

Détection de Bateaux de Plaisance dans une Marina par Deep Learning

Romane Scherrer*, Erwan Aulnette**, Thomas Quiniou*, Jöel Kasarhérou**
Pierre Kolb**, Nazha Selmaoui-Folcher*

* ISEA, Université de la Nouvelle-Calédonie
romane.scherrer@hotmail.fr,

**L2K Innovation, Nouvelle-Calédonie

Résumé. Un système acoustique autonome basé sur deux hydrophones, une carte audio et un nano ordinateur a été installé à l'entrée d'une marina pour détecter le passage des bateaux. Une succession de corrélations croisées est calculée par le système pour déterminer le retard temporel entre les signaux des hydrophones et construire un signal fonction de la trajectoire des bateaux. Depuis son installation, le nano ordinateur effectue la détection des bateaux à l'aide d'un algorithme basé sur le traitement du signal avec une justesse de 80%. Pour améliorer les performances du système, un réseau de neurones à convolution (CNN) est entraîné avec les données acquises pour effectuer une détection en temps réel. Deux tâches de classifications sont considérées (binaire et multi-classes) pour à la fois détecter un bateau et son sens de navigation. Finalement, un CNN entraîné est implémenté dans un nano ordinateur pour évaluer son temps d'exécution.

1 Introduction

Depuis l'invention du sonar, les signaux acoustiques marins ont suscité un intérêt croissant. Bien que les premières inventions aient été centrées sur des applications militaires, plusieurs études récentes proposent d'utiliser les signaux acoustiques pour de toutes autres applications, comme la régulation du trafic maritime (Zwemer et al., 2018; Fillinger et al., 2009), la surveillance sous-marine (Fillinger et al., 2010) ou encore l'évaluation de l'impact des sons sur la vie marine (Codarin et al., 2009; Holles et al., 2013).

Puisque le signal acoustique émanant d'un bateau possède plusieurs sources (hélice, moteur, hydrodynamique, vibrations...) qui produisent des signaux à différentes fréquences, la plupart des méthodes développées pour détecter, classifier ou suivre des bateaux sont basées sur l'analyse fréquentielle ou temps-fréquence. Plusieurs méthodes ont ainsi été développées pour détecter les harmoniques du signal et pour extraire la signature acoustique des bateaux. Dans la plupart des cas, ces méthodes sont basées sur l'analyse du spectre (Guo et al., 2020), du spectre DEMON (Chung et al., 2011) et du cepstre (Santos-Domínguez et al., 2016; Das et al., 2013).

Ces dernières années, l'intelligence artificielle et les réseaux de neurones ont été appliqués avec succès dans des domaines d'application très variés. Une architecture de réseau de

neurones très largement utilisée en vision par ordinateur est celle des réseaux de neurones à convolution (CNN). Cette architecture est notamment connue pour sa capacité à extraire des caractéristiques représentatives à l'aide de ses couches de convolution et pour avoir fourni des résultats remarquables dans de nombreux problèmes comme la classification d'images (Krizhevsky et al., 2012) ou la reconnaissance vocale (Palaz et al., 2015). Motivées par le succès des CNN, plusieurs études proposent d'utiliser cette architecture pour accomplir de la classification sur des séries temporelles (Cui et al., 2016; Le Guennec et al., 2016; Zhao et al., 2017).

Les systèmes acoustiques passifs développés pour acquérir les signaux acoustiques sous-marins sont souvent basés sur l'utilisation de plusieurs hydrophones connectés à une unité centrale (Fillinger et al., 2010; Guo et al., 2020). Le retard temporel entre les signaux reçus par les hydrophones peut alors être mesuré, ce qui permet l'estimation de la position angulaire de la source. Contrairement aux méthodes précédemment citées qui sont basées sur l'analyse fréquentielle ou temps-fréquence pour détecter la présence d'un bateau dans un environnement bruyant, nous proposons ici d'utiliser uniquement la mesure du retard entre deux hydrophones. Le problème de détection d'un bateau peut alors être vu comme un problème de classification de série temporelle.

Nos travaux s'inscrivent dans une démarche d'amélioration de la sécurité d'une marina en détectant automatiquement l'arrivée et le départ d'un bateau en utilisant un système acoustique intelligent et abordable en termes de coût. Par sécurité, nous faisons référence au risque d'intrusion dans la marina. Le système est encore en développement et depuis son installation à Port Brunelet, Nouvelle-Calédonie en juillet 2019, il a enregistré des milliers de signaux acoustiques qui sont analysés pour améliorer la justesse de détection. Dans ce papier, nous étudions la capacité d'un CNN à détecter en temps réel la présence d'un bateau dans un marina et son sens de navigation.

Le papier est organisé de la façon suivante. La section 2 introduit la méthodologie utilisée pour suivre la position angulaire d'un bateau et donne un aperçu du processus d'acquisition du système. La section 3 décrit la phase de traitement des signaux acoustiques enregistrés. Les sections 4 et 5 présentent l'architecture du CNN utilisé et les résultats de classification. Finalement, la section 5 donne les conclusions et perspectives.

2 Méthodologie

2.1 Retard temporel

Considérons pour cette analyse deux hydrophones H_1 et H_2 séparés par une distance L et notons $h_i(t)$ le signal perçu par l'hydrophone H_i . La position du bateau forme un angle $\alpha(t)$ avec la normale du segment reliant les hydrophones. En considérant que $D \gg L$ (champ lointain) et notant c la vitesse de propagation du son dans l'eau de mer (approx. 1500 m/s), nous pouvons montrer que le front d'onde émis par le bateau atteint les hydrophones avec un décalage temporel $\Delta T(t)$ tel que :

$$\Delta T(t) = \frac{L}{c} \sin(\alpha(t)) \quad (1)$$

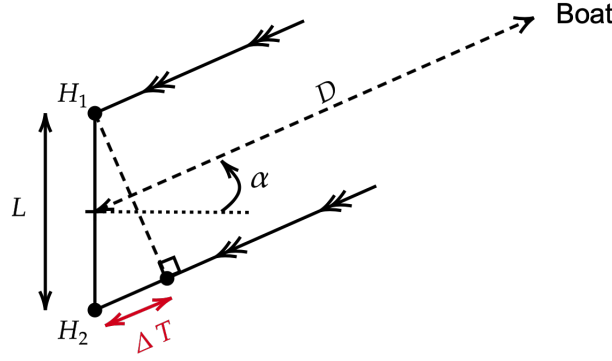


FIG. 1 – Les signaux enregistrés par les hydrophones sont des versions décalées temporellement du même signal.

Le décalage temporel entre $h_1(t)$ et $h_2(t)$ dépend de la position angulaire du bateau et change lorsque la source se déplace (α étant fonction du temps t). Ainsi, pour déterminer la direction de navigation du bateau, il est possible d'analyser uniquement la variation de ΔT dans le temps.

La corrélation croisée est une mesure de similarité entre deux signaux et peut être appliquée pour déterminer le retard temporel (déplacement) d'un signal par rapport à un autre. Pour deux fonctions continues $h_1(t)$ et $h_2(t)$, nous rappelons que la corrélation croisée est définie par :

$$XCorr(h_1, h_2)(\tau) = \int_{-\infty}^{+\infty} h_1^*(t)h_2(t + \tau)dt \quad (2)$$

où * représente le conjugué complexe et τ le décalage temporel.

Si les signaux $h_1(t)$ et $h_2(t)$ sont décalés de ΔT , alors la corrélation croisée possède un maximum à $\tau = \Delta T = \frac{L}{c} \sin(\alpha)$. Ainsi, pour suivre la trajectoire angulaire $\alpha(t)$ d'un bateau avec des signaux discrets, nous avons choisi de corréler les signaux h_1 et h_2 dans de courtes fenêtres de N points. L'argument du maximum de la corrélation dans chaque fenêtre est ensuite stocké pour former un signal discret $\Delta T[t_i]$ qui est fonction de la trajectoire du bateau. Afin de gagner du temps de calcul et d'économiser des ressources hardware, il n'est pas nécessaire de calculer la corrélation croisée $XCorr(\tau)$ pour tous les τ variant sur un intervalle égale à $2N - 1$. En effet, en ignorant les trajets multiples, le front d'onde issu du bateau atteint les hydrophones avec un décalage temporel maximum égale à $|\Delta T| \leq \frac{L}{c} = \tau_{max}$. Ainsi, le calcul de la corrélation croisée sur une fenêtre de taille N et la recherche du maximum peut être effectuée sur un intervalle $\tau \in [-\tau_{max}, \tau_{max}]$.

La figure 2 montre un exemple du processus décrit pour des signaux enregistrés d'un bateau naviguant dans une marina. Nous pouvons constater que le signal $\Delta T[t_i]$ construit par corrélation croisée possède bien une forme sinusoïdale, ce qui est cohérent avec l'équation 1.

Détection de Bateaux de Plaisance dans une Marina par Deep Learning

De plus, le sens de navigation du bateau (entrant ou sortant de la marina) peut être déduit en fonction de la variation du signe de $\Delta T[t_i]$. Dans cet exemple, le signal passe progressivement d'une valeur positive ($\alpha \in [0, \pi]$) à négative ($\alpha \in [\pi, 2\pi]$) ce qui correspond pour notre configuration d'hydrophones à un bateau sortant de la marina. Il est important de noter que la forme de $\Delta T[t_i]$ dépend de la vitesse de navigation du bateau et de sa trajectoire. De plus, le bruit ambiant et les potentielles interférences contribuent également à la corrélation croisée et peuvent rendre le signal très bruité (voir la figure 5 pour plus d'exemples).

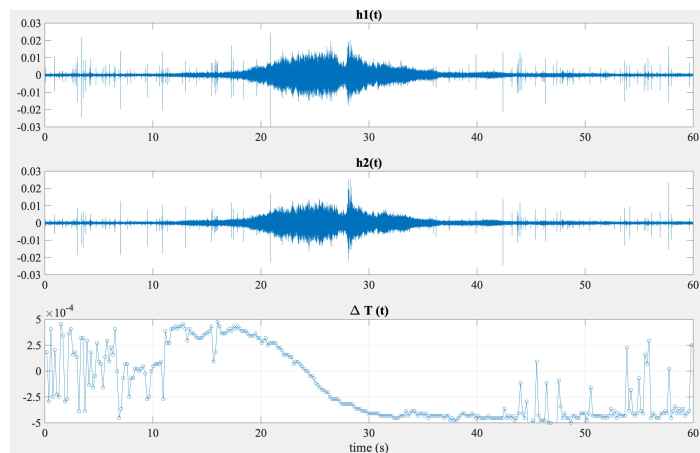


FIG. 2 – h_1, h_2 and le signal construit par corrélation croisée d'un bateau sortant de la marina.

2.2 Présentation du système autonome et l'acquisition des données

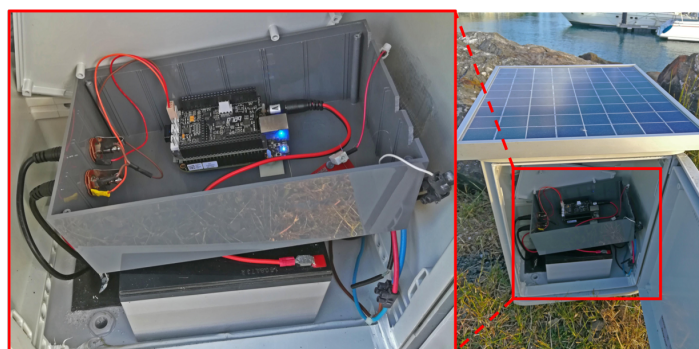


FIG. 3 – Système autonome développé pour la détection de bateaux dans la marina.

Dans le cadre de notre projet qui consiste à améliorer la sécurité d'une marina, nous avons développé un système autonome (figure 3) basé sur du matériel à prix abordable. L'objectif

principal est de déterminer en temps réel l’arrivée et le départ des bateaux et de quantifier le trafic dans la marina. Depuis juillet 2019, le système est opérationnel et a sauvé des milliers de signaux qui ont été traités pour entraîner un CNN et améliorer la robustesse et la précision de détection du système.

Notre système acoustique passif utilise deux hydrophones omnidirectionnels (H1A, *Aquarian hydrophones*) lestés à deux mètres sous le niveau d’eau à l’entrée de la marina (figure 4). Les hydrophones sont séparés d’une distance $L = 0.8$ m et sont connectés à une carte audio à faible latence (*Bela*¹) et un nano-ordinateur (*BeagleBone Black*²). Le système enregistre les signaux h_1 et h_2 perçus par les hydrophones à une fréquence d’échantillonnage de 44.1 kHz. Toutes les 186 ms, la corrélation croisée entre les signaux est calculée et l’argument du maximum est ajouté dans un fichier qui est enregistré sur une carte SD toutes les 10 minutes.

Le système est entièrement autonome et fonctionne avec une batterie rechargée par un panneau solaire. Afin d’économiser de l’énergie et parce qu’il n’y a pas de bateaux entrants ou sortants de la marina à la tombée de la nuit, le système s’éteint automatiquement à 18h et s’allume à 6h. Pendant la journée, le système sauvegarde, par heure, 6 fichiers contenant les résultats des corrélations croisées.

Pendant la première phase de développement du projet, un premier algorithme a été implémenté pour distinguer les bateaux du bruit ambiant. L’algorithme fonctionne de la manière suivante. Tout d’abord, pour chaque changement de signe dans le signal $\Delta T[t_i]$, une fenêtre de N points centrée sur le passage en zéro est appliquée. Puis, dans cette fenêtre, le nombre de points ayant le même signe avant et après le changement de signe est calculé. Finalement, si ce nombre de points est supérieur à un seuil prédéfini manuellement alors l’algorithme prédit qu’un bateau sort ou entre de la marina. La limite de cet algorithme est que le signal construit par corrélation croisée est souvent très bruité et dépend de la vitesse et de la trajectoire du bateau. Pour ces raisons, un seuil statique prédéfini ne permet pas de détecter tous les passages de bateaux dans la marina et la justesse de l’algorithme initial a été évaluée à 80%.

Pour améliorer la robustesse et le taux de prédiction de notre système, les signaux construits par corrélation croisée qui ont été sauvegardés entre juillet et novembre 2019 ont été traités pour entraîner un CNN.

3 Préparation des données

Pour chaque fichier enregistré, le signal construit par corrélation croisée est découpé en petites séries temporelles de 322 points (correspondant à un signal de 1 minute dans le domaine temporel). Afin d’éviter de couper brutalement une trajectoire, au début ou à la fin d’une séquence, un recouvrement de 50 points a été pris en compte. La longueur des séries temporelles a été choisie en prenant en compte plusieurs considérations. Tout d’abord, en fonction de sa vitesse, un bateau met entre 20 et 30 secondes pour entrer ou sortir de la marina, ce qui correspond à un signal de 100 à 160 points. Ainsi, un signal de 322 points est suffisamment long pour détecter le passage d’un bateau. De plus, l’inconvénient d’utiliser uniquement le signal construit par corrélation est qu’il est difficile de séparer la contribution de plusieurs bateaux

1. <https://bela.io/products/>

2. <https://beagleboard.org/black>

Détection de Bateaux de Plaisance dans une Marina par Deep Learning



FIG. 4 – Vue aérienne du Port Brunelet. Les points rouges représentent la position des hydrophones

dans la corrélation (par exemple deux bateaux qui entrent dans la marina en même temps). Bien que les trajectoires de plusieurs bateaux puissent être séparées si leurs contributions dans la corrélation croisée ne se chevauchent pas (Fillinger et al., 2011), cette séparation doit être faite avant de sauvegarder le signal construit par corrélation croisée. Or dans la première phase de développement de notre projet, le système suivait en temps réel la trajectoire d'un seul bateau et n'a pas enregistré les signaux h_1 et h_2 perçus par les hydrophones. Ainsi, en présence de plusieurs bateaux, le signal construit par corrélation croisée "suit" le bateau le plus bruyant et "saute" d'un bateau à l'autre en fonction de leurs contributions dans la corrélation croisée. Ainsi, prendre un signal de 322 points permet de diminuer la probabilité d'avoir plusieurs bateaux qui entrent ou sortent de la marina pendant la période d'observation. Cependant, toutes les séries temporelles du jeu de données qui présentent plusieurs bateaux ont été supprimées de l'étude (approx. 50).

Approximativement 10 700 séries temporelles ont été labellisées manuellement en fonction de leur nature (bruit, bateau entrant ou bateau sortant). Par bruit, nous faisons référence au bruit de fond, aux bruits de nature biologique et aux différents bruits générés par les bateaux qui restent à l'intérieur ou à l'extérieur de la marina sans franchir la barrière acoustique. Pour augmenter le nombre de séries temporelles qui présentent un bateau entrant ou sortant (classes sous représentées dans le jeu de données), toutes les séries temporelles de ces classes ont été doublées en prenant le signe opposé de chaque série. En effet, la trajectoire angulaire d'un bateau entrant peut être vue comme l'opposé de celle d'un bateau sortant.

Après avoir effectué l'augmentation des données, le jeu de données comprend 19 844 séries temporelles représentant du bruit et 1 524 série temporelles représentant un passage de bateau (équitablement réparties entre les classes "bateau entrant" et "bateau sortant"). Finalement, toutes les séries temporelles ont été normalisées entre 0 et 1 en fonction de leurs maximum et minimum. Hormis la normalisation, aucun autre traitement ou transformation n'a été effectué sur les séries temporelles qui ont servi à entraîner le CNN.

4 Deep Learning

L'objectif principal est de distinguer un passage de bateau du bruit ambiant en utilisant uniquement le signal construit par corrélation croisée. Pour cela, nous avons considéré deux

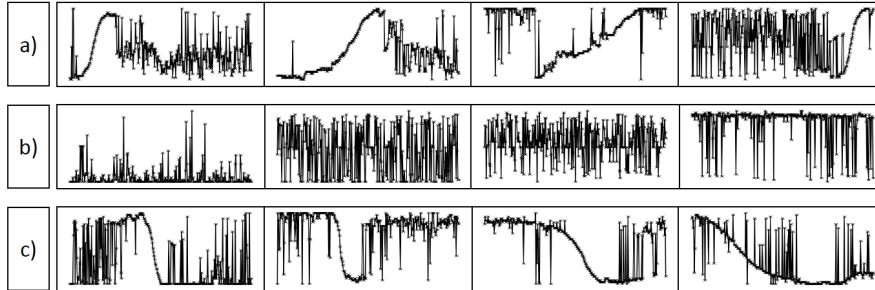


FIG. 5 – Quatre exemples de séries temporelles par classes (a) bateau entrant, (b) bruit et (c) bateau sortant

scénarios. Le premier consiste en une classification des séries temporelles en trois classes : *bateau entrant*, *bateau sortant* et *bruit*. Pour le second scénario, toutes les séries temporelles qui représentent un passage de bateau (indépendamment de leurs sens de navigation) sont regroupées dans une même classe notée *bateau* et le problème peut être vu comme une classification binaire (*bateau* vs *bruit*). Pour les deux scénarios, un CNN est construit et entraîné avec la librairie open source Keras³ v2.2. et Tensorflow⁴ v1.14 en back-end sous Python.

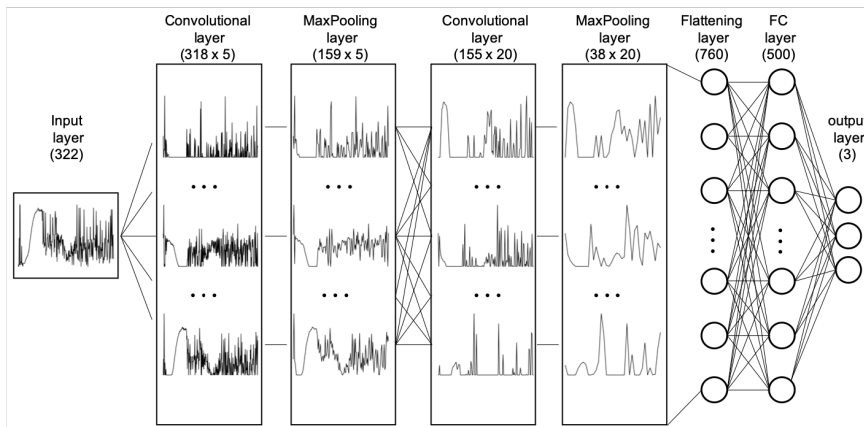


FIG. 6 – Architecture du CNN.

L'architecture des CNN utilisée pour les deux classifications est présentée sur la figure 6 et est basée sur t-LeNet (Le Guennec et al., 2016) qui est une version modifiée de LeNet (LeCun et al., 1989) pour la classification de séries temporelles. Notre modèle possède deux couches de convolution 1D qui sont suivies par une couche entièrement connectée (FC) et d'un classifieur softmax. Pour chaque convolution, la fonction d'activation ReLu est utilisée et la taille des kernels est de 5. La première couche de convolution utilise 5 filtres et est suivie

3. <https://keras.io>

4. <https://www.tensorflow.org>

par une couche de pooling de taille 2. La seconde couche de convolution utilise 20 filtres et est suivie d'une couche de pooling de taille 4 et d'un dropout avec une probabilité de 0,5. Finalement, la couche FC possède 500 neurones et la couche de sortie possède un nombre de neurones égale au nombre de classes.

Les performances des modèles sont évaluées grâce à l'approche train-validation-test avec un ratio de 60 :20 :20. Le CNN est entraîné avec l'optimiseur Adam et un learning rate initial de 0,001 avec un decay de 0,9. Pendant la phase d'apprentissage, la perte d'entropie croisée est calculée et pondérée pour forcer le modèle à prêter attention aux séries temporelles des classes sous-représentées. Le nombre d'épochs est initialisé à 500 mais l'apprentissage est arrêté lorsque la perte sur le jeu de validation n'a pas diminué pendant 10 epochs consécutifs. Pour chacun des scénarios, 40 CNNs sont entraînés avec différents hyperparamètres pour comparer leurs performances sur le jeu de validation et sélectionner le meilleur modèle. Finalement, les tableaux 1, 2 et 3 rapportent les résultats obtenus sur le jeu de test.

5 Résultats

5.1 Classification binaire

La figure 7 synthétise les résultats obtenus par les 10 meilleurs CNNs en fonction des différents hyperparamètres. Pour ce scénario, toutes les séries temporelles représentant un passage de bateau (indépendamment de la direction de navigation) sont regroupées dans une classe nommée *bateau*. Quatre hyperparamètres ont été considérés : le nombre de neurones (500, 400, 300, 200 et 100), la taille du batch (32 et 64), le learning rate (0,001 et 0,0001) et l'initialiseur de kernel (random normal et glorot uniform). Les cinq métriques utilisées pour évaluer les modèles sont définies de la manière suivante :

1. *Le rappel* qui est défini comme étant la proportion des séries temporelles appartenant à la classe *bateau* qui sont correctement prédites par le modèle.
2. *La spécificité* qui est définie comme étant la proportion des séries temporelles appartenant à la classe *bruit* qui sont correctement prédites par le modèle.
3. *la précision* qui quantifie le nombre de prédictions positives qui appartiennent réellement à la classe *bateau*.
4. *le F1-score* qui combine la précision et le rappel ($F1 - score = 2 \times precision \times rappel / (precision + rappel)$).
5. *La justesse équilibrée* (notée Bal.acc) qui est définie comme étant la moyenne de la spécificité et du rappel.

Pour choisir le meilleur CNN, les performances du modèle sur le jeu de validation ont été triées en fonction du F1-score pour minimiser le nombre de faux négatifs et de faux positifs. Par conséquent, le meilleur modèle possède 200 neurones dans la couche FC, un learning rate de 0,001, un batch size de 32 et un initialiseur glorot uniform.

La table 1 présente la matrice de confusion obtenue sur le jeu de test. Pour 304 séries temporelles dans la classe *bateau*, 280 sont correctement prédites par le modèle et 24 sont confondues avec le bruit de fond. Pour la classe *bruit*, seulement 14 des 3972 séries temporelles sont confondues avec un passage de bateau. Finalement, ce modèle donne une justesse équilibrée

de 0.96, un F1-score de 0.94, une précision de 0.95, un rappel de 0.93 et une spécificité de 0.99.

Hyperparameters	Score (%)			
	bal_Accuracy	F1-score	Precision	Recall
[200, 32, 0.001, 'GU'] -	96.3	94.6	96.4	92.8
[300, 32, 0.001, 'RN'] -	96.4	94.4	95.8	93.1
[100, 64, 0.001, 'GU'] -	96.5	94.3	95.2	93.4
[400, 64, 0.001, 'GU'] -	96.1	94.3	96.1	92.5
[300, 32, 0.001, 'GU'] -	95.9	94.2	96.4	92.2
[500, 32, 0.001, 'GU'] -	95.4	94.1	97.6	90.9
[300, 64, 0.001, 'GU'] -	95.8	94.1	96.4	91.8
[200, 64, 0.001, 'RN'] -	94.9	93.4	97	90
[400, 32, 0.001, 'GU'] -	95	93.3	96.6	90.3
[200, 64, 0.001, 'GU'] -	95	93.2	96.3	90.3

FIG. 7 – Performance des 10 premiers CNNs pour la classification binaire sur le jeu de validation pour différents hyperparamètres [Nombre de neurones, batch size, learning rate, kernel initializers ('RN' : random normal, 'GU' : glorot uniform)]

TAB. 1 – Matrice de confusion de la classification binaire

		Classe prédite	
		Bruit	Bateau
Classe réelle	Bruit	3956	14
	Bateau	24	280

5.2 Classification multi-classes

Pour ce scénario, les CNNs doivent effectuer une classification multi-classes afin de détecter un passage de bateau dans la marina et également son sens de navigation. Les trois classes sont nommées *bruit*, *sortant* et *entrant*. Comme précédemment, 40 CNNs ont été entraînés avec différents hyperparamètres et le meilleur modèle a été sélectionné en fonction de son F1-score sur le jeu de validation (figure 8).

Le tableau 2 présente la matrice de confusion obtenue sur le jeu de test. Parmi les 304 séries temporelles qui appartiennent aux classes bateau *entrant* ou *sortant*, seulement 19 sont incorrectement prédites dans la classe *bruit* ce qui montre bien la capacité du CNN à identifier la présence d'un bateau. De plus, 12 séries temporelles sur 3972 qui appartiennent à la classe *bruit* sont confondues avec un passage de bateau. Par conséquent, ce modèle présente de meilleurs résultats que le CNN sélectionné pour faire une classification binaire. De plus, il est capable de prédire la direction des bateaux puisqu'il n'a confondu que 6 fois la direction des bateaux sur l'ensemble du jeu de test.

Pour mieux évaluer les performances du CNN, une transformation one-vs-rest est réalisée sur la matrice de confusion multi-classes. Ainsi, pour chaque classe i , une matrice de confusion binaire est calculée de telle sorte que la classe i est considérée comme la classe positive et les classes $j \neq i$ comme la classe négative. Finalement, les performances du CNN pour chaque classes sont présentées dans le tableau 3. Nous pouvons remarquer que les performances des

Détection de Bateaux de Plaisance dans une Marina par Deep Learning

Hyperparamètres	Score (%)			
	bal_Accuracy	F1-score	Precision	Recall
[500, 64, 0.001, 'RN']	98	96.2	96.2	96.2
[200, 32, 0.001, 'GU']	96.9	95.3	96.5	94
[100, 32, 0.001, 'GU']	96.2	95.2	98	92.5
[500, 32, 0.001, 'RN']	95.9	95.2	98.7	91.8
[300, 32, 0.001, 'RN']	96.7	95.1	96.5	93.7
[200, 64, 0.001, 'GU']	96.6	95.1	96.8	93.4
[100, 32, 0.001, 'RN']	96.6	95.1	96.8	93.4
[200, 32, 0.001, 'RN']	96	95	98	92.2
[500, 32, 0.0001, 'GU']	96	94.9	97.7	92.2
[100, 64, 0.001, 'GU']	96.3	94.8	97	92.8

FIG. 8 – Performance des 10 premiers CNNs pour la classification multi-classes sur le jeu de validation. Hyperparamètres : [Nombre de neurones, batch size, learning rate, kernel initializers ('RN' : random normal, 'GU' : glorot uniform)]

CNNs sur les classes *entrant* et *sortant* sont sensiblement similaires et donnent les scores moyens suivants : une justesse équilibrée de 0.96, un rappel de 0.91, une précision de 0.94 et un F1-score de 0.93.

TAB. 2 – Matrice de confusion - classification multi-classes

		Classe prédite		
		Sortant	Bruit	Entrant
Classe réelle	Sortant	147	10	3
	Bruit	5	3960	7
	Entrant	3	9	132

TAB. 3 – Performances de la classification multi-classes

Classe	Justesse éq.	Rappel	Précision	F1-Score	spécificité
Sortant	0.958	0.919	0.948	0.933	0.998
Bruit	0.968	0.997	0.995	0.996	0.938
Entrant	0.958	0.917	0.930	0.923	0.998

5.3 Implémentation d'un CNN sur un nano ordinateur

Le CNN qui effectue la classification multi-classes a été choisi pour remplacer l'algorithme basé sur le traitement du signal initialement utilisé pendant le développement de notre système. Pour s'assurer que le CNN entraîné puisse fonctionner sur un nano ordinateur, il a été implémenté dans un Raspberry PI 3 modèle B + fonctionnant avec le système d'exploitation PI OS (anciennement appelé Raspbian). La carte équipe un SoC Broadcom BCM2837B0 qui dispose d'un cluster ARM Cortex-A53 quad-core, fonctionnant à 1,4 GHz. Pour mesurer le temps d'exécution, un seul coeur de processeur du Raspberry Pi a été utilisé par TensorFlow. Nous avons exécuté 10 000 fois une boucle de contrôle dans laquelle le CNN effectue la classification d'une série temporelle. Nous avons constaté que le temps d'exécution moyen pour prédire si un bateau entre ou sort de la marina est de 9,75 ms. Ce résultat montre que la prédiction peut

être faite en temps réel. Des travaux supplémentaires seront néanmoins nécessaires pour implémenter le CNN dans notre logiciel développé (allocation mémoire, parallélisme des tâches, ...) mais la prédiction du CNN sera utilisée comme déclencheur pour enregistrer les signaux bruts des hydrophones lorsqu'un bateau est détecté.

6 Conclusion et perspectives

Dans cet article, nous avons présenté notre système autonome basé sur deux hydrophones connectés à une carte audio à deux entrées et un nano ordinateur. Nous avons décrit comment le système utilise les signaux des hydrophones pour construire une barrière acoustique passive capable de détecter la circulation des bateaux à l'intérieur d'une marina. Les signaux des hydrophones sont corrélés par notre système autonome pour trouver le retard et pour construire un signal qui dépend de la trajectoire du bateau. En examinant comment ce signal évolue au fil du temps, le système peut détecter un bateau et sa direction.

Pour améliorer la robustesse de notre système, nous avons traité les signaux enregistrés et les avons manuellement labellisés en fonction de leur nature (bruit de fond, entrée et sortie de bateau). En conséquence, le jeu de données se compose de 21 378 séries temporelles dont 19 844 bruits de fond et 1 534 bruits rayonnés par bateau. Un total de 80 CNNs ont été entraînés hors ligne avec différents hyperparamètres pour choisir le meilleur modèle qui a atteint une justesse de 0,96.

La prochaine étape du développement de notre système acoustique est l'implémentation du CNN entraîné. Le résultat de la classification effectuée par le CNN sera utilisé comme déclencheur pour sauvegarder automatiquement les signaux des hydrophones lorsqu'un bateau est détecté. L'objectif est d'acquérir suffisamment de données pour effectuer la classification des signatures acoustiques des bateaux et pour détecter d'éventuels intrus à l'intérieur de la marina. D'autres travaux seront nécessaires pour suivre simultanément plusieurs bateaux.

Références

- Chung, K. W., A. Sutin, A. Sedunov, et M. Bruno (2011). DEMON acoustic ship signature measurements in an urban harbor. *Advances in Acoustics and Vibration 2011*(i).
- Codarin, A., L. E. Wysocki, F. Ladich, et M. Picciulin (2009). Effects of ambient and boat noise on hearing and communication in three fish species living in a marine protected area (Miramare, Italy). *Marine Pollution Bulletin 58*(12), 1880–1887.
- Cui, Z., W. Chen, et Y. Chen (2016). Multi-Scale Convolutional Neural Networks for Time Series Classification. *CoRR abs/1603.0*.
- Das, A., A. Kumar, et R. Bahl (2013). Marine vessel classification based on passive sonar data : The cepstrum-based approach. *IET Radar, Sonar and Navigation 7*(1), 87–93.
- Fillinger, L., P. De Theije, M. Zampolli, A. Sutin, H. Salloum, N. Sedunov, et A. Sedunov (2010). Towards a passive acoustic underwater system for protecting harbours against intruders. *2010 International Waterside Security Conference, WSS 2010*.
- Fillinger, L., A. Sutin, et A. Sedunov (2009). Cross-correlation of ship noise for water traffic monitoring. *The Journal of the Acoustical Society of America 126*(4), 2251.

Détection de Bateaux de Plaisance dans une Marina par Deep Learning

- Fillinger, L., A. Sutin, et A. Sedunov (2011). Acoustic ship signature measurements by cross-correlation method. *The Journal of the Acoustical Society of America* 129(2), 774–778.
- Guo, W., S. Piao, J. Guo, Y. Lei, et K. Iqbal (2020). Passive detection of ship-radiated acoustic signal using coherent integration of cross-power spectrum with doppler and time delay compensations. *Sensors (Switzerland)* 20(6).
- Holles, S. H., S. D. Simpson, A. N. Radford, L. Berten, et D. Lecchini (2013). Boat noise disrupts orientation behaviour in a coral reef fish. *Marine Ecology Progress Series* 485, 295–300.
- Krizhevsky, A., I. Sutskever, et G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks. pp. 1097–1105.
- Le Guennec, A., S. Malinowski, et R. Tavenard (2016). Data Augmentation for Time Series Classification using Convolutional Neural Networks. *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*.
- LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, et L. D. Jackel (1989). Backpropagation applied to digit recognition. *1*(4), 541–551.
- Palaz, D., M. Magimai-Doss, et R. Collobert (2015). Analysis of cnn-based speech recognition system using raw speech as input. In *Proceedings of Interspeech*, pp. 11–15. ISCA.
- Santos-Domínguez, D., S. Torres-Guijarro, A. Cardenal-López, et A. Pena-Gimenez (2016). ShipsEar : An underwater vessel noise database. *Applied Acoustics* 113, 64–69.
- Zhao, B., H. Lu, S. Chen, J. Liu, et D. Wu (2017). Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics* 28(1), 162–169.
- Zwemer, M. H., R. G. Wijnhoven, et P. H. De With (2018). Ship detection in harbour surveillance based on large-Scale data and CNNs. *VISIGRAPP 2018 5*(Visigrapp), 153–160.

Summary

An autonomous acoustic system based on two bottom-moored hydrophones, a two-input audio board and a small single-board computer was installed at the entrance of a marina to detect entering/exiting boats. Windowed time lagged cross-correlations are calculated by the system to find the consecutive time delays between the hydrophone signals and to compute a signal which is a function of the boats' angular trajectories. Since its installation, the single-board computer performs online prediction with a signal processing-based algorithm which achieved an accuracy of 80%. To improve system performance, a convolutional neural network (CNN) is trained with the acquired data to perform real-time detection. Two classification tasks were considered (binary and multiclass) to both detect a boat and its direction of navigation. Finally, a trained CNN was implemented in a single-board computer to ensure that prediction can be performed in real time.