Domain Adaptation for Visual Applications

Gabriela Csurka

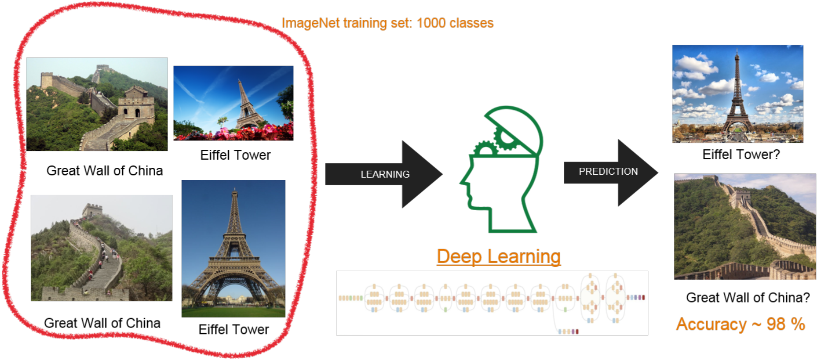
NAVER LABS Europe (NLE), Meylan, France

gabriela.csurka@naverlabs.com

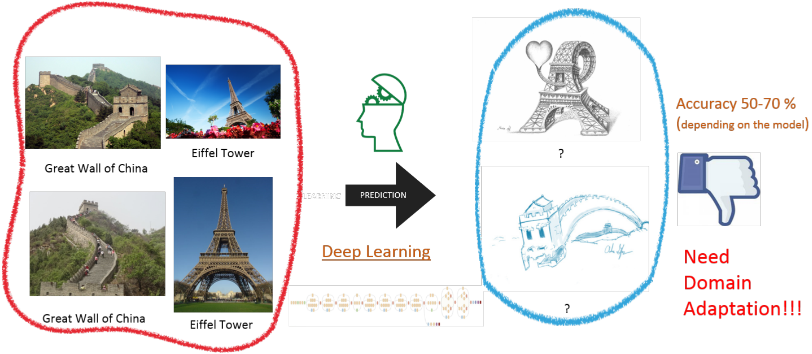
Outline

1. Motivation
2. Shallow Domain Adaptation methods
3. DA using Deep Learning
4. Deep Domain Adaptation Methods
5. Beyond image classification

Image Classification



Domain Shift

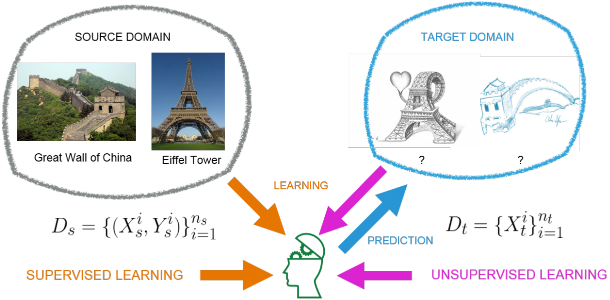


Train: Images

Test: Drawings

Domain adaptation (DA)

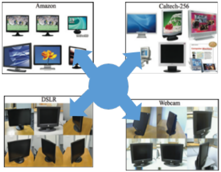
Leveraging labeled data in one or more related domains, referred to as *source domains*, to learn a classifier for data in a *target domain*.



- Unsupervised (US) DA when no label is available in the target domain
- Semi-supervised (SS) DA when a few labels are available in the target domain

Example scenarios

Object recognition



Object detection

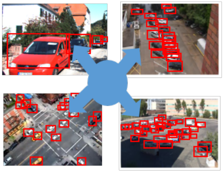
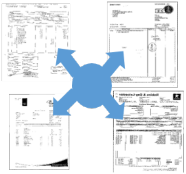
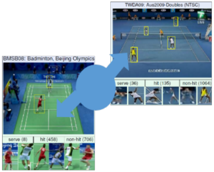


Image segmentation

Document image categorization



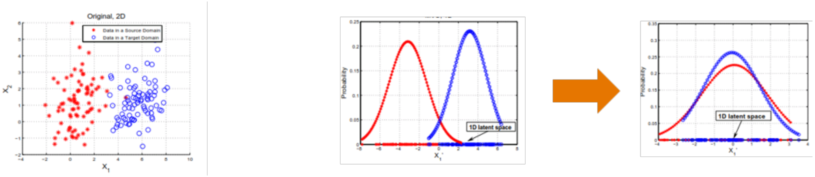
Action recognition



Person Re-identification

Key idea: solve the distribution mismatch

By finding feature representation/embedding where the distributions between source and target match.



The distribution mismatch, is measured by the Maximum Mean Discrepancy[1] (MMD):

$$MMD(S,T) = \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} \psi(\mathbf{x}_i^s) - \frac{1}{N_t} \sum_{j=1}^{N_t} \psi(\mathbf{x}_j^t) \right\|_{\mathcal{H}}$$

in the Reproducing Kernel Hilbert Space (RKHS).

[1] **Borgwardt *et al.*, Integrating structured biological data by kernel maximum mean discrepancy, Bioinformatics (22), 2006.**

Outline

1. Motivation
2. Shallow Domain Adaptation methods
3. DA using Deep Learning
4. Deep Domain Adaptation Methods
5. Beyond image classification

Shallow Domain Adaptation Methods

- **Instance re-weighting**
  - Correcting the sample bias, Dudik *et al.*, NIPS'05, Sugiyama *et al.*, NIPS'07
  - Transfer Adaptive Boosting, Dai *et al.*, ICML'07, Al-Stouhi *et al.*, PKDD'11
- **Parameter adaptation**
  - Adjust SVM parameters, Yang *et al.*, MM'07, Bruzzone *et al.*, PAMI'10
  - Multiple Kernel Learning, Duan *et al.*, CVPR'10
- **Feature augmentation methods**
  - Frustratingly easy feature augmentation, Daume *et al.*, CORR'09
  - Geodesic Flow Sampling (GFS), Gopalan *et al.*, ICCV'11
  - Geodesic Flow Kernel, (GFK), Gong *et al.*, CVPR'12
- **Feature space alignment**
  - Subspace Alignment (SA), Fernando *et al.*, ICCV'13
  - Correlation Alignment (CORAL), by Sun *et al.*, AAAI'15
- **Feature space transformation**
  - Unsupervised and supervised (most popular, see next)
  - Local transformation, FarajiDavar *et al.*, BMVC'14, Courty *et al.*, CORR'15
- **Heterogeneous feature transformation**
  - Dictionary Learning, Shekhar *et al.*, CVPR'13
  - Heterogeneous Spectral Mapping, Shi *et al.*, ICDM'10
  - Domain Adaptation Manifold Alignment, Wang *et al.*, IJCAI'11

Feature space transformation

Learning a common feature projection ϕ by minimizing the distributions mismatch.

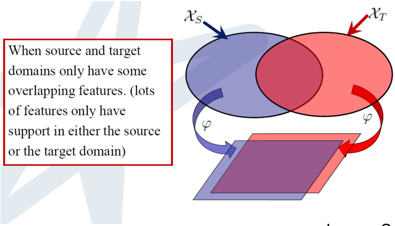


Image: Courtesy to Dong Xu.

Unsupervised feature transformation

- learns the transformation without using any class labels

Supervised feature transformation

- exploits class labels only from source (US scenario)
- exploits class labels from both source and target (SS scenario)

US feature transformation methods

- **Transfer Component Analysis** (TCA), Pan *et al.*, IJCAI'09
  - minimizes the distance between source and target means in the subspace
- **Marginalized Denoising Autoencoders** (MDA), Chen *et al.*, ICML'12
  - reconstructs original features from their noised counterparts
- **Domain Invariant Projection** (DIP), Baktashmotlagh *et al.*, ICCV'13
  - compares directly the distributions in the RKHS
- **Transfer Sparse Coding** (TSC), Long *et al.*, CVPR'13
  - learns robust sparse representations
- **Statistically Invariant Embedding** (SIE), Baktashmotlagh *et al.*, CVPR'14
  - minimizes the Hellinger distance on a Riemannian manifold
- **Transfer Joint Matching** (TJM), Long *et al.*, CVPR'14
  - combines MMD minimization and instance re-weighting

US results on the Office-Caltech dataset

	C ->A	D ->A	W ->A	A ->C	D ->C	W ->C	A ->D	C ->D	W->D	A ->W	C ->W	D ->W	Avg
SA	52.7	38.0	39.4	41.6	**44.8**	34.7	**46.4**	49.0	78.9	40.7	42.7	83.4	49.4
CORAL	52.1	37.7	36.0	45.1	33.8	33.7	39.5	45.9	86.6	44.4	46.4	84.7	48.8
GFK	54.1	33.1	36.6	40.1	39.2	28.9	35.7	44.6	81.2	38.6	39	80.3	46.0
TCA	38.2	32.1	30.1	27.8	31.7	29.3	33.1	41.4	87.3	37.6	38.6	86.1	42.8
SIE	46.7	37.4	**41.3**	42.7	34.6	35.0	40.3	44.1	73.9	42.0	45.2	74.3	46.5
JDA	44.8	33.1	32.8	39.4	31.5	31.2	39.5	45.2	89.2	38.0	41.7	89.5	46.3
TJM	58.6	35.1	40.8	**45.7**	39.6	**34.8**	42.0	49.0	83.4	42.0	48.8	82.0	50.2
ATTM	**60.9**	**38.7**	39.7	42.9	32.4	34.0	39.5	**50.3**	**89.8**	**50.5**	**62.0**	**88.8**	**52.5**
MDA	54.1	37.3	38.8	44.6	33.2	35.4	39.5	44.6	82.8	36.6	48.8	82.3	48.2

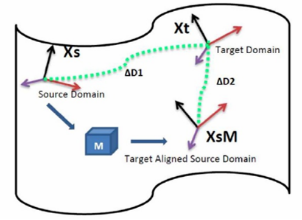
SA - Subspace Alignment, Fernando *et al.*, ICCV'13
CORAL - Correlation Alignment, Sun AAAI'15
GFK - Geodesic Flow Kernel, B. Gong *et al.*, CVPR'12
TCA - Transfer Component Analysis, Pan *et al.*, IJCAI'09
SIE - Statistically Invariant Embedding, Baktashmotlagh *et al.*, CVPR'14
JDA - Joint Distribution Adaptation, Long *et al.*, ICCV'14
TJM - Transfer Joint Matching, Long *et al.*, CVPR'14
ATTM - Adaptive Transductive Transfer Machines, Farajidavar *et al.*, BMVC'14
MDA - Marginalized Denoising Autoencoders, Chen *et al.*, ICML'12

Feature Space Alignment: SA and CORAL

Subspace Alignment (SA)

Learns an alignment M between the PCA subspaces $\mathbf{X}_s$ and $\mathbf{X}_t$ of the source and target space respectively:

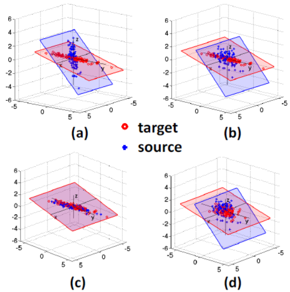
$$\mathbf{M}^* = argmin||\mathbf{X}_s\mathbf{M} - \mathbf{X}_t||$$



Correlation Alignment (CORAL)

The main idea is a ”whitening” of the source data using its covariance $\mathbf{C}_s$ followed by a ”re-coloring” using the target covariance matrix $\mathbf{C}_t$:

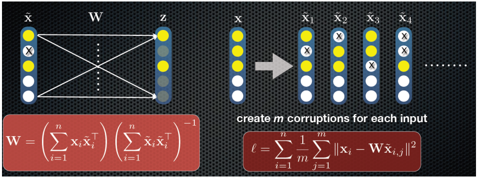
$$\mathbf{X}_s * \mathbf{C}_s^{-1/2} * \mathbf{C}_t^{-1/2}$$



Marginalized Denoising Autoencoders[2]

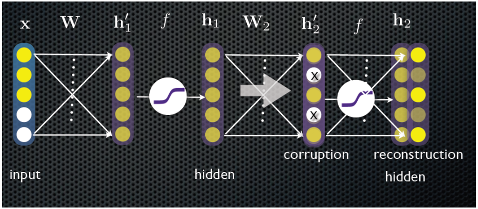
MDA

Using drop-out noise and marginalizing out the corruption yields closed form solution for W, which depends only on the data covariance and the drop-out noise level.



Stacked MDA

Can be easily made "deep", by stacking several MDA layers. Nonlinearities between layers and concatenation of several layers improves the results.



---

[2] Chen *et al.*, Marginalized Stacked Denoising Autoencoders for Domain Adaptation, ICML'12

Supervised feature transformation methods

Exploit class labels from source and when available from target.

- **Max-Margin Domain Transform** (MMDT), Hoffman *et al.*, ICLR'13
  - optimizes jointly the transformation and classifier
- **Joint Distribution Adaptation** (JDA), Long *et al.*, ICCV'13
  - adapts both the marginal and the conditional distribution between domains
- **Adaptation Regularization based TL** (ARTL), Long *et al.*, TDKE'14
  - combines structural risk, manifold consistency and discrepancy loss
- **Joint Distribution Optimal Transport** (JDOT), Courty, *et al.*, NIPS'17
  - minimizes an optimal transport between joint distributions

Extensions proposed for unsupervised feature transformation methods:

- **Semi-supervised Transfer Component Analysis** (SSTCA), Pan *et al.*, TNN'11
- **Domain Invariant Projection** (DIP-CC), Baktashmotlagh *et al.*, ICCV'13
- **Regularized Domain Instance Denoising** (eMDA), Csurka *et al.*, TASK-CV'16

Metric Learning (ML) based DA methods

Exploit class labels from both source and target.

- **Regularized Distance Metric Learning** (R-DML), Zha *et al.*, IJCAI'09
  - uses either Log-determinant or Manifold regularization
- **Information-Theoretic Metric Learning** (ITML), Saenko *et al.*, ECCV'10
  - uses Information-Theoretic Metric to learn a distance across domains
- **Bayes Nearest Neighbor based DA** (NBNN-DA), Tommasi *et al.*, ICCV'13
  - combines Naive Bayes Nearest Neighbor sample selection with ITML
- **Domain Specific Class Means** (MLDSCM), Csurka *et al.*, TASK-CV'14
  - minimizing soft-max distances to domain specific class means
  - can take advantage of multiple sources

SS results on Office-Caltech dataset

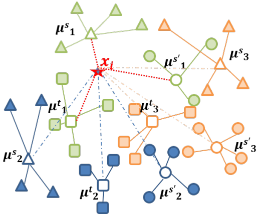
	C ->A	D ->A	W ->A	A ->C	D ->C	W ->C	A ->D	C ->D	W->D	A ->W	C ->W	D ->W	Average
GFK	46.1	46.2	32.1	39.6	33.9	32.1	50.9	55	74.1	56.9	57	74.6	49.9
SA	45.3	45.8	44.8	38.4	**35.8**	**34.1**	55.1	56.6	**82.3**	60.3	60.7	**84.8**	**53.7**
MMDT	49.4	46.9	47.7	36.4	34.1	32.2	56.7	56.5	67	64.6	63.8	74.1	52.5
SSTCA	47.1	40.1	41.5	**40.4**	34.2	33.5	39	41.7	77.8	41.1	36.2	80.5	46.1
ITML	33.7	30.3	32.3	27.3	22.5	21.7	33.7	35	51.3	36	34.7	55.6	34.5
MLDSCM	**50.6**	**48.8**	**48.4**	34.9	34.2	33.4	**62.1**	**61.6**	64.7	**66.1**	**65.1**	71.5	**53.4**

- **GFK** - Geodesic Flow Kernel, B. Gong *et al.*, CVPR'12
- **SA** - Subspace Alignment, Fernando *et al.*, ICCV'13
- **MMDT** - Max-Margin Domain Transforms, Hoffman *et al.*, ICLR'13
- **SSTCA** - Semi-Supervised TCA, Pan *et al.*, TNN'11
- **ITML** - Information Theoretic Metric Learning, Saenko *et al.*, ECCV'10
- **MLDSCM** - ML for Domain Specific Class Means, Csurka *et al.*, TASK-CV'14

Domain Specific Class Means[3] (DSCM)

DSCM

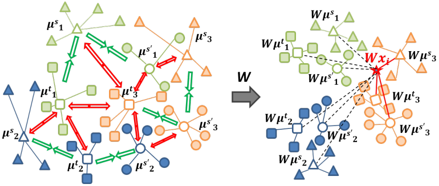
Easily handles multiple domains d, where x_i is assigned to the class for which the weighted soft-max distances to the corresponding domain specific class means (μ_i^d) is minimal.



MLDSCM

Learns a transformation **W** minimizing the weighted soft-max distances for each instance:

$$p(c|\mathbf{x}_i) = \frac{\sum_d w_d e^{\left(-\frac{1}{2}\|\mathbf{W}\mathbf{x}_i - \mathbf{W}\mu_c^d\|\right)}}{\sum_{c'} \sum_d w_d e^{\left(-\frac{1}{2}\|\mathbf{W}\mathbf{x}_i - \mathbf{W}\mu_{c'}^d\|\right)}}$$



[3] Csurka *et al.*, Domain adaptation with a domain specific class means classifier. TASK-CV'14

To summarize

Early methods require labeled target examples

- *e.g.* instance re-weighting, parameter adaptation,
- hence, can be applied only to semi-supervised DA scenario

Simple methods can performs pretty well

- *e.g.* Subspace Alignment, Correlation Alignment, MDA
- they have no/few parameters or closed form solutions

Feature space transformation are the most popular ones

- many of them relies on the Maximum Mean Discrepancy (MMD)
- can be unsupervised or supervised exploiting labels
- optimizing conditional or joint distributions helps

Best performing methods

- *e.g.* Transfer Joint Matching, Adaptive Transductive Transfer Machines
- joint marginal and conditional distributions and instance re-weighting

Outline

1. Motivation
2. Shallow Domain Adaptation methods
3. DA using Deep Learning
4. Deep Domain Adaptation Methods
5. Beyond image classification

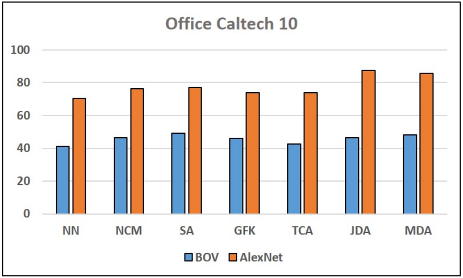
DA using Deep Learning

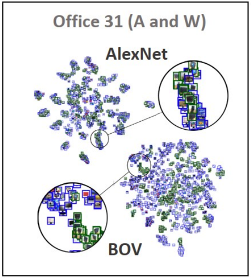
- **Shallow methods using deep features**
  - use the deep model as feature extractor
  - apply any shallow DA method using these features
- **Using fine-tuned deep architectures**
  - fine-tune the deep model on the source
  - apply the fine-tuned model on the target
- **Shallow methods using fine-tuned deep features**
  - fine-tune the deep model on the source
  - use the fine-tuned model as feature extractor
  - apply any shallow DA method using these features
- **Deep DA models**
  - deep Siamese architectures built for domain adaptation
  - the streams, corresponding to source and target, are initialized with a deep model fine-tuned on the source

Shallow methods using deep features

Deep models used as feature extractors[4]

- Activations of the deep CNN model can be used as image representation.
- Popular models are: AlexNet, VGG, ResNet or GoogleNet.
- Best candidates are layers preceding the softmax layer (fc6, fc7, PreLogit).

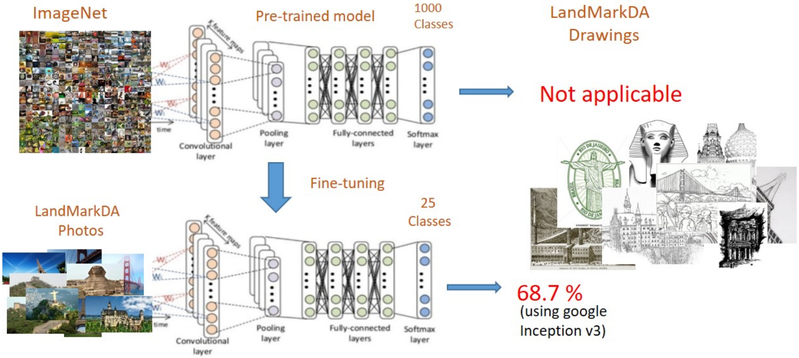




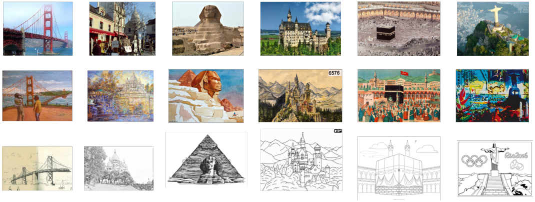
- Using deep features in shallow methods allows a gain above 20%.
- These features being more abstract decreases the domain bias.

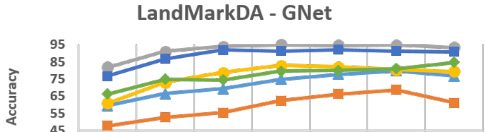
[4] Donuahe *et al.*, DeCAF: A deep convolutional activation feature for generic visual recognition, ICML'14.

Fine-tuning the model on the source



Fine-tuning on the LandmarkDA[5] dataset

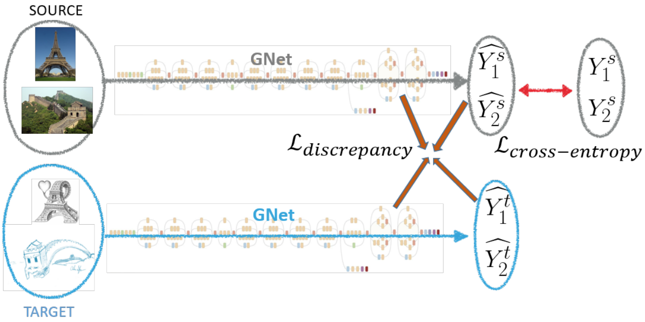




	FT(Lg)	FT(7c)	FT(7a)	FT(6e)	FT(6c)	FT(6a)	FT(all)
Ph->Pt	59.5	66.4	69.3	74.9	77.8	79.7	76.7
Ph->Dr	47.8	52.7	55.3	62.4	66.3	68.7	61.1
Pt->Ph	81.9	91	93.9	94.9	94.6	94.7	93.3
Pt->Dr	60.8	72.9	78.7	82.8	82.1	80.3	79.3
Dr->Ph	76.6	86.8	91.9	91.2	92	91.3	90.8
Dr->Pt	66.3	74.8	74.4	79.5	79.9	80.9	84.7

[5] https://www.researchgate.net/publication/319208011_LandMarkDA_domain_adaptation_dataset.

Discrepancy based DAN

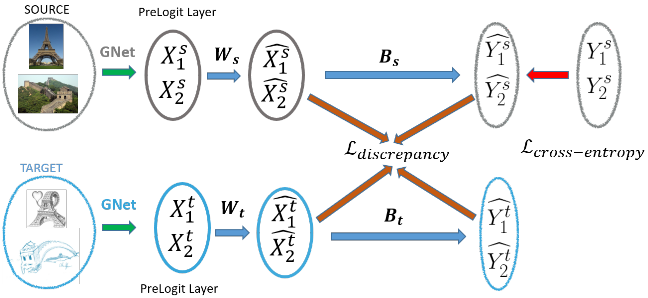


Joint distribution discrepancy[6] (JDD):

$$\mathcal{L}_{JDD} = \left\| \frac{1}{N_s} \sum_{i}^{N_s} \phi(\mathbf{X}_i^s) \otimes \psi(\widehat{\mathbf{Y}}_i^s) - \frac{1}{N_t} \sum_{i}^{N_t} \phi(\mathbf{X}_i^t) \otimes \psi(\widehat{\mathbf{Y}}_i^t) \right\|_{\mathcal{F}\otimes\mathcal{G}}^2 ,$$

[6] Long *et al.*, Deep Transfer Learning with Joint Adaptation Networks, CORR'15

Shallow Adaptation Network (SDAN)

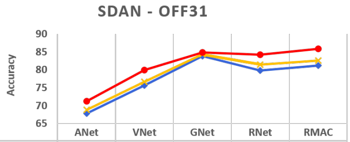


Joint distribution discrepancy[7] (JDD):

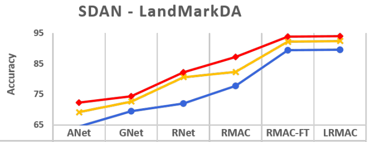
$$\mathcal{L}_{JDD} = \left\| \frac{1}{N_s} \sum_{i}^{N_s} \phi(\widehat{\mathbf{X}}_i^s) \otimes \psi(\widehat{\mathbf{Y}}_i^s) - \frac{1}{N_t} \sum_{i}^{N_t} \phi(\widehat{\mathbf{X}}_i^t) \otimes \psi(\widehat{\mathbf{Y}}_i^t) \right\|_{\mathcal{F}\otimes\mathcal{G}}^2 ,$$

[7] Long *et al.*, Deep Transfer Learning with Joint Adaptation Networks, CORR'15

SDAN results



	ANet	VNet	GNet	RNet	RMAC
BL-NA	67.8	75.7	83.9	79.8	81.2
SDAN-MMD-X	68.8	76.7	84.3	81.5	82.6
SDAN-JDD-XY	71.3	80	84.9	84.2	85.9



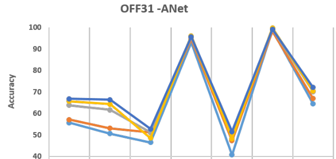
	ANet	GNet	RNet	RMAC	RMAC-FT	LRMAC
BL-NA	64.4	69.4	72	77.7	89.4	89.6
SDAN-MMD-X	69.1	72.6	80.6	82.3	92.2	92.4
SDAN-JDD-XY	72.3	74.3	82.2	87.2	93.8	94

Findings:

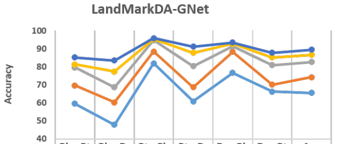
- SDAN allows for significant improvement over training only on the source.
- JDD results show the importance of considering the prediction layer ($\widehat{\mathbf{Y}}$).
- BL with GoogleNet or RestNet features can be better that SDAN obtained with AlexNet or VGG features.
- On Office31, GoogleNet BL is better than any SOA method built on AlexNet (best 80.4% with LRT, Sener *et al.*, NIPS'16)

Always compare methods built on the same original architectures and shallow methods using features extracted from the same deep model!

Deep DA model versus deep features



	A ->D	A ->W	D ->A	D ->W	W ->A	W->D	AVG
NA	55.7	50.6	46.5	93.1	40.7	98.9	64.4
CORAL	57.1	53.1	51.1	94.6	47.3	98.2	66.9
FT	63.8	61.6	51.1	95.4	49.8	99	70.1
FT+ CORAL	65.7	64.3	48.5	96.1	48.2	99.8	70.4
D-CORAL	66.8	66.4	52.8	95.7	51.5	99.2	72.1



	Ph->Pt	Ph->Dr	Pt->Ph	Pt->Dr	Dr->Ph	Dr->Pt	Avg
NA	59.5	47.8	81.9	60.8	76.6	66.3	65.5
SDAN	69.6	60.3	88.5	68.6	88.3	70.1	74.2
FT	79.7	68.7	94.7	80.3	91.3	80.9	82.6
FT + SDAN	81.3	77.4	95.3	87.8	92.7	85.2	86.6
DDAN	85.2	83.5	95.9	91.2	93.4	87.7	89.5

Findings:

- If the domain shift is small (D↔W), almost no gain is obtained with adaptation.
- Fine-tuning the deep model on the source outperforms the shallow model.
- If the target (A,Ph) is closer to the initial domain (ImageNet) than the source (W,D,Pt,Dr), fine tuning on the source seems sufficient. In the opposite case, adaptation yields strong improvements.
- **Shallow methods using fine-tuned model deep features is close to best.**

To summarize

Main advantages of shallow methods

- they are simple and low cost solutions
- same architecture can be applied to any vectorial representation
- it is important to use strong representations (deep features)

Main advantages of deep methods

- they can adjust the feature representation to the problem
- if appropriately trained they often outperform the shallow methods

Shallow methods using fine-tuned deep features

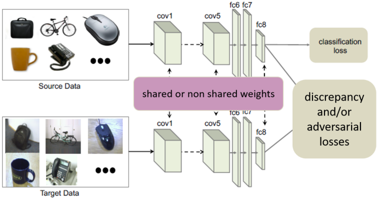
- combines the strength of deep learning and domain adaptation
- close to results obtained with the corresponding deep architecture
- no need to build DA dedicated (Siamese) deep architectures
- it requires less computational cost, easier to deploy on mobile phone
- the fine-tuning can be done in advance, before seeing the target

Outline

Discriminative models

Main framework

- Siamese architecture
- pre-trained on the source
- cross-entropy on source
- discrepancy and/or
adversarial losses



Minimizing the feature discrepancy

► DAN, Long *et al.*, ICML'15, DeepCORAL, Sun *et al.*, TASK-CV'16

Minimizing the joint feature/label distributions

► DeepJDOT, Damodaran *et al.*, ECCV'16, JAN, Long *et al.*, ICML'17

Encouraging domain confusion (adversarial)

► DANN, Ganin *et al.*, JMLR'16, ADDA, Tzeng *et al.*, CVPR'17

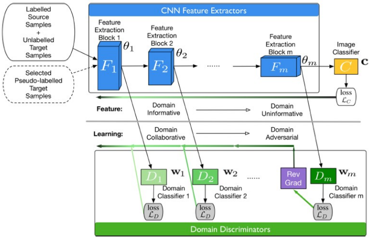
Combine discrepancy minimization with adversarial learning

► MCDDA, Saito, *et al.*, CVPR'18, CDAN, Long *et al.*, NIPS'18

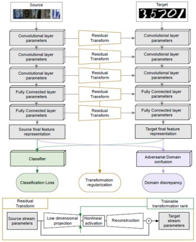
Office 31	CNN	DAN	JAN	DANN	ADDA	CDAN
AlexNet	70.1	72.9	76	74.3	74.7	**76.9**
ResNet-50	76.1	80.4	84.3	82.2	82.9	**87.7**

Per layer based adaptations

CAN



RPT



Automatic Domain Alignment Layers (ADial) Carlucci *et al.*, ICCV'17

- designed to match the source and target feature distributions to a reference one

Collaborative and Adversarial Network, (CAN), Zhang *et al.*, CVPR'18

- to learn simultaneously domain-informative and uninformative features

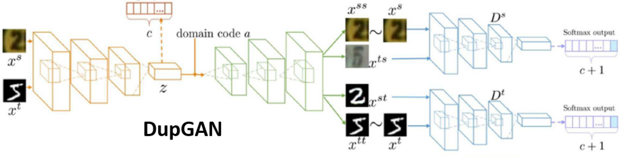
Residual Parameter Transfer, (RPT) Rozantsev *et al.*, CVPR'18

- used to learn the parameter adaptation between source and target layers

Office31	CNN	JAN	CDAN	ADial
AlexNet	70.1	76	76.9	**77.1**

Office31	CNN	JAN	CDAN	RPT	CAN
Resnet	76.1	84.3	**87.7**	81.7	87.2

Generative models



Coupled Generative Adversarial Networks (CoGAN), Liu *et al.*, NIPS'16

- couples two GANs, each corresponding to one of the domains

Pixel-Level Domain Adaptation (PixelDA), Bousmalis *et al.*, CVPR'17

- adapts source images to appear as if drawn from the target domain

Domain Transfer Network (DTN), Taigman, *et al.*, ICLR'17

- relies on cross-domain image translation

Aligning Domains using GAN, (ADGAN), Sankaranarayanan *et al.*, CVPR'18

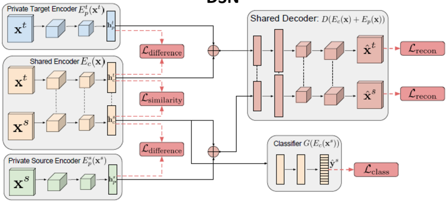
- combines joint feature learning with adversarial image generation

Duplex Generative Adversarial Network (DupGAN), Hu *et al.*, CVPR'18

- uses a duplex discriminator, one for each domain

Auto-encoder based models

DSN



Deep Reconstruction Classification Network (DRCN), Ghifary *et al.*, ECCV'16

- alternates between source label prediction and target data reconstruction

Domain Separation Networks (DSN), Bousmalis *et al.*, NIPS'16

- shared and domain specific encodings and one shared decoding

MNIST

USPS

SVHN

	CNN	ADDA	DeepCoral	MCDDA	DeepJDOT	CoGAN	DupGAN	DRCN	DSN
MNIST->USPS	86.8	89.4	89.33	94.2	95.7	95.6	**96**	91.8	95
USPS->MNIST	77.5	90.1	91.5	94.1	96.4	93.2	**98.8**	73.7	97.6
SVHN->MNIST	68.1	76	59.6	96.2	**96.7**		92.5	82	82.7

Curriculum learning

Learning Transferable Representations (LTR), Sener *et al.*, NIPS'16

- jointly optimizing representation, domain transformation and label inference

Associative Domain Adaptation (ADA), Haeusser *et al.*, ICCV'17

- reinforcing label associations between domains in the embedding space

Asymmetric Tri-training (ATriDA) *et al.*, PMLR'17

- leverages three classifiers trained simultaneously with real and pseudo-labels

Mixture of Alignments of Scatter Tensors (MAST), Koniusz *et al.*, CVPR'17

- aligns higher-order scatter statistics between source and target domains

Self-ensembling for visual domain adaptation, (SelfEns) French *et al.*, ICLR'18

- exponential moving average of the student network weights

Similarity Learning Network (SimNet), Pinheiro, CVPR'18

- learns domain-invariant features and categorical prototype representations jointly

	DeepJDOT	CoGAN	DupGAN	SelfEns	SimNet	ADA
USPS->MNIST	96.4	93.2	**98.8**	98.1	95.6	97.6

Office31	CDAN	Adial	LTR
AlexNet	76.9	77.1	**80.4**

To summarize

Discriminative models

- they are easy to train, straightforward and popular approaches
- minimizing joint feature/label distributions is better (DeepJDOT, JAN)
- best is to combine them with adversarial learning (MCDAA, CDAN)
- per layer based adaptation can bring further improvements

Generative and reconstruction models

- mainly tested on classes with relatively low intra-class variation (digits)
- they outperform the discriminative models (except DeepJDOT) on these data
- best GANs (DupGAN) and best reconstruction (DSN) performs on par

Exploiting pseudo labels

- these methods works in general pretty well
- LTR seems to have the highest gain

However !

The above observations are far from being conclusive

- the results come from various papers
- for the same method results may vary (I took the best)
- only few methods were compared on the same datasets
- the methods often used different deep architectures
- the used datasets are small and not challenging enough
- not clear how the parameters of each model was tuned !!

How to tune the parameters of DA models?

Prohibited:

- tuning the model based on the test results
- using at any time target labels, even with cross validation

Possible, but not always optimal:

- tuning based on the results obtained on the source
- using the model with fixed parameters for all the experiments
- consider reverse cross-validation, Ganin *et al.*, JMLR'16

Preferred, but not always obvious:

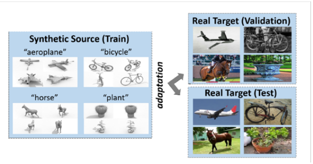
- using measures nor requiring target labels (distribution divergence)
- using a validation domain similar but not the same as target

What we need?

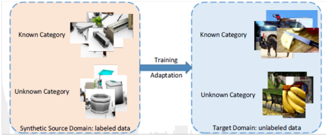
More DA challenges and Leader-boards

- VisDa 2017 and 2018 Challenges, with continuous leaderboard on CodaLab

Visual Classification



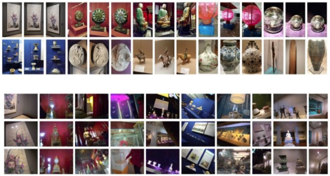
Open Set Classification



More challenging datasets:

- OpenMIC , Koniusz, ECCV'18, CMPlaces , Castrejón, CVPR'16

MIC Open



CMPlaces

Outline

1. Motivation

2. Shallow Domain Adaptation methods

3. DA using Deep Learning

4. Deep Domain Adaptation Methods

5. Beyond image classification

Beyond image classification

Casting the problem as classification

- video concept detection: Yang *et al.*, ICCV'13
- activity recognition: FarajiDavar *et al.*, BMVC'12, Zhu *et al.*, BMVC'13
- 3D pose estimation: Yamada *et al.*, ECCV'12

Data augmentation and synthetic data

- pose estimation, Shotton *et al.*, CVPR'11, Su *et al.*, ICCV'15
- detection: Pepik *et al.*, CVPR'12, Peng *et al.*, ICCV'15
- segmentation: Ros *et al.*, CVPR'16, Satkin *et al.*, BMVC'12
- tracking: Gaidon *et al.*, CVPR'16, Vazquez *et al.*, PAMI'14
- actions recognition, De Souza *et al.*, CVPR'17

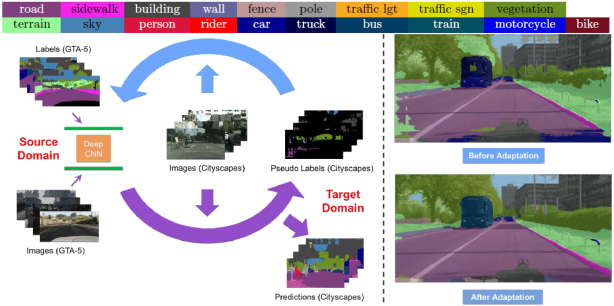
Model adaptation between domains

- adapting or designing deep models for various tasks

Semantic segmentation -1-

Curriculum learning based

- Curriculum domain adaptation, Zhang, *et al.*, ICCV'17
- Class-Balanced Self-Training, Zou, *et al.*, ECCV'17

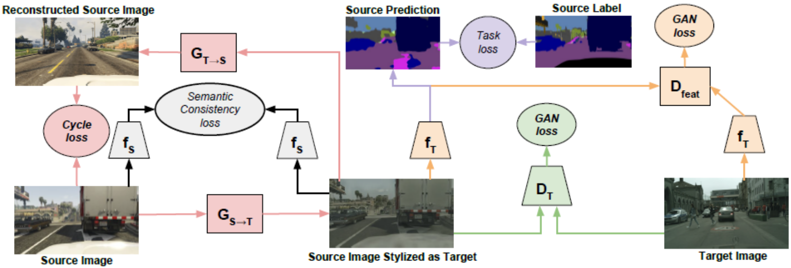


Main idea: start from easier tasks, and then refine relying on predicted labels

Semantic segmentation -2-

GAN based image transformation

- Cycle-consistent adaptation framework (CYCADA), Hoffman *et al.*, CORR'18
- Learning from Synthetic Data, Sankaranarayanan *et al.*, CVPR'18
- Representation Adaptation Networks (RAN), Zhang *et al.*, CORR'18

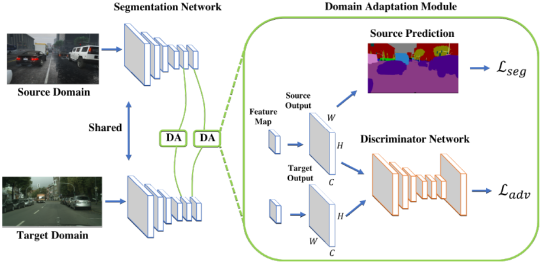


Main idea: style transfer in general from synthetic to real images

Semantic segmentation -3-

GAN integrated into the segmentation framework

- FCN in wild, Hoffman *et al.*, CORR'17
- Adapt Structured Output, Tsai *et al.*, CORR'18
- Conservative Loss, Zhu *et al.*, CORR'18
- Semi-Supervised Semantic Segmentation, Hung *et al.*, CORR'18
- Conditional Generative Adversarial Network, Hong *et al.*, CVPR'18



Main idea: forcing target features to resemble to source features

Object detection

CNNs pre-trained with image level annotations

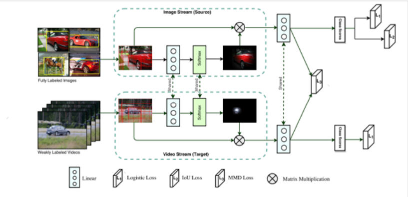
- combined with region proposals, Oquab *et al.*, CVPR'14, Girshick *et al.*, CVPR'14
- learns to transform the classifier into object detector, Hoffman *et al.*, NIPS'14

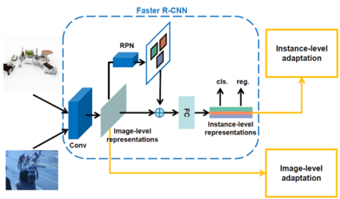
Style-transfered with CycleGAN

- Fine-tune the model on transformed images, Inoue *et al.*, CVPR'18

Adapting the CNN based object detectors

- Align R-CNN features with SA, Raj *et al.*, BMVC'15
- Minimize MMD between image level features features, Chanda *et al.*, BMVC'17
- Gradient reverse layer both at image and at instance level, Chen *et al.*, CVPR'18





And several other tasks

Person Re-ID

- Person Transfer GAN, Wei, *et al.*, CVPR'18
- Pose Transferrable Person Re-Identification, Liu *et al.*, CVPR'18
- Camera Style Adaptation, Zhong*et al.*, CVPR'18
- Learning from rendered 3D humans, Bak *et al.*, ECCV'18

Action recognition

- 3D Body Skeletons via Kernel Feature Maps, Yusuf and Koniusz, BMVC'18

Depth estimation

- AdaDepth, Kundu *et al.*, CVPR'18

3D keypoint estimation

- Regression and view consistency loss, Zhou *et al.*, ECCV'18

Autonomous vehicle control command

- Real-to-Virtual Domain Unification, Yang, *et al.*, ECCV'18

Recent Book on Domain Adaptation

© 2017

Domain Adaptation in Computer Vision Applications

Editors: **Csurka**, Gabriela (Ed.)

- **Introductory part**
  - a comprehensive survey and a deeper look at dataset bias
- **Part I: Shallow Domain Adaptation Methods**
  - GFK, SA, TCA, DME, ATTM, MSDA
- **Part II: Deep Domain Adaptation Methods**
  - deepCoral, DANN, Deep Transfer Across Domains and Tasks
- **Part II: Beyond Image Classification**
  - Segmentation, object and object part detection, re-identification
- **Beyond Domain Adaptation: Unifying Perspectives**
  - domain generalization, multi-domain multi-task learning