The Effect of Training Dataset Size on SAR Automatic Target Recognition Using Deep Learning

Weidong Kuang and Wenjie Dong
Department of Electrical and Computer Engineering
University of Texas Rio Grande Valley
1201 University Dr., Edinburg, TX 78539, USA
weidong.kuang@utrgv.edu

Liang Dong
Department of Electrical and Computer Engineering
Baylor University
One Bear Place, Waco, TX 76798, USA
liang_dong@baylor.edu

Abstract—**Synthetic aperture radar (SAR) is an effective remote sensor for target detection and recognition. Deep learning has a great potential for implementing automatic target recognition based on SAR images. In general, Sufficient labeled data are required to train a deep neural network to avoid overfitting. However, the availability of measured SAR images is usually limited due to high cost and security in practice. In this paper, we will investigate the relationship between the recognition performance and training dataset size. The experiments are performed on three classifiers using MSTAR (Moving and Stationary Target Acquisition and Recognition) dataset. The results show us the minimum size of the training set for a particular classification accuracy.**

Keywords-Synthetic Aperture Radar (SAR); Automatic Target Recognition (ATR); Convolutional Neural Networks (CNNs)

I. INTRODUCTION

Synthetic aperture radar (SAR) [1] can operate in all-weather day-and-night conditions and generate high resolution images of targets. Thus, SAR is particularly suitable for target recognition, reconnaissance, surveillance, etc. Due to scattering mechanism and speckle noise in SAR imagery, the interpretation and understanding of SAR images are very different from optical images. Recognizing a target in a SAR image by human eyes is time consuming and often unreliable. It is desirable to develop automatic target recognition (ATR) algorithms for SAR images.

Motivated by numerous successful applications in the computer vision community, deep learning [2] is now attracting wide attention in SAR remote sensing tasks [3]. However, SAR images are substantially different from optical images in many aspects. Compared to optical images, SAR images have the following special characteristics: large dynamic range, speckling noise, complex value for each pixel, and limited datasets available (due to high cost). Furthermore, SAR images are sensitive to radar parameters (e.g., wavelength, geometric parameters between radar and targets) and target postures. These characteristics limit the potentials of computer vision techniques (developed based on optical images) in SAR image understanding. Although deep learning-based frameworks in general are applicable to SAR images, it is crucial and challenging to address special issues specific to SAR images.

One of the most challenging issues is the availability of measured SAR image data because collecting SAR image data is prohibitively expensive. In this paper, we will investigate the effect of training dataset size on the classification accuracy. Specifically, we train the classifiers using a subset of the MSTAR training dataset, and then test the trained classifiers using the MSTAR test dataset. By changing the size of the subset, we can obtain the dependency of classification accuracy on the training dataset size.

II. MSTAR DATASET

The MSTAR benchmark data set [4] is widely used to test and compare the performance of SAR-ATR algorithms. MSTAR datasets were collected by the Sandia National Laboratory SAR sensor platform. The collection was jointly sponsored by the Defense Advanced Research Projects Agency and the Air Force Research Laboratory as part of the MSTAR program. Hundreds of thousands of SAR images containing ground targets were collected, including different target types, aspect angles, depression angles, serial number, and articulation, and only a small subset of which are publicly available on the website. The publicly released data sets include ten different categories of ground targets (armored personnel carrier: BMP-2, BRDM-2, BTR-60, and BTR-70; tank: T-62, T-72; rocket launcher: 2S1; air defense unit: ZSU-234; truck: ZIL-131; bulldozer: D7). They were collected using an X-band SAR sensor, with a one-foot resolution spotlight mode, full aspect coverage (in the range of 0° to 360°), illustrated in Fig.1. Examples of SAR images of ten types of targets and their corresponding optical images are shown in Fig. 2. Table I lists the number of images and the image size for each category at two different depression angles: 17° for training set and 15° for testing set, in the MSTAR dataset.

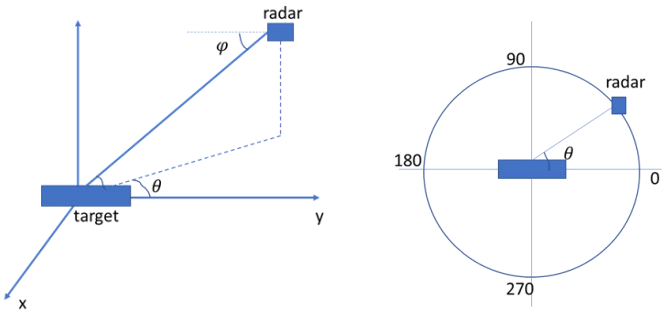

Fig.1 Geometry of SAR and top view of aspect angels: φ is depression angle and θ is aspect angle, φ=15°, 17°, 30°, 44°. θ was sampled from 0 to 360 for each depression angle. Images for φ=17° are used for training set while φ=15° for test set.

978-1-6654-0754-0/22/$31.00©2022 IEEE

2022 IEEE 12th International Conference on Electronics Information and Emergency Communication (ICEIEC) | 978-1-6654-0754-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICEIEC54567.2022.9835077

Authorized licensed use limited to: The University of Texas Rio Grande Valley. Downloaded on March 08,2023 at 15:38:23 UTC from IEEE Xplore. Restrictions apply.

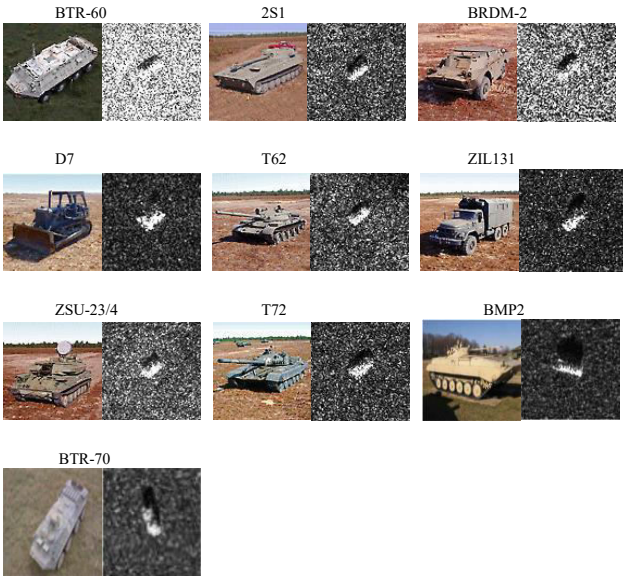

Fig.2 Examples of MSTAR images for ten targets

TABLE I. MSTAR DATASET AT STANDARD OPERATION CONDITION

Target	Standard Operation Condition			
	Depression (15°) (test)		Depression (17°) (train)	
	Image size	# images	Image size	# images
BMP-2	(128,128)	587	(128,128)	698
BRDM-2	(129,128)	274	(129,128)	298
BTR-60	(128,128)	195	(128,128)	256
BTR-70	(128,128)	196	(128,128)	233
T-62	(173,172)	273	(173,172)	299
T-72	(128,128)	582	(128,128)	691
2S1	(158,158)	274	(158,158)	299
ZSU-23-4	(158,158)	274	(158,158)	299
ZIL-131	(193,192)	274	(193,192)	299
D7	(178,177)	274	(178,177)	299
Total		3203		3671

III. CLASSIFIER ARCHITECTURES FOR SAR ART

To obtain a confident and reliable result, we selected three well recognized classifiers based on MSART dataset from literatures [5][6][7]. The architectures are described in Table II, Table III, and Table IV, respectively.

Classifier 1 consists of three convolution layers with ReLU activation function and a fully connected (FC) layer with softmax activation for 10 classes. The kernel sizes of three convolution layers are 9x9, 5x5, and 4x4, respectively. All convolution layers use a stride of 1, and no zero-padding. Max pooling is used to reduce the output data volume following the first two conv layers. In classifier 2, there are four conv layers, one FC layer with dropout for regularization, and the last FC layer with softmax for classification. All conv layers have the same kernel size (3x3) and one zero-padding. Classifier 3 is similar to classifier 2, except that classifier 3 has two more FC layers before the last FC (softmax) layer. Note that three classifiers have different input sizes and the numbers of conv channels.

TABLE II. CLASSIFIER 1 [5]

Layer	Layer name	Conv Kernel size	Output size
0	Input		128x128x1
1	Conv (ReLU)	9x9	120x120x18
	Max Pooling (6x6)		20x20x18
2	Conv (ReLU)	5x5	16x16x36
	Max Pooling (4x4)		4x4x36
3	Conv (ReLU)	4x4	1x1x120
	flatten		120
4	FC (softmax)		10

TABLE III. CLASSIFIER 2 [6]

Layer	Layer name	Conv kernel size	Output size
0	Input		48x48x1
1	Conv (ReLU)	3x3, padding=1	48x48x9
	Max pool (2x2)		24x24x9
2	Conv (ReLU)	3x3, padding=1	24x24x18
	Max pool (2x2)		12x12x18
3	Conv (ReLU)	3x3, padding=1	12x12x36
	Max pool (2x2)		6x6x36
4	Conv (ReLU)	3x3, padding=1	6x6x60
	Flatten		2160
5	FC (ReLU)		60
	Dropout (p=0.1)		
6	FC (softmax)		10

TABLE IV. CLASSIFIER 3 [7]

Layer	Layer name	Conv kernel size	Output size
0	Input		64x64x1
1	Conv (ReLU)	3x3, padding=1	64x64x16
	Max pool		32x32x16
2	Conv (ReLU)	3x3, padding=1	32x32x32
	Max pool		16x16x32
3	Conv (ReLU)	3x3, padding=1	16x16x64
	Max pool		8x8x64
4	Conv (ReLU)	3x3, padding=1	8x8x128
	Max pool		4x4x128
5	Flatten		2048
	FC (ReLU) Dropout (p=0.1 or 0.2)		1000
6	FC (ReLU)		500
7	FC (ReLU)		250
8	FC (softmax)		10

IV. EXPERIMENTS AND RESULTS

In this section, we will present the details of experiments and the results obtained. All classifiers described in Section III are implemented and trained in the PyTorch framework. To investigate the effect of training set size on the classification accuracy, we start with training a classifier using the full training set (i.e. 3671 examples), and measure the classification accuracy using the test set. Then we form a reduced (e.g. 75%) training set by randomly drawing examples from the full training set, and train the classifier from scratch using the reduced training set. The trained classifier will be tested by the same test set. We repeat this process for a different reduced

Authorized licensed use limited to: The University of Texas Rio Grande Valley. Downloaded on March 08,2023 at 15:38:23 UTC from IEEE Xplore. Restrictions apply.

training set with a different size (e.g. 50%). As a result, we can obtain a plot of the accuracy performance versus training set size.

A. Training Datasets

In the experiments, we train each classifier at six different dataset sizes. These training sets are defined in Table V. The full training set includes 3671 examples which are distributed in 10 classes, as shown in Table I. However, in the full MSTAR training set, the classes BMP-2 and T72 have significant larger number of examples (e.g. 698 and 691) than other classes (less than 300). Thus, to make the sample distribution across classes uniform in the training set, we reduce both the numbers of examples for BMP-2 and T72 to 299. The resulting training set has 2880 examples, and we define its relative size is "1.0". To generate a smaller training set (e.g. 0.75), we randomly draw a percentage (e.g. 75%) of examples from each class in the training set 1.0

TABLE V. TRAINING SETS WITH DIFFERENT SIZES

Training set relative size	# of examples	Note
1.2	3671	Entire train set in MSTAR
1.0	2880	Class balanced
0.75	2880x75%=2160	Draw 75% examples from each class in 1.0 size set
0.5	2880x50%=1440	Draw 50% examples from each class in 1.0 size set
0.25	2880x25%=720	Draw 25% examples from each class in 1.0 size set
0.1	2880x10%=288	Draw 10% examples from each class in 1.0 size set

B. Training Settings

Two preprocesses are performed on the SAR images in training set: 1) normalize each image by the set mean and standard deviation; and 2) center crop the image sizes (e.g. 193x192) to the input size (e.g. 64x64 for classifier 3) required by the corresponding classifier.

In the PyTorch framework, we use the following settings to train classifiers:

- Epochs: 20 or 30 or 60
- Batch size: 64 examples
- Weight initialization: Xavier uniform
- Optimizer: Adam, learning rate =0.01, or 0.001
- Learning rate schedule: step size=10, gamma=0.1.
- Dropout for regularization: p=0.1
- Loss: cross entropy

C. Results

To monitor the training process, it is desirable to plot the loss and classification accuracy as the training processing is going on. Since three classifiers deliver the similar loss and accuracy curves, we only present these plots for classifier 3 as examples. Fig.3 shows the loss during the training process with the full training set. Since the full training set has 3671 examples and the batch size is 64, each epoch includes 58 batches, i.e. the neural network weights are updated 58 times during one epoch. For display purpose, we average the loss over every 10 batches, and thus obtain 5 loss values for one epoch, and 150 loss values for the total 30 epochs during the entire training process. Note that the value of loss is plotted in log10() scale.

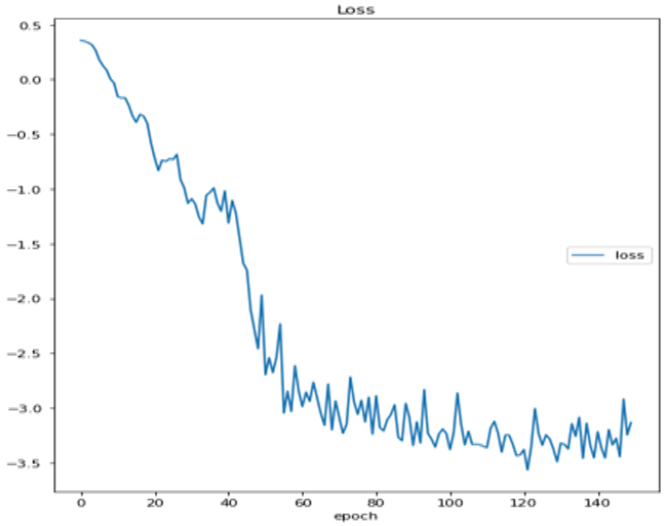

Fig.3 Loss plot during the training process: classifier 3, learning rate=0.001, dropout =0.1

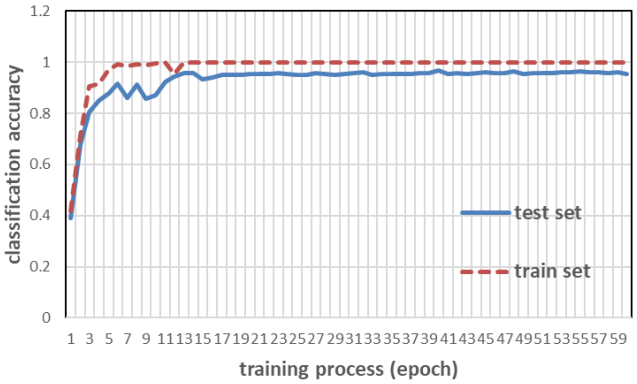

Fig.4 The measurement of classification accuracies on training set and test set as the training process is going on: classifier 3

Fig.4 shows the training progress in terms of classification accuracy. During the early training epochs, the accuracies are improved quickly. After epoch 20, the accuracies maintain almost the same. This implies that the training process can stop at epoch 20. For this particular case shown in Fig.4, the performance of the classifier on test set (about 96%) is constantly and slightly lower than that on the training set (100%). This implies that a minor overfitting problem occurs with the trained model. Various regularization techniques [2] are available to alleviate the overfitting problem if overfitting is severe.

After the training process has been completed (e.g. after 20 or 30 epochs), we test it using the test set, and obtain the confusion matrix, as show in Fig.5. From the confusion matrix,

Authorized licensed use limited to: The University of Texas Rio Grande Valley. Downloaded on March 08,2023 at 15:38:23 UTC from IEEE Xplore. Restrictions apply.

we can obtain the information of the classification accuracy on each class. is calculated as 96%.

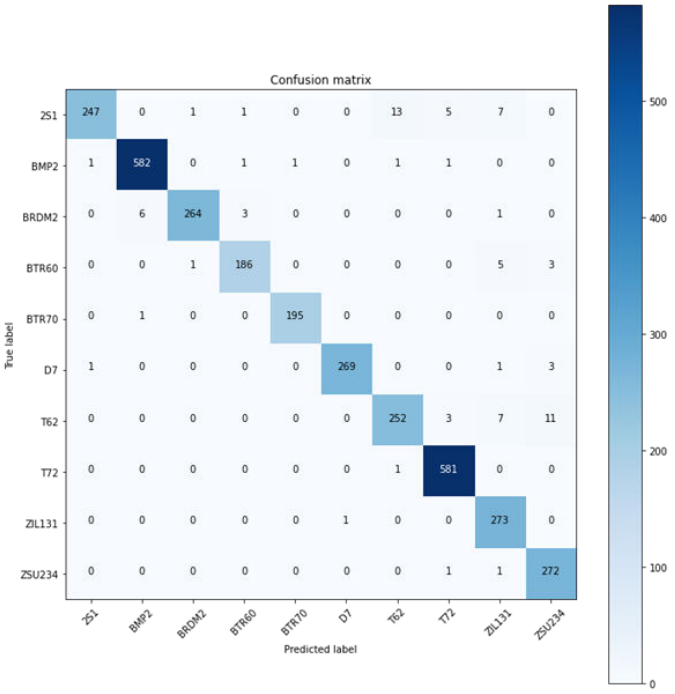

Fig.5 Confusion matrix of the trained classifier 3 testing on the test set

To compare the three classifiers and find the effect of training set size on classification accuracy, we trained each classifier using different training set sizes specified in Table V. The results are summarized in Table VI., and visualized in Fig. 6.

TABLE VI. SUMMARY OF CLASSIFIER ACCURACY AT DIFFERENT TRAINING SET SIZE

Training set	Classifier 1	Classifier 2	Classifier 3
1.2(3671)	91%	96%	97%
1.0(2880)	88%	92%	95%
0.75(75% of 2880)	80%	89%	92%
0.5(50% of 2880)	78%	88%	91%
0.25(25% of 2880)	73%	77%	79%
0.1(10% of 2880)	56%	55%	59%

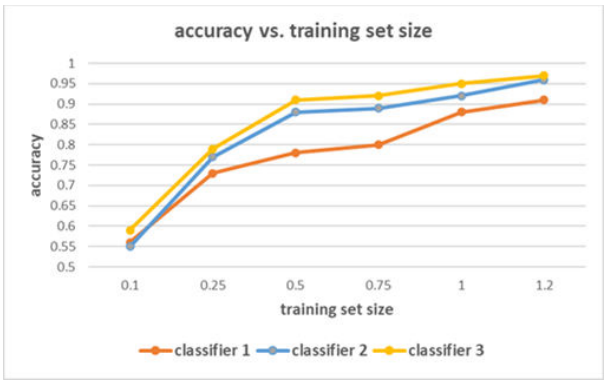

Fig.6 Classifiers' performance versus training dataset size

From Fig.6 we can make the following observations in the context of MSTAR dataset: 1) when the training set is reduced to below 50% of 2880 examples (144 examples per class), the accuracy of classifiers is getting significantly worse; 2) surprisingly, even when the training set has only about 30 examples per class (i.e. 0.1 size), the accuracy is still above 55%; and 3) classifier 2 and classifier 3 have similar performance, and they outperform classifier 1.

V. CONCLUSIONS

In general, a large dataset is required to train a deep neural network for a classification task. This requirement may set an obstacle for its applications when the availability of measured data is very limited. In this paper, we investigated the quantitative relationship between the performance (i.e. classification accuracy) and the training dataset size in the context of MSTAR dataset. The training set with thousands of SAR images for 10 classes can achieve an accuracy above 95% if a good neural network architecture is chosen. However, the training set with hundreds of images is not sufficient to achieve an acceptable performance.

To deal with the lack of large datasets in SAR ATR applications, there are two research directions: 1) data augmentation [8][9] by generating simulated SAR image data; and 2) meta-learning [10] by learning from different data domains.

REFERENCES

[1] John C. Curlander, Robert N. McDonough, Synthetic Aperture Radar: Systems and Signal Processing, Wiley publisher, 1992.

[2] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT press, 2017.

[3] Xiao Xiang Zhu, et al, Deep Learning Meets SAR: concepts, models, pitfalls, and perspectives, accepted by IEEE Geoscience and Remote Sensing Magazine, vol.9, no.4, 2021, pp. 143-172.

[4] https://www.sdms.afrl.af.mil/index.php?collection=mstar

[5] M. Wilmanski, C. Kreucher, and J. Lauer, "Modern approaches in deep learning for SAR ATR," in Proc. SPIE Algorithms for Synthetic Aperture Radar XXIII, (2016).

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

[7] Lewis, B., Scarnati, T., Sudkamp, E., Nehrbass, J., Rosencrantz, S., and Zelnio, E., "A SAR database for ATR development: Synthetic and Measured Paired Labeled Experiment (SAMPLE)," in Proc. SPIE Algorithms for Synthetic Aperture Radar XXVI, (2019).

[8] Ding J, Chen B, Liu H W, et al. Convolutional neural network with data augmentation for SAR target recognition. IEEE Geosci Remote Sens Lett, 2016, 13: 364–368.

[9] Cui Z Y, Zhang M R, Cao Z J, et al. Image data augmentation for SAR sensor via generative adversarial nets. IEEE Access, 2019, 7: 42255–42268.

[10] Timothy M.H., Antreas A. Paul M. and Amos J. S., "Meta-learing in neural networks: a survey," IEEE Transctions on Pattern Analysis and Machine Inteligence, accepted, 2021.

Authorized licensed use limited to: The University of Texas Rio Grande Valley. Downloaded on March 08,2023 at 15:38:23 UTC from IEEE Xplore. Restrictions apply.