remote sensing

Article

Feedback-Assisted Automatic Target and Clutter Discrimination Using a Bayesian Convolutional Neural Network for Improved Explainability in SAR Applications

Nicholas Blomerus [1,2,†], Jacques Cilliers [2,†], Willie Nel [2], Erik Blasch [3] and Pieter de Villiers [1,*,†]

[1] Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0028, South Africa
[2] Council for Industrial and Scientific Research, Pretoria 0184, South Africa
[3] Air Force Research Laboratory, Dayton, OH 45324, USA
* Correspondence: pieter.devilliers@up.ac.za
† These authors contributed equally to this work.

Abstract: In this paper, a feedback training approach for efficiently dealing with distribution shift in synthetic aperture radar target detection using a Bayesian convolutional neural network is proposed. After training the network on in-distribution data, it is tested on out-of-distribution data. Samples that are classified incorrectly with high certainty are fed back for a second round of training. This results in the reduction of false positives in the out-of-distribution dataset. False positive target detections challenge human attention, sensor resource management, and mission engagement. In these types of applications, a reduction in false positives thus often takes precedence over target detection and classification performance. The classifier is used to discriminate the targets from the clutter and to classify the target type in a single step as opposed to the traditional approach of having a sequential chain of functions for target detection and localisation before the machine learning algorithm. Another aspect of automated synthetic aperture radar detection and recognition problems addressed here is the fact that human users of the output of traditional classification systems are presented with decisions made by "black box" algorithms. Consequently, the decisions are not explainable, even to an expert in the sensor domain. This paper makes use of the concept of explainable artificial intelligence via uncertainty heat maps that are overlaid onto synthetic aperture radar imagery to furnish the user with additional information about classification decisions. These uncertainty heat maps facilitate trust in the machine learning algorithm and are derived from the uncertainty estimates of the classifications from the Bayesian convolutional neural network. These uncertainty overlays further enhance the users' ability to interpret the reasons why certain decisions were made by the algorithm. Further, it is demonstrated that feeding back the high-certainty, incorrectly classified out-of-distribution data results in an average improvement in detection performance and a reduction in uncertainty for all synthetic aperture radar images processed. Compared to the baseline method, an improvement in recall of 11.8%, and a reduction in the false positive rate of 7.08% were demonstrated using the Feedback-assisted Bayesian Convolutional Neural Network or FaBCNN.

Keywords: synthetic aperture radar; automatic target recognition; bayesian convolutional neural network; feedback-assisted Bayesian convolutional neural network; explainable artificial intelligence; deep machine learning; epistemic uncertainty; uncertainty estimation

Citation: Blomerus, N.; Cilliers, J.; Nel, W.; Blasch, E.; de Villiers, P. Feedback-Assisted Automatic Target and Clutter Discrimination Using a Bayesian Convolutional Neural Network for Improved Explainability in SAR Applications. *Remote Sens.* **2022**, *14*, 6096. https://doi.org/10.3390/rs14236096

Academic Editors: Chao Wang, Alberto Moreira, Mihai Datcu, Fan Wu and Ruya Xiao

Received: 3 October 2022
Accepted: 24 November 2022
Published: 1 December 2022

1. Introduction

1.1. Overview

In recent years, substantial progress has been made in the development and application of synthetic aperture radar (SAR) sensors and techniques. Military organisations around the world use SAR for joint, intelligence, and surveillance (JISR) operations. JISR operations are often time-critical [1]—it is a process that requires a high level of efficiency and coordination between decision makers and action takers. The process of extracting relevant information, especially targets, from SAR data is a time-consuming process, with highly trained human SAR analysts having to manually evaluate hundreds of kilometres of SAR data. In addition, limitations in sensor hardware may cause challenges such as low resolution (coarser than one meter) images, which result in targets only being represented by a few pixels. These challenges result in delays in SAR-based JISR operations. Consequently, this leads to research into the automation of the process of finding and classifying targets in SAR images by means of automatic target recognition (ATR) of targets within SAR data.

Advancements in machine learning (ML) algorithms and the increased capability of hardware have made the challenge of ATR more practical [2]. As a result, numerous ML techniques have been applied to the problem of ATR applied to SAR images [3–7], with some deep neural network (DNN) implementations achieving a remarkable classification accuracy of 99.3 % on codified data sets [8].

JISR activities can greatly benefit from the use of ML algorithms for ATR using SAR images as it significantly reduces the time to analyse and classify potential targets of interest. In addition to being time-sensitive, JISR tasks may have severe consequences since the decisions made have a direct effect on human lives. However, current state-of-the-art deep learning (DL) algorithms have become "black-box" models that make decisions without any methods for explaining the decisions. This has caused end-users to question the reasoning of these algorithms, especially in applications where lives are at risk [9]. A major drawback of current DL algorithms is that due to their non-transparent nature when an incorrect prediction is made, the model does not provide a reliable indication as to how confident it was in its decision [10]. Discrepancies in confidence values can be caused by the activation layer. For example, traditional softmax layers may cause the whole DL network to suffer from over or under-confident predictions [11]. In addition, when the model is given samples that are not contained within the training dataset, the model can behave in unexpected ways. In the same way that the output of the softmax causes overconfidence, the output of the model for out-of-distribution (OOD) data is often over-confident [12]. A possible solution to this is to use alternative methods to access the uncertainty of the model's prediction. Modern techniques for estimating uncertainty in DL networks include using dropout training [13] and Bayes by Backprop [14]. The Bayes by Backprop technique can be further manipulated into a Convolutional Neural Network (CNN) as shown in [15]. This implementation forms a model that achieves comparable, but slightly worse classification accuracy to traditional CNNs while having the advantage of producing uncertainty estimations and is known as the Bayesian Convolutional Neural Network (BCNN).

In this paper, a method is proposed that leverages uncertainty estimates to feedback data to improve the model. The advantages of using uncertainty estimates from DL algorithms to improve the interpretability (e.g., model construction) between the human user and the ML algorithms are illustrated. Using the uncertainty estimations, a direct comparison of robustness to over-confident predictions between a standard CNN and a BCNN, for in- and out-of-distribution samples is made. A method is proposed to visualise the uncertainty over the SAR image to improve the explainability (e.g., decision outputs) and interpretability of the ML algorithm decisions for human users. A target detector is created using the BCNN with the incorporation of uncertainty estimation.

The rest of the paper is organised as follows: Section 1.2 is a literature review and Section 2 presents the datasets used in this study. In Section 3.1 the theory of the BNN and uncertainty estimation method is presented. The target detection implementation as well as the proposed method to visualise the uncertainty over a region is discussed in Sections 3.2 and 3.4. The feedback-assisted training method is discussed in Section 3.4. The experimental setup is discussed in detail in Section 3.6. This is followed by the results section in Section 4. Lastly, Section 5 presents a discussion and Section 6 conclusions.

1.2. Related Works

1.2.1. Comparison of ATR Systems

In this section, various ATR systems for SAR applications are selected from the literature and are categorised based on the techniques applied by each for target recognition. The selected ATR systems are compared based on their classification performance. Owing to the vast amount of published work on the topic of ATR of SAR images [16], only the systems that utilise feature-based techniques were selected, as they are the main focus of this study. For methods that use the MSTAR dataset, refer to Section 2 for detailed information regarding the different operating conditions under which this dataset was recorded.

Template matching is the simplest method that requires the lowest computational complexity. Recognition is achieved through the matching of SAR data with stored templates of a range of sensor-to-target geometrical variations for each target. The appropriate target is selected using metrics of similarity between the offline templates and input data, such as correlation and matching filters. An example of template-matching is presented in [17], where the attributed scattering centres (ASC) are matched to binary target regions. Image segmentation is performed using basic thresholding to obtain the binary target regions. Then, the binary region is correlated to the stored template region. Lastly, weighted scores are combined from the correlations to determine the appropriate class. An average classification accuracy of 98.34% was achieved using the MSTAR dataset under standard operation conditions (SOC).

An example of a support vector machine implementation is presented in [18]. The training of the SVM used a Gaussian kernel, with the kernel size selected based on the mean Euclidean distance between features in the input data. From the results obtained, the SVM achieved a best classification accuracy of 90.99% while only using three classes of the MSTAR dataset under SOC. Moreover, the variability of the aspect angle had a large effect on the classification accuracy and a trade-off was made between the sector size and the training accuracy.

A representative example of a CNN is given in [19]. The ATR system uses image segmentation methods based on morphological operations in order to reduce background recognition. The CNN is implemented using a large-margin softmax batch normalisation structure which increases separability in the SAR data after pre-processing. Owing to the structure of the network, an increase in the convergence rate is recorded as well as reduced proneness to overfitting. The method was capable of achieving a classification accuracy of 96.44% on the MSTAR dataset under SOC while being robust in extended operation conditions (EOC), such as large depression angles and configuration of the target variants.

Ensemble learning utilises a finite number of different learning methods. Ensembles use the combination of the outputs from multiple learning algorithms to improve prediction capabilities. The main ensemble methods are bagging, stacking, and boosting. An example of a method utilising ensemble methods is given in [20], where a novel pose rectification and image normalisation process is introduced which reduces the variations of the input samples before the feature extraction process. To extract highly discriminative features from ground targets, wavelet decomposition techniques are used. Wavelets allow for a rich edge detection feature set to be extracted that consists of horizontal and vertical edges. Dimensionality reduction is performed to retain the most discriminative features. Decision tree classifiers are utilised to discriminate between the features. A statistical analysis of the input data is used to train each base discriminant tree classifier to support the ensemble learning.

A comparison of feature-based techniques that have been directly applied to ATR using SAR images is presented in Table 1. It was found that the methods that used CNNs, on average, achieved the highest classification accuracy. The top performing CNN implementation [8] slightly outperformed the top performing ensemble implementation [3] using both SOC and EOC for training on the MSTAR dataset. Both implementations contained a CNN and, from the literature, it is apparent that CNNs achieved the best classification performance [21]. Despite being the best performing ATR method in terms of classification accuracy, CNNs are considered to be "black-box" models and are less explainable than BNs and SVMs. Furthermore, the problem of over-confident predictions is still prevalent in CNNs and needs to be considered for ATR systems.

Table 1. Comparison of Various Feature-Based ATR of SAR Techniques.

	Refs	Classifier Method	Dataset	Features	Classification Accuracy
Template-Matching	[17]	Template-Matching using weighted scores	MSTAR	Binary target regions	98.34%
Linear Discriminate Functions	[18]	SVM using Gaussian Kernel	MSTAR using three targets	Image fed into pose estimator The pose of the image is used as the feature vector	90.99 %
Neural Network	[19]	CNN	MSTAR	Image segmentation is performed using morphological operations	96.44%
	[8]	DCNN	MSTAR	Image segmentation is performed and super resolution image obtained using Generative Adverserial Network	99.31%
Ensemble Learning	[3]	Fusion of CNN and SVM	MSTAR	Images are converted to dB and data augmentation is performed	98.56%
	[20]	Ensemble of decision trees using AdaBoost	MSTAR	Combination of texture and edges	97.5%

1.2.2. Explainable Artificial Intelligence

As more ML models are being deployed each day, with increasing levels of complexity, it is apparent that future technological developments can greatly benefit from their use. Technology has come to a pivotal point where the decisions of these models directly affect the lives of humans. Therefore, the demand has increased for the explanation of the decisions made by these ML algorithms [10]. With the increase in the application of AI systems, the introduction of non-transparent models such as DNNs has occurred. These models have been shown to produce impressive results by using efficient training methods and contain a vast number of parameters [22]. With the increase in deployment of these "black-box" models being allowed to make important decisions in various fields, the end-users' need for improved transparency has also increased. The decisions made by these models are often made without reason and are not accompanied by any logical explanations. This can be dangerous in situations where human lives may be affected. Explainable artificial intelligence (XAI) aims to address the problem of the "black-box" model in the following ways: producing models that are explainable while maintaining learning performance, facilitating the user's trust in the algorithms, allowing humans to understand the decision-making process, and aiding in the interpretation of the model's output [9].

The XAI framework contains methods which can be applied to increase the interpretability of DNNs. Such methods can be applied to previously trained networks and are often referred to as post-hoc methods [9]. A method for visual explanations from deep networks is presented in [23], called Gradient-weighted Class Activation Mapping (Grad-CAM), and it utilises the class-specific gradient data entering into the last convolutional layer of a CNN to construct a rough map of the critical information for the classifier in the image. Grad-CAM computes the gradient with respect to the feature maps of the last convolutional layer. Once computed, the gradients are fed back after a global average pool operation is performed to obtain the weights. These weight values are used to extract the critical information of the feature maps for each class. To generate the heat maps, used in this paper, the weights of the features are combined and followed by a ReLu activation since only features with a positive influence on the class of interest are desired.

Grad-CAM is an important XAI tool since it allows for understanding in situations where unexpected predictions are made and, thus, ensuring that the classifier is operating as expected and derives its predictions based on information from the desired target [24]. In recent years, the challenge of explainability in ATR has gained much attention with a significant increase in research addressing this challenge. In [25], an example of the application of an XAI method is presented for the task of image classification using the MSTAR dataset. The CNN model used consisted of three convolutional layers, two max-pooling layers, and one dense layer followed by a softmax layer. Data augmentation is performed by performing various rotations and transitions. The CNN achieved a classification accuracy of 98.78% under SOC. Local Interpretable Model-Agnostic Explanations (LIME) were used to provide model explanations through the visualisation of predictions. LIME allows for the visualisation of the boundary of key characteristics of the target that contributed to the prediction of the CNN.

The quantification of uncertainty in BCNNs provides additional trust to the user through the estimation and visualisation of uncertainty in a model's predictions. The BCNN has been applied in numerous fields such as medicine, finance, computer vision, and surveillance [26–29]. In [29], a novel model called the Bayes-SAR Net is proposed. The BCNN achieved comparable classification accuracy when compared to a CNN, with only a slight decrease in accuracy while gaining the benefits of having uncertainty estimation capabilities. Uncertainty estimates were formed from the mean and covariance of the estimated posterior distribution. The model was trained on polarimetric SAR data and it was concluded that the BCNN was more robust to adversarial noise. Adversarial noise in this instance refers to noise that exploits vulnerabilities in an ML system. In [30], a taxonomy for uncertainty representation and evaluation for modelling and decision-making in information fusion is presented and is further extended in [31]. It contains a discussion on the different types of uncertainties and where they enter a sensing or fusion system. This taxonomy was applied to investigate the effects of uncertainties of the BCNN.

2. Datasets

A focus for contemporary XAI methods is transfer learning, or domain adaptation, between training in one domain and deploying in another (e.g., data collected with different sensors). For comparison purposes with existing publications, use was made of the MSTAR data in this research for training only. Testing was performed using a NATO-SET 250 dataset (to be detailed later), to highlight the challenges of distribution shift. A *distribution shift* is a result of changes in the distributions of the input data. In this particular case, the change results in the switch from the MSTAR to the NATO-SET 250 dataset. If the distribution shift is too large, it can result in a decrease in the accuracy of the model; therefore, it is important to select samples that do not make too many changes to the activation patterns within the models and shift the distribution in a manner that benefits the desired output [32]. In this study, in-distribution data refers to samples that are similar to the training dataset distribution, while OOD data samples do not follow the distribution of the training dataset. The concept of in and out-of-distribution data is better explained with an example from the MSTAR data. Suppose a classifier is trained to classify between United States (US) tanks and armoured personnel carriers (APCs), then the in-distribution data samples are samples of only US tanks and APCs. Subsequently, trying to classify between Soviet era tanks and APCs will result in reduced performance, since samples of Soviet era tanks and APCs belong to the OOD dataset. Soviet era tanks and APCs do not have the same probability distributions over features as the US tanks and APCs, and as such are termed *OOD samples*. Initially, the BCNN was trained on the MSTAR dataset; therefore, the MSTAR initially forms the in-distribution data. During the second round, training samples are selected from the NATO-SET 250 dataset, which forms the OOD. The MSTAR dataset is a standard dataset used throughout the literature and is thus an appropriate benchmark dataset for evaluating the performance of each method [33].

The MSTAR dataset consists of two collections recorded using the X-band Sandia National Laboratory SAR sensor, containing magnitude data [33–36]. The first collections of SAR images contain three targets, the T-72 (T-72 tank), BMP2 (infantry fighting vehicle), and BTR-70 (armoured personnel carrier) and were each collected at depression angles of 15° and 17 °. The second collection contains SAR images of seven targets, the 2S1, BDRM-2, BTR-60, D7, T62, ZIL-131, and ZSU-23/4 recorded at depression angles of 15°, 17°, and 30°. Each SAR image sample is 128 × 128 pixels in size consisting of the magnitude data. Samples from the MSTAR dataset are shown in Figure 1, where the optical images of BMP2, BTR70, T72, BTR-60,2S1, BRDM2, D7, T62, ZIL-131, and ZSU23/4 are shown in the upper panel, and the corresponding SAR images for each target are shown in the corresponding lower panel.

The examples in the figure above use the the two available target configurations, namely, SOC and EOC. The data set for the SOC consists of ten classes with two depression angles of 15° and 17°. There is minimal difference between the two depression angles for SOC of only 2°, but this could lead to variation in radar reflectivity. The EOC data set includes the same ten classes, with the addition of discrete values in depression angles between 45° and 15°, noise variation (range from –10 to 10 dB), occlusion variation (occlusion levels up to 50% of the target), and resolution variation (range resolution from 0.3 m to 0.7 m). Given that the SOC only has a slight variation in depression angle, the target detection/classification performance metrics achieved in this research were higher than when compared to the EOC evaluations. Assume henceforth that each technique was evaluated using the MSTAR dataset unless stated otherwise.

The four SAR scenes used to generate samples for the BCNN detector were supplied by the NATO-SET 250 work group. They are similar to the MSTAR dataset since they contain multiple scenes captured at various depression angles and different orientations; however, they differ in the sensor used, the geographical locations in which they were captured as well as having differing types of terrain. Figure 2 shows the four different scenes from the NATO-SET 250 dataset with bounding boxes around the targets. Both the MSTAR and NATO-SET datasets were captured using horizontal polarisation in X-band. The NATO-SET 250 dataset was captured at a resolution of 0.3 m.

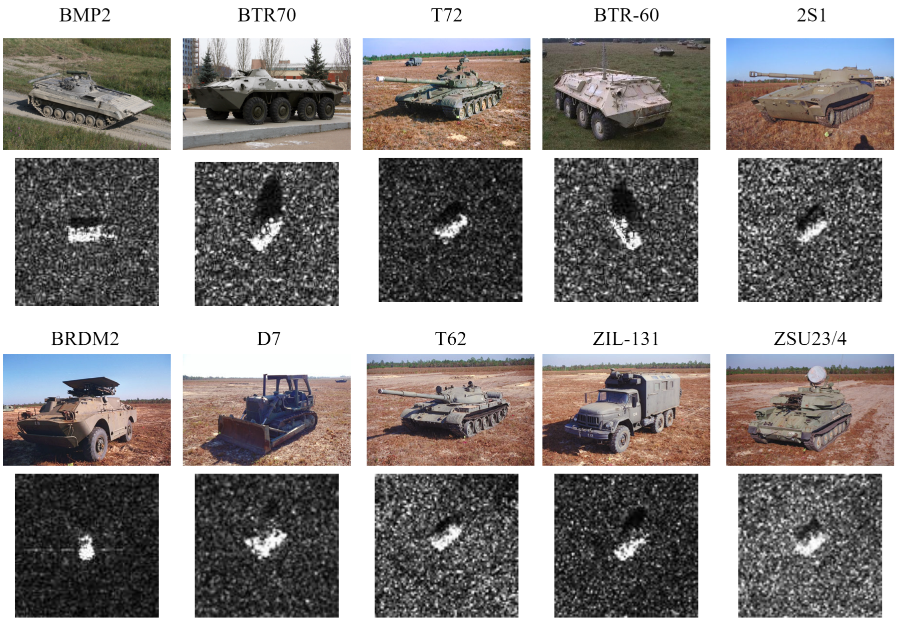


Figure 1. Examples of MSTAR training samples. The MSTAR dataset contains a total of ten targets of various vehicle types. The optical photographs of the targets are shown above their corresponding SAR image.

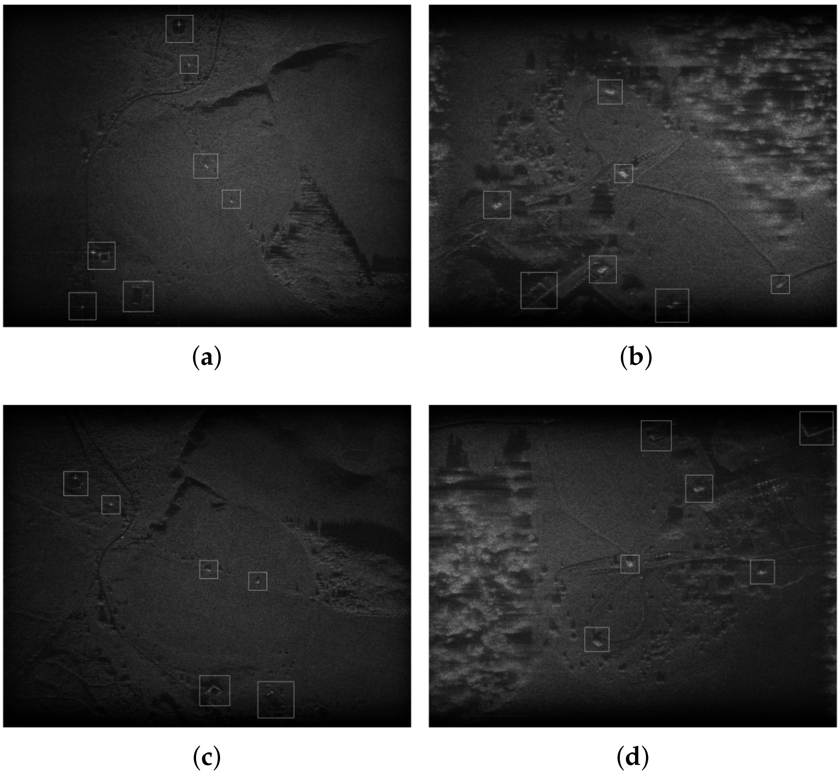
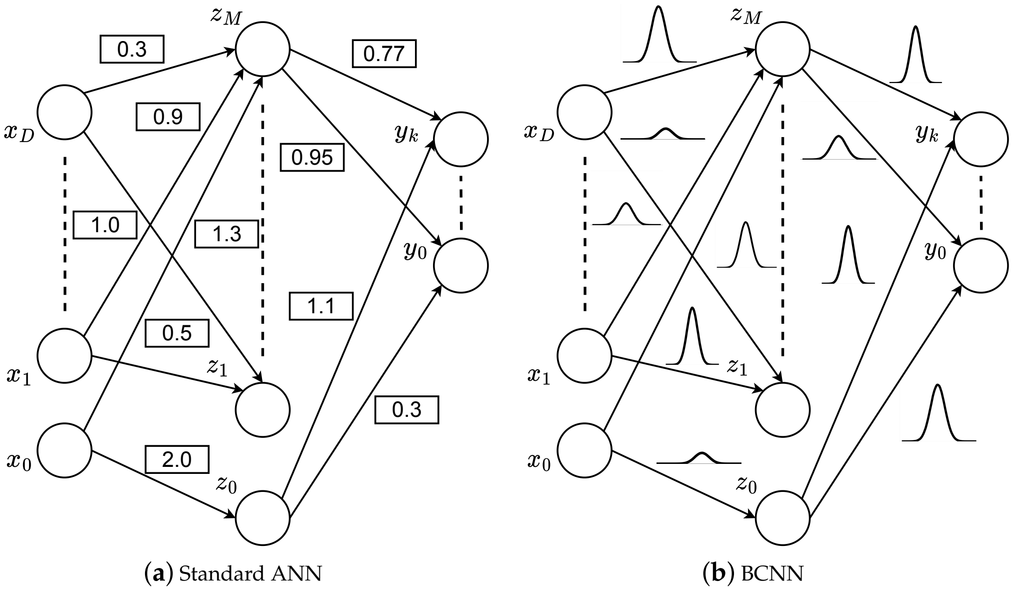


Figure 2. SAR scenes provided from the NATO-SET 250 work group with corresponding bounding boxes around the targets. (**a**) Scene 1 was captured over a large open grass area, it also contains a small portion of trees to the bottom right. (**b**) Scene 2 contains a large collection of trees adjacent to a few man-made structures. (**c**) Scene 3 was captured over a similar region as scene 1 but at a different orientation. This scene highlights more of the tree and hill areas (**d**) Scene 4 is the same area as scene 2 but captured at a different orientation, at this angle more of the forest area is included.

3. Methods

3.1. Bayesian Convolutional Neural Network

The implementation for the BCNN is based on the method of variational inference, namely, Bayes by backpropagation. Bayes by backpropagation is proposed in [14]. It introduces an efficient novel algorithm for regularisation augmented by Bayesian inference on the weights, allowing for the application of a straightforward learning algorithm like back-propagation. It is shown that by introducing uncertainty in the weights, the model gains improved capability by expressing increased uncertainty in areas with little to no data, resulting in a model that is more robust to over-fitting while offering uncertainty estimates through its parameters in the form of probability distributions. In a BCNN network, all of the weights are expressed by probability distributions, in contrast to regular ANNs that use single-valued weights, as shown comparatively in Figure 3.



(**a**) Standard ANN (**b**) BCNN

Figure 3. Side by side comparison of the structures of a standard ANN and a BCNN. The weights of the ANN are represented by discrete values, whereas the BCNN makes use of probability distributions.

Bayes by back-propagation determines the true posterior distribution $p(w|\mathcal{D})$ of the weights w given the training data $\mathcal{D}$. Bayesian inference on the weights of an ANN is intractable due to the vast number of parameters; therefore, an approximate distribution $q_\theta(\mathbf{w})$ is defined and the objective of the training is to determine an approximate distribution as close as possible to the true posterior distribution. Variational inference is applied to learn parameters θ of a distribution on the weights $q(\mathbf{w}|\theta)$. The optimal parameters θ^* are expressed in terms of the Kullback-Leibler divergence as:

$$\theta^* = arg\min_\theta KL[q_\theta(\mathbf{w}|\mathcal{D})||p(w|\mathcal{D})]. \quad (1)$$

This optimisation problem is known as variational free energy [37] and is optimised by the minimisation of the cost function with respect to θ. The cost function is denoted by:

$$\mathcal{F}(\mathcal{D},\theta) = KL[q(\mathbf{w}|\theta)||p(\mathbf{w})] - log\ P(\mathcal{D}|\mathbf{w}), \quad (2)$$

where $p(\mathbf{w})$ is the prior distributions of the weights. The cost function is minimised using gradient descent and variational inference searching for q_θ that is closest to the true posterior. The exact cost can be represented as:

$$\mathcal{F}(\mathcal{D},\theta) = \sum_{i=1}^{n} \log\ q - \theta(\mathbf{w}^{(i)}|\mathcal{D}) - \log\ p(\mathbf{w}^{(i)}) - \log\ p(\mathcal{D}|\mathbf{w}^{(i)}). \tag{3}$$

The weights $\mathbf{w}^{(i)}$ represent the ith Monte Carlo (MC) sample sampled from the variational posterior $q(\mathbf{w}^{(i)}|\theta)$.

The BCNN implementation used in this study is based on [15] as it provides an efficient method to perform variational inference using a CNN. The efficiency is achieved through the use of two convolution operations to determine the mean and variance of the weights. The BCNN implementation introduces a probability distribution over the weights in the convolutional layers, as well as in the weights of the fully-connected layers.

3.2. Target Detector

In traditional ATR systems, a key stage in the processing chain is the "Focus-of-Attention", since it is responsible for evaluating the entire scene for Regions of Interest (ROIs). The attention stage can significantly reduce the computation time required to identify every target in a scene by only passing ROIs to the classification algorithms [38] and, thus, minimising the computationally taxing process of passing each sample through the classification algorithm which may be a DNN. However, for this research, there was no limitation on the computational time and the main focus was on improving the explainability of ATR systems. As a result, the entire SAR scene was used in order for the uncertainty in the detections to be evaluated. Thus, the BCNN is used as a detector to discriminate between targets and clutter, as well as classifying the target types for each detection. Figure 2 shows the four different scenes with bounding boxes around the targets. To train the detector, the data from the MSTAR was used as the targets, and samples were manually selected from regions known not to contain targets in the NATO-SET 250 dataset for the clutter samples. The Feedback-assisted Bayesian convolutional neural network (FaBCNN) was trained to detect two classes—targets and clutter. During the detection process, a sliding window was used to pass samples to the network. At each iteration, data selected by the window was passed through the network and a classification was made. Once a window was correctly classified as a target, a green box was drawn around that sample to indicate that it was correctly detected as a target. Windows correctly detected as clutter are left blank to emphasise the detected targets. All false detections have red boxes drawn around the window. To evaluate the performance of the detector, the precision was calculated for each scene using the correct and incorrect detections.

3.3. Uncertainty Estimation

In order to determine the predicted class, the predictive distribution for the output y^* and the test input x^* are used. The variational predictive distribution is approximated from the predictive distribution, and its variational predictive distribution is given by:

$$q_{\hat{\theta}}(y^* \mid x^*) = \int_{\mathbf{w}} p(y^* \mid x^*, \mathbf{w}) q_{\hat{\theta}}(\mathbf{w}) d\mathbf{w}, \tag{4}$$

and since the integral is intractable, an estimator of the predictive distribution is used:

$$\hat{q}_{\hat{\theta}}(y^* \mid x^*) = \frac{1}{T} \sum_{t=1}^{T} p(y^* \mid x^*, \hat{\mathbf{w}}_t), \tag{5}$$

where w_t is drawn from the variational distribution $q_{\hat{\theta}}$, and T is the number of samples. The variance of the variational predictive distribution is also known as the predictive variance. The variance is also referred to as uncertainty. The uncertainty is separated into aleatoric and epistemic uncertainty with the aleatoric representing the intrinsic randomness in the data while the epistemic uncertainty is generated from the variability in the weights. The combined uncertainty is given by:

$$\mathrm{Var}_{q_{\hat{\theta}}(y^*|x^*)}(y^*) = E_{q_{\hat{\theta}}(y^*|x^*)}\left\{y^{*\otimes 2}\right\} - E_{q_{\hat{\theta}}(y^*|x^*)}(y^*)^{\otimes 2}, \tag{6}$$

where $\otimes$ denotes the outer product. The epistemic uncertainty is a result of the variance of the weights $\mathbf{w}$, given the data [39].

The technique to explicitly compute the uncertainty as two separate types was developed in [40]. However, this method has two key constraints. Firstly, this method estimates the variance of linear predictors, which is not the case for classifiers, instead, it should model the predictive probabilities. Secondly, the aleatoric uncertainty does not factor in the correlations from the diagonal matrix modelling. The challenges are addressed in [41] and the predictive variance is reduced to

$$\mathrm{Var}_{q_{\hat{\theta}}(y^*|x^*)}(y^*) = \underbrace{\frac{1}{T}\sum_{t=1}^{T}\mathrm{diag}(\hat{p}_t) - \hat{p}_t^{\otimes 2}}_{\text{aleatoric}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}(\hat{p}_t - \bar{p})^{\otimes 2}}_{\text{epistemic}}. \tag{7}$$

The mean prediction is $\bar{p} = \frac{1}{T}\sum_{t=1}^{T}\hat{p}_t$ and $\hat{p}_t = \mathrm{softmax}(f_{w_t}(x^*))$, where the softmax is the activation function applied to the output of the model. The uncertainty estimation described above is used in Section 3.4 to generate the uncertainty heat maps and to select high uncertainty incorrect samples to perform the feedback training. Following the processes to generate the uncertainty heat maps and uncertainty feedback, more details on the model initialisation and training are provided in Section 3.5 and lastly, before the results are shown the experimental setup is described in Section 3.6.

3.4. Uncertainty Heat Maps and Feedback

By calculating the uncertainty estimates over the entire scene, a 2-D representation of the uncertainty in the detections is constructed. This 2-D image is then used to determine if the network's confidence over the scene improved when high-confidence incorrect detections are fed back into the model.

The uncertainty heat map shows regions of high and low confidence, with bright areas representing high uncertainty and darker regions representing low uncertainty. This is similar to the Grad-CAM method that visualises regions that contributed the most to the prediction [23] but instead the highlighted regions correspond to the model's uncertainty. Another related paper that focuses on explainability in SAR is [25]. It uses the LIME algorithm to highlight areas in the SAR image that contributed the most to the prediction, whereas the uncertainty heat map approach presented in this paper highlights the areas where classification uncertainty is the highest. The method to generate the uncertainty heat maps is illustrated in Figure 4. Firstly, a sliding 2-D window is applied to the scene. The sliding window method has two parameters—crop size and step size. The crop size is the fixed window size and the step size is the distance the window is displaced after each iteration. The step size of the sliding window is determined through trial and error; however, the lower limit of the window is constrained by the crop size of the dataset, which is 60 × 60. The crop size of the sliding window is set to the crop size of the dataset to allow for more of the targets to be centred in the window. The data selected by the window is fed through the BCNN and using the ensemble of softmax probabilities, the epistemic uncertainty is then calculated using Equation (7) (Figure 4a,b). The epistemic uncertainty values are stored in a 2-D grid, which forms the base of the uncertainty heat map (Figure 4c). The uncertainty heat map is then normalised to unity (Figure 4d). For the uncertainty map to be superimposed onto the test scene, an interpolation function is used to transform the dimension from (44 × 34) to (1360 × 1074) (Figure 4e). First, the uncertainty heat map is scaled to correspond to a maximum brightness of 255, similar to the SAR scene (Figure 4f). Then, the uncertainty heat map is superimposed onto the scene to show regions of high uncertainty. Low-uncertainty regions should have a pixel value of zero and not contribute when superimposed (f).

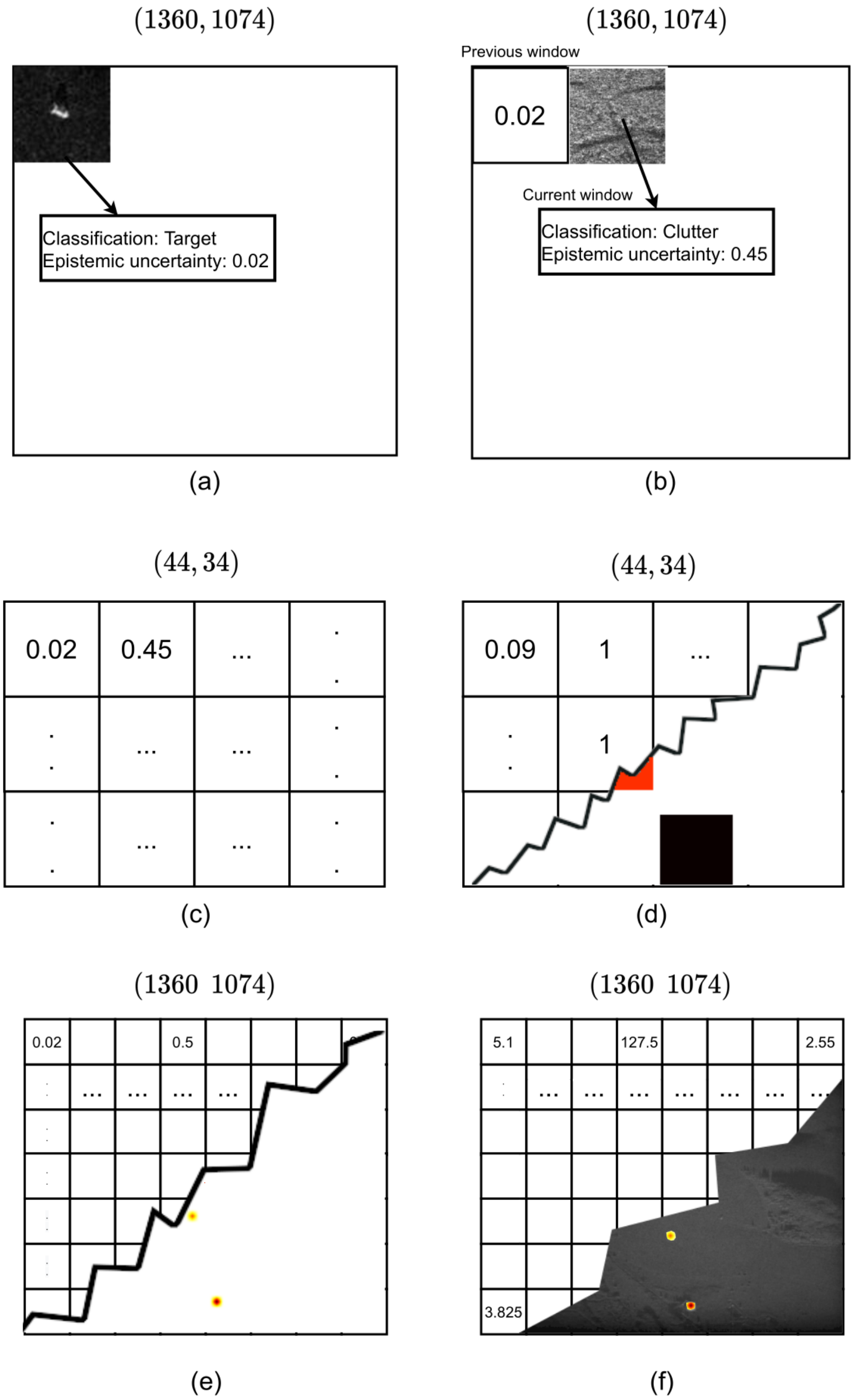


Figure 4. Illustration of the method to produce uncertainty heat maps. (**a**) Each window is passed through the classifier and the classification results and epistemic uncertainty is captured. (**b**) Shows the uncertainty value of the previous window, as well as the current classifier output and uncertainty value. (**c**) Shows all uncertainty values for the SAR scene, which is the base of the uncertainty heat map. (**d**) The uncertainty heat map is normalised. (**e**) The heat map is passed through an interpolation function. (**f**) The interpolated heat map is scaled to a maximum value of 255. Note that in (**d**–**f**) the plot has been split to show the numerical values in the matrix in the upper half and an example of the corresponding heat map or image in the lower half.

The novel contribution of this work is to mitigate distribution shift by feeding back a selection of samples that were incorrectly classified with high certainty in an additional round of training. The hypothesis is that distribution shift is mitigated efficiently by focusing on incorrect predictions that were made with high certainty on the dataset upon which the classifier was not trained, which is important for applications where false positives may have adverse effects on the user. This investigation aims to determine if there is a reduction in incorrect predictions with high certainty after retraining the BCNN with the selected samples that were incorrectly classified with high certainty. The method is described in Algorithm 1. The intersection over union (IoU) threshold is the minimum overlap between ground truth and prediction boxes for the classifier output to be considered a true positive. A popular object detection algorithm called You Only Look Once (YOLO) [42] typically uses a threshold of $iouThreshold$ $=${0.5–0.6}. As a higher IoU threshold (e.g., 0.6) increases the overlap needed for a true positive; therefore, increasing the difficulty of detection—the lower threshold of $iouThreshold = 0.5$ is used instead. The confidence threshold is selected based on the histogram of the uncertainty heat map. The lower quartile of the histogram is used, and the confidence threshold used is $confidenceThreshold = 0.05$.

Algorithm 1 Uncertainty Feedback.

```
Input: SAR scene (1360 × 1074), step size = 20
Output: uncertaintyHeatmap (1360 × 1074)
  Initialisation: Uncertainty heat map = 0
  load BCNN model
  for $i = 0$ to $numXWindows$ do
    for $j = 0$ to $numYWindows$ do
      Move sliding window
      Calculate prediction and epistemic uncertainty
      Update uncertainty heat map
    end for
  end for
  Store all windows coordinates
  Normalise uncertaintyheat map
  for window in all windows do
    for bb in ground truth bounding boxes do
      Calculate IoU between window and bb
    end for
    if any(IoU $\geq iouThreshold$) and epistemicUncertainty $\leq confidenceThreshold$ and prediction = 'clutter' then
      Append window to confident incorrect targets list
    else if all(IoU = 0) and epistemicUncertainty $\leq confidenceThreshold$ prediction = 0 then
      Append window to confident incorrect clutter list
    end if
  end for
  Create dataset using incorrect high confident samples
  Retrain BCNN using a much lower learning rate on new dataset
  Perform target detection
  Generate uncertainty heat map
```

3.5. Model Initialisation and Training

The architecture of the BCNN contains three convolutional layers with three fully-connected layers. A key attribute of the structure is the max pooling layers which were introduced to reduce the overall size of the model [43]. This structure was selected owing to high classification performance and a relatively low number of layers compared to more modern architectures such as VGG16 [44]. Figure 5 illustrates the dimensions at each layer of the network. A brief description of the properties of the architecture follows.

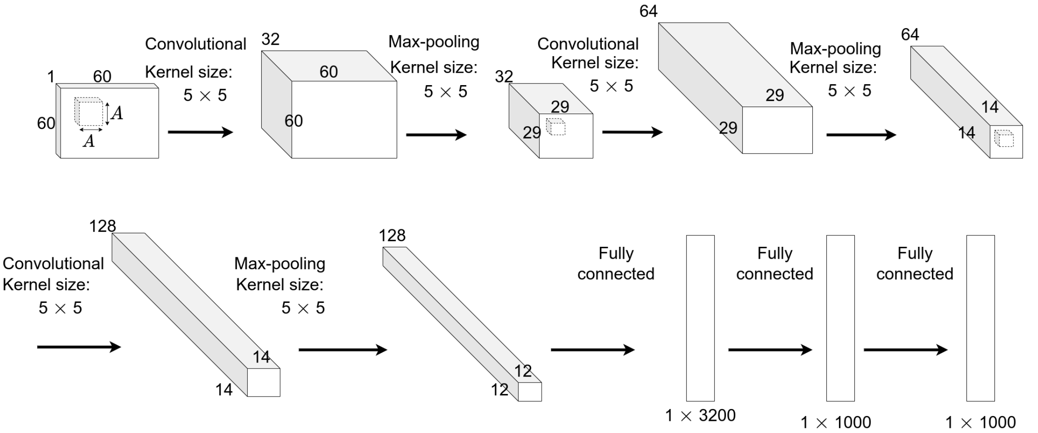


Figure 5. Illustration of input and output dimensions for a three convolutional and three fully-connected architecture. Both the CNN and BCNN use the same architecture.

The BCNN network requires two Gaussian distributions for the initialisation of the priors and the weights. The variance can never be zero, therefore, the variance is represented as σ = softplus(ϵ) where ϵ is randomly selected from a uniform distribution. The prior distribution is initialised with a zero mean and variance of 0.1, while the posterior distribution is initialised with a mean of zero and a ϵ randomly sampled from –5 to 0.1, similar to [14]. From [14], the parameters were found experimentally by using the validation error to find the parameters that resulted in the lowest validation error.

As described in the previous subsection, two convolution operations are performed to determine the mean and variance. In order to ensure that the variance is non-negative, the activation function for the convolutional layers is selected to never output a value equal to or less than zero. The softplus activation function is used as it tends to zero as $x \to -\infty$. The equation for the softplus is given by:

$$\text{softplus}(x) = \frac{1}{\beta}\log(1 + \exp(\beta x)). \tag{8}$$

Here β is set to 1 for all BCNN models trained. The softplus activation is different to the ReLu near zero, where the softplus is smooth and the ReLu goes through zero.

The Adam optimiser is used to perform the updated steps of the variational parameters. Adam is a combination of Root Mean Squared Propagation (RMSprop) and stochastic gradient descent with momentum [45]. Adam benefits from the advantages of adaptive gradient algorithms and RMSprop. It stores the learning rate which improves performance with sparse gradients and the learning rates are adjusted using the mean of the magnitudes of the gradient of weights, making it more resilient to saddle points. The Adam optimiser has proven to converge faster than both RMSprop and stochastic methods [45].

The training method is described in Algorithm 2. During each forward pass, the activation is sampled to calculate the KL divergence. The reparameterisation trick is used to sample from each convolutional layer. Training ends after a fixed number of epochs, and the number of epochs is determined during the hyper-parameter optimisation process along with other hyper-parameters. $numSamples$ is the number of times the activation is sampled per iteration and is fixed to twenty in order to reduce the total training time.

Algorithm 2 Bayes by Backpropagation Learning.

```
Input: Dataset $\mathcal{D} = (x_i, y_i)$, learning rate, batch size
  Initialisation : Priors and posteriors of weights
  for $epoch = 0$ to $numEpochs$ do
    for batch in $numBatches$ do
      for $i = 0$ to $numSamples$ do
        Sample weights
        Calculate KL divergence
      end for
      Calculate ELBO using Equation (3)
      Determine the gradient of the variational parameters $\theta$
      Update the mean $\mu$ and variance $\rho$ of the weights
      Calculate the training and validation accuracy
      Calculate the training and validation loss
    end for
  end for
```

Hyper-parameters are the parameters that control the training of the network and include the learning rate, number of epochs, batch size, and momentum. Hyper-parameter optimisation can be a time-consuming task if performed manually or when using a grid-search method. To improve the efficiency of optimising the networks, a Bayesian model-based optimisation method is employed. Bayesian optimisation methods record previous test results that are used to create a statistical model of the hyper-parameter mappings. This maps the probability of an evaluation for a specific cost function. The hyper-parameter optimisation method used is from [46]. The optimisation is performed for both the traditional CNN and BCNN. The optimised hyper-parameters are listed below for BCNN in Table 2.

Table 2. Optimised Hyper-Parameters for BCNN.

Parameter	Value
Initial priors	$\mu = 0, \sigma = 0.1$
Learning rate	0.00035393
Batch size	8
Number of epochs	105
Patience	11
Activation function	softplus

3.6. Experimental Setup

In order to evaluate the feedback method, the following performance metrics are used: precision, recall and false positive rates. These metrics are selected as the feedback may have different effects on each metric. Since high certainty incorrect samples are fed back into the network, there should be a decrease in false positive detection rates and subsequently an increase in precision and recall. The objective of this is to reduce the risks of AI systems making errors that are safety critical. To assess the degree of improvement a baseline model is used without any additional training with fed back samples. In the evaluation process, three options are used as shown in Figure 6. In the first option high certainty incorrect samples are used for the feedback process. The second option uses random samples and lastly, the third option uses all the available training data. Ten-fold cross-validation is used, where for each fold, a SAR scene in the NATO-SET 250 data is not used until the very end as a test set to evaluate the three options. This ensures that data contamination is eliminated by not using any of the test data for both the initial and feedback rounds of training.

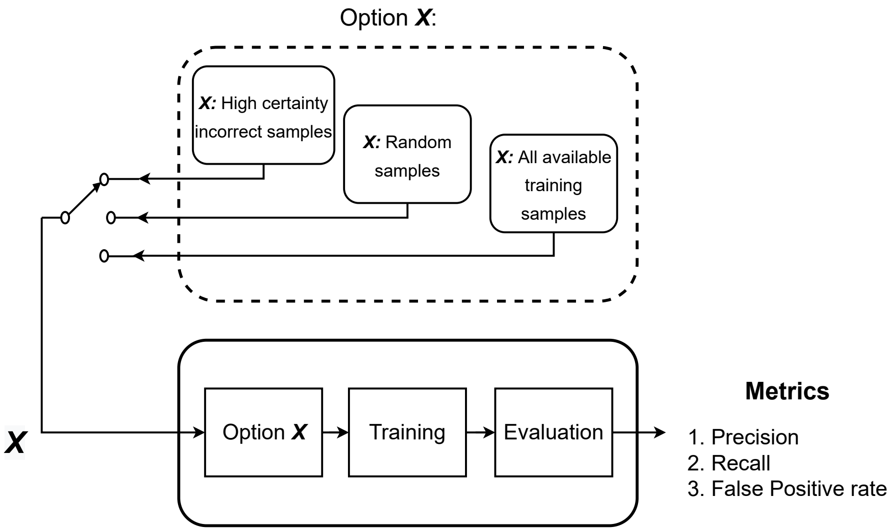


Figure 6. Illustration of options used to evaluate the feedback of high certainty incorrect samples.

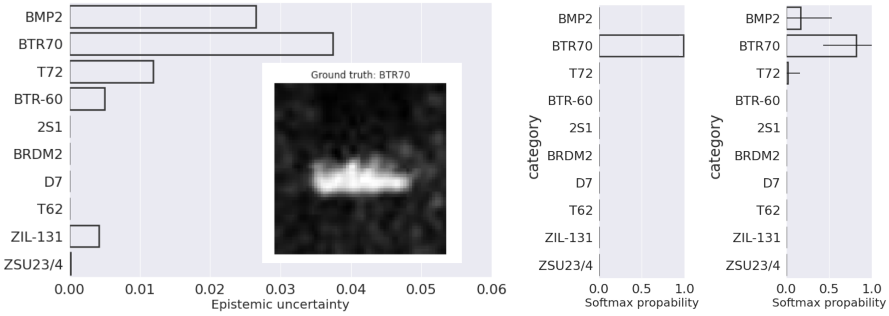
4. Results

4.1. Predictive Uncertainty in BCNNs

In this section, the uncertainty estimates from the BCNN are evaluated for in-distribution and OOD samples. Samples to evaluate the predictive uncertainty were taken from the MSTAR dataset.

The softmax probabilities are shown for both a CNN and BCNN. The BCNN results have added error bars to display the predictive variance. Along with the softmax outputs, the epistemic uncertainty was calculated. For the epistemic uncertainty, the softplus activation function was used and was normalised similarly to the softmax function. The hyper-parameters were selected using a Bayesian model-based optimisation method from [46].

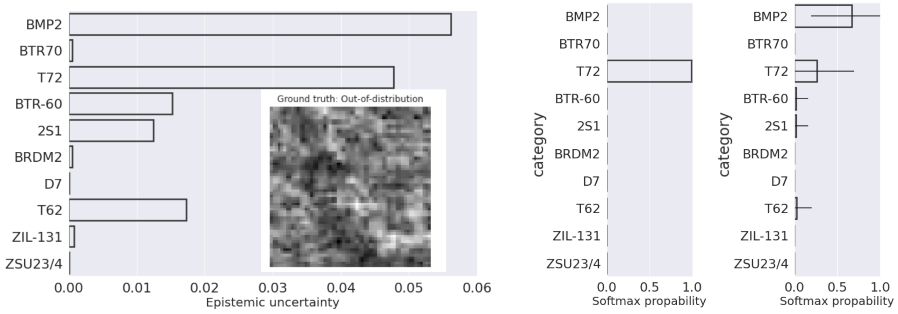
Two test examples are presented to the BCNN and CNN to evaluate the predictive uncertainty. The comparisons of the predictions between the BCNN and CNN are shown for in-distribution and OOD data is shown in Figures 7b and 8b. The corresponding epistemic uncertainty estimates are shown in Figures 7a and 8a.



(**a**) Epistemic uncertainty

(**b**) Softmax probabilities

Figure 7. Comparison of CNN and BCNN predictions for in-distribution sample of a BTR70. (**a**) Epistemic uncertainty of the sample. (**b**) Softmax probability for both the CNN and BCNN. The softmax graph with error bars correspond to the BCNN. The CNN had a predicted softmax probability of 100%.



(**a**) Epistemic uncertainty

(**b**) Softmax probabilities

Figure 8. Comparison of CNN and BCNN predictions for an OOD clutter sample. (**a**) Epistemic uncertainty. This was the highest recorded average uncertainty for both in- and out-of-distribution samples. (**b**) Comparison of CNN and BCNN predictions for OOD. The BCNNs produced five classes with probabilities greater than zero, while the CNN only predicted one class. The OOD input is a clutter sample.

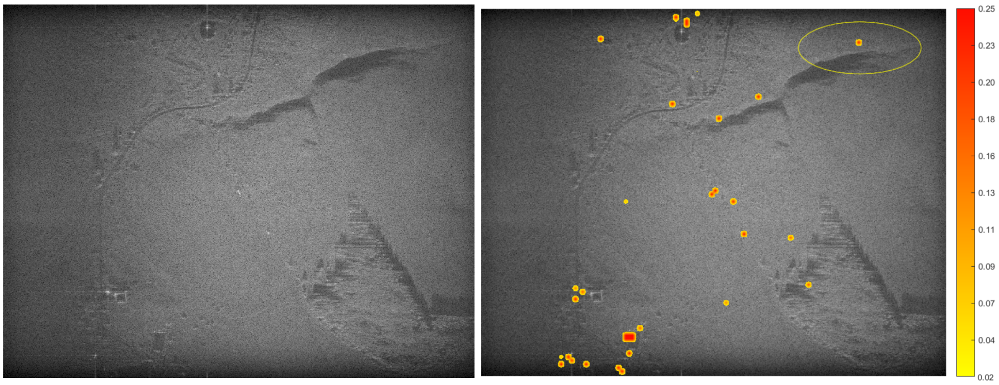
From the initial results, it can be seen that the CNN makes over-confident predictions when compared to the BCNN. This is further confirmed by using the epistemic uncertainty to gain additional information about the confidence of the BCNN. An important observation is that the epistemic uncertainty for the predicted class is the highest in all of the examples. This can be interpreted as the predicted class having the highest uncertainty, and this corresponds with the error bars in the softmax outputs with the longest bar being over the predicted class. The ability to quantify uncertainty comes at a cost, since the CNN achieves a target detection accuracy of 96.8%, whereas the BCNN achieves a slightly reduced accuracy of 93.1% on the MSTAR dataset.

4.2. Uncertainty Heat Maps

The uncertainty heat maps are generated using the method described in Section 4 and are shown in Figures 9b and 10b. The area of dense trees and the regions over the targets have numerous high-uncertainty regions. The highest uncertainty regions are concentrated in the bottom left over the buildings. It is noted that there is also a small cluster of trees with high uncertainty.

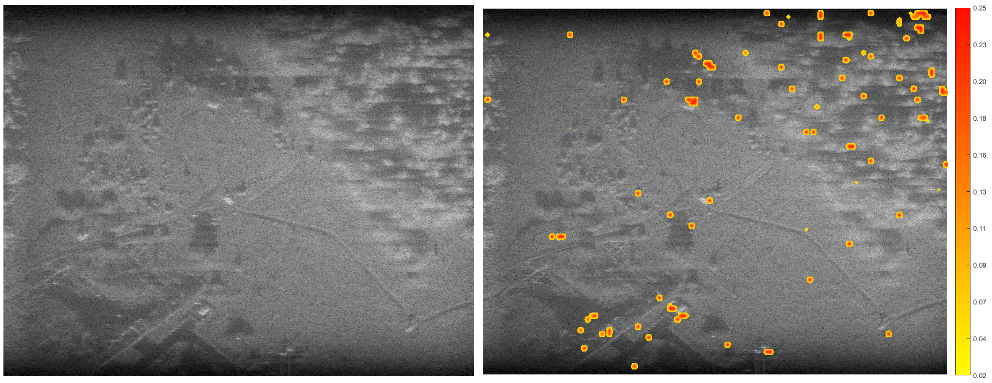
From Figures 9b and 10b, the most distinct elements are the regions that contain targets. Across both scenes, the uncertainty over the targets was the highest. This is attributed to the results in Section 4.1, where it is shown that the predictive uncertainty is always the highest for the predicted class compared to the rest of the classes. It is noted that the predictive uncertainty refers to the epistemic uncertainty and not the softmax probability.

This is apparent even for false detections—such as when the BCNN classifies a tree as a target. However, there were regions with a high uncertainty that were correctly classified as clutter. An example of this can be observed in Figure 9b near the large shadow close to the top right hill in the area contained in the yellow ellipse.



(**a**) (**b**)

Figure 9. (**a**) SAR image of scene 1. (**b**) Uncertainty heat map superimposed onto scene 1, where the yellow ellipse contains low-certainty clutter detection.



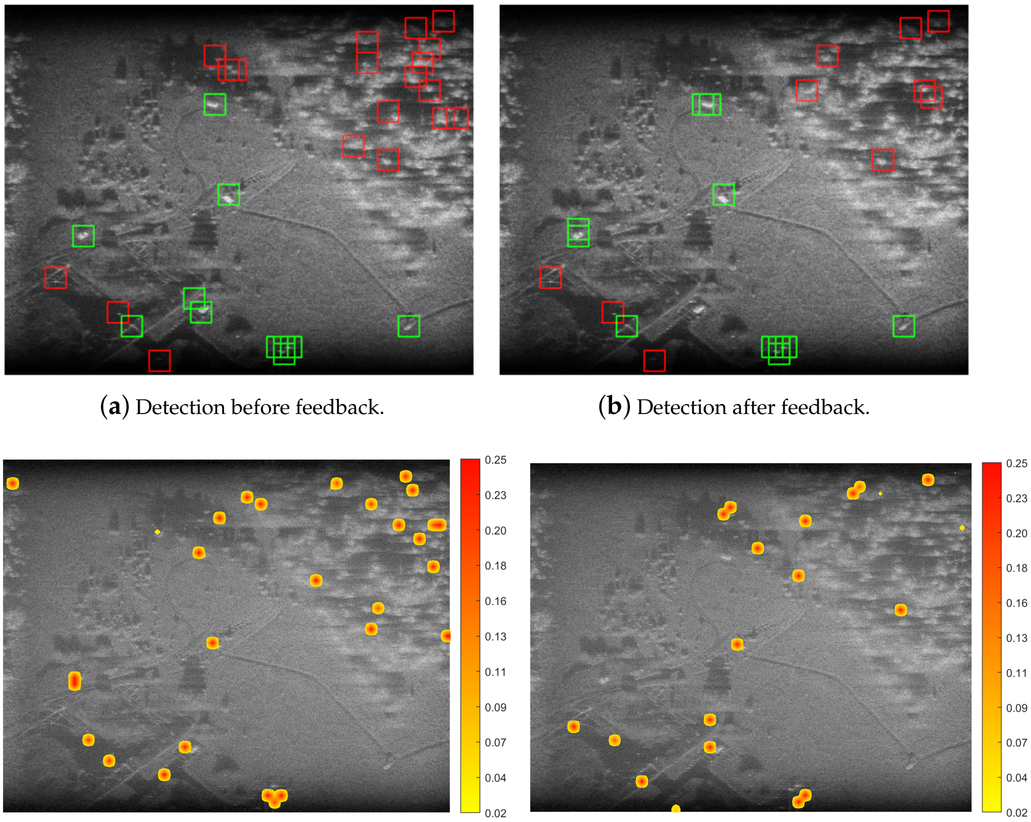
(**a**) (**b**)

Figure 10. (**a**) SAR image of scene 2. (**b**) Uncertainty heat map superimposed onto scene 2. A group of high uncertainty detections are observed over the forest area.

4.3. Feedback of High-Confidence Incorrect Samples

From Figure 2, scenes 2 and 3 were used to illustrate the effects of feeding back high-confident incorrect samples. The results are shown in Figures 11 and 12. The feedback procedure was executed as follows:

1. The detection performance and uncertainty heat maps were determined for each scene before feedback was performed (Figure 12a,c). Multiple runs were performed to gather confident incorrect samples. A total of ten MC runs were performed for the results obtained.
2. Once the BCNN was retrained using fed back samples (this is the feedback process), the detection performance and uncertainty heat map were determined again for comparison (Figure 12b,d). To retrain and not completely alter the current configuration of the weight parameters, the learning rate was reduced by a factor of 25. This ensured that the BCNN was able to adjust its weights appropriately for the new data but not make significant changes that would drastically change the detection performance.



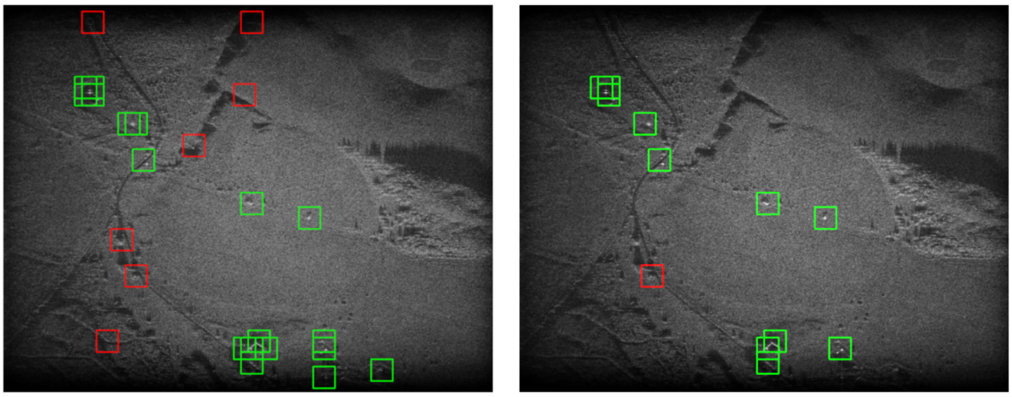
(**a**) Detection before feedback.

(**b**) Detection after feedback.

(**c**) Uncertainty heat map before feedback.

(**d**) Uncertainty heat map after feedback.

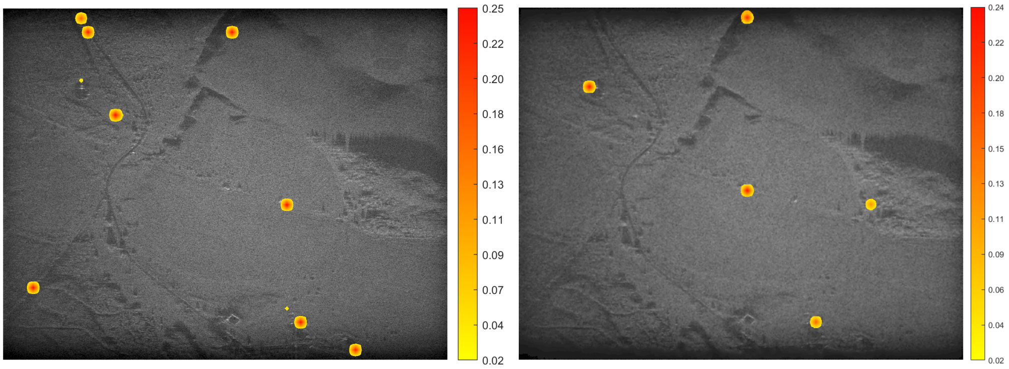
Figure 11. Comparison of the effect of partial retraining BCNN on confident incorrect detections for scene 2. From (**a**,**b**) it can be observed that there was a reduction in the number of false positives (red bounding boxes) over the forest region while maintaining similar detection performance (green bounding boxes) over the remaining regions. Interestingly, there was also a reduction in the number of high uncertainty regions over the forest area between (**c**,**d**).



(**a**) Detection before feedback.

(**b**) Detection after feedback.

Figure 12. *Cont.*



(**c**) Uncertainty heat map before feedback.

(**d**) Uncertainty heat map after feedback.

Figure 12. Comparison of the effect of partial retraining BCNN on confident incorrect detections for scene 3. The main takeaway from (**a**,**b**) is the reduction of false detections in the center of the scene over the hill areas. From (**c**,**d**) there is an reduction in the number of high uncertainty areas.

After feedback, there was a reduction in the number of false positive detections. Noteworthy areas were the line in the centre with the hills. In addition, the number of high uncertainty areas decreased. To compare the precision and uncertainty estimation, ten MC runs were conducted to determine the mean of the precision (units in %), recall (units in %), false positive rate (units in %) and uncertainty (dimensionless quantity) for pre- and post-feedback for all four scenes in Figure 2. The results are tabulated in Tables 3 and 4.

Table 3. Comparison of Detection Performance for Pre- and Post-Feedback.

Method	Precision %	Recall %	False Positive Rate %	Uncertainty
Baseline	75.2 ± 19	8.6 ± 2.35	0.0147 ± 0.0121	0.00107 ± 0.00041
High certainty incorrect samples	**79.7 ± 19**	**9.2 ± 2.08**	0.0129 ± 0.0126	**0.00104 ± 0.00040**
Random samples	78.3 ± 15	8.8 ± 1.91	**0.0127 ± 0.0104**	0.00111 ± 0.00043
All available training samples	72.0 ± 17	8.4 ± 1.99	0.0180 ± 0.0126	0.00131 ± 0.00040

Table 4. Comparison of Average Improvement for Pre- and Post-Feedback.

Method	Precision %	Recall %	False Positive Rate %	Uncertainty %
High certainty incorrect samples	**5.958**	**7.079**	−11.803	**−2.942**
Random samples	4.091	1.898	**−13.397**	3.82
All available training samples	1.481	5.65	1.124	8.772

From Table 4 it can be concluded that the option of high certainty incorrect samples achieved the highest average precision/recall and second lowest average false positive rate. The best improvement in the false positive rate was for the random samples option, but the corresponding improvement in precision and recall was low compared to the high uncertainty incorrect option.

5. Discussion

5.1. Results

In this paper, a method is proposed that feeds back uncertainty estimates from a BCNN. Then using the uncertainty estimates an investigation is performed to determine the effects of feeding back high-confident incorrect samples on the performance of a detector.

In Section 4.1, the results illustrated the use of the predictive variance of the BCNN, with a comparison between predictions made from a CNN and the BCNN for in- and out-of-distribution samples. From the comparison, it is observed that the BCNN is appropriately confident in its predictions, as is apparent from the softmax outputs and the average epistemic uncertainty of the BCNN. For the in-distribution samples, there is an increased spread in the softmax probabilities, whereas the CNN was over-confident having absolute certainty in a single class.

This trend was also prevalent for the OOD samples. In addition, it was recorded that there is a significant increase in the average epistemic uncertainty, which further indicates that the BCNN was less confident when processing the OOD samples.

It is apparent that the CNN has no implicit method of compensating for the distribution shift in the OOD data, and the BCNN provides a solution through uncertainty estimation. When using epistemic uncertainty, it is possible to allow the BCNN to withhold its decision when the uncertainty is above a specific threshold. This allows for undecided samples to be evaluated by a human specialist rather than making an incorrect overconfident prediction.

The uncertainty heat maps provide a 2-D visualisation of the confidence of the model over each region, where the brighter regions indicate a higher uncertainty and the dimmer regions correspond to a low uncertainty. The regions of high uncertainty appear in areas where target detection is more challenging, such as large areas of trees or areas that cast deep shadows. The high uncertainty over the targets is a result of the epistemic uncertainty where the predictive class always has the highest uncertainty.

The effect of feeding back confident incorrect samples resulted in fewer missed detections. This can be observed with the overall increase in precision shown in Table 3 for both scenes. However, it also resulted in fewer correct detections, but the average precision was improved. Compared to the baseline method, an improvement in recall of 7.08%, and a reduction in the false positive rate of 11.8% was demonstrated. This was to be expected as the model was retrained on data that had previously caused false detections. The feedback of confident incorrect samples reduces the number of high-uncertainty predictions for each scene. A significant decrease in the high-uncertainty regions is observed in Figure 12d. This may be attributed to the decrease in the number of target detections. However, regions with known targets have a similar uncertainty to the uncertainty map before the feedback, which indicates that the network had learnt that those samples were indeed targets and adjusted its predictions to compensate for incorrect detections. In Figure 12d where learning is observed, where the number of missed detections in the centre of the image is significantly reduced after the feedback, while the network was still able to detect the majority of the targets correctly.

From Table 4, it can be seen that the average epistemic uncertainty is reduced after retraining of the network. Hence, the network is more confident regarding its predictions. It is observed that the feedback of high-uncertainty incorrect samples is beneficial to the network. The FaBCNN may not be practical during real-time applications where ground truth may not be available , but it is practical in a controlled environment during the training of the network. The FaBCNN may assist in training and evaluation to improve detection performance and uncertainty estimation. Finally, the FaBCNN will work in military operations where one might have intelligence about a small portion of the enemy deployment i.e., one's own forces can visually recognise the enemy targets or have intelligence about a portion of the enemy formation, but obviously, this only covers a small area. A drawback of the FaBCNN is the increased computational cost that is caused by representing the weights as Gaussian distributions and not as single valued weights.

The computer used to train and evaluate the model was an AMD Ryzen 3900 12-core processor with a Nvidia Geforce RTX 2070 Super. The average time to train and perform inference on a single sample was recorded for both a CNN and the FaBCNN and shown in Table 5.

Table 5. Computational time of FaBCNN versus a CNN with the same architecture.

Operation	CNN (s)	FaBCNN (s)
Training of target detector	150.707	921.688
Feedback training	N/A	711.096
Single inference	0.000593	0.006836

There is almost an order of magnitude increase in both the training and inference times of the FaBCNN compared to the CNN.

5.2. Future Research

The FaBCNN implementation uses a rudimentary sliding window to perform the target detection and uncertainty heat map generation, and the approach consisted of feeding the individual windows into the BCNN to estimate the uncertainty. The FaBCNN method is computationally expensive as the classifier computes the output for each window position. Although computational complexity was not the focus of this research, the combination of the reduction in computational complexity combined with improved explainability could be an avenue for future research. Current detection algorithms such as YOLO have been shown to perform with a high degree of speed and precision [42,47]. In addition, YOLO factors in the targets with different resolutions and aspects and is adaptable to complex datasets with multiple overlapping classes. The next step would be the incorporation of the FaBCNN with a YOLO implementation to achieve a state-of-the-art detection algorithm with the benefits of the improvement in explainability through uncertainty estimations.

Lastly, an extension of the FaBCNN to operate on complex-valued data should be investigated. In this paper, only the magnitude data was used for training and evaluation. It has been shown that a significant amount of information is contained in the complex valued data compared to the magnitude-only data [48]. Implementing a complex-valued BCNN should improve the detection performance of the network, especially for the application of ATR using SAR measurements. This approach could also be extended to incorporate polarisation data which has been shown to improve classification performance in radar systems [49–51]. Another avenue of pursuit would be the investigation of training the BCNN on synthetic data generated using electromagnetic modelling tools suitable for electrically large targets [52,53], and then evaluating the BCNN performance on measured data. This data augmentation would give a military user the ability to train the BCNN for expected targets which have not been measured by the radar yet.

6. Conclusions

In this paper, the uncertainty estimations from a BCNN are evaluated and a method for feedback based on the uncertainty is proposed as a second training step, called the FaBCNN. The predictive uncertainty indicates that the BCNN makes fewer over-confident predictions than the CNN while providing insight into the confidence of its predictions. When both networks are presented with samples that are in and out-of-distribution to the MSTAR dataset, the BCNN demonstrates a significant improvement over the CNN with regard to over-confidence. The softmax value of the CNN implies an over-confident prediction, allocating a 100% probability to a single incorrect class. However, the BCNN probabilities are much less peaked and with the aid of the epistemic uncertainty, it is apparent that the network is not over-confident. This difference when testing on OOD data emphasises the necessity for alternative DNN implementation such as the BCNN, for its uncertainty estimation capabilities and increased robustness to over-confident predictions.

As a result, the BCNN is capable of dealing with OOD samples and responds accordingly by refusing to classify them. In this situation, human operators may be notified to address the uncertain sample. This is in contrast to traditional CNNs that would proceed with an incorrect high-confidence prediction.

The FaBCNN demonstrated that it can distinguish targets from clutter over various SAR scenes. Compared to the baseline method (BCNN), an improvement in recall of 11.8%, and a reduction in the false positive rate of 7.08% were demonstrated using the FaBCNN. It was found that the detector was able to correctly detect the majority of the targets. The regions in the images containing trees resulted in the highest number of missed detections. This is a result of the similarities between single trees and targets in the MSTAR dataset since they have a shadow with a brighter region in the centre.

The uncertainty heat maps provide a 2-D visualisation of the confidence of the model over each region where the brighter regions indicate a higher uncertainty and the dimmer regions correspond to a low uncertainty. The regions of high uncertainty appear in areas in which target detection is challenging, such as large areas of trees or areas that cast deep shadows. In the experiments performed in this work, the high uncertainty over the targets was a result of the epistemic uncertainty, where the predictive class always had the highest uncertainty. The uncertainty maps improve the explainability of the system outputs by utilising uncertainty estimations. The FaBCNN is able to provide the user with an indications of how confident it is through the visualisation of the uncertainty overlayed onto the SAR scene. Thus, the detection map, supplemented with an uncertainty heat map, allows the user to have more trust in the target detector. Ultimately, this is a step in the direction of improving the current ML methodologies to foster increased confidence, interpretability, and transparency.

An additional advantage of the uncertainty estimates is that they may be used to improve the performance of the network and reduce the number of high-uncertainty predictions. It was found that by feeding back high-confident incorrect samples, the precision of the detector is improved and an overall reduction in average epistemic uncertainty is observed. The FaBCNN may be used as the last step before a model is deployed to make small adjustments to the network to reduce the number of high-confident incorrect detection or adapt to an unseen set of targets.

Author Contributions: Conceptualization, N.B., E.B. and P.d.V.; Methodology, N.B.; Validation, J.C.; Investigation, N.B.; Data curation, W.N. and P.d.V.; Writing—original draft, N.B. and P.d.V.; Writing—review & editing, J.C.; Visualization, J.C.; Supervision, P.d.V.; Project administration, E.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Radar and Electronic Warfare department at the CSIR.

Data Availability Statement: The NATO-SET 250 dataset is not publicly available; however, the MSTAR dataset can be found at the following url: https://www.sdms.afrl.af.mil/index.php?collection=mstar (accessed on 5 January 2022).

Acknowledgments: The authors would like to thank the NATO-SET 250 work group for providing the dataset in which the CSIR participated.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ANN	Artificial Neural Network
APC	Armoured Personal Carrier
ATR	Automatic Target Recognition
BN	Bayesian Network
BNN	Bayesian Neural Network
BCNN	Bayesian Convolutional Neural Network
CNN	Convolutional Neural Network
CSIR	Council for Scientific and Industrial Research
DL	Deep Learning
EOC	Extended Operating Conditions
FaBCNN	Feedback-assisted Bayesian Convolutional Neural Network
Grad-CAM	Gradient-Weighted Class Activation Mapping

JISR	Joint Intelligence, Surveillance, and Reconnaissance
MAP	Maximum A Posterior
ML	Machine Learning
MC	Monte Carlo
MSTAR	Moving and Stationary Target Acquisition and Recognition
NATO-SET	North Atlantic Treaty Organisation Sensors and Electronic Technology
OOD	Out-Of-Distribution
RCS	Radar Cross Section
SAR	Synthetic Aperture Radar
SOC	Standard Operating Condition
XAI	eXplainable Artificial Intelligence
YOLO	You Only Look Once

References

1. Berens, P. *Introduction to Synthetic Aperture Radar (SAR)*; NATO: Brussels, Belgium, 2006; pp. 1–10.
2. Majumder, U.; Blasch, E.P.; Garren, D. *Deep Learning for Radar and Communications Automatic Target Recognition*; Artech: London, UK, 2020.
3. Wagner, S. Combination of convolutional feature extraction and support vector machines for radar ATR. In Proceedings of the Fusion 2014—17th International Conference On Information Fusion, Salamanca, Spain, 7–11 July 2014; pp. 1–6.
4. Flórez-López, R. Reviewing RELIEF and its extensions: A new approach for estimating attributes considering high-correlated features. In Proceedings of the IEEE International Conference On Data Mining, Maebashi City, Japan, 9–12 December 2002; pp. 605–608.
5. Zhang, G. Neural networks for classification: A survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2000**, *30*, 451–462. [CrossRef]
6. Sain, S.; Vapnik, V. *The Nature of Statistical Learning Theory*; Springer: New York, NY, USA, 2000.
7. Tang, Y.; Srihari, S. Efficient and accurate learning of Bayesian networks using chi-squared independence tests. In Proceedings of the International Conference on Pattern Recognition (ICPR), Tsukuba, Japan, 11–15 November 2012; pp. 2723–2726.
8. Shi, X.; Zhou, F.; Yang, S.; Zhang, Z.; Su, T. Automatic Target Recognition for Synthetic Aperture Radar Images Based on Super-Resolution Generative Adversarial Network and Deep Convolutional Neural Network. *Remote Sens.* **2019**, *11*, 135. [CrossRef]
9. Delser, J. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Inf. Fusion* **2020**, *58*, 82–115.
10. Russel, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Prentice Hall: Upper Saddle River, NJ, USA, 2003.
11. Haas, J.; Rabus, B. Uncertainty Estimation for Deep Learning-Based Segmentation of Roads in Synthetic Aperture Radar Imagery. *Remote Sens.* **2021**, *13*, 1472. [CrossRef]
12. Inkawhich, N.A.; Davis, E.K.; Inkawhich, M.J.; Majumder, U.K.; Chen, Y. Training SAR-ATR Models for Reliable Operation in Open-World Environments. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 3954–3966. [CrossRef]
13. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Proceedings of the ICML'16: Proceedings of the 33rd International Conference on International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016.
14. Blundell, C.; Cornebise, J.; Wierstra, D. Weight uncertainty in neural networks. In Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; Volume 37, pp. 1613–1622.
15. Shridhar, K.; Laumann, F.; Liwicki, M. A Comprehensive Guide to Bayesian Convolutional Neural Network with Variational Inference. *arXiv* **2019**, arXiv:1901.02731.
16. Blasch, E.P. Review of Recent Advances in AI/ML using the MSTAR data. *Proc. SPIE* **2020**, *11393*, 53–63.
17. Tan, J.; Fan, X.; Wang, S.; Ren, Y. Target Recognition of SAR Images via Matching Attributed Scattering Centers with Binary Target Region. *Sensors* **2018**, *18*, 3019. [CrossRef]
18. Zhao, Q.; Principe, J.C. Support vector machines for SAR automatic target recognition. in *IEEE Trans. Aerosp. Electron. Syst.* **2001**, *37*, 643–654. [CrossRef]
19. Zhou, F.; Wang, L.; Bai, X.; Hui, Y. SAR ATR of Ground Vehicles Based on LM-BN-CNN. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 7282–7293. [CrossRef]
20. Zhao, X.; Jiang, Y.; Stathaki, T. Automatic Target Recognition Strategy for Synthetic Aperture Radar Images Based on Combined Discrimination Trees. *Comput. Intell. Neurosci.* **2017**, *2017*, 7186120. [CrossRef] [PubMed]
21. Zhang, W.; Zhu, Y.; Fu, Q. Adversarial Deep Domain Adaptation for Multi-Band SAR Images Classification. *IEEE Access* **2019**, *7*, 78571–78583. [CrossRef]
22. Soldin, R.J. SAR Target Recognition with Deep Learning. In Proceedings of the IEEE Appl. Imag. Pattern Recognition Workshop (AIPR), Washington, DC, USA, 9–11 October 2018; pp. 1–8. . [CrossRef]
23. Selvaraju, R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.***2020**, *128*, 336–359. [CrossRef]
24. Blasch, E.P. Deep Learning in AI for Information Fusion Panel Discussion. *Proc. SPIE* **2019**, *11018*, 110180Q.

25. Mandeep, H.; Pannu, S.; Malhi, A. Deep learning-based explainable target classification for synthetic aperture radar images. In Proceedings of the 2020 13th International Conference on Human System Interaction (HSI), Tokyo, Japan, 6–8 June 2020; pp. 34–39.
26. Zhao, G.; Liu, F.; Oler, J.; Meyerand, M.; Kalin, N.; Birn, R. Bayesian convolutional neural network based MRI brain extraction on nonhuman primates. *Neuroimage* **2018**, *175*, 32–44. [CrossRef] [PubMed]
27. Ticknor, J. A Bayesian regularized artificial neural network for stock market forecasting. *Expert Syst. Appl.* **2013**, *40*, 5501–5506. [CrossRef]
28. Tursunov, A.; Mustaqeem, J.Y.; Choeh, J.Y.; Kwon, S. Age and Gender Recognition Using a Convolutional Neural Network with a Specially Designed Multi-Attention Module through Speech Spectrograms. *Sensors* **2021**, *21*, 5892. [CrossRef]
29. Dera, D.; Rasool, G.; Bouaynaya, N.; Eichen, A.; Shanko, S.; Cammerata, J.; Arnold, S. Bayes-SAR Net: Robust SAR image classification with uncertainty estimation using bayesian convolutional neural network. In Proceedings of the IEEE International Radar Conference (RADAR), Washington, DC, USA, 28–30 April 2020; pp. 362–367.
30. De Villiers, J.; Jousselme, A.; Pavlin, G.; Laskey, K.; Blasch, E.; Costa, P. Uncertainty evaluation of data and information fusion within the context of the decision loop. In Proceedings of the 19th International Conference on Information Fusion (FUSION), Heidelberg, Germany, 5–8 July 2016; pp. 766–773.
31. De Villiers, J.; Pavlin, G.; Jousselme, A.L.; Maskell, S.; Waal, A.; Laskey, K.; Blasch, E.; Costa, P. Uncertainty representation and evaluation for modelling and decision-making in information fusion. *J. Adv. Inf. Fusion* **2018**, *13*, 198–215.
32. Quionero-Candela, J.; Sugiyama, M.; Schwaighofer, A.; Lawrence, N.D. *Dataset Shift in Machine Learning*; MIT Press: Cambridge, MA, USA, 2009.
33. Lewis, B.; Scarnati, T.; Sudkamp, E.; Nehrbass, J.; Rosencrantz, S.; Zelnio, E. A SAR dataset for ATR development: The synthetic and measured paired labeled experiment (SAMPLE). In *Algorithms for Synthetic Aperture Radar Imagery XXVI*; SPIE: Baltimore, MD, USA, 2019; pp. 39–54.
34. DARPA. Moving and Stationary Target Acquisition Recognition (MSTAR), Program Review, Denver, 1996. Available online: https://www.sdms.afrl.af.mil/index.php?collection=mstar&page=targets (accessed on 5 January 2021).
35. Blasch, E.P. Fusion of HRR and SAR information for Automatic Target Recognition and Classification. In Proceedings of the International Conference on Information Fusion, Sunnyvale, CA, USA, 6–9 July 1999; pp. 1221–1227
36. Blasch, E.P. Assembling an Information-fused Human-Computer Cognitive Decision Making Tool. *IEEE Aerosp. Electron. Syst. Mag.* **2000**, *15* 11–17. [CrossRef]
37. Neal, R.; Hinton, G. A View Of The Em Algorithm That Justifies Incremental, Sparse, And Other Variants. In *Learning in Graphical Models*; Springer: Berlin, Germany, 2000; Volume 89.
38. Kaplan, L. Improved SAR target detection via extended fractal features. *IEEE Trans. Aerosp. Electron. Syst.* **2001**, *37*, 436–451. [CrossRef]
39. Tomsett, R.; Preece, A.; Braines, D.; Cerutti, F.; Chakraborty, S.; Srivastava, M.; Pearson, G.; Kaplan, L. Rapid Trust Calibration through Interpretable and Uncertainty-Aware AI. *Patterns* **2020**, *1*, 100049. [CrossRef] [PubMed]
40. Kendall, A.; Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? In Proceedings of the 31st International Conference On Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Brooklyn, NY, USA: 2017; pp. 5580–5590.
41. Kwon, Y.; Won, J.; Kim, B.; Paik M. Uncertainty quantification using Bayesian neural networks in classification: Application to ischemic stroke lesion segmentation. *IEEE Access* **2018**, *6*, 1–6.
42. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
43. Krizhevsky, A.; Sutskever, I.; Hinton, G. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
44. Simonyan, K.; Zisserman, A. A Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
45. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference On Learning Representations, ICLR 2015—Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015.
46. Pelikan, M.; Goldberg, D.; Cantú-Paz, E. BOA: The Bayesian optimization algorithm (department of general engineering. In Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation, San Francisco, CA, USA, 12–16 July 1999; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 1999; pp. 525–532.
47. Huang, Q.; Zhu, W.; Li, Y.; Zhu, B.; Gao, T.; Wang, P. Survey of target detection algorithms in SAR images. In Proceedings of the IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 March 2021; Volume 5, pp. 175–1765.
48. Cilliers, J.E. Information Theoretic Limits on Non-Cooperative Airborne Target Recognition by Means of Radar Sensors. Ph.D. Thesis, Department of Electronic and Electrical Engineering, University College London, London, UK, 2018.
49. Novak, L.M.; Halversen, S.D.; Owirka, G.; Hiett, M. Effects of polarization and resolution on SAR ATR. *IEEE Trans. Aerosp. Electron. Syst.* **1997**, *33*, 102–116. [CrossRef]
50. Cilliers, J.E.; Smit, J.C.; Baker, C.J.; Woodbridge, K. On the gain in recognition performance due to the addition of polarisation in an X-band high range resolution radar evaluated for F-18 and F-35 targets using asymptotic EM techniques. In Proceedings of the 2015 IEEE Radar Conference (RadarCon), Arlington, VI, USA, 10–15 May 2015; pp. 1296–1299;. [CrossRef]

51. Cilliers, J.E.; Potgieter, M.; Blaauw, C.; Odendaal, J.W.; Joubert, J.; Woodbridge, K.; Baker, C.J. Comparison of the mutual information content between the polarimetric monostatic and bistatic measured RCS data of a 1:25 boeing 707 model. In Proceedings of the 2020 IEEE International Radar Conference (RADAR), Washington, DC, USA, 28–30 April 2020; pp. 389–394. [CrossRef]
52. Smit, J.C.; Cilliers, J.E.; Burger, E.H. Comparison of MLFMM, PO and SBR for RCS investigations in radar applications. In Proceedings of the IET International Conference on Radar Systems (Radar 2012), Glasgow, UK, 22–25 October 2012; pp. 1–5. [CrossRef]
53. Potgieter, M.; Cilliers, J.E.; Blaauw, C. The use of sigma hat for modelling of electrically Large practical radar problems. In Proceedings of the 2020 IEEE International Radar Conference (RADAR), Washington, DC, USA, 28–30 April 2020; pp. 186–191. [CrossRef]