Un réseau hybride CNN-LSTM pour la classification de navires à partir d'une base frugale des images SAR

Abdelmalek Toumi, Jean-Christophe Cexus, Mahdi Abid, Ali Khenchaf,

LAB-STICC, UMR CNRS 6285, Ensta-Bretagne
29806 Brest Cedex 9, France

{abdelmalek.toumi, jean-christophe.cexus, mahdi.abid, ali.khenchaf}@ensta-bretagne.fr

Résumé − Ce travail propose une architecture de classification issue des techniques de l'apprentissage profond et en particulier des réseaux neuronaux convolutifs et récurrents pour la classification de navires dans des images SAR. Plus particulièrement, il s'agit de classifier des navires par apprentissage supervisé à partir d'une base d'images SAR préalablement étiquetées. Il s'avère que l'un des principaux obstacles à l'utilisation de l'apprentissage profond est la nécessité d'avoir un très grand nombre de données annotées qui ne sont pas toujours disponibles. L'objectif de cette étude est donc de proposer une architecture hybride spécifique : CNN-LSTM (Convolutional Neural Networks - Long Short-Term Memory) permettant d'exploiter des ensembles de données frugales (peu nombreuses) et relativement déséquilibrés. Afin de comprendre l'intérêt de ce type de méthode, nous proposons de la comparer à des algorithmes d'apprentissage profond fréquemment décrits dans la littérature.

Abstract − This work proposes a classification architecture based on deep learning techniques and in particular convolutional and recurrent neural networks for the ships classification based on SAR images. More specifically, the objective is to classify ships using supervised learning from a database of previously labeled SAR images. It turns out that one of the main obstacles to the use of deep learning is the need to have a very large amount of annotated data which is not always available. Therefore, the objective of this study is to propose a specific hybrid architecture, CNN-LSTM (Convolutional Neural Networks - Long Short-Term Memory), allowing one to exploit frugal (few) and relatively unbalanced data sets. In order to understand the interest of this type of method, we propose to compare it to deep learning algorithms frequently described in the literature.

1 Introduction

L'imagerie radar à synthèse d'ouverture (SAR) offre un potentiel important dans le cadre de la surveillance maritime avec sa couverture mondiale ainsi que son indépendance vis-à-vis des conditions météorologiques. Afin de tirer parti de ce potentiel, l'apprentissage profond peut être utilisé pour traiter automatiquement de très grandes quantités de données dans diverses problématiques : détection [1]classification [2; 3], segmentation [4] ...

Il est apparu que les performances de telles architectures (CNN, LSTM ...) sont remarquables à condition de disposer d'un très grand volume de données annotées [5]. Pour évaluer leurs architectures, de nombreux travaux s'appuient généralement sur des benchmarks publics tels que ImageNet. Malheureusement, acquérir des données et les annoter sont des tâches relativement longues et difficiles, en particulier pour des images SAR. Face à cet obstacle du manque de données annotées, plusieurs techniques visant à augmenter artificiellement le volume de données ont été proposées comme le Transfert Learning [6; 7] ou encore les Generative Adversarial Network [8].

Dans cette étude, nous proposons d'exploiter le potentiel d'analyse et d'extraction offert par le CNN combiné avec une architecture LSTM capable de mémoriser, de prédire et de capturer des dépendances à long terme dans les données. Ce type de combinaison n'est pas nouvelle et existe pour la vidéo [9], le texte [10], ...A noter le travail de Wang *et al.* [11] sur des images SAR de la base MSTAR qui propose d'exploiter un réseau CNN-LSTM couplé avec un module d'attention dans le cadre d'une problématique multi-vues de cibles terrestres immobiles. L'originalité de l'approche décrite ici repose d'une part sur le développement d'une architecture hybride relativement différente (absence d'un module d'attention, structures différentes et optimisations des paramètres et hyper-paramètres du réseau) et, d'autre part la réalisation de la fonction de reconnaissance de navires à partir d'une base frugale d'images radar.

Ce travail porte sur l'utilisation d'une architecture hybride CNN-LSTM pour la classification de navires dans des images radar SAR. La section 2 décrit les spécificités de l'architecture et propose une méthodologie permettant de déterminer certains paramètres optimaux du réseau. Dans la section 3, après une brève description de la base OpenSARShip, un aperçu du potentiel de l'architecture est donné en la comparant à des méthodes classiques. La dernière section propose de conclure et offre quelques perspectives.

2 Architecture CNN-LSTM

2.1 Description de l'architecture

Dans le cadre de ce papier, une méthode hybride est développée afin d'identifier le type de navires à partir d'images SAR. La figure 1 illustre l'architecture conçue en combinant deux réseaux : Convolutional Neural Networks (CNN) et Long Short-Term Memory (LSTM). Le CNN

est utilisé pour extraire les caractéristiques complexes des images et le LSTM est utilisé comme classifieur. La description de l'architecture est décrite dans le tableau 2.

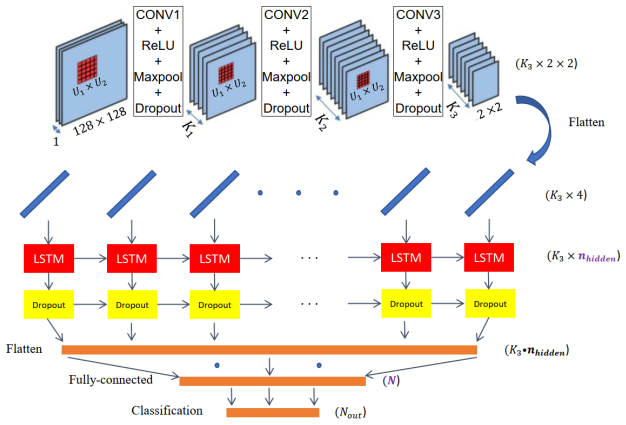

FIGURE 1 – Architecture du CNN-LSTM.

Le CNN est un réseau classique de type *feed-forward* régie par une architecture composée de couches de convolutions (convolution layers - CONV), de couches de réduction (pooling layers - POOL), de couches de correction (activation functions de type ReLU) et d'une couche entièrement connectée (fully-connected layers - FC). Ce type de réseau connaît de larges applications et offre de nombreux avantages [12]. Cependant, il nécessite une grande quantité de données d'entrées annotées et le nombre de paramètres à optimiser peut vite devenir très important. Dans ce travail, nous avons choisi de connecter/transférer la dernière couche convolutive à la partie LSTM de l'architecture afin d'identifier les dépendances entre les différentes caractéristiques.

Dans [13], les auteurs proposent un moyen de contourner les problèmes de disparition/explosion du gradient pour permettre aux réseaux récurrents d'apprendre les dépendances à long terme en introduisant un mécanisme de contrôle : la mémoire à long et court terme (Long Short-Term Memory - LSTM). Elle possède un état cellulaire, une sorte de mémoire interne dans laquelle de nouveaux contenus sont ajoutés et d'anciens contenus sont oubliés à chaque pas de temps. La sortie de la couche LSTM est de K_3 vecteurs de longueur n_{hidden}, où n_{hidden} est la taille de l'état caché, qui sera optimisé lors de la sélection du modèle. Afin d'éviter un sur-apprentissage, des couches d'abandon (dropout layers) de 50% sont appliquées aux sorties des couches cachées, puis une couche FC avec N neurones relie l'état caché à la couche FC de la fonction Softmax. Enfin, cette couche FC finale est utilisée pour prédire les catégories N_{out} présentées dans l'ensemble de données à classifier.

2.2 Élaboration du modèle CNN-LSTM

A partir d'une configuration initiale, il s'agit de déterminer la configuration des paramètres de l'architecture CNN-LSTM les plus 'pertinents' au regard de la problématique et de la base frugale OpenSARShip (Section 3.1). Pour faciliter l'élaboration de l'architecture, la taille des noyaux convolutifs est supposée la même pour toutes les couches CONV. Les masques dont la taille est comprise entre (2,2) et (28,28) sont testés (Figure. 2). Il apparaît que les paramètres optimaux pour (U_1, U_2) sont proches de (18, 18).

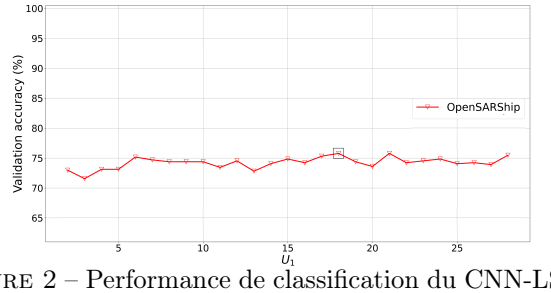

FIGURE 2 – Performance de classification du CNN-LSTM sur l'ensemble de validation en fonction de la taille des noyaux convolutifs (U_1, U_2) avec U_1=U_2, (K_1, K_2, K_3)=(64, 64, 128), n_{hidden} = 128 et N = 128.

Le Tableau 3 présente la précision (Validation accuracy) du modèle sur la base de validation en fonction de (K_1, K_2, K_3). Les hyperparamètres offrant la meilleure précision sont alors retenus.A partir des hyperparamètres identifiés précédemment, le tableau 4 présente la précision du modèle pour différentes valeurs de n_{hidden} correspondant à la taille de l'état caché de la couche FC. Le tableau 5 présente quand à lui la précision en fonction du nombre de neurones (N) de la couche FC. Finalement, les paramètres optimaux pour le réseau CNN-LSTM sont récapitulés dans le tableau 1.

TABLE 1 – Paramètres optimaux du CNN-LSTM proposé pour la base OpenSARShip et performances de validation.

Paramètres optimums				Validation
(U_1,U_2)	(k_1,k_2,k_3)	n_{hidden}	N	accuracy (%)
(18,18)	(64,64,128)	128	128	75.78

3 Résultats et discussions

3.1 OpenSARShip et pré-traitement

La base OpenSARShip [14] est utilisée dans la littérature scientifique récente pour l'évaluation des algorithmes de classification sur des bases frugales d'images SAR [15]. Cet ensemble de données est composé d'imagettes SAR annotées de navires, extraites d'images produites par les satellites Sentinel-1. Les caractéristiques des navires sont variables et présentent un nombre très variable d'instances par classes, ainsi qu'une grande variabilité dans les dimensions et les résolutions des images. La base comprend 5673 objets de 68 classes différentes. A noter que certaines classes ont un très petit nombre d'instances ne permettant pas d'effectuer un apprentissage profond efficace et suffisant pour extraire des informations pertinentes. Pour surmonter ce problème, il est généralement admis de ne conserver que les classes (navires) les plus représentées

[7; 15]. Dans le cadre de cette étude, nous retenons trois classes : *Cargo, Bulk Carrier, Container Ship* issues d'une acquisition en polarisation VV (Ground Range Detected - GRD) et ayant une taille minimale de 70×70 pixels afin de garantir une résolution minimale [15]. Pour définir le modèle il est nécessaire de manipuler des images ayant la même taille (input size). Les images sont ainsi toutes re-dimensionnées en 128×128 pixels.

3.2 Simulations et Résultats

Pour l'évaluation des algorithmes d'apprentissage profond, les données sont divisées en trois sous-ensembles : une base d'entraînement, une base de validation et une base de test. Le Tableau 6 présente la répartition des trois classes de navires dans les trois sous-ensembles.

Dans le tableau 7, les performances du réseau hybride CNN-LSTM sont présentées et comparées avec des architectures classiques de la littérature. Le tableau 7 montre que la plupart des architectures existantes fournissent une précision nettement inférieure à celle de l'architecture hybride proposée. Cet écart de performance peut s'expliquer par le fait que les architectures convolutives profondes ne sont pas adaptées pour la base OpenSARShip dont le nombre d'instances par classe est relativement faible et déséquilibré. Par ailleurs, le réseau VGG16 présente la meilleure précision mais au prix d'un nombre de paramètres entraînables beaucoup plus important (134.27M paramètres). En comparant ce dernier avec la solution proposé, VGG16 nécessite 21 fois plus de paramètres entraînables et avec un temps d'entraînement 5 fois plus important pour un gain de précision autour de 1%. Ainsi, le réseau hybride CNN-LSTM permet d'atteindre une précision relativement intéressante tout en améliorant de manière significative la précision du modèle sur les différentes classes, et surtout en réduisant le nombre de paramètres entraînables. De plus, il s'avère que ce type de réseau hybride peut rapidement converger, ce qui est bénéfique en termes de temps d'apprentissage sans affecter la précision de la prédiction.

4 Conclusions

Dans cet article, une architecture hybride CNN-LSTM est proposée et ses performances sont analysées dans le cadre de la classification de navires à partir de la base OpenSARShip qui présente peu de données. Afin d'optimiser le choix de certains hyper-paramètres, un ensemble de simulations est réalisé afin de sélectionner la configuration adéquate et optimale pour obtenir la meilleure performance de généralisation possible. Les performances du CNN-LSTM sont comparées à plusieurs architectures fréquemment utilisées dans le domaine du traitement des images optiques. Il s'avère que les résultats en terme de performances de classification et de temps d'apprentissage, taille de modele (nombre de paramètres) montrent le réel intérêt de remplacer les couches denses classique des CNN par une couche LSTM pour la classification des images SAR. Il serait intéressant de valider cette approche dans le cadre d'autre base comme FUSAR-Ship. Pour aller plus loin, l'utilisation d'architectures plus complexes pourrait être étudiée. En particulier, l'utilisation de ces architectures combinées avec un mécanisme d'attention pourrait être utilisées afin de sélectionner les parties pertinentes des images et orienter le réseau lors de la phase d'extraction de caractéristiques.

Références

[1] M. Yasir *et al.*, "Ship detection based on deep learning using SAR imagery : a systematic literature review," *Soft Computing*, vol. 27(1), 2023.

[2] H. Parikh *et al.*, "Classification of SAR and polSAR images using deep learning : A review," *International Journal of Image and Data Fusion*, vol. 11(1), 2020.

[3] A. Toumi *et al.*, "A proposal learning strategy on CNN architectures for targets classification," in *Int. Conf. ATSIP*, 2022.

[4] Z. Sun *et al.*, "Review of road segmentation for SAR images," *MDPI, Remote Sensing*, vol. 13(5), 2021.

[5] M. Najafabadi *et al.*, "Deep learning applications and challenges in big data analytics," *Springer Journal of big data*, vol. 2(1), 2015.

[6] Z. Huang *et al.*, "Classification of large-scale high-resolution SAR images with deep Transfer Learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 18(1), 2020.

[7] D. Zhang *et al.*, "Transfer Learning with convolutional neural networks for SAR ship recognition," in *IOP Conference Series*, vol. 322(7), 2018.

[8] I. Goodfellow *et al.*, "Generative Adversarial Networks," *Communications of the ACM*, vol. 63(11), 2020.

[9] S. Li *et al.*, "Unsupervised variational video hashing with 1d-CNN-LSTM networks," *IEEE Transactions on Multimedia*, vol. 22(6), 2019.

[10] A. Yenter *et al.*, "Deep CNN-LSTM with combined kernels from multiple branches for imdb review sentiment analysis," in *IEEE Int. Conf. UEMCON*, 2017.

[11] C. Wang *et al.*, "Multiview attention CNN-LSTM network for SAR automatic target recognition," *IEEE Journal Applied Earth Observations and Remote Sensing*, vol. 14, 2021.

[12] J. Gu *et al.*, "Recent advances in Convolutional Neural Networks," *Elsevier Pattern recognition*, vol. 77, 2018.

[13] S. Hochreiter *et al.*, "Long Short-Term Memory," *Neural Computation*, vol. 9(8), 1997.

[14] L. Huang *et al.*, "OpenSARShip : a dataset dedicated to Sentinel-1 ship interpretation," *IEEE Journal, Applied Earth Observations and Remote Sensing*, vol. 11(1), 2018.

[15] Z. Huang *et al.*, "What, Where, and How to Transfer in SAR target recognition based on Deep CNNs," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58(4), 2020.

TABLE 2 – Description du réseau hybride CNN-LSTM.

Layer	Type	Kernel	Kernel size	Stride	Input size
1	Convolution2D	K_1	$U_1 \times U_2$	1	$1 \times 128 \times 128$
2	Pool	-	4×4	4	$K_1 \times 128 \times 128$
3	Convolution2D	K_2	$U_1 \times U_2$	1	$K_1 \times 32 \times 32$
4	Pool	-	4×4	4	$K_2 \times 32 \times 32$
5	Convolution2D	K_3	$U_1 \times U_2$	1	$K_2 \times 8 \times 8$
6	Pool	-	4×4	4	$K_3 \times 8 \times 8$
7	LSTM	-	-	-	$K_3 \times 4$
8	FC	N	-	-	$K_3.n_{hidden}$
9	Softmax	N_{out}	-	-	N

TABLE 3 – Performance de classification du CNN-LSTM sur l'ensemble de validation en fonction de (K_1, K_2, K_3) dans les couches CONV avec (U_1, U_2) optimisées et n_{hidden} = 128, N=256.

	(K_1, K_2, K_3)							
	(32,32,64)	(32,64,64)	(64,64,128)	(64,128,128)	(128,128,256)	(128,256,256)	(256,256,512)	(256,512,512)
Accuracy (%)	74.53	74.06	**75.78**	74.84	74.84	74.53	74.84	75.00

TABLE 4 – Performance de classification du CNN-LSTM sur l'ensemble de validation en fonction de (n_{hidden}) avec les hyperparamètres optimisées (U_1, U_2) et (K_1, K_2, K_3) et avec N=128.

	n_{hidden}					
	32	64	96	128	160	192
Accuracy (%)	75.31	75.47	73.91	**75.78**	74.69	74.84

TABLE 5 – Performance de classification du CNN-LSTM sur l'ensemble de validation en fonction de (N) avec les hyperparamètres optimisées (U_1, U_2), (K_1, K_2, K_3) et n_{hidden}.

	N					
	64	128	184	256	320	384
Accuracy (%)	75.32	**75.78**	75.11	74.98	75.21	74.84

TABLE 6 – Nombre d'instances par classe dans les trois sous-ensembles résultant de la division des données d'OpenSARShip.

	Training 80% Entire Training	Validation 20% Entire Training	Entire Training 80% Dataset	Test 20% Dataset
Cargo	99	25	124	31
Bulk Carrier	335	84	419	105
Container Ship	104	26	130	33

TABLE 7 – Comparaison des performances du réseau CNN-LSTM proposé avec des architectures existantes sur la base OpenSARShip.

Architecture	# of parameters	Training time (s)	Number of epochs	Test loss	Test accuracy (%)
VGG16	134.27M	718.57	270	**2.0185**	**72.19**
ResNet50	23.51M	3502.18	1854	5.8233	57.99
Xception	20.81M	1958.26	978	4.1148	65.09
DenseNet121	6.96M	5578.21	1447	8.4319	56.80
EfficientNetB0	4.01M	254.74	**111**	2.9316	52.07
MobileNetV2	2.23M	621.78	398	2.5883	56.21
Proposed CNN-LSTM	6.17M	**132.61**	166	2.5124	**70.41**