

# IMPROVING SAR AUTOMATIC TARGET RECOGNITION USING SIMULATED IMAGES UNDER DEEP RESIDUAL REFINEMENTS

Miriam Cha<sup>1,2</sup>, Arjun Majumdar<sup>2</sup>, H.T. Kung<sup>1</sup>, Jarred Barber<sup>2</sup>

<sup>1</sup>Harvard University, <sup>2</sup>MIT Lincoln Laboratory

## ABSTRACT

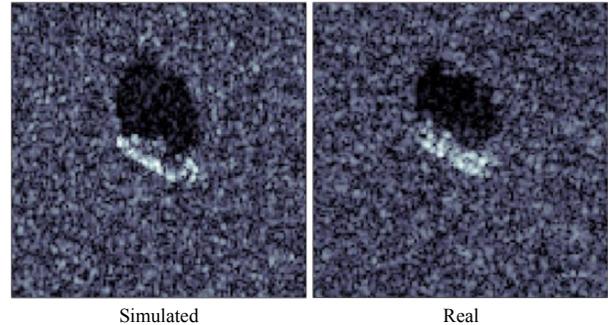
In recent years, convolutional neural networks (CNNs) have been successfully applied for automatic target recognition (ATR) in synthetic aperture radar (SAR) data. However, it is challenging to train a CNN with high classification accuracy when labeled data is limited. This is often the case with SAR ATR in practice, because collecting large amounts of labeled SAR data is both difficult and expensive. Using a simulator to generate SAR images offers a possible solution. Unfortunately, CNNs trained on simulated data may not be directly transferable to real data. In this paper, we introduce a method to refine simulated SAR data based on deep residual networks. We learn a refinement function from simulated to real SAR data through a residual learning framework, and use the function to refine simulated images. Using the MSTAR dataset, we demonstrate that a CNN-based SAR ATR system trained on simulated data under residual network refinements can yield much higher classification accuracy as compared to a system trained on simulated images, and so can training on real data augmented with these simulated data under refinements compared to training with real data alone.

**Index Terms**— Synthetic Aperture Radar (SAR), automatic target recognition, residual networks (ResNet)

## 1. INTRODUCTION

Synthetic aperture radar (SAR) automatic target recognition (ATR) systems aim to classify objects present in areas of interest within SAR data. Compared with classical image classification in optical imagery, a SAR ATR system needs to handle distinguishing characteristics of the SAR modality, such as a high dynamic range and spatial frequency. Recently, convolutional neural networks (CNNs) have achieved state-of-the-art performance in SAR ATR [1, 2, 3] while requiring substantial amounts of labeled training data. Unfortunately,

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited. This material is based upon work supported by the Assistant Secretary of Defense for Research and Engineering under Air Force Contract No. FA8721-05-C-0002 and/or FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Assistant Secretary of Defense for Research and Engineering.



**Fig. 1:** Real (right) and simulated (left) SAR images of a vehicle (BTR70) at  $17^\circ$  depression and  $300^\circ$  aspect angles

the process of collecting SAR images with ground truth labels is usually expensive and time-consuming. In order to fill in the data gap, developing SAR simulators aiming at generating realistic SAR data has been a focus of SAR ATR research.

Malmgren-Hansen *et al.*[4] propose pre-training a CNN-based ATR system on unlabeled simulated images of arbitrary vehicle types to improve network convergence. Odegaard and Cochlin [5] augment training data by combining real and simulated SAR images to improve ship classification. However, training a system exclusively on simulated SAR data may lead the network to learn features only present in the simulated images and fail to generalize to the real data environment.

One solution for closing the gap between simulated and real SAR data is to improve the simulator. However, faithfully modeling the underlying physics of SAR is computationally expensive. This limits the amount of data that can be generated for training purposes. High fidelity simulation also requires accurate target models. This increases the cost of creating a large number of these models. Furthermore, increasing the realism in simulated SAR images using human visual inspection is quite challenging, because contrast sensitivity in human visual system is poor when spatial correlation is low and significantly degrades at high frequency [6]. Figure 1 shows a real SAR image of a BTR70 vehicle in the MSTAR dataset on the right and the corresponding simulated image on the left at  $17^\circ$  depression and  $300^\circ$  aspect angles. The visual difference between the simulated and the real images is not obvious. However, when only simulated images are used in

training of a CNN classifier, SAR ATR performance on real images is very poor due to the gap between simulated and real image distributions (see Section 5.3).

In this paper, we introduce a method for refining simulated SAR images so that the refined images can be used in the training of a classifier to improve SAR ATR system. Specifically, we develop a deep residual learning framework [7] that takes in a simulated image provided by a SAR simulator based on a point scatterer model and outputs a refined image that estimates the real image distribution. We investigate the use of  $\ell_1$ - and  $\ell_2$ -norm losses. We present experiments on MSTAR dataset [8] to show that models trained with the refined images can perform better than the model trained on simulated images. In addition, we demonstrate that the refined images can be combined with real images to further improve classification performance when the labeled data is limited.

The rest of this paper is organized as follows. In Section 2, we provide background on SAR statistics and residual learning. In Section 3, we describe SAR simulator model used in our experiment. Section 4 details our model to refine simulated SAR images. Section 5 provides experimental evaluation of our model trained on the MSTAR dataset, and Section 6 concludes the paper.

## 2. BACKGROUND

### 2.1. Synthetic aperture radar (SAR)

Synthetic aperture radar (SAR) is an important modality in the remote sensing community due to its ability to form high resolution images in all-weather conditions. According to the literature [9, 10], we can assume that the statistical model for the signal received by the SAR sensor corresponding to a pixel in the SAR image has a zero-mean circularly complex Gaussian distribution. Thus, the magnitude  $x$  follows a Rayleigh distribution [11]:

$$p(x) = 2\lambda x \exp(-\lambda x^2), \quad x \geq 0 \quad (1)$$

where  $\lambda$  is the scale parameter.

### 2.2. Deep residual networks (ResNets) for SAR

Given training pairs of real and simulated SAR images, our refiner is aimed to learn a mapping  $R$  from a simulated image  $\mathbf{z}$  to an image  $\hat{\mathbf{x}}$  approximating the corresponding real image  $\mathbf{x}$  such that

$$\hat{\mathbf{x}} = R(\mathbf{z}), \quad \mathbf{z} \sim p_{\mathbf{z}} \quad (2)$$

where  $p_{\mathbf{z}}$  is the prior distribution for simulated images, which reflects the Rayleigh distribution of real images.

Residual networks (ResNet) have been shown to be an effective model for learning image-to-image transformations. A residual block, introduced by He *et al.* [7], consists of a residual function  $\mathcal{F}$  and an identity skip-connection. Traditionally,

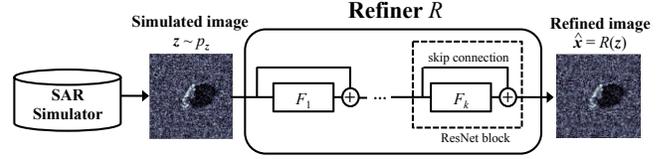


Fig. 2: Our overall architecture of refining simulated SAR images

the residual function  $\mathcal{F}$  contains convolution and rectified linear unit (ReLU) layers. A residual block can be expressed as

$$y' = y + \mathcal{F}(y) \quad (3)$$

where  $y$  and  $y'$  are the input and output of the block, respectively. A deep residual network contains many stacks of residual blocks and can be written as

$$y_j = y_{j-1} + \mathcal{F}_j(y_{j-1}) \quad (4)$$

where  $y_{j-1}$  is the input to the  $j$ th residual block and  $\mathcal{F}_j$  contains the weight layers.

Similar to our task, super-resolution requires generating improved images conditioned on noisy or incomplete data. Ledig *et al.* [12] present convincing evidence in favor of deep ResNet as a strong candidate architecture to learn mapping between low-resolution and high-resolution images. According to Ledig *et al.*, the use of deep residual networks can achieve state-of-the-art peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) index on standard image super-resolution benchmark. Johnson *et al.* [13] learn an image transformation network based on a deep ResNets with perceptual loss functions for image super-resolution and style transfer. In this work, we adopt the deep ResNet architecture to learn simulated-to-real image transformation.

## 3. SAR SIMULATION

Our SAR simulator for generating synthetic SAR images is based on a point-scattering model [14]. The far-field radar return for an ideal point scatterer at location vector  $\mathbf{w}$  is given by the equation:

$$s(f) = \gamma e^{\frac{4\pi i f}{c} \mathbf{r} \cdot \mathbf{w}} \quad (5)$$

where  $f$  is the RF frequency,  $\mathbf{r}$  is the unit vector pointing towards the radar, and  $\gamma$  is a complex radar cross section value determining the strength of the return and a relative phase. The full radar range profile can be generated by sweeping  $f$  across the radars bandwidth followed by an inverse Fourier transform. In practice, this is done in the time domain directly with interpolation for performance reasons.

Our point-scattering model works by loading in a three-dimensional target model, then coating it with a dense grid of point scatterers. For each pulse, the software then computes the return from each point scatterer and integrates the returns up to form the full target return. Once the range profiles are computed, they are fed into a backprojection algorithm that forms a two-dimensional SAR image.

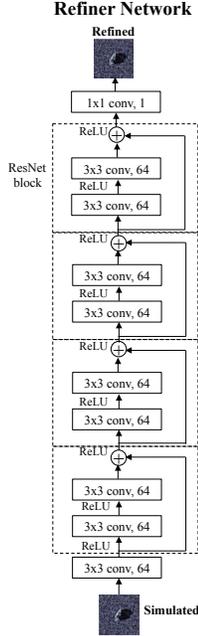


Fig. 3: Deep residual architecture of the refiner network

#### 4. SIMULATED SAR IMAGE REFINEMENT VIA DEEP RESIDUAL LEARNING

Figure 2 depicts our overall refinement network architecture. We denote simulated input image  $\mathbf{z}$  and its real counterpart  $\mathbf{x}$ . For a given class of objects (*e.g.*, a vehicle type), a simulated image  $\mathbf{z}$  is obtained by the SAR simulator described in Section 3. This simulated image is paired with a real training image in the same class. A training example is such a simulated-real SAR image pair. A refined image  $\hat{\mathbf{x}}$  is generated via  $\hat{\mathbf{x}} \leftarrow R(\mathbf{z})$ . An  $i$ th simulated or real image is designated as  $\mathbf{z}^{(i)}$  or  $\mathbf{x}^{(i)}$ , respectively. When applying  $n$  training examples  $\{(\mathbf{z}^{(1)}, \mathbf{x}^{(1)}), \dots, (\mathbf{z}^{(n)}, \mathbf{x}^{(n)})\}$ ,  $R$  outputs corresponding refined images  $\{\hat{\mathbf{x}}^{(1)}, \dots, \hat{\mathbf{x}}^{(n)}\}$ .

Figure 3 provides architectural details of the refiner  $R$ . The network aims to refine input simulated images to improve real SAR ATR accuracy in the sense that the SAR ATR CNN trained on these simulated images under refinements, or real images augmented with these simulated images, can achieve improved accuracy. To achieve this purpose, we adopt a deep residual network that augments the features of target objects by introducing more low-level details through residual learning.

An input simulated image is convolved with  $3 \times 3$  filters producing 64 feature maps. The output is processed through four ResNet blocks each consisting of two convolutional layers containing 64 feature maps. The output of the last ResNet block convolved with  $1 \times 1$  convolutional layer yields the refined image. The refiner network is optimized by gradient descent on a loss function  $L_r$  described below.

---

#### Algorithm 1 Mini-batch training of refiner network $R$

---

- 1: **input:** Sets of real SAR images  $\mathbf{x}^{(i)} \in \mathcal{X}$  and simulated images  $\mathbf{z}^{(i)} \in \mathcal{Z}$ , batch size  $n$ , number of refiner step  $K_r$ , and Adam hyperparameters  $\alpha_a, \beta_1, \beta_2$ .
- 2: **output:** Refiner model  $R(\mathbf{z}; \theta_r)$
- 3: **for**  $k = 1, \dots, K_r$
- 4:   Sample  $n$  mini-batch pairs of simulated and real SAR images  $\{(\mathbf{z}^{(1)}, \mathbf{x}^{(1)}), \dots, (\mathbf{z}^{(n)}, \mathbf{x}^{(n)})\}$ .
- 5:   Update the refiner by taking gradient steps on mini-batch loss in Eq. (6):

$$\theta_r \leftarrow \text{Adam} \left( \nabla_{\theta_r} \frac{1}{n} \sum_{i=1}^n L_r^{(i)}, \theta_r, \alpha_a, \beta_1, \beta_2 \right)$$

- 6: **until for**
- 

The loss function computes the pixel-wise error between a real image and a corresponding refined image as

$$L_r^{(i)} = \|\Psi(\mathbf{x}^{(i)}) - \Psi(R(\mathbf{z}^{(i)}))\|_p \quad (6)$$

where  $\|\cdot\|_p$  is the  $\ell_1$ - or  $\ell_2$ -norm and  $\Psi(\cdot)$  is a feature transformation function. We use the transformation function as  $\Psi(\mathbf{x}) = \log((\mathbf{x} - 0.5) + \epsilon)$  where  $\epsilon$  is a small positive number. Such function remedies high spatial frequencies in SAR images by penalizing pixel differences in the area of the target more heavily than the background or the radar shadow. Similar to image super-resolution [12, 13], SAR image refinement involves one-to-one mapping between simulated and real images. Minimizing the pixel-wise error encourages visual similarity between images. For mini-batch training, loss terms for each samples in a mini-batch are combined and normalized for a single gradient update. We present training algorithm for our model in Algorithm 1.

## 5. EXPERIMENTS

### 5.1. Dataset

We evaluate the proposed SAR image refinement model on Moving and Stationary Target Acquisition and Recognition Radar (MSTAR) dataset<sup>1</sup> using simulated SAR images as described in Section 3. The MSTAR is a standard dataset used in the SAR ATR community. The dataset consists of  $128 \times 128$  SAR imagery of 10 vehicle types. The training set consists of 3671 images collected at  $17^\circ$  depression angle and test set contains 3203 images at  $15^\circ$  depression angle. The training set is split into 80% training and 20% validation.

### 5.2. Experimentation Details

For SAR ATR network, we utilize a CNN-based SAR ATR system with various hyperparameters shown in Table 1. This architecture is adopted from the network used by Wilmanski *et al.*[1]. We train the network using real, simulated, or refined

<sup>1</sup><https://www.sdms.afrl.af.mil/index.php?collection=mstar>

**Table 1:** Network architecture used for SAR ATR

Layer	Output size	Comment
Input	48x48x1	
Convolutional	48x48x9	ReLU
Maxpooling	24x24x9	
Convolutional	24x24x18	ReLU
Maxpooling	12x12x18	
Convolutional	12x12x36	ReLU
Maxpooling	6x6x36	
Convolutional	6x6x60	ReLU
Flatten	2160	
Fully Connected	60	ReLU
Dropout	60	
Fully Connected	10	Softmax

**Table 2:** SAR ATR performance results

Train	Test	Accuracy (%)
Real	Real	94.46
Simulated	Simulated	81.70
Simulated	Real	19.50
Refined	Real	<b>79.95</b>

images collected at  $17^\circ$  depression angle. Then the trained networks are tested on real data at  $15^\circ$  depression angle.

We preprocess by remapping the SAR images to  $[-0.5, 0.5]$  after histogram clipping. Then the images are cropped into  $48 \times 48$  size while keeping the target at the center. Our models are trained with mini-batch stochastic optimization with a mini-batch size of 64. We use the ADAM optimizer [15] with the hyperparameters  $\alpha_a = 0.001$ ,  $\beta_1 = 0.9$ , and  $\beta_2 = 0.999$ . We use refiner steps  $K_r = 50,000$ . In this work, we adopt the ResNet-based framework from Shrivastava *et al.* [16] to implement the refiner.

### 5.3. ATR Performance Comparison

We first validate the ATR model in Table 1 by training and testing with real, simulated, or refined data at different depression angles. Notice in Table 2, the accuracies for the models trained and tested with data obtained from the same manner (real or simulated) are high. However, as we have motivated earlier, when the model is trained with simulated data, it performs poorly on real test data. Suppose now that we refine the simulated images by our deep residual refiner network, train the ATR model using the resulting refined images, and test with the real data. As shown in Table 2, we observe a significant 60.45% absolute percentage improvement when training on the refined images compared to training on simulated images. We use  $\ell_2$ -norm for refinement as it consistently yields better performance than  $\ell_1$ -norm. Note that all reported results are the averaged accuracies over 8 trials.

Next, we test the effectiveness of our proposed model with limited real training data. To simplify the presentation, we consider here only one scenario that uses 10% of real data (*i.e.*, 367 real images as opposed to 3671 real images) in

**Table 3:** SAR ATR performance results using 10% of real training data

Train	Test	Accuracy (%)
Real	Real	71.37
Simulated	Real	19.50
Refined $\ell_1$	Real	44.88
Refined $\ell_2$	Real	55.05
Refined+Real	Real	<b>74.68</b>

training the refiner. We evaluate using both  $\ell_1$ - and  $\ell_2$ -norm losses. We then use the refined simulated images to train the SAR ATR CNN. Table 3 presents a summary that compares the ATR performance when only 10% of real training data is used. When 10% of real training data is used and tested on real test data, the classification accuracy is 71.37%. We refine simulated training data using our refiner network trained on 10% of real training images. Training with resulting refined images achieves 25.38% (note 44.88 vs. 19.50) improvement for  $\ell_1$ -norm and 33.55% (note 55.05 vs. 19.50) for  $\ell_2$ -norm compared to training with simulated. But the achieved performance 55.05% is still much lower compared to training only with real data, which achieves 71.37%.

When the 10% of real data is augmented with 3671 refined simulated images (generated by the refiner trained with 10% of real data), we observe a 3.31% (note 74.68 vs. 71.37) absolute percentage improvement. This result suggests that simulated images refined by our deep residual refiner are of sufficiently high quality that they can help mitigate the overfitting problem of the SAR ATR CNN when only a small number of real images is available for its training.

## 6. CONCLUSION

In this paper, we improve the utility of simulated SAR images for training SAR ATR system. We learn a refiner function from simulated to real SAR images through deep residual training and use the function to refine simulated SAR images. Based on MSTAR dataset, we demonstrate that SAR ATR system trained on refined images can yield 60.45% absolute percentage improvement compared to the system trained on simulated images. We also demonstrate that a combination of refined and real images gives higher accuracy than using only real images when real training data is limited. It is known in the literature that generative adversarial nets (GANs) [17] can be an effective way of generating synthetic images and image refinements [12, 18]. As a future work, it is of interest to compare between ResNet-based approach of this paper and GAN-based approaches and their evaluations in larger dataset.

## 7. REFERENCES

- [1] M. Wilmanski, C. Kreucher, and J. Lauer, "Modern approaches in deep learning for SAR ATR," *SPIE*, 2016.
- [2] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional neural network with data augmentation for sar target recognition," *IEEE Geoscience and Remote Sensing Letters*, 2016.
- [3] S. Chen and H. Wang, "SAR target recognition based on deep learning," *International Conference on Data Science and Advanced Analytics (DSAA)*, 2014.
- [4] D. Malmgren-Hansen, A. Kusk, J. Dall, A. Nielsen, R. Engholm, and H. Skriver, "Improving SAR automatic target recognition models with transfer learning from simulated data," *IEEE Geoscience and Remote Sensing Letters*, 2017.
- [5] N. Odegaard, A. O. Knapskog, C. Cochin, and J. C. Louvigne, "Classification of ships using real and simulated data in a convolutional neural network," *IEEE Radar Conference (Radar-Conf)*, 2016.
- [6] J. Mannon and D. Sakrison, "The effects of a visual fidelity criterion of the encoding of images," *IEEE Transactions on Information Theory*, 1974.
- [7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CVPR*, 2016.
- [8] T. Ross, S. Worrell, V. Velten, J. Mossing, and M. Bryant, "Standard SAR ATR evaluation experiments using the MSTAR public release dataset," *Proc. SPIE*, 1998.
- [9] T. Marzetta, "EM algorithm for estimating the parameters of a multivariate complex rician density for polarimetric SAR," *ICASSP*, 1995.
- [10] M. Preiss, D. Gray, and N. Stacy, "Detecting scene changes using synthetic aperture radar interferometry," *IEEE Transactions on Geoscience and Remote Sensing*, 2006.
- [11] G. Moser, J. Zerubia, and S. Serpico, "Sar amplitude probability density function estimation based on a generalized gaussian model," *IEEE Transactions on Image Processing*, 2006.
- [12] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al., "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint arXiv:1609.04802*, 2016.
- [13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711.
- [14] J. Holtzman, V. Frost, J. Abbott, and V. Kaupp, "Radar image simulation," *IEEE Transactions on Geoscience Electronics*, 1978.
- [15] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [16] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," *CVPR*, 2017.
- [17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *NIPS*, 2014.
- [18] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.