Transfer Learning: Theory, Algorithms and Applications —A Tutorial

Lei Zhang

Chongqing University

Chongqing, China

http://www.leizhang.tk

2021.02.01

重慶大學 CHONGQING UNIVERSITY

LiVE Group 视觉智能与学习研究中心

Contents

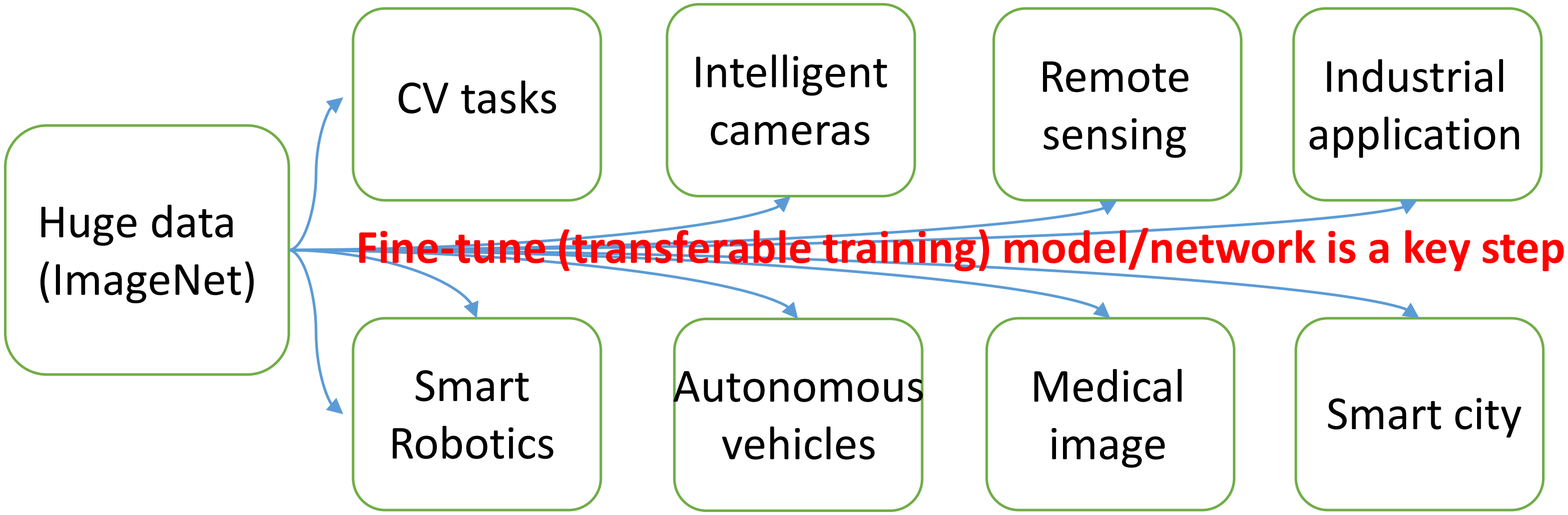
- Part I: Background and Preliminary
- Part II: Concept, Theory and Algorithms
- Part III: Applications of TL/DA Algorithms
- Summary

Contents

- Part I: Background and Preliminary
- Part II: Concept, Theory and Algorithms
- Part III: Applications of TL Algorithms
- Summary

Fine-tune is all you need

- Transfer learning has been a widely used technique in a wide spread of applications.
- In deep learning era, you may hear from about the “fine-tune” technique for down-stream tasks.



Fine-tune (transferable training) model/network is a key step

A Revisit of Machine Learning

- Machine learning is a modeling technique with statistics for parameters estimation of unknown fun.
- To be simple, given a dataset (X, y) with label y, a statistical learning model is to find a mapping f(.) between X and y, such that

$$y=f(x)$$

- A learning problem to be solved is how to find f(.)?
- Many learning techniques from shallow to deep.
- Gradient descent based techniques.

A Revisit of Machine Learning

- To find a feasible (optimal) mapping (solution) f(.), machine learning is transformed to an *optimization* technique.
- A general optimization (*minimization*) problem of learning is,

$$R[\mathrm{Pr}, \theta, l(x, y, \theta)] = \mathbf{E}_{(x,y)\sim \mathrm{Pr}}\left[l(x, y, \theta)\right]$$

- R[.] is the *expected* risk defined by the loss function with input (X,y) sampled from a probabilistic distribution *Pr* and parameter θ of f(.)
- Pr should be an independent identical distribution (i.i.d.)

A Revisit of Machine Learning

- However, due to the infinite space of the data distribution, we can only have a subset of the data (training data).
- So, the expected risk minimization is transformed into an *empirical* risk optimization problem,

$$R[\mathrm{Pr}, \theta, l(x, y, \theta)] = \mathbf{E}_{(x,y)\sim \mathrm{Pr}} \left[l(x, y, \theta) \right]$$

$$R_{\mathrm{emp}}[Z, \theta, l(x, y, \theta)] = \frac{1}{m} \sum_{i=1}^{m} l(x_i, y_i, \theta)$$

- *m* is the size (number) of the finite training subset sampled from the distribution Pr.

A Revisit of Machine Learning

- Generally, by only optimizing the empirical risk, we could not obtain a friendly solution. Overfitting on the training subset often happens.
- So, *regularization* technique is commonly used in the empirical risk optimization problem,

$$R_{\text{reg}}[Z, \theta, l(x, y, \theta)] := R_{\text{emp}}[Z, \theta, l(x, y, \theta)] + \boxed{\lambda\Omega[\theta]}$$

- $\Omega[\theta]$ is the regularization on model parameters.
- Regularization plays a vital role in ML fields.

A Revisit of Machine Learning

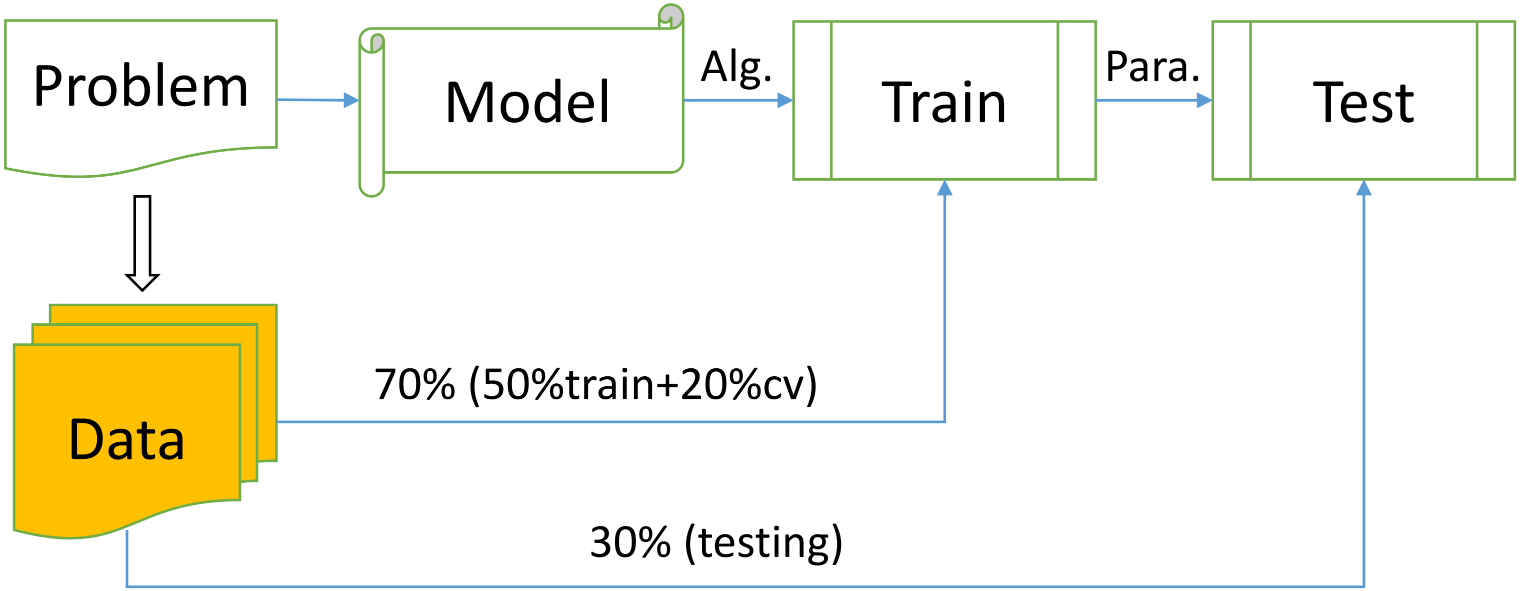
- Generalization is the final objective of ML task.
- The optimized parameter θ of the mapping function $f(.)$ on a training subset sampled from Pr should have *generalization* ability on a test subset sampled from an *i.i.d.* distribution Pr'.
- The expected risk of a test subset is estimated by

$$R[\mathrm{Pr}', \theta, l(x, y, \theta)] = \mathbf{E}_{(x,y)\sim \mathrm{Pr}'}\left[l(x, y, \theta)\right]$$

- θ is the solved parameters with training subset.

A Revisit of Machine Learning

- Okay, now we can have a view of a general machine learning framework with problem definition, data collection, model selection and evaluation protocol.



A Revisit of Machine Learning

- So, anyone can easily deploy a machine learning task, finish your project and enjoy your life.

- Really?

- Machine learning modeling should also have some conditions.

So angry but keep smiling

Label is all you need

- For learning a classifier/predictor based on (X, y), you should first have label y.
- Actually, data collection is sometimes expensive, but label is more expensive and needs cost-ineffective manual power.
- An idea is to "borrow" the sufficiently labeled data from another domain.
- Chinese idioms :"他山之石，可以攻玉"--《诗经》

Label problem is solved, so is it now okay? No!

Probably Approximate Correct(PAC)

- PAC theory is an important basis of statistical ML.
- PAC refers to three basic problems,
- 1) **Sample complexity**: learning a *hypothesis h* needs a *reasonable* number of samples;
- 2) **Computational complexity**: learning a *hypothesis h* needs a *reasonable* computation complexity;
- 3) **Learning reliability**: the hypothesis h has a *low error rate* (empirical risk) on training set **S**, and a *high success rate* on a random test sample **x**.

Probably Approximate Correct(PAC)

- Error rate on training set (can be accurately calculated)

$$Error_{\mathbf{S}}(h) = \frac{1}{n}\sum\nolimits_{x\in\mathbf{S}} \delta(h(x) \neq GT)$$

- Failure rate on a random test sample (to be estimated)

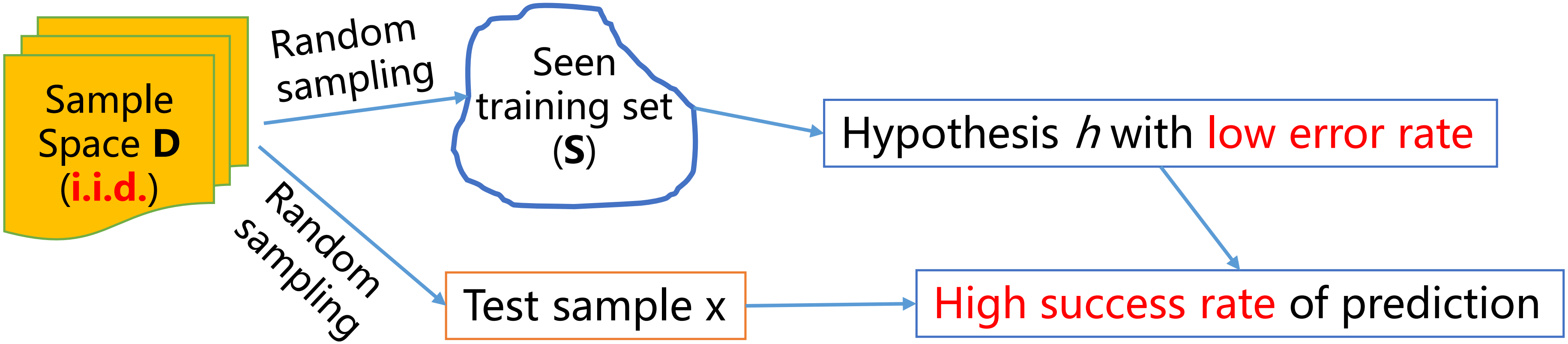
$$Error_{\mathbf{D}}(h) = Pr_{x\in\mathbf{D}}(h(x) \neq GT)$$

- **Definition of PAC**:

A problem can be learnable if and only if the learner can output a hypothesis with *arbitrary low error rate* in *arbitrary high probability*, by using a *reasonable number of data* and *reasonable computation complexity*.

Probably Approximate Correct(PAC)

- A figure to describe PAC


Sample Space **D** (**i.i.d.**)
Random sampling
Seen training set (**S**)
Hypothesis *h* with low error rate
Random sampling
Test sample x
High success rate of prediction

- A prior assumption is the i.i.d. condition.
- The training set and test sample should be sampled from an independent identical distribution (i.i.d.)

A Preliminary of Transfer Learning

Problem definition:

Given a target task (D_T) without labels (few labels), for learning a reliable predictor/classifier on domain D_T,

Not feasible?

- A sufficiently labeled, semantic related but distribution different source task (D_S) is leveraged as auxiliary training data.
- Two key points:

1) Overcomes the label deficiency problem;

2) But introduces non i.i.d. problem between D_T and D_S

A Preliminary of Transfer Learning

Differences from semi-supervised learning (i.i.d.)

- Marginal distribution $P(X_l) = P(X_u)$
- Label space $P(Y_l|X_l) = P(Y_u|X_u)$

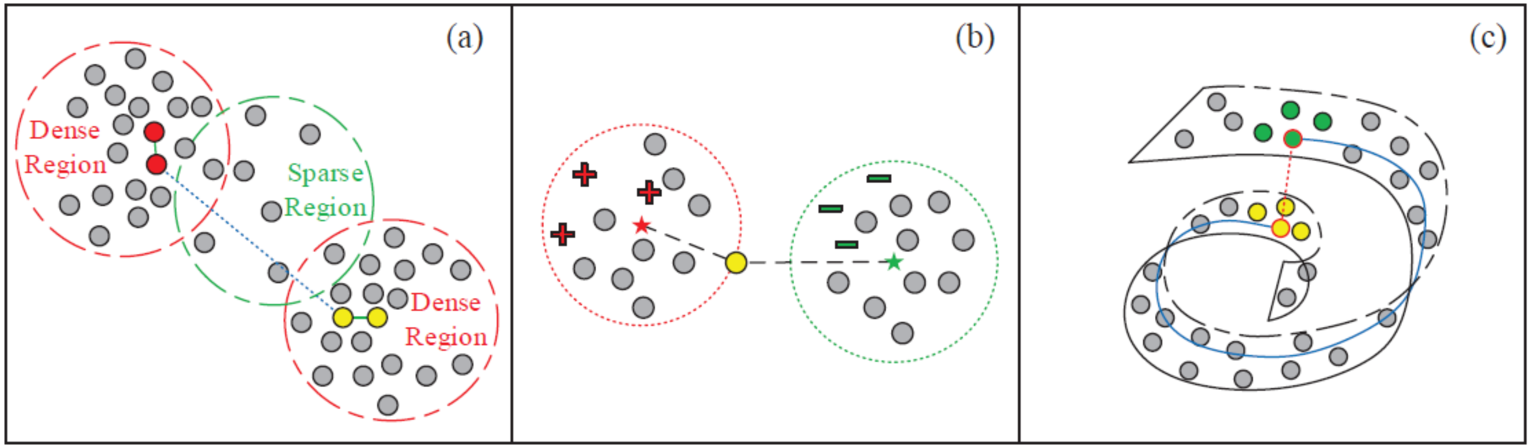
1) **Smooth assumption**: data is distributed with different density and two samples in high density have same labels;

2) **Cluster assumption**: data has inherent cluster structure and two samples in the same cluster have same labels;

3) **Manifold assumption**: data has a low-dimensional manifold and two samples in local neighbor have same labels.

A Preliminary of Transfer Learning

Differences from semi-supervised learning (i.i.d.)

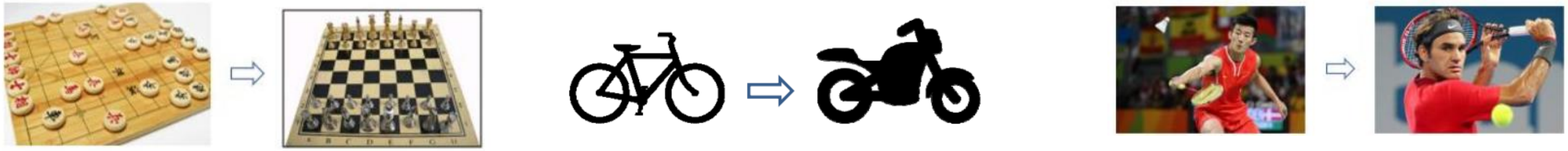


(a) ***Smoothness*** assumption (b) ***Cluster*** assumption (c) ***Manifold*** assumption

A Preliminary of Transfer Learning

Toy Examples:

Semantic related but distribution different tasks



Behavior learning skills (domain common knowledge)

Computer Vision
(image classification)

Natural Language Processing
(translation)

Text Recognition

A Preliminary of Transfer Learning

Mission and Objective:

Transfer learning is solving a class of uncommon machine learning problems, i.e. label deficiency and probability distribution discrepancy.

Revisit the expected risk of test data:

$$
\begin{aligned}
R[\mathrm{Pr}', \theta, l(x, y, \theta)] &= \mathbf{E}_{(x,y)\sim \mathrm{Pr}'}\left[l(x, y, \theta)\right] \\
&= \mathbf{E}_{(x,y)\sim \mathrm{Pr}}\left[\underbrace{\frac{\mathrm{Pr}'(x,y)}{\mathrm{Pr}(x,y)}} l(x, y, \theta)\right]
\end{aligned}
$$

If Pr=Pr’ (i.i.d. for traditional ML),

$$
\mathbf{E}_{(x,y)\sim \mathrm{Pr}}\left[l(x, y, \theta)\right] = \mathbf{E}_{(x,y)\sim \mathrm{Pr}'}\left[l(x, y, \theta)\right]
$$

Else, the trained model is not transferable to test.

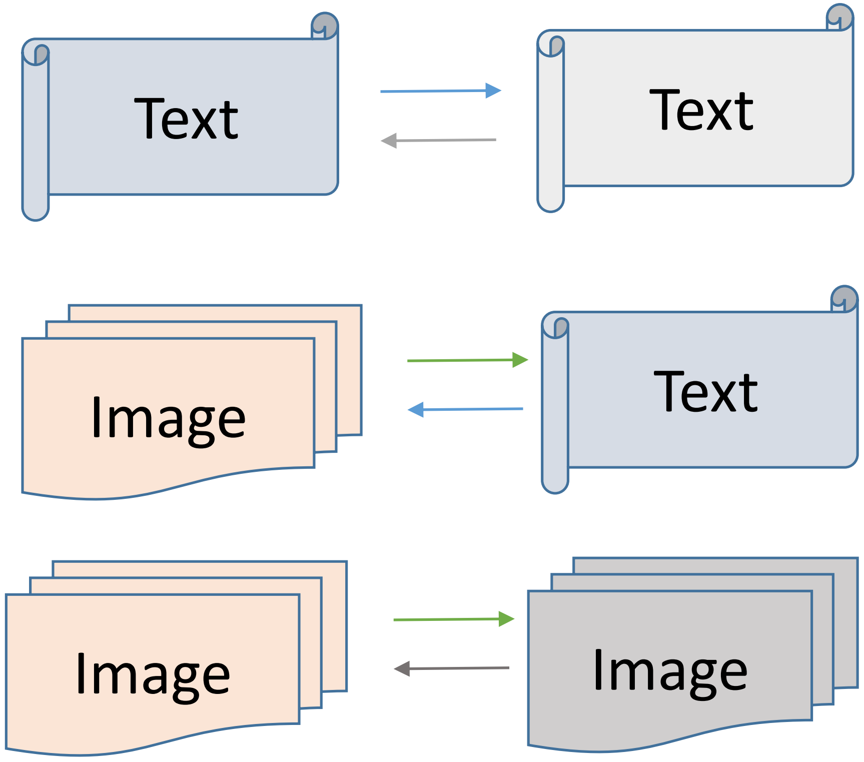
A Preliminary of Transfer Learning

Scenarios of non i.i.d.:

Data of Heterogeneity
(language, blur, etc.)

Data of Heterogeneity
(Media, modality)

Data of Heterogeneity
(background, viewpoint, pose
, modality, etc.)



A Preliminary of Transfer Learning

Weak Learning:

The concept of "weak learning" originates from the era of Boosting and AdaBoost (30 years ago).

Amazingly, the past "weak learning" is equivalent to "strong learning". In a word,

"A problem can be weak-learned if and only if it can be strong-learned."

Currently, the weak learning is really a weak problem rather than a strong problem.

A Preliminary of Transfer Learning

Weak Learning:

1. Weakly-supervised learning (Zhihua Zhou, 2018)-incomplete, inexact, inaccurate of labels

2. Transfer learning (Pratt L.Y., 1991; Qiang Yang, 2010)

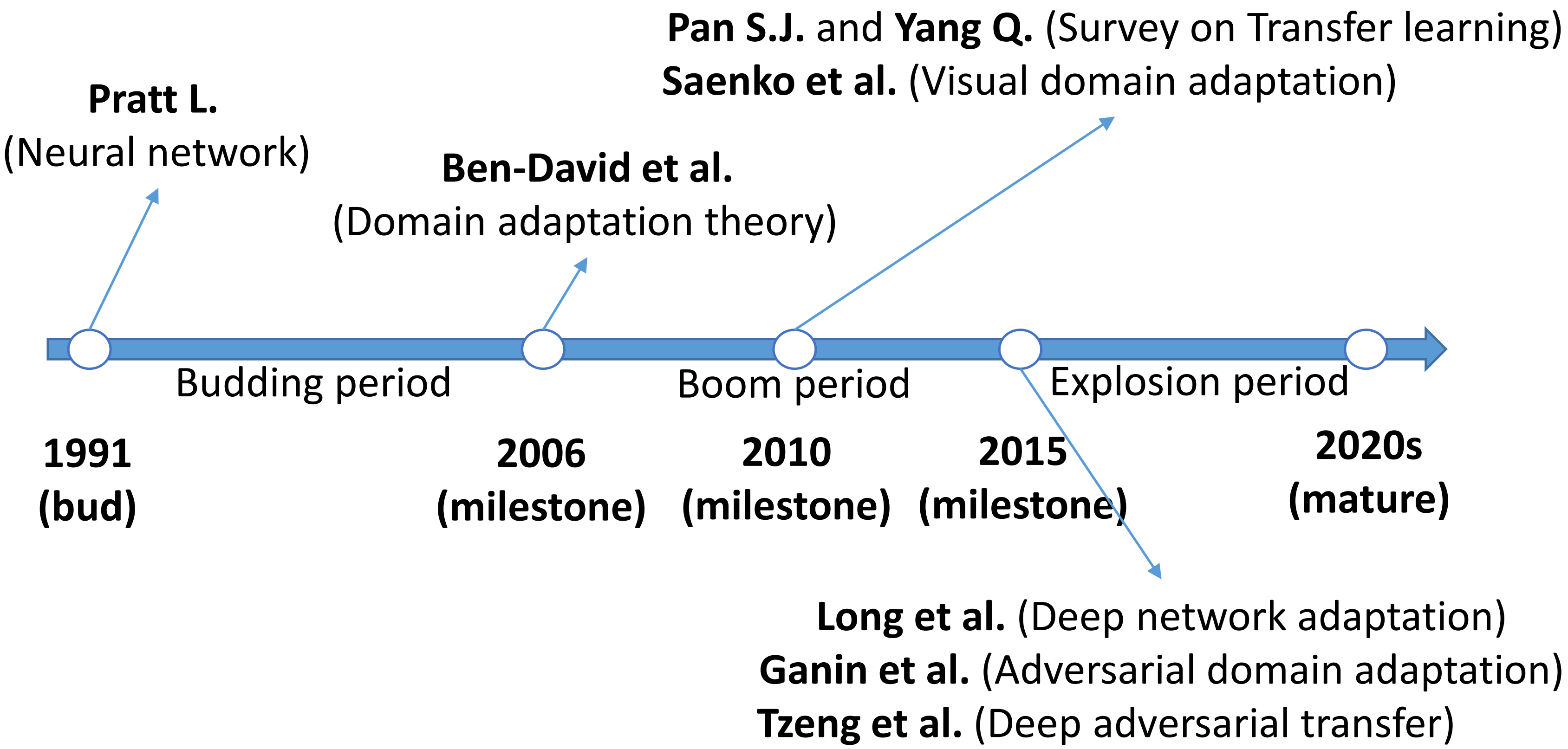
3. Domain adaptation (Shai Ben-David, 2006)

p.s. Transfer learning and domain adaptation hold the same perspective for common knowledge learning between different domains.

In this tutorial, alternated usage of both names (TL vs. DA) frequently happens.

A Preliminary of Transfer Learning

History of Transfer Learning (1990s-2020s):



Pratt L.
(Neural network)

Ben-David et al.
(Domain adaptation theory)

Pan S.J. and **Yang Q.** (Survey on Transfer learning)
Saenko et al. (Visual domain adaptation)

Budding period

Boom period

Explosion period

1991
(bud)

2006
(milestone)

2010
(milestone)

2015
(milestone)

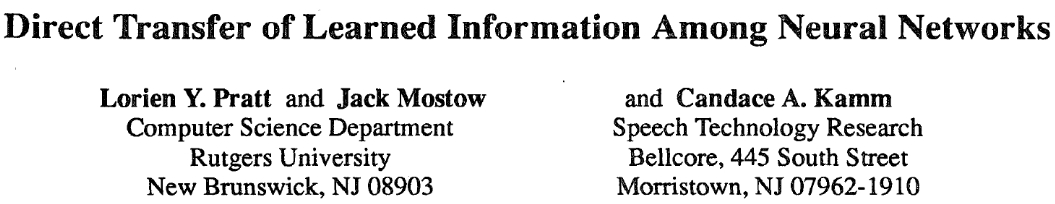
2020s
(mature)

Long et al. (Deep network adaptation)
Ganin et al. (Adversarial domain adaptation)
Tzeng et al. (Deep adversarial transfer)

A Preliminary of Transfer Learning

History of Transfer Learning (1991-1993, bud):

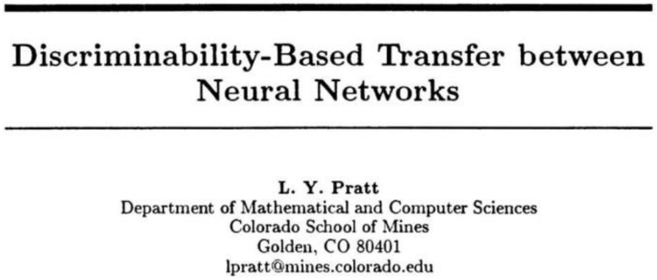
Originally, the "transfer" concept was proposed by L.Y. Pratt in 1991 (AAAI) and 1993 (NIPS) between neural networks.



Direct Transfer of Learned Information Among Neural Networks

Lorien Y. Pratt and **Jack Mostow**
Computer Science Department
Rutgers University
New Brunswick, NJ 08903

and **Candace A. Kamm**
Speech Technology Research
Bellcore, 445 South Street
Morristown, NJ 07962-1910



Discriminability-Based Transfer between Neural Networks

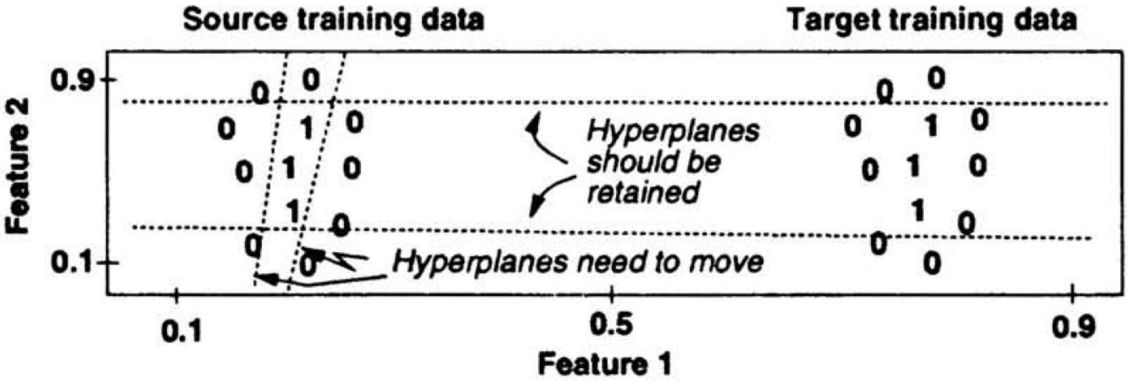
L. Y. Pratt
Department of Mathematical and Computer Sciences
Colorado School of Mines
Golden, CO 80401
lpratt@mines.colorado.edu

- L. Pratt, J. Mostow, and C. Kamm, Direct transfer of learned information among neural networks, *AAAI*, 1991.
- L. Pratt, "Discriminability-based transfer between neural networks," in *NIPS*, 1993.

A Preliminary of Transfer Learning

History of Transfer Learning (1991-1993, bud):

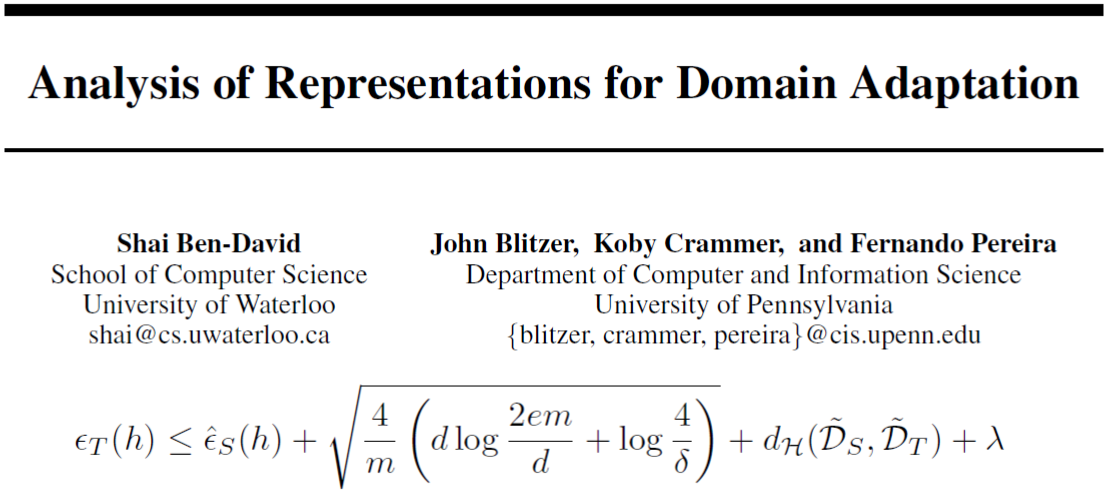
- L. Pratt, J. Mostow, and C. Kamm, Direct transfer of learned information among neural networks, *AAAI*, 1991.
- Motivation: **"how to use information from one neural network to help a second network learn a related task"**.
- Focus: **"learning on a target problem is sped up by using the weights obtained from a network trained for a related source task"**
- L. Pratt, "Discriminability-based transfer between neural networks," in *NIPS*, 1993.



A Preliminary of Transfer Learning

History of Transfer Learning (2006-2015, milestone):

- **15 Years later**, **in 2006**, Shai Ben-David from University of Waterloo, published one paper in domain adaptation theory in NIPS 2006, and theoretically prove the expected error upper bound of target domain.



Analysis of Representations for Domain Adaptation

Shai Ben-David
School of Computer Science
University of Waterloo
shai@cs.uwaterloo.ca

John Blitzer, Koby Crammer, and Fernando Pereira
Department of Computer and Information Science
University of Pennsylvania
{blitzer, crammer, pereira}@cis.upenn.edu

$$\epsilon_T(h) \leq \hat{\epsilon}_S(h) + \sqrt{\frac{4}{m}\left(d \log \frac{2em}{d} + \log \frac{4}{\delta}\right)} + d_{\mathcal{H}}(\tilde{\mathcal{D}}_S, \tilde{\mathcal{D}}_T) + \lambda$$

Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2006). Analysis of representations for domain adaptation. In: *Advances in neural information processing systems*

A Preliminary of Transfer Learning

History of Transfer Learning (2006-2015, milestone):

- in 2010, Qiang Yang from Hong Kong University of Science and Technology, published the first survey on **transfer learning**.
- in 2010, Kate Saenko from UC Berkeley published the first paper on **domain adaptation**, in ECCV, a top computer vision conference.
- from 2010-2015, a number of papers on transfer learning and domain adaptation were published.
- In this period, a number of classical TL/DA models and algorithms in classifiers and features are emerged.

S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Trans. Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, 2010.
K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting visual category models to new domains,” in ECCV, 2010.

A Preliminary of Transfer Learning

History of Transfer Learning (2015-now, explosion):

Deep transfer learning and deep domain adaptation era.

- in 2012, Bengio Y. published one paper on deep learning for transfer learning, in JMLR
- in 2014, Donahue et al. proposes "**fine-tune**" transfer strategy from a pre-trained convolutional neural network and published in ICML 2014.
- *Fine-tune* has become a generic transfer learning strategy in many applications, such as medical image, remote sensing image, etc.

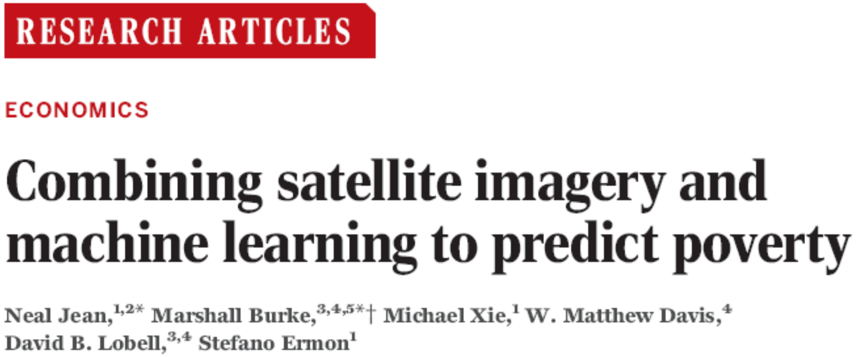
Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," JMLR, vol. 27, pp. 17–37, 2012.
J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in ICML, 2014.

A Preliminary of Transfer Learning

History of Transfer Learning (2015-now, explosion):

Fine-tune based deep transfer learning application:

ImageNet



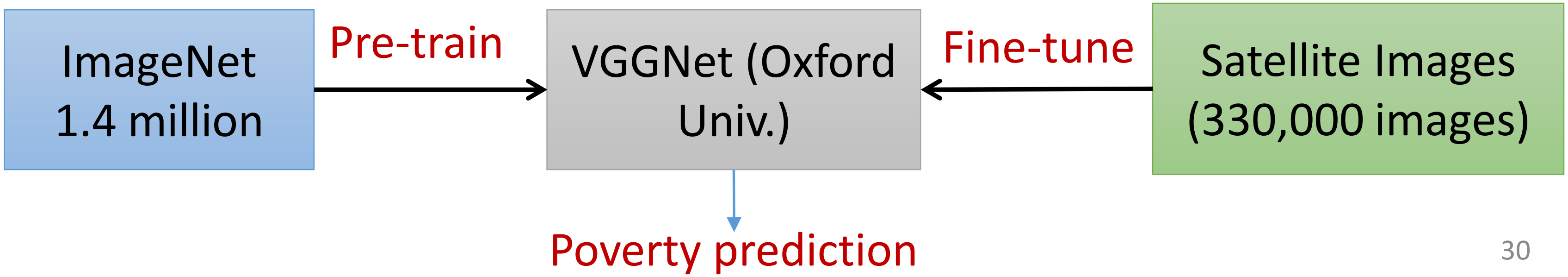
RESEARCH ARTICLES

ECONOMICS

Combining satellite imagery and machine learning to predict poverty

Neal Jean,[1,2*] Marshall Burke,[3,4,5*†] Michael Xie,[1] W. Matthew Davis,[4] David B. Lobell,[3,4] Stefano Ermon[1]

Science 2017, Stanford Univ.



A Preliminary of Transfer Learning

History of Transfer Learning (2015-now, explosion):

Deep convolutional network adaptation for transferable representation.

- in 2015, Long et al. firstly published one paper on **deep network adaptation** based on MMD optimization.
- in 2015, Ganin et al. firstly proposed **adversarial domain adaptation** by using a gradient reversal layer for minimax optimization.
- in 2015, Tzeng et al. proposed **deep adversarial transfer** by solving a minimax gaming optimization as GAN.

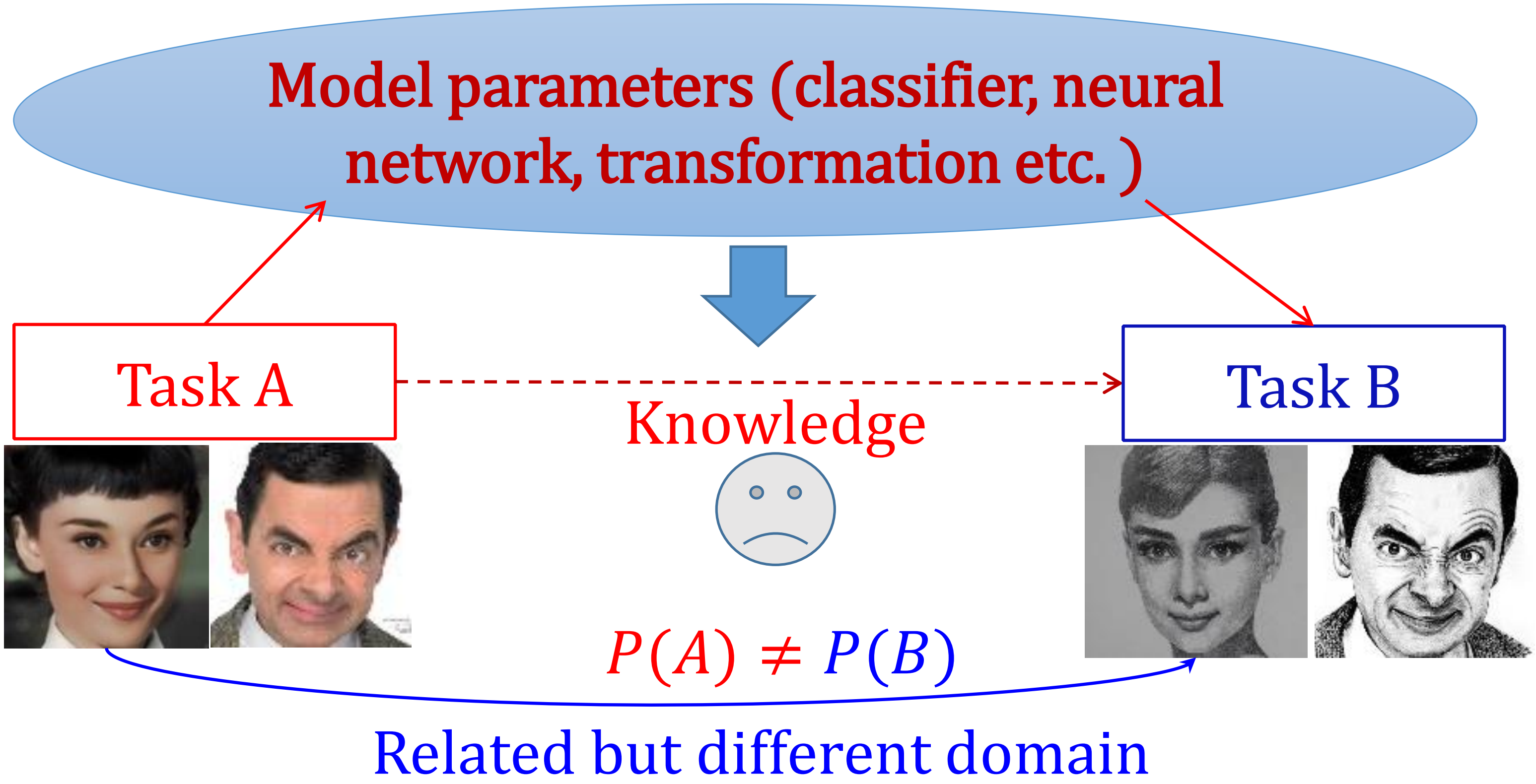
M. Long, Y. Cao, J. Wang, and M. I. Jordan, “Learning transferable features with deep adaptation networks,” in ICML, 2015.
Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in arXiv, 2015.
E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, “Simultaneous deep transfer across domains and tasks,” ICCV, vol. 30, no. 31, pp. 4068–4076, 2015.

Contents

- Part I: Background and Preliminary
- Part II: Concept, Theory and Algorithms
- Part III: Applications of TL/DA Algorithms
- Summary

Concept

What is transfer learning (cross-modal face recog.)?



Model parameters (classifier, neural network, transformation etc.)

Task A

Knowledge

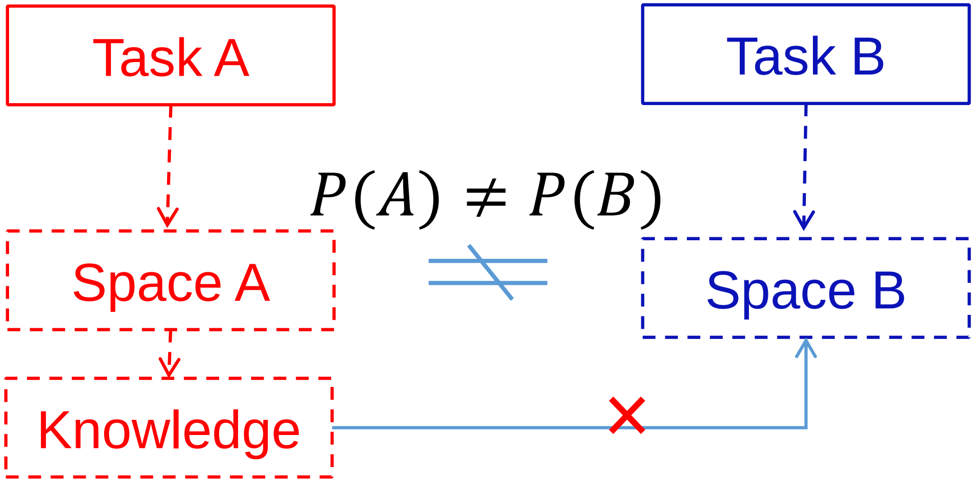
Task B

$P(A) \neq P(B)$

Related but different domain

Concept

What is transfer learning (handwritten digits recog.)?



Task A

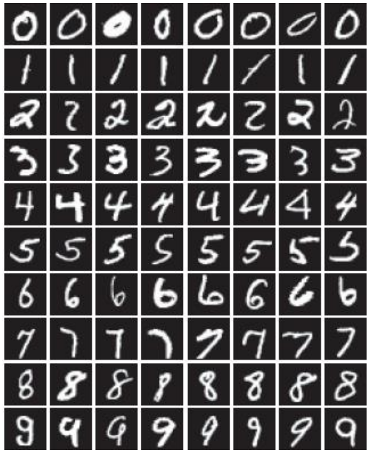
Space A

Knowledge

$P(A) \neq P(B)$

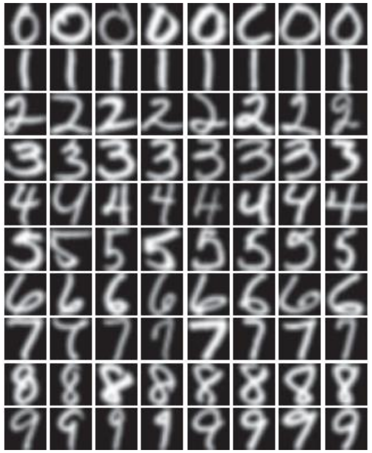
Task B

Space B



Task A

(MNIST)

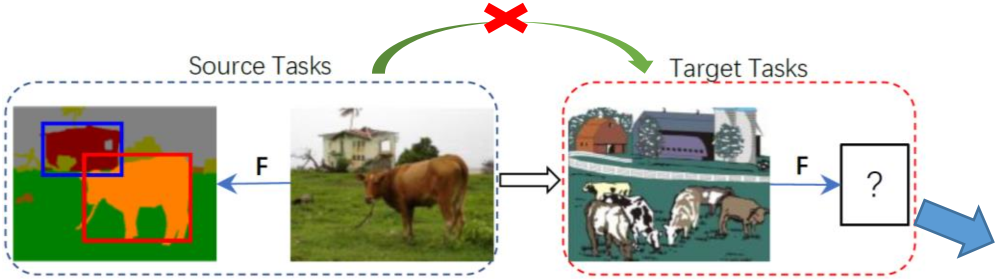


Task B

(USPS)

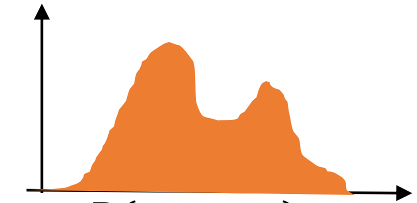
Concept

What is transfer learning (computer vision)?



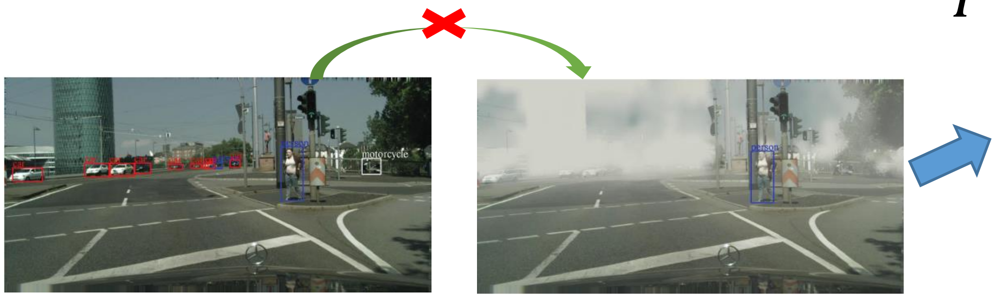
Object detection, segmentation and classification

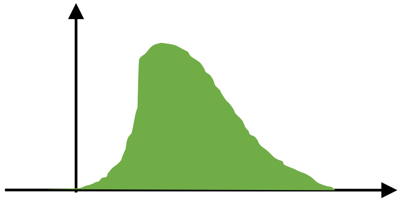
(domain shift)



$P(source)$

$$P(source) \neq P(target)$$





$P(target)$

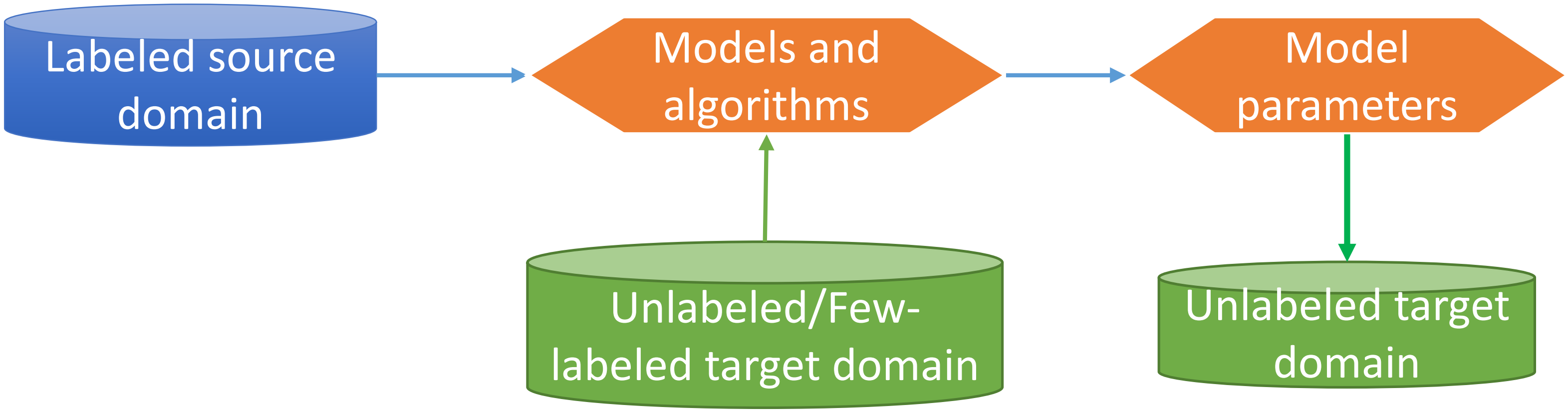
Visual perception in foggy weather

(domain shift)

Concept

What is transfer learning?

Transfer learning or domain adaptation is leveraging a sufficiently labeled, distribution different but semantic related source domain for training and recognizing target domain samples.



Theory

Why are transfer learning models or algorithms effective and reliable?

In other words, how to guarantee the models or algorithms to have low generalization error on target data?

Ben-David Shai et al. induced a generalization bound of domain adaptation, which is widely used as a theoretical guidance for models and algorithms.

Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2006). Analysis of representations for domain adaptation. In: *Advances in neural information processing systems*

Theory

Shai Ben-David's generalization bound theorem:

- To be simple, the expected target error $\epsilon_T(h)$ is bounded as (proof based on triangular inequality is removed)

$$\epsilon_T(h) \leq \hat{\epsilon}_S(h) + \sqrt{\frac{4}{m}\left(d \log \frac{2em}{d} + \log \frac{4}{\delta}\right)} + d_{\mathcal{H}}(\tilde{\mathcal{D}}_S, \tilde{\mathcal{D}}_T) + \lambda$$

- $\mathcal{H}$ is the set of hypothesis.
- The upper bound of $\epsilon_T(h)$ consists of four terms.

Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2006). Analysis of representations for domain adaptation. In: *Advances in neural information processing systems*

Theory

Shai Ben-David's generalization bound theorem:

- $\hat{\epsilon}_S(h)$ is the source error, $d_{\mathcal{H}}\left(\widetilde{\mathcal{D}}_S, \widetilde{\mathcal{D}}_T\right)$ is the $\mathcal{H}$ −divergence and λ is the combined error of an ideal hypothesis h^*.

1)
$$\begin{aligned} \epsilon_S(h) &= \mathrm{E}_{\mathbf{z}\sim\tilde{\mathcal{D}}_S}\left[\mathrm{E}_{y\sim\tilde{f}(\mathbf{z})}\left[y \neq h(\mathbf{z})\right]\right] \\ &= \mathrm{E}_{\mathbf{z}\sim\tilde{\mathcal{D}}_S}\left|\tilde{f}(\mathbf{z}) - h(\mathbf{z})\right| . \end{aligned}$$

2)
$$d_{\mathcal{H}}(\mathcal{S},\mathcal{T}) = 2\left(1 - \min_{h\in\mathcal{H}}\left(err_{\mathcal{S}}(h(\mathbf{x})) + err_{\mathcal{T}}(h(\mathbf{x}))\right)\right) = 2\left(1 - 2\min_{h\in\mathcal{H}} err\left(h(x)\right)\right)$$

3)
$$\lambda = \min\nolimits_{h\in\mathcal{H}} \epsilon_S(h) + \epsilon_T(h)$$

Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2006). Analysis of representations for domain adaptation. In: *Advances in neural information processing systems*

Theory

From $\mathcal{H}$-divergence to $\mathcal{H}\Delta\mathcal{H}$-divergence

- For a hypothesis space $\mathcal{H}$, the *symmetric difference hypothesis space* $\mathcal{H}\Delta\mathcal{H}$ is defined.

$$g \in \mathcal{H}\Delta\mathcal{H} \quad \Longleftrightarrow \quad g(\mathbf{x}) = h(\mathbf{x}) \oplus h'(\mathbf{x}) \quad \text{for some } h, h' \in \mathcal{H}$$

- where $\oplus$ is the XOR function. In words, every hypothesis g ∈ $\mathcal{H}\Delta\mathcal{H}$ is the set of disagreements between two hypotheses h, h' in $\mathcal{H}$,

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) &= 2 \sup_{h,h' \in \mathcal{H}} \left| \Pr_{x \sim \mathcal{D}_S}\left[h(x) \neq h'(x)\right] - \Pr_{x \sim \mathcal{D}_T}\left[h(x) \neq h'(x)\right] \right| \\ &= 2 \sup_{\eta \in \mathcal{H}\Delta\mathcal{H}} \left| Pr_{\tilde{D}_S}[z : \eta(z) = 1] - Pr_{\tilde{D}_T}[z : \eta(z) = 1] \right| \end{aligned}$$

Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2010). A theory of learning from different domains. In: *Machine Learning, 79, 151-175.*

Theory

Shai Ben-David's generalization bound theorem:

$$\epsilon_T(h) \le \epsilon_S(h) + \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + \lambda$$

Essence of domain adaptation:

- A minimax problem

maximization

View 1: $$\min d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) = \boxed{2 \sup_{\eta \in \mathcal{H}\Delta\mathcal{H}} \left| Pr_{\tilde{\mathcal{D}}_S}[z : \eta(z) = 1] - Pr_{\tilde{\mathcal{D}}_T}[z : \eta(z) = 1] \right|}$$

View 2: $$\min_f d_{\mathcal{A}}(\mathcal{S}, \mathcal{T}) \Leftrightarrow \min_f 2\left(1 - 2\min_{h \in \mathcal{H}} err\,(h(x))\right) \Leftrightarrow \max_f \min_{h \in \mathcal{H}} err\,(h(x))$$

Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2010). A theory of learning from different domains. In: *Machine Learning, 79, 151-175.*

Theory

Question:

Does Shai Ben-David’s domain adaptation theory really guarantee the success of transfer learning?

Not always! It is conditional.

Theory

When does transfer learning not work?

- Theorem 1: Necessity of small $d_{\mathcal{H}}\left(\widetilde{\mathcal{D}}_S, \widetilde{\mathcal{D}}_T\right)$.
- Theorem 2: Necessity of small λ (combined error).

If and only if both theorem 1 and theorem 2 meet at the same time, otherwise, transfer learning does not work.

In words, the domain discrepancy should be small.

Ben-David, S., Luu T., Lu T. and Pal D. (2010). Impossibility Theorems for Domain Adaptation. In: AISTATS.

Distribution Difference Measure

Distribution alignment is the key part of transfer learning.

How to measure distribution difference between two distributions *P* and *Q*? Some typical statistics.

- MMD (Maximum Mean Discrepancy) (Gretton et al. NIPS’06, NIPS’09, JMLR’12)
- HSIC (Hilbert Schmidt Independence Criterion) (Gretton et al. ALT’05; Yan et al. TCYB’17, Wang et al. ICCV’17, CRTL)
- Bregman divergence (Si et al. TKDE’10, TSL)
- Moment statistics (Herath et al. CVPR’17, ILS; Sun et al. arXiv’17, CORAL; Peng et al. ICCV’19)

Distribution Difference Measure

Maximum Mean Discrepancy (MMD)

- Gretton et al. NIPS'06, NIPS'09, JMLR'12 from MPI, Germany proposed MMD. A non-parametric statistic for testing whether two distributions are different.
- By using smooth functions "Rich" and "Restrictive".

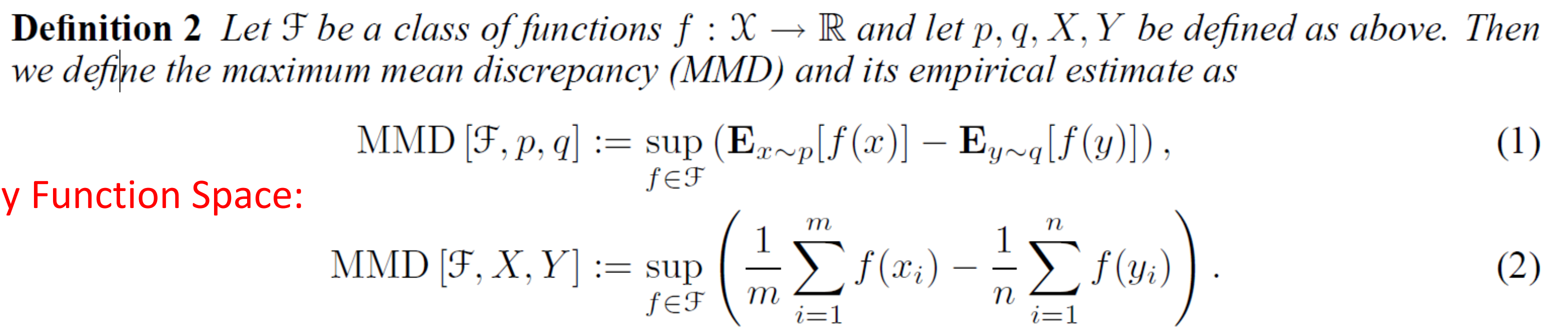
1. MMD(p,q) vanishes if and only if p=q.

2. MMD empirical estimation can easily converge to its expectation.

- In MMD, the unit balls in universal reproducing kernel Hilbert space are used as smooth functions.
- Gaussian and Laplacian kernels are proved to be universal.

Distribution Difference Measure

Maximum Mean Discrepancy (MMD)

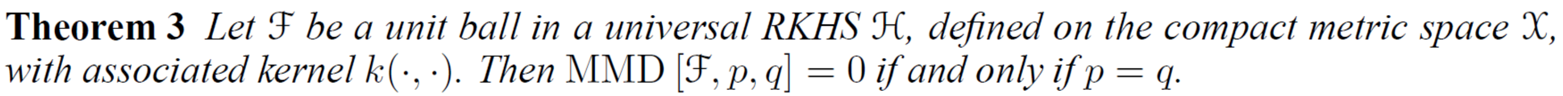


Definition 2 *Let $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to \mathbb{R}$ and let p, q, X, Y be defined as above. Then we define the maximum mean discrepancy (MMD) and its empirical estimate as*

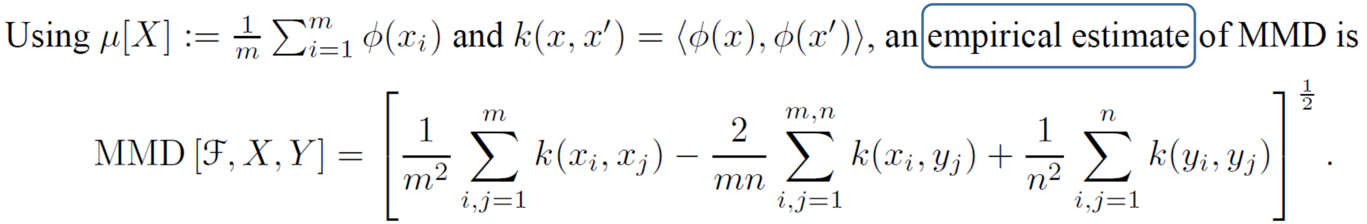
Arbitrary Function Space:

$$\mathrm{MMD}\left[\mathcal{F}, p, q\right] := \sup_{f \in \mathcal{F}} \left(\mathbf{E}_{x \sim p}[f(x)] - \mathbf{E}_{y \sim q}[f(y)]\right), \tag{1}$$

$$\mathrm{MMD}\left[\mathcal{F}, X, Y\right] := \sup_{f \in \mathcal{F}} \left(\frac{1}{m}\sum_{i=1}^{m} f(x_i) - \frac{1}{n}\sum_{i=1}^{n} f(y_i)\right). \tag{2}$$



Theorem 3 *Let $\mathcal{F}$ be a unit ball in a universal RKHS $\mathcal{H}$, defined on the compact metric space $\mathcal{X}$, with associated kernel $k(\cdot,\cdot)$. Then* $\mathrm{MMD}\left[\mathcal{F}, p, q\right] = 0$ *if and only if* $p = q$.



Using $\mu[X] := \frac{1}{m}\sum_{i=1}^{m}\phi(x_i)$ and $k(x, x') = \langle\phi(x), \phi(x')\rangle$, an empirical estimate of MMD is

RKHS:

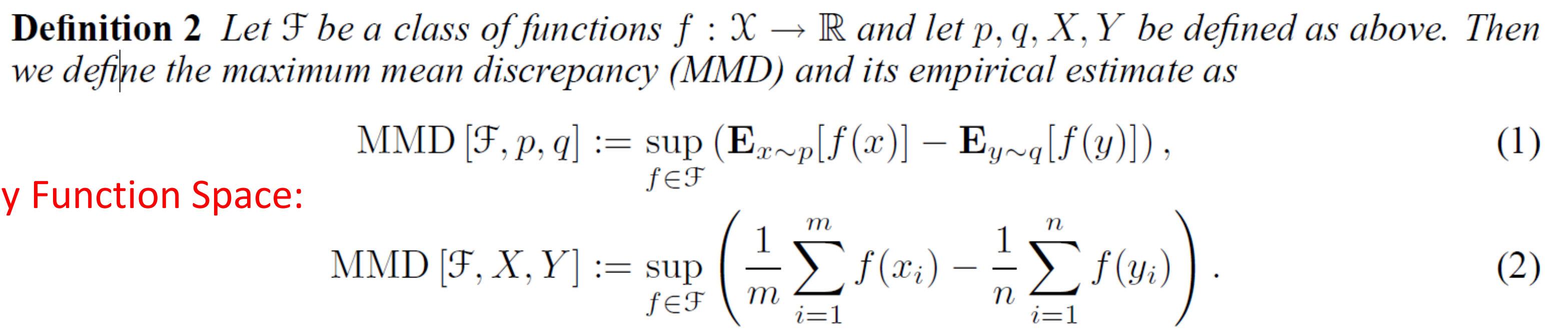
$$\mathrm{MMD}\left[\mathcal{F}, X, Y\right] = \left[\frac{1}{m^2}\sum_{i,j=1}^{m} k(x_i, x_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k(x_i, y_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(y_i, y_j)\right]^{\frac{1}{2}}.$$

- **Kernel is helping us to simplify the computation in infinite dimensional space**
- **To be simple, MMD is the upper bound of the domain mean discrepancy**

http://www.gatsby.ucl.ac.uk/~gretton/mmd/mmd.htm

Distribution Difference Measure

Maximum Mean Discrepancy (MMD)



Definition 2 *Let $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to \mathbb{R}$ and let p, q, X, Y be defined as above. Then we define the maximum mean discrepancy (MMD) and its empirical estimate as*

Arbitrary Function Space:

$$\mathrm{MMD}\left[\mathcal{F}, p, q\right] := \sup_{f \in \mathcal{F}} \left(\mathbf{E}_{x \sim p}[f(x)] - \mathbf{E}_{y \sim q}[f(y)]\right), \tag{1}$$

$$\mathrm{MMD}\left[\mathcal{F}, X, Y\right] := \sup_{f \in \mathcal{F}} \left(\frac{1}{m}\sum_{i=1}^{m} f(x_i) - \frac{1}{n}\sum_{i=1}^{n} f(y_i)\right). \tag{2}$$

If f(x)=x, MMD is a first-order moment statistic;

If f(x)=x^2, MMD is a second-order moment statistic;

- Moment match does not guarantee the distribution similarity.
- So, MMD can measure the discrepancy in arbitrary function space.

http://www.gatsby.ucl.ac.uk/~gretton/mmd/mmd.htm

Theory---->Algorithm

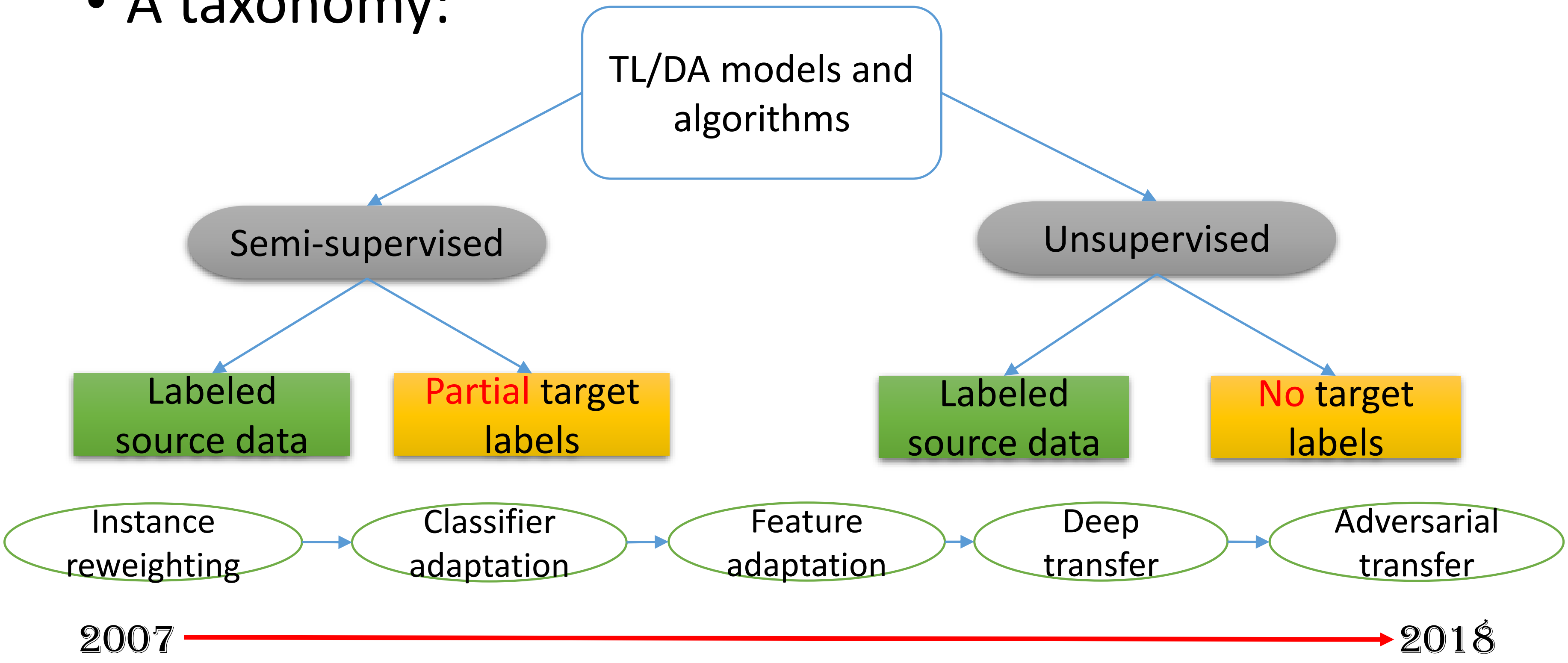
- Induced by the generalization bound theory, a number of models and algorithms are emerged, by focusing on three key points.

1) Source error $\epsilon_S(h)$
2) Domain discrepancy $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T)$
3) Combined error $\lambda = \min_{h\in\mathcal{H}} \epsilon_S(h) + \epsilon_T(h)$

Algorithm

How to design TL/DA models and algorithms?

- A taxonomy:



TL/DA models and algorithms

Semi-supervised

Unsupervised

Labeled source data

Partial target labels

Labeled source data

No target labels

Instance reweighting

Classifier adaptation

Feature adaptation

Deep transfer

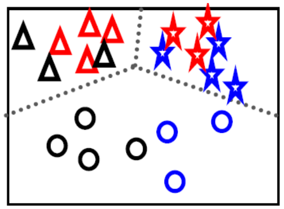
Adversarial transfer

2007

2018

Algorithm

Algorithm progress in the past 15 years



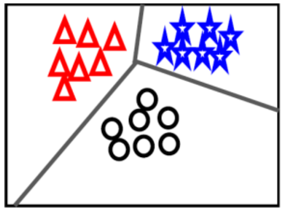
$$\min_{\beta} \|E_{x'\sim P'_r}[\Phi(x')] - E_{x\sim P_r}[\beta(x)\Phi(x)]\|$$

$$s.t. \quad \beta(x) \geq 0, E_{x\sim P_r}[\beta(x)] = 1$$

Instance re-weighting

Learn the instance weights of source data with domain alignment to target data.

2007 --

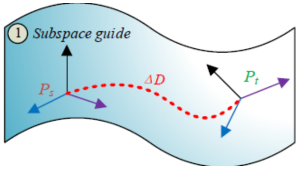


$$f(x) = f^a(x) + \Delta f(x)$$
$$= f^a(x) + w^T\phi(x)$$

Classifier transfer

Learn a generic classifier on source domain by leveraging a few labeled or unlabeled samples in target domain.

2007 --

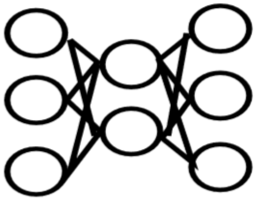


$$\min_{W} F(W, X_S, X_T, Y_S, Y'_T)$$
$$+ \lambda d^2_{\{m,c\}}(X_S, X_T, W)$$

Feature-level transfer

Learn a common subspace or a transformation to minimize domain discrepancy.

2010 --



$$\min_{\Theta} \frac{1}{N_s}\sum_{i=1}^{N_s} \mathcal{J}(\theta(x_i), y_i)$$
$$+ \lambda \sum_{l} d^2_{ma}(\mathcal{D}^l_s, \mathcal{D}^l_t)$$

Deep transfer

Using the CNN model, learn feature representations with maximum mean discrepancy across domains minimized.

2014 --

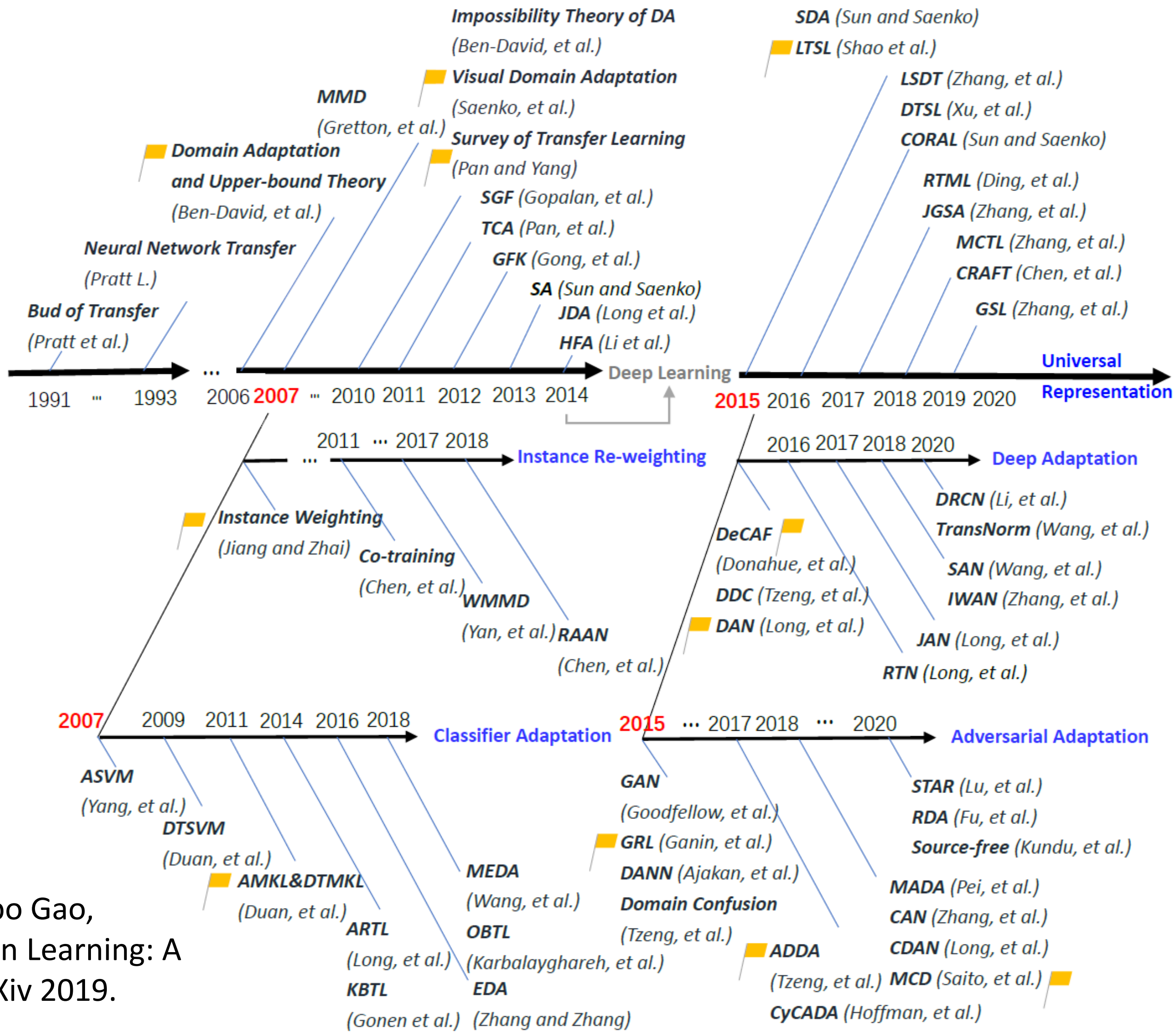
$$\begin{cases} \min\limits_{\theta_C,\theta_F} \mathcal{L}_C(\mathcal{D}_S, \mathcal{Y}_S; \theta_C, \theta_F) \\ \qquad - \lambda\mathcal{L}_D(\mathcal{D}_S, \mathcal{D}_T, \theta_D; \theta_F) \\ \min\limits_{\theta_D} \mathcal{L}_D(\mathcal{D}_S, \mathcal{D}_T, \theta_F; \theta_D) \end{cases}$$

Adversarial transfer

Using the idea of GAN, learn the feature generation model with domain confusion.

2015 --

Algorithm



Bud of Transfer *(Pratt et al.)*

Neural Network Transfer *(Pratt L.)*

Domain Adaptation and Upper-bound Theory *(Ben-David, et al.)*

MMD *(Gretton, et al.)*

Impossibility Theory of DA *(Ben-David, et al.)*

Visual Domain Adaptation *(Saenko, et al.)*

Survey of Transfer Learning *(Pan and Yang)*

SGF *(Gopalan, et al.)*

TCA *(Pan, et al.)*

GFK *(Gong, et al.)*

SA *(Sun and Saenko)*

JDA *(Long et al.)*

HFA *(Li et al.)*

1991 ... 1993 ... 2006 2007 ... 2010 2011 2012 2013 2014 → Deep Learning

SDA *(Sun and Saenko)*

LTSL *(Shao et al.)*

LSDT *(Zhang, et al.)*

DTSL *(Xu, et al.)*

CORAL *(Sun and Saenko)*

RTML *(Ding, et al.)*

JGSA *(Zhang, et al.)*

MCTL *(Zhang, et al.)*

CRAFT *(Chen, et al.)*

GSL *(Zhang, et al.)*

2015 2016 2017 2018 2019 2020 → Universal Representation

2011 ... 2017 2018 → Instance Re-weighting

Instance Weighting *(Jiang and Zhai)*

Co-training *(Chen, et al.)*

WMMD *(Yan, et al.)*

RAAN *(Chen, et al.)*

2016 2017 2018 2020 → Deep Adaptation

DeCAF *(Donahue, et al.)*

DDC *(Tzeng, et al.)*

DAN *(Long, et al.)*

DRCN *(Li, et al.)*

TransNorm *(Wang, et al.)*

SAN *(Wang, et al.)*

IWAN *(Zhang, et al.)*

JAN *(Long, et al.)*

RTN *(Long, et al.)*

2007 2009 2011 2014 2016 2018 → Classifier Adaptation

ASVM *(Yang, et al.)*

DTSVM *(Duan, et al.)*

AMKL&DTMKL *(Duan, et al.)*

ARTL *(Long, et al.)*

KBTL *(Gonen et al.)*

MEDA *(Wang, et al.)*

OBTL *(Karbalayghareh, et al.)*

EDA *(Zhang and Zhang)*

2015 ... 2017 2018 ... 2020 → Adversarial Adaptation

GAN *(Goodfellow, et al.)*

GRL *(Ganin, et al.)*

DANN *(Ajakan, et al.)*

Domain Confusion *(Tzeng, et al.)*

ADDA *(Tzeng, et al.)*

CyCADA *(Hoffman, et al.)*

STAR *(Lu, et al.)*

RDA *(Fu, et al.)*

Source-free *(Kundu, et al.)*

MADA *(Pei, et al.)*

CAN *(Zhang, et al.)*

CDAN *(Long, et al.)*

MCD *(Saito, et al.)*

Lei Zhang and Xinbo Gao, Transfer Adaptation Learning: A Decade Survey, arXiv 2019.

Algorithm

Principle of Instance Re-weighting (Jiang and Zhai, ACL 2007; Huang et al. NIPS 2007):

- Revisit the expected risk of test set in Pr':

$$R[\Pr', \theta, l(x,y,\theta)] = \mathbf{E}_{(x,y)\sim\Pr'}\left[l(x,y,\theta)\right] = \mathbf{E}_{(x,y)\sim\Pr}\left[\underbrace{\frac{\Pr'(x,y)}{\Pr(x,y)}}_{} l(x,y,\theta)\right]$$

- If Pr=Pr', it degenerates to the traditional ML; Otherwise, we let their ratio between them be $\beta(x,y)$
- Then, the regularized empirical risk becomes:

$$R_{\text{reg}}[Z, \beta, l(x,y,\theta)] := \frac{1}{m}\sum_{i=1}^{m} \beta_i l(x_i, y_i, \theta) + \lambda\Omega[\theta]$$

- β_i is the re-weighting coefficient w.r.t. sample i.

Algorithm

Principle of Instance Re-weighting (Jiang and Zhai, ACL 2007; Huang et al. NIPS 2007):

- Kernel mapping based re-weighting and reduces the domain discrepancy:

$$\begin{aligned} &\min_{\beta} \|E_{x' \sim P'_r}[\Phi(x')] - E_{x \sim P_r}[\beta(x)\Phi(x)]\| \\ &s.t. \quad \beta(x) \geq 0, E_{x \sim P_r}[\beta(x)] = 1 \end{aligned}$$

- Similarly, re-weighted maximum mean discrepancy

$$d^2_{wmmd} = \|\frac{1}{\Sigma_{i=1}^{M} \alpha_{y_i^s}} \sum_{i=1}^{M} \alpha_{y_i^s} \phi(x_i^s) - \frac{1}{N} \sum_{j=1}^{N} \phi(x_j^t)\|^2_{\mathcal{H}}$$

Algorithm

Principle of Classifier Adaptation (Yang et al. ACM MM'07; Duan et al. CVPR'12, TPAMI'12; Wang et al. ACM MM'18):

- Learn a common classifier on **source domain**, by leveraging a few labeled/ unlabeled target samples from **target domain**
- **Assumption**: There exists a *delta function* between the auxiliary classifier (source) f_a and the new classifier (target) *f*.

$$f(\mathbf{x}) = f^a(\mathbf{x}) + \boxed{\Delta f(\mathbf{x})} = f^a(\mathbf{x}) + \boxed{\mathbf{w}^T \phi(\mathbf{x})}$$

Standard SVM

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i$$

$$\text{s.t.} \quad \xi_i \geq 0, \quad y_i \mathbf{w}^T \phi(\mathbf{x}_i) \geq 1 - \xi_i, \quad \forall(\mathbf{x}_i, y_i) \in \mathcal{D}_l^p$$

Adaptive SVM →

ASVM

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i$$

$$\text{s.t.} \quad \xi_i \geq 0$$

$$y_i f^a(\mathbf{x}_i) + \boxed{y_i \mathbf{w}^T \phi(\mathbf{x}_i)} \geq 1 - \xi_i.$$

Algorithm

Principle of Classifier Adaptation (Yang et al. ACM MM'07; Duan et al. CVPR'12, TPAMI'12; Wang et al. ACM MM'18):

- With similar idea, from SVM to MKL (multi-kernel learning):

$$f^T(\mathbf{x}) = \sum_{p=1}^{P} \beta_p f_p(\mathbf{x}) + \underbrace{\sum_{m=1}^{M} d_m \mathbf{w}'_m \varphi_m(\mathbf{x}) + b}_{\Delta f(\mathbf{x})}$$

- Introduces the domain discrepancy

$$\mathrm{DIST}_k(\mathcal{D}^A, \mathcal{D}^T) = \left\| \frac{1}{n_A} \sum_{i=1}^{n_A} \varphi(\mathbf{x}_i^A) - \frac{1}{n_T} \sum_{i=1}^{n_T} \varphi(\mathbf{x}_i^T) \right\|_{\mathcal{H}}$$

- Adaptive MKL (AMKL)

$$\min_{\mathbf{d} \in \mathcal{M}} G(\mathbf{d}) = \frac{1}{2} \Omega^2(\mathbf{d}) + \theta J(\mathbf{d}),$$

where

$$J(\mathbf{d}) = \min_{\mathbf{w}_m, \beta, b, \xi_i} \frac{1}{2} \left(\sum_{m=1}^{M} d_m \|\mathbf{w}_m\|^2 + \lambda \|\beta\|^2 \right) + C \sum_{i=1}^{n} \xi_i,$$
$$\text{s.t.} \quad y_i f^T(\mathbf{x}_i) \geq 1 - \xi_i, \xi_i \geq 0,$$

Algorithm

Principle of Classifier Adaptation (Yang et al. ACM MM'07; Duan et al. CVPR'12, TPAMI'12; Wang et al. ACM MM'18):

Representative work (*zero padding feature augmentation, low-rank solution* *and* *delta function***):**

① Daumé III, et al. ACL'07(Frustrating Easy Adaptation, EA)

② Li, et al. TPAMI'14 (HFA)

$$\Phi^s(\boldsymbol{x}) = \langle \boxed{\boldsymbol{x}}, \boldsymbol{x}, \mathbf{0} \rangle, \quad \Phi^t(\boldsymbol{x}) = \langle \boxed{\boldsymbol{x}}, \mathbf{0}, \boldsymbol{x} \rangle$$

kernelize

$$\Phi^s(x) = \langle \Phi(x), \Phi(x), \mathbf{0} \rangle$$
$$\Phi^t(x) = \langle \Phi(x), \mathbf{0}, \Phi(x) \rangle$$

Examples in Re-ID (WeiShi Zheng and Jianhuang Lai):

- View-specific transform for Re-ID (IJCAI'15, TPAMI'18)
- Deep zero padding

Algorithm

Principle of Classifier Adaptation (Yang et al. ACM MM'07; Duan et al. CVPR'12, TPAMI'12; Wang et al. ACM MM'18):

Representative work (*zero padding feature augmentation, low-rank solution* and *delta function*):

③ Li, et al. TPAMI'18 (LRE-SVMs)
④ Zhang, et al. IEEE Sens.'17 (MFKS)

$$R_{reg}[W, l(X_S, X_T, W)] = \sum R_{emp}[w_i, l(X_S, X_T, w_i)]$$

$$+\|[w_1, w_2, \cdots, w_D]\|_*$$

⑤ Joachmis, ICML'1999 (T-SVM)
⑥ Yang, et al. ACM MM'07 (ASVM)
⑦ Duan, et al. TPAMI'12 (AMKL)
⑧ Duan, et al. TPAMI'13 (DTSVM, DTMKL)

$$f(\mathbf{x}) = f^a(\mathbf{x}) + \Delta f(\mathbf{x}) = f^a(\mathbf{x}) + \mathbf{w}^T\phi(\mathbf{x})$$

Algorithm

Principle of Feature Adaptation:

- Subspace unification (Pan et al. TKDE'10; TNNLS'11; Hoffman et al. IJCV'14; Kan et al. IJCV'14)
- Manifold alignment (Gopalan et al. ICCV'11, SGF; Gong, et al. CVPR'12, GFK; Fernando, et al. ICCV'13, SA)
- Feature reconstruction/representation (Jhuo, et al. CVPR'12, RDALR; Shao, et al. IJCV'14, LTSL; Zhang et al. TIP'16, LSDT; Xu et al. TIP'16, DTSL)

Algorithm

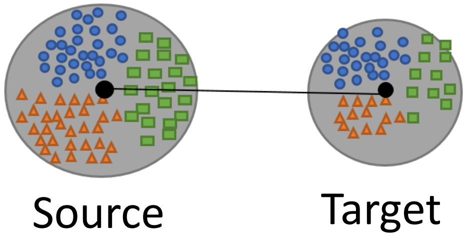
✓**Subspace unification:**

- General paradigm (domain-common/shared subspace learning)

$$R_{reg}[\mathbf{W}, l(X_S, X_T, \mathbf{W})] = R_{emp}[\mathbf{W}, l(X_S, X_T, \mathbf{W})]+\Omega[\mathbf{W}]$$

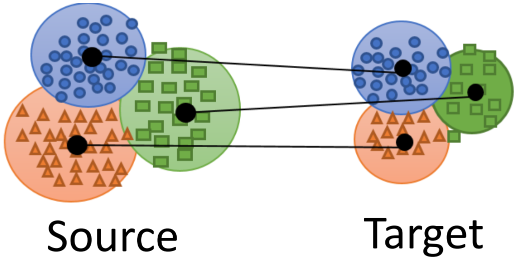
Marginal distribution consistency

$$P\big(\phi(X_S)\big) \approx P\big(\phi(X_T)\big)$$



Conditional distribution consistency

$$P\big(\phi\big(X_S^i\big)|y_S^i\big) \approx P\big(\phi\big(X_T^i\big)|y_T^i\big), i = 1, \cdots, C$$

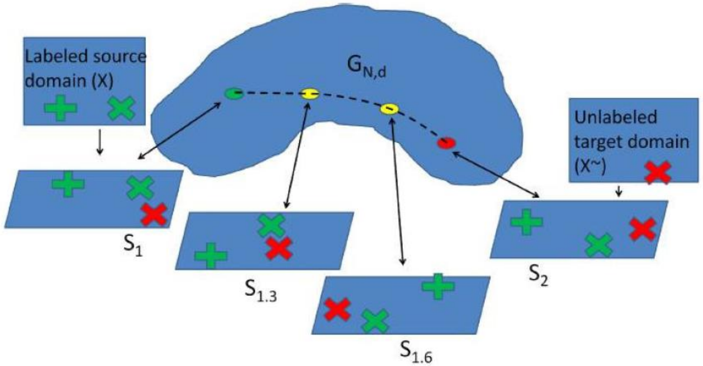


- **W** is a transformation matrix.

Algorithm

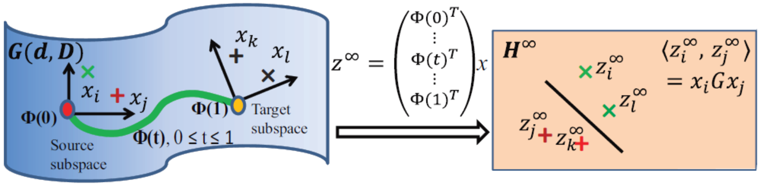
✓Manifold alignment:

- General paradigm (learn mapping)



SGF

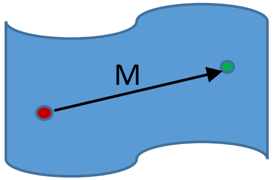
find some intermediate representation along the geodesic path



GFK

construct kernels along the geodesic path to model the domain shift by using infinite subspaces

G is a positive semi-definite mapping matrix



$$F(M) = ||X_S M - X_T||_F^2$$

$$M^* = argmin_M(F(M))$$

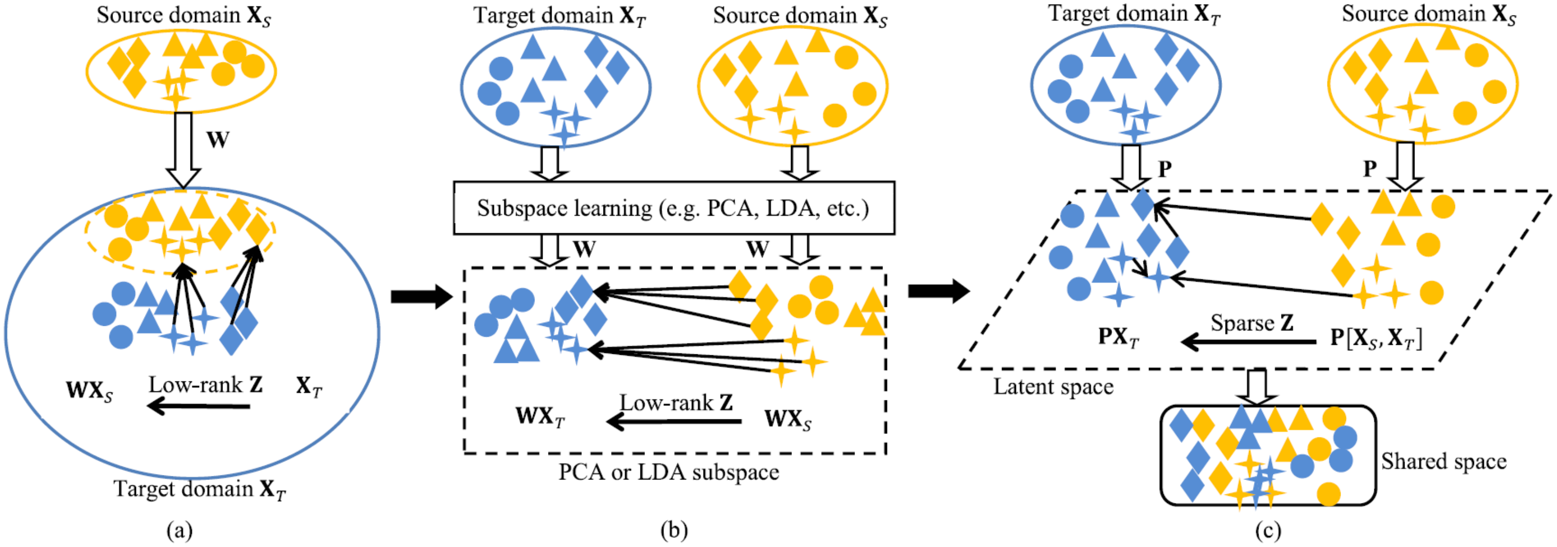
SA

learn the linear mapping M that makes the subspace closer

Algorithm

✓Feature reconstruction:

- General paradigm (low-rank/sparse coding)


Source domain $\mathbf{X}_S$
W
$\mathbf{WX}_S$
Low-rank **Z**
$\mathbf{X}_T$
Target domain $\mathbf{X}_T$
(a)
Target domain $\mathbf{X}_T$
Source domain $\mathbf{X}_S$
Subspace learning (e.g. PCA, LDA, etc.)
W
W
$\mathbf{WX}_T$
Low-rank **Z**
$\mathbf{WX}_S$
PCA or LDA subspace
(b)
Target domain $\mathbf{X}_T$
Source domain $\mathbf{X}_S$
P
P
$\mathbf{PX}_T$
Sparse **Z**
$\mathbf{P}[\mathbf{X}_S, \mathbf{X}_T]$
Latent space
Shared space
(c)

LRR: **strength** (better locality of data, block-wise structure, neighbor to neighbor reconstruction)
weakness (strong assumption of independent subspaces and sufficient data, easy to get trivial solution)

$$\min_{W,Z,E} F(W) + \Re(Z) + \Omega(E)$$
$$\text{s.t. } f(X_T) = f(X_S)Z + E$$

F(.) is subspace learning fun.
f(.) is transformation fun.

Algorithm

✓**Feature reconstruction:**

- General paradigm (low-rank/sparse coding)

For a better basis:

Domain adaptive dictionary (Rama Chellappa. CVPR’13, SDDL)

- **1) domain-shared dictionary**

$$\min_{D,P,\alpha} \sum_{k\in\{s,t\}} \|P_{(k)}X_{(k)} - D\alpha_{(k)}\|_F^2 + \Re(D,P,\alpha)$$

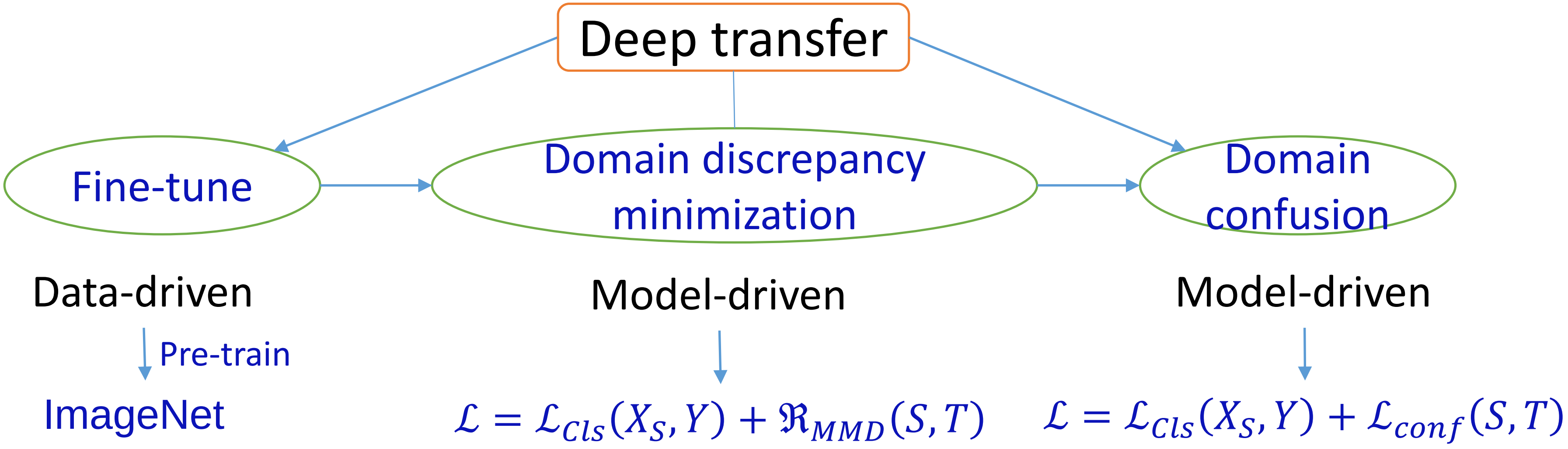
- **2) domain-specific dictionary**

$$\min_{D,P,\alpha} \sum_{k\in\{s,t\}} \|X_{(k)} - D_{(k)}\alpha_{(k)}\|_F^2 + \Omega(\alpha_s,\alpha_t)$$

- **α denotes domain-specific coding coefficient w.r.t. the basis of dictionary D.**

Algorithm

Principle of Deep Network Adaptation:



Deep transfer

Fine-tune

Domain discrepancy minimization

Domain confusion

Data-driven

Pre-train

ImageNet

Model-driven

$\mathcal{L} = \mathcal{L}_{Cls}(X_S, Y) + \Re_{MMD}(S, T)$

Model-driven

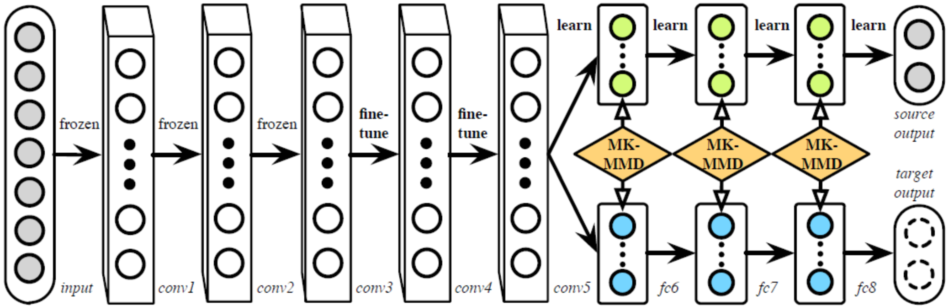
$\mathcal{L} = \mathcal{L}_{Cls}(X_S, Y) + \mathcal{L}_{conf}(S, T)$

- Fine-tune is a kind of intuitive and data-driven transfer learning method, which depends on pretrained models with task-specific source database (e.g., ImageNet)

Algorithm

Deep network with discrepancy alignment:

- Learn general feature representation with *domain discrepancy minimization* in supervised manner (Tzeng, arXiv'14; Long et al. ICML'15, NIPS'16; Yan, et al. CVPR'17; Rozantsev et al. CVPR'18)

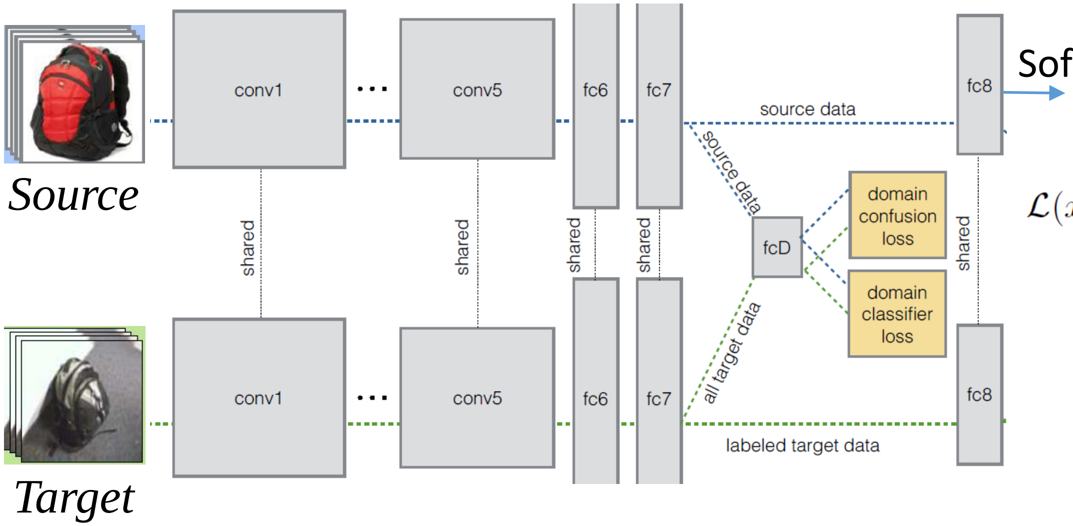


$$\mathcal{L}_{\texttt{fixed}} = \mathcal{L}_{\texttt{class}} + \mathcal{L}_{\texttt{disc}}$$

Algorithm

Domain confusion:

- Learn general feature representation with *domain confusion/domain alignment* (Ajakan et al. NIPS'14, DANN; Tzeng et al. ICCV'15, DDC; Murez et al. CVPR'18)



Softmax cross-entropy loss

$$\begin{aligned}\mathcal{L}(x_S, y_S, x_T, y_T, \theta_D;\theta_{\text{repr}}, \theta_C) = \\ \mathcal{L}_C(x_S, y_S, x_T, y_T; \theta_{\text{repr}}, \theta_C) \\ + \lambda \mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{\text{repr}}) \\ + \nu \mathcal{L}_{\text{soft}}(x_T, y_T; \theta_{\text{repr}}, \theta_C).\end{aligned}$$

Goal: learning domain-invariant representation

Algorithm

Principle of Deep Adversarial Adaptation:

- Learn feature generation model with *domain confusion* (Ganin et al. JMLR’16; Tzeng et al. CVPR’17, ADDA; Chen et al. CVPR’18, RAAN; Saito et al. CVPR’18, MCD; Pinheiro, CVPR’18)
- This kind of models are approaching the minimax essence of domain adaptation.

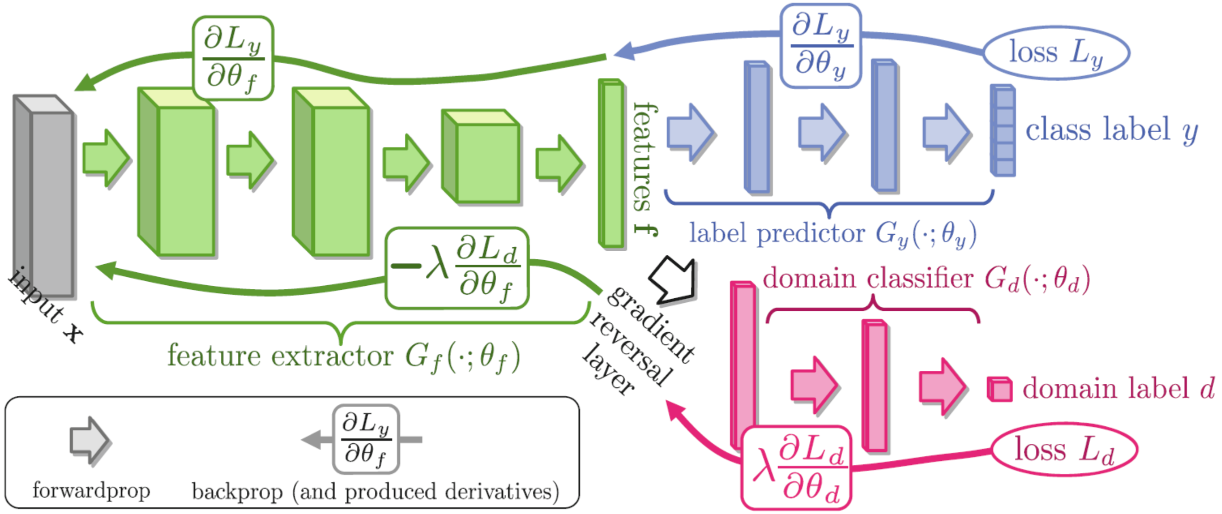
maximization

View 1: $$\min d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) = \boxed{2 \sup_{\eta \in \mathcal{H}\Delta\mathcal{H}} \left| Pr_{\tilde{\mathcal{D}}_S}[z : \eta(z) = 1] - Pr_{\tilde{\mathcal{D}}_T}[z : \eta(z) = 1] \right|}$$

View 2: $$\min_f d_{\mathcal{A}}(\mathcal{S}, \mathcal{T}) \Leftrightarrow \min_f 2\left(1 - 2 \min_{h \in \mathcal{H}} err\left(h(x)\right)\right) \Leftrightarrow \max_f \min_{h \in \mathcal{H}} err\left(h(x)\right)$$

Algorithm

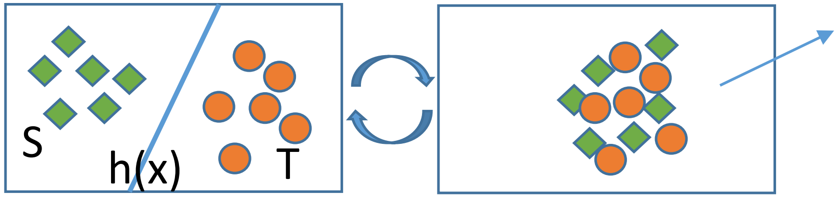
Principle of Deep Adversarial Adaptation:



$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
class label y
features $\mathbf{f}$
label predictor $G_y(\cdot;\theta_y)$
$-\lambda\frac{\partial L_d}{\partial \theta_f}$
input x
feature extractor $G_f(\cdot;\theta_f)$
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
domain label d
$\lambda\frac{\partial L_d}{\partial \theta_d}$
loss L_d
forwardprop
$\frac{\partial L_y}{\partial \theta_f}$
backprop (and produced derivatives)

Gradient reversal layer for adversarial domain adaptation

$$\max_{f}\min_{h\in\mathcal{H}} err\left(h(x)\right)$$



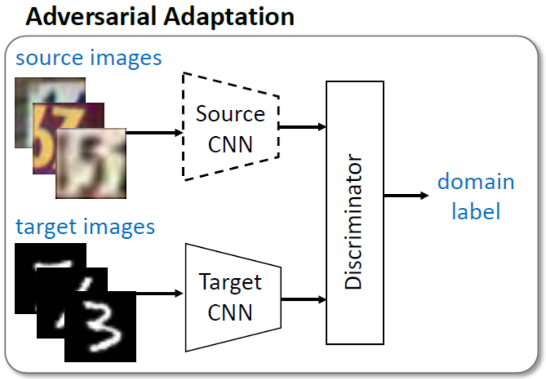
$\min\limits_{h\in\mathcal{H}} err\big(h(x)\big)$

$\max\limits_{f} err\big(h(x)\big)$

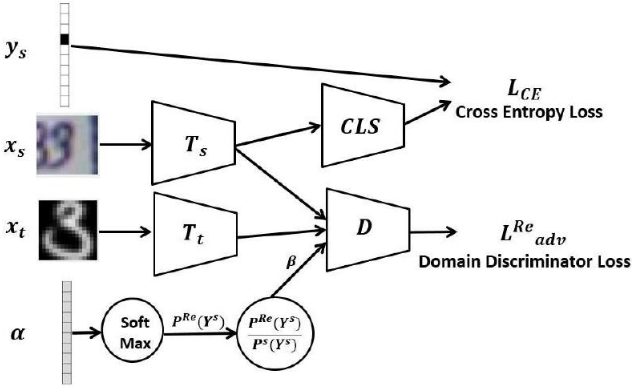
Aligned (confusion)

Algorithm

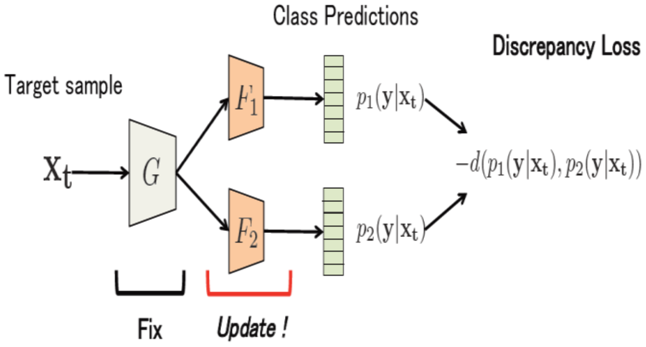
Principle of Deep Adversarial Adaptation:



ADDA
Adversarial Discriminative
Domain Adaptation



RAAN
Re-weighted Adversarial
Adaptation Network



MCD
Maximize Classifier
Discrepancy

Algorithm

NEW DIRECTIONS/TOPICS in TL/DA:

✓**Category shift problem (catastrophic misalignment)**

UniDA (sample-level weighting idea based on entropy and uncertainty) to simultaneously solve **partial** and **open-set** DA.

✓**Inaccuracy of Target pseudo-labels**

Clustering based idea and Progressive training.

✓**Multi-source DA**

Aggregate multiple sources to one source domain, +DA.

✓**Source-free DA**

Source data is unseen (**data privacy**), and you can only access the source pre-trained model. **Essence**: Parameter transfer.

✓**Domain generalization**

✓**……**

This will be introduced in another slides, pls wait in patience

Contents

- Part I: Background and Preliminary
- Part II: Concept, Theory and Algorithms
- Part III: Applications of TL/DA Algorithms
- Summary

Applications

Transfer Learning + X

Applications

- **Application fields (machine learning related AI topics)**

1) Computer vision
2) Natural language processing
3) Big data analysis
4) Smart instruments
5) Medical image analysis
6) Remote sensing
7) …

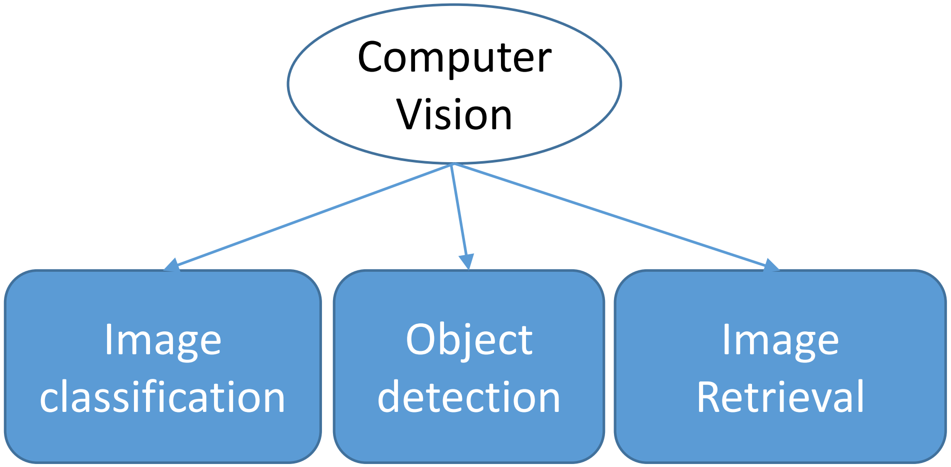
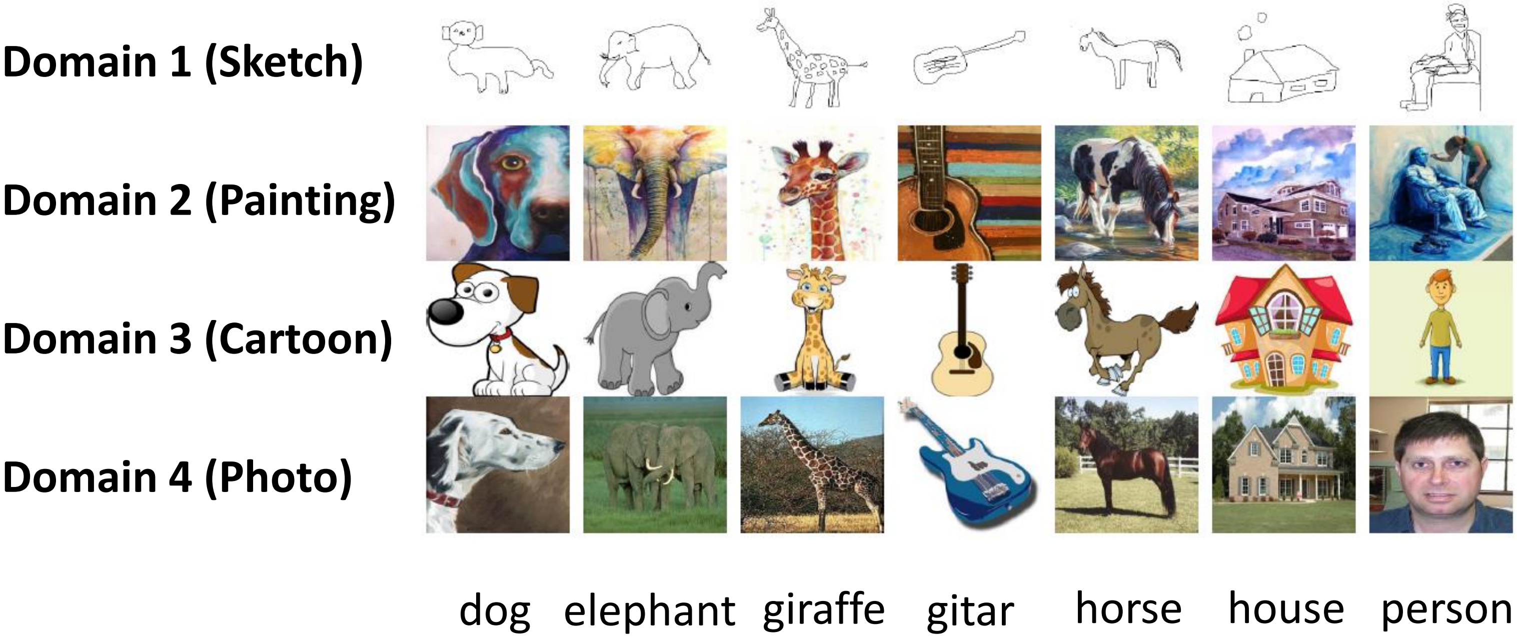


Image Classification

- Image classification is the benchmark task for testing each new TL/DA model and algorithm.
- Cross-domain multi-class classification is standard protocol.



Transferable Image Classification

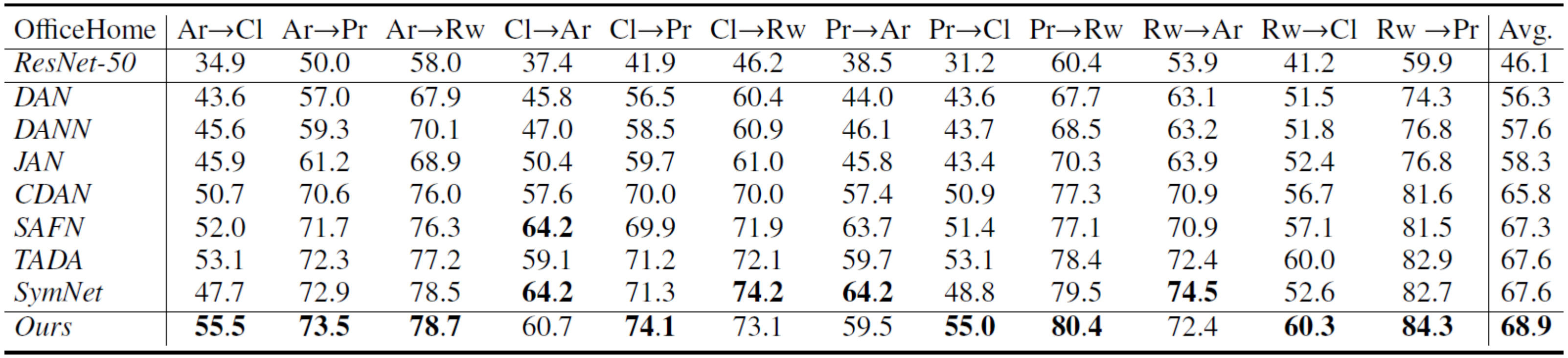
Benchmark Datasets

- Office-31 (3DA, 3 domains, 31 classes, 4652 images)
- Office+Caltech-10 (4DA, 4 domains, 10 classes, 2533 images)
- MNIST+USPS (3 domains, 10 classes, 67291 images)
- Multi-PIE (5 domains, 68 ids, 41368 faces)
- COIL-20 (2 domains, 20 classes, 1440 images)
- MSRC+VOC2007 (2 domains, 6 classes, 9000+images)
- IVLSC (4 domains, 5 classes, 15000+images)
- Office-Home (4 domains, 65 classes, 15500 images)
- ImageCLEF (3 domains, 12 classes, 1800 images)
- P-A-C-S (4 domains, 7 classes, ~10000 images)
- VisDA (2 domains, 12 classes, 280000 images)

Transferable Image Classification

SoTA Performance

Cross-domain classification accuracy on Office-Home (Resnet-50 backbone)



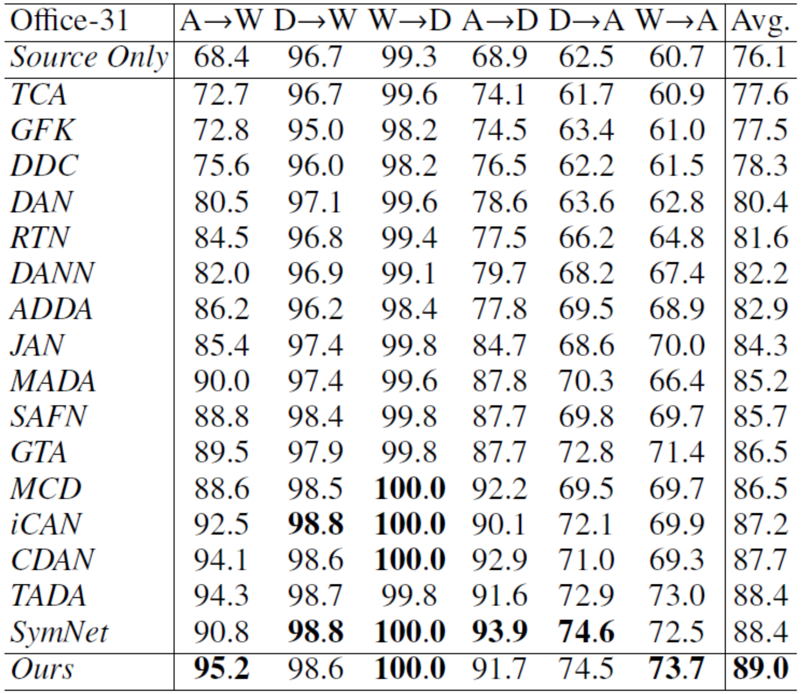
OfficeHome	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw →Pr	Avg.
ResNet-50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
SAFN	52.0	71.7	76.3	**64.2**	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
TADA	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
SymNet	47.7	72.9	78.5	**64.2**	71.3	**74.2**	**64.2**	48.8	79.5	**74.5**	52.6	82.7	67.6
Ours	**55.5**	**73.5**	**78.7**	60.7	**74.1**	73.1	59.5	**55.0**	**80.4**	72.4	**60.3**	**84.3**	**68.9**

S. Wang and L. Zhang, “Self-adaptive Re-weighted Adversarial Domain Adaptation,” IJCAI, 2020.
Long, et al. “Conditional Adversarial Domain Adaptation,” NeurIPS, 2018.

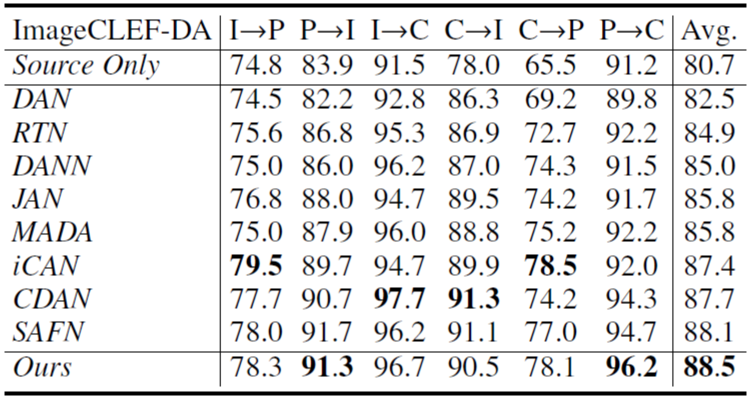
Transferable Image Classification

SoTA Performance

Cross-domain classification accuracy on Office-31, ImageCLEF (Resnet-50 backbone)

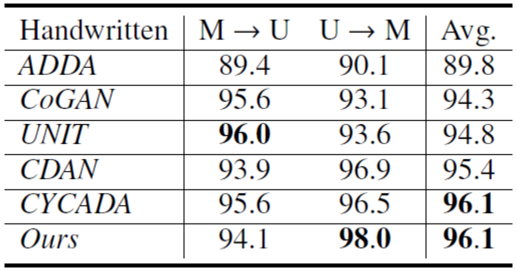


Office-31	A→W	D→W	W→D	A→D	D→A	W→A	Avg.
Source Only	68.4	96.7	99.3	68.9	62.5	60.7	76.1
TCA	72.7	96.7	99.6	74.1	61.7	60.9	77.6
GFK	72.8	95.0	98.2	74.5	63.4	61.0	77.5
DDC	75.6	96.0	98.2	76.5	62.2	61.5	78.3
DAN	80.5	97.1	99.6	78.6	63.6	62.8	80.4
RTN	84.5	96.8	99.4	77.5	66.2	64.8	81.6
DANN	82.0	96.9	99.1	79.7	68.2	67.4	82.2
ADDA	86.2	96.2	98.4	77.8	69.5	68.9	82.9
JAN	85.4	97.4	99.8	84.7	68.6	70.0	84.3
MADA	90.0	97.4	99.6	87.8	70.3	66.4	85.2
SAFN	88.8	98.4	99.8	87.7	69.8	69.7	85.7
GTA	89.5	97.9	99.8	87.7	72.8	71.4	86.5
MCD	88.6	98.5	**100.0**	92.2	69.5	69.7	86.5
iCAN	92.5	**98.8**	**100.0**	90.1	72.1	69.9	87.2
CDAN	94.1	98.6	**100.0**	92.9	71.0	69.3	87.7
TADA	94.3	98.7	99.8	91.6	72.9	73.0	88.4
SymNet	90.8	**98.8**	**100.0**	**93.9**	**74.6**	72.5	88.4
Ours	**95.2**	98.6	**100.0**	91.7	74.5	**73.7**	**89.0**



ImageCLEF-DA	I→P	P→I	I→C	C→I	C→P	P→C	Avg.
Source Only	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DAN	74.5	82.2	92.8	86.3	69.2	89.8	82.5
RTN	75.6	86.8	95.3	86.9	72.7	92.2	84.9
DANN	75.0	86.0	96.2	87.0	74.3	91.5	85.0
JAN	76.8	88.0	94.7	89.5	74.2	91.7	85.8
MADA	75.0	87.9	96.0	88.8	75.2	92.2	85.8
iCAN	**79.5**	89.7	94.7	89.9	**78.5**	92.0	87.4
CDAN	77.7	90.7	**97.7**	**91.3**	74.2	94.3	87.7
SAFN	78.0	91.7	96.2	91.1	77.0	94.7	88.1
Ours	78.3	**91.3**	96.7	90.5	78.1	**96.2**	**88.5**

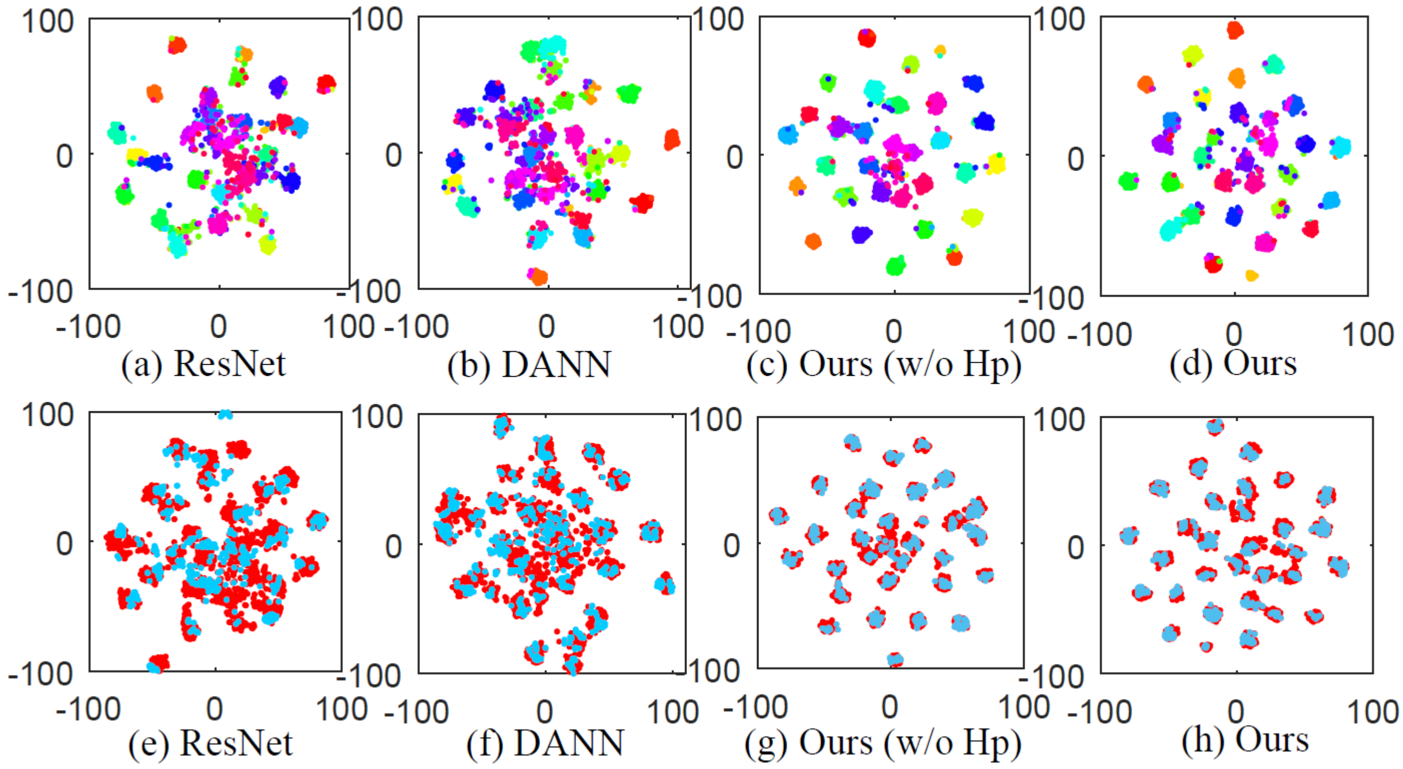
Classification accuracy on MNIST-USPS



Handwritten	M → U	U → M	Avg.
ADDA	89.4	90.1	89.8
CoGAN	95.6	93.1	94.3
UNIT	**96.0**	93.6	94.8
CDAN	93.9	96.9	95.4
CYCADA	95.6	96.5	**96.1**
Ours	94.1	**98.0**	**96.1**

Transferable Image Classification

Domain adaptive/transferable feature visualization



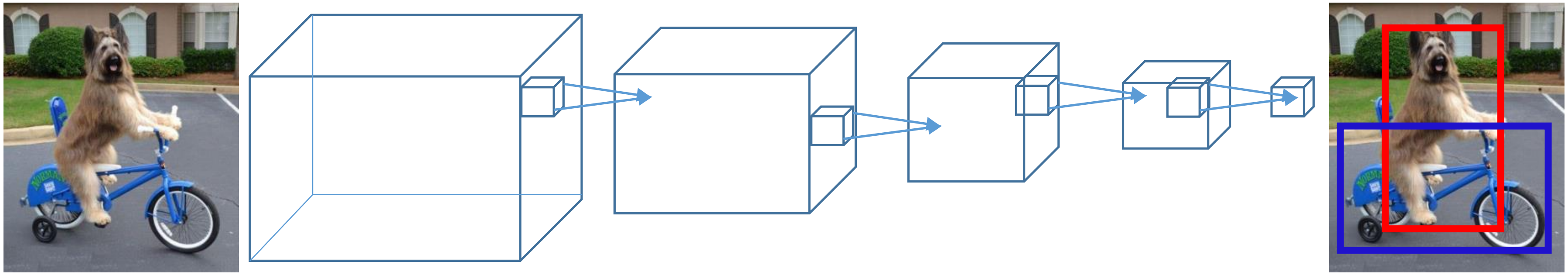
(a) ResNet (b) DANN (c) Ours (w/o Hp) (d) Ours

(e) ResNet (f) DANN (g) Ours (w/o Hp) (h) Ours

Object Detection

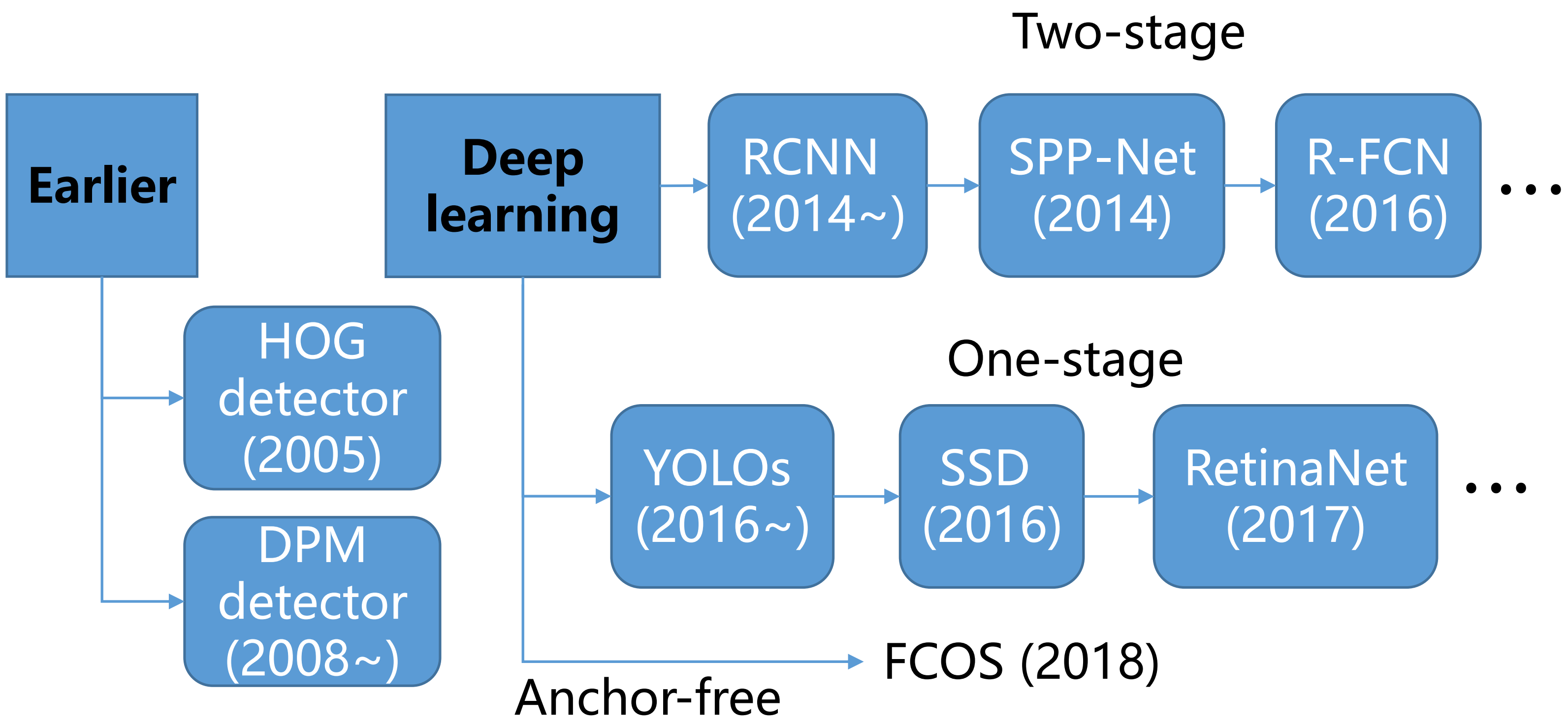
Definition of a detection task

- Image classification focuses on high-level abstract feature semantics
- Object detection is a multi-task issue (classification vs. localization) , the low-level features are also useful



Object Detection

A number of models in object detection



Object Detection

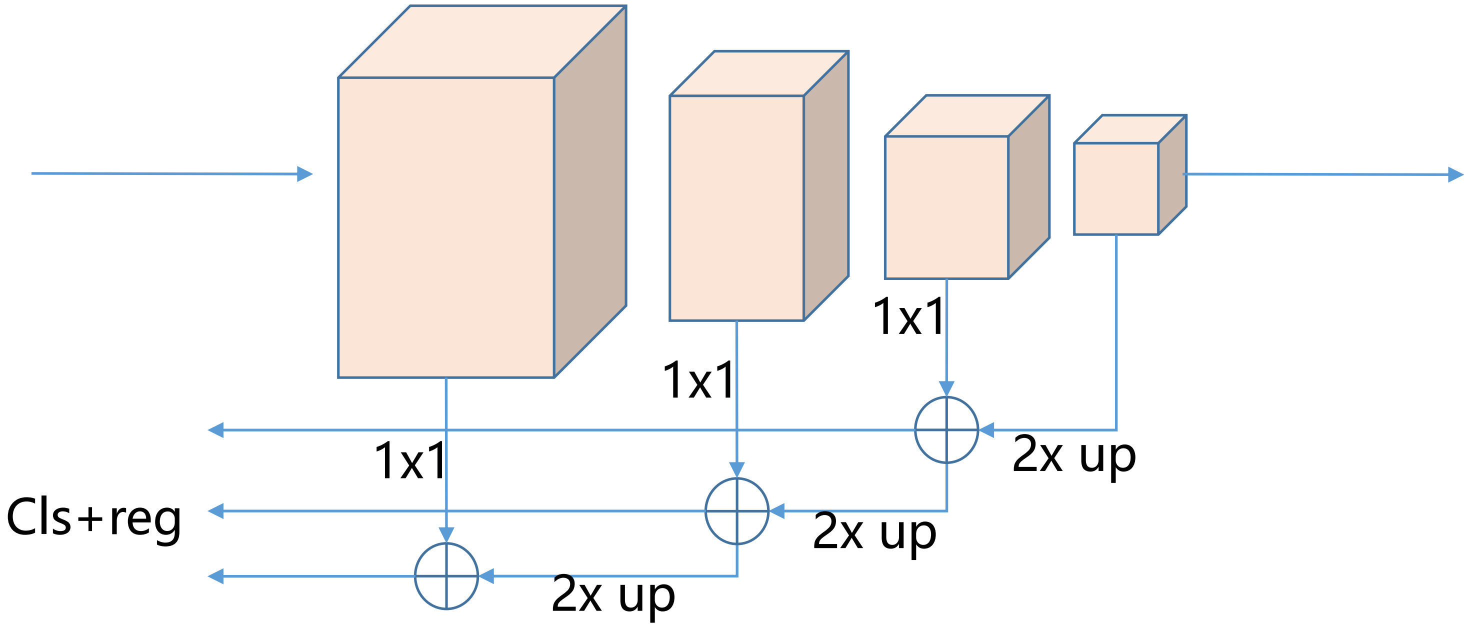
Imbalance Problems of object detection

- Problem 1: Scale imbalance (object of different size, scale) SSD, FPN
- Problem 2: Class imbalance (foreground vs. background, easy vs. hard) Focal loss, OHEM
- Problem 3: IoU imbalance (IoU levels, low vs. high quality, outliers) Libra RCNN (balanced L1), GHM, IoU based losses
- Problem 4: Loss imbalance (cls. vs. reg.) PISA, classification aware localization loss, paid less attention

Object Detection

Solving scale imbalance problem: Feature Pyramid Network

- Lower levels have better spatial resolution
- Higher levels have stronger semantics information

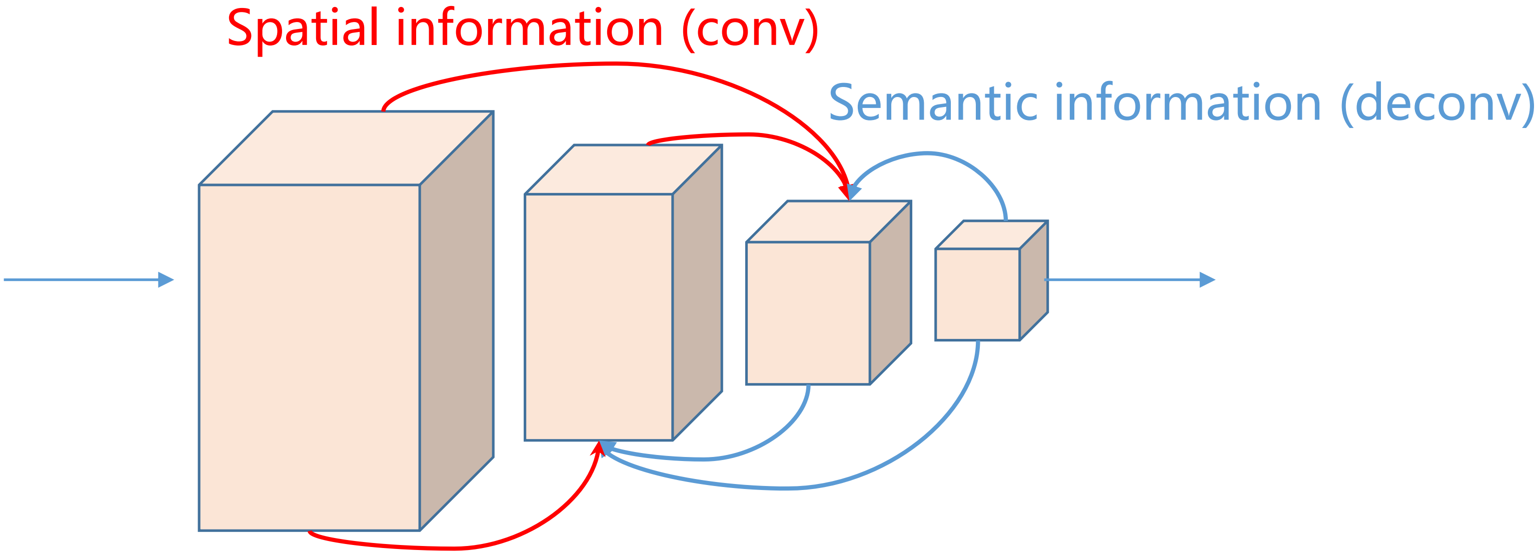


- In FPN, high-level semantic information is back-propagated to the low-level, but information loss is serious.

Object Detection

Single-shot Two-pronged Detector (TPNet):

- A new architecture of two-pronged bi-directional interaction and transfer between low-levels and high-levels
- A Rectified Intersection of Union (RIoU) loss

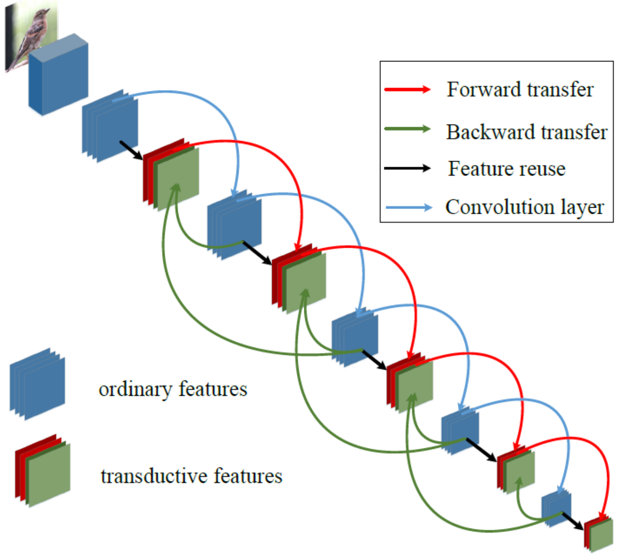


Keyang Wang and Lei Zhang, Single-Shot Two-Pronged Net with Rectified IoU Loss, ACM MM 2020. Oral paper)

Object Detection

Single-shot Two-pronged Detector (TPNet):

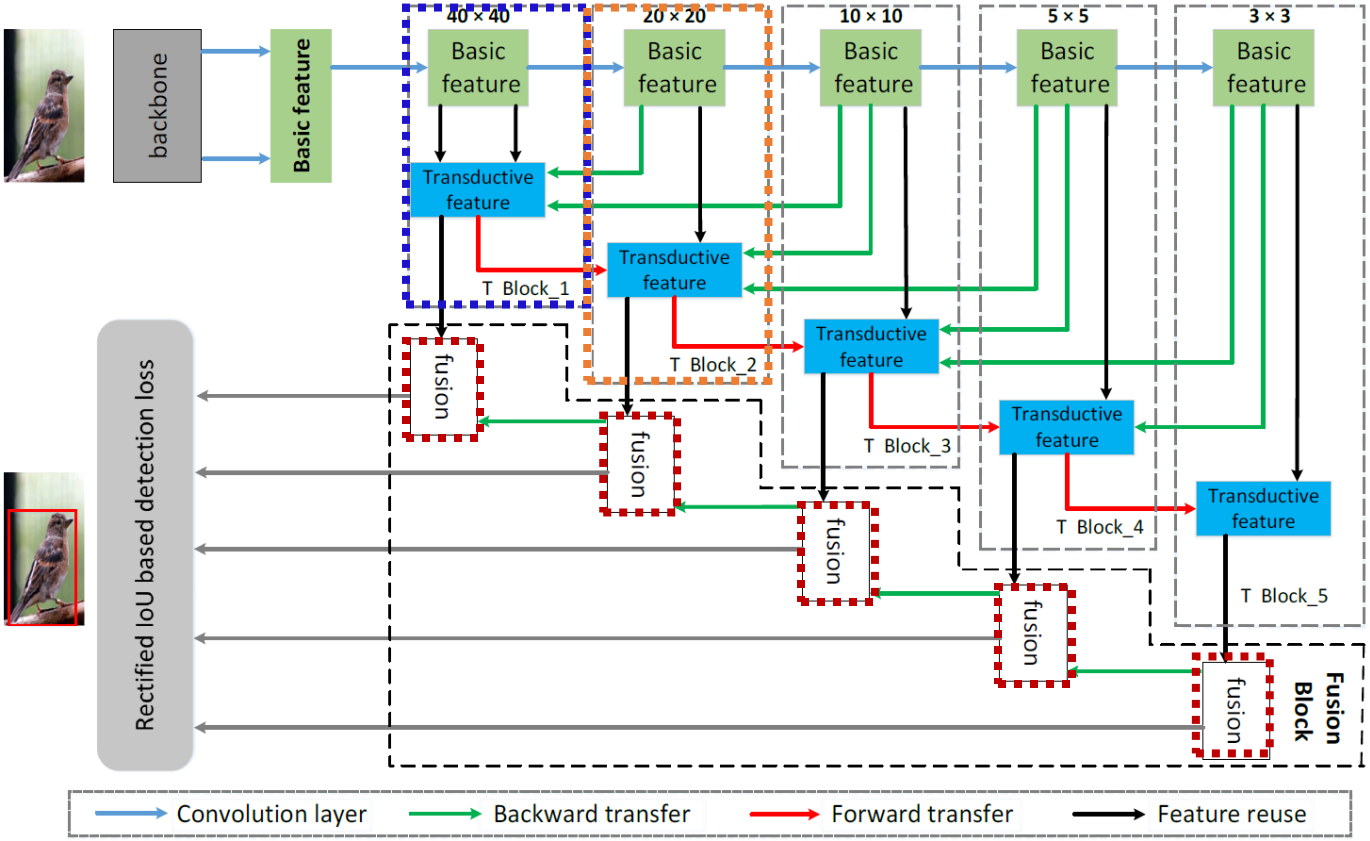
- A new architecture of two-pronged bi-directional interaction and transfer between low-levels and high-levels
- A Rectified Intersection of Union (RIoU) loss



Keyang Wang and Lei Zhang, Single-Shot Two-Pronged Net with Rectified IoU Loss, ACM MM 2020. Oral paper)

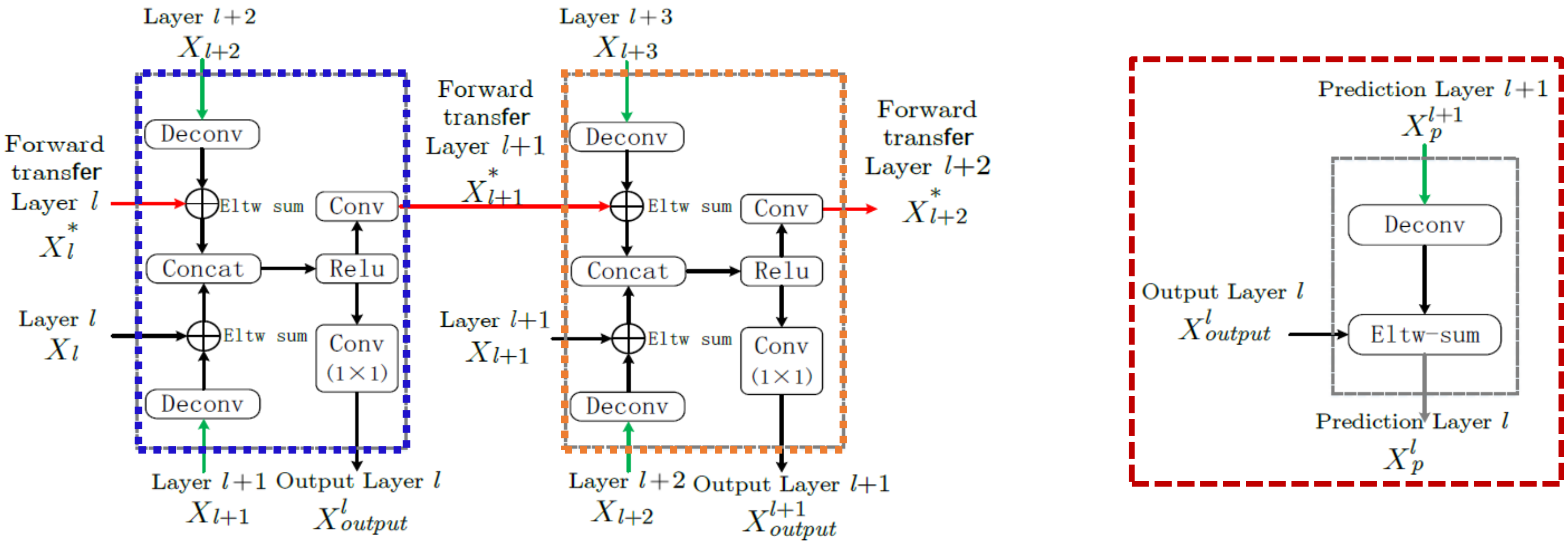
Object Detection

- **TPNet architecture**



Object Detection

- **Transductive block and Fusion block**



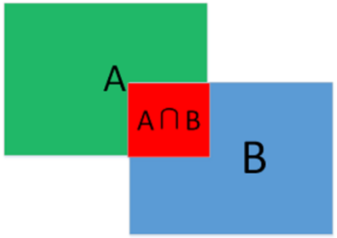
(a) T block L

(b) T block L+1

(c) Fusion block

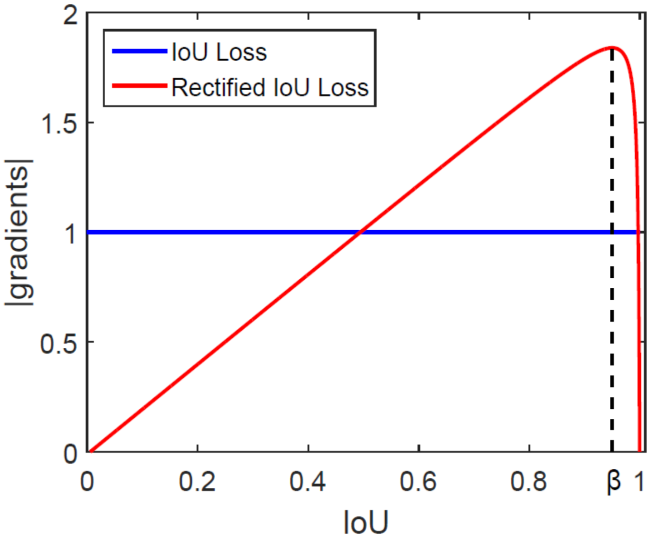
Object Detection

- **Gradient guided RIoU loss**



$$IOU = \frac{A \cap B}{A \cup B}$$

Revisit the standard IoU loss $\mathcal{L}_{IoU} = 1 - IoU$



We can define the gradient (red curve) as:

$$|gradients(IoU)| = |\frac{\partial \mathcal{L}_{RIoU}}{\partial IoU}| = (aIoU + b) + \frac{k}{(IoU - c)}$$

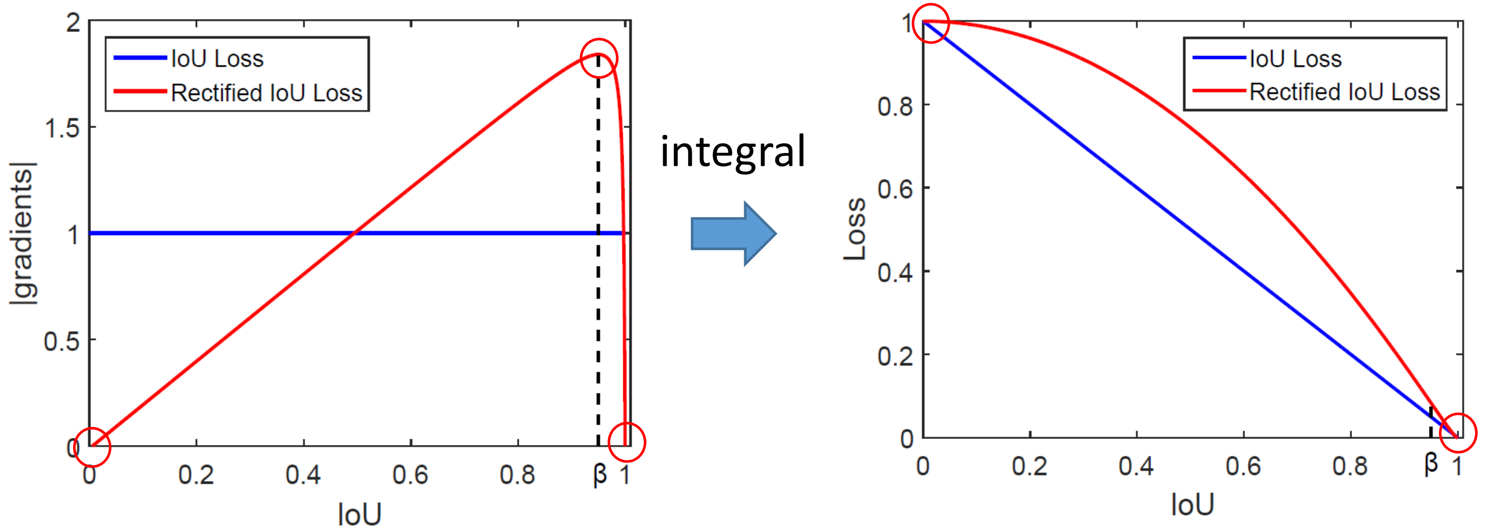
integral

$$\mathcal{L}_{RIoU} = 1 - (\frac{a}{2}IoU^2 + bIoU + k ln|IoU - c| + t)$$

Note 5 model parameters analytically determined with 5 equations.

Object Detection

- **Gradient guided RIoU loss**



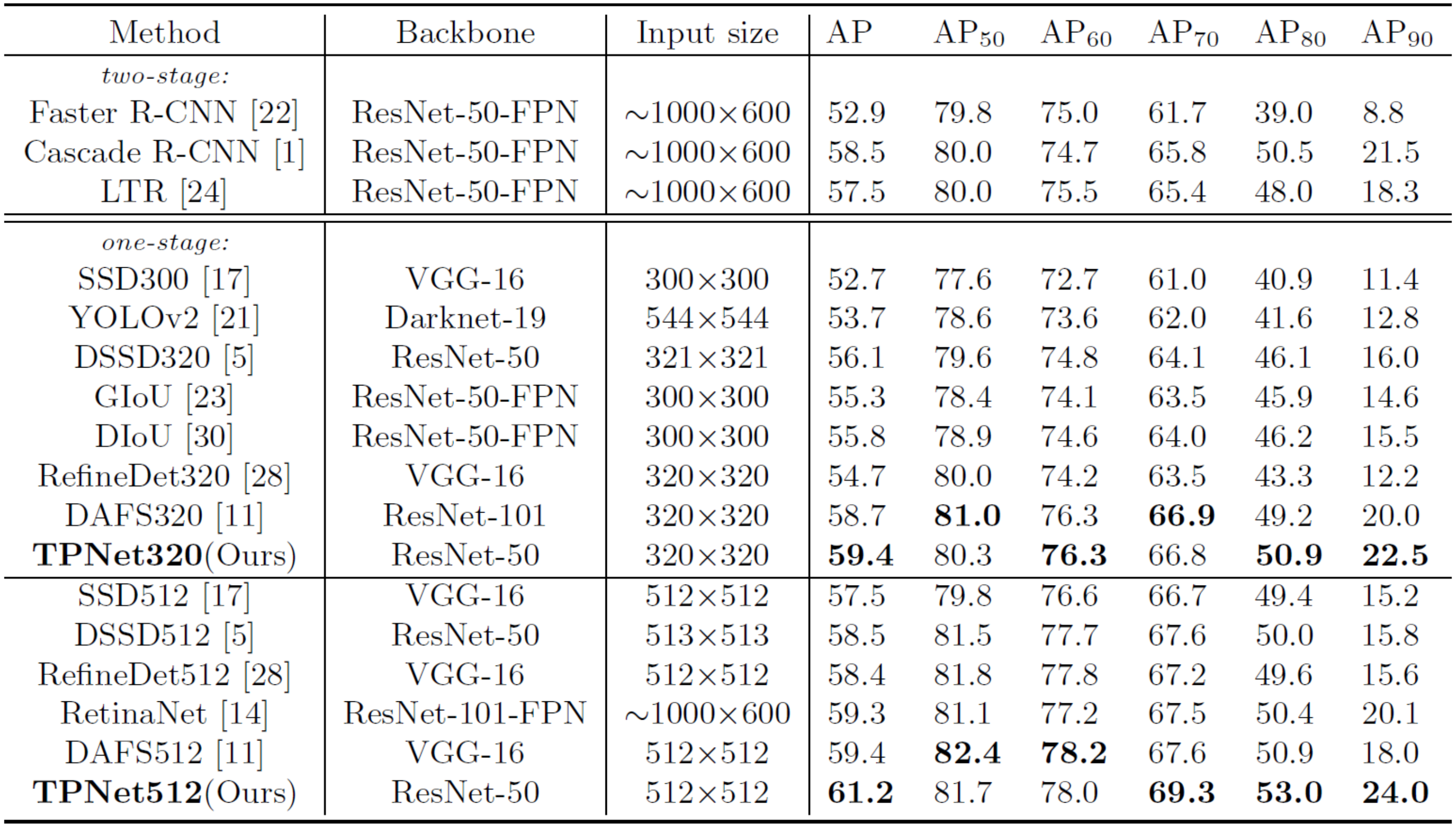
$$\begin{cases} b - \dfrac{k}{c} = 0 \\ a + b + \dfrac{k}{1-c} = 0 \\ c - \sqrt{\dfrac{k}{a}} = \beta \end{cases}$$

$$\begin{cases} 1 - kln|c| - t = 1 \\ 1 - \dfrac{a}{2} - b - kln|1-c| - t = 0 \end{cases}$$

$$\mathcal{L}_{det} = \mathcal{L}_{cls} + \mathcal{L}_{RIoU} + \rho(B_p^{ct}, B_g^{ct})$$

Object Detection

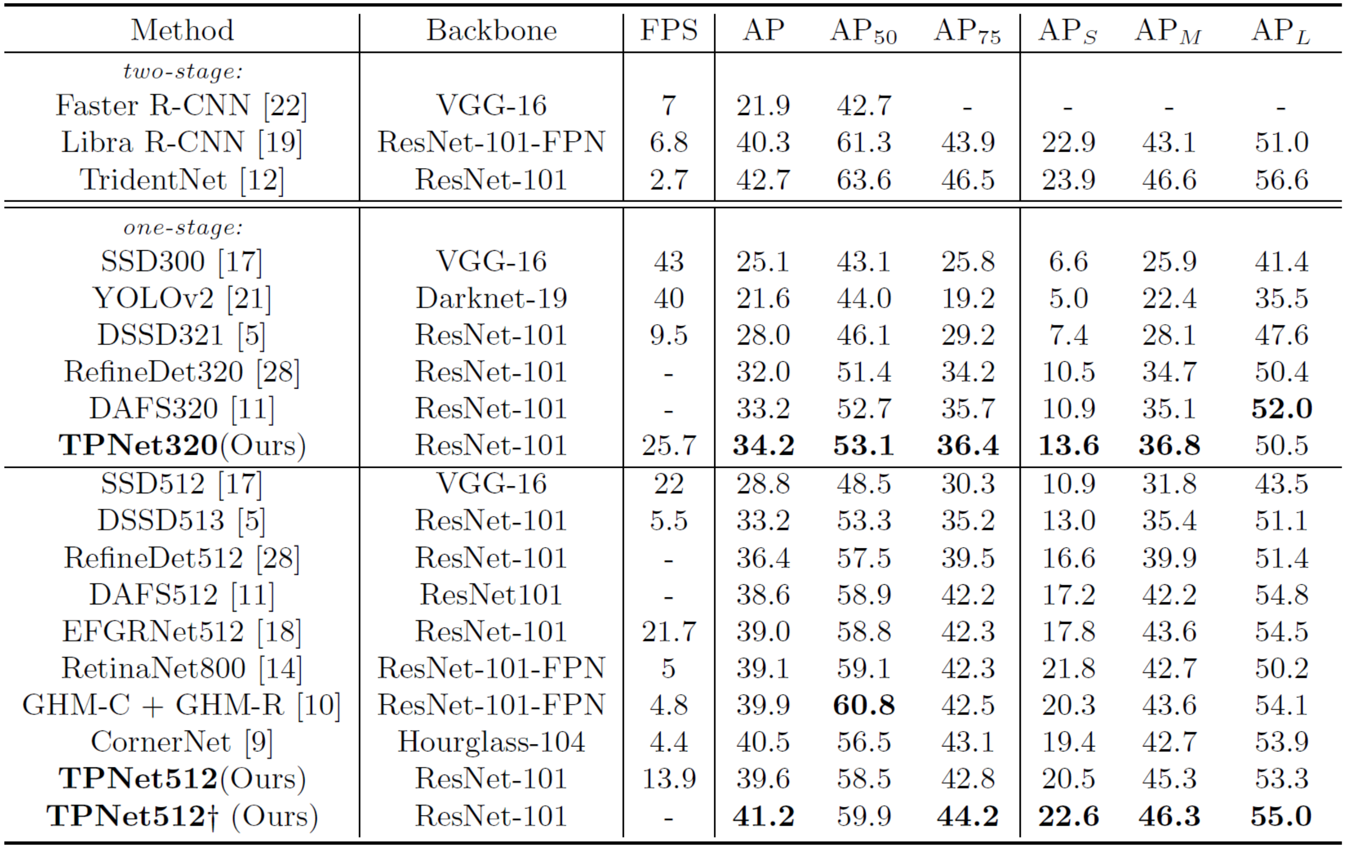
- **Experiments on PASCAL VOC**



Method	Backbone	Input size	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
two-stage:								
Faster R-CNN [22]	ResNet-50-FPN	~1000×600	52.9	79.8	75.0	61.7	39.0	8.8
Cascade R-CNN [1]	ResNet-50-FPN	~1000×600	58.5	80.0	74.7	65.8	50.5	21.5
LTR [24]	ResNet-50-FPN	~1000×600	57.5	80.0	75.5	65.4	48.0	18.3
one-stage:								
SSD300 [17]	VGG-16	300×300	52.7	77.6	72.7	61.0	40.9	11.4
YOLOv2 [21]	Darknet-19	544×544	53.7	78.6	73.6	62.0	41.6	12.8
DSSD320 [5]	ResNet-50	321×321	56.1	79.6	74.8	64.1	46.1	16.0
GIoU [23]	ResNet-50-FPN	300×300	55.3	78.4	74.1	63.5	45.9	14.6
DIoU [30]	ResNet-50-FPN	300×300	55.8	78.9	74.6	64.0	46.2	15.5
RefineDet320 [28]	VGG-16	320×320	54.7	80.0	74.2	63.5	43.3	12.2
DAFS320 [11]	ResNet-101	320×320	58.7	**81.0**	76.3	**66.9**	49.2	20.0
TPNet320(Ours)	ResNet-50	320×320	**59.4**	80.3	**76.3**	66.8	**50.9**	**22.5**
SSD512 [17]	VGG-16	512×512	57.5	79.8	76.6	66.7	49.4	15.2
DSSD512 [5]	ResNet-50	513×513	58.5	81.5	77.7	67.6	50.0	15.8
RefineDet512 [28]	VGG-16	512×512	58.4	81.8	77.8	67.2	49.6	15.6
RetinaNet [14]	ResNet-101-FPN	~1000×600	59.3	81.1	77.2	67.5	50.4	20.1
DAFS512 [11]	VGG-16	512×512	59.4	**82.4**	**78.2**	67.6	50.9	18.0
TPNet512(Ours)	ResNet-50	512×512	**61.2**	81.7	78.0	**69.3**	**53.0**	**24.0**

Object Detection

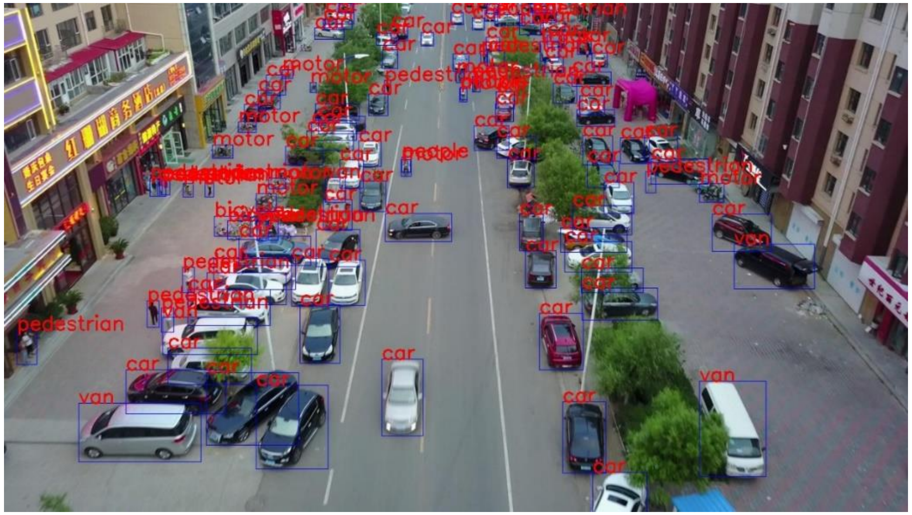
- **Experiments on MS COCO**



Method	Backbone	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
two-stage:								
Faster R-CNN [22]	VGG-16	7	21.9	42.7	-	-	-	-
Libra R-CNN [19]	ResNet-101-FPN	6.8	40.3	61.3	43.9	22.9	43.1	51.0
TridentNet [12]	ResNet-101	2.7	42.7	63.6	46.5	23.9	46.6	56.6
one-stage:								
SSD300 [17]	VGG-16	43	25.1	43.1	25.8	6.6	25.9	41.4
YOLOv2 [21]	Darknet-19	40	21.6	44.0	19.2	5.0	22.4	35.5
DSSD321 [5]	ResNet-101	9.5	28.0	46.1	29.2	7.4	28.1	47.6
RefineDet320 [28]	ResNet-101	-	32.0	51.4	34.2	10.5	34.7	50.4
DAFS320 [11]	ResNet-101	-	33.2	52.7	35.7	10.9	35.1	**52.0**
TPNet320(Ours)	ResNet-101	25.7	**34.2**	**53.1**	**36.4**	**13.6**	**36.8**	50.5
SSD512 [17]	VGG-16	22	28.8	48.5	30.3	10.9	31.8	43.5
DSSD513 [5]	ResNet-101	5.5	33.2	53.3	35.2	13.0	35.4	51.1
RefineDet512 [28]	ResNet-101	-	36.4	57.5	39.5	16.6	39.9	51.4
DAFS512 [11]	ResNet101	-	38.6	58.9	42.2	17.2	42.2	54.8
EFGRNet512 [18]	ResNet-101	21.7	39.0	58.8	42.3	17.8	43.6	54.5
RetinaNet800 [14]	ResNet-101-FPN	5	39.1	59.1	42.3	21.8	42.7	50.2
GHM-C + GHM-R [10]	ResNet-101-FPN	4.8	39.9	**60.8**	42.5	20.3	43.6	54.1
CornerNet [9]	Hourglass-104	4.4	40.5	56.5	43.1	19.4	42.7	53.9
TPNet512(Ours)	ResNet-101	13.9	39.6	58.5	42.8	20.5	45.3	53.3
TPNet512† (Ours)	ResNet-101	-	**41.2**	59.9	**44.2**	**22.6**	**46.3**	**55.0**

Object Detection

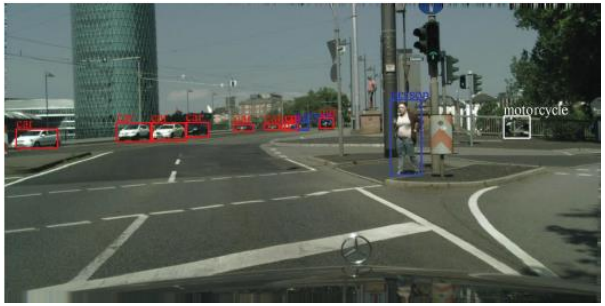
- **Visualization**



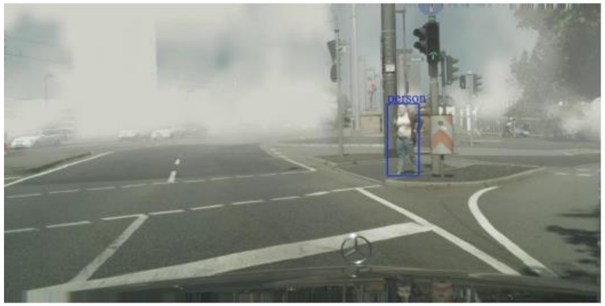
Flaws of Object Detectors

Motivation

- Labels of a specific target domain are not free to use.
- Leveraging a related source domain is natural.
- Existing detectors (one stage vs. two-stage) are not transferable.



Normal weather (source)

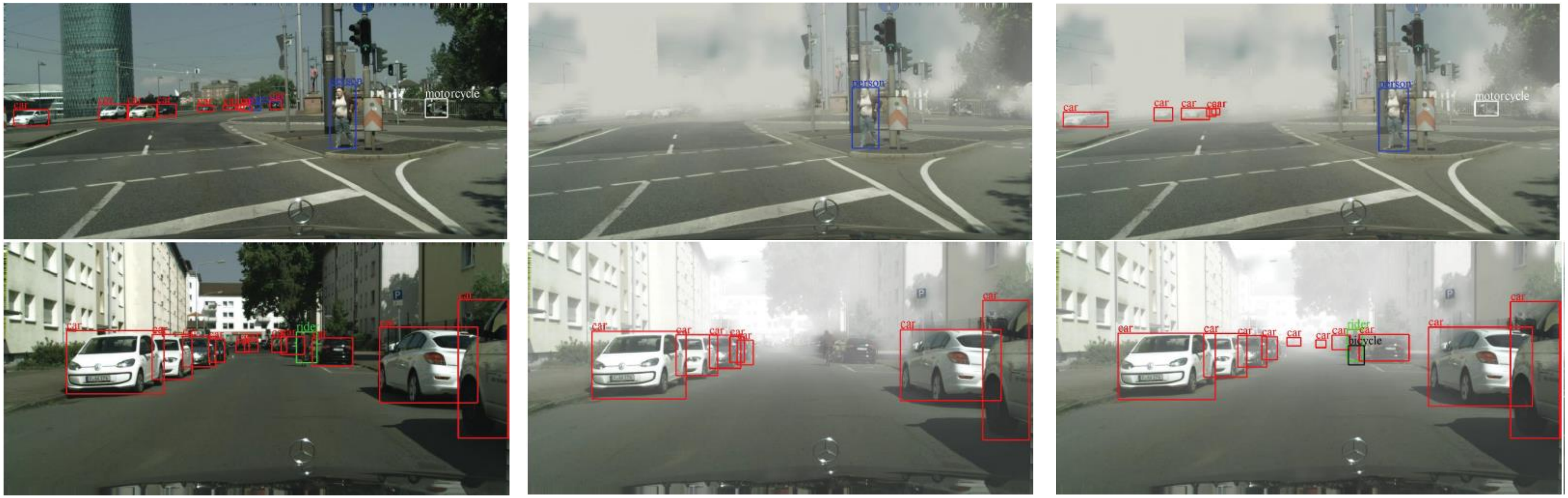


Foggy weather (target)

Image degradation (noise, e.g. haze) has a clear impact on learning

Flaws of Object Detectors

Perspective of distribution mismatch



High-quality detection (Faster RCNN)

Low-quality detection (Faster RCNN)

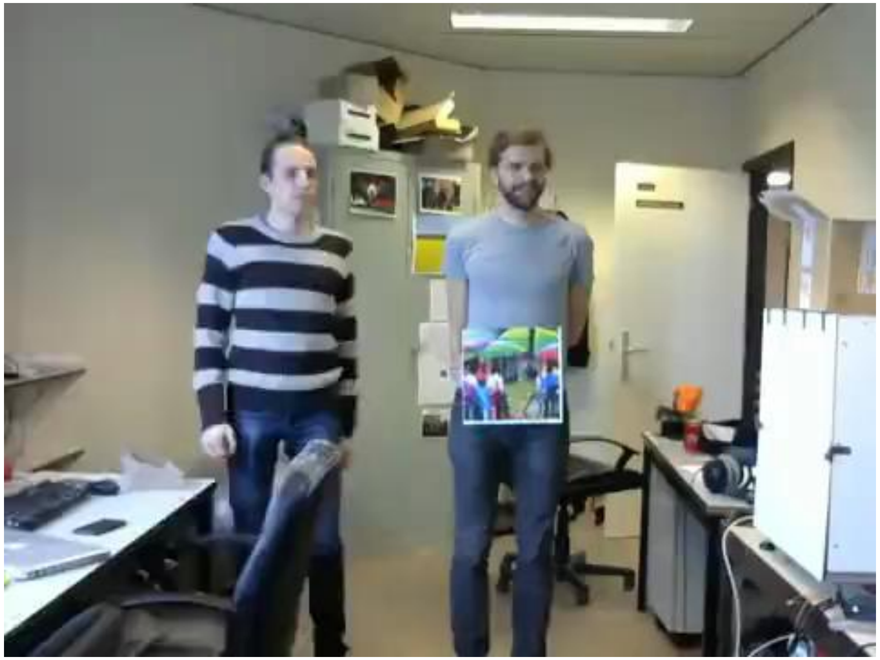
High-quality detection (Transfer Learning)

Z. He and L. Zhang, Multi-adversarial Faster RCNN for Unrestricted Object Detection, ICCV, 2019.

Flaws of Object Detectors

Perspective of adversarial sample

- Attack YOLO-v2 to fool surveillance cameras (AI is not safe).



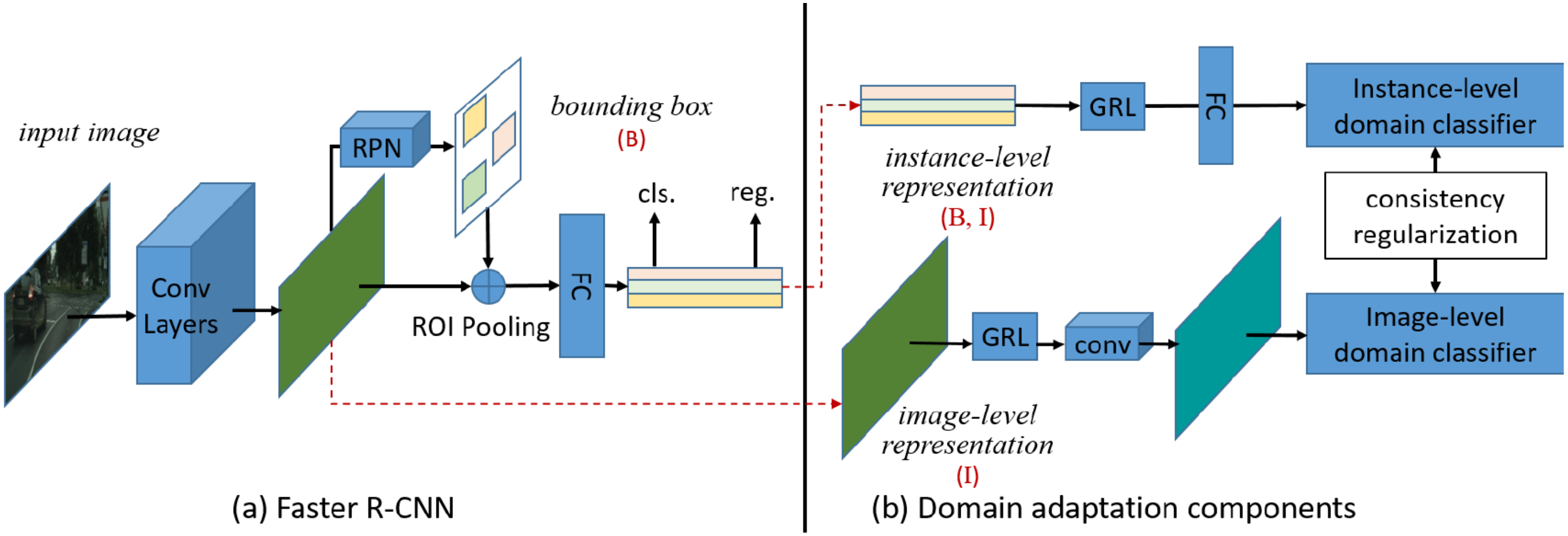
Simen Thys, W.V. Ranst, Fooling automatic surveillance cameras: adversarial patches to attack person detection, arXiv, 2019.

Domain-adaptive Object Detection

DAF model based on $\mathcal{H}$-divergence

$$\min_f d_{\mathcal{H}}(\mathcal{S},\mathcal{T}) \Leftrightarrow \max_f \min_{h\in\mathcal{H}} \{err_{\mathcal{S}}(h(\mathbf{x})) + err_{\mathcal{T}}(h(\mathbf{x}))\}$$

1) Image-level alignment; 2)Instance-level alignment


input image
Conv Layers
RPN
bounding box
(B)
ROI Pooling
FC
cls.
reg.
instance-level representation
(B, I)
GRL
FC
Instance-level domain classifier
consistency regularization
Image-level domain classifier
GRL
conv
image-level representation
(I)

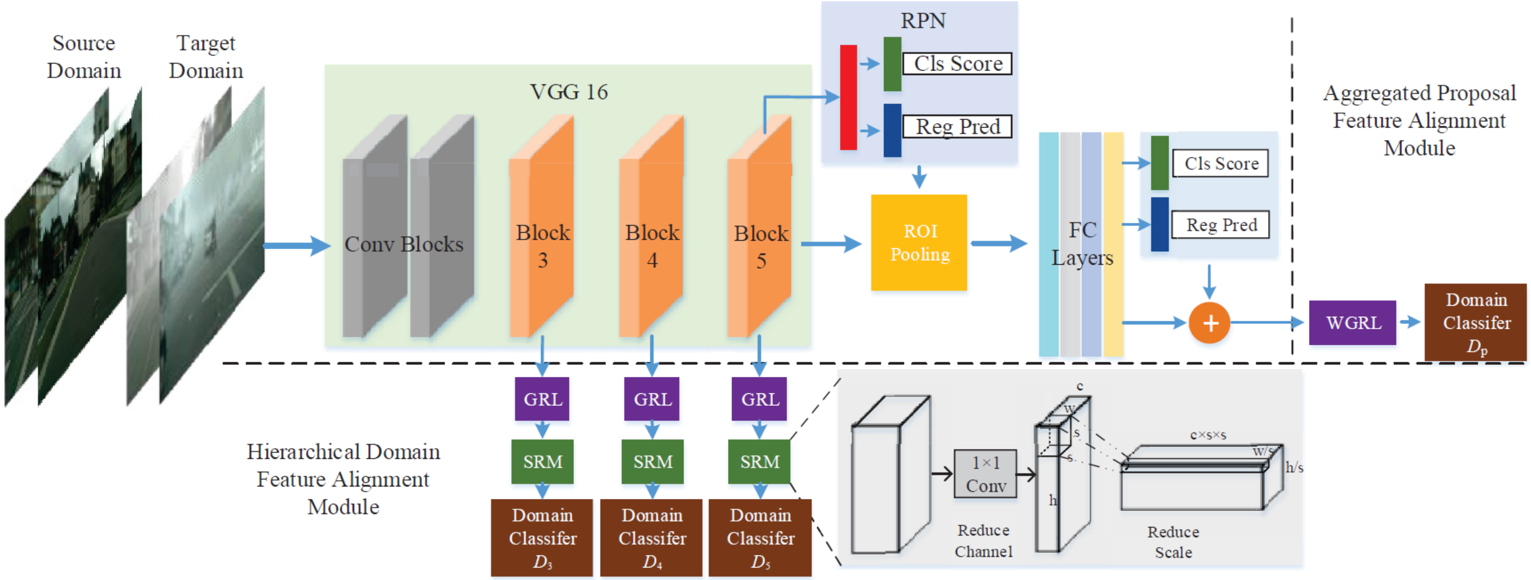
(a) Faster R-CNN

(b) Domain adaptation components

Chen et al., Domain adaptive faster-rcnn for object detection in the wild, CVPR, 2018.

Domain-adaptive Object Detection

MAF model in ICCV'19



- Multiple GRLs for adversarial domain adaptation

Z. He and L. Zhang, Multi-adversarial Faster RCNN for Unrestricted Object Detection, ICCV, 2019.

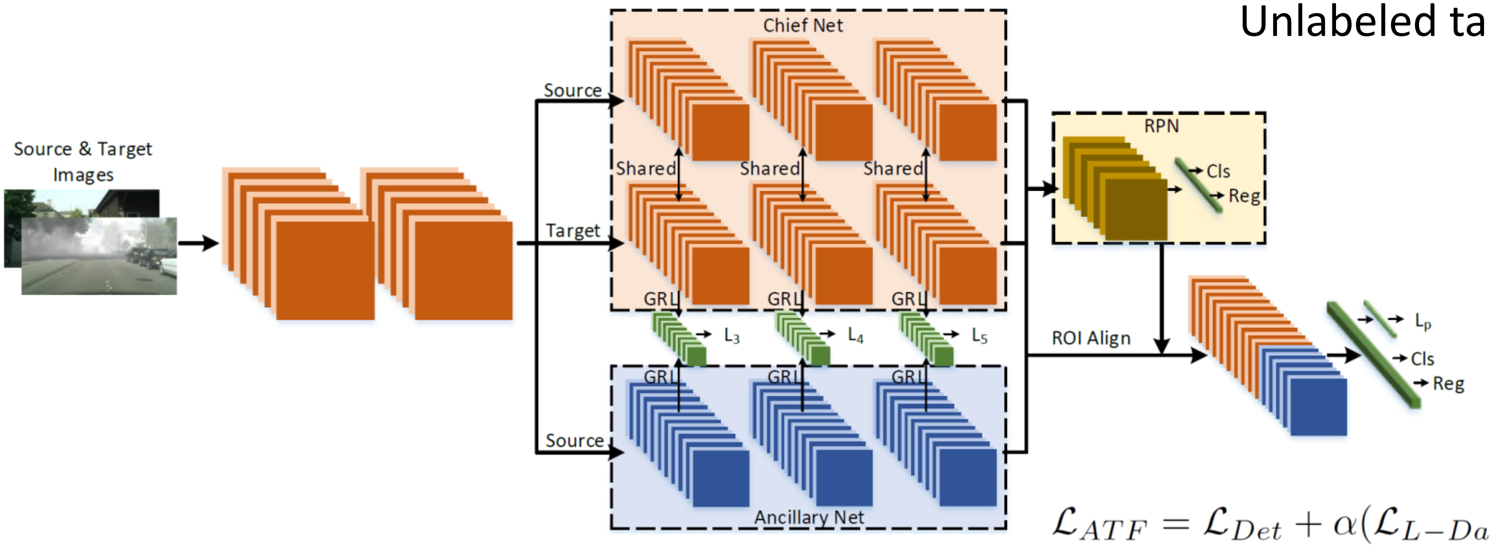
Domain-adaptive Object Detection

ATF model in ECCV'20

Out of control

$$\epsilon_T(h) \leq \hat{\epsilon}_S(h) + \sqrt{\frac{4}{m}\left(d\log\frac{2em}{d} + \log\frac{4}{\delta}\right)} + d_{\mathcal{H}}(\tilde{\mathcal{D}}_S, \tilde{\mathcal{D}}_T) + \boxed{\lambda}$$

Unlabeled target data



$$\mathcal{L}_{ATF} = \mathcal{L}_{Det} + \alpha(\mathcal{L}_{L-Dac} + \sum_{k=3}^{5}\mathcal{L}^{k}_{G-Dac})$$

Z. He and L. Zhang, Domain Adaptive Object Detection via Asymmetric Tri-way Faster-RCNN, ECCV, 2020.

Domain-adaptive Object Detection

Experiments on cross-domain detection

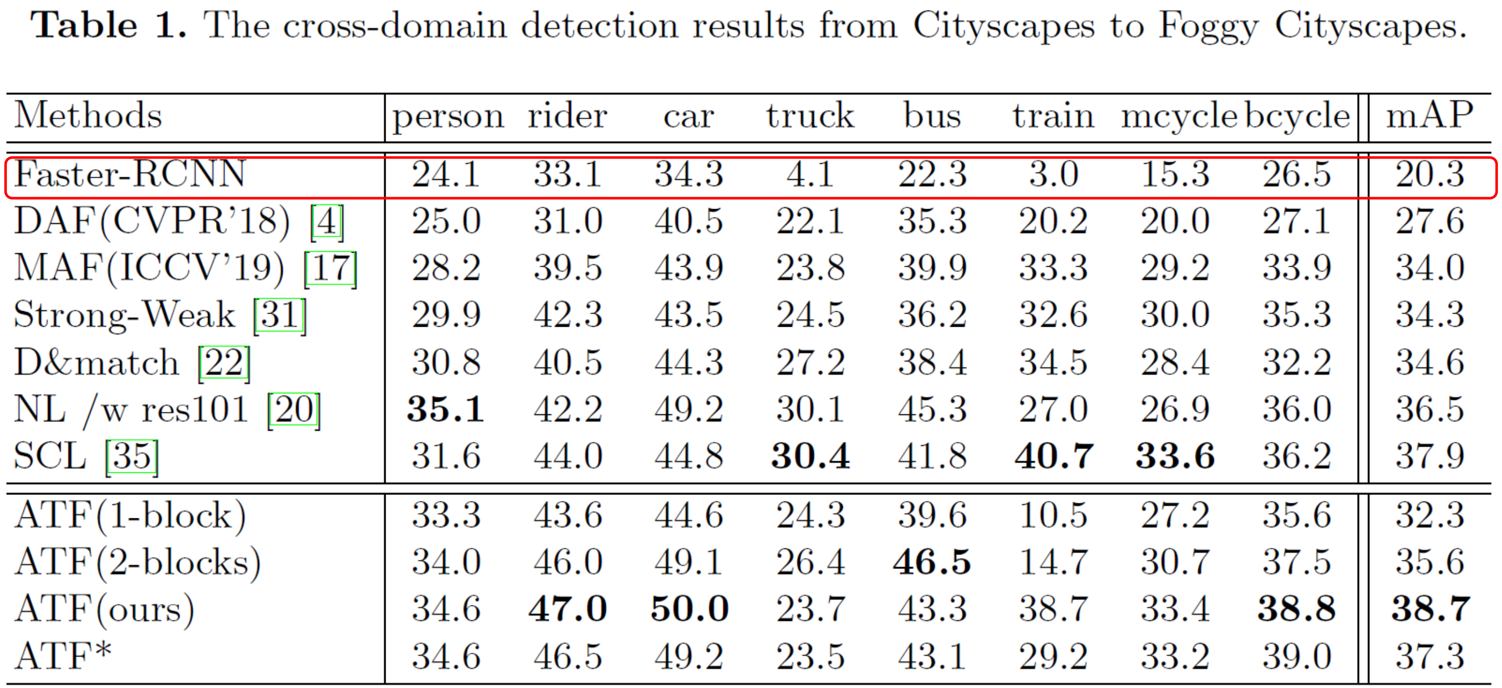


Table 1. The cross-domain detection results from Cityscapes to Foggy Cityscapes.

Methods	person	rider	car	truck	bus	train	mcycle	bcycle	mAP
Faster-RCNN	24.1	33.1	34.3	4.1	22.3	3.0	15.3	26.5	20.3
DAF(CVPR'18) [4]	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
MAF(ICCV'19) [17]	28.2	39.5	43.9	23.8	39.9	33.3	29.2	33.9	34.0
Strong-Weak [31]	29.9	42.3	43.5	24.5	36.2	32.6	30.0	35.3	34.3
D&match [22]	30.8	40.5	44.3	27.2	38.4	34.5	28.4	32.2	34.6
NL /w res101 [20]	**35.1**	42.2	49.2	30.1	45.3	27.0	26.9	36.0	36.5
SCL [35]	31.6	44.0	44.8	**30.4**	41.8	**40.7**	**33.6**	36.2	37.9
ATF(1-block)	33.3	43.6	44.6	24.3	39.6	10.5	27.2	35.6	32.3
ATF(2-blocks)	34.0	46.0	49.1	26.4	**46.5**	14.7	30.7	37.5	35.6
ATF(ours)	34.6	**47.0**	**50.0**	23.7	43.3	38.7	33.4	**38.8**	**38.7**
ATF*	34.6	46.5	49.2	23.5	43.1	29.2	33.2	39.0	37.3

18% improvement with transfer learning

Domain-adaptive Object Detection

Experiments on cross-domain detection

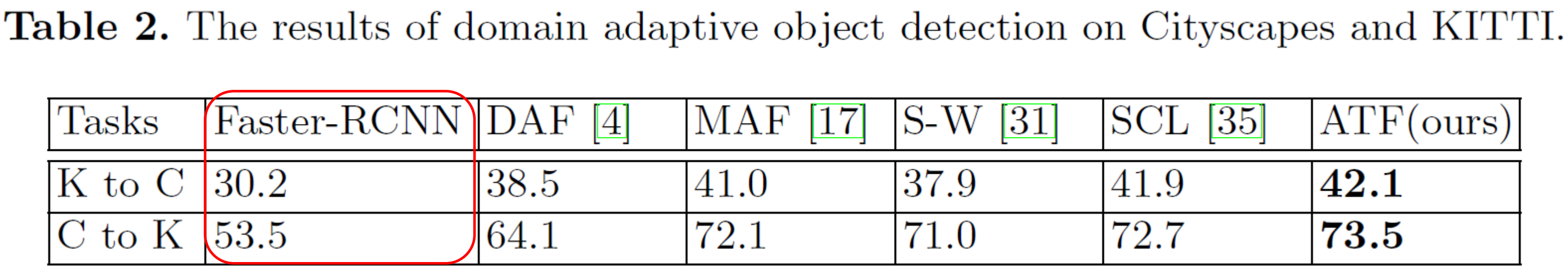


Table 2. The results of domain adaptive object detection on Cityscapes and KITTI.

Tasks	Faster-RCNN	DAF [4]	MAF [17]	S-W [31]	SCL [35]	ATF(ours)
K to C	30.2	38.5	41.0	37.9	41.9	**42.1**
C to K	53.5	64.1	72.1	71.0	72.7	**73.5**

12% and 20% improvement with transfer learning

Domain-adaptive Object Detection

Experiments on cross-domain detection

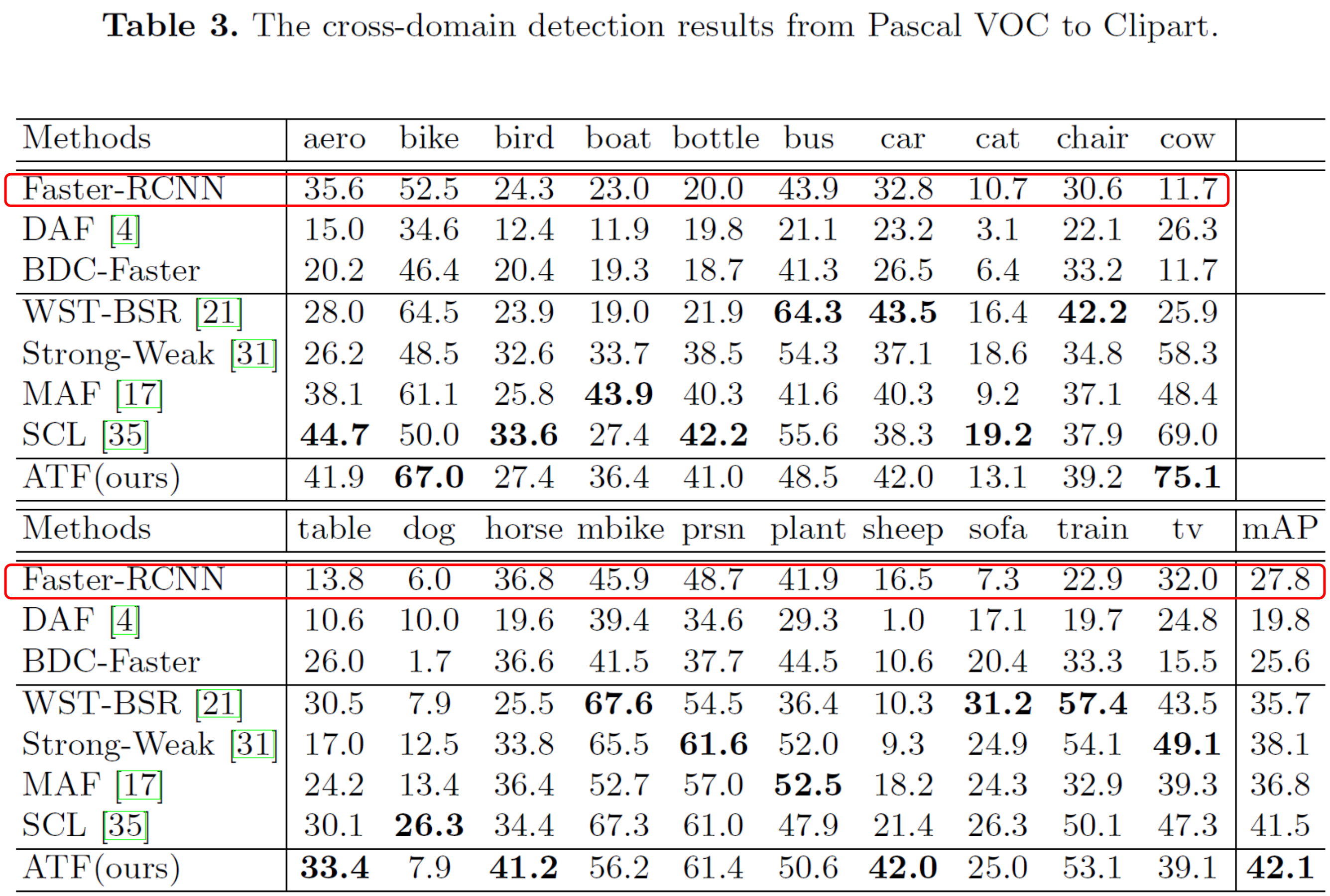


Table 3. The cross-domain detection results from Pascal VOC to Clipart.

Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	prsn	plant	sheep	sofa	train	tv	mAP
Faster-RCNN	35.6	52.5	24.3	23.0	20.0	43.9	32.8	10.7	30.6	11.7	13.8	6.0	36.8	45.9	48.7	41.9	16.5	7.3	22.9	32.0	27.8
DAF [4]	15.0	34.6	12.4	11.9	19.8	21.1	23.2	3.1	22.1	26.3	10.6	10.0	19.6	39.4	34.6	29.3	1.0	17.1	19.7	24.8	19.8
BDC-Faster	20.2	46.4	20.4	19.3	18.7	41.3	26.5	6.4	33.2	11.7	26.0	1.7	36.6	41.5	37.7	44.5	10.6	20.4	33.3	15.5	25.6
WST-BSR [21]	28.0	64.5	23.9	19.0	21.9	**64.3**	**43.5**	16.4	**42.2**	25.9	30.5	7.9	25.5	**67.6**	54.5	36.4	10.3	**31.2**	**57.4**	43.5	35.7
Strong-Weak [31]	26.2	48.5	32.6	33.7	38.5	54.3	37.1	18.6	34.8	58.3	17.0	12.5	33.8	65.5	**61.6**	52.0	9.3	24.9	54.1	**49.1**	38.1
MAF [17]	38.1	61.1	25.8	**43.9**	40.3	41.6	40.3	9.2	37.1	48.4	24.2	13.4	36.4	52.7	57.0	**52.5**	18.2	24.3	32.9	39.3	36.8
SCL [35]	**44.7**	50.0	**33.6**	27.4	**42.2**	55.6	38.3	**19.2**	37.9	69.0	30.1	**26.3**	34.4	67.3	61.0	47.9	21.4	26.3	50.1	47.3	41.5
ATF(ours)	41.9	**67.0**	27.4	36.4	41.0	48.5	42.0	13.1	39.2	**75.1**	**33.4**	7.9	**41.2**	56.2	61.4	50.6	**42.0**	25.0	53.1	39.1	**42.1**

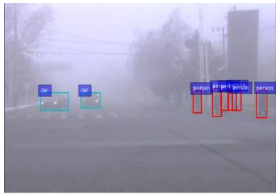
14% improvement with transfer learning

Open issue

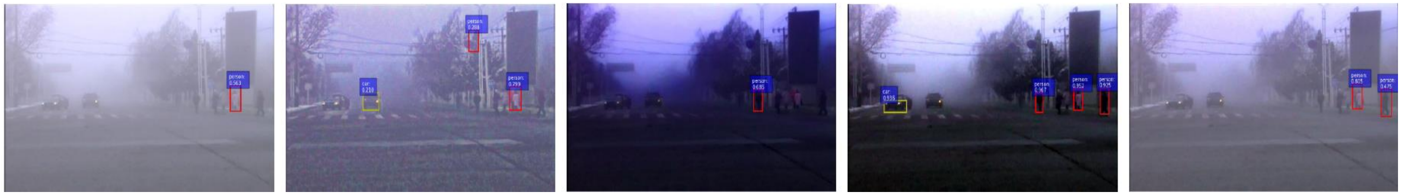
Something You May Concern:

Q: Does degradation removal **HELP** Object Detection and Image Classification (e.g., dehazing)?

View 1: New research finds that dehaze **DOES NOT** help **object detection** and **image classification**. The reason is that dehaze does not add NEW information beneficial to high-level tasks.



Groundtruth



5 dehazing models

B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, Z. Wang, “Benchmarking Single Image Dehazing and Beyond**,”** IEEE Trans. Image Processing, 2019.
Y. Pei, Y. Huang, Q. Zou, X. Zhang, S. Wang, “Effects of Image Degradation and Degradation Removal to CNN-based Image Classification,” IEEE Trans. Patten Analysis and Machine Intelligence, 2019.

Open issue

Something You May Concern:

Q: Does degradation removal **HELP** Object Detection and Image Classification (e.g., dehazing)?

View 2: Adversarial samples.

Panda noise Gibbon (99.3% confidence)

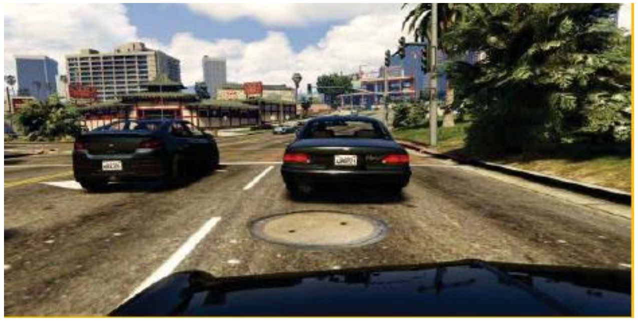
Ekin Dogus Cubuk, Barret Zoph, Samuel Stern Schoenholz, Quoc V. Le, Intriguing Properties of Adversarial Examples

Szegedy et al, Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images

Semantic Segmentation

Lower-level prediction (pixel-level):

- Semantic segmentation needs pixel-level labeling
- Transfer from synthetic domain to real domain.



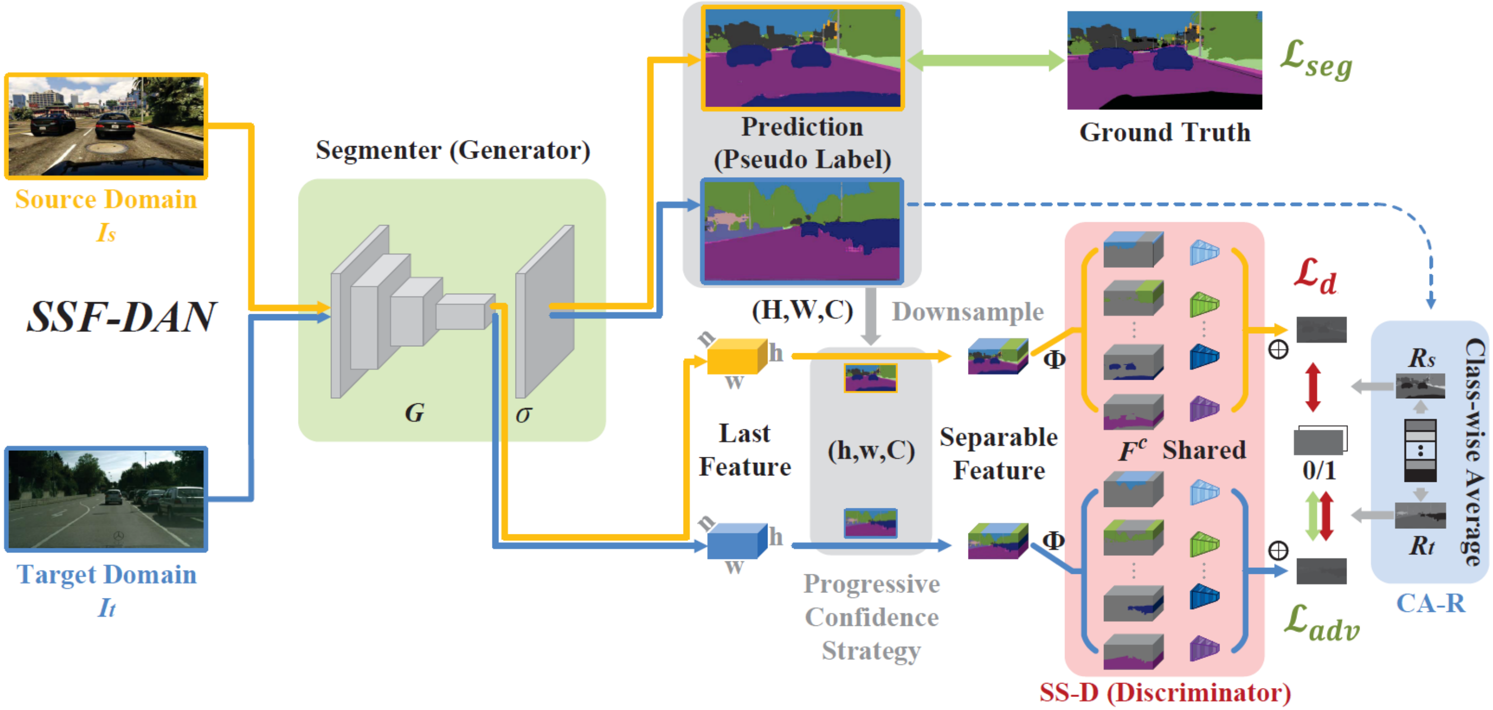
From Computer game

From Cityscapes

Du et al. SSF-DAN: Separated Semantic Feature based Domain Adaptation Network for Semantic Segmentation, ICCV 2019.

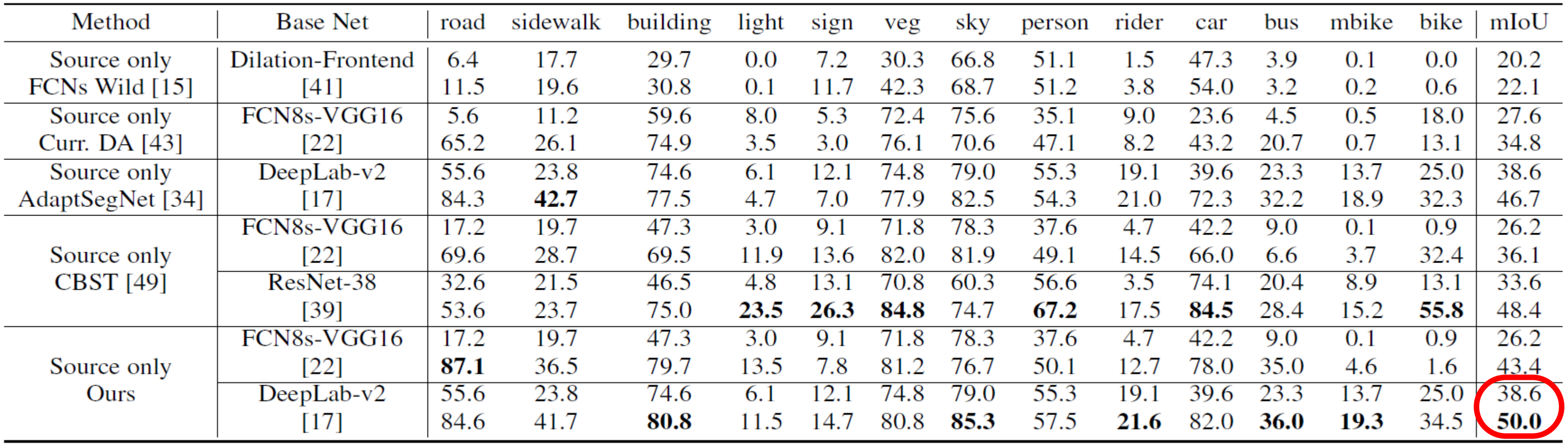
Semantic Segmentation

Domain-adaptive semantic segmentation framework:



- H-divergence based adversarial domain adaptation theory
- GRL minimax optimization

Semantic Segmentation



Method	Base Net	road	sidewalk	building	light	sign	veg	sky	person	rider	car	bus	mbike	bike	mIoU
Source only	Dilation-Frontend	6.4	17.7	29.7	0.0	7.2	30.3	66.8	51.1	1.5	47.3	3.9	0.1	0.0	20.2
FCNs Wild [15]	[41]	11.5	19.6	30.8	0.1	11.7	42.3	68.7	51.2	3.8	54.0	3.2	0.2	0.6	22.1
Source only	FCN8s-VGG16	5.6	11.2	59.6	8.0	5.3	72.4	75.6	35.1	9.0	23.6	4.5	0.5	18.0	27.6
Curr. DA [43]	[22]	65.2	26.1	74.9	3.5	3.0	76.1	70.6	47.1	8.2	43.2	20.7	0.7	13.1	34.8
Source only	DeepLab-v2	55.6	23.8	74.6	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	38.6
AdaptSegNet [34]	[17]	84.3	**42.7**	77.5	4.7	7.0	77.9	82.5	54.3	21.0	72.3	32.2	18.9	32.3	46.7
	FCN8s-VGG16	17.2	19.7	47.3	3.0	9.1	71.8	78.3	37.6	4.7	42.2	9.0	0.1	0.9	26.2
Source only	[22]	69.6	28.7	69.5	11.9	13.6	82.0	81.9	49.1	14.5	66.0	6.6	3.7	32.4	36.1
CBST [49]	ResNet-38	32.6	21.5	46.5	4.8	13.1	70.8	60.3	56.6	3.5	74.1	20.4	8.9	13.1	33.6
	[39]	53.6	23.7	75.0	**23.5**	**26.3**	**84.8**	74.7	**67.2**	17.5	**84.5**	28.4	15.2	**55.8**	48.4
	FCN8s-VGG16	17.2	19.7	47.3	3.0	9.1	71.8	78.3	37.6	4.7	42.2	9.0	0.1	0.9	26.2
Source only	[22]	**87.1**	36.5	79.7	13.5	7.8	81.2	76.7	50.1	12.7	78.0	35.0	4.6	1.6	43.4
Ours	DeepLab-v2	55.6	23.8	74.6	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	38.6
	[17]	84.6	41.7	**80.8**	11.5	14.7	80.8	**85.3**	57.5	**21.6**	82.0	**36.0**	**19.3**	34.5	**50.0**

12%

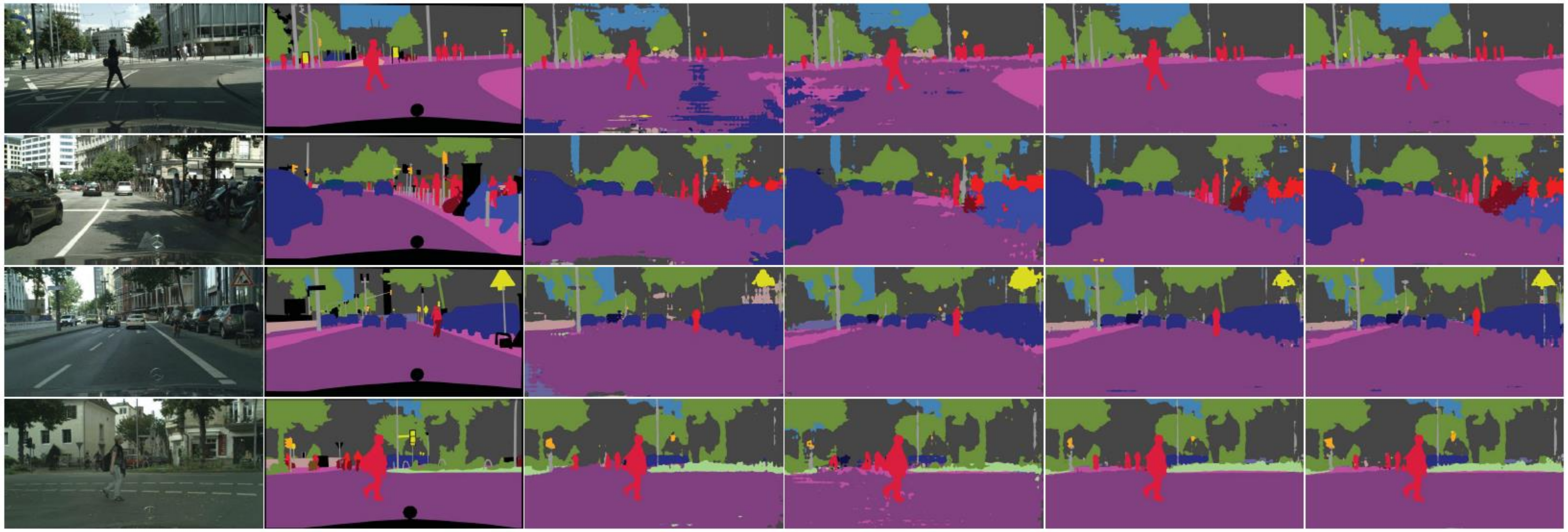
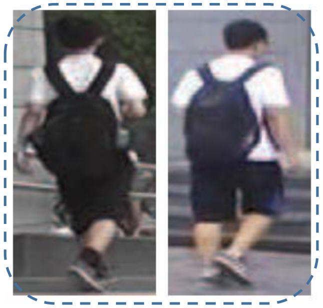


Image Retrieval

A classical similarity match task:

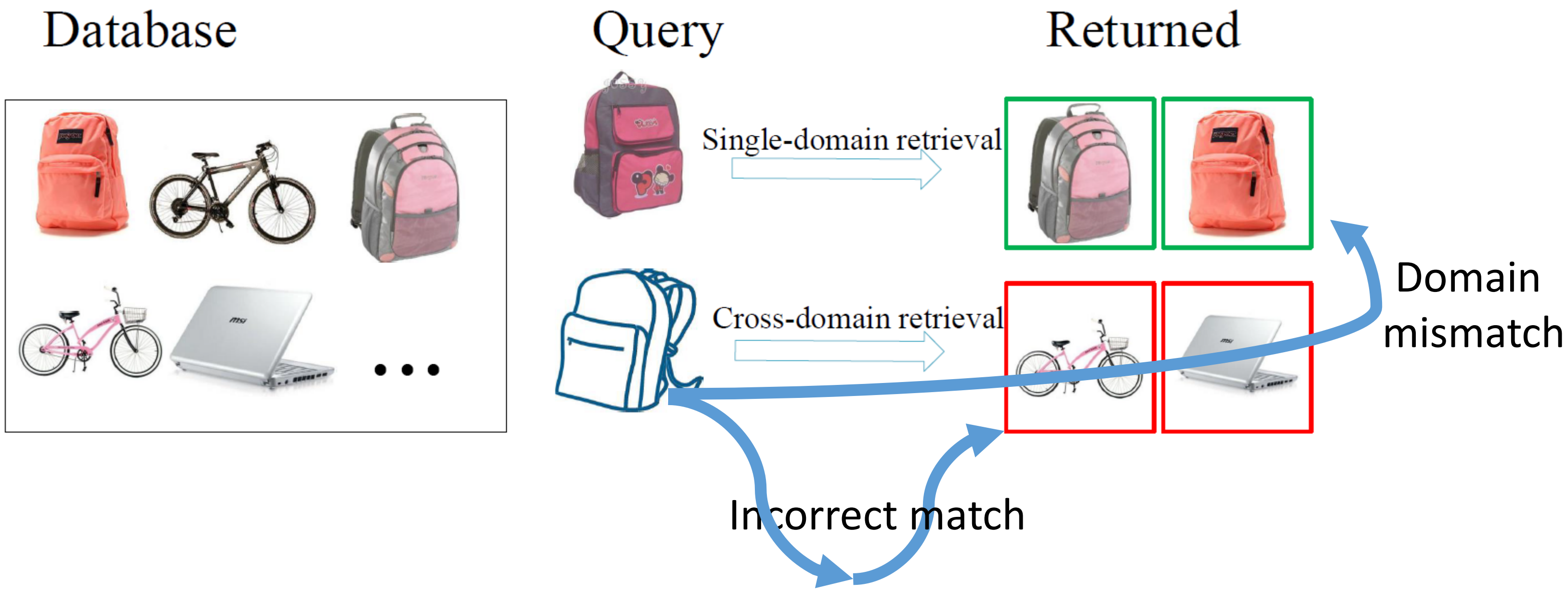
- Cross-modal retrieval (text--image)
- Hashing retrieval (image--image)
- Person Re-identification (person--person)
- Person Search (image--person)



F. Huang, L. Zhang, Probability Weighted Compact Feature for Domain Adaptive Retrieval, CVPR 2020.
J. Liu, L. Zhang, Optimal Projection Guided Transfer Hashing for Image Retrieval, AAAI 2019.

Domain-adaptive Image Retrieval

Consider a cross-domain retrieval problem:



F. Huang, L. Zhang, Probability Weighted Compact Feature for Domain Adaptive Retrieval, CVPR 2020.

Domain-adaptive Image Retrieval

Probability Weighted Compact Feature Learning:

- Hashing for compact feature (binary feature of low memory)
- Focal triplet hashing in Bayesian perspective (BP)
- The posterior probability of compact feature f_i,f_j,f_k w.r.t. to x_i,x_j,x_k, given their similarity s_{ij} and s_{ik}

$$\prod_{i,j,k \in \mathbf{X}} p\left(f_i, f_j, f_k | s_{ij}, s_{ik}\right) \Leftrightarrow$$

$$\prod_{i,j,k \in \mathbf{X}} p\left(s_{ij}, s_{ik} | f_i, f_j, f_k\right) p\left(f_i\right) p\left(f_j\right) p\left(f_k\right)$$

F. Huang, L. Zhang, Probability Weighted Compact Feature for Domain Adaptive Retrieval, CVPR 2020.

Domain-adaptive Image Retrieval

Probability Weighted Compact Feature Learning:

- Hashing for compact feature (binary feature of low memory)
- Focal triplet hashing in Bayesian perspective (BP)
- The posterior probability of compact feature f_i,f_j,f_k w.r.t. to x_i,x_j,x_k, given their similarity s_{ij} and s_{ik}

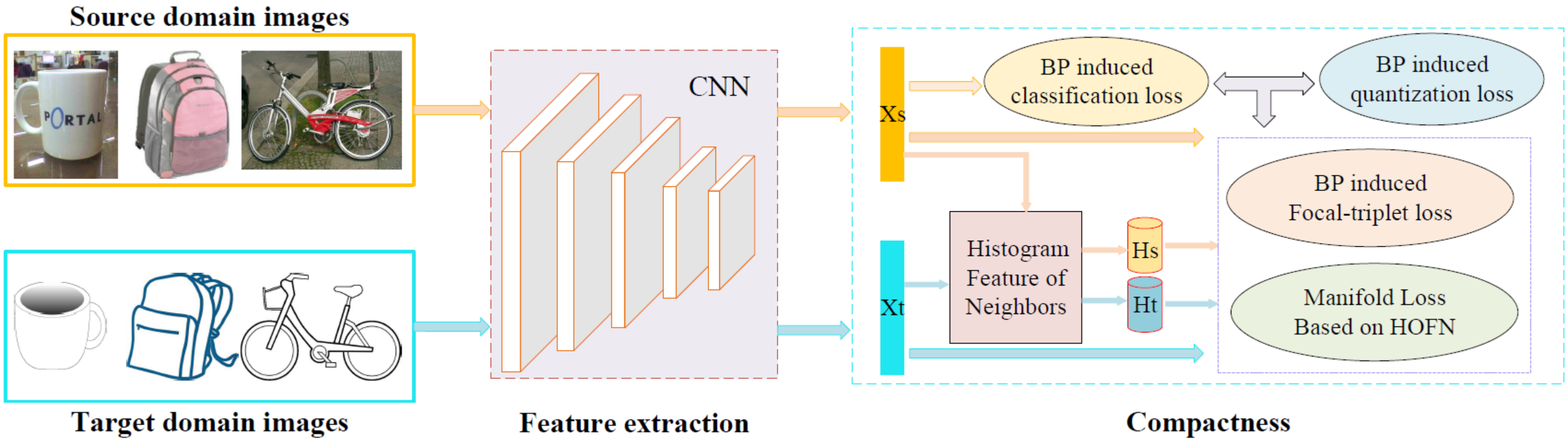
$$\max \sum_{i,j,k \in \mathbf{X}} \omega_{ijk} \log p\left(s_{ij}, s_{ik} | f_i, f_j, f_k\right) + \sum_{i \in \mathbf{X}} \log p\left(f_i\right) + \sum_{j \in \mathbf{X}} \log p\left(f_j\right) + \sum_{k \in \mathbf{X}} \log p\left(f_k\right)$$

- p(.) is sampled from a Gaussian distribution

$$p\left(f_i\right) = e^{-\theta d(\mathbf{b}_i, f_i)} \cdot e^{-\lambda_1 d(\mathbf{y}_i, \mathbf{C}^\top \mathbf{b}_i)} \cdot e^{-\lambda_2 \|\mathbf{C}\|^2}$$

Domain-adaptive Image Retrieval

Probability Weighted Compact Feature Learning:


Source domain images
PORTAL
CNN
Xs
BP induced classification loss
BP induced quantization loss
BP induced Focal-triplet loss
Histogram Feature of Neighbors
Hs
Ht
Xt
Manifold Loss Based on HOFN
Target domain images
Feature extraction
Compactness

$$\min_{\mathbf{W},\mathbf{C},\mathbf{B_t},\mathbf{B_s}} \mathcal{T}ri + \theta\mathcal{Q} + \lambda_1\mathcal{C} + \lambda_2\|\mathbf{C}\|^2 + \lambda_3\mathcal{M}$$

$$s.t. \mathbf{W}^\top\mathbf{W} = \mathbf{I}, \mathbf{B_t} = sgn(\mathbf{W}^\top\mathbf{X_t}), \mathbf{B}_s = sgn(\mathbf{W}^\top\mathbf{X}_s)$$

Domain-adaptive Image Retrieval

Experiments on cross-domain retrieval:

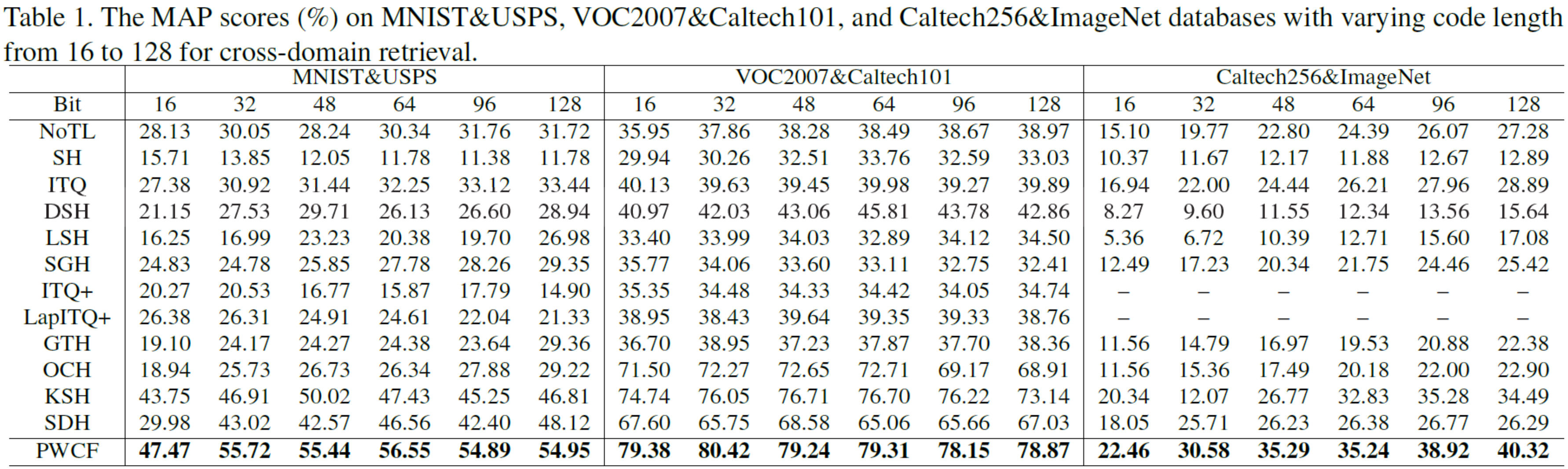


Table 1. The MAP scores (%) on MNIST&USPS, VOC2007&Caltech101, and Caltech256&ImageNet databases with varying code length from 16 to 128 for cross-domain retrieval.

	MNIST&USPS						VOC2007&Caltech101						Caltech256&ImageNet					
Bit	16	32	48	64	96	128	16	32	48	64	96	128	16	32	48	64	96	128
NoTL	28.13	30.05	28.24	30.34	31.76	31.72	35.95	37.86	38.28	38.49	38.67	38.97	15.10	19.77	22.80	24.39	26.07	27.28
SH	15.71	13.85	12.05	11.78	11.38	11.78	29.94	30.26	32.51	33.76	32.59	33.03	10.37	11.67	12.17	11.88	12.67	12.89
ITQ	27.38	30.92	31.44	32.25	33.12	33.44	40.13	39.63	39.45	39.98	39.27	39.89	16.94	22.00	24.44	26.21	27.96	28.89
DSH	21.15	27.53	29.71	26.13	26.60	28.94	40.97	42.03	43.06	45.81	43.78	42.86	8.27	9.60	11.55	12.34	13.56	15.64
LSH	16.25	16.99	23.23	20.38	19.70	26.98	33.40	33.99	34.03	32.89	34.12	34.50	5.36	6.72	10.39	12.71	15.60	17.08
SGH	24.83	24.78	25.85	27.78	28.26	29.35	35.77	34.06	33.60	33.11	32.75	32.41	12.49	17.23	20.34	21.75	24.46	25.42
ITQ+	20.27	20.53	16.77	15.87	17.79	14.90	35.35	34.48	34.33	34.42	34.05	34.74	–	–	–	–	–	–
LapITQ+	26.38	26.31	24.91	24.61	22.04	21.33	38.95	38.43	39.64	39.35	39.33	38.76	–	–	–	–	–	–
GTH	19.10	24.17	24.27	24.38	23.64	29.36	36.70	38.95	37.23	37.87	37.70	38.36	11.56	14.79	16.97	19.53	20.88	22.38
OCH	18.94	25.73	26.73	26.34	27.88	29.22	71.50	72.27	72.65	72.71	69.17	68.91	11.56	15.36	17.49	20.18	22.00	22.90
KSH	43.75	46.91	50.02	47.43	45.25	46.81	74.74	76.05	76.71	76.70	76.22	73.14	20.34	12.07	26.77	32.83	35.28	34.49
SDH	29.98	43.02	42.57	46.56	42.40	48.12	67.60	65.75	68.58	65.06	65.66	67.03	18.05	25.71	26.23	26.38	26.77	26.29
PWCF	**47.47**	**55.72**	**55.44**	**56.55**	**54.89**	**54.95**	**79.38**	**80.42**	**79.24**	**79.31**	**78.15**	**78.87**	**22.46**	**30.58**	**35.29**	**35.24**	**38.92**	**40.32**

Significant improvement with domain adaptation

Contents

- Part I: Background and Preliminary
- Part II: Concept, Theory and Algorithms
- Part III: Applications of TL/DA Algorithms
- Summary

Summary vs. Future

In this talk, a systematic introduction of transfer learning and domain adaptation in background, theory, algorithms and applications.

- Statistical machine learning is conditional on i.i.d. distribution
- Ben-David proves an upper bound of domain adaptation
- Deep learning triggers the progress of transfer learning
- Transfer learning is showing a bloom suitation
- A wide spread of applications in many fields

Summary vs. Future

In AI era, researchers are exploring universal techniques.

- Vision perception in open environment
- Natural language processing in translation and interaction.
- Brain like learning
- From perception to cognition
- Causality and reasoning
- Attack and defense
- AI ethics connecting society (privacy, safety, etc.)
- …

Thank you

LiVE Group

视觉智能与学习研究中心