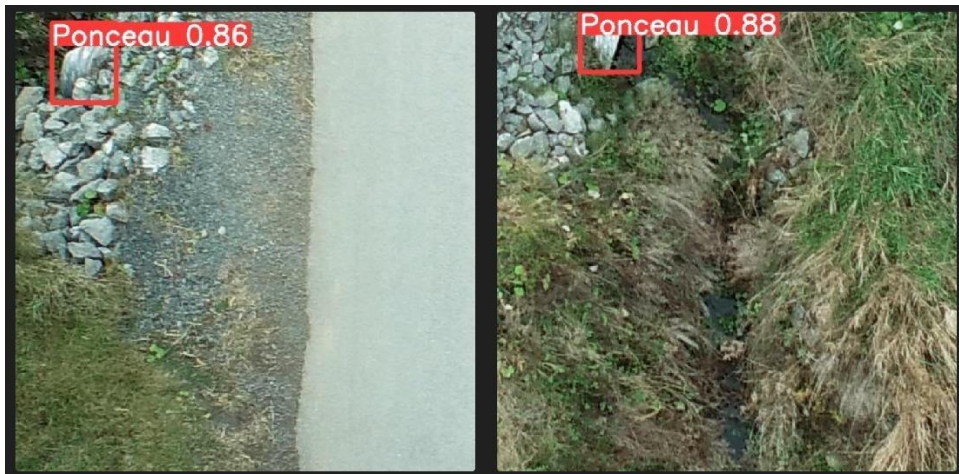


Figure 22: Résultat de l'inférence sur les imagettes

À partir de la détection sur les des imagettes (fig. 26) obtenues, une reconstitution de l'image originale a été réalisée. La figure 28 présente l'image reconstituée avec les ponceaux détectés, représentés par des boîtes de délimitation (*bounding box* en bleu). Vu la taille de l'image, il est difficile d'apercevoir les boîtes de détection.



Figure 23: Image reconstituée à partir de l'inférence sur les imagerie

Les flèches en rouge indiquent la position des boites de délimitation détectées autour des ponceaux.

Une deuxième inférence a été réalisée sur une image quelconque. La procédure est la même que l'image précédente : générer les *imagerie* afin d'avoir la même résolution. Ensuite, procéder à la reconstitution de l'image. La figure 29 présente le résultat de cette inférence.



Figure 24: Deuxième inférence sur une image teste d'une résolution de 5472 x 3648

4.2 Discussion

4.2.1 Performances des modèles

- Le modèle Faster R-CNN

Les figures 16 et 18 présentent la fonction de pertes d'entraînement sur l'ensemble des jeux de données. La perte d'entraînement est une mesure de la performance d'un modèle pendant la phase d'entraînement. Il s'agit de la moyenne des pertes de chaque échantillon d'entraînement du modèle.

On observe que ces pertes diminuent au fil des époques (fig. 16 et 18). La perte d'entraînement (*train_loss*) est plus faible lorsque le nombre d'époques est élevé. Il est de 0.11 et 0.07 pour 100 et 300 époques respectivement (tableau 6). Ces valeurs restent tout de même faibles, surtout pour 300 époques, ce qui indique que l'entraînement converge et que 300 époques semblent suffisantes.

Le F1-score ([3.3.4.1](#)) pendant la phase d'entraînement augmente en fonction du nombre d'époques (figures 15 et 17). Il est de 0.65 pour 100 époques et de 0.96 pour 300 époques (tableau 6). Ces valeurs indiquent que le modèle a une bonne performance en termes de

précision et de rappel. Bien que le score soit faible pour 100 époques, il est très bon pour 300 époques. À partir de 300 époques, le modèle parvient très bien à détecter les ponceaux.

Pour mieux évaluer le modèle, des images sans ponceaux (vrais négatifs) ont été ajoutées aux jeux de données tests afin de s'assurer que le modèle a gardé ses performances. Pour les images sans ponceaux, le modèle n'a rien détecté. Ces résultats indiquent que le modèle a correctement identifié les observations des classes positives (les ponceaux) et des classes négatives (les non-ponceaux), ce qui démontre sa haute précision. Il faut noter que le F1-score élevé indique que le modèle a une bonne performance en termes de précision et de rappel, tandis qu'un F1-score faible indique que le modèle peut rencontrer des difficultés à distinguer les ponceaux des non-ponceaux.

Il est possible d'améliorer le F1-score en augmentant le nombre d'époques pendant l'entraînement, par exemple, en entraînant le modèle jusqu'à 500 époques. Cependant, il est crucial de surveiller la possibilité de surapprentissage (*overfitting*).

Les signes courants de surapprentissage incluent une diminution de la précision de la validation, une précision d'entraînement élevée par rapport à la précision de validation. Pour l'instant, avec 100 époques, le F1-score obtenu lors de l'entraînement est relativement proche de celui obtenu lors de la validation (0.65 pour l'entraînement et 0.68 pour la validation); avec 300 époques, le F1-score est également les valeurs du F1-score sont également proches 0.96 pour l'entraînement et 1 pour la validation).

Si l'augmentation du nombre d'époques conduit à une situation de surapprentissage, il convient de réduire le nombre d'époques et de recommencer.

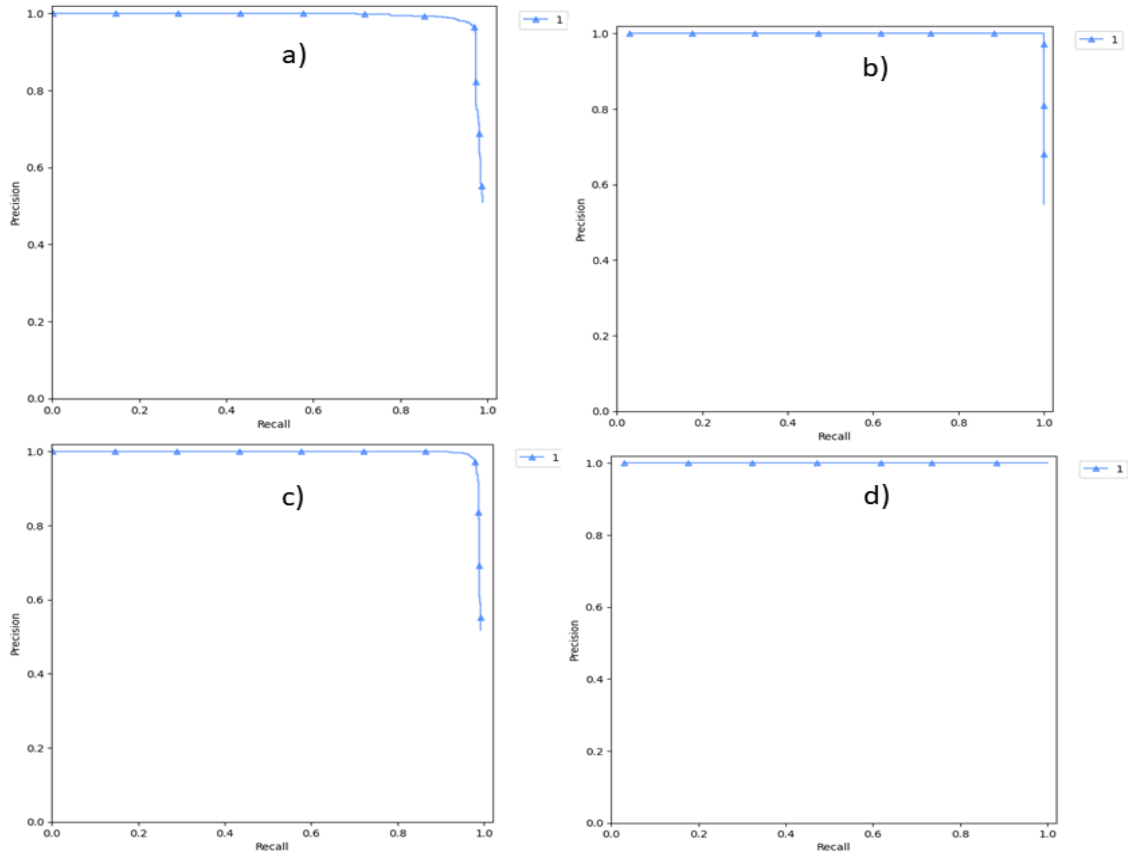


Figure 25: Courbe de Précision x Rappel du modèle Faster R-CNN pour les 100 et 300 époques sur l'ensemble des données d'entraînement et de validation. a) données d'entraînement (100 époques); b) données de validation (100 époques); c) données d'entraînement (300 époques); d) données de validation (300 époques).

La figure 30 présente les courbes de précision en fonction du rappel. On peut remarquer que, pour l'ensemble des jeux de données (entraînement ou validation), la précision reste élevée pendant que le rappel diminue.

Pour la suite du projet, le meilleur modèle des deux époques est à privilégier bien que sa durée d'entraînement soit longue. La durée d'entraînement pour 300 époques est de 16807 secondes soit 4.7 heures (tableau 4).

- Le modèle YOLOv8

La figure 21 présente les courbes des métriques issues de l'entraînement du modèle YOLOv8. L'entraînement a été effectué sur 125 époques. Sur cette figure, on peut observer que les valeurs des pertes d'entraînement (*train/cls_loss*) et de perte de validation (*val/cls_loss*) diminuent conjointement. Bien que les valeurs de la perte de validation restent relativement constantes à partir de 100 époques, elles demeurent tout de même faibles. De plus, la précision est au-dessus de 0.99 avec un mAP de 0.98, un rappel de 0.96 et un F1-score de 0.97. Cela signifie qu'il n'y a pas de surapprentissage.

La matrice de confusion (fig. 20) montre que 97 % des ponceaux prédits sont réellement des ponceaux, tandis que 3 % des non-ponceaux ont été classés comme étant des ponceaux. Cela peut s'expliquer par le manque de diversité des ponceaux. Pour remédier à cela, il est possible d'ajouter les *imagerie* mal détectés aux jeux de données d'entraînement, ce qui permettra au modèle de s'ajuster afin d'éviter ces fausses détections.

L'amélioration du modèle YOLOv8 peut être fait, mais en augmentant les jeux de données à partir de la plateforme *roboflow* et en générant une nouvelle version du modèle. Toutefois, cela n'est pas nécessaire étant donné que nous avons obtenu une excellente précision qui est de 100 % (tableau 7) sans avoir de surapprentissage.

L'un des avantages du modèle YOLOv8 est qu'il permet de faire une détection à partir d'une vidéo en temps réel. L'inférence du modèle à partir de la vidéo se fait de la même manière que sur les images. Il suffit juste de spécifier l'extension du fichier.

4.2.3 Discussion des résultats de la détection des ponceaux

Les résultats de la prédiction des deux modèles sont présentés dans la section [4.1.2](#). Les résultats montrent que, dans l'ensemble, les modèles détectent les ponceaux avec une bonne précision. Cependant, il existe des exceptions, c'est le cas de la figure 23 où une fausse détection est observée. En effet, afin de comprendre cette fausse détection, il a fallu observer l'image originale ayant servi à générer l'*imagerie*. On remarque que c'est une branche d'un arbre qui semble avoir la même texture qu'un ponceau en béton. Ce qui a engendré cette confusion.

Bien que les deux modèles aient détecté correctement les ponceaux sur les jeux de données tests, il est important de noter que certaines fausses détections (fig.31) sont dues à un manque de diversité des données.

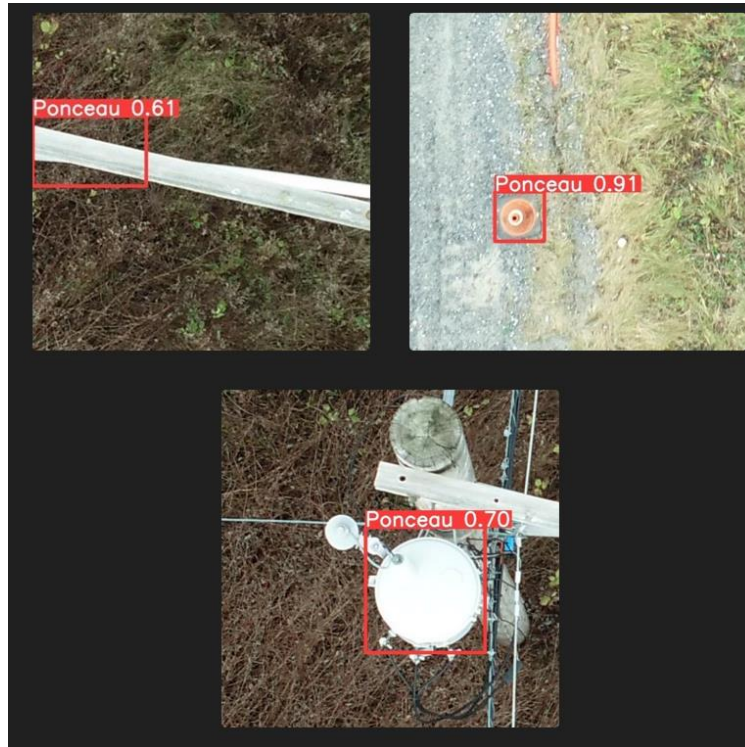


Figure 26: Fausses détections des ponceaux avec le modèle YOLOv8

La figure 31 met en évidence des fausses détections sur les images tests du modèle YOLOv8. On observe que certaines structures, telles que le fer, les objets en béton, sont détectées comme des ponceaux par le modèle. Cela s'explique par le manque de diversité et de quantité suffisante de données d'entraînement, ce qui rend difficile la distinction précise des ponceaux.

Étant donné que le modèle détecte presque tous les ponceaux (*Recall* élevé), supprimer les fausses détections est rapide. Il suffit d'ignorer ces *imagettes* pendant le post-traitement. Les deux modèles (YOLOv8 et Faster R-CNN (300 époques) ont des rappels respectivement de 96 % et 93 %. Ceci signifie que seuls 4 % (pour YOLOv8) ou 7 % (pour Faster R-CNN) des ponceaux n'ont pas réellement été identifiés.

Tableau 8 : résumé des métriques de l'entraînement des modèles Fast R-CNN et YOLOv8

Modèle	F1-score	Précision	Recall	Nombre d'époques	Temps d'exécution (heures)
Fast R-CNN (300 epochs)	96%	99%	93%	300	4.6
YOLOv8	98.48%	100%	97%	125	0.786

Le Tableau 8 présente le résumé des deux architectures. On constate que le modèle Fast R-CNN est très couteux en termes de temps d'exécution par rapport au modèle YOLOv8. Le temps d'entraînement du modèle Fast R-CNN est de 4.6 heures pour 300 époques contre 0.78 heures pour le modèle YOLOv8 pour 125 époques. Le rappel du modèle YOLOv8 est supérieur au modèle Fast R-CNN (97 % contre 93 %). Nguyen *et al.*(2021) ont effectué une comparaison des deux architectures. Bien que les auteurs aient comparé le modèle Faster R-CNN et la deuxième version de YOLO, ces résultats peuvent être considérés dans le cas de cet essai surtout que l'architecture YOLOv8 est plus performant que YOLOv2 (Terven et Cordova-Esparza, 2023). D'autres études telles que Munin *et al.*, (s.d.), Nepal et Eslamiat (2022), ont démontré que l'architecture du modèle YOLOV8 donne de meilleurs résultats en termes de performance que le Faster R-CNN.

Dans le cadre de ce projet, le modèle final sera utilisé pour générer une couche vectorielle indiquant l'emplacement de chaque ponton détecté. La table attributaire devra inclure le score de détection, les coordonnées X et Y, ainsi que, éventuellement, le nom de l'image d'origine et la pente dans laquelle se trouve le ponton. Cette approche vise à accélérer l'inspection des pontons.

4.2.4 Limites du projet et recommandations

Les deux modèles développés présentent des limites malgré leurs avantages. Tout d'abord, pour le modèle Faster R-CNN, la durée d'entraînement pour 300 époques du modèle est assez longue (tableau 6) et il est moins performant en termes de détection, comme indiqué dans la section précédente.

Le modèle YOLOv8 a été développé sur la plateforme *roboflow* qui est une plateforme propriétaire. Il n'est pas possible d'avoir accès aux ressources totales (GPU, mémoire RAM, etc.), surtout que le modèle a été entraîné avec un compte gratuit. Ce type de compte rend les données publiques contrairement à un compte payant. De plus, le nombre d'entraînements est limité à un seul par jour, or, il faudrait faire plusieurs tests afin de choisir le meilleur modèle. Il n'est également pas possible d'augmenter les données plus de trois fois par rapport à sa quantité initiale. Enfin, avec un compte gratuit, il n'est pas possible de faire des entraînements précis (*Accurate Train*).

La collecte de terrain n'a permis d'acquérir que 150 images., ce qui est insuffisant pour entraîner un CNN. Même avec une augmentation des données, cela demeure insuffisant.

La forme des ponceaux n'a pas été prise en compte pendant la formation des deux modèles. Il faudrait en tenir compte afin d'améliorer leur performance. Aussi, plusieurs caractéristiques peuvent être prises en compte. Par exemple, tenir compte de l'environnement dans lequel se trouvent les ponceaux : la présence d'eau, la présence de routes, etc.

De plus les images acquises ne sont pas géoréférencées. Il est impossible de générer une couche vectorielle afin de comparer avec la couche issue de l'inventaire qui a permis de planifier la collecte de terrain.

La résolution des images d'entraînement est très importante pendant l'entraînement du modèle. La création des *imassettes* est certes une alternative, mais cela a un impact sur la qualité de l'information. En effet, les *imassettes* sont tout simplement un découpage de l'image de base en petites tuiles afin d'avoir une dimension adaptée à la formation du modèle, ce qui entraîne une perte de la qualité de l'information

Afin que le système soit totalement opérationnel, il serait nécessaire de disposer de nouveaux jeux de données pour des tests afin d'évaluer la robustesse du modèle. Il serait intéressant que les jeux de données proviennent d'une zone différente de celles utilisées pour l'entraînement.

Enfin, étant donné le temps alloué à cet essai, il n'est pas possible de réaliser une évaluation économique du rendement du projet, notamment en termes de gain en temps et de l'impact financier potentiel pour la compagnie Maxxum Gestion d'Actifs.

5. Conclusion

Le développement d'un outil de détection automatique des ponceaux revêt une grande importance pour la surveillance des infrastructures routières. Les ponceaux sont des éléments clés dans les systèmes de drainage et leur maintenance régulière est essentielle pour assurer la sécurité des usagers de la route.

L'utilisation de drones pour la surveillance des ponceaux offre une couverture plus large et plus précise que les méthodes traditionnelles. L'acquisition des images de drone des ponceaux existants a permis de constituer une base de données importante pour le développement du modèle de détection d'objets. Le succès du modèle dépend en grande partie de la qualité et de la quantité des données d'entraînement disponibles.

Les modèles de détections des ponceaux développés par les méthodes CNN ont permis d'identifier les ponceaux de manière précise et automatisée à partir des images de drone. Cette méthode permet une analyse plus rapide et plus efficace des données, réduisant ainsi les coûts et le temps nécessaire à l'inspection manuelle des ponceaux. Cette innovation contribuera à améliorer les méthodes de collecte des données de Maxxum Gestion d'Actif dans le cadre de ses projets des plans d'intervention d'infrastructures routières.

6 Références

- Akhatar, M. (2023) Deep Learning approach applied to drone imagery for the automatic detection of forest fire.
- Alan, M. (1950) Turing. Computing Machinery and Intelligence. Mind LIX, 2236: 433-460. Oxford University Press, Oxford, United Kingdom.
- ArcGIS Field Maps (2022)Esri.com.
- ARSENAULT, S. et GAGNON, A. (2004) Systeme d'inspection des ponceaux. 2004 CONGRES ET EXPOSITION ANNUELS DE L'ASSOCIATION DES TRANSPORTS DU CANADA.
- Babayan, P. V., Ershov, M. D. et Erokhin, D. Y. (2019) Neural network-based vehicle and pedestrian detection for video analysis system. 2019 8th Mediterranean Conference on Embedded Computing (MECO), p. 1-5. IEEE.
- Chang, Y.-L., Tan, T.-H., Lee, W.-H., Chang, L., Chen, Y.-N., Fan, K.-C. et Alkhaleefah, M. (2022) Consolidated convolutional neural network for hyperspectral image classification. Remote Sensing, vol. 14, n°7, p. 1571.
- Chen, H. et Cao, P. (2019) Deep learning based data augmentation and classification for limited medical data learning. 2019 IEEE international conference on power, intelligent computing and systems (ICPICS), p. 300-303. IEEE.
- Chen, Y.-N., Fan, K.-C., Chang, Y.-L. et Moriyama, T. (2023) Special Issue Review: Artificial Intelligence and Machine Learning Applications in Remote Sensing. Remote Sensing (Vol. 15, p. 569). Multidisciplinary Digital Publishing Institute.
- Chollet, F. (2021) Deep learning with Python. Simon and Schuster.
- Cooper, S. B. et Van Leeuwen, J. (2013) Alan Turing: His work and impact. Elsevier.

- Cornioley, P. (2018) Intégration d'un module d'apprentissage profond dans l'architecture logicielle d'un SIG Web. Mémoire présenté pour l'obtention du grade de Maître en science, Université de Sherbrooke.
- Cottrell, G. W. (2006) New life for neural networks. *Science*, vol. 313, n°5786, p. 454-455.
- Dan, A., Oosterbaan, J. et Jamet, P. (2002) Contribution des reseaux de neurones artificiels (RNA) a la caracterisation des pollutions de sol. Exemples des pollutions en hydrocarbures aromatiques polycycliques (HAP). *Comptes Rendus Geoscience*, vol. 334, n°13, p. 957-965.
- Diaby, I., Germain, M. et Goïta, K. (2021) Evidential data fusion for characterization of pavement surface conditions during winter using a multi-sensor approach. *Sensors*, vol. 21, n°24, p. 8218.
- Dong, S., Wang, P. et Abbas, K. (2021) A survey on deep learning and its applications. *Computer Science Review*, vol. 40, p. 100379.
- Dortier, J.-F. (2014) L'intelligence artificielle. Espoirs et réalisations. Le cerveau et la pensée (Vol. 2, p. 102-112). Éditions Sciences Humaines.
- Drew, P. J. et Monson, J. R. T. (2000) Artificial neural networks. *Surgery*, vol. 127, n°1, p. 3-11.
- Dwyer, B. et Nelson, J. (2022) Roboflow (version 1.0). URL <https://roboflow.com>.
- Elshawi, R., Wahab, A., Barnawi, A. et Sakr, S. (2021) DLBench: a comprehensive experimental evaluation of deep learning frameworks. *Cluster Computing*, vol. 24, p. 2017-2038.

Fukushima, K. (1980) Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, vol. 36, n°4, p. 193-202.

George (2015) Wotton. *Histoire du Québec*.

Ghoury, S., Sungur, C. et Durdu, A. (2019) Real-time diseases detection of grape and grape leaves using faster r-cnn and ssd mobilenet architectures. *International conference on advanced technologies, computer engineering and science (ICATCES 2019)*, p. 39-44.

Girshick, R., Donahue, J., Darrell, T. et Malik, J. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 580-587.

Google Colab (2019) [Google.com](https://colab.research.google.com/).

Gouvernement du Québec (2019) *Classification fonctionnelle du réseau routier*. Gouv.qc.ca.

He, K., Gkioxari, G., Dollár, P. et Girshick, R. (2017) Mask r-cnn. *Proceedings of the IEEE international conference on computer vision*, p. 2961-2969.

Hinton, G. E., Osindero, S. et Teh, Y.-W. (2006) A fast learning algorithm for deep belief nets. *Neural computation*, vol. 18, n°7, p. 1527-1554.

Hou, L., Samaras, D., Kurc, T. M., Gao, Y., Davis, J. E. et Saltz, J. H. (2016) Patch-based convolutional neural network for whole slide tissue image classification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 2424-2433.

HOURRANE, O. (2018) *Réseaux neuronaux récurrents et LSTM*. Datasciencetoday.net.

Huang, T., Yang, G. et Tang, G. (1979) A fast two-dimensional median filtering algorithm. *IEEE transactions on acoustics, speech, and signal processing*, vol. 27, n°1, p. 13-18.

- Janiesch, C., Zschech, P. et Heinrich, K. (2021) Machine learning and deep learning. *Electronic Markets*, vol. 31, n°3, p. 685-695.
- Jordan, M. I. et Mitchell, T. M. (2015) Machine learning: Trends, perspectives, and prospects. *Science*, vol. 349, n°6245, p. 255-260.
- Karas, A. (2023) Télédétection spatiale de dépôts d'avalanche pour le suivi de zones à risque avalancheux.
- Kaur, P., Khehra, B. S. et Mavi, E. B. S. (2021) Data augmentation for object detection: A review. *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, p. 537-543. IEEE.
- Kim, S., Wimmer, H. et Kim, J. (2022) Analysis of Deep Learning Libraries: Keras, PyTorch, and MXnet. *2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA)*, p. 54-62. IEEE.
- Kiyindou, A. (2021) Intelligence artificielle et société africaine: pour un développement intelligent. *TERRITOIRES INTELLIGENTS ET SOCIÉTÉS APPRENANTES*, p. 12.
- Kpegouni, G. et Bouroubi, Y. (2022) Applications des techniques d'apprentissage profond aux images satellites THR pour l'aide au recensement de population en Afrique. Thèse de doctorat, Université de Sherbrooke.
- Kpegouni, G., Bouroubi, Y., Coulombe, H., Echevin, D., Lauzier-Hudon, E. et Germain, M. (2023) Application of deep-learning techniques to very-high-resolution satellite images supporting population censuses in developing countries. *Journal of Applied Remote Sensing*, vol. 17, n°2, p. 024506-024506.
- Larocque, V. (2020) Méthode de conception de traverses à gué aménagées pour des cours d'eau en milieu forestier. Thèse de doctorat, Université Laval.

- LeCun, Y., Bengio, Y. et Hinton, G. (2015) Deep learning. *nature*, vol. 521, n°7553, p. 436-444.
- Lee, C.-J. et Wang, S.-D. (1999) Fingerprint feature extraction using Gabor filters. *Electronics Letters*.
- Lessard, F., Jutras, S., Perreault, N. et Guilbert, É. (2022) Performance of automated geoprocessing methods for culvert detection in remote Forest environments. *Canadian Water Resources Journal/Revue canadienne des ressources hydriques*, p. 1-10.
- Lin-Kwong-Chon, C., Todoroff, P., Le Mezo, L., Benne, M. et Amangoua Kadjo, J. J. (2023) Multi-level deep-based classification of land use and land cover: a case study on Réunion Island. *ISSCT*.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. et Berg, A. C. (2016) Ssd: Single shot multibox detector. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, p. 21-37. Springer.
- Mahendru, M. et Dubey, S. K. (2021) Real time object detection with audio feedback using Yolo vs. Yolo_v3. *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, p. 734-740. IEEE.
- Make Sense (2023) [Makesense.ai](https://makesense.ai).
- Mäkinen, V., Oksanen, J. et Sarjakoski, T. (2019) Automatic determination of stream networks from DEMs by using road network data to locate culverts. *International Journal of Geographical Information Science*, vol. 33, n°2, p. 291-313.
- Marques, E. (2023) Étude à fine échelle de l’îlot de chaleur urbain par modélisation bayésienne à partir de données opportunes.

- Ministère des Affaires municipales et de l'Habitation (MAMH) (2023) Répertoire des municipalités.
- Mohammadi, P., Rashidi, A., Malekzadeh, M. et Tiwari, S. (2023) Evaluating various machine learning algorithms for automated inspection of culverts. *Engineering Analysis with Boundary Elements*, vol. 148, p. 366-375.
- MTQ, M. des T. du Q. (2019) Manuel d'inspection des ponceaux. Gouv.qc.ca.
- Munawar, H. S., Aggarwal, R., Qadir, Z., Khan, S. I., Kouzani, A. Z. et Mahmud, M. P. (2021) A gabor filter-based protocol for automated image-based building detection. *Buildings*, vol. 11, n°7, p. 302.
- Munawar, H. S., Qayyum, S., Ullah, F. et Sepasgozar, S. (2020) Big data and its applications in smart real estate and the disaster management life cycle: A systematic analysis. *Big Data and Cognitive Computing*, vol. 4, n°2, p. 4.
- Munin, A., Folarin, A., Munin-Doce, A., Alonso-Garcia, L., Diaz-Casas, V., Ferreno-Gonzalez, S. et Ciriano-Palacios, J. M. (s.d.) Real Time Vessel Detection Model Using Deep Learning Algorithms for Controlling a Barrier System. Available at SSRN 4423353.
- Naïma Humblot (2021) Deep learning : comprendre les réseaux de neurones artificiels (artificial neural networks) - Ryax Technologies. Ryax Technologies.
- Najafi, M. et Bhattachar, D. V. (2011) Development of a culvert inventory and inspection framework for asset management of road structures. *Journal of King Saud University-Science*, vol. 23, n°3, p. 243-254.
- Nepal, U. et Eslamiat, H. (2022) Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs. *Sensors*, vol. 22, n°2, p. 464.

- Nguyen, H. H., Ta, T. N., Nguyen, N. C., Pham, H. M. et Nguyen, D. M. (2021) Yolo based real-time human detection for smart video surveillance at the edge. 2020 IEEE Eighth International Conference on Communications and Electronics (ICCE), p. 439-444. IEEE.
- OpenMMLab (2023a) Dive into YOLOv8: How does this state-of-the-art model work? Medium. Medium.
- OpenMMLab (2023b) Dive into YOLOv8: How does this state-of-the-art model work? Medium. Medium.
- Paquette, M. et Association québécoise des transports, Aqt. (2015) La gestion des ponceaux au ministère des Transports du Québec. Aqtr.com.
- Paradis-Lacombe, P. (2018) Caractérisation de l'état et de la durabilité des traverses de cours d'eau sur les chemins forestiers. Thèse de doctorat, Université Laval.
- Proia, N. et Pagé, V. (2009) Caractérisation d'une méthode Bayésienne de détection de bateaux dans des images satellites optiques. ORASIS'09-Congrès des jeunes chercheurs en vision par ordinateur.
- Rao, M. N. (2023) A Comparative Analysis of Deep Learning Frameworks and Libraries. International Journal of Intelligent Systems and Applications in Engineering, vol. 11, n°2s, p. 337-342.
- Redmon, J., Divvala, S., Girshick, R. et Farhadi, A. (2016) You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, p. 779-788.
- Ren, S., He, K., Girshick, R. et Sun, J. (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, vol. 28.

- Rougier, N. P. (2021) Intelligence artificielle et singularité.
- Russell, S. J. (2010) Artificial intelligence a modern approach. Pearson Education, Inc.
- S. K, A. K. Srivastava, S. Allam, et D. Lilaramani (2021) Comparative analysis on Deep Convolution Neural Network models using Pytorch and OpenCV DNN frameworks for identifying optimum fruit detection solution on RISC-V architecture. 2021 IEEE Mysore Sub Section International Conference (MysuruCon), p. 738-743.
- Sabo, N., Foucher, S., Lauzier-Hudon, E. et Bouroubi, Y. (2022) Adaptation des architectures ADDA et semi-ADDA pour la détection d'objets par apprentissage profond sur les images satellites THR. Thèse de doctorat, Université de Sherbrooke.
- Sadiq, R., Rodriguez, M. J. et Mian, H. R. (2019) Empirical Models to Predict Disinfection By-Products (DBPs) in Drinking Water: An Updated Review☆. *In* J. Nriagu (dir.), Encyclopedia of Environmental Health (Second Edition) (p. 324-338). Oxford : Elsevier.
- Schmitt, A., Le Blanc, B., Corsini, M.-M., Lafond, C. et Brůžek, J. (2001) Les réseaux de neurones artificiels. Un outil de traitement de données prometteur pour l'anthropologie. *Bulletins et mémoires de la Société d'Anthropologie de Paris. BMSAP*, vol. 13, n°13 (1-2).
- Shinde, P. P. et Shah, S. (2018) A review of machine learning and deep learning applications. 2018 Fourth international conference on computing communication control and automation (ICCUBEA), p. 1-6. IEEE.
- Simon, W. (2022) List of Open-Source Annotation Tools for Machine Learning Research. Simonwenkel.com.
- Skalski, P. (2019) Make Sense. [Github.io](https://github.com/skalski-io).

- Solawetz, J. (2020) Why and How to Implement Random Rotate Data Augmentation. Roboflow Blog. Roboflow Blog.
- Tabti, S. (2016) Modélisation des images par patchs pour leur restauration et leur interprétation. Applications à l'imagerie SAR. Thèse de doctorat, Télécom ParisTech.
- Terven, J. et Cordova-Esparza, D. (2023) A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond. arXiv preprint arXiv:2304.00501.
- Thambawita, V., Strümke, I., Hicks, S. A., Halvorsen, P., Parasa, S. et Riegler, M. A. (2021) Impact of image resolution on deep learning performance in endoscopy image classification: An experimental study using a large dataset of endoscopic images. *Diagnostics*, vol. 11, n°12, p. 2183.
- Transports Canada (2023) Utiliser votre drone de façon sécuritaire et légale. Transports Canada.
- Twum, F., Missah, Y. M., Oppong, S. O. et Ussiph, N. (2022) Textural Analysis for Medicinal Plants Identification Using Log Gabor Filters. *IEEE Access*, vol. 10, p. 83204-83220.
- Ultralytics YOLOv8 Docs (2015)Ultralytics.com.
- Vélo Québec et Association des réseaux cyclables du Québec, A. (2012) Fiche d'entretien des ponceaux.
- Wira, P. (2009) Réseaux de neurones artificiels: architectures et applications. Cours en ligne, Université de Haute-Alsace.
- Xiao, Y., Tian, Z., Yu, J., Zhang, Y., Liu, S., Du, S. et Lan, X. (2020) A review of object detection based on deep learning. *Multimedia Tools and Applications*, vol. 79, p. 23729-23791.

Yadav, P. K., Thomasson, J. A., Hardin, R., Searcy, S. W., Braga-Neto, U., Popescu, S. C., Martin, D. E., Rodriguez, R., Meza, K. et Enciso, J. (2023) Detecting volunteer cotton plants in a corn field with deep learning on UAV remote-sensing imagery. *Computers and Electronics in Agriculture*, vol. 204, p. 107551.

YAPICI, M. M. et Topaloğlu, N. (2021) Performance comparison of deep learning frameworks. *Computers and Informatics*, vol. 1, n°1, p. 1-11.

YOLOv8 Ultralytics: State-of-the-Art YOLO Models (2023) LearnOpenCV – Learn OpenCV, PyTorch, Keras, Tensorflow with examples and tutorials.

Zhao, Z.-Q., Zheng, P., Xu, S. et Wu, X. (2019) Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, vol. 30, n°11, p. 3212-3232.