Unified Deep Supervised Domain Adaptation and Generalization

West Virginia University

Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, Gianfranco Doretto

International Conference on Computer Vision 2017

Introduction

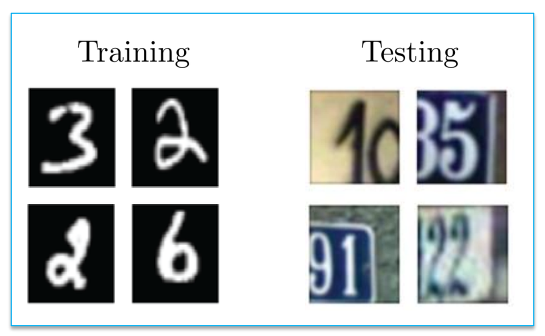
Problem: Improving visual recognition performance when source and target distributions are different for two scenarios:

- **Domain Adaptation:** When very few labeled target samples are available in training.
- **Domain Generalization:** When no information about target domain is available in training.

Why: There is always **covariate shift** between training and testing distributions.

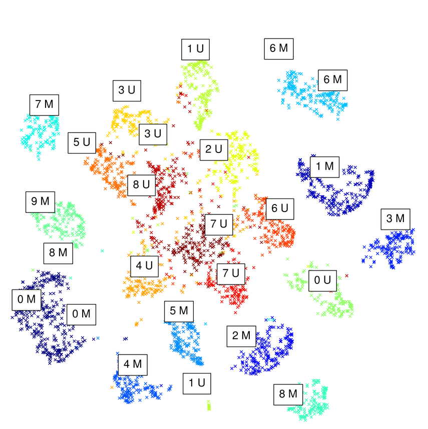
How: By exploiting Siamese structure and **Semantic Alignment loss** and **Separation loss** together with **pointwise** surrogates.

Covariate shift



Visualization of Digits in Feature Space

- Training *LeNet* using MNIST dataset.
- Testing on the USPS and MNIST datasets.
- Accuracy: MNIST = %95 and USPS = %65
- **2D visualization of samples in the feature space:**



- Samples from the same class but different distributions lie far from each other.
- Samples from the same class and same distribution lie close to each other.

Existing Supervised Domain Adaptation Methods

SDA assumes that few target samples per class are available in training.

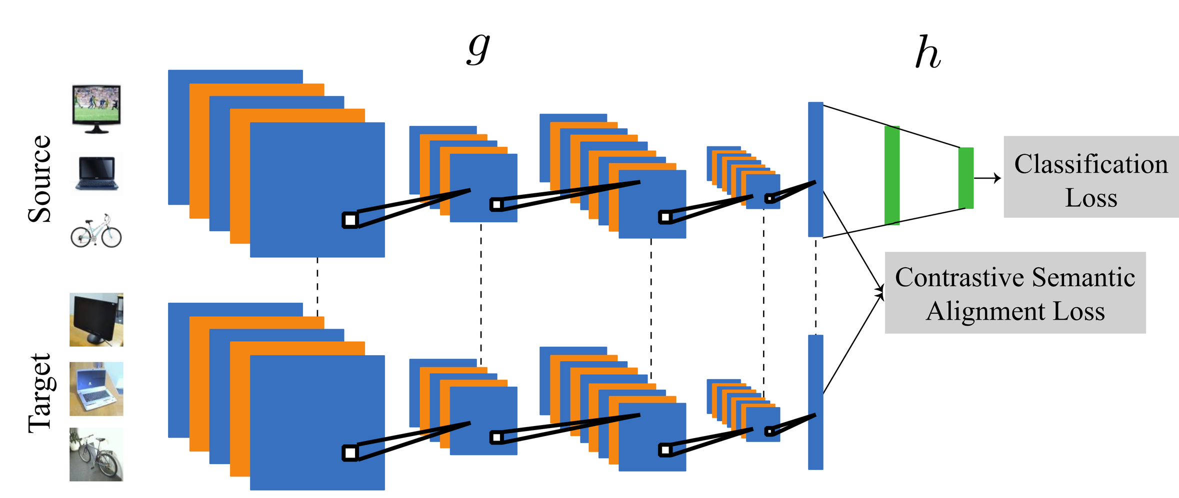
SDA tries to minimize the distance between corresponding classes.

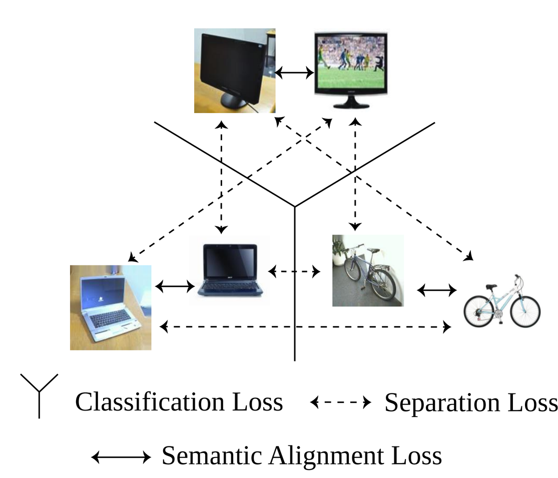
- Maximum Mean Discrepancy (MMD).
- Mean between two distributions.
- Second-or higher-order scatter tensors.

[1] E. Tzeng, et all. Deep domain confusion: Maximizing for domain invariance. arXiv, 2014.
[2] E. Tzeng, et all. Simultaneous deep transfer across domains and tasks. In ICCV, 2015.
[3] P. Koniusz, et all. Domain adaptation by mixture of alignments of second-or higher-order scatter tensors. In CVPR, 2017.

Our Approach to Solve Domain Adaptation

- Minimizing the distance between corresponding classes in the embedding space.
- Maximizing the distance between different classes and distributions in the embedding space.
- Using **pointwise surrogates** instead of distribution distances.





Semantic Alignment Loss for Positive Pairs

$$\mathcal{L}_{SA}(g) = \sum_{a=1}^{C} d(p(g(X_a^s)), p(g(X_a^t))) = \sum_{i,j} d(g(x_i^s), g(x_j^t))$$

$$d(g(x_i^s), g(x_j^t)) = \frac{1}{2}\|g(x_i^s) - g(x_j^t)\|^2$$

Penalty when two samples do not embed into the same point.

Separation Loss for Negative Pairs

$$\mathcal{L}_{S}(g) = \sum_{a,b|a\neq b} k(p(g(X_a^s)), p(g(X_b^t))) = \sum_{i,j} k(g(x_i^s), g(x_j^t))$$

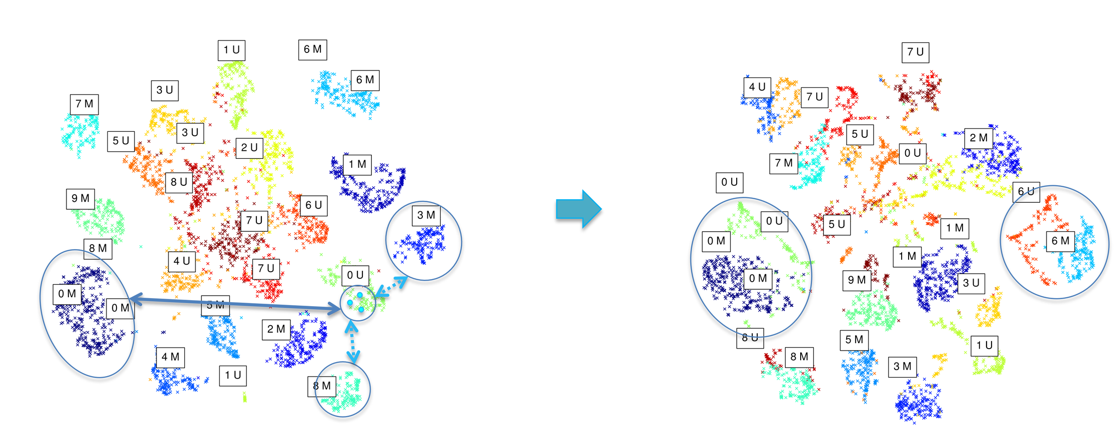
$$k(g(x_i^s), g(x_j^t)) = \frac{1}{2}\max(0, m - \|g(x_i^s) - g(x_j^t)\|)^2$$

Penalty when the distance between two samples is less than a threshold.

Total Loss

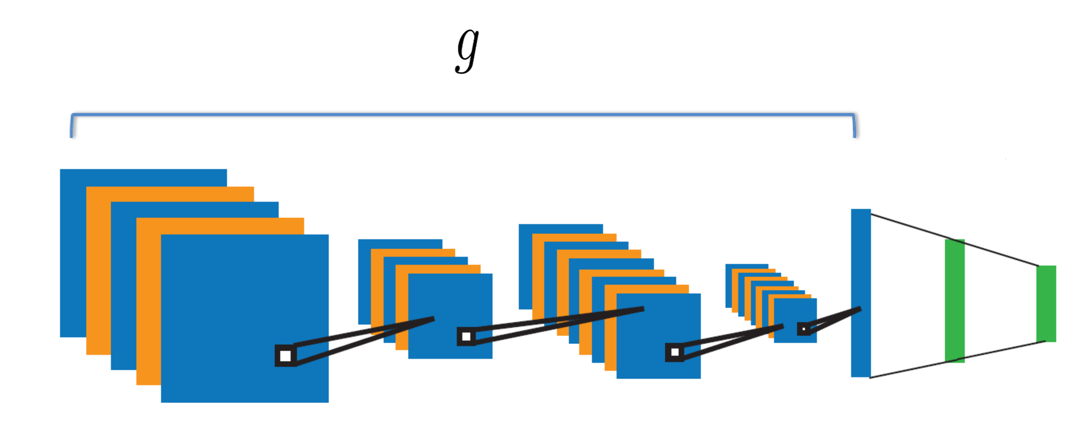
$$\mathcal{L}_{CCSA}(f) = \mathcal{L}_C(h \circ g) + \mathcal{L}_{SA}(g) + \mathcal{L}_S(g)$$

Note. $\mathcal{L}_{SA}(g)$ and $\mathcal{L}_S(g)$ can be defined in many ways. Here we only discussed one possible choice.

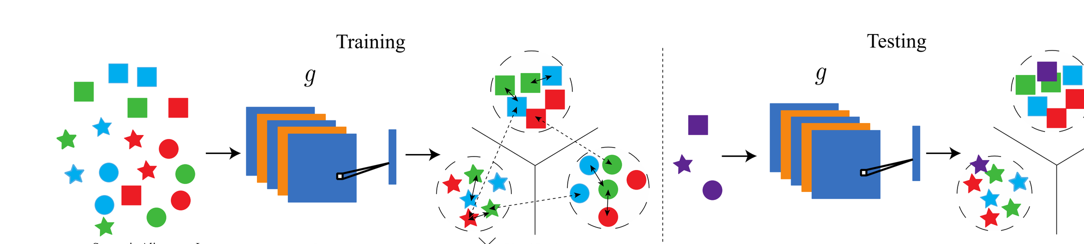


- This allows a target sample to be paired with all the source samples independently from other target samples.
- Even one target sample per class can **significantly** increase performance.
- Separation loss makes the the embedding space more separable (makes everything easy for the classifier).
- Easy to implement & is consistent with mini-batch learning.

Extension to Domain Generalization



- Domain generalization is looking for finding a domain invariant embedding function g.
- Our proposed method pulls together samples from the same class and different distributions.
- Our proposed method pushes apart samples from the different classes and distributions.
- We can use the same network structure and the same losses.

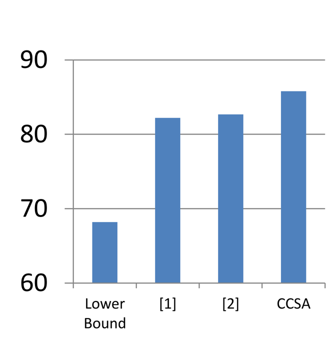


Domain Adaptation Results

Office Dataset

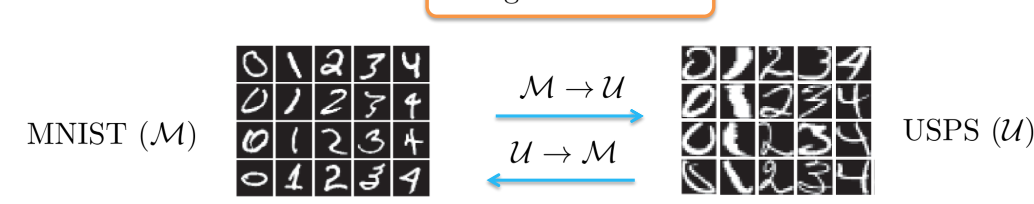
- Source: 20 samples per class for $\mathcal{A}$. 8 samples per class for the $\mathcal{D}$ and $\mathcal{W}$ and 3 target samples per class.
- Prediction function h: An fc layer with softmax activation.
- Embedding function g: Convolutional layers of VGG-16 followed by 2 fc layers with size of 1024 and 128.

	Lower Bound	[1]	[2]	CCSA
$\mathcal{A} \to \mathcal{D}$	62.3 ± 0.8	86.1 ± 1.2	86.3 ± 0.8	**89.0 ± 1.2**
$\mathcal{A} \to \mathcal{W}$	61.2 ± 0.9	82.7 ± 0.8	84.5 ± 1.7	**88.2 ± 1.0**
$\mathcal{W} \to \mathcal{A}$	51.6 ± 0.9	65.0 ± 0.5	65.7 ± 1.7	**72.1 ± 1.0**
$\mathcal{W} \to \mathcal{D}$	95.6 ± 0.7	97.6 ± 0.2	97.5 ± 0.7	97.6 ± 0.4
$\mathcal{D} \to \mathcal{A}$	58.5 ± 0.8	66.2 ± 0.3	66.5 ± 1.0	**71.8 ± 0.5**
$\mathcal{D} \to \mathcal{W}$	80.1 ± 0.6	95.7 ± 0.5	95.5 ± 0.6	**96.4 ± 0.8**
Average	68.2	82.21	82.68	**85.8**

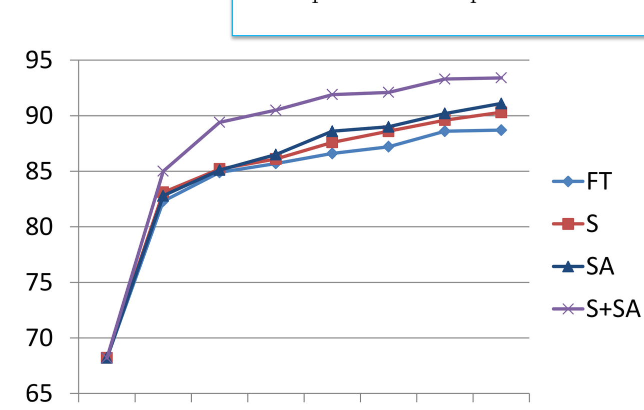


[1] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In ICCV, 2015.
[2] P. Koniusz, Y. Tas, and F. Porikli. Domain adaptation by mixture of alignments of second-or higher-order scatter tensors. In CVPR, 2017.
[3] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In CVPR, 2017.
[4] M.Y. Liu and O. Tuzel. Coupled generative adversarial networks. In Advances in NIPS, 2016.

Digits Datasets



- Randomly selected 2000 samples from MNIST.
- Randomly selected 1800 samples from USPS.
- Repeated the experiments for several number of target samples per class (from one to seven).
- Repeated each experiment 10 times.



$\mathcal{M} \to \mathcal{U}$

	FT	S	SA	S+SA
$n = 1$	82.3	83.1	82.8	85.0
$n = 2$	84.9	88.2	88.1	89.4
$n = 3$	85.7	86.1	86.5	90.1
$n = 4$	86.6	87.6	88.6	91.4
$n = 5$	87.2	88.6	89.0	92.4
$n = 6$	88.6	89.6	90.2	93.0
$n = 7$	88.7	90.3	91.1	92.9
$n = 8$	88.9	90.8	91.2	92.8
Unsupervised Adversarial Methods				
ADDA [3]	89.4			
CoGAN [4]	91.2			

Domain Generalization Results

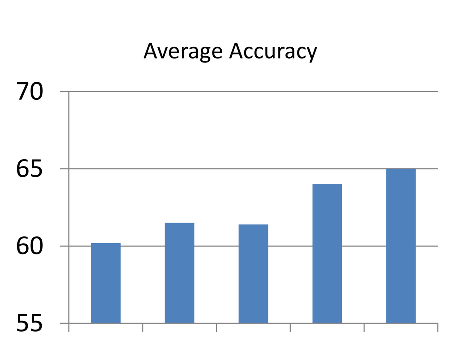
VLCS Dataset

VOC2007 ($\mathcal{V}$) Caltech-101 ($\mathcal{C}$) LabelMe ($\mathcal{L}$) SUN09 ($\mathcal{S}$)

- 5 shared object categories (bird, car, chair, dog, and person)

- Pairs: For each source sample, we randomly selected 5 samples from remaining source.
- In order to compare our results with the state-of-the-art, we used DeCaF-fc6 features and repeated the experiments 20 times.
- Embedding function g: 2 fc layers with size of 1024 and 128 with ReLU activation.
- Prediction function h: An fc layer with softmax activation.

	Lower Bound			Domain Generalization			
	1NN	SVM	LB	UML [5]	LRE-SVM [6]	SCA [7]	Ours
$\mathcal{L},\mathcal{C},\mathcal{S} \to \mathcal{V}$	57.2	58.4	59.1	56.2	60.5	64.3	**67.1**
$\mathcal{V},\mathcal{C},\mathcal{S} \to \mathcal{L}$	52.4	55.2	55.6	58.5	59.7	59.6	**62.1**
$\mathcal{V},\mathcal{L},\mathcal{S} \to \mathcal{C}$	90.5	85.1	86.1	91.1	88.1	88.9	**92.3**
$\mathcal{V},\mathcal{L},\mathcal{C} \to \mathcal{S}$	56.9	55.2	54.6	58.4	54.8	**59.2**	59.1
$\mathcal{C},\mathcal{S} \to \mathcal{V},\mathcal{L}$	55.0	55.5	55.3	56.4	55.0	**59.5**	59.3
$\mathcal{C},\mathcal{L} \to \mathcal{V},\mathcal{S}$	52.6	51.8	50.9	**57.4**	52.8	55.9	56.5
$\mathcal{V},\mathcal{C} \to \mathcal{L},\mathcal{S}$	56.6	59.9	60.1	55.4	58.8	**60.7**	60.2
Average	60.1	60.2	60.3	61.5	61.4	64.0	**65.0**



[5] C. Fang, Y. Xu, and D. N. Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In International Conference on Computer Vision, 2013.
[6] Z. Xu, W. Li, L. Niu, and D. Xu. Exploiting low-rank structure from latent domains for domain generalization. In ECCV, pages 628643, 2014.
[7] M. Ghifary, D. Balduzzi,W. B. Kleijn, and M. Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. TPAMI, 2017.