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

Jingyang Lu, Chenggang Yu, Erik Blasch, Roman Ilin, Hua-mei Chen, Dan Shen, Nichole Sullivan, Genshe Chen, Robert Kozma

Intelligent Fusion Technology, Inc.
Air Force Research Laboratory
The University of Memphis
01/15/2021

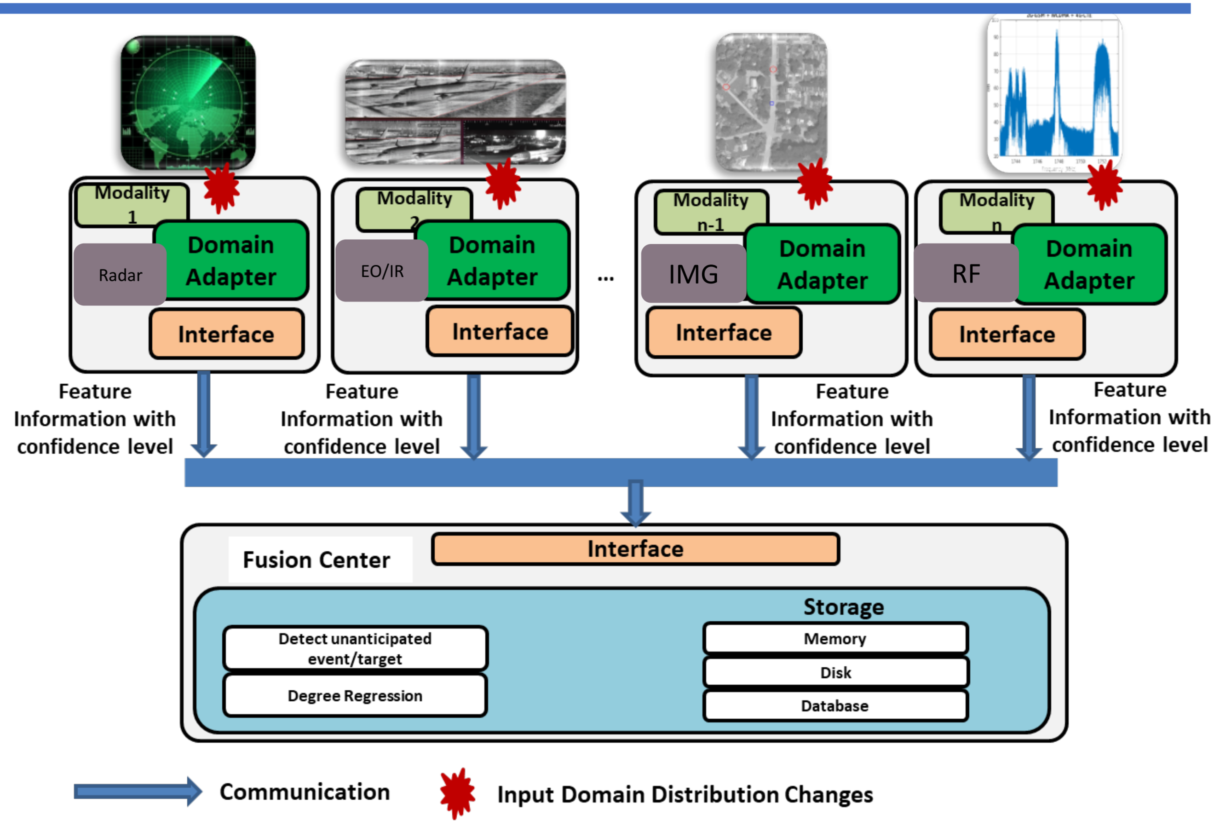
Outline

- Introduction and Motivation
- Problem Formulation
- Benchmark Dataset Test
- Data Fusion Approaches
- Conclusion and Future Work

Machine Learning based Domain Adaptation for Multiple Source Classification and Fusion

Motivation

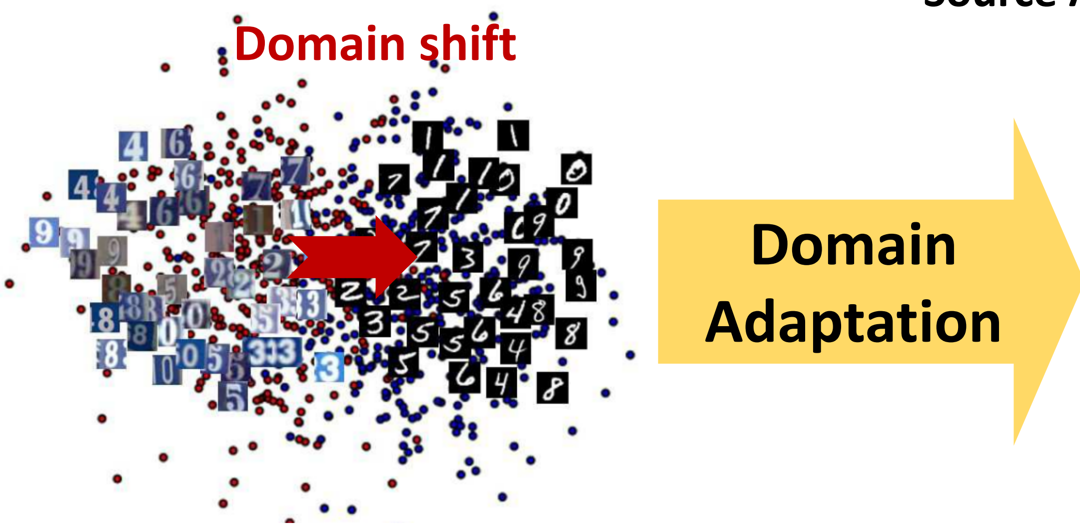
- Classifier accuracy decreases due to the domain shift
- Higher false alarm rates and consequently decreases trust in the classifier system
- Quick adaptation to changes in domain distributions without retraining the classifiers



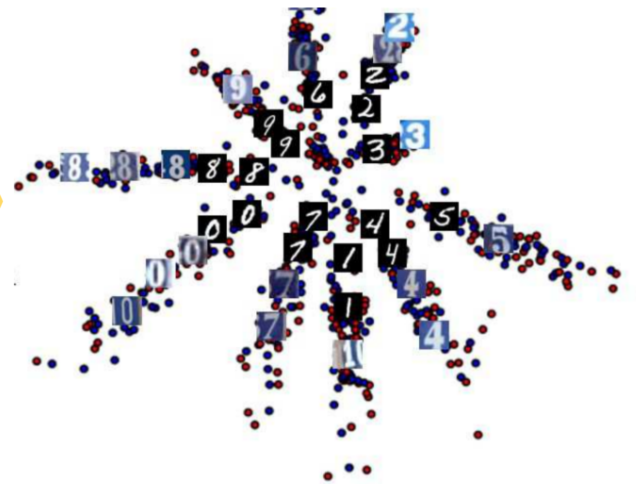
The benefits of the proposed solution, Machine Learning based Domain Adaptation (MLB-DA) :

- **Focused on learning features that combine: (i) discriminativeness and (ii) domain invariance.**
- **Does not need to retrain the model to adapt to input distribution change.**
- **Provides a sound foundation for the more realistic *Open Set Domain Adaptation* scenario.**

Problem Formulation



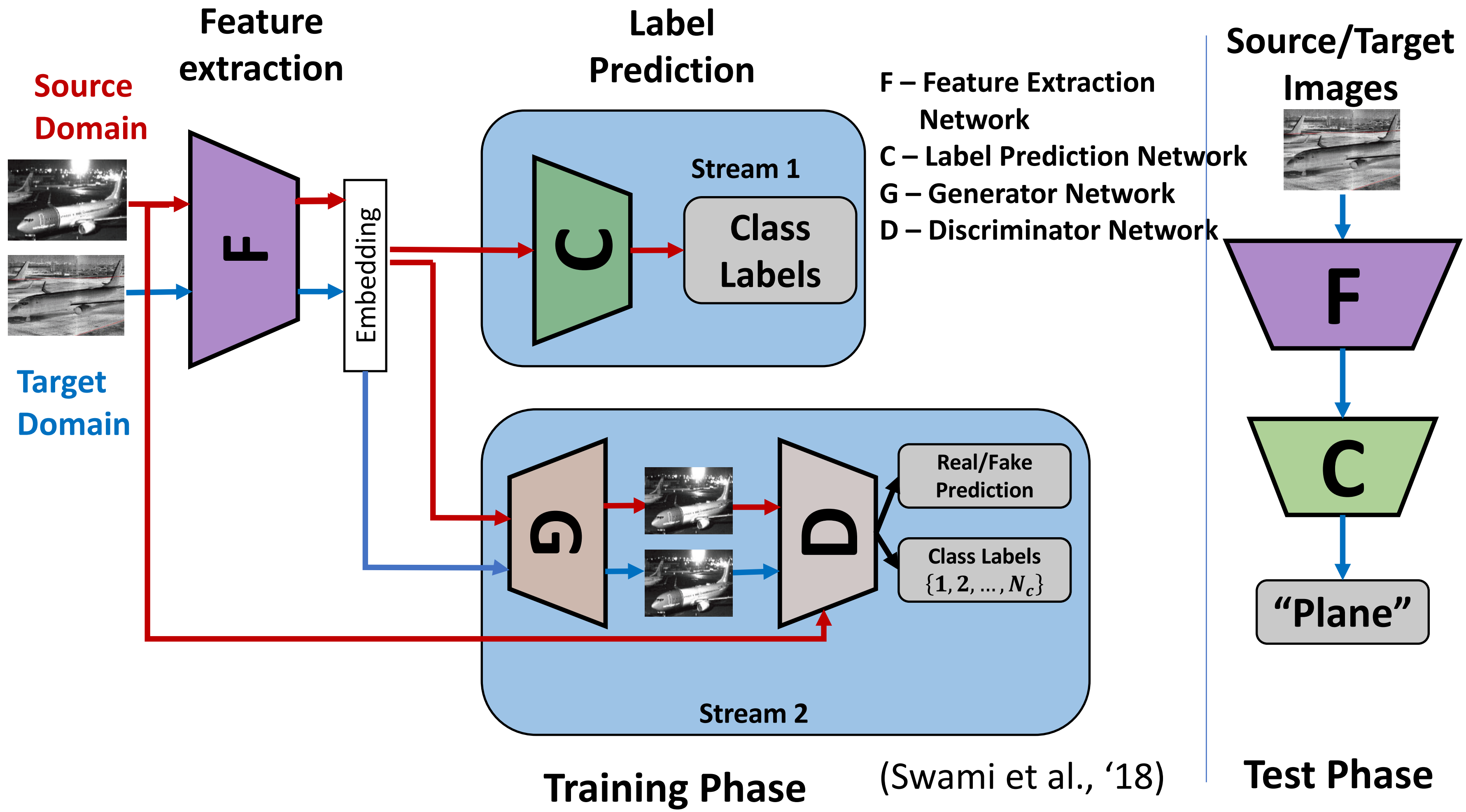
Samples Of The Same Class From Both Source And Target Domains Are Close To Each Other



Source Domain Target Domain

- Domain Adaptation Attempts To Mitigate The Discrepancy Between Source And Target Domain.
- After Adaptation, The Source And Target Domains Are Expected To Share The Same Or Similar Distribution.

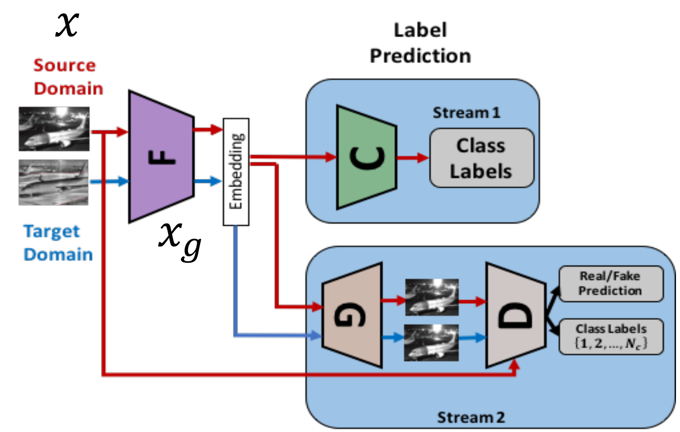
Domain Adaptation for Each Modality



The proposed MLB-DA is designed by employing a variant of the conditional GAN called Auxiliary Classifier GAN where the discriminator is modeled as a multi-class classifier instead of providing conditioning information at the input

Domain Adaptation for Each Modality

1. Given a real data x as input to F, the input to the generator network G is $x_g = [F(x), z, l]$,
where z is random noise vector $z \in \mathbb{R}^d$ sampled from $N(0,1)$;
l is 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.
2. A classifier network C that takes as input the embedding generated by F and predicts a multiclass distribution $C(x)$



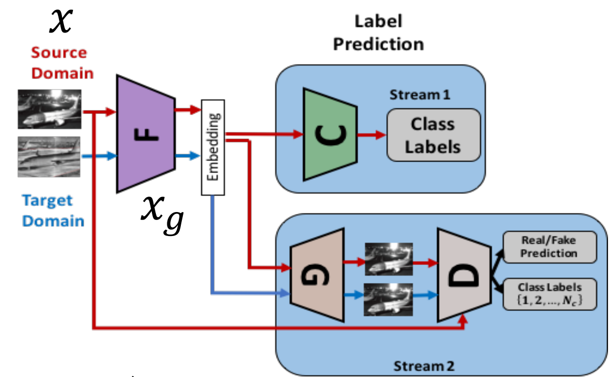
3. The discriminator mapping D takes the real input data x or the generated input $G(x_g)$ as input and outputs two distributions:

(1) $D_{data}(x)$: the probability of the input being real, which is modeled as a binary classifier

(2) $D_{cls}(x)$: the class probability distribution of the input x, which is modeled as a N_c-way classifier.

It should be noted that, for target data, as the class labels are unknown, $D_{data}(x)$ is only used to update the gradients

Cost Function Domain Adaptation for Each Modality



1. In the case of source inputs, the gradients are generated using the following loss functions,

$$L_{data,src} + L_{cls,src} = \boldsymbol{E}_{\boldsymbol{x}\sim\boldsymbol{S}} \max_{D} log D_{data}(x)) + \log\left(1 - D_{data}\left(G\left(x_g\right)\right)\right) + \log\left(D_{cls}(x)_y\right)$$

1st D Update

The third entity in the cost function is utilized as the label data information is available in the source domain dataset.

2. Based on the loss function for D, Generator (G) is updated based on the combination of adversarial loss and classification loss.

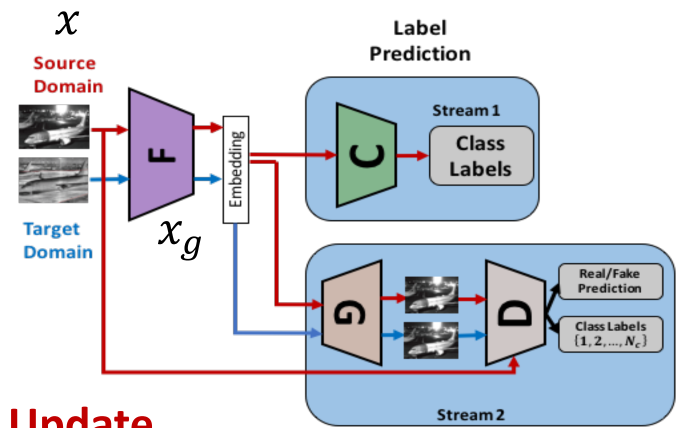
$$L_G = \min_{G} E_{x\sim S} \; -\log\left(D_{cls}\left(G\left(x_g\right)\right)_y\right) + \log\left(1 - D_{data}\left(G\left(x_g\right)\right)\right)$$

Source domain data is used to update the G

In our proposed frame work, target domain data is also used to update the G

Cost Function Domain Adaptation for Each Modality

3. F, C Update



$$L_c = \min_C \min_F E_{x\sim\mathcal{S}} - \log\left(C\left(F(x)\right)_y\right),$$

$$L_{cls,src} = \min_F E_{x\sim\mathcal{S}} - \alpha \log\left(D_{cls}\left(G(x_g)\right)_y\right)$$

1st F Update

F is also updated using the adversarial gradients which is similar to the loss function for G

4. D is updated to determine the generated target domain as fake as follows,

$$L_{adv,tgt} = \max_D E_{x\sim\mathcal{T}} \log(1 - D_{data}(G(x_g)))$$

2nd D Update

In order to transfer the knowledge of target distribution to the embedding, F is updated using the gradients from D_{data} that corresponds to the generated target data being classified as real,

$$L_{F_{adv}} = \min_F E_{x\sim\mathcal{T}} \beta \log(1 - D_{data}(G(x_g)))$$

2nd F Update

Training Process for Domain Adaptation

Algorithm Iterative Training Procedure Of MLB-DA

1: Training Iterations = N

2: For t in 1: N do

3: Sample k raw data with labels from source domain $S: \{s_i, y_i\}_i^k$
Let $f_i = F(s_i)$ be the embeddings computed for the source images
Sample k images from target domain $\mathcal{T}: \{t_i\}_i^k$
Let $h_i = F(t_i)$ be the embeddings computed for the target images
Sample k random noise samples $\{z_i\}_{i=1}^k \sim \mathcal{N}(0,1)$.
Let f_{g_i} and h_{g_i} be the concatenated inputs to the generator.

4: Update discriminator (D) using the following objectives:

$$\boldsymbol{L_D = L_{data,src} + L_{cls,src} + L_{adv,tgt}}$$

Target domain data

5: Update the generator (G), only for source data, through the discriminator (D) gradients computed using

$$\boldsymbol{L_G = min_G \frac{1}{k} - log\left(D_{cls}\left(G\left(f_{g_i}\right)\right)_{y_i}\right) + log\left(1 - D_{data}\left(G\left(f_{g_i}\right)\right)\right) \boxed{+ log\left(1 - D_{data}\left(G\left(h_{g_i}\right)\right)\right)}}$$

6: Update the embedding F using a linear combination of the adversarial loss and classification loss. Update the classifier C for the source data using a cross entropy loss function.

$$\boldsymbol{L_F = L_c + \alpha L_{cls,src} + \beta L_{F_{adv}}}$$

- $L_C = \min_C \min_F \frac{1}{k} \sum_i^k - \log\left(C(f_i)_{y_i}\right)$
- $L_{cls,src} = \min_F \frac{1}{k} \sum_i^k - \log\left(D_{cls}\left(G\left(f_{g_i}\right)\right)_{y_i}\right)$
- $L_{F_{adv}} = \min_F \frac{1}{k} \sum_i^k \log\left(1 - D_{data}\left(G\left(h_{g_i}\right)\right)\right)$

Benchmark Dataset Test

1. GTA performance evaluation based on digits dataset.
2. Study the new dataset UCM and AID including the Baseball field, beach, medium residential, sparse residential, and parking lot. *
3. Improve the GTA approach: the feature extraction model F is replaced by the ResNet-50 in order to extract efficient feature from the input data.
4. Implement GTA Domain Adaptation From AID to UCM, the numerical results show GTA approaches can efficiently classify the data from target domain.
5. Conduct the GTA approach sensitivity analysis.

Benchmark Dataset- Digits Dataset

Results shown in the original paper

Method	MN → US (p)	MN → US (f)	US → MN	SV → MN
Source only	75.2 ± 1.6	79.1 ± 0.9	57.1 ± 1.7	60.3 ± 1.5
RevGrad [4]	77.1 ± 1.8	-	73.0 ± 2.0	73.9
DRCN [5]	91.8 ± 0.09	-	73.7 ± 0.04	82.0 ± 0.16
CoGAN [15]	91.2 ± 0.8	-	89.1 ± 0.8	-
ADDA [32]	89.4 ± 0.2	-	90.1 ± 0.8	76.0 ± 1.8
PixelDA [1]	-	**95.9**	-	-
Ours	**92.8 ± 0.9**	95.3 ± 0.7	**90.8** ± 1.3	**92.4** ± 0.9

Implemented by IFT
62% (Source Only)
88.90% (GTA)

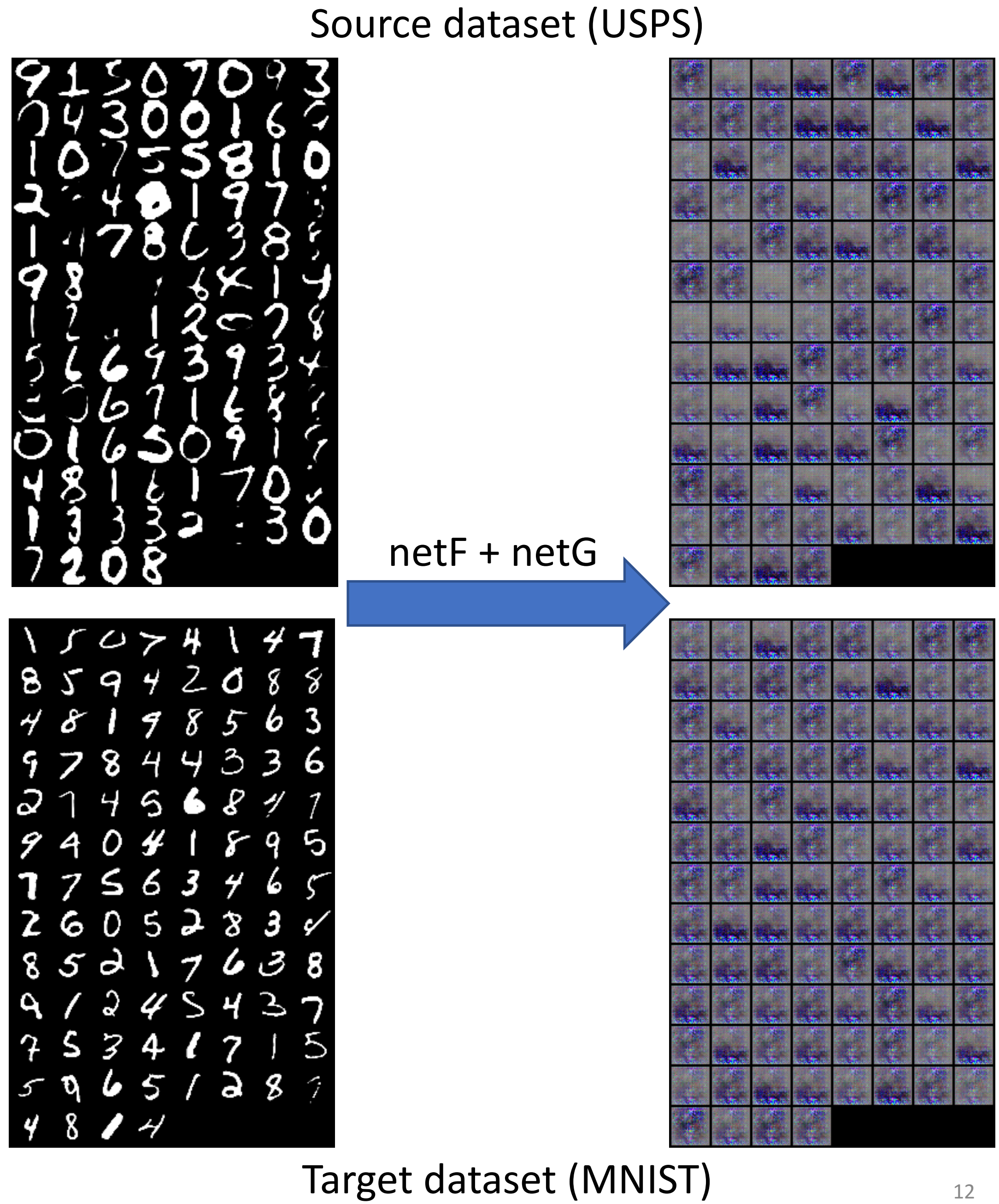
Implemented by IFT
89.9% accuracy (Source Only)
93.8 % (GTA)

Implemented by IFT
74.5 % (Source Only)
97.0% (GTA)

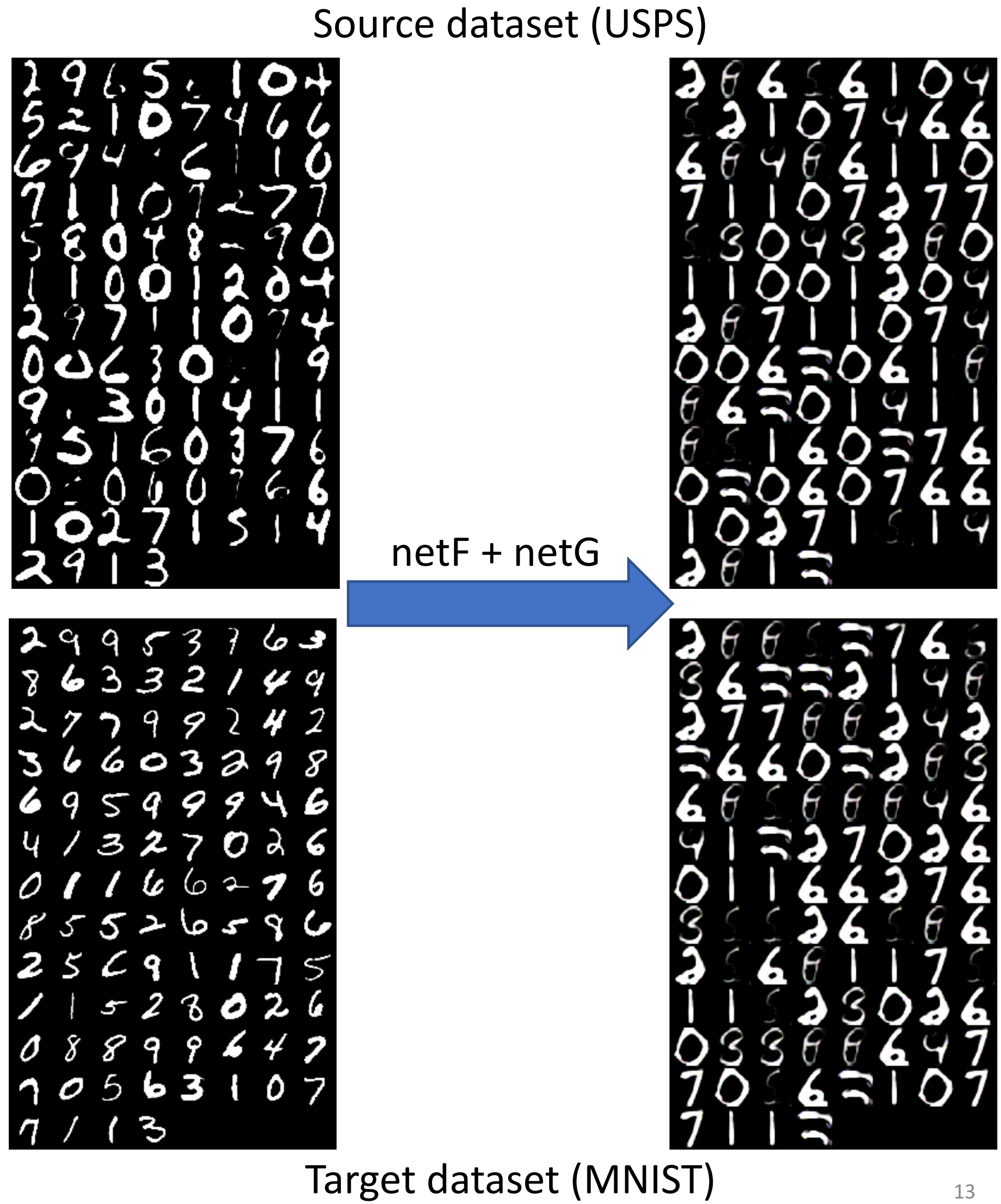
Each model was trained 500 epochs

Validation: In the experiment set up, for example, SVHN->MN, the target domain data is for the MNIST, SVHN is the source domain data, and after each epoch of training, a fixed subset of data from source domain is used to validation, which is different from the test.

100 samples from the two datasets are transferred by **netF + netG** after **1** round of training

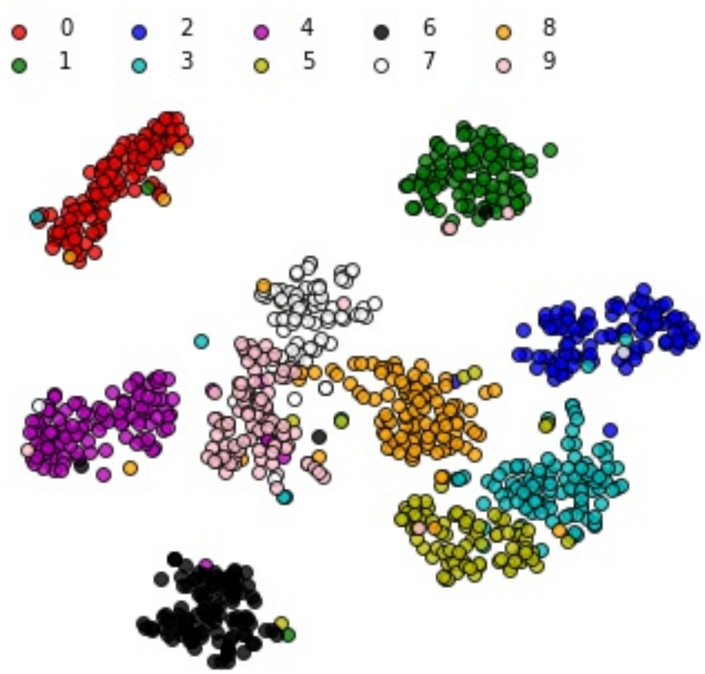


100 samples from the two datasets are transferred by **netF + netG** after **190** rounds of training

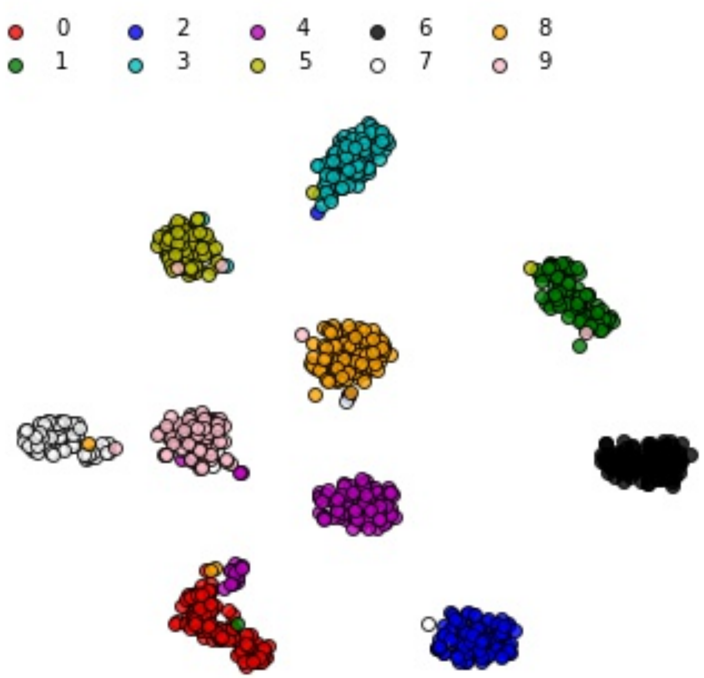


TSNE visualization of target data

Source only



GTA



TSNE visualization of target data (MNIST) separation by features out of netF that is trained by source data (USPS) only and by GTA (Each point represent one sample randomly selected from the MNIST testing set. Same 1000 random samples are used in the two plots)

Benchmark Dataset- Aerial Datasets

1. UCM

- Manually extracted images from United States Geological Survey National Map Urban Area Imagery
- 21 classes
- Image size is 256x256 pixels
- Ground Sample Distance (GSD) 1 foot/pixel
- 100 images per class ([UCM] Yi Yang et. al., "Bag-Of-Visual-Words and Spatial Extensions for Land-Use Classification," ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), 2010)

2. AID

- More than 10,000 aerial images
- 30 classes
- Multi source Google Earth images from various countries
- Image size is 600x600 pixels
- Multi GSD (8 meter to 0.5 meter)

GTA Domain Adaptation From AID to UCM

Source dataset: AID
Training set size: 1129
Testing set size: 489

Target dataset : UCM
Training set size: 350
Testing set size: 150

Task: Classifying images into five categories: Baseball field, beach, medium residential, sparse residential, and parking lot

GTA Networks Architectures

NetF

ResNet 50
(Last layer out)

NetC

Linear
2048 → 5

NetG

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 X3
ConvT (64ch, 4x4, 2, 1) BN, ReLU
ConvT (3ch, 4x4, 2, 1) Tanh

NetD

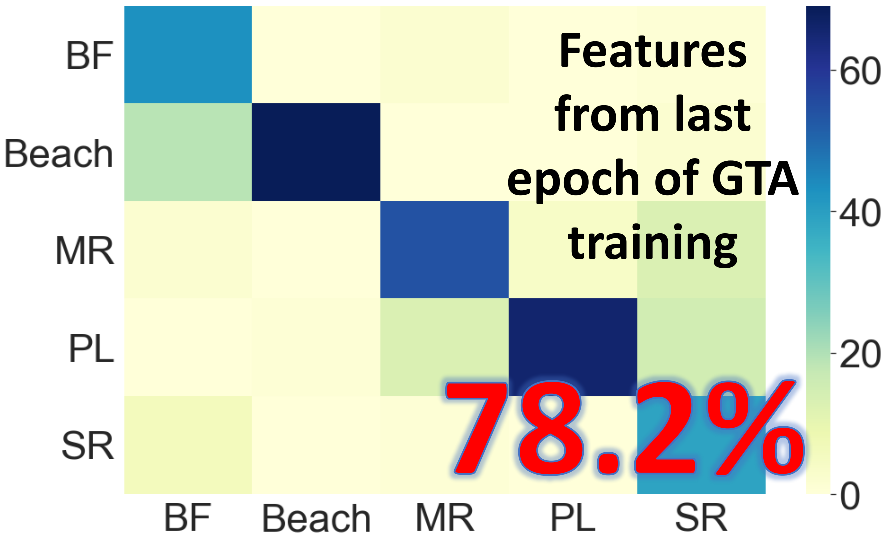
Conv (128ch, 5x5, 1,2) BN, ReLU, Max(4x4)
Conv (128ch, 5x5, 1,2) BN, ReLU, Max(2x2) X3
Conv (128ch, 5x5, 1,2) BN, ReLU, Max(7x7)
Linear 128 → 500
Linear 500 → 500
Linear 500 → 5

Benchmark Dataset- Aerial Datasets

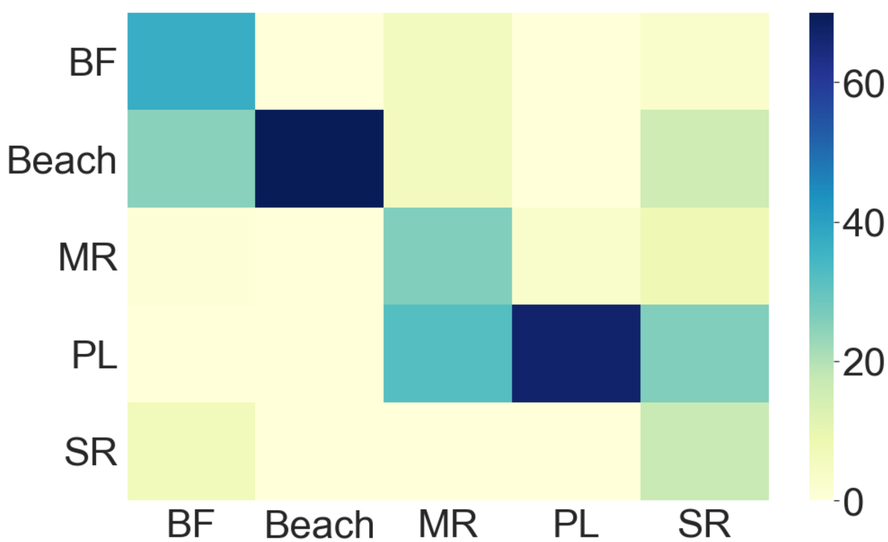
		Case 1	Case 2	Case 3	Case 4	Case 5
Parameters	Learning rate	0.0004	0.0004	0.0004	0.0001	0.0004
	Learning rate decay	0.0002	0.0002	0.0002	0.001	0.001
	Alpha	0.05	0.01	0.08	0.05	0.05
	Beta	0.05	0.01	0.08	0.05	0.05
Target accuracy	Source only	69.7	69.7	69.7	69.7	69.7
	Best GTA model	66.4	56.2	66.6	54.7	65.1
	Last GTA model	**78.2**	48.5	**75.6**	60.3	65.7

~12.0% Improvement

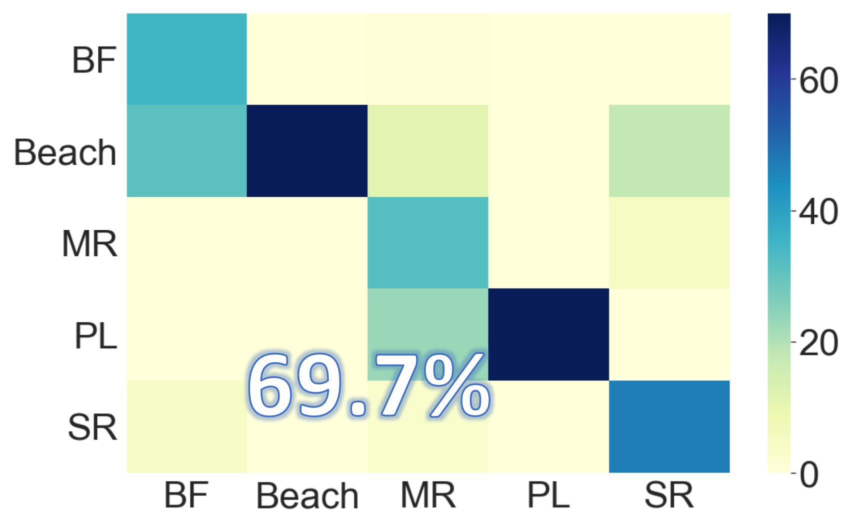
Experiment 1



Learning rate	**0.0004**
Learning rate decay	0.0002
Alpha	0.05
Beta	0.05



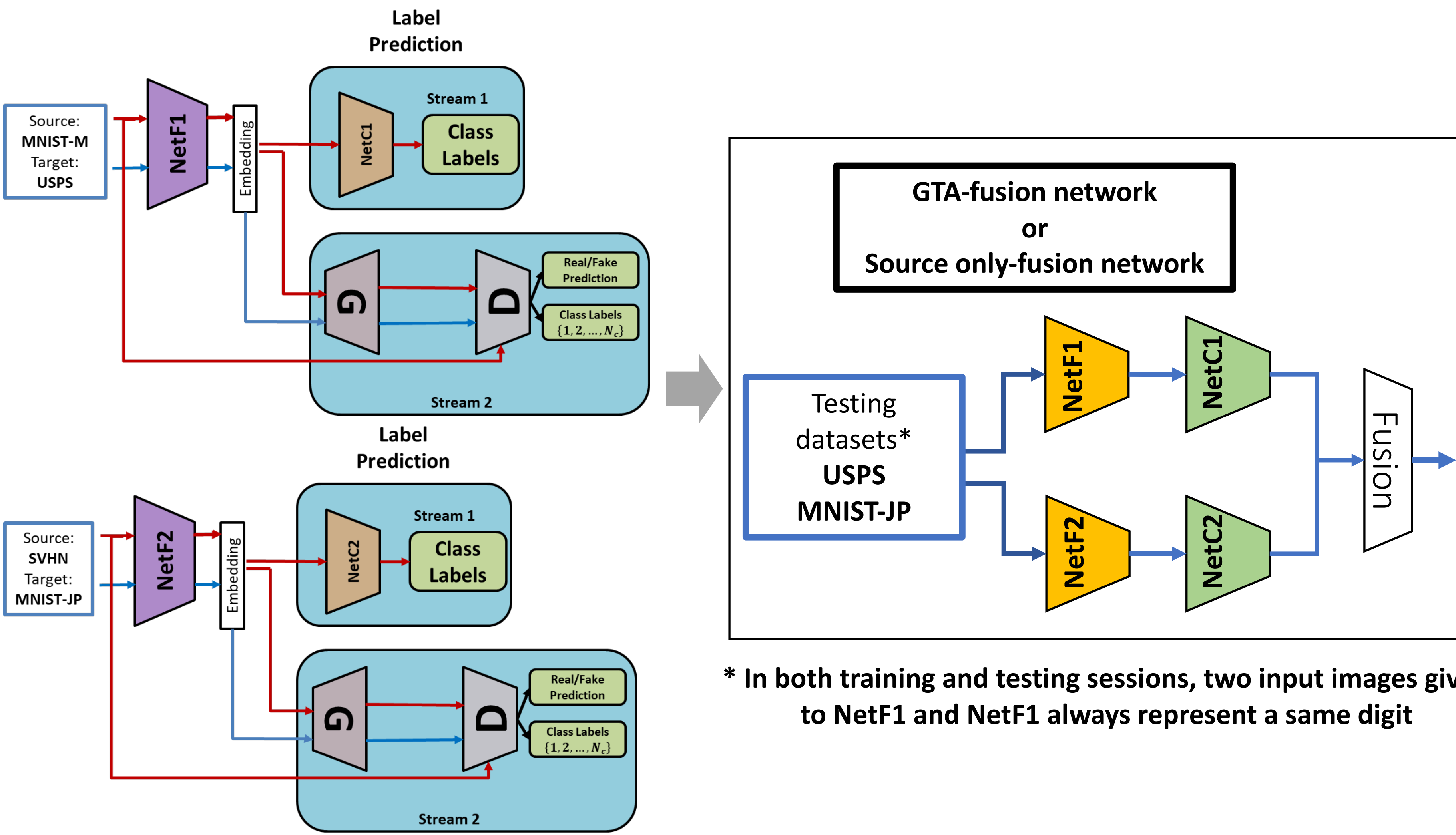
Features from best GTA model



Features from source only training

Development of Data Fusion Approaches from Different Sensors and Different Modalities

Decision Level Fusion for Heterogeneous Multiple Sensor Modalities



* In both training and testing sessions, two input images given to NetF1 and NetF1 always represent a same digit

Decision Level Fusion for Heterogeneous Multiple Sensor Modalities

- Entropy $\mathcal{H}_k$ for each sensor k

$$\mathcal{H}_k = -\sum_{i=1}^{N} p_{k_i} \log(p_{k_i}), i = 1, \dots, N$$

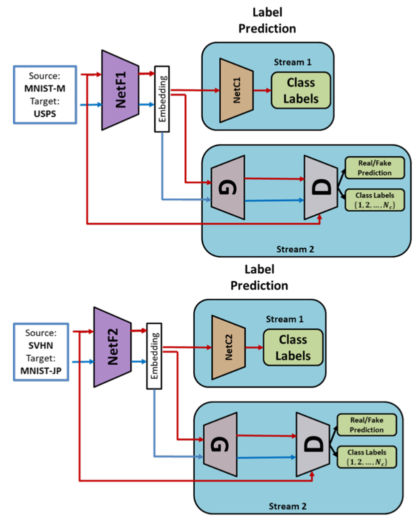
- Decision by each sensor k

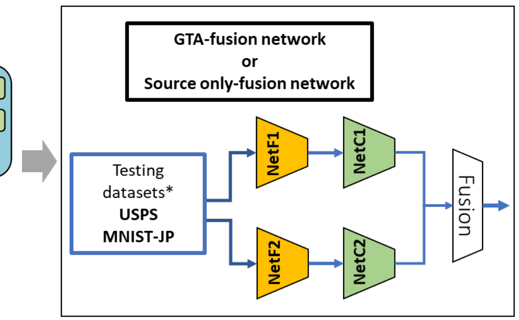
$$d_k = argmax(p_{k_i}), i = 1, \dots, N$$

- Final decision by Fusion Center

$$D = d_{opt}, H_{opt} \leq H_k \; k = 1, \dots, K$$

Where K is the total number of senor modalities.





* In both training and testing sessions, two input images given to NetF1 and NetF1 always represent a same digit

For a system with two sensor modalities:

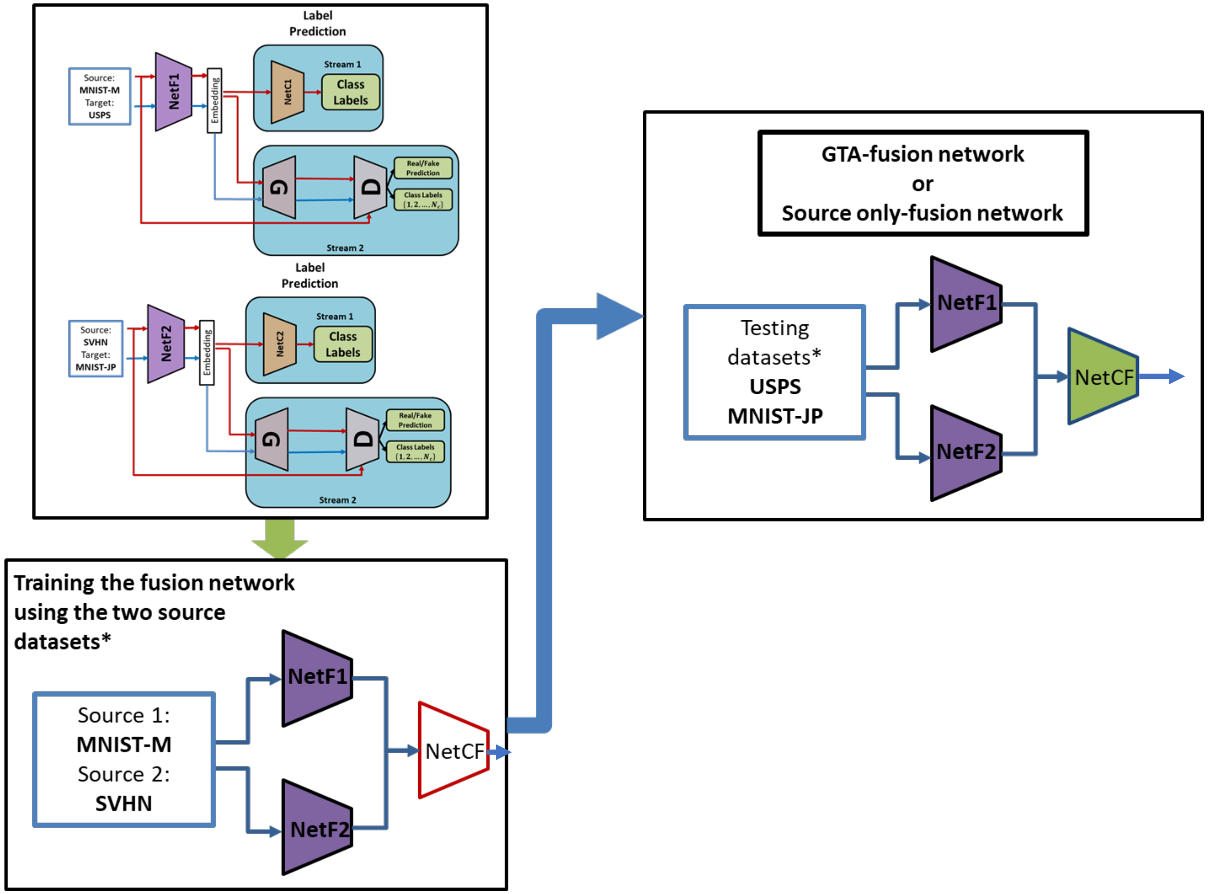
$$d = argmax(p_i), i = 0, \dots, 9$$

$$H = -\sum_{i=1}^{N} p_i \log(p_i), i = 0, \dots, 9$$

$$D = \begin{cases} d_1, if \; H_1 < H_2 \\ d_2, if \; H_1 > H2 \end{cases}$$

In order to make a final prediction D from the predictions of the two decision networks, we assessed each prediction's reliability by computing an entropy, where p0 through p9 are 10 output values from one netC.

Feature Level Fusion for Heterogeneous Multiple Sensor Modalities



*** In both training and testing sessions, two input images given to NetF1 and NetF1 always represent a same object.**

Architectures for Fusion Networks

NetF

ConvT (64ch, 5x5, 1,0) BN, ReLU, Max(2x2)
ConvT (64ch, 5x5, 1,0) BN, ReLU
Conv (128ch, 4x4, 1,0) BN, ReLU

NetC

Linear 128 → 128 ReLU
Linear 128 → 10 SoftMax*

NetCF**

Linear 256 → 256
Linear 256 → 128 RelU
Linear 128 → 10

NetG

ConvT (512ch, 4x4, 2, 2) BN, ReLU
ConvT (256ch, 4x4, 2, 2) BN, ReLU
ConvT (128ch, 4x4, 2, 2) BN, ReLU
ConvT (64ch, 4x4, 2, 2) BN, ReLU
ConvT (1ch, 4x4, 2, 2) BN, ReLU

NetD

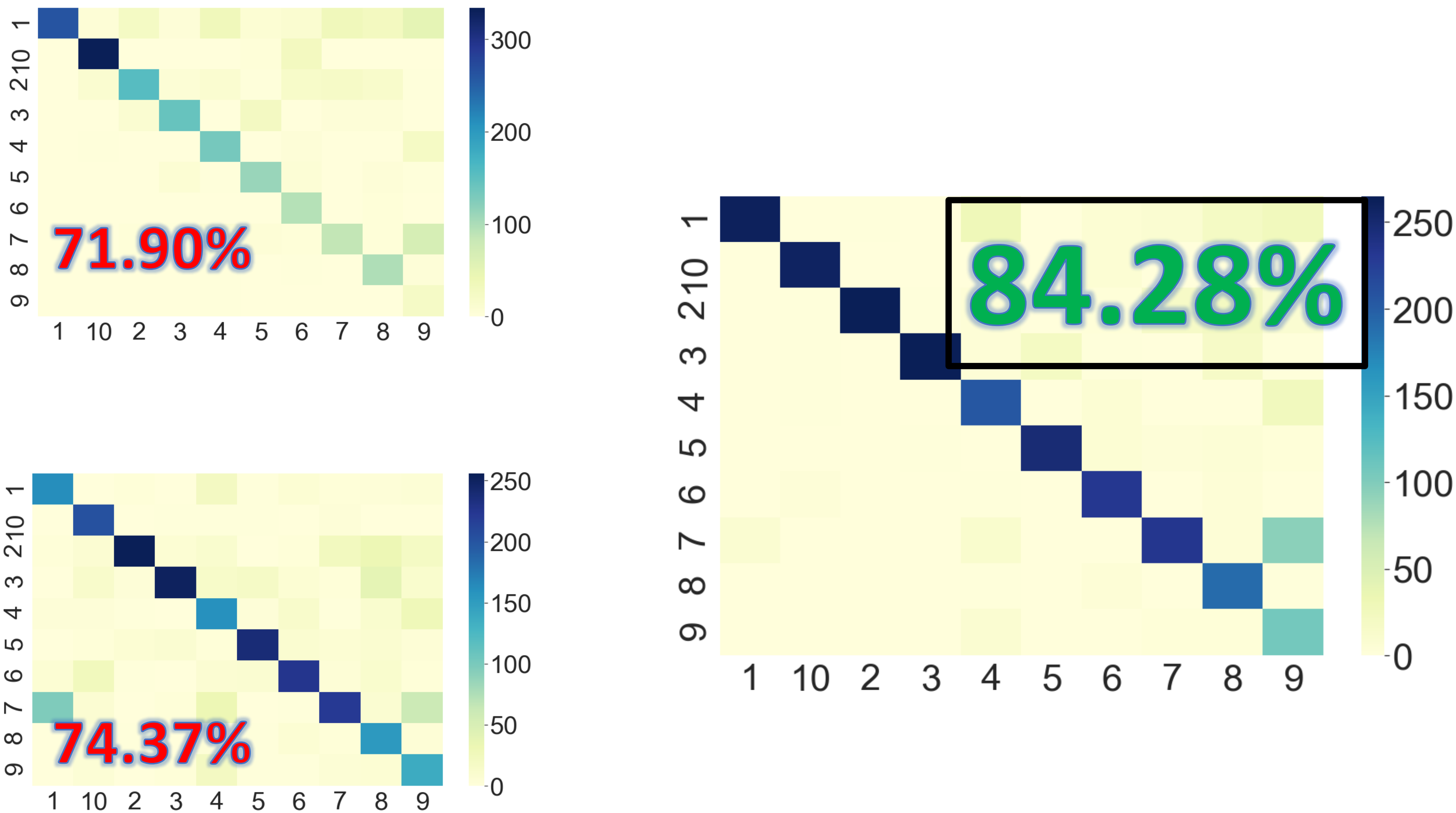
Conv (64ch, 3x3) BN, LeakyRelU(0.2), Max(2x2)
Conv (128ch, 3x3) BN, LeakyRelU(0.2), Max(2x2)
Conv (256ch, 3x3) BN, LeakyRelU(0.2), Max(2x2)
Conv (128ch, 3x3) BN, LeakyRelU(0.2), Max(4x4)
Linear 128 → 10 128 →2

* SoftMax is applied only when conducting decision-level fusion
** The network that fuses two NetFs

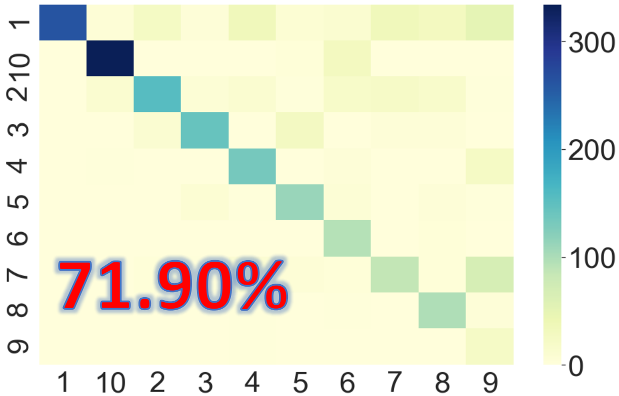
Data Fusion Approaches Performance for Multiple Sensor Modalities

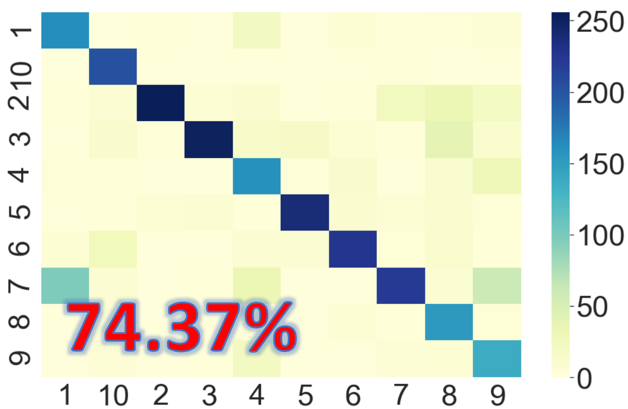
Testing dataset	Single GTA-trained network		Feature-level fusion	Decision-level fusion
	MNIST-M →USPS	**SVNH → MNIST-JP**		
USPS	**71.90**	58.44		
MNIST-JP	56.89	**74.37**		
USPS+MNIST-JP			**86.07**	**84.28**

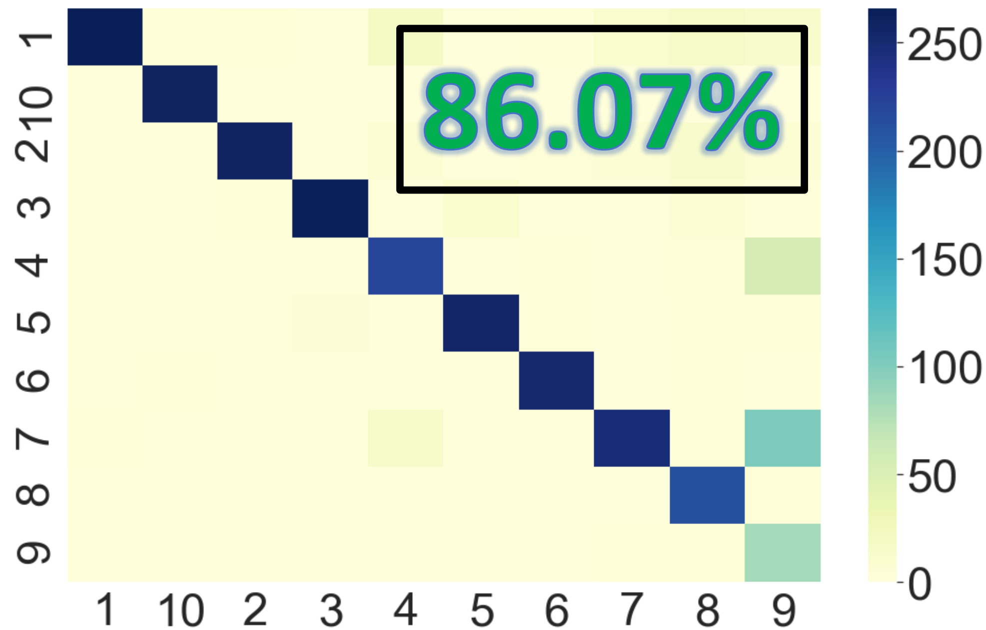
Decision Level Fusion for Heterogeneous Multiple Sensor Modalities



Feature Level Fusion for Heterogeneous Multiple Sensor Modalities







Conclusion

1. Design and implemented the proposed MLB-DA approach and test it with Digits/UCM-AID dataset for cross class sets domain adaptation.

2. Developed the framework for data fusion from different sensors, and can be extended to different modalities.

3. Decision-level fusion (84% accuracy) and feature-level fusion (86% accuracy) are both implemented.

4. Initial benchmark and feasibility study of proposed approach have shown MLB-DA outperforms (min 10%) previous results of GTA for a single sensor.

thank you!