



HAL
open science

Comparative performances of CNN models for SAR Targets classification

Abdelmalek Toumi, Jean-Christophe Cexus, Ali Khenchaf

► **To cite this version:**

Abdelmalek Toumi, Jean-Christophe Cexus, Ali Khenchaf. Comparative performances of CNN models for SAR Targets classification. 2024 IEEE 7th International Conference on Advanced Technologies, Signal and Image Processing (ATSIP), Jul 2024, Sousse, Tunisia. pp.122-127, 10.1109/ATSIP62566.2024.10638905 . hal-04805277

HAL Id: hal-04805277

<https://hal.science/hal-04805277v1>

Submitted on 2 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Comparative performances of CNN models for SAR Targets classification

Abdelmalek Toumi, Jean-Christophe Cexus, Ali Khenchaf

ENSTA Bretagne - Lab-STICC, UMR CNRS 6285

2, Rue François Verny, 29806 Brest Cedex 9, France.

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

Abstract—This article focuses on the development of classification architectures for Synthetic Aperture Radar (SAR) image analysis, particularly for target recognition. Several methods based on deep artificial neural networks are explored and compared. Specifically, this work investigates the study of pretrained architectures such as Xception and DenseNet, which were originally developed and trained on the ImageNet database containing optical images. However, adaptation of these architectures is necessary within the context of SAR radar images. The goal was to take advantage of the powerful feature extraction capabilities of these models to effectively classify objects in radar images. We evaluated and compared the classification performances of each technique using the publicly MSTAR dataset. This led to the proposal and development of new architectures based on the Xception and DenseNet models. Using these models, it is possible to achieve impressive recognition rates close to 99.5% on the test dataset, surpassing several benchmarks reported in the scientific literature on the same dataset.

Keywords—Automatic Target Recognition, Deep Learning, SAR classification, Transfer Learning, Fine Tuning.

I. INTRODUCTION

The identification and classification of radar targets from SAR images offer a significant advantage over the challenges humans face when analyzing such images, unlike conventional images, especially optical ones. This difficulty stems from the characteristics of radar images, the observed scenes within uncontrolled environments, their resolution, and the limited color dynamics. Consequently, there is considerable interest in categorization and object detection within the field of remote sensing [1]-[5].

Our objective is to leverage SAR data to design a recognition architecture based on existing deep learning models. However, the unique characteristics of SAR image databases pose challenges in directly applying established models such as DenseNet121 [6], Xception [7], ResNet [8], MobileNet [9], and other which are predominantly developed for optical images like those in ImageNet [10]. Adapting these models for SAR images often requires significant modifications, potentially altering their architectures based on the desired performance for recognition applications [11]. Previous literature has addressed this issue by proposing dedicated architectures for SAR image tasks, including classification, detection, and classification, or detection alone. Such research often draws inspiration from and compares with architectures developed for various image types, including classical or optical images.

In this context, our focus lies on developing, implementing and comparing neural networks customized for SAR image processing. We have also endeavored to enhance existing

models from scientific literature to better suit radar image analysis.

In Section II, we explain the potential of transfer learning and fine-tuning strategies applied to the CNN architectures proposed. Here, we present the CNN architectures proposed for target recognition on SAR images, along with the experimentation conducted to evaluate them. Section III discusses the obtained results and performances of the proposed methods. Finally, we conclude with our findings and outline future perspectives.

II. MACHINE LEARNING METHODS FOR TARGET RECOGNITION

This section quickly presents the different architectures used in this study: The Multilayer Perceptron (PM) and several Convolutional Neural Network (CNN) models.

A. Multilayer Perceptron (PM)

A Multilayer Perceptron (PM) is a type of artificial neural network that consists of multiple layers of interconnected neurons. PM are powerful models capable of learning complex patterns and relationships in data. They are widely used in various fields, including pattern recognition, classification, regression, and time series prediction.

Despite their effectiveness, PM have certain limitations. They require a large amount of labeled data for training, and their performance et memory resource depends heavily on the choice of hyperparameters such as number of layers, number of neurons per layer, and activation functions. Additionally, MLPs are prone to overfitting, especially when dealing with noisy or small datasets.

TABLE I. ARCHITECTURE OF THE PM MODEL.

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 88, 88, 1)]	0
flatten_1 (Flatten)	(None, 7744)	0
dense_1 (Dense)	(None, 1024)	7930880
dropout (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 512)	524800
dropout_1 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 10)	5130
Total params: 8,460,810		
Trainable params: 8,460,810		
Non-trainable params: 0		

In summary, PMs are a fundamental building block of machine learning and have proven to be effective models for a wide range of tasks. However, they require careful tuning

and consideration of various factors to achieve optimal performance. In this study, the PM is used as the minimum reference with a recognition rate of 93.90% based on the tests. Its architecture is presented in Table I and the classification results in Section III.

B. Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is other type of artificial neural network specifically designed for images. CNNs are particularly effective in tasks involving image recognition, object detection, and classification. One of the key advantages of CNNs is their ability to automatically learn hierarchical representations of data. By exploiting the spatial correlations present in images, CNNs can learn to extract meaningful features at different levels of abstraction, leading to superior performance compared to traditional machine learning methods.

CNNs are characterized by their architecture, which consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers apply convolution operations to the input data, which involves sliding a filter (also known as a kernel) across the input image to extract local features. These features are then passed through activation functions to introduce non-linearity into the network.

In this study, we build upon the framework developed by C. Coman *et al.* [12], which introduces a specific architecture designed for MSTAR data, denoted as CNN¹. Its architecture is presented in Table II and the classification results in Section III.

TABLE II. ARCHITECTURE OF THE CNN1 MODEL.

Layer (type)	Output Shape	Param #
conv2d_24 (Conv2D)	(None, 86, 86, 32)	320
conv2d_25 (Conv2D)	(None, 84, 84, 32)	9248
max_pooling2d_1 (MaxPooling2)	(None, 42, 42, 32)	0
dropout_1 (Dropout)	(None, 42, 42, 32)	0
flatten_1 (Flatten)	(None, 56448)	0
dense_2 (Dense)	(None, 128)	7225472
dense_3 (Dense)	(None, 10)	1290
Total params: 7,236,330		
Trainable params: 7,236,330		
Non-trainable params: 0		

In summary, Convolutional Neural Networks are a powerful class of models for image processing tasks, capable of learning complex patterns and representations directly from raw pixel data. Their effectiveness, versatility, and scalability have made them the go-to choose for a wide range of computer vision applications.

Subsequently, it is a question of studying to what extent less specific CNN architectures can offer equivalent or better recognition rates. For this, we studied 4 CNN architectures well known in the literature:

- Xception model [7]: is a deep CNN architecture introduced in 2017 by Chollet (creator of Keras, a popular machine learning library in Python). The

main advantage of the Xception architecture lies in its innovative approach to convolution. Instead of using traditional convolution filters, Xception uses deep convolution modules called “Depthwise Separable Convolutions”. These modules help reduce the number of parameters while maintaining network performance, which can lead to more computationally and memory efficient models.

- Inception ResNetV2 [13]: is a deep CNN architecture that combines the concepts of the Inception architecture [14] with the residual blocks of ResNet [8]. It was introduced by Google as part of the TensorFlow project. This architecture was designed to achieve better performance in image classification, object detection and semantic segmentation tasks. It is renowned for its ability to extract complex features at different resolution levels, making it a popular choice for many computer vision applications.
- MobileNetV2 [9]: is a deep CNN architecture designed to run efficiently on mobile devices by reducing the number of parameters and calculations. It builds upon an original architecture, aiming to improve both performance and efficiency. It is based on a structure inverted residual where the convolution layers are placed between the linear activation layers.
- DenseNet121 [6]: is a deep CNN architecture proposed by Huang *et al.* in 2017. It connects each layer to all previous layers in a dense manner, which promotes maximum information flow through the network. This architecture allows better use of the features learned at different spatial scales.

To do this, it involves modifying and improving the behavior of the 4 architectures via two techniques which differ in their approach: Fine-Tuning method and Transfer Learning method. The goal of fine-tuning and transfer learning is to exploit pre-trained neural network models to improve their performance for new problems or types of data (here SAR images: MSTAR database – section III-A). In both cases, the goal is to take the knowledge gained from a pre-trained model on a large general data set and apply it to a new related task (Table III). For all architectures, the weights of the convolutional layers of these architectures are those optimized on the ImageNet database.

TABLE III. SUMMARY KEY DIFFERENCES BETWEEN TRANSFER LEARNING AND FINE-TUNING.

Feature	Transfer Learning (TL-)	Fine-Tuning (FT-)
Objective	Leverage pre-trained knowledge for a new task	Further specialize a pre-trained model
Layers Updated	Early layers frozen, final layers retrained	More layers (potentially all) updated
Training Time	Faster	Slower
Data Requirement	Less data needed	May require more data

1) Transfer Learning method on architectures (TL-)

This method involves transferring knowledge learned from one task or data set to another related task or data set. By

initializing the model with weights learned from a pre-trained model, transfer learning can speed up training and improve generalization, especially when the target dataset is small.

A pre-trained model, such as an ImageNet-trained Xception, ResNetV2... serves as the initial foundation. The lower layers of the model, responsible for extracting fundamental features like edges and shapes, are typically kept frozen (with weights unchanged) during training. Conversely, the upper layers, responsible for capturing higher-level, task-specific features, are retrained using a new dataset tailored to address your specific problem.

First, it is necessary to modify the initial architectures somewhat. Notably the last connected layer of all architectures, we process the 10 classes of the MSTAR database against 1000 classes used for example in Xception [7]. To do this, it is necessary to replace the last layer of the CNN architecture with a layer of 10 outputs whose weights are defined by learning on the MSTAR classes.

The other layers remain fixed with the optimized coefficients obtained by training on the ImageNet database and are used in the feature extraction. It is also possible to add a hidden layer well suited in size, but this technique was not retained in our study case because it did not allow us to improve performance. The learning results obtained are shown section III.

2) Fine Tuning method on architectures (FT-)

Fine-tuning involves adjusting the parameters of a pre-trained model using a smaller dataset specific to the new problem. This involves further specializing a pre-trained model for a specific task by adjusting its weights. This process allows the model to adapt its previously learned representations to better fit the new task, potentially achieving higher accuracy than training from scratch.

For training, it is often possible to freeze the initial layers of the CNN models, and only adapt the final layers for the new classification problem. Freezing all convolutional layers corresponds to the previously presented TL-models. In this phase of Fine-tuning the coefficients of the previous model will vary slightly. Indeed, we will use the TL-models obtained previously which already has optimized coefficients. It is therefore a question of modifying weakly at each iteration the weights of the last selected convolutional layers, to adapt it progressively to our database, without completely modifying the knowledge already acquired. We denote by FT-models the resulting architectures which integrates the training of the last two convolutional layers of the TL-models in addition to the final layer (classifier). The learning results obtained are shown section III.

3) Complete Training methods on architectures (CT-)

To analyze the performance limits of the architectures presented previously, we implemented the global training strategy only on the Xception model. Here all layers of the FT-Xception model are retrained, this means all weights are recalculated and optimized based on MSTAR database. We designate this completely reconstructed architecture as CT-

Xception. The initial coefficients during training are those of the FT-Xception model. The learning results obtained are shown section III.

III. RESULTS

This section presents a comprehensive analysis of the obtained results. We will evaluate the performance of the models across various evaluation criteria.

A. Data Description

This section focuses on applying various classification methods to the public MSTAR database¹. The MSTAR database serves as the foundation for our experiments. This publicly available dataset consists of 5172 grayscale images, each with a resolution of 128 x 128 pixels (see Table IV).

For training purposes, the MSTAR data is divided into a learning base and a test base. An important feature of the learning base is the balanced distribution of images across target classes. This ensures that the model receives an equal amount of training data for each target type, promoting balanced learning. The test base also exhibits a relatively balanced distribution across most target classes, with the exception of a few outliers. Figure 1 provides a visual overview of the MSTAR database's contents.

TABLE IV. NUMBER OF IMAGES PER CLASS (TARGET) IN THE LEARNING / TEST MSTAR DATABASE.

Classe	2S1	BMP2	BRDM2	BTR60	BTR70	D7	T62	T72	ZIL131	ZSU23	Total
Train (17 ⁶)	299	233	298	256	233	299	299	232	299	299	2747
Test (15 ⁶)	274	195	274	195	196	274	273	196	274	274	2425



Fig. 1. Samples of raw SAR images of some targets from the MSTAR database.

We compare the performance of our architectures and our results to those presented by other methods. This comparison aims to understand the complex and often unpredictable relationship between the data content and the chosen architecture. Oftentimes, we lack information regarding the distribution nature of the data in the feature space, which can be extensive (equivalent to the number of pixels in an image). Consequently, we resort to heuristic approaches when selecting the model (architecture) to ensure optimal performance.

¹ The Air Force Moving and Stationary Target Recognition Database (MSTAR). Available online: <https://www.sdms.afrl.af.mil/datasets/mstar/> (accessed on 14 March 2024).

TABLE V. RESULTS USING PRE-TRAINED ARCHITECTURES.

Pre-Trained Architectures	Recognition rate by class %						Total number of parameters	Number of convolutional layers (Depth)
	Transfert Learning (TL-)			Fine-Tuning (FT-)				
	Train	Test	Nbr-En	Train	Test	Nbr-En		
Xception	99,96	89,96	20 490	100	96,41	4 769 290	20 881 970	126
Inception ResNetV2	99,93	92,41	15 370	100	93,65	4 145 706	54 352 106	572
MobileNetV2	94,00	89,04	627 210	100	92,49	1 347 210	2 885 194	88
DenseNet121	99,98	88,00	10 250	100	90,44	176 394	7 047 754	121

TABLE VI. SUMMARY OF CLASSIFICATION RESULTS.

Models	Recognition rate by class % (test base)										Overall rate %
	251	BMP2	BRDM_2	BTR70	BTR_60	D7	T62	T72	ZIL131	ZSU_23_4	
PM	92,36	63,77	96,35	98,47	89,23	98,90	96,38	99,49	98,90	98,54	93,90
CNN ¹	100	84,10	100	93,36	96,92	99,27	98,53	100	100	100	97,69
CT-Xception	100	95,38	100	100	98,97	100	100	100	100	100	99,55
FT-Xception	98,17	69,23	100	96,43	96,92	98,17	99,63	98,98	99,63	100	96,41
FT-Inception ResNetV2	93,43	72,82	98,54	82,14	93,85	96,71	98,90	94,39	99,27	97,44	93,65
FT-MobileNetV2	93,06	70,26	97,44	85,20	85,13	95,98	98,53	95,92	99,63	94,16	92,49
FT-DenseNet121	88,32	57,95	96,35	88,77	80,00	92,70	95,24	92,35	98,54	98,54	90,44
TL-Xception	93,80	65,13	94,16	89,29	87,18	90,51	89,01	91,32	95,25	93,79	89,96
TL-Inception ResNetV2	94,53	65,64	98,17	88,26	91,28	93,35	94,50	93,37	97,08	95,98	92,41
TL-MobileNetV2	89,05	69,56	98,90	83,76	55,60	96,53	89,38	94,38	93,35	93,06	89,04
TL-DenseNet121	85,76	48,72	93,43	87,75	81,02	89,78	93,40	90,82	98,17	98,54	88,00

TABLE VII. COMPARISON OF THE SC-XCEPTION MODEL WITH OTHER MODELS FROM THE SCIENTIFIC LITERATURE.

Models	Recognition rate by class % (test base)										Overall rate %
	251	BMP2	BRDM_2	BTR60	BTR70	D7	T62	T72	ZIL131	ZSU_23_4	
SC-Xception (proposed method)	100	95.38	100	98.97	100	100	100	100	100	100	99.55
AM-CNN [15]	98.90	100	99.64	96.41	100	99.27	99.63	100	99.64	100	99.35
FCAB-CNN [16]	100	98.46	99.27	98.98	100	100	99.27	100	100	98.17	99.51
GoogleNet-APNB-ISEB [17]	99.80	99.50	99.80	99.20	100	99.30	100	99.80	99.80	99.40	99.72
CA-MCN [18]	99.27	100	99.64	99.49	100	99.27	99.63	100	99.64	100	99.59
MA-MobileNetV2 [19]	100	100	100	98.89	100	100	100	99.49	100	100	99.85
PA-DCNN [20]	97.40	99.80	99.30	95.90	100	99.30	99.30	99.90	99.30	99.60	99.50
Assembled-CNN [21]	100	99.49	98.90	100	93.30	98.90	99.63	98.90	100	100	99.05
ATR-CNN [22]	99.27	83.58	91.24	89.74	98.54	96.70	98.54	95.41	93.67	80.67	93.00

To evaluate recognition quality, we utilize metrics such as class-specific recognition rates and the Confusion Matrix based on the Test set. These metrics possess the capability to simultaneously highlight well-identified classes and visualize correlations among database classes. Additionally, we employ training curves to analyze the neural network's behavior, including its stability and convergence speed. This enables us to track and optimize the selection of hyperparameters to enhance generalization ability.

B. Comparison of Architectures

In this part, we present the summary of the results obtained and we give the performance of the models through different evaluation criteria. Tables V, VI and VII provide a summary of the results obtained on the MSTAR database.

In Table V, we present obtained results using other pretrained architectures with the same approach used for Xception architecture on connected layer (classifier) and the

last two convolutional layers. We denote by Nbr-En the number of trainable parameters. It can be seen from Table VI that the FT-Xception achieves the best rate on the test compared to other models.

The network PM shown a recognition rate of 93.90% on the test dataset. This interesting result provided by this simple network may be because the useful information is all centered in the SAR images. We note that the best results are obtained for the T72 class (Table VI).

As shown in the Table VI, the CNN¹ model achieves an average recognition rate of 97.69% (a gain of +3.79 compared to PM). We see that the CNN¹ architecture presents better performances compared to a classic PM. However, CNN¹ presents difficulties in classifying the BMP2 target which is strongly confused with the T72 and BTR70 targets (according to the confusion matrix not presented here due to lack of space).

The overall and per-class rates shown in Table VI are obtained based on the MSTAR test dataset (the overall rate is slightly different from the average per-class rate because the image number per target varies). We can see that all models (FT-models and TL-models) have difficulties in correctly classifying the BMP2 target which is generally confused with the BTR70 and T72 classes. Remember that the two classes BTR70 and BMP2 have the smallest number of images in the learning base, which can be one of the causes of this difficulty.

The comparisons presented in Table VII show the contribution of our approach SC-Xception and the improvement made compared to the best results obtained in the scientific literature dealing with the same MSTAR database.

The CT-Xception model achieved a very satisfactory overall and class recognition rate which favors the use of learning transfer. As a result, the overall training strategy leads to the best performance. These results confirm our hypotheses that complete training on specific databases such as SAR images (MSTAR base) can bring a real gain in terms of recognition of targets. Though, it involves optimizing all the weights of the pre-trained model as well as requires relearning for each problem treated. This relearning is unfortunately very costly in terms of time and requires a large volume of data, things that we often do not have available.

Additionally, another approach is to develop a tailored architecture that considers constraints related to learning time, generalization, and model complexity. In our study, the model selection was carried out following a heuristic approach which is often adopted by a large scientific community that makes it necessary to go through the experimentation of several models. Indeed, in the absence of a precise mathematical formalism that describes the decision-making mechanisms by these networks, the deterministic approach remains often difficult to put in place. This problem is rather studied by another field under development, that of the explainability of AI.

IV. CONCLUSIONS

This work tackled the challenge of automatic radar target recognition using machine learning techniques. The results obtained through studying various neural networks clearly demonstrated the effectiveness of these methods in

classification. A key focus of our approach was leveraging architectures developed in the scientific literature through transfer learning and fine tuning. We employed a methodical approach, first examining the performance of pre-trained models after adding a classifier. We then evaluated these models after fine-tuning their final convolutional layers specifically for the MSTAR dataset. Notably, the FT-Xception model achieved the best results within this fine-tuning stage. Finally, by completely training the FT-Xception model, we proposed the CT-Xception model. This final model exhibited excellent generalization capability, achieving a remarkable recognition rate on the test set, surpassing the best results reported in the literature. It's important to note that transfer learning in this work was performed using models pre-trained on the ImageNet database, which contains natural images.

We propose a potentially more efficient solution: transfer learning from pre-trained architectures on radar images themselves. For example, the CT-Xception model could be further exploited and adapted using this technique for recognizing radar targets in other databases or the robustness of the FT-models or CT-Xception model against Gaussian white noise.

REFERENCES

- [1] M. Yasir, *et al.*, "ShipGeoNet: SAR Image-based Geometric Feature Extraction of Ships using Convolutional Neural Networks," IEEE Transactions on Geoscience and Remote Sensing, 2024.
- [2] J.-C. Cexus, *et al.*, "Target Recognition from ISAR Image using Polar Mapping and Shape Matrix," Proceeding of the IEEE 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 2020.
- [3] A.-J. Gallego, *et al.*, "Automatic Ship Classification from Optical Aerial Images with Convolutional Neural Networks," Remote Sens., Vol. 10, no. 4, 2018.
- [4] Z. Huang, *et al.*, "Transfer Learning with Deep Convolutional Neural Network for SAR Target Classification with Limited Labeled Data," Remote Sensing, vol. 9, n° 9, 2017.
- [5] A. Karine, *et al.*, "Radar Target Recognition using Salient Keypoint Descriptors and Multitask Sparse Representation," Remote Sens., vol. 10, no. 6, 2018.
- [6] G. Huang, *et al.*, "Densely Connected Convolutional Networks," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- [7] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- [8] K. He, *et al.*, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- [9] M. Sandler, *et al.*, "Mobilenetv2: Inverted Residuals and Linear Bottlenecks," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
- [10] O. Russakovsky, *et al.*, "Imagenet Large Scale Visual Recognition Challenge," International Journal of Computer Vision, vol. 115, 2015.
- [11] A. Toumi, *et al.*, "A Proposal Learning Strategy on CNN Architectures for Targets Classification," Proceedings of the IEEE 6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 2022.
- [12] C. Coman, *et al.*, "A Deep Learning SAR Target Classification Experiment on MSTAR dataset," Proceedings of the IEEE 19th International Radar Symposium (IRS), 2018.
- [13] C. Szegedy, *et al.*, "Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning," Proceedings of the AAAI Conference on Artificial Intelligence, 2017.

- [14] C. Szegedy, *et al.*, "Rethinking the Inception Architecture for Computer Vision," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- [15] M. Zhang, *et al.*, "Convolutional Neural Network with Attention Mechanism for SAR Automatic Target Recognition," IEEE Geosci. Remote Sens. Lett. 2020.
- [16] R. Li, *et al.*, "SAR Target Recognition Based on Efficient Fully Convolutional Attention Block CNN," IEEE Geosci. Remote Sens. Lett. 2022.
- [17] Z. Wang, *et al.*, "Land-Sea Target Detection and Recognition in SAR Image Based on Non-Local Channel Attention Network," IEEE Trans. Geosci. Remote Sens. 2022.
- [18] H. Xu, *et al.*, "Multi-Scale Capsule Network with Coordinate Attention for SAR Automatic Target Recognition," Proceedings of the 7th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR), 2021.
- [19] T. Tang, *et al.*, "Vehicle Target Recognition in SAR Images with Complex Scenes Based on Mixed Attention Mechanism," Information, vol. 15, no 3, 2024.
- [20] Y. Zhong, *et al.*, "Enlightening Deep Neural Networks with Knowledge of Confounding Factors," Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), 2017.
- [21] Z. Huang, *et al.*, "Transfer Learning with Deep Convolutional Neural Network for SAR Target Classification with Limited Labeled Data," Remote Sens. Vol. 9, no. 9, 2017.
- [22] A. El Housseini, *et al.*, "Deep Learning for Target Recognition from SAR Images," Proceedings of the Seminar on Detection Systems Architectures and Technologies (DAT), 2017.