Joint Embedding and Classification for SAR Target Recognition

Jiayun Wang, Patrick Virtue, and Stella X. Yu

Abstract—Deep learning can be an effective and efficient means to automatically detect and classify targets in synthetic aperture radar (SAR) images, but it is critical for trained neural networks to be robust to variations that exist between training and test environments. The layers in a neural network can be understood as successive transformations of an input image into embedded feature representations and ultimately into a semantic class label. To address the overfitting problem in SAR target classification, we train neural networks to optimize the spatial clustering of points in the embedded space in addition to optimizing the final classification score. We demonstrate that networks trained with this dual embedding and classification loss outperform networks with classification loss only. We study placing the embedding loss after different network layers and find that applying the embedding loss on the classification space results in the best SAR classification performance. Finally, our visualization of the network’s ten-dimensional classification space supports our claim that the embedding loss encourages greater separation between target class clusters for both training and testing partitions of the MSTAR dataset.

Index Terms—synthetic aperture radar (SAR), automatic target recognition (ATR), embedding, deep learning

I. INTRODUCTION

Synthetic aperture radar (SAR) captures images of objects that are suitable for target classification, reconnaissance and surveillance, regardless of the day-and-night conditions. Target recognition in SAR images has many potential applications in military and homeland security, such as friend and foe identification, battlefield surveillance, and disaster relief.

Automatic target recognition (ATR) for SAR images often proceeds in two stages: detection and classification. At the detection stage, candidate target patches are extracted from SAR images; they may include not only targets of interest, but also false alarm clutter such as trees, cars, buildings, etc. These detected image regions, or chips, are then sent to the classification stage, where a classification algorithm predicts the target class of each chip. In this work, we focus on improving the chip classification accuracy.

Traditional SAR target classification methods rely on extracting informative features and then training a classifier to map those features to the semantic target class labels. Various features have been explored, including raw pixel intensity [1], magnitudes of the two dimensional (2-D) DFT coefficients [2], and a transformation of the SAR image into polar coordinates followed by PCA compression [3]. Associated feature classifiers include support vector machines [1], radial basis function networks [2] and nearest neighbor approaches [3].
Deep learning methods have recently been used for SAR target recognition, integrating feature extraction and classification into a single neural network [5], [6], [4], [7]. Despite the appreciable achievements that these deep models have achieved in SAR target recognition, the complexity of the real-world SAR images still presents challenging differences between the training scenario and different test scenarios, including variations in the depression angle, SAR sensors, landscape, target articulation, and target configuration.

The typical overfitting problem in SAR target recognition is zero training error (Fig. 1), i.e., there is no room left for improving the learning process on the training set, yet there remains a large classification error (e.g. 9%) on the test set.

We argue that the deep network should be trained not only to improve the classification accuracy on each class, but also to increase the separation of features between classes. A larger separation allows a larger margin of errors when the test set deviates from the training set, thereby reducing the gap between training and testing accuracies.

Classification by deep learning can be understood as successive transformation of feature representations, where raw intensities in the pixel space get transformed into the labels in the semantic space. All the representations, including intermediate ones, can be considered an embedding of the input images. Through classifier learning, the distance in the representation becomes more indicative of individual semantic classes.

We propose to augment the network training objective by replacing the standard single classification loss with a dual...
classification and embedding loss, the latter aimed at increasing the distance between semantic classes in the embedded feature space. With this additional embedding loss, training performance will still be high, but now when the test data vary away from the training data, the larger separation in the embedded space reduces the chance for mis-classification.

The idea of combining a feature embedding loss with a target classification loss has been explored in the person re-identification (Person-ReID) task. This task, however, is not performing target identity classification, but determining whether two photos are of the same person or a different unknown person [8], [9], [10], [11], [12]. The feature embedding loss is a form of metric learning extending properties from samples on known identities to unknown identities, it has thus developed naturally in a Person-ReID classification formulation.

We are the first to explore adding a feature embedding loss to the usual classification loss in the recognition of known identities. We desire the learned features to be maximally separated from different classes, while supporting perfect classification of individual classes on the training set. Consequently, the deep neural net becomes more robust to appearance deviation of unseen new instances at the test set.

In addition to the novel application of the dual classification and embedding loss to target recognition tasks, we also further improve the algorithm by imposing the embedding loss at a later representation stage, in the classifier space, instead of the feature space that is commonly done in the Person-ReID literature (Fig. 2).

Our paper makes the following contributions:

- We demonstrate that the usual single classification loss alone suffers overfitting in SAR target recognition. We add the embedding loss to train the deep convolutional neural network (CNN), in order to learn more discriminative feature representations and improve SAR target recognition accuracies.
- We adopt a Siamese network architecture (Fig. 2) for our dual loss criterion. Compared with the existing single classifier architecture, our network can enhance the performance of the state-of-the-art.
- We investigate which CNN layer would be most effective to apply the embedding loss. Instead of applying the embedding loss to the feature space, we find out that it is best to apply to the classifier space where the features are linearly mapped to the same dimension as the number of output classes and yet before the nonlinear softmax normalization takes place.
- We conduct experiments to evaluate various aspects of the proposed method on the benchmark dataset MSTAR. The results outperform the previous SAR classification methods. We provide further insights through visualization of the learned discriminative feature space.

II. PROPOSED METHOD

Fig. 2 provides an overview of our approach. We first choose a published CNN network architecture on the MSTAR dataset as a baseline architecture, so that we can establish our results in comparison with the current state-of-the-art. We will then describe our new dual loss functions and various feature options to apply the new embedding loss. Our ideas can be applied to other CNN classifier architectures as well.

A. Network architecture

Our CNN architecture is based upon Chen’s A-ConvNet [5] with the following modifications.

1) We insert an additional convolution block to handle input chips of size $128 \times 128$, rather than cropping the input $88 \times 88$.
2) We add an additional fully connected layer to create an explicit feature vector of length 256.
3) We add batch normalization before the ReLU layers.

The resulting network consists of five convolutional blocks and two fully connected blocks. Each convolutional block, or “CBRP” block, is a series of layers consisting of convolution, batch normalization, ReLU activation, and then pooling. The fourth block is an exception, as it does not have a pooling layer. The five convolutional layers have square kernels of sizes 5, 5, 6, 5 and 3. The pooling layers all use kernel size $2 \times 2$. The number of output channels for the five convolutions are 16, 32, 64, 128 and 128. The output size of each block are $62 \times 62 \times 16, 29 \times 29 \times 32, 12 \times 12 \times 64, 8 \times 8 \times 128, 3 \times 3 \times 128$.

After the convolutional blocks, we use a fully connected layer followed by a ReLU layer to produce a 256 dimensional feature vector. While each CNN layer produces an intermediate feature representation of the input image, we refer to this last 256-dimensional space in which this vector lives in as the feature space. We use a fully connected layer to learn a linear transform from the feature space to a $k$-dimensional classifier space, where $k$ is the total number of classes. For the MSTAR dataset, $k = 10$.

B. Dual loss

Sun et al. [8] have shown that person re-identification is better achieved by simultaneously learning two tasks, one for recognizing the identity of person (which is available for the training images), and the other for learning a feature representations that maps images of the same person closer than those of different persons. The former focuses on pushing images of different identities apart, whereas the latter focuses on pulling images of the same identities together. Both tasks are best achieved when images are tightly clustered within the same identities and well separated between identities. At the test time, one can infer whether two new face images belong to the same albeit unknown identity.

We adopt the same dual loss idea to the classification task, i.e. to classify whether a new image belongs to one of the classes seen during training. Specifically, we combine classification and embedding losses to classify vehicle targets in SAR images. The dual loss leads to better generalization than the single classification loss, because the embedding loss enforces larger margins between classes.

Our dual loss is formulated as follows:

$$L = L_{\text{classification}} + \alpha L_{\text{embedding}} + R(W) \quad (1)$$
where $L_{\text{contrastive}}$ is the standard softmax cross-entropy loss, $L_{\text{embedding}}$ is an embedding loss, $\alpha$ is a hyper parameter that controls the relative importance of the embedding loss, and $R(W)$ is the standard weight decay regularization term.

Minimizing the classification loss forces the CNN to learn the correct predictions on individual images, whereas minimizing the embedding loss forces the CNN to develop an intermediate representation with better cohesion within each class and/or better separation between classes.

We consider two embedding loss functions, namely, contrastive loss and center loss. They both help learn more discriminative embedding representations: The contrastive loss aims to pull data in the same class closer and to push those from different classes apart, whereas the center loss aims to decrease the inter-class variation within individual classes.

**Contrastive Embedding Loss** Consider a pair of chips $(x_a, x_b)$ and their associated labels $(y_a, y_b)$. The goal is to pull the network activation vectors $f(x_a)$ and $f(x_b)$ closer if they belong to the same class and push them apart if they belong to different classes. The contrastive loss of for a pair of chips is:

$$L_{\text{contrastive}} = \begin{cases} L_{\text{sim}}, & \text{if } y_a = y_b \\ L_{\text{dis}}, & \text{if } y_a \neq y_b \end{cases}$$

(2)

$$L_{\text{sim}} = \max(0, \|f(x_a) - f(x_b)\|^2 - m_s)$$

(3)

$$L_{\text{dis}} = \max(0, m_d - \|f(x_a) - f(x_b)\|^2)$$

(4)

where $f(x)$ is the network output for the layer upon which the contrastive loss is applied; and $m_s$ and $m_d$ are hyperparameters for similar and dissimilar margins, respectively. For images of the same label, $L_{\text{sim}} > 0$ only if $\|f(x_a) - f(x_b)\|^2 > m_s$, whereas for images of different labels, $L_{\text{dis}} > 0$ only if $\|f(x_a) - f(x_b)\|^2 < m_d$. That is, the contrastive loss acts to pull the images of the same label closer than $m_s$, and to push the images of different labels farther apart than $m_d$.

In our experiments, we set $m_s = 0$. As shown in Fig. 4, the CNN for the contrastive loss is implemented as a Siamese network [13], where an image chip pair is fed in parallel to two identical networks with shared weights.

**Center Embedding Loss.** First proposed by Wen et al. to learn discriminative feature representations for face recognition [9], the center loss function reduces the variance of feature vectors with the same class label, promoting tighter class clusters in the embedding representation. The center loss is applied to a minibatch of training points as follows:

$$L_{\text{center}} = \frac{1}{2B} \sum_{i=1}^{B} \|f(x_i) - c_{y_i}\|^2$$

(5)

where $B$ is the batch size; $x_i$ is the $i$th chip in the batch; $f(x)$ is the network output for the layer upon which the embedding loss is applied; $y_i$ is the label of $x_i$; and $c_{y_i}$ is the mean of all $f(x)$ vectors where the class of $x$ is $y$.

**C. Feature Space vs. Classifier Space**

This section investigates which network layers we shall apply our embedding loss to. We first define several specific embedding spaces in our network: feature space, classifier space, and probability space, as illustrated in Fig. 2.

The feature space refers to the last network layer representation before the network maps it to the classifier space where the dimensionality is the same as the output – the total number of classes $k = 10$. The probability space is the
result of normalizing the 10-D classifier space via the softmax operation. Specifically:

\begin{align}
\text{feature space: } & \quad f(x) \\
\text{classifier space: } & \quad W f(x) + b \\
\text{probability space: } & \quad \hat{p}_i = \frac{e^{w_i f(x)}}{\sum_{j=1}^{k} e^{w_j f(x)}}, \quad i = 1, \ldots, k
\end{align}

where $W$ is the $10 \times 256$ weight matrix, $b$ is the length 10 bias vector learned in the final fully connected layer, and $w_i$ is the $i$-th row of $W$. $\hat{P} = [\hat{p}_1, \ldots, \hat{p}_k]^T$ is a 10-D vector in the probability space, which is directly used to determine the most likely class label (Fig. 2 right).

For the Re-ID task that employs both classification and embedding losses, we experiment with the network architecture described in Section II-A. For the sake of comparison, we also implement and test the following three possible dual loss modifications to our baseline approach with a single classification loss.

1) Add the embedding loss to the feature space, following the previous face Re-ID work.
2) Add the embedding loss to classifier space. Unlike the previous face ReID work with a dual loss, we apply the embedding loss (center or contrastive loss) to the classifier space.
3) Add the embedding loss to probability space. In attempt to move the embedding loss even further down the network, we apply the contrastive loss in the probability space, which is exactly where the we apply the classification loss.

C. Results

Table II lists the classification accuracy of all the methods. Fig. 3 shows the confusion matrices of contrastive loss. The following is a summary of our results:

1) **Our baseline is better than the state of the art.** Compared to Ding *et al.* [4], the overall accuracy improves 9.0%, and the unseen set result improves 10.6%.

2) **Adding an embedding loss in the feature space improves test classification accuracy.** Compared with our baseline, where only the classification loss is used, contrastive loss and center loss both improve the classification accuracy on the test set. For the center loss, compared with baseline, applying the embedding loss on feature space improves the overall classification accuracy by 4.9% and the unseen classification accuracy by 1.7%. For the contrastive loss, compared with the baseline, applying the embedding loss on feature space improves the overall classification accuracy by 0.9% and the unseen classification accuracy by 1.6%.

3) **Moving the embedding loss from the feature space to the classifier space further improves the test accuracy.** For the center loss, moving from feature space to classifier space further improves overall classification accuracy by 1.9% and the unseen classification accuracy by 4.0%. When moving contrastive loss from feature space to classifier space, we further improve the overall
We focus on the distribution of the test points that are never shown to the network (dots in Fig. 3). With the classification loss only, the intra-class variance is visibly large (Fig. 3 left). Although the inter-class overlapping of training data is rare, due to high intra-class variance, the classification loss still confuses the two classes 8.6% of the time. When the embedding loss is applied to the feature space, the intra-class variance is visibly decreased (Fig. 3 center). However, the difference between the two classes is still small, leading to 8.0% inter-class confusion. When we move the embedding loss to the classifier space, the intra-class variance is further decreased, and the inter-class separation is more significant (Fig. 3 right). Interestingly, the distribution of the two classes is more aligned with their respective axes, allowing the network to be more confident of its classification and reducing the confusion between 2S1 and BRDM2 to only 1.2%.

**t-SNE visualization.** t-SNE technique can visualize high-dimensional data by giving each data point a location in a two or three-dimensional map.

In Fig. 6, we visualize the classifier space learned by applying the embedding loss in different spaces. The quality of the class clusters in this t-SNE visualization mirrors the quantitative accuracy results in Table II. Moving the contrastive embedding loss from the feature space to the classifier space, the cluster of each class has smaller intra-class spread and bigger inter-class distance. In addition, the number of misclassified points in the test set are reduced when we apply the embedding loss to the classifier layer. t-SNE visualization shows that the center loss applied to the feature space induces good clustering in the classifier space.

Fig. 7 shows the t-SNE embedding of the contrastive loss applied to the classifier space, with each test point represented by its associated chip. This visualization shows a uniform distribution of both vehicle rotation and position within the chip, indicating that the trained network is invariant to both translation and target orientation.

### TABLE I

**Setup of the MSTAR dataset (before augmentation) [4]**

| Dataset | BMP2 | BTR70 | T72  | BTR60  | 2S1  | BRDM2 | D7   | T62   | ZIL131 | ZSU23/4 |
|---------|------|-------|------|--------|------|-------|------|-------|--------|---------|
| Training | 233  | 233   | 196  | 256    | 999  | 299   | 299  | 299   | 299    | 299     |
| Testing  | 196  | 195   | 196  | 195    | 274  | 274   | 274  | 274   | 274    | 274     |

### TABLE II

**Classification accuracy on translation augmented test set**

| Network                                      | Seen  | Unseen | Overall |
|----------------------------------------------|-------|--------|---------|
| Ding [4]                                     | 83.0  | 80.4   | 82.4    |
| Baseline: classification loss only           | 91.6  | 91.0   | 91.4    |
| Baseline + center loss on feature space      | 97.4  | 92.7   | 96.3    |
| Baseline + contrastive loss on feature space | 97.2  | 92.6   | 92.3    |
| Baseline + center loss on classifier space   | 98.7  | 96.7   | 98.2    |
| Baseline + contrastive loss on classifier space | 98.5  | 97.2   | 98.2    |
| Baseline + contrastive loss on probability space | 90.1  | 92.4   | 90.6    |

classification accuracy by 5.9% and unseen classification accuracy by 4.6%. On unseen test set, the influence of applying embedding loss on classifier space is more significant.

4) **Moving the embedding loss from the classifier space to the probability space hurts the classification accuracy.** When applying the contrastive loss after the softmax non-linear layer, the overall classification accuracy decreases 7.6% and the unseen classification accuracy decreases 4.8%. It is not surprising, as it distracts the network from minimizing the target classification loss.

Overall, adding a feature embedding loss improves the classification accuracy at the test time. Applying the embedding loss in the classifier space leads to higher overall classification accuracy than that applying it in the feature space. Additionally, although both embedding loss functions perform equivalently well, the contrastive loss performs slightly better than the center loss on unseen test images.

D. Analysis

In order to better understand the learned space for target classification, we use both orthogonal projection and t-Distributed Stochastic Neighbor Embedding (t-SNE) [14] to visualize the learned 10-D classifier space.

**Direct visualization of Two Confusing classes.** Fig. 3 plots the representation in the classifier space for two specific target classes, 2S1 and BRDM2.
IV. MSTAR OUTLIER REJECTION

In SAR-ATR systems, the detector may return some unseen targets of unknown classes which could confuse the system. We conduct outlier rejection experiments to evaluate the confuser rejection performance of the proposed methods.

A. Data

We follow the experimental setup on outlier rejection from Chen [5], training the neural network on three target classes, BMP-2, BTR-70, and T-72, and testing on the original targets and additional confuser targets, 2S1 and ZIL-131. In this experimental scenario, the test set is composed of 588 images of three known targets (196 images per class) and 548 images of two confuser targets (274 images per class).

B. Methods

The output of the CNN gives the posterior probability of classification for each class. Given a threshold $\tau$, the target image will be declared as a confuser if all the posterior probabilities are lower than $\tau$. Denote $P_d$ as the probability of detection, and $P_{fa}$ as the probability of false alarm. These values are formulated as follows:

\[ P_d = \frac{N(\text{prediction} = \text{target} | \text{real label} = \text{target})}{N(\text{real label} = \text{target})} \quad (9) \]

\[ P_{fa} = \frac{N(\text{prediction} = \text{target} | \text{real label} = \text{confuser})}{N(\text{real label} = \text{confuser})} \quad (10) \]

Varying $\tau$, we collect many pairs of detection probability and false alarm probability. The receiver operating character-
The ROC curves of different methods are shown in Fig. 8. At $P_d = 90\%$, the false alarm probabilities are 56.1%, 39.9%, 20.5% for our baseline, our center loss applied in the classifier space, our contrastive loss applied in the classifier space respectively. For each embedding loss, given a fixed detection probability, applying the loss in the classifier space gives a lower false alarm probability than applying that loss in the feature space.

### D. Analysis

From these experimental results, we draw the following conclusions:

1) In terms of rejection performance, dual loss performs better than single classification loss only. Since the learned classifier space is more discriminative, the ATR system becomes more confident in rejecting outliers.

2) Applying the embedding loss in the classifier space performs better on outlier rejection than that in the feature space. Moving the embedding loss from the feature space to the classifier space can improve outlier rejection per-
performance in addition to improving classification accuracy.

3) Contrastive loss performs better than center loss. Center loss makes the classifier more discriminative by decreasing the intra-class variance. Contrastive loss, on the other hand, also increases the inter-class difference, which leads to improved outlier rejection.

V. CONCLUSION

We present a joint embedding and classification framework for SAR target recognition. We demonstrate that applying the embedding loss in the classifier space leads to better performance than applying it in the feature space. Two types of embedding loss functions, contrastive loss and center loss, are compared. Both can achieve higher target classification accuracy for SAR target recognition task. In addition, we find that contrastive loss performs better than center loss on outlier rejection tasks.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Zhirong Wu for valuable discussions and feedback.

REFERENCES

[1] Q. Zhao and J. C. Principe, “Support vector machines for SAR automatic target recognition,” IEEE Transactions on Aerospace and Electronic Systems, vol. 37, no. 2, pp. 643–654, 2001.
[2] Y. Sun, Z. Liu, S. Todorovic, and J. Li, “Adaptive boosting for SAR automatic target recognition,” IEEE Transactions on Aerospace and Electronic Systems, vol. 43, no. 1, 2007.
[3] J.-I. Park and K.-T. Kim, “Modified polar mapping classifier for SAR automatic target recognition,” IEEE Transactions on Aerospace and Electronic Systems, vol. 50, no. 2, pp. 1092–1107, 2014.
[4] J. Ding, B. Chen, H. Liu, and M. Huang, “Convolutional neural network with data augmentation for SAR target recognition,” IEEE Geoscience and Remote Sensing Letters, vol. 13, no. 3, pp. 364–368, 2016.
[5] 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.
[6] E. R. Keydel, S. W. Lee, and J. T. Moore, “MSTAR extended operating conditions: A tutorial,” in Aerospace/Defense Sensing and Controls. International Society for Optics and Photonics, 1996, pp. 228–242.
[7] Z. Lin, K. Ji, M. Kang, X. Leng, and H. Zou, “Deep convolutional highway unit network for sar target classification with limited labeled training data,” IEEE Geoscience and Remote Sensing Letters, 2017.
[8] Y. Sun, Y. Chen, X. Wang, and X. Tang, “Deep learning face representation by joint identification-verification,” in Advances in neural information processing systems, 2014, pp. 1988–1996.
[9] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in European Conference on Computer Vision. Springer, 2016, pp. 499–515.
[10] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
[11] S. Zhou, J. Wang, J. Wang, and Y. Gong, “Point to set similarity based deep feature learning for person re-identification,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[12] J. Wang, S. Zhou, J. Wang, and Q. Hou, “Deep ranking model by large adaptive margin learning for person re-identification,” Pattern Recognition, 2017.
[13] J. Bromley, I. Guyon, Y. LeCun, E. Sackinger, and R. Shah, “Signature verification using a “Siamese” time delay neural network,” in Advances in Neural Information Processing Systems, 1994, pp. 737–744.
[14] L. v. d. Maaten and G. Hinton, “Visualizing data using t-SNE,” Journal of Machine Learning Research, vol. 9, no. Nov, pp. 2579–2605, 2008.