

Targeted adversarial discriminative domain adaptation

Hua-Mei Chen,^a Andreas Savakis,^b Ashley Diehl,^c Erik Blasch^{©,d},
Sixiao Wei,^a and Genshe Chen^{a,*}

^aIntelligent Fusion Technology, Inc., Germantown, Maryland, United States

^bRochester Institute of Technology, Rochester, New York, United States

^cAir Force Research Laboratory, Wright-Patterson Air Force Base, Ohio, United States

^dAir Force Office of Scientific Research, Arlington, Virginia, United States

Abstract. Domain adaptation is a technology enabling aided target recognition and other algorithms for environments and targets with data or labeled data that is scarce. Recent advances in unsupervised domain adaptation have demonstrated excellent performance but only when the domain shift is relatively small. We proposed targeted adversarial discriminative domain adaptation (T-ADDA), a semi-supervised domain adaptation method that extends the ADDA framework. By providing at least one labeled target image per class, used as a cue to guide the adaption, T-ADDA significantly boosts the performance of ADDA and is applicable to the challenging scenario in which the sets of targets in the source and target domains are not the same. The efficacy of T-ADDA is demonstrated by cross-domain, cross-sensor, and cross-target experiments using the common digits datasets and several aerial image datasets. Results demonstrate an average increase of 15% improvement with T-ADDA over ADDA using just a few labeled images when adapting to a small domain shift and afforded a 60% improvement when adapting to large domain shifts. © The Authors. Published by SPIE under a Creative Commons Attribution 4.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.JRS.15.038504](https://doi.org/10.1117/1.JRS.15.038504)]

Keywords: semi-supervised domain adaptation; transfer learning; domain shift; adversarial learning; generative adversarial network.

Paper 210201SS received Mar. 31, 2021; accepted for publication Jul. 6, 2021; published online Jul. 19, 2021.

1 Introduction

Aided target recognition (AiTR) focuses on developing automatic target recognition (ATR) to aid a human user.¹ Three examples of AiTR include discriminability with data fusion,² extendibility over data sparsity,³ and interpretability from data compression.⁴ Data fusion supports AiTR through enhancing target recognition by combining data from two or more sensors. Data fusion has demonstrated numerous capabilities for applications such as infrared (IR) and millimeter-wave IR for object detection,⁵ electro-optical (EO) and IR for object tracking,⁶ and EO and radar for enhanced situation awareness.⁷ Combining EO with radar signatures allows for machine processing of multiresolution data with sparsity and complexity.⁸ These data fusion methods afford interpretability of data for task success. Examples include interpretability over compressed imagery data,⁹ 3D volumetric lidar data,¹⁰ and classifier assessment.¹¹ One of the challenges is to develop efficient methods for large volumes of data from which developments in deep learning (DL) have become popular.

Deep convolutional neural networks (CNNs) trained on large datasets have demonstrated excellent performance on computer vision tasks such as object classification,¹² change detection,¹³ and ATR¹⁴ from EO and radar data. However, the data distribution in the target domain, where testing takes place, may be different from the data distribution in the source domain, where training occurs. Domain adaptation (DA) aims to overcome the domain shift, or dataset bias,¹⁵ that reduces classifier performance when classification takes place in a target domain. The shift in the data distribution may be due to differences in illumination, sensor type, perspective, background, and target classes. Conventional deep transfer learning utilizes pretrained CNN

*Address all correspondence to Genshe Chen, gchen@infucitontech.com

models for feature extraction and performs fine-tuning for training on a labeled dataset of interest. Unsupervised DA deals with unlabeled data in the target domain after training with labeled data in the source domain. Many unsupervised DA approaches have demonstrated excellent performance, but only when the domain shift is small.

For applications such as transferring knowledge from one set of targets to another set of targets, any unsupervised DA approach is doomed to fail as the class correspondence is typically ambiguous and without further information, and an unsupervised DA method will have limited knowledge on how the adaptation should proceed. Figure 1 shows an example of DA ambiguity. In Fig. 1(a), the red uppercase letters represent the source domain, and the blue lowercase letters represent the target domain. For an unsupervised DA, the target feature vectors in classes a, b, and c will be adapted to nearby source classes A, B, and C, respectively. Figure 1(b) shows what an unsupervised DA approach can achieve. Without knowing the correspondence between the classes in the source and target domains, adjacent classes in the source and target domains will be merged, thus representing the best that an unsupervised DA method can achieve. Obviously, the adaptation results are not necessarily correct without a domain mapping.

The need for the algorithm to know where the target classes a, b, and c should be adapted to for correct adaptation¹⁶ motivates the targeted adversarial discriminative DA (T-ADDA) approach. T-ADDA assumes the availability of at least one labeled target image per target class (i.e., one labeled target feature vector per class). The labeled target feature vectors are indicated by the dark blue, underlined lowercase letters in Fig. 2(a). By enforcing all labeled target feature vectors to move toward their targeted source class centers as indicated by the dashed lines, T-ADDA adapts the target model, so the resulting target classes in the target domain correctly match the corresponding source classes as shown in Fig. 2(b).

This paper is organized as follows. Section 2 provides a brief review of unsupervised DA approaches and a deeper look into the unsupervised DA approach that T-ADDA was built upon, i.e., Adversarial Discriminative DA (ADDA). The proposed T-ADDA is detailed in Sec. 3 and followed by implementation methods in Sec. 4. In Sec. 5, four experimental results using digit

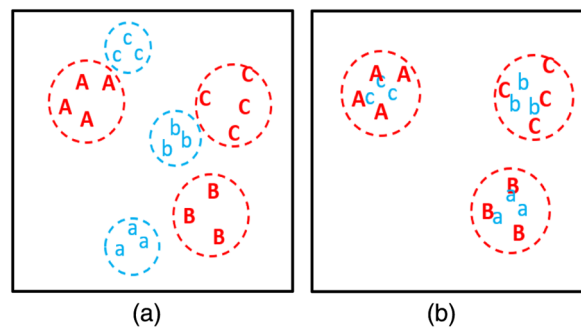


Fig. 1 Illustration of unsupervised DA.

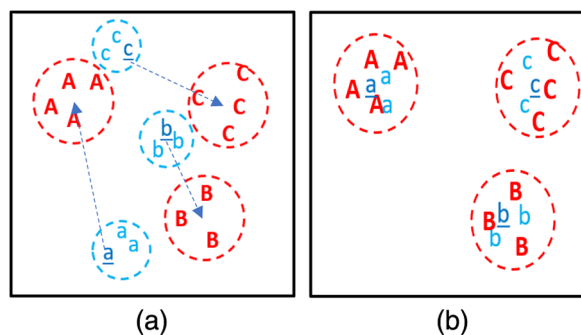


Fig. 2 Illustration of the idea that motivates T-ADDA.

datasets and real aerial image datasets (AID) are presented. Finally, concluding remarks are provided in Sec. 6.

2 Literature Review of Domain Adaptation

2.1 Unsupervised Domain Adaptation

Subspace alignment (SA)¹⁷ is one of the early unsupervised DA approaches that performs a transformation on the source and target domain representations to generate features that are domain invariant. Other methods that perform subspace alignment include CORAL¹⁸ and manifold aligned label transfer DA.¹⁹ Adversarial learning is often used by DA methods. The domain adversarial neural networks method²⁰ uses a gradient reversal layer to learn features that are class discriminative and domain invariant. Domain symmetric networks (SymNets)²¹ are based on a symmetric design of source and target task classifiers and adversarial training with a domain confusion scheme for learning domain invariant representations.

The work that is most relevant to the proposed T-ADDA semi-supervised domain adaptation method is the unsupervised DA framework by Tzeng.²² In fact, the T-ADDA approach can be considered an extension of ADDA from unsupervised learning to semi-supervised learning.

ADDA is a generalized framework for adversarial DA that combines discriminative modeling, untied weight sharing, and a generative adversarial network (GAN) loss. ADDA first learns a discriminative representation using the labels in the source domain and then a separate encoding that maps the target data to the same space using an asymmetric mapping learned through a domain-adversarial loss. It is a simple, flexible, yet surprisingly powerful approach that achieves state-of-the-art visual adaptation results on standard DA datasets.

All of the above unsupervised DA methods assume that the initial domain shift is relatively small and that adjacent classes in the source and target domains correspond to the same target class. However, a small domain shift assumption may not be true if the source and target domains are very different. When the domain shift is large, extra information in terms of a few labeled target images is needed, and it is known as semi-supervised DA (SSDA).

2.2 Semi-Supervised Domain Adaptation

SSDA is an important task; however, it has not been fully explored with regard to DL-based methods.²³ One notable SSDA work was the minimax entropy DA by Saito et al.²³ In minimax entropy DA, domain invariant class prototypes are defined as the weight vectors of the classifier C , which takes normalized feature vectors as its input and outputs the probability of classes with a softmax activation function. Then, the weight vectors are updated during training to maximize the entropy measured by the similarity between W , the weight vectors associated with the classifier C , and the unlabeled target features. Next, the feature extractor F is updated to minimize the entropy on unlabeled target examples to yield discriminative features extracted by F . At the same time, C and F are trained to classify both labeled source examples and a few labeled target examples correctly by minimizing the cross-entropy.

2.3 Adversarial Discriminative Domain Adaptation

ADDA²¹ uses a GAN framework along with an adversarial loss for DA. Details of ADDA are provided next to set the stage for describing the proposed T-ADDA. To begin, we need source images \mathbf{X}_s and labels Y_s drawn from a source domain distribution $p_s(x, y)$ and target images \mathbf{X}_t drawn from a target domain distribution $p_t(x, y)$, where there are no labels available. The goal is to learn a target feature encoder M_t and a target classifier C_t that can correctly classify target images into one of K categories at test time, despite the lack of target domain annotations. Since direct supervised learning on the target is not possible, DA instead learns a source feature encoder M_s along with a source classifier C_s and then adapts that model for use in the target domain. The adaptation is accomplished by minimizing the distance between the two empirical source and target distributions $M_s(\mathbf{X}_s)$ and $M_t(\mathbf{X}_t)$ and setting $C_t = C_s$.

The source classification model is trained using the standard supervised cross-entropy loss as

$$\min_{M_s, C_s} \mathcal{L}_{\text{cross-entropy}}(\mathbf{X}_s, Y_s) = \mathbb{E}_{(\mathbf{x}_s, y_s) \sim (\mathbf{X}_s, Y_s)} \left[- \sum_{k=1}^K \mathbb{1}_{[k=y_s]} \log C_s(M_s(\mathbf{x}_s)) \right]. \quad (1)$$

To minimize the empirical source ($M_s(\mathbf{X}_s)$) and target ($M_t(\mathbf{X}_t)$) distributions, the adversarial learning of ADDA consists of alternating the following two optimizations:

$$\min_D \mathcal{L}_{\text{adv}_D}(\mathbf{X}_s, \mathbf{X}_t, M_s, M_t), \quad (2)$$

and

$$\min_{M_t} \mathcal{L}_{\text{adv}_M}(\mathbf{X}_t, D), \quad (3)$$

where D is a domain discriminator that classifies whether a data point is drawn from the source or target domain. Equation (2) states that the domain discriminator D is optimized according to an adversarial domain discrimination loss function $\mathcal{L}_{\text{adv}_D}$, which is defined as

$$\mathcal{L}_{\text{adv}_D} = -\mathbb{E}_{\mathbf{x}_s \sim \mathbf{X}_s} [\log D(M_s(\mathbf{x}_s))] - \mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t} [\log(1 - D(M_t(\mathbf{x}_t)))]], \quad (4)$$

and Eq. (3) states that the target encoder M_t is optimized according the GAN loss function \mathcal{L}_{GAN} , which is defined as

$$\mathcal{L}_{\text{adv}_M} = \mathcal{L}_{\text{GAN}} = -\mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t} [\log(D(M_t(\mathbf{x}_t)))]]. \quad (5)$$

It is worth noting that the source encoder M_s is optimized during pretraining and is fixed during the above adversarial learning process.

3 Proposed T-ADDA Approach

3.1 Assumption

T-ADDA, as illustrated in Fig. 2, makes two assumptions. The first assumption is that the source and target features of different target classes are well separated and clustered, and the second assumption is that all target feature points of the same classes will follow the movements of the few labeled target feature points to result in the desired adaptation result. The success of T-ADDA relies on the validity of the above two assumptions.

To enforce the validity of the first assumption, the combined cross-entropy and center loss function²⁴ is adopted to encourage separation and clustering of source feature vectors. However, it is not straightforward to enforce clustering of target feature vectors, which are encoded by the initial target feature extractor (target feature encoder). In T-ADDA, clustering of target feature vectors is supported experimentally by carefully choosing the initial target feature encoder.

The second assumption, that all target feature points of the same classes will follow the movements of the few labeled target feature points to result in the desired adaptation result, is enforced by adversarial learning as described in ADDA and is validated by extensive experiments shown in the results.

3.2 Targeted Adversarial Discriminative Domain Adaptation

When there are no labeled target images, the proposed T-ADDA is identical to ADDA reviewed in Sec. 2. When few labeled target images are available, three types of input data can be distinguished in T-ADDA: (1) the labeled source data \mathbf{X}_s , (2) the target data \mathbf{X}_t , and (3) the few labeled target data $\mathbf{X}'_t \subset \mathbf{X}_t$. The use of \mathbf{X}_s and \mathbf{X}_t in T-ADDA is identical to ADDA as described in Eqs. (4) and (5). When few labeled target images are available, i.e., \mathbf{X}'_t is not an empty set, the target encoder M_t defined in Sec. 2 is additionally optimized according to the following feature class matching (FCM) loss function using \mathbf{X}'_t .

$$\mathcal{L}_{FCM} = \sum_{i=1}^n \|S_{C_i} - \mathbf{x}'_i\|_2^2, \tag{6}$$

where \mathbf{x}'_i is the feature vector extracted from the i 'th labeled target image and S_{C_i} is the corresponding source feature class centers, which are extracted after the source model is trained. Figure 3 shows the overview of the proposed T-ADDA approach, which consists of three steps. In Step 1, a source model is pretrained using the source domain training dataset with either cross-entropy or combined cross-entropy and center loss functions, which is described in the next subsection. Once the source model is pretrained, T-ADDA computes and saves the center of features in each class. The key contribution of the proposed T-ADDA is performed in Step 2 and attempts to adapt a target encoder M_t so that the features extracted by it cannot be distinguished from the features extracted by the source encoder M_s . In this step, \mathcal{L}_{adv_D} , \mathcal{L}_{adv_M} , and \mathcal{L}_{FCM} , given in Eqs. (4)–(6), respectively, are optimized alternately. Finally, in Step 3, the target model is formed by the adapted target encoder concatenated with the classification layer(s) of the source model and is used to classify images in the target domain.

3.3 Center Loss

Through supervised training, via minimizing categorical cross entropy loss, discriminative features are guaranteed to be generated, but well-clustered features are not guaranteed. The idea of center loss was originally presented in Ref. 24 and was adopted in Ref. 19 for DA. It was shown that by combining cross-entropy loss and center-loss functions, well clustered features are generated, and the accuracy of classifiers can be improved,²⁴ which is confirmed by our experiments. Thus, in this section, the center loss is presented and then employed throughout our experiments for improved source model performance and, thus, improved T-ADDA performance.

Center loss function is formulated as

$$\mathcal{L}_C = \frac{1}{2} \sum_{i=1}^m \|x_i - C_{y_i}\|_2^2, \tag{7}$$

where x_i and y_i are the i 'th feature vector and its label and $C_{y_i} \in \mathbb{R}^d$ denotes the y_i 'th class center of deep features. The formulation with Eq. (7) encourages each encoded feature point to move toward the corresponding class center C_{y_i} . In Ref. 24, attention was paid to updating the dynamic class center during the training process. However, T-ADDA adopts a two-stage training process to simplify the implementation. In the first stage, the source model is trained using the cross-

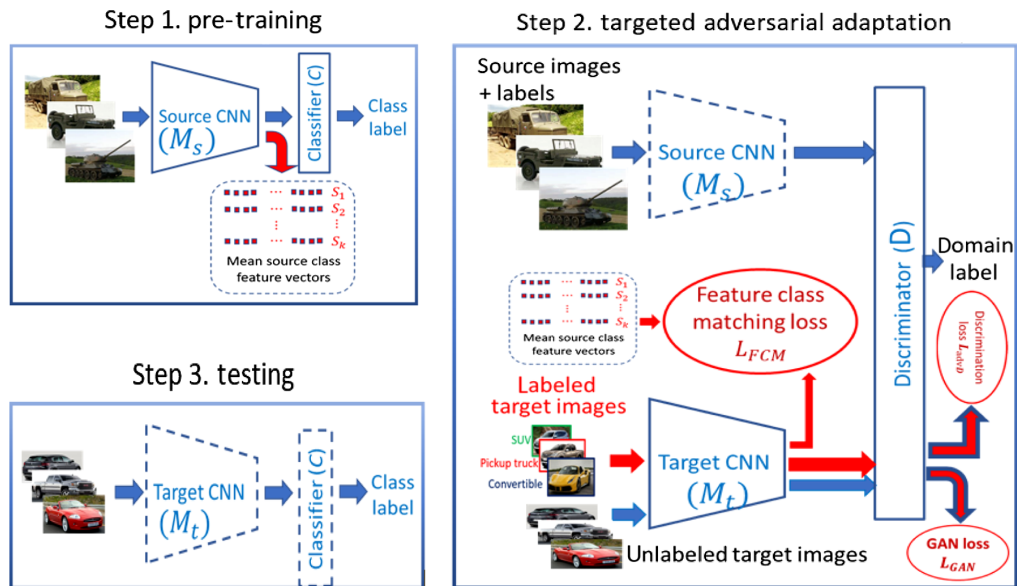


Fig. 3 Overview of the proposed T-ADDA approach.

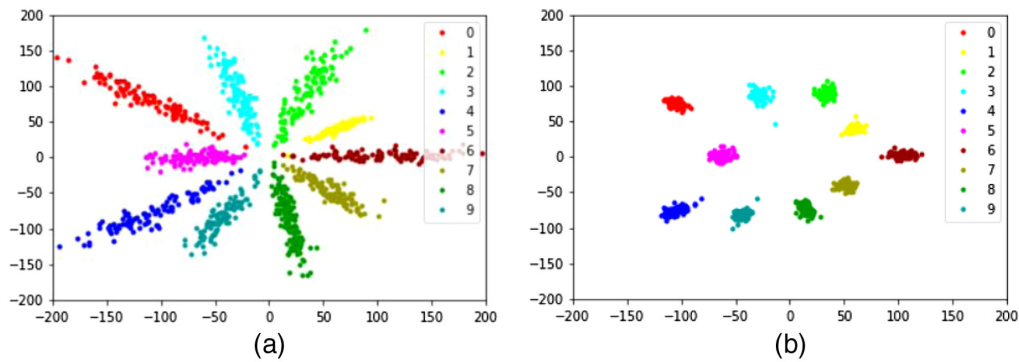


Fig. 4 MNIST features obtained using (a) cross-entropy and (b) combined center and cross-entropy loss as the function to be minimized.

entropy loss function only and computes the centers of all classes to be used in Eq. (6) to encourage feature clustering. In the second stage, the computed class centers were then used in Eq. (6) to compute center loss. The complete loss to be minimized is the combination of cross entropy and center loss given as

$$\mathcal{L} = \lambda \cdot \mathcal{L}_C + \mathcal{L}_S, \quad (8)$$

where \mathcal{L}_S denotes the standard cross entropy loss and \mathcal{L}_C is the center loss given in Eq. (7). A visual comparison of features resulting from cross entropy loss and combined cross entropy and center loss is given in Fig. 4. Figure 4 uses MNIST data and a LeNet++ source model²⁴ to generate the plots by setting the feature dimension to be two. We note that, in T-ADDA, the computed source class centers are used in the feature matching loss function given in Eq. (6), where the source class centers are denoted by S_{C_i} .

4 Implementation

The pseudo code of the proposed T-ADDA approach is provided in Fig. 5.

Given source domain data **X_source**, source class centers **sourceCenters**, unlabeled target domain data **Y_target_unlabeled**, labeled target domain data **Y_target_labeled**, as well as the following neural networks *sourceEncoder*, *targetEncoder*, *discriminator*, *gan*, and *fcm*. The loss functions employed by *discriminator*, *gan*, and *fcm* are eq.(4), eq.(5) and eq.(6) respectively. Denote **Y_target** the union of **Y_target_unlabeled** and **Y_target_labeled**.

```

=====
for epoch = 1 to epoch_max
  for batch = 1 to batch_max
    sourceFeatures = sourceEncoder(X_source)
    targetFeature = targetEncoder(Y_target)
    discriminator.train_on_batch([sourceFeatures, targetFeature],
                                disc_y=[1,...,1,0,...,0])
    gan.train_on_batch(Y_target, gen_y=[1,...,1])
    fcm.train_on_batch([Y_target_labeled, sourceCenters], [dummy_y,
                                                         dummy_y])
  end_batch
end_epoch

```

Fig. 5 Pseudo code of the proposed T-ADDA.

For the experimental results involving digits datasets (Secs. 5.2.1–5.2.2), we constructed the source model based on LeNet++.²⁴ Table 1 shows the summary of our LeNet++ based model, which is a variation of LeNet++ by incorporating batch normalization and dropout layers. The source encoder is formed from the InputLayer up to the layer ip1. The dimension of the feature space is fixed at 500. The dense layer ip2 serves as a linear 10 class classifier. The LeNet++ based source models, after being trained with source domain datasets, are used as the initial target models for adaptation in Experiments 1 and 2 involving digit datasets.

For the experimental results involving real AID (Secs. 5.2.3–5.2.4), we built our source model based on ImageNet²⁵ pretrained models, as we found that the LeNet++ based model was not able to extract well-separated and clustered features. Specifically, we employed the ImageNet pretrained DenseNet²⁶ model provided in Keras, and the layer GlobalMaxPooling2D was adopted to reduce the CNN features from dimension $7 \times 7 \times 1024$ to 1024 to form the base model. Then, a dense (fully connected) layer *fc1* was added to form the source encoder. To complete the source classification model, we added another fully connected layer *fc2* to the source encoder as the classifier. The purpose of the *fc1* layer is to further reduce the feature space dimension to a pre-determined size, in this case, 256. To construct the initial target encoder, two strategies were considered: (1) adopting the source encoder as the initial target encoder and (2) concatenating the base model

Table 1 Summary of the implemented LeNet++ based model.

Layer (type)	Output shape	#Parameters
Input (InputLayer)	(none,32,32,3)	0
Conv2D_01 (conv2D)	(none,32,32,32)	2432
BN_01 (BatchNorm)	(none,32,32,32)	128
Conv2D_02 (Conv2D)	(none,32,32,32)	25632
BN_02 (BatchNorm)	(none,32,32,32)	128
Max_pooling2d_02 (MaxPooling2D)	(none,16,16,32)	0
Conv2D_03_1 (Conv2D)	(none,16,16,64)	51264
BN_03_1 (BatchNorm)	(none,16,16,64)	128
Max_pooling2d_02 (MaxPooling2D)	(none,16,16,32)	0
Conv2D_03_2 (Conv2D)	(none,16,16,64)	102464
BN_03_2 (BatchNorm)	(none,16,16,64)	256
Max_pooling2d_03 (MaxPooling2D)	(none,8,8,64)	0
Conv2D_04_1 (Conv2D)	(none,8,8,128)	204928
BN_04_1 (BatchNorm)	(none,8,8,128)	512
Conv2d_04_2 (Conv2D)	(none,8,8,128)	409728
Max_pooling2d_04 (MaxPooling2D)	(none,4,4,128)	0
Activation_04 (Activation)	(none,4,4,128)	0
BN_04_2 (BatchNorm)	(none,4,4,128)	512
Flatten (Flatten)	(none,2048)	0
Dropout (Dropout)	(none,2048)	0
IP1 (Dense)	(none,500)	1024500
IP 2 (Dense)	(none,10)	5010
Total Parameters: 1,827,750		

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 1024)	263168
leaky_re_lu_1 (LeakyReLU)	(None, 1024)	0
dense_2 (Dense)	(None, 512)	524800
leaky_re_lu_2 (LeakyReLU)	(None, 512)	0
dense_3 (Dense)	(None, 1)	513

Fig. 6 Structure of the implemented discriminator in T-ADDA.

with the $fc1$ layer from the source encoder. The first strategy tunes the feature extraction CNN toward classifying the specific source classes, and the second strategy keeps the feature extractor intact. In our experiments, we found that the first approach worked better when the source and target domains shared the same object classes, and the second approach was preferred when new classes were introduced in the target domain. We believe that the reason is that when source and target domains share the common target classes, the fine-tuned feature extraction layers are able to extract well separated and clustered features in the source domain as well as in the target domain. However, if new classes appear in the target domain, since the feature extraction layers are fine-tuned toward extracting features that well separate the source classes, it may not have the power to extract the features that distinguish different target classes in the target domain. In this case, it is preferred to retain all feature extraction layers that are pretrained using a large amount of the ImageNet dataset.

Next, the network discriminator in both cases consists of three dense layers as shown in Fig. 6.

The network GAN is formed by concatenating *targetEncoder* and discriminator, and only the *targetEncoder* is trainable. Finally, the network FCM is implemented similarly to how the source model is trained by combined center loss and cross-entropy loss functions. However, in *FCM*, only the center loss is employed. The label *dummy_y* is randomly generated as no label is required for employing the center loss function.

5 Experimental Procedures and Results

5.1 Datasets

The proposed T-ADDA is first evaluated in two experiments involving three datasets with 10 digit classes and then tested in two experiments involving six aerial datasets. Brief descriptions about each dataset are provided below.

5.1.1 Modified National Institute of Standards and Technology

The Modified National Institute of Standards and Technology (MNIST)²⁷ database consists of 70,000 grayscale handwritten digit images. Among them, 60,000 images are sequestered for the training set, and the remaining 10,000 are saved for the test set. The MNIST database is commonly used for developing and testing various image processing systems.

5.1.2 Street View House Numbers

The Street View House Numbers (SVHN)²⁸ dataset, obtained from house numbers in Google Street View images, is a real-world image dataset for developing machine learning and object recognition algorithms with minimal requirements on data preprocessing and formatting. The image size of SVHN is 32×32 . It can be seen as similar in style to MNIST (e.g., the images are of small cropped digits), but it incorporates an order of magnitude more labeled data (over 600,000 digit images) and comes from a significantly harder, unsolved, real world problem

(recognizing digits and numbers in natural scene images). Among them, 73,257 images are for training, and 26,032 are for testing.

5.1.3 Devanagari Handwritten Character

The Devanagari handwritten character (DHC) dataset²⁹ is a database of handwritten Devanagari characters consisting of 46 classes of characters, including ten Devanagari digits, with 2000 examples each. The image size of DHC dataset is 32×32 .

In Fig. 7, example digits images from MNIST, SVHN, and DHC databases are provided for comparison. We note that the same digits in Arabic and Devanagari numerals do not necessarily have the same meaning. For example, the Arabic digit 9 resembles Devanagari digit 1

5.1.4 Aerial Image Datasets

AID³⁰ contains over 10,000 aerial images from 30 classes. The image size is 600×600 pixels, obtained at multiple ground sampling distances (GSDs) (8 to 0.5 m). The source is Google Earth images from various countries.

5.1.5 The University of California Merced

The University of California Merced (UCM) landmass dataset³¹ has 2100 images representing 21 classes with 100 images per class. The UCM images are of size 256×256 pixels, at GSD of 1 foot/pixel. They are manually extracted images from the United States Geological Survey National Map Urban Area Imagery.

5.1.6 xView

The xView 2018 dataset³² is one of the largest publicly available datasets of overhead imagery. It contains around 1 million labeled object samples divided across 60 classes with the option of using either 3-band or 8-band imagery. The images were obtained from the WorldView-3 satellite at 0.3-m ground sample distance. The xView dataset is an imbalanced dataset that has some classes with a few instances and some with many instances.

5.1.7 DOTA

The DOTA dataset³³ is a large-scale dataset designed for the development and evaluation of object detectors for aerial imagery. It contains 2806 aerial images from different sensors and platforms. Image sizes range from about 800×800 to 4000×4000 pixels and contain objects exhibiting a wide variety of scales, orientations, and shapes. There are sixteen object categories in DOTA-v1.0, including plane, ship, and storage tank.

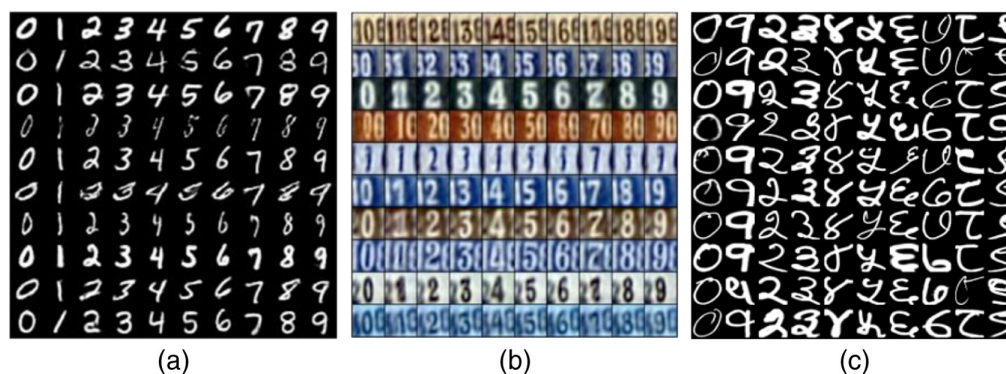


Fig. 7 Arabic numerals in (a) MNIST and (b) SVHN databases and Devanagari numerals in (c) DHC database.

5.1.8 NWPU

The NWPU-RESISC45 dataset³⁴ consists of 31,500 images divided into 45 scene classes. Each class includes 700 images that have a size of 256×256 pixels. The spatial resolution varies from about 30 to 0.2 m per pixel for most of the classes except for island, lake, mountain, and snowberg, which have lower spatial resolutions.

5.1.9 Remote Sensing Image Classification Benchmark

The Large-Scale Remote Sensing Image Classification Benchmark (RSI-CB) dataset³⁵ consists of two parts: the RSI-CB256 dataset and the RSI-CB128 dataset. They both have spatial resolutions of 0.3 to 3 m. RSI-CB256 contains 35 categories and more than 24,000 images of size 256×256 . RSI-CB128 contains 45 categories and more than 36,000 images of size 128×128 . Both datasets have six common categories: agricultural land, construction land and facilities, transportation and facilities, water and water conservancy facilities, woodland, and other land, and there are various subcategories within them.

5.2 Experiments and Results

Three transfer learning scenarios are considered in four experiments. Each experiment, including the considered scenario, experimental procedure, and experimental result, is provided below.

5.2.1 Experiment 1

In the first experiment, we consider the transfer learning from simulated data to measured data. For this scenario, SVHN is employed as the simulated data, as they were collected from printed house numbers, and MNIST is employed as the measured data, as they were hand-written digits. In the first stage, we performed source model training using cross-entropy as the loss function to be minimized. Then, we computed and saved the centers of source classes S_i , $i = 1, \dots, K$, in the feature space, where K is the number of source classes. Next, we performed source model training by minimizing the combined cross-entropy and center loss function. This completed the first stage of source model training.

In the second stage, adversarial DA, we used the source model as the initial target model followed by randomly selecting N target images for labeling, $10 \geq N \geq 0$ and then performed T-ADDA. When N equals 0, it reduces to ADDA. This process was repeated 10 times, and the results were averaged together. Finally, in the last stage, we combined the classification layer of the source model and the adapted target encoder to evaluate the performance of the target model before and after adaptation. Table 2 shows the common settings used in Experiment 1 and Experiment 2.

The accuracy of the cross entropy trained source classifier on source validation data is 92.86%, and the accuracy of the combined cross entropy and center loss trained source classifier on source validation data is 93.65%. Intuitively, these two values can be used as the upper bounds of target classifier performance after adaptation. Table 3 and Fig. 8 show the numerical and graphical results of the experiment. Clearly, T-ADDA is very effective with an improved performance of 3% to 18% over ADDA when N is increased from 1 to 10. In addition, we observe that the standard deviation decreases with increased N . This indicates that the target images that are selected for labeling have an impact on the adaptation result. How to effectively select target images for labeling within a given selection budget is a topic to investigate in the future. Also, the results showed the combined cross entropy and center loss consistently outperformed cross entropy loss by 2% to 4%. This indicates that a better clustered source domain is beneficial to performing DA via T-ADDA. Figure 9 shows two t -distributed stochastic neighbor embedding (t -SNE) visualizations of the source domain containing features of the ten digits classes, and the t -SNE visualization of features in the target domain. In Fig. 9(a), features are extracted from the cross-entropy trained source model, in Figs. 9(b) and 9(c), features are extracted from the combined cross-entropy and center loss trained source model. We note that the target features extracted from the cross-entropy trained source model are very similar to the ones extracted

Table 2 Settings employed in the first two experiments.

Datasets	Digit datasets (MNIST, SVHN, and DHC)	
Base model	LeNet++ based	
Input size	32 × 32 × 3	
Feature space dimension	256	
Epochs for source training	11	
Optimizer	SGD	
Source training learning rate	Cross-entropy	Center-loss
	0.001	0.001
λ	N/A	0.05
Discriminator learning rate	0.002	
GAN network learning rate	0.0002	
FCM network learning rate	0.002	
Epochs for adaptation	61	

Table 3 Numerical results of Experiment 1: SVHN to MNIST adaptation.

N source model	Source only	0 (ADDA)	2	4	6	8	10
Cross-entropy	0.599	0.715 ± 0.050	0.799 ± 0.026	0.846±0.012	0.877 ± 0.010	0.897 ± 0.009	0.907 ± 0.005
Center loss	0.616	0.755 ±.027	0.824 ± 0.020	0.874 ± 0.008	0.901 ± 0.008	0.915 ± 0.007	0.929 ± 0.005

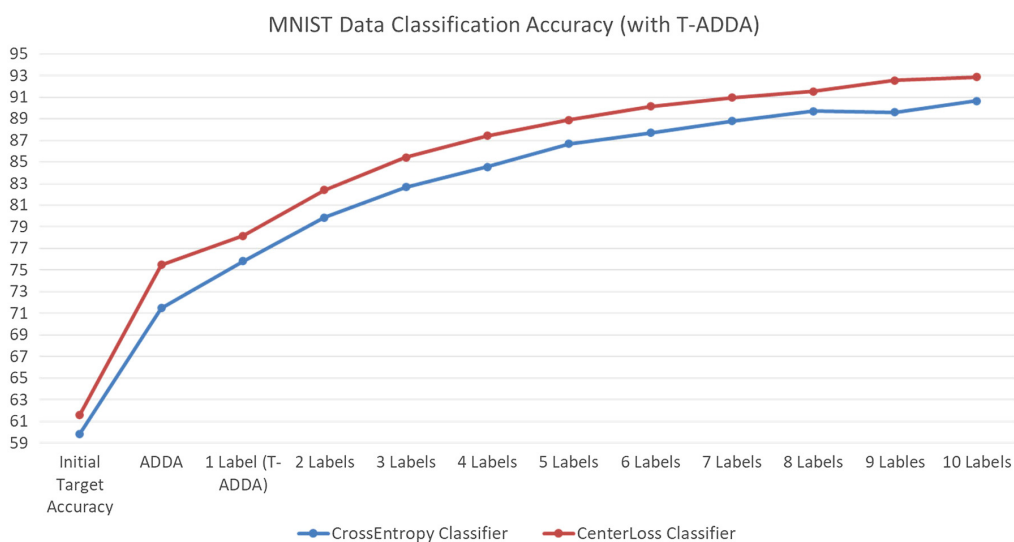


Fig. 8 Graphical result of Experiment 1.

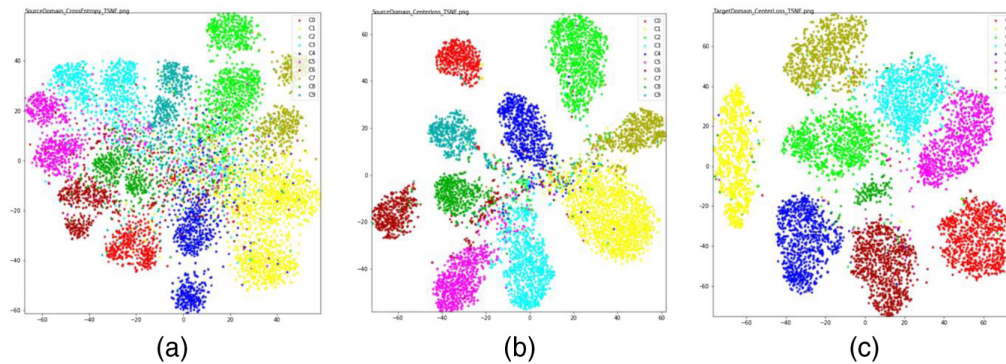


Fig. 9 t-SNE visualization of the (a) source features obtained from cross-entropy trained source model; (b) source features obtained from combined cross-entropy and center-loss trained source model; and (c) target features obtained from combined cross-entropy and center-loss trained source model.

from the combined cross-entropy and center loss trained source model; thus, it is not shown. It is worth noting that the target features are well separated and clustered in this case. Also notice that, in both cases, the performance of T-ADDA when 10 target images ($\sim 1\%$ of the total target images) are randomly selected for labeling approaches the upper bounds established by evaluating the source classifier on source validation data.

5.2.2 Experiment 2

The transfer learning scenario that learns how to classify a set of targets from a classifier that is trained to classify a different set of targets was considered in Ref. 36, in which the authors utilize one-labeled sample per class to transfer the classification from lung cancer to breast cancer. Two experiments, i.e., Experiments 2 and 4, are conducted under this scenario. In Experiment 2, SVHN and DHC datasets are employed. Though images of numerals from zero to nine are employed in both datasets, from Fig. 7 we see that only 0, 2, and 3 are visually similar and represent the same numerals. Others are either new to one another, i.e., 1, 4, 5, 7, and 8 in SVHN, or represent different numerals, i.e., 6 and 9 in SVHN.

Table 4 and Fig. 10 show the numerical and graphical results of the experiment. In this experiment, ADDA failed; this was expected as the domain shift is more likely to be large and the adapted target domain will not necessarily match the source domain in terms of class labels. On the other hand, T-ADDA is very effective with an improved performance from 18% to 80% over ADDA when N is increased from 1 to 10. However, the improvement of combined cross entropy and center loss trained source classifier over the cross-entropy trained source classifier is reduced. It is interesting to note that, when 10 target images per class ($< 0.6\%$) are randomly selected for labeling, the adaptation result from cross entropy trained source classifier reaches the performance upper bound, and the adaptation result from combined cross entropy and center loss trained source classifier exceeds the performance upper bound established by applying source classifier on source validation data. This is indicated by the bold values in Table 4.

The outstanding performance from T-ADDA of SVHN to DHC adaptation that exceeds both the performance of SVHN to MNIST adaptation shown in Experiment 01 and the intuitive

Table 4 Numerical results of Experiment 2: SVHN to DHC digits adaptation.

N source model	Source only	0 (ADDA)	2	4	6	8	10
Cross-entropy	0.142	0.153 \pm 0.049	0.651 \pm 0.094	0.855 \pm 0.016	0.898 \pm 0.013	0.915 \pm 0.014	0.929 \pm 0.008
Center loss	0.142	0.134 \pm .029	0.667 \pm 0.108	0.863 \pm 0.022	0.906 \pm 0.013	0.921 \pm 0.010	0.939 \pm 0.010

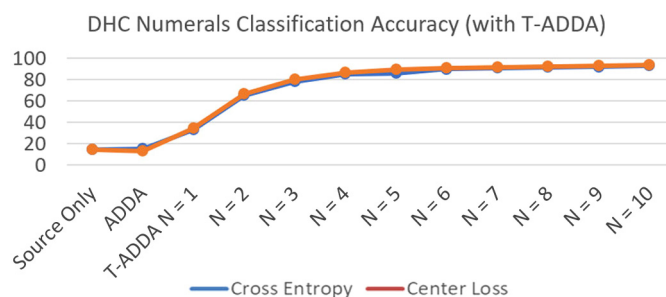


Fig. 10 Graphical result of Experiment 2.

performance upper bounds may be attributed to the lack of diversity of the DHC data within the same classes as compared with that in the MNIST and SVHN datasets. In other words, we anticipate that the DHC features encoded by the SVHN trained source encoder are very well separated and clustered. This is confirmed by the t -SNE visualization shown in Fig. 11(a). As a comparison, Fig. 11(b) shows the t -SNE visualization of features of the SVHN validation dataset. Clearly, DHC features are better separated than SVHN validation set features, which explains why the target classifier outperforms the source classifier when they are evaluated against target and source domain datasets, respectively.

To visualize the adaptation result, Fig. 12 shows the parametric t -SNE³⁷ visualizations of DHC features (a) before adaptation, (b) after ADDA adaptation, and (c) after T-ADDA adaptation. The parametric t -SNE model was trained by the features of the source training set extracted by the source encoder as shown in Fig. 12(d), to which the DHC features are adapted. Finally, it is interesting to get the classification performance for each target class by observing the confusion matrices resulting from the target model before and after T-ADDA. By observing Fig. 13(a), the confusion matrix resulting from the initial target model has relatively good performance for digits 0, 2, and 3, with classification accuracies of 0.68, 0.6, and 0.47, respectively. This is consistent with the observation that these three digits share very similar forms. After adaptation, distributions of all ten target classes are very close to the distributions of the corresponding ten source classes as observed from Fig. 13(b) with the lowest classification accuracy associated with numeral seven in the source domain. In this case, about 60% of numeral seven in the DHC dataset are correctly classified as numeral seven in the SVHN dataset, and about 20% are misclassified as numeral one in SVHN dataset.

5.2.3 Experiment 3

In the next two experiments, aerial images are used. In Experiment 3, we consider the transfer learning scenario from one imaging condition to another. For this scenario, we formed augmented xView and augmented DOTA datasets and performed DA from the former to the latter.

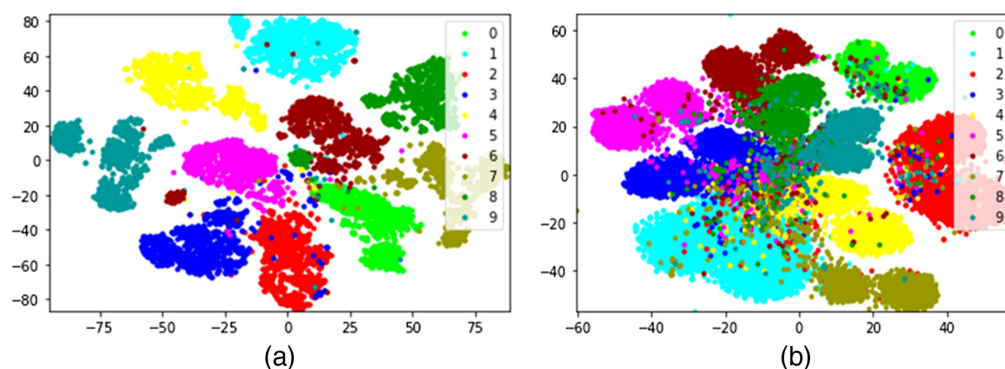


Fig. 11 The t -SNE visualization of target features encoded by source encoders (a) DHC features encoded by SVHN trained source encoder and (b) features of SVHN validation set encoded by SVHN trained source encoder.

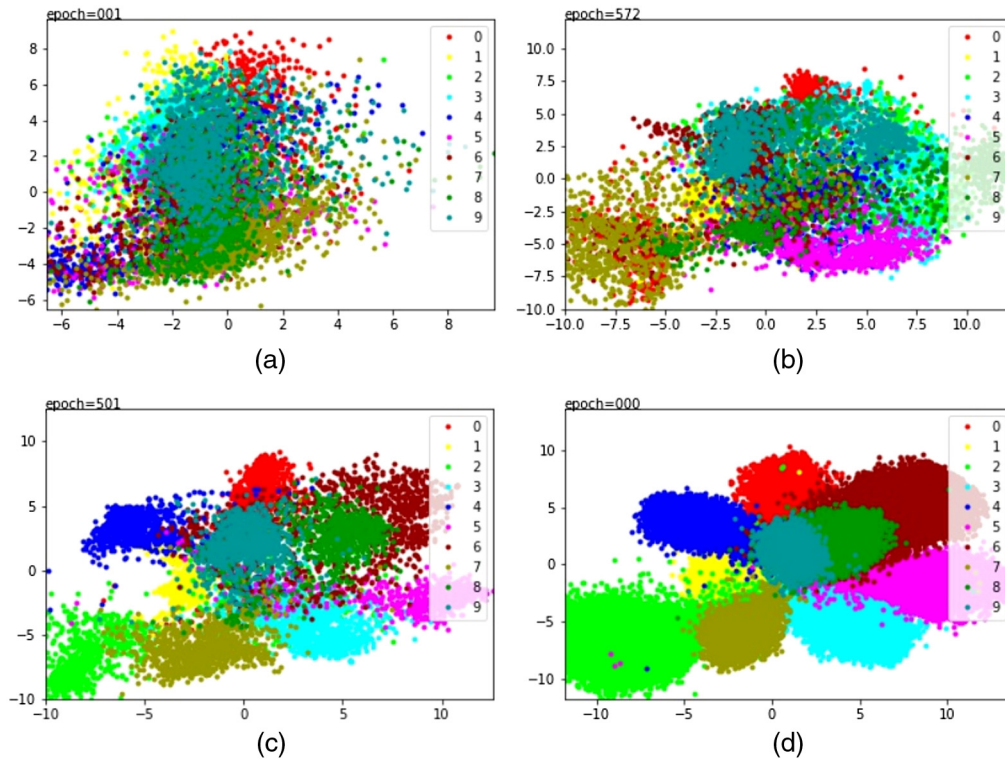


Fig. 12 DHC features visualized by *t*-SNE (a) before adaptation; (b) after ADDA adaptation; and (c) after T-ADDA adaptation ($N = 10$); (d) SVHN features encoded by source encoder.

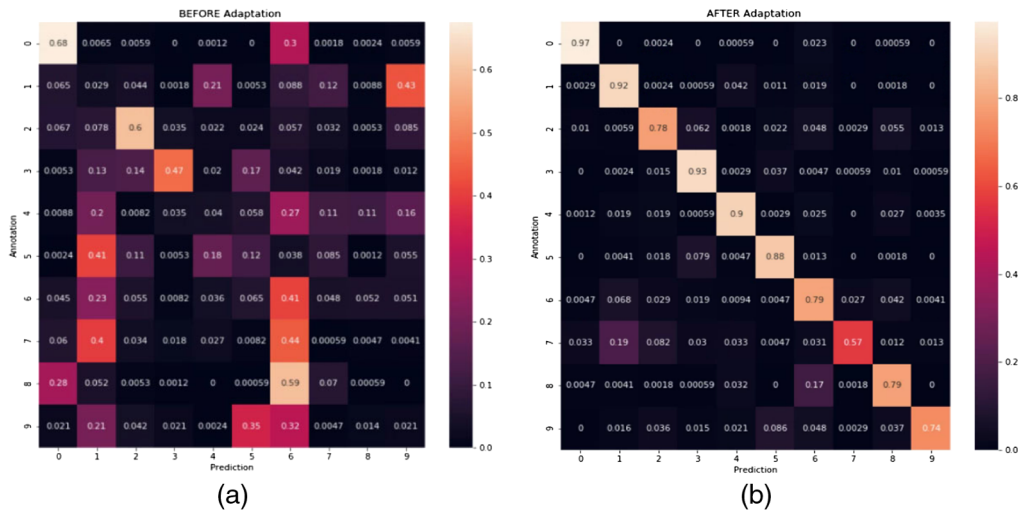


Fig. 13 Confusion matrix resulting from the target model (a) before and (b) after T-ADDA ($N = 10$).

The augmented datasets are formed by the following five classes from the xView and DOTA datasets: Airplane, Large Vehicle, Small Vehicle, Ship, and Storage Tanks, and the following three classes from the NWPU and RSI-CB datasets: Parking Lot, Runway, and Bridge. Several example images in selected classes from xView, DOTA, NWPU and RSI-CB are shown in Fig. 14. Table 5 shows the common settings used in Experiment 3 and Experiment 4. It is worth noting that, in this case, both the source and target domains have the same eight target classes. In Table 6, we list all eight classes and the number of images used in each class.

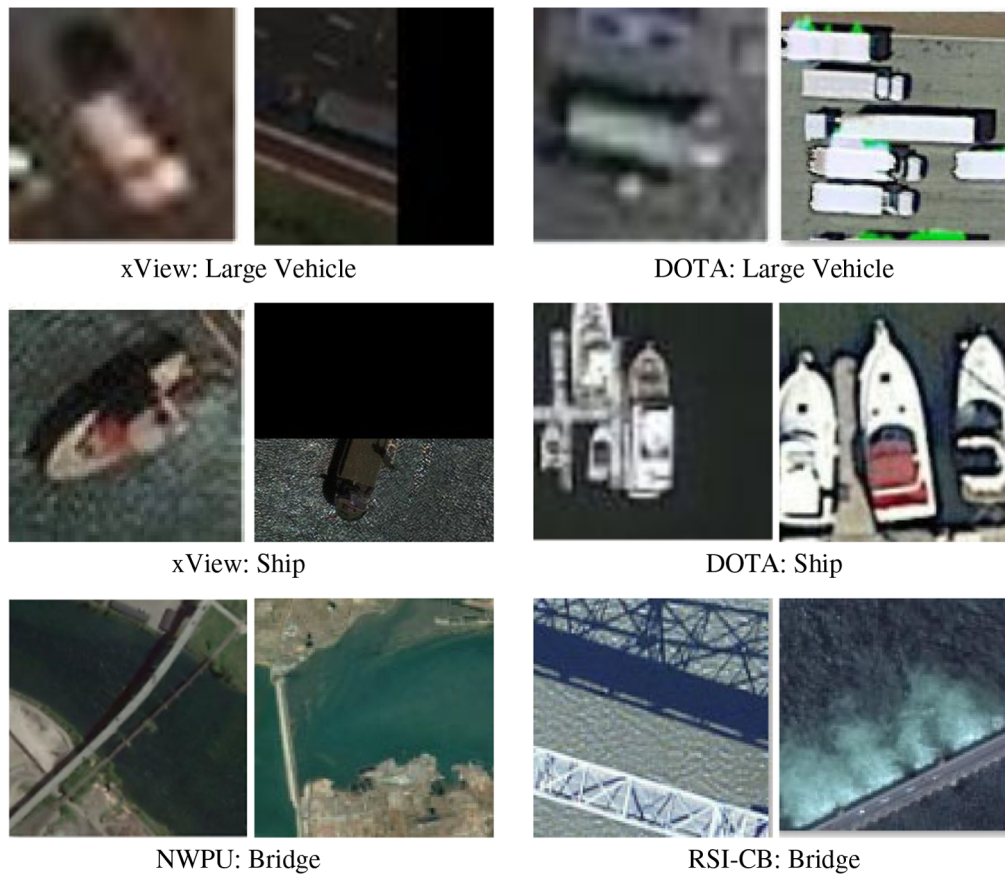


Fig. 14 Example image used in Experiment 3.

Table 5 Common settings in Experiments 3 and 4.

Datasets	Aerial datasets	
Base model	DenseNet	
Input size	224 × 224 × 3	
Feature space dimension	256	
Epochs for source training	61	
Optimizer	SGD	
Source training learning rate	Cross-entropy	Center-loss
	0.002	0.001
λ	N/A	0.05
Discriminator learning rate	0.001	
GAN network learning rate	0.0005	
FCM network learning rate	0.003	
Epochs for adaptation	61	

Table 6 Augmented xView and augmented DOTA datasets.

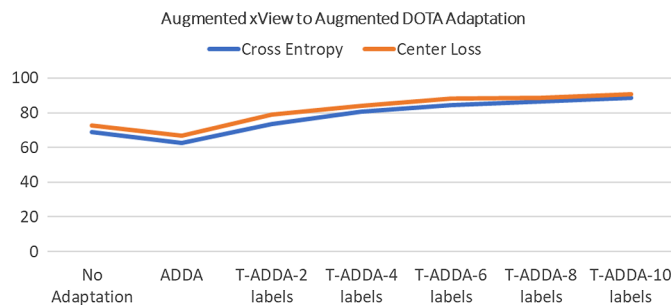
Class label	Augmented xView (source domain)	Augmented DOTA (target domain)
C0	xView:Airplane (1000)	DOTA:Airplane (1000)
C1	xView:LargeVehicle (3000)	DOTA:LargeVehicle (1000)
C2	xView:SmallVehicle (5000)	DOTA:SmallVehicle (1000)
C3	xView:Ship (2000)	DOTA:Ship (1000)
C4	xView:StorageTanks (1712)	DOTA:StorageTanks (1000)
C5	NWPU:ParkingLot (700)	RSI-CB:ParkingLot (1000)
C6	NWPU:Runway (700)	RSI-CB: Runway (1000)
C7	NWPU:Bridge (700)	RSI-CB Bridge(1000)

In this experiment, DenseNet was selected as the base model. After constructing the source model, we fine-tuned all parameters with the training set of the source data and employed the trained source encoder as the initial target encoder because the same target classes were involved in both the source and target domains. The number of labeled target images N is set to be 0, 2, 4, 6, 8, and 10, and for each value of N , we ran the experiment five times and reported the mean \pm standard deviation of the classification accuracies in the target domain. Table 7 and Fig. 15 show the numerical and graphical results of the experiment. Surprisingly, in this case, ADDA failed to improve target classification accuracy after adaptation. We believe that this is because the domain shift is not small enough for successful ADDA adaptation. However, T-ADDA still shows promising results when N is increased from 2 to 10. Again, the results showed that combined cross entropy and center loss consistently outperformed cross entropy loss by 1.2% to 5.5%. This indicates that a better clustered source domain is beneficial to performing DA via T-ADDA.

Also notice that in both cases, the performance of T-ADDA when 6 or above target images (0.6% to 1% of the total target images) are randomly selected for labeling exceeds the upper bounds established by evaluating the source classifier on source validation data. We conjecture

Table 7 Numerical results of Experiment 3: augmented xView to augmented DOTA adaptation.

N source Model	Source only	0 (ADDA)	2	4	6	8	10
Cross-entropy	0.690	0.627 \pm 0.021	0.736 \pm 0.026	0.807 \pm 0.017	0.844 \pm 0.018	0.866 \pm 0.007	0.886 \pm 0.017
Center loss	0.726	0.669 \pm 0.020	0.791 \pm 0.020	0.840 \pm 0.018	0.883 \pm 0.004	0.888 \pm 0.004	0.906 \pm 0.001

**Fig. 15** Graphical result of Experiment 3.

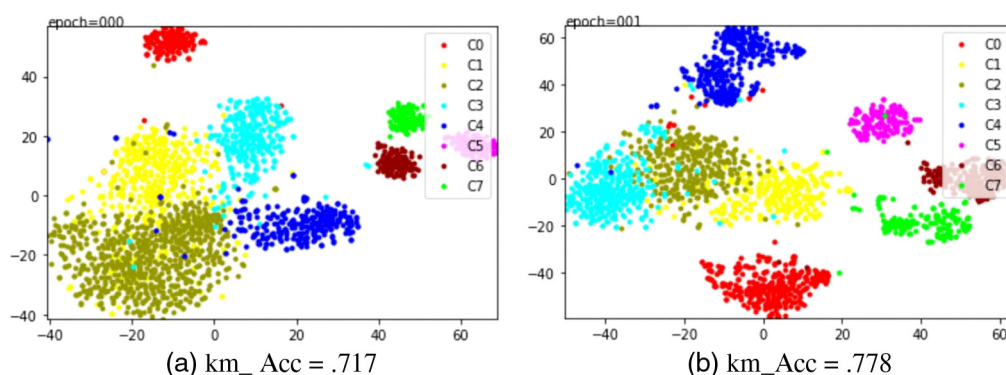


Fig. 16 (a) t -SNE visualization of source validation data features encoded by the source encoder and (b) t -SNE visualization of target features encoded by the source encoder.

that the reason for this is that target features are better separated and clustered than the source validation data features. This is confirmed by the t -SNE visualization plots provided in Fig. 16. To quantify the clustering quality, we applied K -means clustering³⁸ and computed the clustering accuracies, which are indicated under each t -SNE plot.

5.2.4 Experiment 4

In the last experiment, we again consider the scenario involving different classes of targets in the source and target domains. The datasets employed were AID and UCM. Ten classes from the AID dataset and ten classes from UCM dataset were selected for the experiment. Among them, five classes were common in both the AID and UCM datasets, and five classes were unique to each dataset.

Table 8 shows the ten source and target domain classes along with the number of images in each class. Sample images from some of the common classes are shown in Fig. 17, and those from some of the unique classes are shown in Fig. 18.

For source model training, we first randomly split the source images into training (85%) and validation (15%) sets. All images in the training set were used for source model training. After training, the source model was evaluated against the source validation set as well as the entire target domain images, which were the 10 classes from the UCM dataset, and there were 100 images in each target class. For T-ADDA adaptation, the target encoder was initialized with

Table 8 Source and target classes used in Experiment 4.

Class label	AID (source domain)	UCM (target domain)
C0	Baseball field (220)	Baseball field (100)
C1	Beach (400)	Beach (100)
C2	Medium residential (290)	Medium residential (100)
C3	Parking lot (390)	Parking lot (100)
C4	Sparse residential (300)	Sparse residential (100)
C5	Church (240)	Airplane (100)
C6	Desert (302)	Golf course (100)
C7	Industrial (390)	Runway (100)
C8	Mountain (340)	Storage tanks (100)
C9	Port (380)	Tennis (100)

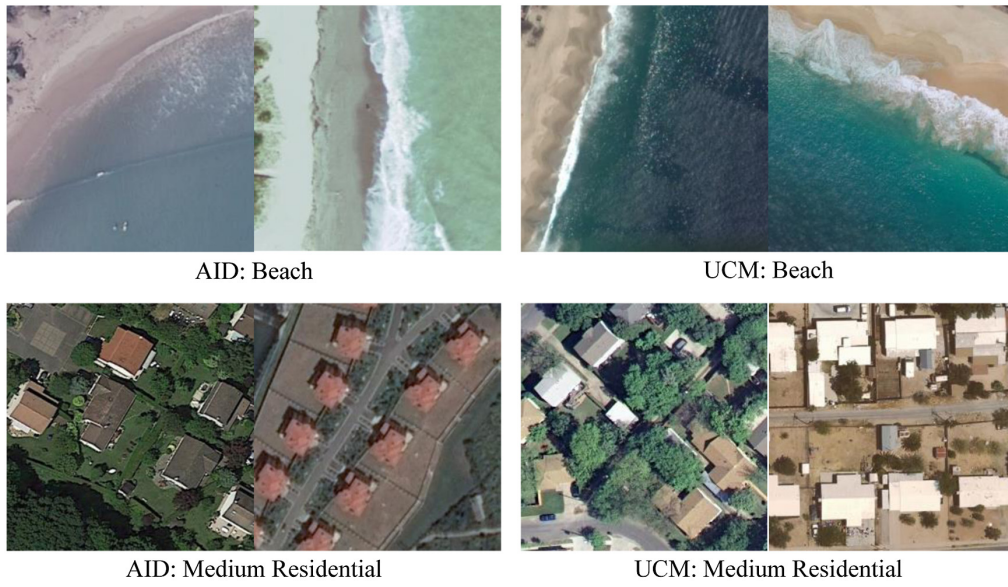


Fig. 17 Example images of some common classes in (a) AID and (b) UCM.



Fig. 18 Example images of some unique classes in (a) AID and (b) UCM.

the ImageNet pretrained DenseNet model with the feature space dimension equals to 256 and the number of classes set to 10. For each number of labeled target images $N = 0, 2, \dots, 10$, we ran the experiment five times and reported the mean \pm standard deviation of the classification accuracies in the target domain.

The accuracy of the cross entropy trained source classifier on source validation data was 98.4%, and the accuracy of the combined cross entropy and center loss trained source classifier on source validation data was 98.8%. Intuitively, these two values can be used as the upper

Table 9 Numerical results of Experiment 4: AID to UCM adaptation.

N Source model	Source only	0 (ADDA)	2	4	6	8	10
Cross-entropy	0.209	0.203 \pm 0.060	0.645 \pm 0.077	0.860 \pm 0.051	0.903 \pm 0.013	0.915 \pm 0.007	0.937 \pm 0.012
Center loss	0.185	0.148 \pm 0.036	0.633 \pm 0.088	0.850 \pm 0.018	0.924 \pm 0.017	0.920 \pm 0.008	0.953 \pm 0.014

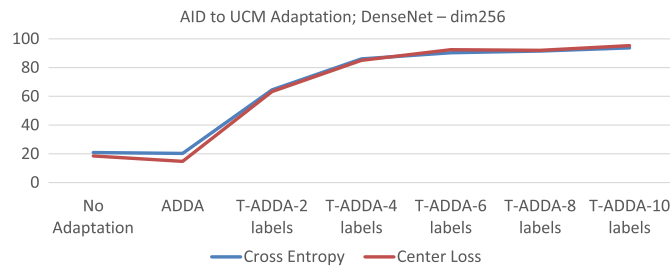


Fig. 19 Graphical result of Experiment 4.

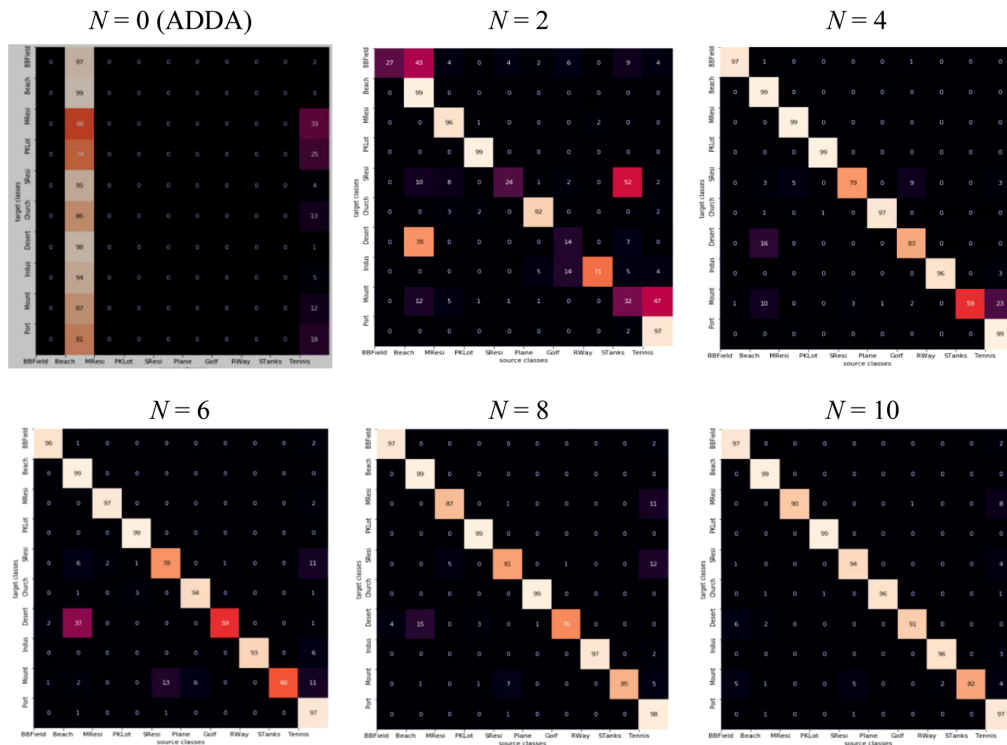


Fig. 20 Confusion matrices for $N = 0, 2, \dots, 10$.

bounds of the target classifier performance after adaptation. Table 9 and Fig. 19 show the numerical and graphical results of the experiment. As expected, ADDA failed to improve the target classification accuracy after adaptation due to different target classes in the source and target domains. On the other hand, T-ADDA shows promising results even when N is equal to 2, with the accuracy improving from about 20% to above 60%. However, in this experiment, the advantage of the combined cross entropy and center loss was not as clear as in other experiments. To see the classification accuracy of each individual class, in Fig. 20, we provide the confusion matrices for $N = 0, 2, \dots, 10$. From Fig. 20, no clear difference between the five common classes and the five classes unique to each domain is observed.

6 Conclusions

The paper describes a robust DA framework, T-ADDA. It is a semi-supervised approach that provides the required robustness for scenarios in which the initial domain shift is large. Digit image datasets and real AID were employed to demonstrate the proposed T-ADDA framework. Three scenarios were tested including transferring knowledge from simulated data to measure data (SVHN to MNIST), transferring knowledge from one set of targets to another and different set of targets (SVHN to DHC and AID to UCM), and transferring knowledge from one imaging condition/sensor to new imaging conditions or sensors (augmented xView to augmented DOTA). Our experimental results show that T-ADDA is very effective in all three scenarios. When the available labeled target images are as few as two images per class, T-ADDA increases performance over ADDA by at least 8% in the simulated-to-measured scenario, 12% in the sensor-to-sensor scenario, and over 40% in the target-to-target scenario.

Acknowledgments

Intelligent Fusion Technology Inc. was supported under contract No. FA8649-20-P0-352. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the United States Air Force.

References

1. E. Blasch et al., "Wide-area motion imagery (WAMI) exploitation tools for enhanced situation awareness," in *IEEE Appl. Imagery Pattern Recognit. Workshop* (2012).
2. Y. Zheng, E. Blasch, and Z. Liu, *Multispectral Image Fusion and Colorization*, SPIE Press, Bellingham, Washington, DC (2018).
3. R. Niu et al., "Joint sparsity based heterogeneous data-level fusion for target detection and estimation," *Proc. SPIE* **10196**, 101960E (2017).
4. E. Blasch et al., "Prediction of compression-induced image interpretability degradation," *Opt. Eng.* **57**(4), 043108 (2018).
5. H-M. Chen and P.K. Varshney, "Automatic registration of infrared and millimeter-wave images for concealed weapon detection," *Proc. SPIE* **3719**, 152–160 (1999).
6. Y. Wu et al., "Multiple source data fusion via sparse representation for robust visual tracking," in *Int. Conf. Inf. Fusion* (2011).
7. E. Blasch, E. Bosse, and D. A. Lambert, *High-Level Information Fusion Management and Systems Design*, Artech House, Norwood, MA (2012).
8. D. Shen et al., "A joint manifold leaning-based framework for heterogeneous upstream data fusion," *J. Algor. Comput. Technol.* **12**(4), 311–332 (2018).
9. D. Shen et al., "Network survivability oriented Markov games (NSOMG) in wideband satellite communications," in *IEEE/AIAA Digital Avionics Syst. Conf.* (2014).
10. Y. Duan et al., "Feasibility of an Interpretability metric for LIDAR Data," *Proc. SPIE* **10645**, 1064506 (2018).

11. H-M. Chen, G. Chen, and E. Blasch, "On the development of a classification based automated motion imagery interpretability prediction," *Lect. Notes Comput. Sci.* **12668**, 75–88 (2021).
12. B. Jia et al., "Space object classification using deep neural networks," in *IEEE Aerospace Conf.* (2018).
13. A. Savakis et al., "Change detection in satellite imagery with region proposal networks," *Defense Syst. Inf. Anal. Center (DSIAC) J.* **6**(4), 23–28 (2019).
14. U. Majumder, E. Blasch, and D. Garren, *Deep Learning for Radar and Communications Automatic Target Recognition*, Artech House, Massachusetts (2020).
15. A. Torralba and A. Efros, "Unbiased look at dataset bias," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.* (2011).
16. J. Lu et al., "Deep learning based domain adaptation with data fusion for aerial image data analysis," *Lect. Notes Comput. Sci.* **12668**, 118–133 (2021).
17. B. Fernando et al., "Unsupervised visual domain adaptation using subspace alignment," in *Int. Conf. Comput. Vision* (2013).
18. B. Sun and K. Saenko, "Deep CORAL: correlation alignment for deep domain adaptation," *Lect. Notes Comput. Sci.* **9915**, 443–450 (2016).
19. B. Minnehan and A. Savakis, "Deep domain adaptation with manifold aligned label transfer," *Mach. Vis. Appl.* **30**, 473–485 (2019).
20. Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.* **17**(1), 2096–2030 (2016).
21. Y. Zhang et al., "Domain-symmetric networks for adversarial domain adaptation," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.* (2019).
22. E. Tzeng et al., "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.* (2017).
23. K. Saito et al., "Semi-supervised domain adaptation via minimax entropy," in *Proc. IEEE Int. Conf. Comput. Vision* (2019).
24. Y. Wen et al., "A discriminative feature learning approach for deep face recognition," *Lect. Notes Comput. Sci.* **9911**, 499–515 (2016).
25. O. Russakovsky et al., "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vision* **115**(3), 211–252 (2015).
26. G. Huang et al., "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 4700–4708 (2017).
27. <http://yann.lecun.com/exdb/mnist/>.
28. <http://ufldl.stanford.edu/housenumbers/>.
29. <https://archive.ics.uci.edu/ml/datasets/Devanagari+Handwritten+Character+Dataset>.
30. G.-S. Xia et al., "AID: a benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.* **55**(7), 3965–3981 (2017).
31. Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst.*, pp. 270–279 (2010).
32. D. Lam et al., "xview: objects in context in overhead imagery," <http://xviewdataset.org/> (2018).
33. G-S. Xia et al., "DOTA: a large-scale dataset for object detection in aerial images," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 3974–3983 (2018).
34. G. Cheng, J. Han, and X. Lu. "Remote sensing image scene classification: benchmark and state of the art," *Proc. IEEE* **105**(10), 1865–1883 (2017).
35. H. Li et al., "RSI-CB: a large scale remote sensing image classification benchmark via crowdsourcing data," <https://github.com/lehaifeng/RSI-CB> (2017).
36. O. Mendoza-Schrock et al., "Manifold transfer subspace learning (MTSL) for high dimensional data—applications to handwritten digits and health informatics," in *Int. Conf. Image Process., Comput. Vision, and Pattern Recognit.* (2017).
37. L. J. P. van der Maaten, "Learning a parametric embedding by preserving local structure," in *Artif. Intell. and Stat.* (2009).
38. S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory* **28**(2), 129–137 (1982).

Hua-Mei Chen received his BS degree in electro-physics from National Jiaotong University, Taiwan, and his master's degree and PhD in electrical engineering from Syracuse University. He

was an assistant professor in the Department of Computer Science and Engineering, University of Texas at Arlington from 2002 to 2008. He joined Intelligent Fusion Technology Inc. in 2015 and was promoted as a principal scientist in 2017. His research areas are in digital signal/image/video processing and in machine learning.

Andreas Savakis is professor of computer engineering and director of the Center for Human-aware Artificial Intelligence at Rochester Institute of Technology. He received his PhD in electrical and computer engineering with a mathematics minor from North Carolina State University and was with the Kodak Research Labs before joining RIT. His research has generated over 120 publications and 12 U.S. patents. His interests include computer vision, deep learning, DA, visual tracking, pose estimation, and scene analysis.

Ashley Diehl is a research electronics engineer with the Air Force Research Laboratory (AFRL) in Dayton, Ohio. She began her career in the early 2010s as an undergraduate researcher. For the last eight years, her work has focused on designing classification frameworks for vibrometry and EO imagery. Her research interests include hierarchical learning, transfer learning, and few-shot learning. She is also a doctoral candidate in the electrical engineering program at Wright State University.

Erik Blasch is an Air Force Research Laboratory Program officer. He received his BS degree in mechanical engineering from MIT and his PhD in electrical engineering from Wright State with master's degrees in ME, industrial Eng., EE, medicine, military studies, economics, and business. His assignments include USAF Reserve colonel (ret) and adjunct associate professor with interests in information-fusion and human-machine integration. He is the author of 900+ papers, 35 patents, and 8 books. He is an AIAA associate, IEEE member, and SPIE fellow.

Sixiao Wei received his MS degree from the Department of Computer and Information Sciences, Towson University in 2014 and his BS degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 2010. Currently, he has been a research scientist at Intelligent Fusion Technology, Inc., Germantown, Maryland, since summer 2014. His research interests are wireless cyber security, computer networking, cloud computing, and big data.

Genshe Chen is the CTO of Intelligent Fusion Technology, Inc. He received his BS and MS degrees in electrical engineering and his PhD in aerospace engineering in 1989, 1991, and 1994, respectively, all from Northwestern Polytechnical University, Xian, China. He has been the PM/PI/Technical Lead for 100+ projects. He served as a technical conference chair of SPIE DSS Sensors and Systems for Space Applications for many years. He is a senior member of SPIE.