A Multi-view CNN for SAR ATR

Katherine M. Banas, Chris Kreucher
KBR Government Solutions
900 Victors Way, Suite 220, Ann Arbor, MI, 48108, USA
kmbanas@umich.edu, ckreuche@umich.edu

***Abstract*—This paper describes an approach for exploiting 3D information to improve synthetic aperture radar (SAR) automatic target recognition (ATR) performance. Historically, SAR ATR is performed using a single look, where an individual SAR image formed over a particular azimuth dwell is used as input to a classification algorithm. Recent data collection and processing developments enable techniques which exploit information from multiple looks simultaneously. Using more information from distinct geometries promises to improve both algorithm robustness and overall classification performance. Our approach to using this additional information is to adopt techniques for 3D image recognition from the natural image literature and adapt them to the SAR modality. We find SAR ATR performance can be improved substantially using a multi-view network employing a backbone architecture known to perform well on SAR ATR and enhanced with SAR-specific preprocessing and augmentation.**

***Index Terms*—Synthetic Aperture Radar, Automatic Target Recognition, 3D, Classification, Identification, AI/ML techniques**

I. Introduction

This paper describes an approach to using 3D information to improve synthetic aperture radar (SAR) automatic target recognition (ATR). Typically, SAR ATR is done using single-look data, where a single 2D (range versus cross-range) image is presented to an algorithm for a classification call. Modern data collection systems have increasing ability to georegister data, which means that it is straightforward to identify multiple looks at a single unknown object on the ground and provide an algorithm with far more than a single look at a target before requiring classification. This ability promises to dramatically improve classification performance – not only due to averaging random instantiations, but also stemming from the fact that different collection geometries will reveal different target features.

There is a large body of related research in the natural image domain (also referred to as the electro-optical, or EO domain) which aims to exploit the 3D structure of objects for improved classification. Qi [1] divides deep learning-based 3D ATR methods into two categories: view-based and model-based. View-based methods classify a 3D shape given a collection of 2D projections ("views" or "images"), whereas model-based approaches use a direct 3D representation of the object such as a point cloud [2], voxel grid [3], or mesh [4].

We find a view-based approach appealing for SAR ATR for two reasons: first, the radar community has compiled large datasets of collected and simulated 2D SAR images (e.g., MSTAR [5] and SAMPLE [6]); second, the EO literature has found [7] that building classifiers of 3D objects from 2D image projections provides superior generalization over direct 3D (voxel, points) representations, likely due to the maturity of convolutional neural network (CNN) algorithms on the 2D image classification problem.

The MVCNN (Multi-view CNN) algorithm [8] is a seminal work describing a view-based CNN for multi-look classification. The method was originally demonstrated on the ModelNet [9] dataset, a standard challenge problem in 3D EO classification. MVCNN employs a backbone 2D image classification network and partitions it into the first N_1 layers, which extract features from individual views, and the final N_2 layers, which output a class prediction given the aggregated features from the set of all views. The features extracted from each view are aggregated by a view-pooling layer (i.e., max-pooling over the view dimension) before being passed to the final set of layers. One obvious approach to combining views is to simply average the class scores predicted for each view by a 2D CNN. MVCNN improves upon this naïve method with view-pooling and additional trainable weights after the view-pooling layer, as these provide the ability to select features from only the most informative views and learn from the dependencies across a full set of views.

Other methods [10]–[12] are analogous to MVCNN in that they utilize a backbone network and then later combine the individually extracted features. They are distinguished from MVCNN in that they typically group the individual view features using more sophisticated methods than view-pooling. In particular, view-GCN [13] represents multiple views using a view graph defined by the look angle of each view. It combines single-view features via local and non-local graph convolution and selective view sampling, which explicitly consider the spatial relationship between views and discriminability of individual views.

SAR imagery collected from airborne platforms will typically be unevenly spaced in azimuth and elevation, including gaps in coverage. While view-GCN's view aggregation carefully considers spatial relationships among views and can therefore prevent some of the information loss possible with MVCNN's harsher max-pooling aggregation, view-GCN is highly dependent on view order and geometric configuration whereas MVCNN is totally flexible. Thus, we expect MVCNN to perform better on multi-view SAR data with variable sensor geometry and have chosen to use it here. We have elected to use the AConvNets [14] model as our backbone due to its robust performance on 2D SAR datasets, and as such refer to our approach as MVAConvNets.

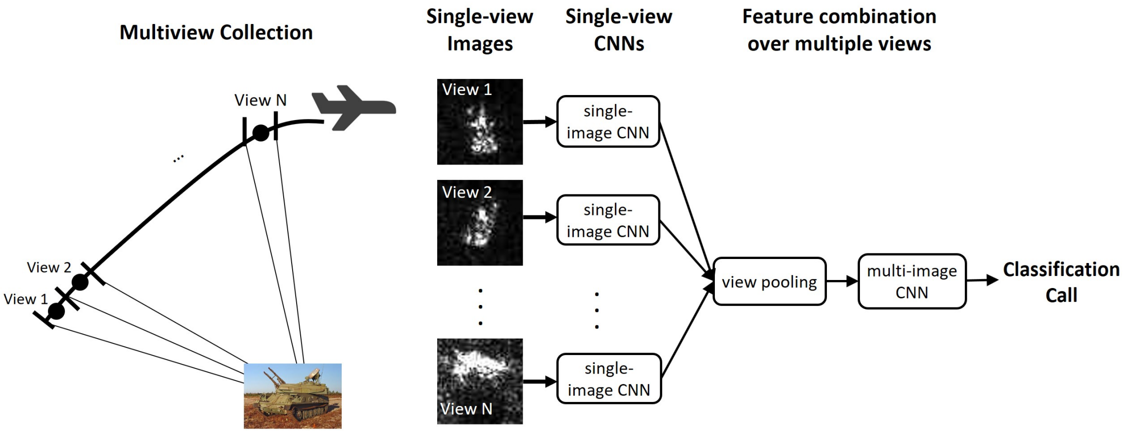

Fig. 1. The multi-view CNN approach to SAR ATR, adapted from [8]. The network operates as follows. First, the collection platform forms SAR imagery of a target at multiple views. Next, features are extracted from each view using a backbone single-image CNN block. Finally, the single-view features are pooled and passed through a multi-view CNN block to generate a classification call.

The rest of this paper proceeds as follows. First, Section II describes our technical approach, which is to adopt the MVCNN framework, combine it with a backbone architecture shown to work well on SAR problems, and layer in SAR domain-specific data preprocessing and data augmentations. Next, Section III describes a set of experiments which train on synthetic data and test on collected data. These experiments illustrate the utility of multi-view SAR ATR as well as explore the space of parameters that must be selected to instantiate the algorithm. Finally, Section V summarizes results and concludes the paper.

II. Technical Approach

This section describes our technical approach to combining multiple SAR views into a single classifier. We start with the MVCNN architecture, which we briefly outline in subsection II-A. Subsection II-B lays out how we use the AconvNets [14] model as our backbone network as it has been shown to work well with single-view SAR ATR data. Finally, subsection II-C describes the mechanisms we use for training the classifier which respect the unique characteristics of SAR data.

A. Overview of MVCNN

The Multi-view Convolutional Neural Network (MVCNN) algorithm [8] combines multiple 2D projections (“views”) of an object using a two-stage process. The first stage operates on single views, utilizing the first N_1 layers of a backbone network to extract features from individual images. In this stage, the same CNN layer weights are applied to all views. Individual view features are then combined with an element-wise max pooling step across the views. Finally, the pooled features are sent through the remaining CNN layers, which produce an output classification call on the collection of images. In our implementation, we use all but the last layer of AConvNets for feature extraction, and the final layer for classification after view-pooling. We also investigated the effect of performing view-pooling at earlier layers, but we saw empirically that performance was not very sensitive to where view-pooling was done.

We train the algorithm using purely synthetic data and test it using collected data. Training occurs in two stages. First, all layers of the backbone network are trained using a large collection of simulated single-image data using the ordinary approach to training single-view CNNs. Next, a view-pooling layer is inserted between the first N_1 layers and final N_2 layers of the backbone network, and the training process is resumed on groups of multiple views. We allow all weights to be updated in the second stage of training. Our nominal configuration is to use 12 views at a common elevation spaced $30°$ apart, but we perform trade studies varying these parameters later. We have also found it useful to add noise to the azimuth angles used in each view (i.e., to use azimuth angles that differ a bit from the nominal ones) and to perform dropout on entire view feature vectors prior to the view-pooling layer; we hypothesize that these augmentations add robustness when collected datasets have uneven angular spacing and missing views or views with poor discriminability.

At test time, the fully trained network operates on groups of collected images. Figure 1 is an illustration, adapted from [8], which graphically describes how set of images from a collection is classified. First, N image views are formed from the aperture flown by the sensor. Next, each image view is passed through the single-image CNN block, and those features are combined at the view-pooling step. Finally, the multi-image CNN block uses these inputs to make a classification call.

B. Backbone Network

Before combining views, the multi-view approach derives features from a CNN for each view which is referred to as the “backbone” network. In the EO multi-view literature some of the backbone networks used include VGG [15] and ResNet [16], [17]. These were selected because they perform well on the single-image EO classification problem. We elected to use the AConvNets architecture [14] for our models. AConvNets has shown excellent performance on single-view collected SAR data [14], [18], [19]. The network is trained using the cross-entropy loss with softmax activation and the ADADELTA optimizer.

C. SAR-Specific Data Treatment

Most multi-view (and even single-view) deep learning techniques in the literature are applied to EO data. As such, the large body of published deep learning algorithms typically include preprocessing and data augmentation steps tailored to EO. The SAR modality has a number of properties that differentiate it from EO which must be addressed in the algorithm design for a robust, high-performing network. These factors are of particular importance when using synthetic training data. We briefly review the factors and our approach here.

Preprocessing: Preprocessing refers to a manipulation of the input data which standardizes the input stream and changes the scale to allow a network to more easily learn features from the data.

As EO data is 3-channel (RGB) with features that are meaningfully captured on a linear scale, EO networks preprocess data with this in mind. Popular approaches to EO preprocessing include linear scaling from 0 to 1 [20], zero-centering and scaling between 0 and 255 [16], and scaling between -1 and 1 [21].

In contrast, SAR data has a dramatically larger range of pixel values, typically spanning four or five decades. Therefore, learning algorithms using SAR data benefit from preprocessing which prevents a small number of pixels from dominating, while still maintaining the descriptive information of all pixels. Researchers have applied a number of methods in this aim, including binarization [22], quantization [23], [24], keeping only the top N largest-magnitude pixels [25], clipping [6], as well as approaches which employ sophisticated morphological operations and speckle reduction [26].

We have elected to use a quantization approach [27], where we keep the top 400 pixels and quantize the data into 6 levels.

Data Augmentation: Data augmentation is an approach where training samples are randomly perturbed to produce additional samples which reflect the variation expected at test time. This is well-known to increase robustness in the learned model.

EO data augmentation includes image translations, rotations, and scaling to model the fact that similar images at different angles and scale factors should be identically classified. In addition, EO data can be augmented with noise modeled to match what would be seen in the imaging plane.

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 fact that though EO data can be well-modeled as diffuse scattering, meaning that the reflected energy is roughly independent of the illumination direction (from 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. Therefore, SAR data cannot be simply rotated or stretched to produce realistic additional samples. Furthermore, noise and clutter in the SAR modality is best modeled as Rayleigh or Weibull, rather than Gaussian. Finally, various timing and positioning errors manifest themselves in phase errors which are best modeled in complex-valued imagery.

With this as background, we perform the following *SAR-specific* steps [28] in our data augmentation at training time: we add random quadratic phase error of maximum magnitude 150°, add Rayleigh noise to achieve a 0 dB target to clutter ratio, randomly select 400 of the top 600 pixels to preprocess and zero the rest, randomly swap adjacent pixels with probability 0.08, and randomly perturb the image azimuth by up to one-tenth of the azimuth spacing between views.

III. Results

This section describes a set of experiments with MVAConvNets to demonstrate improved performance by combining multiple image views.

We performed experiments using two datasets: (i) the Moving and Stationary Target Acquisition and Recognition (MSTAR) [5] collected data coupled with a synthetic dataset we generated using CAD models and ray-tracing software, and (ii) the Synthetic and Measured Paired and Labeled Experiment (SAMPLE) dataset [6], a set of paired collected and synthetic data released by the Air Force Research Laboratory (AFRL). We first describe the training and testing data, and then move to the experiment and its results.

A. MSTAR Dataset

The MSTAR [5] dataset is a publicly available collected SAR image dataset widely used in the ATR literature. 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. All images are "chips" of target-centered data of size 128×128. 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 instead. We evaluate the performance on the 3203 collected chips from the MSTAR-15 dataset.

The synthetic SAR dataset we use for training was generated using a 3D CAD model of each MSTAR target via 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. We synthesized data from all 360° of azimuth at 1° spacing.

B. SAMPLE Dataset

The Synthetic and Measured Paired and Labeled Experiment (SAMPLE) dataset [6] is a related SAR dataset 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 [5] and a matching set of synthetic data created by AFRL. 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.

We use the synthetic set as training chips and the collected set as validation chips as is typically done in the literature [29]. In addition, we have elected to divide the synthetic and collected sets in accordance with [30], where the training data comes from elevations $14° - 16°$ and the test data is at $17°$ elevation. The publicly released SAMPLE dataset restricts azimuth to between approximately $10°$ and $80°$ azimuth. As a result, there are far fewer test chips in the dataset, which contains a total of 806 training chips and 539 test chips.

IV. Experiments

We first show the results of an experiment using MVAConvNets with the MSTAR dataset. We ran the first (i.e., single-view) stage of training for 25 epochs using individual synthetic images. We determined empirically that this was enough epochs for the training set accuracy to nearly reach its peak without overfitting. At each epoch, we recorded the performance of the trained model against the collected test set. After the single-image network was fully trained, we trained the second stage network, which combines N_{views} images together for classification, for an additional 25 epochs. The network was evaluated on the test set using collections of N_{views} chips of the same target with adjacent views spaced ΔAz degrees apart. Since the test set is not perfectly spaced in azimuth, we simply took the closest image to the desired angle. Figure 2 shows the performance of the synthetically-trained network against the collected data at each epoch in each training stage. This is our baseline configuration with $N_{views} = 12$ and $\Delta Az = 30°$. We ran the experiment 50 times and show one standard deviation error bars around the mean performance.

An important aspect of MVAConvNets' performance is its dependence on the number of views and the azimuth spacing between each view. We expect multiple views to be helpful for two reasons: (1) even in the limiting case where views are nearly at the same angle, multiple views can help make more accurate predictions by reducing variance in a statistical sense; and (2) in the case where views are spaced far apart, the geometric diversity of the view collection aids prediction by providing additional information (though the effect of variance reduction may be less pronounced as combining views is less like averaging). Figures 3 and 4 show performance when the azimuth spacing is reduced (Figure 3) and when both the azimuth spacing and number of views is reduced (Figure 4).

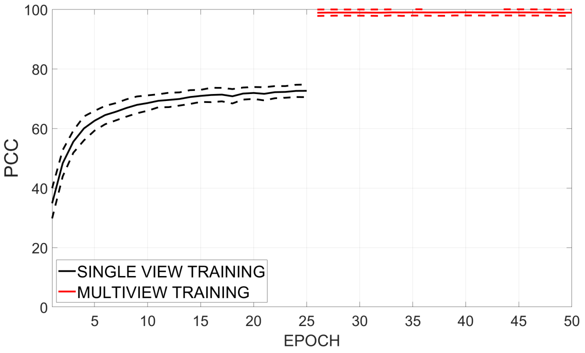

Fig. 2. The performance of MVAConvNets on the MSTAR synthetic vs. real experiment using baseline parameters ($N_{views} = 12, \Delta Az = 30°$). The solid lines show mean performance, and the dashed lines show the standard deviation. There are 50 trials. The first stage trains the single view algorithm. It asymptotes to approximately 70% probability of correct classification (PCC). The second stage aggregates over views, reaching more than 95% PCC.

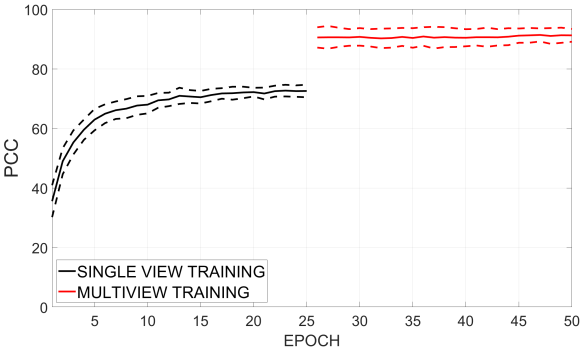

Fig. 3. The performance of MVAConvNets on the MSTAR synthetic vs. real experiment with 12 views as in the baseline but only using a $5°$ spacing of the views ($N_{views} = 12, \Delta Az = 5°$). We find the multi-image performance below 90%, approximately 6 points below the baseline.

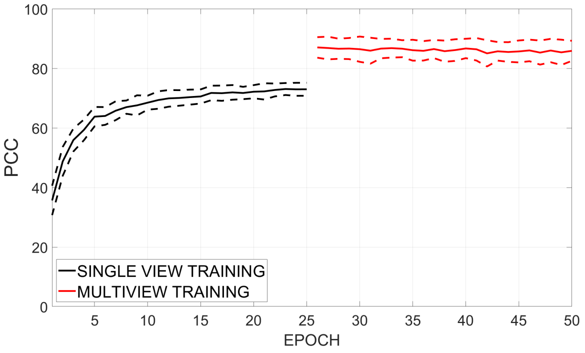

Fig. 4. The performance of MVAConvNets on the MSTAR synthetic vs. real experiment with 8 views, spaced $5°$ apart ($N_{views} = 8, \Delta Az = 5°$). The multi-image performance is just over 80%.

Figure 5 shows a grid search study of the performance of MVAConvNets versus the azimuth spacing and number of views, again averaged over 50 trials. We find that performance improves with more views and larger spacing between views. While this is intuitive, the figure is mainly useful for performance prediction, where an operator knows how many looks were collected and the azimuth spacing and can then predict

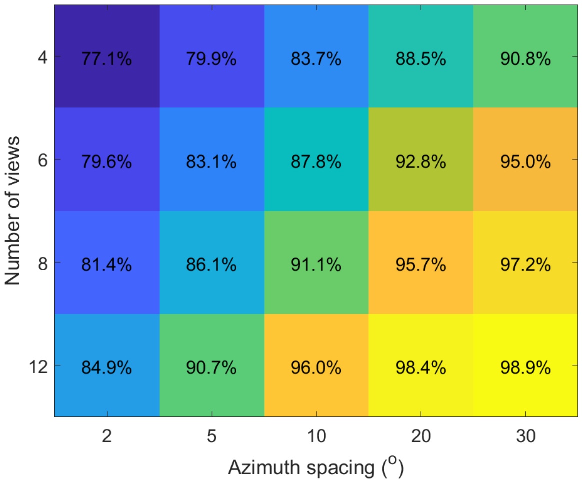

Fig. 5. The multi-image performance versus number of views and the spacing between views on the MSTAR synthetic vs. real experiment. The values in each cell are the test set correct classification probability, averaged over the final 25 training epochs. Broadly speaking, performance improves with more views and larger spacing between views.

in advance the utility of the algorithm. It is worth noting that even the view configuration with the least geometric diversity ($N_{views} = 4, \Delta Az = 2°$) yields a performance improvement over the baseline single-view CNN (see the single-view training curves in Figures 2 through 4).

A natural question arises as to whether algorithm performance simply depends on the azimuth span of the collected data (i.e., $N_{views} \cdot \Delta Az$). Figure 6 is a plot of classification performance versus total azimuth span, with curves for different numbers of views. It shows that the dominant factor in performance is in fact total azimuth span of the test set. A secondary effect is that generally speaking for a fixed total azimuth span, more looks are better.

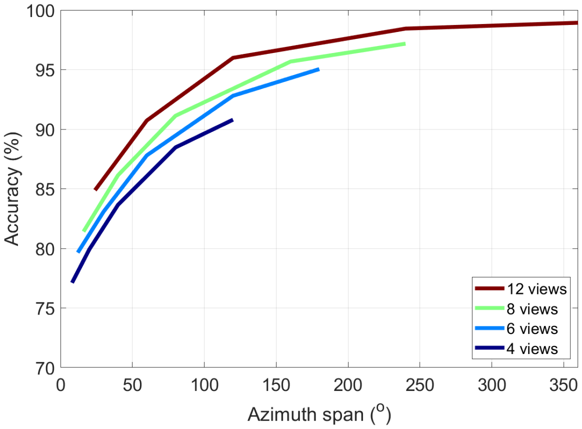

Fig. 6. Mean performance (PCC) on the MSTAR synthetic vs. real experiment over 50 Monte Carlo trials and the last 25 training epochs, versus total azimuth span (i.e., $N_{views} \cdot \Delta Az$). Each curve is for a different N_{views}. In general, performance improves as azimuth span increases, and as N_{views} increases for a fixed azimuth span.

It is important that our multi-view approach continues to perform acceptably when the data is not collected at a regularly spaced set of angles. Since we train on synthetic data, we continue to assume the training set has data at every 1° in azimuth as described in Section III-A. However, we now assume the test data has no such regularity. We selected 36 collected chips from the MSTAR set randomly. This resulted in chips with random azimuth gaps. On average, the spacing is 10°, but in practice we find gaps as large as 30°. We executed our trained algorithm on this new randomly thinned dataset and show the results in Figure 7. We find the performance on this set with non-uniform angles is similar to the fully populated dataset. There is a small performance reduction at low azimuth span.

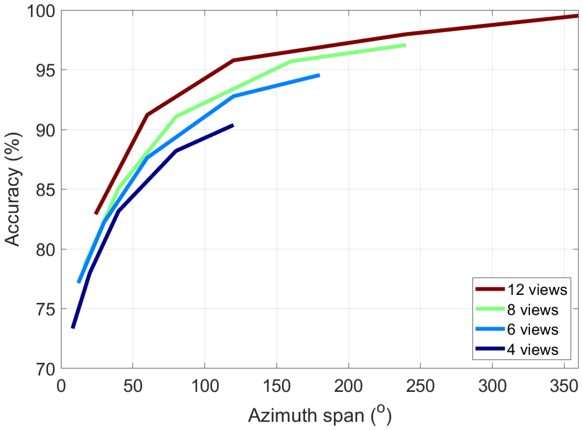

Fig. 7. Mean performance (PCC) on the MSTAR synthetic vs. real experiment over 50 Monte Carlo trials and the last 25 training epochs, versus total azimuth span (i.e., $N_{views} \cdot \Delta Az$) with a test dataset that has random gaps in azimuth spacing. The network performance is similar to the regularly spaced test data.

We performed a similar experiment using the SAMPLE dataset. The results are shown in Figures 8 and 9. Broadly speaking, we find similar behavior – performance improves with total azimuth extent, and for a fixed total azimuth extent more views are better. However, due to the limited azimuthal extent (10° to 80°) of the publicly released SAMPLE data and uneven number of samples at each azimuth we do not find an exactly monotonic relationship.

V. Conclusion

This paper described an approach to using 3D information for improved Synthetic Aperture Radar (SAR) Automatic Target Recognition (ATR). Our approach is inspired by multi-look networks developed for natural images, but also incorporates some SAR domain-specific components to specialize to the radar modality. We adopt a two-stage multi-view approach, where first a backbone network is trained on single image features, then these features are combined via view-pooling, and the final layers use the multi-input features to perform classification in the second stage. In experiments which use synthetic training data and collected test data, we find a substantial improvement in target classification performance using the multi-view network.

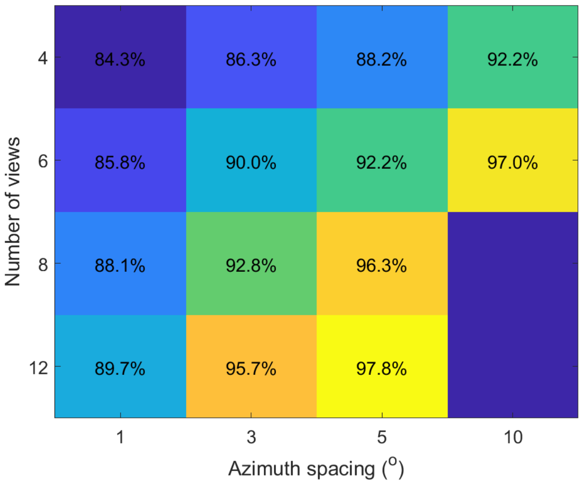

Fig. 8. The multi-image performance versus number of views and the spacing between views on the SAMPLE experiment. The values in each cell are the test set correct classification probability, averaged over the final 25 training epochs. Because of the limited azimuth extent (i.e., 10° to 80°) in the SAMPLE dataset, some $(N_{views}, \Delta Az)$ combinations could not be tested.

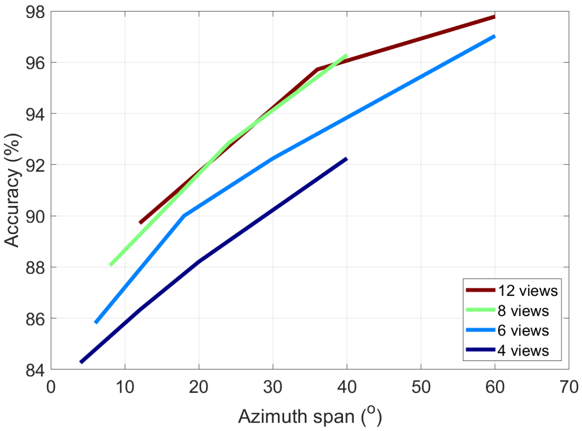

Fig. 9. Mean performance (PCC) of MVAConvNets over 50 Monte Carlo trials and the last 25 training epochs, versus total azimuth span (i.e., $N_{views} \cdot \Delta Az$) of the views for the SAMPLE dataset.

References

[1] S. Qi, X. Ning, G. Yang, L. Zhang, P. Long, W. Cai, and W. Li, "Review of multi-view 3d object recognition methods based on deep learning," *Displays*, vol. 69, 2021.

[2] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics*, vol. 38, pp. 1–12, 2019.

[3] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 922–928.

[4] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spheirical harmonic representation of 3d shape descriptors," in *Symposium on Geometry Processing*, 2003, pp. 156–164.

[5] 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.

[6] 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*. SPIE, 2019, pp. 39–54.

[7] J.-C. Su, M. Gadelha, R. Wang, and S. Maji, "A deeper look at 3d shape classifiers," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshop*, 2018.

[8] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015.

[9] [Online]. Available: https://modelnet.cs.princeton.edu/

[10] J. Pei, Y. Huang, W. Huo, Y. Xue, and Y. Zhang, "Multi-view SAR ATR based on networks ensemble and graph search," in *2018 IEEE Radar Conference (RadarConf18)*, 2018, pp. 355–360.

[11] X. Zhang, J. Pei, Y. Ma, and Q. Yi, "Multiview feature extraction and discrimination network for SAR ATR," in *IEEE International Geoscience and Remote Sensing Symposium*, 2023, pp. 7042–7045.

[12] Z. Han, M. Shang, Z. Liu, C.-M. Vong, Y.-S. Liu, M. Zwicker, J. Han, and C. L. P. Chen, "Seqviews2seqlabels: Learning 3d global features via aggregating sequential views by rnn with attention," *IEEE Transactions on Image Processing*, no. 2, pp. 658–672, 2019.

[13] X. Wei, R. Yu, and J. Sun, "View-gcn: View-based graph convolutional network for 3d shape analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1850–1859.

[14] 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.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:14124313

[16] 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.

[17] R. J. Soldin, "SAR target recognition with deep learning," in *2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 2018, pp. 1–8.

[18] G. Dong, N. Wang, and G. Kuang, "Sparse representation of monogenic signal: With application to target recognition in sar images," *IEEE Signal Processing Letters*, pp. 952–956, 2014.

[19] A. W. Denton and D. A. Garren, "Deep-layer training of cnn for sar with two-stage data augmentation," in *2023 IEEE Radar Conference (RadarConf23)*, 2023, pp. 1–6.

[20] 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.

[21] 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.

[22] K. Hansen, "Updated algorithms for peaky template matching," tech. rep, Sandia National Labs, Tech. Rep., 1994.

[23] 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, 1999, pp. 320–331.

[24] 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. SPIE, 2015.

[25] K. Banas, T. Hill, C. Kreucher, B. Raeker, K. Simpson, and K. Weeks, "Novel view synthesis with compressed sensing as data augmentation for SAR ATR," in *Proceedings SPIE DCSC*, 2023.

[26] G. F. Araujo, R. Machado, and M. I. Pettersson, "Assessment of preprocessing techniques in a model-based automatic target recognition algorithm for the sample dataset," in *Image and Signal Processing for Remote Sensing XXVIII*. SPIE, October 2022.

[27] 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, 2022, ch. 1, pp. 1–23.

[28] C. Kreucher, "SAR-ATR using EO-based deep networks," in *2023 IEEE Radar Conference (RadarConf23)*. IEEE, 2023.

[29] 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. SPIE, 2019.

[30] N. Inkawhich, M. J. Inkawhich, E. K. Davis, and U. K. M. et. al, "Bridging a gap in SAR-ATR: Training on fully synthetic and testing on measured data," *IEEE J. Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 2942–2955, 2021.