SAR-ATR Using EO-based Deep Networks

Chris Kreucher
KBR Government Solutions
900 Victors Way, Suite 220
Ann Arbor, MI, 48108, USA
e-mail: chris.kreucher@us.kbr.com

***Abstract*—In recent years there has been widespread adoption of Deep Convolutional Neural Networks to electro-optical (EO) image classification, most famously using the ImageNet database to form challenge problems. The Synthetic Aperture Radar (SAR) classification problem, typically referred to as Automatic Target Recognition (ATR), is a related topic that has received less attention. While there have been some custom networks proposed for SAR-ATR, the size of the literature is significantly smaller than that for EO classification. A natural question arises as to how well the state-of-the-art EO networks designed for natural (optical) image classification perform on standard SAR ATR problems. This paper evaluates a number of well-known EO architectures (including DenseNet, ResNet, Inception and Xception) on a standard SAR ATR problem and identifies the factors that drive performance. We also perform a comparison to existing SAR-ATR networks in the literature. We recognize four important "pillars" of successful SAR ATR: data, architecture, augmentation and preprocessing. While the first two are well studied, the latter two are also of critical importance. In fact, we find that off-the-shelf EO networks can perform well on SAR - ATR with appropriate preprocessing and data augmentation.**

Keywords — synthetic aperture radar, machine learning, artificial intelligence, classification, identification

I. Introduction

Deep Convolutional Neural Networks have enjoyed widespread success in visual image classification problems [1]–[7], most famously using the ImageNet database [8] in a series of large scale visual recognition challenges [9]. Synthetic Aperture Radar (SAR) Automatic Target Recognition (ATR) is a related problem which primarily aims to classify man-made objects such as vehicles or aircraft using the radar modality instead of an optical sensor. While there have been some Deep Learning architectures networks proposed specifically for SAR-ATR [10]–[15], the size of the literature is significantly smaller than that focused on the electro-optical (EO) application.

A natural question arises as to how well networks designed for optical image classification perform in standard SAR ATR problems. Furthermore, it is of interest to determine what modifications are necessary to the architecture, data augmentation, or data preprocessing stages to unleash the utility of these networks in the SAR ATR problem.

This paper evaluates a number of well-known EO architectures – including DenseNet [1], ResNet [2], EfficientNet [3], Inception [5], MobileNet [6], and Xception [7] – on a standard SAR ATR dataset called SAMPLE [16] and identifies the factors that drive performance. SAMPLE is a publicly available dataset which contains both synthetically generated and collected SAR images ("chips") of military vehicles that have been ground truthed and have been studied elsewhere in the literature [17].

Our study identifies four important "pillars" of successful SAR ATR – the data, the network architecture (or model), the data augmentation algorithm, and the preprocessing step. While the first two factors have been studied, the latter two factors are also of critical importance to train a functioning Deep Learning based SAR ATR algorithm. In fact, we find that off-the-shelf EO networks can perform well on SAR ATR when we use appropriate preprocessing and data augmentation.

This paper proceeds as follows. First, in Section II, we describe the training and test data used for our studies. Next, in Section III we enumerate the network architectures we use in the study. Third, Section IV discusses the SAR-tailored set of data augmentations we use during training. Fourth, Section V describes the preprocessing methods typically used by EO networks and two preprocessing techniques specialized to SAR. Finally, Section VI shows the results of our extensive study which investigates more than thirty network architectures and six different preprocessing methods by capturing their performance statistically. These results show that many EO architectures can achieve fairly good ATR performance when coupled with the write preprocessing scheme. Finally, Section VII provides a conclusion and some comments.

II. Training and Testing Data

Our experiments use the Synthetic and Measured Paired and Labeled Experiment (SAMPLE) dataset [16] which has been recently released by the Air Force Research Laboratory (AFRL). SAMPLE includes a publicly available SAR dataset that consists of 10 target classes of collected data from the MSTAR flight test [18] and a recently created matching set of synthetic data. All images are "chips" of target-centered data of size 128 × 128. The synthetic data is created using CAD models of the target chips and a ray-tracing approach to provide a fully synthetic set that matches the collected chips in azimuth, elevation, and target mode. Figure 1 shows an example chip from each class.

This material is based upon work supported by the Air Force Research Laboratory (AFRL) under Contract No. FA8651-20-C-0046. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the AFRL.

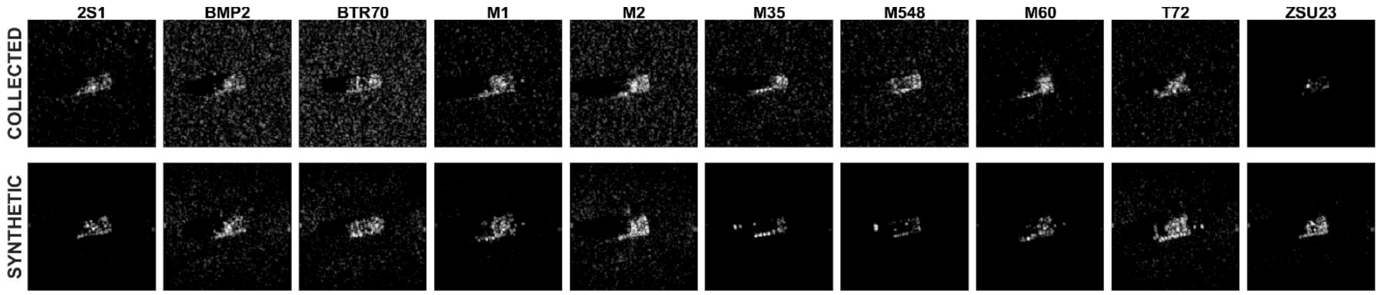

Fig 1. Example collected and synthetic chips from the 10-class SAMPLE dataset. Images are detected and displayed log-scale with 30dB of dynamic range.

We use the synthetic set as training chips and the collected set as validation chips as is typically done in the literature [19]. Ultimately, all inputs to our network are magnitude-detected (i.e., we use the absolute value of the pixels, discarding the phase). Recent work [20] has studied using complex data with SAMPLE, finding that for the networks they investigated, complex-data networks did not significantly outperform magnitude-only networks. Fully exploiting the complex SAR data in ATR continues to be an open area of study.

In addition, we have elected to divide the synthetic and collected sets in accordance with [17] where the training data comes from elevations $14° - 16°$ and the test data is at $17°$ elevation. As a result, there are a total of 806 training chips and 539 test chips.

III. Network Architectures

Our study includes a suite of broadly cited image classification networks from the literature, many of which have implementation in the deep learning API for Python called Keras. We include DenseNet [1], ResNet [2], EfficientNet [3], [4], Inception [5], MobileNet [6], and Xception [7]. Each of these architectures includes a collection of particular architectures (e.g., DenseNet129, DenseNet169 and DenseNet201) which represent variations in architecture.

Image classification architectures typically expect 3-channel (RGB) data as input whereas the (detected) SAMPLE SAR data is single-channel. The two obvious ways of dealing with this are channel blanking (i.e., putting the detected SAR data into the R channel and zeros into the G and B channels) or channel stuffing (i.e., replicating the detected data in all three channels). A preliminary investigation we performed indicated no major performance difference between the two approaches and so for this study we elected to do channel blanking. More sophisticated approaches, such as those that make use of polarization, complex data, or apply pre-filters to give the network three looks at the data simultaneously are of interest but beyond the scope of this study.

In addition to these image classification networks, we also compare the performance of published algorithms that have been developed specifically for the purpose of SAR-ATR, including AConvNets [10], LM-BN-CNN [11], MorganNet [12], TemplateNet [13], and SMPL [16][1]. This comparison is a second reason for using simple channel blanking for the image classification architectures because more sophisticated prefiltering would confound the comparison between the different classes of algorithms.

IV. Data Augmentation

As commonly done, we apply data augmentation to the training set at each stage of the training. This is a well-known approach to increase robustness in the learned model. In our experiments, we apply a SAR-specific set of data augmentations. We find that successful SAR ATR networks are enabled by augmentation approaches that are tailored to the unique aspects radar that distinguish it from EO.

One property of SAR that differentiates it from EO is that it is not invariant under rotation. In contrast to optical images, which can be rotated arbitrarily and produce realistic images, SAR images must respect the illumination direction. This stems from the specular nature of SAR data. While EO measured data is often well-modeled as diffuse scattering, meaning the reflected energy is (roughly) independent of the illumination direction (the sun or other light source), SAR data exhibits specular scattering meaning the reflected signal is strongly dependent on the direction of the transmitted signal.

A second distinguishing property of SAR – especially when used with man-made objects – is the large dynamic range. In contrast to EO imagery which is displayed meaningfully on a linear scale, typically SAR imagery is viewed on a log scale or with a quantization-based remapping.

Furthermore, noise and clutter in the SAR modality is best modeled as Rayleigh or Weibull, rather than Gaussian noise. Finally, various timing and positioning errors manifest themselves in phase errors which are best modeled in the complex imagery.

With this as background, we perform the following "SAR specific" steps in our data augmentation process:

- **Random Shift**. Training chips are randomly translated in the horizontal and vertical dimensions (range and cross-range) to model the imperfect centering of the collected data. The shift is a uniform random variable with maximum shift of 10% of the image size.

[1] [16] includes a description of the dataset, a network architecture we refer to as SMPL, and a preprocessing technique we refer to as "clipping".

- **Random Phase Error**. Training Chips have a quadratic phase error (QPE) added to model residual defocus possibly present in the collected data. The QPE added to each chip is selected from a Gaussian random variable with $\sigma = 150°$.
- **Random Target to Clutter Ratio**. Training chips have Rayleigh clutter added to model the background clutter present in the collected chips. For each chip, we first randomly select the desired target to clutter ratio (TCR). We elected to draw the TCR from a Gaussian with mean $0dB$ and standard deviation of $3dB$. We then add random Rayleigh clutter scaled to achieve this target TCR.
- **Random Pixel Swapping**. A subset of the top-N pixels are amplitude swapped with a neighbor. This models small differences between collected and training data either due to actual differences between the physical vehicle being collected and the CAD model or small differences in recorded viewing direction from actual viewing direction. We have elected to perform this perturbation on each pixel with probability 0.08, and when a pixel is selected, a random neighbor pixel is selected and its amplitude is used in place of the current pixel.

V. PREPROCESSING

We investigate a number of preprocessing approaches, including the standard preprocessing methods in image classification efforts (i.e., de-meaning and scaling) and methods tailored to the SAR modality. As mentioned earlier, one important difference between SAR data and EO data is that SAR data includes a dramatically larger range of pixel values, typically spanning four or five decades.

We find empirically that networks trained using preprocessing methods that are insensitive to the large variation in SAR pixel magnitudes (such as the simple scaling often employed with natural image classification) perform poorly as the weights tend to be overly swayed by a small number of large-amplitude image pixels. In contrast, preprocessing methods which are sensitive to large amplitude spreads such as quantization and clipping lead to more effective networks.

The preprocessing techniques we consider are:

- **DenseNet-style** [1] – Input chips are simply scaled to between 0 and 1.
- **ResNet-style (caffe)** [2] – Input chips are zero-centered with respect to the ImageNet dataset, i.e., first scaled to be between 0 and 255, then set to have the same mean as the "R" channel in ImageNet.
- **Inception-style** [5] – Input chips are simply scaled between -1 and 1.
- **Top-N** – Input chips are preprocessed by keeping only the $N = 5000$ highest-amplitude pixels (the remaining are set to zero). Chips are then scaled between 0 and 1.
- **Clipping** [16] – Input chips are preprocessed by first clipping large amplitude pixels (identified by an outlier test) and then setting all with amplitude more than $d = 64dB$ below the maximum to zero. Finally, the data is scaled between 0 and 1.
- **Quantization** [13], [21]–[23] – Input chips are preprocessed by keeping the N highest amplitude pixels and quantizing to N_l levels. We elect to use $N = 400$ and $N_l = 6$ in these experiments.

VI. EXPERIMENTAL RESULTS

Our primary result is a performance comparison of the architectures discussed in Section III. These architectures include both image classification models and models developed specifically for the SAR problem. The comparisons are carried out using the SAMPLE data discussed in Section II, where the 10-class training data is synthetically generated and the testing data comes from an airborne collect. Of particular interest is the performance of the architectures with the different preprocessing approaches discussed in Section V.

The performance of a trained network on a test dataset is inherently stochastic, stemming from the random batching and data augmentation that happens at every training epoch. Therefore, our approach to characterizing the performance of a network will be to run T trials where each trial fully trains the network for E epochs. We then report the mean performance and standard deviation (over the T training episodes) at final epoch E. Here we use $E = 60$ epochs and $T = 100$ trials, consistent with what is done elsewhere [17].

Figures 2, 3, and 4 illustrate the SAR-ATR classification performance using the ResNet, Densenet, *Ception, and EfficientNet family of architectures with the different preprocessing approaches described in Section V. Each bar is centered at the mean validation performance and is one standard deviation tall. Broadly speaking, we find SAR-tailored preprocessing approaches (primarily quantization but usually clipping as well) generate superior performance over those that use EO-type preprocessing (i.e., simple scaling).

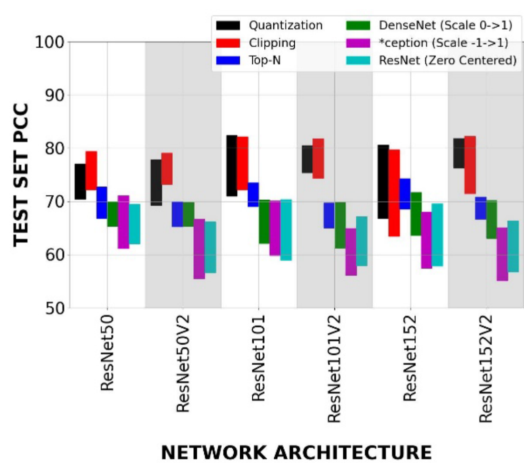

Fig 2. Test set Percent Correctly Classified (PCC) for six different ResNet configurations and the six different preprocessing approaches.

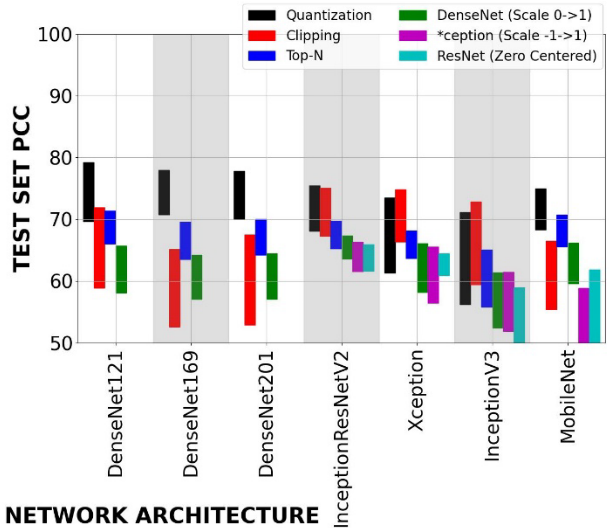

Fig 3. Test set Percent Correctly Classified (PCC) for DenseNet, Inception, Xception and MobileNet configurations and the six different preprocessing approaches.

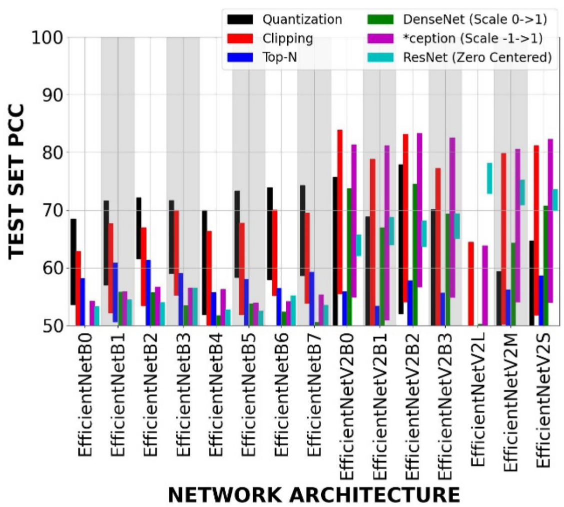

Fig 4. Test set Percent Correctly Classified (PCC) for fifteen different EfficientNet configurations and the six different preprocessing approaches.

Finally, Figure 5 summarizes the EO-network results from the proceeding figures by selecting one representative variant of each architecture and showing the performance with quantization preprocessing and DenseNet preprocessing. Furthermore, we also show the performance of published architectures that were designed specifically for SAR-ATR (TemplateNet [13], MorganNet [12], LM-BN-CNN [11], AconvNets [10], and SAMPLE [16]). Generally, we find that quantization-based preprocessing achieves the best performance. Additionally, while the SAR-specific networks perform best, EO networks using quantization preprocessing also perform credibly.

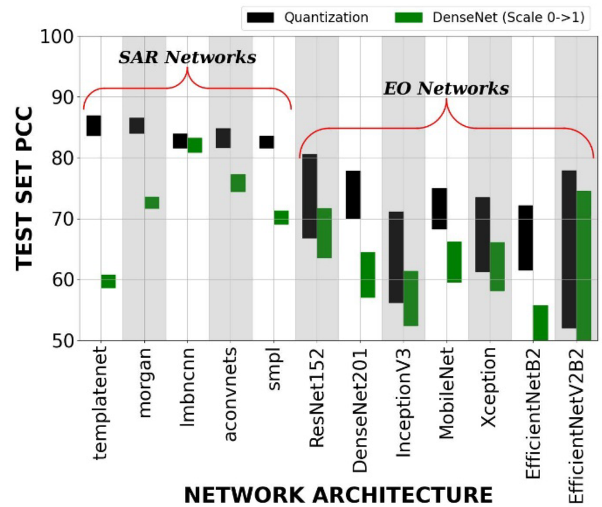

Fig. 5. Performance of networks designed for SAR-ATR and representative off-the-shelf EO networks with different preprocessing schemes. Generally, we find that quantization-based preprocessing achieves higher performance than simple scaling. Additionally, while the SAR-specific networks perform best, EO networks using quanization preprocessing also perform credibly.

VII. Conclusion

This paper studied the performance of Deep Learning approaches for SAR-ATR. Of particular interest here is the application of state-of-the-art EO architectures to the SAR problem. While it is well known that the model architecture and training data are strong drivers of performance, we find that SAR-tailored preprocessing and data augmentation play an crucial role algorithm performance. In fact, while SAR-specific networks provide the best performance, we find that many EO networks with appropriate front-end manipulations can perform fairly well on the SAR-ATR problem.

References

[1] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261–2269.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[3] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in Proceedings of the 36th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114.

[4] ——, “Efficientnetv2: Smaller models and faster training,” in Proceedings of the 38th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 10 096–10 106.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818–2826.

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

[7] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1800–1807.

[8] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” Neural Information Processing Systems, vol. 25, 01 2012.
[9] O. Russakovsky and J. Deng, “ImageNet large scale visual recognition challenge,” International Journal of Computer Vision, pp. 211–252, 2015.
[10] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, “Target classification using the deep convolutional networks for SAR images,” IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 8, pp. 4806–4817, 2016.
[11] F. Zhou, L. Wang, X. Bai, and Y. Hui, “SAR ATR of ground vehicles based on LM-BN-CNN,” IEEE Transactions on Geoscience and Remote Sensing, vol. 56, no. 12, pp. 7282–7293, 2018.
[12] D. A. E. Morgan, “Deep convolutional neural networks for ATR from SAR imagery,” in Algorithms for Synthetic Aperture Radar Imagery XXII, E. Zelnio and F. D. Garber, Eds., vol. 9475, International Society for Optics and Photonics. SPIE, 2015, pp. 116 – 128.
[13] M. P. Masarik, C. Kreucher, K. Weeks, and K. Simpson, “End-to-end ATR leveraging deep learning,” in Synthetic Aperture Radar (SAR) Data Applications, M. Rysz, P. Pardalos, A. Tsokas, K. Dipple, and K. Fair, Eds. Springer, 2023.
[14] M. Wilmanski, C. Kreucher, and J. Lauer, “Modern approaches in deep learning for SAR ATR,” in Proceedings of the 2016 SPIE Defense, Security, and Sensing Symposium, vol. 9843, 2016, pp. 98 430N–98 430N–10.
[15] Y. Li, J. Wang, Y. Xu, H. Li, Z. Miao, and Y. Zhang, “Deep SAR-net: Deep convolutional neural networks for sar target recognition,” in 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), 2017, pp. 740–743.
[16] B. Lewis, T. Scarnati, E. Sudkamp, J. Nehrbass, S. Rosencrantz, and E. Zelnio, “A SAR dataset for ATR development: The Synthetic and Measured Paired Labeled Experiment (SAMPLE),” in Algorithms for Synthetic Aperture Radar Imagery XXVI, E. Zelnio and F. D. Garber, Eds., vol. 10987, International Society for Optics and Photonics. SPIE, 2019, pp. 39 – 54.
[17] N. Inkawhich, M. J. Inkawhich, E. K. Davis, U. K. Majumder, E. Tripp, C. Capraro, and Y. Chen, “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, vol. 14, pp. 2942– 2955, 2021.
[18] T. Ross, S. Worrell, V. Velten, J. Mossing, and M. Bryant, “Standard SAR ATR Evaluation Experiments using the MSTAR Public Release Data Set,” SPIE Conference on Algorithms for Synthetic Aperture Radar Imagery V, vol. 3370, no. April 1998, pp. 566–573, 1198.
[19] T. Scarnati and B. Lewis, “A deep learning approach to the Synthetic and Measured Paired and Labeled Experiment (SAMPLE) challenge problem,” in Algorithms for Synthetic Aperture Radar Imagery XXVI, E. Zelnio and F. D. Garber, Eds., vol. 10987, International Society for Optics and Photonics. SPIE, 2019, p. 109870G.
[20] ——, “Complex-valued neural networks for synthetic aperture radar image classification,” in 2021 IEEE International Radar Conference. IEEE, May 2021, pp. 1–6.
[21] M. S. Horvath and B. D. Rigling, “Multinomial pattern matching revisited,” in Algorithms for Synthetic Aperture Radar Imagery XXII, E. Zelnio and F. D. Garber, Eds., vol. 9475, International Society for Optics and Photonics. SPIE, 2015, pp. 137 – 150.
[22] W. W. Irving and G. J. Ettinger, “Classification of targets in synthetic aperture radar imagery via quantized grayscale matching,” in Algorithms for Synthetic Aperture Radar Imagery VI, vol. 3721. International Society for Optics and Photonics, 1999, pp. 320–331.
[23] K. Hansen, “Updated algorithms for peaky template matching,” tech. rep, Sandia National Labs, Tech. Rep., 1994.