Optical Engineering

OpticalEngineering.SPIEDigitalLibrary.org

Integrating deep learning-based data driven and model-based approaches for inverse synthetic aperture radar target recognition

Rajkumar Theagarajan
Bir Bhanu
Tugba Erpek
Yik-Kiong Hue
Robert Schwieterman
Kemal Davaslioglu
Yi Shi
Yalin E. Sagduyu

SPIE.

Rajkumar Theagarajan, Bir Bhanu, Tugba Erpek, Yik-Kiong Hue, Robert Schwieterman, Kemal Davaslioglu, Yi Shi, Yalin E. Sagduyu, "Integrating deep learning-based data driven and model-based approaches for inverse synthetic aperture radar target recognition," *Opt. Eng.* **59**(5), 051407 (2020), doi: 10.1117/1.OE.59.5.051407

Integrating deep learning-based data driven and model-based approaches for inverse synthetic aperture radar target recognition

Rajkumar Theagarajan,[a,*] Bir Bhanu,[a] Tugba Erpek,[b]
Yik-Kiong Hue,[b] Robert Schwieterman,[b] Kemal Davaslioglu,[b]
Yi Shi,[b] and Yalin E. Sagduyu[b]
[a]University of California, Center for Research in Intelligent Systems, Riverside, California, United States
[b]Intelligent Automation, Inc., Rockville, Maryland, United States

Abstract. We explore the blending of model-based and deep learning approaches for target recognition in inverse synthetic aperture radar (ISAR) imagery. It evaluates five different approaches, namely, a model-based geometric hashing approach, a supervised deep learning approach, and three different blending models that fuse the model-based and deep learning approaches. The model-based approach extracts scattering centers as features and requires domain experts to identify and characterize important features of a target, which makes the training process very costly and hard to generalize when the image quality degrades in low signal-to-interference-plus-noise-ratio conditions. Next, a deep learning algorithm using a convolutional neural network is considered to extract the spatial features when raw ISAR data are used as input. This approach does not need an expert and only requires the labels of images for training. Finally, three model-based and deep learning approaches are blended together at the feature level and decision level to benefit from the advantages of both approaches, achieving a higher performance. The results show that the blending of the two approaches achieves a high performance while providing explainable inferences. The performance of the five different approaches is evaluated under varying conditions of occlusion, clutter, masking of the target, and adversarial attacks. It is empirically shown that the model-based and deep learning approaches are able to complement each other and can achieve better classification accuracy upon fusing the integrated approach. © *2020 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.OE.59.5.051407]

Keywords: automatic target recognition; data fusion; deep learning; geometric hashing; inverse synthetic aperture radar.

Paper 20200300SS received Mar. 18, 2020; accepted for publication Apr. 14, 2020; published online Apr. 25, 2020.

1 Introduction

Computer vision with inverse synthetic aperture radar (ISAR) imagery finds rich applications such as remote sensing, surveillance, and deep space exploration.[1–4] ISAR images are formed with a high-resolution radar, and the images represent a two-dimensional distribution of target scatterers in the range (line of sight of radar) and cross-range (perpendicular to range) directions. The measurement of ISAR images is a complex task that requires uniform motion of the target with respect to radar. The current systems for ISAR-based target recognition, in general, employ model-based systems that are designed by experts.[5–12] These systems typically involve the careful crafting of meaningful features and their extraction to allow the algorithms to match existing models with the ISAR images. Such features include the distances and angles between the scattering centers of ISAR imagery. Although this approach provides explainable results to the ISAR algorithms, the system performance degrades significantly for low ISAR image quality. This degradation may occur due to various reasons such as low signal-to-inference-plus-noise-ratio,

*Address all correspondence to Rajkumar Theagarajan, E-mail: rthea001@ucr.edu

0091-3286/2020/$28.00 © 2020 SPIE

observation intervals, multipaths, and viewing geometry of the target. Additionally, the image collection platforms for ISAR can produce far more imagery (terabytes per day per aircraft), which image analysts find too difficult to handle.

The recent advances in deep learning and neural networks have demonstrated significant performance improvements with electro-optical imaging when applied to target recognition tasks. These data driven models extract spatial features and can be trained without the need for a subject matter expert. Although bringing such advantages, using deep learning tools, such as convolutional neural networks (CNNs), for ISAR image classification brings its own challenges. The deep learning approaches require extensive labeled data and their inferences are not as explainable as those of the model-based approach.

The scope of this paper is the development of several approaches for target recognition and their performance evaluation starting with ISAR imagery of target vessels and ending with the vessel classification. The specific challenges that we address in this paper are automated recognition of maritime vessels under different distortion conditions namely: occlusion, clutter, and adversarial attacks. Recognition methods involving template matching are not useful in these cases because distortions or occlusions change global features such as the object outline and major axis.[13–15] Constrained models of parts and joint articulation used in optical images[13,16] are not appropriate for the relatively low resolution and complex part interactions of ISAR images.

Our approach for target recognition is specifically designed for ISAR imagery. In contrast to the passive vision systems, the scale of the ISAR image is fixed by characteristics of the radar. However, where optical images are mainly formed as a result of diffuse reflections from a noncoherent source of light, ISAR images are formed primarily from the echo signal backscattered due to the induced electric and magnetic currents on the surface and volume of the target's body; they are nonliteral and vary quickly and abruptly with small pose angle variations.[17] The peaks (local maxima) in radar return are related to the physical geometry of an object. The relative locations of these scattering centers are independent of translation and serve as distinguishing features.

In this paper, we evaluate five different approaches for classifying maritime targets from ISAR images. (i) A model-based geometric hashing (MBGH) approach that uses the distance in range and cross range among scattering centers, (ii) a CNN-based approach that is trained to extract features in a supervised manner, (iii) a decision level fusion (DLF) that takes the average of the predictions of the CNN and model-based approaches as the final prediction, (iv) late feature fusion (LFF) that uses the features of the trained CNN and model-based approaches to train a support vector machine (SVM) classifier, and (v) early feature fusion (EFF) that uses the features extracted from the model-based approach as prior information along with the input image to train the CNN. We used the computer-aided design (CAD) models of two maritime targets and generated ISAR images using the radar cross-section simulations that incorporate dynamic effects of the ship motion and radar platform. Figure 1 shows rendered images of the two maritime targets used in this work.

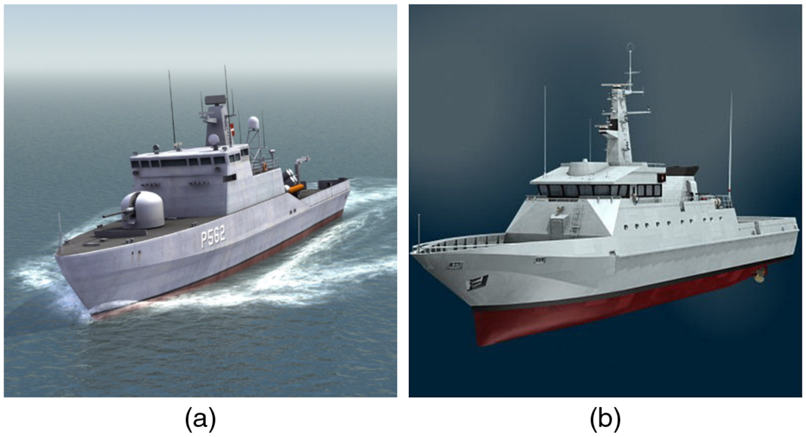

Fig. 1 Rendered images of the two maritime vessels used: (a) Flyvefisken class patrol boat and (b) offshore patrol vessel OPV54 Pluvier.

The reminder of this paper is organized as follows. Section 2 describes the related work and our contributions for automatic target recognition using ISAR imagery. The simulation setup and ISAR data are explained together with the technical approaches and an ablation study in Sec. 3. Experimental results under various distortions are presented and discussed in detail in Sec. 4 and Sec. 5 concludes our paper.

2 Related Works and Contributions of This Paper

After the Imagenet competition[18] entry of Krizhevsky et al.,[19] where the authors used CNNs for extracting features and classifying more than 1000 classes, there has been a wide spread adoption of CNNs and deep learning within the field of computer vision. Although there has been a wide spread adoption of CNNs, to date there are only few works[20–23] that use deep learning methods for the classification of ISAR images. The reason for this is that the process of generating and collecting ISAR images is very expensive and time-consuming. Moreover, the radars used for collecting ISAR imagery are predominantly used in the defense and military organizations making the data classified for public use. On the other hand, there have been many approaches[24–31] using deep learning for the classification of synthetic aperture radar (SAR) images. The reason for this is the availability of the MSTAR dataset,[32] which has 10 different ground targets whose images were collected using an X-band SAR sensor, at one-foot resolution spotlight mode, with full aspect coverage (0 deg to 360 deg). In the following, we discuss about the related work done for the two areas, namely: classification of targets using SAR imagery and classification of targets using ISAR imagery, followed by our contributions.

2.1 *Related Works for Classification of SAR Images*

Table 1 shows a summary of the related work for the classification of targets using SAR imagery. Geng et al.[24] used a deep convolutional autoencoder (DCAE) to extract features from SAR images and perform classification. The autoencoder consists of eight layers: a convolutional layer, a scale transformation layer to aggregate neighbor information, four layers based on sparse autoencoders to optimize features for classification, and the last two layers for postprocessing. As compared to the manually hand-crafted approaches,[33,34] the DCAE approach performs better classification on a dataset collected from the TerraSAR-X satellite.

Table 1 Summary of the related work for classification of SAR images.

Authors	Comments
Geng et al.[24]	Used a DCAE to extract features and perform classification using SAR images. This approach performs better as compared to handcrafted feature extraction methods on the TerraSAR-X dataset.
Zhou et al.[25]	Used polarimetric SAR images obtained using the AIRSAR data of San Francisco, California, to train and extract features from a four-layer CNN.
Huang et al.[26]	Used transfer learning by training a CNN on the MSTAR dataset and fine tuning on SAR images to classify different landscapes. The authors verify the learned features by reconstructing the input image using the features.
Zhao et al.[27]	Used DisDBN and an ensemble of classifiers to learn high-level features from SAR images.
Gong et al.[28]	Used a sparse autoencoder and unsupervised clustering to detect ternary changes in SAR images. The autoencoder transforms the log-ratio image into a feature space for weakly training the CNN.
Zhang et al.[29]	Used a CV-CNN for SAR image classification. The authors designed a complex valued back propagation algorithm.
Gao et al.[30]	Used a dual-CNN for classifying polarimetric SAR images. In this approach, one branch extracts polarization features and the other extract features from the Pauli RGB images.

Zhou et al.[25] used deep CNNs for the classification of polarimetric synthetic aperture radar (POLSAR) images. The POLSAR data are first converted into a normalized 6-D real feature vector. The six-channel images are then passed to a four-layer CNN with cascaded convolutional layers. The POLSAR images were obtained using the airborne synthetic aperture radar (AIRSAR) data of San Francisco, California. These images were used for classifying different types of land areas such as urban and vegetated areas present within a geographic location.

Huang et al.[26] designed a CNN architecture to classify different landscapes in SAR images. The authors had a very limited dataset of landscape images, hence they pretrained their CNN architecture on the MSTAR dataset and transferred the weights to fine tune the CNN on their dataset. Additionally, in order to make sure the CNN architecture learns the correct features and does not overfit, the authors reconstruct the input image using the extracted feature. The authors conclude that if the reconstructed images and the input images look similar then the features learned for classification should be correct.

Zhao et al.[27] used discriminant deep belief networks (DisDBN) to learn high-level features from SAR images to classify landscapes into water bodies, farmlands, buildings, and trees. The discriminant features are learned by combining ensemble learning with a deep belief network in an unsupervised manner. First, subsets of SAR image patches are marked with pseudolabels to train weak classifiers. Second, a specific SAR image patch is characterized by a set of projection vectors that are obtained by projecting the SAR image patch onto a weak decision space spanned by each weak classifier. Finally, the discriminant features are generated by feeding the projection vectors to a deep belief network for classification.

Gong et al.[28] used a CNN-based sparse autoencoder and unsupervised clustering to detect ternary changes in SAR images. First, the sparse autoencoder is used to transform the log-ratio difference image into a suitable feature space for extracting key features and suppressing outliers and noise. Next, the learned features are clustered into three classes, which are taken as pseudolabels for weakly training a CNN. In order to select reliable training samples for the CNN, the authors select training samples from the feature maps learned by the sparse autoencoder. After obtaining training samples and corresponding pseudolabels, the CNN model is trained using back propagation with stochastic gradient descent. To evaluate their approach, the authors used SAR images acquired from the ESAERS-2 satellite over the city San Francisco, CA and the Radarsat-2 satellite over the Yellow River Estuary in China.

Zhang et al.[29] proposed a complex-valued CNN (CV-CNN) for SAR image classification. The authors used both the amplitude and phase information of SAR images as input to the CV-CNN. Based on this, the authors derived a complex valued back propagation algorithm using stochastic gradient descent. The approach is evaluated on the polarimetric SAR image for classifying different terrain types into 15 classes collected from Flevoland, Netherlands, and Oberpfaffenhofen, Germany.

Gao et al.[30] used a dual branch-CNN (dual-CNN) for classifying polarimetric SAR images. In their approach, one branch of the CNN is used to extract polarization features from the six-channel real matrix derived from the complex coherency matrix. The other branch of the CNN is used to extract spatial features of a Pauli RGB image and both of these features are combined using a fully connected layer sharing the polarization and spatial property. The authors evaluated their approach on the AIRSAR data for classifying 14 different land terrains collected from Flevoland, Netherlands.

2.2 *Related Works for Classification of ISAR Images*

Table 2 shows a summary of the related works done for classification using ISAR images. He et al.[20] used the point scatter model of aircraft from ISAR images to train a CNN-based autoencoder. The authors used simulated images of aircraft to train the autoencoder to reconstruct the input image in an unsupervised manner. After training, the authors extract the latent space feature, which is given as input to a neural network for classification.

Hu et al.[21] used the deep- alternating direction method of multipliers (ADMM)-net (DAN) proposed by Ref. 40 to reconstruct high-quality ISAR images. DAN is constructed by mapping the iterative steps of the traditional convex optimization algorithm, ADMM,[41] into a deep network. The authors used three stages of the DAN and each stage corresponds to an ADMM

Table 2 Summary of the related work for classifying ISAR images.

Authors	Comments
He et al.[20]	Used an autoencoder trained on point scatter models of ISAR images to extract features for classification.
Hu et al.[21]	Used a three-stage DAN to recreate high-quality ISAR images from under-sampled ISAR images.
Xue et al.[22]	Localized and extracted semantic relationship between different objects in ISAR images by learning a graph-based attribute association.
Xue et al.[23]	Leveraged weakly supervised learning to localize objects in ISAR images using ARPN and a joint sparse learning.
Mamatha and Kumar[35]	Used the watershed transformation and wavelet decomposition to classify maritime vessels from ISAR images.
Lee et al.[36]	Proposed a preprocessing step that exploits the cross-range resolution of ISAR images to improve the classification performance.
Paladini and Martorella[37]	Used the Cloude–Pottier decomposition to extract orientation-invariant feature vectors to preserve the polarimetric structure.
Zelković et al.[38]	Used the pulse reflection shape and Doppler shifts to extract features defined by the size and shape for classifying aircraft.
Martorella et al.[39]	Used polarimetric ISAR images to match the 3-D scattering center of unknown targets with the 2-D scattering center of template images.
This paper	Compares a MBGH and deep learning approach for classifying maritime vessels from ISAR images. The authors fuse the two approaches at the classifier level and feature level and evaluate them under varying distortion conditions. Experimental results show that the two approaches are complementary to each other.

iteration. The constructed DAN is trained using the training set pairs of two-dimensional randomly undersampled data and high-quality image generated by the Range–Doppler method.

Xue and Tong[22] detected and localized objects in an ISAR image using deep relation graph learning. The authors used dilated deformable convolutions to improve the sampling and transformation ability of the CNN, which, in turn, increases the output resolution of the feature maps. Next, the authors proposed a graph-based attribute association learning to extract heterogeneous features and learn the semantic relation between different objects.

Xue and Tong[23] localized objects in ISAR images using advanced region proposal networks (ARPN) and a weakly supervised joint sparse learning. The ARPN uses a pairwise ranking loss and a triplet ranking loss for extracting features from the weakly annotated and unannotated ISAR images. The joint sparse learning helps extract more discriminative features while also learning shared and individual features and their correlation.

Mamatha and Kumar[35] used ISAR images to classify two different maritime vessels. The authors used the watershed transform and multiresolution wavelet transform to derive the target features. The wavelet analysis divides the image into approximate and details subsignals. The approximate and subsignals, which consist of the wavelet coefficients, statistical moments, and standard deviation of the ISAR image are taken as input to the classifier. The authors showed that wavelet coefficients obtained after first level of decomposition perform better on ISAR images without segmentation, whereas the wavelet coefficients obtained after third level of decomposition perform better on ISAR images with segmentation.

Lee et al.[36] proposed a preprocessing step that exploits the cross-range resolution (CRR) of ISAR images to improve their classification performance. The authors first obtained the CRR of an unknown target using cross-range scaling algorithms and normalize them such that they are identical to that of the ISAR images in the training dataset. By doing so, the deterioration in classification performance resulting from the CRR between the unknown target and the training images can be avoided.

Paladini et al.[37] classified nonhomogeneous targets from ISAR images by performing a macroscopic and detailed target analysis. First, the Cloude–Pottier decomposition is used as a starting point in order to find orientation-invariant feature vectors that are able to represent the average polarimetric structure of complex targets. After formulating the feature space, a nearest neighbor classification is done. Additionally, a validation process is performed by analyzing experimental data of simple targets collected in an anechoic chamber and airborne Electromagnetics Institute Surface Aperture Radar images of eight ships.

Zeljković et al.[38] classified aircraft from ISAR images using the pulse reflection shape and Doppler shifts of the parts that are in maneuver resulting in rotation of a target. The authors simulated five different aircraft and tested their images using features defined by size and shape in a prescribed holding pattern such as (i) number of pixels on the target, (ii) average values of the x and y coordinates of the pixels on the target, and (iii) size of the object along the x and y coordinates.

Martorella et al.[39] used polarimetric ISAR images (POLISAR) to match scattering points of an unknown target with the scattering centers of template images. The authors first obtained the three-dimensional (3-D) point like scattering center and compare it with the two-dimensional (2-D) scattering centers of the template images. Since the comparison is made between a set of scattering centers that belong to a 2-D domain and a set of scattering centers that belong to a 3-D domain, the image projection plane has to be identified (prior to or contextually) with the matching process. The authors addressed this problem by solving the alignment problem proposed by Cooke et al.[42]

2.3 *Contributions of this Paper*

- A robust decision level and feature level fusion approach that automatically classifies ISAR images of a target under various distortions using model-based and deep learning approaches.
- An ablation study is done for the model-based approach, deep learning approach, and the three different fusion approaches under different distortions namely: occlusions, clutter, masking, and adversarial attacks.
- From the experimental results, we empirically find that the model-based approach and deep learning approach complement each other.

3 Technical Approach

In this section, we first describe the ISAR imaging system and image data simulation followed by presenting the details of the MBGH approach, the deep learning approach, and the fusion of the approaches at the classifier and feature level.

3.1 *ISAR Imaging System and Data Simulation*

The ISAR imaging simulation is performed using the ANSYS Electronics Desktop software.[43] The ISAR imaging is formed by coherently processing echoes from a moving target and exploiting information from frequency shifts caused by motion of the target relative to the radar. Figure 2 shows an illustration for reformatting the received echo signals from the frequency-azimuth domain (f,θ) to the spatial frequency domain (f_x,f_y). The received echo signal $y(f,\theta)$ of a target at frequency f and azimuth angle θ can be expressed as follows:

$$y(f,\theta)=\sum_{k=1}^{L}a_k\,\exp\left[-j\frac{4\pi f}{c}(x_k\,\cos\,\theta+y_k\,\sin\,\theta)\right]+u(f,\theta), \tag{1}$$

where L is the number of scattering centers on the target, a_k is the amplitude of the k'th scattering center, and (x_k,y_k) is the position of the k'th scattering center in the spatial domain. u is the additive white Gaussian noise (AWGN) with zero mean and variance σ^2 and c is the speed of light. The received radar cross section uniformly sampled in the frequency-azimuth domain (f,θ) and (f_x,f_y) are related by $f_x=\frac{2f}{c}\,\cos\,\theta$ and $f_y=\frac{2f}{c}\,\sin\,\theta$. Data uniformity in the spatial

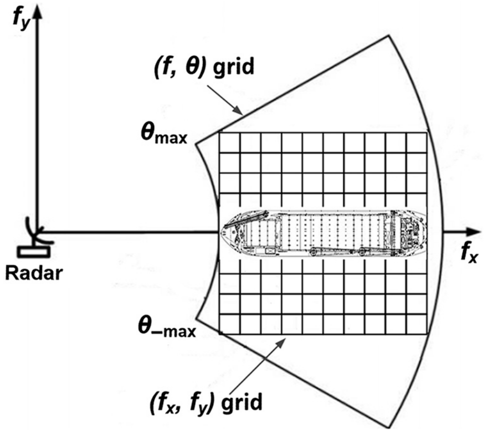

Fig. 2 Polar reformatting from (f, θ) domain to (f_x, f_y) domain.

frequency domain (f_x, f_y) is essential for Fourier transform. Therefore, in order to generate a focused ISAR image, the received data should be converted from polar-formatted samples to Cartesian-formatted samples with uniform spatial frequency sampling spacings Δf_x and Δf_y, before applying the Fourier transform. After polar reformatting, the discrete version of Eq. (1) is given by

$$y(m, n) = \sum_{k=1}^{L} a_k \exp\{-j2\pi[x_k f_x(m) + y_k f_y(n)]\} + u(m, n), \tag{2}$$

where

$$f_x(m) = f_x(0) + m\Delta f_x, \quad m = 1,2, \ldots, M$$
$$f_y(n) = f_y(0) + n\Delta f_y, \quad n = 1,2, \ldots, N$$

and M and N are the number of interpolated data samples in f_x and f_y, respectively. After polar reformatting, the spatial reflectivity function of the target can be easily reconstructed by means of a 2-D Fourier transform. The down-range and cross-range resolutions are inversely proportional to the spatial frequency bandwidths $M\Delta f_x$ and $N\Delta f_y$, respectively. In order to increase the spatial frequency bandwidths $M\Delta f_x$ and $N\Delta f_y$ for obtaining high-resolution ISAR images, the frequency bandwidth of the radar system and relative angular motion of the radar and target should be sufficiently large. Otherwise, the Fourier transform will result in limited resolution ISAR images, causing unreliable classification performance.

3.1.1 *Parameters of the ISAR Radar*

The ISAR radar is mounted on an aerial platform and the look angle is determined using the dynamics of ship motion. In our simulations, we assume the ship has minimal or no roll or yaw motions. For reasons related to restrictions on privacy and confidentiality, we are not allowed to disclose the parameters of the ISAR radar in this paper. Using the CAD models of the Flyvefisken class patrol boat and the offshore patrol vessel OPV54 Pluvier as shown in Fig. 1, we generate the ISAR images at azimuth angles (counterclockwise direction): 15 deg, 22 deg, 30 deg, 150 deg, and 165 deg. The 0-deg azimuth reference for the target is the direction of the ship's bow toward the radar as shown in the illustration of Fig. 2. Figures 3 and 4 show the simulated ISAR images for the two CAD models.

3.2 *Model-Based Geometric Hashing for Target Classification*

Figure 5 shows the overall framework for the MBGH approach proposed by Jones and Bhanu.[11] After obtaining the ISAR magnitude image of the ships, we first normalize the image and segment the body of the ship. The body of the ship is segmented by suppressing the peak-signal-to-noise

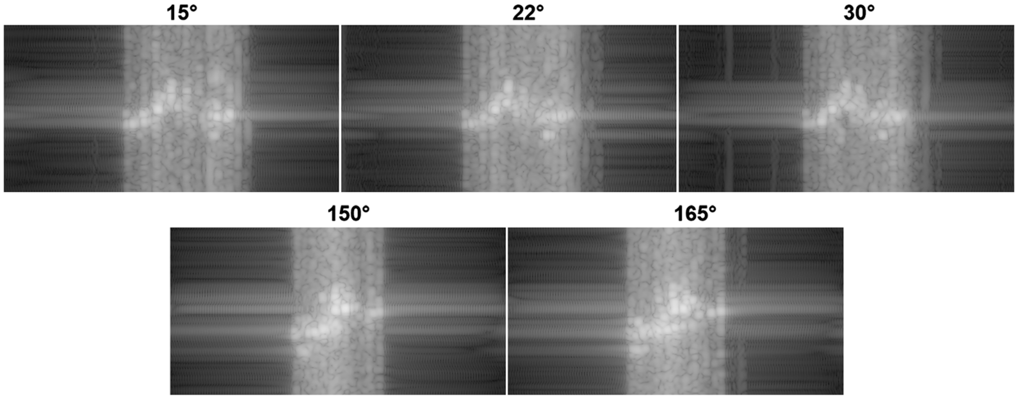

Fig. 3 ISAR images of the Flyvefisken class patrol boat at azimuth angles 15 deg, 22 deg, 30 deg, 150 deg, and 165 deg. The *x* axis of the corresponding images is the range and *y* axis is the cross range.

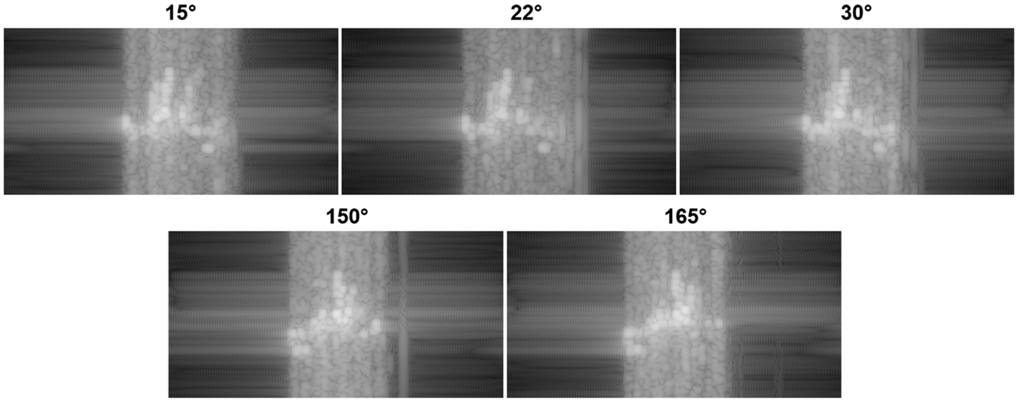

Fig. 4 ISAR images of the offshore patrol boat OPV54 Pluvier at azimuth angles 15 deg, 22 deg, 30 deg, 150 deg, and 165 deg. The *x* axis of the corresponding images is the range and *y* axis is the cross range.

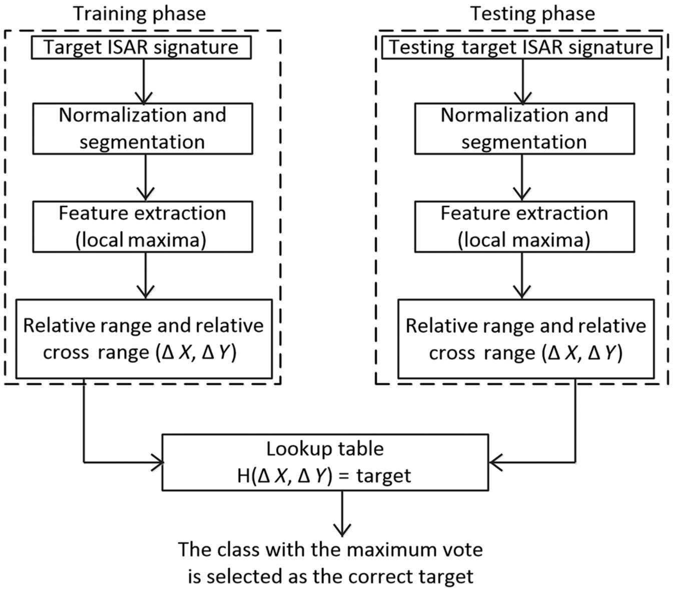
Training phase
Target ISAR signature
Normalization and segmentation
Feature extraction (local maxima)
Relative range and relative cross range (Δ X, Δ Y)
Testing phase
Testing target ISAR signature
Normalization and segmentation
Feature extraction (local maxima)
Relative range and relative cross range (Δ X, Δ Y)
Lookup table H(Δ X, Δ Y) = target
The class with the maximum vote is selected as the correct target

Fig. 5 Overall framework of the MBGH approach.

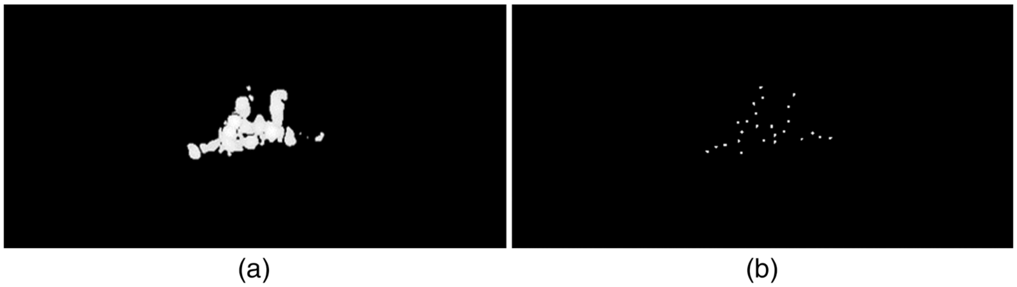

Fig. 6 (a) Segmented body and (b) scattering centers of the OPV54 Pluvier ship at azimuth angle = 30 deg. The *x* axis of the corresponding images is the range and *y* axis is the cross range.

ratio (PSNR) below a threshold. In our approach, we set the threshold to gray-scale pixel value of 210. We evaluated different threshold values within the range of (190 to 230) and empirically chose the threshold as 210 as it provided the best segmentation results without losing too much information on the target's body. In order to automatically segment the body of the target, it is possible to use approaches such as CFAR[44–46] or estimate the noise statistics[47,48] in the future. After segmenting the body of the ship, we compute the scattering centers of the ship. The scattering centers are obtained by computing the local maxima within a 3×3 sliding window. Unlike optical images where the images are formed as a result of diffuse reflections from a noncoherent source of light, ISAR images are formed from the echo signal backscattered due to the induced electric and magnetic currents on the surface and volume of the target's body. Hence, a high intensity at a particular point in the ISAR image means that there was a high reflection from that point on the target, resulting in a scattering center. Figures 6(a) and 6(b) show the segmented body of the OPV54 Pluvier ship at azimuth angle = 30 deg and its associated scattering centers, respectively.

After obtaining the M scattering centers, we treat each scattering center as a reference center and compute its relative range ΔX and relative cross range ΔY with respect to other scattering centers. Range refers to the distance along the x axis (horizontal axis) and cross range refers to the distance along the y axis (vertical axis). This results in $M(M-1)/2$ features per azimuth angle. We repeat this for all the azimuth angles of the ship and record all of the $(\Delta X, \Delta Y)$ pairs into a lookup table together with the target type and azimuth angle. Figures 7(a) and 7(b) show the visualization of the $(\Delta X, \Delta Y)$ pairs for the Flyvefisken and OPV54 Pluvier ships for the azimuth angles 15 deg, 22 deg, 30 deg, 150 deg, and 165 deg, respectively.

After obtaining the lookup table that consists of the $(\Delta X, \Delta Y)$ pairs for all of the azimuth angles for both models of the ships, during testing we extract the $M(M-1)/2$ pairs of $(\Delta X, \Delta Y)$ for the unknown target. For each $(\Delta X, \Delta Y)$ pair of the unknown target, we check the lookup table at those corresponding locations to see if there was a similar $(\Delta X, \Delta Y)$ pair for each of the ships. If so, the corresponding classes and azimuths get a vote. This is repeated for all the $(\Delta X, \Delta Y)$ pairs of the unknown target and at the end the class with the greatest number of votes is chosen as the class of the unknown target.

3.3 *Convolutional Neural Networks for Target Classification*

In this section, we explain in detail the architecture of the CNN and data augmentation for training the CNN. Table 3 shows the architecture of the CNN. The input to the CNN is the magnitude ISAR image of size 256×512. The CNN consists six convolution layers, each layer consists of k filters or kernels of size $m \times n \times q$, where m and n are the dimensions of the kernels and q is the number of filters. The size of the filters gives rise to the locally connected structures, which are convolved with the image to produce k feature maps of size $m - n + 1$. A rectified linear unit nonlinearity is applied to each convolution layer and every convolution layer is followed by batch normalization. After every convolutional layer, a fully connected layer is used to complete the CNN. The output of the fully connected layer has one neuron per class (two classes in our case), which corresponds to the Flyvefisken and OPV54 Pluvier ships. A softmax activation is applied to the fully connected layer, thus each neuron's output represents the posterior class probability.

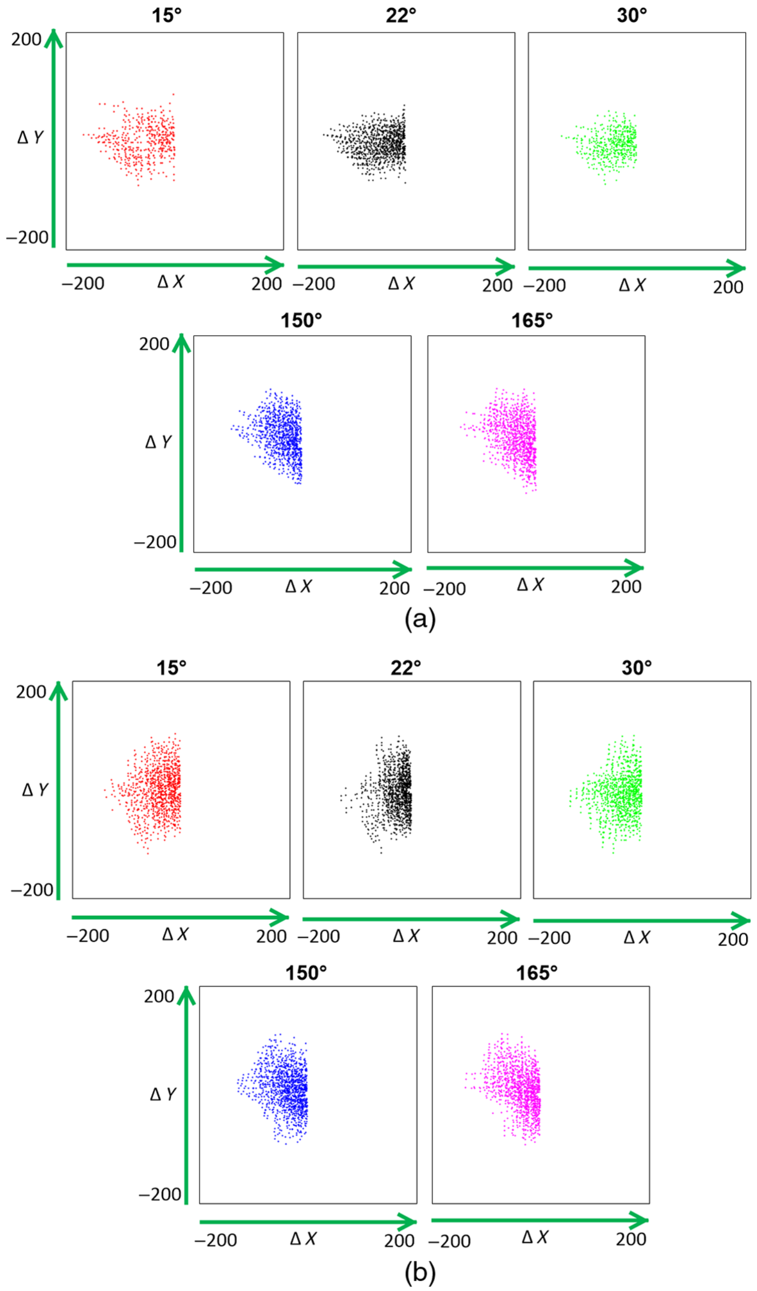

Fig. 7 Visualization of the $(\Delta X, \Delta Y)$ pairs of the (a) Flyvefisken class patrol boat and (b) offshore patrol boat OPV54 Pluvier at azimuth angles 15 deg, 22 deg, 30 deg, 150 deg, and 165 deg, respectively.

3.3.1 *Data augmentation for training the CNN*

In order to train the CNN, we augment our dataset by translating the locations of the scattering centers in the ISAR image. The scattering centers are obtained by computing the local maxima across a 3×3 window in the image as described in Sec. 3.2. After obtaining the M scattering centers, we take a $N \times N$ window centered around a scattering center and swap all the pixel values within the window with the pixel values from within another $N \times N$ window selected randomly on the body of the ship. Figure 8 shows an example of swapping the scattering centers for the Flyvefisken ship at an azimuth angle of 15 deg.

In Fig. 8, the blue dots indicate the scattering centers, the red bounding box is a 7×7 window centered around a scattering center, and the green bounding box is a 7×7 bounding box selected

Table 3 Architecture of the CNN.

Layer	Output dimension	Kernel dimension	Stride	Padding	Number of filters
Convolution 1	128 × 256	5 × 5	2	2	16
Convolution 2	64 × 128	5 × 5	2	2	32
Convolution 3	32 × 64	5 × 5	2	2	64
Convolution 4	16 × 32	5 × 5	2	2	96
Convolution 5	8 × 16	5 × 5	2	2	128
Convolution 6	4 × 8	5 × 5	2	2	128
Fully connected layer	2	—	—	—	—

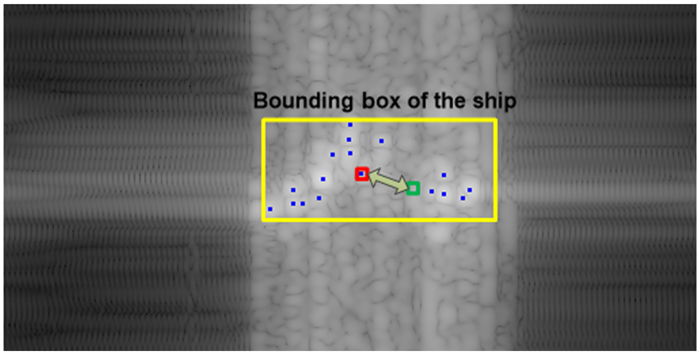

Fig. 8 Example of swapping a scattering center of the Flyvefisken ship at an azimuth angle of 15 deg. The *x* axis of the image is the range and *y* axis is the cross range.

randomly within the body of the ship. The pixel values within the two bounding boxes are swapped and the resulting image is augmented to the dataset. For each scattering center, we swap the window around it with 10 different randomly selected windows, which results in 10 different images. This is repeated for every scattering center and for all azimuth angles for each ship. In our experiments, we chose the size of the swapping windows to be 7 × 7. We evaluated windows of sizes 3 × 3, 5 × 5, 7 × 7, 9 × 9, 11 × 11, 13 × 13 and empirically found that 7 × 7 was the largest window possible that encompasses exactly one scattering center while also not overlapping background information when swapping windows close to the boundaries of the target. This resulted in a total of 1230 images for the Flyvefisken ship and 1150 images for the OPV54 Pluvier ship. Finally, after augmenting the dataset, we use these 2380 images for training the CNN. It should be noted that these 2380 images are not exactly the same as the original images shown in Figs. 3 and 4. Additionally, the original images shown in Figs. 3 and 4 were used only for testing our approaches.

3.4 *Fusion of the CNN and Model-Based Approaches*

In this section, we explain in detail the three different types of fusion performed namely: DLF, LFF, and EFF.

3.4.1 *Decision level fusion*

DLF is the simplest form of fusion between any two or more approaches where the output probabilities of each individual approach are averaged in order to get the final output probability. In our approach, after obtaining the output score for each class from the CNN, we convert the score of each class into a probability distribution over all the classes using the Softmax function given by

$$\mathrm{Pr}(X_i) = \frac{\mathrm{Exp}(X_i)}{\sum_{j=0}^{k} \mathrm{Exp}(X_j)}, \tag{3}$$

where $\mathrm{Pr}(X_i)$ is the probability of the output class X_i, $i = 1,2, \ldots, k$ and k is the number of classes. Similarly, the output of the model-based approach is converted into a probability distribution for each class using the following equation:

$$\mathrm{Pr}(X_i) = \frac{\text{total number of images classified as the class } X_i}{\text{total number of images}}. \tag{4}$$

Finally, we take the average of the probability distributions of the two individual approaches and the class with the highest probability is taken as the final classification.

3.4.2 *Late feature fusion*

In this approach, after training the CNN and the model-based approach, we extract the feature vectors from the individual approaches and concatenate them into a single feature vector. The concatenated feature vector is then used as input to train an SVM, which gives us the final prediction. The feature vector from the CNN is extracted by taking the output of the penultimate layer (convolution 6) as shown in Table 3. After extracting the feature vector from the CNN, we apply a sigmoid nonlinearity to the feature vector, such that the scale of the feature vector is between 0 and 1. Next, the feature vector is squashed into a single dimension resulting in a feature vector of size 4096×1 ($4 \times 8 \times 128 = 4096$).

The $(\Delta X, \Delta Y)$ feature vector from the model-based approach has a dimension of $M(M-1) \times 2$, where M is the number of scattering centers and it varies with respect to the azimuth angle between the target and the ISAR. From this, we take only the first 256 $(\Delta X, \Delta Y)$ pairs as the feature vector and normalize it with respect to the dimensions of the image. The resulting feature vector has a dimension of 256×2 and is within the scale of 0 to 1. Then, the 256×2 feature vector is then squashed into a single dimension (512×1) as shown in Fig. 9. Finally, the concatenated 4608×1 ($4096 + 512$) dimensional vector is then given as input to train an SVM with linear kernel for classification. Additionally, in cases where there are not many scattering centers visible on the target ($M < 17$), such that the dimension of the extracted feature vector from the model-based approach is less than 256×2, we concatenate zeros to the end of the feature vector in order to fit its size.

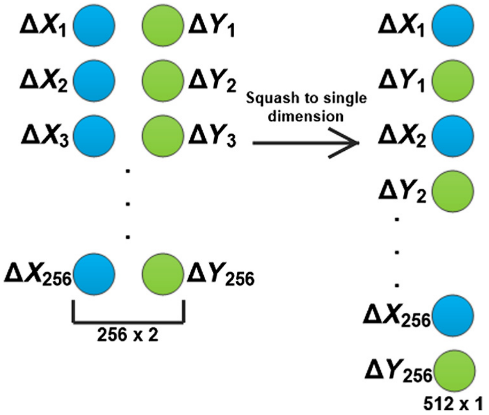

Fig. 9 Squashing the 256×2 $(\Delta X, \Delta Y)$ feature vector from the model-based approach into a 512×1 feature vector.

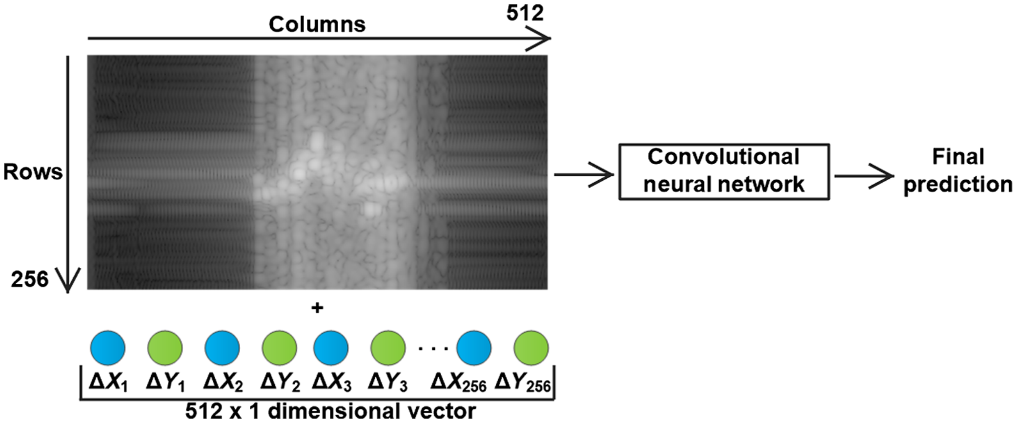
Columns
512
Rows
256
Convolutional neural network
Final prediction
+
ΔX_1 ΔY_1 ΔX_2 ΔY_2 ΔX_3 ΔY_3 ... ΔX_{256} ΔY_{256}
512 x 1 dimensional vector

Fig. 10 Concatenating the 512 × 1 feature vector extracted from the model-based approach to the bottom row of the input image, which is then passed to the CNN for training.

3.4.3 *Early feature fusion*

In this approach, we first extract the 512 × 1 feature vector from the model-based approach as discussed in Sec. 3.4.2. This feature vector is then concatenated to the bottom row of the input image as shown in Fig. 10. The resulting image is then passed as input to train the CNN. By doing so, the CNN is able to utilize the relative distances between the scattering centers as prior information to further improve its classification when compared to using the ISAR image alone. Experimental results in Sec. 4 show that by providing this prior information to the CNN, we are able to improve the classification accuracy.

3.5 *Ablation Study for the Performance of the Classifiers Under Different Articulations on the ISAR Image*

In this section, we explain in detail the process for generating different distortions on ISAR images. We experimented with three different distortions namely: occlusion and clutter, masking, and adversarial attacks. The reason for introducing distortions to our ISAR images is to simulate a real-world noisy environment where these distortions could affect the performance of our classifier.

3.5.1 *Occlusion and clutter*

Occlusions are artifacts in the ISAR image that hide the scattering centers, whereas clutters are artifacts that may create new scattering centers in the ISAR image that should not exist. Occlusions and clutters can occur in the real world when there is an unmodeled motion causing the target image to defocus resulting in scattering centers being in wrong positions. Although this error can be corrected using suitable mechanical design or by the use of autofocus techniques, it is a good ablation study to see how the individual classifiers and their fusion perform under such scenarios.

In our approach in order to introduce occlusion, we randomly select $X\%$ of scattering centers and mask them by replacing the pixel values of the selected scattering centers with the background value. Clutters are generated by randomly adding $X\%$ of new scattering centers. This is done by randomly selecting a pixel within the body of the ship and replacing the pixel value to 255. In our experiments, we keep the ratio of occlusion to clutters to 1, meaning that for $X\%$ of occlusion in an image we generate $X\%$ of clutter. Figure 11 shows an example for 30% generated occlusion and clutter with window sizes of 5 × 5 for the Flyvefisken ship at an azimuth angle of 15 deg, the red bounding boxes indicate clutter, the yellow bounding boxes indicate occlusion, and the blue dots are the remaining original scattering centers. In Fig. 11, the ISAR image initially had 17 scattering centers, 30% of occlusion means that we randomly selected 5 scattering centers (30% of $17 \approx 5$) and replaced the pixel values to the background pixel value. Next, in order to generate the clutter, we randomly selected 5 pixel locations on the body of the ship and replaced the value of each pixel to 255.

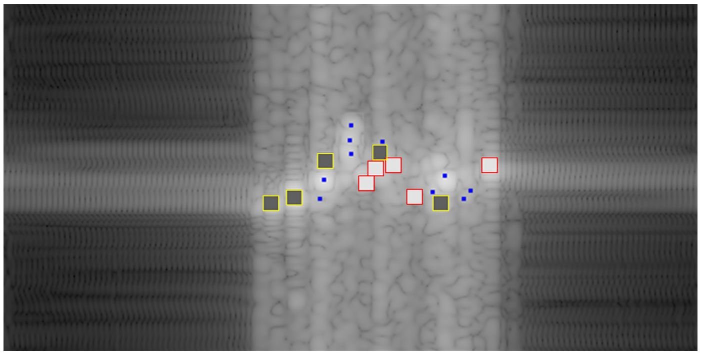

Fig. 11 Example of 30% generated occlusion and clutter with window sizes of 5 × 5 for the Flyvefisken ship at an azimuth angle of 15 deg. The red bounding boxes indicate clutters, the yellow bounding boxes indicate occlusion, and the blue dots are the remaining original scattering centers. (For illustration purposes, the occlusion and clutter are enlarged). The *x* axis of the image is the range and *y* axis is the cross range.

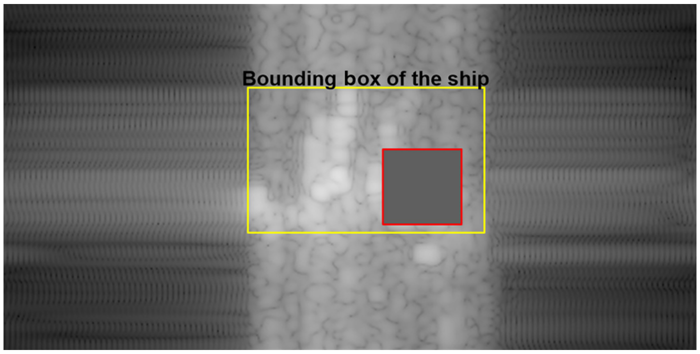

Fig. 12 Example of masking 30% of the body of the OPV54 Pluvier ship at azimuth angle of 15 deg. The red bounding box of dimension 55 × 55 is the mask that blocks ∼30% of the area of the ship. The *x* axis of the image is the range and *y* axis is the cross range.

3.5.2 *Masking the body of the target*

In the real-world scenarios, there are situations where the aircraft is trying maneuver its position in order to get the target into the field of view of ISAR radar. During such situations, only a partial area of the body of the target is visible. Hence, we perform an ablation study to see how the individual classifiers and their fusion perform under such situations. In order to mask the body of the target, we select the smallest possible square window on the body of the ship that can mask $X\%$ of the area of target. Next, we replace all the values within this window to the background pixel value. Figure 12 shows an example where 30% of the body of the OPV54 Pluvier ship was masked. In Fig. 12, the red bounding box is the mask with dimensions 55 × 55 that blocks ∼30% of the area of the ship. The area of the ship is computed as the number of pixels on the body after segmenting the ship.

3.5.3 *Adversarial attack on the CNN*

In Secs. 3.5.1 and 3.5.2, we discussed in detail about distortions related to occlusion, clutter, and masking. All of these approaches involve manipulating the location of the scattering centers and the superstructure of the target. In the real-world scenarios, distortions can also occur due to the presence of noises from adversarial attacks on the ISAR images. Adversarial examples are inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake. Hence, we perform an ablation study to see how adding adversarial noise to the ISAR images affects the performance of the individual classifiers and their fusion. Remote monitoring of maritime targets is a very critical and sensitive military application that requires

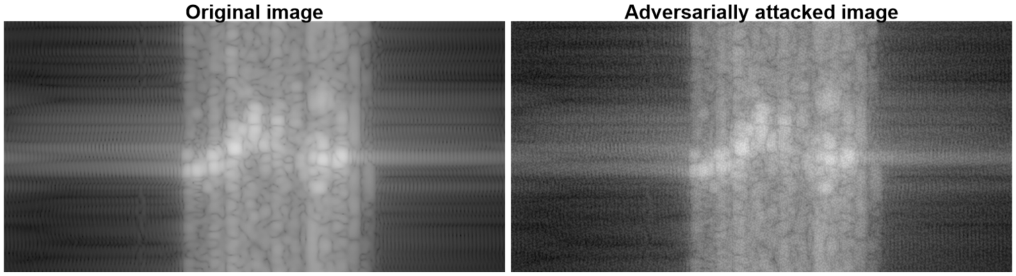

Fig. 13 Example of an adversarial attack with $\epsilon_{\text{attack}} = 8$ on an ISAR image of the Flyvefisken ship at azimuth angle of 15 deg. The original image on the left was predicted correctly as the Flyvefisken ship, whereas the adversarial image on the right was misclassified as the OPV54 Pluvier ship even though the two images look visually similar to a human observer. The x axis of the corresponding images is the range and y axis is the cross range.

the system to be robust against adversarial attacks. Adversarial noise can be added into the system when the aircraft is transmitting the ISAR images to the ground station.[49–52] In our approach, we induce the adversarial noise in the ISAR image by attacking the CNN using the fast gradient sign method (FGSM) proposed by Goodfellow et al.[53]

For any given image $\mathbf{X}$, adversarial attacks try to find a small perturbation (adversarial noise) δ with $\|\delta\|_\infty \le \epsilon_{\text{attack}}$ such that a classifier gives a misclassification for $\mathbf{X}^{\text{adv}} = \mathbf{X} + \delta$. ϵ_{attack} is a parameter that sets the perturbation limit for each pixel in the image $\mathbf{X}$. In the FGSM attack, the perturbation δ is generated based on the equation given by

$$\delta = \epsilon_{\text{attack}} \text{sign}[\nabla_{\mathbf{X}} L(\mathbf{X}, y)], \tag{5}$$

where $\nabla_{\mathbf{X}} L(\mathbf{X}, y)$ is the gradient of the loss function used to train the model and y is the class label. This approach uses the sign of the gradients computed for every pixel to determine the direction and magnitude of perturbation with which to change the corresponding pixel value in order to result in a misclassification. After generating the adversarial noise δ, we create the adversarial image $X_{\text{adv}} = X + \delta$. Figure 13 shows an example of an original image compared to its corresponding adversarial image that was misclassified by the CNN with $\epsilon_{\text{attack}} = 8$. From Fig. 13, we can observe that the adversarial image visually looks grainier compared to the original image resembling an AWGN. In this ablation study, we observe the performance of the individual classifiers and their fusion under adversarial attacks with strength $\epsilon_{\text{attack}} = 2, 4, 8, 10, 12$. We did not use $\epsilon_{\text{attack}} > 12$ because the adversarial noise becomes too strong making the resulting ISAR image indistinguishable even to the human observer.

4 Experimental Results

In this section, we evaluate the individual classifiers and their fusion in the presence of the different distortions discussed in this paper namely: occlusion, clutter, masking, and adversarial noise. In order to train the MBGH approach, we used the original five images of each class shown in Figs. 3 and 4 to compute the lookup table as shown in Fig. 5. In order to train the CNN, we used the 2380 augmented images described in Sec. 3.3.1.

4.1 *Performance of the Classifiers in the Presence of Occlusion and Clutter*

In this section, we evaluate the individual approaches in the presence of varying percentages of occlusion and clutter. Table 4 shows the performance accuracy and comparison of our individual classifiers and their different fusions. For each occlusion percentage and window, we generated 200 images for every azimuth angle using the original images shown in Figs. 3 and 4. Since we have ISAR images for the two classes generated at five different azimuth angles, this results in 1000 images per window per occlusion/clutter percentage for each class. We used these occlusion/clutter images for evaluating the performance of the classifiers in Table 4.

Table 4 Performance accuracy (%) and comparison of the individual classifiers and their different fusions in the presence of varying percentages of occlusion and clutter.

Dimensions of the window	Approach[a]	Percentage of occlusion and clutter						
		20%	30%	40%	50%	60%	70%	80%
3 × 3	MBGH	93.51	81.46	74.82	68.33	65.21	63.27	60.56
	CNN	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	LFF	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	EFF	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	DLF	100.00	100.00	100.00	100.00	100.00	100.00	100.00
5 × 5	MBGH	91.64	79.60	70.93	66.76	63.20	62.37	60.91
	CNN	100.00	100.00	100.00	99.84	99.84	99.26	99.02
	LFF	100.00	100.00	100.00	100.00	100.00	100.00	99.76
	EFF	100.00	100.00	100.00	100.00	100.00	99.56	99.14
	DLF	100.00	100.00	100.00	100.00	100.00	99.92	99.76
9 × 9	MBGH	89.25	77.72	72.27	63.01	60.18	55.27	52.44
	CNN	100.00	100.00	100.00	100.00	96.24	93.28	91.10
	LFF	100.00	100.00	100.00	99.36	97.38	93.46	90.95
	EFF	100.00	100.00	100.00	98.78	95.20	94.22	91.79
	DLF	100.00	100.00	100.00	100.00	96.24	94.32	92.14

[a] MBGH, model-based geometric hashing; LFF, late feature fusion; EFF, early feature fusion; and DLF, decision level fusion.

We experimented with occlusion and clutter percentages from 20% to 80% and window sizes of 3×3, 5×5, and 9×9. From Table 4, we can observe that, as the percentage of occlusion/clutter and the window dimension increase, the performance of the model-based approach drastically declines. In the worst case scenario of 80% occlusion and clutter with 9×9 dimensional windows, the performance of the model-based approach is only 52.44%, which is almost close to random prediction. The reason for this is that as we occlude and add new scattering centers the values of the $(\Delta X, \Delta Y)$ pairs of the distorted images start to vary significantly compared to the values in the lookup table. Hence, this finding shows that the model-based approach is very sensitive to the locations of the scattering centers, which is also shown to be consistent with the work done by Jones and Bhanu.[11]

On the contrary, the performance of the CNN did not significantly drop with increasing percentages of occlusion and clutter as compared to the model-based approach. In the worst case scenario of 80% occlusion and clutter with 9×9 dimensional windows, the CNN achieved 91.10% accuracy. Additionally, with DLF, this accuracy increases to 92.14%. The key reason for this is that when we create occlusion and clutter, they are mostly concentrated within the body of target leaving the superstructure of the target largely untouched. Hence, the CNN is able to detect the superstructure of the target, which leads to a high classification accuracy compared to the model-based approach.

4.2 *Performance of the Classifiers in the Presence of Masking*

From the previous Sec. 4.1, we can observe that the CNN is able to detect the superstructure of the ship resulting in a higher performance accuracy compared to the model-based approach in the presence of occlusion and clutter. In this section, we evaluate the performance of the individual

Table 5 Performance accuracy (%) and comparison of the individual classifiers and their different fusions in the presence of varying percentages of masking.

	Masking percentage				
Approach	20%	30%	40%	50%	60%
MBGH	77.83	74.29	72.38	68.16	64.64
CNN	61.99	58.29	52.94	51.13	50.28
LFF	70.24	68.35	66.57	66.12	64.88
EFF	72.58	70.02	70.16	68.29	66.16
DLF	76.04	74.82	72.35	70.98	70.14

classifiers and their fusions in the presence of masking. Table 5 shows the performance accuracy and comparison of our individual classifiers and their fusion in the presence of masking. For each percentage of masking we generated 40 images for every azimuth angle using the original images shown in Figs. 3 and 4. Since we have ISAR images for the two classes generated at five different azimuth angles, this results in 200 images per masking percentage for each class. We used these masked images for evaluating the performance of the classifiers in Table 5.

From Table 5 we can observe that as the percentage of masking increases, the performance of the CNN declines rapidly. The CNN achieved 61.99% for 20% masking and only 50.28% for 60% masking. Comparing this with the model-based approach, the model-based approach was able to achieve a higher performance accuracy for all percentages of masking compared to the CNN. The reason for this is that even though a percentage of the body of target is masked, the scattering centers in the unmasked portions of the target are not affected. Hence, the model-based approach is able to use the information of the unaffected scattering centers to predict the targets. When we performed EFF and LFF, the performance accuracy was higher compared to just using the CNN as a standalone classifier, but the performance accuracy was still lower compared to using the model-based classifier as a standalone classifier. This indicates that the CNN was not able to extract reliable features and was pulling down the performance of the feature level fusion as a whole. On performing DLF, we were able to achieve slightly higher accuracy of 70.14% at 60% masking compared to all other classifiers.

4.3 *Performance of Classifiers in the Presence of Adversarial Attack*

In Secs. 4.1 and 4.2, we evaluated our classifiers by manipulating the positions of the scattering centers and the superstructure of the target. In this section, we evaluate the performance of the classifier in the presence of adversarial noise. Table 6 shows the performance of the different classifiers in the presence of the FGSM adversarial attack. For each ϵ_{attack}, we generated 40

Table 6 Performance accuracy (%) and comparison of the individual classifiers and their different fusions in the presence of varying strength of adversarial attack.

	Adversarial attack strength (ϵ_{attack})				
Approach	2	4	8	10	12
MBGH	93.31	90.12	86.58	85.11	83.05
CNN	77.22	64.98	62.17	59.55	53.48
LFF	80.66	77.42	79.15	74.10	72.73
EFF	77.35	70.62	68.43	70.11	66.80
DLF	89.87	89.16	87.20	85.37	84.38

adversarial images for every azimuth angle using the original images shown in Figs. 3 and 4. Since we have ISAR images for the two classes at five different azimuth angles, this results in 200 images per ϵ_{attack} for each class.

From Table 6, we can observe that as the strength of the adversarial attack increases, the performance of the CNN drastically declines and achieves only 53.48% when $\epsilon_{\text{attack}} = 12$. The reason for this is that adversarial attacks are intentionally designed to cause perturbation in pixel values of an image such that these perturbations cause a covariate shift in the feature space of the CNN leading to a misclassification.

On the contrary, the model-based approach had the highest accuracy of 93.31% and 90.12% when $\epsilon_{\text{attack}} = 2$ and 4, respectively. Additionally, the performance of the model-based approach did not decline rapidly with increasing ϵ_{attack} as compared to the CNN. The reason for this is that, although the adversarial attack perturbs the pixel values in the image, these perturbations are not strong enough to affect the difference in pixel values between the scattering centers and their surrounding neighborhood pixels. But, as ϵ_{attack} increases beyond 12, the perturbations become so severe as to be corrected. When fusing the two approaches, we can observe that the LFF slightly outperforms the EFF. Moreover, as ϵ_{attack} increases beyond 4, the DLF is able to outperform all the other classifiers.

4.4 *Discussion of Results*

In this section, we discuss the performance of the individual classifiers and their fusions in the presence of the various distortions as discussed above.

4.4.1 *Performance of the model-based geometric hashing approach*

From Table 4, we can see that, as the percentage of scattering centers that are cluttered/occluded increases, the performance of the model-based approach declines significantly compared to the other approaches. This is evident from the fact that the model-based approach uses the relative distances $(\Delta X, \Delta Y)$ between scattering centers in order to predict the class. By manipulating the locations of these scattering centers using occlusion/clutter, the values of the $(\Delta X, \Delta Y)$ pairs start to vary significantly compared to the $(\Delta X, \Delta Y)$ pairs in the lookup table.

Furthermore, in Table 5, as the size of the mask increases, the performance of the model-based approach steadily declines but not as much as compared to the CNN. The reason for this is that although part of the body of the target is occluded, the locations of the scattering centers that are visible are not altered, hence the model-based approach is able to utilize the $(\Delta X, \Delta Y)$ pairs from the visible scattering centers to outperform the CNN. This is even more evident from Table 6 where in the presence of adversarial noise, the model-based approach still outperformed the CNN because the locations of the scattering centers were not altered. This shows that the model-based approach is sensitive to the locations of the scattering centers.

4.4.2 *Performance of the CNN*

From Table 4, we can see that as the percentage of scattering centers that are cluttered/occluded increases, the performance of the CNN is unaffected as compared to the model-based approach. The reason for this is that compared to the model-based approach, the CNN is not dependent on the locations of the scattering centers, but rather it learns to extract features that are closely related to the superstructure of a target. This is evident from Table 5, whereas the percentage of masking of the body of a target increases, the performance of the CNN drastically declines. The CNN achieved only 50.28% when 60% of the body of a target is masked.

Additionally, in Table 6, as the strength of the adversarial attack (ϵ) increases, the performance of the CNN rapidly declines. The reason for this is that FGSM adversarial attacks add an adversarial noise to the image such that the noise tries to push the direction of the gradient of the CNN in a different direction compared to the gradient of the original image, resulting in a misclassification. Since the gradient is derived based on the features learned by the CNN, by perturbing the gradient, the CNN extracts erroneous features that may not be related to the superstructure of the ship resulting in a misclassification. Based on these results, we can conclude that the CNN and model-based approaches are able to extract features that are almost complementary to each other.

4.4.3 *Performance of the feature level and decision level fusion*

From Table 4, we can see that by combining the features extracted by the model-based approach and CNN, the LFF and EFF approaches are able to outperform the model-based approach and CNN. Similarly, in Table 5, as the percentage of the masking increases, the performance of the CNN decreases, but when we fuse the features extracted by model-based approach along with the CNN, the LFF and EFF approaches are able to improve the classification performance compared to the individual classifiers. Furthermore, in Tables 4 and 5, the DLF is able to slightly outperform the feature level fusion resulting in higher classification accuracy.

In Table 6, the LFF and EFF approaches outperform the CNN, but fall short when compared to the model-based approach. Additionally, the LFF outperformed the EFF approach in the presence of the adversarial attacks. When performing DLF, since the locations of the scattering centers are not altered, the model-based approach has a higher probability of correct classification compared to the CNN. Hence, the final classification obtained from the DLF is able to overpower the individual classification of the CNN resulting in improved performance compared to the CNN and the model-based approach.

4.5 *Additional Experiments*

In this section, we describe additional experiments by adding two more targets into our dataset and experiment with different data representations: in-phase and quadrature phase (I/Q), magnitude only, fast Fourier transform (FFT) of complex radar input, and magnitude and phase. Since ISAR images are very high-dimensional data, we evaluate the performance of dimensionality reduction using the *t*-distributed stochastic neighbor embedding (t-SNE)[54] algorithm, and then use the resulting feature vector to train an SVM model. For reasons related to restrictions on privacy and confidentiality, we shall refer to the additional two targets as target 1 and target 2, respectively.

All the experimental results reported in Secs. 4.1 to 4.3 used only the magnitude images of the targets for classification under different distortion conditions. In this section, we conduct experiments to empirically observe the classification performance when using different input data representations as mentioned above without any distortion. For this purpose, we generated ISAR images for the two additional targets using the same setup as described in Sec. 3.1.1 with azimuth angles (counter clockwise direction) of 15 deg, 22 deg, 30 deg, 150 deg, and 165 deg.

In order to train the CNN, we performed data augmentation on the ISAR images of the four targets. The data augmentation was done by adding random noise to both the magnitude and phase signals. The noise added to each pixel in the magnitude image was randomly selected between the range of ±10% of the original pixel value. Similarly, the noise added to each pixel in the phase image was randomly selected between ±5 deg of the original pixel value. We selected these ranges empirically such that the resulting image does not significantly deviate when compared with the original image. This multiplicative noise is similar to simulating speckles in SAR images.[55] Speckle is caused by coherent processing of reflected signals from multiple scattering centers of the target. By simulating such kinds of noise, we are making the image to be more realistic as it would be in a real-world noisy environment. This is also similar to the color jittering function[19,56] that is used for augmenting datasets while training CNNs. This adds more variability to the images compared to using AWGN and also helps to improve the robustness in classification by our CNN.

Based on this, we generated 365 images for target 1, 310 images for target 2, 25 images for OPV54 Pluvier ship, and 45 images for Flyvefisken patrol boat. In order to train the CNN, we randomly selected 80% of the data from each class for training and 20% for testing. It should be noted that in this section, we did not use the 2380 images of the OPV54 Pluvier and Flyvefiken ships that were described in Sec. 3.3.1 for training the CNN. The reason for this is that we wanted to observe the classification performance when we augment the dataset by adding random noise to both the magnitude and phase signals of the ISAR image.

4.5.1 *Classification using CNN and different data types*

We trained a MobileNet-v2 architecture using the different data representations mentioned above. MobileNet-v2 is a CNN architecture proposed by Sandler et al.[57] that is designed to have small number of parameters (~2.2 million parameters) compared to a deeper CNN model such as ResNet[56] that has ~23.6 million parameters. Despite having a very small number of parameters compared to ResNet, MobileNet-v2 has been shown to achieve comparable performance to ResNet using depthwise separable convolutions.

When using multiple data representations as input, we concatenate the selected data representations along with the input channel. For example, when using I/Q data, the size of the resulting ISAR image becomes 256 × 512 × 2. This is similar to using three channels for RGB color images or one channel for grayscale images. Table 7 shows the classification performance for the four targets using the different data types and Tables 8–11 show the confusion matrices for the different data representations used in Table 7.

From Table 7, it can be seen that using the I/Q data resulted in the highest classification accuracy (99.3%) followed using the magnitude data (98.7%). We observed that when using the magnitude and phase data together, the training of the CNN was less stable compared to using the other data representations and the accuracy on the testing dataset drops down to 79.3%. Based on this, our results suggest that concatenating the phase and magnitude images with the input channels may cause the CNN to overfit on the training data resulting in drop in the performance.

Table 7 Classification accuracy on the testing dataset for the four targets using different data representations.

Input type	Accuracy (%)
I/Q	99.3
Magnitude	98.7
Magnitude/phase	79.7
FFT of complex radar input	85.2

Table 8 Confusion matrix for classification using I/Q data of the four targets.

Prediction →	Target 1	Target 2	Flyvefisken	OPV54
Target 1	72	0	0	0
Target 2	1	61	0	0
Flyvefisken	0	0	5	0
OPV54	0	0	0	9

Table 9 Confusion matrix for classification using magnitude data of the four targets.

Prediction →	Target 1	Target 2	Flyvefisken	OPV54
Target 1	70	1	1	0
Target 2	0	62	0	0
Flyvefisken	0	0	5	0
OPV54	0	0	0	9

Table 10 Confusion matrix for classification using magnitude and phase data of the four targets.

Prediction →	Target 1	Target 2	Flyvefisken	OPV54
Target 1	57	15	0	0
Target 2	15	47	0	0
Flyvefisken	0	0	5	0
OPV54	0	0	0	9

Table 11 Confusion matrix for classification using FFT data of the four targets.

Prediction →	Target 1	Target 2	Flyvefisken	OPV54
Target 1	50	13	9	0
Target 2	0	62	0	0
Flyvefisken	0	0	5	0
OPV54	0	0	0	9

4.5.2 *Dimensionality reduction using t-SNE and SVM-based classification*

Since ISAR imagery is a multidimensional data, in this section, we perform dimensionality reduction using t-SNE and then perform an SVM-based classification. t-SNE is a nonlinear dimensionality reduction technique that is used for visualizing high-dimensional data. The t-SNE algorithm consists of two stages.

1) In this first stage, t-SNE calculates the prior distribution over high-dimensional data. Consider two images x_i and x_j. t-SNE first calculates the joint probability p_{ij} using the two conditional probabilities $p_{j|i}$ and $p_{i|j}$. The conditional probability $p_{j|i}$ is defined as

$$\frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|x_i - x_k\|^2/2\sigma_i^2)}, \tag{6}$$

where σ^2 denotes the bandwidth of the Gaussian kernel. Next, the joint probability p_{ij} is simply calculated as $p_{ij} = (p_{i|j} + p_{j|i})/2N$, where the probabilities when $i = j$ is set to zero such that $p_{ij} = 0$ and N is the number of samples.

2) In the second stage, t-SNE defines a prioir distribution over the points in the low-dimensional representation to minimize the Kullback–Leibler (KL) divergence between the two distributions. For a low-dimensional representation of d simensions, the t-SNE transformations of N samples are represented as $y_i \in \mathbb{R}^d$ for $i = 1, \dots, N$. These points use the following similarity function q_{ij} that is defined as

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k\neq i}(1 + \|y_i - y_k\|^2)^{-1}}. \tag{7}$$

Using these similarity measures, the t-SNE method employs the KL divergence of the reduced dimension distribution Q from the data distribution P and solves $\mathrm{KL}(P||Q) = \sum_{i\neq j} p_{ij}\,\log(p_{ij}/q_{ij})$. When the clusters in the reduced dimensions reflect separation of target types, they can be used to infer an unknown target's class. Additionally, unknown targets can be inferred as unknown based on their projected distance to the existing clusters.

Next, we use the low-dimensional embedding as input to train an SVM classifier. In the t-SNE algorithm, there are two parameters to tune, namely: (a) the initialization algorithm and

(b) the perplexity. The initialization algorithm determines the size, distance, and shape of the clusters belonging to the low-dimensional embeddings. The perplexity is the parameter that balances the local and global aspects of embeddings.

In our approach, we consider two initialization algorithms: (i) random initialization and (ii) principal component analysis (PCA). We experimented with different values of perplexity between 5 and 50 and found that a perplexity value of 25 with random initialization provided the best separation between the classes. Figure 14 shows the t-SNE plot for the four targets using their magnitude image with random and PCA initialization and perplexity values of 25 and 40. In Fig. 14, t-SNE projects a manifold from a high-dimensional space to a lower dimension and the axes have no logical meaning. This is similar to PCA. The axes are not meant to be interpretable, they just define a 2-D space into which higher dimensional space is projected, while preserving relative proportional distances as much as possible.

From Fig. 14, we can observe that random initialization provides the best separation, whereas PCA-based initialization fails to separate the clusters belonging to target 1 and target 2. Based on the observation from Fig. 14, we use random initialization with perplexity = 25 in order to obtain the low-dimensional embeddings of the magnitude images, which is then used for training an SVM classifier. We split the dataset randomly with 80% of each class as training data and 20% for testing. After obtaining the low-dimensional features for both the training and testing datasets, we train an SVM classifier with a radial basis function kernel given by

$$f(x) = \exp(-\gamma\|x - x'\|^2), \tag{8}$$

where x is the input feature vector. There are two parameters γ and C that need to be tuned while training SVM. The parameter γ determines the output of SVM as shown in the kernel function in Eq. (8). γ redefines the Euclidean distance in feature space into the distance in kernel space, thus there are few neighbors in the training set for smaller γ compared to a larger γ. The parameters C

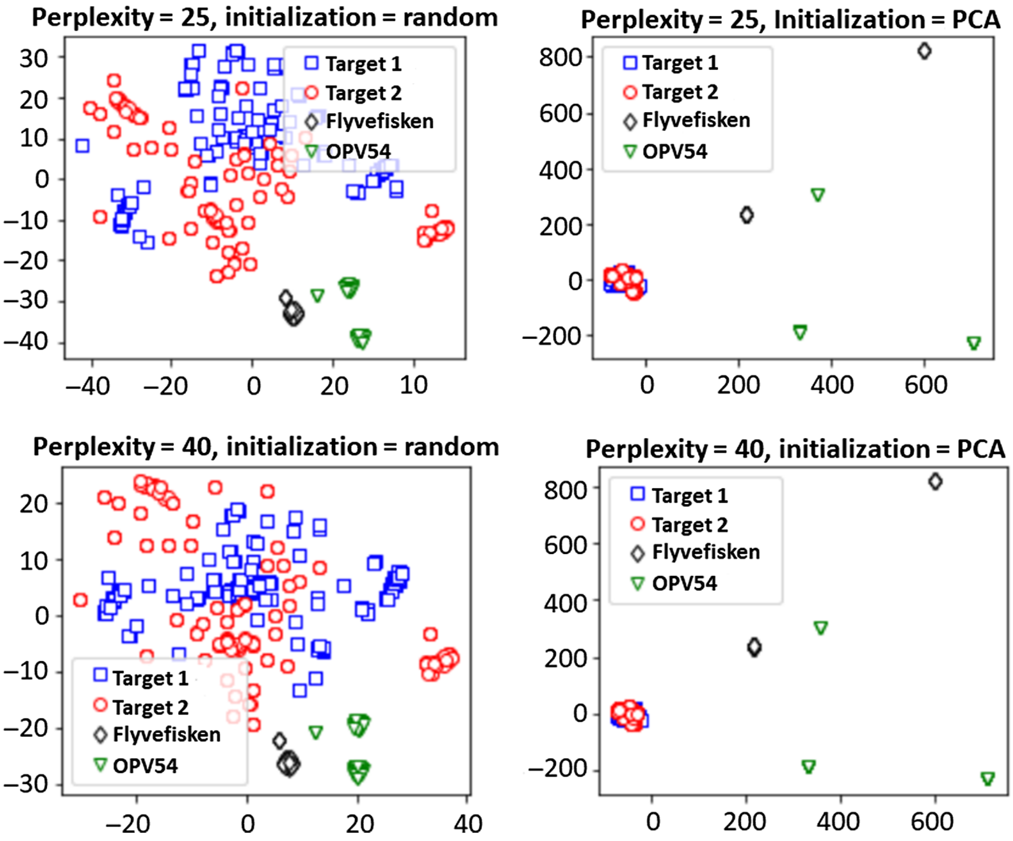

Fig. 14 t-SNE plot for the magnitude images of the four targets with random and PCA initialization and perplexity values of 25 and 40.

balances the trade-off between the model complexity and empirical error. When C is large, SVM mainly fits the samples in the training set, resulting in overfitting. On the other hand, when C is small, SVM tries to minimize the model coefficients, resulting in a very simple model leading to underfitting. In our approach, we empirically chose the values of $\gamma = 10$ and $C = 10$ by trial and error. Based on this setup, the SVM classifier was able to achieve an accuracy of 98% on the testing dataset.

5 Conclusions

We studied the target recognition problem in the context of ISAR imagery for maritime ship classification and compared traditional model-based approaches and deep learning approaches. We presented five different approaches: (i) model-based approach that uses scattering centers on the target for classification, (ii) supervised deep learning approach that uses a trained CNN for ISAR target recognition, (iii) DLF that takes the normalized average of the predictions of the CNN and model-based approaches as the final prediction, (iv) LFF that uses the features of the trained CNN and model-based approaches to train an SVM classifier, and (v) EFF that uses the features extracted from the model-based approach as prior information along with the input image to train the CNN. We evaluated all five approaches in the presence of three different distortions, namely: occlusion, clutter, masking of the body, and adversarial attack. After analyzing the results, we empirically find that the model-based approach is sensitive to the locations of the scattering centers, whereas the CNN is sensitive to the superstructure of the target. Experimental results show that the model-based approach achieves lower performance compared to the CNN when the locations of the scattering centers are affected due to occlusion and clutter. On the other hand, the CNN achieves lower performance compared to the model-based approach when body of the target is masked. This empirical finding suggests that the CNN and model-based approaches are able to extract features that are almost complementary to each other. Moreover, when we fuse these two approaches at both the feature level and decision level, we are able to achieve an increase in the performance, which is also robust against different distortions. Based on these findings, we can safely conclude that using an ensemble of approaches, where each approach is able to provide unique features of a given target, we can design a target classifier that is robust in prediction under various distortions compared to using stand-alone classifiers.

Acknowledgments

This work was supported by the U.S. Naval Air Warface Center, under Contract No. N68335-19-C-0108. The content of the information does not necessarily reflect the position or the policy of the U.S. Government and no official endorsement should be inferred.

References

1. F. Zhu et al., "Compressive sensing in ISAR spectrogram data transmission," in *IEEE Conf. Synth. Aperture Radar*, pp. 89–92 (2009).
2. X. Bai et al., "High-resolution three-dimensional imaging of spinning space debris," *IEEE Trans. Geosci. Remote Sens.* **47**(7), 2352–2362 (2009).
3. F. Zhou et al., "Nonparametric Bayesian 3-D ISAR imaging of space debris," in *IEEE Int. Geosci. and Remote Sens. Symp.*, pp. 4595–4598 (2018).
4. S.-J. Lee, S.-H. Park, and K.-T. Kim, "Improved classification performance using ISAR images and trace transform," *IEEE Trans. Aerosp. Electron. Syst.* **53**(2), 950–965 (2017).
5. A. Karine et al., "Target recognition in radar images using weighted statistical dictionary-based sparse representation," *IEEE Geosci. Remote Sens. Lett.* **14**(12), 2403–2407 (2017).
6. A. Lazarov and C. Minchev, "ISAR geometry, signal model, and image processing algorithms," *IET Radar Sonar Navig.* **11**(9), 1425–1434 (2017).
7. A. Manno-Kovacs et al., "Automatic target classification in passive ISAR range-crossrange images," in *IEEE Radar Conf.*, pp. 206–211 (2018).

8. F. Rice, T. Cooke, and D. Gibbins, "Model based ISAR ship classification," *Digital Signal Process.* **16**(5), 628–637 (2006).
9. B. Bhanu and Y. Lin, "Genetic algorithm based feature selection for target detection in SAR images," *Image Vision Comput.* **21**(7), 591–608 (2003).
10. G. Jones, III and B. Bhanu, "Recognizing articulated objects in SAR images," *Pattern Recognit.* **34**(2), 469–485 (2001).
11. G. Jones and B. Bhanu, "Recognition of articulated and occluded objects," *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(7), 603–613 (1999).
12. G. Jones and B. Bhanu, "Recognizing occluded objects in SAR images," *IEEE Trans. Aerosp. Electron. Syst.* **37**(1), 316–328 (2001).
13. M. Vespe, C. J. Baker, and H. D. Griffiths, "Outline structural representation for radar target classification based on non-radar templates," in *CIE Int. Conf. Radar*, pp. 1–4 (2006).
14. J. G. Verly, R. L. Delanoy, and C. H. Lazott, "Principles and evaluation of an automatic target recognition system for synthetic aperture radar imagery based on the use of functional templates," *Proc. SPIE* **1960**, 57–72 (1993).
15. J. H. Yi, B. Bhanu, and M. Li, "Target indexing in SAR images using scattering centers and the Hausdorff distance," *Pattern Recognit. Lett.* **17**(11), 1191–1198 (1996).
16. A. Beinglass and H. J. Wolfson, "Articulated object recognition, or: how to generalize the generalized Hough transform," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 461–466 (1991).
17. B. R. Mahafza, *Introduction to Radar Analysis*, Chapman and Hall/CRC, New York (2017).
18. J. Deng et al., "Imagenet: a large-scale hierarchical image database," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 248–255 (2009).
19. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, pp. 1097–1105 (2012).
20. X. He, N. Tong, and X. Hu, "Automatic recognition of ISAR images based on deep learning," in *CIE Int. Conf. Radar*, pp. 1–4 (2016).
21. C. Hu et al., "Inverse synthetic aperture radar imaging using a deep ADMM network," in *IEEE Int. Radar Symp.*, pp. 1–9 (2019).
22. B. Xue and N. Tong, "Real-world ISAR object recognition using deep multimodal relation learning," *IEEE Trans. Cybern.* (2019).
23. B. Xue and N. Tong, "DIOD: fast and efficient weakly semi-supervised deep complex ISAR object detection," *IEEE Trans. Cybern.* (2018).
24. J. Geng et al., "High-resolution SAR image classification via deep convolutional autoencoders," *IEEE Geosci. Remote Sens. Lett.* **12**(11), 2351–2355 (2015).
25. Y. Zhou et al., "Polarimetric SAR image classification using deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.* **13**(12), 1935–1939 (2016).
26. Z. Huang, Z. Pan, and B. Lei, "Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data," *Remote Sens.* **9**(9), 907 (2017).
27. Z. Zhao et al., "Discriminant deep belief network for high-resolution SAR image classification," *Pattern Recognit.* **61**, 686–701 (2017).
28. M. Gong, H. Yang, and P. Zhang, "Feature learning and change feature classification based on deep learning for ternary change detection in SAR images," *ISPRS J. Photogramm. Remote Sens.* **129**, 212–225 (2017).
29. Z. Zhang et al., "Complex-valued convolutional neural network and its application in polarimetric SAR image classification," *IEEE Trans. Geosci. Remote Sens.* **55**(12), 7177–7188 (2017).
30. F. Gao et al., "Dual-branch deep convolution neural network for polarimetric SAR image classification," *Appl. Sci.* **7**(5), 447 (2017).
31. S. Tian et al., "SAR object classification using the DAE with a modified triplet restriction," *IET Radar, Sonar Navig.* (2019).
32. E. R. Keydel, S. W. Lee, and J. T. Moore, "MSTAR extended operating conditions: a tutorial," *Proc. SPIE* **2757**, 228–243 (1996).
33. C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.* **2**(3), 27 (2011).
34. H. Xie et al., "Multilayer feature learning for polarimetric synthetic radar data classification," in *IEEE Geosci. and Remote Sens. Symp.*, pp. 2818–2821 (2014).

35. B. Mamatha and V. V. Kumar, "ISAR image classification with wavelet and watershed transforms," *Int. J. Electr. Comput. Eng.* **6**(6), 3087–3093 (2016).
36. S. J. Lee et al., "Classification of ISAR images using variable cross-range resolutions," *IEEE Trans. Aerosp. Electron. Syst.* **54**(5), 2291–2303 (2018).
37. R. Paladini, M. Martorella, and F. Berizzi, "Classification of man-made targets via invariant coherency-matrix eigenvector decomposition of polarimetric SAR/ISAR images," *IEEE Trans. Geosci. Remote Sens.* **49**(8), 3022–3034 (2011).
38. V. Zeljković et al., "Automatic algorithm for inverse synthetic aperture radar images recognition and classification," *IET Radar, Sonar Navig.* **4**(1), 96–109 (2010).
39. M. Martorella et al., "Target recognition by means of polarimetric ISAR images," *IEEE Trans. Aerosp. Electron. Syst.* **47**(1), 225–239 (2011).
40. J. Sun, H. Li, and Z. Xu, "Deep ADMM-Net for compressive sensing MRI," in *Adv. Neural Inf. Process. Syst.*, pp. 10–18 (2016).
41. S. Boyd et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.* **3**(1), 1–122 (2011).
42. T. Cooke et al., "Use of 3D ship scatterer models from ISAR image sequences for target recognition," *Digital Signal Process.* **16**(5), 523–532 (2006).
43. "Ansys® Electronics Desktop," ANSYS Inc., Canonsburg, Pennsylvania, https://www.ansys.com/products/electronics/ansys-hfss (2019).
44. D. Tao, A. P. Doulgeris, and C. Brekke, "A segmentation-based CFAR detection algorithm using truncated statistics," *IEEE Trans. Geosci. Remote Sens.* **54**(5), 2887–2898 (2016).
45. P. Lombardo and M. Sciotti, "Segmentation-based technique for ship detection in SAR images," *IEE Proc.-Radar, Sonar Navig.* **148**(3), 147–159 (2001).
46. I. McConnell and C. J. Oliver, "Segmentation-based target detection in SAR," *Proc. SPIE* **3869**, 45–54 (1999).
47. A. Neri, S. Colonnese, and G. Russo, "Automatic moving-object and background segmentation by means of higher order statistics," *Proc. SPIE* **3024**, 246–256 (1997).
48. G. Yang et al., "Study on statistics iterative thresholding segmentation based on aviation image," in *IEEE Int. Conf. Software Eng., Artif. Intell. Networking, and Parallel/Distrib. Comput.*, vol. 2, pp. 187–188 (2007).
49. A. Ortiz et al., "On the defense against adversarial examples beyond the visible spectrum," in *IEEE Mil. Commun. Conf.*, pp. 1–5 (2018).
50. P. Svenmarck et al., "Possibilities and challenges for artificial intelligence in military applications," in *NATO Big Data and Artif. Intell. for Mil. Decis. Making Specialists Meeting* (2018).
51. L. Aurdal et al., "Adversarial camouflage (AC) for naval vessels," *Proc. SPIE* **11169**, 111690K (2019).
52. Y. Shi et al., "Spectrum data poisoning with adversarial deep learning," in *IEEE Mil. Commun. Conf.*, pp. 407–412 (2018).
53. I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv:1412.6572 (2014).
54. L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
55. P. Singh and R. Shree, "Analysis and effects of speckle noise in SAR images," in *IEEE Int. Conf. Adv. Comput. Commun. Autom.*, pp. 1–5 (2016).
56. K. He et al., "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 770–778 (2016).
57. M. Sandler et al., "Mobilenetv2: Inverted residuals and linear bottlenecks," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 4510–4520 (2018).

Rajkumar Theagarajan received his BE degree in electronics and communication engineering from Anna University, Chennai, India, in 2014, and his MS degree in electrical and computer engineering from the University of California at Riverside (UCR), Riverside, CA, USA, in 2016. He is currently pursuing his PhD in electrical and computer engineering at the Center for Research in Intelligent Systems of the UCR. His research interests include computer vision, image processing, pattern recognition, and machine learning.

Bir Bhanu received his SM and EE degrees in electrical engineering and computer science from Massachusetts Institute of Technology, Cambridge, MA, USA, his PhD in electrical engineering from the University of Southern California, Los Angeles, CA, USA, and his MBA degree from the University of California at Irvine, Irvine, CA, USA. He is the founding professor of electrical engineering at the UCR, Riverside, CA, USA, and served as its first chair from 1991 to 1994. He is currently the Bourns Endowed University of California Presidential Chair in engineering, the distinguished professor of electrical and computer engineering, and the founding director at the Interdisciplinary Center for Research in Intelligent Systems (1998 to 2019) and the Visualization and Intelligent Systems Laboratory at UCR. He has published extensively and has 18 patents. Prioir to joining UCR, he was a senior Honeywell fellow at Honeywell Inc. He is a fellow of IEEE, AAAS, IAPR, SPIE, and AIMBE. His research interests include computer vision, pattern recognition and data mining, machine learning, artificial intelligence, image processing, image and video database, graphics and visualization, robotics, human–computer interactions, and biological, medical, military, and intelligence applications.

Tugba Erpek received her MSc degree in electrical and computer engineering from George Mason University in 2007. She is currently pursuing her PhD in electrical and computer engineering at Virginia Tech focusing on machine learning for communication systems. She is a senior research engineer at the Intelligent Automation, Inc. (IAI). Prior to joining IAI, she worked as a senior communications systems engineer at Shared Spectrum Company. Her research interests include wireless communications, machine learning, deep learning, MIMO systems, cognitive radio, dynamic spectrum access, network protocols, and routing algorithms.

Yik-Kiong Hue received his BS, MS, and PhD degrees in electrical engineering from The Ohio State University, Columbus, OH, in 2001, 2003, and 2006, respectively. He is a lead research scientist at IAI His research interests include radar, imaging, channel modeling, antenna, and wireless communications.

Robert Schwieterman received his BSEE degree from Purdue University in 2016 and his MSCS degree from Georgia Institute of Technology in 2018. As a research scientist at Intelligent Automation Inc., he has designed and implemented machine learning algorithms for classification and object detection tasks on a variety of domains. His research interests include spectral graph theory and machine learning.

Kemal Davaslioglu received his PhD in electrical and computer engineering from the University of California, Irvine, in 2015. He is a senior research scientist at IAI. Prior to joining IAI, he has held long-term internships at Broadcom Inc., Irvine, CA, USA, where he worked on beamforming algorithms and 60-GHz channel characterization. His research interests include resource allocation in wireless networks, machine learning, adversarial learning, and cybersecurity.

Yi Shi received his PhD from Virginia Tech in 2008. He is a lead research scientist at IAI and an adjunct faculty at Virginia Tech. His research interests include algorithm design, optimization, machine learning, and communication networks. He has been an editor for IEEE Communications Surveys and Tutorials and a TPC chair for IEEE and ACM workshops and symposium. He was a recipient of IEEE INFOCOM 2008 Best Paper Award, IEEE INFOCOM 2011 Best Paper Award Runner-Up, ACM WUWNet 2014 Best Student Paper Award, and IEEE HST 2018 Best Paper Award.

Yalin E. Sagduyu received his BS degree in electrical and electronics engineering from Bogazici University, Turkey and his MS and PhD degrees in electrical and computer engineering from the University of Maryland at College Park. He is the director of the Networks and Security at the Intelligent Automation Inc., Rockville, MD, USA. His research interests include networks, communications, optimization, and machine learning. He was a recipient of the Best Paper Award at IEEE HST 2018. He has chaired workshops at ACM MobiCom, ACM WiSec, IEEE CNS, and IEEE ICNP, has served as the track chair of IEEE PIMRC, IEEE MILCOM, and IEEE GlobalSIP, and has served on the Organizing Committee of IEEE GLOBECOM.