Improving SAR ATR using synthetic data via transfer learning

Brian O. Raeker[a], Tyler A. Hill[a], Chris Kreucher[a], Katherine M. Banas[a], Kevin Tactac[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

Attempts to use synthetic data to augment measured data for improved synthetic aperture radar (SAR) automatic target recognition (ATR) performance have been hampered by domain mismatch between datasets. Past work which leveraged synthetic data in a transfer learning framework has been successful but was primarily focused on transferring generic SAR features. Recently SAMPLE, a paired synthetic and measured dataset was introduced to the SAR community, enabling demonstration of good ATR performance using 100% synthetic data. In this work, we examine how to leverage synthetic data and measured data to boost ATR using transfer learning. The synthetic dataset corresponds to the MSTAR 15^{o} dataset. We demonstrate that high quality synthetic data can enhance ATR performance even when substantial measured data is available, and that synthetic data can reduce measured data requirements by over 50% while maintaining classification accuracy.

Keywords: transfer learning, synthetic aperture radar, automatic target recognition, MSTAR dataset, SAMPLE dataset, deep learning

1. INTRODUCTION

Synthetic aperture radar (SAR) automatic target recognition (ATR) is often limited by the availability of labeled measured data. Off-the-shelf networks trained on measured electro-optical (EO) imagery fail to generalize straightforwardly to SAR due to the significant differences in imaging characteristics between domains, with SAR imagery having significantly higher sparsity and dynamic range. An alternative data source supplementing measured data is synthetically generated data. Synthetic data is attractive since SAR data is expensive to collect, and the significant performance gains allowed by ATR based on convolutional neural networks (CNN) require large training datasets [1].

Early SAR transfer learning (TL) experiments leveraged unlabeled data to improve ATR performance [2]. Later work demonstrated performance improvements combining synthetic data with the same classes as the measured data, but training with synthetic data alone yielded poor performance [3]. The SAMPLE dataset made high quality paired synthetic and measured data available [4], which enabled very good (>90%) ATR performance using 100% synthetic data [5]. This excellent performance on purely synthetic data motivates the present work, which investigates how transfer learning can be used to further improve SAR ATR performance when the simulated data fidelity is high. Prior transfer learning experiments used well matched classes of MSTAR targets [6] and SAMPLE targets [7], but did not demonstrate the effectiveness of transfer learning as a function of the quantity of measured data, or perform studies as to when transfer learning on these SAR datasets is most effective. In this work we demonstrate substantial performance gains of >10% absolute performance improvement when 10% of measured MSTAR data is available for training, using a synthetic dataset created to correspond to MSTAR. For comparison, we examine experiments where the synthetic dataset is not well-matched to the validation set, and while we still observe performance improvements when transferring to limited measured data, they are less substantial than the well-matched synthetic data case, suggesting class specific features can be successfully transferred.

This paper is organized as follows. First, Section 2 describes the experiments performed in this study. Section 3 describes the effect of freezing layers in a transfer learning experiment. In Section 4, we provide experiment results comparing classification performance of TL against the performance with a single training stage. Finally, Section 5 provides conclusions from this effort.

2. SAR ATR EXPERIMENT APPROACH

2.1 ATR Framework

Our experiments use a SAR ATR test harness to evaluate ATR algorithms and transfer learning capabilities. The harness includes implementations of more than a dozen deep learning models from the literature (including AConvNets, DenseNet, ResNet, Inception, EfficientNet and their variants) [8]. Multiple data preprocessing approaches (such as scaling, quantization, and clipping) and data augmentation strategies are also available. The testbed interfaces with several standard and user-defined datasets, including the MSTAR [9] and SAMPLE [4] datasets. The testbed is configurable to allow any network architecture to be trained with any dataset using any combination of preprocessing and augmentation methods and validated against any other dataset. We have used this test setup to systematically evaluate the impact of each portion of the ATR algorithm on classification performance.

We have found that SAR ATR performs well with the following settings which we use to obtain the results presented in this paper:

- Data preprocessing: Select the top $N = 500$ pixels in magnitude to form the image and normalize by the maximum value.
- Data augmentation: A random image shift in each pixel dimension is applied to each image during training. This provides resilience to targets being off-center in the image.
- Deep learning network: Network architecture is user defined. We used the AConvNets and DenseNet architectures in this paper.
- Loss function: cross entropy loss.

Classification performance is measured as the probability of correct classification (PCC) of validation dataset samples, which is a number between 0 (completely incorrect classification) and 1 (correct classification for all samples). Typically, we run multiple trials for each experiment to provide results showing average performance, reducing variation due to independently trained networks and the randomly chosen samples when subsetting the target stage training dataset. Classification performance for each experiment is typically reported as the average and standard deviation of the PCC for the last 10 epochs of 10 trials (100 performance results).

2.2 SAR Image Datasets

Our work uses the publicly available MSTAR [9] and SAMPLE [4] datasets. The MSTAR dataset contains 10 target classes (2S1, BMP-2, BRDM-2, BTR-60, BTR-70, T-62, T-72, Caterpillar D7, and a ZSU 23/4), and the SAMPLE dataset also contains 10 target classes (2S1, BMP-2, M548, M1, BTR-70, T-72, M2, M35, M60, ZSU 23/4). Five target classes are present in each dataset and the remaining five classes are unique to each dataset. The two datasets are obtained from the same data collection, with some chips of the shared 5 classes identically present in both the MSTAR and SAMPLE datasets. Figure 1 plots the azimuth and elevation imaging angles for each dataset, allowing the identical chips to be identified.

The MSTAR dataset is typically binned into images formed at two different elevation angles. The MSTAR datasets are termed MSTAR17 for the data measured at elevation angles near 17 degrees, and MSTAR15 for data measured near 15 degrees.

Figure 1 shows there are various amounts of overlapping chips between MSTAR and SAMPLE. The SAMPLE dataset provides unique chips at elevation angles near 16 degrees but contains a mixture of identical and unique chips at elevation angles near 17 degrees depending on the class. Two target classes also contained chips at elevation angles near 15 degrees. In the following experiments we removed all chips from the SAMPLE set that were members of the MSTAR15 set. This allows us to use MSTAR15 for validation since the network will not be trained on these chips. The angular diversity of these publicly available SAR datasets allows evaluation of transfer learning utility to ATR performance improvement.

Synthetic data was generated using asymptotic ray-tracing techniques from 3D CAD models of the MSTAR targets. The data was simulated at X-band, with bandwidth and aperture chosen to achieve 1 foot resolution. The image chips were then formed by backprojection of the synthetic data at 1 degree increments in azimuth at 15 degrees elevation for each of the 10 targets using HH, HV, VH, and VV polarizations, yielding 1440 synthetic chips per target class.

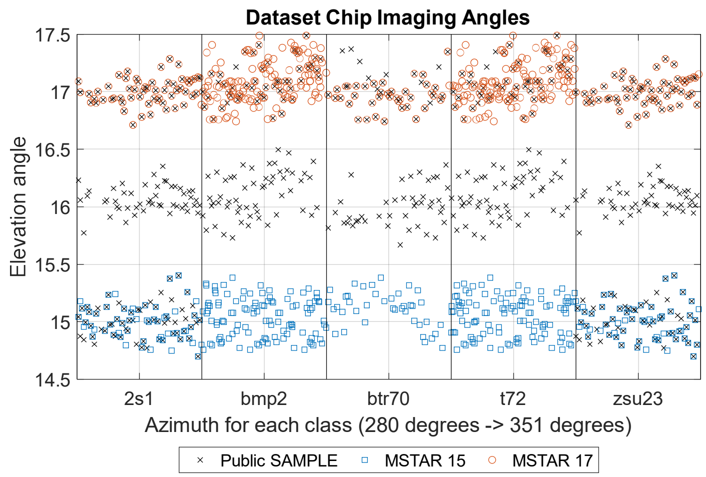

Figure 1: Azimuth and elevation angles of target chips for the shared target classes in the MSTAR and SAMPLE datasets.

2.3 Transfer Learning Methodology

Transfer Learning is a method of transferring knowledge gained by training a network on an initial ("source") dataset and applying it to a separate related problem (the "targeted" dataset) [10]. Knowledge transfer is performed by initializing network weights during the target dataset training stage with those learned from the source training.

Our transfer learning studies have the general form of: (i) training on a large amount of synthetic data in the source stage and then (ii) using various amounts of measured training data to continue training the network parameters in the target stage. Classification performance of the network is validated with a separate measured dataset. Typically, the measured training and validation datasets contain the same target classes, but the source-stage synthetic training dataset may contain different target classes from the measured datasets.

We used two architectures, AConvNets [11] and DenseNet [12], in our transfer learning studies. Transfer learning also allows "freezing" (preventing updates) some or all the learned weights before training with a second set of data. Different layers within the network can be frozen in the second training stage, we describe this approach using the AConvNets architecture shown Figure 2. As illustrated, there are five separate places where layer freezing may have utility in the AConvNets architecture. For each location, layers between the input image and the given location are frozen.

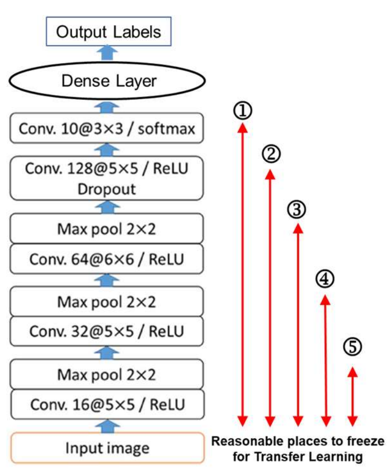

Figure 2: The AConvNets network architecture and potential freeze locations. Figure adapted from [11].

3. LAYER FREEZING DURING TRANSFER LEARNING

In this section, we briefly study the effect that freezing network layers between training stages has on the overall transfer learning classification performance. When a layer is frozen, the network weights in the layer are not updated during the next training stage. This prevents these layers from learning during the subsequent training stage. Since different parts of the network learn different features, freezing layers prevents these learned features from changing due to additional training data.

Different configurations of frozen layers are applied during repeated instances of a transfer learning experiment to identify the resulting change in PCC performance. AConvNets is used for this purpose and consists of a sequence of five convolutional and pooling layers. Figure 2 displays the network architecture. This provides several locations to freeze layers between transfer learning stages, identified as locations 1-5. Layers between the input image and the chosen location are frozen.

A series of experiments were performed to evaluate the performance impact of freezing layers during the transfer learning process. Locations 2, 3, and 4 were chosen to freeze to. Each trial was trained for 50 epochs on the entire synthetic MSTAR15 data, then for 50 epochs on a random subset of the measured MSTAR17 dataset. The network was validated against measured MSTAR15. In this case, the target and source stage training datasets are a high-quality match to the validation dataset in terms of target class and imaging parameters. Five trials were performed for each experiment, with performance statistics gathered over the last ten epochs of each trial. Figure 3 shows the results as a box plot of PCC performance as a function of frozen layers and target training stage set size.

As seen in Figure 3, there is a slight performance boost when freezing layers to locations 4 and 3 when training with 5%-10% of the target stage MSTAR17 dataset. However, as more measured data is used in the target training stage it becomes more advantageous to freeze no layers (allow the entire network to continue to train). Freezing most of the network (to location 2) provides the lowest performance, which is expected since the network cannot adequately learn from the available high-quality measured MSTAR17 training data.

In general, this experiment shows that freezing a few early network layers provides a performance improvement when the synthetic source stage dataset is well-matched to the measured validation dataset and there is a small quantity of measured target training stage data available. Otherwise, it is advantageous to not freeze the network, allowing it to continue fully learning on the target stage training set. Similar results related to layer freezing are also obtained in [13]. Therefore, no layers are frozen in the subsequent transfer learning experiments.

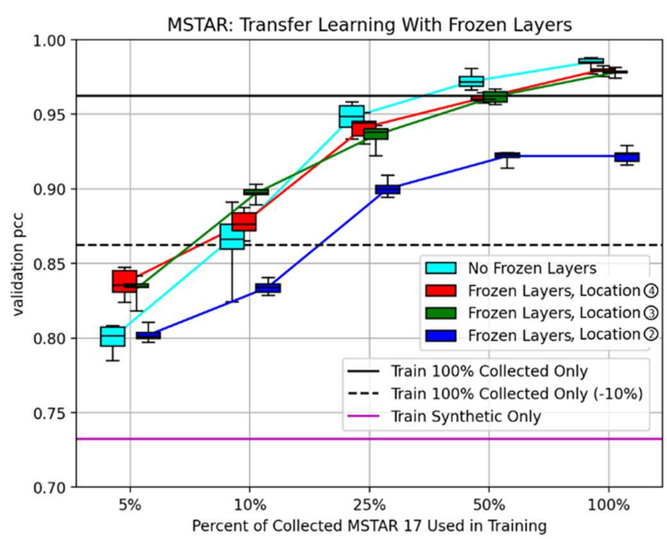

Figure 3: Frozen layers performance when training on MSTAR 15 synthetic transferring to MSTAR 17 measured and validating with MSTAR 15 measured.

4. TRANSFER LEARNING STUDIES

This section describes a series of transfer learning experiments using different training datasets and details the resulting classification performance. Our focus is to assess the performance of TL techniques with synthetic datasets that both match and mismatch the target classes relative to the measured datasets. The general structure of these experiments is to initially train on a large synthetic image dataset (source training stage) for 100 epochs, then transfer the network to a target training stage where training is performed with a measured image dataset. The network is validated against a separate measured image dataset.

Classification performance dependence on the measured training set size is examined through multiple executions of each TL experiment. For each execution a subset of the measured training dataset is randomly chosen for target stage training. Doing so allows characterizing the improvement of classification performance attributable to the transfer learning process with various quantities of measured training data.

A function fit was performed to the average PCC of 10 independent trials to approximate performance when using between 5% and 100% of the target training stage measured data. The function used to fit this data is given by equation

$$f(s, x_1, \dots, x_6) = x_1 e^{-|x_2|s} + x_3 e^{-|x_4|s} + x_5 e^{-|x_6|s} \quad (1)$$

where s is the subset fraction of the target stage training dataset. Coefficients $x_1, \dots, x_6$ were optimized to best match the function to the mean performance of the experiments. The standard deviation for the curve is identified as a transparent background around the average function-fit performance. Two curves are generated for each experiment set: 1) baseline performance where the network is only trained on a subset of the measured training data set samples, and 2) transfer learning performance where the network is trained on synthetic data for 100 epochs then transferred to continue training with the same subset fraction of measured training data as in the baseline performance case. Training with measured data was performed for more than 200 epoch to ensure the network completely learned from the measured training set.

4.1 Same Target Classes

In this section, we examine TL when the synthetic training dataset contains the same target classes as in the measured datasets and is therefore well-matched to the measured datasets. Separate experiments were performed which use different combinations of MSTAR and SAMPLE in the synthetic source stage and measured target stage, but the same validation dataset. Table 1 summarizes the datasets used for each experiment. Note that experiments involving both MSTAR and SAMPLE (experiments 2a-2e) only use the shared 5 classes (BMP, BTR70, T72, ZSU) and chips with an azimuth between 280 and 351 degrees due to the limitations of the Public SAMPLE dataset.

Table 1: Datasets used for TL experiments with the same target classes.

Exp.	Source Stage Training Set (Synthetic)	Target Stage Training Set (Measured)	Validation Set (Measured)	Number of Target Classes
1	MSTAR 15	MSTAR 17	MSTAR 15	10
2a	MSTAR 15	MSTAR 17	MSTAR 15	5
2b	MSTAR 15	SAMPLE16/17	MSTAR 15	5
2c	MSTAR15	SAMPLE16	MSTAR15	5
2d	SAMPLE	SAMPLE16/17	MSTAR15	5
2e	SAMPLE	MSTAR17	MSTAR15	5

The first experiment trains using the synthetic MSTAR15 data set in the source stage, the measured MSTAR17 data set in the target stage, and validates against the measured MSTAR15 data set. Figure 4 and Figure 5 display the classification PCC performance for AConvNets and DenseNet. Transfer learning provides a PCC increase for each randomly selected subset size of the target stage training data. The PCC increase is much larger when using a small amount of measured training data than for cases with many training data training samples.

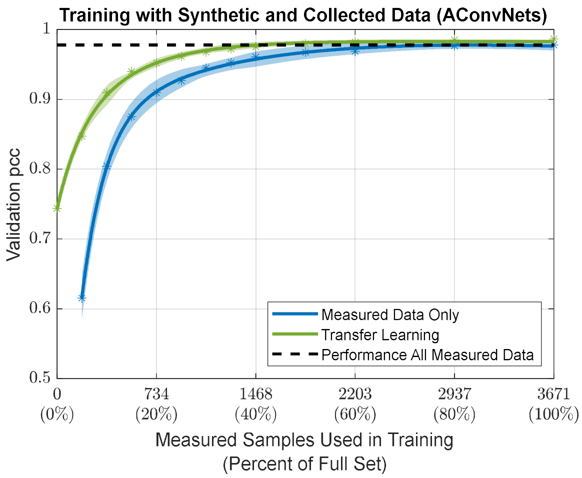

Figure 4: Transfer learning experiment results when training on synthetic MSTAR15, transferring to measured MSTAR17, and validating with measured MSTAR 15. The AConvNets architecture was used.

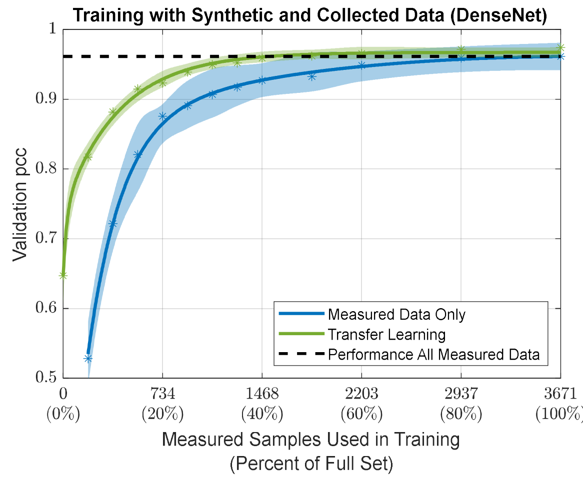

Figure 5: Transfer learning experiment results when training on synthetic MSTAR15, transferring to measured MSTAR17, and validating with measured MSTAR 15. The DenseNet architecture was used.

The transfer learning experiments in Figure 4 demonstrate that transfer learning using synthetic data allows for a 50% or more reduction in measured data while maintaining classification performance. For example, training with synthetic data and 10% of the MSTAR17 dataset yields equivalent performance to training with only 20% of the MSTAR17 dataset. This shows that the measured MSTAR17 data is very well matched to the measured MSTAR15 validation set and will dominate the classification performance of the trained network. Since both MSTAR datasets were obtained during the same collection with the same physical target vehicles, it is expected that the datasets would closely match in both target signature and background clutter.

The second set of experiments examines scenarios where the training datasets are not as well-matched to the measured validation data set. This series of experiments uses SAMPLE training datasets that are a poorer match to the measured MSTAR15 validation dataset to demonstrate the impact of transfer learning with synthetic data.

Figure 6 displays the classification performance for each transfer learning scenario, where the AConvNets network architecture was used. Experiment 2a is the highest-fidelity training data match to the validation dataset, where each stage utilizes MSTAR data. Transfer learning provides a performance increase when few measured training chips are used, but a much smaller benefit when a larger amount of measured MSTAR15 data is trained with. These results match those of experiment 1 since the same data was used.

Experiments 2b-2d each train with the measured SAMPLE dataset during the target training stage. As seen in Figure 6, the average PCC of each baseline case is about 0.85, suggesting that the measured Public SAMPLE dataset is not as well matched to MSTAR15 as compared to MSTAR17. Experiments 2b-2c both train with synthetic MSTAR15 during the source stage, leading to a large classification performance improvement. Experiment 2d trains with synthetic SAMPLE during the source stage, which leads to a slight performance improvement, smaller than observed for experiments 2b-2c. In all cases, initial training with synthetic data via transfer learning helps alleviate the dataset mismatch problem and leads to better performance regardless of the number of measured training samples used.

Experiment 2e demonstrates it is possible to obtain significantly increased performance transfer learning from a synthetic to a measured dataset as compared to training on either of those datasets individually, shown using publicly available MSTAR and SAMPLE datasets. Training using synthetic SAMPLE data from [4] we found 64.4% PCC validating on MSTAR15, and 77.2% when training on ~45 measured MSTAR17 chips, but 90.6% PCC when using these two training datasets together with transfer learning.

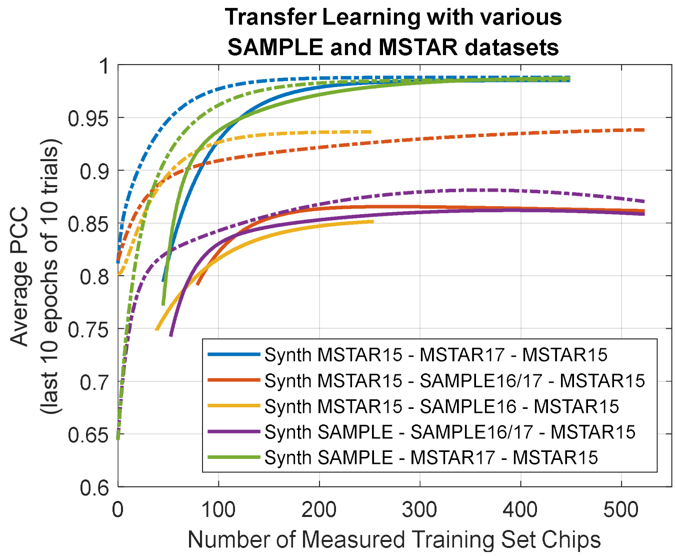

Figure 6: TL experiments involving both MSTAR and SAMPLE datasets. The legend lists datasets as (synthetic source training set)-(measured target training set)-(measured validation set). Solid lines represent the case of training with only measured data and dashed lines are the TL case. Network is AConvNets.

The results from the transfer learning experiments performed in this section are summarized into a few main conclusions. First, transfer learning leads to PCC improvement for all amounts of measured training stage data but provides the most improvement when small amounts of measured data are available to finish training. Second, performance improvement due to transfer learning is increased when the source dataset is well matched to the validation dataset. Finally, transfer learning performance improvement is large when measured training data is not as well-matched to the validation data as compared to the synthetic data (red and yellow curved of Figure 6), and in this case the performance benefit of transfer learning persists even when larger quantities of measured data is available.

4.2 Different Target Classes

In this section, the synthetic dataset used in the transfer learning experiment does not contain target classes present in the measured datasets (target classes are disjoint). We perform a transfer learning experiment where the synthetic dataset contains MSTAR target classes disjoint from the validation set. This experiment trains using the synthetic MSTAR 15 dataset (classes: 2S1, BMP, BTR70, T72, ZSU) in the source stage, the measured MSTAR 17 dataset (Classes: BRDM, BTR60, D7, T62, ZIL) in the target stage, and validates against the measured MSTAR 15 dataset (Classes: BRDM, BTR60, D7, T62, ZIL). The synthetic dataset contains samples from a disjoint class set compared to the classes contained in the measured datasets.

Figure 7 and Figure 8 display the classification PCC performance using AConvNets and DenseNet. Transfer learning provides a slight PCC increase compared to the baseline network trained only on the subset of the measured training set. Note that using only synthetic data during training (zero measured training samples used in the target training stage) results in a PCC that is very low since none of the synthetic training data samples represent classes contained in the measured datasets. This causes the network to misclassify the validation dataset. As more measured training data (which contains the desired target classes) is used in the transfer learning experiment, the PCC rapidly increases.

Overall, transfer learning from synthetic data of disjoint classes can be beneficial, with consistent modest performance benefit. This is due to the synthetic training set only containing disjoint target classes so only generic SAR features are transferrable and becomes apparent when comparing to the results in Section 4.1 where class specific features are transferred.

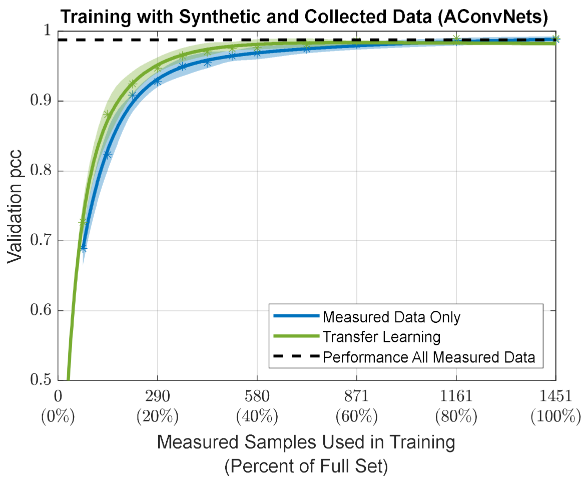

Figure 7 Transfer learning experiment results when training on synthetic MSTAR15 (disjoint), transferring to measured MSTAR 17, and validating with MSTAR15. The AConvNets architecture was used.

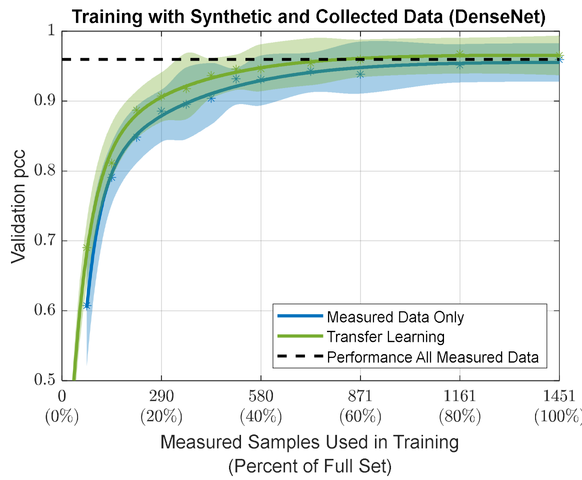

Figure 8 Transfer learning experiment results when training on synthetic MSTAR15 (disjoint), transferring to measured MSTAR 17, and validating with MSTAR15. The DenseNet architecture was used.

5. CONCLUSION

The good ATR performance training using 100% synthetic data corresponding to the SAMPLE and MSTAR datasets demonstrates that networks trained on synthetic data can learn class specific features. We see that the absolute performance gains from transfer learning when minimal measured data is available are much larger when the source dataset shares the same classes and imaging parameters as the validation set, supporting that class specific information is transferrable.

When measured data is scarce (i.e. <50 chips per class for a 10 class experiment) it can be useful to freeze layer weights when transferring, however, when more data is available performance is improved by letting all weights fine tune. We show the benefit of transfer learning persists across different network architectures. When synthetic data is plentiful and measured data is scarce, the benefits of transfer learning may be increased for deeper networks as suggested by our results on AConvNets as compared to DenseNet, but this merits further study.

Transfer learning leads to the largest performance improvements when available measured data is scarce, relatively less well matched to the validation data, and when the synthetic data is well matched to the validation dataset. We show it is possible to achieve significantly improved performance (>13% absolute PCC increase to over 90% PCC) using transfer learning from a synthetic to a measured dataset relative to training on these datasets individually using the publicly available synthetic SAMPLE and measured MSTAR datasets. Transfer learning from plentiful class matched synthetic data can allow 50% or more reduction in measured data requirements to achieve equivalent performance.

6. ACKNOWLEDGEMENTS

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] M.Wilmanski, C. Kreucher, and J. Lauer, “Modern approaches in deep learning for SAR ATR,” SPIE, 2016.

[2] D. Malmgren-Hansen, A. Kusk, J. Dall, A. A. Nielsen, R. Engholm, and H. Skriver, “Improving SAR automatic target recognition models with transfer learning from simulated data,” IEEE Geosci. Remote Sens. Lett., vol. 14, no. 9, pp. 1484–1488, Sep. 2017

[3] M. Cha, A. Majumdar, H. T. Kung, and J. Barber, “Improving SAR automatic target recognition using simulated images under deep residual refinements,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2018, pp. 2606–2610.

[4] Lewis, B., Zelnio, E., et. Al, "A SAR dataset for ATR development: the Synthetic and Measured Paired Labeled Experiment (SAMPLE)," Proc. SPIE 10987, 14 May 2019

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

[6] J. M. Arnold, L. J. Moore, and E. G. Zelnio, “Blending synthetic and measured data using transfer learning for synthetic aperture radar (SAR) target classification,” Proc. SPIE, vol. 10647, pp. 48–57, 2018.

[7] C. Clum, D. G. Mixon, and T. Scarnati, “Matching component analysis for transfer learning,” SIAM J. Math. Data Sci., vol. 2, no. 2, pp. 309–334, 2020.

[8] M. Masarik, C. Kreucher, K. Weeks, and K. Simpson. End-to-End ATR Leveraging Deep Learning. M. Rysz, A. Tsokas, K. Dipple, K.. Fair, and P. Pardalos, editors, Synthetic Aperture Radar (SAR) Data Applications, chapter 5. Springer, 2023.

[9] 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, vol. 3370, no. April 1998, pp. 566–573, 1198

[10] Pan, S., and Yang, Q., "A Survey on Transfer Learning," in IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, Oct. 2010

[11] 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, pp. 4806–4817, 2016.

[12] G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger, "Densely Connected Convolutional Networks“ Conference on Computer Vision and Pattern Recognition (CVPR), 2017, 4700-4708.

[13] Yosinski, J., Clune, j., Bengio, Y., and Lipson, H., “How transferable are features in deep neural networks?”, Advances in Neural Information Processing Systems 27 (NIPS ’14), NIPS Foundation, 20