Domain Adaptive Faster R-CNN for Object Detection in the Wild

Yuhua Chen, Wen Li,
Christos Sakaridis, De ngxin Dai, Luc Van Gool

Computer Vision Lab, ETH Zurich, VISICS, ESAT/PSI, KU Leuven

Presented by Chaofan Tao

March 21, 2019

Outline

1. Introduction
2. Related Work
   - Object Detection
   - Domain Adaptation
3. Preliminaries
   - Faster RCNN
   - Distribution Alignment with H-divergence
4. Method
   - Image-Level & Instance-Level & Consistency Regularization
   - Proposed Objective Functions
5. Experiments
   - Experimental Results
   - Ablation Study
6. Conclusion & Remarks

Introduction

Background

1. The problem of domain shift in object detection.

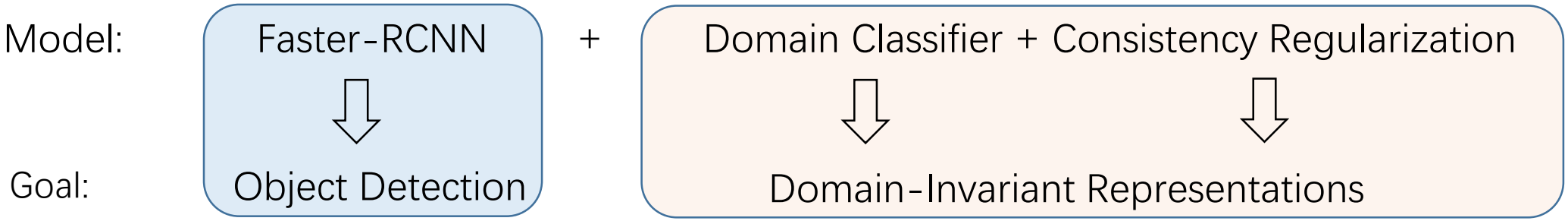
   (*viewpoints, object appearance, backgrounds, illumination, image quality, etc.*)

2. Annotating bounding boxes is expensive time-consuming.

Illustration of different datasets for autonomous driving.
Contained Dataset (from top to bottom-right)
KITTI, Cityscapes, Foggy Cityscapes, SIM10K

It is challenging to apply an object detection model learned from one domain to another domain !

Introduction



Main Contribution

1. Address this problem of object detection in the scenario of **unsupervised domain adaptation**.
2. Provide a **theoretical analysis** of the proposed model from a probabilistic perspective.
3. We integrate the proposed components into the Faster R-CNN model, and the resulting system can be trained in an **end-to-end** manner.

Related Work

Object Detection

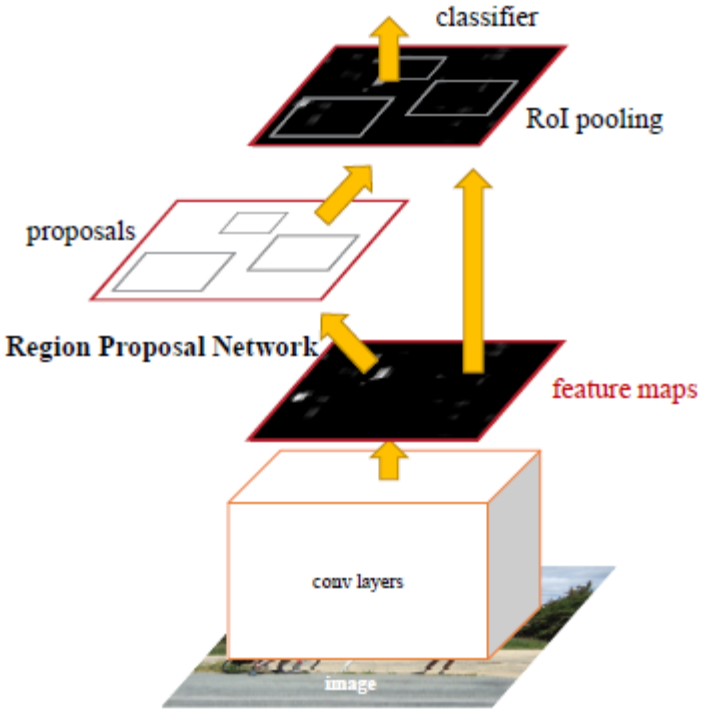
1. Fast R-CNN proposed **ROI-pooling** layer after the last convolutional layer; Integrate **multi-task loss** with CNN instead of using SVMs.
2. Instead of selective search, Faster R-CNN produces object proposals with a **Region Proposal Network** (RPN) that simultaneously predicts bounding boxes and objectness scores.

Domain Adaptation

1. Domain adaptation has been widely studies for image classification in computer vision.
   kernel learning, metric learning, subspace learning, geodesic flow kernel, covariance matrix alignment etc.
2. Much less attention has been paid to domain adaptation for other tasks.
   semantic segmentation, object detection, learn detectors from videos/3D models/synthetic models , etc.

Preliminaries

Faster-RCNN



Based on the feature maps, Region Proposal Network (RPN) generates candidate object proposals

RoI-wise classifier predicts the category label from a feature vector obtained using RoI-pooling

Detailed architecture about RPN and training method is available in "Faster R-CNN Towards Real-Time Object Detection with Region Proposal Networks", NIPS, 2015

Preliminaries

Distribution Alignment with H-divergence

We use h: x ⟶ {0, 1} denotes domain classifier, then the discrepancy between two classifiers is:

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

Goal: make the domain classifier confuse about the domain of input.

If the error is high for the best domain classifier, the two domains are hard to distinguish, so they are close to each other, and vice versa.

Detailed training method and theoretical insights about relation between domain classifier and upper bound is available in "Unsupervised Domain Adaptation by Backpropagation", ICML, 2015

Method

Framework

Model ~ Faster-RCNN + Domain Classifier + Consistency Regularization

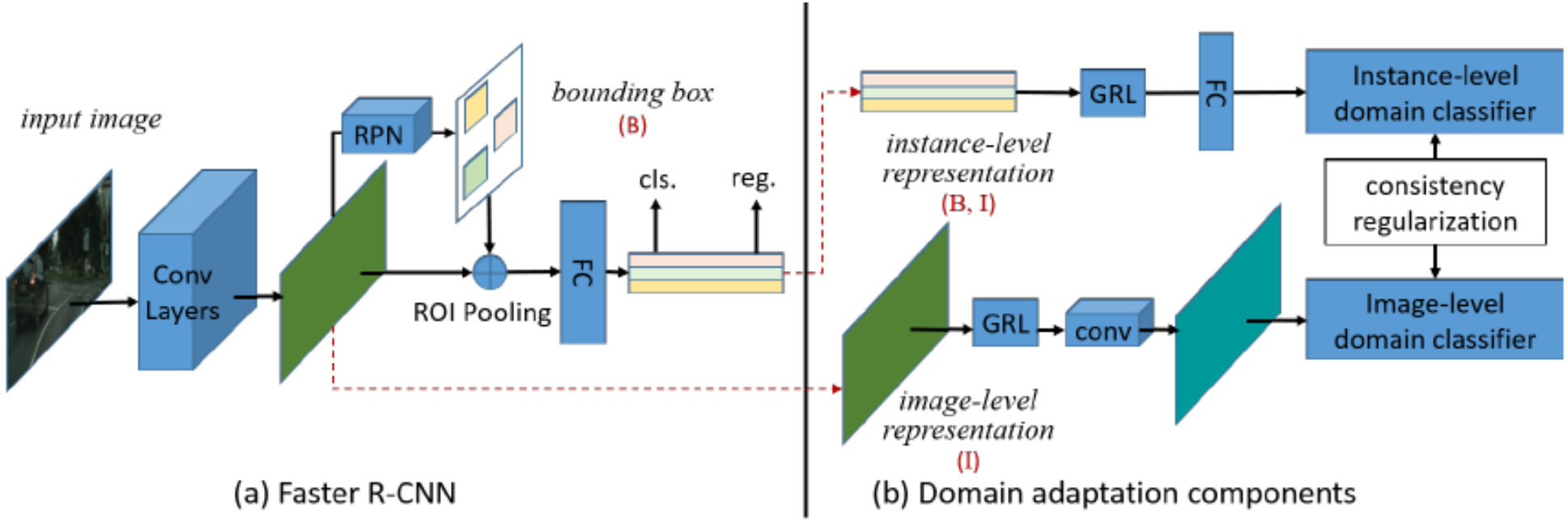


Figure 2. **An overview of our Domain Adaptive Faster R-CNN model:** we tackle the domain shift on two levels, the image level and the instance level. A domain classifier is built on each level, trained in an adversarial training manner. A consistency regularizer is incorporated within these two classifiers to learn a domain-invariant RPN for the Faster R-CNN model.

Method

C: category B: bounding box
I: image D: domain label

Domain shift: $P_{\mathcal{S}}(C,B,I) \neq P_{\mathcal{T}}(C,B,I)$.

Image-Level Adaptation

$$P(C,B,I) = P(C,B|I)P(I).$$

Assuming that domain shift is caused by the difference on the marginal distribution P(I).

We enforce $P_{\mathcal{S}}(I) = P_{\mathcal{T}}(I)$ by pixel-wise binary prediction with the following objective function:

$$\mathcal{L}_{img} = -\sum_{i,u,v}\left[D_i \log p_i^{(u,v)} + (1-D_i)\log(1-p_i^{(u,v)})\right].$$

the local of (u,v) on the feature map and i-th image

Instance-Level Adaptation

$$P(C,B,I) = P(C|B,I)P(B,I).$$

Assuming that domain shift is caused by the difference on the marginal distribution $P(B,I) = P(B|I)P(I)$

We enforce $P_{\mathcal{S}}(B,I) = P_{\mathcal{T}}(B,I)$ by binary prediction on the RoI-pooled feature with the following objective function:

$$\mathcal{L}_{ins} = -\sum_{i,j}\left[D_i \log p_{i,j} + (1-D_i)\log(1-p_{i,j})\right].$$

the j-th region proposal and i-th image

Method

Consistency Regularization

Goal: alleviate the bias in estimating $P(B|I)$. By using the Bayes' theorem, we obtain:

$$P(D|B,I)P(B|I) = P(B|D,I)P(D|I).$$

P(B | I) is a **domain-invariant** bounding box predictor, and P(B | D,I) is a **domain-dependent** bounding box predictor. Since bounding box for target is inaccessible, we **enforce the consistency between two domain classifiers**, i.e., P(D | B, I) = P(D | I), we could learn P(B | D, I) to approach P(B | I) by the following objective function:

$$L_{cst} = \sum_{i,j} \| \frac{1}{|I|} \sum_{u,v} p_i^{(u,v)} - p_{i,j} \|_2,$$

Final objective function: $L = L_{det} + \lambda(L_{img} + L_{ins} + L_{cst})$

Testing phase: use Faster-RCNN only

Experiments

Experimental Results

SIM 10k -> Cityscapes

	img	ins	cons	car AP
Faster R-CNN				30.12
Ours	✓			33.03
		✓		35.79
	✓	✓		37.86
	✓	✓	✓	**38.97**

Table 1. The average precision (AP) of *Car* on the *Cityscapes* validation set. The models are trained using the *SIM 10k* dataset as the source domain and the *Cityscapes* training set as the target domain. *img* is short for *image-level alignment*, *ins* for *instance-level alignment* and *cons* is short for our *consistency loss*

1. Only *car* is reported since *car* is the only category annotated in both dataset.

2. Note that the *Cityscapes* dataset is not dedicated to detection, thus we take the tightest rectangles of its instance masks as ground-truth bounding boxes.

Experiments

Experimental Results

Cityscapes -> Foggy Cityscapes

	img	ins	cons	person	rider	car	truck	bus	train	mcycle	bicycle	mAP
Faster R-CNN				17.8	23.6	27.1	11.9	23.8	9.1	14.4	22.8	18.8
Ours	✓			22.9	30.7	39.0	20.1	27.5	17.7	21.4	25.9	25.7
		✓		23.6	30.6	38.6	20.8	**40.5**	12.8	17.1	26.1	26.3
	✓	✓		24.2	**31.2**	39.1	19.1	36.2	19.2	17.1	27.0	26.6
	✓	✓	✓	**25.0**	31.0	**40.5**	**22.1**	35.3	**20.2**	**20.0**	**27.1**	**27.6**

Table 2. Quantitative results on the *Foggy Cityscapes* validation set, models are trained on the *Cityscapes* training set.

KITTI -> Cityscapes & Cityscapes -> KITTI

	img	ins	cons	K → C	C → K
Faster R-CNN				30.2	53.5
Ours	✓			36.6	60.9
		✓		34.6	57.6
	✓	✓		37.3	62.7
	✓	✓	✓	**38.5**	**64.1**

Table 3. Quantitative analysis of adaptation result between *KITTI* and *Cityscapes* . We report AP of *Car* on both directions. *e.g.* K → C and C → K.

The improvement **generalizes well across different categories**, which suggests that the proposed technique can also reduce domain discrepancy across different object classes

Experiments

Ablation Study: does each individual improve the performance of detection?

KITTI → Cityscapes

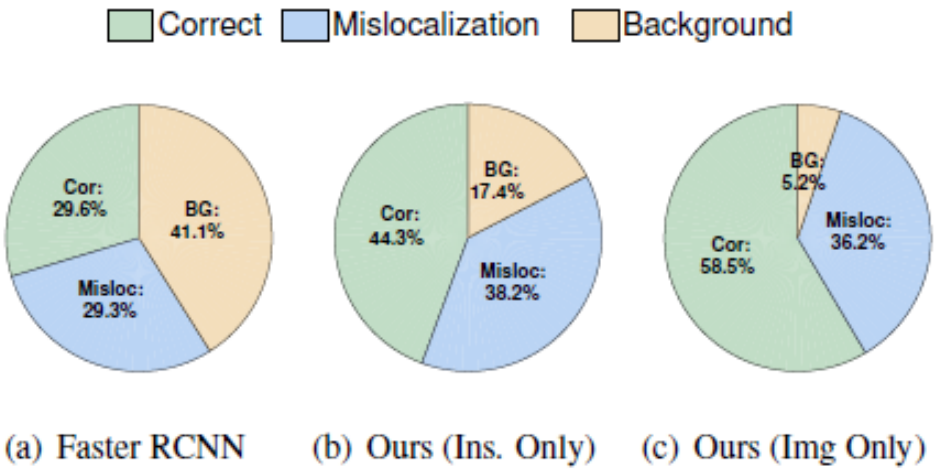


Figure 3. Error Analysis of Top Ranked Detections

We select 20, 000 predictions with highest confidence for the **Faster R-CNN model**, our model with **only image-level adaptation**, and our model with **only instance-level adaptation**, respectively

Correct: detection has an overlap > 0.5 with ground-truth.
Mis-localized: detection has a overlap with ground-truth of 0.3 to 0.5,
Background: detection has an overlap < 0.3

Discussion

1. Both *Ins. Only* and *Img. Only* improves the performance of detection.
2. In comparison, *Ins. Only* gives higher background error than *Img. Only*.

Experiments

Ablation Study: does the target scale matters?

KITTI → Cityscapes

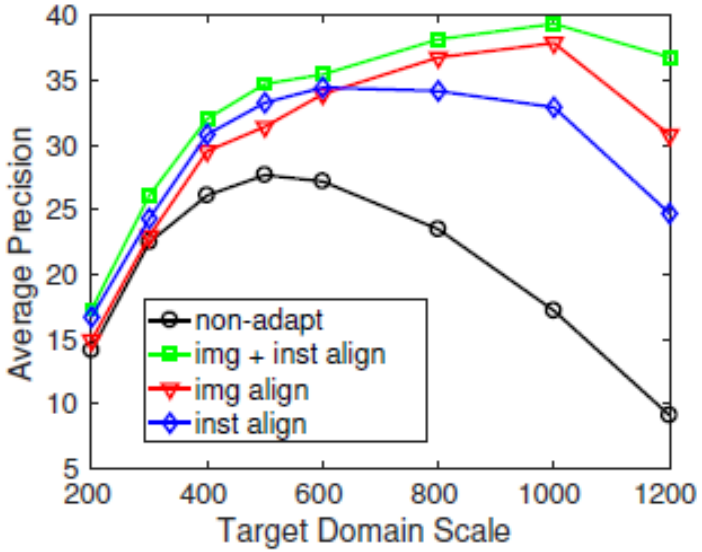


Figure 4. **AP at different scales**: Source images from *KITTI* are fixed at a scale of 500 pixels, and we resize the target images from *Cityscapes* to different scales.

we refer to **the shorter length of an image** as its scale.

Q: Why average precision drops with the variation of scale?
A: the scale change is a **global transformation**, which affects all instances and background.

Q: Why our model is robust against the variation of scale?
A: Global domain shift is mainly tackled by image-level alignment, and instance-level alignment is used to minimize instance-level discrepancy. **Using both always yields the best results** across all scales.

Experiments

Ablation Study: Are the consistency regularization beneficial?

KITTI → Cityscapes

	Faster R-CNN	Ours(w/o)	Ours
mIoU	18.8	28.5	**30.3**

Table 4. Mean best Overlap between with groundtruth bounding boxes by top 300 proposals from RPN in different models, in which Ours(w/o) denotes our model without using consistency regularization.

The maximum achievable mean overlap (mIoU) between the **top 300 proposals** from RPN and the **ground-truth** is used for measuring the benefit of consistency regularization .

Discussion

1. Our model improves Faster R-CNN due to the use of image-level and instance-level adaptation.
2. Consistency regularizer encourages the RPN to be more robust.

Conclusion

1. Based on our theoretical analysis for cross-domain object detection, we propose an **image-level adaptation component** and an **instance-level component** to alleviate the performance drop caused by domain shift.
2. Our model can be trained end-to-end and is validated on various domain shift scenarios, the adaptive method outperforms baseline Faster R-CNN by a clear margin.

Remark

1. This is the first work proposed to improve Faster R-CNN for cross-domain object detection.
2. If this work is extended to detect unseen category rather than the shared category in different domains (e.g., *car* in SIM 10k -> Cityscapes), this model is more of utility value.
3. An intuitive improvement may be obtained by integrating MMD-based regularization (e.g. JAN) into image-level and instance-level feature.

Q & A