Deep Learning Based Domain Adaptation with Data Fusion for Aerial Image Data Analysis

Jingyang Lu[1], Chenggang Yu[1], Erik Blasch[2], Roman Ilin[2], Hua-mei Chen[1], Dan Shen[1], Nichole Sullivan[1], Genshe Chen[1](✉), and Robert Kozma[3]

[1] Intelligent Fusion Technology, Inc., Germantown, MD 20876, USA
gchen@intfusiontech.com
[2] Air Force Research Laboratory, Dayton, OH 45435, USA
[3] The University of Memphis, Memphis, TN 38152, USA

Abstract. Current Artificial Intelligence (AI) machine learning approaches perform well with similar sensors for data collection, training, and testing. The ability to learn and analyze data from multiple sources would enhance capabilities for Artificial Intelligence (AI) systems. This paper presents a deep learning-based multi-source self-correcting approach to fuse data with different modalities. The data-level fusion approach maximizes the capability to detect unanticipated events/targets augmented with machine learning methods. The proposed Domain Adaptation for Efficient Learning Fusion (DAELF) deep neural network adapts to changes of the input distribution allowing for self-correcting of multiple source classification and fusion. When supported by a distributed computing hierarchy, the proposed DAELF scales up in neural network size and out in geographical span. The design of DAELF includes various types of data fusion, including decision-level and feature-level data fusion. The results of DAELF highlight that feature-level fusion outperforms other approaches in terms of classification accuracy for the digit data and the Aerial Image Data analysis.

Keywords: Motion imagery · Domain adaptation · Aerial image analysis

1 Introduction

Deep learning, as an element of machine learning (ML), has revolutionized many traditional data fusion approaches including wavelet fusion [1, 2], manifold fusion [3, 4] and target tracking [5–7]. Data fusion approaches include data-level, feature-level, and decision-level fusion for such applications as audio-video [8], video-text [9], and visual-infrared fusion [10]. The data fusion methods for aerial sensing extend to situation awareness [11] and temporal awareness [12]. The combination of deep learning-based multi-source analysis and data-level fusion provide a self-correcting approach to combine data of different modalities. Cognitively-motivated approaches provide flexibility and robustness of sensory fusion required under partially unknown conditions and in response to unexpected scenarios [4, 14]. Both machine learning and heterogeneous data-level fusion can enhance detection of unanticipated events/targets through the use of

A. Del Bimbo et al. (Eds.): ICPR 2020 Workshops, LNCS 12668, pp. 118–133, 2021.
https://doi.org/10.1007/978-3-030-68793-9_9

domain adaptation, see Fig. 1. The proposed ***Domain Adaptation for Efficient Learning Fusion*** (DAELF) deep neural network approach adapts to changes of the input distribution allowing self-correcting multiple source classification and fusion. When supported by a scalable distributed computing hierarchy, DAELF *scales up* in neural network size, *scales out* in geographical span, and *scales across* modalities.

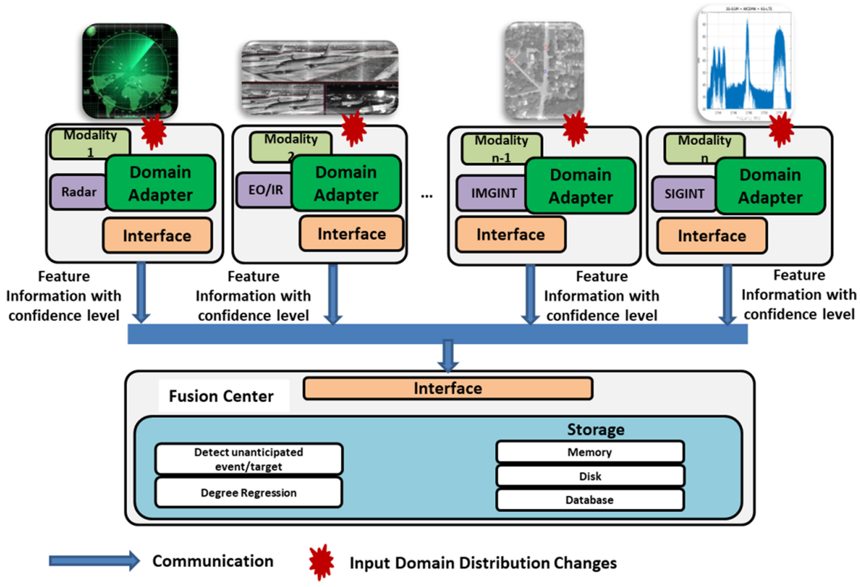


Fig. 1. Machine Learning based Domain Adaptation for Multiple Source Classification and Fusion

Generalizing models learned on one domain to another novel domain has been a major challenge in the quest for universal object recognition, especially for aerial motion imagery [15]. The performance of the learned models degrades significantly when testing on novel domains due to the presence of *domain shift* [16]. In Fig. 1, the proposed Domain Adaptation for Efficient Learning Fusion (DAELF) highlights heterogeneous data fusion for unanticipated event/target detection. The data from different sensing modalities are processed through a ML-based domain adapter, which can leverage unsupervised data to bring the source and target distributions closer in a learned joint feature space. DAELF includes a symbiotic relationship between the learned embedding and a generative adversarial network (GAN). Note, the GAN in DAELF supports joint multimodal analysis as contrasted to traditional GAN methods, which use the adversarial framework for generating realistic data and retraining deep models with such synthetic data [17, 18].

Based on the single source unsupervised domain adaptation (UDA), DAELF is an innovative approach to align multiple source domains with the target domain, which incorporates the moment Matching Component (MC) with GANs into deep neural network (DNN) to train the model in an end-to-end fashion. The key advantages of the DAELF approach include:

- Learning features that combine (i) discriminativeness and (ii) domain-invariance achieved by jointly optimizing the underlying features as well as two discriminative classifiers operating on these features. Namely, (i) the *label predictor* that predicts class labels and is used both during training and at test time and (ii) the *domain classifier* that discriminates between the source and the target domains during training;
- Adapting classifiers to the target domain with different distributions *without retraining* new input data. DAELF leverages unsupervised data to bring the source and target domain distributions closer in a learned joint feature space;
- Leveraging an adversarial data generation approach to directly learn the shared feature embedding using labeled data from source domain and unlabeled data from target domain. The novelty of the DAELF approach is in using a *joint generative discriminative method*: the embeddings are learned using a combination of classification loss and data generation procedure that is modeled using a variant of GANs. Then, given the availability of multiple sources data, which aims to transfer knowledge learned from multiple labeled source domains to an unlabeled target domain by dynamically aligning moments of their feature distribution; and
- Incorporating decision-level and feature-level fusion for enhanced target/event detection robust performance.

Deep learning has been utilized to uncover rich, hierarchical models [19] that represent probability distributions of various labeled data in different domains such as natural aerial images, audio waveforms containing speech, and symbols in natural language corpora. For a problem lacking labeled data, it may be still possible to obtain training sets that are large enough for training large-scale deep models, but they suffer from the *domain shift* in data from the trained data to that of the actual data encountered at the application time.

To account for domain shift, methods are needed to learn features that combine discriminativeness and domain-invariance in order to address environmental changes. While the parameters of the classifier are optimized in order to minimize errors on the training set, the parameters of the underlying deep feature mapping are optimized in order to minimize the loss of the label classifier and to maximize the loss of the domain classifier. The label classifier update works adversarially to the domain classifier, and it encourages domain-invariant features to emerge in the course of the optimization.

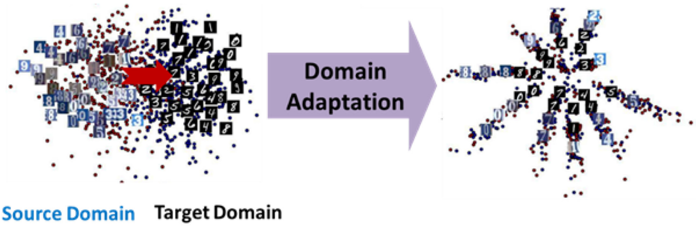
The rest of the paper is as follows. Section 2 describes the methods of domain adaptation with adversarial networks. Section 3 provides results using the Aerial Image Data (AID) dataset and Sect. 4 concludes the paper.

2 Methods

For data analysis, consider the tasks where $\boldsymbol{X} = \{x_i\}_{i=1}^N$ is the input space and $\boldsymbol{Y} = \{y_i\}_{i=1}^N$ is the label space. It is assumed that there exists a source-domain distribution $\mathcal{S}(x, y)$ and target-domain distribution $\mathcal{T}(x, y)$ over the samples in $\boldsymbol{X}$. There are three types of *domain adaptation* shown in Table 1.

Table 1. Types of domain adaptation

	Source and Target domain	Source and Target tasks
Inductive Domain Adaptation	Same	Different but related
Transductive Domain Adaptation	Different but related	Same
Unsupervised Domain Adaptation	Different but related	Different but related



(a) (b)

Fig. 2. Illustration of Domain Adaptation of samples of the same class from Source and Target Domains that are (a) sep-arated and (b) close to each other.

For unsupervised domain adaptation, the source distribution using labeled data from $\boldsymbol{X}$ is only accessible for the machine model training. The problem of unsupervised domain adaptation (Fig. 2) can be stated as learning a predictor that is optimal in the joint distribution space by using labeled source domain data and unlabeled target domain data sampled from $\boldsymbol{X}$. The objective is to learn an embedding map $F : \boldsymbol{X} \rightarrow \mathbb{R}^{\boldsymbol{d}}$ and a prediction function $C : \mathbb{R}^{\boldsymbol{d}} \rightarrow \boldsymbol{Y}$. In DAELF, both F and C are modeled as deep neural networks. The predictor has access to the labels only for the data sampled from source domain and not from the target domain during the training process, so F implicitly learns the domain shift between source domain distribution $\mathcal{S}(x, y)$ and target domain distribution $\mathcal{T}(x, y)$. Likewise, a GAN-based approach is proposed to bridge the gap between the source and target domains. The target can be accomplished by using both generative and discriminative process which takes as much information as possible to learn the invariant features existing between the source and target domain.

2.1 Generative Adversarial Network

Generative Adversarial Networks (GANs) [1, 18] are utilized in many machine learning methods in domain adaptation. In a traditional GAN, two competing mappings are learned, the discriminator D and the generator G, both of which are modeled as deep neural networks. G and D play minmax the game, where D tries to classify the generated samples as fake and G tries to fool D by producing examples that are as realistic as possible. In order to train a GAN, the following optimization problem is solved in an iterative manner,

$$\min_{G} \max_{D} V(D, G) = E_{x \sim p_{data}}\big[\log D(x)\big] + E_{z \sim p_{noise}}\big[\log(1 - D(G(z))\big] \qquad (1)$$

where $D(x)$ represents the probability that x comes from the real data distribution rather than the distribution modeled by the generator $G(z)$, where z are noise variables. As an extension to traditional GANs, *conditional GANs* enable conditioning the generator and discriminator mappings on additional data such as a class label or an embedding. They have been shown to generate data on the class label or the embedding respectively. As in training a traditional GAN, the conditional GAN involves optimizing the following minimax objective, conditioned on the variable y:

$$\min_G \max_D E_{x \sim p_{data}}(\log(D(x|y))) + E_{z \sim p_{noise}} \log(1 - D(G(z|y))) \tag{2}$$

Building on the development of traditional GANs, conditional GANs, and multi-modal GANs, the next sections highlights a domain adaptation approach using GANs.

2.2 Domain-Adversarial Neural Networks

The proposed DAELF is designed by employing a variant of the conditional GAN called *Auxiliary Classifier GAN* (AC-GAN) [20] by Sankaranarayanan, et al., where the discriminator is modeled as a multi-class classifier instead of providing conditioning information at the input, as shown in Fig. 3.

The AC-GAN set up for the domain adaptation is as follows:

- *Sampling*: Given a real data set x as input to F, the input to the generator network G is $x_g = [F(x), z, l]$, which is a concatenated version of the encoder embedding $F(x)$, a random noise vector $z \in \mathbb{R}^d$ sampled from $N(0, 1)$ and a one-hot encoding of the class label, $l \in \{0, 1\}^{(N_c+1)}$ with N_c real classes and $\{N_c + 1\}$ being the fake class. For all target samples, since the class labels are unknown, l is set as the one-hot encoding of the fake class $\{N_c + 1\}$.
- *Classifier*: The classifier network C that takes as input the embedding generated by F and predicts a multiclass distribution $C(x)$, i.e. the class probability distribution of the input x, which is modeled as a N_c-way classifier.
- *Discriminator*: The discriminator mapping D takes the real input data x or the generated input $G(x_g)$ as input and outputs two distributions: (1) $D_{data}(x)$: the probability of the input being real, which is modeled as a binary classifier, and (2) $D_{cls}(x)$: the class probability distribution of the input x, which is modeled as a N_c-way classifier. To clarify the notation, $D_{cls}(x)_y$ implies the probability assigned by the classifier mapping D_{cls} from input x to y. It should be noted that, for target domain data, since class labels are unknown, only D_{data} is used to backpropagate the gradients. Please refer to [20] for additional details. It is worth mentioning that in order to better improve the training performance, the target domain data is also used to update the generator (G), which is denoted as follows,

$$L_G = \min_G E_{x \sim s} - \log\left(D_{cLs}(G(x_g))_y\right) + \log(1 - D_{data}(G(x_g))) + \log(1 - D_{data}(G(h_{g_i}))) \tag{3}$$

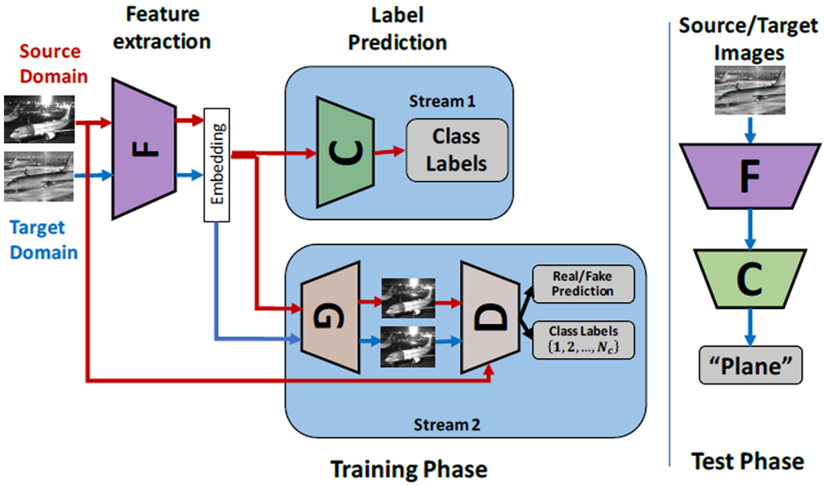


Fig. 3. Illustration of DAELF AC-GAN Approach [adapted from 20] ("F" denotes Feature Extraction Network, "C" denotes Label Prediction Network, "G" denotes Generator Network, and "D" denotes Discriminator Network)

2.3 Fusion Network Model

A fusion network model integrates two sources of input. For clarity, **netF** and **netC** is equivalent to **F** and **C** denoted in Fig. 3. If each sensor has a domain adaptation network (**netF**) followed by a *centralized fusion network* (**netC**) as in Fig. 4. Each **netF** is first trained by a different pair from the source/target dataset. See Table 3 in the Results section, where two pair datasets *MNIST* → *USPS* and *SVHN* → *JP* are used to demonstrate the validity of the proposed DAELF sensor fusion. The weights of the two netFs are then brought into the centralized fusion network and the netC is trained by using two source datasets. The two networks {**netF** and **netC**} are trained and the whole fusion network is able to predict both target domain inputs.

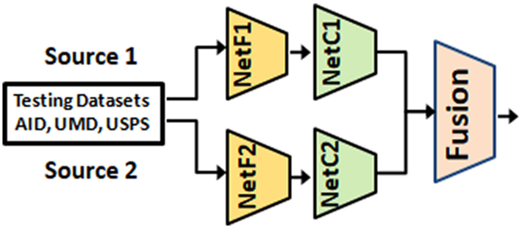


Fig. 4. Decision-Level Fusion for Multiple Sensors as GTA or Source-only fusion network

Two fusion approaches widely used are decision-level fusion (DLF) shown in Fig. 4 and feature-level fusion (FLF) shown in Fig. 5. The FLF consists of two separately trained feature networks (netF1 and netF2) followed by one decision network that takes the concatenation of the outputs of the two feature networks (i.e., two embedding feature vectors) as inputs. The decision network needs to be trained by the two source domain training datasets with matched class labels.

Compared to feature-level fusion (FLF), decision-level fusion (DLF) does not need a second training. DLF consists of the two classification networks, each was formed by a feature network (netF) and a decision network (netC) that were trained by the ***Generate to Adapt*** (GTA) method [20] with using one pair of source/target domain data. DAELF

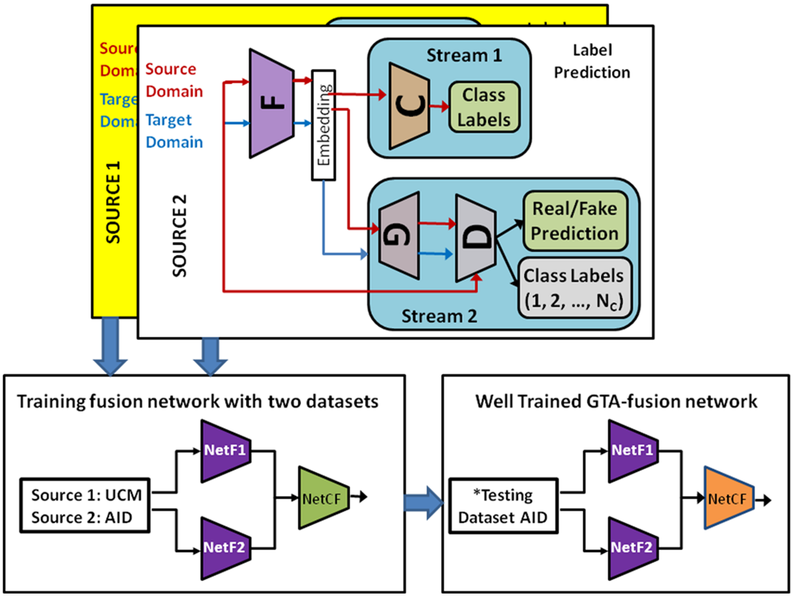


Fig. 5. Feature-level Data Fusion for Multiple Sensors

employs a strategy to predict the input images' class label according to the outputs of the two decision networks, which is explained as follows.

The last layer of each *netCF* has 10 outputs that represent the class labels of 10 digits from 0 to 9. A class label d described in Eq. (4) is predicted if the corresponding output value is the maximum among the 10 outputs. In order to make a final prediction D from the predictions of the two decision networks, DAELF assesses each prediction's reliability by computing an entropy H using Eq. (5), where p_0 through p_9 are 10 output values from one *netCF*. The final prediction would be the one that has a smaller entropy (Eq. (6)).

$$d = argmax(p_i, i = 0, 1, \ldots 9) \tag{4}$$

$$H = -\sum_{i=0}^{9} p_i \log(p_i) \tag{5}$$

$$D = \begin{cases} d_1 & if \;\; H_1 < H_2 \\ d_2 & if \;\; H_2 < H_1 \end{cases} \tag{6}$$

DAELF uses the two separately trained neural networks to form a fusion network to simulate a two-sensor two-modality system (Fig. 5). Because there are two sensors used to detect the same object, then it is required that every two images feeding to the fusion network must have an identical class label, which is also the true output of the network.

The next section demonstrates the DAELF approach for different scenarios.

3 Simulation Results

3.1 Classification of Digits Dataset

Comparing to other standard image datasets, the three DIGITS datasets, USPS (U.S. Postal Service), MNIST (Modified National Institute of Standards and Technology database), and SVHN (Google Street View House Number) are simple, and the domain shift from one to the other is relatively small [16]. The datasets are widely used as the first set of data in the testing of various domain adaptation algorithms. The original algorithm has two ways in training a network to classify images of handwriting digits:

1) **Source-only** that trains a network (formed by netF and netCF) with labeled source training data only;
2) ***Generate to Adapt*** (GTA) that trains netF and netCF separately. NetF is trained by labeled source training data and unlabeled target training data through a GAN, while netC is trained by source training data only.

A target testing dataset was used to evaluate the performance of the network (netF plus netC) trained by the two different ways. Various datasets exist for comparison: MNIST, USPS, and SVHN. Table 2 compares the classification accuracies obtained to that of the results by using Source-only approach. In all three domain adaptation cases, the network trained by GTA significantly outperformed the network trained by the source-only method. Through inspecting the clustering of embedding features, we found that it is possible to achieve an accuracy as high as 96% if we are able to modify the model selection strategy. This potential improvement by *model selection* is discussed in the next section.

Table 2. Performance Comparison

	MNIST→USPS	USPS→MNIST	SVHN→MNIST
Source only	79.1 ± 0.9	57.1 ± 1.7	60.3 ± 1.5
GTA	95.3 ± 0.7	90.8 ± 1.3	92.4 ± 0.9
DAELF	93.8	97.0	88.9

3.2 Visualization and Potential Improvement of Embedding Features

DAELF employs a *T-distributed Stochastic Neighbor Embedding* (TSNE) method to visualize the embedding features produced by netF. TSNE is a widely-used feature reduction and visualization method that transfers samples in a high-dimensional space to a low-dimensional space while retaining their relative distribution in the original space. Therefore, a cluster of samples on a 2D graph indicates a similar cluster of these samples in their original high dimensional space.

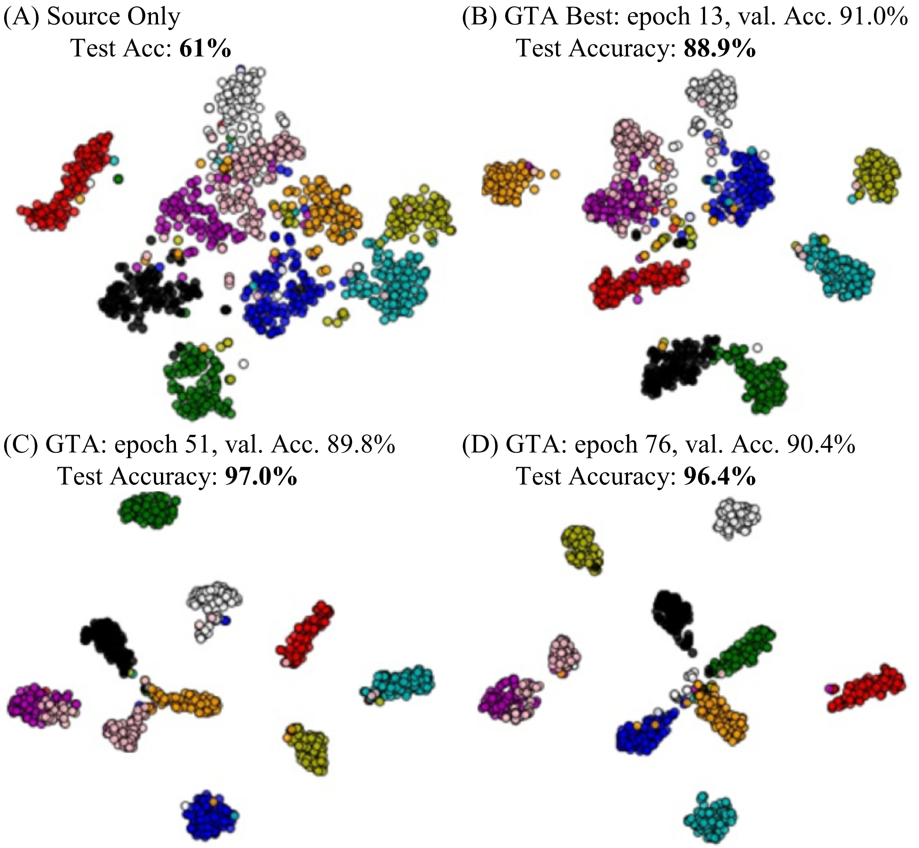


Fig. 6. 2D view of embedding features of a batch of target testing data. TSNE method was used to map the 128 features generated by **netF** that was trained by (A) source-only mode, (B) GTA mode at when maximal validation accuracy was reached, (C) GTA mode at epoch 51, (D) GTA mode at epoch 76.

By visually inspecting the distribution of target samples' embedding features (128 dimensions) that were mapped onto a 2D graph via the TSNE method, the results are promising. Figure 6 shows the 2D maps of embedding features for MNIST testing data generated by netF that were trained by SVHN as source training data (Fig. 6A, source only), and by SVHN as source and MNIST as target training data (Fig. 6B to Fig. 6D, GTA).

Comparing embedding features obtained through GTA and source-only training, GTA features could better separate testing images of 10 digits into distinct clusters, which led to a significantly improvement of classification accuracy for target testing data from 61% to 88%. Interestingly, DAELF didn't obtain the best performance from the GTA trained netF that was selected when the validation accuracy reached maximum at epoch 13. On the contrary, DAELF obtained significantly higher testing accuracies for netF selected after more training iterations, for example at epoch 51, epoch 76. At these times, the validation accuracy (on source data) was slightly decreased from 91.0% to 89.8% and 90.4%. However, the testing accuracy increased from 88.9% to 97.0% and 96.4%. Correspondingly, the clusters of the testing images of 10 digits are more clearly separated on the 2D graphs by the embedding features from netF selected later at epochs 51 and 76 (Fig. 6C and Fig. 6D).

The results demonstrate that the validation accuracy measured on source domain data may not be the ideal metric for selecting the optional model (netF) to classify target domain data. Since domain adaptation is driven by both source and target domain data during GTA mode training, a model's performance on source domain could be a trade-off to its performance on the target domain. Therefore, the selection of a model solely based on its best performance on the source domain data could be sub-optimal for the target domain data. An optimal model selection strategy should balance the performance on both domains.

A model's performance on target domain cannot be directly estimated without knowing target sample labels. In this case, a *surrogate metric* is needed to indirectly estimate a model's potential performance on the target domain. One of such surrogate metrics could be based on the clustering of target domain data in the embedding feature space as its correlation with target domain performance has been shown in Fig. 6. To achieve correlation without knowing the labels of target samples, it is possible to rely on the labeled source training samples to determine the clustered regions in the embedding feature space and quantify how well the target training samples may fall into those dense regions.

3.3 Data Fusion for Multiple Sensors

Using the four DIGIT datasets simulates two sensor modalities. The four datasets include two datasets (SVHN and USPS) that have been used in previous studies by Taigman, *et al.* [16], and the two new datasets, *MNIST-N (noise)* and MNIST-JP (*Japanese*). MNIST-N consists of images derived from MNIST by adding background *noise.* MNIST-JP consists of is a dataset similar to MNIST but the images of hand writing digits were written by *Japanese.* We used these two new datasets in order to increase learning difficulty so that the performance improvement of the fusion approaches could be observed.

We separated the four datasets into two pairs and applied the GTA algorithm to train two separate neural networks. The first neural network was trained by using SVHN as source domain data and MNIST-JP as target domain data (SVHN → MNIST-JP). The second neural network as trained by using MNIST-N as source domain data and USPS as target domain data (MNIST-N → USPS). The two networks were evaluated by testing data from the target domain, i.e., MNIST-JP and USPS, respectively.

We evaluated the performances of the feature-level and the decision-level fusion approaches and compared them with single GTA-trained networks. Table 3 lists the classification accuracy when each method was used to predict testing datasets, which were not used in any training processes.

The GTA-trained network can effectively improve the classification accuracy for target domain data. DAELF shows improvement here again in each single GTA trained network. The network trained by MNIST-N → USPS achieved 71.9% (Fig. 7) accuracy for USPS testing data and the network trained by SVHN → MNIST-JP achieved 74.37%

(Fig. 8) accuracy for MNIST-JP testing data. However, the two networks don't perform well for new domain data. The former network only achieved 56.89% accuracy for MNIST-JP and the latter network achieved 58.44% accuracy for USPS.

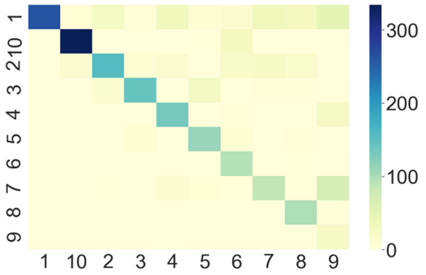


Fig. 7. MNIST-N → USPS

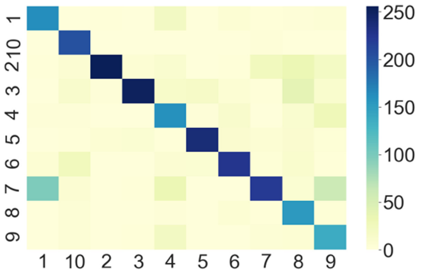


Fig. 8. SVHN → MNIST-JP

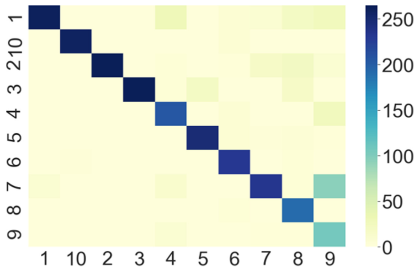


Fig. 9. Decision Level Data Fusion for Multiple Sensors

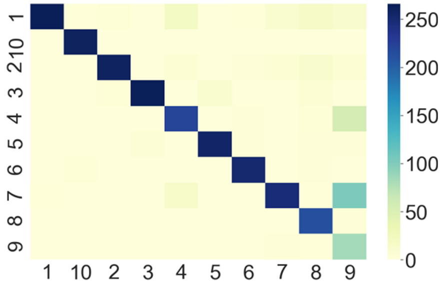


Fig. 10. Feature Level Data Fusion for Multiple Sensors

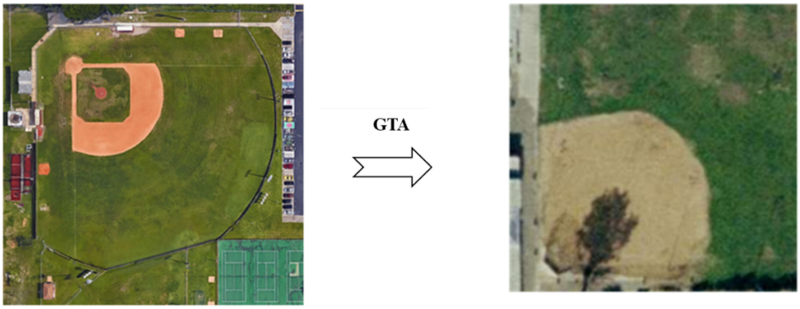
After incorporating the two networks together, either through feature-level or decision-level fusion, the new system outperformed any single network for every one of the two testing datasets. The two fusion methods achieved 84.28% (Fig. 9) and 86.07% (Fig. 10) accuracy, respectively. This more than 10% increase demonstrates the effectiveness of our proposed fusion approaches.

Table 3. Classification accuracies achieved by single GTA-trained and the fusion networks: Feature-Level Fusion (FLF), Decision-Level Fusion (DLF)

Testing Dataset	Single GTA-trained network		FLF	DLF
	MNIST-M→USPS	SVHN→MNIST-JP		
USPS	**71.90**	*58.44*		
MNIST-JP	*56.89*	**74.37**		
USPS + MNIST-JP			**86.07**	**84.28**

3.4 Classification of Aerial Image Dataset

Aerial imagery analysis provides a good showcase for advances in deep learning [21]. Using the DAELF model, it was modified to enable the classification of aerial images. We chose two datasets: *Aerial Image Dataset* (AID) and the *University of California, Merced* (UCM) dataset as source and target domain datasets, respectively. AID is a new large-scale aerial image dataset that collected images from the Google Earth imagery. The dataset contains 10000 600 × 600-pixel land images that are categorized in 30 scenes. The UCM is a similar land image dataset, which contains 2100 256 × 256-pixel images that are categorized in 21 scenes (100 images per scene). The images were manually extracted from large images from the USGS National Map Urban Area Imagery collection for various urban areas around the country. In order to test the DAELF model, we only used five classes of images from each dataset in the model development. These classes are: baseball field, medium residential area, sparse residential area, beach, and parking lot. We randomly chose 70% of images from AID and UCM to form source and target training datasets and used the remaining images as testing datasets. Figure 11 shows two example images from AID and UCM.



(A) Source domain: Images in AID dataset

(B) Target domain: Images in UCM dataset

Fig. 11. Domain adaptation between AID and UCM datasets.

The DAELF network's architecture for domain adaption was tailored between AID and UCM. In particular, the Resnet-50 network with pre-trained weights was used and the last layer removed as netF, and one linear layer as netC. Figure 12 illustrates the architectures of netF, netC, netG, and netD (replacing those of Fig. 3 with similar constructs of F, C, G, and D). Since the input image size for Resnet-50 is 224 × 224 pixels, both the AID and UCM images were re-sized to 224 × 224 before feeding them to the network.

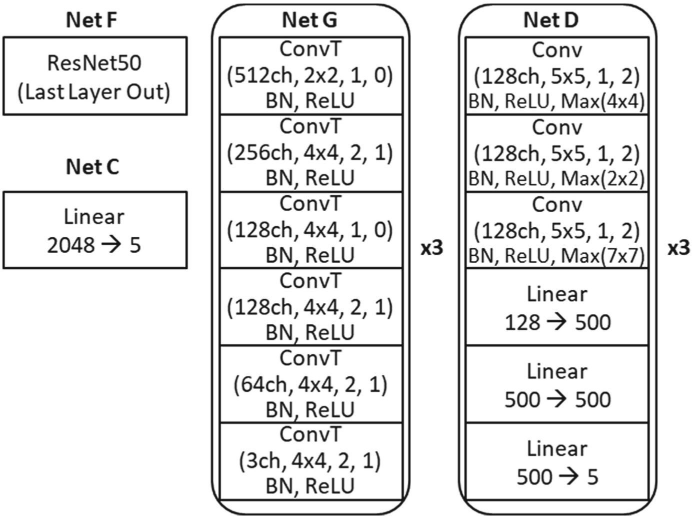

Net F
ResNet50 (Last Layer Out)
Net C
Linear 2048 → 5
Net G
ConvT (512ch, 2x2, 1, 0) BN, ReLU
ConvT (256ch, 4x4, 2, 1) BN, ReLU
ConvT (128ch, 4x4, 1, 0) BN, ReLU
ConvT (128ch, 4x4, 2, 1) BN, ReLU
ConvT (64ch, 4x4, 2, 1) BN, ReLU
ConvT (3ch, 4x4, 2, 1) BN, ReLU
x3
Net D
Conv (128ch, 5x5, 1, 2) BN, ReLU, Max(4x4)
Conv (128ch, 5x5, 1, 2) BN, ReLU, Max(2x2)
Conv (128ch, 5x5, 1, 2) BN, ReLU, Max(7x7)
Linear 128 → 500
Linear 500 → 500
Linear 500 → 5
x3

Fig. 12. Architectures for domain adaptation between AID and UCM dataset

DAELF was developed as a method for domain adaptation and data fusion. To achieve optimal performance, different combinations of parameters are explored in training the network. The parameters and the performance of 'source only' and GTA method are listed in Table 4. By choosing parameters properly, DAELF was able to obtain significant improvement for the GTA method when using the last trained model after 1000 epochs. Compared with the corresponding 'source only' method, the GTA accuracy can increase up to 12%. Figure 13 shows the TSNE method for the target domain testing images for the results in Table 4.

Table 4. Effect of parameters for the performance of the GTA method

		1	2	3	4	5
Parameters	Learning Rate	0.0004	0.0004	0.0004	0.0001	0.0004
	Learning Rate decay	0.0002	0.0002	0.0002	0.0010	0.0010
	Alpha	0.05	0.01	0.08	0.05	0.05
	Beta	0.05	0.01	0.08	0.05	0.05
Testing Accuracy	Source Only	69.7	69.7	69.7	69.7	69.7
	Best GTA Method	66.4	**56.2**	66.7	54.7	65.1
	Last GTA Method	**78.7**	48.5	**75.6**	**60.3**	**65.7**

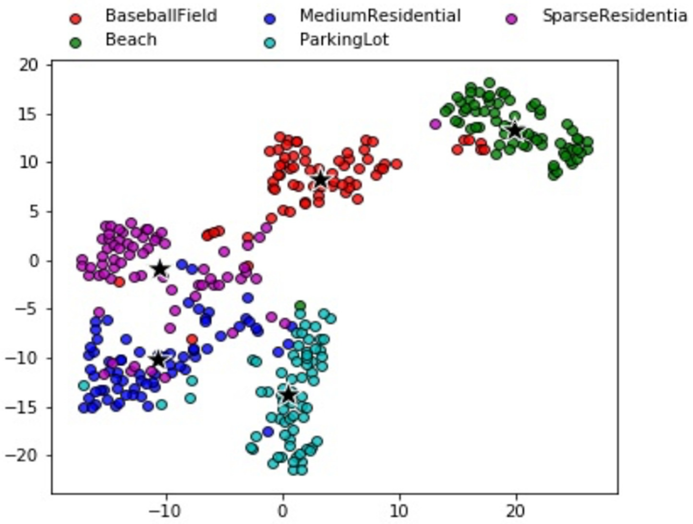


Fig. 13. 2D view of embedding features extracted by the TSNE method for the target domain testing images for Table 4 with accuracy of 78.7%.

4 Discussion and Conclusions

The paper introduces a deep neural network-based method DAELF, which adapts to changes of the input distribution allowing for self-correcting of multiple source classification and fusion. The DAELF results showed that optimum performance can be achieved, which reaches or even exceeds state-of-art approaches in common datasets. The performance of the DAELF depends on various hyper-parameters, each of which must be tuned to achieve optimum. The optimization is a sensitive process, requiring great attention and significant computational efforts. Hence, future results seek to better interpret the selection of the hyper-parameters for different scenarios.

It is known that the training process of GAN models may exhibit oscillations and instabilities, which is called *generator collapse* [18]. There are various methods to address these issues. Two such methods which have been used in our studies as ongoing work:

- ***Unrolled GAN***: The original GAN framework is a minimax optimization problem, which is practically unfeasible to solve for optimal parameters of discriminator and generator. Instead, it is solved by iteratively using gradient descent on G and gradient ascent on D. *Unrolled GANs* [22] are simultaneous recurrent networks (SRN), which extend the time horizon of the iterative solution, when the theory of ordered derivatives in *backpropagation through time* (BPTT) is directly applicable [23, 24]. SRNs provide are a natural way to improve the performance of GANs by considering the unfolding iterations over a given time horizon, e.g., 10–20 iterations. The stability and convergence are improved using unrolled GANs.

- ***Wasserstein GAN***: WGAN is an alternative to traditional GAN training, by replacing the original Kullback-Leibler (KL)-based distance measure by a new, mathematically justified function [25, 26]. Results demonstrated that Wasserstein loss stabilizes the performance of the DAELF system. The method has been extended to Wasserstein GAN as well. Performance stabilization is extremely important when using GANs for domain transfer applications, as when the data changes, sometimes in an unpredictable way, stability issues can arise.

In conclusion, this paper develops a deep learning-based multi-source self-correcting approach to fuse data with different modalities at the data-level to maximize their capabilities to detect unanticipated events/targets. The Domain Adaptation for Efficient Learning Fusion (DAELF) deep neural network approach adapts to changes of the input distribution allowing self-correcting across multiple source classifications. When supported by a distributed computing hierarchy, DAELF scales in data size, geographical span, and sensor modalities. From the aerial data sets analysis, feature-level fusion (FLF) outperforms decision-level fusion (DLF) approaches in terms of classification accuracy.

Acknowledgements. This material is based on research sponsored by Air Force under contract FA864920P0350. 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. Zheng, Y., Blasch, E., Liu, Z.: Multispectral Image Fusion and Colorization, SPIE Press, Bellingham (2018)
2. Zhang, R., Bin, J., Liu, Z., et al.: WGGAN: a wavelet-guided generative adversarial network for thermal image translation. In: Naved, M. (ed.), Generative Adversarial Networks for Image-to-Image Translation. Elsevier (2020)
3. Shen, D., et al.: A joint manifold leaning-based framework for heterogeneous upstream data fusion. J. Algorithms Comput. Technol. (JACT) **12**(4), 311–332 (2018)
4. Vakil, A., Liu, J., Zulch, P., et al.: A survey of multimodal sensor fusion for passive RF and EO information integration. In: IEEE Aerospace and Electronics System Magazine (2020)
5. Bunyak, F., Palaniappan, K., Nath, S.K., Seetharaman, G.: Flux tensor constrained geodesic active contours with sensor fusion for persistent object tracking. J. Multimedia **2**(4), 20 (2007)
6. Jia, B., Pham, K.D., et al.: Cooperative space object tracking using space-based optical sensors via consensus-based filters. IEEE Tr. Aerosp. Electron. Syst. **52**(3), 1908–1936 (2016)
7. Shen, D., Sheaff, C., Guo, M., et al.: Enhanced GANs for satellite behavior discovery. In: Proc SPIE, p. 11422 (2020)
8. Nicolaou, M.A., Gunes, H., Pantic, M.: Audio-visual classification and fusion of spontaneous affective data in likelihood space. In: ICPR (2010)
9. Muller, H., Kalpathy–Cramer, J.: The image CLEF medical retrieval task at ICPR 2010 — information fusion to combine visual and textual information. In: ICPR (2010)
10. Li, H., Wu, X.J., Kittler, J.: Infrared and visible image fusion using a deep learning framework. In: ICPR (2018)
11. Blasch, E., Seetharaman, G., Palaniappan, K., Ling, H., Chen, G.: Wide-area motion imagery (WAMI) exploitation tools for enhanced situation awareness. In: IEEE Applied Imagery Pattern Recognition Workshop (2012)

12. Palaniappan, K., et al.: Moving object detection for vehicle tracking in wide area motion imagery using 4D filtering. In: International Conference on Pattern Recognition (ICPR) (2016)
13. Kozma, R.: A cognitively motivated algorithm for rapid response in emergency situations. In: IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA) (2017)
14. Kozma, R.: Intentional systems: review of neurodynamics, modeling, and robotics implementation. Phys. Life Rev. **5**(1), 1–21 (2008)
15. Majumder, U., Blasch, E., Garren, D.: Deep Learning for Radar and Communications Automatic Target Recognition. Artech House, Norwood (2020)
16. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. arXiv preprint arXiv:1611.02200 (2016)
17. Goodfellow, I.J.: NIPS 2016 tutorial: Generative adversarial networks. arXiv:1701.00160v4 (2016)
18. Goodfellow, I.J., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NIPS), p. 27 (2014)
19. Tzeng, E., Devin, C., Hoffman, J., Finn, C., Abbeel, P.: Adapting deep visuomotor representations with weak pairwise constraints. arXiv, https://arxiv.org/abs/1511.07111 (2015)
20. Sankaranarayanan, S., Balaji, Y., Castillo, C.D.: Generate to adapt: aligning domains using generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 8503–8512 (2018)
21. Savakis, A., Nagananda, N., Kerekes, J.P., et al.: Change detection in satellite imagery with region proposal networks. Defense Syst. Inform. Anal. Center (DSIAC) J. **6**(4), 23–28 Fall (2019)
22. Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J.: Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163 (2016)
23. Werbos, P.J.: Backpropagation through time: what it does and how to do it. Proc. IEEE **78**(10), 1550–1560 (1990)
24. Ilin, R., Kozma, R., Werbos, P.J.: Beyond feedforward models trained by backpropagation: a practical training tool for a more efficient universal approximator. IEEE Trans. Neural Netw. **19**(6), 929–937 (2008)
25. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. arXiv preprint arXiv:1701.07875 (2017)
26. Gulrajani, F., Ahmed, M., Arjovsky, V., Dumoulin Courville, A.C.: Improved training of Wasserstein GANs. In: Advances in Neural Information Processing Systems, pp. 5767–5777 (2017). https://arxiv.org/pdf/1704.00028.pdf.