

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

**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

□ TODO

Je tiens à remercier ...

Table des matières

Liste des figures	III
Liste des tableaux	III
Liste des abréviations	IV
1 Introduction	1
1.1 Mise en contexte	1
1.2 Problématique	2
1.3 Objectifs	4
2 Cadre théorique	5
2.1 Le nano ordinateur	5
2.2 La segmentation sémantique	6
3 Matériel et méthodes	8
3.1 Site d'étude	8
3.2 Modèles et jeux de données	11
3.3 Matériel et logiciels	15
3.4 Méthodologie	21
3.5 Documentation	23
3.6 Recherche	23
3.6.1 Revue de littérature	23
3.6.2 Étude du site d'implantation	24
3.6.3 Sélection des données et des modèles de réseaux de neurones	25
3.6.4 Choix de l'équipement pour le nano ordinateur	26
3.6.5 Exploration des solutions logicielles	26
3.7 Environnement de travail	26
3.7.1 Préparation du nano ordinateur	26
3.7.2 Collecte des données	34
3.7.3 Mise en place des solutions logicielles	35
3.8 Évaluation	37
3.8.1 Stratégie de test de l'inférence	39
3.8.2 Stratégie de collecte des indicateurs de performance matériel	39
3.8.3 Résolutions évaluées	40
3.8.4 Segmentation avec des images	40

3.8.5	Segmentation avec des vidéos	42
3.9	Adaptation	44
3.9.1	Choix du modèle de l'architecture FCN	45
4	Résultats	46
4.1	Performances matérielles	46
4.1.1	Stockage de données	46
4.1.2	Performances système	46
4.2	Performances de l'inférence	53
4.2.1	Images	53
4.2.2	Vidéos	54
4.3	Adaptation	55
5	Interprétation et discussion des résultats	56
5.1	Performances matérielles	56
5.1.1	Stockage de données	56
5.1.2	Performances système	56
5.2	Performances de la segmentation	59
5.2.1	Images	59
5.2.2	Vidéos	59
6	Conclusion et recommandations	59
6.1	Objectif principal	59
6.2	Limites	61
6.2.1	Limites matérielles	61
6.2.2	Limites applicatives	61
6.3	Optimisation	61
6.3.1	Optimisation matérielle	61
6.3.2	Optimisation logicielle	62
6.4	Accès distant	63
6.5	Documentation	63
Références		64
7	Annexes	67
7.1	Exemples de nano ordinateurs qui supportent les SDK pour l'IA	67
7.2	Communication avec l'Association des Piétons et Cyclistes du Pont Jacques-Cartier	67

Liste des figures

1	Segmentation semantic	6
2	Relation entre Intelligence Artificielle, Apprentissage Machine et Apprentissage Profond	7
3	Vue aérienne du pont Jacques-Cartier	9
4	Carte du site d'implantation	10
5	Schéma de la configuration de la piste multifonctionnelle	11
6	Carte mère Jetson Nano de NVIDIA	16
7	Diagramme de l'architecture du NVIDIA JetPack	17
8	Organigramme de la méthodologie à haut niveau	21
9	Organigramme des détails de la méthodologie pour évaluer les performances	22
10	Organigramme des détails de la méthodologie pour évaluer les performances après une phase d'adaptation	22
11	Préparation du nano ordinateur	27
12	Montage du nano ordinateur	27
13	Carte mère, accessoires et périphériques du nano ordinateur	28
14	Éléments pour l'évaluation des performances	38
15	Diagramme d'architecture de la segmentation d'une vidéo	43
16	Diagramme d'architecture de la segmentation avec la caméra	43
17	Méthodologie du traitement et adaptation	45
18	Les périodes du diagramme des performances système	48
19	Diagramme des performances système : la fréquence	49
20	Diagramme des performances système : la mémoire	49
21	Diagramme des performances système : le I/O total en % de la segmentation	50
22	Diagramme des performances système : le I/O en KBytes de la segmentation	50
23	Diagramme des performances système : le I/O total du disque en KBytes	51
24	Diagramme des performances système : les températures	51
25	Diagramme des performances système : la consommation	52
26	Segmentation sémantique de l'image b1-09517 générée par le modèle	53
27	Segmentation sémantique de l'image b378-61 générée par le modèle	54

Liste des tableaux

1	Tableau des données	13
2	Solutions logicielles de l'essai	18

3	Cartes microSD	30
4	Résolutions et images par seconde (FPS) qui seront évaluées	41
5	Classes et palettes de couleur	41
6	Comparaison des performances du "data read" entre un SDD M.2 NVMe et une microSD	46
7	Résolutions et images par seconde (FPS) testés	55
8	Comparaison des trois nano ordinateurs supportant les SDK pour l'IA	67

Liste des abréviations

AM Apprentissage Machine.

AP Apprentissage Profond.

APC-PJC Association des Piétons et Cyclistes du Pont Jacques-Cartier.

CPU Central Processing Unit.

CSV Comma Separated Value.

FCN Fully Convolutional Network.

FPS Frames per second.

GPU Ggraphics Processing Unit.

GT Ground truth.

IA Intelligence Artificielle.

IoT Internet of Things.

IoU Intersection over Union.

L4T Linux for Tegra.

PJCCI Les Ponts Jacques Cartier et Champlain Incorporée.

SDK Software Development Kit.

SSD Solid State Drive.

TFLOPS teraFLOPS.

1 Introduction

1.1 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 nano ordinateurs) à différents endroits du pont. Chacun de ces nano ordinateurs possède trois différents types de capteurs : vision ; son ; et météorologiques (température, humidité, etc.). Chaque nano ordinateur 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 nano ordinateur du fabricant NVIDIA, le Jetson Nano (NVIDIA, 2019a), 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" (CHONG et al., 1992) (DETTMERS, 2015) (BEAM, 2017) (JIACONDA, 2019) (ZHENG et al., 2020) (KURENKOV, 2015).

Les systèmes informatiques performants sont de plus en plus miniatures, on parle de nano ordinateurs et des objets connectés ("Internet of Things" ou "IoT") (BLANCO-FILGUEIRA et al., 2019) (SHARMA et al., 2019). 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 (ZHENG et al., 2020) (BERNAS et al., 2017) (ABOUZAHIR et al., 2017) (BLANCO-FILGUEIRA et al., 2019).

Les réseaux de neurones ont aussi très rapidement progressé depuis 2012 (BEAM, 2017), permettant d'offrir des alternatives aux solutions de détection et de classifications, entre autres (PATHAK et EL-SHARKAWY, 2019). Les réseaux de neurones convolutifs entiers ("FCN" en anglais, pour "Fully Convolutional Network") sont les derniers à avoir émergé ("state-of-art") (ZHENG et al., 2020) et à profiter au domaine de la vision et de la détection d'objets (NGUYEN et al., 2019) (ZHENG et al., 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 nano ordinateurs (LONG et al., 2015) (BLANCO-FILGUEIRA et al., 2019). 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 (MODY et al., 2018). Leur application par des nano ordinateurs est un défi en raison de la faible consommation d'énergie (Watts) et de la puissance de calcul limité de ces derniers (COPEL, 2016).

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.

1.2 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 clima-

tiques 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. 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 nano ordinateur 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 (NVIDIA, 2019b) (NGUYEN et al., 2019) (CHONG et al., 1992), 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 requiert de l'expérience. C'est un processus d'essais-erreurs qui est très coûteux 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 (KOH, 2018). La recherche dans la littérature permet heureusement de limiter les choix et donner des pistes (ZHENG et al., 2020) (NGUYEN et al., 2019) (NVIDIA, 2019b). La problématique de la conception existe toujours, car

le modèle a besoin d'être étudié, adapté et ré entrainé, jusqu'à l'obtention de résultats probants. Mais la paramétrisation des hyper paramètres n'est plus nécessaire (supposément), ce qui est très avantageux.

Même si les tests du modèle donnent des résultats très satisfaisants en théorie, la réalité du terrain peut surprendre. Le test du modèle doit se faire dans des conditions réelles avec de nouvelles données (images), celles qui sont captées par le système hôte sur le terrain d'implantation : dans le jargon de l'intelligence artificielle, c'est l'"inférence" (COPEL, 2016) (NVIDIA, 2019b). Il est assez probable que le modèle doive retourner à une phase d'adaptation. De plus, le système hôte, dans notre cas le nano ordinateur NVIDIA Jetson Nano, est conçu avec une architecture matérielle limitée (GPU, CPUs, mémoire, taux de transfert, alimentation) et verra, au besoin, son architecture matérielle adaptée et remise en question.

1.3 Objectifs

L'objectif principal de l'essai est d'évaluer la capacité du nano ordinateur "NVIDIA Jetson Nano" à exécuter, en temps réel, un modèle de réseau de neurones à convolution entier (FCN) permettant la segmentation sémantique d'une vidéo d'une piste multifonctionnelle. Pour y arriver, différents sous-objectifs ont été établis.

Voici en bref les sous-objectifs, avec un peu plus de détails par la suite pour chacun d'eux :

- Évaluer les limites de la plateforme, matérielle et applicative.
- Trouver des moyens d'optimiser la plateforme, au besoin, d'un point de vue matériel, mais aussi applicatif.
- Permettre un accès à distance sécurisé au nano ordinateur.
- Documenter l'approche, les tests, et les résultats ;

Le premier sous-objectif est de déterminer quelles sont les limites de la plateforme, d'un point de vue matériel (GPU, CPUs, 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 (NGUYEN et al., 2019) (ZHENG et al., 2020) (NVIDIA, 2019b). 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'entraîne-

ment 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 nano ordinateur 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, librairies 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.

2 Cadre théorique

2.1 Le nano ordinateur

Le nano ordinateur de cet essai doit être compris comme étant un ordinateur miniature, ayant une taille et des capacités qui lui permettent d'être installé ("embedded system") dans une voiture, un drone, un tracteur ou être accroché à un poteau. Le terme "On the Edge" s'y approprie mieux que "IoT" ("Internet of Things"), puisqu'étant sur le terrain il se trouve directement proche des données, ce qui lui donne l'avantage de pouvoir faire des traitements en temps réel. Les premiers systèmes embarqués reconnus comme tels, sont ceux installés dans le missile Minuteman (1961 ; ref) et la navette Apollo (1960 ; ref). Les avancées technologiques ont permis de les rendre de plus en plus compactes et performants, et le Jetson Nano de NVIDIA est le dernier né de la compagnie NVIDIA permettant d'inférer en temps réel des modèles d'intelligence artificielle, sans ajout de périphériques. Du même constructeur, ses grands frères sont le Jetson Xavier and le Jetson TX2, plus performants, et donc plus onéreux. Son concurrent direct est le Raspberry Pi, mais il nécessite une extension USB Movidius Intel pour l'inférence de modèles IA.

TODO
TODO

Ce qui caractérise principalement un ordinateur miniature, est le fait qu'il soit assez petit pour pouvoir être embarqué dans un système plus gros, tel qu'un robot ou du matériel médical. Son cout est bas, en raison des performances qui sont limitées par une conception répondant à un besoin très spécifique. Tous les éléments matériels requis sont contenus sur une même carte. Une fois installé et paramétré, le système se doit d'être fiable et opérationnel sur le long terme. Mais il doit

aussi être interchangeable, au besoin, rapidement et facilement. La consommation électrique est généralement faible. Étant généralement opérationnel sur le terrain, proche des données, il est responsable d'une tâche bien particulière, qu'il doit accomplir efficacement. Il n'y a généralement pas d'interface utilisateur, et l'accès au système se fait à distance ou via une console. Il est composé de capteurs, au besoin d'une caméra. Le même système peut-être déployé en grande quantité, comme dans le contexte de notre essai, où plusieurs pairs seront déployés le long de la piste multifonctionnelle ; un autre exemple est celui des constellations de nano satellites.

L'annexe 7.1 montrent les nano ordinateurs qui supportent les SDK pour l'IA.

2.2 La segmentation sémantique

La segmentation sémantique d'images ou de vidéos est une technique de télédétection du domaine de la vision par ordinateur. Elle permet de délimiter (segmenter) différentes parties (sémantique) d'une image. Les méthodes de segmentation ont été améliorées ces dernières années par les récentes avancées dans le domaine de l'apprentissage profond.

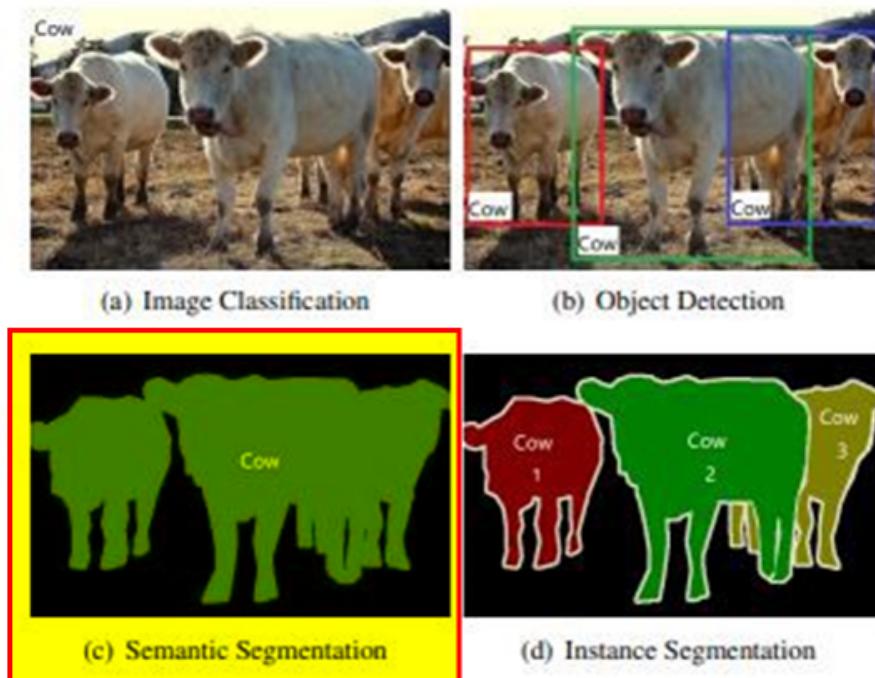


FIGURE 1 – Segmentation semantic(WU et al., 2019, p. 1)

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

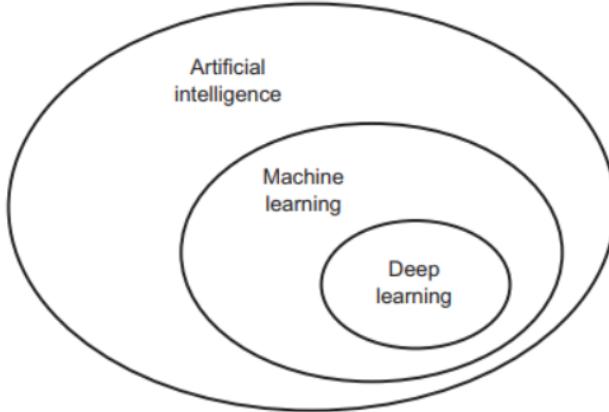


FIGURE 2 – Relation entre Intelligence Artificielle, Apprentissage Machine et Apprentissage Profond (CHOLLET, 2018, p. 4)

Les concepts de l’Intelligence Artificielle (AI) existent depuis les années 1950 (CHOLLET, 2018, p. 4) (ALOM et al., 2018, p. 1), et ont continué à se développer par vague, jusqu’à leur nouvelle popularité des 15 dernières années. En effet, trois raisons principales ont permis à ce domaine de renaître de nouveau (CHOLLET, 2018, p. 20) : 1) la capacité et la puissance des machines ; 2) des jeux de données plus larges ; 3) des algorithmes plus avancés. Les deux moments clés, preuves de cette renaissance, sont : 1) la possibilité d’entrainer des architectures de réseaux de neurones profonds (DNN) (2006) (ALOM et al., 2018, p. 6) ; et 2) l’architecture du réseau de neurones convolutionnels AlexNet permet de gagner le challenge ImageNet contre les approches traditionnelles(ALOM et al., 2018, p. 11).

En 2016 (ALOM et al., 2018, p. 14), l’architecture FCN (Fully Convolutional Network, réseau (de neurones) convolutionnel entier) a permis aux tâches réservées à la segmentation d’images d’être plus efficace que les méthodes traditionnelles de la vision par ordinateur. Cette nouvelle méthode s’applique désormais à tous les domaines connexes à l’analyse d’images, tels que l’imagerie médicale, la conduite autonome de véhicules, la robotique, la télédétection d’images satellites, la sécurité par caméra vidéo, l’agriculture de haute précision. Aujourd’hui, elle peut s’exécuter en temps réel sur des systèmes embarqués proche des données.

3 Matériel et méthodes

3.1 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 (PJCCI, 2018b), la piste multifonctionnelle du pont Jacques-Cartier (figure 3) relie Montréal intramuros, proche de la station de métro De Lorimier / René-Lévesque, et la rive sud, à Longueuil, proche de la station de métro Longueuil et de son terminal de bus. Elle est longue d'une distance de 2.7km et est située d'un seul côté du pont, côté sud (figure 4). Elle est surtout utilisée par les cyclistes, et moindrement par les passants (PJCCI, 2018b).

Sa configuration est bien particulière (PJCCI, 2018a) (figure 5) : 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 3 – Vue aérienne du pont Jacques-Cartier (flickr PJCCI)¹.

1. <https://www.flickr.com/photos/pjcci/6830109134>

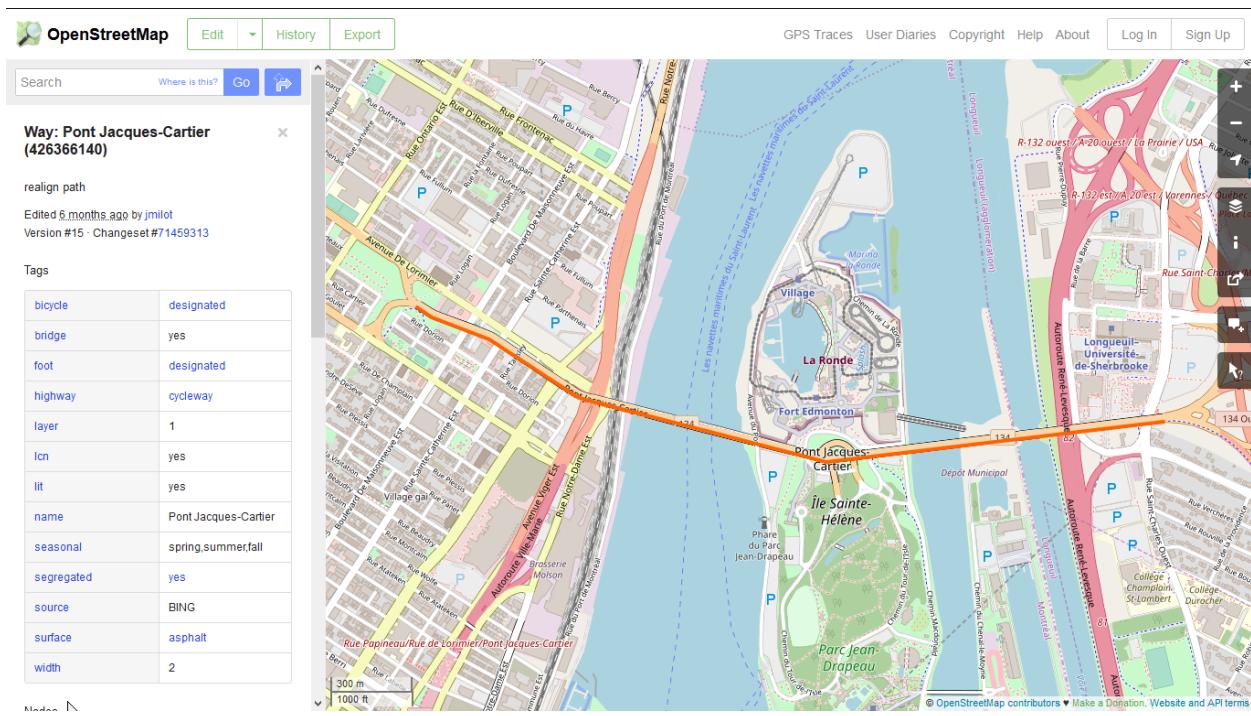


FIGURE 4 – Description de la zone géographique du site d’implantation : le pont Jacques-Cartier et la piste multifonctionnelle en orange sur le pont (copie-écran OpenStreetMap).

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

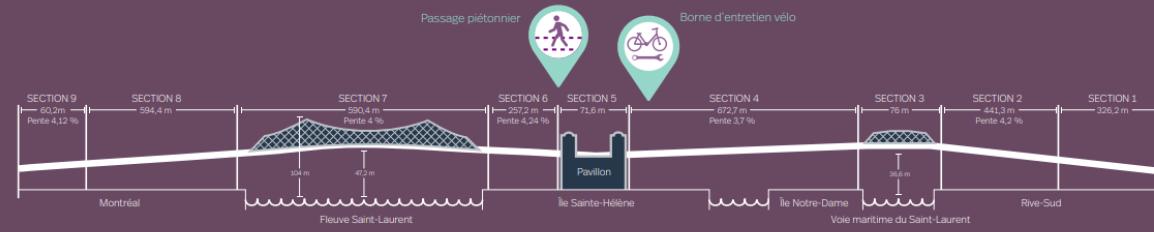
- + Longueur (2,7 km)
- + Dégagement latéral étroit (de 2,5 m à 1,8 m en hiver)
Normes du MTMDET : 3,5 m
- + Pentes longues et abruptes (4,2 %)
- + Virages serrés
- + Enclavement (garde-corps de chaque côté)

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



Représentation schématisée qui n'est pas à l'échelle

AUTOMNE 2016

FIGURE 5 – Schéma de la configuration de la piste multifonctionnelle (PJCCI)².

3.2 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 (NGUYEN et al., 2019) et (CHONG et al., 1992). Beaucoup de références ont été publiées ces deux dernières années sur le sujet de la segmentation sémantique, ils existent donc de multiples alternatives inspirantes.

2. https://jacquescartierchamplain.ca/wp-content/uploads/2018/10/IMG_Fiche_piste-multi_pont_JC_FR_vfinale_web_2018-10-10.pdf

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 (DUSTIN, 2019), 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 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. (des piétons et cyclistes du pont JACQUES-CARTIER, 2020) (des piétons et cyclistes pont JACQUES-CARTIER, 2020)

Approche prévue pour le traitement des données

Il y a deux phases à cet essai : 1) l'inférence avec des modèles déjà prêts ; et 2) 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, celles avec lesquelles le modèle a été, ou sera, entraînées. La résolution et la qualité de l'image vidéo seront nivélées 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 qui ont été découvertes et qui seront potentiellement

utilisées dans le cadre de l'essai, incluant la référence à l'architecture du modèle d'apprentissage profond.

Tableau 1 – Tableau des données

	Spécification
1	<p>Réseau : SegNet Jeu de données : CamVid Vidéo : 10 minutes Résolution/s : HD</p> <p>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.</p> <p>https://github.com/PengKiKi/camvid</p>
2	<p>Réseau : MFANet Jeu de données : Cityscapes Nombre d'images : 5000 Résolutions : 1280x1024</p> <p>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.</p> <p>leejy@ustb.edu.cn</p>
3	<p>Réseau : RESNet18 Jeu de données : Cityscapes Nombre d'images : 25 000 Résolutions : 360x720, 512x256, 1024x512, 2048x1024</p> <p>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.</p> <p>https://github.com/tynguyen/MAVNet/tree/master/data/perch_drone</p>

Spécification	
4	<p>Réseau : RESNet18</p> <p>Jeu de données : DeepScenes</p> <p>Nombre d'images : 15 000</p> <p>Résolutions : 576x320, 864x480</p> <hr/> <p>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.</p> <p>http://deepscale.cs.uni-freiburg.de</p>
5	<p>Réseau : RESNet18</p> <p>Jeu de données : Synthia</p> <p>Nombre d'images : 220 000</p> <p>Résolutions : 1280x760</p> <hr/> <p>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. RESNet18 a été entraîné avec ce jeu. Il n'a pas été testé avec le Jetson Nano.</p> <p>http://3dvision.princeton.edu/datasets.html</p>
6	<p>Jeu de données : Association des piétons et cyclistes pont Jacques-Cartier</p> <p>Nombre d'images : 313</p> <p>Résolutions : variées</p> <hr/> <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</p> <p>http://pontjacquescartier365.com/videos-pont-jacques-cartier</p>

Spécification	
7	<p>Jeu de données : images et vidéo sur Internet Nombre d'images : entre 30-50 Résolutions : variées</p> <hr/> <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	<p>Jeux de données : KITI Road/Lane Detection</p> <hr/> <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>
9	<p>Jeux de données : Personnel</p> <hr/> <p>Nombre d'images et vidéos : 188 images de 1080x1920 ; 19 vidéos de 30-60 secondes de 1080x1920 et 60FPS.</p> <hr/> <p>Ce jeu de données contient des vidéos de différentes sections de pistes cyclables de mon quartier. Les vidéos ont été prises au mois de mars, dans des conditions ensoleillées, mais avec des endroits de la piste ombragée, sèche ou mouillée, bordée d'herbe ou de neige, parfois avec des passants. 188 images ont été extraites des vidéos.</p>

3.3 Matériel et logiciels

Le nano ordinateur

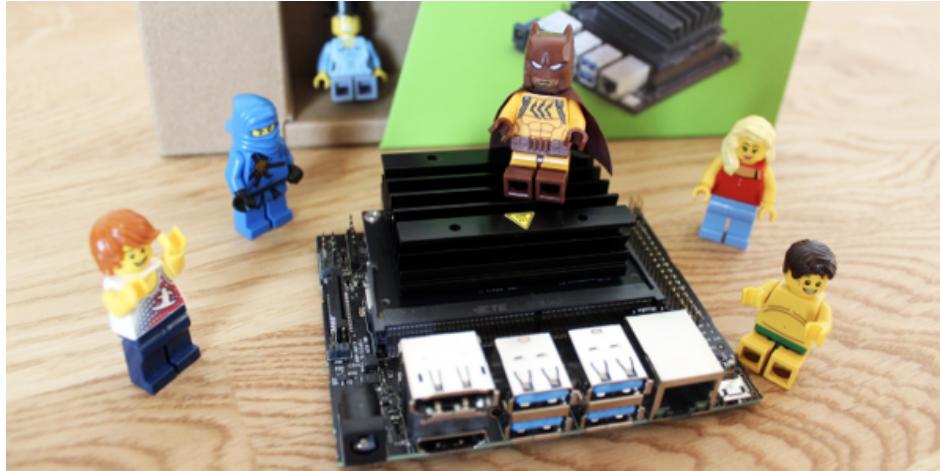


FIGURE 6 – Carte mère Jetson Nano de NVIDIA, représenté avec des Lego pour démontrer sa petite taille

L'objet d'étude de cet essai est le nano ordinateur "Jetson Nano" du fabricant "NVIDIA" (figure 6). Ce modèle a été choisi car il a été conçu par la compagnie NVIDIA spécifiquement pour répondre au besoin d'inférence en temps réel sur le terrain, afin d'éviter le transfert de données et le traitement à distance et différé.

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, où l'architecture la plus populaire et portable est x86-64. Il est composé d'un quad-core ARM Cortex-A57 @ 1.43 GHz, qui est conçu pour ce genre de nano ordinateur, comme le Raspberry Pi. Les performances GPU sont de 128-cores du Maxwell @ 921 MHz, 0.5 TFLOPS (16 FP = 16 bits FP = 2 bytes Floating Points). Par comparaison la PlayStation 4 Pro (2016) supporte +4 TFLOPS. La mémoire est limitée à 4Gb LPDDR4 @ 1.6 GHz. Les autres caractéristiques à considérer sont le port pour une carte microSD, un port Ethernet 10/100/1000Mbs, un port HDMI, un hub USB 4 ports 3.0, un connecteur pour une caméra, et un port PCIe.

Logiciels

De même que pour les périphériques, les solutions logiciels principaux qui seront utilisés dans le cadre de l'essai sont résumés dans le tableau suivant, où il est indiqué leur nom, le type de licence, leur version, leurs rôles et responsabilités, 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 microSD et du disque SDD interne M.2 NVMe, l'utilitaire "hdparm" a été utilisé. Il est nécessaire de l'installer car il n'est pas inclus de base avec le système L4T.

Le SDK qui sera utilisé avec le nano ordinateur sera celui fourni par NVIDIA et qui se nomme "JetPack"^{3 4}. La version 4.4⁵ sera celle avec laquelle les tests de performance ont été exécutés. Il contient le système d'exploitation "Linux for Tegra" (L4T)⁶ (version L4T 32.4.3), qui est une version de la distribution Linux Ubuntu 18.04 mise à la saveur de NVIDIA. Jetpack contient aussi d'autres librairies qui sont nécessaires pour l'inférence, tel que Cuda, CuDNN et TensorRT.

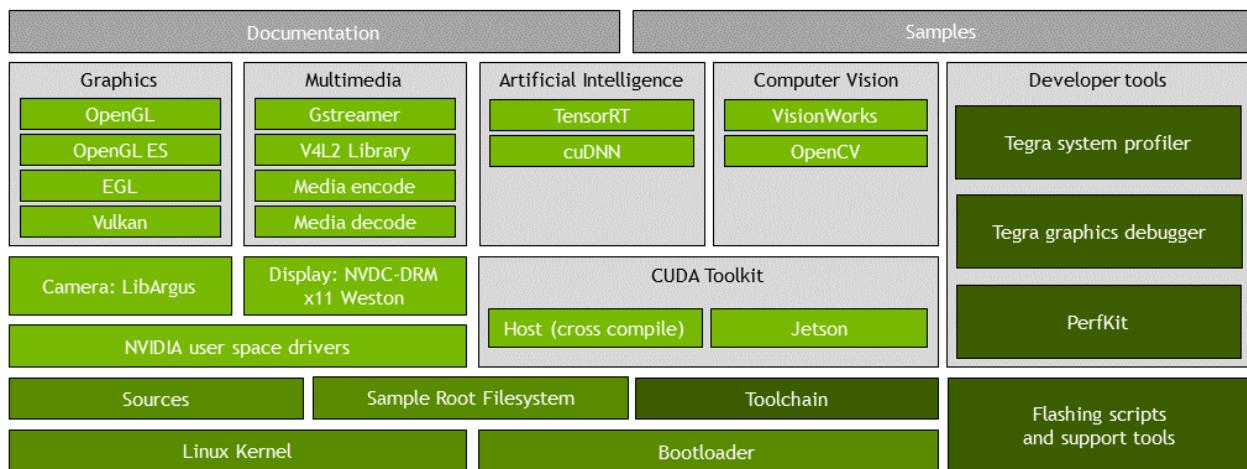


FIGURE 7 – Diagramme de l'architecture du NVIDIA JetPack⁷

Python et le C++ sont les langages utilisés par le framework de DeepLearning de NVIDIA. Python est utilisé comme langage accessible et appelle les extensions écrites en C++ et qui optimisent les accès aux ressources systèmes tel que les CPUs et GPUs, les traitements des images et vidéos, les boucles et les traitements mémoires intensifs.

La librairie d'apprentissage profond qui sera utilisée est PyTorch, bonifié avec une version adaptée par NVIDIA de torchvision, qui fournit des modèles d'architecture et des utilitaires pour la vision par ordinateur (computer vision). Des versions bien spécifiques sont nécessaires et il est important

3. <https://developer.nvidia.com/embedded/jetpack>

4. <https://docs.nvidia.com/jetson/jetpack/introduction/index.html>

5. <https://developer.nvidia.com/embedded/jetpack-archive>

6. <https://developer.nvidia.com/embedded/linux-tegra>

7. <https://docs.nvidia.com/jetson/l4t/index.html#page/Tegra%2520Linux%2520Driver%2520Package%2520Development%2520Guide%2520Overview.html%23>

de s'y conformer au risque de tomber dans une investigation bien couteuse en temps et énergie⁸.

Le nano ordinateur inclut un GPU qui est mis à contribution lors de l'inférence. Le compilateur de NVIDIA pour GPU 'cuda' est nécessaire pour re générer le .onnx lors de la phase d'adoption. La version doit concorder avec la bonne version de PyTorch. La version adaptée (fork) de torchvision doit être recompilée avec la bonne version de pytorch et cuda.

Enfin pour re générer le .onnx lors de la phase d'adaptation, les librairies TensorRT et ONNX ont été utilisées, en compagnie de l'utilitaire 'trtexec' qui permet de valider et tester le fichier .onnx généré.

Lors de la phase d'évaluation des performances systèmes, les utilitaires tegrastats, free, iotop ont été utilisés.

Tableau 2 – Solutions logicielles de l'essai

Language	Version	Licence	Rôles et responsabilités
JetPack	4.4	NVIDIA	Kit de développement de logiciels incluant le système d'exploitation L4T, et les librairies et utilitaires nécessaires pour l'inférence avec le nano ordinateur.
L4T	32.4.3	NVIDIA	Le système d'exploitation "Linux for Tegra" conçu par NVIDIA pour leurs solutions d'inférence légères, comme pour le nano ordinateur.
Python	2.7	GPL	Language plus accessible que le C++.
C++ (gcc)	7.3.1 ⁹	GPL	Certaines extensions du cadre applicatif de NVIDIA pour l'inférence sont écrites en C++, pour des raisons d'optimisation.
pytorch	1.1.0	BSD 3-Clause	Cadres d'application logicielle ("framework") pour l'apprentissage machine et profond.
torchvision	0.0.3	BSD 3-Clause	Branche de torchvision adaptée par NVIDIA ¹⁰ ; Doit être recompilée avec la version de pytorch 1.1.0 et cuda 10.0.
cuda	10.0	NVIDIA	Compilateur de code C++ pour GPU.

8. <https://forums.developer.nvidia.com/t/trying-to-regenerate-onnx-for-jetson-nano/125494?u=vincelf>

9. <https://developer.nvidia.com/embedded/linux-tegra>

10. <https://github.com/dusty-nv/vision.git> et ensuite branche v0.3.0

Language	Version	Licence	Rôles et responsabilités
TensorRT	6.0.1.5	NVIDIA	SDK pour générer des modèles au format ONNX, optimisés et interopérables, pour l'inférence.
ONNX	1.7.0	MIT	Librairie qui permet de générer un format interopérable pour l'inférence de modèles d'architecture construits avec différent framework de machine learning (Caffe, PyTorch, TensorFlow, etc).
trtexec	-	NVIDIA	Utilitaire qui a permis de tester le .onnx qui a été re généré.
gstreamer	1.14.5	LGPL	Utilitaire qui a permis d'alimenter le modèle de la segmentation avec la vidéo.
v4l2loopback	0.12.5	GPL	Utilitaire qui a permis de créer un matériel vidéo virtuelle permettant de remplacer la caméra, permettant ainsi au modèle d'être alimenté par une vidéo et non la caméra. Une fois installé, le matériel vidéo virtuel est accessible via "/dev/video1", "/dev/video0" étant réservé pour la caméra.
hdparm	9.54	GPL	Utilitaire permettant de tester la capacité de lecture d'une unité de stockage, tel que'un SSD NVMe et différentes cartes microSD.
tegrastats	-	NVIDIA	La commande offre différents indicateurs système tel que l'utilisation des processeurs, la température, la consommation, et qui sont utiles pour observer le comportement du système lors des tests de performance de la segmentation.
free	3.3.12	GPL	La commande offre le statut de la mémoire totale, utilisée, libre, swap, cachée, etc. Elle est utile pour observer le comportement de la mémoire du nano ordinateur lors des tests de performance de la segmentation.
iotop	0.6	GPL	La commande offre le statut des opérations "I/O" de lecture & écriture sur le disque, totale ou pour le processus de segmentation. Elle est utile pour observer le comportement des opérations sur le disque du nano ordinateur lors des tests de performance de la segmentation.

Language	Version	Licence	Rôles et responsabilités
segnet-console	-	NVIDIA	<p>La commande permet de segmenter une image. L'architecture est donnée en argument. Lors de l'essai, celle qui a été évaluée est "fcn-resnet18-deepscene-576x320". Les options qui doivent être utilisées pour que l'image générée soit évaluée doivent être "--visualize=mask --filter-mode=point --alpha=0". L'image originale est fournie en avant-dernière place de la commande, et l'image générée en dernière place. Il est possible aussi de démarrer l'inférence avec sa propre architecture grâce aux options "--model", "--prototxt", "--labels", "--colors", "--input_blob" et "--output_blob".</p>
segnet-camera	-	NVIDIA	<p>La commande permet de démarrer la segmentation avec la caméra, ou optionnellement avec un matériel vidéo virtuel (fournie par "v4l2loopback") grâce à l'option "--camera=/dev/video1", comme cela a été le cas durant le projet pour évaluer la segmentation avec des vidéos au lieu de la caméra. L'architecture est donnée en argument. Lors de l'essai, celle qui a été évaluée est "fcn-resnet18-deepscene-576x320". La résolution peut être précisée grâce aux options "width" et "height", mais durant l'évaluation les valeurs par défaut ont été conservées (width=1280px et height=720px). Il est possible aussi de démarrer l'inférence avec sa propre architecture grâce aux options "--model", "--prototxt", "--labels", "--colors", "--input_blob" et --output_blob". Il n'est pas possible de conserver la vidéo segmentée, il a été tenté de sauvegarder les images qui sont rafraîchies à l'écran dans la fenêtre XWindow mais le code est trop intrusif et impacte négativement les performances de la segmentation et du nano ordinateur. L'évaluation des performances n'est donc que visuelle, et cette limitation pourrait remettre en question le désir de détecter les délimitations de la piste cyclable en temps réel, puisqu'il n'y a pas de moyen de récupérer le résultat généré.</p>

3.4 Méthodologie

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

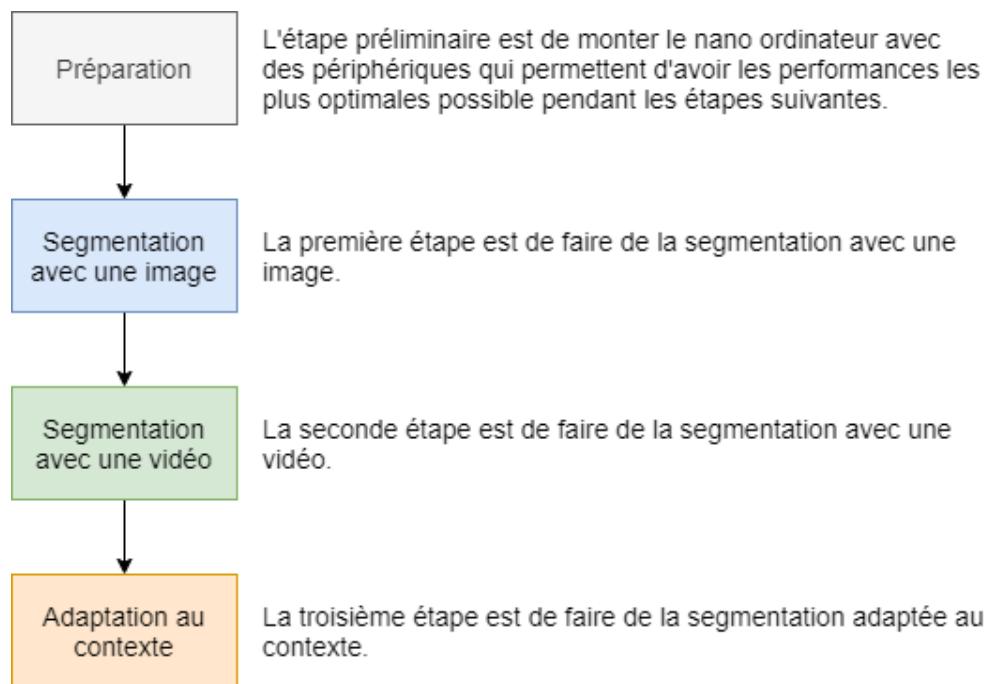


FIGURE 8 – Organigramme de la méthodologie à haut niveau

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.

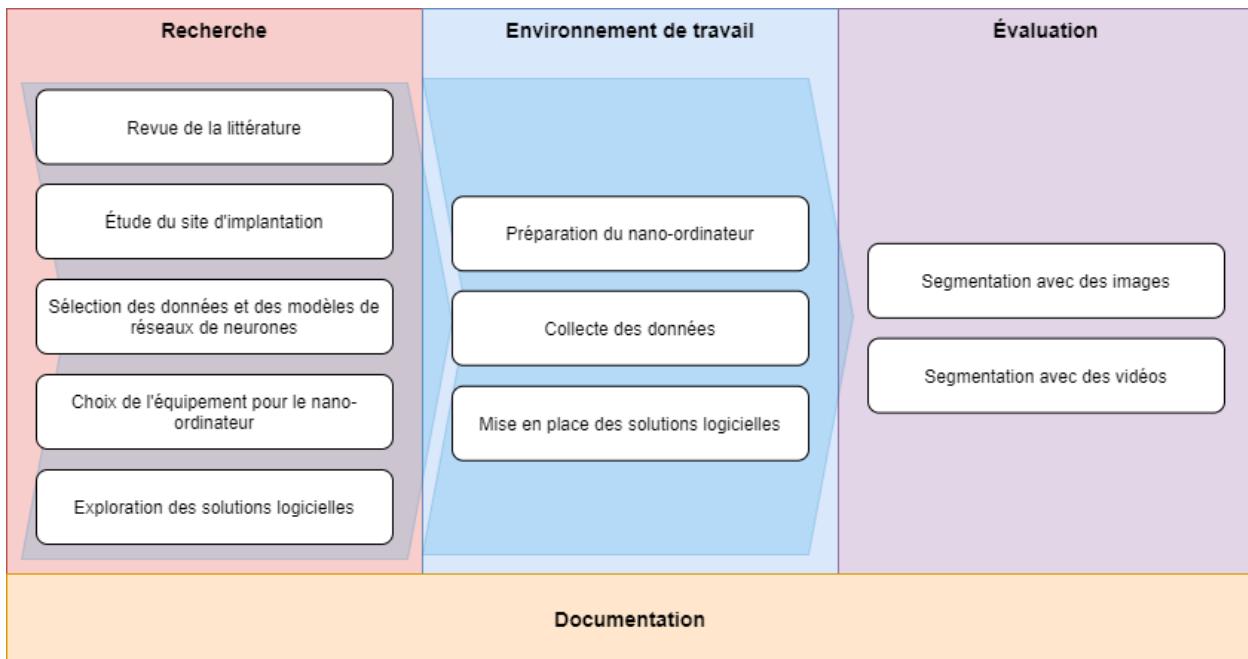


FIGURE 9 – Organigramme des détails de la méthodologie pour évaluer les performances

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

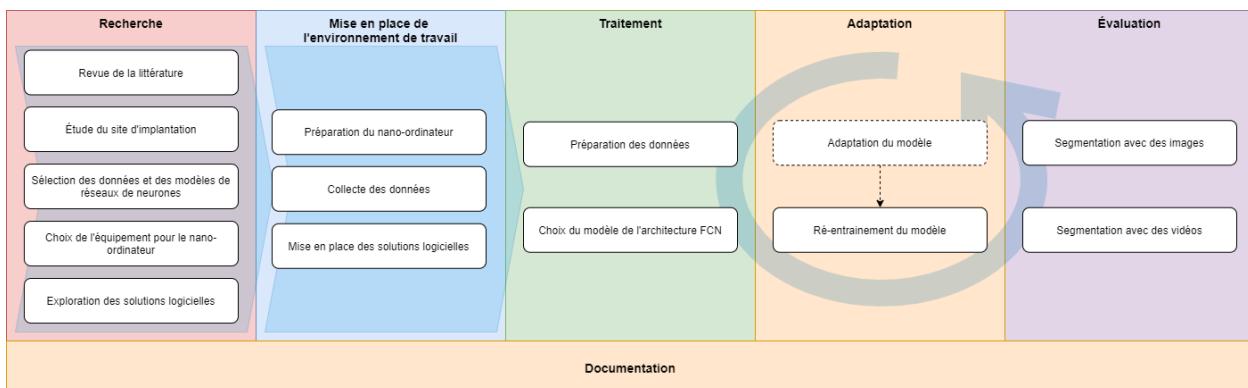


FIGURE 10 – Organigramme des détails de la méthodologie pour évaluer les performances après une phase d'adaptation

Les phases de la méthodologie présentée dans l'organigramme de la figure 10 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 nano ordinateur le système d'exploitation, l'environnement de développement et de tests pour l'inférence.

- Itération entre les étapes suivantes :
- Inférence avec le nano ordinateur 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.

Finalement, il est à noter que le cheminement de l'essai a été entièrement documenté pendant tout le déroulement de l'essai.

3.5 Documentation

La méthodologie a été entièrement documentée pendant tout le déroulement de l'essai. Elle se retrouve pour référence dans un blogue public sur le site de "github.io"¹¹. Cette méthodologie de documentation permet entre autres, de très facilement documenter, de ne pas perdre des notes très importantes, de suivre le cheminement, de pouvoir retrouver des notes, même si elles ont été effacées ou modifiées, puisque toute modification est sauvegardée dans un repository Git.

Par ailleurs, tous les documents de rédaction LaTeX, les images, les scripts et code source qui ont été utiles et utilisés durant l'essai ont été géré dans un repository Git public avec "github.com"¹².

Ces sources d'information viennent bonifier grandement ce rapport et il est même recommandé de s'y référer pour atteindre un certain niveau détails et de compréhension.

3.6 Recherche

3.6.1 Revue de littérature¹³

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.

11. <https://vince71f.github.io/>

12. <https://github.com/vince71f/gae724>

13. 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).

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 : 1) les références se rapprochant le plus du sujet de l'essai ; 2) l'histoire et les antécédents des réseaux de neurones ; 3) 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, où 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.

3.6.2 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.

3.6.3 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 (NGUYEN et al., 2019) et (CHONG et al., 1992). 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 (DUSTIN, 2019), 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. (des piétons et cyclistes du pont JACQUES-CARTIER, 2020) (des piétons et cyclistes pont JACQUES-CARTIER, 2020)

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).

3.6.4 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 nano ordinateur, 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".

3.6.5 Exploration des solutions logicielles

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

3.7 Environnement de travail

3.7.1 Préparation du nano ordinateur

L'organigramme de la figure 11 présente les activités qui composent la préparation du nano ordinateur.

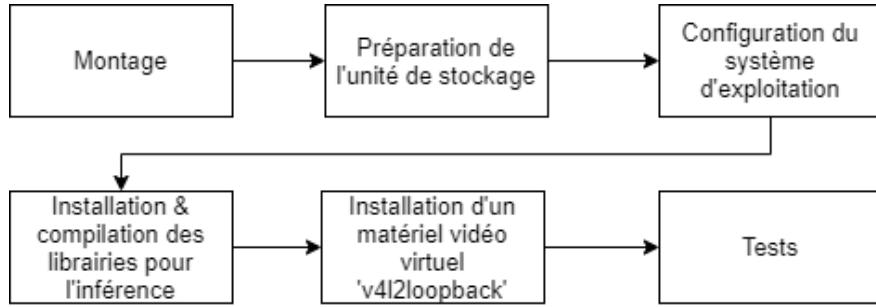


FIGURE 11 – Préparation du nano ordinateur

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 12 présente les activités qui composent le montage du nano ordinateur.

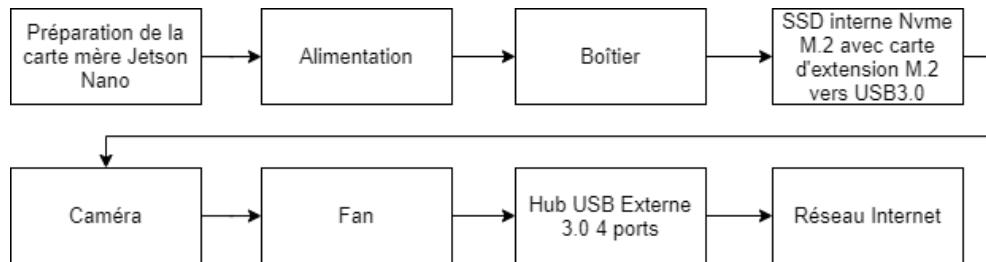


FIGURE 12 – Montage du nano ordinateur

La figure 13 présente la carte mère, les accessoires et périphériques du nano ordinateur qui a été monté.

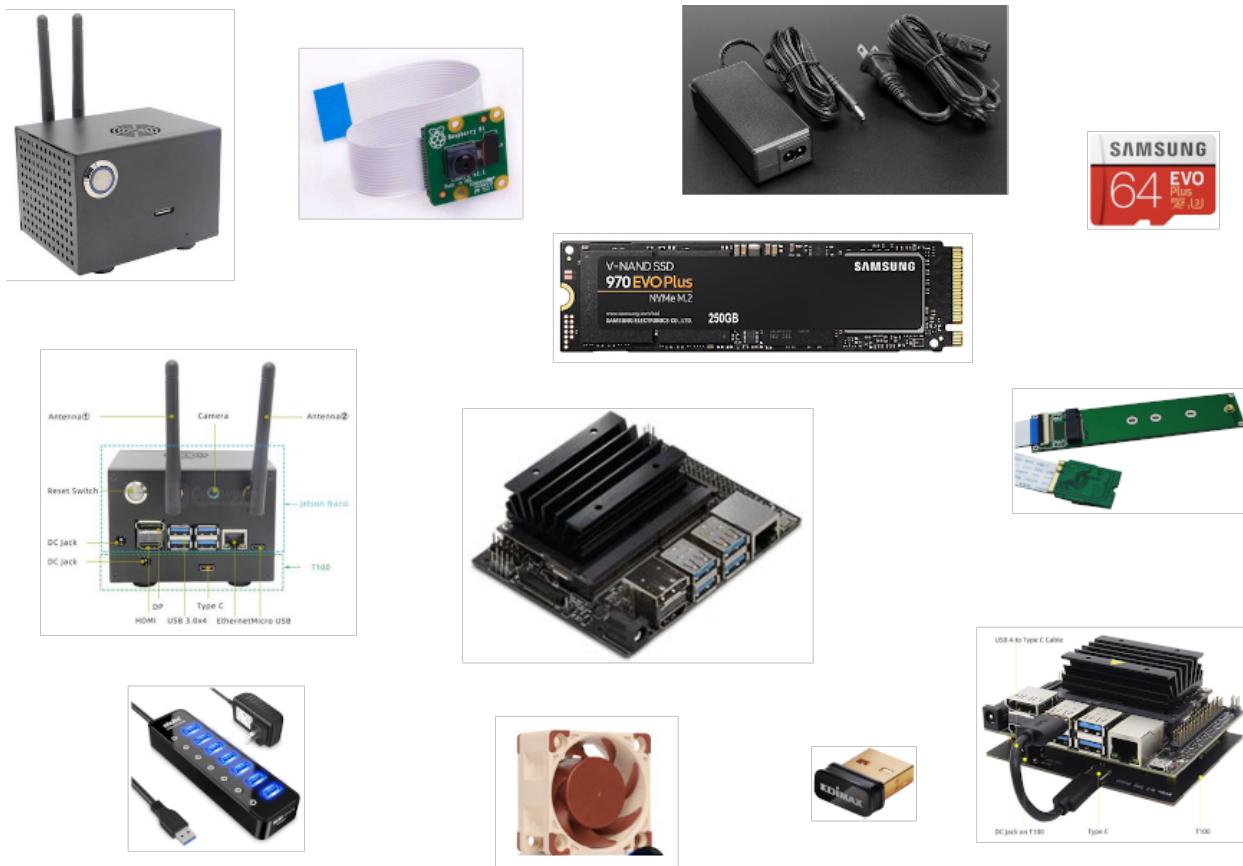


FIGURE 13 – Carte mère, accessoires et périphériques du nano ordinateur

Préparation de la carte mère Jetson Nano

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/2 amp 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 5 watts due 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 5 volts 4 amp 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

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 5 volt 2 amp. Mais le besoin en énergie augmente avec les périphériques qui s'accumulent, tel qu'une caméra. Il est prudent de choisir un adaptateur 5 volt 4 amp 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

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 boitier en métal. Le boitier 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 boitier 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.

Unité de stockage

Le nano ordinateur est conçu pour fonctionner avec un système d'exploitation hébergé sur une carte microSD. Il existe différentes cartes microSD, et certaines sont beaucoup plus performantes que les autres. Malheureusement les cartes microSD 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 continu à l'extérieur, l'utilisation un disque SSD interne comme alternative semble logique.

Carte microSD

Il existe différentes cartes microSD, 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 microSD seront évaluées :

Tableau 3 – Cartes microSD



microSD Samsung EVO 64Gb Plus XC I Grade 3 Class 10

microSD Samsung EVO 64Gb Select XC I Grade 3 Class 10

microSD Scan Disk Ultra 32Gb HC I Class 10

Disque SSD

Pour un appareil destiné à être continuellement en service et à l’extérieur, 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 microSD est tout de même nécessaire pour "bootstrapper" le système d’exploitation. Il n’est pas nécessaire d’avoir une carte microSD performante puisqu’elle n’est utilisée que pour démarrer le système qui se trouve sur le SSD interne.

Un disque SSD et une carte microSD sont différents type de matériel pour différents usages. Le disque SSD est 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, et qui est un appareil destiné à être continuellement en service et à l’extérieur. L’unité de stockage doit être non seulement performante, mais aussi endurante. 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 microSD, 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.

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.

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 destiné 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 ne supporte pas les SSD M.2 NVMe connecté au port USB¹⁴. 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.

Différentes options pour optimiser l'alimentation ont été explorées : a) utiliser un HUB USB externe et auto alimenté ; b) brancher un câble Ethernet au lieu d'utiliser un Dongle Wifi ; c) allumer le ventilateur dès le démarrage du nano ordinateur ; d) et l'option de fournir 6 amp directement supportée par la carte mère via les pins ; e) explorer les solutions sur les forums de discussion^{15 16}.

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

14. À noter que la carte d'extension T100 est discontinuée et remplacée par la T130

15. <https://www.kingston.com/en/community/articledetail/articleid/48543>

16. <https://geekworm.com/products/nvidia-jetson-nano-nvme-m-2-ssd-shield-t100-v1-1>

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.

la caméra qui a été sélectionnée est la version 2 de celle du fournisseur Raspberry Pi, le concurrent direct du nano ordinateur NVIDIA Jetson Nano. Cette caméra a été éprouvée avec le temps, est performante, elle semble être la plus adaptée pour ce genre de projet.

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 boitier, il est encore plus indispensable d'installer un ventilateur. Le ventilateur choisi a pu être installé dans le boitier, même si le boitier 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'extérieure, évitant un risque de surchauffe, la capacité du ventilateur étant tout de même limité (petit modèle).

Hub USB externe 3.0 4 ports

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 gourmant 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 11Mbps.

Le PoE n'a pas été évalué.

Préparation de l'unité de stockage

Le nano ordinateur est conçu pour fonctionner avec une microSD, et NVIDIA fournit uniquement de la documentation à cet effet. L'option d'utiliser un disque SSD interne est disponible sur Internet, mais n'est pas supporté officiellement par NVIDIA. Il existe néanmoins des articles à ce sujet dans le forum des développeurs¹⁷.

Carte microSD

NVIDIA fournit de la documentation très claire et simple afin de préparer la carte microSD (formatage) et installer l'image du JetPack.¹⁸.

Disque SSD

La procédure d'installation pour installer JetPack sur le SSD interne est disponible sur le site "jetsonhacks.com"¹⁹. À noter qu'une carte microSD est tout de même nécessaire pour "bootstrapper"

17. <https://forums.developer.nvidia.com/t/how-to-connect-ssd-to-jetson-nano/74053>

18. <https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#write>

19. <https://www.jetsonhacks.com/2019/09/17/jetson-nano-run-from-usb-drive/>

le système d'exploitation. Il n'est pas nécessaire d'avoir une carte microSD 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, fuseau horaire, 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.

3.7.2 Collecte des données

Le jeu d'images de DeepScene est celui qui semble le plus approprié car il a été conçu pour détecter les chemins dans la forêt. De plus il existe une version du modèle qui a été entraîné avec ce jeu. Comme il possède un jeu d'images vérité terrain (GT), il sera bien utile pour évaluer la segmentation prédite et un gain de temps non négligeable dans le cadre d'un essai. Le jeu d'images de CityScape est très complet pour les scènes urbaines, mais comme il est moins spécialisé dans la détection de chemins ou de piste, son utilisation ne sera pas priorisée. Il contient toutefois des

images vérité terrain de routes, ce qui est avantageux dans notre contexte et le favorise par rapport au deux derniers que nous avons à notre disposition. En effet le jeu d'images et de vidéos de l'APC-PJC, et celui que j'ai monté en prenant des vidéos de pistes cyclables de mon quartier, sont des jeux intéressants pour tester les résultats de la segmentation avec des images ou des vidéos qui viennent du site d'études, ou similaire, que celui de chemins forestiers. De plus ces images et vidéos sont loin des conditions parfaites (luminosité, qualité du sol, angle de vue, etc).

3.7.3 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), le cadre applicatif ('framework') et les librairies nécessaires pour l'inférence, tel que Python, pytorch, les modèles pré-entraînés au format ONNX, le compilateur CUDA, et le SDK TensorRT.

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 met 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 véritable terrain (GT).

NVIDIA

Compte NVIDIA

NVIDIA met 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 re générer 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 impacte

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 re générer un ONNX, par exemple. Malheureusement la capacité du nano ordinateur ne permet pas de travailler efficacement avec un container docker, 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.

3.8 Évaluation

L'évaluation des performances se décomposent des différents éléments suivants, et qui sont présentés dans le diagramme de la figure 14 :

- Tout d'abord, les indicateurs de performances.
- Les performances matérielles durant l'inférence seront évaluées grâce aux indicateurs fournies par les utilitaires 'Tegrastats' de NVIDIA, 'free' et 'iostop'. Ces utilitaires sont brièvement décrit dans le tableau 2.

- Les performances de la segmentation seront évaluées avec les indicateurs classiques²⁰ : le "IoU" ("Intersection over Union", ou "Jaccard index") ; et le F1 score (ou "Dice coefficient").
- Ensuite, différentes résolutions d'images et de vidéos seront utilisées pour déterminer lesquelles sont supportées par le modèle évalué, tel qu'indiqué dans la section "3.8.3 Résolutions évaluées".
- Les images et vidéos qui sont à notre disposition pour être évaluée proviennent de différentes sources de données, tel que décrit dans la section "3.7.2 Collecte des données".
- Enfin plusieurs modèles ont été sélectionnés comme candidats intéressants pour l'évaluation, et décrit dans la section "3.9.1 Choix du modèle de l'architecture FCN".

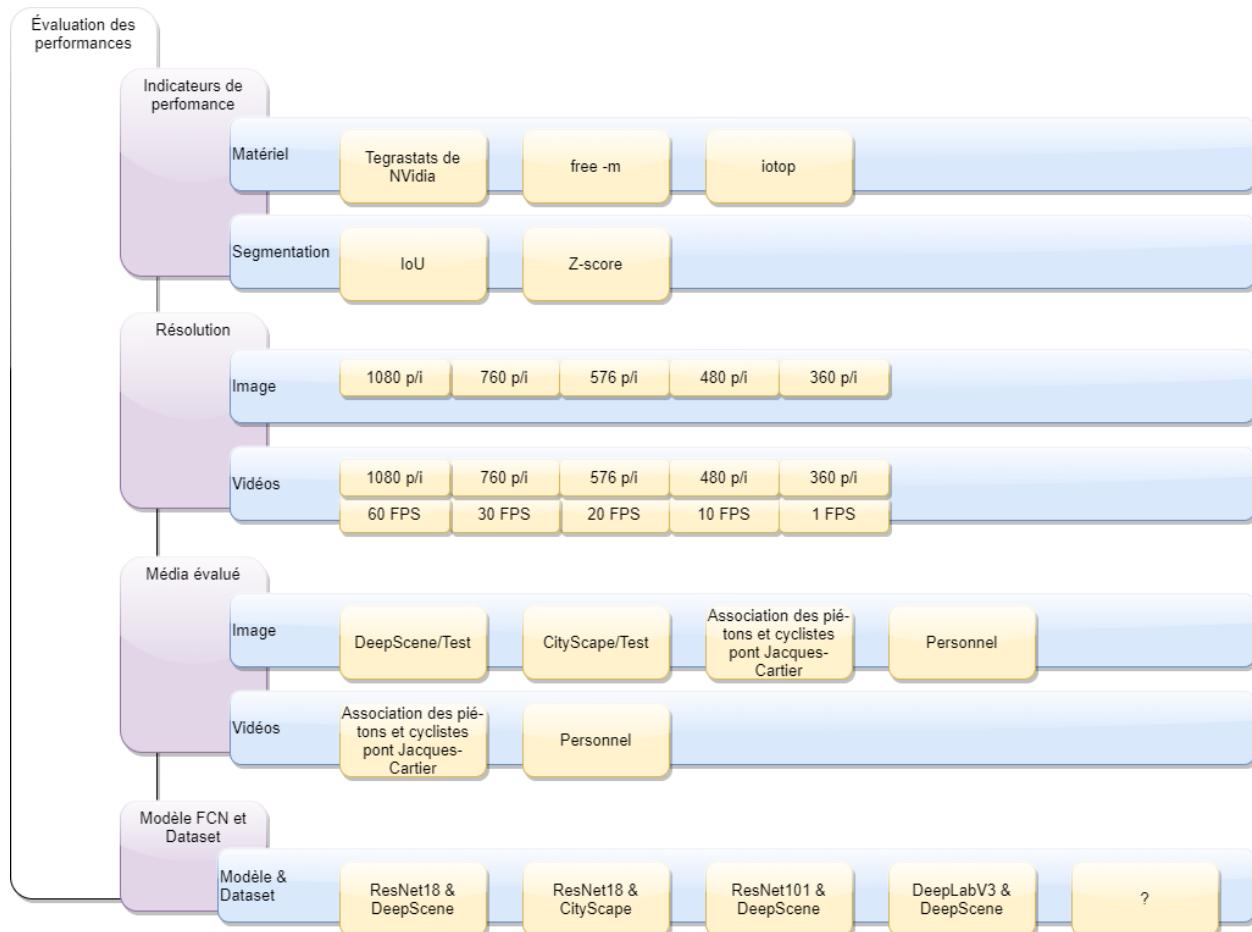


FIGURE 14 – Éléments pour l'évaluation des performances

20. <https://ilmonteux.github.io/2019/05/10/segmentation-metrics.html>

3.8.1 Stratégie de test de l'inférence

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 entraînés 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.

3.8.2 Stratégie de collecte des indicateurs de performance matériel

La méthodologie de la collecte des indicateurs est la suivante²¹ :

- La collecte est démarrée après un démarrage frais, manuellement, via un script shell, qui exécute chaque utilitaire, et attend l'interruption du test.
- Chaque utilitaire qui est utilisé pour collecter les mesures, possède son propre fichier.
- La date et l'heure de chaque indicateur collecté sont précisées.
- Afin de faciliter la documentation et l'analyse du test, des points d'intérêt sont ajoutés dans un fichier séparé pour marquer un moment particulier du test, avec la date, l'heure et un libellé. Ce point d'intérêt est fait grâce à une commande "shell" qui vient ajouter une trace dans ce fichier.
- Chaque indicateur est collecté toutes les secondes.

21. <https://vince71f.github.io/2020/05/26/metrics.html>

- Une fois le test complété, la collecte est arrêté manuellement.
- Chaque fichier est ensuite transformé en fichier CSV, via des commandes shell.
- À partir des fichiers CSV un script Python génère les graphiques automatiquement.

Chaque indicateur est une colonne du fichier CSV. Il existe le même nombre d'indicateurs à tout moment. La date et l'heure sont un champ.

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

Ensuite les tests débutent.

Les indicateurs collectés permettent de créer des graphiques qui montrent la progression de chacun.

Les performances matérielles du Jetson Nano sont évaluées grâce à différents utilitaires : "tegrastats" fournis par NVIDIA ; "free" ; et "iostop".

Les performances de la segmentation sont évaluées grâce au IoU et au F1 score pour la classe du chemin / route. Une fonction Python est utilisée. Les fonctions IoU et le F1 score utilisent l'image prédite (généré par le modèle FCN) et l'image vérité terrain (GT). Les images originales sont donc présélectionnées selon leur intérêt et l'image vérité terrain (GT) créée. L'image prédite et vérité terrain (GT) doivent utiliser la même palette de couleurs et doivent être de la même résolution. Pour les images qui ne possèdent pas d'image vérité terrain (GT), celle-ci est créée à la main avec l'éditeur "Gimp". Comme la résolution de la segmentation de l'image prédite par le modèle de NVIDIA est très faible ("carrée"), l'image vérité terrain (GT) ne sera pas précise au pixel près. Le besoin est d'évaluer et non d'entrainer, l'importance de la précision de la classification est moindre dans ce cas.

3.8.3 Résolutions évaluées

Voici dans le tableau 4 les résolutions et images par seconde qui seront évaluées.

3.8.4 Segmentation avec des images

Préparation et post-traitement

Tableau 4 – Résolutions et images par seconde (FPS) qui seront évaluées

Résolutions
320x576, 480x640, 720x1280, 768x1024, 768x1152, 800x1152, 832x1024, 864x1024, 832x1120, 832x1152, 768x1280, 800x1280, 864x1152, 900x1152, 900x1280, 960x1600, 1080x1920, 1024x1024
Images par seconde (FPS)
60/1, 30/1, 15/1, 1/1

Afin de pouvoir mesurer les performances de la segmentation (IoU, F1 score), les classes et la palette de couleur entre l'image vérité terrain (GT) et celles prédites doivent être les mêmes.

L'image vérité terrain (GT) du jeu de donnée original DeepScene ne possède pas la même palette de couleur ni exactement les mêmes classes que celle du modèle.

Un travail d'uniformisation est nécessaire avant la segmentation, qui est résumée dans le tableau 5.

Tableau 5 – Classes et palettes de couleur

DeepScene		NVIDIA		Consolidée	
Classes	RGB	Classes	RGB	Classes	RGB
Road	170-170-170	Trail	200-155-75	Trail	170-170-170
Grass	0-255-0	Grass	85-210-100	Grass	0-255-0
Vegetation	102-102-51	Vegetation	15-100-20	Vegetation	102-102-51
Tree	0-60-0	-	-	Vegetation	102-102-51
Sky	0-120-255	Sky	0-120-255	Sky	0-120-255
Obstacle	0-0-0	Obstacle	255-185-0	Obstacle	0-0-0

De plus, l'image segmentée prédite par le modèle ne possède pas précisément la même palette de couleur que celle qui est configurée, il y a quelques différences minimes dans les codes couleurs

RGB (par exemple 0-119-255 au lieu de 0-120-255), mais qui doivent être arrangées afin de pouvoir être correctement évaluées.

Un travail de traitement de l'image segmentée prédictive est nécessaire avant l'évaluation de la segmentation.

Segmentation et évaluation

Afin de tester la performance de la segmentation du modèle, deux images du jeu de données de DeepScene seront utilisées, car ce jeu contient déjà les images vérités terrain (GT), un gain de temps non négligeable dans le cadre de l'essai. Uniquement la classe "Trail" sera évaluée.

L'architecture fournit à l'utilitaire "segnet-console" est "fcn-resnet18-deepscene-576x320"²².

Un script Python²³ est utilisé afin de mesurer le IoU et le F1 score de la classe de l'image prédictive par le modèle.

3.8.5 Segmentation avec des vidéos

Préparation et pré-traitement

L'évaluation de la segmentation avec des vidéos va s'effectuer non pas avec la caméra, mais avec un matériel vidéo virtuel. En effet, il n'est pas réaliste de pouvoir travailler sur le terrain. La commande "segnet-camera" permet de fournir en option le matériel qui doit être utilisé, par exemple "/dev/video0" pour la caméra. Le module "v4l2loopback"²⁴ permet de créer un matériel vidéo virtuel "/dev/video1". Ce matériel permet de recevoir un flux vidéo, qui pourra alors alimenter l'utilitaire "segnet-camera", comme le ferait la caméra. Le flux vidéo sera produit par l'utilitaire "gstreamer" avec comme données d'entrées le fichier de la vidéo et dirigé vers le matériel vidéo virtuel "/dev/video1".

La difficulté réside dans le fait que le matériel vidéo virtuel et le flux vidéo doivent être compatibles avec ce que l'utilitaire "segnet-camera" s'attend, et qui a été conçu pour être compatible avec une caméra.

22. segnet-console-{}-network=fcn-resnet18-deepscene-{}-visualize=mask-{}-alpha=10000images/city_0.jpgoutput.jp

23. <https://gist.github.com/ilmonteux/8340df952722f3a1030a7d937e701b5a>

24. <https://github.com/umlaeute/v4l2loopback>

Le diagramme de la figure 15 résume à haut niveau les relations entre ces éléments. Pour comparaison, le diagramme de la figure 16 montre la segmentation avec la caméra.

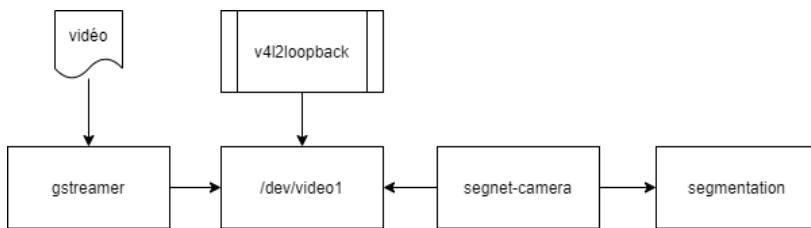


FIGURE 15 – Diagramme d'architecture de la segmentation d'une vidéo

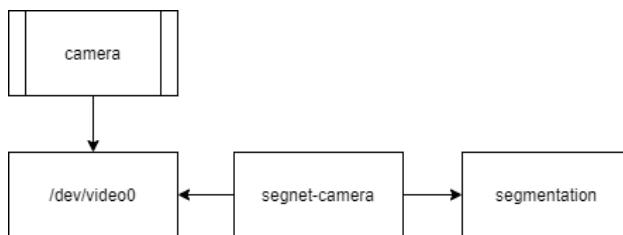


FIGURE 16 – Diagramme d'architecture de la segmentation avec la caméra

Segmentation et évaluation

Les tests de performance de la segmentation de vidéos se déroulent de la manière précisée dans la section "3.8.1 Stratégie de test de l'inférence".

L'un des avantages de l'utilitaire "gstreamer" est de pouvoir contrôler la résolution et le nombre d'images par seconde (FPS) de la vidéo qui doit être segmentée. Les différentes résolutions et FPS qui désirent être exécutées sont préparées dans un script "shell" écrit pour l'occasion. Le script s'occupe de démarrer gstreamer avec les bons paramètres, et en parallèle de démarrer la segmentation avec "segnet-camera". Un jeu de résolution peut être testé unitairement²⁵, ou plusieurs en séquence²⁶.

Les résolutions et images par seconde qui ont été testées sont résumées dans le tableau 7.

Deux vidéos ont été utilisées pour tester la segmentation. La première vidéo est utilisée pour tester l'inférence avec une vidéo du site d'étude, et qui a été fournie gracieusement par l' APC-PJC. Cette

25. https://github.com/vince71f/gae724/blob/master/run_deepscene.sh

26. https://github.com/vince71f/gae724/blob/master/run_deepscene_batch.sh

première vidéo est intéressante, car elle est filmée en mouvement par un cycliste. Dans un interval de 30 secondes, l'angle de vue change rapidement. La piste cyclable est bordée d'un muret côté sud, et de la route avec les voitures qui circulent côté nord. Même si la journée est ensoleillée, la surface de la piste est aussi à un moment humide.

La seconde vidéo est utilisée pour tester la segmentation avec les différentes résolutions et images par seconde. C'est une vidéo d'une petite piste cyclable qui est dans mon quartier, et que j'ai prise en marchant avec mon téléphone intelligent. La vidéo est intéressante, car dans un interval de 30 secondes l'état de la piste passe d'une scène ensoleillée à ombragée, sèche à mi-sèche, avec un petit ou gros banc de neige en bordure, ou qui s'aventure un peu sur la piste, bordée d'herbe mouillée ou sèche.

3.9 Adaptation

La phase d'adaptation sera initiée si le temps le permet, et s'il est jugé bon d'améliorer la qualité de la segmentation. Il y a deux types d'adaptation possible : matérielle et logicielle.

L'adaptation matérielle sera jugée nécessaire si le nano ordinateur ne peut être utilisé tel quel pour répondre aux objectifs. Par exemple si le nano ordinateur devient non utilisable (lent ou sans réponse) après un certain temps. La stratégie sera de diagnostiquer le comportement et d'évaluer un remède. L'une des pistes de solution privilégiée sera de lui apporter plus de puissance, comme de l'ajout de mémoires ou un espace de stockage plus performant. Plus complexe, optimiser le processus, par exemple certains traitements, serait aussi une option.

L'adaptation logicielle, quant à elle, sera jugée nécessaire si la prédiction de la segmentation est en deçà des attentes, ou inutilisable. Le choix d'adapter un modèle avec les méthodes de "Transfer Learning" et "Domain adaptation" sera la première option privilégiée, car cela procure un gain de temps non négligeable : il n'y a pas besoin de passer à travers tout le processus "essai-erreur", couteux en temps, en énergie et en ressources matérielles, d'apprentissage et de paramétrisation du modèle. Des efforts conséquents seront par contre nécessaires pour générer les images vérités terrain GT avec le jeu de données local, celui de l'APC-PJC de préférence, personnel ou autre sinon.

Le diagramme de la figure 17 présente la méthodologie qu'il faut suivre pour adapter un modèle à un nouveau jeu de données et à une autre résolution. La première étape est de sélectionner un modèle déjà entraîné et qui semble pouvoir être le meilleur candidat pour aider à répondre à la pro-

blématique. C'est un travail de recherche et de test minutieux, qui est le plus important de toutes les étapes. La seconde activité est conséquente en efforts : préparer le jeu de données, incluant les images vérité terrain, les bonnes résolutions ; et déterminer les classes et la palette de couleurs nécessaires. L'étape suivante est d'étudier l'architecture du modèle, afin de l'adapter au jeu de données, aux classes, et au besoin modifier les couches de l'architecture afin d'avoir une segmentation la plus précise et fine possible. Une fois ces étapes de préparation complétées, le ré entraînement du modèle peut s'effectuer, et la segmentation évaluée. Cette phase d'adaptation est un processus itératif, qui peut être représenté par la figure ??.

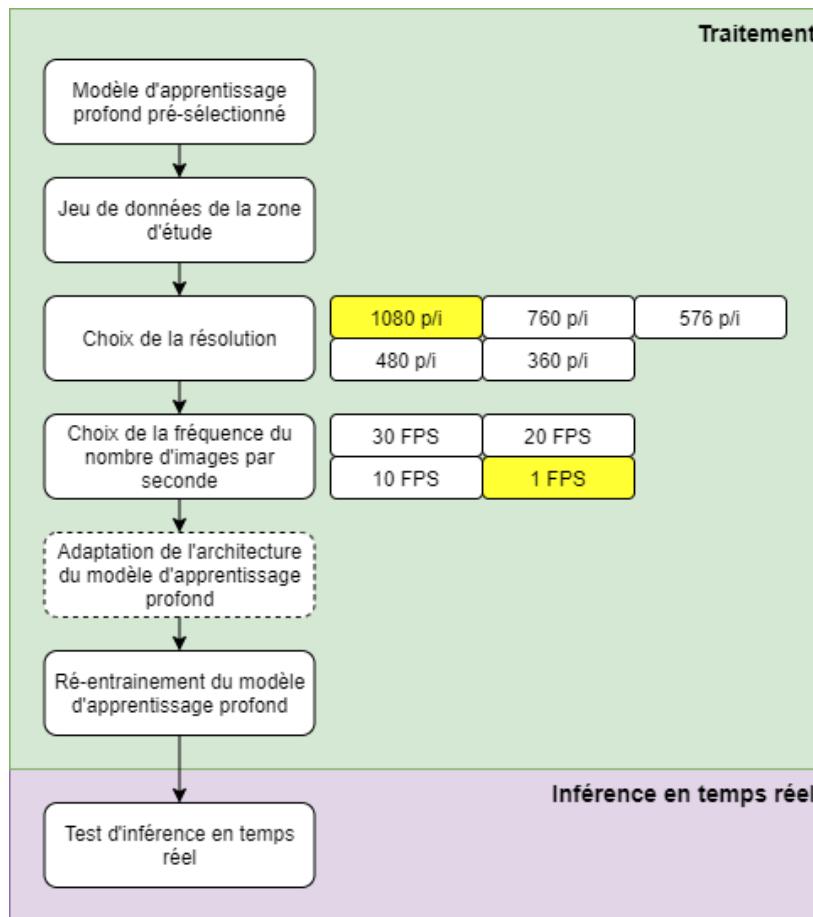


FIGURE 17 – Méthodologie du traitement et adaptation

3.9.1 Choix du modèle de l'architecture FCN

Le premier modèle qui sera évalué sera celui de SegNet18 entraîné avec le jeu de données "DeepScene", et fourni par NVIDIA. Le second de la liste, et qui est aussi déjà fourni par NVIDIA, est celui de SegNet18 entraîné avec le jeu de donnée "CityScape". Les deux autres architectures, ResNet101 & DeepScene et DeepLabV3 & DeepScene, ne sont pas disponibles et devront être

préparées et entraînées, mais elles sont attrayantes du point de vue de leur réputation et potentiel, et vouloir les adapter au contexte de l'essai semble logique. Une dernière boîte vide est disponible, afin de laisser une porte ouverte à une potentielle opportunité d'entraîner un modèle tout à fait personnalisé, par exemple une adaptation de l'architecture de DeepLabV3 avec le jeu de données de l'APC-PJC.

4 Résultats

4.1 Performances matérielles

4.1.1 Stockage de données

Pour tester les performances de la microSD et du disque SSD interne M.2 NVMe, l'utilitaire "hdparm" peut être facilement utilisé.

Tableau 6 – Comparaison des performances du "data read" entre un SSD M.2 NVMe et une microSD

Disk reads	MB	sec	MB/sec
Samsung 970 EVO Plus 250GB M.2 NVMe Internal Solid State Drive	1004	3	334.15
microSD Scan Disk Ultra 32Gb HC I Class 10	122	3.03	40.22
microSD Samsung EVO 64Gb Plus XC I Grade 3 Class 10	256	3.02	84.71
microSD Samsung EVO 64Gb Select XC I Grade 3 Class 10	92	3.01	30.54

4.1.2 Performances système

Les diagrammes suivants présentent l'état du nano ordinateur avant la segmentation, pendant et après. Les indicateurs qui sont observés sont ceux de la mémoire, la fréquence, le I/O, la consommation, la température. Afin de montrer l'impact potentiel de l'application Chromium, elle est démarrée entre deux segmentations, et pendant la segmentation.

La carte microSD "Scan Disk Ultra 32Gb class 10 HC I" a été utilisée pour les tests de performance système. La carte microSD "Samsung EVO 64Gb Plus class 10 HC I" n'était malheureusement plus fonctionnelle au moment des tests, celle-ci ayant été réservée pour tenter d'adapter le modèle aux images terrain locales.

Le test infère en temps réel la vidéo qui est capturée avec la caméra du nano ordinateur. Le réseau FCN qui est utilisé est celui fournit par NVIDIA "fcn-resnet18-deepscene-576x320". Ce modèle détecte automatiquement la résolution la plus appropriée avec cette caméra, c'est-à-dire 30 images par seconde (FPS) et une résolution de 1280x720. Le test dure 1400 secondes, un peu de plus de 23 minutes. Il peut se diviser en onze périodes d'observation, qui sont brièvement décrites ci-dessous :

1. La première période est celle entre la 1re seconde et la 200e seconde, et qui permet d'observer l'état du système au démarrage du nano ordinateur sans opération mise à part celle de la collecte des statistiques.
2. La seconde période est entre la 200e seconde et la 400e, et qui correspond à la première segmentation avec la caméra. Elle permet d'observer le système lors du premier démarrage de la segmentation.
3. La troisième période est celle entre la 400e seconde et le premier démarrage de Chromium. Elle permet d'observer la réaction du système après l'arrêt de la segmentation.
4. La quatrième période est celle entre le premier démarrage de Chromium et son arrêt. Elle permet d'observer le comportement du système lors de l'utilisation de Chromium, qui est suspecté de ralentir le système, lorsqu'actif (observations faites durant l'essai).
5. La cinquième période est celle entre l'arrêt de Chromium et le démarrage de la seconde segmentation avec la caméra. Cette période permet d'observer la réaction du système après l'arrêt de Chromium.
6. La sixième période est celle entre le démarrage de la seconde segmentation avec la caméra et son arrêt. Cette période permet d'observer la réaction du système pendant la seconde segmentation.
7. La septième période est celle entre l'arrêt de la seconde segmentation et le démarrage de la troisième segmentation avec la caméra. Elle permet d'observer la réaction du système après l'arrêt de la segmentation la seconde fois.
8. La huitième période est celle entre le démarrage de la troisième segmentation et le démarrage de Chromium la seconde fois. Cette période permet d'observer la réaction du système pendant le démarrage de la segmentation la troisième fois.
9. La neuvième période est celle entre le deuxième démarrage de Chromium et son arrêt. Elle permet d'observer le comportement du système lors de l'utilisation de Chromium pendant l'inférence.
10. La dixième période est celle entre l'arrêt Chromium la seconde fois et l'arrêt de la troisième segmentation. Cette période permet d'observer la réaction du système après l'arrêt de Chromium pendant l'inférence.

11. La onzième période est celle entre l'arrêt de la troisième segmentation et l'arrêt du test et de la collecte des statistiques. Elle permet d'observer la réaction du système après l'arrêt de la segmentation la troisième fois.

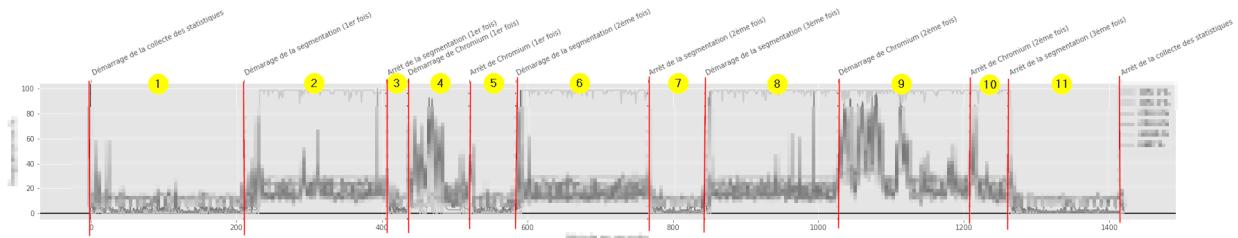


FIGURE 18 – Les périodes du diagramme des performances système

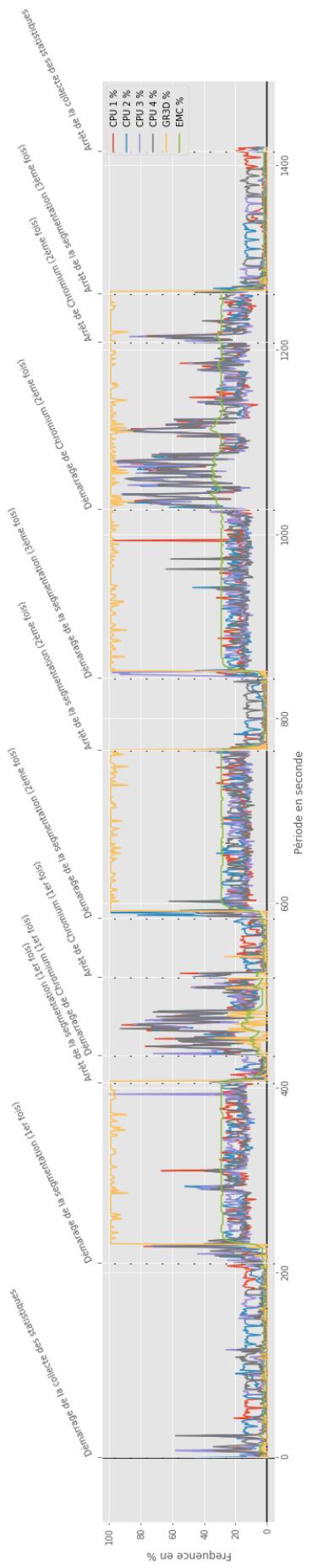


FIGURE 19 – Diagramme des performances système : la fréquence

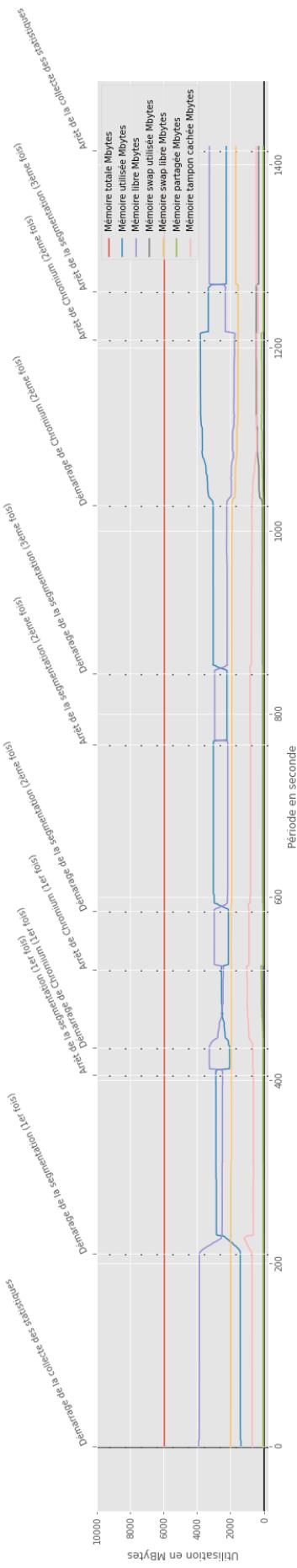


FIGURE 20 – Diagramme des performances système : la mémoire

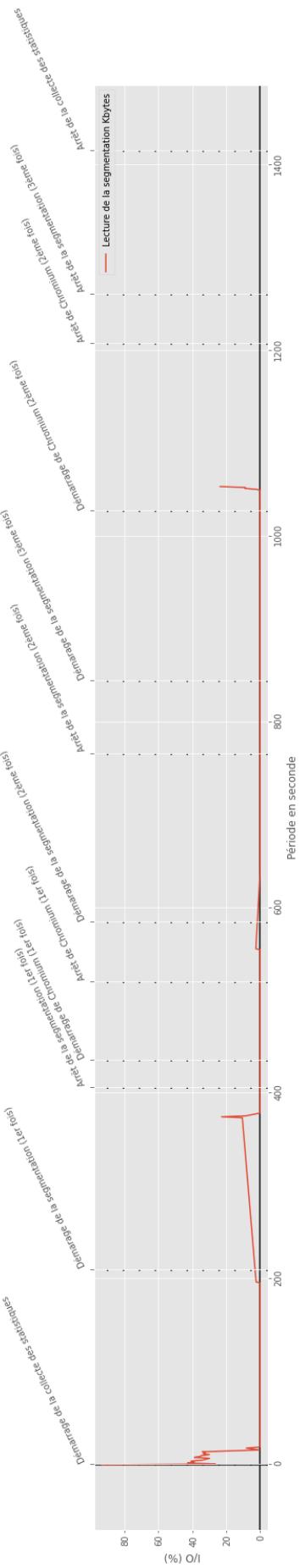


FIGURE 21 – Diagramme des performances système : le I/O total en % de la segmentation

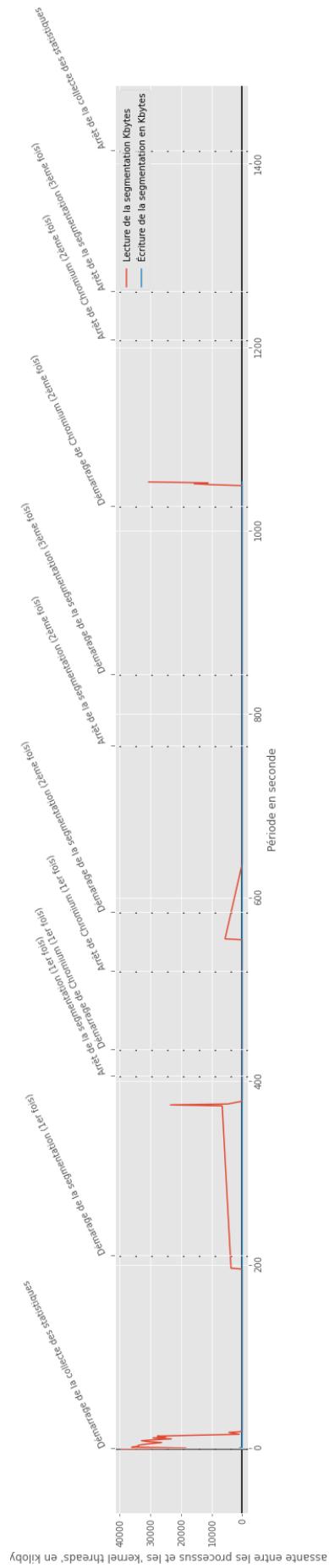


FIGURE 22 – Diagramme des performances système : le I/O en KBytes de la segmentation

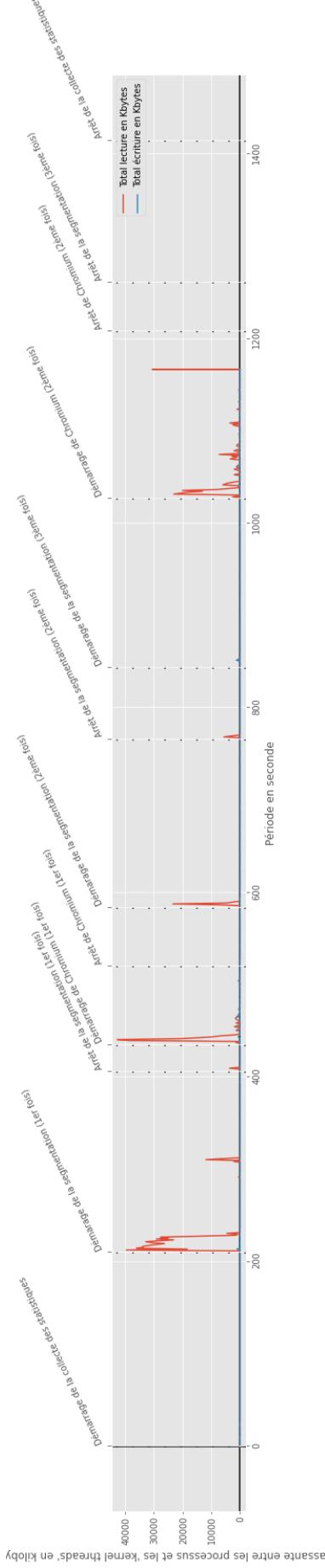


FIGURE 23 – Diagramme des performances système : le I/O total du disque en KBytes

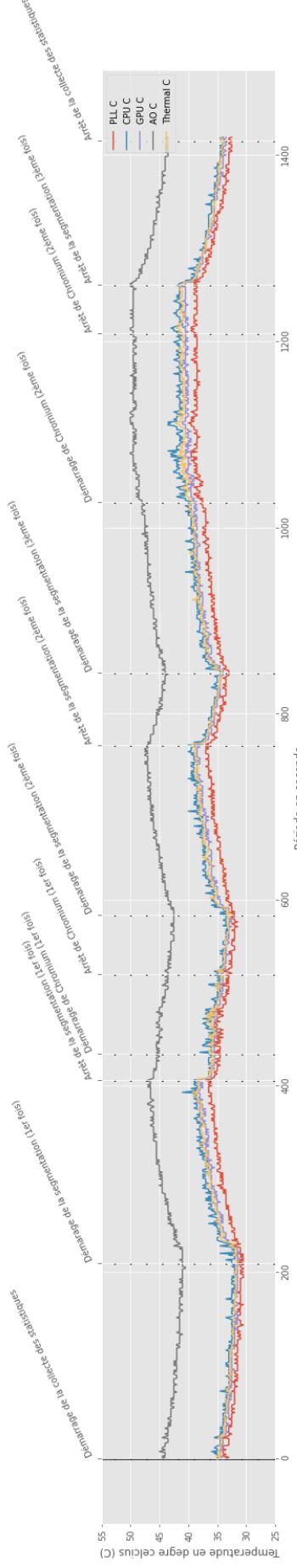


FIGURE 24 – Diagramme des performances système : les températures 27

27. PLL : Phase locking loop thermal sensor; AO : Always on thermal sensor. <https://forums.developer.nvidia.com/t/operating-temperature-range-on-jetson-nano/73555/10>

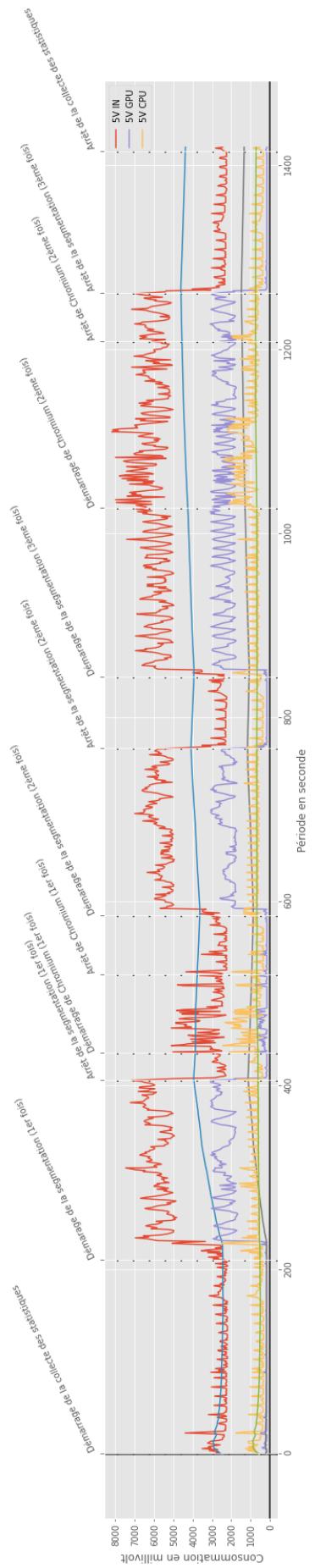


FIGURE 25 – Diagramme des performances système : la consommation

4.2 Performances de l'inférence

4.2.1 Images

Les tests ont été faits avec le modèle "fcn-resnet18-deepscene-576x320" fourni par NVIDIA.

Lors de l'entraînement et l'inférence, le script montre un IoU moyen du modèle de 75%. Mais l'objet d'intérêt de l'essai n'est pas la qualité de la segmentation de l'image complète, mais seulement de la piste cyclable. Certains efforts ont dû être dépensés²⁸ afin de pouvoir observer le IoU et le F1 score de la segmentation sémantique de la piste cyclable uniquement.

Le résultat de la segmentation sémantique peut-être visualisé avec ces deux photos, prises du jeu de donnée de test de la forêt de Freiburg et utiliser comme jeu de données de test pour le modèle. L'image utilisée possède une version vérité terrain (GT). L'image générée est l'image prédictive et peut être comparée avec l'image vérité terrain (GT), tant que la palette de couleur est identique à la version vérité terrain (GT).

Il s'avère que le IoU et F1 score sont assez élevés pour les deux photos pour la classe "Chemin".

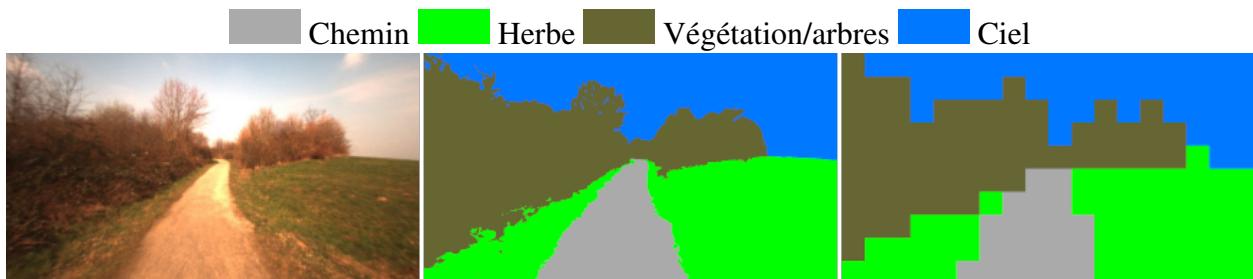


FIGURE 26 – (gauche) Image originale (b1-09517); (centre) vérité terrain (GT); (droite) segmentation sémantique générée par le modèle. Le IoU et le F1 score pour le chemin sont de +80%.

28. https://github.com/vince7lf/vince7lf.github.io/blob/master/_notebooks/2020-06-21-image_pred_color.ipynb

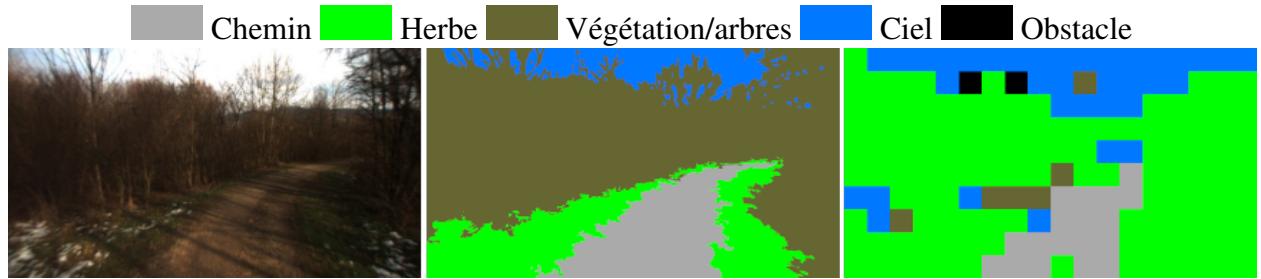


FIGURE 27 – (gauche) Image originale (b378-61) ; (milieu) vérité terrain (GT) ; (droite) segmentation sémantique générée par le modèle. Le IoU pour le chemin est +69%.

4.2.2 Vidéos

Il y a deux vidéos qui ont été utilisées pour tester les performances de la segmentation avec une vidéo. Comme il n'est pas évident de montrer une vidéo dans un essai, des liens sont mis à disposition. Ces vidéos sont disponibles dans le projet "Vision Météo" de Teams de l'Université de Sherbrooke. Chaque vidéo a été créée en filmant avec un téléphone intelligent l'écran du nano ordinateur pendant que la segmentation est exécutée. Cela produit une vidéo HD 1080p 30 FPS. Lors de la seconde vidéo, les performances du système et les statistiques "tegrastats" sont affichées en plus de la segmentation.

J'ai tenté de capturer le résultat (vidéos/images) de l'inférence directement depuis le nano ordinateur, mais ce n'est pas une bonne idée, car trop intrusif, l'inférence est ralentie. Deux images sont produites par le modèle : "overlay" et "mask", qui sont directement rafraîchies dans un XWindow.

- Lien²⁹ vers une courte vidéo de 30 secondes démontrant l'inférence en temps réel de la segmentation sémantique d'une vidéo de la piste cyclable du pont Jacques-Cartier dans des conditions ensoleillées, mais avec un angle de vue qui change rapidement. Inférence effectuée en 30 FPS 1280x720 avec le modèle "fcn-resnet18-deepscene-576x320" ;
- Lien³⁰ vers une vidéo longue de +8 minutes présentant l'inférence en temps réel de la segmentation sémantique d'une vidéo d'une piste cyclable dans des conditions ensoleillée, mais mouillée, avec présence de neige. Durée de plus de 8 minutes. La taille de la vidéo est de 800Mb. La même vidéo, d'une durée de 30 secondes, est utilisée successivement avec différentes images par seconde (60 / 30 / 15 / 1 FPS) et résolutions (720x1280 / 480x640 / 320x480 / 240x320). Le modèle est "fcn-resnet18-deepscene-576x320". Selon le titre de la

29. https://usherbrooke.sharepoint.com/sites/ProjetVisionMto/Documents/%20partages/General/projet_visionmeteo/videos/gae724_lefv2603/resultats/20200221_020044.mp4

30. https://usherbrooke.sharepoint.com/sites/ProjetVisionMto/Documents/%20partages/General/projet_visionmeteo/videos/gae724_lefv2603/resultats/20200412_232155.mp4

fenêtre xWindow présentant la segmentation, le FPS est autour de 23-26 FPS.

Voici un tableau montrant les différentes résolutions et images par seconde (FPS) qui ont été testées avec le modèle :

Tableau 7 – Résolutions et images par seconde (FPS) testés

Résolutions qui fonctionnent
320x576, 480x640, 720x1280, 768x1024, 768x1152, 800x1152, 832x1024, 864x1024
Résolutions qui ne fonctionnent pas
832x1120, 832x1152, 768x1280, 800x1280, 864x1152, 900x1152, 900x1280, 960x1600, 1080x1920, 1024x1024
Images par seconde (FPS) supportées
60/1, 30/1, 15/1, 1/1

4.3 Adaptation

Une tentative d'adaptation a été initiée. La première étape a été de vouloir re générer le fichier .onnx tel que NVIDIA le fournit, sans autre effort d'adaptation. En effet les commandes fournies par NVIDIA pour segmenter une image ou une vidéo permettent de préciser un modèle personnalisé, tel que le fichier .onnx, les classes, les codes couleurs. L'idée est donc de bénéficier de cette possibilité. Le seconde étape était de créer un jeu de données adapté au contexte, c'est-à-dire avec des photos de différentes sections de pistes cyclables avec une qualité de surface variable (sèche, mouillée, avec neige, ensoleillé, ombragé, etc.), avec les images fournies par l'APC-PJC et mon jeu personnel. Les difficultés attendues étaient d'uniformiser les résolutions des photos pour le modèle SegNet18, mais surtout de créer des images vérités terrain (GT), qui est beaucoup plus chronophage que difficile. Une fois le tout complété, la dernière étape serait de re entraîner le modèle SegNet18 avec ce jeu, et re générer le nouveau .onnx.

Malheureusement la première étape, de re générer le fichier .onnx tel quel, ne s'est pas déroulé aussi simplement qu'espéré, et a remis en question la suite de l'adaptation. La génération du fichier .onnx a tout d'abord réussi assez facilement, mais une erreur à l'exécution a remis en question

l'intégrité de ce fichier. Une longue période d'investigation a débuté³¹³², et finalement le fichier .onnx a pu être re généré avec succès après la phase de ré entraînement. Malheureusement le script fourni par NVIDIA qui adapte les photos du jeu de données de DeppScene pour le jeu d'entraînement et de test destiné au ré-entraînement du modèle SegNet18, génère des photos toutes noires, l'investigation s'est arrêtée à ce point.

5 Interprétation et discussion des résultats

5.1 Performances matérielles

5.1.1 Stockage de données

Les tests montrent que le SSD interne est de 4 à 11 fois plus efficaces qu'une carte microSD, pour l'opération de lecture de données.

5.1.2 Performances système

Performances globales

Concernant les performances globales du nano ordinateur, il est à noter que celui-ci est capable d'exécuter l'inférence en temps réel pour une durée prolongée (23 minutes dans ce cas), et rester réactif aux commandes. L'exemple qui le démontre est le démarrage du navigateur Chromium entre deux segmentations, et pendant la segmentation.

Fréquence

La commande "tegrastats" offre la fréquence des CPUs (4 pour le nano ordinateur), le GR3D (GPU) et EMC. On peut noter que l'inférence prend 100% du GR3D pendant toute la durée. Les CPUs sont tous utilisés équitablement pendant l'inférence, en dépassant rarement les 30% d'utilisation. En fait la période qui montre une exploitation élevée des CPUs est lors de l'utilisation de Chromium, où l'ensemble des CPUs sont employés entre 0% et 90%.

Il faut donc rester vigilant quant à l'utilisation des CPUs pendant l'inférence sur le long terme, au

31. https://github.com/vince7lf/vince7lf.github.io/blob/master/_posts/2020-05-13-train-onnx-nano.md

32. <https://forums.developer.nvidia.com/t/trying-to-regenerate-onnx-for-jetson-nano/125494?u=vincelf>

risque de perdre le système en raison d'un ralentissement progressif dû à un manque de ressources processeurs CPUs.

Mémoire

La commande "free -m" offre l'utilisation mémoire du système en Mb. Le nano ordinateur au démarrage ne consomme qu'environ 1.5Gb de mémoire totale, et possède 4Gb de libres sur un total de 6 Gb(d'où proviennent les 6Gb du graphique ?). À la fin du test de 25 minutes, il ne reste TODO qu'environ 3Gb de mémoire libre, un peu plus de 2Gb semble resté utilisé. De la mémoire swap a commencée à être consommée lors du démarrage de Chromium pendant la 3e segmentation, et ne semble jamais avoir été libérée. La mémoire tampon cachée est aussi sensiblement utilisée et revient un peu en dessous de son niveau original à la fin du test.

De même que pour l'utilisation des processeurs, il semble être préférable de rester vigilant lors de l'utilisation opérationnelle du nano ordinateur, la segmentation consommant de la mémoire qui semble ne plus être disponible pour les autres ressources du système, comme le démontre l'état de la mémoire totale libre à la suite de l'arrêt de la 1re segmentation.

I/O

La commande "iostat" offre les performances I/O du nano ordinateur pendant le test de 25 minutes. Le I/O de la segmentation est très raisonnable, de même que celle du système. Il n'y a quasiment pas d'opération visible en écriture, même la collecte des statistiques durant le test, aux secondes, n'apparaît pas. Les opérations en lecture sont plus visibles, mais très ponctuelles. La période la plus occupée en lecture semble être due durant le démarrage de la segmentation la première fois : le système semble lire le modèle en mémoire, et le conserver en mémoire, car les opérations en lecture suivantes sont peu ou non visibles pendant le démarrage des segmentations suivantes.

Cela expliquerait l'augmentation de l'utilisation de la mémoire à la suite de la segmentation.

Température

La commande "tegrastats" offre grâce à des capteurs intégrés à la carte mère la température de différents éléments matériels du nano ordinateur. La commande "sudo jetson_clock" est démarrée manuellement dès que le système est démarré, permettant de profiter de la fréquence maximale d'utilisation supportée par le nano ordinateur. Le succès de la commande est simple à vérifier : le

ventilateur se met à ventiler aussitôt³³.

La température dans la pièce au moment du test est de 27C. Au démarrage, on note que la température mesurée de la plupart des capteurs thermiques, sauf pour le AO ("Always on") est entre 33C et 36C. Le démarrage de la 1re segmentation fait graduellement monter la température, entre 37C et 39C, jusqu'au point d'arrêt de la segmentation, après 200 secondes approximativement, et qui diminue graduellement approximativement pendant 200 secondes vers son point d'origine lorsqu'elle est arrêtée. Le démarrage de Chromium pendant cette période semble ralentir un peu le refroidissement. L'observation lors de la seconde segmentation est identique à la première. La troisième segmentation est plus longue, 400 secondes, et voit la température se stabiliser entre 41C et 43C. L'arrêt de la segmentation voit la température baisser et revenir assez rapidement à sa température d'origine.

Le capteur thermique AO ("Always on") est plus particulier, puisqu'il mesure une température de 10C supérieures aux autres capteurs. Selon le modérateur Trumany de NVIDIA³⁴, "AO_therm is used for a truly robust thermtrip and as an LP0 wake source, as other zones will cease to operate during LP0.". Mes compétences en la matière ne me permettent pas d'expliquer clairement ce renseignement, mais cela semble signifier que ce capteur est plus robuste que les autres et devient l'indicateur de référence pour gérer une surchauffe.

Il est donc à noter que l'opérationnalisation constante de la segmentation aurait un impacte non négligeable sur la durée de vie du Jetson Nano. Selon la documentation de NVIDIA, une carte Jetson Xavier TX2i qui opère 24/7, selon certaines conditions, a une durée de vie théorique de 4,4 années³⁵

Au besoin, plus d'informations peuvent être trouvées dans le guide de conception thermique du Jetson Nano³⁶

33. https://docs.nvidia.com/jetson/14t/index.html#page/Tegra\%20Linux\%20Driver\%20Package\%20Development\%20Guide/power_management_nano.html

34. <https://forums.developer.nvidia.com/t/operating-temperature-range-on-jetson-nano/73555/10>

35. https://docs.nvidia.com/jetson/14t/index.html#page/Tegra\%2520Linux\%2520Driver\%2520Package\%2520Development\%2520Guide\%2Fjetson_module_support.html

36. https://developer.download.nvidia.com/assets/embedded/secure/jetson/Nano/docs/Jetson_Nano_Thermal_Design_Guide_TDG-09383-001_v1.3.pdf?2P65awpy13RwXu6jWjsqFgresjNSqhO-N2uI3BNH2Wcbp9LNh91GF3UtmC3JgEWd6MX2-BC5xoL80tY5Wpl5cEltIMR4IawEflJehkxKH3yDAgxV-HpXyOo5Ge8a32mdntMcfRzjRZZTP2-hsJ1IuT5FB7G36zhkCva7uPS9ntgWDff-w1W0LBJLH5DvpE1qU-3yZM5hjSz9g9cpFM

Consommation

La commande "tegrastats" offre de visualiser la consommation du nano ordinateur, soit globale, pour les CPUs et pour le GPU. En mode opérationnel continu, cela peut avoir une importance sur le budget, car la consommation est clairement beaucoup plus élevée pendant la segmentation. Il peut être observé aussi qu'elle est beaucoup plus volatile avec Chromium démarrée.

5.2 Performances de la segmentation

5.2.1 Images

La segmentation prédite pour la classe "Trail" est assez surprenante. Le IoU est de 89% et de 69% respectivement dans le cas des deux images évaluées, ce qui est très encourageant. Par contre les délimitations de la segmentation pour le chemin sont décevantes et questionnables, car le modèle retourne une résolution très faible, l'image est très grossièrement "pixelisée", de gros carrés sont utilisés pour délimiter chaque classe. C'est probablement dû au fait que l'architecture du modèle segnet18 n'utilise que 18 couches, et qu'il n'y a donc que peu de représentations possibles pour les classes.

5.2.2 Vidéos

La segmentation des vidéos n'a pas pu être évaluée avec des indicateurs de performances. C'est donc subjectivement que l'on peut donner une appréciation. Comme les images se succèdent très rapidement, ce n'est pas non plus évident. Grossièrement je n'ai pas trouvé le résultat de la segmentation très bonne. Les vidéos proviennent d'un contexte autre que celui du jeu de données d'entraînement (la forêt), la surface de la piste est bien différente.

Je ne sais pas trop comment la segmentation en temps réel de vidéos peut être utilisée, car il n'y a pas moyen d'évaluer la qualité de la segmentation sans avoir la vérité terrain (GT).

6 Conclusion et recommandations

6.1 Objectif principal

L'objectif principal de l'essai était d'évaluer la capacité du nano ordinateur "NVIDIA Jetson Nano" à exécuter, en temps réel, un modèle de réseau de neurones à convolution entier (FCN) permettant

la segmentation sémantique d'une vidéo d'une piste multifonctionnelle. Il faut découper en plusieurs faits cet objectif afin de bien pouvoir l'évaluer :

- Le nano ordinateur est capable d'inférer un modèle FCN pour segmenter sémantiquement une vidéo représentant une piste cyclable.
- La segmentation sémantique découlant de l'inférence n'a pas pu être mesurée, il n'y a aucun moyen qui m'est connu afin de récupérer un indicateur, un coefficient ou un score me permettant de juger si la segmentation d'une vidéo est bonne ou non, comme pour une image ou la mesure du IoU ou du F1 score est possible si l'image de la vérité terrain (GT) est disponible.
- L'image générée par le modèle FCN SegNet18 a une résolution très faible, de l'ordre de 19x10. La délimitation de la segmentation, entre chaque classe, est donc très très grossière.
- Le "temps réel" a été simulé, et n'est donc pas celui qui sera utilisé sur le terrain.
- le nano ordinateur et le modèle FCN supporte l'inférence d'une vidéo HD (résolution de 720x1280 = 720p) avec un nombre d'images par seconde de 60/1 FPS.

D'un point de vue performance matérielle et logicielle, le nano ordinateur est capable d'inférer avec un modèle FCN une vidéo pour la segmenter. Par contre, d'un point de vue qualitatif, 1) la qualité de la segmentation ne peut pas être mesurée. De plus, 2) la segmentation prédite est très imprécise.

La première contrainte qualitative semble être un défaut majeur. Mais si on replace l'objectif dans le contexte de la détection de la délimitation d'une piste cyclable, à partir d'un point de vue fixe, on peut s'interroger sur le besoin de faire de la télédétection en temps réel avec une vidéo en haute résolution.

La deuxième contrainte pourrait potentiellement être améliorée en utilisant un modèle dont l'architecture est plus performante, mais implicitement plus complexe, telle que le modèle SegNet101 ou DeepLabV3, mais qui risque d'être aussi plus demandant en ressources matérielles, GPU, CPU et mémoire. Ce qui risque de remettre en question les performances matérielles et logicielles du nano ordinateur. C'est ainsi probables la raison pour laquelle NVIDIA procure uniquement des jeux de modèles préentraînés de segmentation sémantique avec SegNet18 pour le nano ordinateur.

6.2 Limites

6.2.1 Limites matérielles

Au sujet des limites matérielles, durant l'inférence, il n'y a aucune limite qui est ressortie lors des tests de performance. Par contre il a été lu qu'un mode opérationnel 24/7 n'offrait qu'une durée de vie de 4.4 années au nano ordinateur.

6.2.2 Limites applicatives

Au sujet des limites applicatives, durant l'inférence, il n'y a aucune limite qui est ressortie lors des tests de performance. Par contre il a été observé durant l'essai que le nano ordinateur ne devrait pas être utilisé comme machine de développement, pour par exemple pour re entraîner un modèle. L'entraînement du modèle SegNet18 n'a pas fonctionné dans un environnement virtuel Python, ni dans un conteneur Docker sur le nano ordinateur, celui-ci arrête de fonctionner. Il n'y a pas eu d'investigation, mais il semble que le nano ordinateur atteint une limite mémoire qui le ralenti jusqu'à un arrêt de fonctionnement. DIGITS ne peut pas non plus être utilisé, car il n'est pas compatible avec l'architecture ARM du nano ordinateur. Si l'objectif est d'améliorer le modèle en le ré entraînant à la demande en mode opérationnel, l'entraînement et l'inférence ne peuvent cohabiter simultanément, cela me semble donc impossible aujourd'hui, à moins d'investiguer et de trouver un moyen d'optimiser les ressources.

Durant l'essai, il a aussi été observé que l'utilisation prolongée de Chromium peut impacter les performances du nano ordinateur en le ralentissant grandement.

6.3 Optimisation

6.3.1 Optimisation matérielle

Plusieurs initiatives ont été tentées afin d'optimiser le matériel. L'optimisation requise est celle d'utiliser un adaptateur 5V 4Amp, recommandé et fiable, afin de fournir assez de puissance au nano ordinateur lorsque d'autres périphériques viennent s'y raccorder, comme une caméra et un ventilateur. Profiter du PoE de l'interface réseau n'a pas été testé, mais cela semble aussi être une option rapide et simple à mettre en place pour assister l'adaptateur. Enfin, forcer le démarrage du ventilateur dès le démarrage du nano ordinateur est une autre optimisation simple, mais efficace à appliquer. Par contre, je ne recommande pas l'utilisation d'un dongle ou adaptateur Wifi, celui-ci étant très énergivore, peu efficace, non fiable, ni stable. Il prendrait de plus un pourcentage d'utilisation non négligeable du Hub USB 3.0.

La seconde optimisation qui a été logiquement tentée est celle d'utiliser un SSD à la place d'une microSD, car il y aurait, selon moi, beaucoup d'avantages. Pour des raisons de performances d'abord, le gain peut-être d'au moins 4 fois plus grand en opération de lecture I/O. Ensuite, en durée de vie, une carte microSD est fragile et ne peut être considérée comme un système fiable sur le long terme. D'un point de vue capacité de stockage, un SSD peut offrir beaucoup mieux. Enfin, un SSD est plus adapté à la gestion d'un système opérationnelle et la manipulation de petits fichiers. En contrepartie, un SSD va demander plus de puissance (Watt) au nano ordinateur, et générer plus de chaleur. Ma recommandation serait de trouver un disque SSD interne au format NVMe, connecteur de type M.2, assez pratique pour être branché au port PCIe du nano ordinateur.

Une autre optimisation matérielle qui n'est pas à négliger est le boitier. Vu que le système a été conçu pour être en opération continue sur le terrain, un boitier bien conçu permet de le protéger sur le long terme. Il doit être bien adapté à ses périphériques, que sont la caméra et le ventilateur, et optionnellement un SSD interne.

6.3.2 Optimisation logicielle

La version du modèle SegNet18 fournit par NVIDIA s'exécute avec fluidité, sans que l'on sente que le nano ordinateur puisse devenir non réactif. Au démarrage de l'inférence, il y a une brève période de 2-3 secondes où le nano ordinateur ne répond plus. Mais sinon, il est tout à fait possible d'utiliser le nano ordinateur pendant l'inférence d'une vidéo ou avec la caméra, et même avec 5-6 onglets d'ouverts dans Chromium. Lorsque le nombre d'onglets, ou d'instances de Chromium, devient trop grand, il a été observé que le nano ordinateur devenait lent, limite non fonctionnel, jusqu'à la fermeture des onglets. Ceci est probablement dû à une limitation mémoire.

Autrement, certaines corrections au code C++ ont dû être apportées au code source original fourni par NVIDIA : l'image de la caméra est à l'envers (et je ne pouvais monter la caméra dans le sens opposé dans le boitier) ; le pipeline gstreamer interne de l'application est trop spécifique pour supporter un flux vidéo autre que celui provenant de la caméra ; et la taille de la fenêtre XWindow qui s'ouvre pour afficher la segmentation de la vidéo est programmée pour prendre tout l'écran, nous faisant perdre ainsi l'accessibilité et visibilité aux autres fenêtres.

Segmentation

Comme observé durant les tests, la résolution de la segmentation avec le modèle SegNet18 est très

faible. Le désavantage majeur dans le contexte de cet essai est que les délimitations des classes sont très approximatives, incluant celle du chemin. Même si le IoU et le F1 score sont pourtant très acceptable pour cette classe. Il semble que ce serait l'élément prioritaire à améliorer.

Adaptation

Même si la phase d'adaptation a pu être initiée durant l'essai, elle n'a pas durée très longtemps : re générer le même fichier interopérable .onnx avec le code source original a été très laborieux. Il est vrai que NVIDIA propose, avec DIGITS, un environnement de re entraînement et d'adaptation des modèles qu'ils offrent. Mais dans le contexte de cet essai, je n'avais à ma disposition que l'environnement de Compute Canada. Néanmoins je pense qu'il est important de pouvoir le faire tout en gardant le contrôle de son environnement, par exemple pour permettre d'adapter le modèle de notre choix, plus performant, tel que SegNet101 ou DeepLabV3, entraîné avec le jeu de données DeepScene, et l'adapter à un jeu de données personnalisé. Le questionnement est de savoir comment le nano ordinateur réagit avec l'inférence d'un modèle beaucoup plus gros et complexe que SegNet18. Dans une autre perspective, il serait bon de considérer un modèle de nano ordinateur plus performant, tel que le Jetson Xavier AGX.

6.4 Accès distant

L'un des sous-objectifs était de permettre un accès à distance sécurisé au nano ordinateur. Pour des raisons de temps, aucune activité de recherche ni de test n'a été effectuée dans le cadre de cet essai.

6.5 Documentation

Un gros effort de documentation du cheminement de l'essai a été fait. La documentation des activités, des procédures, des scripts, des modifications, des erreurs, des références, et bien d'autres éléments encore, est disponible publiquement dans le blogue sur GitHub³⁷.

³⁷. <https://github.com/vince71f/vince71f.github.io>

Références

- ABOUZAHIR, S., SADIK, M. & SABIR, E. (2017). IoT-Empowered Smart Agriculture : A Real-Time Light-Weight Embedded Segmentation System. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 319-332. https://doi.org/10.1007/978-3-319-68179-5_28
- ALOM, Z., TAHA, T. M., YAKOPCIC, C., WESTBERG, S., SIDIKE, P. & NASRIN, M. S. (2018). The History Began from AlexNet : A Comprehensive Survey on Deep Learning Approaches, 39.
- BEAM, A. (2017). *Deep Learning 101 - Part 1 : History and Background*. https://beamandrew.github.io/deeplearning/2017/02/23/deep_learning_101_part1.html
- BERNAS, M., PLACZEK, B. & SAPEK, A. (2017). Edge Real-Time Medical Data Segmentation for IoT Devices with Computational and Memory Constrains. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 119-128. https://doi.org/10.1007/978-3-319-67077-5_12
- BLANCO-FILGUEIRA, B., GARCÍA-LESTA, D., FERNÁNDEZ-SANJURJO, M., BREA, V. M. & LÓPEZ, M. (2019). Deep Learning-Based Multiple Object Visual Tracking on Embedded System for IoT and Mobile Edge Computing Applications. *IEEE Internet of Things Journal*, 5423-5431. <https://doi.org/10.1109/JIOT.2019.2902141>
- CHOLLET, F. (2018). *Deep learning with Python* [OCLC : ocn982650571]. Manning Publications Co.
- CHONG, C. P., SALAMA, C. A. T. & SMITH, K. C. (1992). Real-Time Edge Detection and Image Segmentation. *Analog Integrated Circuits and Signal Processing*, 117-130. <https://doi.org/10.1007/BF00142412>
- COPEL, M. (2016). *What's the Difference Between Deep Learning Training and Inference ?* <https://blogs.nvidia.com/blog/2016/08/22/difference-deep-learning-training-inference-ai/>
- des piétons et cyclistes du pont JACQUES-CARTIER, A. (2020). PontJacques-Cartier365.com. <http://pontjacquescartier365.com>
- des piétons et cyclistes pont JACQUES-CARTIER, A. (2020). Flickr Association des piétons et cyclistes pont Jacques-Cartier. <https://www.flickr.com/photos/pontjacquescartier>
- DETTMERS, T. (2015). *Deep Learning in a Nutshell : History and Training*. <https://devblogs.nvidia.com/deep-learning-nutshell-history-training/>
- DUSTIN, F. (2019). *Realtime Semantic Segmentation on Jetson Nano in Python and C++*. <https://www.linkedin.com/pulse/realtime-semantic-segmentation-jetson-nano-python-c-dustin-franklin>

- JIACONDA. (2019). *A Concise History of Neural Networks*. <https://towardsdatascience.com/a-concise-history-of-neural-networks-2070655d3fec>
- KOH, J. Y. (2018). *Model Zoo - Deep Learning Code and Pretrained Models for Transfer Learning, Educational Purposes, and More*. <https://modelzoo.co/>
- KURENKOV, A. (2015). *A 'Brief' History of Neural Nets and Deep Learning*. <https://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/>
- LONG, J., SHELHAMER, E. & DARRELL, T. (2015). Fully Convolutional Networks for Semantic Segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431-3440. <https://doi.org/10.1109/CVPR.2015.7298965>
- MODY, M., KUMAR, D., SWAMI, P., MATHEW, M. & NAGORI, S. (2018). Low Cost and Power CNN/Deep Learning Solution for Automated Driving. *Proceedings - International Symposium on Quality Electronic Design, ISQED*, 432-436. <https://doi.org/10.1109/ISQED.2018.8357325>
- NGUYEN, T., SHIVAKUMAR, S. S., MILLER, I. D., KELLER, J., LEE, E. S., ZHOU, A., OZASLAN, T., LOIANNO, G., HARWOOD, J. H., WOZENCRAFT, J., TAYLOR, C. J. & KUMAR, V. (2019). MAVNet : An Effective Semantic Segmentation Micro-Network for MAV-Based Tasks. *arXiv:1904.01795 [cs]*. <http://arxiv.org/abs/1904.01795>
- NVIDIA. (2019a). *Jetson Nano*. <https://developer.nvidia.com/embedded/jetson-nano>
- NVIDIA. (2019b). *Jetson Nano : Deep Learning Inference Benchmarks*. <https://developer.nvidia.com/embedded/jetson-nano-dl-inference-benchmarks>
- PATHAK, D. & EL-SHARKAWY, M. (2019). Architecturally Compressed CNN : An Embedded Realtime Classifier (NXP Bluebox2.0 with RTMaps). *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 0331-0336. <https://doi.org/10.1109/CCWC.2019.8666495>
- PJCCI. (2018a). Fiche de la piste multifonctionnelle du pont Jacques-Cartier. https://jacquescartierchamplain.ca/wp-content/uploads/2018/10/IMG_Fiche_piste-multi_pont_JC_FR_vfinale_web_2018-10-10.pdf
- PJCCI. (2018b). Rapport post-mortem sur le projet pilote d'entretien hivernal de la piste multifonctionnelle du pont Jacques-Cartier. https://jacquescartierchamplain.ca/wp-content/uploads/2018/10/RPP_piste_PJC_2018-10-10-1.pdf
- SHARMA, N., SHAMKUWAR, M. & SINGH, I. (2019). *The History, Present and Future with IoT*. Springer Science ; Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-030-04203-5_3

- WU, X., SAHOO, D. & HOI, S. C. H. (2019). Recent Advances in Deep Learning for Object Detection [arXiv : 1908.03673]. *arXiv :1908.03673 [cs]*. Récupérée 9 août 2020, à partir de <http://arxiv.org/abs/1908.03673>
- ZHENG, J., LI, J., LIU, Y. & ZHANG, W. (2020). Real-Time Semantic Segmentation Network for Edge Deployment. In Y. JIA, J. DU & W. ZHANG (Éd.), *Proceedings of 2019 Chinese Intelligent Systems Conference* (p. 243-249). Springer Singapore. https://doi.org/10.1007/978-981-32-9698-5_28

7 Annexes

7.1 Exemples de nano ordinateurs qui supportent les SDK pour l'IA

Tableau 8 – Comparaison des trois nano ordinateurs supportant les SDK pour l'IA

NVIDIA Jetson Nano	NVIDIA Jetson Xavier AGX	Raspberry Pi 4B + Intel NCS2
99USD	599USD	134USD (55USD + 79USD)
45 x 69.6 mm, 250 gr, 5-10 W	100 x 87 mm, 630 gr, 10-15-30 W	56 x 85.60 mm + 27x72 mm, 45 gr + 18.1 gr, 15 W
128-core NVIDIA Maxwell GPU	512-core NVIDIA Volta GPU with 64 Tensor Cores	Intel Movidius Myriad X VPU 16 SHAVE cores
Quad-Core ARM Cortex-A57 MPCore	8-core NVIDIA Carmel Arm v8.2 64-bit CPU 8MB L2 + 4MB L3	Quad-core ARM Cortex-A72 64-bit @ 1.5 GHz
4 GB 64-bit LPDDR4	32 GB 256-bit LPDDR4	4GB LPDDR4
0.47 TFLOPS@FP16	5.5-11.5 TFLOPS@FP16; 20-32 TOPS@INT8	4 FLOPS@FP16, 1 TOPS@INT8

7.2 Communication avec l'Association des Piétons et Cyclistes du Pont Jacques-Cartier

L'Association des Piétons et Cyclistes du Pont Jacques-Cartier (APC-PJC) a été contacté afin de leur demander la permission d'utiliser leurs fichiers multimédias de la piste cyclable du pont Jacques-Cartier, tel que leurs images et leurs vidéos. Voici les détails de la communication et les conditions d'utilisation.

7/31/2020

Courriel - Vincent Le Falher - Outlook

RE: Bonjour !

Mickaël Germain <Mickael.Germain@USherbrooke.ca>

Mer 2020-02-19 23:30

À : Vincent Le Falher <Vincent.Le.Falher@USherbrooke.ca>; Piétons-cyclistes pont Jacques-Cartier <apc.pontjc@gmail.com>

Bonjour,

Merci pour les informations. Nous n'avons pas d'entente avec PICCI pour le rapport de maîtrise. Nous ne partagerons pas vos informations sans votre accord.

Cordialement,
Mickaël

De : Vincent Le Falher <Vincent.Le.Falher@USherbrooke.ca>

Envoyé : 18 février 2020 17:55

À : Piétons-cyclistes pont Jacques-Cartier

Cc : Mickaël Germain

Objet : Re: Bonjour !

Bonjour M. Démontagne,

Je copie mon directeur de projet pour le notifier des conditions entourant l'usage des fichiers médias (photos et vidéos) que vous nous permettrez gentiment d'utiliser aux fins de mon essai de recherche pour étude.

Je comprends vos demandes et je les appliquerai en bonne et due forme, cela me fera plaisir.

@Mickaël stp noter qu'il est important de notifier l'association si les fichiers médias sont utilisés par d'autres étudiants, *surtout* dans le contexte du projet avec PICCI.

+ De plus sais-tu s'il existe une entente ou convention avec PICCI par rapport à l'utilisation du rapport de maîtrise ?

@M. Démontagne je vous remercie encore pour votre aide, et, si vous le désirez, je pourrais vous tenir au courant de l'évolution de mon projet. Mon objectif est de compléter mon essai avant la mi-aout 2020.

Au plaisir de discuter de nouveau avec vous, au besoin.

Vincent

--

Vincent Le Falher
514-229-3863

De : Piétons-cyclistes pont Jacques-Cartier <apc.pontjc@gmail.com>

Envoyé : 18 février 2020 17:11

À : Vincent Le Falher <Vincent.Le.Falher@USherbrooke.ca>

Objet : Re: Bonjour !

Bonjour Vincent,

Suite à notre conversation de ce mardi midi, je vous autorise à utiliser les photos et vidéos disponibles sur le [compte Flickr](#) de l'Association ainsi qu'aux vidéos disponibles sur notre [compte YouTube](#).

Le dossier complet (photos, vidéos) contient environ 250 Go de données (classement par années). Celui qui correspond sensiblement aux données du compte Flickr environ 30 Go (classé par thèmes).

Nous vous demanderons simplement de **mentionner l'Association des piétons et cyclistes du pont Jacques-Cartier** lors de l'attribution des crédits des médias. Nous vous demandons également que les médias utilisés dans le cadre de votre projet de maîtrise servent à alimenter uniquement votre projet. **S'ils devaient être transférés à PICCI pour être exploités dans le cadre des opérations de la Société (entretien, études, etc.) nous vous demanderons de nous avertir.** D'autre part, si vous avez une **entente ou convention avec PICCI** par rapport à l'utilisation de votre rapport de maîtrise, nous vous demanderons de nous la faire parvenir pour que nous puissions prendre connaissance de son contenu.

<https://www.flickr.com/photos/151964858@N02/albums>
<https://www.youtube.com/channel/UCD2cmxKIEP88LmZ1chTKHw>

Cordialement,

François Démontagne,

Président de l'association des piétons et cyclistes du pont Jacques-Cartier

Le sam. 15 févr. 2020, à 13 h 32, Vincent Le Falher <Vincent.Le.Falher@usherbroke.ca> a écrit :

Bonjour M. Démontagne. Merci pour votre intérêt. Je vais tenter de vous contacter Mardi sur l'heure du déjeuner. À Mardi. Vincent.

De : Piétons-cyclistes pont Jacques-Cartier <apc.pontjc@gmail.com>

Envoyé : 12 février 2020 16:02

À : Vincent Le Falher <Vincent.Le.Falher@USherbrooke.ca>

Cc : Mickaël Germain <Mickael.Germain@USherbrooke.ca>

Objet : Re: Bonjour !

<https://outlook.office.com/mail/search/id/AAQkAGRkMzY2ZjNLTJjZTMtNGI2My1hNTc2LTQ0ZlxMzExNzYxMAAQAGWvRXbRHROhn9Ro85udEw%3D> 1/2

7/31/2020

Courriel - Vincent Le Falher - Outlook

Bonjour M. Le Falher,

Nous disposons effectivement d'un certain nombre de photos et vidéos de la piste dans différentes conditions d'utilisation.

Je vous propose de me contacter en soirée ou entre 12 h et 13 h au 514-927-6366 pour discuter de votre projet.

Au plaisir,

François Démontagne

Président de l'Association des piétons et cyclistes du pont Jacques-Cartier

Le mer. 12 févr. 2020, à 13 h 00, Vincent Le Falher <Vincent.Le.Falher@usherbrooke.ca> a écrit :

Bonjour cher responsable de l'Association des piétons et cyclistes du PJC.

Mon nom est Vincent Le Falher. Je suis étudiant à l'université de Sherbrooke et je suis en plein essai de recherche en géomatique appliquée dans le cadre de la Maîtrise en géographie, cheminement géodéveloppement durable. Et mon sujet de recherche a pour site d'étude la piste multifonctionnelle du pont Jacques Cartier. Bhe oui 😊 !

Voici une page d'introduction et de mise en contexte de mon projet de recherche, si cela vous intéresse: <https://vince7lf.github.io/about.html>

En gros, je travaille sur un système qui permettrait de détecter automatiquement les délimitations de la piste cyclable, peu importe l'angle de vue et les conditions de la surface (mouillée, neige, etc).

Et j'aurais besoin de "données", c'est à dire des images et des vidéos de la piste multifonctionnelle. J'en ai besoin pour tester et adapter des modèles de reconnaissance d'images. Et c'est super car vous en avez ... pas mal quand même (<https://www.flickr.com/photos/pontjacquescartier/>). J'ai trouvé le tout grâce à votre site (<http://pontjacquescartier365.com/contact/>), que j'ai trouvé en faisant des recherches d'images via Google.

Le premier objectif de mon email est de vous demander la permission d'utiliser ces images et ces vidéos. À des fins de recherche pour ma maîtrise. Le second objectif, si vous me donner la permission, est de pouvoir récupérer ces images et vidéos, incluant toutes leurs différentes résolutions.

Si vous voulez en discuter, cela me fera plaisir, je suis disponible au:

514-229-3863

Ou via email aussi.

J'attends de vos nouvelles avec impatience.

En attendant, je vous souhaite une excellente journée ! Merci

Vincent Le Falher

--

--

Association des piétons et cyclistes du pont Jacques-Cartier

<https://www.facebook.com/association.pietons.cyclistes.pont.jacques.cartier/>
<https://twitter.com/APCPontJCcartier>

--

--

Association des piétons et cyclistes du pont Jacques-Cartier

<https://www.facebook.com/association.pietons.cyclistes.pont.jacques.cartier/>
<https://twitter.com/APCPontJCcartier>