A SAR Dataset for ATR Development: the Synthetic and Measured Paired Labeled Experiment (SAMPLE)

Benjamin Lewis[a], Theresa Scarnati[a], Elizabeth Sudkamp[a], John Nehrbass[b], Stephen Rosencrantz[b], and Edmund Zelnio[a]

[a]Air Force Research Laboratory, Sensors Directorate, WPAFB, OH
[b]Wright State University, Dayton, OH

ABSTRACT

The publicly-available Moving and Stationary Target Acquisition and Recognition (MSTAR) synthetic aperture radar (SAR) dataset has been an valuable tool in the development of SAR automatic target recognition (ATR) algorithms over the past two decades, leading to the achievement of excellent target classification results. However, because of the large number of possible sensor parameters, target configurations and environmental conditions, the SAR operating condition (OC) space is vast. This leads to the impossible task of collecting sufficient measured data to cover the entire OC space. Thus, synthetic data must be generated to augment measured datasets. The study of synthetic data fidelity with respect to classification tasks is a non-trivial task. To that end, we introduce the Synthetic and Measured Paired and Labeled Experiment (SAMPLE) dataset, which consists of SAR imagery from the MSTAR dataset and well-matched synthetic data. By matching target configurations and sensor parameters among the measured and synthetic data, the SAMPLE dataset is ideal for investigating the differences between measured and synthetic SAR imagery. In addition to the dataset, we propose four experimental designs challenging researchers to investigate the best ways to classify targets in measured SAR imagery given synthetic SAR training imagery.

Keywords: MSTAR, Synthetic Data, Synthetic Aperture Radar (SAR), Automatic Target Recognition (ATR), Target Classification

1. INTRODUCTION

Synthetic aperture radar (SAR) is a sensor of great practical use for both military and civilian applications. Unlike electro-optical (EO) sensors, this active sensing modality is able to perform in all illuminations and most weather types. While SAR imagery resolution is not as fine as that of EO sensors, the resolution is dependent on the radar bandwidth and the imaging platform's synthetic aperture[1] and independent of the imaging platform standoff range. Thus, a SAR imaging system is able to provide images under most environmental conditions with consistent resolution. Though the SAR imaging system is well understood, limitations in collecting a sufficient amount of SAR data occur due to three main factors: (i) sensor parameters, (ii) target configurations and (iii) environmental conditions. Collectively, we refer to these factors as the SAR operating condition (OC) space.

SAR imaging systems can operate at various frequency bands over many different flight paths. Thus, the resolution of SAR images collected in the same location, but formed from data collected under different sensor parameters will vary. For example, coarse resolution could lead to a large object being represented by only a handful of pixels, while a more fine resolution would allow the same object to be comprised of a few dozen pixels. Beyond resolution discrepancies, the reflectivity of materials at the frequency bands typically used by radar is not intuitive to most humans. For this reason, SAR images are often referred to as "non-literal". Specular materials that are not visible in EO images can have large reflections in radar images. The positioning of the radar relative to a given target also significantly affects the amount of radio energy returned to the sensor, causing the appearance of a given target to significantly change depending on the angle from which it is viewed. Additionally, similar targets in different configurations imaged at the same locations will produce different radar

Contact information: {benjamin.lewis.13, theresa.scarnati.1, elizabeth.sudkamp.1, john.nehrbass.1.ctr, stephen.rosencrantz.1.ctr, edmund.zelnio}@us.af.mil

returns. Variations between images containing identical targets are also caused by environmental factors such as ground vegetation and soil moisture content. Clearly, because of the large number of OCs that affect the appearance of a SAR image, it is impossible to capture a sufficient quantity of measured images to describe the entirety of the SAR OC space.

The MSTAR dataset[2] is one of the most comprehensive measured SAR datasets available to the research community. It consists of a collection of one-foot resolution SAR images collected by the Air Force Research Laboratory, Sandia National Laboratory, and the Defense Advanced Research Projects Agency (DARPA) during the latter half of the 1990s. This dataset has been beneficial for radar researchers who are interested in automatic target recognition (ATR) tasks. Methods such as template-based algorithms,[3,4] transfer learning,[5] and convolutional neural networks (CNNs)[6,7] have been used to identify targets in the SAR imagery. Most of these algorithms use data from the MSTAR dataset for both training and evaluation purposes. As such, the training and evaluation data share OC characteristics such as background, radar operating parameters, weather, and target articulation. This, in turn, can result in the ATR algorithms overfitting the MSTAR OC space. Moving forward, we would like to be able to classify targets from a larger OC space, which requires more data. In particular, CNNs require a large amount of varied training data to achieve good generalization. Unfortunately, as discussed before, there is an insufficient amount of measured data to span the entire OC space or build large enough training sets. Thus, we must augment measured SAR datasets with computer generated, i.e., synthetic, data.

Generating SAR data via electromagnetic computational tools is a fairly mature technology. Several commercial or open-source projects, including RaySAR,[8] CohRaS,[9] SARsim,[10] and SARViz[11] have been developed and provide synthetic SAR images through a variety of methods, including ray bouncing and scattering center prediction. These tools have been used throughout the literature to generate datasets that serve as a proxy for the MSTAR dataset.[5,12] However, these datasets are likely not based on computer aided design (CAD) models that were well-truthed to MSTAR vehicles. For example, in Figure 3 of Ref 5, it is apparent that the model is very different from the photographed vehicle. These errors will propagate to the SAR images formed from that model, thus leading to discrepancies that affect the quality of algorithms developed with the synthetic data. Specifically, the articulation of individual parts on a target in a CAD model has a significant effect on the final appearance of a SAR image. Due to the specular nature of many surfaces at radar wavelengths, a change that would be nearly imperceptible in an EO image may create a large local difference in SAR imagery. Canonical shapes such as corner reflectors (trihedrals), dihedrals, flat plates, and cylinders have signatures that behave differently in the radar domain than in visible light. An analytical discussion of how the geometry of a target influences its appearance in a SAR image can be found in Ref 13.

Few papers that reference generated vehicular SAR datasets provide information about the quality of the synthetic data with respect to collected measured data. However, in Ref 12, the authors test the ability of a CNN to classify measured data given synthetic training data. They report an accuracy of approximately 20% on MSTAR imagery when the network is trained using synthetic imagery. This performance is a far cry from the 99% accuracy reported with the networks that are solely trained and tested using MSTAR data.[6,7] While additional accuracy may be gained through use of a more advanced CNN, more care could be taken to accurately configure models and match known OCs.

Much research is necessary to help bridge this gap between measured and synthetic SAR imagery. Several papers have investigated transforming synthetic images such that they appear more like measured SAR imagery.[5,12,14] This line of work is predicated on the idea that the main detriments to performance are the assumptions and computational shortcuts used when simulating electromagnetic data. Results in this line of inquiry have been promising, but such work can only benefit further from a dataset in which the CAD models used to generate imagery have been rigorously compared to the measured targets. In the spirit of enabling and accelerating this research, we present the Synthetic and Measured Paired Labeled Experiment (SAMPLE) dataset, which is based on the MSTAR dataset. Creation of this dataset involved careful truthing of CAD models to the configurations of the vehicles measured during the MSTAR collect. In this way, we have created a dataset in which the variance due to CAD model articulation has been minimized. Thus, it is positioned to be a helpful synthetic augmentation to the publicly-available MSTAR dataset for further ATR development. Alongside this paper, we are also releasing a portion of the SAMPLE dataset for public use. Using the publicly released data, the second half of this paper defines a challenge problem designed to spur innovation in solving the synthetic/measured SAR data gap.

The rest of this paper is organized as follows. In Section 2 we discuss the careful generation of synthetic data that are matched to the measured MSTAR collect. The collection of matched synthetic and measured data is coined the SAMPLE database. We then analyze the fidelity of our synthetic data in Section 3. Section 4 discusses four potential experimental designs using the SAMPLE database. Each experimental design describes possible metrics of performance and possible implementations. We conclude in Section 5.

2. SAMPLE DATASET GENERATION

In this section, we discuss the creation of the SAMPLE dataset. This dataset is composed of measured SAR images from the MSTAR dataset and simulated SAR images based on carefully truthed CAD models. The simulated models are based on metadata that was recorded during the MSTAR Program, enabling us to position the CAD models in the same way that the measured vehicles were placed during the collect. Thus, we are sufficiently able to match target and radar OCs.

2.1 Model Truthing

When generating radar data, the complexity and positioning of the CAD models is extremely important. The behavior of electromagnetic waves transmitted by radars at microwave frequencies is very different from that of the visible spectrum light measured by EO imaging systems. Thus, intuition about visible light is not a good guide when envisioning interactions of radar signals with scenes or targets. In short, even small details on a target must receive proper attention. The typical wavelength of a transmitted radar pulse is on the order of centimeters, which creates an approximate lower bound on the size of the vehicle parts that must be examined.

While the MSTAR dataset contains SAR imagery of a variety of vehicles at various elevations, locations, and in different years, we decided to scope our effort to a subset of vehicles according to the data and CAD models that were available. The CAD models, which were another product of the MSTAR Program, provided a reasonable starting point. Ostensibly, these models had been truthed to the vehicles in the manner which we describe here, but a close inspection of the models showed that this was not always the case. Many models lacked crucial components. Additionally, models of some targets did not exist. Thus, vehicles with highly unsatisfactory CAD models were excluded from consideration in this dataset. We further chose to limit our effort to targets with available measured data between 15° and 17° in elevation. This gives the dataset a degree of homogeneity, as the similar collection geometry for every target implies that most of the images will have many properties in common. We also chose to scope our truthing efforts to targets imaged during the same flight. These limitations left us with the ten vehicles and corresponding serial numbers listed in Table 1.

Vehicle	2S1	BMP2	BTR70	M1	M2	M35	M548	M60	T72	ZSU23-4
Serial #	B01	9563	C71	0AP00N	MV02GX	T839	C245HAB	3336	812	D08

Table 1: A list of the vehicles and serial numbers included in the SAMPLE Dataset

Further, the CAD models were in an outdated file format and initially could only be read using a proprietary, outdated, difficult-to-use viewer. Thus, during the course of this work, the models were updated to the AFacet file format.[15] This format supports separately articulable assemblies which can be positioned programatically or with an HTML viewer.

During the MSTAR data collect, the MSTAR team collected a rich set of traditional camera (EO) images of the various MSTAR targets. This imagery served as the baseline truth to which we compared our CAD models. In general, EO images were taken of each target at intervals of 45° in azimuth at elevations of 0°, 15°, 30°, and 45°. These images were taken from the ground, rather than an aircraft; a few examples are shown in Figures 1a and 1c. Close-up shots of each target were taken as necessary to show details that were not clear in wider shots. In addition to imagery, notes about the position of various parts of each target were available and used as necessary during the truthing process. These notes and images provided sufficient information for us to be

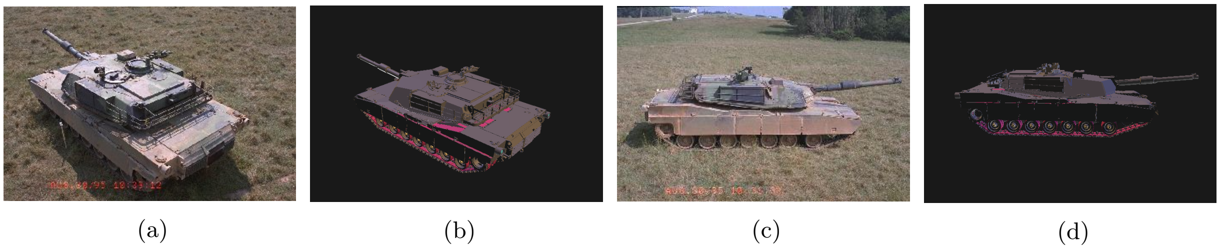

(a) (b) (c) (d)

Figure 1: A comparison between EO images taken of an M1 during the MSTAR data collect (a, c) and truthed CAD models of the same vehicle (b, d) from two viewpoints. Note that even small details, such as the position of the gun, hatches, and the cable on the side of the vehicle agree well between the two images. Coloration of the CAD models is provided to identify part assemblies and does not denote any specific property that affects the electromagnetic simulation.

confident in our efforts to match the models to the ground truth. Displayed in Figure 1 is an example of M1 tank images along with the final, corresponding, truthed CAD model at two different look angles.

An iterative process was used to adjust the CAD models to a satisfactory degree of agreement with the set of EO images. During each iteration, all discrepancies between the CAD model and corresponding EO image were corrected in the CAD model. This process is inherently manual and requires a human-in-the-loop. To minimize errors, the process was repeated a number of times for each target until agreement was reached.

2.2 Radar Simulation

Prediction of the electromagnetic signatures of a target was performed by asymptotic ray-tracing techniques. This is a large, computationally intense data generation problem and requires significant computer resources. We used the Generic Parallel System[16] to parallelize the data generation process to accomplish the simulation in a reasonable amount of time.

Material properties for each surface on the target were assigned based on notes from the MSTAR collect as well as intelligent reasoning about the composition of parts. Each material, such as glass, paint, metal, or rubber, was assigned appropriate electromagnetic property values. During simulation, the electromagnetic solver considered these values when computing the radar return from a target.

For simulation, each target was placed on top of a ground plane defined by a stochastic rough surface, which provides a background for each image. By so doing, each synthetic SAR image exhibits a background with speckle. Matching of the synthetic speckle to the background clutter present in the measured MSTAR images will be the study of future research efforts.

Radar data was generated in the form of a data dome. Radar returns were computed around each target from far-field points of view spaced at 0.04° intervals in both azimuth and elevation dimensions. The data dome around each target spanned all 360° in azimuth between 14° and 18° in elevation.

2.3 Image Formation

For each measured image in our subset of the MSTAR dataset, we created a matched synthetic SAR image. MSTAR image files contain metadata about the conditions under which each image was collected. Information such as target type, radar bandwidth, collection azimuth and elevation angle, target position, range resolution and cross-range resolution are stored in the metadata along with a number of other parameters. These parameters are sufficient to reconstruct the geometry and collection parameters of the image. This relevant metadata was extracted from each MSTAR file and used to create a synthetic SAR image under the same image formation parameters. Many of the relevant parameters, which are consistent across the dataset, are displayed in Table 2.

From the metadata in the MSTAR file, the synthetic phase history was extracted from the data dome described in Subsection 2.2. We assumed a constant elevation across all data points and constructed an azimuth aperture centered on the azimuth angle detailed in the MSTAR metadata. Note that this azimuth angle was defined relative to the heading of each vehicle, such that the front of a target at zero degrees azimuth always

Parameter	**Value**	**Parameter**	**Value**
Range resolution	0.30 m	Bandwidth	591 MHz
Range pixel spacing	0.20 m	Center frequency	9.6 GHz
Range extent	25.8 m	Image size	128 × 128
Cross-range resolution	0.30 m	Polarization	HH
Cross-range pixel spacing	0.20 m	Elevations	14°-18°
Cross-range extent	25.8 m	Taylor weighting	-35 dB

Table 2: Typical parameter values from the MSTAR dataset. These parameters were used when forming synthetic SAR images in order to provide good agreement with the MSTAR images. These parameters are available in files from the publicly released portion of the MSTAR dataset.[2]

points to the right of the image. Based on the central azimuth angle, range and cross-range resolutions, elevation angle, and bandwidth, we computed the locations at which an airborne imaging platform would need to collect radar data. In other words, these computed locations are the points along a flight path at which the radar would transmit pulses in order to image the scene. The radar pulses at these locations were extracted from the data dome and combined to form a phase history. In cases where the exact computed pulse location was not available in the data dome (e.g., 5.2063° azimuth angle and 15.23° elevation angle), data was simulated using two-dimensional linear (bi-linear) interpolation from the nearest defined points.

We applied a Taylor window with -35 dB sidelobe suppression to the phase history data. As phase history windowing decreases the resolution of an image, the synthetic aperture was broadened to counteract this windowing. This windowing function was applied to the synthetic data in accordance with the MSTAR metadata. Complex images of size 128 × 128 pixels were then formed using the polar format algorithm, a commonly used, Fast Fourier Transform (FFT) based, SAR image formation technique.[17] This is the same image formation technique that was used to form the MSTAR images.

We aligned the measured and synthetic images on a per-pair basis. In each synthetic image, the target is centered at the origin of the imaging plane. Thus, alignment was performed by circularly shifting each measured image to align with the synthetic image. The alignment process was performed using a quantized version of the magnitude of each image, where a mask was placed around target regions. Target masks were created using the Variance-Based Joint Sparsity algorithm,[18,19] which is used to select pixels that correspond to strong, target-like returns. Each mask was filtered using a RANSAC-based[20] algorithm to reject outlier points. Within the convex hull of this mask, each image was quantized into seven levels. The normalized cross-correlation between a given quantized MSTAR image and its synthetic counterpart was used to compute the number of pixels in each direction the MSTAR image must be moved in order to align with the synthetic image.

Finally, the data products were saved to disk. Each complex image was saved as a MATLAB .mat file. Basic information about the collection geometry, such as azimuth and elevation, were saved to the file as well. Table 4 provides a description of metadata included within each .mat file. The files were named using the template <vehicle_type> <real_or_synthetic> <center_azimuth> <elevation> <serial_number>. This inclusion of relevant metadata in the filename enables quick indexing of files without reading the contents of the file. It also enables real and synthetic files to be matched based on filename alone.

Figure 2 shows a randomly-chosen set of images from the completed SAMPLE dataset. Here, measured MSTAR images are displayed in the top row, and the matching synthetic images are shown on the bottom. Recall that we made no effort to match the ground planes and backgrounds of the MSTAR images; for this reason, the background of the synthetic images are somewhat darker than in the measured images.

3. SAMPLE DATASET FIDELITY

Even with the best efforts to match the synthetic data to the measured data, there are still inevitably inherent differences between the two domains due to computational precision, CAD model fidelity, and the inescapable errors of human judgment. However, simulated data based on less well-matched models will not be any more accurate and can introduce variability in the SAR images that is entirely controllable. One of the main goals

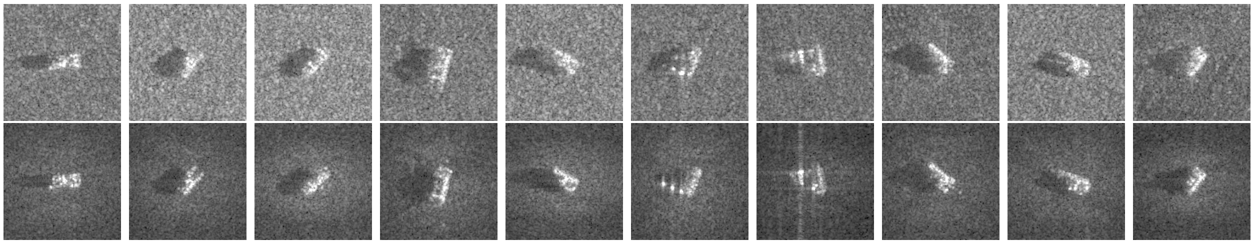


Figure 2: An example of one image of each vehicle in the SAMPLE dataset. Measured MSTAR images are on the top row, and the corresponding synthetic images are on the bottom row. The order of the vehicles from left to light is the same as the ordering in Table 1. We see that details such as shadows, orientation, and relative return magnitudes are in good agreement.

of the SAMPLE dataset is to minimize controllable variance such that profitable research can be conducted regarding the underlying differences between the measured and synthetic data domains.

A mismatch between the measured configuration of the vehicles and the CAD models may partly explain the difficulty in obtaining satisfactory results when training ATR algorithms only on synthetic data. While this claim is not a simple one to validate, one method to do so would be to compare the measured data classification accuracy of a neural network trained on non-matched generated data. Unfortunately, we do not have access to the non-matched datasets used in comparable papers from the literature,[5, 12] and the computational cost of generating such data ourselves is non-trivial. Instead, we compare our dataset to results from papers that report such classification accuracy (e.g., neural networks trained on synthetic data and tested on measured data). To our knowledge, Cha *et al*[12] is the only paper which reports these results. While the main goal of that paper deals with making synthetic SAR data look more like measured SAR data, the authors present classification results for a network trained on synthetic data and tested on measured data; these results are of interest in the context of our paper. Their measured dataset is the MSTAR measured images, as are our images. Their synthetic SAR images appear to have been generated specifically for their experiment. It is unclear, but unlikely, that the models used in generating their data were truthed in the same way the SAMPLE dataset models were. Their dataset, like our SAMPLE dataset, also consists of ten vehicular targets, although it is unclear if these are the same vehicles as ours. In short, we attempted to replicate their experiment in as many way as we were able in order to provide insight into the fidelity of the SAMPLE dataset with respect to the MSTAR data and to show that the SAMPLE dataset matches MSTAR better than a generic synthetic dataset.

Layer	**Output Size**	**Nonlinearity**
Input	48×48×1	
Convolutional Max Pooling	48×48×9 24×24×9	ReLU
Convolutional Max Pooling	24×24×18 12×12×18	ReLU
Convolutional Max Pooling	12×12×36 6×6×36	ReLU
Convolutional	6×6×60	ReLU
Flatten	2160	
Fully Connected	60	ReLU
Dropout	60	
Fully Connected	10	Softmax

Table 3: Architecture of a convolutional neural network used to compare the differences of real and synthetic data. Architecture originally described in Cha *et al.*[12]

To ensure fair results, we duplicated the classification network described by Cha *et al.* The architecture of this network is reproduced here in Table 3 for reference. In order to match their experiment, we trained the network on all synthetic images collected at a 17° depression angle, then evaluated the network on measured data collected at a 15° depression angle. Before input, the images were cropped to a 48×48 patch around the center of the image. Pixel values were normalized to the range [-1, 1]. We used the Adam optimizer with hyperparameters $lr = 0.001$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ when training the network. Training ran for 100 epochs before performing evaluation on the measured data. Over the course of 10 runs, we obtained an average test accuracy of 27.8% using the SAMPLE dataset; for comparison, Cha *et al* achieved a reported 19.5% accuracy. While our performance is also not particularly impressive, it demonstrates that the SAMPLE dataset is, in some fashion, a better match to MSTAR than a dataset that was not truthed as carefully, such as that used by Cha *et al.* We claim that this increased performance is due to our efforts to match the CAD models to the MSTAR targets.

Note that the network described in Table 3 is not a particularly advanced classifier, and the lack of network complexity produces an artificially low accuracy for this test. With a state-of-the-art classifier such as DenseNet,[21] we are able to achieve a much better baseline performance on this test. Using PyTorch, we trained an off-the-shelf DenseNet with the Adam optimizer for 20 epochs. Over the course of ten runs, we achieved an average classification accuracy of 54.2%. We also performed an additional test in which Gaussian noise was added to the synthetic images during training. This noise was added to account for the environmental condition mismatch that is still present within the SAMPLE database. Doing so encourages network generalization and leads to higher classification accuracy in general. Note that we did not add such noise during our comparison to Cha *et al* because the authors made no mention of using this technique. When adding Gaussian noise to the images during DenseNet training, we obtained an even higher classification score of 77.7%, averaged over ten runs. These high accurate results speak not only to the power of complex networks, but also to the quality of the SAMPLE dataset.

4. CHALLENGE PROBLEM

Convolutional neural networks are tremendously successful at classifying objects in electro-optical images. However, as discussed in Section 1, with SAR data, off-the-shelf classifiers are insufficient because there are limited measured SAR data available and SAR images are not invariant to object manipulations. To overcome these issues, measured SAR datasets could potentially be augmented with synthetically generated data. As shown in Section 3, even when every care is taken to match the synthetic data to pre-existing measured data, synthetic data do not sufficiently model real world phenomena. This means that if we train off-the-shelf classifiers with synthetic data and test the algorithm on measured data (as would be the case in a operational situation), results are poor.

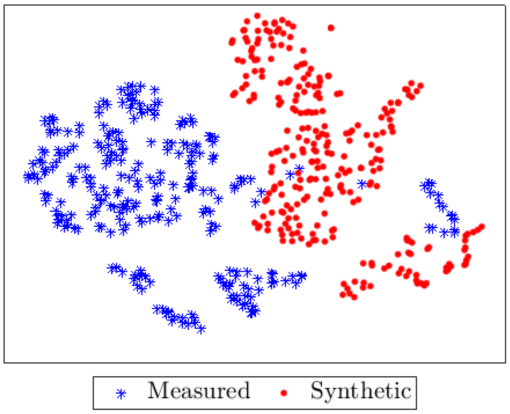


Figure 3: The t-SNE[22] two-dimensional representation of the SAMPLE dataset described in Section 2.

To demonstrate the disconnect between the well-matched measured and synthetic data described in Section 3, the data are projected onto the the two-dimensional manifold that best preserves the local structure of the high dimensional data using t-Distributed Stochastic Neighbor Embedding (t-SNE).[22] Here, the data input into t-SNE are consider the original SAR images, unwrapped to a large vector. Note that the t-SNE dimensionality reduction technique is not supplied data labels a-priori. In Figure 3 we plot the two-dimensional data resulting from applying t-SNE to our dataset and then overlay the measured/synthetic labels onto the results. Clearly, the measured and synthetic data are disjoint. In this problem, we challenge the research community to develop techniques for closing this measured/synthetic gap so that we can appropriately classify measured SAR data using synthetic SAR training data.

4.1 Publicly Released Data

With this challenge problem, we publicly release a subset of the SAMPLE dataset that is appropriate for running small scale, meaningful classification experiments. This dataset contains complex image data (in .mat format), and magnitude-only images (in .png format) for the ten targets mentioned in Table 1. The data are sampled in azimuth from 10° to 80° and in elevation from 14° to 17°. Additional metadata, described in Table 4, are included in each .mat file. Note that these metadata were used to form each complex image as described in Section 2.3. For each measured image, the corresponding matched synthetic image is provided. The number of samples for each target class within the available publicly released dataset is given in Table 5. To gain access to the publicly released data, please email the authors of this paper.

Variable	Meaning	Type	Units
`azimuth`	center azimuth angle	float	degrees
`complex_img`	complex image data	128×128 float	-
`elevation`	elevation angle	float	degrees
`polarization`	radar polarization	string	-
`range_resolution`	range resolution	float	meters
`target_name`	target name	string	-
`xrange_resolution`	cross-range resolution	float	meters

Table 4: Description of metadata included with each .mat file.

4.2 Experimental Setups

We now describe four potential experiments to be conducted under this challenge problem setup. Along with the publicly released data, researchers will be granted access to Python codes for the reproduction of each experiment via a Jupyter notebook. Within each experiment definition we also describe suggested metrics of performance so that researchers may be able to appropriately compare experimental results.

4.2.1 Classifying Measured Data from Synthetic Training Data

One of the main goals of this challenge problem is to be able to train a classification algorithm on synthetic data in a way that is useful for classifying measured data. In this experiment we will analyze two scenarios. First, in Experiment 4.1, we evaluate overall classification accuracy as the amount of measured data in the training set is reduced while simultaneously increasing the amount of synthetic data in the training set. Second, in Experiment 4.2, we explore how the classification algorithm performs as the number of classes without measured training data increases.

For each experiment, the withheld testing data are defined as the measured data from each class collected at a 17° elevation angle. No data (measured or synthetic) collected from a 17° elevation angle will be included in the training set.

Define the number of training samples for class j as

$$T_j = T_j^m + T_j^s, \quad j = 1, ..., 10, \tag{4.1}$$

where T_j^m and T_j^s respectively define the number of measured and synthetic training images from class j. The number of measured images in the training set for class j is defined as

$$T_j^m = k(N_j^m - S_j^m), \quad j = 1, ..., 10, \tag{4.2}$$

where N_j^m is the total number of available measured images in class j, S_j^m is the number of measured testing images withheld from class j, and $k \in [0,1]$ represents the fraction of images in the training set that are measured. Then, the number of synthetic images in the training set for class j is

$$T_j^s = (N_j^m - S_j^m) - T_j^m = (N_j^s - S_j^m) - T_j^m, \tag{4.3}$$

where N_j^s is the total number of synthetic images available for class j. Note that because we have supplied a matched dataset, $N_j^m = N_j^s$ for all $j = 1, ..., 10$. An example of the image distribution break down when $k = 0.5$ is displayed in Table 6. Notice that there are no synthetic data in the testing set and the sum of the number of training and testing images equals the total number of measured *or* synthetic images in the set. That is, for any $k \in [0,1]$, $S_j^s = 0$, and $T_j^m + T_j^s + S_j^m = N_j^m = N_j^s$ for all $j = 1, ..., 10$.

Experiment 4.1. *For this experiment, the amount of measured training data is decreased by changing the value of k in (4.2) simultaneously for all classes $j = 1, ..., 10$. Each time a measured image is removed from the training set of class j, its matched synthetic image is put in its place. Specifically, in (4.2) define*

$$k \equiv k_i = i\Delta, \quad i = 0, ..., 20, \tag{4.4}$$

with $\Delta = 0.05$. Then, we can evaluate the performance of a classification algorithm when trained with increasing amounts of synthetic data and tested on measured data. Two metrics of performance for this experiment include, but are not limited to, (i) the average probability of correct classification among all classes for a particular k, and (ii) the confusion matrices associated with each run of the experiment.

Experiment 4.1 is a useful investigation of how to effectively augment limited measured training data with synthetic data. Such investigations are of interest because, relative to the entire OC space of SAR, little measured data is available for training ATR algorithms. By systematically excluding measured data from the training set and replacing it with synthetic data in this way, much can be learned that can carry over to using synthetic data to augment the OCs necessary to enhance classification performance.

Experiment 4.2. *For this experiment, the classification algorithm is trained such that J classes contain no measured training data. That is, $k = 0$ in (4.2) for a subset of J classes. This subset of J classes is then trained on only synthetic data. The remaining $10 - J$ classes are trained on only measured data, i.e. $k = 1$ in (4.2). Here, $J = 1, 2, ..., 10$, and for each selection of J there are*

$$\binom{10}{J} = \frac{10!}{J!(10-J)!}$$

possible combinations of class subsets from which measured data are withheld from the training set. Thus the metrics of performance for this experiment include, but are not limited to, the average probability of correct classification over (i) all classes, (ii) the J classes trained on synthetic only (no measured) data, and (iii) the

Class	**Measured**	**Synthetic**	**Total**
2S1	177	177	354
BMP2	108	108	216
BTR70	96	96	192
M1	131	131	262
M2	129	129	258
M35	131	131	262
M548	129	129	258
M60	178	178	356
T72	110	110	220
ZSU23	177	177	354
Total	1366	1366	2732

Table 5: Distribution of publicly available SAMPLE data for each class.

$10 - J$ *classes trained on measured only (no synthetic) data. Confusion matrices may be useful for comparison among results.*

Experiment 4.2 is a practical investigation of how classification networks perform when no measured data is available for specific classes. This is an interesting problem because it requires innovation to successfully perform cross-domain transfer learning in a way that encourages the algorithm to know something about SAR imagery in general.

We now provide the result of conducting Experiment 4.1 and Experiment 4.2 on the publicly released data discussed in Section 4.1. The classification network used for these experiments is a convolutional neural network (CNN) with four convolutional layers and four fully connected layers. The details of the CNN are outlined in Table 7. The CNN is implemented in Pytorch,[23] where a cross entropy loss function is minimized via the Adam optimization algorithm with a learning rate equal to 0.001. We train the algorithm using batch sizes of 16 images for 60 epochs. During training, 15% of the training images were used for validation.

For each k_i in (4.4), Experiment 4.1 is conducted 100 independent times and the results are averaged together. Figure 4 displays the result of averaging the probability of correct classification among all 10 classes over the 100 independent runs of Experiment 4.1. Figure 5 displays the confusion matrices resulting from conducting one run of Experiment 4.1 with $k_0 = 0$ and $k_{20} = 1$. From these results, we see that as we augment the training set with additional synthetic data, classification accuracy significantly decreases. Specifically, when $k < 0.4$, the probability of correct classification falls below 0.9.

Class	j	N_j^m	N_j^s	S_j^m	S_j^s	T_j^m	T_j^s
2S1	1	177	177	59	0	59	59
BMP2	2	108	108	52	0	28	28
BTR70	3	96	96	51	0	23	22
M1	4	131	131	52	0	40	39
M2	5	129	129	54	0	38	37
M35	6	131	131	54	0	39	38
M548	7	129	129	54	0	38	37
M60	8	178	178	60	0	59	59
T72	9	110	110	53	0	29	28
ZSU23	10	177	177	59	0	59	59
Totals		1366	1366	548	0	412	406

Table 6: Data distribution for Experiment 1 with $k = 0.5$.

Layer	**Output Size**	**Comment**
Input	$64 \times 64 \times 1$	
Convolutional Max Pooling	$64 \times 64 \times 16$ $32 \times 32 \times 16$	ReLU
Convolutional Max Pooling	$32 \times 32 \times 32$ $16 \times 16 \times 32$	ReLU
Convolutional Max Pooling	$16 \times 16 \times 64$ $8 \times 8 \times 64$	ReLU
Convolutional Max Pooling	$8 \times 8 \times 128$ $4 \times 4 \times 128$	ReLU
Flatten	2048	
Fully Connected	1000	ReLU
Fully Connected	500	ReLU
Fully Connected	250	ReLU
Fully Connected	10	

Table 7: Classification network used in Experiment 4.1 and Experiment 4.2.

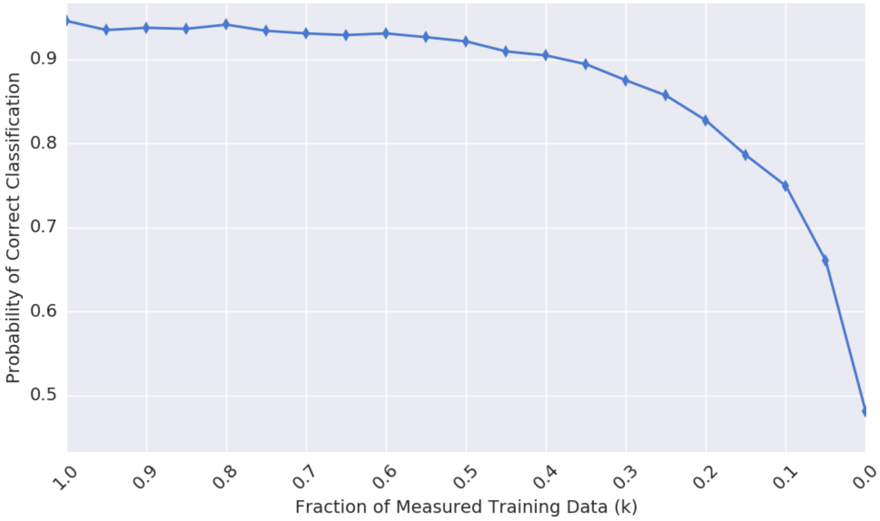
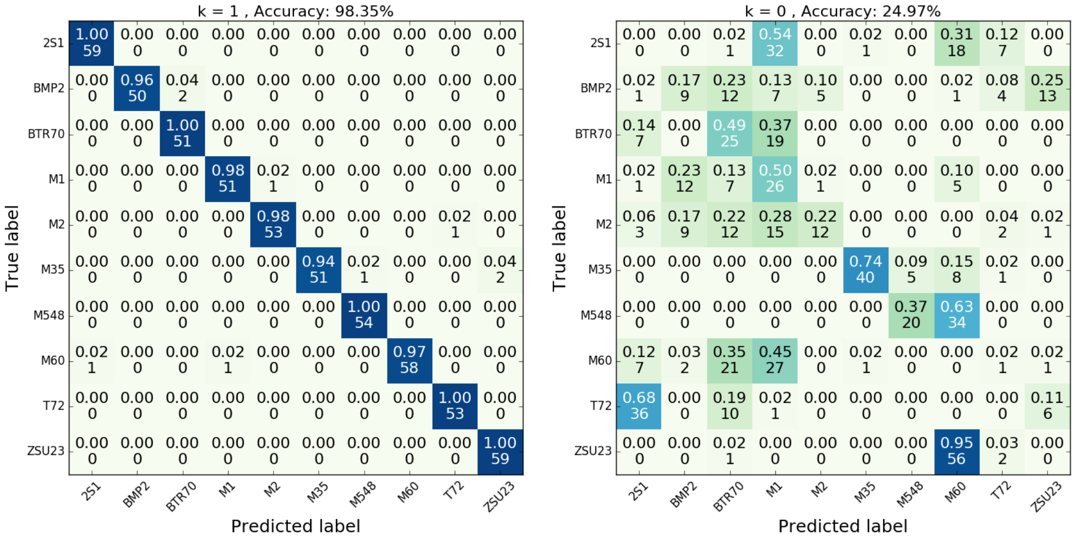


Figure 4: The result of conducting Experiment 4.1 with the publicly released data discussed in Section 4.1. Probability of correct classification was calculated for each value of k as the average probability of correct classification over 100 independent trials of Experiment 4.1.



k = 1 , Accuracy: 98.35%

	2S1	BMP2	BTR70	M1	M2	M35	M548	M60	T72	ZSU23
2S1	1.00 59	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
BMP2	0.00 0	0.96 50	0.04 2	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
BTR70	0.00 0	0.00 0	1.00 51	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
M1	0.00 0	0.00 0	0.00 0	0.98 51	0.02 1	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
M2	0.00 0	0.00 0	0.00 0	0.00 0	0.98 53	0.00 0	0.00 0	0.00 0	0.02 1	0.00 0
M35	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.94 51	0.02 1	0.00 0	0.00 0	0.04 2
M548	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	1.00 54	0.00 0	0.00 0	0.00 0
M60	0.02 1	0.00 0	0.00 0	0.02 1	0.00 0	0.00 0	0.00 0	0.97 58	0.00 0	0.00 0
T72	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	1.00 53	0.00 0
ZSU23	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	1.00 59

k = 0 , Accuracy: 24.97%

	2S1	BMP2	BTR70	M1	M2	M35	M548	M60	T72	ZSU23
2S1	0.00 0	0.00 0	0.02 1	0.54 32	0.00 0	0.02 1	0.00 0	0.31 18	0.12 7	0.00 0
BMP2	0.02 1	0.17 9	0.23 12	0.13 7	0.10 5	0.00 0	0.00 0	0.02 1	0.08 4	0.25 13
BTR70	0.14 7	0.00 0	0.49 25	0.37 19	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
M1	0.02 1	0.23 12	0.13 7	0.50 26	0.02 1	0.00 0	0.00 0	0.10 5	0.00 0	0.00 0
M2	0.06 3	0.17 9	0.22 12	0.28 15	0.22 12	0.00 0	0.00 0	0.00 0	0.04 2	0.02 1
M35	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.74 40	0.09 5	0.15 8	0.02 1	0.00 0
M548	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.37 20	0.63 34	0.00 0	0.00 0
M60	0.12 7	0.03 2	0.35 21	0.45 27	0.00 0	0.02 1	0.00 0	0.00 0	0.02 1	0.02 1
T72	0.68 36	0.00 0	0.19 10	0.02 1	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.11 6
ZSU23	0.00 0	0.00 0	0.02 1	0.00 0	0.00 0	0.00 0	0.00 0	0.95 56	0.03 2	0.00 0

Figure 5: Confusion matrices resulting from conducting Experiment 4.1 with the CNN described in Table 7 for (left) $k_{20} = 1$, and (right) $k_0 = 0$ in (4.4). Note that within each block of the confusion matrices we display the probability of correct classification above the number of images classified as such.

Figure 6 displays the t-SNE[22] two-dimensional representation of the final feature layer of the CNN described in Table 7 when given the measured testing data but trained with a measured/synthetic split defined by k in (4.2). We see that as k decreases, i.e., the training set consists of a larger fraction of synthetic data, class separability is not maintained. We conjecture this is because features produced by the network trained on synthetic data are not representative of the features present in measured data. This helps to explain why we are experiencing the poor classification accuracy for small k in (4.4). As researchers begin to approach this challenge problem with the given data, one possible goal is to maintain classification accuracy above 90% even when k is small.

Displayed in Figure 7 is the result of conducting Experiment 4.2 on the publicly released data described in Section 4.1. The probability of correct classification among the $10 - J$ classes trained on only measured data monotonically decreases as J increases. However, the probability of correct classification among the J classes

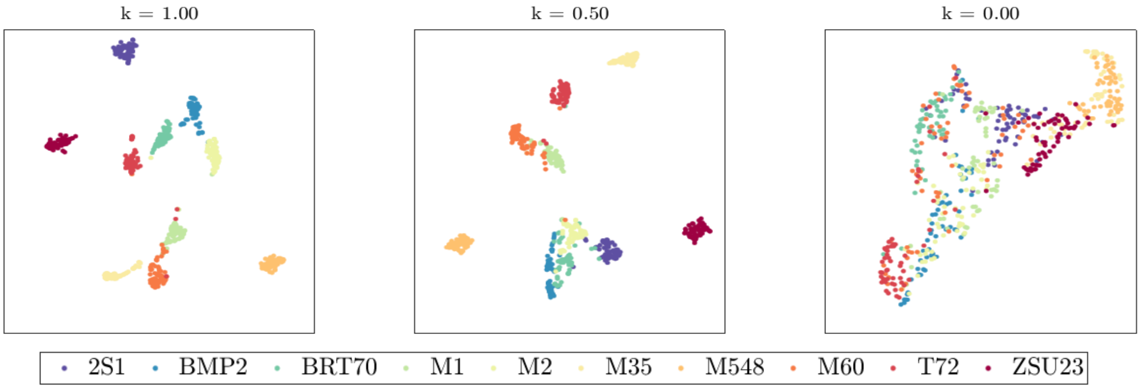


Figure 6: The two-dimensional t-SNE representation of the last feature layer of the CNN described in Table 7 given measured testing data after trained with (left) all measured data ($k = 1$), (middle) half measured and half synthetic data ($k = 0.5$), and (right) all synthetic data ($k = 0$).

trained on only synthetic data monotonically increases as J increases. Shown in Figure 7, these two curves intersect when $J = 7$. At this point, the average probability of correct classification among all classes begins to increase. Further investigations into this experiment should help to explain this phenomenology.

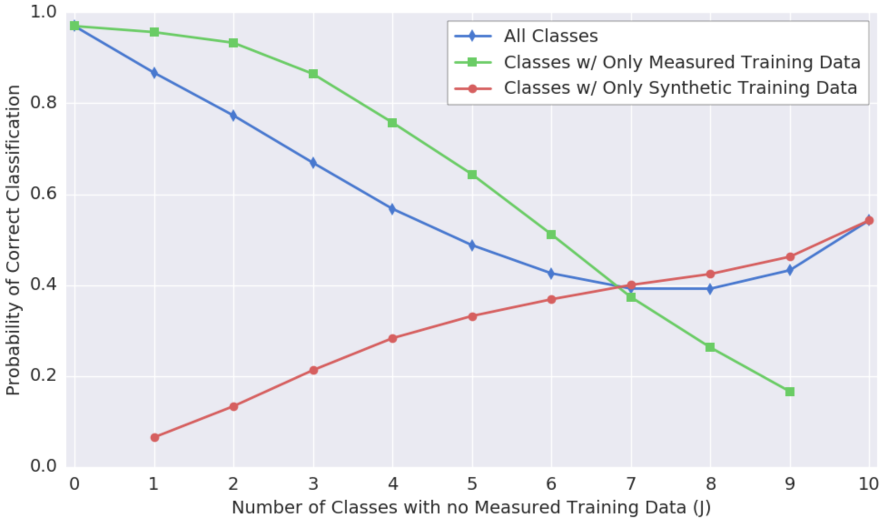


Figure 7: The result of conducting Experiment 4.2 on the publicly released data described in Section 4.1.

4.2.2 The Open Set Problem

As is, the classification network posed in Table 7 forces each testing image to be labeled as one of the ten classes considered. Most neural networks, especially CNNs, demonstrate this closed set behavior. However, recognition in the real world should avoid classifying unknown objects at test time; failure to do so results in false classification, which may have adverse or catastrophic effects. One possible solution is to adapt networks in a way to create new classes based on the rejected data. This problem is classically known as the open set problem.[24–29] The publicly released data described in Section 4.1 contains ten classes. By completely eliminating data from a subset of classes in the training set, the SAMPLE dataset may be easily used to research the open set problem. An example of one possible open set experiment is described in Experiment 4.3.

Experiment 4.3. *Given the data for 10 classes, remove all training data from a subset of J classes. These J classes are deemed out of library confusers. For the remaining $10 - J$ classes, choose k in (4.2) to determine the* *amount of measured and synthetic data included in the training set (as in Experiment 4.1). Measured data from all 10 classes at a 17° elevation angle are withheld from training for testing purposes.*

The result of conducting Experiment 4.3 with $J = 2$ and $k = 1$ using the CNN described in Table 7 is displayed in the confusion matrix found in Figure 8. Here, the two classes with data withheld from the training set are the 2S1 and the BMP2. We see 95.63% classification accuracy over the the 8 classes which were included in the training set. However, with no attempt to reject out of library confusers, the proposed CNN always incorrectly classifies the 2S1 and BMP2 testing data.

4.2.3 The Super-Class Problem

Another possible experiment is to classify targets within the data by not only their class labels but also by their super-class labels. Then, it is possible to fuse classification results based on this hierarchy of labels given to the training data.[30,31] Two possible sets of super-class labels for the publicly released data described in Section 4.1 are `{tank, truck}` and `{wheeled, tracked}`. These super-class labels for each of the 10 classes is described in Table 8.

We do not present results of the super-class experiment here, but note that classification codes can easily be changed by re-labeling targets in the SAMPLE dataset to the appropriate super-class label. All of the previous experiments may be performed on the super-class problem.

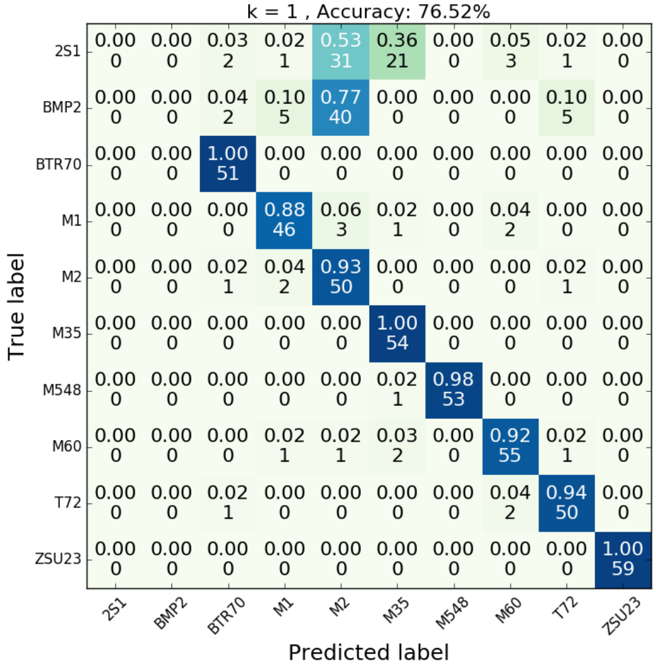


Figure 8: The result of conducting Experiment 4.3 while withholding training data from the 2S1 and BMP2 classes. Here $k = 1$ indicates that all training data came from the measured set. Note that within each block of the confusion matrix we display the probability of correct classification above the number of images classified as such.

Class	Type	Wheeled?
2S1	Tank	Tracked
BMP2	Tank	Tracked
BTR70	Tank	Wheeled
M1	Tank	Tracked
M2	Tank	Tracked
M35	Truck	Wheeled
M548	Truck	Tracked
M60	Tank	Tracked
T72	Tank	Tracked
ZSU23	Tank	Tracked

Table 8: Possible Super-Classes.

5. CONCLUSIONS

In this paper, we have presented the SAMPLE dataset, which includes simulated SAR images that are well-matched to measured images from the MSTAR dataset. The SAMPLE dataset benefits from careful truthing to MSTAR vehicle positions, accurate electromagnetic simulation, and particular attention to image formation parameters. Because of this, it is an ideal tool for investigating how to better use simulated radar data to improve SAR ATR classification. Internally, the SAMPLE dataset has been in use for numerous years, leading to several useful investigations into the problems posed here.[14, 32, 33]

Given the data described in Section 4.1, the number of possible experiment definitions is vast. In this paper, we have provided three detailed experimental designs, Experiments 4.1, 4.2 and 4.3, that not only define open problems throughout the machine learning/deep learning literature, but also are relevant to SAR ATR development. With each experiment, we provide possible metrics of performance. We briefly discuss an additional experiment possibility in Section 4.2.3, but leave the experimental design open-ended. Beyond the experiments presented here, we encourage other productive research using this dataset, including work in domain transfer, manifold joining, or background style transfer. To conclude, we challenge researchers to go out and do great things with the data!

ACKNOWLEDGMENTS

We gratefully acknowledge the funding of the Air Force Research Laboratory (AFRL) via the ATR Center summer internship program. We are also thankful for mentorship and instruction from numerous colleagues at AFRL and Wright State University.

Distribution A: Approved for public release. Distribution unlimited. PA Approval #88ABW-2019-1483.

REFERENCES

[1] Jakowatz, C. V., Wahl, D. E., Eichel, P. H., Ghiglia, D. C., and Thompson, P. A., [*Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach: A Signal Processing Approach*], Springer Science & Business Media (2012).

[2] Sandia National Laboratory, "MSTAR overview." https://www.sdms.afrl.af.mil/index.php?collection=mstar (1995). [Online; accessed 19-May-2017].

[3] Owirka, G. J., Verbout, S. M., and Novak, L. M., "Template-based SAR ATR performance using different image enhancement techniques," in [*SPIE 1999 Algorithms for Synthetic Aperture Radar*], 302–320 (1999).

[4] Ross, T., Worrell, S., Velten, V., Mossing, J., and Bryant, M., "Standard SAR ATR evaluation experiments using the mstar public release data set," in [*SPIE 1998 Algorithms for Synthetic Aperture Radar*], (1998).

[5] Malmgren-Hansen, D., Kusk, A., Dall, J., Nielsen, A. A., Engholm, R., and Skriver, H., "Improving SAR automatic target recognition models with transfer learning from simulated data," *IEEE Geoscience and remote sensing Letters* **14**(9), 1484 – 1488 (2017).

[6] Chen, S., Wang, H., Xu, F., and Jin, Y. Q., "Target classification using the deep convolutional networks for SAR images," *IEEE Transactions on Geoscience and Remote Sensing* (2016).

[7] Fox, M. R. and Narayanan, R. M., "Application and performance of convolutional neural networks to SAR," in [*Proc.SPIE*], 10633 – 10633 – 12 (2018).

[8] Auer, S. J., *3D synthetic aperture radar simulation for interpreting complex urban reflection scenarios*, PhD thesis, Technische Universität München (2011).

[9] Hammer, H. and Schulz, K., "Coherent simulation of SAR images," in [*Image and Signal Processing for Remote Sensing XV*], **7477**, 74771G, International Society for Optics and Photonics (2009).

[10] Allan, J. and Collins, M., "SARSIM: A digital SAR simulation system," (09 2007).

[11] Balz, T. and Stilla, U., "Hybrid GPU-based single-and double-bounce SAR simulation," *IEEE Transactions on Geoscience and Remote Sensing* **47**(10), 3519–3529 (2009).

[12] Cha, M., Majumdar, A., Kung, H., and Barber, J., "Improving SAR automatic target recognition using simulated images under deep residual refinements," in [*2018 IEEE International Conference on Acoustics, Speech and Signal Processing*], (2018).

[13] Moore, L. and Majumder, U., "An analytical expression for the three-dimensional response of a point scatterer for circular synthetic aperture radar," in [*SPIE 2010 Algorithms for Synthetic Aperture Radar*], (2010).

[14] Lewis, B., Liu, J., and Wong, A., "Generative adversarial networks for SAR image realism," in [*SPIE 2018 Algorithms for Synthetic Aperture Radar*], 10 (2018).

[15] Rosencrantz, S., Nehrbass, J., Zelnio, E., and Sudkamp, E., "AFacet: A geometry based format and visualizer to support SAR and multisensor signature generation," in [*SPIE 2018 Algorithms for Synthetic Aperture Radar*], (2018).

[16] Nehrbass, J. W., "Generic parallel system (HPC-GPS)," in [*Algorithms for Synthetic Aperture Radar Imagery XXVI*], **10987**, 10987–9, International Society for Optics and Photonics (2019).

[17] Doerry, A. W., "Basics of polar-format algorithm for processing synthetic aperture radar imagery," tech. rep., Sandia National Laboratory, Albuquerque, NM (2012).

[18] Scarnati, T. and Gelb, A., "Variance based joint sparsity reconstruction of synthetic aperture radar data for speckle reduction," in [*Algorithms for Synthetic Aperture Radar Imagery XXV*], **10647**, 106470R, International Society for Optics and Photonics (2018).

[19] Gelb, A. and Scarnati, T., "Reducing effects of bad data using variance based joint sparsity recovery," *Journal of Scientific Computing* , 1–27 (2017).

[20] Fischler, M. A. and Bolles, R. C., "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM* **24**, 381–395 (June 1981).

[21] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q., "Densely connected convolutional networks," in [*Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*], (2017).

[22] Maaten, L. v. d. and Hinton, G., "Visualizing data using t-SNE," *Journal of machine learning research* **9**(Nov), 2579–2605 (2008).

[23] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A., "Automatic differentiation in pytorch," in [*NIPS-W*], (2017).

[24] Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., and Boult, T. E., "Toward open set recognition," *IEEE transactions on pattern analysis and machine intelligence* **35**(7), 1757–1772 (2013).

[25] Bendale, A. and Boult, T. E., "Towards open set deep networks," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 1563–1572 (2016).

[26] Scheirer, W. J., Jain, L. P., and Boult, T. E., "Probability models for open set recognition," *IEEE transactions on pattern analysis and machine intelligence* **36**(11), 2317–2324 (2014).

[27] Li, F. and Wechsler, H., "Open set face recognition using transduction," *IEEE transactions on pattern analysis and machine intelligence* **27**(11), 1686–1697 (2005).

[28] Cruz, S., Coleman, C., Rudd, E. M., and Boult, T. E., "Open set intrusion recognition for fine-grained attack categorization," in [*IEEE International Symposium on Technologies for Homeland Security (HST)*], 1–6, IEEE (2017).

[29] Henrydoss, J., Cruz, S., Rudd, E. M., Boult, T. E., et al., "Incremental open set intrusion recognition using extreme value machine," in [*16th IEEE International Conference on Machine Learning and Applications (ICMLA)*], 1089–1093, IEEE (2017).

[30] Zhou, Y., Hu, Q., and Wang, Y., "Deep super-class learning for long-tail distributed image classification," *Pattern Recognition* **80**, 118–128 (2018).

[31] Salakhutdinov, R., Torralba, A., and Tenenbaum, J., "Learning to share visual appearance for multiclass object detection," in [*CVPR 2011*], 1481–1488, IEEE (2011).

[32] Arnold, J., Moore, L., and Zelnio, E., "Blending synthetic and measured data using transfer learning for synthetic aperture radar (SAR) target classification," in [*SPIE 2018 Algorithms for Synthetic Aperture Radar*], (2018).

[33] Friedlander, R., Levy, M., Sudkamp, E., and Zelnio, E., "Deep learning model-based algorithm for SAR ATR," in [*SPIE 2018 Algorithms for Synthetic Aperture Radar*], (2018).