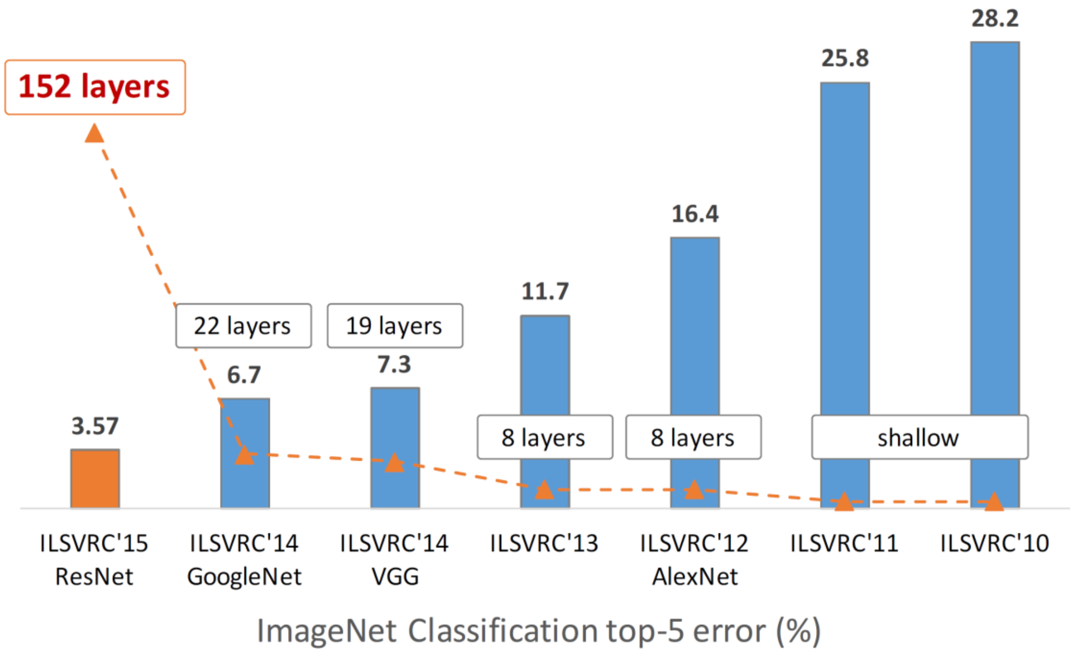
Towards Weakly-Supervised Visual Understanding

Zhiding Yu   Learning & Perception Research, NVIDIA

zhidingy@nvidia.com

Introduction

The Benefit of Big Data and Computation Power



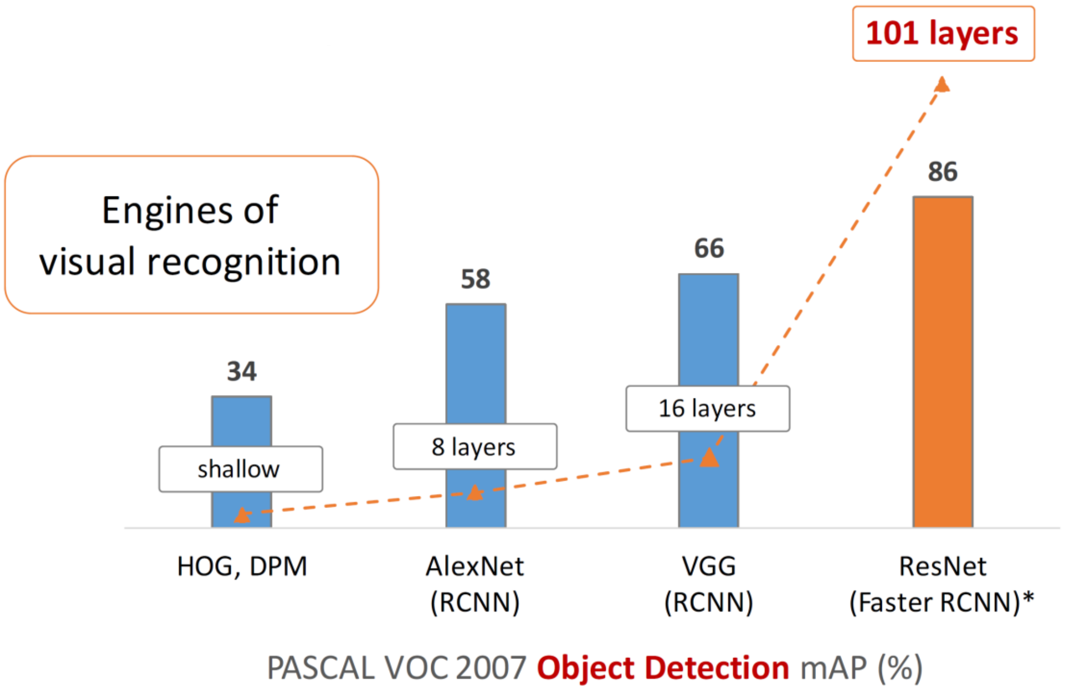


Figure credit: Kaiming He et al., Deep Residual Learning for Image Recognition, CVPR16

Beyond Supervised Learning

Reinforcement Learning
(Cherry)

Supervised Learning
(Icing)

Unsupervised Learning
(Cake)

“The revolution will not be supervised!”

— Alyosha Efros

“If intelligence was a cake, unsupervised learning would be the cake, supervised learning would be the icing on the cake, and reinforcement learning would be the cherry on the cake.” — Yann LeCun

Weakly-Supervised Learning

Image credit: https://firstbook.org/blog/2016/03/11/teaching-much-more-than-basic-concepts/

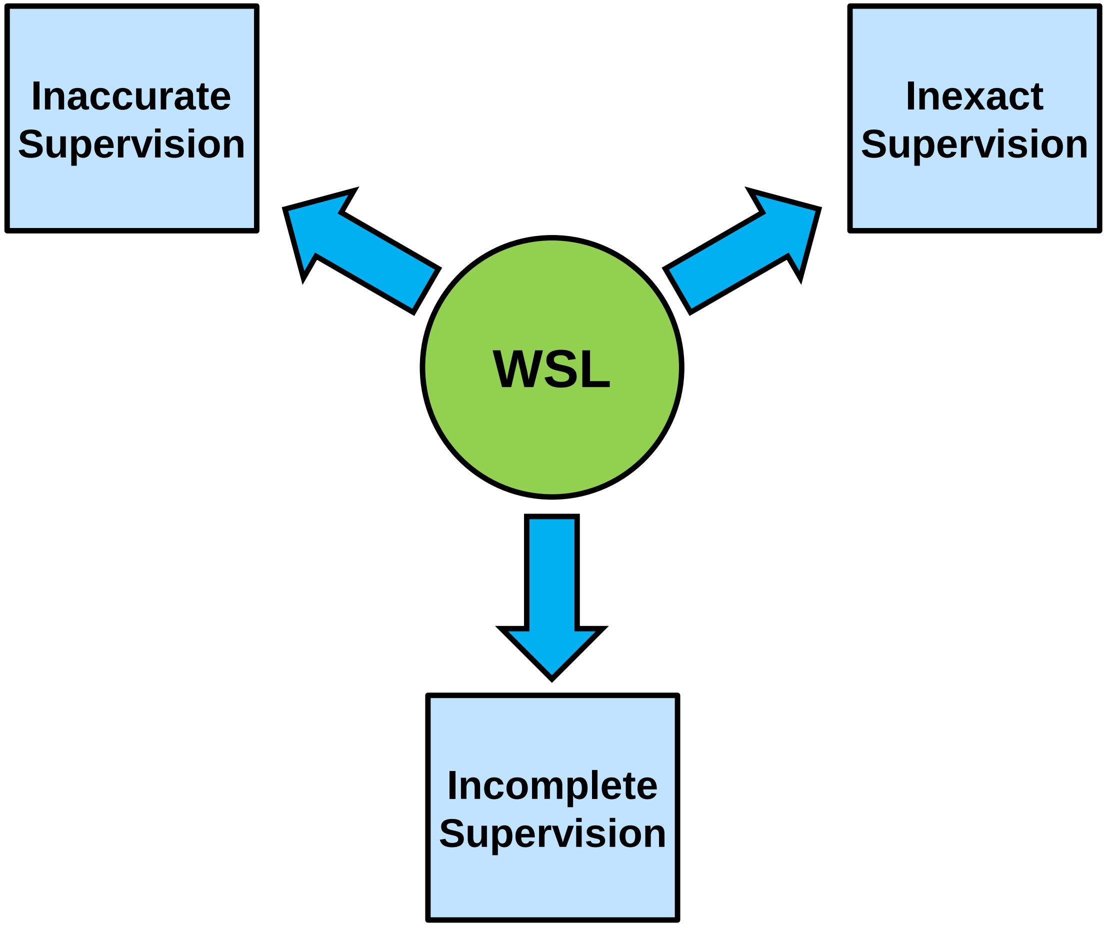
From Research Perspective

- Similar to how human learns to understand the world
- Good support for “continuous learning”

From Application Perspective

- Good middle ground between unsupervised learning and supervised learning
- Potential to accommodate labels in diverse forms
- Scalable to much larger amount of data

Weakly-Supervised Learning



Weakly-Supervised Learning

Inaccurate Supervision

- Wrong/misaligned labels
- Ambiguities
- Noisy label learning

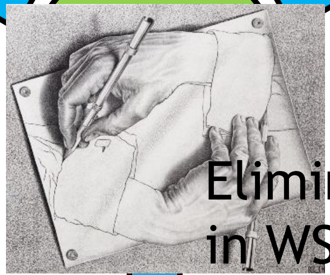
Inexact Supervision

- Seg/Det with cls label/bbox/point
- Multiple instance learning
- Attention models

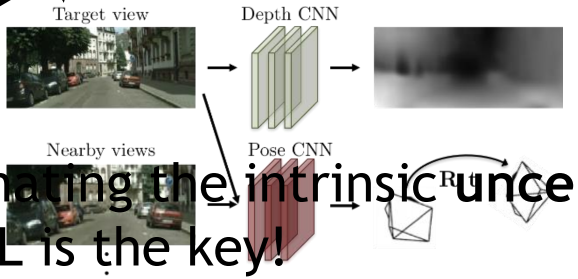
Incomplete Supervision

- Semi-supervised learning
- Teacher-student models
- Domain adaptation

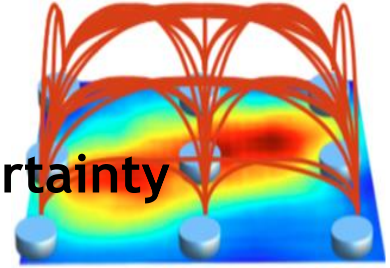
Eliminating the intrinsic uncertainty in WSL is the key!



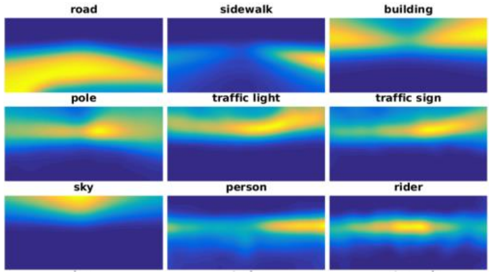
Self-supervision



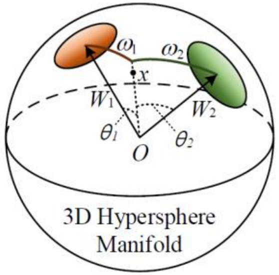
Meta-supervision



Structured info



Domain prior



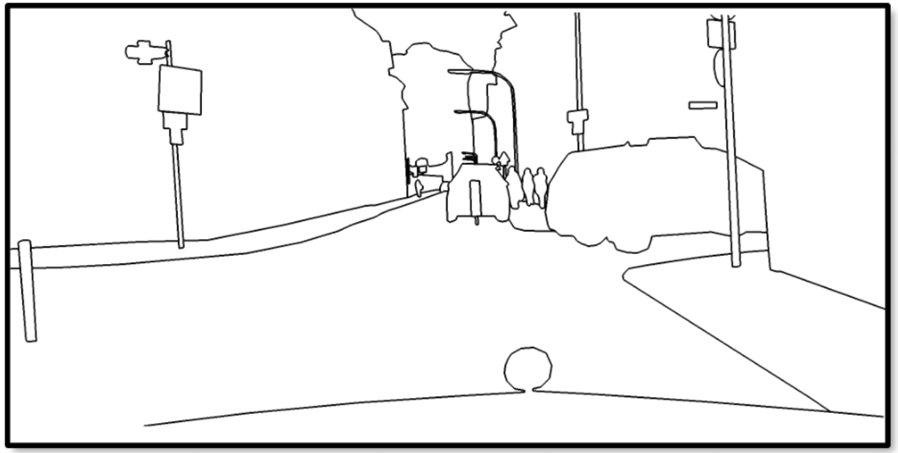
Normalization

Learning with Inaccurate Supervision

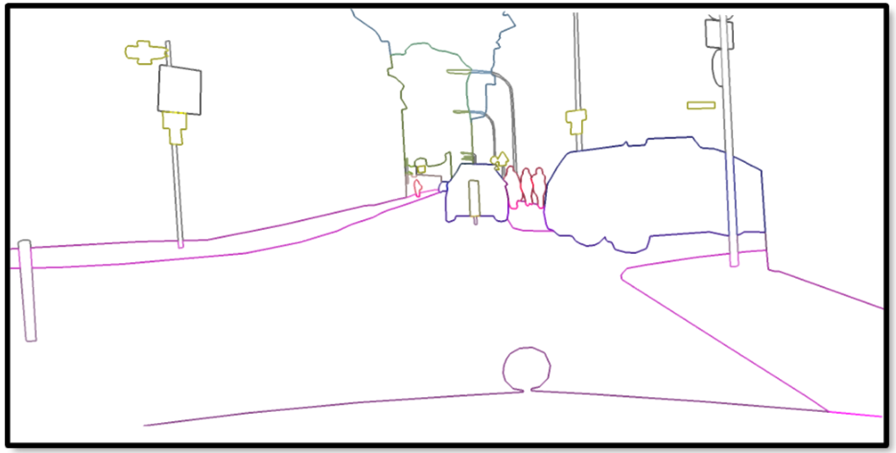
Category-Aware Semantic Edge Detection

Original Image

Perceptual Edges

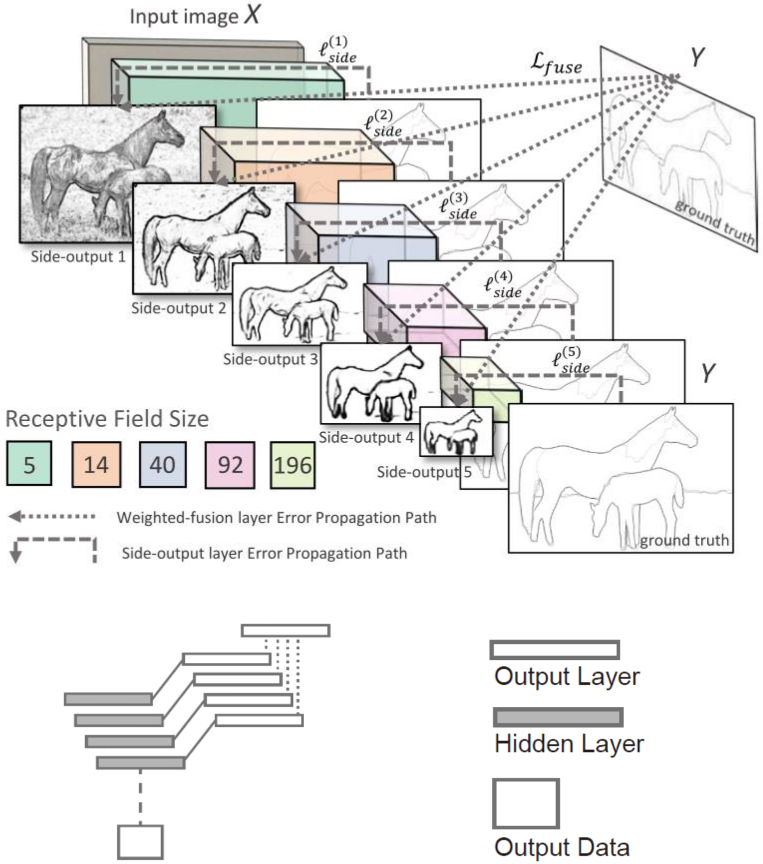


Semantic Edges

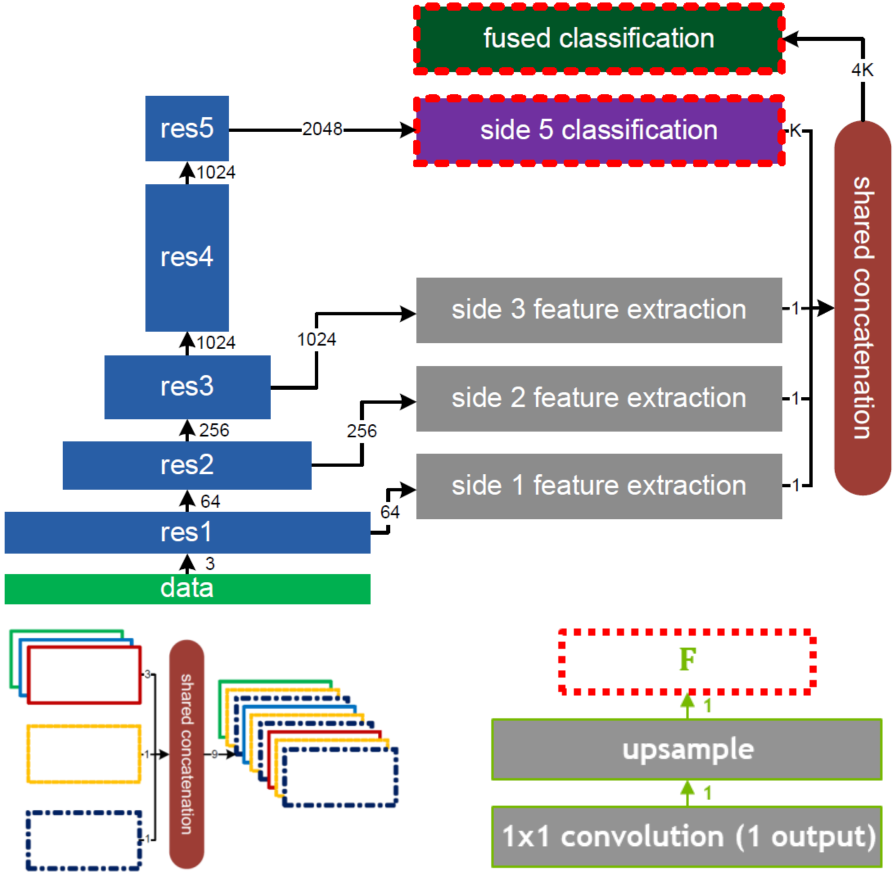


Category-Aware Semantic Edges

Category-Aware Semantic Edge Detection



Saining Xie et al., **Holistically-Nested Edge Detection**, ICCV15



Zhiding Yu et al., **CASENet: Deep Category-Aware Semantic Edge Detection**, CVPR17

Human Annotations Can Be Noisy!

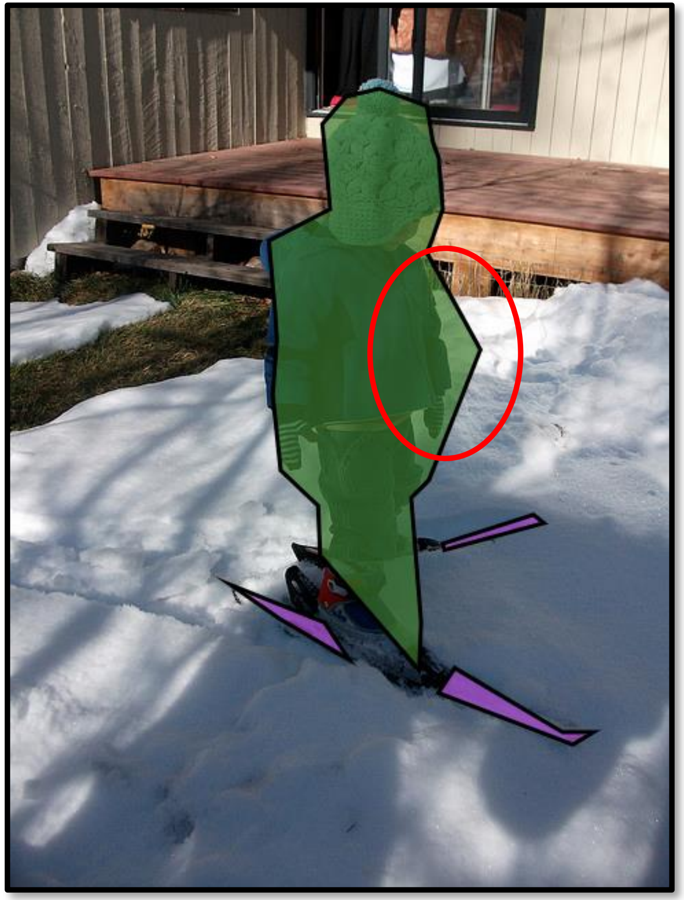
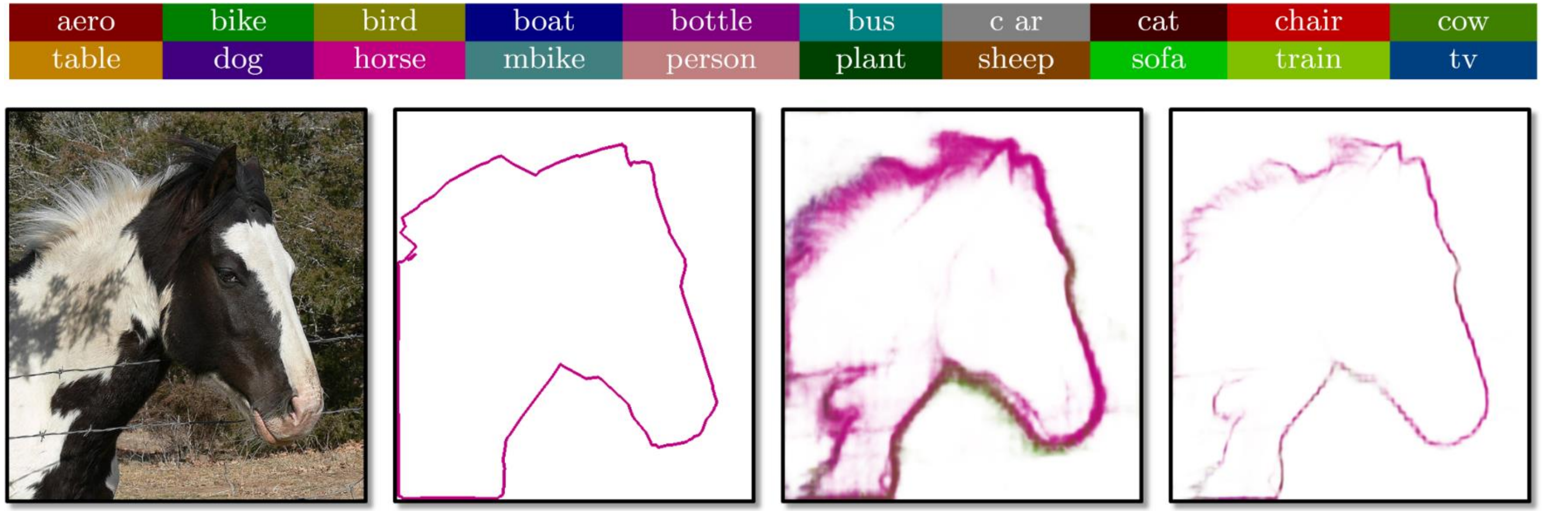


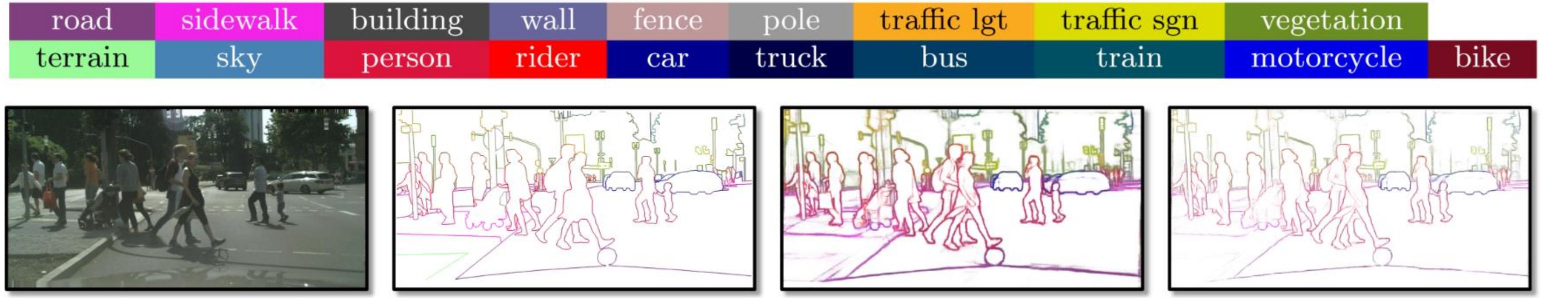
Image credit: Microsoft COCO: Common Objects in Context (http://cocodataset.org)

Motivations of This Work

Automatic edge alignment



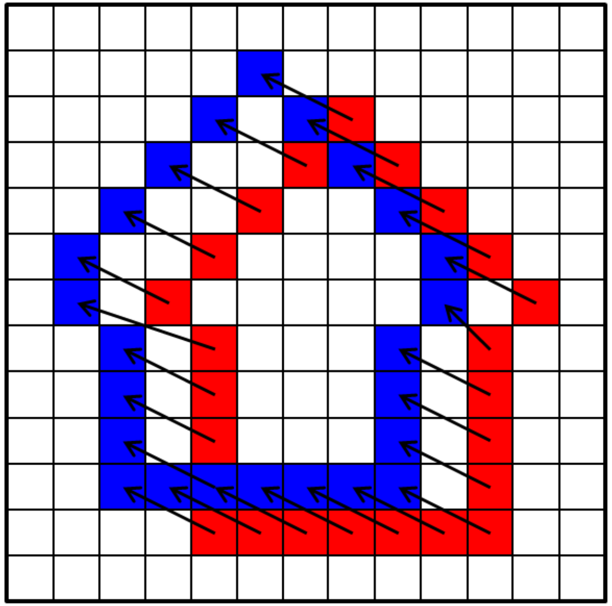
(a) Original image (b) Ground truth (c) CASENet (d) SEAL



(e) Original image (f) Ground truth (g) CASENet (h) SEAL

Producing high quality sharp/crisp edges during testing

The Proposed Learning Framework



$\mathbf{p} = (x_p, y_p), \mathbf{q} = (x_q, y_q)$: Pixel index

$k \in \{1, \ldots, K\}$: Semantic class index

$\mathbf{y} = \{y_{\mathbf{q}}^k \in \{0, 1\}\}$: Human annotation

$\hat{\mathbf{y}} = \{\hat{y}_{\mathbf{p}}^k \in \{0, 1\}\}$: Aligned edge label

■ $y_{\mathbf{q}}^k = 1$ ■ $\hat{y}_{\mathbf{p}}^k = 1$ $\nwarrow$ $m(\mathbf{q}) - \mathbf{q}$

Traditional edge learning:

$$\max_{\mathbf{W}} \mathcal{L}(\mathbf{W}) = P(\mathbf{y}|\mathbf{x}; \mathbf{W})$$

Simultaneous edge alignment & learning:

$$\max_{\hat{\mathbf{y}}, \mathbf{W}} \mathcal{L}(\hat{\mathbf{y}}, \mathbf{W}) = P(\mathbf{y}, \hat{\mathbf{y}}|\mathbf{x}; \mathbf{W}) = P(\mathbf{y}|\hat{\mathbf{y}})P(\hat{\mathbf{y}}|\mathbf{x}; \mathbf{W})$$

$$= \prod_k \underbrace{P(\mathbf{y}^k|\hat{\mathbf{y}}^k)}_{\text{Edge prior}} \underbrace{P(\hat{\mathbf{y}}^k|\mathbf{x}; \mathbf{W})}_{\text{Network likelihood}}$$

Edge prior model

$$P(\mathbf{y}^k|\hat{\mathbf{y}}^k) \propto \sup_{m \in \mathcal{M}(\mathbf{y}^k, \hat{\mathbf{y}}^k)} \prod_{(\mathbf{p},\mathbf{q}) \in E_m} \exp\left(-\frac{\|\mathbf{p} - \mathbf{q}\|^2}{2\sigma^2}\right)$$

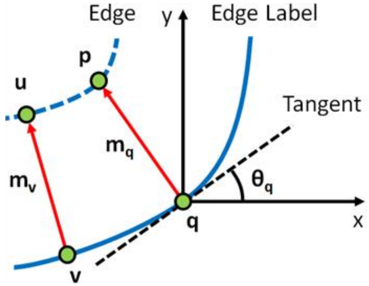
$$= \exp\left(-\inf_{m \in \mathcal{M}(\mathbf{y}^k, \hat{\mathbf{y}}^k)} \sum_{(\mathbf{p},\mathbf{q}) \in E_m} \frac{\|\mathbf{p} - \mathbf{q}\|^2}{2\sigma^2}\right)$$

Network likelihood model

$$P(\hat{\mathbf{y}}^k|\mathbf{x}; \mathbf{W}) = \prod_{\mathbf{p}} P(\hat{y}_{\mathbf{p}}^k|\mathbf{x}; \mathbf{W})$$

$$= \prod_{\mathbf{p}} h_k(\mathbf{p}|\mathbf{x}; \mathbf{W})^{\hat{y}_{\mathbf{p}}^k}(1 - h_k(\mathbf{p}|\mathbf{x}; \mathbf{W}))^{(1 - \hat{y}_{\mathbf{p}}^k)}$$

Issue with isotropic Gaussian kernels:



Biased Gaussian kernel and neighbor smoothness:

$$P(\mathbf{y}|\hat{\mathbf{y}}) \propto \sup_{m \in \mathcal{M}(\mathbf{y}, \hat{\mathbf{y}})} \prod_{(\mathbf{p},\mathbf{q}) \in E_m} \exp(-\mathbf{m}_{\mathbf{q}}^{\top} \boldsymbol{\Sigma}_{\mathbf{q}} \mathbf{m}_{\mathbf{q}}) \prod_{\substack{(\mathbf{u},\mathbf{v}) \in E_m, \\ \mathbf{v} \in \mathcal{N}(\mathbf{q})}} \exp(-\lambda \|\mathbf{m}_{\mathbf{q}} - \mathbf{m}_{\mathbf{v}}\|^2)$$

$$\mathbf{m}_{\mathbf{q}} = \mathbf{p} - \mathbf{q}, \text{ and } \mathbf{m}_{\mathbf{v}} = \mathbf{u} - \mathbf{v}$$

$$\boldsymbol{\Sigma}_{\mathbf{q}} = \begin{bmatrix} \frac{\cos(\theta_{\mathbf{q}})^2}{2\sigma_x^2} + \frac{\sin(\theta_{\mathbf{q}})^2}{2\sigma_y^2} & \frac{\sin(2\theta_{\mathbf{q}})}{4\sigma_y^2} - \frac{\sin(2\theta_{\mathbf{q}})}{4 * \sigma_x^2} \\ \frac{\sin(2\theta_{\mathbf{q}})}{4\sigma_y^2} - \frac{\sin(2\theta_{\mathbf{q}})}{4\sigma_x^2} & \frac{\sin(\theta_{\mathbf{q}})^2}{2\sigma_x^2} + \frac{\cos(\theta_{\mathbf{q}})^2}{2\sigma_y^2} \end{bmatrix}$$

Zhiding Yu et al., **Simultaneous Edge Alignment and Learning**, ECCV18

Learning and Optimization

Optimization as the following assignment problem:

$$\min_{m \in \mathrm{M}} \ \mathcal{C}(m) = \mathcal{C}_{Unary}(m) + \mathcal{C}_{Pair}(m)$$

$$= \sum_{(\mathbf{p},\mathbf{q}) \in E_m} \left[\mathbf{m}_\mathbf{q}^\top \Sigma_\mathbf{q} \mathbf{m}_\mathbf{q} + \log((1-\sigma(\mathbf{p}))/\sigma(\mathbf{p})) \right]$$

$$+ \lambda \sum_{(\mathbf{p},\mathbf{q}) \in E_m} \sum_{\substack{(\mathbf{u},\mathbf{v}) \in E_m, \\ \mathbf{v} \in \mathcal{N}(\mathbf{q})}} \|\mathbf{m}_\mathbf{q} - \mathbf{m}_\mathbf{v}\|^2$$

Take iterated conditional mode like optimization:

$$\text{Initialize:} \ m^{(1)} = \arg\min_{m \in \mathrm{M}} \ \mathcal{C}_{Unary}(m)$$

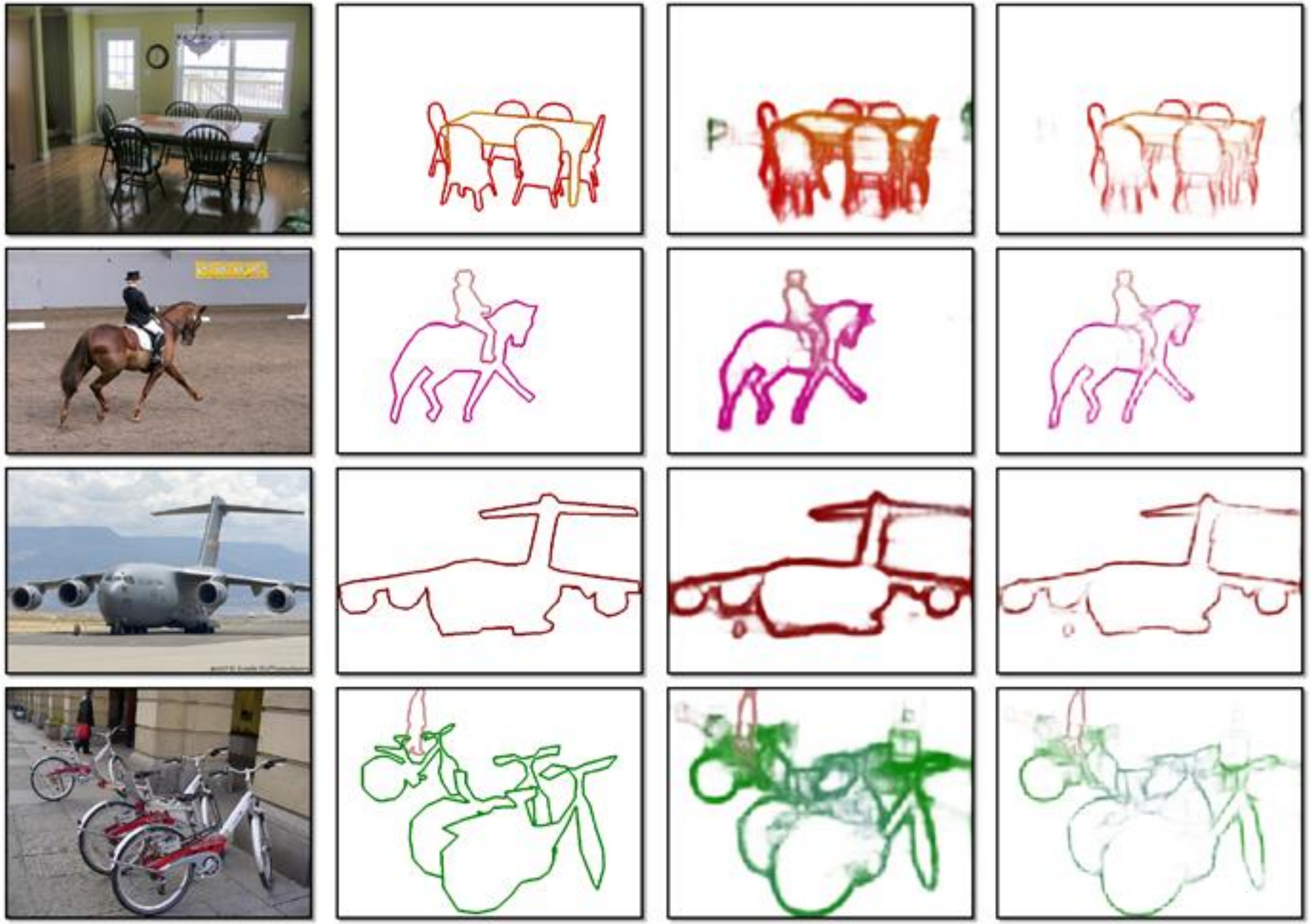
$$\text{Assign:} \ m^{(t+1)} = \arg\min_{m \in \mathrm{M}} \ \mathcal{C}_{Unary}(m) + \mathcal{C}_{Pair}(m, m^{(t)})$$

$$\text{Update:} \ \mathcal{C}_{Pair}(m, m^{(t)}) \to \mathcal{C}_{Pair}(m, m^{(t+1)})$$

Relaxation by decouple mappings in pairwise cost:

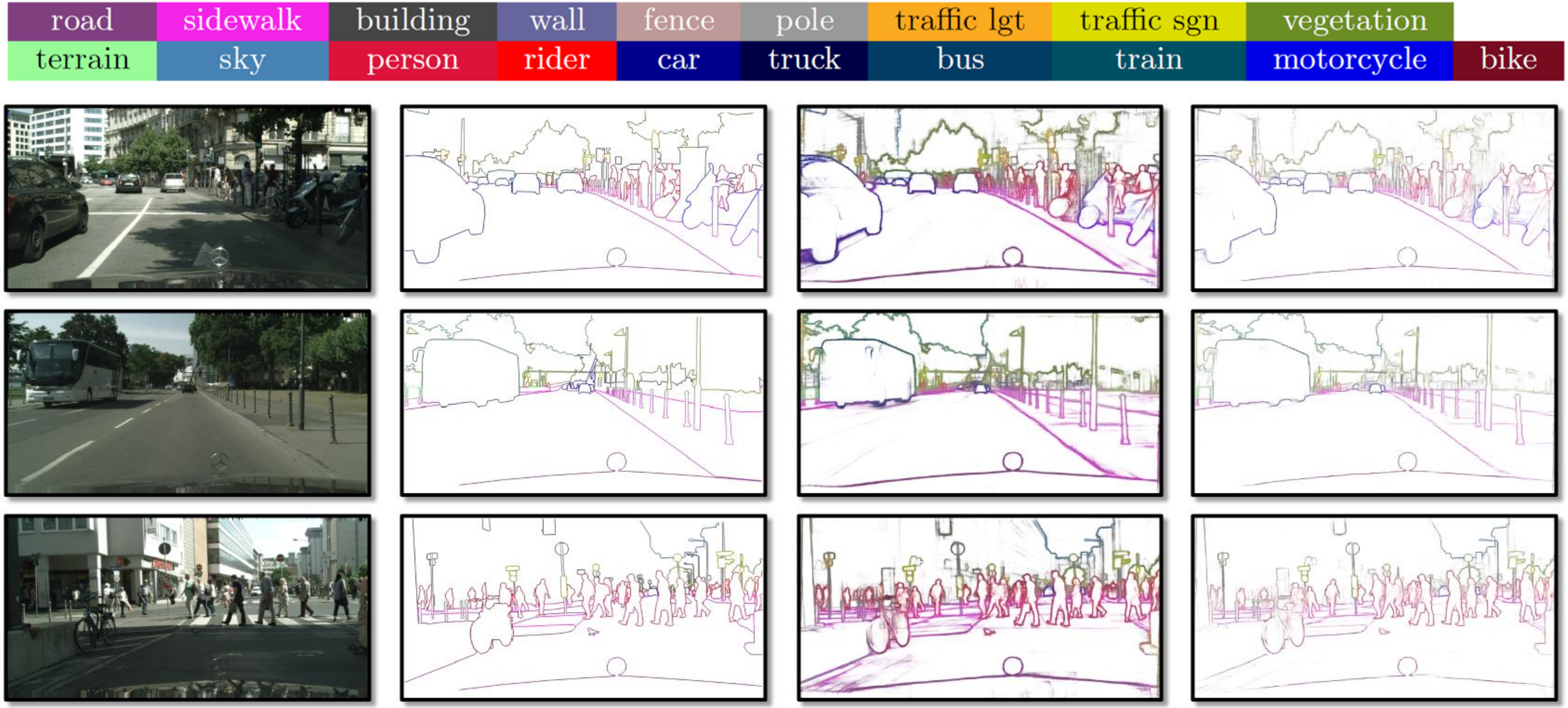
$$\mathcal{C}_{Pair}(m, m') = \sum_{(\mathbf{p},\mathbf{q}) \in E_m} \sum_{\substack{(\mathbf{u},\mathbf{v}) \in E_{m'}, \\ \mathbf{v} \in \mathcal{N}(\mathbf{q})}} \|\mathbf{m}_\mathbf{q} - \mathbf{m}_\mathbf{v}\|^2$$

Experiment: Qualitative Results (SBD)

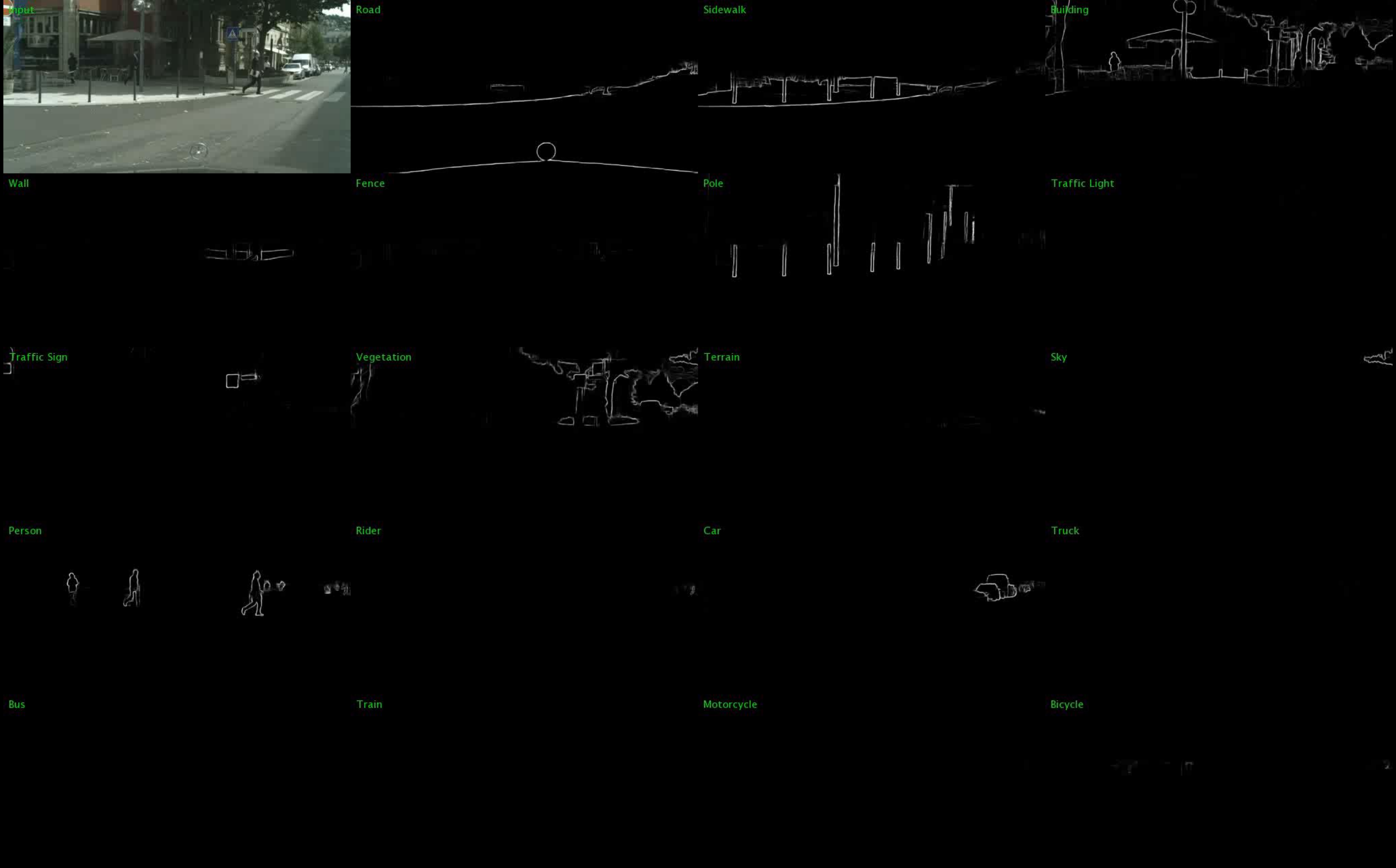


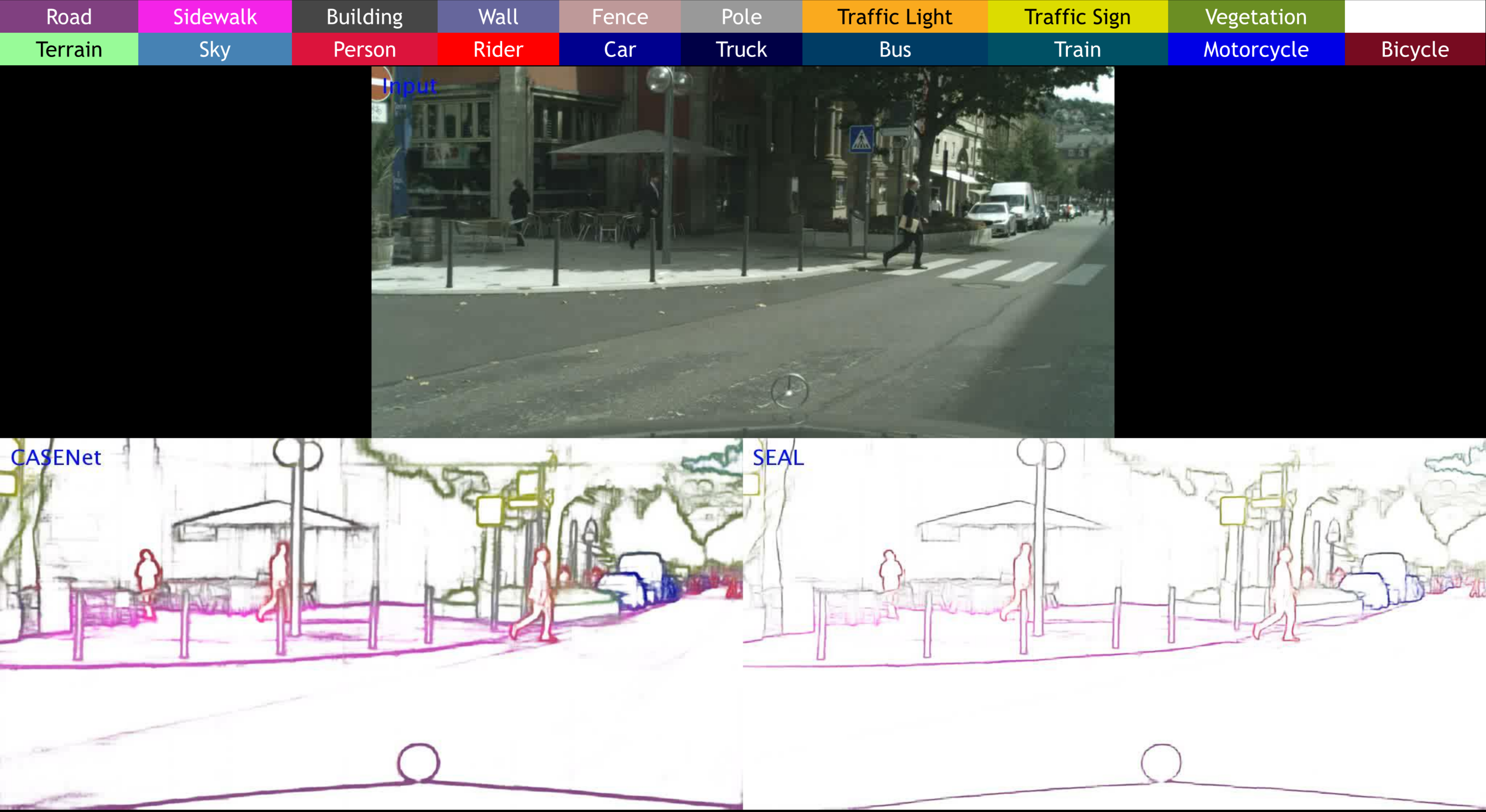
Original GT CASENet SEAL

Experiment: Qualitative Results (Cityscapes)

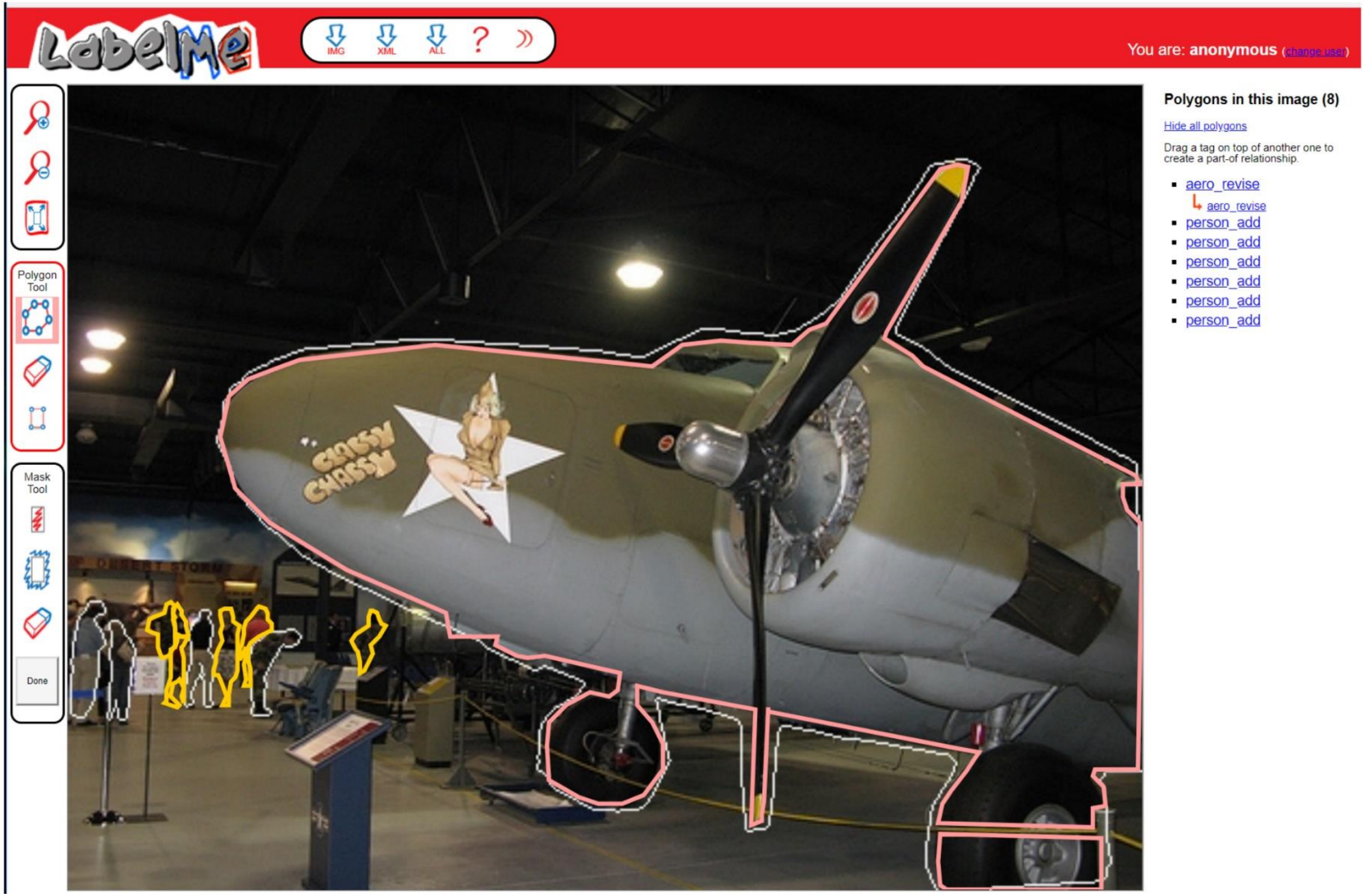


Original GT CASENet SEAL


Input
Road
Sidewalk
Building
Wall
Fence
Pole
Traffic Light
Traffic Sign
Vegetation
Terrain
Sky
Person
Rider
Car
Truck
Bus
Train
Motorcycle
Bicycle


Road
Sidewalk
Building
Wall
Fence
Pole
Traffic Light
Traffic Sign
Vegetation
Terrain
Sky
Person
Rider
Car
Truck
Bus
Train
Motorcycle
Bicycle
Input
CASENet
SEAL

SBD Test Set Re-Annotation



Experiment: Quantitative Results

MF scores on the re-annotated SBD test set. Results are measured by %.

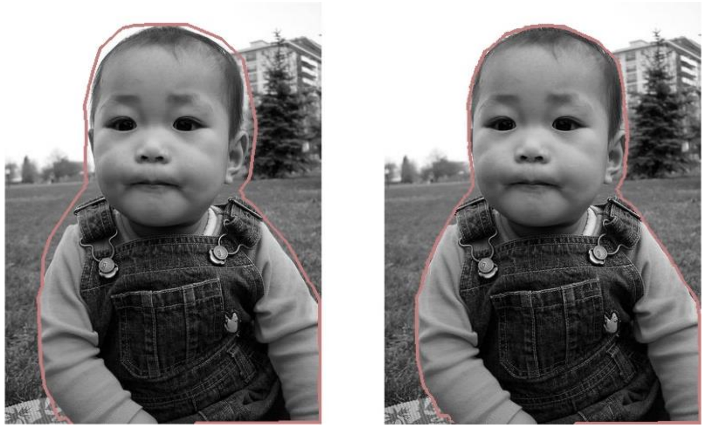
Metric	Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
MF (Thin)	CASENet	83.6	75.3	82.3	63.1	70.5	83.5	76.5	82.6	56.8	76.3	47.5	80.8	80.9	75.6	80.7	54.1	77.7	52.3	77.9	68.0	72.3
	CASENet-S	**84.5**	**76.5**	**83.7**	**65.3**	71.3	**83.9**	**78.3**	84.5	**58.8**	76.8	50.8	81.9	**82.3**	**77.2**	82.7	**55.9**	78.1	54.0	**79.5**	69.4	**73.8**
	CASENet-C	83.9	71.1	82.5	62.6	71.0	82.2	76.8	83.4	56.5	**76.9**	49.2	81.0	81.1	75.4	81.4	54.0	**78.5**	53.3	77.1	67.0	72.2
	SEAL	**84.5**	**76.5**	**83.7**	64.9	**71.7**	83.8	78.1	**85.0**	**58.8**	76.6	**50.9**	**82.4**	82.2	77.1	**83.0**	55.1	78.4	**54.4**	79.3	**69.6**	**73.8**
MF (Raw)	CASENet	71.8	60.2	72.6	49.5	59.3	73.3	65.2	70.8	51.9	64.9	41.2	67.9	72.5	64.1	71.2	44.0	71.7	45.7	65.4	55.8	62.0
	CASENet-S	75.8	65.0	78.4	56.2	64.7	76.4	71.8	75.2	55.2	68.7	45.8	72.8	77.0	68.1	76.5	47.1	75.5	49.0	70.2	60.6	66.5
	CASENet-C	80.4	67.1	79.9	57.9	65.9	77.6	72.6	79.2	53.5	72.7	45.5	76.7	79.4	71.2	78.3	**50.8**	77.6	50.7	71.6	61.6	68.5
	SEAL	**81.1**	**69.6**	**81.7**	**60.6**	**68.0**	**80.5**	**75.1**	**80.7**	**57.0**	**73.1**	**48.1**	**78.2**	**80.3**	**72.1**	**79.8**	50.0	**78.2**	**51.8**	**74.6**	**65.0**	**70.3**

MF scores on the Cityscapes validation set. Results are measured by %.

Metric	Method	road	sidewalk	building	wall	fence	pole	t-light	t-sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike	mean
MF (Thin)	CASENet	86.2	74.9	74.5	47.6	**46.5**	72.8	70.0	73.3	79.3	57.0	86.5	80.4	66.8	88.3	49.3	64.6	**47.8**	**55.8**	71.9	68.1
	CASENet-S	**87.6**	77.1	**75.9**	**48.7**	46.2	**75.5**	**71.4**	75.3	80.6	59.7	86.8	81.4	68.1	**89.2**	**50.7**	**68.0**	42.5	54.6	72.7	**69.1**
	SEAL	**87.6**	**77.5**	**75.9**	47.6	46.3	**75.5**	71.2	**75.4**	**80.9**	**60.1**	**87.4**	**81.5**	**68.9**	88.9	50.2	67.8	44.1	52.7	**73.0**	**69.1**
MF (Raw)	CASENet	66.8	64.6	66.8	39.4	40.6	71.7	64.2	65.1	71.1	50.2	80.3	73.1	58.6	77.0	42.0	53.2	39.1	46.1	62.2	59.6
	CASENet-S	79.2	70.8	70.4	42.5	42.4	73.9	66.7	68.2	74.6	54.6	82.5	75.7	61.5	82.7	46.0	59.7	39.1	47.0	64.8	63.3
	SEAL	**84.4**	**73.5**	**72.7**	**43.4**	**43.2**	**76.1**	**68.5**	**69.8**	**77.2**	**57.5**	**85.3**	**77.6**	**63.6**	**84.9**	**48.6**	**61.9**	**41.2**	**49.0**	**66.7**	**65.5**

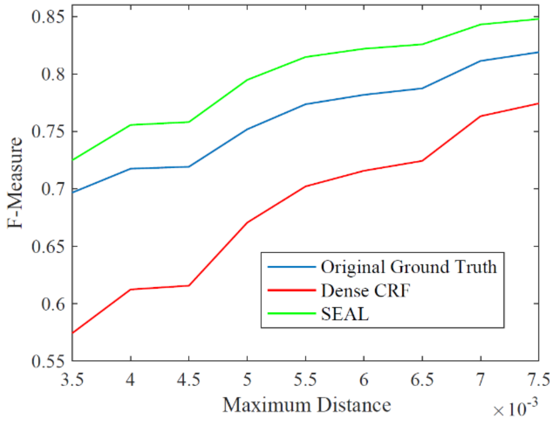
Experiment: Automatic Label Refinement

Alignment on Cityscapes (red: before alignment, blue: after alignment)



Original GT

SEAL



Learning with Incomplete Supervision

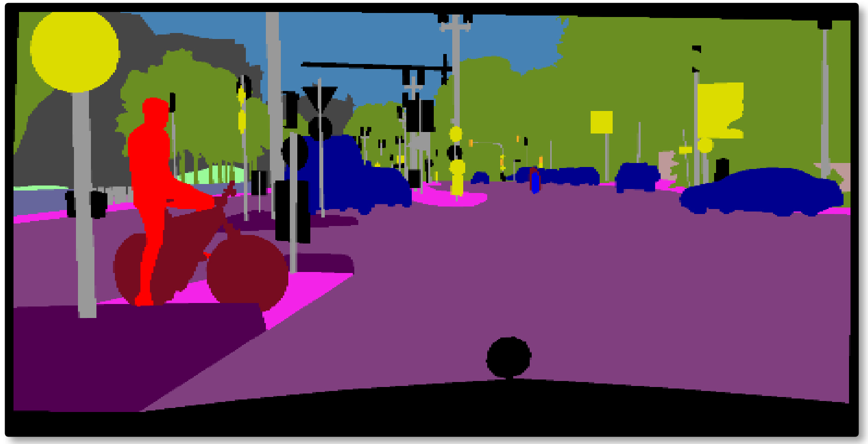
Obtaining Per-Pixel Dense Labels is Hard

Real application often requires model robustness over scenes with large diversity

- Different cities, different weather, different views
- Large scale annotated image data is beneficial

Annotating large scale real world image dataset is expensive

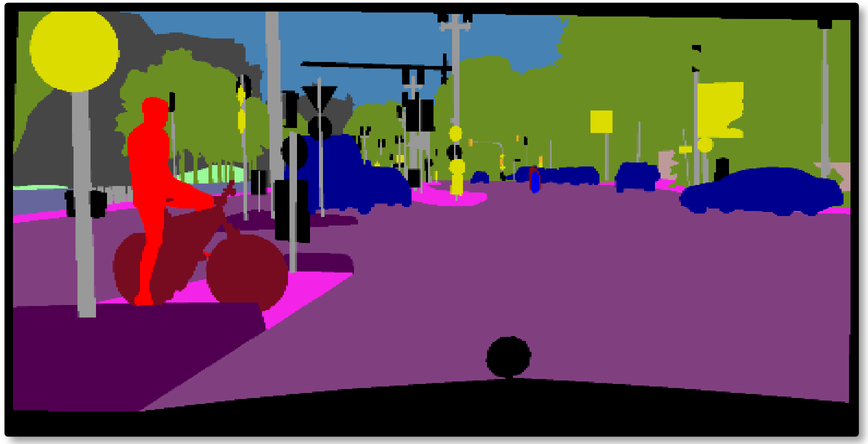
- Cityscapes dataset: 90 minutes per image on average



road	sidewalk	building	wall	fence	pole	traffic lgt	traffic sgn	vegetation	
terrain	sky	person	rider	car	truck	bus	train	motorcycle	bike

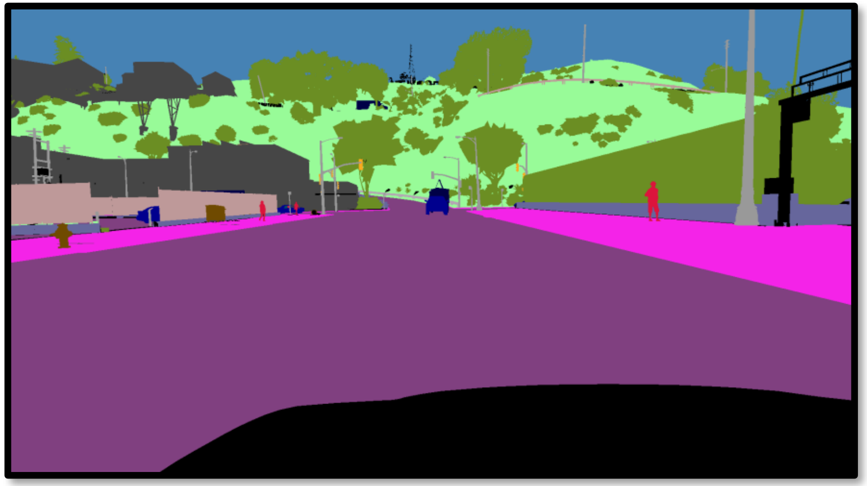
Use Synthetic Data to Obtain Infinite GTs?

Original image from Cityscapes



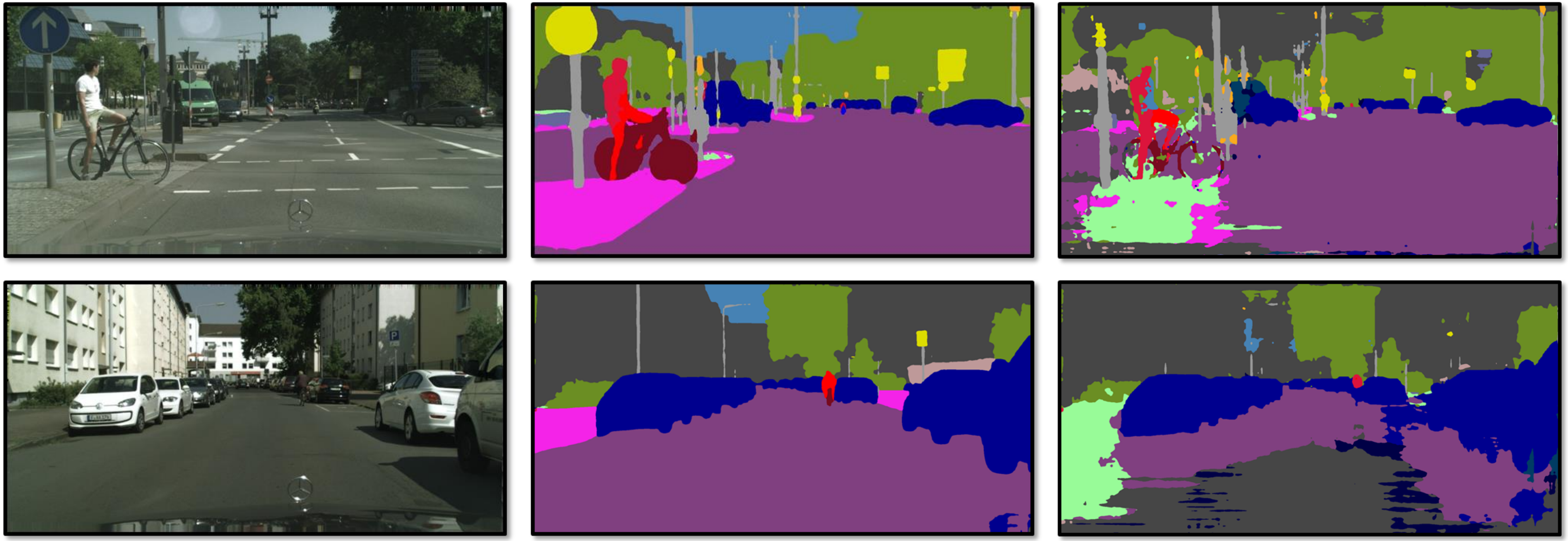
Human annotated ground truth

Original image from GTA5



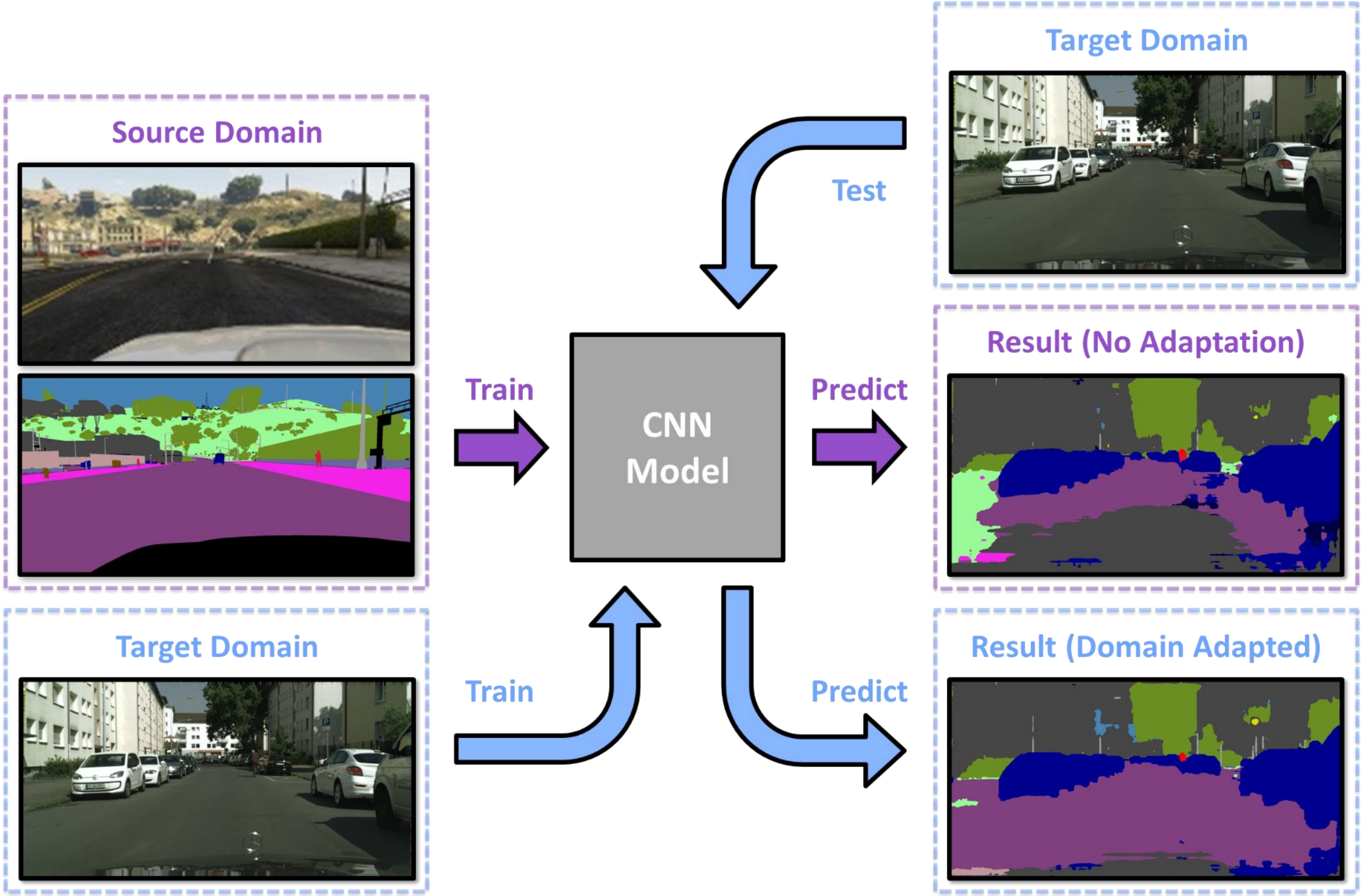
Ground truth from game Engine

Drop of Performance Due to Domain Gaps

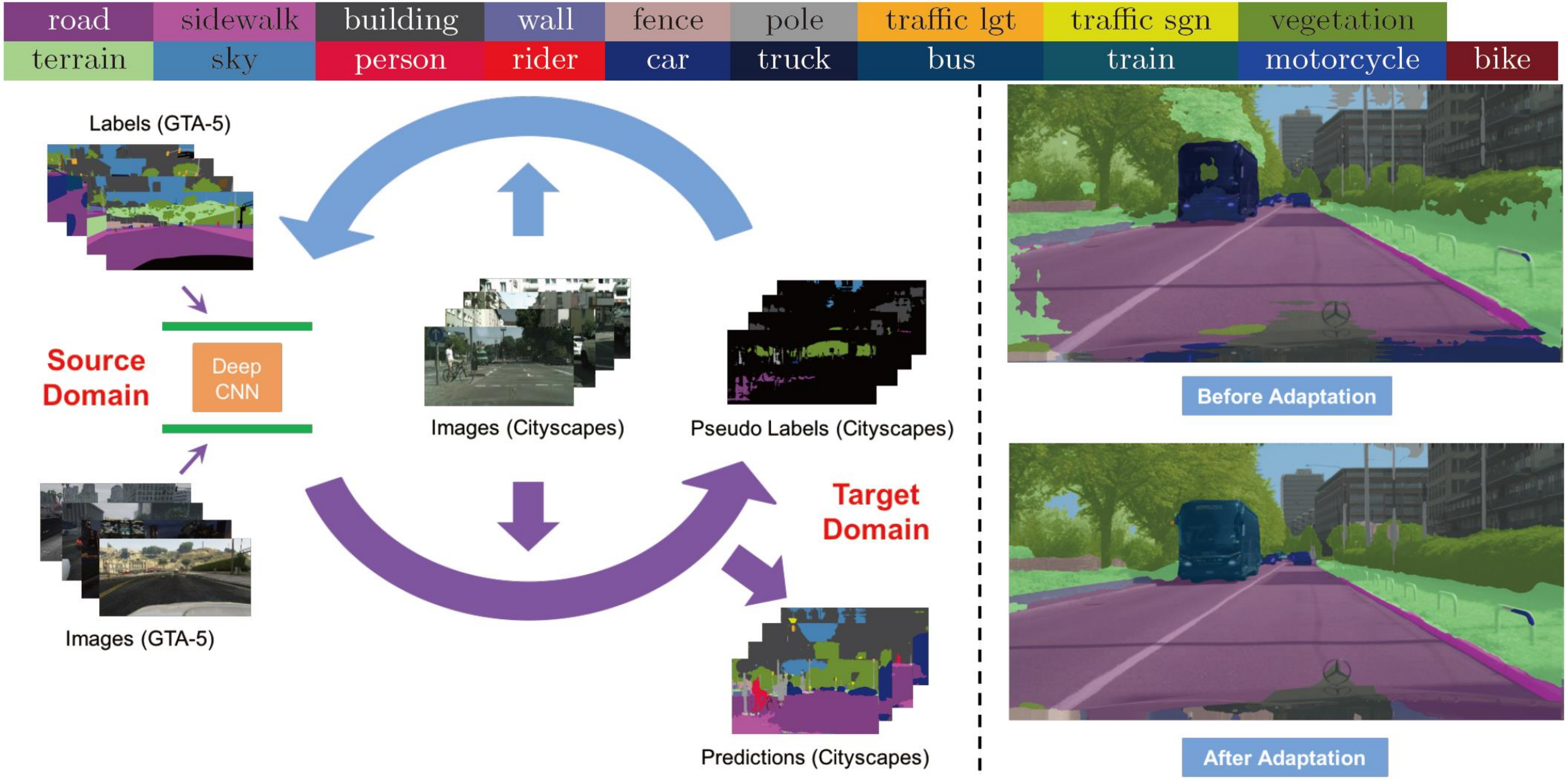


Cityscapes images | Model trained on Cityscapes | Model trained on GTA5

Unsupervised Domain Adaptation



Domain Adaptation via Deep Self-Training



Yang Zou*, Zhiding Yu* et al., **Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-Training**, ECCV18

Preliminaries and Definitions

Fine-tuning for Supervised Domain Adaptation

$$\min_{\mathbf{w}} \mathcal{L}_S(\mathbf{w}) = -\sum_{s=1}^{S}\sum_{n=1}^{N} \mathbf{y}_{s,n}^{\top} \log(\mathbf{p}_n(\mathbf{w}, \mathbf{I}_s)) - \sum_{t=1}^{T}\sum_{n=1}^{N} \mathbf{y}_{t,n}^{\top} \log(\mathbf{p}_n(\mathbf{w}, \mathbf{I}_t))$$

where: $\mathbf{I}$: input image (crop) $\mathbf{p}$: pixel class probability vector $\mathbf{y}$: pixel label vector

$\mathbf{w}$: network parameters s: source image index t: target image index

Self-Training for Unsupervised Domain Adaptation

$$\min_{\mathbf{w},\hat{\mathbf{y}}} \mathcal{L}_U(\mathbf{w}, \hat{\mathbf{y}}) = -\sum_{s=1}^{S}\sum_{n=1}^{N} \mathbf{y}_{s,n}^{\top} \log(\mathbf{p}_n(\mathbf{w}, \mathbf{I}_s)) - \sum_{t=1}^{T}\sum_{n=1}^{N} \hat{\mathbf{y}}_{t,n}^{\top} \log(\mathbf{p}_n(\mathbf{w}, \mathbf{I}_t))$$

$$s.t.\ \hat{\mathbf{y}}_{t,n} \in \{\mathbf{e}^{(i)} | \mathbf{e}^{(i)} \in \mathbb{R}^C\}, \forall t, n$$

where: $\hat{\mathbf{y}}$: pseudo label vector $\mathbf{e}^{(i)}$: one-hot vector

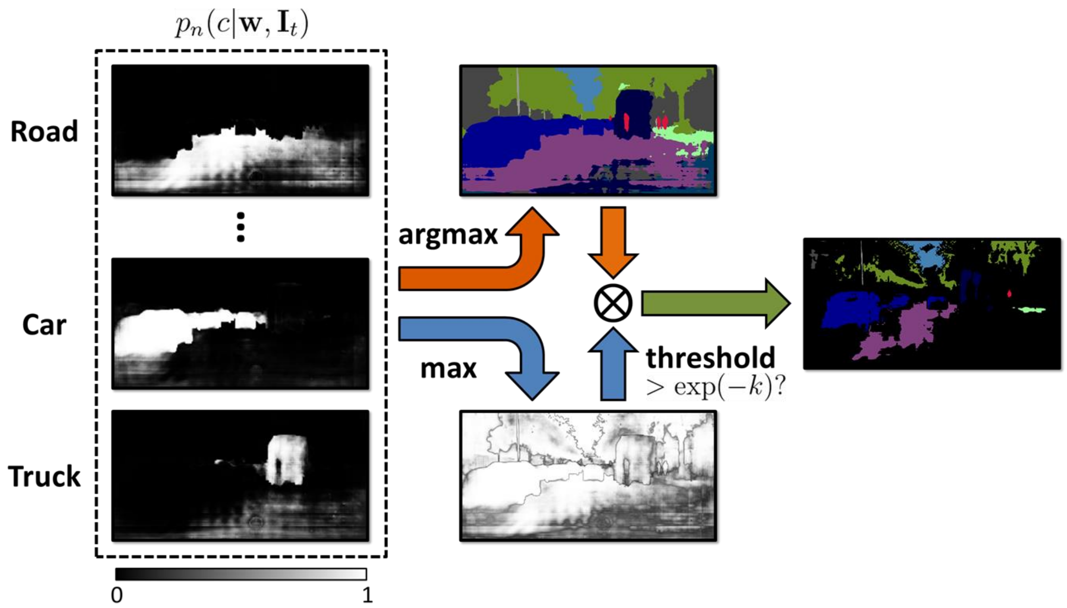
Self-Training (ST) with Self-Paced Learning

$$\min_{\mathbf{w},\hat{\mathbf{y}}} \mathcal{L}_{ST}(\mathbf{w},\hat{\mathbf{y}}) = -\sum_{s=1}^{S}\sum_{n=1}^{N} \mathbf{y}_{s,n}^{\top} \log(\mathbf{p}_n(\mathbf{w},\mathbf{I}_s)) - \sum_{t=1}^{T}\sum_{n=1}^{N} [\hat{\mathbf{y}}_{t,n}^{\top} \log(\mathbf{p}_n(\mathbf{w},\mathbf{I}_t)) + k|\hat{\mathbf{y}}_{t,n}|_1]$$

$$s.t.\ \hat{\mathbf{y}}_{t,n} \in \{\{\mathbf{e}^{(i)}|\mathbf{e}^{(i)} \in \mathbb{R}^C\} \cup \mathbf{0}\}, \forall t, n$$
$$k > 0$$

The cost can be minimized via mixed integer programming, which leads to the following solution:

$$\hat{y}_{t,n}^{(c)*} = \begin{cases} 1, \text{ if } c = \arg\max_c p_n(c|\mathbf{w},\mathbf{I}_t), \\ \quad p_n(c|\mathbf{w},\mathbf{I}_t) > \exp(-k) \\ 0, \text{ otherwise} \end{cases}$$



Class-Balanced Self-Training

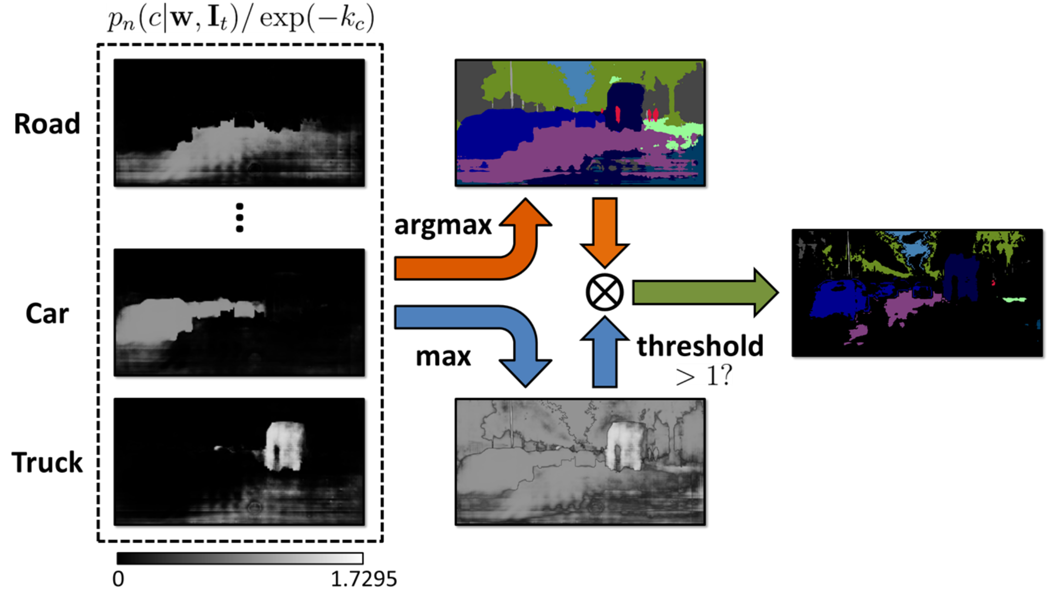
$$\min_{\mathbf{w},\hat{\mathbf{y}}} \mathcal{L}_{CB}(\mathbf{w},\hat{\mathbf{y}}) = -\sum_{s=1}^{S}\sum_{n=1}^{N}\mathbf{y}_{s,n}^{\top}\log(\mathbf{p}_n(\mathbf{w},\mathbf{I}_s)) - \sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{c=1}^{C}[\hat{y}_{t,n}^{(c)}\log(p_n(c|\mathbf{w},\mathbf{I}_t)) + k_c\hat{y}_{t,n}^{(c)}]$$

$$s.t.\ \hat{\mathbf{y}}_{t,n} = [\hat{y}_{t,n}^{(1)},...,\hat{y}_{t,n}^{(C)}] \in \{\{\mathbf{e}^{(i)}|\mathbf{e}^{(i)} \in \mathbb{R}^C\} \cup \mathbf{0}\}, \forall t,n$$

$$k_c > 0, \forall c$$

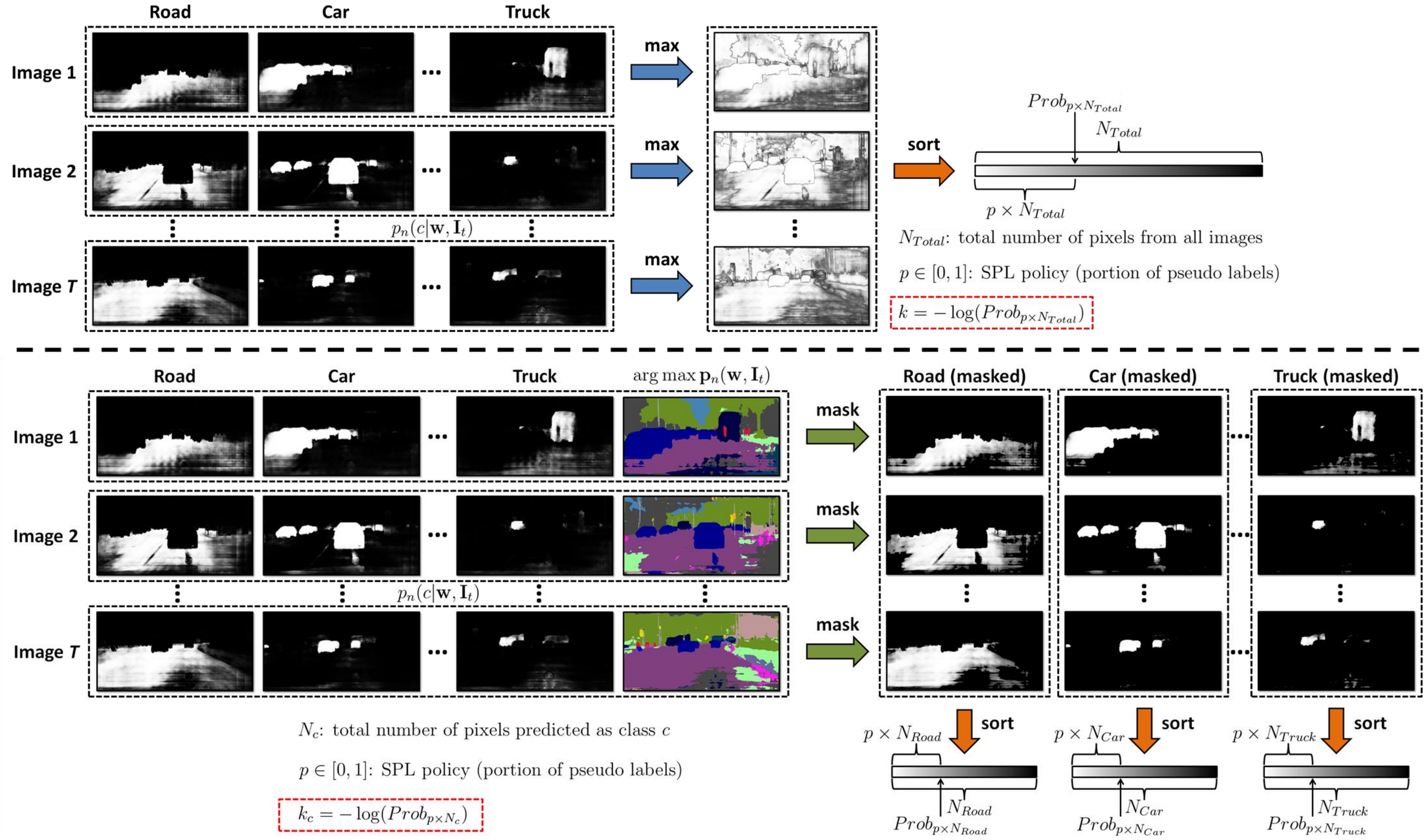
Again using mixed integer programming, one obtains the following solution:

$$\hat{y}_{t,n}^{(c)*} = \begin{cases} 1, \ \textbf{if}\ c = \arg\max\limits_{c} \dfrac{p_n(c|\mathbf{w},\mathbf{I}_t)}{\exp(-k_c)}, \\ \qquad \dfrac{p_n(c|\mathbf{w},\mathbf{I}_t)}{\exp(-k_c)} > 1 \\ 0, \ \text{otherwise} \end{cases}$$

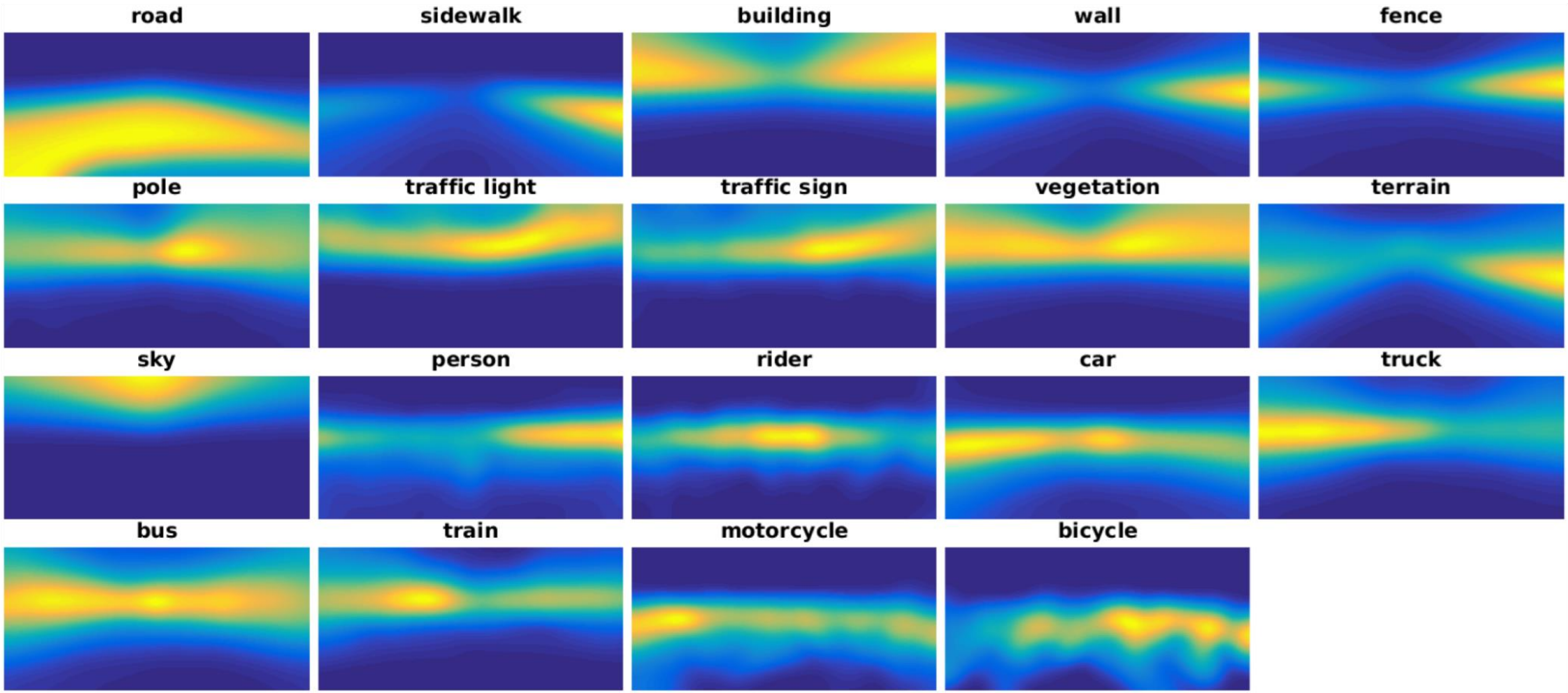


Self-Paced Learning Policy Design

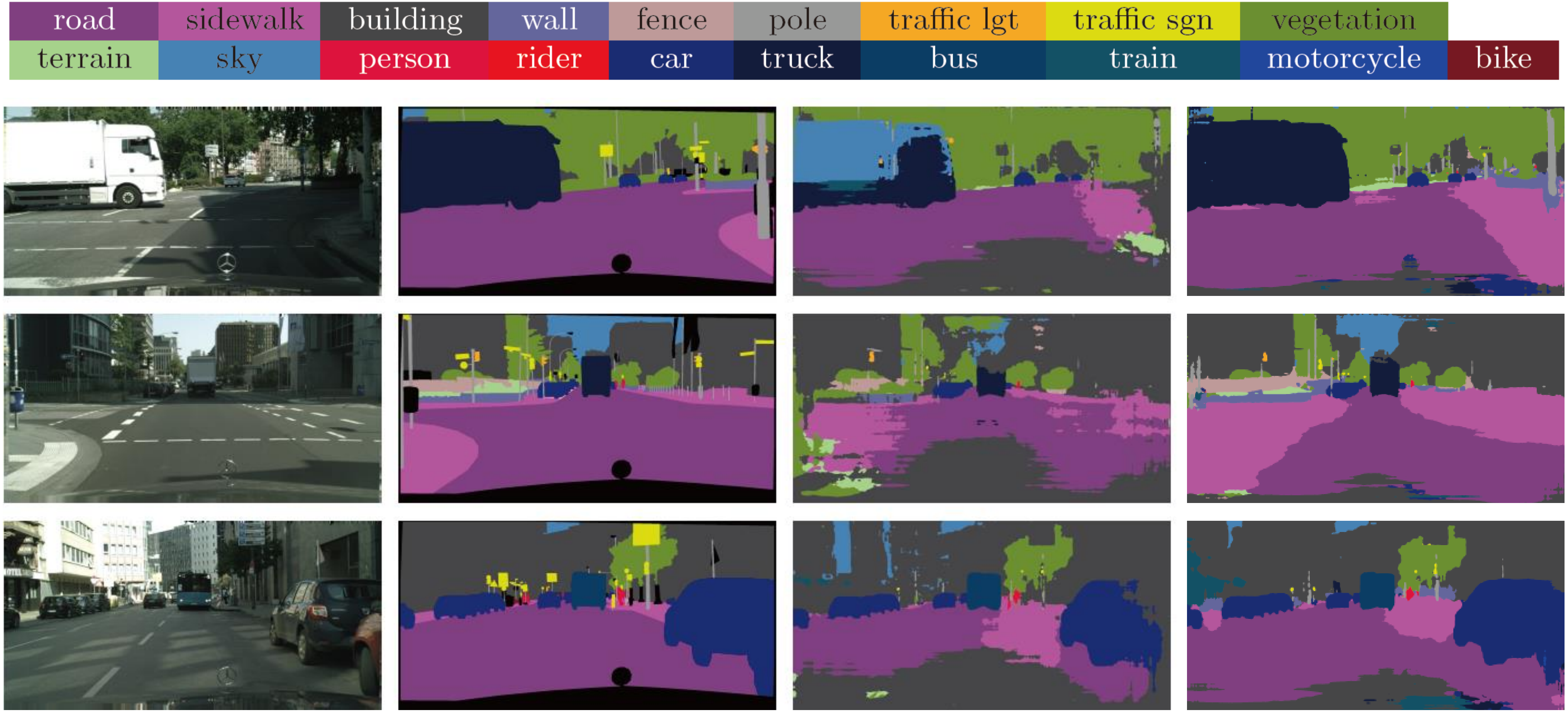
The both k and k_c in ST and CBST can be easily determined with a single SPL policy parameter p:



Incorporating Spatial Priors



Experiment: GTA to Cityscapes



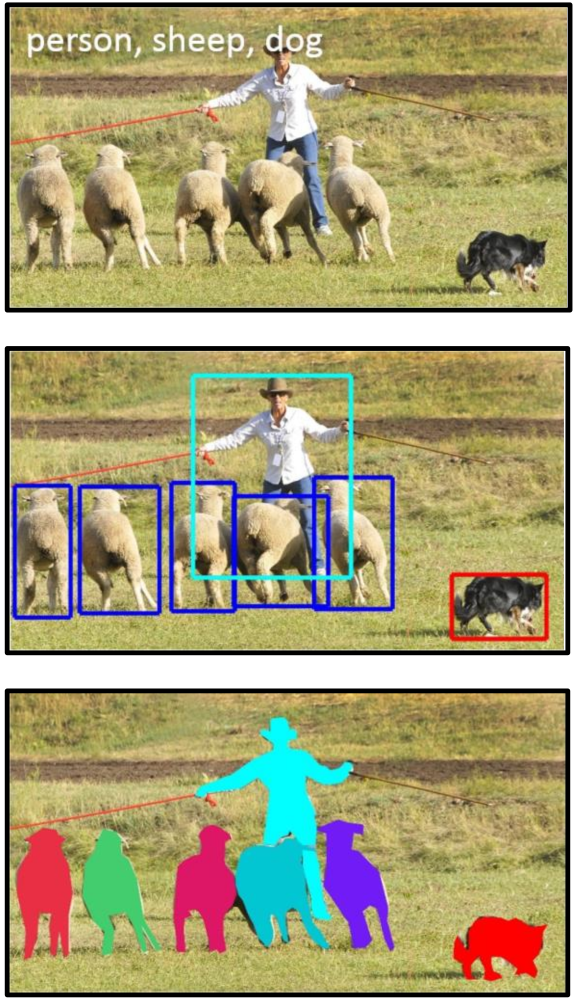
Original Image | Ground Truth | Source Model | CBST-SP

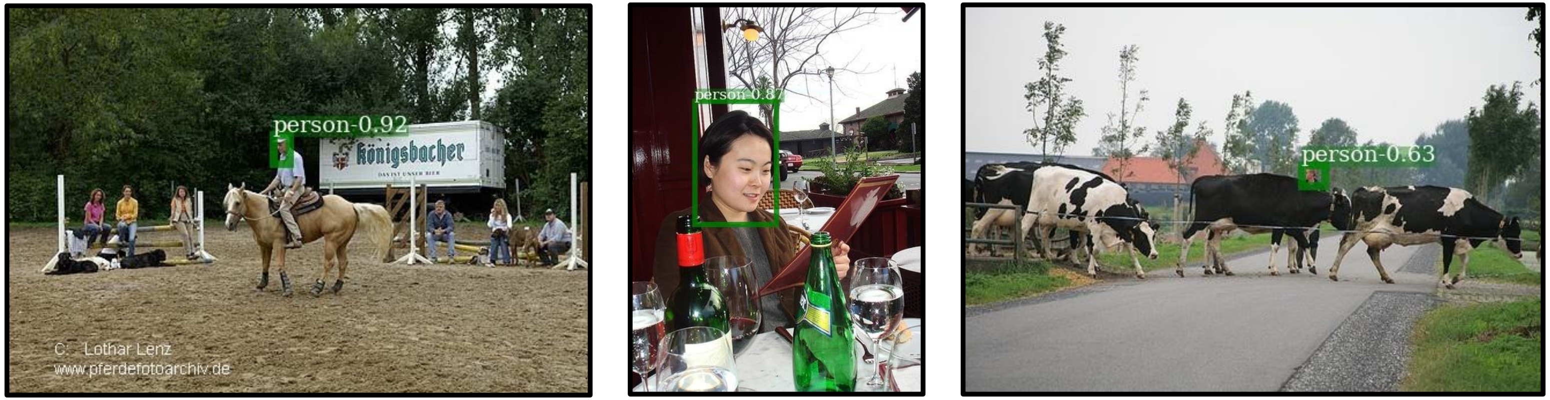
Experiment: GTA to Cityscapes

Method	Base Net	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg.	Terrain	Sky	PR	Rider	Car	Truck	Bus	Train	Motor	Bike	mIoU
Source only [18]	Dilation-Frontend	31.9	18.9	47.7	7.4	3.1	16.0	10.4	1.0	76.5	13.0	58.9	36.0	1.0	67.1	9.5	3.7	0.0	0.0	0.0	21.2
FCN wild [18]	[43]	70.4	32.4	62.1	14.9	5.4	10.9	14.2	2.7	79.2	21.3	64.6	44.1	4.2	70.4	8.0	7.3	0.0	3.5	0.0	27.1
Source only [45]	FCN8s-VGG16	18.1	6.8	64.1	7.3	8.7	21.0	14.9	16.8	45.9	2.4	64.4	41.6	17.5	55.3	8.4	5.0	6.9	4.3	13.8	22.3
Curr. DA [45]	[21]	74.9	22.0	71.7	6.0	11.9	8.4	16.3	11.1	75.7	13.3	66.5	38.0	9.3	55.2	18.8	18.9	0.0	16.8	16.6	28.9
Source only [17]	FCN8s-VGG16	26.0	14.9	65.1	5.5	12.9	8.9	6.0	2.5	70.0	2.9	47.0	24.5	0.0	40.0	12.1	1.5	0.0	0.0	0.0	17.9
CyCADA [17]	[21]	85.2	37.2	76.5	21.8	15.0	23.8	22.9	21.5	80.5	31.3	60.7	50.5	9.0	76.9	17.1	28.2	4.5	9.8	0.0	35.4
Source only [17]	Dilated ResNet-26	42.7	26.3	51.7	5.5	6.8	13.8	23.6	6.9	75.5	11.5	36.8	49.3	0.9	46.7	3.4	5.0	0.0	5.0	1.4	21.7
CyCADA [17]	[44]	79.1	33.1	77.9	23.4	17.3	32.1	33.3	31.8	81.5	26.7	69.0	62.8	14.7	74.5	20.9	25.6	6.9	18.8	20.4	39.5
Source only [30]	ResNet-50	64.5	24.9	73.7	14.8	2.5	18.0	15.9	0	74.9	16.4	72.0	42.3	0.0	39.5	8.6	13.4	0.0	0.0	0.0	25.3
ADR [30]	[16]	87.8	15.6	77.4	20.6	9.7	19.0	19.9	7.7	82.0	31.5	74.3	43.5	9.0	77.8	17.5	27.7	1.8	9.7	0.0	33.3
Source only [24]	DenseNet	67.3	23.1	69.4	13.9	14.4	21.6	19.2	12.4	78.7	24.5	74.8	49.3	3.7	54.1	8.7	5.3	2.6	6.2	1.9	29.0
I2I Adapt [24]	[19]	85.8	37.5	80.2	23.3	16.1	23.0	14.5	9.8	79.2	**36.5**	**76.4**	53.4	7.4	82.8	19.1	15.7	2.8	13.4	1.7	35.7
Source only [36]	DeepLab-v2	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6
MAA [36]	[19]	86.5	36.0	**79.9**	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
Source only	FCN8s-VGG16	64.0	22.1	68.6	13.3	8.7	19.9	15.5	5.9	74.9	13.4	37.0	37.7	10.3	48.2	6.1	1.2	1.8	10.8	2.9	24.3
ST	[18]	83.8	17.4	72.1	14.3	2.9	16.5	16.0	6.8	81.4	24.2	47.2	40.7	7.6	71.7	10.2	7.6	0.5	11.1	0.9	28.1
CBST		66.7	26.8	73.7	14.8	9.5	28.3	25.9	10.1	75.5	15.7	51.6	47.2	6.2	71.9	3.7	2.2	5.4	18.9	32.4	30.9
CBST-SP		**90.4**	50.8	72.0	18.3	9.5	27.2	28.6	14.1	82.4	25.1	70.8	42.6	14.5	76.9	5.9	12.5	1.2	14.0	28.6	36.1
Source only	ResNet-38	70.0	23.7	67.8	15.4	18.1	40.2	41.9	25.3	78.8	11.7	31.4	**62.9**	**29.8**	60.1	21.5	26.8	7.7	28.1	12.0	35.4
ST	[41]	90.1	56.8	77.9	28.5	23.0	41.5	45.2	39.6	84.8	26.4	49.2	59.0	27.4	82.3	39.7	45.6	**20.9**	**34.8**	**46.2**	41.5
CBST		86.8	46.7	76.9	26.3	**24.8**	42.0	46.0	38.6	80.7	15.7	48.0	57.3	27.9	78.2	24.5	49.6	17.7	25.5	45.1	45.2
CBST-SP		88.0	56.2	77.0	27.4	22.4	40.7	47.3	**40.9**	82.4	21.6	60.3	50.2	20.4	**83.8**	35.0	**51.0**	15.2	20.6	37.0	46.2
CBST-SP+MST		89.6	**58.9**	78.5	**33.0**	22.3	**41.4**	**48.2**	39.2	**83.6**	24.3	65.4	49.3	20.2	83.3	**39.0**	48.6	12.5	20.3	35.3	**47.0**

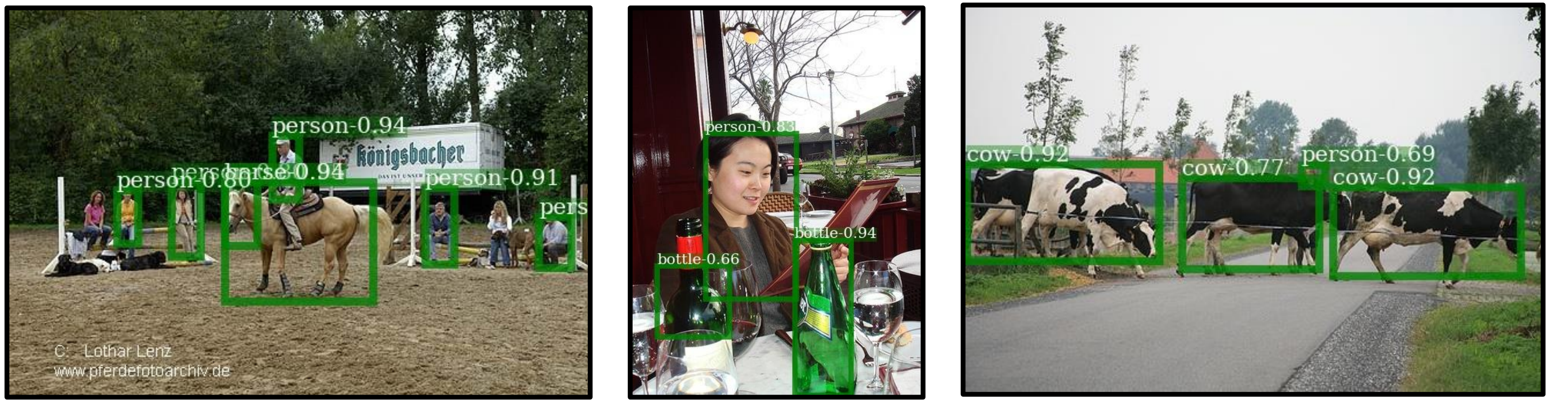
Learning with Inexact Supervision

Learning Instance Det/Seg with Image-Level Labels





Previous Method (WSDDN)



Our Proposed Method

Work in progress with Zhongzheng Ren, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz et al.

Conclusions and Future Works

Conclusions and Future Works

Conclusions

- WSL methods are useful in a wide range of tasks, such as Autonomous Driving, IVA, AI City, Robotics, Annotation, Web Video Analysis, Cloud Service, Advertisements, etc.
- Impact from a fundamental research perspective towards achieving AGI.

Future works

- A good WSL platform that can handle a variety of weak grounding signals and tasks.
- Models with better designed self-sup/meta-sup/structured info/priors/normalization.
- Large-scale weakly and unsupervised learning from videos.
- Weak grounding signal with combination to robotics and reinforcement learning.

Thanks You!