Generate To Adapt： Aligning Domains using Generative Adversarial Networks

Swami Sankaranarayanan* Yogesh Balaji* Carlos D. Castillo Rama Chellappa

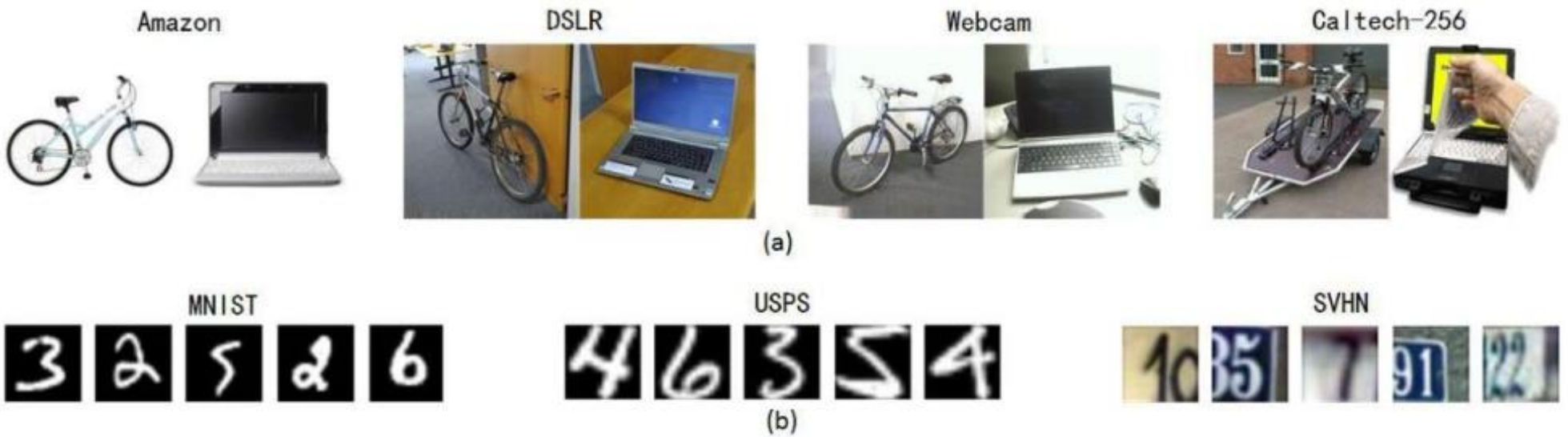
UMIACS, University of Maryland, College Park

CVPR 2018

Content：

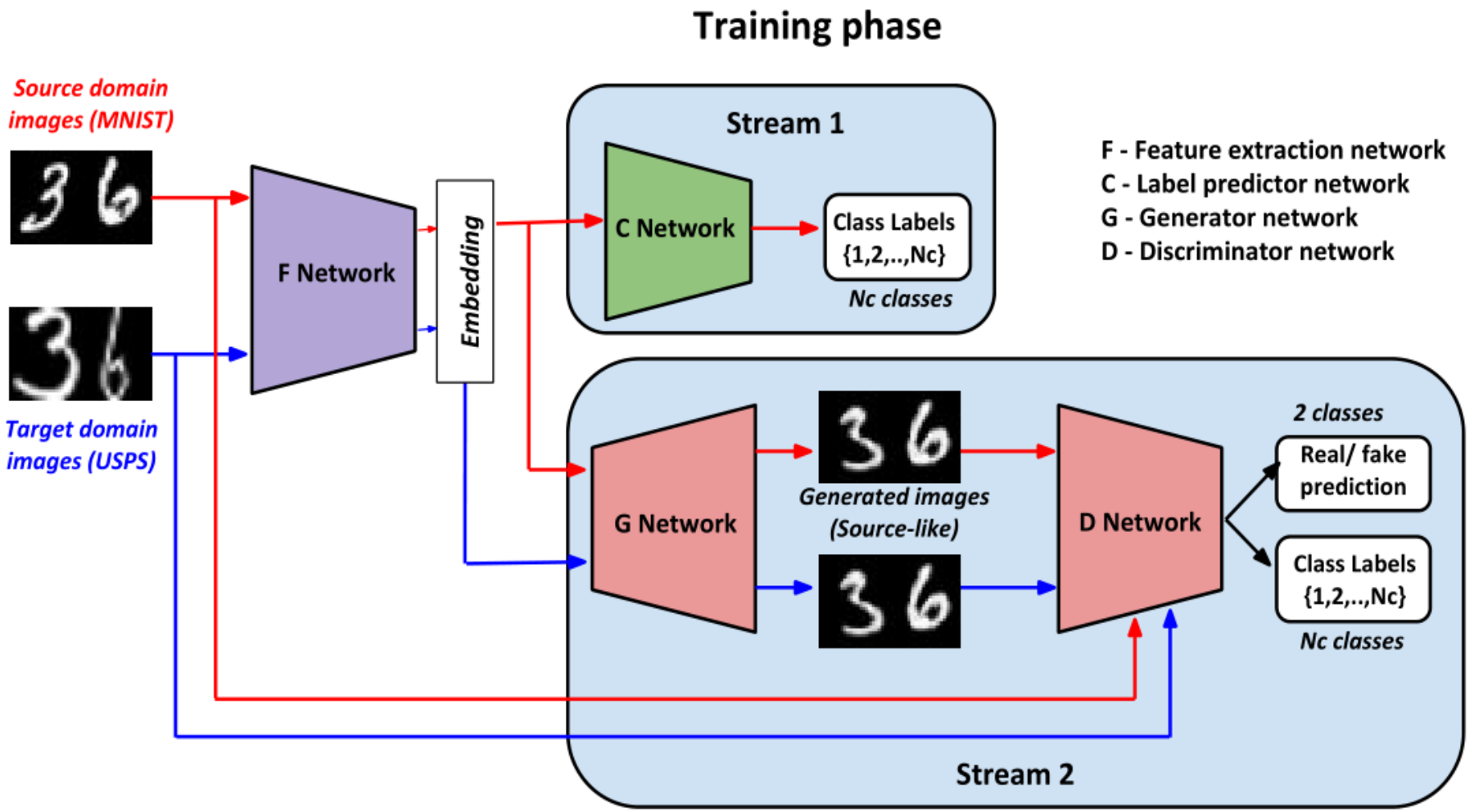
Introduciton:

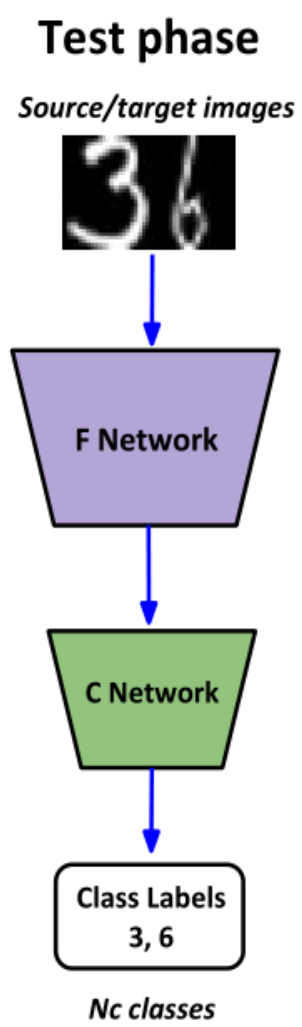
Due to many factors (e.g., illumination, pose, and image quality), there is always a distribution change or domain shift between two domains that can degrade the performance.



Introduction:

- 1. propose an approach that leverages unsupervised data to bring the source and target distributions closer in a learned joint feature space.
- 2. We accomplish this by inducing a symbiotic relationship between the learned embedding and a generative adversarial network.
- 3. by far the only GAN-based method that has been shown to work well across different datasets such as OFFICE and DIGITS.


Training phase
Source domain images (MNIST)
Target domain images (USPS)
F Network
Embedding
Stream 1
C Network
Class Labels {1,2,..,Nc}
Nc classes
F - Feature extraction network
C - Label predictor network
G - Generator network
D - Discriminator network
G Network
Generated images (Source-like)
D Network
2 classes
Real/ fake prediction
Class Labels {1,2,..,Nc}
Nc classes
Stream 2


Test phase
Source/target images
F Network
C Network
Class Labels 3, 6
Nc classes

$$x_g = [F(x), z, l]$$

Approach：

1. Given source images as input, D outputs two distributions D_{data} and D_{cls}

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

2. G is updated using a combination of adversarial loss and classification loss。

$$L_G = \min_{G} \mathbf{E}_{x\sim\mathcal{S}} - \log(D_{cls}(G(x_g))_y) + \log(1 - D_{data}(G(x_g))),$$

D data (x): the probability of the input being real；
D cls (x): the class probability distribution of the input x；
D cls (x)y ： the probability assigned by the classifier mapping D cls for input x to class y

3. F and C are updated。

$$L_C = \min_C \min_F \mathbf{E}_{x\sim\mathcal{S}} - \log(C(F(x))_y),$$
$$L_{cls,src} = \min_F \mathbf{E}_{x\sim\mathcal{S}} - \alpha \log(D_{cls}(G(x_g))_y))$$

4. The target embeddings output by F along with the random noise vector z and the fake label encoding l are input to G. The generated target images G(x_g) are then given as input to D.

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

Algorithm 1 Iterative training procedure of our approach

training iterations = N

for t in 1:N **do**

Sample k images with labels from source domain $\mathcal{S}$: $\{s_i, y_i\}_{i=1}^{k}$

Let $f_i = F(s_i)$ be the embeddings computed for the source images.

Sample k images from target domain $\mathcal{T}$: $\{t_i\}_{i=1}^{k}$

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.

Update discriminator using the following objectives:

$$L_D = L_{data,src} + L_{cls,src} + L_{adv,tgt} \tag{3}$$

- $L_{data,src} = \max_D \frac{1}{k} \sum_{i=1}^{k} \log(D_{data}(s_i)) + log(1 - D_{data}(G(f_{g_i})))$
- $L_{cls,src} = \max_D \frac{1}{k} \sum_{i=1}^{k} \log(D_{cls}(s_i)_{y_i})$
- $L_{adv,tgt} = \max_D \frac{1}{k} \sum_{i=1}^{k} \log(1 - D_{data}(G(h_{g_i})))$

Update the generator, only for source data, through the discriminator gradients computed using real labels.

$$L_G = \min_G \frac{1}{k} \sum_{i=1}^{k} - \log(D_{cls}(G(f_{g_i}))_{y_i}) + \log(1 - D_{data}(G(f_{g_i}))) \tag{4}$$

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.

$$L_F = L_C + \alpha\, L_{cls,src} + \beta\, L_{F_{adv}} \tag{5}$$

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

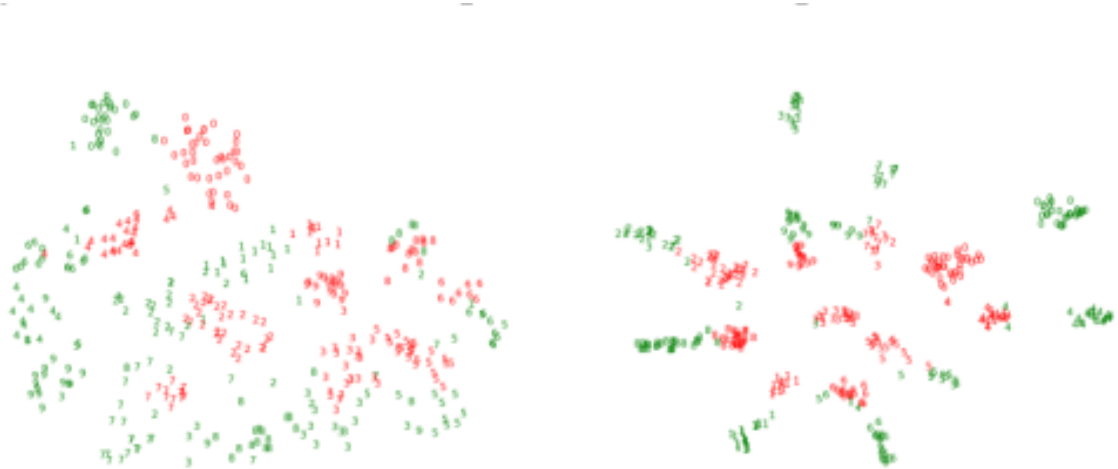
end for

Experiments and Results:

1. Digit Experiments

Table 1: Accuracy (mean ± std%) values for cross-domain recognition tasks over five independent runs on the digits based datasets. The best numbers are indicated in **bold** and the second best are underlined. – denotes unreported results. MN: MNIST, US: USPS, SV: SVHN. MN→US (p) denotes the MN→US experiment run using the protocol established in [17], while MN→US (f) denotes the experiment run using the entire datasets. (Refer to Digits experiments section for more details)

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



(a) Non adapted (b) Adapted

Figure 2: TSNE visualization of SVHN → MNIST adaptation. In (a), the source data shown in *red* is classified well into distinct clusters but the target data is clustered poorly. On applying the proposed approach, as shown in (b), both the source and target distributions are brought closer in a class consistent manner.

2. OFFICE experiments

Table 2: Accuracy (mean ± std%) values on the OFFICE dataset for the standard protocol for unsupervised domain adaptation [6]. Results are reported as an average over 5 independent runs. The best numbers are indicated in **bold** and the second best are underlined. − denotes unreported results. A: Amazon, W: Webcam, D: DSLR

Method	A → W	D → W	W → D	A → D	D → A	W → A	Average
ResNet - Source only [9]	68.4 ± 0.2	96.7 ± 0.1	99.3 ± 0.1	68.9 ± 0.2	62.5 ± 0.3	60.7 ± 0.3	76.1
TCA [23]	72.7 ± 0.0	96.7 ± 0.0	99.6 ± 0.0	74.1 ± 0.0	61.7 ± 0.0	60.9 ± 0.0	77.6
GFK [6]	72.8 ± 0.0	95.0 ± 0.0	98.2 ± 0.0	74.5 ± 0.0	63.4 ± 0.0	61.0 ± 0.0	77.5
DDC [33]	75.6 ± 0.2	76.0± 0.2	98.2 ± 0.1	76.5 ± 0.3	62.2 ± 0.4	61.5 ± 0.5	78.3
DAN [16]	80.5 ± 0.4	97.1 ± 0.2	99.6 ± 0.1	78.6 ± 0.2	63.6 ± 0.3	62.8 ± 0.2	80.4
RTN [18]	84.5 ± 0.2	96.8 ± 0.1	99.4 ± 0.1	77.5 ± 0.3	66.2 ± 0.2	64.8 ± 0.3	81.6
RevGrad [4]	82.0 ± 0.4	96.9 ± 0.2	99.1 ± 0.1	79.4 ± 0.4	68.2 ± 0.4	67.4 ± 0.5	82.2
JAN [19]	85.4 ± 0.3	97.4 ± 0.2	**99.8** ± 0.2	84.7 ± 0.3	68.6 ± 0.3	70.0 ± 0.4	84.3
Ours	**89.5** ± 0.5	**97.9** ± 0.3	**99.8** ± 0.4	**87.7** ± 0.5	**72.8** ± 0.3	**71.4** ± 0.4	**86.5**

3. Synthetic to Real experiments

In this experiment, we use CAD synthetic dataset and a subset of PASCAL VOC dataset as our source and target sets respectively.

Table 3: Accuracy (mean ± std%) values over five independent runs on the Synthetic to real setting. The best numbers are indicated in **bold**.

Method	CAD → PASCAL
VGGNet - Source only	38.1 ± 0.4
RevGrad [4]	48.3 ± 0.7
RTN [18]	43.2 ± 0.5
JAN [19]	46.4 ± 0.8
Ours	**50.4** ± 0.6

Ablation Study

Table 5: Ablation study for OFFICE A→W setting

Setting	Accuracy(in %)
Stream 1 - Source only	68.4
Stream 1 + Stream 2 (C_1 only)	80.5
Stream 1 + Stream 2 ($C_1 + C_2$)	89.5

Conclusion:

- We proposed a joint adversarial discriminative approach that transfers the information of the target distribution to the learned embedding using a generator-discriminator pair.

- address the problem using experiments on three different tasks