



RECHERCHES POUR LE DEVELOPPEMENT

Série Sciences Technologie

N°09-2024

Antananarivo - Madagascar

**Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique**

ISSN 1025 – 3459

RECHERCHES POUR LE DEVELOPPEMENT

Série Sciences Technologie

N°09

2024

**RECHERCHES
POUR
LE DEVELOPPEMENT**

Série

Sciences Technologie

N° 09-2024

Membres du Comité de lecture :

- Pr RAMIARISON Claudine
- Pr RAKOTOARIVELO Marie Laure
- Dr RAHARIJAONA Nivoniaina Fahendrena
- Dr VOLAMPENO Sylviane

Ce numéro a été édité avec le concours de

Université d'Antananarivo

et

Centre d'Information et de Documentation
Scientifique et Technique

Toute correspondance concernant les publications
RECHERCHES POUR LE DEVELOPPEMENT
doit être adressée au :

Centre d'Information et de Documentation
Scientifique et Technique
BP 6224 – Email : edition.cidst@gmail.com
Antananarivo - Madagascar

ISSN 1025-3459

SOMMAIRE

Guidage d'un robot mobile par detection et suivi d'objet

RAKOTOMANANJARA Jean Tony, RAHAJANIAINA Andriamasinoro,
RATIARISON Adolphe Andriamanga

06

Modélisation d'une carte de bruit des avions par la méthode d'apprentissage automatique

RAFANAMBINANTSOA Valohery Clermont, RAMAHERY Yvan Emiliau,
MAHATODY Thomas et RAZAFINDRAKOTO Raft

21

GUIDAGE D'UN ROBOT MOBILE PAR DETECTION ET SUIVI D'OBJETS

par

RAKOTOMANANJARA Jean Tony⁽¹⁾, RAHAJANIAINA Andriamasinoro ⁽¹⁾,
RATJARISON Adolphe Andriamanga⁽²⁾

⁽¹⁾ Mention Mathématiques, Informatiques et Applications, Université de Toamasina

⁽²⁾ Mention Physique et Applications, Université d'Antananarivo

RESUME

Dans le cadre d'une étude visant à améliorer le guidage des robots mobiles, cet article explore l'intégration de technologies telles que MobileNetV3 version Small, Single Shot multibox Detector (SSD) version Lite, le filtre de Kalman et les capteurs à ultrasons. La méthodologie combine l'utilisation de MobileNetV3 version Small et Single Shot multibox Detector (SSD) version Lite pour la détection et la classification des objets, avec l'algorithme Kernelized Correlation Filter (KCF) pour le suivi et l'Unscented Kalman Filter (UKF) pour prédire les trajectoires. Un capteur à ultrasons a été utilisé pour mesurer la distance des objets par rapport au robot, permettant d'améliorer le contrôle de ce dernier. Les résultats ont montré une amélioration de la précision de la détection d'objets en temps réel, même dans des environnements encombrés, avec un score F1 de 75%. L'interprétation des résultats souligne les avancées technologiques tout en mettant en évidence les défis persistants de la détection dans des environnements complexes, soulignant ainsi la nécessité de poursuivre la recherche pour une adoption plus large de ces technologies innovantes.

Mots clés : Robot mobile, Détection d'objet, Suivi d'objet, Guidage robotique, MobileNet, Single Shot multibox Detector (SSD), OpenCV, Filtre de Kalman.

ABSTRACT

In the context of a study aimed at improving mobile robot guidance, this article explores the integration of technologies such as MobileNetV3 Small version, Single Shot multibox Detector (SSD) Lite version, Kalman filter, and ultrasonic sensors. The methodology combines the use of MobileNetV3 Small version and Single Shot multibox Detector (SSD) Lite version for object detection and classification, with the Kernelized Correlation Filter (KCF) algorithm for tracking and Unscented Kalman Filter (UKF) for trajectory prediction. An ultrasonic sensor was used to measure object distances from the robot, enhancing robot control. Results demonstrated improved real-time object detection accuracy, even in cluttered environments,

achieving an F1 score of 75%. Interpretation of the findings highlights technological advancements while underscoring persistent challenges in detection within complex environments, thereby emphasizing the need for continued research to facilitate broader adoption of these innovative technologies.

Keywords : Mobile robot, object detection, object tracking, robotic guidance, mobileNet, Single Shot Multibox Detector (SSD), OpenCV, Kalman filter.

INTRODUCTION

Au fil des années, l'évolution des robots mobiles a connu une progression spectaculaire, passant de simples exécutants de tâche en machines autonomes dotées de capacités avancées telles que la détection d'obstacles et la navigation autonome grâce à l'intelligence artificielle et à l'apprentissage automatique. L'intégration de capteurs sophistiqués et de réseaux neuronaux a permis des avancées significatives, transformant l'interaction avec les robots, que ce soit pour des tâches quotidiennes simples ou des missions complexes de recherche et de sauvetage. Toutefois, malgré ces progrès, le guidage des robots mobiles reste un domaine confronté à de nombreux défis, notamment la capacité à suivre efficacement un objet dans des environnements complexes et encombrés, la gestion optimale des ressources des algorithmes de guidage, et les contraintes de calcul imposées par le traitement d'image en temps réel. Le guidage efficace des robots mobiles est crucial dans diverses applications industrielles, logistiques et médicales, nécessitant la conception et la mise en œuvre de systèmes adaptés aux attentes spécifiques et au domaine d'application (Siegwart *et al.*, 2011).

Les robots peuvent utiliser des algorithmes de détection pour identifier et suivre des objets spécifiques dans leur champ de vision, ce qui est utile pour des tâches telles que la manipulation d'objets, le tri, et la surveillance. Plusieurs méthodes existent pour réaliser ces tâches, mais la présente étude sera axée sur les techniques les plus récentes, comme celles présentées par (Mekhennene et Sekhri, 2022) et (Shi *et al.*, 2019).

Mekhennene A. et Sekhri M. ont utilisé la méthode YOLO (Farhadi *et al.*, 2016), et le filtre de Kalman pour la détection et le suivi d'objets (MATLAB & Simulink, 2024). Ils ont choisi YOLO pour sa capacité à utiliser des réseaux de neurones afin de fournir une détection d'objets en temps réel, permettant d'identifier les objets présents dans une image ou une séquence d'images. Une fois les objets détectés, le filtre de Kalman est utilisé pour estimer leur trajectoire et leur position future. La combinaison de ces deux méthodes rend le système entièrement automatique et capable de suivre chaque objet mobile tout en cartographiant sa trajectoire. Cette

approche présente de nombreux avantages : YOLO permet une détection rapide et en temps réel, tandis que le filtre de Kalman assure un suivi précis des objets en mouvement, ce qui est particulièrement utile dans des domaines tels que la surveillance et la navigation. Cependant, le système présente des inconvénients. La précision du suivi dépend fortement de la qualité de la détection initiale par YOLO, et des erreurs de détection peuvent entraîner des erreurs de suivi. Le système peut également rencontrer des difficultés dans des situations complexes, comme des mouvements rapides ou des environnements encombrés, et la combinaison des deux méthodes exige une puissance de calcul considérable, ce qui peut ralentir l'exécution de l'algorithme. Enfin, le suivi des objets sur de longues périodes peut poser problème, car le filtre de Kalman n'est pas toujours robuste face à des objets très dynamiques. En résumé, la combinaison de YOLO et du filtre de Kalman offre des avantages en termes de détection en temps réel et de suivi automatique d'objets, mais des défis subsistent concernant la précision de la détection et la gestion des situations complexes.

Tenant compte de tout ceci, la problématique posée se rapporte à comment améliorer le guidage des robots mobiles, notamment leur contrôle ? Dans le vaste domaine de la robotique mobile, la détection et le suivi d'objets jouent un rôle crucial pour permettre aux robots de fonctionner de manière autonome et efficace dans divers domaines tels que la logistique industrielle et l'assistance personnelle. L'étude vise à combiner des méthodes pour créer un système de guidage innovant pour les robots mobiles, en mettant l'accent sur la détection et le suivi d'objets, avec pour objectif une combinaison optimale de rapidité, de précision et de capacité à prévoir les mouvements futurs.

MATERIELS ET METHODES

Environnement de développement

Pour ce projet, un environnement basé sur Ubuntu Mate 22.04 a été développé. Le ROS (Robot Operating System) a été installé pour la gestion des capteurs et des communications du robot. Python a été utilisé pour le langage de programmation principal qui offre, par ailleurs, une flexibilité et une facilité d'intégration avec les bibliothèques de vision par ordinateur.

Robot Operating System

Le Robot Operating System (ROS) est un ensemble d'outils et de bibliothèques open source conçu pour le développement d'applications robotiques. Dans la mesure où il n'est pas un système d'exploitation en soi, ROS nécessite un système d'exploitation pour fonctionner, et Ubuntu Mate 22.04 a été choisi avec la version

ROS Humble pour le projet robotique, car elle est compatible avec le système. L'un des principaux avantages de ROS est sa compatibilité avec plusieurs langages de programmation tels que Python, Java et C++. Son principe fondamental est de permettre l'exécution simultanée de nombreux processus, avec la capacité d'échanger des informations de manière synchrone ou asynchrone.

Matériels utilisés

Pour mener à bien le projet, les équipements suivants ont été utilisés :

- 1 Raspberry Pi 4
- 1 Pi Camera
- 1 clé USB 32Go
- 1 servomoteur
- 1 capteur à ultrason
- 1 contrôleur de moteur
- 1 Châssis robotique
- 4 roues et 4 moteurs
- 1 BMS (Battery Management System)
- 1 module d'abaisseur de tension 5V
- 2 batteries 7,4V (3,7V x2)
- Des jumpers

Pour la détection, l'utilisation de MobileNet et SSD (Single Shot Detector) ont été adoptés, en particulier la version MobileNetV3-SSDLite Small (Xiong et *al.*, 2021), qui offre un équilibre idéal entre précision et rapidité. Une fois les objets détectés, l'algorithme KCF (Kernelized Correlation Filter) a été utilisé pour le suivi, car réputé pour sa rapidité et sa robustesse. Pour anticiper la position future des objets dans un environnement dynamique, UKF (Unscented Kalman Filter) a été appliqué ; il s'agit d'un outil puissant pour prédire les trajectoires (Julier et Uhlmann, 1997). Pour détecter les obstacles proches, un capteur à ultrasons a été intégré, ce qui a permis au robot de s'arrêter ou de changer de direction en cas de détection.

Détection d'objet

Le MobileNetV3 SSD Lite Small est une combinaison de l'architecture MobileNetV3, conçue pour des applications mobiles et embarquées, et du SSD Lite, une version allégée du Single Shot Detector adaptée aux contraintes des appareils mobiles.

MobileNetV3

MobileNetV3 combine des techniques avancées comme les blocs MobileNetV2 (Sandler et *al.*, 2018), les couches squeeze-and-excitation (Hu et *al.*, 2018), et les connexions de skip pour optimiser l'efficacité des calculs et la performance. Les caractéristiques principales de MobileNetV3 incluent l'utilisation de blocs de convolution séparables profonds, qui divisent les convolutions profondes et ponctuelles pour réduire le coût de calcul, et les blocs inversés résiduels avec connexions linéaires, qui augmentent l'efficacité des couches convolutives en diminuant les opérations. De plus, les modules squeeze-and-excitation ajoutent des poids adaptatifs aux canaux convolutifs pour améliorer la représentation des caractéristiques importantes, tandis que l'activation H-Swish, plus efficace et rapide que ReLU, optimise encore davantage les performances du réseau.

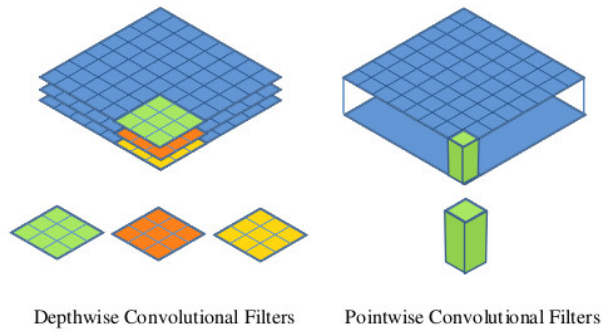


Figure 1 : Convolution séparable profonde

SSD Lite

SSD Lite est une version simplifiée de SSD, adaptée pour des appareils à faible puissance. C'est un modèle de détection d'objets qui permet d'identifier la présence et la localisation d'objets multiples dans une image, et il est conçu pour accomplir ces tâches en une seule passe ; contrairement à certaines méthodes qui utilisent plusieurs étapes pour détecter et classifier les objets (Liu, et *al.*, 2016).

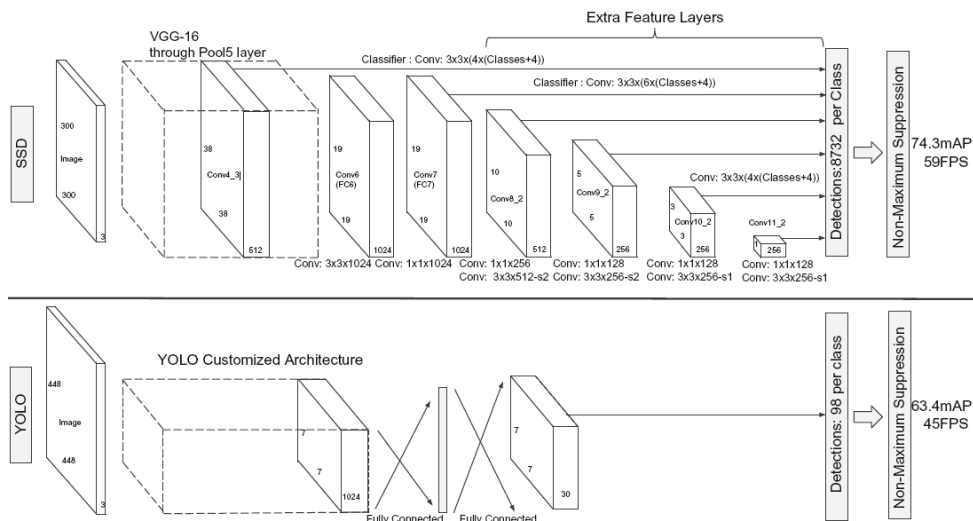


Figure 2 : Comparaison entre SSD et YOLO

Les principales caractéristiques de SSD Lite comprennent la détection d'objets multicouches qui utilise plusieurs couches pour détecter les objets à différentes échelles et tailles, et la prédiction directe qui permet de prédire directement les classes d'objets et les boîtes de délimitation sans étape de proposition préalable. De plus, SSD Lite est optimisé pour l'exécution sur mobile, réduisant les paramètres et les opérations de convolution pour une meilleure efficacité.

Spécifications du MobileNetV3 SSD Lite Small

Le MobileNetV3 SSD Lite Small se caractérise par une taille d'entrée typique de 320 x 320 pixels, ajustable selon les besoins de l'application, et utilise MobileNetV3-Small comme backbone pour extraire les caractéristiques de l'image. Il intègre des convolutions séparables profondes et ponctuelles pour réduire les calculs tout en maintenant la performance, et ajoute des couches de détection SSD Lite après l'extraction des caractéristiques pour prédire les boîtes de délimitation et les classes d'objets. De plus, il applique une suppression des non-maximum (NMS) pour filtrer les détections redondantes, et est optimisé pour l'efficacité mobile, étant rapide et léger, idéal pour les appareils à faible puissance et les applications en temps réel.

Intégration du capteur à ultrason

Un capteur à ultrason est un dispositif électronique conçu pour détecter la distance entre lui-même et un objet en utilisant des ondes sonores ultrasonores. Ce type de capteur émet des signaux ultrasonores et mesure le temps nécessaire pour que ces signaux rebondissent sur l'objet cible et reviennent au capteur. En se basant sur la durée du trajet de ces ondes sonores, le capteur peut calculer la distance entre

lui-même et l'objet, fournissant ainsi des informations précises sur la proximité spatiale (Medak et *al.*, 2021). Le but de l'utilisation d'un capteur à ultrason ici est de détecter les obstacles devant le robot pour éviter la collision. La distance entre le capteur et l'objet se calcule :

$$\text{Distance} = (\text{Temps écoulé} * \text{Vitesse du son}) / 2 \quad (1)$$

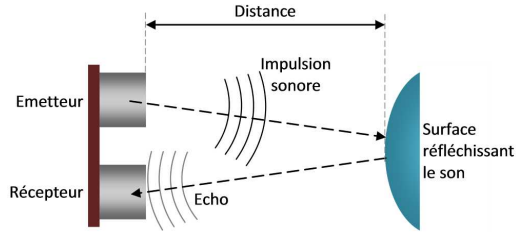


Figure 3 : Principe de fonctionnement de l'Ultrason

Suivi d'objet

Kernelized Correlation Filter

L'algorithme KCF (Kernelized Correlation Filter) utilise des techniques modernes et des filtres spéciaux pour suivre les objets de manière efficace et intelligente. Ces filtres sont spécialement entraînés pour distinguer un objet cible de son arrière-plan. L'algorithme commence par utiliser les données d'une image initiale où l'objet est déjà marqué ou identifié, puis il continue à suivre cet objet à travers les images successives. L'aspect clé de KCF est son utilisation d'une méthode "Kernelized", qui transforme les données d'entrée dans un espace de caractéristiques de dimension supérieure. Cette transformation permet de mieux saisir les relations complexes entre l'objet cible et son environnement, améliorant ainsi significativement la précision du suivi. En outre, KCF est particulièrement efficace en termes de calcul. Il utilise une méthode de résolution rapide pour l'entraînement et la détection des objets, en exploitant les caractéristiques répétitives des données, ce qui diminue le temps requis pour le traitement. Cette combinaison d'efficacité et de précision fait de KCF un choix populaire pour le suivi d'objets en temps réel dans les applications de vision par ordinateur (Viola et Jones, 2001).

Filtre de KALMAN

Le filtre de Kalman est un estimateur récursif qui fait des prédictions sur l'état futur d'un système dynamique en utilisant une combinaison des estimations précédentes et des nouvelles mesures de données, tout en minimisant l'erreur quadratique moyenne. Le principe clé derrière le filtre de Kalman est de fournir une manière efficace de combiner les informations de l'état précédent avec les nouvelles

informations obtenues des mesures, en tenant compte des incertitudes associées (Grewal et Andrews, 2014).

Principe de fonctionnement

Le filtre de Kalman, outil essentiel en ingénierie et en traitement du signal, opère en plusieurs phases pour estimer avec précision l'état d'un système. Initialement, il prédit l'état futur du système en utilisant un modèle établi, mais cette prédiction n'est pas fixe. À mesure que de nouvelles données ou mesures deviennent disponibles, le filtre ajuste ses prédictions initiales, intégrant ainsi les incertitudes présentes dans la prédiction initiale et les nouvelles mesures. En parallèle, il évalue l'erreur de covariance, mesurant l'incertitude associée à l'estimation de l'état, ce qui est crucial pour évaluer la fiabilité des prédictions. Le "Gain de Kalman" est un aspect clé du filtre, déterminant l'importance relative des nouvelles mesures par rapport à l'estimation actuelle, équilibrant ainsi la confiance entre le modèle et les données.

Enfin, le filtre met régulièrement à jour à la fois le Gain de Kalman et l'erreur de covariance, affinant ainsi les estimations futures et améliorant la précision du filtre au fil du temps. Grâce à ces étapes, le filtre de Kalman est capable de fournir des estimations très précises, même dans des situations où les données sont bruitées ou incomplètes.

Performance du modèle

D'après les résultats du test de performance fait sur la dataset coco avec 5000 images annotées, le modèle a atteint une précision de 0,66, ce qui signifie que 66% des prédictions positives faites par le modèle étaient correctes. Un rappel de 0,86 a été obtenu, ce qui indique que le modèle a réussi à identifier correctement 86% des observations positives des données. Enfin, le score F1, qui est une moyenne pondérée de la précision et du rappel, est de 0,75. Cela signifie qu'il existe un bon équilibre entre la capacité du modèle à préserver la précision et à maximiser la capacité de rappel.

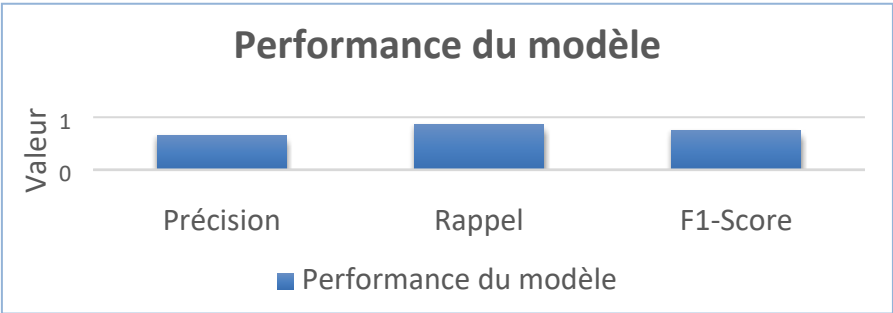


Figure 4 : Performance du modèle

RESULTAT ET DISCUSSION

La Figure 7 illustre l'interface de guidage robotique.

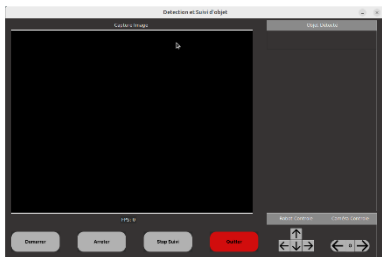


Figure 5 : Interface de Guidage Robotique

Quant aux résultats de la détection et suivi d'objets, les résultats sont les suivants :

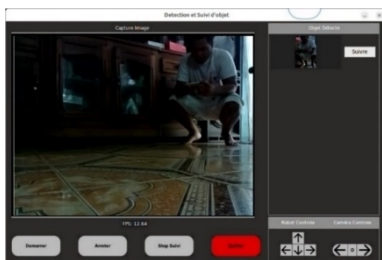


Figure 6 : Détection d'objets 1



Figure 7 : Suivi d'objet 1



Figure 8 : Suivi d'objet 1 terminer



Figure 9 : Suivi d'objet 2

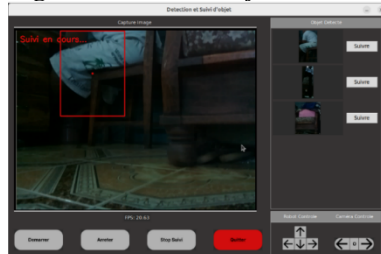


Figure 10 : Suivi d'objet 2.1

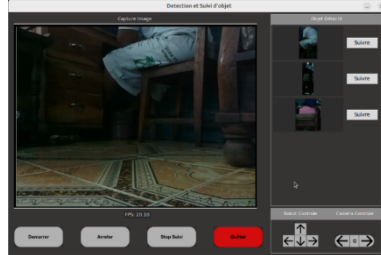


Figure 11 : Suivi d'objet 2 terminer

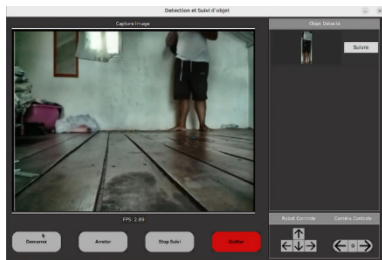


Figure 12 : Détection d'objets 3

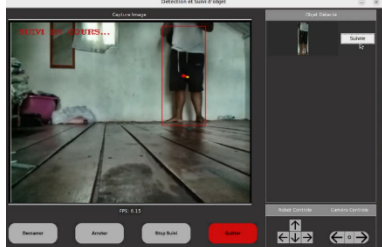


Figure 13 : Suivi d'objet 3



Figure 14 : Suivi d'objet 3.1

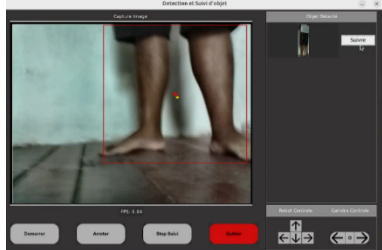


Figure 15 : Suivi d'objet 3.2



Figure 16 : Suivi d'objet 3 terminer



Figure 17 : Suivi d'objet 4

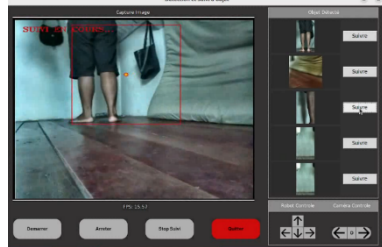


Figure 18 : Suivi d'objet 4.1



Figure 19 : Suivi d'objet 4.2



Figure 20 : Suivi d'objet 4.3



Figure 21 : Suivi d'objet 4.4



Figure 22 : Suivi d'objet 4.5



Figure 23 : Suivi d'objet 4 terminer

Les Figures ci-dessus montrent chaque étape de la détection et suivi d'objet en temps réel. La Figure 6 présente le résultat de la détection d'objets du système où chaque objet détecté est affiché dans le panneau "Objets Détectés" à droite. Ce panneau peut contenir jusqu'à cinq objets détectés simultanément. À côté de chaque image de l'objet détecté se trouve un bouton "Suivre" permettant de lancer le processus de suivi de l'objet.

Les Figures 7, 9, 10, 12, 13, 14, 15, 17, 18, 19, 20, 21 et 22 montrent le processus, en cours, de suivi d'objet. Une boîte englobante est tracée autour de l'objet suivi, ce qui permet de suivre visuellement le succès du suivi. Dans la Figure 8, 11, 16 et 23 la boîte englobante n'est plus visible, ceci indique que le robot a atteint l'objet et que le processus de suivi est terminé.

Dans la phase de test, le système a démontré une capacité efficace à détecter et à suivre les objets en temps réel. Certains problèmes ont été résolus avec succès, notamment la gestion des ressources utilisées par l'algorithme, la gestion du traitement d'image en temps réel, ainsi que l'amélioration de la prise de décision rapide.

Cependant, malgré ces améliorations, certaines limites ont été relevées, il s'agit de difficultés à suivre des objets dans des situations complexes et des environnements très encombrés.

Analyses comparatives

Dans l'étude de (Shi et *al.*, 2019), l'utilisation combinée de SSD et MobileNet pour le suivi de véhicules nécessite des ressources informatiques substantielles, limitant ainsi leur déploiement sur des plateformes embarquées. En comparaison, notre approche avec MobileNetV3-SSD Lite Small est conçue pour être plus légère et mieux adaptée aux contraintes de ressources, offrant une solution efficace et flexible pour les systèmes embarqués, tout en maintenant des performances de détection et de suivi robustes.

Par rapport à l'algorithme de détection d'objets basé sur une version améliorée de YOLOV4 proposé par (Hu et *al.*, 2022), l'approche adoptée se distingue par l'utilisation de MobileNetV3-SSD Lite Small pour la détection, garantissant un équilibre optimal entre rapidité et précision. Pour le suivi, l'algorithme KCF a été utilisé, il est reconnu pour sa rapidité et sa robustesse face aux mouvements rapides. En outre, l'intégration de l'UKF permet d'anticiper les mouvements futurs des objets, améliorant ainsi la capacité du robot à naviguer dans des environnements dynamiques. Le système a été complété par l'intégration de capteurs à ultrasons pour la détection d'obstacles proches, permettant une réactivité accrue du robot en cas de

danger, offrant ainsi une solution intégrée et prédictive adaptée aux applications variées de la logistique industrielle et de l'assistance personnelle.

CONCLUSION ET PERSPECTIVES

Dans le cadre de ce travail, il a été vérifié que l'intégration de technologies telles que MobileNet, SSD, le filtre de Kalman et les capteurs à ultrasons offrent des opportunités sans précédent pour des applications allant de la logistique industrielle à l'assistance personnelle. Toutefois, malgré les progrès significatifs réalisés dans ce domaine, des défis subsistent, notamment en ce qui concerne la précision de la détection, la gestion des environnements complexes et la consommation de ressources computationnelles. Néanmoins, les résultats prometteurs obtenus à ce jour, tels que la précision du modèle, développé, de détection d'objets, indiquent un potentiel considérable pour une adoption plus large de ces technologies. En continuant à innover et à surmonter les défis techniques, il est possible de façonner un avenir où les robots mobiles joueront un rôle essentiel dans la vie quotidienne, offrant des solutions efficaces et intelligentes à diverses problématiques. Cependant, il est important de noter qu'il manque le choix automatique du sujet dans l'approche adoptée, ce qui signifie que le robot n'est pas entièrement autonome et dépend de l'homme pour choisir l'objet à suivre. Pour remédier à cette situation, il est envisagé d'intégrer un système permettant au robot de choisir d'une manière automatique un sujet à suivre suivant des critères bien définis.

REFERENCES BIBLIOGRAPHIQUES

- 1) Grewal, M. S., & Andrews, A. P. (2014). Kalman filtering: Theory and Practice with MATLAB. (J. Wiley, & Sons, Éd.s.)
- 2) Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). Mobilenets : Réseaux de neurones convolutifs efficaces pour les applications de vision mobile. préimpression arXiv arXiv:1704.04861.
- 3) Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-Excitation Networks. Proceedings of the IEEE conference on computer vision and pattern recognition, (pp. 7132-7141).
- 4) Hu, Y., Liu, G., Chen, Z., & Guo, J. (2022). Object detection algorithm for wheeled mobile robot based on an improved yolov4. Applied Sciences, 12(9), 4769.
- 5) Julier, S. J., & Uhlmann, J. K. (1997). New extension of the Kalman filter to nonlinear systems. Signal processing, sensor fusion, and target recognition VI, (pp. 182-193).
- 6) Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd : Détecteur multibox mono-coup. Proceedings of the European Conference on Computer Vision (pp. 21-37). Springer.

- 7) Lixing, L., Xu, W., Xin, Y., Hongjie, L., Jianping, L., & Pengfei, W. (2023). Path planning techniques for mobile robots: Review and prospect. *Expert Systems with Applications*, 227, 120254.
- 8) Medak, D., Posilović, L., Subašić, M., Budimir, M., & Lončarić, S. (2021). Automated defect detection from ultrasonic images using deep learning. (IEEE, Éd.) *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 68(10), 3126-3134.
- 9) Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 4510-4520).
- 10) Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). *Introduction to autonomous mobile robots*. MIT press.
- 11) Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. {*Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, 1, pp. I-I.
- 12) Vu, V., & Bguyen, D. (2021, Décembre 15). Application of MobileNet-SSD Deep Neural Network for Real-Time Object Detection and Lane Tracking on an Autonomous Vehicle. *Advances in Asian Mechanism and Machine Science*, 113. (N. Khang, N. Hoang, & M. Ceccarelli, Éds.) doi:https://doi.org/10.1007/978-3-030-91892-7_53
- 13) Xiong, Y., Liu, H., Gupta, S., Akin, B., Bender, G., Wang, Y., . . . Chen, B. (2021). Mobilenets: Searching for object detection architectures for mobile accelerators. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 3825-3834).

Modélisation d'une carte de bruit des avions par la méthode d'apprentissage automatique

Par

¹RAFANAMBINANTSOA Valohery Clermont ; ²RAMAHERY Yvan Emiliau,;

³Professeur Mahatody Thomas, ⁴Professeur RAZAFINDRAKOTO Raft

¹Ingénieur

²développeur chez DevelopA

³Directeur de l'ENI, rattaché au Laboratoire Informatique et Mathématiques Appliquées au Développement

⁴rattaché au Laboratoire de Recherche Appliquée Multidisciplinaire.

Ecole doctorale en modélisation informatique, Université de Fianarantsoa, Madagascar

RESUME :

La mesure du bruit des avions est très importante pour l'industrie du transport aérien, les riverains et la mairie. Aux Etats Unis d'Amérique, la FAA¹ a son outil de gestion de carte de bruit des avions sous l'appellation AEDT²; en France c'est l'outil Bruitparif qui satisfait la directive européenne d'élaboration de cartes stratégiques de bruit. Pour Madagascar, les réglementations ne sont pas encore claires en dehors du suivi environnemental.

Cet article propose une modélisation de carte de bruit des avions par la méthode d'apprentissage automatique pour le cas de l'aéroport international d'Ivato : un dataset construit avec un système de gestion de base de données relationnel en partant de la détection d'un radar secondaire. Ces données, après avoir suivi un processus de nettoyage de données, vont être entraînées et testées avec les fonctions telles que la naïve bayésienne, l'arbre de décision et la forêt aléatoire. En se basant sur la matrice de confusion comme métrique, nous avons apprécié le résultat fourni par l'arbre de décision avec des métriques « precision (98%), recall (97%) et f1-score (97%) ».

Comme perspectives d'amélioration, l'utilisation de la logique floue serait appréciée pour une classification multiclasse en utilisant des degrés de bruit du nuisible, tolérable, plus tolérable au non nuisible. Nous nous focaliserons aussi sur la

¹ Federal Aviation Administration est l'administration fédérale qui a pour mission de gérer les dossiers en matière d'aviation aux Etats Unis d'Amérique.

² Aeronautical Environmental Design Tool est un outil permettant d'estimer la consommation en carburant, l'émission, le bruit et leurs conséquences sur l'air.

durée d'exposition d'une valeur de bruit pour un temps donné, le temps de passage de l'avion étant un paramètre à considérer.

Mots clés : AEDT, Bruitparif, BADA, carte de bruit, apprentissage automatique, naïve bayésienne, arbre de décision, forêt aléatoire

ABSTRACT :

The measure of the aircrafts noise is very important for the industry of flight transportation, for the residents and for the hall city. In the United States of America, the FAA owns the tool for managing aircrafts noise map called AEDT; in France Bruitparif is the tool which satisfied the European recommendation for noise map strategy elaboration. Here in Madagascar, the specific rules related to aircraft noise are not yet so clear except the environmental purposes.

This document proposes a modelisation of aircraft noise map using machine learning for the case of Ivato international airport: a dataset built with a relational database management system obtained from secondary radar detection. These data, after entering in a cleaning data processing, will be trained, and tested with functions such as naïve Bayes, decision tree and random forest. Based on confusion matrix as metrics, we appreciated the result of decision tree with metrics as 98% for precision, 97% for recall and 97% for f1-score.

For future updates, the use of fuzzy logic would be appreciated like multiclass classification using some degrees of noise values like harmful, tolerable, more tolerable, and not harmful. We will also focus on the exposition duration for an amount of noise for a given time, the time of flying is also a parameter to be considered.

Keywords : AEDT, Bruitparif, BADA, noise map, machine learning, naïve bayes, decision tree, random forest

INTRODUCTION

De nombreuses études ont porté sur les nuisances sonores aux environs de l'aéroport, et ont déclenché des réglementations et des recommandations puisque « *les avions génèrent des bruits par leur moteur mais aussi par la circulation de l'air autour de l'appareil* » (Cidb, 2024). Une autre étude basée sur l'effet de bruit pendant la nuit a aussi vu le jour afin de donner une corrélation entre l'exposition au bruit des avions la nuit et la qualité du sommeil qui dégrade l'état de santé de l'homme (Nassur, 2018). Chaque année, le gestionnaire de l'aéroport international d'Ivato réalise un suivi environnemental de l'exploitation de l'aéroport. Les mesures de bruit ont pour but de caractériser le niveau de bruit ambiant à proximité et dans l'enceinte de l'aéroport durant son exploitation (egis inframad, 2019).

MATERIELS ET METHODES

Travaux similaires

Au Canada, les autorités se partagent la responsabilité du bruit en provenance de différentes sources, notamment des transports. Le gouvernement fédéral règlemente les gênes sonores issues du transport aérien, le domaine qui nous intéresse (Laplace, 2022). Une analyse des effets du bruit des avions sur les êtres humains est essentielle notamment pour les gestionnaires des aéroports que pour les responsables de l'aménagement des terrains aux alentours de ces aérodromes (Canada, 2013).

En France, un projet dénommé « noisecapture » porté par l'ifsttar et le CNRS a vu le jour en 2018. Le projet consiste à la mise en œuvre d'une approche collaborative pour produire des données avec des smartphones afin de proposer une méthode d'évaluation de l'environnement sonore. Par ailleurs, les données collectées sont librement accessibles par ceux qui veulent à leur tour ajouter des fonctionnalités supplémentaires (Picaut, Nicolas Fortin, & Pierre Aumond).

Notre contribution

Nous utilisons un radar³ de surveillance, un radar secondaire qui est un système de détection de cible utilisé dans le service de la navigation aérienne afin de détecter sa position, et le cas échéant d'avoir d'autres informations supplémentaires.

L'objectif est ici de construire une carte de bruit des avions aux alentours d'un aéroport par la méthode d'apprentissage automatique.

Le dataset de notre machine d'apprentissage est conçu en quatre phases :

- la première phase consiste à extraire du radar MSSR d'Antananarivo les points qui constituent les trajectoires des avions les plus bruyants en décodant les données ASTERIX⁴ CAT 48 via la librairie asterix4py⁵; la période considérée va du 12 septembre au 28 septembre 2022.
- la deuxième phase à l'utilisation de la base de données des avions appelée « aircraft database », fruit de détections issues de 6000 senseurs (radar, ADS-B⁶, TCAS⁷) de par le monde (opensky-network, 2023);

³ Radio Detecting And Ranging

⁴ All purpose STructured Eurocontrol suRveillance Information eXchange catégorie 48 qui standardise la transmission des détections issues d'un mono radar.

⁵ Asterix4py est une librairie python utilisée pour décoder et parser les données codées en binaire sous format ASTERIX

⁶ Automatic Dependant Surveillance Broadcast est un senseur dépendant fonctionnant sur la diffusion de signal en onde électromagnétique sur la fréquence 1090Mhz en squitter long

⁷ Traffic Collision Avoidance System est un instrument installé à bord d'un avion qui permet l'éviction de collision avec d'autres avions

- la troisième étape est le tri au hasard des points dans l'espace d'approche.
- la quatrième et dernière étape consiste au nettoyage et à la transformation des données nettoyées avec un format utilisable par la machine .

La figure 1 représente les phases de conception de la carte de bruit :

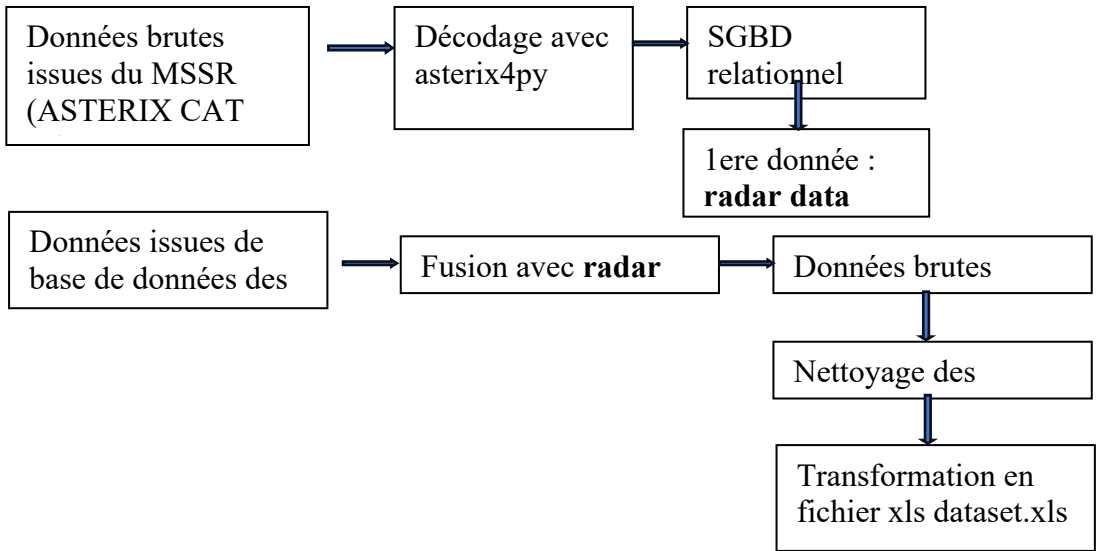


Figure 1 : Phases de conception de la carte de bruit

Choix des algorithmes

Le jeu de données ou *dataset* dont nous disposons constitue une ressource précieuse, il constitue le coeur de notre machine d'apprentissage. Il faut pouvoir l'utiliser à bon escient, afin de pouvoir à la fois **choisir** un modèle et **l'entraîner** (ce que nous avons fait dans le chapitre pratique précédent), mais aussi de pouvoir **tester** la qualité de ce modèle.

Les attributs **x1**, **x2** et **x3** seront utilisés et ces attributs correspondent à :

- x1 : la latitude du point de la zone
- x2 : la longitude du point de la zone
- x3 : l'altitude du point de la zone

Pour ce faire, nous allons définir les hypothèses suivantes :

- une trajectoire donnée relative à l'atterrissage et au décollage est un ensemble de points x_i avec $x_i = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$
- $\exists$ une boule $B(g_i, r)$ définie par un centre g_i et un rayon $r > 0$
- $\exists r_0 > 0$ tel qu'à l'intérieur de $B(g_i, r_0)$, le bruit est nuisible sinon le bruit est non nuisible. Pour la gêne pendant le jour et la nuit, cette valeur de r_0 va changer.

De tout ce qui précède, nous pouvons modéliser notre sujet comme un problème de classification binaire qui consiste à déterminer si un point dans l'espace aérien (en phase d'approche) est qualifié de bruit nuisible ou non.

Nous choisissons trois algorithmes de classification avec les différentes étapes suivantes :

- Utilisation du même fichier de dataset avec 692876 données après la phase de nettoyage et de transformation, la figure 2 représentant un aperçu du dataset :

	A	B	C	D	E
1		latPoint	lonPoint	altPoint	y
2	0	-18,7213	47,50076	1515,287	1
3	1	-19,8308	47,59254	1452,059	0
4	2	-18,8602	48,44859	1877,605	0
5	3	-18,2793	48,2478	1007,763	1
6	4	-18,806	47,73083	1111,209	1
7	5	-19,0239	47,68184	1291,186	1
8	6	-18,2829	47,78024	1892,092	1
9	7	-17,9274	47,24241	2123,434	0
10	8	-18,9655	47,68775	1869,493	1
11	9	-18,8288	47,90102	1856,233	0
12	10	-18,8307	47,65472	1943,967	1
13	11	-18,7449	47,46639	1858,751	1
14	12	-18,6363	47,95912	1897,37	1
15	13	-18,7546	46,63049	1143,499	1
16	14	-18,4827	47,56754	1569,708	1
17	15	-18,8123	46,79234	1388,872	1
18	16	-19,2671	47,16281	2046,493	1
19	17	-19,3659	47,39475	2085,287	1
20	18	-18,4034	47,65856	2019,5	1
21	19	-18,8442	48,15328	1654,65	0
22	20	-18,1245	47,55082	1454,612	1
23	21	-18,4506	47,61111	1877,756	1

Figure 2 : Aperçu du dataset

- Quant à la phase d'entraînement, elle est propre pour chaque algorithme, et la répartition est construite avec 80% de données pour l'entraînement et 20% des données restantes pour le test.

RESULTATS

Nous allons mettre en évidence les différents résultats obtenus de nos apprentissages automatiques construits avec les trois différents algorithmes utilisés pour le même dataset.

Construction avec la naïve bayésienne :

La figure 3 montre la métrique pour le modèle construit avec naïve bayésienne.

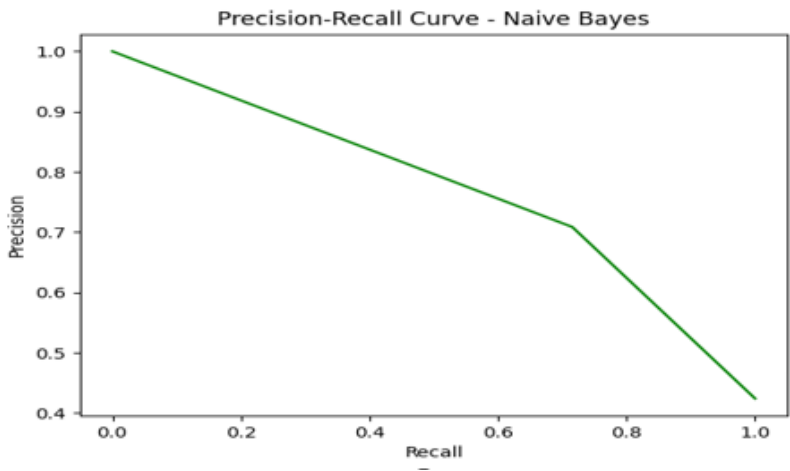


Figure 3 : Précision et recall avec naïve bayésienne

Et la figure 4 illustre toutes les métriques issues de la fonction naïve bayésienne.

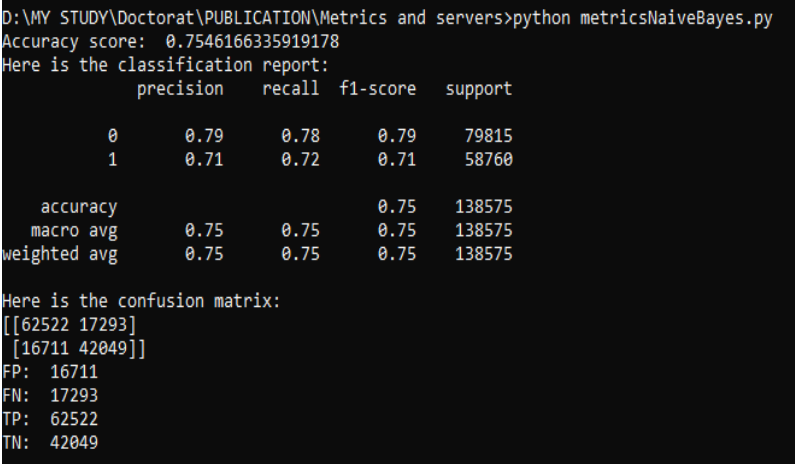


Figure 4 : Matrice de confusion de naïve bayésienne

Construction avec l’arbre de décision

La figure 5 affiche la précision et le recall pour le modèle construit avec l’arbre de décision.

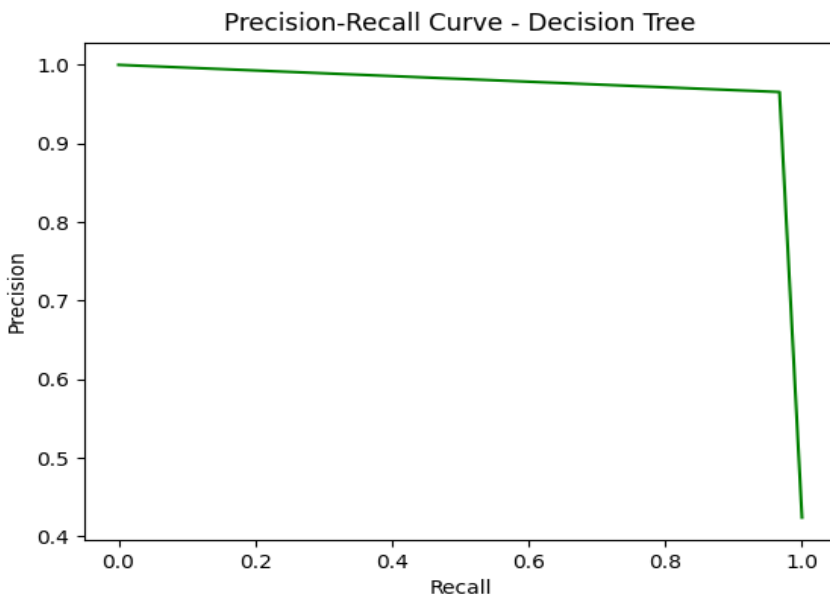


Figure 5 : Précision et Recall avec l'arbre de décision

La figure 6 représente toutes les métriques pour ce modèle.

```
D:\MY STUDY\Doctorat\PUBLICATION\Metrics and servers>python metricsDecisionTree.py
Accuracy score: 0.9714667147753924
Here is the classification report:
              precision    recall  f1-score   support

     0           0.98       0.97       0.98       79815
     1           0.97       0.97       0.97       58760

   accuracy          0.97
  macro avg          0.97
 weighted avg          0.97

Here is the confusion matrix:
[[77792  2023]
 [ 1931 56829]]
FP: 1931
FN: 2023
TP: 77792
TN: 56829
```

Figure 6 : Matrice de confusion de l'arbre de décision

Construction avec la forêt aléatoire

La figure 7 met en évidence deux métriques pour le modèle construit avec la forêt aléatoire.

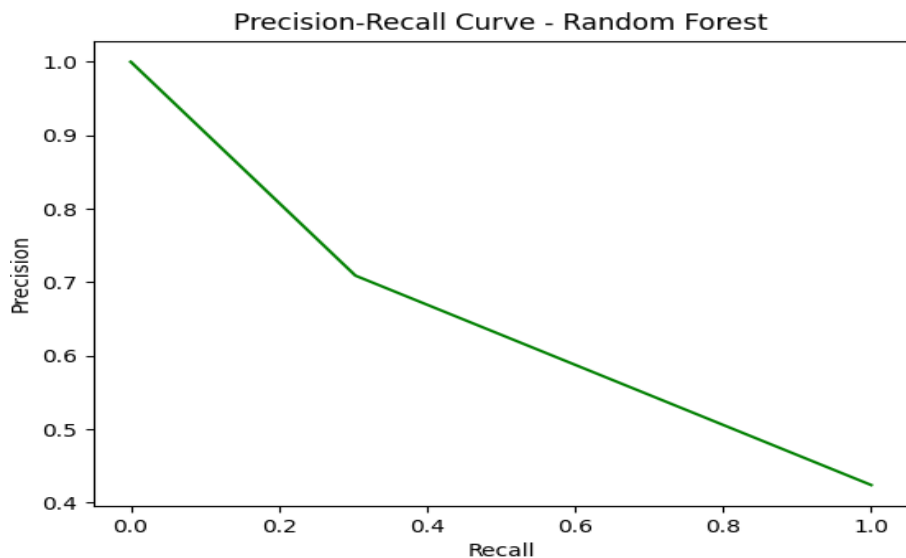


Figure 7 : Précision et recall avec la forêt aléatoire

La figure 8 représente la matrice de confusion pour la forêt aléatoire.

```
D:\MY STUDY\Doctorat\PUBLICATION\Metrics and servers>python metricsRandomForest.py
Accuracy score: 0.6518419628360094
Here is the classification report:
              precision    recall  f1-score   support

     0       0.64       0.91       0.75       79815
     1       0.71       0.30       0.43       58760

 accuracy          0.65       138575
 macro avg       0.67       0.61       0.59       138575
weighted avg       0.67       0.65       0.61       138575

Here is the confusion matrix:
[[72496  7319]
 [40927 17833]]
FP: 40927
FN: 7319
TP: 72496
TN: 17833
```

Figure 8 : Matrice de confusion de la forêt aléatoire

Les résultats correspondant aux entrainements associés aux trois fonctions sont donnés par le tableau 1 ci-après :

	Fonction		
Nom de la métrique	Naïve bayésienne	Arbre de décision	Forêt aléatoire
Précision (0, 1)	(79%, 71%)	(98%, 97%)	(64%, 71%)
Recall (0, 1)	(78%, 72%)	(97%, 97%)	(91%, 30%)
F1-score (0, 1)	(79%, 71%)	(98%, 97%)	(75%, 43%)

Tableau 1 : Tableau comparatif des métriques liées à chaque fonction

Pour rappel, la métrique **F1-score** désigne le score qui appartient à la famille plus large des F-beta scores mais ici les erreurs faites sont pondérées par un facteur $\frac{1}{2}$. Un des deux paramètres contribuant à la métrique F1-score est la précision ou encore Positive Predictive Value donnée

Plus la précision se rapproche de 100%, plus le modèle est précis. Le deuxième paramètre contribuant à la métrique F1-score est la « sensitivity », cette métrique porte aussi le nom de Recall ou True Positive Rate. Cette métrique se réfère à la probabilité d'un test positif, conditionné en étant vraiment positif.

Nous pouvons déduire qu'en entraînant notre dataset avec les trois fonctions, le résultat donné par l'arbre de décision donne une matrice de confusion aux alentours de 97% : ce pourcentage illustre que notre modèle construit aux alentours de 97% possède une vraisemblance élevée.

Avant de construire notre fonction $y = f(x_1, x_2, x_3)$, nous avons voulu étudier l'interdépendance de nos variables explicatives x_1 , x_2 et x_3 avec les fameuses méthodes de l'analyse en composantes principales ou l'analyse en facteurs communs. Etant donné que la latitude, la longitude et l'altitude sont des variables indépendantes, cette étape ne fut pas considérée.

DISCUSSION

Faute d'informations relatives au Bruitparif, la comparaison fut uniquement axée entre l'AEDT et notre modèle : le tableau 2 résume la comparaison entre ces deux modèles.

Critères	AEDT	Notre modèle
Source de données	Base de données BADA ⁸ d'EUROCONTROL (EUROCONTROL, 2020)	Radar secondaire
Type de données	• Données pour des fins de simulation et non sujet du projet ESSAR⁹ (Cynthia, Thrasher, Theodore, & Boeker, Eric, 2019) • Données non considérées « Real World Data » (EUROCONTROL, 2020)	Données réelles issues de détection de capteur.
Métriques	Métrique « sensitivity » à améliorer (Gabrielian, Puranik, Bendarkar, & Kirby, 2021)	Sensitivity : 97% pour l'arbre de décision.

Tableau 2 : Tableau de comparaison entre l'AEDT et notre modèle

La figure 9 représente l'aperçu de la carte sur laquelle nous avons modélisé la détection de la nuisibilité au bruit dans les zones à l'intérieur du CTR¹⁰ (couleur verte sur la figure 9) et de la TMA¹¹ (couleur rose sur la figure 9) de l'aéroport international d'Ivato.

⁸ Base of Aircraft Data est un modèle de performance avion développé et maintenu par EUROCONTROL pour des fins de simulation et de prédiction de trajectoires des avions

⁹ EUROCONTROL Safety Regulatory Requirement

¹⁰ Control Terminal Region est un espace réglementé destiné à protéger les vols à l'arrivée ou au départ d'un aéroport.

¹¹ Terminal Manoeuvring Area est un espace aérien réglementé destiné à protéger les vols en approche ou au départ d'un ou plusieurs aéroports.

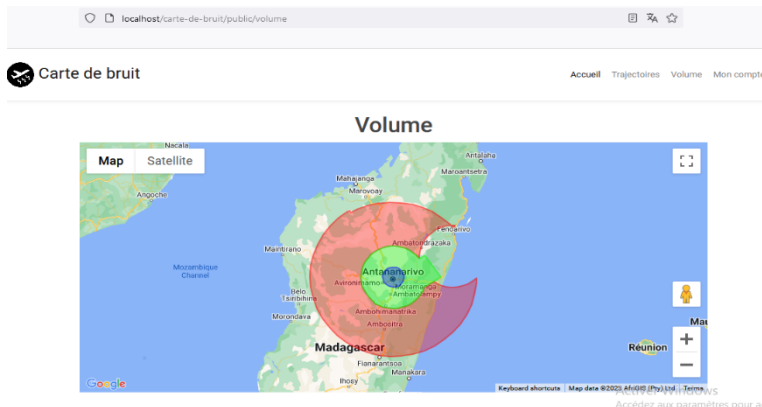


Figure 9 : Aperçu de la carte d'élaboration de la détection de nuisibilité au bruit

Les contours de cette carte ont été conçus à travers un script appelé « **fdp_volumes_definition.sql** » qui contient une table « **arcs** » et dont les valeurs sont des points exprimés en coordonnées géographiques récupérés dans le fichier de définition de la base de données du système de gestion de trafic aérien utilisé par le centre de contrôle de la navigation aérienne.

La figure 10 représente l'aperçu de la table « **arcs** » dudit script. Dans cette figure, la famille des points DZP, HAI et MG existent bel et bien dans le système de gestion de trafic aérien. DZP se réfère à Dzaoudzi, HAI pour Moroni et MG pour Mahajanga.

```
CREATE TABLE `arcs` (
  `name` varchar(8) NOT NULL,
  `start` varchar(8) NOT NULL,
  `end` varchar(8) NOT NULL,
  `center` varchar(16) NOT NULL,
  `centerLatitude` double NOT NULL,
  `centerLongitude` double NOT NULL,
  `arcPrecision` varchar(3) NOT NULL DEFAULT 'LO',
  `categoryID` int(2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `arcs`
--

INSERT INTO `arcs` (`name`, `start`, `end`, `center`, `centerLatitude`, `centerLongitude`, `arcPrecision`,
('DZPE30', 'DZP02', 'DZP03', '124830S0451642E', -12.808333333333334, 45.278333333333336, 'LO', 1),
('HAI107', 'HAI03', 'HAI04', '113054S0431636E', -11.515, 43.276666666666664, 'LO', 1),
('HAI171', 'HAI01', 'HAI02', '113054S0431636E', -11.515, 43.276666666666664, 'LO', 1),
('MG20E', 'MG01', 'MG02', '153927S0462026E', -15.6575, 46.340555555555554, 'LO', 1),
('MG20W', 'MG02', 'MG01', '153927S0462026E', -15.6575, 46.340555555555554, 'LO', 1),
('MGE60', 'MG03', 'TNV12', '153927S0462026E', -15.6575, 46.340555555555554, 'LO', 1),
```

Figure 10 : Aperçu des lignes de code du script « **fdp_volumes_definition.sql** »

Après avoir entraîné les données, nous pouvons entrer un point pris au hasard et trouver si le bruit généré par l'avion et perçu à ce point est nuisible ou non en considérant éventuellement les gènes. La figure 11 met en exergue un résultat non nuisible et donc pas de forte gêne et ne perturbe pas le sommeil.



Figure 11 : Point non nuisible au bruit

Pour rappel, les indicateurs de niveau sonore utilisés dans le cadre de la réalisation des cartes stratégiques de bruit et ils sont portés pour trois périodes de la journée :

- *Lday* : niveau sonore de la période de jour, de 6h à 18h (d'une durée de 12h)
- *Levening* : niveau sonore de la période de soirée, de 18h à 22h (d'une durée de 4h),
- *Lnight* : niveau sonore de la période de nuit, de 22h à 6h (d'une durée de 8h)

L'indicateur *Lden* est le niveau sonore équivalent pondéré sur 24 heures calculé à partir des niveaux sonores pour les périodes jour, soir et nuit, en donnant un poids plus fort au bruit en soirée (+ 5 dB(A)) et la nuit (+ 10 dB(A)) afin de tenir compte de la sensibilité accrue des populations au bruit durant ces deux périodes.

La figure 12 illustre non seulement un cas nuisible mais aussi avec une forte gêne et perturbe le sommeil.

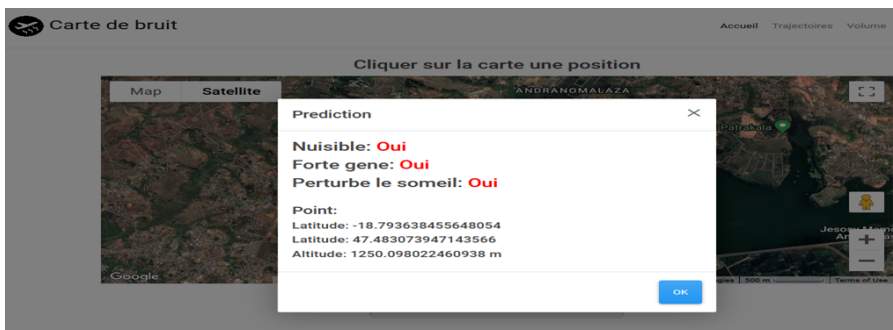


Figure 12 : Point nuisible au bruit

Cette fonctionnalité est obtenue en considérant ces paramètres dans la fonction de détermination de notre variable expliquée y. La figure 13 illustre des lignes de code qui tiennent compte de ces indicateurs de bruit.

```
def getYs(lat1, lon1, alt1):
    bruitInitial = 150
    bruitFinal = 80
    rayonNuisible = getDistanceRadius(bruitInitial, bruitFinal, 1)
    bruitFinal = 55
    rayonDayEvening = getDistanceRadius(bruitInitial, bruitFinal, 1)
    bruitFinal = 40
    rayonNight = getDistanceRadius(bruitInitial, bruitFinal, 1)
    sqlData = "SELECT latitude, longitude, altitude FROM radar_data"
    myCursor.execute(sqlData)
    radarData = myCursor.fetchall()
    yNuisible, yDayEvening, yNight = 0, 0, 0
    d = 0
    for data in radarData:
        lat2 = data[0];
        lon2 = data[1];
        alt2 = data[2];
        d = slant(lat1, lon1, alt1, lat2, lon2, alt2)

        if d < rayonNight:
            yNight = 1

        if d < rayonDayEvening:
            yDayEvening = 1

        if d < rayonNuisible:
            yNuisible = 1
            yNight = 1
            yDayEvening = 1
            break
    return str(yNuisible) + "," + str(yDayEvening) + "," + str(yNight)
```

Figure 13 : Illustration des indicateurs de bruit dans les lignes de code

La carte de bruit permet aussi de donner la trajectoire d'un avion saisi avec son adresse mode S¹² qui lui est propre. Cette trajectoire est aussi visible et matérialisée par le trait en rouge dans l'application (figure 14), elle nous permet entre autres de donner une zone de bruits aux alentours de l'aéroport.



Figure 14 : Trajectoire en approche d'un avion

¹² Chaque avion, pour être détecté par un radar mode S, se voit attribuer une adresse fixe codée sur 24 bits

CONCLUSION ET PERSPECTIVES D'AMELIORATION

Après avoir utilisé les détections issues d'un radar secondaire comme source de données, nous avons pu dans un premier temps, à partir du principe de nuisibilité de bruit à l'oreille humaine, construire une carte de bruit aux alentours d'un aéroport en partant de la fonction de propagation du bruit dans l'espace. Il est considéré ici un problème de classification où la fonction consiste à déterminer la nuisibilité au bruit ou non d'un point aux alentours de l'aéroport. Tout au long du processus de modélisation, nous avons choisi certains algorithmes comme la naïve bayésienne, l'arbre de décision et la forêt aléatoire, et nous avons pu apprécier les métriques données par l'arbre de décision.

L'amélioration du modèle serait l'utilisation de la logique floue au détriment de la logique classique. Pour ce faire, nous allons nous retrouver avec une valeur expliquée réelle entre 0 et 1 : à l'état actuel, le problème consiste à une classification binaire. En combinant les attributs nuisibilité et gêne, nous pourrions transformer l'étude en un problème de classification multiclassées avec des degrés passant du non nuisible, tolérable, plus tolérable au très nuisible.

Outre cela, se focaliser sur la durée d'exposition d'une valeur de bruit pour un temps donné serait aussi appréciable, tout en prenant en compte le temps de passage de l'avion.

BIBLIOGRAPHIE

- 1) Canada, D. g. (2013). Aviation Utilisation des terrains au voisinage des aérodomes. Ottawa, CANADA: Direction des normes.
- 2) Cidb. (2024, janvier 25). www.bruit.fr. Récupéré sur <https://www.bruit.fr/particuliers/bruit-des-transports/trafic-aerien>.
- 3) Cynthia, L., Thrasher, Theodore, & Boeker, Eric. (2019). Technical manual version 3b. Washington DC.
- 4) egis inframad. (2019). Suivi environnemental de l'exploitation de l'aéroport international d'Ivato. Antananarivo: Ravinala airports.
- 5) EUROCONTROL. (2020). Licence agreement for release. Bruxelles: EUROCONTROL.
- 6) Gabrielian, A., Puranik, T. G., Bendarkar, M. V., & Kirby, M. (2021). Noise Model Validation using RealWorld Operations Data. Atlanta: FAA Office of Environment and Energy.
- 7) Laplace, J. (2022, février). Encadrement du bruit environnemental au Canada. Canadian Public Policy.

- 8) Nassur, A. M. (2018). Effets de l'exposition au bruit des avions sur la qualité du sommeil des riverains des aéroports français. Lyon: Université de Lyon.
- 9) opensky-network. (2023, décembre 15). Aircraft database. Récupéré sur opensky-network: <https://opensky-network.org/aircraft-database/>
- 10) Picaut, J., Nicolas Fortin,, & Pierre Aumond. (s.d.). Noisecapture: une démarche Open Science pour la collaborative de cartes de bruit à l'aide de smartphones. HAL open science.