ISPRS Journal of Photogrammetry and Remote Sensing 218 (2024) 312–323

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Generalization in deep learning-based aircraft classification for SAR imagery

Andrea Pulella [a,b,*], Francescopaolo Sica [b], Carlos Villamil Lopez [a], Harald Anglberger [a], Ronny Hänsch [a]

[a] Microwaves and Radar Institute, German Aerospace Center (DLR), Münchener Str. 20, Weßling, 82234, Germany
[b] Institute of Space Technology and Space Applications, University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, Neubiberg, 85579, Germany

ARTICLE INFO

Keywords:
Synthetic Aperture Radar (SAR)
Automatic Target Recognition (ATR)
Object classification

ABSTRACT

Automatic Target Recognition (ATR) from Synthetic Aperture Radar (SAR) data covers a wide range of applications. SAR ATR helps to detect and track vehicles and other objects, e.g. in disaster relief and surveillance operations. Aircraft classification covers a significant part of this research area, which differs from other SAR-based ATR tasks, such as ship and ground vehicle detection and classification, in that aircrafts are usually a static target, often remaining at the same location and in a given orientation for longer time frames. Today, there is a significant mismatch between the abundance of deep learning-based aircraft classification models and the availability of corresponding datasets. This mismatch has led to models with improved classification performance on specific datasets, but the challenge of generalizing to conditions not present in the training data (which are expected to occur in operational conditions) has not yet been satisfactorily analyzed. This paper aims to evaluate how classification performance and generalization capabilities of deep learning models are influenced by the diversity of the training dataset. Our goal is to understand the model's competence and the conditions under which it can achieve proficiency in aircraft classification tasks for high-resolution SAR images while demonstrating generalization capabilities when confronted with novel data that include different geographic locations, environmental conditions, and geometric variations. We address this gap by using manually annotated high-resolution SAR data from TerraSAR-X and TanDEM-X and show how the classification performance changes for different application scenarios requiring different training and evaluation setups. We find that, as expected, the type of aircraft plays a crucial role in the classification problem, since it will vary in shape and dimension. However, these aspects are secondary to how the SAR image is acquired, with the acquisition geometry playing the primary role. Therefore, we find that the characteristics of the acquisition are much more relevant for generalization than the complex geometry of the target. We show this for various models selected among the standard classification algorithms.

1. Introduction

Synthetic Aperture Radar (SAR) is an active remote sensing technology that uses microwaves to create images of the Earth's surface. Due to its ability to see through clouds, SAR is one of the preferred sensor types in situational awareness applications (Roemer et al., 2016; Pulella and Sica, 2021). Automatic Target Recognition (ATR) refers to the automated detection and classification of objects in imagery and is a specific case of situational awareness. In the SAR context, ATR involves the use of image analysis techniques to identify targets such as vehicles, buildings, or other objects of interest based on the characteristics of the SAR backscatter signature. The variety of SAR ATR applications is wide, from detecting and tracking vehicles and similar targets to supporting disaster relief and surveillance (El-Darymli et al., 2016). The latter includes, for example, the detection and tracking of ships at sea, which is essential for maritime security.

SAR ATR was first applied to high-resolution SAR in a military context. The Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset (Diemunsch and Wissinger, 1998) was created by the Defense Advanced Research Projects Agency (DARPA) Air Force Research Laboratory in 1998 to promote and evaluate progress in SAR ATR algorithm development. The dataset consists of a collection of ten military vehicles acquired from different angles. Classical algorithms are mostly based on simple image processing operations such as template matching e.g. as presented in O'Sullivan et al. (2001) and in Srinivas et al. (2014). More recent work uses machine learning (ML) and, in particular, deep learning (DL), usually in the framework

* Corresponding author at: Microwaves and Radar Institute, German Aerospace Center (DLR), Münchener Str. 20, Weßling, 82234, Germany.
E-mail addresses: Andrea.Pulella@dlr.de (A. Pulella), Francescopaolo.Sica@unibw.de (F. Sica), Carlos.VillamilLopez@dlr.de (C. Villamil Lopez), Harald.Anglberger@dlr.de (H. Anglberger), Ronny.Haensch@dlr.de (R. Hänsch).

https://doi.org/10.1016/j.isprsjprs.2024.10.030
Received 31 January 2024; Received in revised form 12 July 2024; Accepted 29 October 2024
Available online 8 November 2024

of supervised learning, i.e. approaches that leverage training datasets of annotated images before being tasked to detect and classify these objects within new images. Examples of these algorithms include Radial Based Function (RBF) neural networks (Sun et al., 2007), Multilayer Perceptrons (MLPs) (Principe et al., 1998), and Support Vector Machines (SVMs) (Zhao and Principe, 2001; Wagner, 2016). In 2015 first results using a deep convolutional neural network (CNN) on the MSTAR dataset were presented (Morgan, 2015). Since then, the number of works using deep learning-based approaches for SAR ATR has increased dramatically. In Chen et al. (2016), the authors presented a sparsely connected layer network that automatically generates features by learning hierarchical representations from SAR images. In Soldin (2018), the authors analyze the effects that different classifier methodologies based on a ResNet-18 have on the performance of the existing and emerging targets. In particular, they consider (1) a scratch model that retrains all layers of the base CNN architecture, (2) a fine-tune model that freezes all layers in the ResNet-18 except the softmax, and (3) a dynamic image model that adds to the fine-tune model an intermediate layer based on the image split into two sub-apertures for data augmentation. All three models are trained on the 10 MSTAR classes and results show that the usage of dynamic imaging provides the best performance. Gu et al. show in Gu et al. (2021) that the usage of an existing neural network, such as the VGG16, already guarantees accuracies of 90% for three military vehicles of the MSTAR with similar appearances. In Jacob et al. (2023), the authors compare the performance of different standard networks, such as InceptionV3, VGG16, VGG19, ResNet50, and MobileNet. The analysis evaluation metrics used during the analysis are the confusion matrix, precision, recall, F1-Score, and mean average precision (mAP). As expected, all the models return good performance when using the MSTAR dataset. VGG16 and MobileNet models appear the best CNNs for this classification task. More sophisticated approaches using CNNs include the Multiple Feature-based CNN (MFCNN) (Cho and Park, 2018), a network capable of detecting targets without applying prior noise suppression by aggregating strong features with high noise levels and smoothed features with lower noise effects. In Pei et al. (2018), the authors address the issue of multiple views in SAR data. A multi-input CNN is presented that learns and extracts classification information from the multi-view images acquired by the SAR platform in different view intervals (different elevation and aspect angles). The advantage of this approach is that it requires only a small number of raw SAR images to generate training samples. The improvement of using multiple angles over a single angle is also demonstrated in Wang et al. (2021), where the authors present a multiview attention convolutional neural network with long short-term memory (LSTM) network to extract and fuse the features from images with adjacent azimuths. Convolutional highway networks are alternative solutions for guaranteeing robust training, by retrieving deeper features from reduced training datasets (Lin et al., 2017). Attention mechanism networks can be used as a solution to capture more valuable information and reduce the computational burden. First studies on attention-based real-valued CNNs using SAR data are reported in Zhang et al. (2020a) and Li et al. (2022). In Lang et al. (2022), the authors design a cascaded multidomain attention module, based on discrete cosine transform and discrete wavelet transform embedded in a four-layer CNN model to perform hierarchical feature representation learning and to further complete the class-specific feature extraction from both the frequency and wavelet transform domains of the input feature maps. The incorporation of multi-domain attention enhances the feature extraction capability and effectively improves the recognition accuracy of the CNN. Capsule networks are designed to replace the traditional scalar output neurons of a CNN with vectors that output multiple values and, as a consequence, improve object recognition in images at the expense of a larger amount of computation. In Shah et al. (2019), the authors present the first capsule network for SAR ATR. It consists of a convolutional layer and two capsule layers, which can be trained on a smaller dataset than that required by a traditional CNN, with a higher accuracy than 98% using MSTAR data. In Guo et al. (2020), the authors suggest a capsule network for high-accuracy recognition based on a vector-based full-connected operation. Results show a robustness greater than a traditional CNN. An advanced version of the capsule network is reported in Ren et al. (2021). Multiple dilated convolutions are used to extract multi-scale features in the encoder network, and refinements are used to extract discriminative features by adaptively highlighting informative features. The correlation among multimodal radar data is crucial for improving algorithm accuracy. In Feng et al. (2021) the authors present a decision fusion framework based on target parts divided according to a set of attributed scattering center (ASC) parameters. The main disadvantage of this solution lies in the simple form of the fusion approach, which measures the average of the predictions retrieved from different DL methods and model-based approaches. A solution proposed in Feng et al. (2022, 2023) is to apply a physics-based approach consisting of the fusion of multiple network layer features to inform deep recognition networks. In Zhang et al. (2024), the authors suggest a multi-scale feature approach that fuses scattering features and deep features by weighted integration to enrich the diversity of features. In Shi (2022), the author presents a multi-feature fusion-based approach to capture target shape, corner features, and texture. In particular, the author uses the Hu moment to describe the shape of the SAR targets, the Harris corner point to extract the corners of the object, and the Gabor features for texture analysis. All the feature descriptors are given as input to three conventional classifiers: Decision Tree, SVM, and MLP. Overall the study shows that the Decision Tree algorithm attained the highest recognition accuracy. Additionally, several studies on the role of data augmentation (Ding et al., 2016), imbalance loss (Zhang et al., 2020c; Cao et al., 2022) structured pruning (Zhang et al., 2020b), and transfer learning (Zhang et al., 2020b; Zhong et al., 2019; Huang et al., 2020; Song et al., 2022; Shang et al., 2018; Chen et al., 2022) have been conducted.

While most of these algorithms have been trained and tested on the MSTAR dataset, some new datasets have been released for SAR ATR, e.g. datasets consisting of very-high-resolution SAR images of the Chinese C-band Gaofen-3 (Guo et al., 2019). However, the number of deep learning-based algorithms dedicated to this task is still much larger than the number of available datasets. This aspect has led to the fact that, although the detection and classification capabilities under specific operational conditions have increased, the generalization to larger and more diverse datasets still needs to be adequately investigated.

Generalization refers to the ability of a trained model to accurately classify or recognize new and previously unseen examples that belong to the same classes as those present in the training data, but have a slightly different appearance. Reasons for a lack in generalization include overfitting of the model, i.e. that it has learned to memorize specific examples rather than learning generalizable patterns or features that can be applied to new data. Another cause is that the model might have learned shortcuts, i.e. spurious correlations that are present in the training data but not during deployment (in our case, for example, when airplanes are parked at the same positions and the model learns to recognize the background/surrounding rather than the object). Alternatively, it can happen when the training data is not representative of the real-world examples the model is expected to encounter, which has enormous implications when SAR ATR is to be used in operational scenarios.

The goal of this paper is to study the classification performance in terms of the generalization ability of deep learning models. Specifically, we aim to understand how well and under what circumstances a given trained model can

- be successfully applied to aircraft classification tasks from high-resolution SAR images, i.e. distinguishing between different classes of airplanes; and
- generalize to new and previously unseen data by considering different locations, environmental conditions, and geometries.

Table 1
Occurrences of the TSX/TDX ST images for each airport, orbit, and incidence angle.

Airport	Orbit	Incidence angle	Number of images
(i)	Ascending	32.5°	3
	Descending	28.5°	3
		41.1°	3
(ii)	Ascending	16.1°	1
		31.1°	13
		43.0°	3
		52.1°	3
	Descending	16.2°	1
		31.2°	7
		43.0°	3
(iii)	Ascending	35.9°	3
	Descending	47.5°	5
(iv)	Ascending	31.1°	3
	Descending	45.8°	3
(v)	Descending	32.6°	3
		44.6°	3

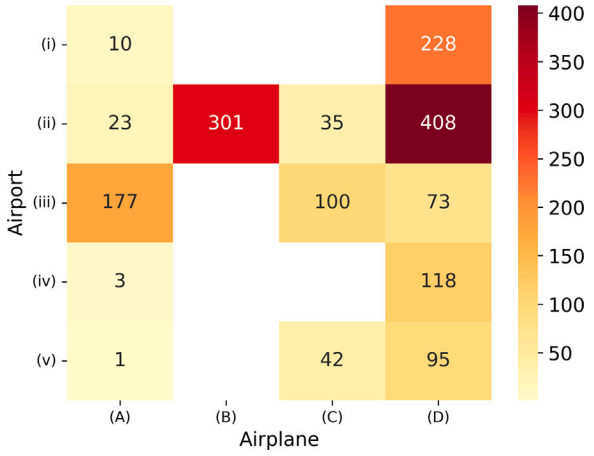

Fig. 1. Number of labels for each airport (rows) and aircraft class (columns).

The remainder of the manuscript is structured as follows: Section 2 presents all the used data and its preparation, while Section 3 describes the method used for classification as well as assessing its performance and generalization capability. The experiments and related discussion are presented in Section 4. Finally, we draw conclusions about the investigation and present an outlook for future work in Section 5.

2. Dataset

The used dataset consists of 60 acquisitions with TerraSAR-X (TSX) and TanDEM-X (TDX) in Staring Spotlight (ST) mode (Mittermayer et al., 2014) with a nominal resolution of 23 cm in azimuth and 58 cm in range, covering four consecutive years from August 8, 2015 to September 3, 2019. The acquired data allows to classify aircraft types located at different airport scenes with varying imaging conditions. The dataset contains five airport scenes where twelve different aircraft classes have been observed in the manual annotation process. The main features of this dataset are the variety of imaging geometries and environmental conditions, thus capturing differences in the signature of the respective objects for e.g. multiple aspects, snow-cover, etc. Table 1 lists the number of data takes per imaging geometry for the respective five airports *(i)-(v)*.

As in all data-driven approaches, the training database plays an essential role for SAR aircraft classification. In the following, we describe the construction of the dataset used for training and evaluation, i.e. the annotation process to create the labels as well as preprocessing to bring the data into a form that is analysis-ready for ML methods.

2.1. Dataset annotation and curation

High-quality annotations are crucial for training and evaluating SAR aircraft classification approaches. Given the difficulties to construct large training datasets of annotated SAR images, label quality is of high importance as the influence of label noise is more severe for smaller datasets.

We used 60 images of five different airports that are annotated by experienced human operators. Quality control consisted of a cross-check with SAR signatures synthetically generated by an internal SAR simulator, which is capable of synthesizing a target given the geometric properties of the acquisition and an external 3D model of the aircraft. In total, 2334 aircrafts are marked in the images of which the 1614 instances belonging to the four most frequent classes are selected. The remaining eight aircraft classes are considered as non-target patches to reinforce the multi-label classification training. More details can be found in Section 2.2. The result is a fairly-well populated multi-class dataset, which is summarized in Fig. 1 (aircraft classes are denoted by *(A)–(D)*, airports by *(i)–(v)*).

Fig. 1 shows the number of available instances for the five available airports with the four selected aircraft classes and demonstrates a small class imbalance in both directions. Most samples are available for the aircraft type *(D)* and airport *(ii)*. Certain aircrafts are annotated only in certain airports, as in the case of class *(B)*, which is only present in airport *(ii)*.

2.2. Patch extraction

ATR can be seen as a two-step process where first potential objects are detected and then subsequently classified (note, however, that not all approaches follow these two steps explicitly). In this work, we focus on the classification part of ATR only and not on the actual detection which means that we require a database of patches showing at most one object instance instead of whole images containing multiple instances.

We apply a semi-automatic strategy to generate patches containing the previously detected targets as *positive samples* and also a set of *negative samples*, i.e. patches that do not contain any of the previously defined aircraft classes. Given the spatial resolution of TSX/TDX images as well as the size of the objects of interest, we use a patch size of 256×256 pixels that completely covers instances of the smallest to the largest target class with a small margin at the borders.

The *negative samples* consist of two different sources: (1) the other eight aircraft types that are sparsely present in the dataset, and (2) background patches selected by joint evaluation of the coherent scattering characteristic in a given patch (Sanjuan-Ferrer et al., 2015) and the associated coefficient of variation, defined as the ratio of the local standard deviation to the local mean of the SAR amplitude. The background samples are generated by randomly selecting 150 negatives from each image in the dataset. The set of negatives in a given image is the combination of three different subsets. The first subset consists of those negatives with a high number of coherent scatterers, which is a good indication of man-made objects. Since the objects of interest usually have a considerable number of scatterers within the patch, man-made objects with similar properties represent hard negative examples that are highly informative. Thus, 60% of all negative samples belong to this subset. The second negative subset consists of patches with a lower number of coherent scatterers than in the first subset, but with a high coefficient of variation. This subset includes different types of surfaces and some specific artificial objects. It contributes 35% to all negative samples. Finally, the third subset provides the remaining 5% and consists of patches with few coherent scatterers and a low coefficient of variation, which typically correspond to relatively homogeneous clutter areas.

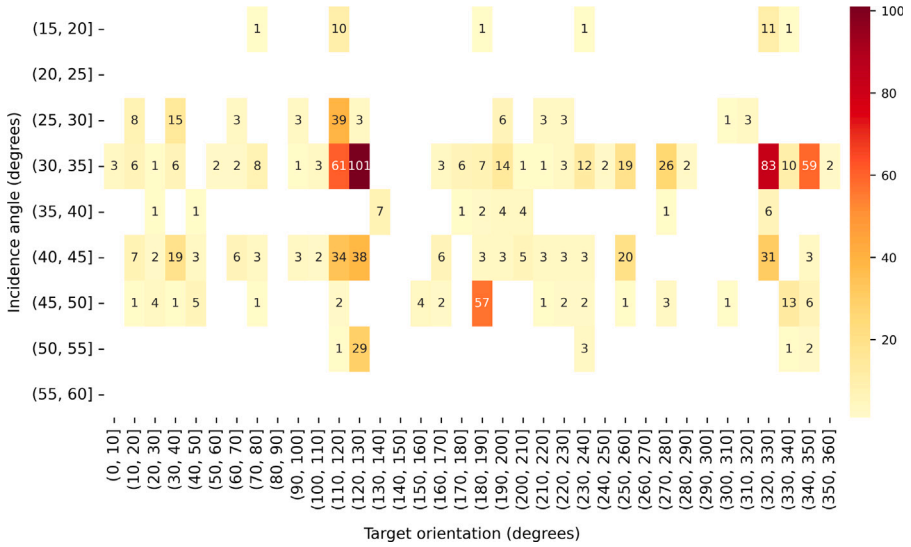

Fig. 2. Distribution of the labels for the aircraft class *(D)* when the incidence angle and the angle of target orientation are varied and all airports are considered.

The proposed patch formation guarantees a diverse dataset with respect to the two main geometric parameters of the TSX ST mode, i.e. the incidence angle and the target orientation. The latter is intuitively the actual orientation of the object with respect to the line of sight (LOS) of the SAR sensor and, especially in our case study, it strongly depends on the specific airport. Fig. 2 collects in a 2-D histogram the number of instances for the airplane class *(D)* as a function of the two angles. For simplicity, we have divided the range of possible incidence angles and target orientation angles into several discrete intervals using a bin size of 5° and 10°, respectively. We observe an irregular distribution of the object instances in the airplane class *(D)* for both dimensions. The different sampling along the incidence angle axis is only a limitation of the selected set of TSX/TDX ST images and can be solved by increasing the number of acquisitions trying to cover the gaps. The irregular distribution of instances along the target orientation axis is the more critical problem. Solving it would require acquiring and annotating more images containing this specific object, e.g. aircraft class *(D)*, in different orientations. This aspect is a limitation of virtually all real datasets as some classes and object orientations are simply more common (e.g., airplanes are often parked in certain positions and orientations at the airport).

It is worth illustrating how the targets appear when the two considered geometric parameters are varied. Given a target orientation of 116°, Fig. 3 compares the radar signature of four airplane instances of class *(D)* selected from the full available set previously shown in Fig. 2. In particular, we can observe the aircraft class *(D)* at different incidence angle values, (a) 47°, (b) 41°, (c) 31°, and (d) 28°, and deduce that the radar shadow becomes more prominent at lower (steeper) incidence angles. On the other hand, Fig. 4 compares the radar brightness of aircraft class *(D)* seen at different target orientation values by keeping the incidence angle constant at 28°. In Fig. 4(a)–(c) and (b)–(d) the aircrafts are in the same parking slot. They do not change orientation to north, but are imaged respectively from ascending and descending orbits. This emphasizes that signatures can vary drastically with imaging conditions, even if the target itself remains stationary.

Finally, the used dataset consists of 1614 aircraft samples across 60 images, containing four different aircraft classes and the previously described set of semi-automated *negative samples* at five different airports. Each sample has a patch size of 256 × 256 pixels and each aircraft is centered in the patch by roughly estimating its center of gravity.

Rather than applying the full data set directly, several test scenarios of increasing difficulty are defined in Table 2 combining cases of having single/multiple classes of interest at single/multiple airports. In particular, we are interested in how performance changes when the task of classifying a single aircraft at a single airport is extended to classify multiple aircraft classes at airports that are not part of the training set.

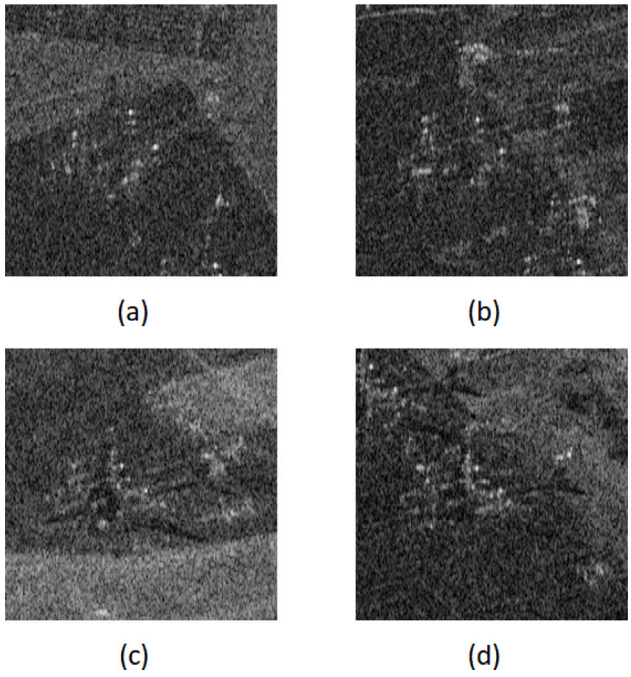

Fig. 3. Radar brightness β_0 of airplane class *(D)*, oriented 116° w.r.t. the North direction, using different incidence angles: (a) 47°, (b) 41°, (c) 31°, (d) 28°.

Table 2
List of the 4 test scenarios considered in Section 4.

	Airplanes	Airports	Targets	Images
Scenario 1	Single *(D)*	Single *(ii)*	408	31
Scenario 2	Multiple	Single *(ii)*	767	31
Scenario 3	Single *(D)*	Multiple	922	60
Scenario 4	Multiple	Multiple	1614	60

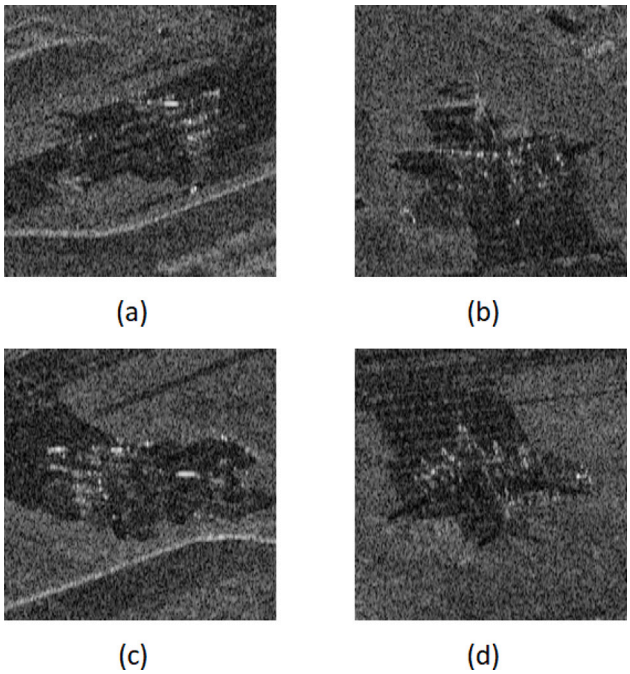

Fig. 4. Radar brightness β_0 of airplane class *(D)*, observed with an incidence angle of 28°, using target orientation angles: (a) 312°, (b) 194°, (c) 67°, (d) 15°.

Table 3
List of the 4 features considered in Section 4. All features are stored as 8-bit unsigned integer arrays.

Feature	Description
β_0	Radar brightness
γ_{cs}	Sublooking process in range direction followed by a false alarm thresholding to highlight the strongest scatterers in the coherence (Sanjuan-Ferrer et al., 2015)
$\hat{A}_{ml}$	Despeckling and sidelobe reduction followed by a 2-D multilooking (Anglberger et al., 2017)
$\hat{A}$	Despeckling and sidelobe reduction without loss of resolution (Anglberger et al., 2017)

Please note, that in both *Scenario 1* and *Scenario 3* the aircraft classes *(A)*, *(B)*, and *(C)* are added to the set of *negative samples*.

2.3. Feature extraction

One of the most important and delicate tasks in SAR aircraft classification is the ability to inject SAR system information into the classification process. In the case of machine learning algorithms, this can be done at different levels of the learning process. Here, we inject this information in the form of input data by computing additional features from SAR images. Specifically, in our work we focused on four different features extracted from Single Look Complex (SLC) images, as described in Table 3. We point out that this is a peculiarity of the used dataset, which originates from SLC images, while most of the available datasets only provide the corresponding SAR detected amplitude. Fig. 5 shows several example patches for the features described in Table 3 and for each aircraft class.

While Fig. 5(c) and (d) are low-pass filtered versions of the radar brightness shown in Fig. 5(a), the coherent scatterer image γ_{CS} shown in Fig. 5(b) is a binary mask indicating the brighter scatterers in the scene with respect to an adaptive threshold. Amplitude normalization is another important preprocessing step that ensures that the data has a consistent scale. This is especially important when comparing data from different sources, as the amplitude of the signals can vary significantly. Normalization is typically performed by transforming the data to zero mean and unit variance. This helps to reduce the impact of amplitude differences, making it easier to compare data and apply machine learning algorithms. In this work, we converted the amplitude features, β_0, $\hat{A}_{ml}$ and $\hat{A}$, in decibels and scaled the results to 8 bits, by stretching the data between −30 dB and 20 dB. In the case of the coherent scatterer, γ_{CS}, the resulting binary image was scaled to 8 bits by assigning clutter to 0 and coherent scatterer to 255.

Table 4
Hyperparameters common to the selected architectures.

Patch size	256	
Batch size	32	
Epochs	50	
Learning rate	Type	Step-based
	Initial value	0.0001
	Drop decay	0.1
	Drop rate	15
Optimizer	RMSProp	

3. Method

This section outlines the details of the employed Deep Learning approach, i.e. the network architecture as well as training and evaluation procedures. For training we use the dataset of TerraSAR-X and TanDEM-X images acquired in ST mode as discussed in Section 2. As a baseline, we use a fairly standard model for classification, plus a selection of DL models from standard image classification algorithms, in order to draw general conclusions that are not dependent on specific model architectures or training procedures. We focus on the question of which factors have the strongest impact on performance, i.e., how classification accuracy changes within the different scenarios with varying degrees of diversity: the distribution of classes, orientations, and locations. Since these variables have a direct influence on the observed SAR signature of an object, it is crucial to evaluate how data-driven models are able to generalize to different configurations.

3.1. Used network architectures

The proposed baseline network (BN) is shown in Fig. 6. It consists of a four-layer encoder, which is used to create an internal representation from an input patch. The latent space is then flattened and transformed via a fully connected layer and a softmax activation function into a probability vector for the K different classes. The different feature maps of the encoder are connected via Residual Blocks (RBs), shown in Fig. 7, which helps to mitigate overfitting and allows for easier optimization. When using multiple input features, we use different branches for each feature, which are then concatenated just before the flattening layer.

The model is optimized via the categorical cross-entropy loss and a step-based learning rate, which drops the learning rate every few epochs using the following formula:

$$\eta_n = \eta_0 d^{\lfloor \frac{1+n}{r} \rfloor} \tag{1}$$

where η_0 is the initial learning rate equal to 0.0001, d is the drop decay set to 0.1, n is the iteration step, r is the drop rate set to 15 epochs, and $\lfloor . \rfloor$ refers to the floor function.

In addition, we consider a selection of standard DL models for classification, namely the VGG16 (Simonyan and Zisserman, 2015), InceptionV3 (Szegedy et al., 2015), ResNet50 (He et al., 2016), Inception-ResNet (Szegedy et al., 2016), and Xception (Chollet, 2017). Table 4 shows the list of hyperparameters common to the six selected architectures and Table 5 summarizes the approximate number of trainable parameters for each network adapted to the specific input shape.

All these models combine convolutional and fully connected layers. In particular, the Visual Geometry Group (VGG) network is characterized by a sequence of 3 × 3 convolutional layers stacked on top

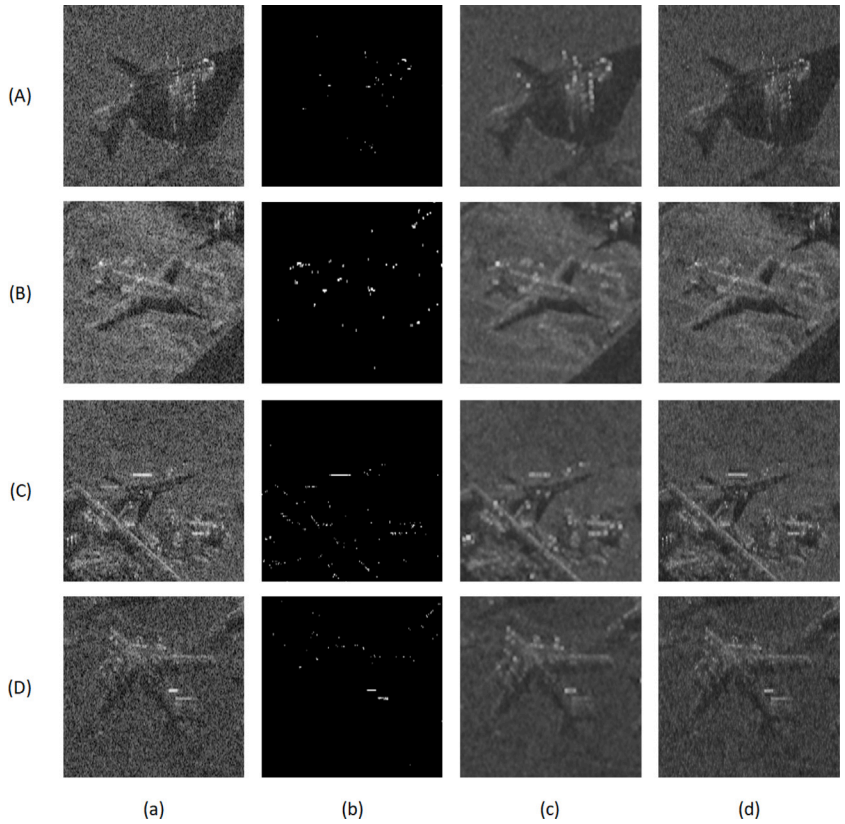

Fig. 5. Overview of the considered features for each aircraft class (A)-(D). From left to right: (a) radar brightness β_0, (b) coherent scatter γ_{cs}, speckle-reduced amplitude (c) with and (d) without multilooking, indicated in Table 3 as $\hat{A}_{ml}$ and $\hat{A}$, respectively.

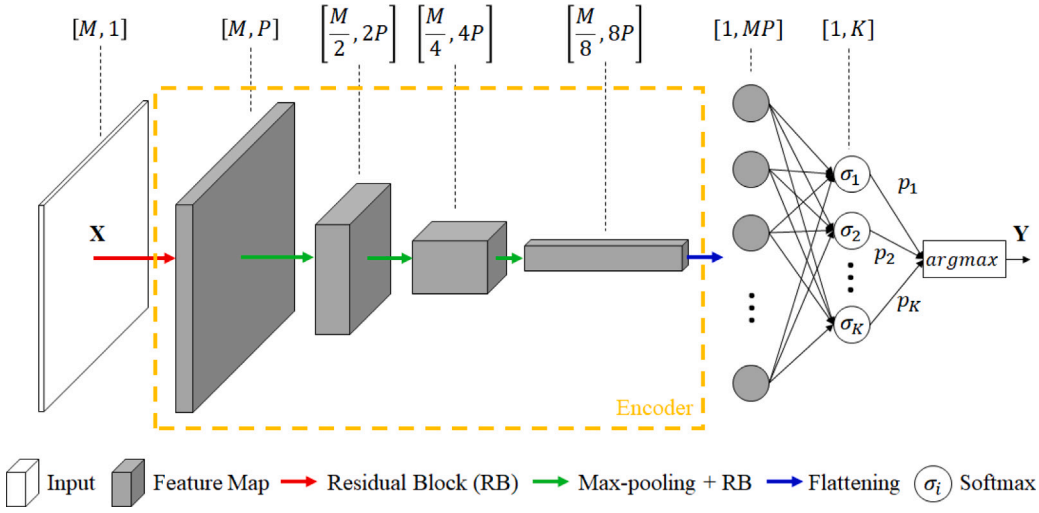

Fig. 6. Baseline network (BN) architecture. The output dimensions at every single layer are shown in brackets, where P is the minimum number of output filters, and M identifies a 2-D ($M \times M$) array. The residual block (RB) is described more in detail in Fig. 7.

of each other in increasing depth; each block is separated by a max pooling layer that gradually reduces the volume size. VGG16 refers to the number of weight layers considered in the network. As shown in Table 5, VGG16 presents the largest number of trainable parameters. InceptionV3 is a modified version of the VGG16 and introduces a block called multi-level feature extractor that simultaneously computes 1×1, 3×3, and 5 × 5 convolutions and stacks their results along the channel dimension before propagating into the next layer of the network. Xception is an extension of the InceptionV3 architecture and consists of depthwise separable convolutions that replace the standard Inception blocks. VGG16, InceptionV3, and Xception have in common a sequential approach as well as the BN architecture. ResNet50 architecture makes use of shortcut connections to solve the vanishing gradient problem. As a result, the architecture is much deeper than VGG16, but the model size is smaller due to the usage of global average pooling rather than fully connected layers. Inception-ResNet is a hybrid architecture that incorporates residual connections into the Inception backbone. By using such a variety of networks, as well as the BN architecture, we aim to ensure that the generalization problem is addressed independently of the used architecture.

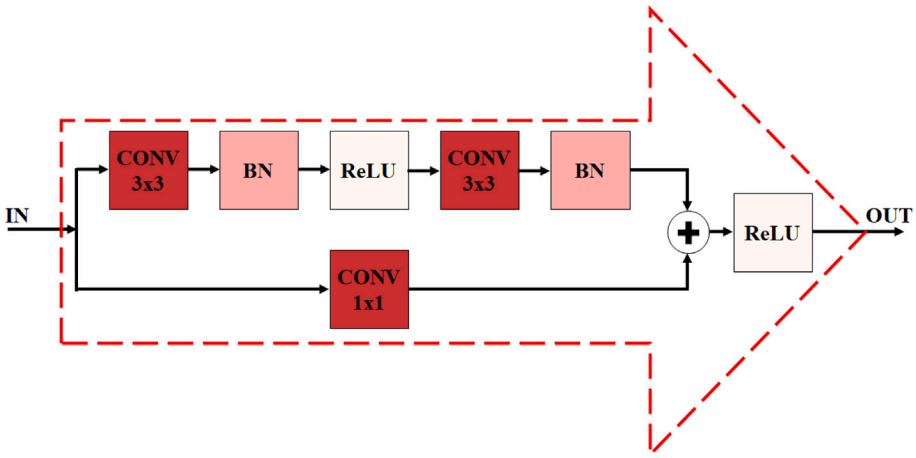

Fig. 7. Structure of the residual blocks used in the BN architecture.

Table 5
Trainable parameters of each used architecture. In the context of our analysis, the network parameters are designed by setting a patch size equal to 256 × 256, and a single input feature.

Architecture	Trainable parameters
BN	11.8 M
VGG16	165.7 M
InceptionV3	21.8 M
ResNet50	23.5 M
Inception-ResNet	54.3 M
Xception	20.8 M

3.2. Data splits

Ideally, the model would be trained on and applied to images acquired over different airports, using different incidence angles, orbits, and seasons containing a multitude of instances for the different aircraft classes. Such a diverse dataset, however, is very difficult and costly to produce. As a consequence, detection and classification models are often trained and evaluated on datasets that only contain a subset of such configurations, e.g. only a single airport, a limited set of incidence angles, etc. To evaluate how much the performance of the proposed methodology degrades when certain conditions in the training and test data differ, we distinguish whether instances of a *Single* or *Multiple* classes of aircrafts shall be classified as well as whether images of a single or multiple airports are used. This results in four different scenarios as shown in Table 2. We also consider four different variations to divide the data into subsets for training and evaluation if images of multiple airports are used. The general split applied to all scenarios is the *Varied Split* which randomly divides patches into training, validation, and test sets using 70%, 10%, and 20% of the available images, including all possible conditions. This creates validation and test sets that are most similar to the training data and thus represents the simplest case. Table 6 shows the number of images and target samples used for training, validation, and testing when applying the *Varied Split* strategy in the different scenarios.

Additionally, we apply an *Airport Split* to all scenarios with multiple airports, i.e. *Scenario 3* and *Scenario 4*. All the images of the airports *(i)*, *(iii)*, *(iv)*, and *(v)* in Fig. 1 are used for training and validation, while the remaining images of airport *(ii)* are dedicated for testing the network. Table 7 reports the number of images and target samples.

We also analyze the effect of different incidence angles in isolation, without the influence of seasonal changes or different locations, by applying the *Incidence Split* to all scenarios with a single airport, i.e. *Scenario 1* and *Scenario 2*. This split is applied only to the test cases of images without snow (the most significant seasonal change occurring over airports). All the images with an incidence angle between 40° and 45° are used for testing, while the other acquisitions are employed for training and validation. Table 8 shows the number of images and target samples used for training, validation and testing.

Finally, we apply a *Season Split* to all scenarios with a single airport to analyze the classification performance for different weather conditions. Specifically, this split uses all winter images (most with snow) for testing, and the rest for training and validation. The effect of seasonal

Table 6
Varied split: Number of images, resulting target and negative samples used for training, validation, and testing in the different scenarios.

Images			
Scenario	Train	Validation	Test
Single *(ii)*	21	3	7
Multiple	40	7	13
Airplane samples *(A)*, *(B)*, *(C)*, *(D)*, *(N)*			
Scenario	Train	Validation	Test
Scenario 1	–,–,–,275,3769	–,–,–,40,539	–,–,–,93,1077
Scenario 2	13,204,25,275,3476	1,30,3,40,496	9,67,7,93,994
Scenario 3	–,–,–,617,6860	–,–,–,114,980	–,–,–,191,1960
Scenario 4	128,204,115,617,6300	31,30,24,114,900	55,67,38,191,1800

Table 7
Airport split: Number of images and resulting target samples used for training, validation, and testing in the scenarios with multiple airports.

Images			
Scenario	Train	Validation	Test
Multiple	49	5	6
Airplane samples *(A)*, *(B)*, *(C)*, *(D)*, *(N)*			
Scenario	Train	Validation	Test
Scenario 3	–,–,–,743,2229	–,–,–,84,319	–,–,–,95,637
Scenario 4	185,282,123,743,2079	28,19,12,84,297	1,0,42,95,594

Table 8
Incidence split: Number of images and resulting target samples used for training, validation, and testing in the scenarios with a single airport.

Images			
Scenario	Train	Validation	Test
Single *(ii)*	25	3	3
Airplane samples *(A)*, *(B)*, *(C)*, *(D)*, *(N)*			
Scenario	Train	Validation	Test
Scenario 1	–,–,–,334,3850	–,–,–,37,459	–,–,–,37,442
Scenario 2	21,241,29,334,3559	1,30,4,37,424	1,30,3,37,408

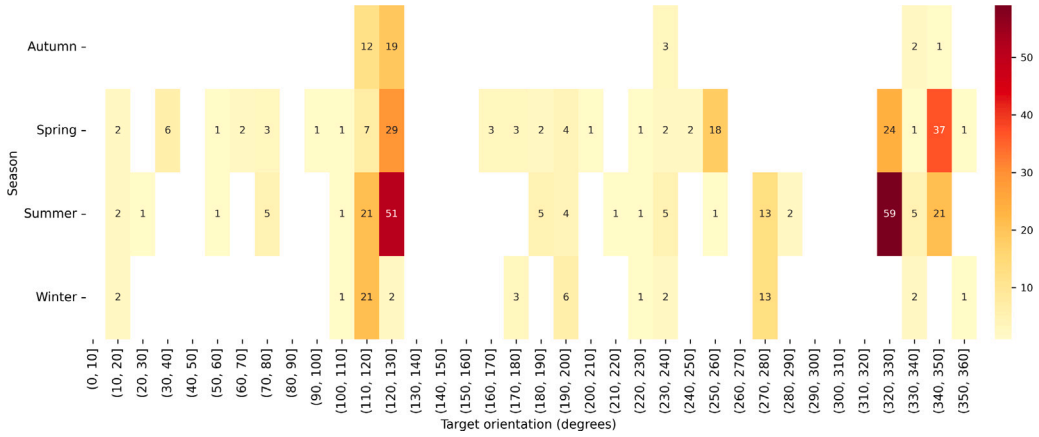

Fig. 8. Distribution of the labels for the aircraft class (D) when the seasonality and the angle of target orientation are varied, airport *(ii)* and common values of the incidence angle between 30° and 35° are considered.

Table 9
Season split: Number of images and resulting target samples used for training, validation, and testing in the scenarios with a single airport.

Images			
Scenario	Train	Validation	Test
Single *(ii)*	16	2	2
Airplane samples *(A)*, *(B)*, *(C)*, *(D)*, *(N)*			
Scenario	Train	Validation	Test
Scenario 1	–,–,–,222,2504	–,–,–,25,269	–,–,–,26,238
Scenario 2	15,154,20,222,2315	1,18,2,25,248	1,19,2,26,216

changes is evaluated in isolation, without the influence of different airports, and only using images with incidence angles between 30° and 35°. Table 9 illustrates the number of images and target samples. Fig. 8 collects the number of instances for the airplane class *(D)* as a function of the target orientation angle and the four seasons in a 2-D histogram.

Overall, we perform an analysis of different test cases, considering scenarios with single/multiple airports, single/multiple aircraft classes, single/multiple input features, and different splitting strategies on the dataset. The results are reported in Section 4.

3.3. Performance metrics

Performance metrics play a crucial role in the assessment of a pre-trained classifier. In this work, we select different evaluation metrics for measuring the quality of binary and multiclass classifiers. In both cases, we consider the overall accuracy which is one of the most common metrics used to evaluate the generalization ability of classifiers and corresponds to the proportion of correct predictions. In a binary problem, the formula for quantifying the overall accuracy, OA, is:

$$OA = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2}$$

where TP are the *true positives*, FP are the *false positives*, TN are the *true negatives*, and FN are the *false negatives*. The overall accuracy has several weaknesses, e.g. it is biased in favor of the majority class (Hossin and Sulaiman, 2015). An often used alternative to have an overview of the classification is the measurement of the average accuracy, AA, which corresponds to the average of each accuracy per class, i.e. it is the sum of accuracy for each class predicted divided by the number of classes. The average accuracy can be directly used in both binary and multiclass problems with

$$AA = \frac{1}{C} \sum_{c=1}^{C} \frac{TP_c}{TP_c + FN_c}, \tag{3}$$

where TP_c and FN_c are the true-positive and false-negative samples of the cth out of C classes.

In the binary problem, other additional metrics are designed for performance evaluation. Precision, Pr, measures the positive samples correctly predicted from the total samples predicted in a positive class and is defined as:

$$Pr = \frac{TP}{TP + FP}. \tag{4}$$

On the other hand, recall, Re, measures the fraction of positives that are correctly classified and can be written as:

$$Re = \frac{TP}{TP + FN}. \tag{5}$$

Finally, F1-score is another used metric that represents the harmonic mean between recall and precision values. In formulas, it is defined as:

$$F1 = \frac{2 \cdot Pr \cdot Re}{Pr + Re}. \tag{6}$$

4. Experiments

This section presents the results obtained by using the six models described in Section 3.1 for different experiments. A key component for improving aircraft classification performance is the identification of relevant features. The BN architecture implemented in *Scenario 1* allows us to evaluate feature importance using a single or a group of features from Table 3. We use the *Varied Split* strategy to separate the training, validation, and test datasets. Table 10 reports the overall accuracy (OA), average accuracy (AA), F1 score, precision (Pr) and recall (Re) of seven different feature combinations. In general, all features perform quite well, with OA, AA, and F1 scores greater than 97%, 92%, and 86%, respectively. The fact that OA is mostly larger than AA is caused by the unbalanced test set and indicates a better performance for the dominant class, i.e. the negative samples. Of the features tested, $\hat{A}_{ml}$ performs best, followed closely by the similar feature $\hat{A}$, illustrating that speckle is one of the dominant factors that reduces the overall detection performance, which can be successfully addressed by despeckling and sidelobe reduction in a preprocessing step. The weakest results are obtained with γ_{cs}, i.e., limiting the signal to scatterers with strong coherence, such as the jet engines and the vertical and horizontal stabilizers of the tail. Details about the shape of the aircraft are dropped. This seems to either cause too much loss of information, or results in input images that are too sparse to be efficiently processed by convolutional networks. Combining different features does not have a significant effect on performance, i.e., using a group of features leads to similar or worse performance than using the strongest feature in the group alone. As shown in Fig. 5, the correlation

Table 10
Evaluation of the BN architecture using as test dataset *Scenario 1*, and a *Varied Split* of the dataset. Each row presents the numerical computation using the different combinations of features.

Features	OA	AA	F1	Pr	Re
(β_0)	98.8	97.39	92.71	89.9	95.7
(γ_{cs})	98.46	93.27	90.0	**93.1**	87.1
$(\hat{A}_{ml})$	**99.32**	**99.63**	**95.88**	92.08	**100**
$(\hat{A})$	99.15	98.55	94.79	91.92	97.85
(β_0, γ_{cs})	97.78	92.41	86.02	86.02	86.02
$(\beta_0, \hat{A}_{ml})$	**99.32**	99.14	95.83	92.93	98.92
$(\beta_0, \hat{A})$	99.23	99.09	95.34	92.0	98.92

Table 11
Evaluation of the BN network in the four scenarios. $\hat{A}_{ml}$ is used as input feature and each dataset is divided using the *Varied Split* strategy.

	OA	AA	F1	Pr	Re
Scenario 1	99.32	99.63	95.88	92.08	100.0
Scenario 2	98.63	79.56	87.99	98.41	79.56
Scenario 3	99.49	98.03	97.11	97.37	96.86
Scenario 4	98.14	88.19	92.13	96.43	88.19

between the selected features is responsible for the performance degradation when they are combined as input to the BN architecture. As a result, the next experiments are performed using the multilooked and despeckled amplitude $\hat{A}_{ml}$. Table 11 describes the performance of the BN architecture for the four different scenarios described in Section 2, using the *Varied Split* strategy.

A relatively high OA of more than 98% is achieved in all scenarios. However, since the test set is unbalanced, the AA is more relevant, and here the different scenarios show significant differences. Extending the classification task to multiple aircraft classes, i.e. *Scenario 2* and *Scenario 4*, leads to a significant decrease in AA by 20% and 10%, respectively. This is mainly caused by minority classes (which explain the stable OA performance), i.e. classes *(A)* and *(C)* in Table 6, which lead to a strong confusion with the background class. Nevertheless, the results clearly show that such classification tasks can be solved with satisfactory accuracy if the requirements regarding the amount and quality (i.e. consistency and diversity) of the training data are met.

Table 12 extends the performance evaluation to the six selected networks. All of them are trained on the *Varied Split* dataset of the four different scenarios, using as input feature $\hat{A}_{ml}$. InceptionV3 performs better overall on this particular dataset. A possible explanation is that it uses Inception layers with different filter sizes applied simultaneously. As a result, the large number of hidden layers allows more complex features to be learned. In the single airport scenarios, i.e. *Scenario 1* and *Scenario 2*, very deep neural networks, such as VGG16 and Inception-ResNet, do not significantly increase the performance indicators compared to light convolutional networks such as the BN architecture.

This behavior is due to the difficulty the networks have in converging with a limited amount of training data compared to their large number of trainable parameters. In *Scenario 4* of Table 12, the F1-score degradation of 2% can be observed when comparing the values from the VGG16 and the BN architectures. This result is correlated with the number of weights of the VGG16 that are an order of magnitude larger than those of the BN architecture (see Table 5). In addition, Table 12 shows that a well-balanced and populated splitting strategy guarantees high performance on any architecture. For the sake of brevity, the following experiments are obtained by leveraging the BN and InceptionV3 networks using feature $\hat{A}_{ml}$ as input for three distinct experiments, i.e. applying them to (i) airport, (ii) incidence, and (iii) season splitting strategies described in Section 3.2.

Table 12
Comparison of classification performance using feature β_0 as input and a *Varied Split* of each scenario. Each row presents the numerical computation of the different selected CNNs. For each scenario, the greater performance metric is highlighted in bold.

	Architecture	OA	AA	F1	Pr	Re
Scenario 1	BN	99.32	99.63	95.88	92.08	**100.0**
	VGG16	99.74	99.37	98.4	97.87	98.92
	InceptionV3	**99.91**	**99.46**	**99.46**	**100.0**	98.92
	ResNet50	99.83	99.42	98.92	98.92	98.92
	Inception-ResNet	99.57	98.79	97.33	96.81	97.85
	Xception	99.66	99.32	97.87	96.84	98.92
Scenario 2	BN	98.63	79.56	87.99	**98.41**	79.56
	VGG16	98.21	79.82	87.66	97.21	79.82
	InceptionV3	**99.23**	**82.2**	88.48	95.81	**82.2**
	ResNet50	98.72	75.46	85.12	97.61	75.46
	Inception-ResNet	99.06	81.88	**89.25**	98.08	81.88
	Xception	98.63	74.8	84.88	98.11	74.8
Scenario 3	BN	99.49	98.03	97.11	97.37	96.86
	VGG16	99.63	98.61	97.89	98.41	97.38
	InceptionV3	**99.77**	**99.16**	**98.69**	**98.95**	**98.43**
	ResNet50	99.4	97.78	96.57	97.34	95.81
	Inception-ResNet	99.67	98.64	98.15	98.94	97.38
	Xception	99.21	96.73	95.47	97.28	93.72
Scenario 4	BN	98.14	88.19	92.13	96.43	88.19
	VGG16	97.81	84.12	90.0	96.76	84.12
	InceptionV3	**99.35**	**95.64**	**96.75**	97.88	**95.64**
	ResNet50	98.14	85.15	90.39	96.32	85.15
	Inception-ResNet	98.7	88.29	93.17	**98.62**	88.29
	Xception	97.35	77.53	85.56	95.45	77.53

4.1. Varied versus Airport splits

This section considers the multiple airport scenarios indicated in Table 2 as *Scenario 3* and *Scenario 4*, respectively. For each scenario, we train the BN and InceptionV3 networks using the *Airport Split* where test samples are from an airport that is not contained in the training set and compare the results against the *Varied Split*, i.e. having train and test samples from all available airports. Table 13 shows that all performance metrics drop considerably whenever we differentiate the airports for training and testing. While classification performance for a single airplane class in multiple airports (*Scenario 3*) is close to perfect when training data of all airports is available, accuracy decreases dramatically when the model is tested on an airport outside of the training set. Precision remains high, but there is a drop in Recall, when using the BN architecture. This indicates that either the acquisition factors of images over the test airport are not well covered by the training set or that both the models and in particular the BN architecture learned shortcuts to identify instances of the target class that are not valid if data comes from a different airport as those in the training set. The representativeness of the training data is also a contributing factor, although it does not rule out other reasons such as the model focusing on spurious correlations. One example of such a shortcut is that airplanes might not be moved between two acquisitions or even if moved, are parked again at the same location. In both cases, there is a strong correlation between a parking spot and the airplane class. The location information is encoded in the background of the image, e.g. nearby stationary objects or ground features. This would cause an information leak from the training to the test set, even if the splits are performed on image level. The results for *Scenario 4* lead to similar conclusions. The drop in performance appears to be less severe given the accuracy statistics in Table 13, mainly because performance is already worse compared to easier scenarios. It can be observed that the complexity introduced by the InceptionV3 architecture guarantees an overall higher classification performance than the BN architecture.

4.2. Varied versus Incidence splits

In this experiment, the BN and InceptionV3 classifiers are performed considering the single airport scenarios, denoted in Table 2 as

Table 13
Evaluation of the BN and InceptionV3 networks in *Scenario 3* and *Scenario 4* using the $\hat{A}_{ml}$ feature. Each row reports the performance using a different split of the test data set. In the case of multiple classes, F1-score, precision, and recall are computed as class-wise averages.

	Architecture	Split	OA	AA	F1	Pr	Re
Scenario 3	BN	Varied	99.49	98.3	97.11	97.37	96.86
		Airport	92.62	71.58	82.04	96.09	71.58
	InceptionV3	Varied	99.77	99.16	98.69	98.95	98.43
		Airport	97.95	92.55	91.53	98.78	85.26
Scenario 4	BN	Varied	98.14	88.19	92.13	96.43	88.19
		Airport	90.16	71.64	71.92	72.2	71.64
	InceptionV3	Varied	99.35	95.64	96.75	97.88	95.64
		Airport	94.67	79.33	80.46	77.87	79.33

Table 14
Evaluation of the BN and InceptionV3 networks in *Scenario 1* and *Scenario 2* using the $\hat{A}_{ml}$ feature. Each row reports the performance using a different split of the test dataset. In the case of multiple classes, F1-score, precision, and recall are computed as class-wise averages.

	Architecture	Split	OA	AA	F1	Pr	Re
Scenario 1	BN	Varied	99.49	98.3	97.11	97.37	96.86
		Incidence	95.13	71.83	56.9	78.57	44.59
	InceptionV3	Varied	99.77	99.16	98.69	98.95	98.43
		Incidence	97.47	83.06	79.03	98.0	66.22
Scenario 2	BN	Varied	98.14	88.19	92.13	96.43	88.19
		Incidence	91.72	55.69	63.4	73.58	55.69
	InceptionV3	Varied	99.35	95.64	96.75	97.88	95.64
		Incidence	98.44	95.05	93.68	92.36	95.05

Scenario 1 and *Scenario 2*, respectively. For each scenario, we trained the network using the *Incidence Split* where test samples are from incidence angles comprised between 40° and 45° not contained in the training set. Similarly to Section 4.1, we compare the results against the *Varied Split*, i.e. having train and test samples from all available incidence angles.

As expected, Table 14 shows a drop in performance in both the scenarios. The acquisition geometry plays a crucial role in the classification because the radar brightness of a target depends on the inclination of the radar antenna beam. Whenever we isolate a portion of the incidence angle range from the training, we lose the information about the signature of the target. As a result, performance indicators such as the F1-score start to decrease mostly due to a decreased Recall.

4.3. Varied versus Season splits

In this section, we train both the BN and InceptionV3 networks on the same single airport scenarios using the $\hat{A}_{ml}$ feature and the *Season Split* where test samples are extracted from winter acquisitions not considered in the training set. Similarly to Section 4.2, we compare the results against the *Varied Split* in Table 15. We observe that performance parameters such as the F1-score, Precision, and Recall are decreasing in the *Season Split* and this is associated with the isolation of a season from the training.

Additionally, the AA metric is worse than the one obtained in Table 14 with the *Incidence Split*. The loss of accuracy can be attributed to changes in the background. Seasonality means a change in the environment due to external weather conditions. Fig. 9 compares the apron of the airport *(ii)* in (a) summer against (b) winter. By visual inspection, we can observe that many airplanes might be misclassified as negatives when surrounded by snow and vice versa. The snow signature is comparable to the brightness of the tails and fuselages of the airplanes.

Table 15
Evaluation of the BN and InceptionV3 networks in *Scenario 1* and *Scenario 2* using the $\hat{A}_{ml}$ feature. Each row reports the performance using a different split of the test dataset. In the case of multiple classes, F1-score, precision, and recall are computed as class-wise averages.

	Architecture	Split	OA	AA	F1	Pr	Re
Scenario 1	BN	Varied	99.49	98.3	97.11	97.37	96.86
		Season	95.52	75.4	66.67	96.3	50.98
	InceptionV3	Varied	99.77	99.16	98.69	98.95	98.43
		Season	95.69	70.38	68.35	96.43	52.94
Scenario 2	BN	Varied	98.14	88.19	92.13	96.43	88.19
		Season	90.0	40.95	53.67	77.87	40.95
	InceptionV3	Varied	99.35	95.64	96.75	97.88	95.64
		Season	91.9	59.63	74.21	98.24	59.63

5. Conclusion and future work

The current landscape of SAR aircraft classification research reveals a notable incongruence between the abundance of deep learning-based aircraft classification algorithms and the scarcity of compatible datasets. This has led to advances in target detection and classification in specific operational scenarios, while leaving the challenge of achieving robust generalization across larger and more diverse datasets unresolved. This, however, is a critical requirement for the practical deployment of SAR aircraft classification systems.

Our central objective is the evaluation of the classification performance and generalization capabilities of a generic deep learning model in the context of SAR aircraft classification. We elucidate the relationship of model performance and different operational scenarios that influence the ability to perform aircraft classification, particularly with respect to high-resolution SAR images. This investigation is conducted by using manually annotated high-resolution SAR data from the TanDEM-X mission, grounding the study in real-world, operationally relevant scenarios. We show how the model's ability to generalize changes when confronted with novel data involving different geographic locations, geometric complexities, and weather conditions. To support our findings, we use several DL models for this investigation, selected from the standard classification algorithms.

The results indicate that achieving generalization in this type of problem remains a largely unsolved challenge. A possible approach to address this problem is simplification by constraining certain parameters, such as focusing on the identification of specific aircraft types, controlling the geometry of SAR acquisitions within a given area, and specifying airport locations. While it is still possible to train models on a specific subset of variables, our results strongly suggest that this is the preferred strategy.

To improve the generalization capabilities of the model, an effective approach might be to augment the training data with simulated SAR target signatures. In addition, progress can be made in improving generalization by exploiting geometry-invariant features, as exemplified by the application of state-of-the-art representation learning techniques.

CRediT authorship contribution statement

Andrea Pulella: Conceptualization, Investigation, Methodology, Software, Writing – original draft, Validation. **Francescopaolo Sica:** Conceptualization, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing. **Carlos Villamil Lopez:** Investigation, Software, Writing – review & editing. **Harald Anglberger:** Writing – review & editing. **Ronny Hänsch:** Conceptualization, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

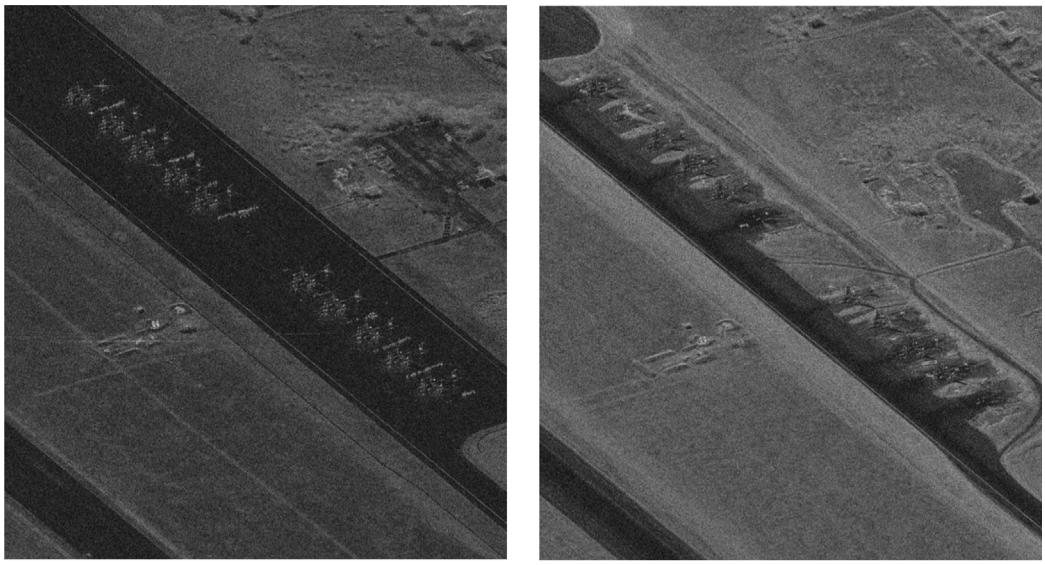

Fig. 9. Two image patches showing the same apron of the airport *(ii)* for different seasons: (a) summer time, (b) winter time with snowdrift around the parked airplanes.

References

Anglberger, H., Kempf, T., Profelt, J., Villamil Lopez, C., Speck, R., 2017. RADIAN Version 2.04 Benutzerhandbuch. Tech. rep., DLR.

Cao, C., Cui, Z., Wang, L., Wang, J., Cao, Z., Yang, J., 2022. Cost-sensitive awareness-based SAR automatic target recognition for imbalanced data. IEEE Trans. Geosci. Remote Sens. 60, 1—16.

Chen, K., Pan, Z., Huang, Z., Hu, Y., Ding, C., 2022. Learning from reliable unlabeled samples for semi-supervised SAR ATR. IEEE Geosci. Remote Sens. Lett. 19, 1—5.

Chen, S., Wang, H., Xu, F., Jin, Y.-Q., 2016. Target classification using the deep convolutional networks for SAR images. IEEE Trans. Geosci. Remote Sens. 54 (8), 4806—4817.

Cho, J.H., Park, C.G., 2018. Multiple feature aggregation using convolutional neural networks for SAR image-based automatic target recognition. IEEE Geosci. Remote Sens. Lett. 15 (12), 1882—1886.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. arXiv:1610.02357.

Diemunsch, J.R., Wissinger, J., 1998. Moving and stationary target acquisition and recognition (MSTAR) model-based automatic target recognition: search technology for a robust ATR. In: Zelnio, E.G. (Ed.), Algorithms for Synthetic Aperture Radar Imagery V. Vol. 3370, International Society for Optics and Photonics, SPIE, pp. 481—492.

Ding, J., Chen, B., Liu, H., Huang, M., 2016. Convolutional neural network with data augmentation for SAR target recognition. IEEE Geosci. Remote Sens. Lett. 13 (3), 364—368.

El-Darymli, K., Gill, E.W., Mcguire, P., Power, D., Moloney, C., 2016. Automatic target recognition in synthetic aperture radar imagery: A state-of-the-art review. IEEE Access 4, 6014—6058.

Feng, S., Ji, K., Wang, F., Zhang, L., Ma, X., Kuang, G., 2022. Electromagnetic scattering feature (ESF) module embedded network based on ASC model for robust and interpretable SAR ATR. IEEE Trans. Geosci. Remote Sens. 60, 1—15.

Feng, S., Ji, K., Wang, F., Zhang, L., Ma, X., Kuang, G., 2023. PAN: Part attention network integrating electromagnetic characteristics for interpretable SAR vehicle target recognition. IEEE Trans. Geosci. Remote Sens. 61, 1—17.

Feng, S., Ji, K., Zhang, L., Ma, X., Kuang, G., 2021. SAR target classification based on integration of ASC parts model and deep learning algorithm. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14, 10213—10225.

Gu, Y., Tao, J., Feng, L., Wang, H., 2021. Using VGG16 to military target classification on MSTAR dataset. In: 2021 2nd China International SAR Symposium. CISS, pp. 1—3.

Guo, Y., Pan, Z., Wang, M., Wang, J., Yang, W., 2020. Learning capsules for SAR target recognition. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13, 4663—4673.

Guo, Q., Wang, H., Xu, F., 2019. Aircraft detection in high-resolution SAR images using scattering feature information. In: 2019 6th Asia-Pacific Conference on Synthetic Aperture Radar. APSAR, pp. 1—5.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Identity mappings in deep residual networks. arXiv:1603.05027.

Hossin, M., Sulaiman, M.N., 2015. A review on evaluation metrics for data classification evaluations. Int. J. Data Min. Knowl. Manag. Process 5, 1—11.

Huang, Z., Pan, Z., Lei, B., 2020. What, where, and how to transfer in SAR target recognition based on deep CNNs. IEEE Trans. Geosci. Remote Sens. 58 (4), 2324—2336.

Jacob, S., Wall, J., Sharif, M.S., 2023. Analysis of deep neural networks for military target classification using synthetic aperture radar images. In: 2023 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies. 3ICT, pp. 227—233.

Lang, P., Fu, X., Feng, C., Dong, J., Qin, R., Martorella, M., 2022. LW-CMDANet: A novel attention network for SAR automatic target recognition. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 15, 6615—6630.

Li, R., Wang, X., Wang, J., Song, Y., Lei, L., 2022. SAR target recognition based on efficient fully convolutional attention block CNN. IEEE Geosci. Remote Sens. Lett. 19, 1—5.

Lin, Z., Ji, K., Kang, M., Leng, X., Zou, H., 2017. Deep convolutional highway unit network for SAR target classification with limited labeled training data. IEEE Geosci. Remote Sens. Lett. 14 (7), 1091—1095.

Mittermayer, J., Wollstadt, S., Prats-Iraola, P., Scheiber, R., 2014. The TerraSAR-X staring spotlight mode concept. IEEE Trans. Geosci. Remote Sens. 52 (6), 3695—3706.

Morgan, D.A.E., 2015. Deep convolutional neural networks for ATR from SAR imagery. In: Zelnio, E.K., Garber, F.D. (Eds.), Algorithms for Synthetic Aperture Radar Imagery XXII. Vol. 9475, International Society for Optics and Photonics, SPIE, pp. 116—128.

O'Sullivan, J.A., DeVore, M.D., Kedia, V., Miller, M.I., 2001. SAR ATR performance using a conditionally Gaussian model. IEEE Trans. Aerosp. Electron. Syst. 37 (1), 91—108.

Pei, J., Huang, Y., Huo, W., Zhang, Y., Yang, J., Yeo, T.-S., 2018. SAR automatic target recognition based on multiview deep learning framework. IEEE Trans. Geosci. Remote Sens. 56 (4), 2196—2210.

Principe, J.C., Kim, M., Fisher, M., 1998. Target discrimination in synthetic aperture radar using artificial neural networks. IEEE Trans. Image Process. 7 (8), 1136—1149.

Pulella, A., Sica, F., 2021. Situational awareness of large infrastructures using remote sensing: The Rome–Fiumicino airport during the COVID-19 lockdown. Remote Sens. 13 (2).

Ren, H., Yu, X., Zou, L., Zhou, Y., Wang, X., Bruzzone, L., 2021. Extended convolutional capsule network with application on SAR automatic target recognition. Signal Process. 183, 108021.

Roemer, H., Kiefl, R., Henkel, F., Cao, W., Nippold, R., Kurz, F., Kippnich, U., 2016. Using airborne remote sensing to increase situational awareness in civil protection and humanitarian relief - the importance of user involvement. In: Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XLI-B8, pp. 1363—1370.

Sanjuan-Ferrer, M.J., Hajnsek, I., Papathanassiou, K., Moreira, A., 2015. A new detection algorithm for coherent scatterers in SAR data. IEEE Trans. Geosci. Remote Sens. 53 (11), 6293—6307.

Shah, R., Soni, A., Mall, V., Gadhiya, T., Roy, A.K., 2019. Automatic target recognition from SAR images using capsule networks. In: Pattern Recognition and Machine Intelligence: 8th International Conference, PReMI 2019, Tezpur, India, December 17-20, 2019, Proceedings, Part II. Springer, pp. 377—386.

Shang, R., Wang, J., Jiao, L., Stolkin, R., Hou, B., Li, Y., 2018. SAR targets classification based on deep memory convolution neural networks and transfer parameters. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11 (8), 2834—2846.

Shi, J., 2022. SAR target recognition method of MSTAR data set based on multi-feature fusion. In: 2022 International Conference on Big Data, Information and Computer Network. BDICN, pp. 626—632.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Soldin, R.J., 2018. SAR target recognition with deep learning. In: 2018 IEEE Applied Imagery Pattern Recognition Workshop. AIPR, pp. 1—8.

Song, Y., Li, J., Gao, P., Li, L., Tian, T., Tian, J., 2022. Two-stage cross-modality transfer learning method for military-civilian SAR ship recognition. IEEE Geosci. Remote Sens. Lett. 19, 1—5.

Srinivas, U., Monga, V., Raj, R.G., 2014. SAR automatic target recognition using discriminative graphical models. IEEE Trans. Aerosp. Electron. Syst. 50, 591—606.

Sun, Y., Liu, Z., Todorovic, S., Li, J., 2007. Adaptive boosting for SAR automatic target recognition. IEEE Trans. Aerosp. Electron. Syst. 43 (1), 112—125.

Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A., 2016. Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv:1602.07261.

Szegedy, Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2015. Rethinking the inception architecture for computer vision. arXiv:1512.00567.

Wagner, S.A., 2016. SAR ATR by a combination of convolutional neural network and support vector machines. IEEE Trans. Aerosp. Electron. Syst. 52 (6), 2861—2872.

Wang, C., Liu, X., Pei, J., Huang, Y., Zhang, Y., Yang, J., 2021. Multiview attention CNN-LSTM network for SAR automatic target recognition. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14, 12504—12513.

Zhang, M., An, J., Yu, D.H., Yang, L.D., Wu, L., Lu, X.Q., 2020a. Convolutional neural network with attention mechanism for SAR automatic target recognition. IEEE Geosci. Remote Sens. Lett. 19, 1—5.

Zhang, X., Feng, S., Zhao, C., Sun, Z., Zhang, S., Ji, K., 2024. MGSFA-Net: Multiscale global scattering feature association network for SAR ship target recognition. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 17, 4611—4625.

Zhang, F., Liu, Y., Zhou, Y., Yin, Q., Li, H.-C., 2020b. A lossless lightweight CNN design for SAR target recognition. Remote Sens. Lett. 11 (5), 485—494.

Zhang, L., Zhang, C., Quan, S., Xiao, H., Kuang, G., Liu, L., 2020c. A class imbalance loss for imbalanced object recognition. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13, 2778—2792.

Zhao, Q., Principe, J.C., 2001. Support vector machines for SAR automatic target recognition. IEEE Trans. Aerosp. Electron. Syst. 37 (2), 643—654.

Zhong, C., Mu, X., He, X., Wang, J., Zhu, M., 2019. SAR target image classification based on transfer learning and model compression. IEEE Geosci. Remote Sens. Lett. 16 (3), 412—416.