Classification of Ships using ISAR Images with Combined Deep Transfer Learning and GAN Processing Framework

Mahadev Mahesh Maitri
Electronics and Communication Engineering
R V College of Engineering
Bengaluru, India

Nethravathi K. A
Assistant Professor, Electronics and Communication Engineering
R V College of Engineering
Bengaluru, India

***Abstract*— Deep learning techniques like Convolutional Neural Network (CNN) are being used for processing of Inverse Synthetic Aperture (ISAR) images to enhance resolution for better imaging quality. Work is seldom done for classification of ships from ISAR images using deep learning techniques. Moreover, the technique requires a large database which is difficult to construct in a real-world scenario. In this work, the problem is addressed using Generative Adversarial Network (GAN), a framework connected by generator and discriminator networks. The combined loss is used to train the GAN which is composed of the adversarial loss and the absolute loss. With the help of absolute loss, the random noise is reconstructed to a meaningful ISAR image of a ship by a generator whereas the adversarial loss enhances weakly scattered points and amplitude by a discriminator which is trained to distinguish between model distribution and data distribution. The GAN is kept training till there is an equilibrium between the networks. For ship classification, a technique called transfer learning is used with Deep Convolutional Neural Network (DCNN) MobileNetV2 as a backbone consisting of linear bottlenecks and inverted residuals and achieve 90% accuracy as experimental results. Comparing this work with existing state- of-the-art methods, GAN used to generate ISAR images and DCNN for ship classification yields better results and more details of target.**

Keywords*— *Inverse Synthetic Aperture Radar (ISAR), Deep Convolutional Neural Network (DCNN), Generative Adversarial Network (GAN), Transfer Learning, Image Classification.

I. INTRODUCTION

Inverse Synthetic Aperture Radar (ISAR) is widely used in remote sensing for the acquisition of radar images to detect and classify ship targets for military applications. Automatic recognition of ships using ISAR images is achieved in various ways such as feature matching between ISAR images and 3-D geometry projection to Image Projection Plane (IPP) [1] and probabilistic recognition of ship using deep learning technique such as combining Faster-Region based Convolutional Neural Network (Faster-RCNN) [2] and Bayesian fusion [3]. Basically, this is narrowed down to two main methods: classification using feature matching and neural network.

The formation of ISAR images is achieved by having static imaging radar and motioned [4] ship along its axis. ISAR generates a series of range-Doppler image frames which is described in [5]. Imaging Radar is able to detect, track and image targets at long range with high accuracy in all weather conditions. Since simulation of ISAR images is tedious using mathematical solvers, this can be overcome by a well-known framework which can generate data similar to data distribution called Generative Adversarial Network (GAN) [6]. For image classification, the typical number of images in the dataset should be more than 500 per class (categories). Initial ISAR images of ships are generated using mathematical solvers and are increased with the help of GAN. This technique is well suited for generation and translation of images. This framework was introduced by Ian J. Goodfellow to generate similar data for the applications of artificial intelligence. In this framework, two models are trained namely, a generative model G and a discriminative model D. The functionality of these two models are, one measures the distribution of data and the other calculates the likelihood of how close the training data is to the output of the generator. When the model is in training mode, G is attempting to increase the likelihood that D would make an error. When both models are assigned with arbitrary functions, there exists a unique approach with G reconstructing the data distribution and D attaining the probability equal to 12 everywhere. This is similar to a two-player game where one tries to win over the other. Similarly, GAN can also be used to improve the resolution of the ISAR images by considering ISAR images of high-resolution in the data distribution [7].

As for the classification of ships, in the conventional radars, ships are classified at a broader level based on features extracted such as Radar Cross Section (RCS) and speed of the ships. These features are not sufficient to classify ships for military applications. With imaging radars of high resolution, more features could be extracted for detailed level of classification. With advent of computer vision algorithms, extraction of features from inverse synthetic aperture radar (ISAR) images is of significance. But the autoencoder, neural network used to learn efficient data codings, extracts the maximum number of features from the ISAR images for classification. This is achieved by Deep Convolutional Neural Network [8]. It has been studied over the last decade as one of the most effective tools, which has become very popular in literature because it can handle a large amount of knowledge. Recently, the concept of providing deeper hidden layers has begun to exceed the success of classical methods in major fields; mainly in pattern recognition. The parameters [9] which are learnt over the iterations are produced by convolutional and fully-connected layers and not by pooling and activation layer. With hardware accelerator devices (e.g. Graphics Processing Units or GPUs) available for computers, machine learning techniques are commonly used in many applications such as

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

image processing [10], natural language processing for speech recognition [11], speech separation [12], etc. In recent years, Automation Target Recognition (ATR) methods [13], [14] for ISAR images based on feature matching have been widely used and achieved very good results. Instead of using completely new neural networks, a new technique called transfer learning [15], [16], [17] is used in order to achieve the greater performance. It is the transfer of knowledge obtained by solving a kind of problem and using it to solve a different but same kind of problem. This technique is used where there is deficiency of the data in the dataset (less than the required). In [18], the recognition of sea targets is achieved from correlation between ISAR images and optical images. After processing them jointly, conditional GAN (cGAN) is used to translate the pix2pix information from ISAR images to corresponding optical images and CNN is trained for target recognition.

In this work, the GAN framework is used to increase the images in the dataset and transfer learning for classifying the different categories of ship. To optimize the generator, a combined loss is applied which is composed of adversarial loss and absolute loss. The absolute loss helps to structure the ISAR images out of random noise. The adversarial loss helps the generator to gain all the data points scattered widely. Lastly, categorical loss helps to classify the ships according to their respective classes.

In the next section, the detailed state-of-the-art solution is provided for GAN and transfer learning with the updating of weights and biases using forward and backpropagation. In section 3, the results obtained from this method are analysed and tools used to achieve the same. Additionally, the simulation of ISAR images of ships using ANSYS Electromagnetics is discussed.

II. METHODOLOGY

A. ISAR Image Pre-Processing

The flowchart of target recognition for ISAR images based on GAN and DCNN using transfer learning is shown in Fig. 1. If the entire ISAR image is labelled as a target, the matching features accounting background characteristics will affect the results of the classification because the target ship only takes up a portion of the ISAR image, thus the accuracy of the classification algorithm is reduced. If the ISAR image consists of background noise such as sea clutters which is strong, this may lead to decrease in the accuracy of the classification. Therefore, the image segmentation technique must be used to extract the target area of interest (ship structure) in center by applying pre-processing technique to ISAR images, which results in an increase in the accuracy of the classification algorithm.

First, the ISAR images are simulated using mathematical solvers, with resolution equal to 1m so that features like length and height are extracted precisely, from the available 3-D models. Using the radon transform [19], [20], the obtained angle at which ship structure is aligned in threshold ISAR images is used to orient the same ship structure to horizontal axis making the hull face at the bottom. The image is threshold by dynamic value to eliminate the sea clutters and speckle noise. Normalizing the ISAR images between -1 and 1 so that learning rate can be as low as possible. The number of samples in the range and cross-range depends on dimension along length and height of the ship i.e. shape of the ISAR images. Indirectly ISAR images of different categories have different shape such as cruise and warship. So, all the images are padded with redundant data symmetrically on both axes making it 256x256 shape. This helps in training and generating the better images from the generator.

B. GAN bases framework

The framework which is designed based on GAN is shown in Fig. 2, including image generation and mapping between generator and discriminator. The main objective of the GAN is to translate the noise to corresponding ISAR image from the data distribution (contains simulated ISAR images) using a generator G_{θ_G} that is achieved by end-to-end mapping of model distribution and data distribution with the help of DCNN. This can be transformed into a problem of optimization which is expressed as:

$$\hat{\theta}_G = \arg\min_{\theta_G} \frac{1}{m}\sum_{i=1}^{m} \log\big(1 - D\big(G(z)\big), x\big) \qquad (1)$$

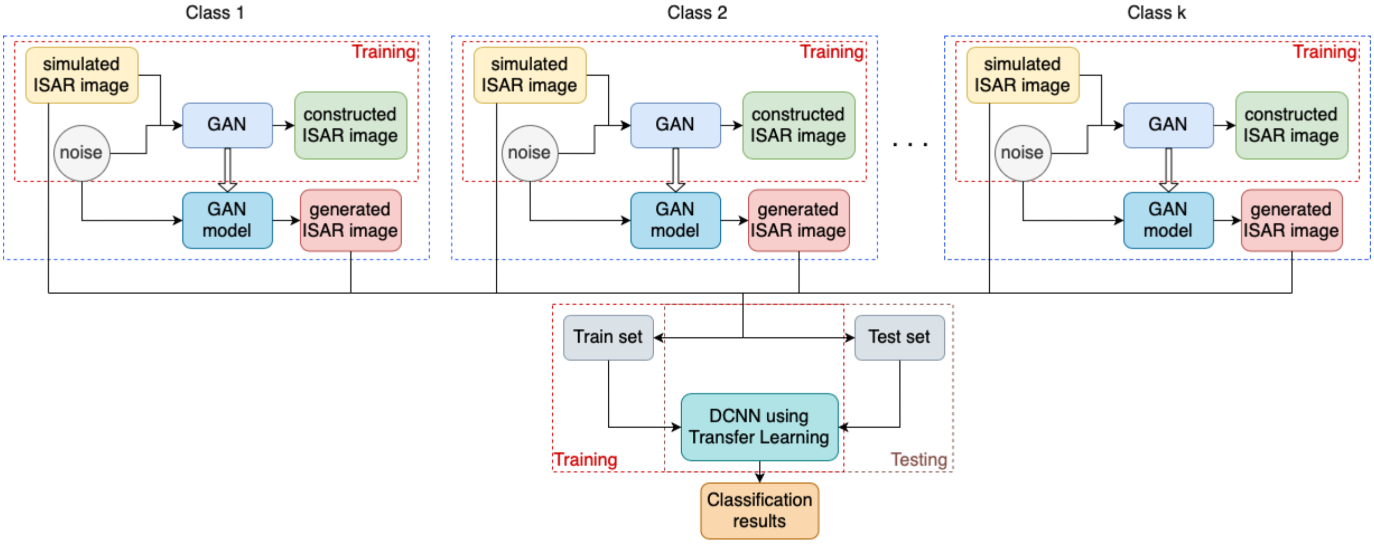


Fig. 1: The flowchart of target recognition for ISAR image based in GAN and DCNN using transfer learning.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

where z is random noise, θ_G is parameters of generator and x is real data (data distribution). From [4], discriminator D_{θ_D} can also be converted to optimization problem which is expressed as:

$$\hat{\theta}_D = \arg\max_{\theta_D} \frac{1}{m}\sum_{i=1}^{m} \log\big(D(x)\big) \qquad (2)$$

where θ_D is parameters of discriminator.

This design makes it possible to train the generator model G in order to bypass the differential discriminator D that is trained to distinguish the model distribution (generator output) from the data distribution (simulated ISAR images).It encourages the generator to construct an ISAR image from random noise and to recover all data points by introducing an adversarial loss.

The data distribution contains simulated ISAR images with help of mathematical solvers and pre-processed for training. The design and construction of both elements of GAN is discussed in the next sub section.

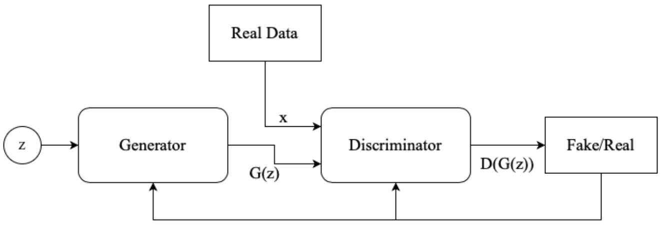


Fig. 2: The framework based on GAN for ISAR image

C. Design of GAN

The experiment carried out in [21] has shown that, in imaging equality, Complex Valued- CNN (CV-CNN) has no significant benefits over Real Valued-CNN with the exception of time consumption. Therefore, absolute values are used to learn the GAN and image classification [22]. The main features appear as bright spots (amplitude) in ISAR images, unlike optical images, which appear in different colour details and rich edges with differentiable backgrounds. To account for image contrast as the most important for the extraction of features that can be found in the Res-Net architecture [23], while preventing network degradation the image contrast is transmitted directly through skip connections.

Inspired by Res-Net architecture, designing your own convolutional neural network to work on image contrast feature extraction is the most important. For this work, the dataflow layers stacked for generators, is shown in Fig. 3, in order to extract the maximum number of features belonging to different ships. In the beginning, the random noise is given to the generator and reshaped to desired shape and subsequently convolutional operations are used to extract the features. The group of 2D-Convolution, Batch Normalization, Rectified Linear Unit and Upsampling (conv2d-BN-ReLU-Upsampling) layers acts as residual blocks, where it helps to structure the noise into ship images and upsample the data at each residual block to obtain desired ISAR image size.

Similarly, CNN is used to design discriminators with flow of data as shown in Fig. 4. It takes the batch of ISAR images from data distribution as input and corresponding output from generator (model distribution) to learn the parameters and improvise the generator to produce better images. The job of a discriminator is to classify the input image into fake or real with respect to data distribution. The seven convolutional layers are stacked in a sequence with an increasing number of feature maps, each layer being increased by a factor of 2 from 16 to 256 except for last two layers as in the Visual Geometry Group Network (VGGNet) which is a very deep convolutional neural network [24]. In the ISAR images, each pixel is related to the cross-section of the target and the range of pixel values is broad, the data distribution images are more susceptible to normalization, which is already accomplished in the pre-processing portion. The dropout layer is used in order to choose multiple independent paths to avoid overfitting of data. The PReLU layer is used to avoid sparse gradients instead of ReLU. The performance can be achieved at the highest level by deleting batch normalization (BN) layers, which reduces the computational complexity of the residual blocks. Since ISAR images are prone to image contrast and it is necessary to normalize the data in blocks, in this work, BN layers are used in the GAN architecture which is proven in [25].

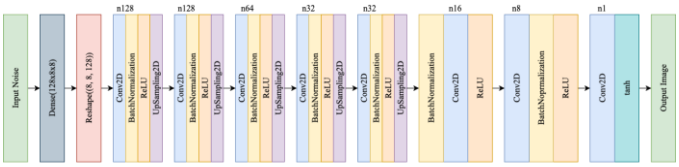


Fig. 3: Stack of dataflow layers for Generator

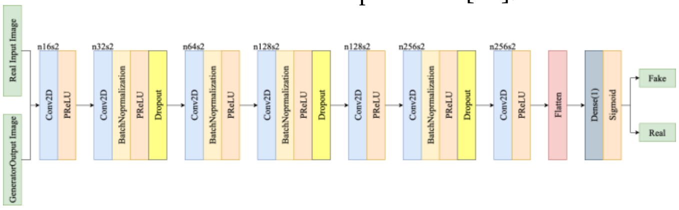


Fig. 4: Stack of dataflow layers for Discriminator

D. Combined Loss

For better performance out of the generator, a definition of a loss function is critical. Mean Squared Error (MSE) is widely used in many applications but cross-entropy helps loss to be calculated in log scale to prevent large range of numbers. Precisely, binary cross-entropy loss is used since it is the log loss it helps in training the network faster compared to others. The binary cross-entropy is defined as follows:

$$loss = -[y\log(\hat{y}) + (1-y)\log(1-\hat{y})] \qquad (3)$$

where y and $\hat{y}$ represent actual class and predicted class, respectively. Because the significant target ship features only appear as bright spots in the ISAR images, if the loss function used for the generator is binary cross-entropy, due to log operation all the feature weights calculation will be relatively equal, resulting in minimal enhancement and loss of some weak point scatters. For discriminator, its job is to classify the generator output between fake or real i.e. binary classification, the loss function used is binary cross-entropy.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

The adversarial loss [6] is defined over all training samples as follows:

$$\min_{G} \max_{D} V(D, G(z)) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z), x))] \qquad (4)$$

where x and z are data distribution and random noise, respectively. In addition to strong point scatters, after normalization, the pixel values (amplitude) of most point scatters in a single ISAR images are extremely close to 1, the regularization parameter is used to ensure that the adversarial loss is at least 10 times lesser than the absolute loss preventing the training process lead by the adversarial loss, this is identified over simulation results.

E. Ship Image Classification

For the ISAR images [26], an important feature is image contrast and weakly scattered data points. In order to account these into classification between different categories of ships, classifiers should be designed to have a lesser number of parameters and no redundant parameters. The classifier could be designed completely from scratch, but the number of images in the dataset for each class is less than what is required, resulting in the use of a new technique called transfer learning [15], [16]. This technique is popular with neural networks where there is a small amount of data. It is the transfer of knowledge obtained by solving a kind of problem and using it to solve a different but same kind of problem. There are a number of well-trained models for transfer learning to image classification such as MobileNetV2 [27], InceptionV2 [28]. For the work, the light-weight model is sufficient for the classification, so MobileNetV2 is considered and it is trained on ImageNet dataset with 1000 classes. It boosts the mobile model's state- of-the-art efficiency on various benchmarks and tasks as well as across a number of different model sizes. This model is a reduced form of DeepLabV3 [29] designed to outperform semantic segmentation on mobile platforms, and this is achieved by having shortcut connections between thin bottleneck layers called inverted residual blocks. The bottle-neck layers for strides 1 and 2 are shown in Fig. 5.

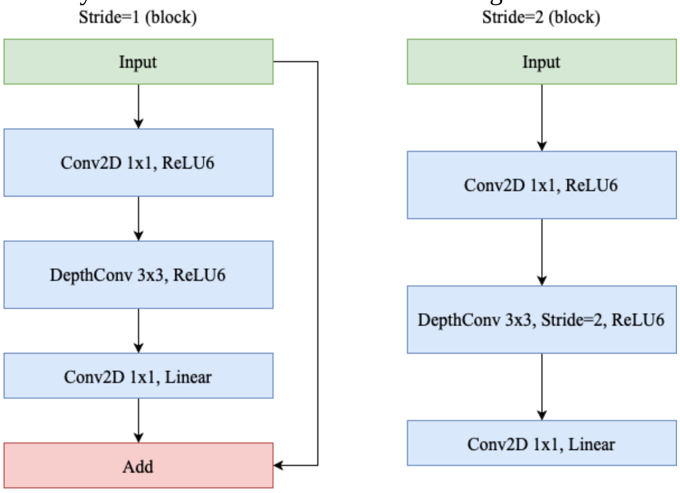


Fig. 5: Bottle-neck layers of MobileNetV2 [27]

This model extends the ideas of MobileNet [30], and the efficient building blocks of this network contain a depthwise separable convolution layer. However, MobileNetV2 is designed with two new blocks and connections in the architecture that are linear bottle-necks between the layers and shortcut connections between the bottle-necks. The sequence of layers stacked in the architecture with stride and number of feature maps (output channels or filters) is shown in Table 1 for the input image size of 224x224. It is used as an encoder and few fully connected layers are appended to it to act as a classifier. In general, the slope of curve with transfer learning is higher than without transfer learning in the graph of training versus performance.

In the proposed method, MobileNetV2 is used for transfer learning where the weights in each kernel is copied from a trained network. Here, the network was trained on ImageNet dataset with 1000 classes and over one million images. The copied weights is utilized to train ISAR images and thus minimizing the training time. If weights are initialized using initializers (like RandomUniform, GlorotNormal, etc.), this resulted in overfitting as there is limited number of images in dataset. As the trained network contains 1000 classes (connections in the last layer), all other connections except 3 (number of ship categories considered in the proposed network) are frozen during training mode as well as testing mode.

TABLE 1. MobileNetV2: stacked layers

Input	Operator	c	n	s
224x224x3	conv2d	32	1	2
112x112x32	bottle-neck	16	1	1
112x112x16	bottle-neck	24	2	2
52x52x24	bottle-neck	32	3	2
28x28x32	bottle-neck	64	4	2
14x14x64	bottle-neck	96	3	1
14x14x96	bottle-neck	160	3	2
7x7x160	bottle-neck	320	1	1
7x7x320	conv2d 1x1	1280	1	1
7x7x1280	avgpool 7x7	-	1	-
1x1x1280	conv2d 1x1	k	-	

where n is number of repetition layers, c is number of output channels and s is stride size. In this work, the model takes input as 3 channels and ISAR images have only 1 channel, this allows duplicating the same ISAR data into all 3 channels. This works the same as a neural network with 1 channel as input. The loss of classifier is calculated as in Eq. 3 as categorical cross-entropy which means it can be extended to N number of categories.

F. Forward Propogation

The convolution layer is a key to the extraction of features from images. This layer involves convolution operation between the output of the previous layer and the kernel to extract the features. When training the neural network, the kernel of the convolution layer is continuously modified by learning feature maps. When for each convolution operation specific convolutional kernels are used, more and more parameters need to be trained, as depth of network increases. The weight-sharing operation is adopted by CNN to decrease the number of the network's training parameters. The convolution process in convolutional layer operation which includes convolution and activation is expressed as:

$$y_j^{(l)} = \sum_i f_i^{(l-1)}(x, y) * w_{ij}^{(l)}(x, y) + b_j^{(l)} \qquad (5)$$

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

where $y_j^{(l)}$ is the j^{th} output of the l^{th} layer, $f_i^{(l)}(x,y)$ is any pixel of an ISAR image on the i^{th} feature map (layer output) of the l^{th} layer, $w_{ij}^{(l)}$ is the kernel matrix operating on i^{th} input feature map to get j^{th} output feature map on the l^{th} layer, $b_j^{(l)}$ is the j^{th} bias of the l^{th} layer, and $*$ denotes the 2-D convolution operation. In order to maximize nonlinear characteristics and to have a stronger classification capability of the network, the nonlinear activation function layer must be connected to each convolution layer:

$$f_j^{(l)}(x,y) = \sigma\left(y_j^{(l)}\right) \qquad (6)$$

where σ is the ReLU non-linear activation function. The final decision layer is a single dimensional array of the length of the number of ship categories when they are connected to a higher dimensional space. In this layer, convolutional kernel and bias are the trainable parameters.

The pooling layers are used to decrease the number of trainable parameters and are realized after convolution layer by means of "compression" of information in computer vision applications. Pooling layer are of two types average and maximum. Here, the maximum pooling is expressed as:

$$f_j^{(l)}(x,y) = \max_{m,n=0,\ldots,k-1} f_j^{(l)}(x+m, y+n) \qquad (7)$$

where k is the shape of pooling window. The features from ISAR images are extracted based on alternate convolutional and pooling layer.

Usually, the last layer of the network is softmax activation function for multi-class classification which is defined as posterior probability of output of previous layer, expressed as:

$$p(y_i|f^{(L)}) = \frac{e^{\left(f_i^{(L)}\right)}}{\sum_{j=1}^{k} e^{\left(f_j^{(L)}\right)}} \qquad (8)$$

where y_i is the predicted class of the i^{th} class, L is the number of layer, $f_j^{(L)}$ is the sum of the weight of the j^{th} node of the fully connected layer, k is the number of class, and $f^{(L)}$ is the input to last softmax activation layer. With the help of softmax activation, the probability vector is normalized and the predicted class is one with a label corresponding to the maximum posterior probability.

After forward propagation [31], the trainable network parameters are updated by some rules. The commonly used loss functions are Mean Squared Error (MSE) and cross-entropy which defines the rules to the network. The cross-entropy loss or log-loss is better for classification which reflects similarity between data distribution and model distribution:

$$Loss(W,b) = -\sum_{i=1}^{k} y^{(i)} \log p\left(y_i|f^{(L)}; W, b\right) \qquad (9)$$

where W and b are weight and bias matrix of the all the layer in the network respectively, and $y^{(i)}$ is the actual label of the i^{th} class. When k is 2, if probability of a class is p then the probability of other class is $(1-p)$, expanding and substituting the probabilities in Eq. 9 results in Eq. 3.

The problem of classification can be summed up as the problem of optimization, i.e. minimizing the loss function when the network is training on data distribution images.

G. *Backward Propogation*

The backpropagation is the building block of the neural network pointed out by Hinton in [32]. The probability vector from the output layer is the predicted class of the model. The error term can be evaluated by taking the difference between the actual label and the predicted label at the output layer:

$$df_i^{(L)} = -[y^{(i)} - p(y_i|f^{(L)}; W, b)] \qquad (10)$$

If the convolutional layer is present at $(l+1)^{th}$ layer, then the error term produced at i^{th} feature map (layer output) of the l^{th} layer is determined by $(l+1)^{th}$ layer's error term, calculated as:

$$df_i^{(l)} = \sigma'\left(f_i^{(l)}\right) \odot \sum_j df_j^{(l+1)} w_{ij}^{(l+1)} \qquad (11)$$

where $\odot$ denotes dot product, and $\sigma'(.)$ is the first derivative of the defined activation function (here, ReLU and PReLU).

If the pooling is present at $(l+1)^{th}$ layer, the error term produced at i^{th} feature map (layer output) of the l^{th} layer is calculated using:

$$df_i^{(l)} = \sigma'\left(f_i^{(l)}\right) \odot \mathrm{Up}\left(df_i^{(l+1)}\right) \qquad (12)$$

where $\mathrm{Up}(.)$ is the upsampling function in the dataflow.

The gradients of the weight and the bias of l^{th} layer is dependent on the error term of l^{th} layer which is calculated as:

$$\frac{\partial Loss}{\partial w_{ji}^{(l)}} = f_i^{(l-1)} df_j^{(l)} \qquad (13)$$

$$\frac{\partial Loss}{\partial b_j^{(l)}} = \sum_{x,y} df_j^{(l)}(x,y) \qquad (14)$$

Applying gradient descent, the weight and bias matrix is updated for each layer in the network as:

$$w \longleftarrow w - \eta \frac{\partial Loss}{\partial w} \qquad (15)$$

where η is the learning rate.

The optimized network parameters for classification is obtained with the help of forward [31] and backward [33], [34] propagation and also stable network. With the ISAR image classification, ISAR images of a ship are given as input to get their class attributes.

III. PERFORMANCE ANALYSIS

A. *Neural Network Training Details and Parameters*

First, the ISAR images of ships are simulated using electromagnetic solvers with resolution equal to 1m on both range and cross-range axis. The pre-processing of ISAR images is done as discussed above. In the dataset, three categories of ship are considered for classification, BNSBangabandhu, Cruise and Monitor-36 [from grabcad.com]. For each category, 6 ISAR images are simulated for training at different look angles where look angle varies from 20° to 70° with a difference of 10°.

The input ISAR images are kept to their original dimensions during the training process. The GAN training process is completed on an NVIDIA Tesla T4, which is accessible from Google Colab based on TensorFlow and takes around 4 hours. For updating the weights, the optimization used is Adam, and the momentum is 0.5, the learning rate of the generator is 0.0001, whereas the discriminator is 0.0002. The batch size for the hardware accelerator is 6, and 10 ISAR images for each category of ship are generated after training. Training lasts for

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

10000 epochs. The number of parameters for learning generators and discriminators is 1.23M and 1.64M. Similarly, for classifiers, Google Colab is used for training based on the same dataflow framework which took about 30 minutes on NVIDIA Tesla T4. RMSProp optimization is used for weight updating and the learning rate is 0.00001 with categorical cross-entropy loss. The batch size is 16 and training lasts about 50 epochs. The number of parameters which are learnable for classifiers is 41.95M where MobileNetV2 is non-trainable and the last layer has 3 neurons for categorical classification of ships. The visualization of classifier's loss and accuracy during the training process is shown in Fig. 6.

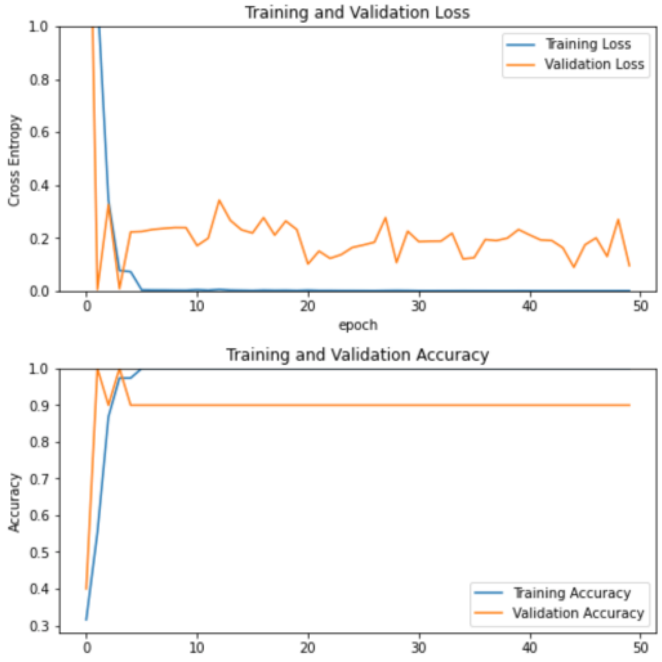


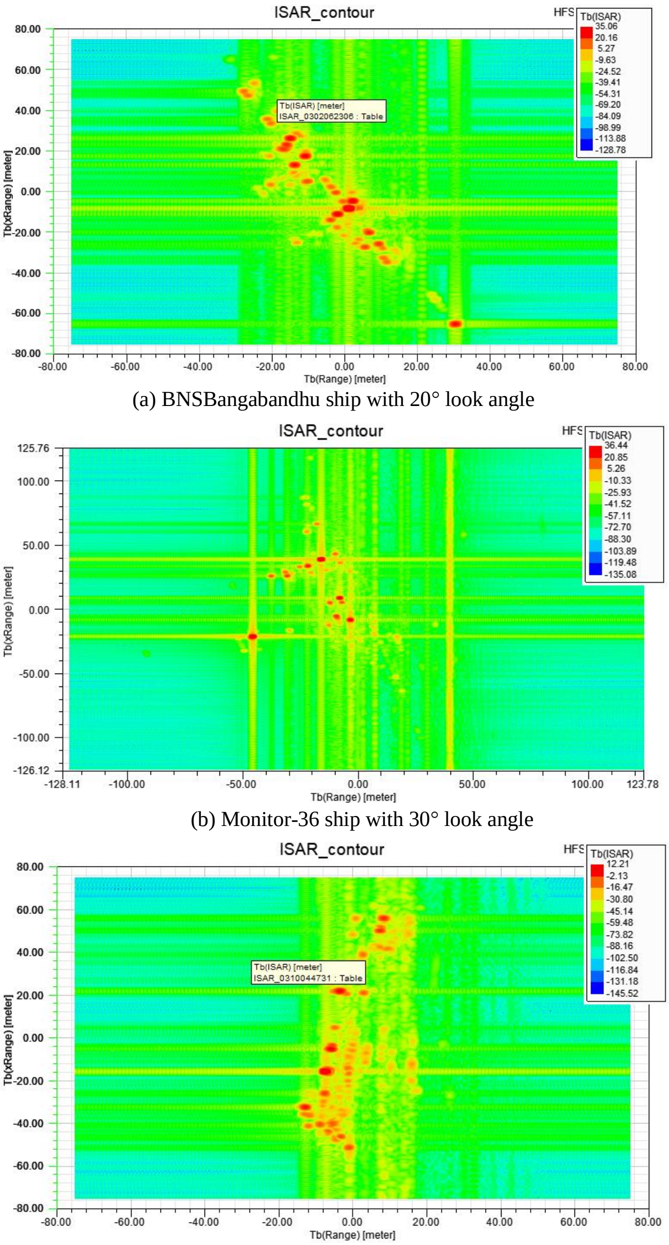
Fig. 6: Visualization of classifier's loss and accuracy

B. *Simulation of ISAR Images*

Initially, the ISAR images are simulated using electromagnetic solvers to train the GAN for generating the duplicates. The simulated and generated images are used for classification. The mathematical solver used to simulate the ISAR images is ANSYS Electromagnetics SBR+ Solver with RADARPre and RADARPost as ACT extensions. The parameters used for simulation is shown in Table II. The simulated ISAR images in range and cross-range axis (in logarithmic scale) are shown in Fig. 7.

The simulated ISAR images are obtained as following:

- Create a HFSS project in ANSYS Electromagnetics SBR+ Solver and import the 3-D model of the ship in workspace.
- Assign the material of the ship as "pec" and change the units as meter. The orientation of ship along x-axis, y-axis and z-axis should be ship length, height and width, respectively. Because the radar is placed in XY plane at an angle (look angle) w.r.t x-axis as per RADARPre.
- Open RADARPre window, enter the radar requirements as shown in Table II and Generate Radar setup.
- Set the Ray Density per Wavelength as 0.1 and check Edge Correction in Setup window.
- The model and setup is simulated, using RADARPost, simulated results are imported and ISAR images are obtained as shown in Fig. 7



(a) BNSBangabandhu ship with 20° look angle

(b) Monitor-36 ship with 30° look angle

(c) Cruise ship with 50° look angle

Fig. 7: ISAR Images in log scale

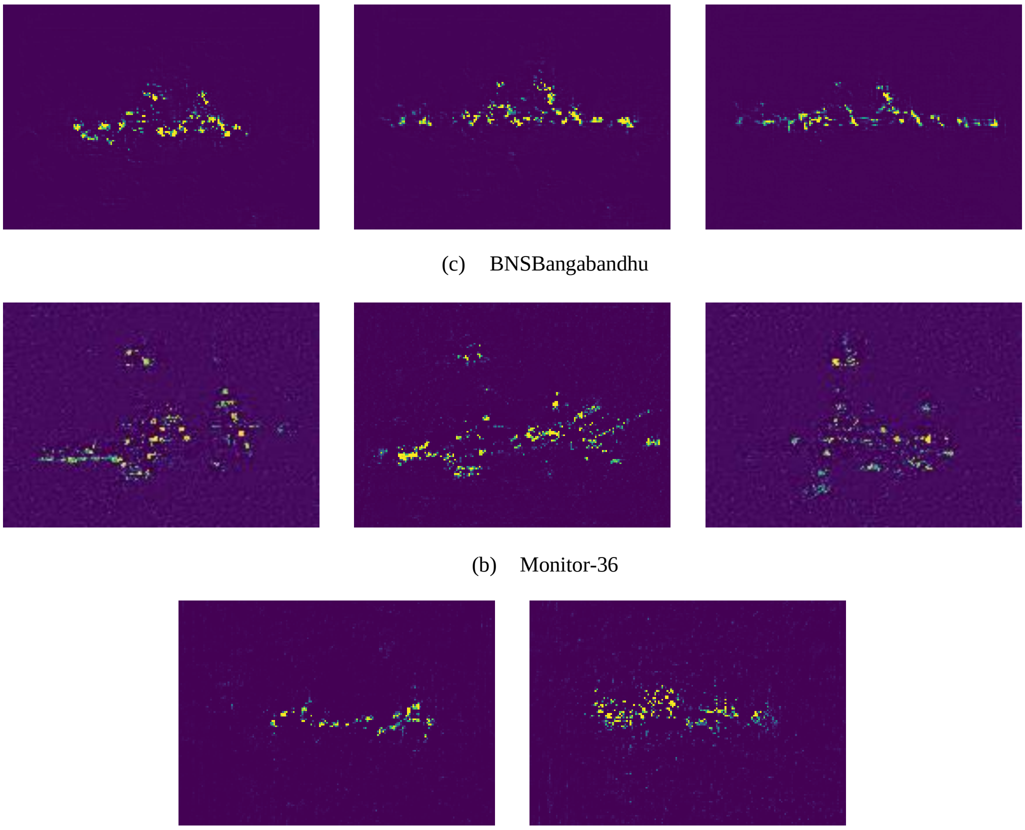
TABLE 2. Simulation parameters for ANSYS Electromagnetics

Parameter	Value
Frequency	9.5 GHz
Look Angle	20°-70° with 10° difference
Range resolution	1 m
Cross-range resolution	1 m
Range	Ship Length
Cross-range	Ship Height

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

C. *Laboratory Results*

In this work, GAN is utilized to generate ISAR images of ships which are similar to data distribution (real data). After training the GAN which is constructed using neural network, the results obtained for first category of ship (BNSBangabandhu) is shown in Fig. 8a. The second category of ship (Monitor-36) is shown in Fig. 8b. Similarly, for the last category of ship (Cruise) is shown in Fig. 8c. To identify the different categories of ship, the features extracted are ship silhouette, masts, length and area. For BNSBangabandhu, length and two masts acts as main features as well as silhouette gives the placement and height of masts, this is also the case for Monitor-36 but two masts are of different height and silhouette accounts for ship structure. As for Cruise, there are no masts but length, area and silhouette is enough to differentiate between other categories of ship. The images are scaled down to size of 180x128 for better visualization with x-axis and y-axis as range and cross-range (in logarithmic scale), respectively. The output images of GAN are compared to data distribution images in terms of length, height, number of masts, mast position, and area occupied by the ship. These can be directly compared with 3-D geometry models or ANSYS simulated images since the resolution along range and cross-range is 1 meter. As part of GAN working, generator and discriminator achieved equilibrium (valued ½) with its objective function where generator tries to minimize and discriminator tries to maximize. With this, GAN results are satisfactory for further applications. As for the ISAR image classification, transfer learning technique is utilized to achieve the state-of-the-art solution. The model used is MobileNetV2 which results an accuracy of 100% on training dataset and 90% accuracy on validation dataset. The results shown here are compared to the existing methods such as Multi-Feature ATR [1] where the author obtained an accuracy of 89.7% and 91.6% for single frame and two frame, respectively, with known aspect angle. The proposed method outperforms in terms of real time prediction and pre-processing towards the algorithm.



(c) BNSBangabandhu

(b) Monitor-36

(a) Cruise

Fig. 8: ISAR images obtained from GAN for the three categories [scale, 1pix = 1m]

IV. Conclusion

A GAN-based framework is developed for generating ISAR images and a combined loss comprising of the adversarial loss and the absolute loss for structuring the noise into ISAR images of ships. This is designed such that image contrast is considered as feature and the weakly scattered points. Comparing this work to existing and recent state-of-the-art methods, the results show that this method is superior in structuring ISAR images from random noise, amplitude, position and considering very weak point scatters. The ISAR image classification also outperforms existing methods and uses a transfer learning technique for classification. In the future, the GAN framework and the classifier can be further studied on several other sides, assuming that all the required imaging parameters are known in real-world scenarios. Instead of simple GAN, conditional GAN can be used to reduce the time consumption where condition is category of ship or InfoGAN can be used to gather maximum information from ISAR images using a variation of the Week-Sleep algorithm. ISAR image dataset needs to be increased by simulation and GAN and considering multi image perspective of ship while simulating i.e. pain view and broadside profile.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

ACKNOWLEDGMENT

The authors would like to thank Dr. Dyana. A from Electronics and Radar Development Establishment (LRDE), Defence Research and Development Organization (DRDO) for her help throughout the process and providing necessary resources.

REFERENCES

[1] D. Pastina and C. Spina, "Multi-feature based automatic recognition of ship targets in ISAR," *IET Radar, Sonar & Navigation*, vol. 3, no. 4, p. 406, 2009.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[3] K. Kim, S. Hong, B. Choi, and E. Kim, "Probabilistic ship detection and classification using deep learning," *Applied Sciences*, vol. 8, no. 6, p. 936, jun 2018.

[4] A. W. Doerry, "Ship dynamics for maritime ISAR imaging." Tech. Rep., feb 2008.

[5] T. Moore, "A new algorithm for the formation of ISAR images," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 2, pp. 714–721, apr 1996.

[6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[7] D. Qin and X. Gao, "Enhancing ISAR resolution by a generative adversarial network," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2020.

[8] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, L. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," 2015.

[9] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, "Recent advances in convolutional neural network acceleration," 2018.

[10] K. He, G. Gkioxari, P. Dolla´r, and R. Girshick, "Maskr-cnn," 2017.

[11] C. Weng, D. Yu, S. Watanabe, and B.-H. F. Juang, "Recurrent deep neural networks for robust speech recognition," in 2014 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, may 2014.

[12] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multi-talker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2017.

[13] A. Kurowska, J. S. Kulpa, E. Giusti, and M. Conti, "Classification results of ISAR sea targets based on their two features," in 2017 *Signal Processing Symposium (SPSympo)*. IEEE, sep 2017.

[14] S. Musman, D. Kerr, and C. Bachmann, "Automatic recognition of ISAR ship images," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4, pp. 1392–1404, 1996.

[15] A. Quattoni, M. Collins, and T. Darrell, "Transfer learning for image classification with sparse prototype representations," in 2008 *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2008.

[16] M. Shaha and M. Pawar, "Transfer learning for image classification," in 2018 *Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, mar 2018.

[17] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," 2018.

[18] G. Li, Z. Sun, and Y. Zhang, "ISAR target recognition using pix2pix network derived from cGAN," in 2019 *International Radar Conference (RADAR)*. IEEE, sep 2019.

[19] D. Pastina and C. Spina, "Slope-based frame selection and scaling technique for ship ISAR imaging," *IET Signal Processing*, vol. 2, no. 3, p. 265, 2008.

[20] D. Pastina, M. Bucciarelli, and C. Spina, "Multi-sensor rotation motion estimation for distributed isar target imaging," in 2009 *European Radar Conference (EuRAD)*, 2009, pp. 282–285.

[21] J. Gao, B. Deng, Y. Qin, H. Wang, and X. Li, "Enhanced radar imaging using a complex-valued convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, 2017.

[22] X. Shi, F. Zhou, S. Yang, Z. Zhang, and T. Su, "Automatic target recognition for synthetic aperture radar images based on super-resolution generative adversarial network and deep convolutional neural network," *Remote Sensing*, vol. 11, no. 2, p. 135, jan 2019.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[25] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in 2017 *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, jul 2017.

[26] M. Inggs and A. Robinson, "Ship target recognition using low resolution radar and neural networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 2, pp. 386–393, apr 1999.

[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2018.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014.

[29] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017.

[30] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.

[31] K. Hirasawa, M. Ohbayashi, M. Koga, and M. Harada, "Forward propagation universal learning network," in *Proceedings of International Conference on Neural Networks (ICNN'96)*. IEEE, 1996.

[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, oct 1986.

[33] A. A. Kohan, E. A. Rietman, and H. T. Siegelmann, "Error forward-propagation: Reusing feedforward connections to propagate errors in deep learning," 2018.

[34] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)