

Projet Ingénieur

Mesurer les Infrastructures Routières

Equipe:

ALLEMAND Fabien
BALAKRISHNAN Sylvain
BONNAIL Julie

Alcatel • Lucent



Table des matières

Liste des figures	2
1 Introduction	3
2 Rappels R1 et R2	3
3 Application Android	3
3.1 Utilité	3
3.2 Développement	4
3.3 État de l'application	4
3.4 Rôle futur envisagé	4
4 Collecte de Données	5
5 Augmentation des Données	5
5.1 Renverser la séquence dans le Temps	7
5.2 Renverser l'Axe de Rotation	8
5.3 Ajout de Bruit	8
5.4 Ajout d'un Signal Porteur	8
5.5 Découpage des Séquences	8
6 Intelligence Artificielle	9
6.1 Création du Jeu de Données	9
6.2 Premier Modèle	9
6.3 Méthodes Envisagées	9
7 Conclusion	10
Bibliographie	11

Liste des figures

1	Données accélérométriques enregistrées sur le robot Scout 2.0 avec le dispositif Arduino . . .	4
2	Capture d'écran d'Android Studio lors du développement de l'application	5
3	Tableur récapitulatif de la collecte de données	6
4	Comparaison d'un enregistrement et de sa version modifiée (renversement du temps)	7
5	Comparaison d'un enregistrement et de sa version modifiée (renversement d'un axe horizontal)	8
6	Comparaison d'un enregistrement et de sa version modifiée (ajout de bruit)	9

1 Introduction

Les infrastructures routières sont essentielles lorsqu'il s'agit de voyager. Qu'il s'agisse de déplacements quotidiens ou de voyages ponctuels, des millions de personnes conduisent leur véhicule sur la route pour aller d'un point A à un point B. Il est indéniable que l'état de détérioration des routes a un impact considérable sur la sécurité des conducteurs et des passagers. Un trou inattendu sur une route peut amener le conducteur à changer brusquement de direction ou à perdre le contrôle de son véhicule. L'effet d'une route mal entretenue sur un véhicule est généralement négligé, mais il semble logique que les trous et les bosses sur une route soient susceptibles d'endommager les véhicules, réduisant ainsi la sécurité et augmentant les coûts d'entretien du véhicule. À plus grande échelle, les transports peuvent être ralentis par des routes détériorées, ce qui signifie que l'ensemble du processus d'échanges économiques se déroule à un rythme plus lent, ce qui nuit à l'économie des villes ou même des pays. Enfin, la force militaire d'un pays peut être évaluée en fonction de l'état des réseaux routiers. En cas d'urgence, les forces militaires doivent se déplacer rapidement. Là encore, l'état des routes est un facteur clé.

L'objectif de ce projet est de développer une solution basée sur l'IA afin de faciliter l'entretien des routes. En formant une IA à reconnaître les dégradations sur une route, les cantonniers pourraient plus facilement entretenir les routes et ainsi améliorer la sécurité et l'expérience des usagers.

Dans cette étude, l'IA sera principalement entraînée sur les données d'accélération mesurées sur les véhicules. Pour fonctionner correctement, le modèle doit être capable de détecter diverses dégradations (bosses et obstacles, trous et fissures ainsi que graviers), quel que soit le type de véhicule d'où proviennent les données. Pour les besoins de l'étude, deux méthodes seront utilisées pour collecter les données d'accélération. Tout d'abord, en utilisant une carte Arduino et une *unité de mesure inertielle* (IMU). Dans un second temps, en utilisant l'accéléromètre contenu dans les smartphones.

Comme preuve de concept, une application pour smartphone sera créée et démontrera l'efficacité de cette méthode d'évaluation de la qualité des routes.

2 Rappels R1 et R2

Comme pour tout projet, les premières étapes avaient consisté à comprendre la problématique, les enjeux et les défis. Nous avons effectué quelques recherches sur l'évaluation de la qualité des routes et sur des sujets connexes afin de connaître les solutions existantes ou les méthodes pertinentes.

Nous avons ensuite pu définir les objectifs des sprints et les organiser en utilisant les *user stories* et la complexité des sprints. Les sprints avaient ensuite été utilisés pour planifier l'avancement du projet.

Le projet avait été divisé en trois parties:

- Collecter des données
- Construire et entraîner une intelligence artificielle
- Développer une application Android

Grâce au petit dispositif à base d'Arduino, nous avons pu réaliser de premiers enregistrements, commencer à analyser ces données et les comparer aux données provenant des deux autres sources que nous avons à disposition (les données de M. Helbert et les données du jeu de données trouvé sur internet).

En parallèle, nous avons aussi débuté le développement d'une application Android pour permettre d'enregistrer des données plus précises et de façon plus efficace.

3 Application Android

3.1 Utilité

Dans le but de pouvoir entraîner un modèle pour prédire le type de dégradations rencontrées sur la route de manière adéquate, il est nécessaire d'avoir un nombre important de données. C'est pourquoi rendre la collecte

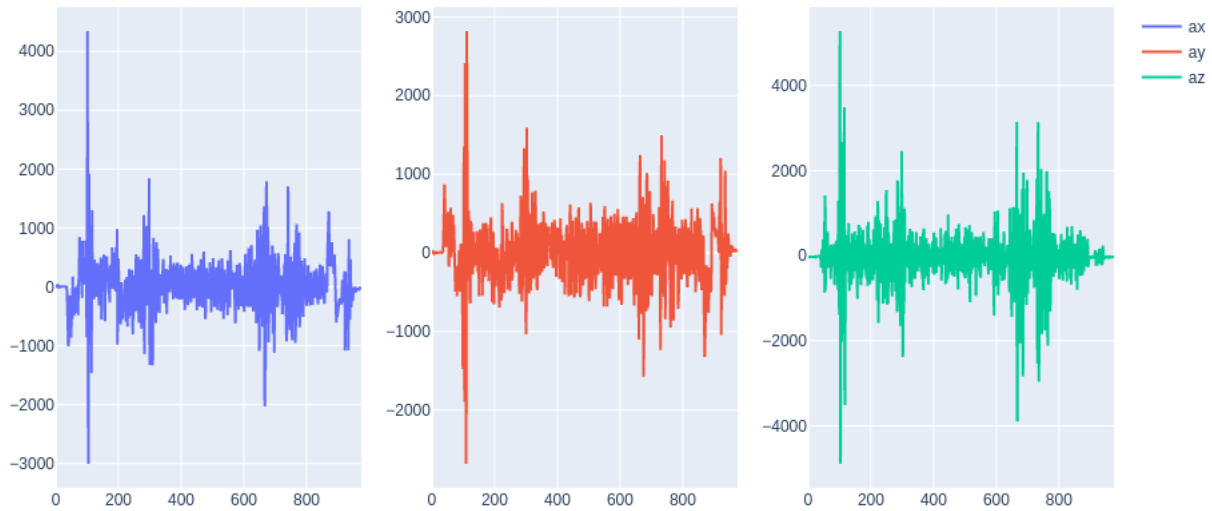


Figure 1: Données accélérométriques enregistrées sur le robot Scout 2.0 avec le dispositif Arduino

de données la plus pratique et efficace possible peut faciliter cette collecte et répondre à notre besoin.

Nous avons donc d'abord utilisé un robot équipé de deux cartes Arduino, mais nous avons pensé qu'il serait plus judicieux de mettre en place une application mobile qui permettrait d'effectuer la même tâche. En effet, l'idée d'utiliser une application mobile est de rendre la collecte de données accessible à toute autre personne munie d'un véhicule. L'utilisation de l'application nécessitera dans un premier temps des restrictions concernant la disposition du téléphone dans le véhicule (position, orientation, support utilisé). Cela permettra de diversifier les données et de les augmenter quantitativement, étant donné qu'il y aura un nombre important de collecteurs de données.

3.2 Développement

L'application actuelle sur laquelle nous travaillons a été codée en Kotlin et en Java à l'aide d'Android Studio qui est un environnement de développement pour développer des applications mobiles Android (Figure 2). La présence de nombreux tutoriels mis en place par Android Studio, ainsi que de forums nous a permis d'avancer sur le développement du code.

3.3 État de l'application

À l'heure actuelle, l'application est capable de collecter les données accélérométriques avec le *timestamp*. Cependant, les enregistrements sont écrits dans des fichiers réservés à l'application, c'est à dire qui ne sont disponibles qu'en passant par Android Studio. Ce défaut rend donc la collecte de données non accessible pour tous les utilisateurs. Nous avons en revanche pu créer une seconde application avec Android Studio qui peut enregistrer les données GPS du téléphone, et cette fois-ci dans un fichier qui est accessible directement depuis le téléphone. La tâche en cours est donc d'implémenter cette particularité d'enregistrement de fichier dans l'application principale.

3.4 Rôle futur envisagé

Une fois que les problèmes d'enregistrement de fichiers seront résolus, la prochaine étape de l'application sera de la relier au modèle d'apprentissage que nous aurons mis en place, et de permettre ainsi de détecter et d'identifier le type de dégradation au moment même où la voiture se déplacera et rencontrera les détériorations de la route.

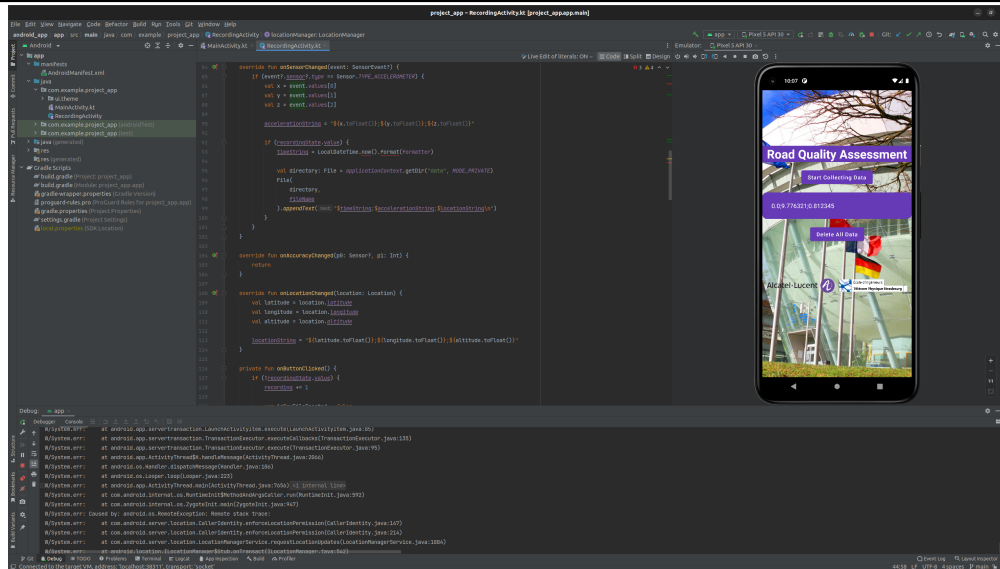


Figure 2: Capture d'écran d'Android Studio lors du développement de l'application

4 Collecte de Données

Dans notre projet de détection de détériorations de la route, la collecte de données est une étape fondamentale pour garantir la qualité et la fiabilité de notre modèle. Nous avons donc réalisé une nouvelle collecte de données en utilisant le robot, le module Arduino et notre application de collecte de données. Cette fois-ci, notre objectif était de collecter un maximum de données de différents types de détériorations de la route, afin d'en avoir suffisamment pour entraîner notre modèle d'apprentissage automatique.

Nous avons collecté environ quatre-vingt données (Figure 3) de différentes détériorations de la route, notamment des dos d'âne, des failles, des fissures et des affaissements. Pour chaque type de détérioration, nous avons collecté en moyenne dix données par dégradation pour garantir la diversité de nos données.

Cependant, nous avons rencontré un problème avec notre carte arduino lors de la collecte de données. En effet, celle-ci n'a enregistré qu'une dizaine de données, sans que nous puissions identifier la cause de cette erreur. Bien que ce soit une déception, nous avons malgré tout pu collecter suffisamment de données avec l'application Android pour poursuivre notre projet de détection de détériorations de la route.

Dans l'ensemble, la collecte de données est une étape cruciale de notre projet, car la qualité des données collectées aura un impact direct sur la qualité et l'efficacité de notre modèle.

5 Augmentation des Données

Le jeu de données obtenu suite à la séance d'enregistrement décrite dans la Section 4 est constitué de quatre-vingt-deux enregistrements résumés dans la Table 1. Tel quel, ce jeu de données est insuffisant pour procéder directement à l'entraînement d'un modèle d'intelligence artificielle. En effet, pour entraîner les modèles les plus simples, il faut être en possession de suffisamment de données pour créer des jeux d'entraînement, de validation et de test. Les méthodes plus avancées relevant de l'apprentissage profond nécessitent encore plus de données d'entraînement pour être utilisées dans de bonnes conditions et offrir des résultats corrects.

Plutôt que de continuer à collecter des données, une démarche longue et peu instructive, nous avons choisi de procéder à de l'augmentation de données.

Les méthodes de *data augmentation* consistent à créer de nouvelles données en altérant légèrement les données déjà présentes dans le *dataset*. Lorsqu'on travaille sur des images, de nouvelles images peuvent être créées en effectuant des rotations, en rognant l'image ou en modifiant certaines propriétés (contraste, luminosité...).

Dans le cadre de ce projet, les données sont sous la forme de séquences numériques selon trois axes. Nous

First Data Collection												
1												2023/04/14
2	Date:											Pole API, Car Park
3	Location:											Phone application
4	Recording device:											-3 ms
5	Acquisition delay:											-333 Hz
6	Acquisition frequency:											Agile-X Scout 2.0
7	Vehicle:											
8	Position of the recording device on the vehicle:											Front left, laying flat, screen on top, top of the phone toward the front of the robot
ID	Time	Obstacle	Obstacle Description	Obstacle Dimension	Location Index	Location Description	File	Direction/Angle	Heading Toward	Speed	Vehicle Side	Vehicle Reaction
10	1	Speed Bump	Short speed bump		1	Speed bump close to the school back entrance	DATA1					
11	2	Speed Bump	Short speed bump		1	Speed bump close to the school back entrance	DATA2					
12	3	Speed Bump	Short speed bump		1	Speed bump close to the school back entrance	DATA3					
13	4	Speed Bump	Short speed bump		1	Speed bump close to the school back entrance	DATA4					
14	5	Speed Bump	Short speed bump		1	Speed bump close to the school back entrance	DATA5					
15	6	Speed Bump	Short speed bump		1	Speed bump close to the school back entrance	DATA6					
16	7	Speed Bump	Short speed bump		1	Speed bump close to the school back entrance	DATA7					
17	8	Speed Bump	Short speed bump		1	Speed bump close to the school back entrance	DATA8					
18	9	Speed Bump	Short speed bump		1	Speed bump close to the school back entrance	DATA9					
19	10	Speed Bump	Short speed bump		1	Speed bump close to the school back entrance	DATA10					
20	11	Crack	Random crack		1	Next to the speed bump mentioned above	DATA11					
21	12	Crack	Random crack		1	Next to the speed bump mentioned above	DATA12					
22	13	Crack	Random crack		1	Next to the speed bump mentioned above	DATA13					
23	14	Crack	Random crack		1	Next to the speed bump mentioned above	DATA14					
24	15	Crack	Random crack		1	Next to the speed bump mentioned above	DATA15					
25	16	Crack	Random crack		1	Next to the speed bump mentioned above	DATA16					
26	17	Crack	Random crack		1	Next to the speed bump mentioned above	DATA17					
27	18	Crack	Random crack		1	Next to the speed bump mentioned above	DATA18					
28	19	Crack	Random crack		1	Next to the speed bump mentioned above	DATA19					
29	20	Crack	Random crack		1	Next to the speed bump mentioned above	DATA20					
30	21	Hole	Round hole		2	At the beginning of the path leading to the RU	DATA21					
31	22	Hole	Round hole		2	At the beginning of the path leading to the RU	DATA22					
32	23	Hole	Round hole		2	At the beginning of the path leading to the RU	DATA23					
33	24	Hole	Round hole		2	At the beginning of the path leading to the RU	DATA24					
34	25	Hole	Round hole		2	At the beginning of the path leading to the RU	DATA25					
35	26	Hole	Round hole		2	At the beginning of the path leading to the RU	DATA26					
36	27	Hole	Round hole		2	At the beginning of the path leading to the RU	DATA27					
37	28	Hole	Round hole		2	At the beginning of the path leading to the RU	DATA28					
38	29	Hole	Round hole		2	At the beginning of the path leading to the RU	DATA29					
39	30	Hole	Round hole		2	At the beginning of the path leading to the RU	DATA30					
40	31	Hole	Round hole		2	At the beginning of the path leading to the RU	DATA31					
41	32	Hole	Round hole		2	At the beginning of the path leading to the RU	DATA32					
42	33	Bump	Tree bump		3	At the beginning of the path leading to the RU	DATA33					
43	34	Bump	Tree bump		3	At the beginning of the path leading to the RU	DATA34					
44	35	Bump	Tree bump		3	At the beginning of the path leading to the RU	DATA35					
45	36	Bump	Tree bump		3	At the beginning of the path leading to the RU	DATA36					
46	37	Bump	Tree bump		3	At the beginning of the path leading to the RU	DATA37					
47	38	Bump	Tree bump		3	At the beginning of the path leading to the RU	DATA38					
48	39	Bump	Tree bump		3	At the beginning of the path leading to the RU	DATA39					
49	40	Bump	Tree bump		3	At the beginning of the path leading to the RU	DATA40					
50	41	Bump	Tree bump		3	At the beginning of the path leading to the RU	DATA41					
51	42	Bump	Tree bump		3	At the beginning of the path leading to the RU	DATA42					
52	43	Crack	Little crack		4	At the beginning of the path leading to the RU	DATA43					
53	44	Crack	Little crack		4	At the beginning of the path leading to the RU	DATA44					
54	45	Crack	Little crack		4	At the beginning of the path leading to the RU	DATA45					
55	46	Crack	Little crack		4	At the beginning of the path leading to the RU	DATA46					
56	47	Crack	Little crack		4	At the beginning of the path leading to the RU	DATA47					
57	48	Crack	Little crack		4	At the beginning of the path leading to the RU	DATA48					
58	49	Crack	Little crack		4	At the beginning of the path leading to the RU	DATA49					
59	50	Crack	Little crack		4	At the beginning of the path leading to the RU	DATA50					
60	51	Crack	Little crack		4	At the beginning of the path leading to the RU	DATA51					
61	52	Crack	Little crack		4	At the beginning of the path leading to the RU	DATA52					
62	53	Speed Bump	Short speed bump		5	At the beginning of the path leading to the CRL	DATA53					
63	54	Speed Bump	Short speed bump		5	At the beginning of the path leading to the CRL	DATA54					
64	55	Speed Bump	Short speed bump		5	At the beginning of the path leading to the CRL	DATA55					
65	56	Speed Bump	Short speed bump		5	At the beginning of the path leading to the CRL	DATA56					
66	57	Speed Bump	Short speed bump		5	At the beginning of the path leading to the CRL	DATA57					
67	58	Speed Bump	Short speed bump		5	At the beginning of the path leading to the CRL	DATA58					
68	59	Speed Bump	Short speed bump		5	At the beginning of the path leading to the CRL	DATA59					
69	60	Speed Bump	Short speed bump		5	At the beginning of the path leading to the CRL	DATA60					
70	61	Speed Bump	Short speed bump		5	At the beginning of the path leading to the CRL	DATA61					
71	62	Speed Bump	Short speed bump		5	At the beginning of the path leading to the CRL	DATA62					
72	63	Bumpy tiles	Old path made of garden tiles		6	At the beginning of the path leading to the RU	DATA63					
73	64	Bumpy tiles	Old path made of garden tiles		6	At the beginning of the path leading to the RU	DATA64					
74	65	Bumpy tiles	Old path made of garden tiles		6	At the beginning of the path leading to the RU	DATA65					
75	66	Bumpy tiles	Old path made of garden tiles		6	At the beginning of the path leading to the RU	DATA66					
76	67	Hole + Crack + Bumps	Round hole, straight crack (jointure between to asphalt surf		7	Beginning of the path leading to the RU	DATA67					
77	68	Hole + Crack + Bumps	Round hole, straight crack (jointure between to asphalt surf		7	Beginning of the path leading to the RU	DATA68					
78	69	Hole + Crack + Bumps	Round hole, straight crack (jointure between to asphalt surf		7	Beginning of the path leading to the RU	DATA69					
79	70	Hole + Crack + Bumps	Round hole, straight crack (jointure between to asphalt surf		7	Beginning of the path leading to the RU	DATA70					
80	71	Hole + Crack + Bumps	Round hole, straight crack (jointure between to asphalt surf		7	Beginning of the path leading to the RU	DATA71					
81	72	Hole + Crack + Bumps	Round hole, straight crack (jointure between to asphalt surf		7	Beginning of the path leading to the RU	DATA72					
82	73	Hole + Crack + Bumps	Round hole, straight crack (jointure between to asphalt surf		7	Beginning of the path leading to the RU	DATA73					
83	74	Hole + Crack + Bumps	Round hole, straight crack (jointure between to asphalt surf		7	Beginning of the path leading to the RU	DATA74					
84	75	Hole + Crack + Bumps	Round hole, straight crack (jointure between to asphalt surf		7	Beginning of the path leading to the RU	DATA75					
85	76	Hole + Crack + Bumps	Round hole, straight crack (jointure between to asphalt surf		7	Beginning of the path leading to the RU	DATA76					
86	77	Speed Bump + Hole	Little speed bump, little hole		8	From parking to the RU	DATA77					
87	78	Speed Bump + Hole	Little speed bump, little hole		8	From parking to the RU	DATA78					
88	79	Speed Bump + Hole	Little speed bump, little hole		8	From parking to the RU	DATA79					
89	80	Speed Bump + Hole	Little speed bump, little hole		8	From parking to the RU	DATA80					
90	81	Speed Bump + Hole	Little speed bump, little hole		8	From parking to the RU	DATA81					
91	82	Speed Bump + Hole	Little speed bump, little hole		8	From parking to the RU	DATA82					

Figure 3: Tableau récapitulatif de la collecte de données

Dégradation	Nombre d'enregistrements
Ralentisseur	20
Racines	10
Fissure	20
Trou	12
Dalles	4
Parcours	16

Table 1: Table résumant la collecte de données

Test

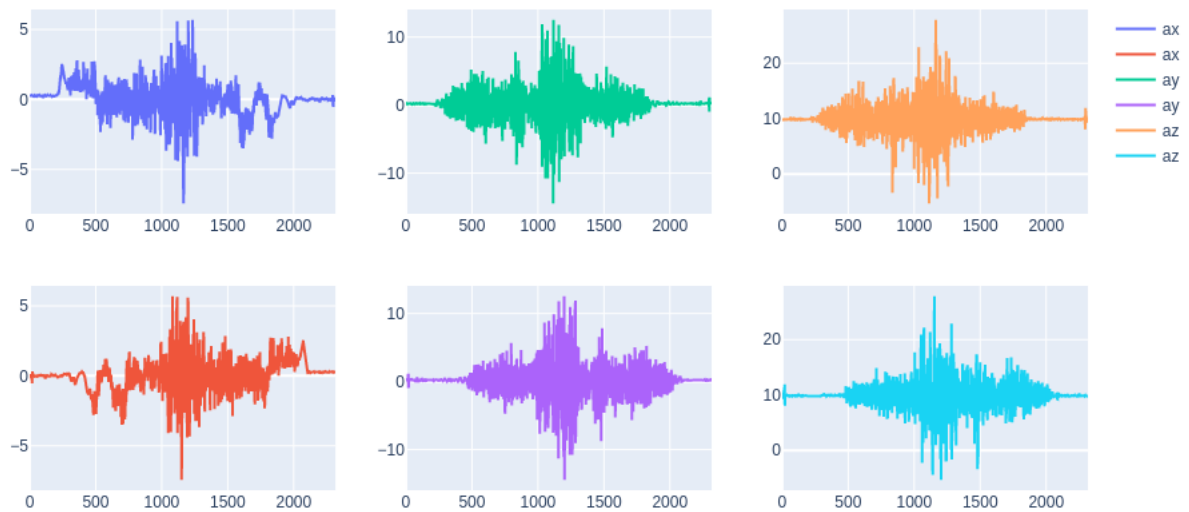


Figure 4: Comparaison d'un enregistrement et de sa version modifiée (renversement du temps)

avons choisi cinq méthodes pour augmenter la quantité de données:

- Renverser la séquence dans le temps
- Inverser l'axe correspondant à la rotation du robot
- Ajout de bruit
- Ajout d'un signal porteur
- Découpage des séquences

5.1 Renverser la séquence dans le Temps

Etant donnée que le robot est très réactif (l'accélération et la décélération sont similaires), nous pouvons considérer que la séquence renversée dans le temps reste représentative. (Figure 4)

Test

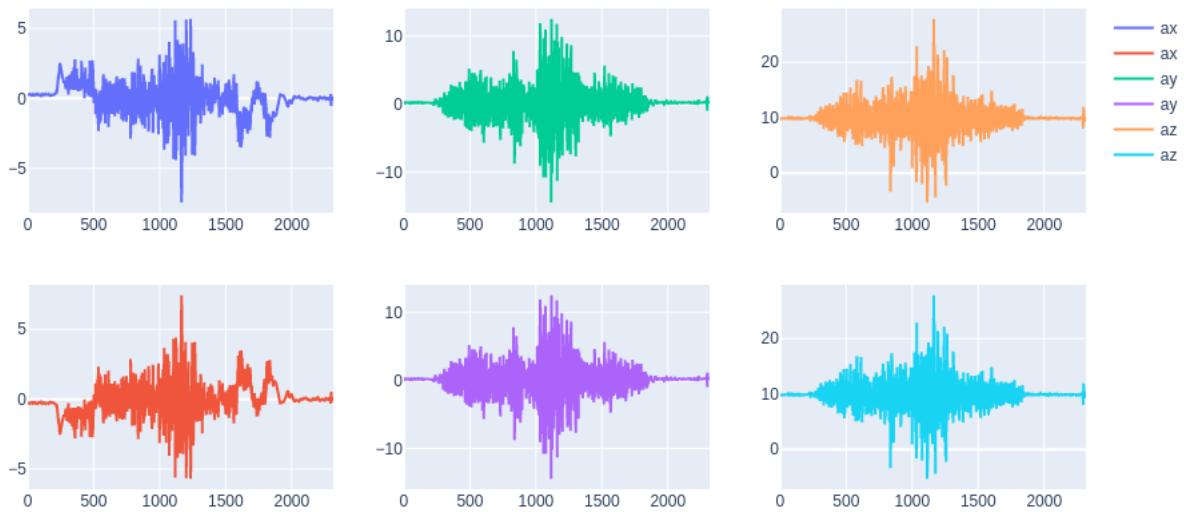


Figure 5: Comparaison d'un enregistrement et de sa version modifiée (renversement d'un axe horizontal)

5.2 Renverser l'Axe de Rotation

De la même façon, inverser la rotation du robot ne modifie pas l'enregistrement du point de vue des dégradations de la route. Les dégradations ont principalement un effet sur l'axe vertical et n'impactent pas la direction du robot donc nous pouvons prendre l'opposée des valeurs sur l'axe X. (Figure 5)

5.3 Ajout de Bruit

Une méthode classique pour augmenter la quantité de données consiste à ajouter du bruit. Cette méthode s'applique dans une certaine mesure aux données accélérométriques recueillies. Un faible bruit peut simuler un bitume moins lisse mais un trop gros bruit signifierait que la route est dans un mauvais état global (gravier...). (Figure 6)

5.4 Ajout d'un Signal Porteur

Des recherches sur internet [1] nous ont montré qu'il est aussi possible d'ajouter un signal porteur pour simuler un dénivelé supplémentaire. Un enregistrement de passage dans un trou sur un sol horizontal peut devenir un enregistrement de passage dans un trou au cours d'une montée.

5.5 Découpage des Séquences

Une dernière méthode simple de *data augmentation* consiste à découper les séquences dans le temps pour isoler les dégradations lors de parcours ou tout simplement pour obtenir des séquences de longueur différentes avec le passage sur la dégradation à différents moments de l'enregistrement (pas nécessairement au milieu) ce qui pourrait induire un biais d'apprentissage.

Finalement, les méthodes précédentes peuvent être combinées afin de générer de nouvelles séquences artificielles.

Test

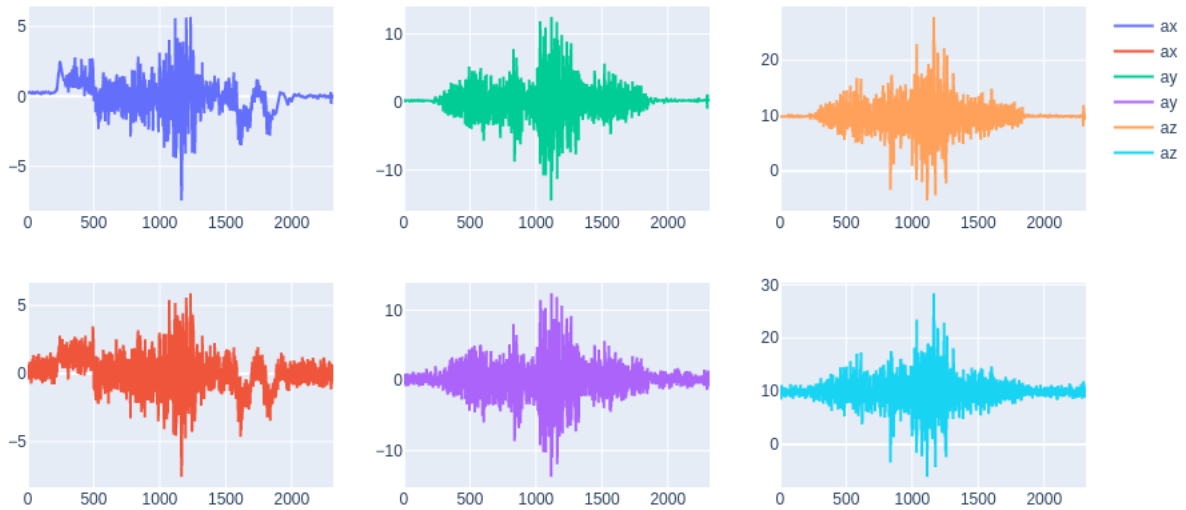


Figure 6: Comparaison d'un enregistrement et de sa version modifiée (ajout de bruit)

6 Intelligence Artificielle

Ayant suffisamment de données pour entraîner un modèle d'intelligence artificielle, nous avons commencé à mettre les données en forme pour faciliter l'apprentissage.

6.1 Création du Jeu de Données

La première étape consiste à créer un véritable jeu de données labélisé en utilisant les enregistrements et les labels recueillis lors des collectes de données. Pour cela nous avons suivi dans les grandes lignes un tutoriel de pré-traitement des données séquentielles pour classification.[3] Nous avons donc créé un grand *dataset* contenant les *inputs* regroupant tous les enregistrements que nous avons stocké sous la forme d'un fichier *csv* et un fichier de labels avec l'extension *.labels* qui regroupe les fichiers et leurs labels.

6.2 Premier Modèle

La mise en place d'un premier modèle en suivant un autre tutoriel s'est avérée infructueuse. Adapter le programme présenté dans le tutoriel ne permettait pas d'utiliser le modèle car il nécessite des séquences de longueurs égales, ce qui n'est pas le cas dans nos données.[2] Il faut donc ajouter une étape supplémentaire de pré-traitement pour obtenir des séquences de même longueur (par découpage ou ajout de *padding*).

6.3 Méthodes Envisagées

Au cours des prochains sprints, nous envisageons de poursuivre le développement de l'intelligence artificielle.

Nous commencerons par des modèles simples comme le modèle *k-Nearest Neighbors* rencontré à plusieurs reprises lors de nos recherches. Nous pensons aussi essayer des méthodes plus avancées basées sur des approches d'apprentissage profond comme par exemple les *Long-Short Term Memory* (LSTM) qui sont très utilisés pour la classification de séquences numériques. Finalement, nous essaierons d'entraîner un modèle de détection d'anomalies. Bien qu'il n'effectue pas une classification des dégradations, ce modèle a l'avantage de détecter

tout type d'anomalie et serait donc capable d'identifier n'importe quel type de dégradation de la chaussée (y compris les dégradations qui ne sont pas représentées dans nos données enregistrées).

Suivant les ressources qui seront à notre disposition, nous devrons éventuellement effectuer l'entraînement des approches *deep learning* par affinement d'un modèle déjà existant pour une tâche similaire (comme la détection de chute) voire plus générale et nous espérons pouvoir mettre en place un programme capable d'effectuer un apprentissage continu grâce à de nouvelles données enregistrées au fur et à mesure.

7 Conclusion

Le gain d'expérience suite à la première collecte de données et l'embrion de l'application Android nous ont permis de construire un jeu de données plus conséquent et de meilleure qualité. Ce *dataset* contient plus de quatre-vingt enregistrements de durées variables correspondant au passage du robot Agile-X Scout 2.0 sur un ou plusieurs obstacles en faisant varier la vitesse.

La taille de ce jeu de données restant insuffisante pour entraîner convenablement un modèle d'IA (jeu d'entraînement, de validation et de test), nous avons dû augmenter artificiellement le nombre d'enregistrements en inversant la séquence dans le temps, inversant l'axe correspondant à la rotation du robot, en ajoutant du bruit ou un signal porteur et en combinant ces méthodes.

Une grande phase de pré-traitement a été réalisée avant de commencer à programmer une intelligence artificielle. La création du jeu de données contenant les séquences numériques a été automatisée et les données ont été labellisées manuellement selon les informations recueillies dans un tableur lors des enregistrements. Cependant, une nouvelle phase de pré-traitement est nécessaire pour uniformiser la longueur des séquences ce qui permettra de construire un premier classifieur.

Plusieurs approches pour classifier les séquences seront testées par la suite: approches classiques (*k-Nearest Neighbors*), approches par apprentissage profond (LSTM) et détection d'anomalies. Les différents modèles devront être évalués méthodiquement afin de sélectionner le modèle le plus adapté pour répondre à la problématique de ce projet. Le modèle pourra alors être déployé et amélioré au cours du temps si nous parvenons à implémenter une méthode d'amélioration continue.

Bibliographie

- [1] Terry T. Um, Franz M. J. Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ICMI 2017, pages 216–220, New York, NY, USA, 2017. ACM.
- [2] Venelin Valkov. Multivariate time series classification tutorial with lstm in pytorch, pytorch lightning and python.
- [3] Venelin Valkov. Multivariate time series data preprocessing with pandas in python — machine learning tutorial.