

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

ESSAI

SEGMENTATION SÉMANTIQUE EN TEMPS RÉEL À PARTIR D'UN NANO ORDINATEUR : ÉTUDE DES PERFORMANCES ET DES LIMITES

*Essai présenté pour l'obtention du grade de Maître en sciences (M.Sc.),
cheminement géodéveloppement durable*

VINCENT LE FALHER

LONGUEUIL
SEPTEMBRE 2020

Remerciements

Je tiens à remercier ...

□ TODO

Table des matières

Liste des figures	3
Liste des tableaux	3
1 Suivi des changements au rapport	1
2 Introduction	2
2.1 Mise en contexte	2
2.2 Problématique	3
2.3 Objectifs	4
3 Cadre théorique (état des connaissances, revue de la littérature)	4
3.1 Cadre théorique au sujet du nano ordinateur	4
3.2 Cadre théorique au sujet de l'apprentissage profond et de la segmentation sémantique	5
4 Matériel et méthodes	5
4.1 Site d'étude	5
4.2 Modèles et jeux de données	9
4.3 Matériel et logiciels	13
4.4 Méthodologie	14
4.5 Recherche	17
4.5.1 Revue de littérature	17
4.5.2 Étude du site d'implantation	17
4.5.3 Sélection des données et des modèles de réseaux de neurones	18
4.5.4 Choix de l'équipement pour le nano ordinateur	18
4.5.5 Exploration des solutions logicielles	19
4.6 Environnement de travail	19
4.6.1 Préparation du nano ordinateur	19
4.6.2 Collecte des données	27
4.6.3 Mise en place des solutions logicielles	27
4.7 Évaluation	30
4.7.1 Stratégie de test	30
4.7.2 Stratégie de collecte des indicateurs de performance	31
4.7.3 Segmentation avec des images	32
4.7.4 Segmentation avec des vidéos	32
4.8 Adaptation	32
4.8.1 Choix du modèle de l'architecture FCN	33
4.8.2 Adaptation du modèle	33
4.8.3 Ré-entraînement du modèle	33
5 Résultats	34
6 Interprétation et discussion des résultats	34

7 Conclusion et recommandations	34
8 Annexes	38

Liste des figures

1 Vue aérienne du pont Jacques-Cartier	6
2 Vue aérienne du pont Jacques-Cartier	7
3 Carte du site d'implantation : le pont Jacques-Cartier et la piste multifonctionnelle en orange sur le pont	8
4 Schéma de la configuration de la piste multifonctionnelle	9
5 Photo de la carte mère Jetson Nano de NVIDIA, représenté avec des Lego pour démontrer sa petitesse	13
6 Organigramme de la méthodologie à haut niveau	14
7 Organigramme de la méthodologie pour évaluer les performances	15
8 Organigramme des détails de la méthodologie pour évaluer les performances	15
9 Organigramme de la méthodologie pour évaluer les performances après une phase d'adaptation théorique	15
10 Organigramme de la méthodologie pour évaluer les performances après une phase d'adaptation réaliste	16
11 Organigramme des détails de la méthodologie pour évaluer les performances après une phase d'adaptation	16
12 Préparation du nano ordinateur	19
13 Montage du nano ordinateur	19
14 Carte mère du nano ordinateur	20
15 Adaptateur 5 volts 4 amps	20
16 Boitier pour le nano ordinateur	21
17 Vue arrière du boitier pour le nano ordinateur	22
18 Disque SSD NVMe M.2 interne 250GB	22
19 Caméra	23
20 Ventilateur	24
21 Hub USB 3.0 externe autoalimenté	25
22 Éléments pour l'évaluation des performances	30
23 Méthodologie du traitement et adaptation	33

Liste des tableaux

1 Suivi des changements	1
2 Tableau des données	11
3 Comparaison des performances du "data read" entre un SDD M.2 NVMe et une micro-sd	34

1 Suivi des changements au rapport

Cette section n'est disponible que pendant la rédaction du rapport. Les changements au rapport sont documentés dans ce tableau. Cela permet de faire un suivi et permet d'aider les personnes qui révisent et commentent le rapport.

Les changements les plus récents apparaissent en premier.

TABLE 1 – Suivi des changements

Version	Date	Changement
v1.1	18 juillet 2020	<ul style="list-style-type: none">— Ajout d'une section changelog pour documenter le suivi des changement et faciliter la révision.— Apport de clarification dans la section Matériel & logiciels (hdparm, présentation SDK, etc).— Début de la rédaction de la section résultat ; inclusion d'un tableau des performances data read SSD vs micro-sd.
v1.0	17 juillet 2020	Première version envoyée à Michaël G.

2 Introduction

2.1 Mise en contexte

La compagnie LES PONTS JACQUES CARTIER ET CHAMPLAIN INCORPORÉE (PJCCI) désire évaluer la mise en service de la piste multifonctionnelle (vélos, piétons, etc.) du pont Jacques-Cartier, à Montréal, durant l'hiver. Pour ce faire, la piste doit rester sécuritaire et dégagée, malgré les évènements météorologiques.

L'université de Sherbrooke, qui participe à cette initiative, propose de mettre en place sur le pont une plateforme de détection innovatrice qui consiste à installer plusieurs paires d'objets connectés ultralégers et performants (des nanoordinateurs) à différents endroits du pont. Chacun de ces nanoordinateurs possède trois différents types de capteurs : vision, son, et météorologiques (température, humidité, etc.). Chaque nanoordinateur d'une paire perçoit le même environnement, mais d'une perspective différente que son homologue : la caméra pointe vers la même surface, mais d'un autre point de vue ; les sons et les données météorologiques sont captés dans le même voisinage. Les données collectées par les capteurs sont traitées en temps réel par des algorithmes de détections performants qui sont adaptés à ce type de problématiques : les réseaux de neurones, du domaine de l'intelligence artificielle. La déduction de l'état de la surface de la piste (sèche, mouillée, glacée, etc.) se fait en fusionnant les différentes perceptions (multicibles) de chaque capteur (multicapteurs).

L'objet principal de cet essai consiste à étudier la capacité du nanoordinateur du fabricant NVIDIA, le Jetson nano [17], à exécuter, en temps réel, un modèle de réseau de neurones entraîné à faire de la segmentation sémantique (classification) d'images de haute résolution qui sont perçues avec la caméra. Les résultats de cette étude permettront de déterminer le modèle de réseau de neurones le plus adapté pour répondre aux besoins du volet vision du projet pour PJCCI.

La détection d'objets et de surface en temps réel est de plus en plus précise et efficace depuis que les performances des systèmes informatisés permettent l'exécution d'algorithmes exigeants, en majeure partie depuis l'utilisation des processeurs graphiques "GPU" [5] [7] [2] [11] [[jia_real-time_2020](#)] [13]).

Les systèmes informatiques performants sont de plus en plus miniatures, on parle de nanoordinateurs et des objets connectés ("Internet of Things" ou "IoT") [4] [22]. Ils permettent la détection en temps réel à des endroits, dans des situations et dans des conditions qui n'étaient pas envisageables il y a encore 10 ans ([\[jia_real-time_2020\]](#) [3] [1] [4]).

Les réseaux de neurones ont aussi très rapidement progressé depuis 2012 [2], permettant d'offrir des alternatives aux solutions de détection et de classifications, entre autres [19]. Les réseaux de neurones convolutifs entiers ("FCN" en anglais, pour "Fully Convolutional Network") sont les derniers à avoir émergé ("state-of-art") [[jia_real-time_2020](#)] et à profiter au domaine de la vision et de la détection d'objets ([16] [[jia_real-time_2020](#)]).

La segmentation sémantique est une forme de classification d'image, pixel par pixel, qui tire profit des dernières évolutions de la classification supervisée grâce aux réseaux de neurones convolutifs entiers, et se permet d'être déduite en temps réel avec des nanoordinateurs ([14] [4]). Les images doivent être de très haute résolution, ce qui nécessite d'avoir à disposition un système informatique capable de fournir une puissance de calcul appropriée, particulièrement pour la manipulation de la mémoire et des nombres flottants pendant l'inférence [15]. Leur application par des nanoordinateurs est un défi en raison de la faible consommation d'énergie (Watts) et de la

puissance de calcul limité de ces derniers [6].

Pour PJCCI, les avantages d'une telle plateforme seraient multiples, et on peut en énumérer plusieurs, sans se limiter à : contrôler l'épandage de sel ; surveiller les conditions de la piste multifonctionnelle ; suivre les effets du gel et du dégel ; optimiser les couts des opérations d'entretien (déplacements, quantité) ; offrir aux usagers des conditions d'accès sécurisées et optimales même en hiver ; effets environnementaux atténus ; détecter ce qu'un spécialiste humain ne pourrait pas ou aurait des difficultés à détecter ; prise de décision et gestion proactive ; planification.

D'un autre côté, les défis ne sont pas à sous-évaluer : la détection doit être précise, fiable et consistante, tout cela afin d'assurer aux usagers un service de qualité dans un contexte sécuritaire.

2.2 Problématique

Dans le cadre du projet pour PJCCI, une plateforme technologique sera mise à la disposition des gestionnaires du pont afin de les aider à prendre les décisions les plus responsables et raisonnables possibles. Mais la mise en opération d'une solution innovante et fiable, qui concilie des algorithmes d'apprentissage profond, du temps réel, des nano ordinateurs, et des conditions climatiques variables, est complexe. Dans une certaine mesure l'essai va contribuer à la recherche de solutions afin de répondre au défi pour le domaine du transport actif et durable d'être soutenu par des solutions technologiques fiables (opérationnelles), l'objectif étant de pouvoir offrir des services de qualité et sécuritaires sur l'ensemble des quatre saisons.

La seconde problématique que l'essai va contribuer à résoudre concerne les limites d'un nano ordinateur. Un nano ordinateur est un ordinateur miniaturisé en taille, mais aussi limité en capacité. Il existe différents fabricants et modèles, de spécifications variées, pour répondre à différents besoins. Le dernier né est le modèle "Jetson nano" du fabricant "NVIDIA ", disponible depuis juin 2019 au prix très abordable de 99\$US, et qui sera le matériel utilisé dans le cadre de cet essai. La compagnie NVIDIA a conçu ce matériel spécialement pour l'inférence de modèles d'apprentissage profond sur une plateforme mobile (drone) ou proche des données ("edge" en anglais). L'inférence nécessite une architecture et une puissance machine différente de celle nécessaire pour l'entraînement. Les modèles de réseaux de neurones sont adaptés et optimisés pour l'inférence. L'essai va permettre de préciser les capacités du Jetson nano pour l'inférence de diverses architectures de réseaux de neurones convolutifs entiers (FCN en anglais) et la segmentation sémantique en temps réel avec des vidéos de différentes propriétés (résolutions et nombre d'images par seconde). Il existe des tests encourageants ([18] [16] [5]), qui seront utilisés comme modèle, même si ceux-ci sont limités à des types d'application qui ne sont pas les mêmes que pour l'essai.

Il est difficile de trouver des jeux de données pour entraîner les réseaux de neurones convolutifs entiers adaptés à la problématique. La technique de "Data augmentation" permet de démarrer d'un modèle qui a déjà appris avec un jeu d'images important (milliers d'images), et de lui faire apprendre davantage, en lui fournissant un plus petit jeu d'images (centaines d'images) de la nouvelle zone d'étude. Par exemple un modèle peut avoir appris à classifier des images de la Californie, États-Unis. Pour lui permettre de classifier des images de la Ville de Sherbrooke, il est souhaitable de lui fournir un nouveau jeu de données spécifique à cette ville afin qu'il s'adapte (ses paramètres) à cette région. Dans le contexte de cet essai, les données acquises sur le terrain seront fournies aux différents modèles qui seront évalués, et qui seront ré entraînés avec ce nouveau jeu d'images adapté à la zone d'étude.

La paramétrisation (des "hyper paramètres") des réseaux de neurones est très "subtile" et "intuitive" et requière de l'expérience. C'est un processus d'essais-erreurs qui est très couteux en temps, et risqué puisqu'il n'y a aucune garantie de succès. La technique de "Transfer Learning" permet d'hériter d'un modèle qui est déjà entraîné et configuré, et de l'adapter pour répondre à ses besoins. Cette technique permet un gain en temps et en énergie (et en argent) important puisque le temps de conception (architecture et configuration) et le temps d'entraînement, de validation et de tests sont diminués de façon non négligeable. La problématique pour l'essai est de trouver le modèle qui est le plus adapté pour répondre au besoin, et il existe des milliers [12]. La recherche dans la littérature permet heureusement de limiter les choix et donner des pistes ([jia_real-time_2020] [16] [18]). La problématique de la conception existe toujours, car le modèle a besoin d'être étudié, adapté et ré entraîné, jusqu'à l'obtention de résultats probants. Mais la paramétrisation des hyper paramètres n'est plus nécessaire (supposément), ce qui est très avantageux.

2.3 Objectifs

Le premier sous-objectif est de déterminer quelles sont les limites de la plateforme, d'un point de vue matériel (GPU, CPU, mémoire, transfert mémoire, consommation, etc.), mais aussi applicatif (entraînement, inférence). Cette phase du projet va permettre d'exécuter différents modèles déjà existants, sans modification, en tenant compte des éléments documentés dans la littérature [16] [jia_real-time_2020] [18]. Selon le déroulement de cette étape, un ou plusieurs modèles seront sélectionnés.

Un autre sous-objectif est d'optimiser ou d'adapter la plateforme, d'un point de vue matériel, mais aussi applicatif, afin d'avoir les meilleures performances et résultats possibles pendant l'entraînement et l'inférence.

Comme les résultats devront être disponibles en tout temps, une connexion à distance sécurisée devra être mise en place. Cette connexion permettra aussi de pouvoir prendre le contrôle du nanoordinateur à distance et de l'administrer.

L'approche, les tests, et les résultats seront documentés. Il y aura beaucoup d'activités relatives à la conception et aux tests, le cheminement complet ne sera pas fourni. Une synthèse sera préférée et les informations les plus pertinentes seront incluses. Les détails de l'installation de l'environnement de développement et des applications, bibliothèques et autres dépendances nécessaires seront inclus, ainsi que ceux de la configuration. Dans le cas où l'objectif principal n'est pas atteint, ou partiellement, la/les raison/s de l'échec seront spécifiées et des pistes de solutions potentielles proposées.

3 Cadre théorique (état des connaissances, revue de la littérature)

Il y a deux sections, la première qui concerne le nano ordinateur et ensuite la seconde, l'apprentissage profond et la segmentation sémantique.

3.1 Cadre théorique au sujet du nano ordinateur

Voici le plan qui est utilisé pour rédiger le cadre théorique au sujet du nano ordinateur.

TODO

- historique et évolution ; une brève présentation de l'historique des nano ordinateurs, de leur apparition à leur place aujourd'hui.
- usages ; quelques exemples d'usages des nano ordinateurs, dans un contexte professionnel.
- architecture ; brève présentation, fonctionnement et comparaison des architectures matérielles des nano ordinateurs, leurs couts, leurs avantages et limitations.

3.2 Cadre théorique au sujet de l'apprentissage profond et de la segmentation sémantique

TODO

Voici le plan qui est utilisé pour rédiger le cadre théorique au sujet des réseaux de neurones et de la segmentation sémantique.

- historique ; une brève présentation de l'historique et du contexte de l'intelligence artificielle, l'apprentissage machine et les réseaux de neurones.

Les concepts de l'Intelligence artificielle (AI) existe depuis les années 1950 et ont continué à se développer jusqu'à leur popularité des 10 dernières années ;

- popularité depuis 10 ans ; argumentation autour des raisons de la renaissance de l'apprentissage machine.

Trois raisons principales ont permis à ce domaine de sortir du champ de la recherche pour celui de l'industrie et la mise en production : amélioration de la capacité et la puissance des machines ; jeux de données plus larges ; algorithmes plus avancés ;

- domaine ; où se situent les réseaux de neurones et la segmentation sémantique dans la hiérarchie de l'IA.

L'apprentissage profond est un sous-domaine de celui de l'apprentissage machine qui est un sous-domaine de celui de l'intelligence artificielle.

La segmentation sémantique d'images ou de vidéos avec des algorithmes d'apprentissage profond fait partie du domaine de la télédétection par la vision.

- applications ; quelques exemples d'applications des réseaux de neurones selon leur type, dont les réseaux de neurones à convolution entiers.

Les applications sont plus sophistiquées, la segmentation sémantique en fait partie.

- principes ; présentation des principes théoriques de la segmentation sémantique.

La segmentation sémantique d'images est une technique élaborée de classification supervisée d'images.

À noter qu'il n'est pas prévu expliquer le fonctionnement des réseaux de neurones, tels que les différentes fonctions (activation, perte), les hyper-paramètres, les différentes architectures et les types de couches.

4 Matériel et méthodes

4.1 Site d'étude

Tel que trouvé dans le rapport détaillé sur le projet pilote d'entretien hivernal de la piste multifonctionnelle du pont Jacques-Cartier [21], la piste multifonctionnelle du pont Jacques Cartier relie Montréal intramuros, proche de la station de métro Papineau, et la rive sud, à Longueuil, proche de la station de métro Longueuil et le terminal de bus. Elle est longue d'une distance de 2.7km et est

située d'un seul côté du pont, côté sud. Elle est surtout utilisée par les cyclistes, et moindrement par les passants. Sa configuration est bien particulière [20] : elle ne longe pas la route adjacente sur toute sa longueur ; elle est interrompue par une voie de sortie de l'île Notre-Dame ; des chicanes sont disposées à certains endroits ; sa largeur varie entre 2.5m et 1.8m ; elle possède une pente assez prononcée à certains moments ; il y a des courbes assez serrées. Elle est fermée l'hiver par mesure de sécurité. Elle est ouverte au début du printemps jusqu'au début de l'hiver, lorsque les conditions ne nécessitent pas d'entretien.



FIGURE 1 – Vue aérienne du pont Jacques-Cartier



a alamy stock photo

AN9ENY
www.alamy.com

FIGURE 2 – Vue aérienne du pont Jacques-Cartier

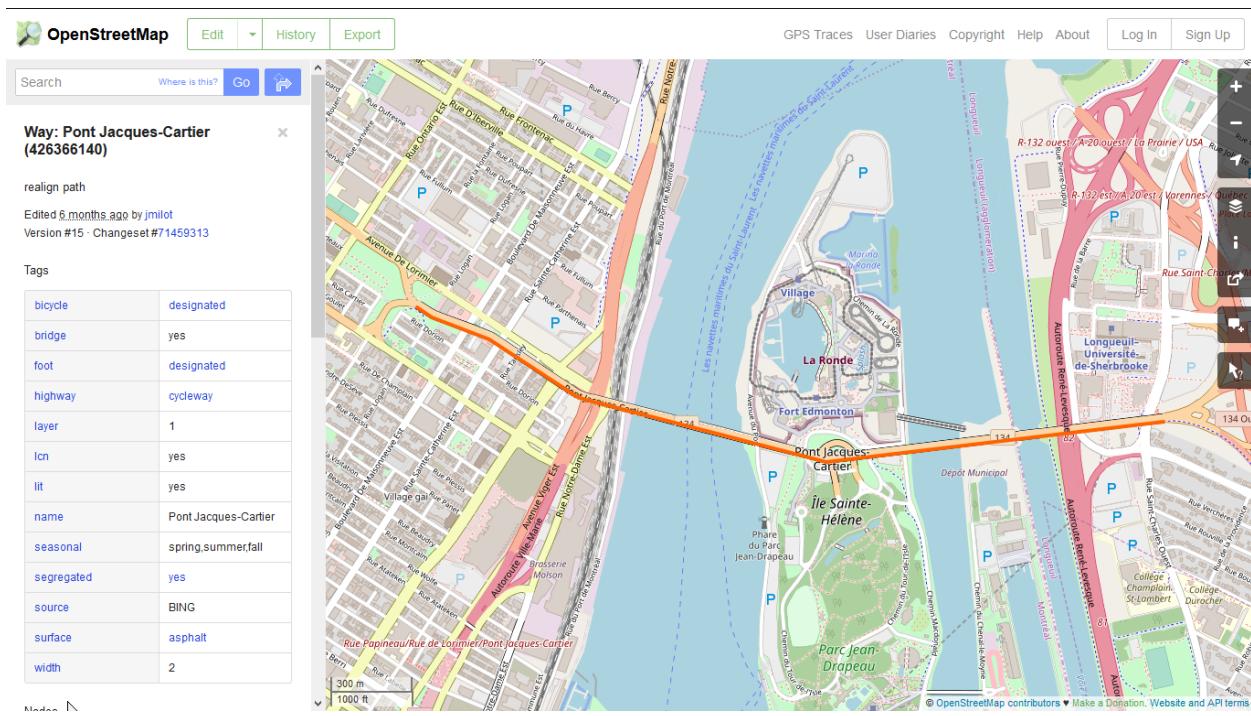


FIGURE 3 – Carte du site d’implantation : le pont Jacques-Cartier et la piste multifonctionnelle en orange sur le pont

Piste multifonctionnelle du pont Jacques-Cartier

La piste multifonctionnelle du pont Jacques-Cartier est un lien singulier aux défis uniques



Pont urbain à la géométrie atypique et exposé à des conditions météorologiques particulières.

Particularités de la piste et enjeux de sécurité

La combinaison de l'ensemble de ces éléments soulève des enjeux de sécurité accrus en hiver.

Géométrie atypique

- + Longueur (2,7 km)
- + Dégagement latéral étroit (de 2,5 m à 1,8 m en hiver)
Normes du MTMDET : 3,5 m
- + Pentes longues et abruptes (4,2 %)
- + Virages serrés
- + Enclavement (garde-corps de chaque côté)

Piste surélevée

- + Dalle de béton (15 cm)
- + Pas de matériaux en profondeur qui l'isolent et atténuent son refroidissement
- + Plus affecté par les variations météorologiques qu'une piste sur remblai
- Danger de chute de glace**
- + Structures métalliques en hauteur (sections 3 et 7)
- + Neige projetée des voies de circulation

Complexité des conditions météorologiques du fleuve Saint-Laurent

- + L'environnement climatique au-dessus du fleuve Saint-Laurent est unique, rigoureux et changeant
- + Milieu très humide, venteux et changeant propice à la formation imprévisible de glace noire qui affecte directement la qualité de la surface de roulement

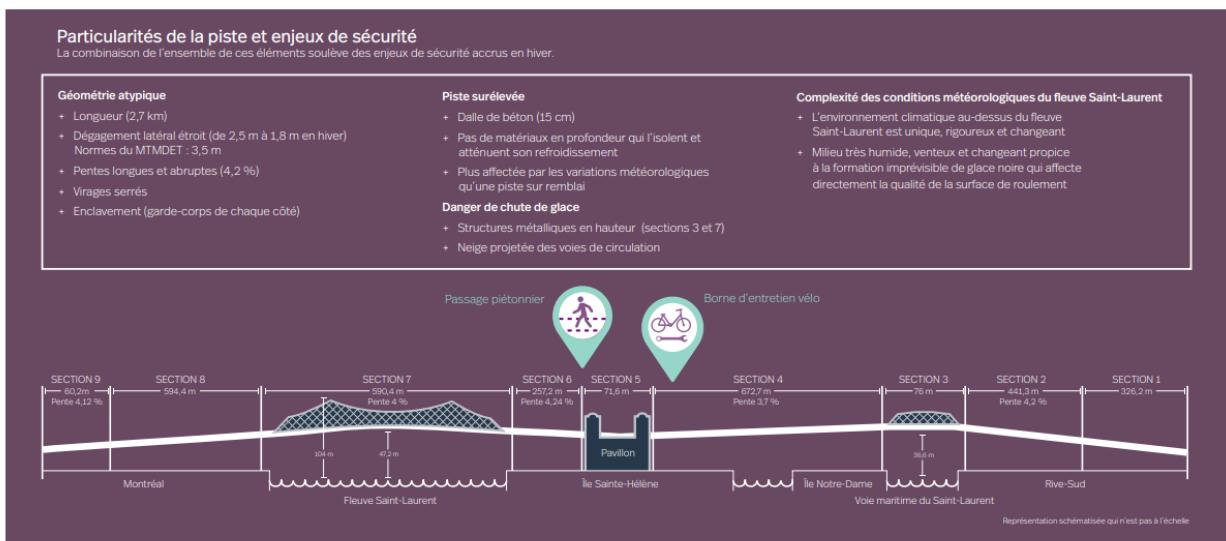


FIGURE 4 – Schéma de la configuration de la piste multifonctionnelle

4.2 Modèles et jeux de données

Données

Les ressources mises à disposition par le constructeur du Jetson nano, NVIDIA, font référence à des jeux de données qui sont disponibles publiquement.

En complément des ressources de NVIDIA, deux références scientifiques seront principalement étudiées, car leurs recherches ont été faites avec le Jetson nano ([16] et [5]). Beaucoup de références ont été publiées ces deux dernières années sur le sujet de la segmentation sémantique, ils existent donc de multiples alternatives inspirantes.

Internet est une mine de données : il existe des forums et des blogues dans lesquels des utilisateurs publient leurs expérimentations de la segmentation sémantique en temps réel avec le Jetson nano ([8]), ou plus génériquement la segmentation sémantique ; des sites comme "modelzoo.co" ou "kaggle.com" sont des entrepôts de données et de modèles FCN prêts à être utilisés ; une autre option a été d'effectuer une recherche d'images ou de vidéos de la piste multifonctionnelle du pont Jacques-Cartier via les sites de recherche tels que Google.

L'Association des piétons et cyclistes du pont Jacques-Cartier existe depuis de nombreuses

**TODo
review**

années pour promouvoir le transport actif et conserver la piste multifonctionnelle du pont Jacques Cartier ouverte durant l'hiver. Ils fournissent, via leurs sites Internet, des collections de vidéos et d'images qui seront utilisées après leur avoir demandé leur autorisation verbale et écrite. [9] [10]

Approche prévue pour le traitement des données

Il y a deux phases à cet essai : l'inférence avec des modèles déjà prêts et l'inférence avec des modèles ré entrainés. Les données utilisées par l'inférence sont des vidéos (d'une certaine résolution et d'un certain nombre d'images par seconde), et celles pour l'entraînement sont des images. Dans les deux cas, les images pour l'entraînement ou l'inférence doivent être d'une taille bien précise, celle avec laquelle le modèle a été, ou sera, entraîné. La résolution et la qualité de l'image vidéo seront nivelées vers le bas afin de déterminer la limite inférieure acceptable pour la détection la plus efficace et fiable possible. La résolution et le nombre d'images par seconde de la vidéo sont contrôlés par le logiciel ("driver" en anglais) de la caméra, et sont configurables.

Tout cela signifie que les vidéos ou nouvelles images devront être traités pour répondre à une certaine taille et résolution requise par le modèle, tout en conservant une qualité élevée (nombre de pixels, niveaux de couleurs). De nouvelles images pour l'entraînement seront extraites des vidéos, et annotées.

Certains cadres d'application logicielle ("framework") d'apprentissage profond (par exemple "Keras") offrent l'option d'augmenter automatiquement le jeu de données avec des techniques d'augmentation de données (par exemple la rotation, le redimensionnement, l'effet miroir), ce qui est très utile et non négligeable.

Voici le tableau de synthèse des données, incluant la référence avec leurs réseaux de neurones.

TABLE 2: Tableau des données

	Spécification	Description
1	réseau : SegNet jeu de données : CamVid vidéos : 10 minutes résolution/s : HD	SegNet est un réseau qui a été créé pour la segmentation sémantique de vidéos. Il a été entraîné avec le jeu de données de CamVid, qui procurent des vidéos de la route avec la même perspective que le conducteur du véhicule. Un modèle entraîné est disponible pour le Jetson nano. https://github.com/PengKIKi/camvid
2	réseau : MFANet jeu de données : Cityscapes nombre d'images : 5000 résolution/s : 1280x1024	MFANet est un réseau qui a été créé en 2019 pour la segmentation sémantique sur des appareils tel que le Jetson nano. Il a été entraîné avec le jeu de données de Cityscapes, qui procurent des images de scènes urbaines. Différentes stratégies d'augmentation de données sont utilisées. Des tests ont été faits avec le Jetson nano. leejy@ustb.edu.cn
3	réseau : RESNet18 jeu de données : Cityscapes nombre d'images : 25 000 résolution/s : 360x720, 512x256, 1024x512, 2048x1024	Cityscapes est un jeu de données qui fournit des images de rues spécifiquement destinées pour la segmentation sémantique. Il peut être utilisé par de nombreux réseaux. RESNet18 a été entraîné avec ce jeu et est disponible en diverses résolutions pour le Jetson Nano. https://github.com/tynguyen/MAVNNet/tree/master/data/perch_drone
4	réseau : RESNet18 jeu de données : DeepScenes nombre d'images : 15 000 résolution/s : 576x320, 864x480	DeepScene propose un modèle et un jeu de données. Le modèle est entraîné avec différents jeux de données, comme Cityscapes, SUN-RGBD, Synthia. Le jeu de données fournit des images de forêt, qui est destinée pour la segmentation sémantique. RESNet18 a été entraîné avec ce jeu et est disponible en deux résolutions pour le Jetson Nano. http://deepscene.cs.uni-freiburg.de
5	réseau : DeepScene jeu de données : Synthia nombre d'images : 220 000 résolution/s : 1280x760	Le jeu de données Synthia fournit des images (et vidéos) de scènes de rue comme celui de Cityscapes, et qui est destiné pour la segmentation sémantique. DeepScene a été entraîné avec ce jeu. Il n'a pas été testé avec le Jetson Nano. http://3dvision.princeton.edu/datasets.html

Spécification	Description
6 jeu de données : Association des piétons et cyclistes pont Jacques-Cartier nombre d'images : 313 résolution/s : variées	<p>L'Association des piétons et cyclistes du pont Jacques-Cartier a une collection d'images et de vidéos de la piste multifonctionnelle du pont Jacques-Cartier. Ce n'est pas un jeu de données qui est prêt à être utilisé pour l'apprentissage tel quel, il doit être préparé. Mais c'est une source de données qui est très importante pour l'essai. Il est envisagé de contacter l'association au besoin afin de leur demander leur collaboration pour la collecte d'autres d'images ou vidéos.</p> <p>https://www.flickr.com/photos/pontjacquescartier http://pontjacquescartier365.com/videos-pont-jacques-cartier</p>
7 jeu de données : images et vidéo sur Internet nombre d'images : entre 30-50 résolution/s : variées	<p>Internet est une source de données non négligeable en termes de données. Quelques images et vidéos de la piste multifonctionnelle du pont Jacques-Cartier, autre que celles fournies par L'Association des piétons et cyclistes du pont Jacques-Cartier, sont disponibles. Ce n'est pas un jeu de données qui est prêt à être utilisé pour l'apprentissage tel quel, il doit être préparé. Mais c'est une source de données qui est très importante pour l'essai.</p> <p>https://google.ca</p>
8 jeux de données : KITI Road/Lane Detection	<p>Ce jeu de données contient 289 images d'entraînement et 290 images de tests d'image de routes urbaines. Il existe une grande multitude de modèles qui sont entraînés avec ce jeu de données.</p> <p>http://www.cvlibs.net/datasets/kitti/eval_road.php</p>

4.3 Matériel et logiciels

Le nano ordinateur

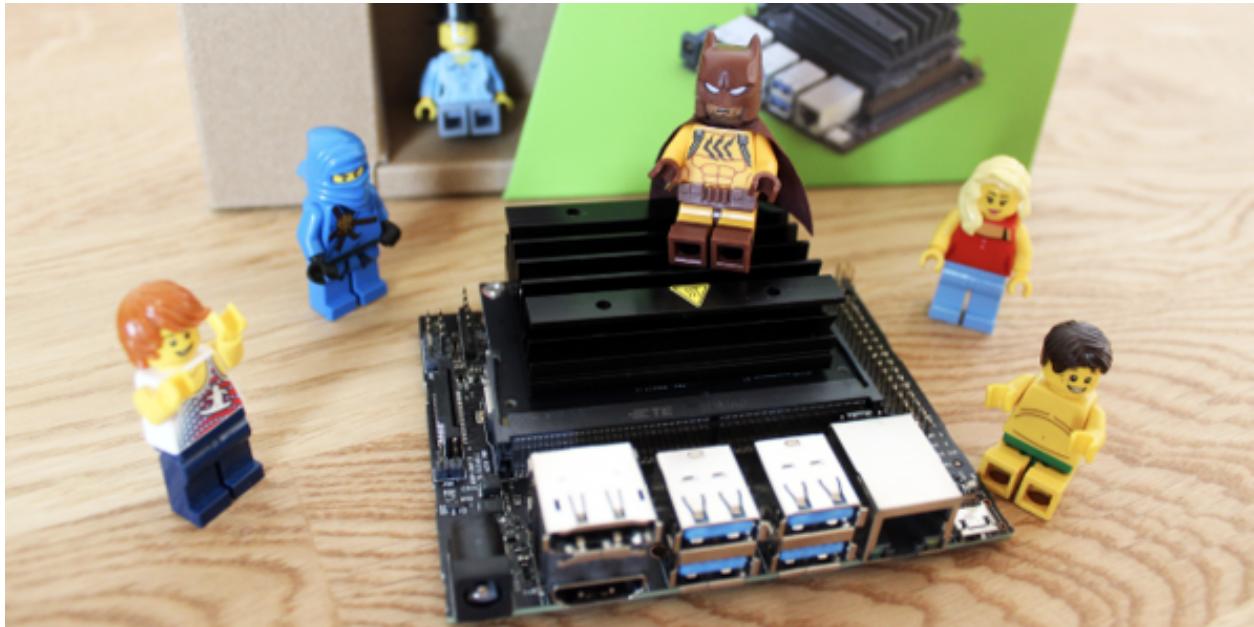


FIGURE 5 – Photo de la carte mère Jetson Nano de NVIDIA, représenté avec des Lego pour démontrer sa petitesse

L'objet d'étude de cet essai est un nano ordinateur. Un nano ordinateur est un ordinateur miniaturisé en taille, mais aussi limité en capacité. Il existe différents fabricants et modèles, de caractéristiques techniques variées, pour répondre à différents besoins. Le dernier né est le modèle "Jetson nano" du fabricant "NVIDIA", disponible depuis juin 2019 au prix très abordable de 99\$US. La compagnie NVIDIA a conçu ce matériel spécialement pour différentes applications d'inférence de modèles d'apprentissage profond sur une plateforme mobile (drone) ou proche des données ("edge" en anglais). Ce modèle a été choisi afin de répondre à l'intérêt que suscitent ses capacités et ses limites.

TODO

Logiciels

TODO

De même que pour les périphériques, les logiciels qui seront utilisés seront résumés dans un tableau, où il sera indiqué leur nom, le type de licence, leur version, leurs avantages et limitations, comme pour le système d'exploitation, l'environnement de développement pour l'apprentissage profond, l'inférence, les logiciels de traitements vidéos et d'images.

Pour tester les performances de la micro-sd et du disque SDD interne M.2 NVMe, l'utilitaire hdparm a été utilisé. Il est nécessaire de l'installer ('sudo apt-get install hdparm') car il n'est pas inclus dans le système UBuntu 18.04 ;

Le SDK qui sera utilisé sera celui fourni par NVIDIA et qui se nomme "JetPack". Il contient le système d'exploitation Linux For Tegra (L4T) qui est une version de la distribution Ubuntu 18.04

mise à la saveur de NVIDIA. Jetpack contient aussi d'autres librairies qui sont nécessaires pour l'inférence, tel que Cuda, CuDn et TensorRT.

Figure/tableau des versions et description des éléments du framework Jetpack. TODO

Le cadre d'application logicielles ("frameworks") d'apprentissage profond qui seront utilisés sont PyTorch avec un fork de torchvision.

Figure/tableau des versions, role et responsabilité de pytorch et le fork de torchvision dans le cadre de l'essai. TODO

Enfin pour regénérer le ONNX, les librairies TensorRT et ONNX ont été utilisé, en compagnie de l'utilitaire 'tensorrt' qui permet de valider et tester l'ONNX généré.

Figure/tableau des versions, role et responsabilité de TensorRT, ONNX et tensorrt dans le cadre de l'essai. TODO

4.4 Méthodologie

Voici à très haut niveau les grandes étapes de cet essai :

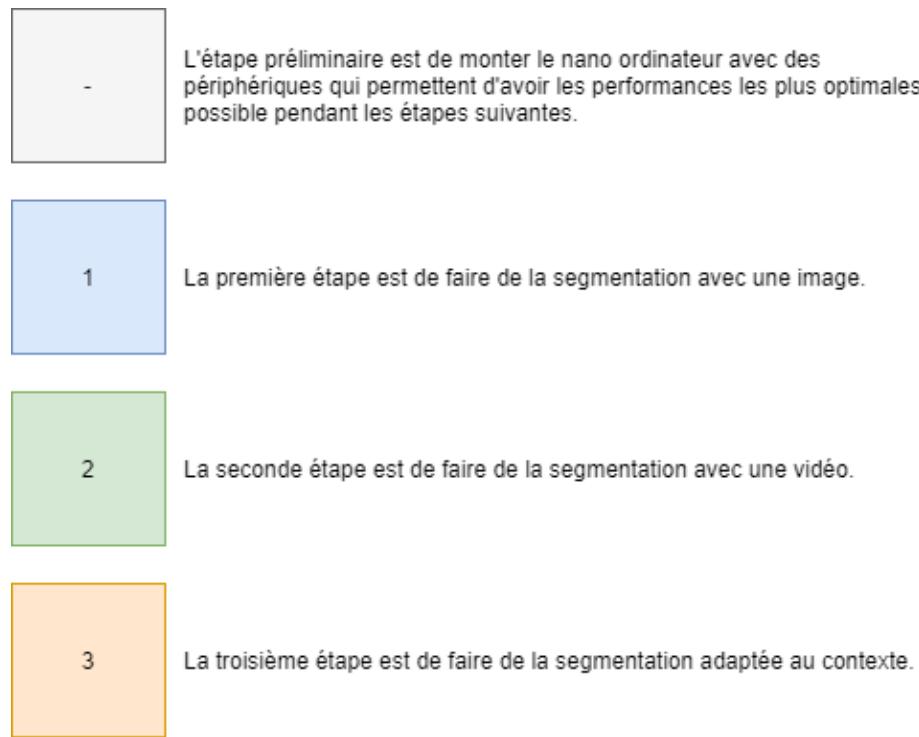


FIGURE 6 – Organigramme de la méthodologie à haut niveau

Pour y parvenir, la méthodologie suivante a été suivie et permet d'évaluer les performances de base de la segmentation sémantique avec le nano ordinateur.



FIGURE 7 – Organigramme de la méthodologie pour évaluer les performances

Si chacun des blocs est exploité, chacun d'eux s'organise autour des activités suivantes :

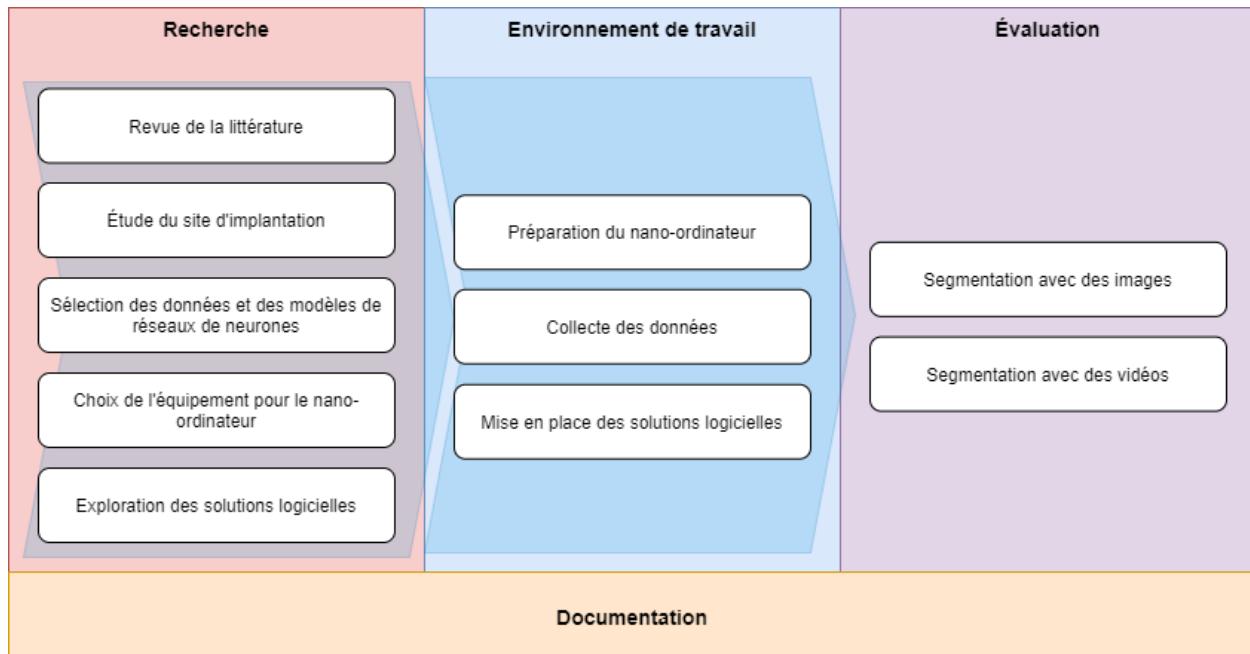


FIGURE 8 – Organigramme des détails de la méthodologie pour évaluer les performances

Si l'évaluation est probante, la méthodologie se verra bonifiée par des étapes d'adaptation et de traitement.

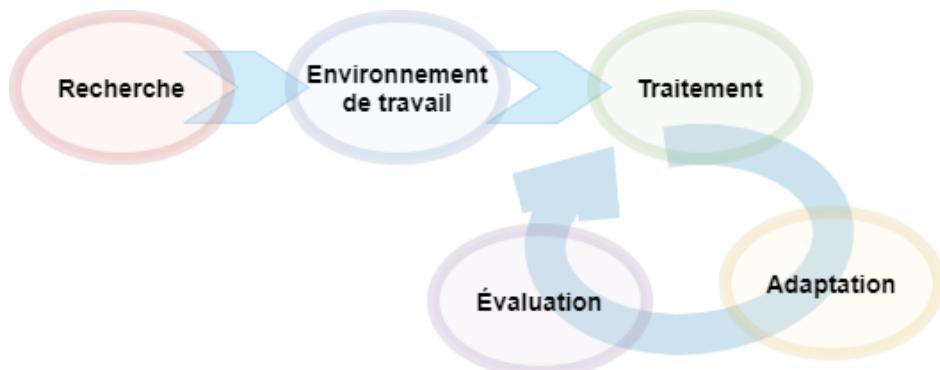


FIGURE 9 – Organigramme de la méthodologie pour évaluer les performances après une phase d'adaptation théorique

Mais sincèrement dans la pratique la méthodologie ressemblera plus à celle-ci :

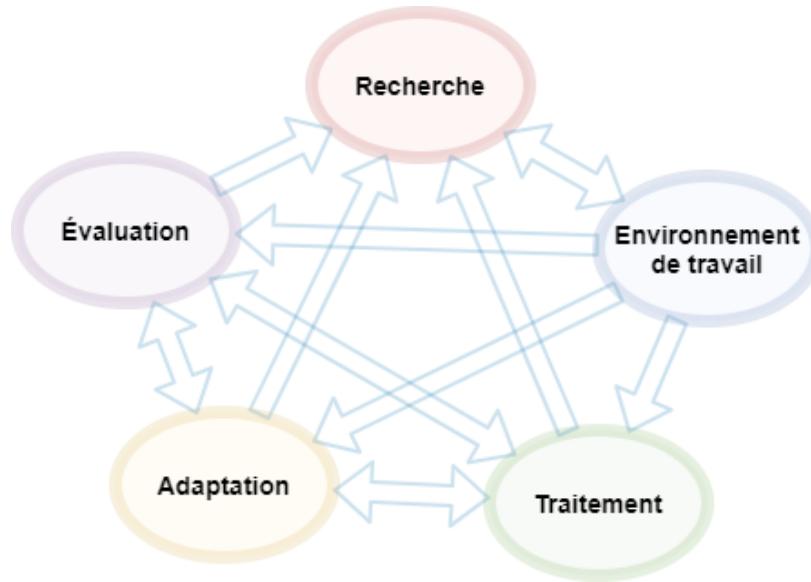


FIGURE 10 – Organigramme de la méthodologie pour évaluer les performances après une phase d'adaptation réaliste

Si chacun des blocs est exploité, chacun d'eux s'organise autour des activités suivantes :

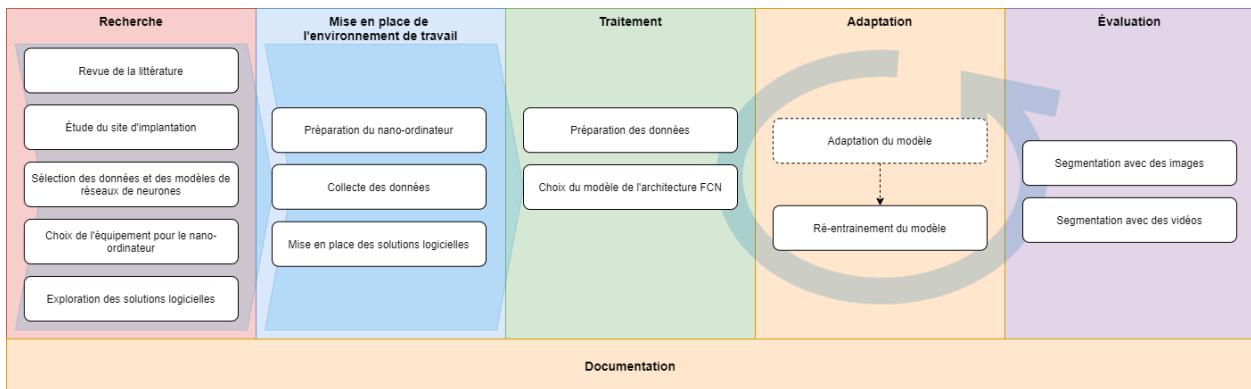


FIGURE 11 – Organigramme des détails de la méthodologie pour évaluer les performances après une phase d'adaptation

Voir l'organigramme de la figure 11 pour la représentation graphique de la méthodologie, et dont les phases peuvent être résumées de la façon suivante :

- Recherche des références, des modèles et des données, ainsi que l'équipement pour le nano ordinateur et des logiciels nécessaires.
- Installation sur le Jetson nano du système d'exploitation, de l'environnement de développement et de tests pour l'inférence.
- Itération entre les étapes suivantes :

- Inférence avec le Jetson nano en utilisant les modèles et les sources de données sélectionnées.
- Adaptation des modèles à différentes résolutions d'images et à la zone d'étude.
- Traitement des données afin de les adapter au requis des modèles.

4.5 Recherche

4.5.1 Revue de littérature

La revue de la littérature a débuté en octobre-novembre 2019, c'est-à-dire quelques mois après la disponibilité du nano ordinateur (juin 2019). La recherche s'est concentrée sur des références traitantes des concepts du sujet de l'essai : la segmentation sémantique, le temps réel, et les nano ordinateurs. Le premier objectif a été de trouver si des études avaient déjà expérimenté le nano ordinateur, en particulier pour la segmentation de vidéos en temps réel. Pendant cette recherche, j'en ai profité pour effectuer une révision de l'évolution des réseaux de neurones convolutionnels entiers (FCN Fully Convolutional Network) et des différentes architectures, et chercher d'autres solutions de détection de la route en temps réel grâce au FCN.

Il a été assez compliqué de trouver des références intégrant les nano ordinateurs. Comme l'objectif de l'essai est de valider les performances d'un nano ordinateur bien spécifique, les mots-clés "NVIDIA Jetson nano" font partie de la stratégie de recherche.

Les réseaux de neurones convolutifs entiers (FCN) sont implicitement inclus dans les résultats puisque c'est le "state-of-art" actuellement pour répondre au besoin de la segmentation sémantique d'images.

Plus de 75 références ont été collectées. Une quarantaine ont été sélectionnées. Cette sélection peut se décomposer en trois catégories : les références se rapprochant le plus du sujet de l'essai ; l'histoire et les antécédents des réseaux de neurones ; du matériel éducatif pour étudier et manipuler les réseaux de neurones.

Je me suis intéressé aux références des années les plus récentes, autour de 2020, 2019 et 2018, car les avancées dans le domaine des réseaux de neurones sont très rapides. Par curiosité je suis allé aussi parfois voir dans les années bien plus éloignées, comme 1998, ou j'ai trouvé un article proposant une solution pour prédire la température de la surface de la route avec des réseaux de neurones.

Je n'ai pas pu trouver de références spécifiquement pour la déduction de l'état de la surface (mouillé, gelée, etc.) d'une piste multifonctionnelle (vélo, piéton).

Il est intéressant de noter que la banque de données SCOPUS retourne plus de 11,000 documents avec l'expression "segmentation AND "real-time"". Il y en a plus de 700 uniquement pour l'année 2019.

4.5.2 Étude du site d'implantation

Le nano ordinateur est destiné à être déployé sur le chemin de la piste multifonctionnelle du pont Jacques-Cartier. L'étude du site a permis de chercher à comprendre, parmi ses caractéristiques, les difficultés de son usage l'hiver. Il a été tenté de comprendre les défis et les raisons, techniques, politiques, sécuritaires, de pouvoir la conserver ouverte toute l'année. Une carte du site permet de

montrer un exemple de configuration où et comment seront installés les nano ordinateurs, et des images de ces zones d'intérêt permet de "visualiser" ce qui sera interprété par le modèle.

Un mot est réservé pour citer "L'Association des piétons et cyclistes du pont Jacques-Cartier" qui est un acteur actif pour le développement du transport actif dans cette région du Québec, et dont les membres sont des usagers habituels de la piste multifonctionnelle, même l'hiver.

4.5.3 Sélection des données et des modèles de réseaux de neurones

Les ressources mises à disposition par le constructeur du Jetson nano, NVIDIA, ont été étudiées pour apprendre et tester le nano ordinateur. Parmi les plus intéressantes, on peut citer le "Jetson Nano Developer Kit", le "NVIDIA Deep Learning Institute", la communauté Jetson, les tutoriels, les "benchmarks". Des jeux de données sont fournis gratuitement.

En complément des ressources de NVIDIA, deux références scientifiques ont été principalement utilisées comme points de départ et comme modèles pour l'essai, car leurs études ont été faites avec le Jetson nano ([16] et [5]). Beaucoup de références ont été publiées ces deux dernières années sur le sujet de la segmentation sémantique, ils existent donc de multiples alternatives inspirantes.

Internet est une mine d'information. Il existe des forums et des blogues dans lesquels des utilisateurs publient leurs expérimentations de la segmentation sémantique en temps réel avec le Jetson nano ([8]), ou plus génériquement la segmentation sémantique. Des sites comme "modelzoo.co" et "kaggle.com" sont des entrepôts de modèles déjà entraînés.

Une autre option est d'effectuer une recherche d'images ou de vidéos de la piste multifonctionnelle du pont Jacques-Cartier via les sites de recherche tels que Google.

L'Association des piétons et cyclistes du pont Jacques-Cartier existe depuis de nombreuses années pour promouvoir le transport actif et conserver la piste multifonctionnelle du pont Jacques Cartier ouverte durant l'hiver. Ils fournissent, via leurs sites Internet, des collections de vidéos et d'images qui pourraient être utilisées. Il serait aussi possible d'entrer en contact avec l'association et leur demander de prendre de nouvelles vidéos. [9] [10]

Les architectures des modèles FCN sélectionnés pour l'essai sont résumés dans un tableau récapitulatif, incluant leur type, leur application et leurs jeux de données respectifs, précisant les différentes variantes entre résolutions et nombre d'images par seconde (FPS).

4.5.4 Choix de l'équipement pour le nano ordinateur

L'objet d'étude de cet essai est un nano ordinateur. Un nano ordinateur est un ordinateur miniaturisé en taille, mais aussi limité en capacité. Il existe différents fabricants et modèles, de caractéristiques techniques variées, pour répondre à différents besoins. Le dernier né est le modèle "Jetson nano" du fabricant "NVIDIA", disponible depuis juin 2019 au prix très abordable de 99\$US. La compagnie NVIDIA a conçu ce matériel spécialement pour différentes applications d'inférence de modèles d'apprentissage profond sur une plateforme mobile (drone) ou proche des données ("edge" en anglais). Ce modèle a été choisi afin de répondre à l'intérêt que suscitent ses capacités et ses limites. Une image du Jetson nano et un tableau de ses caractéristiques techniques seront disponibles.

L'architecture matérielle sera étudiée et présentée avec l'aide d'images, de diagrammes et de textes explicatifs. Les éléments clés seront identifiés.

Afin d'optimiser les performances du Jetson nano, une recherche des périphériques les plus adaptés pour répondre aux besoins de performance (et de budget) de l'essai est essentielle, telle que l'alimentation, le stockage, la caméra. Des images des périphériques seront incluses, et les caractéristiques principales seront présentées dans des tableaux.

Le matériel est commandé par le collaborateur de cet essai "Vision météo".

4.5.5 Exploration des solutions logicielles

TODO

De même que pour les périphériques, les solutions logicielles logiciels nécessaires sont résumés dans un tableau, où il sera indiqué leur nom, leur version, les avantages et limitations, comme le système d'exploitation, l'environnement de développement, l'inférence, les logiciels de traitements vidéos et d'images.

4.6 Environnement de travail

4.6.1 Préparation du nano ordinateur

L'organigramme de la figure 12 présente les activités qui composent la préparation du nano ordinateur.

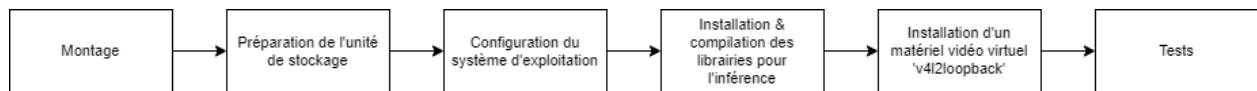


FIGURE 12 – Préparation du nano ordinateur

Montage

Le nano ordinateur est une carte mère livrée sans aucun périphérique ni même boîtier. Vu que les performances logicielles dépendent des performances matérielles, surtout pour une unité telle qu'un nano ordinateur où les capacités matérielles sont très limitées, la première partie de l'essai a été allouée à la sélection des accessoires et périphériques qui vont permettre d'augmenter les performances, protéger et utiliser confortablement le nano ordinateur.

L'organigramme de la figure 13 présente les activités qui composent le montage du nano ordinateur.

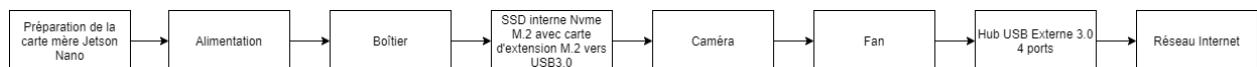


FIGURE 13 – Montage du nano ordinateur

Préparation de la carte mère Jetson Nano



FIGURE 14 – Carte mère du nano ordinateur

Le nano ordinateur qui est livré dans sa boîte est uniquement une carte mère, sans unité de stockage, ni boîtier, clavier, souris, écran, capacité wifi, ou caméra. Il est uniquement livré avec un câble micro-usb qui lui permet d'être démarré avec une alimition minimale de 5 Volts/2Amp et ne consommer que 5 Watts. Aucun système d'exploitation n'est livré non plus. Vu que de l'objectif de l'essai est de tester les capacités du nano ordinateur et que la consommation sera de plus de 5Watt dues aux branchements de multiples périphériques, certaines "broches" sur la carte mère doivent être activées : la broche J48 permet de brancher un adaptateur d'alimentation de 5Volts 4Amp au lieu de l'alimentation micro-usb ; et la broche J38 permet d'activer le PoE (Power-Over-Ethernet) afin d'hériter de l'alimentation du câble Ethernet. Aucune autre préparation sur la carte n'est nécessaire.

Alimentation



FIGURE 15 – Adaptateur 5 volts 4 amps

L'alimentation du nano ordinateur est l'élément matériel le plus important du système. De base le nano ordinateur est livré avec un câble micro-USB, lui permettant d'être alimenté en 5Volt 2Amp. Mais le besoin en énergie augmente avec les périphériques qui s'accumulent, tel qu'une caméra. Il est prudent de choisir un adaptateur 5Volt 4Amp d'un fournisseur recommandé par

NVIDIA, car un changement de puissance sensible en entrée impacte le fonctionnement opérationnel du nano ordinateur. Deux adaptateurs ont été utilisés, l'un recommandé, et l'autre non, afin de tester leur performance.

Dans le cadre de l'essai, l'alimentation du nano ordinateur est utilisée pour alimenter la carte mère, qui comporte entre autres les CPUs, le GPU, le Hub USB 3.0 interne, le contrôleur Ethernet et le port HDMI. Mais aussi la caméra et le ventilateur et optionnellement une carte d'extension M.2 NVMe. Afin d'assister l'adaptateur, un hub USB 3.0 externe a été utilisé pour brancher la souris, le clavier, et à un moment donné le dongle Wifi.

Boitier



FIGURE 16 – Boitier pour le nano ordinateur

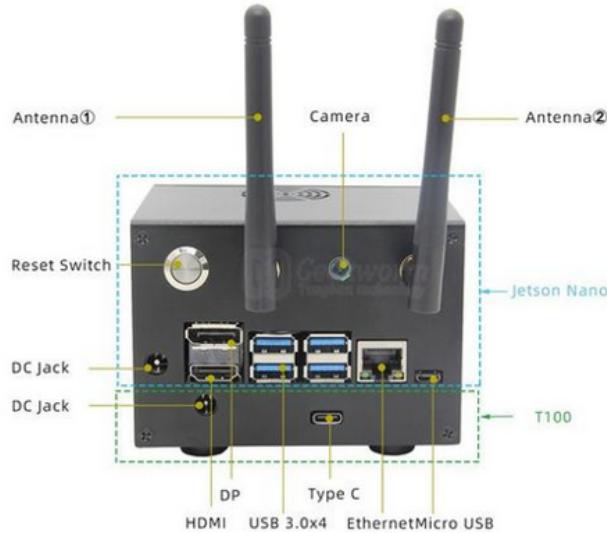


FIGURE 17 – Vue arrière du boîtier pour le nano ordinateur

Afin de protéger le nano ordinateur durant l'essai et l'utiliser dans les conditions les plus proches de son futur mode d'opération, il a été installé dans un boîtier en métal. Le boîtier a été choisi en tenant compte qu'une carte d'extension pour un SSD interne sera installée, ainsi qu'une caméra et un ventilateur. Durant l'essai le nano ordinateur sera manipulé très fréquemment en raison d'un manque d'espace réservé dans la maison. Le boîtier permet donc d'éviter de manipuler le matériel et les connecteurs, les protège, évitant de risquer de les briser, et donc ajouter des délais à l'essai.

SSD interne NVme M.2 avec carte d'extension M.2 vers USB3.0



FIGURE 18 – Disque SSD NVMe M.2 interne 250GB

Un disque SSD est entre 50 et 100 fois plus performant qu'une carte micro-SD. Il est aussi plus adapté pour manipuler les petits fichiers et héberger un système d'exploitation. Il est aussi plus résilient à long terme. C'est donc une option qui ne doit pas être négligée dans le contexte de tests de performance, encore plus avec un nano ordinateur dont les capacités matérielles sont limitées. Néanmoins, il y a un contreparti important dans la situation d'un nano ordinateur : la consommation d'énergie. Un SSD interne va demander plus d'énergie qu'une carte micro-SD, et si le nano ordinateur n'est pas capable de gérer correctement les besoins en énergie de ses extensions matériels, le SSD interne risque d'échouer en pleine opération et le nano ordinateur devenir non fonctionnel soudainement.

Il y a deux choix qui ont été retenus pendant l'essai pour brancher un SSD interne au nano ordinateur : soit via une carte d'extension M.2 MVMe, et connecté via le Hub USB, soit via une carte d'extension M.2 NVMe connectée au port PCIe interne du nano ordinateur, normalement destinée à une carte d'extension Wifi.

Concernant le disque SSD M.2 NVMe connecté à la carte d'extension M.2 via le Hub USB 3.0 interne, le système L4T de NVidia (Ubuntu 18 mis à la saveur NVidia) ne supporte pas les SSD M.2 NVMe connecté au port USB. Il n'est pas reconnu / détecté, il est donc impossible de le formater, de le partitionner, de l'utiliser. Comme il serait risqué pour l'essai de se lancer dans la recompilation du kernel du L4T, une alternative trouvée sur le développeur forum de NVidia est de passer par un adaptateur M.2 MVMe connecté au port PCIe interne.

Malheureusement cette alternative a rapidement été abandonnée. Il a été possible de démarrer et installer le système d'opération sur le SSD M.2, et faire quelques tests, mais pour une raison inconnue, le système n'était pas stable et devenait non opérationnel assez rapidement, le système perdant la connexion au SSD. La durée la plus longue de stabilité observée a été de moins 30 minutes. Une hypothèse est une baisse d'énergie qui survient à un moment et qui impacte l'alimentation du SSD, chaque volt et milliampère étant important pour la stabilité du nano ordinateur. De plus, le raccordement du câble de la carte d'extension M.2 NVMe PCIe avec le SSD M.2 NVMe est très compliqué et risqué pour le câble lui-même. Une autre limitation importante est que cette solution ne permet pas d'utiliser le boîtier, car le SSD M.2 ne rentre pas et ne peut même pas être fixé.

Differentes options pour optimiser l'alimentation ont été explorées : utiliser un HUB USB externe et auto alimenté ; brancher un câble Ethernet au lieu d'utiliser un Dongle Wifi ; allumer le ventilateur dès le démarrage du nano ordinateur ; et l'option de fournir 6Amp directement supportée par la carte mère via les pins ; explorer les solutions sur les forums de discussion. colorred Ref : <https://www.kingston.com/en/community/articledetail/articleid/48543> <https://geekworm.com/products/nvidia-jetson-nano-nvme-m-2-ssd-shield-t100-v1-1>

À noter que la carte T100 est discontinued et remplacée par T130

Caméra

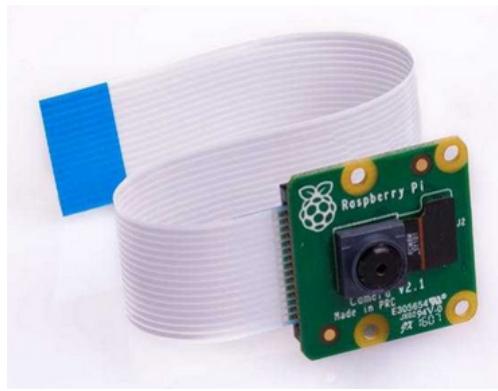


FIGURE 19 – Caméra

L'objectif du nano ordinateur est d'être utilisé pour détecter continuellement les délimitations

de la piste cyclable. Il est évident qu'une caméra doit donc faire partie du système et faire partie de l'évaluation des performances. Néanmoins, durant le déroulement de l'essai, la caméra sera très peu utilisée. En effet il n'est pas évident d'être dans un mode de développement directement sur le terrain. Un matériel vidéo virtuel sera utilisé pour simuler la caméra et alimenter l'inférence avec des vidéos préenregistrées, permettant ainsi d'évaluer les performances de l'inférence avec des vidéos, même si d'un point de vue performance matérielle l'utilisation ne sera pas équivalente. Les performances matérielles de l'inférence en temps réel seront évaluées avec la caméra, même si la vue de la caméra n'est pas la piste cyclable, ce qui n'est pas important pour ce test, peu importe ce qui est détecté.

Ventilateur



FIGURE 20 – Ventilateur

Un système informatique a besoin d'un ventilateur pour évacuer la chaleur produite par ses processeurs et les autres éléments électroniques, et éviter une faute opérationnelle et des bris de matériel. L'objectif du nano ordinateur étant d'être opérationnel continuellement, et ses éléments étant contenus dans un boîtier, il est encore plus indispensable d'installer un ventilateur. Le ventilateur choisi a pu être installé dans le boîtier, même si le boîtier ne possède de support pour le fixer. Le ventilateur est capable de démarrer automatiquement au besoin, mais il est volontairement démarré manuellement dès que le nano ordinateur est démarré. Cela évite que la chaleur ne s'accumule, qu'elle soit tout de suite ventilée à l'extérieur, évitant un risque de surchauffe, la capacité du ventilateur étant tout de même limité (petit modèle).

Hub USB externe 3.0 4 ports



FIGURE 21 – Hub USB 3.0 externe autoalimenté

Le nano ordinateur comprend un hub USB 3.0 4 ports internes, les 4 ports étant connectées via le même contrôleur. Ce hub consomme de l'énergie pour alimenter les périphériques qui y sont connectés, comme un SSD interne ou un dongle Wifi, et gérer l'échange de données. Afin de minimiser les besoins en alimentation et optimiser le plus possible le transfert de données, la souris, le clavier et le dongle USB ont été branchés à un hub USB 3.0 externe autoalimenté. Malheureusement cette option complexifie le déploiement sur le terrain du nano ordinateur. L'alternative pour s'en passer est d'utiliser un câble Ethernet, PoE préférablement, à la place d'un dongle Wifi qui est très gourmand en termes de besoin en alimentation, et chauffe rapidement.

Réseau Internet

Le nano ordinateur comprend un contrôleur Ethernet pour brancher un câble réseau et se brancher sur Internet. Selon la configuration de la carte mère, le nano ordinateur peut hériter de l'alimentation via Ethernet (PoE), via la broche J38. Il comprend aussi un port PCIe interne qui permet de brancher une carte d'extension Wifi. L'autre alternative étant de passer par un dongle USB Wifi, ou un périphérique Wifi externe connecté au port USB.

Dans le cadre de cet essai, le périphérique Wifi externe USB a été utilisé en premier puisque déjà disponible. Malheureusement les performances étaient assez décevantes, le réseau Wifi à la maison n'étant pas non plus très performant dans la pièce où le nano ordinateur était installé (table de la cuisine). Un débit d'environ 5Mbits était disponible. Par curiosité un dongle USB Wifi a été acquis, mais autant décevant. La meilleure alternative pour améliorer le déroulement de l'essai a été de tirer un câble Ethernet et d'installer un routeur secondaire, et de brancher le nano ordinateur à ce nouveau routeur. L'accès Internet a été plus stable et de bien meilleure qualité, la connexion étant d'environ 11Mbs.

Le PoE n'a pas été évalué.

Préparation de l'unité de stockage

Le nano ordinateur est conçu pour fonctionner avec un système d'exploitation hébergé sur une carte micro-SD. Il existe différentes cartes micro-SD, et certaines sont beaucoup plus performantes

que les autres. Malheureusement les cartes micro-SD ne sont pas destinées à exécuter un système d'exploitation à temps plein, et leur espérance de vie reste très limitée. Étant donné que l'objectif du nano ordinateur est d'être en service continue à l'extérieure, l'utilisation un disque SSD interne comme alternative semble logique.

Carte micro-sd

Il existe différentes cartes micro-SD, de multiples constructeurs, et pour différents usages, mais généralement destiné pour stocker des images et vidéos directement par les appareils multimédias. Leur conception est faite pour la manipulation de gros blocs de données, et non des petits fichiers. Trois cartes micro-SD seront évaluées : la carte micro-SD 64Gb EVO Plus (rouge ; Samsung), 64Gb EVO Select (verte ; Samsung), 32Gb Ultra (blanche ; ScanDisk).

Disque SSD

Pour un appareil destiné à être continuellement en service et à l'extérieure, l'unité de stockage doit être non seulement performante, mais aussi endurante. Un disque SSD interne pour un nano ordinateur est soit une carte d'extension M.2 NVMe ou SATA (selon la carte d'extension), connecté au port PCIe ou USB. Les SSD internes Samsung 970 EVO 250GB NVMe M.2 et Samsung 860 EVO M.2 500GB SATA seront évalués. À noter qu'une carte micro-SD est tout de même nécessaire pour "bootstrapper" le système d'exploitation. Il n'est pas nécessaire d'avoir une carte micro-SD performante puisqu'elle n'est utilisée que pour démarrer le système qui se trouve sur le SSD interne.

Configuration du système d'exploitation

La première fois que le système démarre, le système Ubuntu Linux For Tegra (L4T) doit être configuré avec toutes les options personnalisées (langue, clavier, timezone, etc.).

Installation & compilation des librairies pour l'inférence

Les librairies pour la segmentation sémantique d'images et de vidéos via l'inférence de modèles déjà préparée sont mises à disposition par NVIDIA via un projet dans GitHub. La documentation pour l'installation et l'inférence est disponible directement dans la page GitHub.

Installation d'un matériel vidéo virtuel 'v4l2loopback'

L'inférence fournie par NVIDIA est conçue pour utiliser la caméra du nano ordinateur. Ce qui n'est pas forcément "pratique" pour évaluer la segmentation sémantique d'une vidéo d'une piste cyclable. Heureusement un matériel vidéo virtuel permet de simuler la caméra et d'alimenter l'inférence avec une vidéo enregistrée, au lieu de la caméra. La contrepartie concerne l'évaluation des performances : en effet la caméra demande plus de puissance au nano ordinateur que le simulateur logiciel.

Tests

Afin de s'assurer que le nano ordinateur est prêt pour être évalué, des tests matériels et logiciels sont effectués une fois le système monté et stabilisé. Les résultats des tests servent de référence pour évaluer l'état de santé du nano ordinateur.

4.6.2 Collecte des données

TODO

4.6.3 Mise en place des solutions logicielles

Jetson Nano

Le nano ordinateur est destiné à l'inférence. NVIDIA fournit tout un système d'installation, qui est nommé JetPack, et qui contient un système d'exploitation basée sur Ubuntu, Linux For Tegra L4T), et toutes les librairies nécessaires pour l'inférence, tel que le compilateur CUDA, et le SDK TensorRT pour générer le format interopérable ONNX pour l'inférence.

L'architecture du nano ordinateur est ARM 64 bits (aarch64), ce qui le limite pour certaines portabilités de librairies, surtout dans le domaine assez restreint de la recherche, ou l'architecture la plus populaire et portable est x86-64.

Il est composé d'un quad-core ARM Cortex-A57, qui est conçu pour ce genre de nano ordinateur, comme le Raspberry Pi.

Les performances GPU sont faibles, 0.5 TFLOPs (16FP; 16bits/2 bytes floating points). Par comparaison la PlayStation 4 Pro (2016) supporte +4 TFLOPs.

La mémoire est limitée à 4GB.

Les autres ports importants sont le port pour une carte micro-SD, un port Ethernet 10/100/1000Mbs, un port HDMI, un hub USB 3.0 4 ports, un connecteur pour une caméra, et un port PCIe.

Compute Canada

Le nano ordinateur est destiné à l'inférence, et non l'entraînement de modèles. Il n'est pas non plus destiné à être un environnement de développement. Un autre environnement de travail est donc nécessaire pour développer, et doit posséder les capacités matérielles (GPUs, mémoires, espace de stockage) et logicielles (librairies) pour entraîner un modèle. Heureusement mon directeur de projet m'a introduit à Compute Canada, ou Calcul Québec. Compute Canada fournit un espace de travail puissant aux chercheurs et aux universitaires. Il n'est pas évident de posséder à la maison un environnement permettant de faire de l'apprentissage profond. Ce que je ne pouvais faire avec le nano ordinateur, j'ai pu le faire dans l'environnement de Compute Canada, tel que compiler un fork de torchvision, réentraîner des modèles, générer des onnx. Avoir accès à cet environnement de travail a été un élément déterminant dans le cadre de cet essai.

Compte Compute Canada

Compute Canada mets à disposition des ressources matérielles puissantes et l'accès à des librairies de haute technologie telle que pour l'apprentissage profond, permettant d'avoir un environnement de travail professionnel et performant rapidement. Les ressources matérielles à disposition sont des grappes de serveurs, de CPUs et GPUs de différents types, ainsi que de l'espace de stockage. Les librairies sont disponibles via un repository privé, et lorsque certaines étaient manquantes (onnx et onnxruntime), j'ai fait une demande par courriel. L'administrateur a pu rendre disponible l'une des deux (onnx), la seconde (onnxruntime) étant beaucoup plus complexe à installé, pour l'avoir tenté sur le nano ordinateur.

L'autre avantage de l'environnement de Compute Canada est la mise à disposition de Jupyter Notebook, afin de tester rapidement du code Python. Par contre il n'est pas conseillé d'exécuter du code nécessitant des délais, tels que l'entraînement d'un modèle.

L'un des irritants est de ne pas pouvoir exécuter un container docker tel quel. Il faut le convertir au format Singularity. Dans le cadre du projet cela m'aurait facilité la tâche, car NVIDIA fournit des docker prêt à l'utilisation pour le réentraînement. Je n'ai malheureusement pas pris le temps et la chance de convertir un container docker au format Singularity. Je ne sais pas si c'est une activité assez simple ou complexe, mais du peu que j'ai lu cela semble assez "rapide".

Jupyter Notebook

Le besoin de tester du code Python est toujours nécessaire. La console Python n'étant vraiment pas conviviale, un environnement Jupyter Notebook est un compromis incontournable. Heureusement Compute Canada fournit un accès à des notebooks depuis Internet, permettant en plus d'hériter de leur environnement de travail. Il est à noter que les notebooks n'ont pas été utilisés pour entraîner un modèle ou générer des onnx, mais de tester du code Python simple, comme visualiser des images, transformer des tensors, et évaluer la segmentation prédictive générée avec le "ground truth".

NVIDIA

Compte NVIDIA

NVIDIA mets à disposition tout un écosystème éducatif permettant aux développeurs et aux chercheurs d'obtenir de l'aide au sujet de leur produit et librairies. Dans le cadre de l'essai, un compte NVIDIA a été créé, permettant d'accéder au forum de développeurs, et les containers docker par exemple. Il est aussi possible d'accéder à du matériel éducatif grâce à l'institut DeepLearning de NVIDIA, dont l'accès a été commandité par mon directeur de projet. Le forum de développeurs a été un outil très utile dans le cadre de ce projet, car le dépôt d'une question m'a permis de me débloquer. Je n'étais pas capable de régénérer l'ONNX à partir du code source et de la documentation fournie par NVIDIA pour un modèle FCN. Le développeur principal de l'application a répondu et m'a guidé dans la résolution du problème. Les autres ressources ont eu un impact limité dans le cadre de ce projet, puisque par exemple le container docker et DIGITS n'ont pas pu être utilisé. Le code source des modèles est disponible sans nécessiter de compte, de même que les SDKs Jetpack.

NVIDIA DIGITS

NVIDIA fournit aux développeurs un environnement visuel permettant de réentraîner les modèles FCN qu'ils fournissent avec leurs propres dataset. Cet environnement se nomme DIGITS. Malheureusement il est nécessaire d'avoir son propre matériel, le système d'exploitation Ubuntu 18.04 LTS, et très recommandé d'avoir au moins un GPU et un ordinateur performant. Ce qui n'est malheureusement pas mon cas. DIGITS ne s'installe pas sur le nano ordinateur, ni sous Windows, ni même un Ubuntu sous windows (WSL). Cette option a donc été abandonnée rapidement.

Docker NVIDIA

NVIDIA fournit aux développeurs des containers docker, avec tout ce qui est nécessaire pour réentraîner un modèle et régénérer un ONNX, par exemple. Malheureusement la capacité du nano-ordinateur ne permet pas de travailler efficacement avec un container docker, la mémoire du nano ordinateur étant limité à 4Gbit, le nano-ordinateur devient sans réponse, nécessitant un redémarrage forcé (hard-reboot). Cette option a donc été aussi abandonnée rapidement.

NVIDIA DeepStream

Durant le déroulement de l'essai, NVIDIA a mis à disposition un environnement d'apprentissage profond, nommé "DeepStream", facilitant la conception et la génération de modèles, jusqu'à l'inférence. Cet outil n'a pas été évalué, mais pourrait être un outil alternatif pour réentraîner un modèle.

4.7 Évaluation

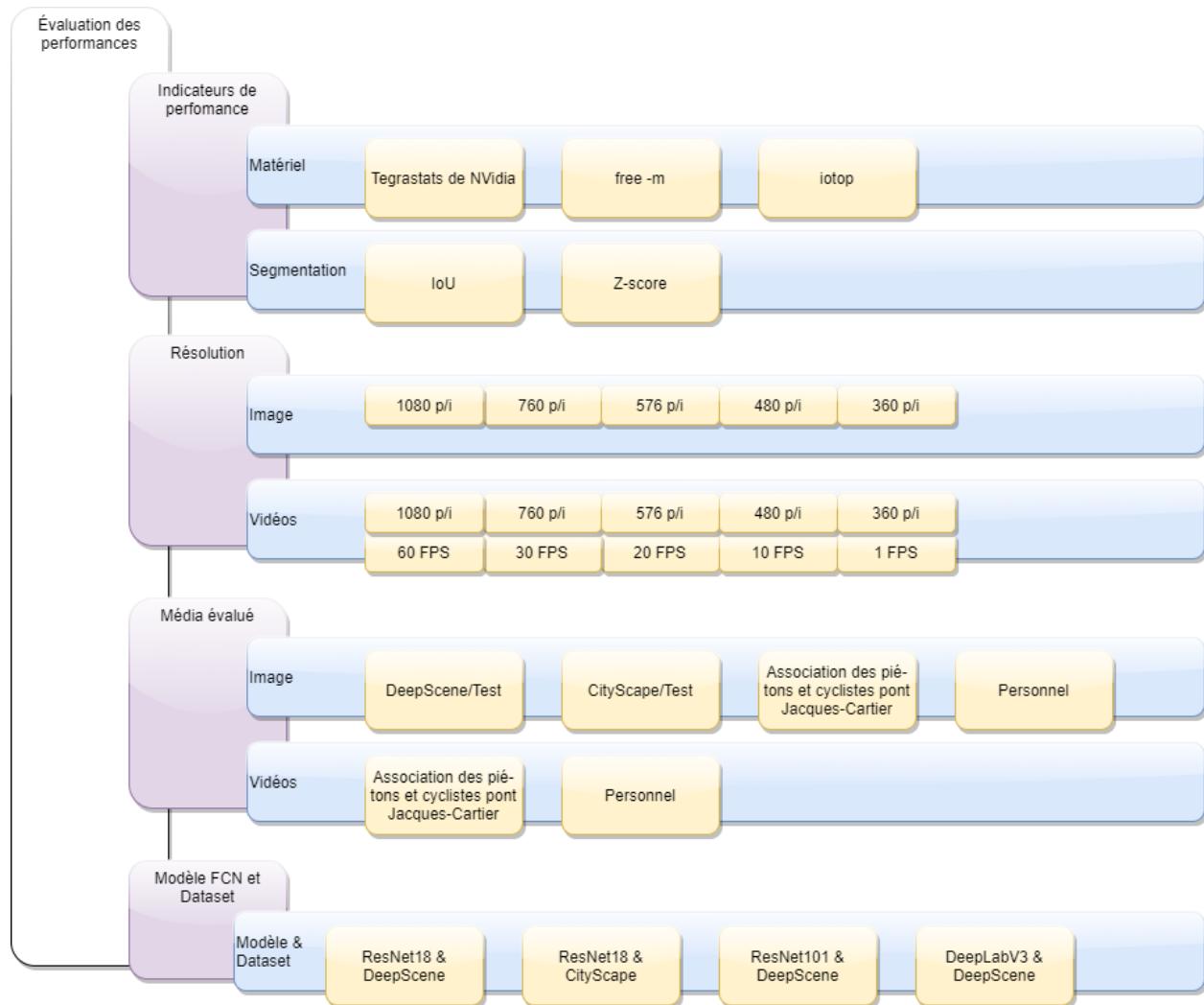


FIGURE 22 – Éléments pour l'évaluation des performances

4.7.1 Stratégie de test

L'objectif principal de l'essai est de déterminer la capacité et les limites du nano ordinateur d'inférer en temps réel des modèles de réseau de neurones à convolution entier pour la segmentation sémantique de vidéos. La stratégie qui sera appliquée sera de tester avec divers modèles et divers niveaux de qualité vidéos, en espérant trouver le compromis qui répond le mieux à cet objectif.

1. Afin de s'assurer du bon fonctionnement du nano ordinateur et d'avoir des résultats de référence propre à notre environnement, l'inférence sera testée avec des modèles existants et pré entrainés pour la segmentation sémantique, avec les images et les vidéos provenant des références, et dont les caractéristiques et les résultats sont disponibles.
2. En espérant que les tests de l'étape #1 précédente donnent les résultats documentés dans les articles de références, ils seront repris avec les mêmes modèles, mais avec les images

et les vidéos du site d'étude possédant la meilleure qualité acquise (1080p/i, 30FPS). Les données sources (images et vidéos) devront subir certains prétraitements à ce effet, afin de répondre aux requis des modèles.

3. Selon les résultats de l'étape #2, les tests se concentreront sur l'inférence avec des vidéos, en réduisant progressivement la résolution (760p/i, 576p/i, 480p/i, 360p/i) et le nombre d'images par seconde (20FPS, 10FSP, 1FPS).
4. Les étapes intermédiaires de l'étape #3 précédente seront de 1) valider les résultats de l'inférence avec des images avant de tester avec les vidéos, et 2) évaluer si les modèles de réseaux de neurones à convolution entiers doivent et/ou peuvent être adaptés facilement, en tenant compte de l'échéancier de l'essai, et ce afin de répondre à l'objectif principal.

4.7.2 Stratégie de collecte des indicateurs de performance

La méthodologie de la collecte des indicateurs est la suivante :

- La collecte est démarrée après un démarrage frais.
- Elle est consolidée dans un seul fichier.
- La date et l'heure du début de la collecte sont précisées.
- Au moment précis de début et de fin de chacun des tests, une entrée spéciale est ajoutée dans le fichier de collecte des indicateurs avec la date et l'heure, un libellé et une description. Ce point de contrôle (checkpoint) est fait grâce à une commande "shell" qui vient ajouter une trace dans le fichier de collecte des indicateurs.
- Chaque indicateur est collecté aux 5 secondes. Ce délai est réévalué au besoin, selon les fréquences des pics observées.

Ensuite tout au long des tests de validations matériels et logiciels les indicateurs sont collectés.

Chaque indicateur est une paire de clé=valeur. Il existe le même nombre d'indicateurs à tout moment. La date et l'heure sont un champ, ainsi que la source (tegrastats, System Monitor, user) et le champ descriptif.

Avant tout début de tests, la collecte est démarrée. Cela permet de prendre une base de référence sans aucune charge.

Les indicateurs de performances sont collectés avant les tests matériels et logiciels lors du montage du nano ordinateur. Cela permet d'observer le "payload" possible d'un équipement ou d'un logiciel, sans lien avec l'inférence. Par exemple Chromium. Ou la caméra avec GStreamer.

Ensuite les tests avec l'inférence débutent. D'abord par des tests de régression, pour s'assurer que l'environnement est toujours fonctionnel et retourne les mêmes résultats que les tests similaires précédents, que nos attentes sont respectées.

Puis par des tests ciblés, en commençant par l'inférence avec des images de diverses résolutions. Et pour finir par l'inférence avec des vidéos de diverses résolutions.

Les indicateurs collectés permettent de créer des graphiques qui montrent la progression de chacun avant, pendant et après le test. Le min, max et avg peut être extrait.

Les performances matérielles du Jetson Nano sont évaluées grâce au System Monitor d'Ubuntu et les métriques tegrastats fournis par NVidia.

Les performances de la segmentation sont évaluées grâce au IoU et au z-score pour la classe du chemin / route. Une fonction Python est utilisée. Les fonctions IoU et le z-score utilisent l'image prédite (générée par le modèle FCN) et l'image Ground Truth. Les images originales sont donc

présélectionnées selon leur intérêt et l'image Ground Truth créée. L'image prédite et Ground Truth doivent utiliser la même palette de couleurs et doivent être de la même résolution. L'image Ground Truth est créée à la main avec l'éditeur Gimp. Comme la segmentation de l'image prédite par le modèle de NVidia est très carrée, l'image Ground Truth ne sera pas précise au pixel prêt. Le besoin est d'évaluer et non d'entrainer, l'importance de la précision de la classification est moindre.

4.7.3 Segmentation avec des images

TODO

4.7.4 Segmentation avec des vidéos

TODO

4.8 Adaptation

TODO

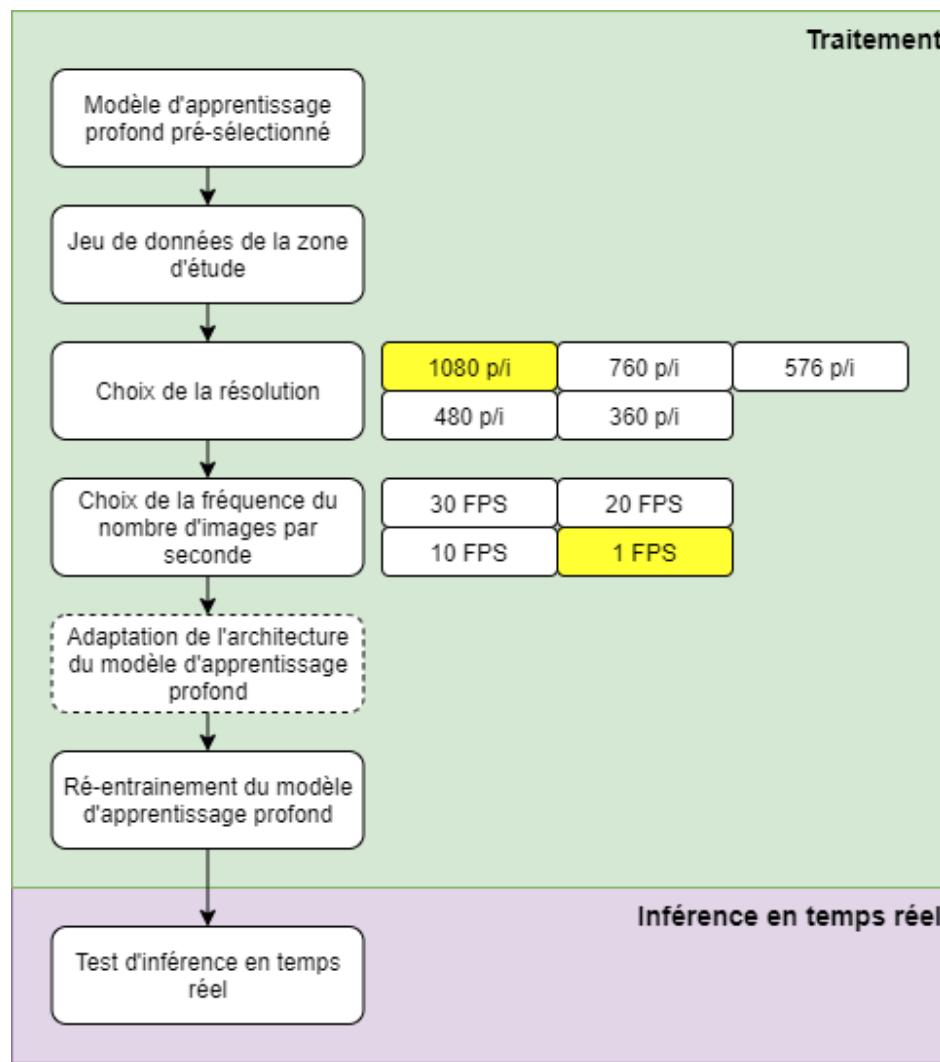


FIGURE 23 – Méthodologie du traitement et adaptation

4.8.1 Choix du modèle de l'architecture FCN

TODO

4.8.2 Adaptation du modèle

TODO

4.8.3 Ré-entraînement du modèle

TODO

5 Résultats

TODO

Voici le plan qui est utilisé pour rédiger les résultats.

- Pour chaque modèle et résolution utilisés, la segmentation sémantique de certaines images et vidéos sera présentée. La segmentation qui a réussi, celle qui est moins précise, et celle qui a échoué seront soulignées. Un résumé du % de succès vs des échecs sera fait, selon les modèles et les résolutions.
- En complément de la section précédente, les performances du Jetson nano pour les divers scénarios de test seront résumés avec différents indicateurs. Ceux qui ont échoué ou n'ont pas été possibles en raison des limitations du nano ordinateur seront indiqués.
- Enfin les performances de l'inférence et des modèles de réseaux de neurones pour la segmentation sémantique seront listées. Des indicateurs de performance classiques et tirés de la littérature seront utilisés.

Pour tester les performances de la micro-sd et du disque SSD interne M.2 NVMe, l'utilitaire hdparm peut être facilement utilisé. Les tests montrent que le SSD interne est plus de 8 fois plus efficace que la micro-sd pour la lecture de données.

TABLE 3: Comparaison des performances du "data read" entre un SSD M.2 NVMe et une micro-sd

Disk reads	MB	sec	MB/sec
SSD	1004	3	334.15
micro-sd	122	3.03	40.22

6 Interprétation et discussion des résultats

TODO

Voici le plan qui est utilisé pour rédiger le cadre théorique au sujet du nano ordinateur.

- Interprétation et discussion des résultats de la segmentation sémantique en temps réel de la zone d'études ;
- Interprétation et discussion des performances du nano ordinateur "Jetson nano" pour l'inférence en temps réel dans le contexte de l'essai.
- Interprétation et discussion des performances de l'inférence et des modèles de réseaux de neurones pour la segmentation sémantique en temps réel de la zone d'étude.

7 Conclusion et recommandations

TODO

Voici le plan qui est utilisé pour rédiger la conclusion.

- Synthèse des réussites et des échecs de l'essai par rapport aux objectifs ;
- Synthèse des capacités et des limites du nano ordinateur "Jetson nano" pour l'inférence en temps réel à des fins de segmentation sémantique de vidéos ;
- Synthèse des capacités et des limites des modèles de réseaux de neurones pour la segmentation sémantique en temps réel de la zone d'études ;

— Recommendations ;

Bibliographie

- [1] S. ABOUZAHIR, M. SADIK et E. SABIR. “IoT-Empowered Smart Agriculture : A Real-Time Light-Weight Embedded Segmentation System”. In : *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2017), p. 319-332. DOI : 10.1007/978-3-319-68179-5_28.
- [2] A. BEAM. *Deep Learning 101 - Part 1 : History and Background*. 2017. URL : https://beamandrew.github.io/deeplearning/2017/02/23/deep_learning_101_part1.html.
- [3] M. BERNAS, B. P{\textbackslash}LACZEK et A. SAPEK. “Edge Real-Time Medical Data Segmentation for IoT Devices with Computational and Memory Constraints”. In : *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2017), p. 119-128. DOI : 10.1007/978-3-319-67077-5_12.
- [4] B. BLANCO-FILGUEIRA et al. “Deep Learning-Based Multiple Object Visual Tracking on Embedded System for IoT and Mobile Edge Computing Applications”. In : *IEEE Internet of Things Journal* (juin 2019), p. 5423-5431. DOI : 10.1109/JIOT.2019.2902141.
- [5] C. P. CHONG, C. A. T. SALAMA et K. C. SMITH. “Real-Time Edge Detection and Image Segmentation”. In : *Analog Integrated Circuits and Signal Processing* (1992), p. 117-130. DOI : 10.1007/BF00142412.
- [6] M. COPEL. *What's the Difference Between Deep Learning Training and Inference ?* Août 2016. URL : <https://blogs.nvidia.com/blog/2016/08/22/difference-deep-learning-training-inference-ai/>.
- [7] T. DETTMERS. *Deep Learning in a Nutshell : History and Training*. Déc. 2015. URL : <https://devblogs.nvidia.com/deep-learning-nutshell-history-training/>.
- [8] F. DUSTIN. *Realtime Semantic Segmentation on Jetson Nano in Python and C++*. Oct. 2019. URL : <https://www.linkedin.com/pulse/realtime-semantic-segmentation-jetson-nano-python-c-dustin-franklin>.
- [9] Association des piétons et cyclistes du pont JACQUES-CARTIER. *PontJacques-Cartier365.com*. 2020. URL : <http://pontjacquescartier365.com>.
- [10] Association des piétons et cyclistes pont JACQUES-CARTIER. *Flickr Association des piétons et cyclistes pont Jacques-Cartier*. 2020. URL : <https://www.flickr.com/photos/pontjacquescartier>.
- [11] JIACONDA. *A Concise History of Neural Networks*. Avr. 2019. URL : <https://towardsdatascience.com/a-concise-history-of-neural-networks-2070655d3fec>.
- [12] J. Y. KOH. *Model Zoo - Deep Learning Code and Pretrained Models for Transfer Learning, Educational Purposes, and More*. 2018. URL : <https://modelzoo.co/>.
- [13] A. KURENKOV. *A 'Brief' History of Neural Nets and Deep Learning*. 2015. URL : <https://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/>.

- [14] J. LONG, E. SHELHAMER et T. DARRELL. “Fully Convolutional Networks for Semantic Segmentation”. In : *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2015, p. 3431-3440. DOI : 10.1109/CVPR.2015.7298965.
- [15] M. MODY et al. “Low Cost and Power CNN/Deep Learning Solution for Automated Driving”. In : *Proceedings - International Symposium on Quality Electronic Design, ISQED*. 2018, p. 432-436. DOI : 10.1109/ISQED.2018.8357325.
- [16] T. NGUYEN et al. “MAVNet : An Effective Semantic Segmentation Micro-Network for MAV-Based Tasks”. In : *arXiv :1904.01795 [cs]* (juin 2019). URL : <http://arxiv.org/abs/1904.01795>.
- [17] NVIDIA. *Jetson Nano*. Mar. 2019. URL : <https://developer.nvidia.com/embedded/jetson-nano>.
- [18] NVIDIA. *Jetson Nano : Deep Learning Inference Benchmarks*. Avr. 2019. URL : <https://developer.nvidia.com/embedded/jetson-nano-dl-inference-benchmarks>.
- [19] D. PATHAK et M. EL-SHARKAWY. “Architecturally Compressed CNN : An Embedded Realtime Classifier (NXP Bluebox2.0 with RTMaps)”. In : *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. Jan. 2019, p. 0331-0336. DOI : 10.1109/CCWC.2019.8666495.
- [20] PJCCI. *Fiche de la piste multifonctionnelle du pont Jacques-Cartier*. 10 oct. 2018. URL : https://jacquescartierchamplain.ca/wp-content/uploads/2018/10/IMG_Fiche_piste-multi_pont_JC_FR_vfinale_web__2018-10-10.pdf.
- [21] PJCCI. *Rapport post-mortem sur le projet pilote d'entretien hivernal de la piste multifonctionnelle du pont Jacques-Cartier*. 10 oct. 2018. URL : https://jacquescartierchamplain.ca/wp-content/uploads/2018/10/RPP_piste_PJC_2018-10-10-1.pdf.
- [22] N. SHARMA, M. SHAMKUWAR et I. SINGH. *The History, Present and Future with IoT*. Intelligent Systems Reference Library. Springer Science et Business Media Deutschland GmbH, 2019. ISBN : 18684394 (ISSN). DOI : 10.1007/978-3-030-04203-5_3.

8 Annexes

TODO