

Technical Note

# Fast and Accurate System for Onboard Target Recognition on Raw SAR Echo Data

Gustavo Jacinto <sup>1</sup>, Mário Véstias <sup>2,3,\*</sup>, Paulo Flores <sup>1</sup> and Rui Policarpo Duarte <sup>2,3</sup>

<sup>1</sup> INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, 1000-039 Lisbon, Portugal; gustavo.jacinto@tecnico.ulisboa.pt (G.J.); paulo.flores@inesc-id.pt (P.F.)

<sup>2</sup> INESC INOV, 1000-029 Lisbon, Portugal; rui.duarte@inov.pt

<sup>3</sup> Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, 1959-007 Lisbon, Portugal

\* Correspondence: mario.vestias@inov.pt

## Highlights

### What are the main findings?

- Direct target recognition from raw SAR echo data with an accuracy close to 100% with low power.
- Recognition can be achieved with neural network models with low computational cost, yet fast enough to enable real-time processing.

### What are the implications of the main findings?

- Target recognition from SAR echo data can run onboard using embedded devices with less computing power.
- Enables faster decision-making in time-critical missions, since data are processed near the sensor.

## Abstract

Synthetic Aperture Radar (SAR) onboard satellites provides high-resolution Earth imaging independent of weather conditions. SAR data are acquired by an aircraft or satellite and sent to a ground station to be processed. However, for novel applications requiring real-time analysis and decisions, onboard processing is necessary to escape the limited downlink bandwidth and latency. One such application is real-time target recognition, which has emerged as a decisive operation in areas such as defense and surveillance. In recent years, deep learning models have improved the accuracy of target recognition algorithms. However, these are based on optical image processing and are computation and memory expensive, which requires not only processing the SAR pulse data but also optimized models and architectures for efficient deployment in onboard computers. This paper presents a fast and accurate target recognition system directly on raw SAR data using a neural network model. This network receives and processes SAR echo data for fast processing, alleviating the computationally expensive DSP image generation algorithms such as Backprojection and RangeDoppler. Thus, this allows the use of simpler and faster models, while maintaining accuracy. The system was designed, optimized, and tested on low-cost embedded devices with low size, weight, and energy requirements (Khadas VIM3 and Raspberry Pi 5). Results demonstrate that the proposed solution achieves a target classification accuracy for the MSTAR dataset close to 100% in less than 1.5 ms and 5.5 W of power.

**Keywords:** SAR echo data; target recognition; neural network; embedded



Academic Editor: Mingsheng Liao

Received: 14 August 2025

Revised: 10 October 2025

Accepted: 22 October 2025

Published: 26 October 2025

**Citation:** Jacinto, G.; Véstias, M.; Flores, P.; Duarte, R.P. Fast and Accurate System for Onboard Target Recognition on Raw SAR Echo Data. *Remote Sens.* **2025**, *17*, 3547.

<https://doi.org/10.3390/rs17213547>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Synthetic Aperture Radar (SAR) technology [1] is widely used for the identification of planet surface objects and scenarios in any weather condition, such as tracking ships and oil spills, terrain erosion, drought and landslides, deforestation, and fires [2]. For all these reasons and more, the modern challenges addressed by SAR make it an extremely important technology. Current Automatic Target Recognition (ATR) works classify targets present in SAR images, which implies the use of an image reconstruction algorithm on captured SAR echo data [3]. This approach is computationally expensive and, therefore, is infeasible for time-sensitive applications such as defense and surveillance. Furthermore, each data transformation accumulates precision errors that reduce the overall classification precision.

Using a neural network to recognize targets directly from the SAR echo data alone is a promising alternative for object classification, as it removes the need to process the data into an image, an often time-consuming operation, and reduces the source of errors. Raw data classification is expected to achieve results with higher accuracy than SAR image classification, since the final image lacks the details still present in the echo data. The use of a neural network to process SAR echo data directly is crucial to develop the speed and cost of SAR ATR tasks and guarantee that these solutions can be implemented in compact low-cost boards.

The neural network models to be executed onboard must be carefully designed to avoid high computing requirements and energy consumption. So, the structure of the proposed neural network model was found through a design model exploration and mapped to an embedded low-cost, yet powerful, device for low energy consumption.

In SAR ATR, situations where the captured target configuration differs from the training data configuration can considerably degrade the accuracy of the results. These conditions are referred to as Extended Operation Conditions (EOC) [4]. Situations where the target configuration is closer to matching the data on which the network was trained are referred to as Standard Operation Conditions (SOC). In this work, both conditions were considered to show the robustness of the proposal.

Training a neural network for ATR tasks requires datasets that contain the raw SAR echo data of labeled targets. Unfortunately, there is a tremendous lack of SAR datasets overall, much less ones that fit these conditions. The only proper dataset that could be found was Moving and Stationary Target Acquisition and Recognition (MSTAR). It is a dataset that contains raw data and images of a good variety of targets in various conditions. The targets are centered in each data sample, making it simpler for ATR applications. It is also the most widely used dataset in the related works, which allows a more straightforward comparison of the accuracy results.

This work proposes an optimized neural network to classify targets captured by SAR directly from the SAR echo data. A simple, yet effective, novel neural network is designed and trained to classify SAR echo data. The proposed network classified both SOC and EOC with extreme accuracy. The neural network was optimized and implemented on two single board computers, Khadas VIM3 and Raspberry Pi 5 due to their small Size, Weight, and Power (SWaP) constraints. The experimental results demonstrated that the proposed networks can classify data from the MSTAR dataset [5] on embedded devices with accuracies above 99% in both SOC and EOC.

## 2. Related Work

Early machine learning algorithms for the classification of objects from SAR images focused on identifying features such as geometric features [6], histograms of oriented gradient [7], fusion features [8], scattering features [9], and statistical features [10], among others.

Recently, with the advent of deep learning, these features are automatically learned, achieving better performance. Object classification applied to SAR has been proposed as a means to extract information from captured scenes to aid in quicker and/or automated decision-making. Some works focus on target classification based on SAR images. The work in ref. [11] was one of the first approaches to consider Convolutional Neural Network (CNN) for SAR image classification. They proposed All-convolutional Network (A-ConvNet), a network with sparse layers to reduce the overfitting problem. They also established the configuration for SOC and EOC in most future classification works that also use MSTAR [5]. SOC implies training with data captured at a  $17^\circ$  depression angle and testing with a  $15^\circ$  depression angle data. EOC implies training with the same type of data but testing with a  $30^\circ$  depression angle data. Any mention of SOC and EOC throughout this article refers to these conditions, unless stated otherwise. The network achieved an accuracy of 99.13% for SOC 87.40% for EOC. The work also explores the introduction of random noise in the input data observing a large impact on the accuracy.

To deal with scarce SAR data, CHU-Net [12], a CNN network with dropout to avoid overfitting from scarce data, was proposed to classify SAR images. When using all data, the model achieved an accuracy close to 99%, which dropped to 94% when trained with a small subset of the training data.

Another work looking for a solution to scarce data [13] proposed Amplitude-phase Convolutional Neural Network (CNN), a CNN that considers both the amplitude and phase of SAR data. This improves the accuracy of the algorithm in EOC when trained with scarce data compared to methods that only consider the amplitude. The CNN achieved accuracies of 98.10% and 93.57% in SOC and EOC, respectively. A two-step approach was considered in ref. [14] to deal with a lack of training data. An initial step uses a CNN to extract features to train a second CNN for SAR image classification.

Hybrid models have also been considered for target classification of SAR images. Some considered the integration of two types of deep learning models such as ref. [15], where a CNN was combined with a Long Short-term Memory (LSTM) model. A CNN was used to extract features, enhanced with a spatial attention module, followed by LSTM that fused features from adjacent azimuths when multiview images were present. This network achieved accuracies of 99.38% and 95.57% in SOC and EOC, respectively. Other hybrid approaches considered a deep learning model combined with traditional machine learning algorithms. In ref. [16], a histogram-oriented gradient was combined with a CNN enhanced with the attention mechanism. Two SAR ship datasets were tested: OpenSARShip [17] and FUSAR-Ship [18]. Their network, HOG-ShipCLSNet, achieved accuracies of 78.15% and 86.69% with each dataset, respectively.

In ref. [19], data augmentation techniques such as clutter transfer were applied to images in the MSTAR dataset [5] to improve the robustness of target recognition, with a specially tuned ResNet18 network. The work achieved an accuracy of 97.2%. With non-ideal ResNet18 parameters and contrast balancing, it achieved an average accuracy of 88.5%. It also considered experiments with changing the background clutter and generating synthetic images, with a minor impact on the accuracy.

Transformers and attention mechanisms are at the core of the most recent SAR image classifiers [20–22]. One of the most recent works [22] achieved accuracies of up to 99.79% in SOC and up to 98.52% in EOC on MSTAR [5]. The work also achieved an approximate accuracy of 84.25% throughout the different versions of the OpenSARShip [17] dataset.

Deep learning-based methods are effective in identifying ships in SAR images. However, echo data have to be preprocessed with correction functions and converted to an image, which is finally processed with a target detection algorithm. Transmission of echo data should be included in the processing flow if data are processed in a ground station.

For real-time target detection onboard, the computational and energy onboard requirements are high. Taking this into account, a new research direction was taken focused on ATR directly applied to the raw SAR data.

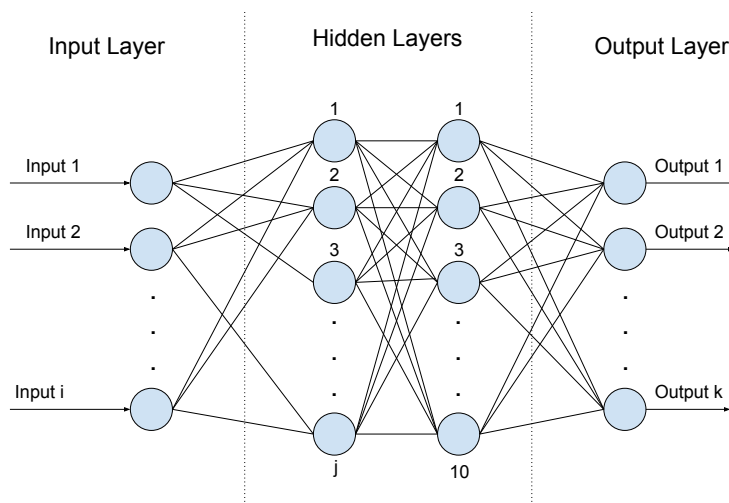
Recent works [23,24] have focused on ATR with raw Ground-based Synthetic Aperture Radar (GBSAR) data. GBSAR is a variation of SAR that is typically applied in indoor environments. These works used a custom-made sensor attached to a rail to capture small objects of different shapes and materials. In ref. [23], a modified ResNet18 network was trained to perform multilabel classification on three bottles of varying materials. Various experiments were conducted, such as different weight initializations and comparing the raw data and image classification results. Raw data classification achieved the best results with a mean F1 score of 88.24%. In ref. [24], the same modified ResNet18 was used to train on GBSAR data with different polarizations mixed into the input in different ways, such as mixing the rows of data or appending horizontal polarity data to vertical polarity data, referred to as JOIN. The previously mentioned article only used data with horizontal polarization. This article also created a siamese model that combined the results of two separate ResNets, trained on each polarization. The JOIN model resulted in the highest accuracy, which was 93.06%.

Fast Range-compressed Detection (FastRCDet) was proposed in ref. [25], a novel lightweight network for ship detection that accepts range-compressed SAR echo data as its input. The network was conceived to detect ships onboard the SAR platform. They also proposed a network to adapt the data to the range-compressed domain. The lightweight network with 2.49 M parameters was able to detect ships with an average accuracy of 77.12%.

SAR image classification works achieve accuracies around 99% but require an SAR image formation step and complex CNN models. Works on SAR echo data classification are still in their infancy, but the results are promising. The work proposed in this paper contributes to ATR on raw SAR echo data. An optimized neural model is designed to achieve high accuracy, avoid overfitting due to scarce data, and reduce memory and computing complexities for onboard execution at low energy.

### 3. Proposed Method: Neural Network

The main goal of this work was to define the smallest and simplest possible neural network architecture that could produce highly accurate SAR classification results. All network sizes that were tested had the base architecture shown in Figure 1. Each hidden layer is dense and uses ReLU as the activation function. In this diagram,  $i$  corresponds to the number of input neurons,  $k$  corresponds to the number of output neurons, and  $j$  is the number of neurons in the first hidden layer. The number of outputs depends on the conditions the network is for: SOC or EOC. The  $k$  value depends on the dataset used and corresponds to the output classes. The  $j$  value is the main focus of the network size experiments. Values between 60 and 20 were tested. By default, the  $i$  value is the size of a raw echo data sample in the dataset being used.



**Figure 1.** The neural network architecture tested for all conditions.

### 3.1. Dataset

The MSTAR dataset [5] is widely used for SAR ATR tasks. It contains labeled raw data and images of various armored vehicles. The dataset also contains data captured in different conditions, mainly different depression angles. It was chosen for its availability, popularity, and simplicity.

The most abundant data present in the dataset had a depression angle of  $17^\circ$ . These data included 7 classes with 300 samples each: 2S1, BRDM\_2, D7, SLICY, T62, ZIL131, and ZSU\_23\_4. These data were determined for training. Data with a depression angle of  $15^\circ$  were divided into the same 7 classes but only contained 274 samples each. Since  $15^\circ$  and  $17^\circ$  depression angles are similar, the  $15^\circ$  depression angle data were used for testing the network in SOC. There were only 4 classes that had data with a  $30^\circ$  depression angle: 2S1, BRDM\_2, SLICY, and ZSU\_23\_4. This extreme angle was chosen for training the proposed networks in EOC.

Inspired by ref. [11], the data with varying depression angles were used to test the robustness of the proposed network. In SOC,  $17^\circ$  data were used to train the network on all 7 classes, and  $15^\circ$  data were used to test. EOC refers to the conditions in which a target can be captured that are outside the expected, leading to lower accuracy ATR results [26]. In this case, MSTAR data, captured with a depression angle of  $30^\circ$ , were used for testing a network trained in  $17^\circ$  angle data to see how the network classified these extreme conditions.

### 3.2. Preprocessing

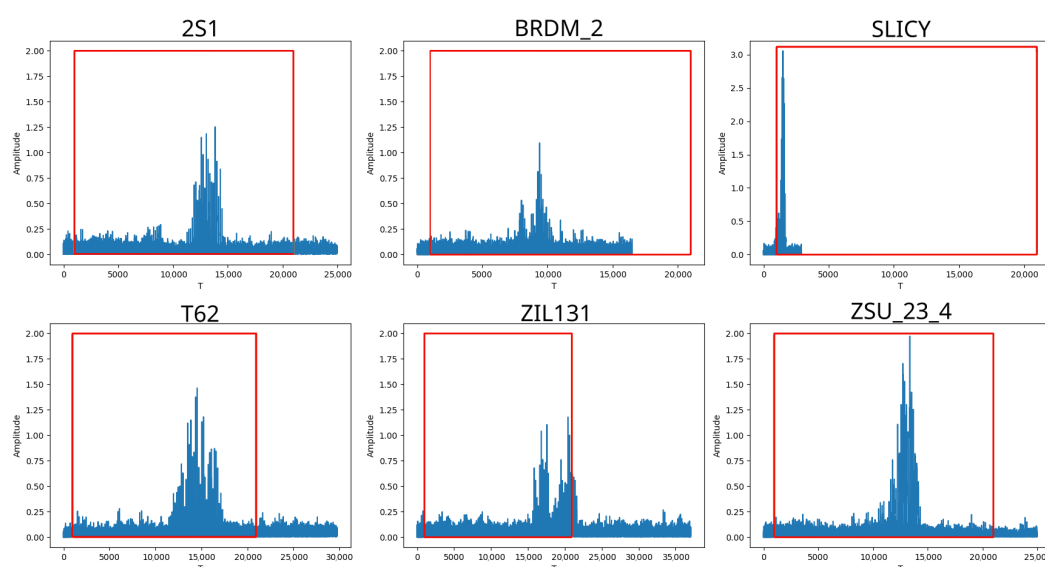
An initial problem was identified with the data present in the dataset: the size of the samples varied between classes. The input size of the network must remain constant. In image classification tasks, it is common to resize the image to  $224 \times 224$ , not only to make the input size constant but also to reduce the processing times of training and inference. However, raw data should not be treated like an image. The chosen solution was to create a range of the data that had to at least contain the target (typically the center of the signal or, after image processing, the center of the image). This range of data is referred to as a “window”. Signals shorter than the window’s size were given 0 padding, while signals that were longer were cut.

Two of the classes, 2S1 and ZSU\_23\_4, had approximately the same size of 25,000 data values. The window was therefore defined as having a range of 0–25,000. The other classes were verified to confirm that a window of such a range did not cut off the target, which is the object to classify. If there are ever other datasets that share this size problem, the same

method can be applied. Otherwise, in datasets with consistent raw data sample sizes, the window serves only in the case described in the following subsection.

### 3.3. Model Optimization

The first experiment to shorten the input size involved reducing the size of the input data by altering the start and length of the data window. The window reductions have a limit, as they must contain the most important part of the signal: the target. As an example for the MSTAR dataset, a window region of 1000–20,000 was determined to be suitable for all classes. This results in an input size of 19,000, a reduction in 6000 data values. Figure 2 displays plots of raw data from randomly selected samples of 6 classes. It also contains a square drawn in each plot corresponding to the window region of 1000–20,000. The peaks in the middle of each signal correspond to the targets. SLICY, the class with the shortest samples, has a target that barely fits in the defined window interval. The network was trained without weight initialization or dropout.



**Figure 2.** The raw data of six different classes plotted on diagrams. The left and right lines of the red squares correspond to the limits of the window, ranging from 1000 to 20,000.

The sampling of the input data is another aspect that was explored. This process consists of using only every Nth data value of the input, effectively reducing its size. Multiples of two between 2 and 8 were tested. In the MSTAR dataset’s case, with the shortened window size and sampling with  $N > 1$ , the input size varied between 10,000 and 2500.

## 4. Results

All the generated network models were trained using categorical cross entropy as the loss function, Adam as the optimizer, a learning rate of 0.001, a batch size of 32, and ten epochs. The learning rate and number of epochs were chosen to match the size of the data, which is small compared to the usual  $224 \times 224$  image in image classification tasks.

The MSTAR dataset was used to experiment on the size of the network. A public tool called “mstar2raw” was used to separate the magnitude data from the full raw data samples. These data were then converted to .csv files.

### 4.1. Standard Operation Conditions

The SOC accuracy results were tested for different sizes of the architecture. The  $17^\circ$  depression angle data were split into training and validation groups (80/20), and the  $15^\circ$

depression angle data, present in the same dataset, were used for testing. Table 1 shows the accuracy of each attempted network. The Layer 1 and Layer 2 columns show the number of neurons in the respective hidden layers of the network. The table shows smaller networks perform better. It is suggested that this is due to the size of the small dataset compared to the larger networks. However, the resulting accuracy differences are extremely small; so, they are most likely the result of a margin of error.

The experiments ended with a network with 20 neurons on the first layer and 10 on the second layer. This stopping point was chosen, because far worse accuracy results were expected to occur for very small networks, especially after the reduction steps mentioned in Section 3.3, which are further explored in the following subsection.

**Table 1.** SOC results on differing networks.

Layer 1	Layer 2	Total Parameters	Estimated Total Size (MB)	Test Accuracy
60	10	1,500,714	5.82	99.896%
50	10	1,250,637	4.87	99.948%
40	10	1,000,527	3.91	100.00%
30	10	750,417	2.96	100.00%
20	10	500,307	2.00	100.00%

This network is very simple, but the size of the input data implies a high number of parameters, since the input size is 25,000, and the input is fully connected to the neurons of the first hidden layer. The window size reduction described in Section 3 is applied in the following experiments.

The results of the window size reduction can be seen in Table 2.

**Table 2.** SOC results of different networks with the smaller window size.

Layer 1	Layer 2	Total Parameters	Estimated Total Size (MB)	Test Accuracy
60	10	1,200,747	4.66	100.00%
50	10	1,000,050	3.89	100.00%
40	10	800,527	3.13	100.00%
30	10	600,417	2.37	100.00%
20	10	400,020	1.60	99.948%

As can be observed from the results in the table, the model size can be reduced to one third with a negligible degradation in accuracy.

Next, the sampling of the input data, as described in Section 3, was applied to further reduce the network. Table 3 shows the results of these experiments, in addition to having the window range of 1000–20,000. The resulting accuracy is extremely high in all experiments with SOC.

#### 4.2. Extended Operation Conditions

Unlike the 17° and 15° depression angle data, MSTAR only contained four classes of 30° angle data: 2S1, BRDM\_2, SLICY, and ZSU\_23\_4. Due to this lower number of classes, the network was trained with the respective classes in 17° depression angles. This is, the network was trained and tested on the same four class types. Similar to the previous experiments in SOC, several networks were trained to determine how small the network could be before noticeable drops in accuracy were observed. The 30° angle data were also used as the validation group to observe drops in accuracy. As in SOC, the initial experiments were performed with a window size of 0–25,000 and a sampling of 1. The results can be seen in Table 4. The network sizes are the same as in SOC, Table 1; so, those columns

were omitted. However, due to the difference in conditions between the training data and the testing data, the accuracy fluctuated more than in SOC. Therefore, new columns were added that display the final and best accuracies of the network, along with the epoch in which it achieved the best accuracy.

**Table 3.** SOC results of different networks with the smaller window size and data sampling.

Layer 1	Layer 2	N	Total Parameters	Estimated Total Size (MB)	Test Accuracy
60	10	2	600,747	2.33	100.00%
		4	300,747	1.17	100.00%
		8	150,747	0.59	99.849%
50	10	2	500,637	1.95	99.948%
		4	250,637	0.98	99.844%
		8	125,637	0.49	99.896%
40	10	2	400,527	1.57	100.00%
		4	200,527	0.78	100.00%
		8	100,527	0.39	99.896%
30	10	2	300,417	1.18	99.948%
		4	150,417	0.59	99.844%
		8	75,417	0.30	99.896%
20	10	2	200,307	0.80	99.896%
		4	100,307	0.40	99.896%
		8	50,307	0.20	99.896%

**Table 4.** EOC results on differing networks .

Layer 1	Layer 2	Final Accuracy	Best Epoch	Best Accuracy
60	10	80.36%	6	81.06%
50	10	98.96%	4	99.22%
40	10	99.04%	2	99.48%
30	10	97.91%	6	98.70%

The results show a large decrease in average accuracy starting in the network architecture, which has 30 neurons in the first layer. Therefore, the one with 40 neurons in that layer was selected for further experimentation. Table 5 shows the results of this network with a window size of 1000–2000 and a varying sampling number. The network sizes are present in the table for ease of access.

**Table 5.** EOC results of a network with the reduced input window size and data sampling.

N	Total Parameters	Estimated Total Size (MB)	Final Accuracy	Best Epoch	Best Accuracy
1	800,494	3.13	96.09%	6	99.39%
2	400,494	1.57	98.96%	8	99.48%
4	200,494	0.78	98.87%	6	99.48%
8	100,494	0.39	97.22%	5	98.18%

#### 4.3. Comparing Results to Related Works

In SOC, the smallest tested network still manages to achieve an accuracy of 99.896%. Therefore, the proposed network for SOC is the network with 20 neurons in the first hidden layer, 10 neurons in the second layer, a window size of 1000–20,000 on the input data, and a data sampling of  $N = 8$ .

In EOC, a small decrease in precision was observed in the smallest tested networks. The proposed network for EOC has 40 neurons, a window size of 1000–20,000, and a data sampling of  $N = 4$ . This was chosen for its accuracy and reduced size. The network with 50 neurons in the first layer shares the same best accuracy but is larger. The best accuracy was observed in the sixth epoch at 99.48% before decreasing to 98.87% by the tenth epoch.

Yoon et al. [22] report values that come close, with 99.79% on SOC and 98.52% on EOC. However, an input image requires the raw data to be processed into an image first, which implies a more power-consuming and time-consuming “preprocessing” step.

From the related works that use raw data as input, the highest accuracy achieved was 93.06% on GBSAR using a modified ResNet. A common thread in these works is the use of deep neural network architectures commonly applied to image classification tasks on raw SAR data. According to our results, simple fully connected neural networks can be trained to achieve higher target classification accuracy, even in EOC. Tables 6 and 7 compare the results of the proposed networks with other networks from the related works, considering SAR images and raw SAR echo data, respectively. The highest accuracies in each category (images and raw data) for each set of conditions, SOC and EOC, are in bold. Comparatively, the proposed network achieves a higher accuracy, but the difference in datasets and methodology prevent a fair comparison. The lack of source code availability for these works further hinders a fair comparisons.

**Table 6.** Accuracies of neural networks used in related works and of the proposed networks for SAR images.

	Dataset	Network	Conditions	Accuracy
IMG	MSTAR [5]	A-ConvNet [11]	SOC	99.13%
			EOC	87.40%
		ResNet18 [19]	contrast-balanced	98.00%
			contrast-balanced	98.90%
		AP-CNN [13]	SOC	98.10%
			EOC	93.57%
		CNN-LSTM [15]	SOC	99.38%
			EOC	95.57%
		Yoon et al. [22]	SOC	<b>99.79%</b>
			EOC	<b>98.52%</b>
OpenSARShip [17]		–	84.25%	
FUSAR-Ship [18]	HOG-ShipCLSNet [16]	–	78.15%	
IMG GBSAR [27]	ResNet18 [23]	–	86.69%	
			97.85%	

**Table 7.** Accuracies of neural networks used in related works and of the proposed networks for raw SAR echo data.

	Dataset	Network	FLOPS	Params	Conditions	Accuracy
RAW	RCSHIP [28]	FastRCDet [25]	8.73 G	2.49 M	–	77.12%
	RAW GBSAR [29]	Modified ResNet18 [24]	≈2 G	≈11 M	–	93.06%
	MSTAR [5]	Proposed	50 K	100 K	SOC	<b>99.90%</b>
			200 K	400 K	EOC	<b>98.87%</b>

From the tables, it can be observed that the proposed models are considerably less complex (number of parameters and operations for a single inference) when compared to the other two works with direct processing of raw SAR echo data. The accuracy is also higher for the proposed models. However, the datasets were different in all cases, since the datasets considered in the compared works could not be accessed.

#### 4.4. Embedded Device Implementation

The proposed networks were first implemented on a desktop computer with an AMD Ryzen 9 7900X3D Central Processing Unit (CPU) with 12 cores, 64 GB of Random-access Memory (RAM), and an Nvidia GeForce RTX 4080 running CUDA 12.0. To test their effectiveness in more limited conditions, the networks were also implemented on a variety of embedded devices to compare each of their inference speeds and power consumption. All network implementations were performed in Python 3.8.3. The tested devices were VIM3 from Khadas and a Raspberry Pi 5. The Khadas VIM3 possesses a quad-core ARM Cortex-A73 CPU running at 2.2 GHz, and the Raspberry Pi has a quad-core ARM Cortex-A76 CPU running at 2.4 GHz. Table 8 shows these metrics. The reported power consumption corresponds to the highest wattage observed during inference. Inference times were measured while inferring with batches of 32 raw data samples. EOC has a longer inference time due to the larger size of the network proposed for these conditions. The difference in power consumption between the NVIDIA Graphics Processing Unit (GPU) and the embedded devices highlights the different applications of such devices, as a flying SAR equipped platform would not be able to host said GPU effectively. The GPU would require too much power. Still, the device is reported here to compare inference times with the related works.

**Table 8.** Power consumption and inference time of the proposed network on the tested embedded devices.

Device	Power Consumption (W)	SOC Inference Time (ms)	EOC Inference Time (ms)
NVIDIA RTX 4080, CUDA 12.0	49	0.23	0.29
Khadas VIM3 [30]	4.084	21.60	21.75
Raspberry Pi 5 [31]	5.458	1.53	1.83

Some of the related works provided inference time metrics as well. These metrics would commonly be displayed in Frames per Second (FPS); so, they were converted to milliseconds for comparison purposes. Table 9 compares the results between the related works. Only Yoon et al. [22] showed the results for the different conditions. The proposed network of Yoon et al. [22], besides having the highest accuracy when compared to other related works, also reports the quickest inference times. However, the network is slower than the network proposed in this article, which was executed on an inferior GPU (RTX 4080 vs. RTX 4090).

**Table 9.** Inference times of the related works.

Network	Device	SOC Inference Time (ms)	EOC Inference Time (ms)
Yoon et al. [22]	NVIDIA RTX 4090, CUDA 12.1	1.80	1.76
HOG-ShipCLSNet [16]	NVIDIA RTX 2080 Ti, CUDA 10.1	4.33	4.33
FastRCDet [25]	Undisclosed	26.30	26.30

## 5. Discussion

The results of the proposed small neural networks were unexpectedly high. These results show promise, but further research has to be pursued to validate the process with different datasets. The work demonstrates that with a small network, it is possible to achieve accurate target detection on low-power embedded systems that can run onboard in real time.

Even data with widely different depression angles were classified with stunning accuracy. Compared to previous works that classify SAR images after image formation, the proposed work achieves better accuracies and can run on low-power embedded devices, while image-based detection must run on high-power devices to achieve the same processing times. The MSTAR dataset however presents ideal scenarios where the targets are always centered. The proposed method works well as a proof of concept. More realistic data are required as further research continues.

The size of the input data can be further reduced by removing the background noise and only feeding target data to the network. Despite the target range of each class being different, a static window range was chosen for consistency. However, there is potential in having an even smaller window if the start of the window is aligned with the start of the target range of each class. This reduction has a large impact on the dimension of the neural network since most parameters of the model are relative to the connections neurons of the input layer and the first hidden layer.

## 6. Conclusions

A small neural network has been proposed for onboard target detection directly from SAR echo data. This avoids the utilization of a costly SAR image formation algorithm and allows its efficient execution onboard with low power.

As the results show, the proposed network for SOC, the smallest tested network, can achieve accuracies as high as 99.896%. EOC calls for a slightly larger network, to achieve accuracies of 99.48%. Comparatively, the proposed network achieves a higher accuracy, but the difference in datasets and methodology prevents a fair comparison. In addition, the sizes of the proposed networks make them energy- and resource-efficient, facilitating their implementation in embedded devices for installation on the moving SAR platform.

Future work will be devoted to more complex scenes featuring overlapping and densely co-located targets. Additionally, different architectures can be further compared to narrow down the best methodology in these new experiments.

**Author Contributions:** Conceptualization, G.J. and R.P.D.; methodology, G.J. and M.V.; software, G.J.; validation, G.J., M.V. and P.F.; investigation, G.J.; writing—original draft preparation, G.J.; writing—review and editing, G.J. and M.V.; supervision, R.P.D., M.V. and P.F.; project administration, R.P.D.; funding acquisition, R.P.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is funded by national funds through Fundação para a Ciência e a Tecnologia, I.P. (FCT) under projects UID/6486/2025 and UID/PRR/6486/2025 and under projects UID/50021/2025 and UID/PRR/50021/2025, project 2023.15325.PEX, and project LISBOA2030-FEDER-00692100-15811.

**Data Availability Statement:** This work used the following publicly available dataset: MSTAR Dataset (<https://www.sdms.af.mil/index.php?collection=mstar>, accessed on 20 March 2025).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Cruz, H.; Véstias, M.; Monteiro, J.; Neto, H.; Duarte, R.P. A Review of Synthetic-Aperture Radar Image Formation Algorithms and Implementations: A Computational Perspective. *Remote Sens.* **2022**, *14*, 1258. [CrossRef]
2. Trinder, J.C. Editorial for Special Issue “Applications of Synthetic Aperture Radar (SAR) for Land Cover Analysis”. *Remote Sens.* **2020**, *12*, 2428. [CrossRef]
3. Mota, D.; Cruz, H.; Miranda, P.R.; Duarte, R.P.; de Sousa, J.T.; Neto, H.C.; Véstias, M.P. Onboard Processing of Synthetic Aperture Radar Backprojection Algorithm in FPGA. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 3600–3611. [CrossRef]

4. Keydel, E.R.; Lee, S.W.; Moore, J.T. MSTAR extended operating conditions: A tutorial. In Proceedings of the Algorithms for Synthetic Aperture Radar Imagery III, Orlando, FL, USA, 8–12 April 1996; Zelnio, E.G., Douglass, R.J., Eds.; SPIE: Bellingham, WA, USA, 1996. [CrossRef]
5. U.S. Air Force and DARPA. Moving and Stationary Target Acquisition and Recognition (MSTAR) Dataset, 2005. Available online: <https://www.sdms.af.mil/datasets/mstar/> (accessed on 20 March 2025).
6. Lang, H.; Wu, S. Ship Classification in Moderate-Resolution SAR Image by Naive Geometric Features-Combined Multiple Kernel Learning. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1765–1769. [CrossRef]
7. Lin, H.; Song, S.; Yang, J. Ship Classification Based on MSHOG Feature and Task-Driven Dictionary Learning with Structured Incoherent Constraints in SAR Images. *Remote Sens.* **2018**, *10*, 190. [CrossRef]
8. Zhou, G.; Zhang, G.; Xue, B. A Maximum-Information-Minimum-Redundancy-Based Feature Fusion Framework for Ship Classification in Moderate-Resolution SAR Image. *Sensors* **2021**, *21*, 519. [CrossRef] [PubMed]
9. Wang, X.; Liu, C.; Li, Z.; Ji, X.; Zhang, X. Superstructure scattering features and their application in high-resolution SAR ship classification. *J. Appl. Remote Sens.* **2022**, *16*, 036507. [CrossRef]
10. Wu, F.; Wang, C.; Jiang, S.; Zhang, H.; Zhang, B. Classification of Vessels in Single-Pol COSMO-SkyMed Images Based on Statistical and Structural Features. *Remote Sens.* **2015**, *7*, 5511–5533. [CrossRef]
11. Chen, S.; Wang, H.; Xu, F.; Jin, Y.Q. Target Classification Using the Deep Convolutional Networks for SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4806–4817. [CrossRef]
12. Lin, Z.; Ji, K.; Kang, M.; Leng, X.; Zou, H. Deep Convolutional Highway Unit Network for SAR Target Classification with Limited Labeled Training Data. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1091–1095. [CrossRef]
13. Deng, J.; Bi, H.; Zhang, J.; Liu, Z.; Yu, L. Amplitude-Phase CNN-Based SAR Target Classification via Complex-Valued Sparse Image. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 5214–5221. [CrossRef]
14. Huang, Z.; Yao, X.; Liu, Y.; Dumitru, C.O.; Datcu, M.; Han, J. Physically explainable CNN for SAR image classification. *ISPRS J. Photogramm. Remote Sens.* **2022**, *190*, 25–37. [CrossRef]
15. Wang, C.; Liu, X.; Pei, J.; Huang, Y.; Zhang, Y.; Yang, J. Multiview Attention CNN-LSTM Network for SAR Automatic Target Recognition. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 12504–12513. [CrossRef]
16. Zhang, T.; Zhang, X.; Ke, X.; Liu, C.; Xu, X.; Zhan, X.; Wang, C.; Ahmad, I.; Zhou, Y.; Pan, D.; et al. HOG-ShipCLSNet: A Novel Deep Learning Network with HOG Feature Fusion for SAR Ship Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5210322. [CrossRef]
17. Huang, L.; Liu, B.; Li, B.; Guo, W.; Yu, W.; Zhang, Z.; Yu, W. OpenSARShip: A Dataset Dedicated to Sentinel-1 Ship Interpretation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 195–208. [CrossRef]
18. Hou, X.; Ao, W.; Song, Q.; Lai, J.; Wang, H.; Xu, F. FUSAR-Ship: Building a high-resolution SAR-AIS matchup dataset of Gaofen-3 for ship detection and recognition. *Sci. China Inf. Sci.* **2020**, *63*, 140303. [CrossRef]
19. Geng, Z.; Xu, Y.; Wang, B.N.; Yu, X.; Zhu, D.Y.; Zhang, G. Target Recognition in SAR Images by Deep Learning with Training Data Augmentation. *Sensors* **2023**, *23*, 941. [CrossRef]
20. Wang, C.; Huang, Y.; Liu, X.; Pei, J.; Zhang, Y.; Yang, J. Global in Local: A Convolutional Transformer for SAR ATR FSL. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 4509605. [CrossRef]
21. Wang, D.; Song, Y.; Huang, J.; An, D.; Chen, L. SAR Target Classification Based on Multiscale Attention Super-Class Network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 9004–9019. [CrossRef]
22. Yoon, J.; Song, J.; Hussain, T.; Khowaja, S.A.; Muhammad, K.; Lee, I.H. Hybrid Conv-Attention Networks for Synthetic Aperture Radar Imagery-Based Target Recognition. *IEEE Access* **2024**, *12*, 53045–53055. [CrossRef]
23. Kačan, M.; Turčinović, F.; Bojanjac, D.; Bosiljevac, M. Deep Learning Approach for Object Classification on Raw and Reconstructed GBSAR Data. *Remote Sens.* **2022**, *14*, 5673. [CrossRef]
24. Turčinović, F.; Kačan, M.; Bojanjac, D.; Bosiljevac, M.; Šipuš, Z. Utilizing Polarization Diversity in GBSAR Data-Based Object Classification. *Sensors* **2024**, *24*, 2305. [CrossRef] [PubMed]
25. Tan, X.; Leng, X.; Sun, Z.; Luo, R.; Ji, K.; Kuang, G. Lightweight Ship Detection Network for SAR Range-Compressed Domain. *Remote Sens.* **2024**, *16*, 3284. [CrossRef]
26. Ross, T.D.; Bradley, J.J.; Hudson, L.J.; O'Connor, M.P. SAR ATR: So what's the problem? An MSTAR perspective. In Proceedings of the Algorithms for Synthetic Aperture Radar Imagery VI, Orlando, FL, USA, 5–9 April 1999; Zelnio, E.G., Ed.; SPIE: Bellingham, WA, USA, 1999. [CrossRef]
27. Turčinović, F. Near-Distance Raw and Reconstructed Ground Based SAR Data. 2022. Available online: <https://data.mendeley.com/datasets/m458grc688/1> (accessed on 12 April 2025).
28. Tan, X.; Leng, X.; Ji, K.; Kuang, G. RCShip: A Dataset Dedicated to Ship Detection in Range-Compressed SAR Data. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 4004805. [CrossRef]
29. Turčinović, F. Ground Based SAR Data Obtained with Different Polarizations. 2024. Available online: <https://data.mendeley.com/datasets/nbc9xpwv96/1> (accessed on 12 April 2025).

30. Khadas. VIM3 Single Board Computer. 2019. Available online: <https://www.khadas.com/vim3> (accessed on 20 October 2025).
31. Raspberry Pi Ltd. Raspberry Pi 5 Single Board Computer. 2023. Available online: <https://www.raspberrypi.com/products/raspberry-pi-5/> (accessed on 20 October 2025).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.