Novel view synthesis with compressed sensing as data augmentation for SAR ATR

Katherine M. Banas[a], Tyler A. Hill[a], Chris Kreucher[a], Brian O. Raeker[a], Kyle Simpson[b], Kirk Weeks[b]
[a]KBR, Inc., 900 Victors Way, Ann Arbor, MI, USA, 48108; [b]Signature Research, Inc., 2045 Fountain Professional Ct., Navarre, FL, USA, 32566

ABSTRACT

This work investigates the application of compressed sensing algorithms to the problem of novel view synthesis in synthetic aperture radar (SAR). We demonstrate the ability to generate new images of a SAR target from a sparse set of looks at said target, and we show that this can be used as a data augmentation technique for deep learning-based automatic target recognition (ATR). The newly synthesized views can be used both to enlarge the original, sparse training set, and in transfer learning as a source dataset for initial training of the network. The success of the approach is quantified by measuring ATR performance on the MSTAR dataset

Keywords: novel view synthesis, data augmentation, synthetic aperture radar, automatic target recognition, basis pursuit denoising, transfer learning, deep learning, MSTAR dataset

1. INTRODUCTION

The application of deep learning (DL) using convolutional neural networks (CNNs) to the problem of synthetic aperture radar (SAR) automatic target recognition (ATR) has resulted in considerable performance gains over conventional template matching methods. However, harnessing the full power of these algorithms, which have enjoyed massive success on electro-optical (EO) imagery, typically requires large training datasets [1]. This is a challenge in SAR, where data is costly to collect. This motivates the use of synthetic training data and data augmentation strategies [2].

Data augmentation can help CNNs learn robust features rather than overfit to nuisance features such as clutter. Due to the widespread use of synthetic data in SAR ATR, a large class of data augmentation strategies focuses on reducing the synthetic/measured data mismatch and requires both data types in training. These methods use strategies such as GANs [3] or more realistic phase noise learned from measured data [4]. There are also data augmentation strategies that leverage SAR-specific features, such as attributed scattering centers (ASCs) [5] which leverage the fact that SAR imagery is sparse. Augmentation strategies valid for the EO domain, such as image rotation, have reduced applicability to SAR imagery due to the different underlying physics [6].

Novel view synthesis has been used as a data augmentation strategy in the EO domain [7], and this work introduces a SAR analogue which can be used to generate SAR imagery with novel operating conditions. To improve performance, we leverage a priori knowledge of SAR image sparsity alongside transfer learning techniques.

Our approach employs methods from the fields of compressed sensing and ℓ_1 minimization for novel view synthesis. Compressed sensing and ℓ_1 minimization techniques have seen increased use in SAR, as targets in SAR imagery (especially man-made targets) are often well-represented as a sparse superposition of point scatterers. We leverage these methods and motivating prior domain knowledge in our work on novel view synthesis. In this work, we perform an experiment using synthetic image chips of the same military vehicle targets as in the MSTAR dataset [8]. We generate chips at a novel set of azimuth (aspect) angles from this original dataset, which contains a small number of chips comprising a sparse set of center azimuth angles. The Basis Pursuit Denoising (BPDN) algorithm is used for novel view synthesis; the chosen BPDN formulation is typically used for sparse image reconstruction, but as the algorithm fills in k-space to solve for a sparse image that fits the data, apertures at novel azimuth angles can be interpolated from this filled k-space to generate new chips. We demonstrate empirical success in using the newly synthesized chips together with the original chips, both as a combined training set, and in a transfer learning (TL) setup, where we initially train a network on the synthesized chips and resume training on the original chips. A performance improvement of 9.6% median probability of correct classification (PCC) is achieved over training on the original, sparse-in-azimuth chips alone.

2. SAR IMAGE DATASETS

2.1 MSTAR and MSTAR-Synthetic Datasets

The MSTAR dataset is a publicly available collected SAR image dataset widely used in the ATR literature [8]. It contains 10 classes of military vehicles: the 2S1, BMP-2, BRDM-2, BTR-60, BTR-70, T-62, T-72, Caterpillar D7, and ZSU 23/4. It is common to partition the MSTAR dataset into a set of chips collected at approximately 17° elevation ("MSTAR-17"), and a set of chips collected at approximately 15° elevation ("MSTAR-15"); then, a classification algorithm can be trained on the MSTAR-17 data and tested on the MSTAR-15 data. We have found that the similarity of these two datasets leads to an overly optimistic evaluation of performance when training on one and testing on the other, so in this work we have opted to use a synthetically generated dataset for training. We evaluate the performance on the standard 3203 collected chips from the MSTAR-15 dataset. The MSTAR dataset (at both 15° and 17° elevation) contains chips covering a wide range of target-centric azimuth look angles, meaning that a training set with only a sparse set of center azimuths will be insufficient for classifying the collected MSTAR data.

To form our original, sparse-in-azimuth training set from which novel views are then synthesized, we simulated synthetic SAR data from a 3D CAD model of each MSTAR target, using asymptotic ray-tracing methods. The data was generated at HH polarization, 15° elevation, and X-band with enough bandwidth to achieve comparable resolution to the collected MSTAR data. All 360° of azimuth were simulated, but for each target we drew a random, sparse set of azimuth angles at which to form imagery. Note the difference between our use of the terms "synthetic" (which we use to refer to the data simulated with the ray-tracing code) and "synthesized" (which we use to refer to novel views of data generated with BPDN from the synthetic data).

2.2 Sampling Azimuths from the MSTAR-Synthetic Dataset

For each target in the MSTAR-synthetic dataset, we randomly selected 50 azimuth angles and formed an image with far-field backprojection and enough aperture about this angle to achieve approximately square image resolution. This results in 500 total original training chips among the 10 targets. For a given target, we first selected $N_{ctr} = 10$ center azimuth angles distributed approximately evenly about the polar annulus, i.e., the i^{th} center azimuth is chosen to be fit this data is given by:

$$\theta_{ctr,i} = \frac{360^\circ}{N_{ctr}}\big((i-1) + 0.25X\big),\ X \sim \mathcal{U}([-1,1]),\ i = 1,\dots,N_{ctr}. \tag{1}$$

Then, for each of these center azimuths, we randomly selected $N_{az} = 5$ nearby azimuth angles, where the j^{th} azimuth in the set near the i^{th} center azimuth is:

$$\theta_{i,j} = \theta_{ctr,i} + \frac{360^\circ}{N_{ctr}} \cdot 0.4Y,\ Y \sim \mathcal{U}([-1,1]),\ j = 1,\dots,N_{az}. \tag{2}$$

By making the center azimuths approximately evenly distributed about the annulus in (1), we avoid large gaps in azimuth between the available chips. But by forcing individual chips to be close to the azimuth centers in (2), we leave large enough gaps such that it is still a challenging problem for BPDN to fill in k-space data between the available apertures.

3. BASIS PURSUIT FOR NOVEL VIEW SYNTHESIS

3.1 Background

ℓ_1 minimization approaches from the compressed sensing literature, such as the Basis Pursuit Denoising (BPDN) algorithm, have been successfully applied to problems such as sparse image reconstruction and gap-filling phase history data in SAR [9]. These algorithms exploit the prior that targets (especially man-made targets like military vehicles) are made up of a sparse collection of point scatterers in the image domain. A typical BPDN formulation for these problems is given in (3), where x is the image pixel or voxel values, b is the measured phase history samples provided to the algorithm, σ is a small positive scalar, and A is the forward linear operator mapping image data to phase history data:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \|x\|_1 \text{ subject to } \|Ax - b\|_2^2 \leq \sigma. \tag{3}$$

If multiple phase history apertures from a target can be coherently combined such that they fill out portions of a 2D or 3D k-space, BPDN can solve for an apodized and superresolved image, and thus fill gaps in k-space from which new apertures can be interpolated to generate more images.

3.2 BPDN for *k*-space Gap Filling

In our experiments, the apertures of synthetic data for a given target are coherent and can thus be combined into a common k-space for BPDN. For our problem, we run BPDN separately on each target and define the measurements b in (3) to be the 50 phase history apertures. We use a forward operator of $A = RP^{-1}FZ$, where Z is the zero-padding operator and F is the Fourier Transform operator. P^{-1} is the Inverse Polar Format operator that interpolates evenly sampled Cartesian k-space grid data to polar phase history data; here, we interpolate the 2D k-space grid to the full 360° polar annulus for the available bandwidth. Finally, R is the restriction operator that selects from the full 360° polar annulus only the samples at azimuth angles that lie within the 50 available apertures.

We use a reweighted ℓ_1 minimization technique as in [10], and instead of directly solving (3), repeatedly solve the weighted BPDN problem, where the solution at the $(i+1)^{\text{th}}$ iteration is:

$$\hat{x}^{(i+1)} = \underset{x}{\operatorname{argmin}} \left\| W^{(i)} x \right\|_1 \text{ subject to } \|Ax - b\|_2^2 \leq \sigma. \tag{4}$$

Here, $W^{(i)}$ is a diagonal matrix with weights $w_j^{(i)},\ j = 1, \ldots, N$. We define $w_j^{(0)} = 1$ for all j, and $w_j^{(i)} = \frac{1}{\sqrt{\left|\hat{x}_j^{(i)}\right| + \varepsilon}}$ for $i \geq 1$. Using the same assumptions as [10] uses to prove that reweighted ℓ_1 minimization with $w_j^{(i)} = \frac{1}{\left|\hat{x}_j^{(i)}\right| + \varepsilon}$ is a majorize-minimize algorithm for minimizing the $\sum_j \log(|x_j| + \varepsilon)$ objective function, it can be shown that the reweighting we use is a majorize-minimize algorithm for solving the comparable $\sum_j \sqrt{|x_j| + \varepsilon}$ objective. This is a non-convex objective more functionally similar to the ℓ_0-norm than the ℓ_1-norm is, and thus its solution is often sparser than the ordinary BPDN solution [10]. At each iteration of reweighted ℓ_1 minimization, we use the spectral projected gradient algorithm in [11] to solve (4).

3.3 Novel View Synthesis

The result of the BPDN algorithm is an estimated sparse image (with a corresponding filled k-space) for each target. After taking the Fourier Transform of this image, we then perform Inverse Polar Format on the resulting, gap-filled k-space to generate a new aperture at each azimuth angle in {0°, 1°, …, 359°}. For each target, we use the same bandwidth and azimuth dwell as the original 50 apertures, and the same far-field backprojection code to form an image from each of the 360 new apertures. This ultimately yields a new training set of 3600 total chips among the 10 targets.

4. NETWORK TRAINING STUDIES

4.1 Network Training Setup

To assess the utility of improving SAR ATR performance via novel view synthesis with reweighted BPDN, we train a CNN using several datasets: the small (50 images per target) MSTAR-synthetic dataset, the dataset synthesized via solving BPDN on each target, the union of the MSTAR-synthetic and BPDN datasets, and a transfer learning (TL) setup in which the network is initially trained on the BPDN dataset and training is resumed on the MSTAR-synthetic dataset. Performance of the network with each training set is evaluated on the MSTAR-15 collected dataset.

We use the AConvNets architecture [12] for our models. AConvNets has shown excellent performance on the MSTAR dataset by limiting the number of trainable parameters to mitigate overfitting on small SAR datasets [12]. We use the following hyperparameters while training AConvNets:

- We train the network using cross-entropy loss with softmax activation.
- We preprocess each chip by finding the top 500 pixels, zeroing all pixels not in the top 500, and dividing pixel magnitudes by the maximum magnitude pixel in the chip such that all pixel magnitudes lie in the interval [0, 1].
- We perform data augmentation in the training loop, which involves shifting the image by a random number of pixels. This encourages the network to perform robustly when targets are off-center in the chip.
- We train the network using an ADADELTA optimizer with a learning rate of 1.0, and a batch size of 32.

When setting up the AConvNets transfer learning (TL) experiment, we recall that in typical use-cases of TL, models are first trained on a "source dataset" with plentiful samples but some characteristics that deviate from the actual data to be classified, and then fine-tuned on a smaller "targeted dataset" more representative of the problem of interest. In our experiment, we follow this paradigm and use the BPDN chips (a larger dataset with some errors relative to collected chips at those azimuths) as the source dataset, and the MSTAR-synthetic chips (a smaller dataset more well-matched to collected data) as the targeted dataset. It is common in TL to "freeze layers", i.e., to fix the learned weights of certain layers after training on the source dataset so that they are not updated when training is resumed on the targeted dataset. In our experiments, we froze layers at the locations labeled 1 through 5 in Figure 1, as well as freezing no layers, before transferring from the source dataset to the targeted dataset. When we freeze at one of the labeled locations in Figure 1, we freeze the layer next to the label as well as all preceding layers.

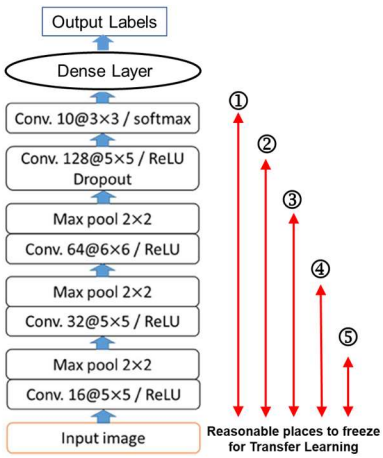

Figure 1. The AConvNets network architecture with freeze locations for transfer learning. Figure adapted from [12].

Each of our non-TL and TL experiments uses a different number of epochs, namely: 2296 epochs when training on MSTAR-synthetic data only (500 chips); 319 epochs when training on BPDN data only (3600 chips); 280 epochs when training on MSTAR-synthetic + BPDN data (4100 chips); and for TL, 160 epochs when initially training on the BPDN data and 1148 epochs when subsequently training on the MSTAR-synthetic data. This ensures that each experiment uses approximately the same number of model updates with the 32-sample batch size. We hold the number of model updates fixed so the performance is affected only by data quality and experimental setup (i.e., TL or not) and is not confounded by the number training steps. Furthermore, these choices for number of epochs ensure that training has roughly converged (performance has plateaued) in all experiments (including in both the source and targeted stages of TL).

As a final note, it is well-known that training deep CNNs is an inherently stochastic process [6], due both to the random data augmentations applied to each sample and the random partitioning of the training set into batches with stochastic gradient descent-style optimizers like ADADELTA [13]. To account for this and assess performance statistically, each experiment was repeated for 5 Monte Carlo trials. We report the probability of correct classification (PCC) on the MSTAR-15 collected dataset for the last 20% of epochs in the 5 trials.

4.2 Results

We report performance of these experiments in Figure 2, which shows box plots of PCC achieved on the test dataset over the last 20% of epochs and the 5 Monte Carlo trials. Overall, median test-set PCCs of 47.7%, 55.6% and 57.3% were achieved by MSTAR-synthetic training data only, combined MSTAR-synthetic and BPDN training data, and TL with layer-freezing at location 2 in Figure 1, respectively. Thus, data augmentation with BPDN novel view synthesis used in a TL context yielded a 9.6% improvement in classification performance over the baseline without novel view synthesis. Note that baseline performance in the MSTAR-synthetic-only experiment is poor relative to other experiments where networks were trained on synthetic datasets and tested on the MSTAR datasets [2]; this is because so few training samples (50 chips per class) at such a small fraction of possible azimuths were used. TL yielded modest gains in the median test-set PCC over naïvely training on the MSTAR-synthetic and BPDN chips together, and significant gains over training on the MSTAR-synthetic chips alone. The best results were obtained by freezing layers at relatively deep points in the

network. We hypothesize that allowing earlier layers to be re-trained may have caused the model to overfit the azimuth-starved MSTAR-synthetic dataset.

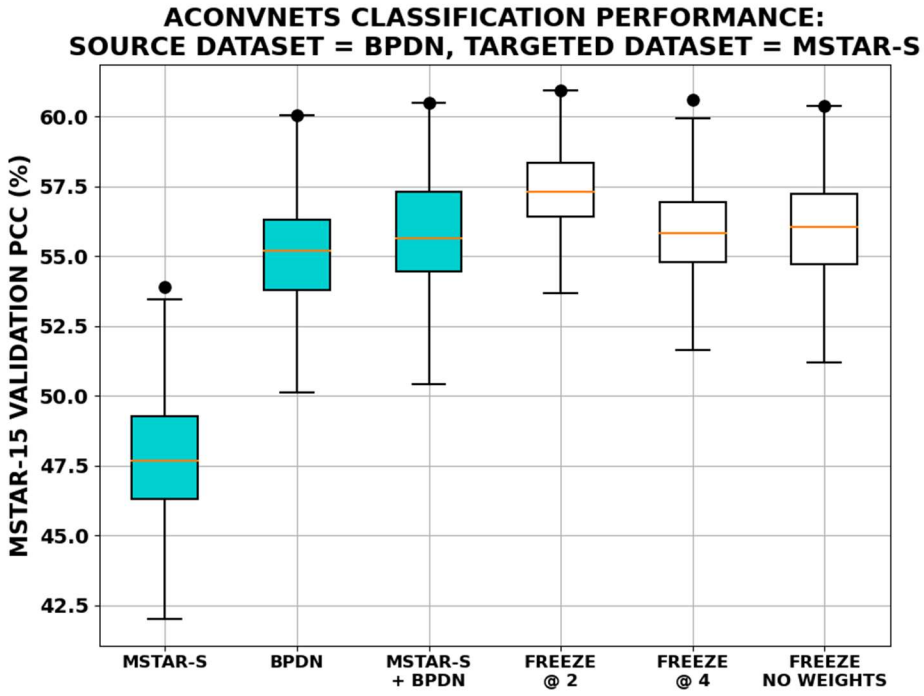

Figure 2. Boxplots of AConvNets performance (PCC) on the 15° MSTAR-collected dataset, achieved in the last 20% of epochs over 5 Monte Carlo trials. Black dots are the best-ever PCC for a given training setup. Single-training set experiments for the MSTAR-synthetic only, BPDN-synthesized only, and MSTAR-synthetic + BPDN datasets are shown in teal. Transfer learning experiments (transferring from BPDN to MSTAR-synthetic) are shown in white for the layer-freezing locations depicted in Figure 1.

When comparing test-set PCC vs. number of model updates among the different experiments in the convergence plot in Figure 3, we see that towards the end of training, TL with the BPDN data consistently has a better mean performance than training on the BPDN data alone or naïvely training with the combined MSTAR-synthetic and BPDN data. However, the mean PCCs of the BPDN experiment and the MSTAR-synthetic + BPDN experiment are still less than one standard deviation below the mean PCC of the TL experiment, so the improvement is not particularly dramatic. It is worth noting that the PCC increases quite quickly in the TL experiment immediately after transferring from training on the BPDN data to training on the MSTAR-synthetic data. Thus, one benefit of using the BPDN data in a TL context is that it gets to a high PCC rapidly (i.e., within a few model updates after transferring), even if this PCC is not significantly higher than that of training on the MSTAR-synthetic and BPDN data together without TL.

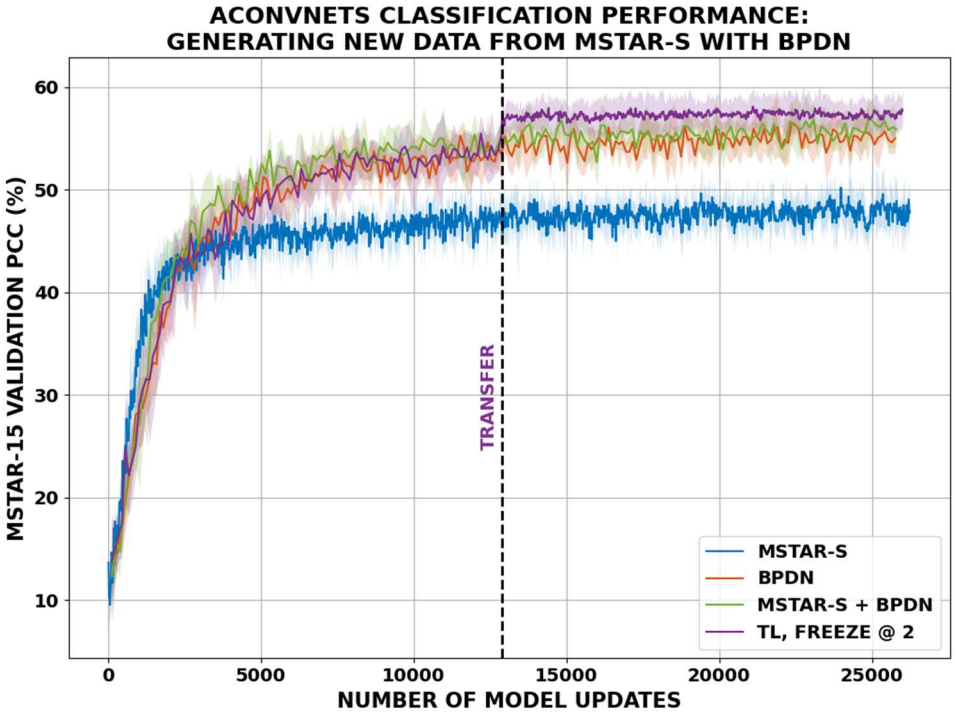

Figure 3. Mean (solid lines) $\pm$ standard deviation (filled transparent areas) PCC on the MSTAR-15 test dataset over the 5 Monte Carlo trials vs. number of model updates. Results are shown for each of the following experiments: training on the MSTAR-synthetic data only ("MSTAR-S"), training on the BPDN data only ("BPDN"), training on the combined MSTAR-synthetic and BPDN datasets ("MSTAR-S + BPDN"), and TL with initially training on the BPDN data and resuming training on the MSTAR-synthetic data and freezing layers at location 2 in Figure 1 ("TL, FREEZE @ 2").

While these experiments demonstrate that synthesizing new chips from existing chips with BPDN and using both datasets in a TL context has utility for SAR ATR, there are limitations on the potential performance gains. BPDN tends to do best at filling in k-space near the measured apertures it has access to, and it particularly struggles to fill in scatterers that are not illuminated in these apertures. This is a fundamental limitation of BPDN, as adding new scatterers to the image would increase the ℓ_1 objective. Consider the T-62 target in Figure 4. The image that BPDN predicts at the missing angles (bottom right) does not contain every scatterer in the true MSTAR-synthetic image at those angles (top right); rather, the only scatterers present are those illuminated in the image that BPDN has access to (top left).

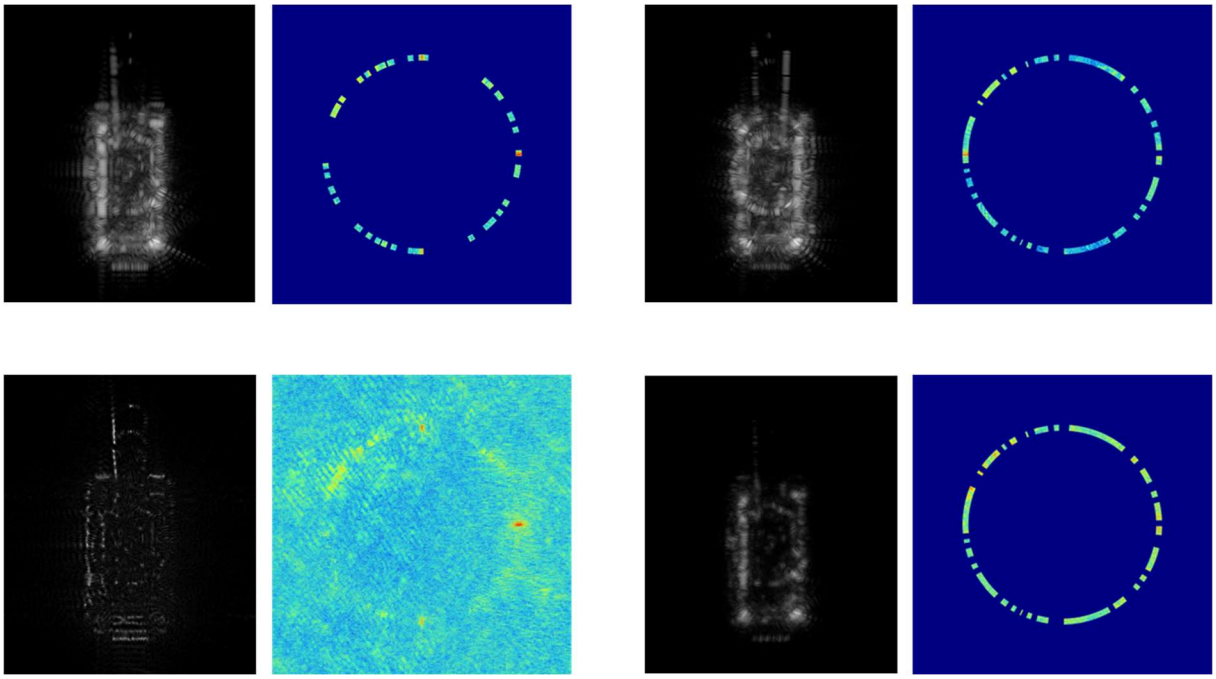

Figure 4. Images and corresponding k-space for the T-62 target. Top left: MSTAR-synthetic data at just the angles provided to BPDN. Top right: MSTAR-synthetic data at just the missing angles not provided to BPDN. Bottom left: result of running BPDN on the apertures shown in the top left. Bottom right: masking the k-space filled in by BPDN to just include the missing angles (same angles shown in the top right).

5. CONCLUSION

We have demonstrated the ability to improve the performance of deep learning-based SAR ATR algorithms on small datasets covering a sparse set of azimuth angles, by generating additional training data via novel view synthesis with the Basis Pursuit Denoising (BPDN) algorithm. If the apertures from the sparse-in-azimuth training set can be coherently combined into a common k-space, BPDN can solve for a sparse image that fits this combined data, and thus fill in gaps in k-space from which new apertures and corresponding training chips can be generated. In our experiments, we used a sparse training set of synthetic data of the MSTAR targets, generated additional chips from it with BPDN, and quantified performance on the collected 15°-elevation MSTAR dataset. Overall, median test-set probability of correct classification (PCC) scores of 47.7%, 55.6% and 57.3% were achieved by MSTAR-synthetic training data only, combined MSTAR-synthetic and BPDN training data, and a transfer learning (TL) setup where the network is initially trained on the BPDN data and subsequently trained on the MSTAR-synthetic data, respectively.

The data synthesized with BPDN (a larger dataset with errors) and the MSTAR-synthetic data (a small dataset well-matched to the collected test dataset) were effective at improving ATR when used in a TL context, especially when layers were frozen at relatively deep points in the network (likely because this prevented overfitting on the small MSTAR-synthetic dataset). While ATR with TL did not dramatically outperform ATR with the BPDN and MSTAR-synthetic data naïvely combined into a single training set, it is beneficial that TL achieved a high test-set PCC quickly (within a few model updates of transferring to training on the MSTAR-synthetic dataset).

The 9.6% improvement in test-set PCC achieved by TL with the novel views synthesized via BPDN versus just training on the original, sparse MSTAR-synthetic dataset is sizeable. However, the overall performance is not on par with other results achieved on MSTAR by training on high-quality synthetic data [2]. This is likely because BPDN struggles to fill gaps in k-space far away from the original apertures it has access to, especially when these gaps should contain scatterers that were not illuminated in the original apertures. This is a fundamental issue with the chosen BPDN formulation, as adding new scatterers would increase the ℓ_1 objective being minimized. In future work, it would be worthwhile to examine whether penalty functions or bases that encourage left-right symmetry of the target could be used to mitigate BPDN's

inability to fill in missing scatterers. It would also be of interest to explore novel view synthesis approaches other than BPDN, such as neural radiance fields (NeRF) methods that have recently been applied to SAR data [14].

6. ACKNOWLEDGMENTS

This research was funded by AFRL/RI under contract FA8750-21-C-1524. We are grateful to Mr. Bernard Clarke and Mr. Vincent Turczyn at AFRL and Dr. Thomas Mitchell at NGA for their guidance and support.

REFERENCES

[1] Wilmanski, M., Kreucher, C. and Lauer, J., "Modern approaches in deep learning for SAR ATR," Zelnio, E. and Garber, F., eds., Proc. 2016 SPIE Defense, Security and Sensing Symposium 9843 (April 2016).

[2] Masarik, M., Kreucher, C., Weeks, K. and Simpson, K. "End-to-End ATR Leveraging Deep Learning," Rysz, M., Tsokas, A., Dipple, K., Fair, K. and Pardalos, P., eds., Synthetic Aperture Radar (SAR) Data Applications, chapter 5. Springer (2022).

[3] Lewis, B., DeGuchy, O., Sebastian, J., & Kaminski, J., "Realistic SAR data augmentation using machine learning techniques," Algorithms for Synthetic Aperture Radar Imagery XXVI Vol. 10987, 12-28, SPIE (May 2019).

[4] Sellers, S. R., Collins, P. J. and Jackson, J.A.. "Augmenting simulations for SAR ATR neural network training." 2020 IEEE International Radar Conference (RADAR), 309-314 (2020).

[5] Lv, J. and Liu, Y. "Data augmentation based on attributed scattering centers to train robust CNN for SAR ATR". IEEE Access, 7, 25459-25473 (2019).

[6] Inkawhich, N., et al., "Bridging a Gap in SAR-ATR: Training on Fully Synthetic and Testing on Measured Data," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 14, 2942-2955 (2021).

[7] Eker, T., Kasaei, S. M., Pereboom-Huizinga, W., & den Hollander, R. "Classifying objects from unseen viewpoints using novel view synthesis data augmentation," Doctoral dissertation (2021).

[8] Ross, T., Worrell, S., Velten, V., Mossing, J., Bryant, M. "Standard SAR ATR Evaluation Experiments using the MSTAR Public Release Data Set," SPIE Conf. Algorithms Synthetic Aperture Radar Imaging V 3370, 566–573 (April 1998).

[9] Coleman, C., Connell, S., Gabl, E., and Walter, J, "Minimum L1 norm SAR image formation," 2012 IEEE Statistical Signal Processing Workshop (SSP), 544-547 (2012).

[10] Candès, E. J., Wakin, M. B. and Boyd S. P., "Enhancing Sparsity by Reweighted L1 Minimization", J. Fourier Anal. Appl. 14, 877-905 (2008).

[11] Van den Berg, E. and Friedlander, M., "Probing the Pareto Frontier for Basis Pursuit Solutions", SIAM Journal on Scientific Computing, vol. 31, no. 2, 890-912 (2009).

[12] Chen, S., Wang, H., Xu, F. and Jin, Y., "Target Classification Using the Deep Convolutional Networks for SAR Images," IEEE Transactions Geoscience Remote Sensing, vol. 54, no. 8, 4806–4817 (2016).

[13] Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.

[14] Wilmanski, M. and Tamir, J., "Differentiable rendering for synthetic aperture radar imagery," arXiv preprint arXiv:2204.01248 (2022).