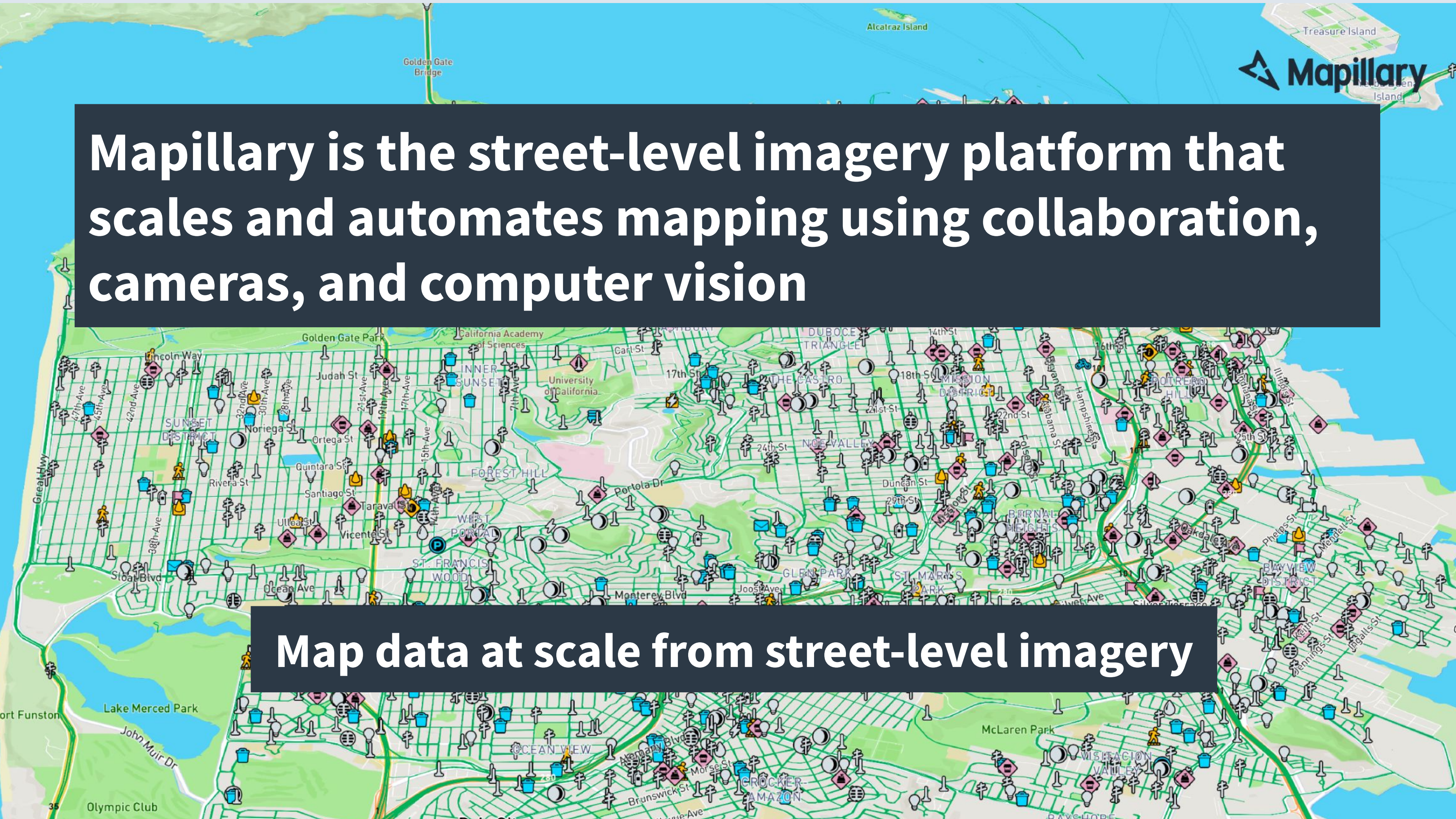
Recognition for Mapping on a Global Scale using Deep Learning and Computer Vision

Peter Kontschieder

Mapillary
Research

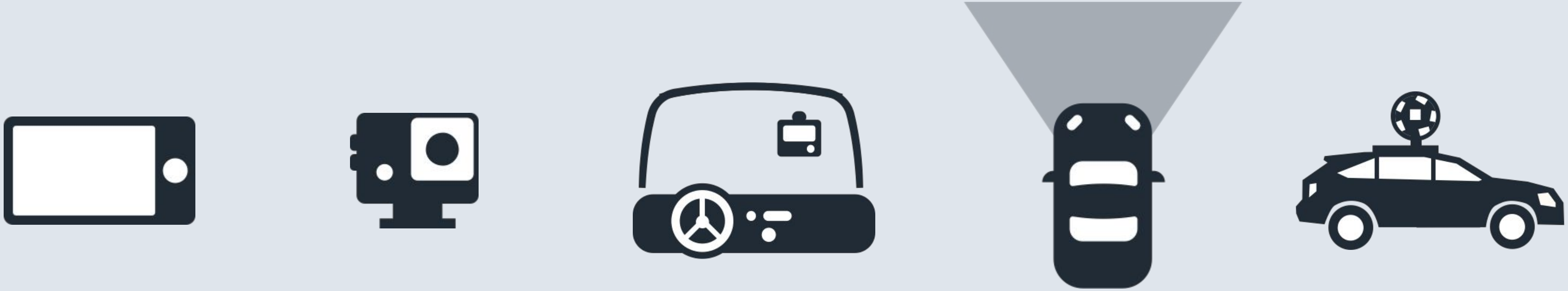

Who We Are

Mapillary is the street-level imagery platform that scales and automates mapping using collaboration, cameras, and computer vision



Map data at scale from street-level imagery

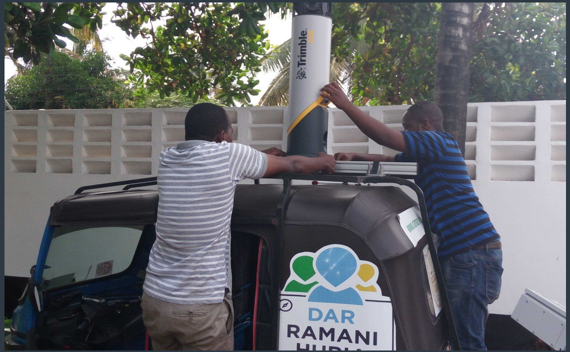
Anyone With Any Camera, Anywhere



560+ million images, >7.6 million km, 38+ billion objects

Empowering A Global Community Of Collaborators

Individuals



NGOs

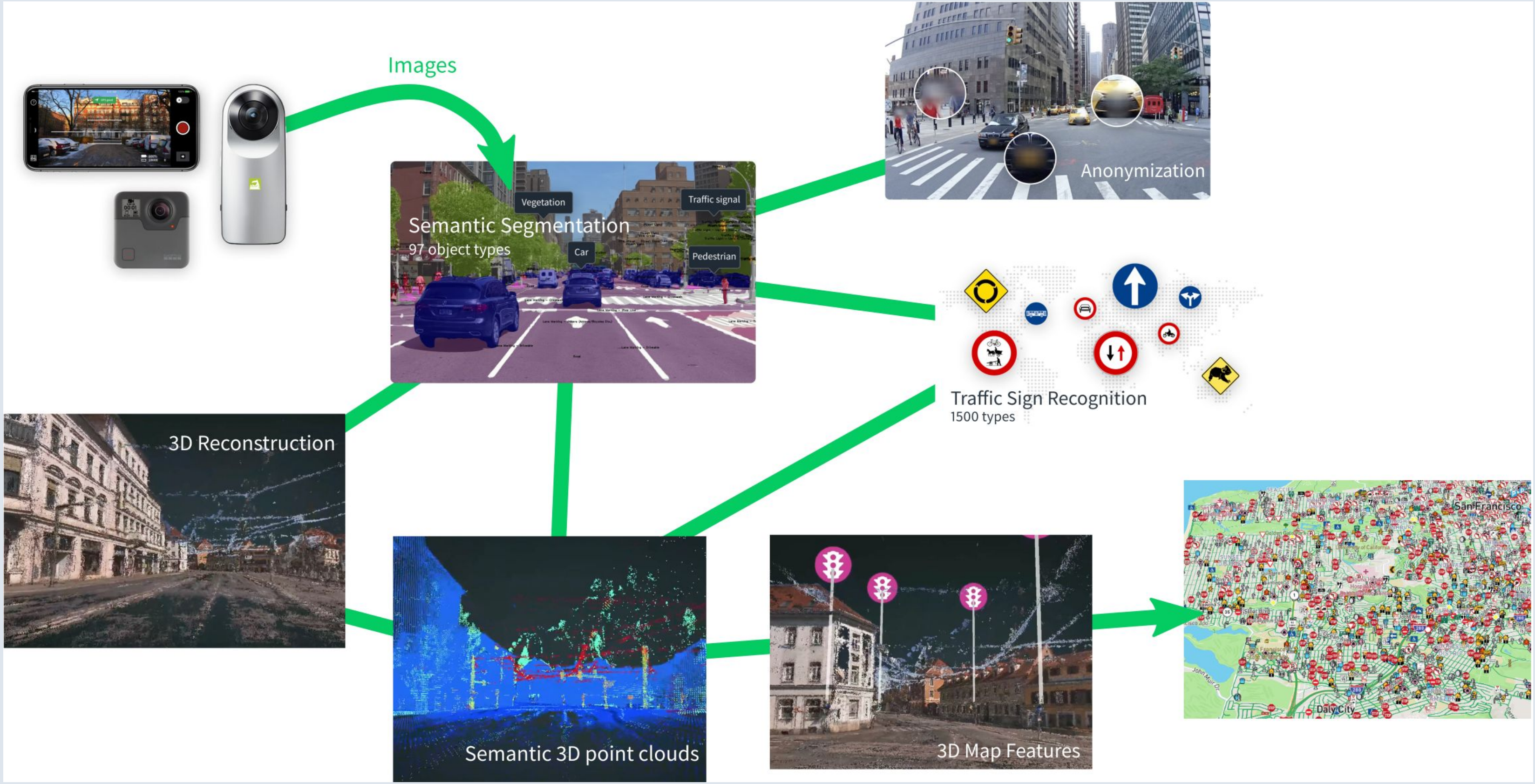
Municipalities & Public Agencies

Geospatial Services

Fleets

Mapping Companies

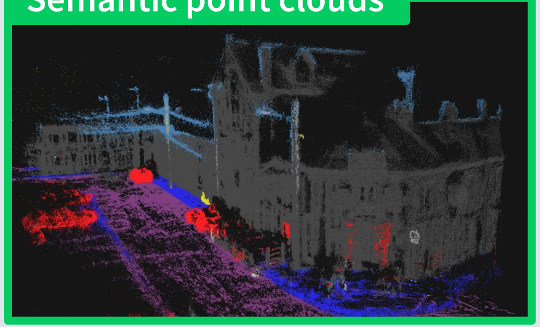
From Images to Map Data



Recognition Algorithms at Work

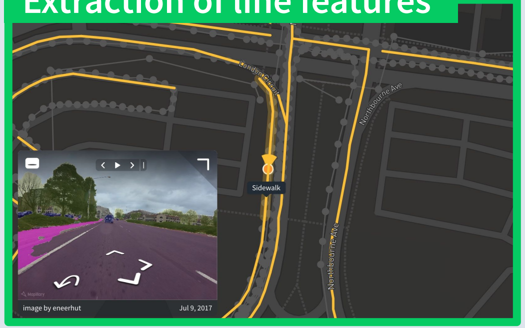
Map data for 97 classes

Semantic point clouds

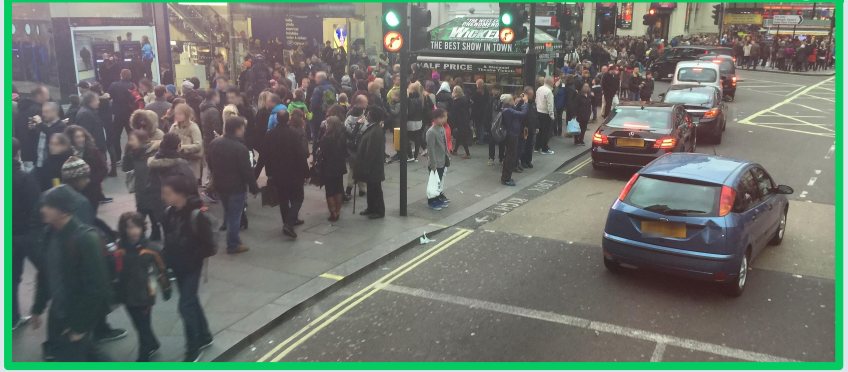


1500 traffic sign classes >100 countries

Extraction of line features



Privacy protection: face and licence place blurring



Research @ Mapillary

Meet Mapillary’s Research Team!

Peter

Lorenzo

Arno

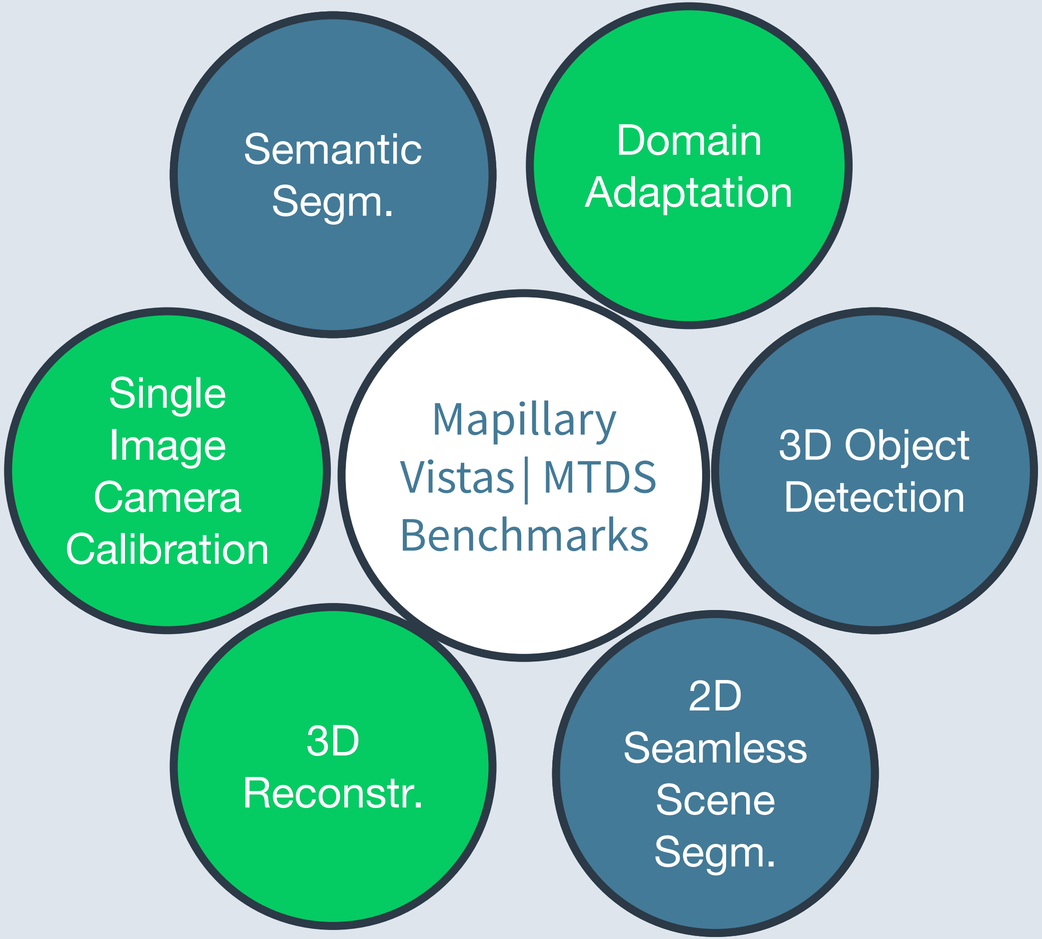
Samuel

Aleksander

Andrea

Mapillary Technology Stack

Select Research Interests



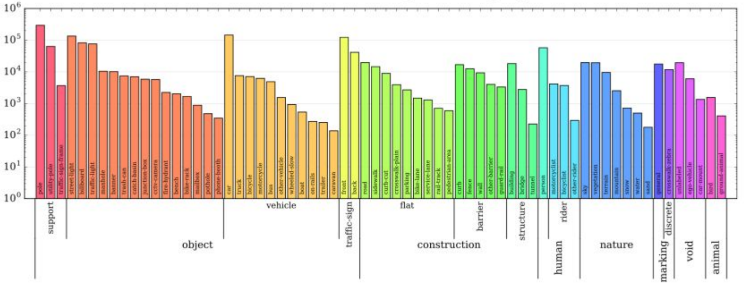
Benchmark Data

The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes

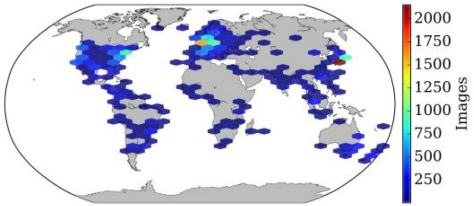
G. Neuhold, T. Ollmann, S. Rota Bulò, P. Kontschieder. (ICCV 2017)
Mapillary Research

Vistas Features and Statistics

Labeled instances per category

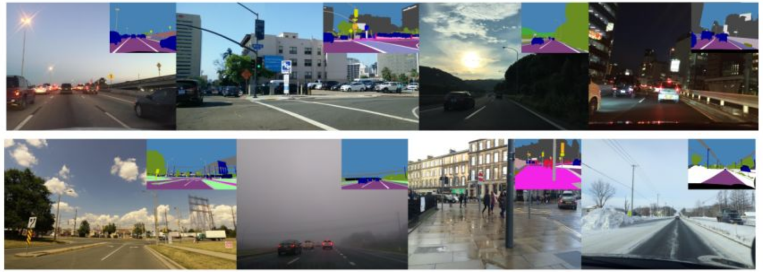


Global geographic reach (6 continents)

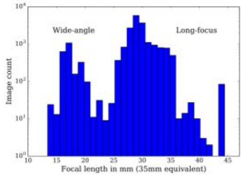


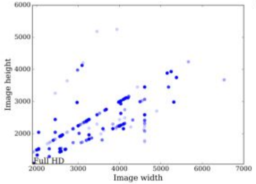
Diverse viewpoints from roads, sidewalks and off-road

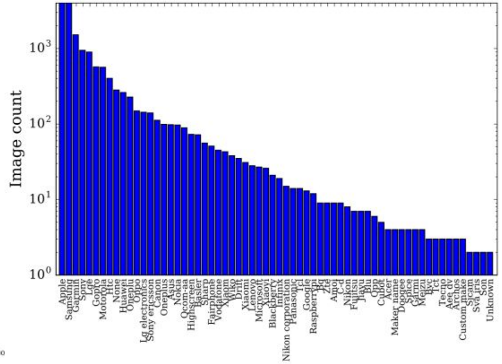
Various weather conditions and capture times



Wide variety of camera sensors, focal lengths
image aspect ratios and types of camera noise







Mapillary Vistas Dataset (ICCV 2017)

- Most diverse publicly available semantic segmentation dataset with street-level imagery
- 25k high-res images with pixel-wise annotations (18k train / 2k val / 5k test)
- 65 object classes, 37 instance-specific (research edition free for non-commercial purposes)
- Global geographic reach, covering 6 continents
- Diverse viewpoints: Roads, sidewalks, off-road
- Wide variety of camera sensors, focal lengths, image aspect ratios, and types of camera noise
- Various weather conditions and capture times

`https://www.mapillary.com/dataset/vistas`

Mapillary Traffic Sign Dataset (MTSD)

- The only publicly available traffic sign dataset with worldwide data
- Largest and most diverse traffic sign dataset
- 52K images with 257K traffic sign annotations
- 48K nearby images with propagated annotations
- 313 traffic sign classes
- Covering most countries from all continents
- Similar image properties as in Vistas

Semantic & Panoptic Segmentation

Map data recognition

Focus on small & underrepresented objects

Deep Architectures

Backbone (encoder)

Head (decoder)

From higher to lower resolution

Few to many feature channels

Features at different scales

Potential to combine different modalities

From lower to higher resolution

Reduction of feature channels

Agglomerate contextual information

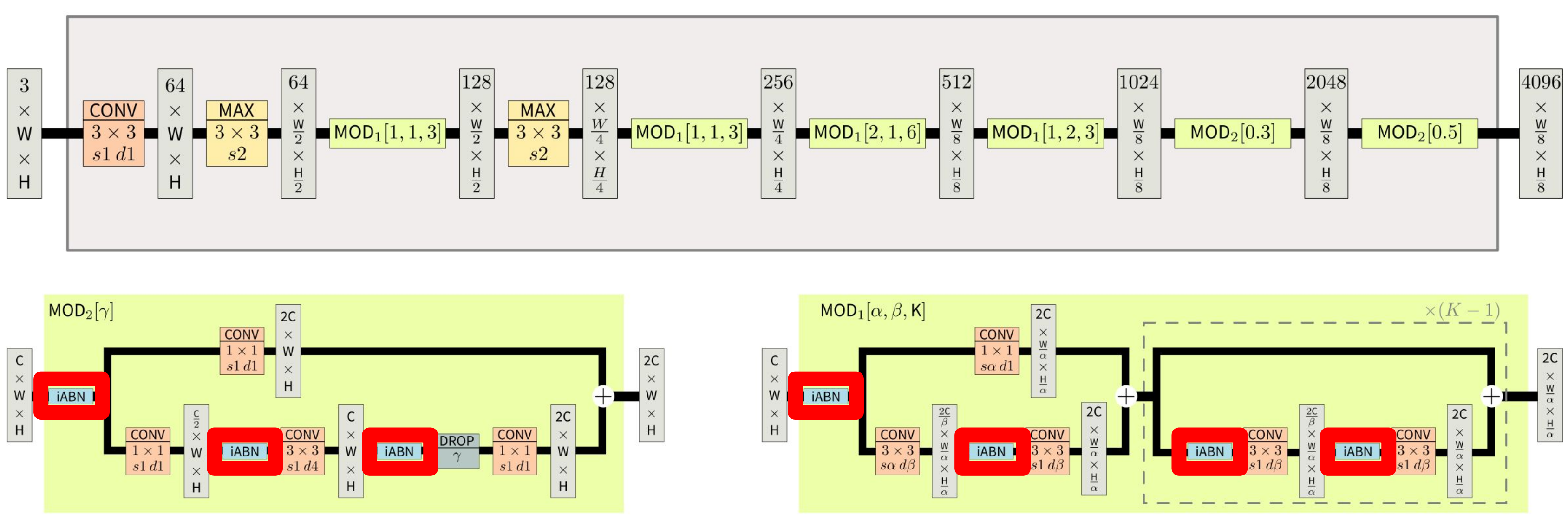
Provide pixel-specific predictions

Multi-task learning for instance segment.

Mapillary's Working Horse: Wide ResNet

ResNet with reduced depth but wider layers (more feature channels)

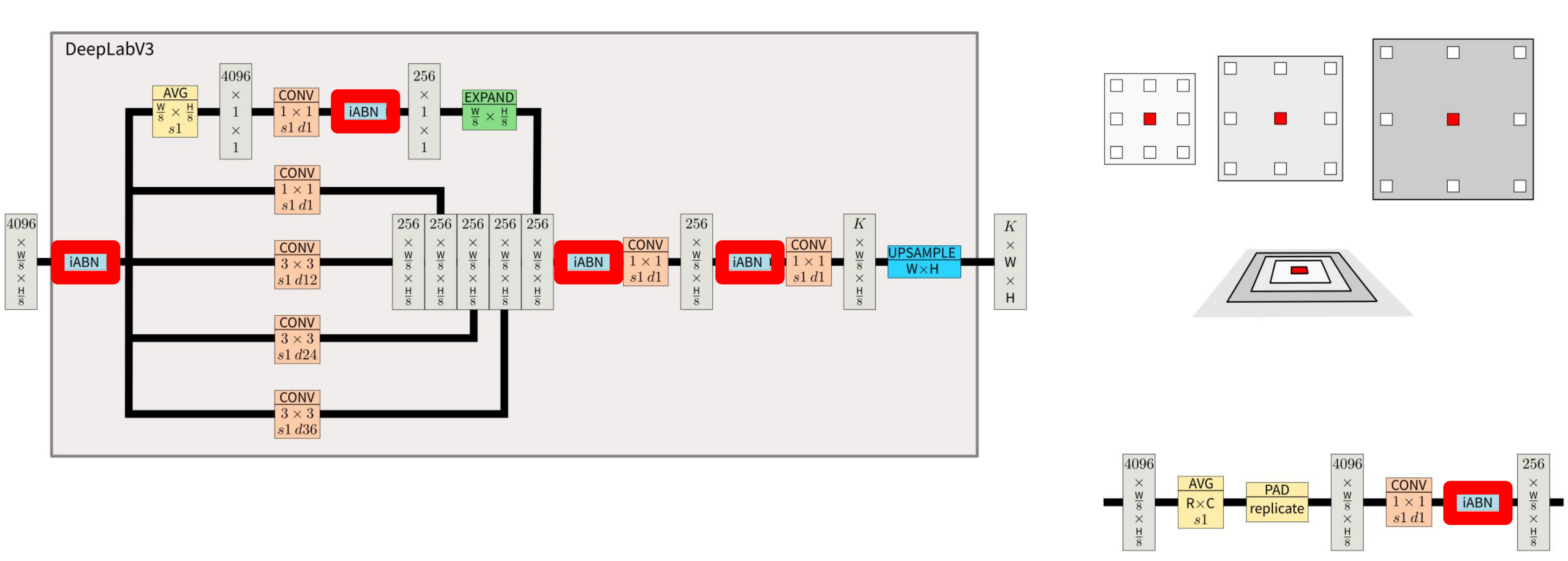
Wide ResNet-38



Wu et al., Wider or deeper: Revisiting the ResNet model for visual recognition. In **PR**, 2019

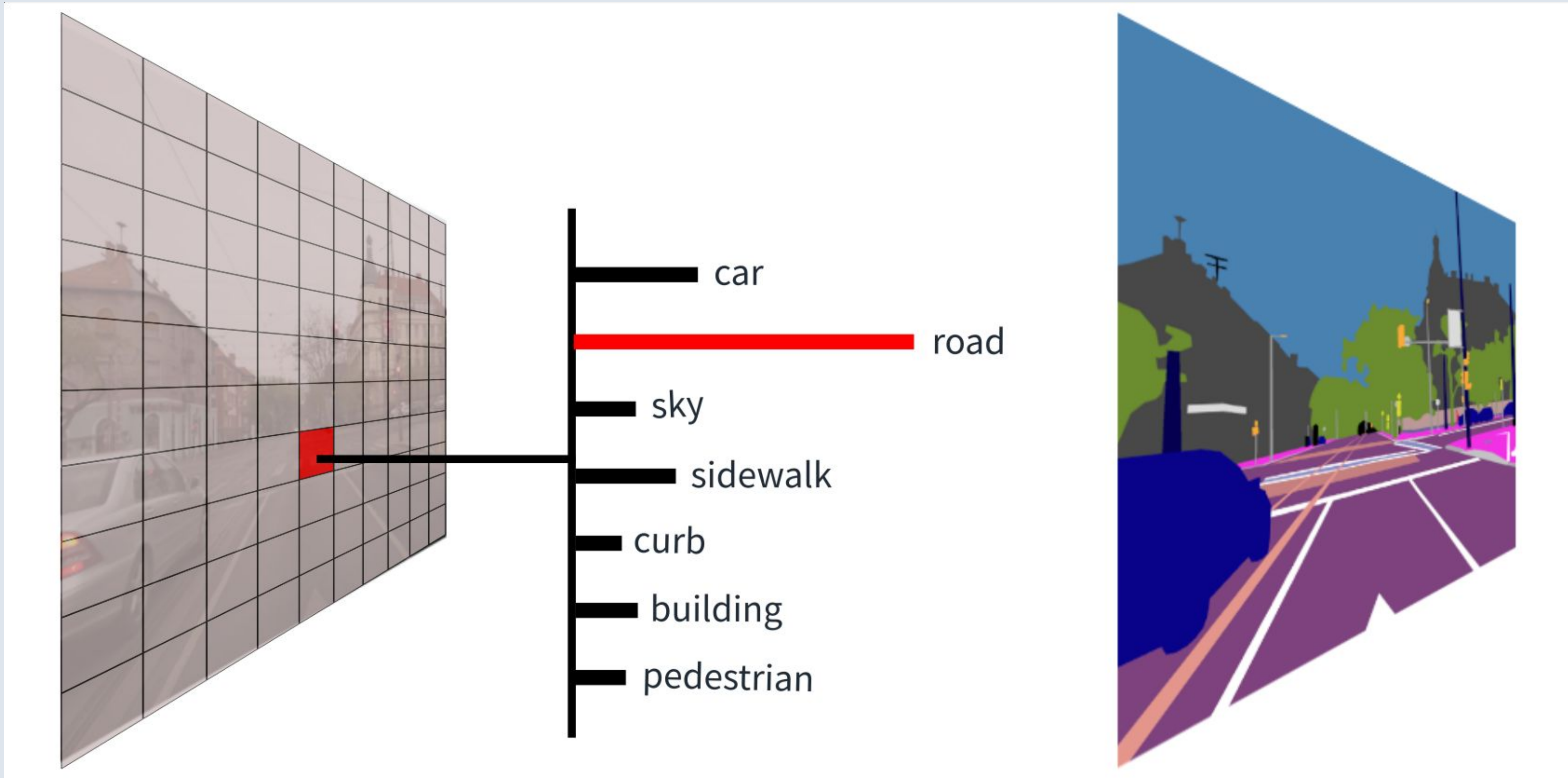
DeepLabV3 Head

Combine global pooling and increasing, dilated convolutions for learning of context



Chen et al., Rethinking Atrous Convolution for Semantic Image Segmentation, **arXiv 2018**

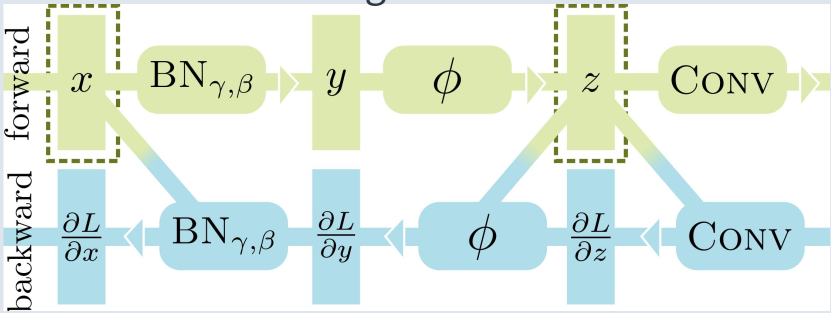
Semantic Segmentation Predictions



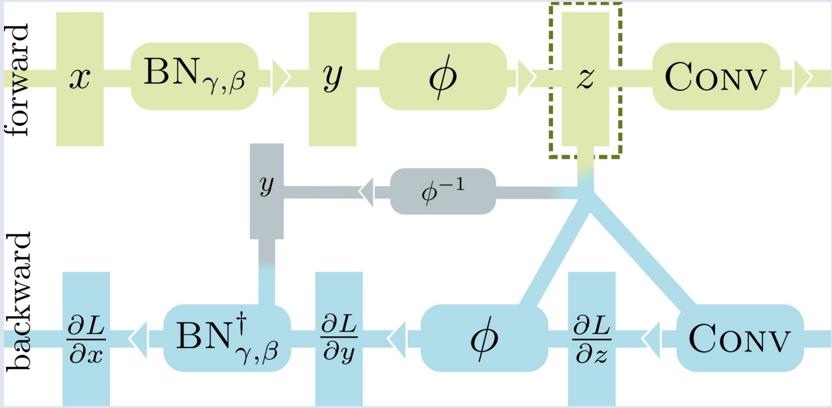
Improving object recognition

Overcoming lack of memory

Conventional setting



In-Place Activated BatchNorm



Code available on arXiv!

Gains approximately 50% GPU memory during training at minor computational overhead (< 2%)

In-Place Activated BatchNorm for Memory-Optimized Training of DNNs. **CVPR'18**

Improving Object Recognition

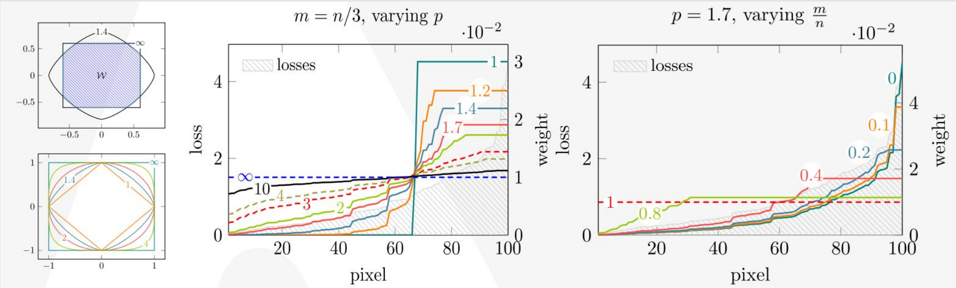
Focus attention of learning algorithm on difficult samples

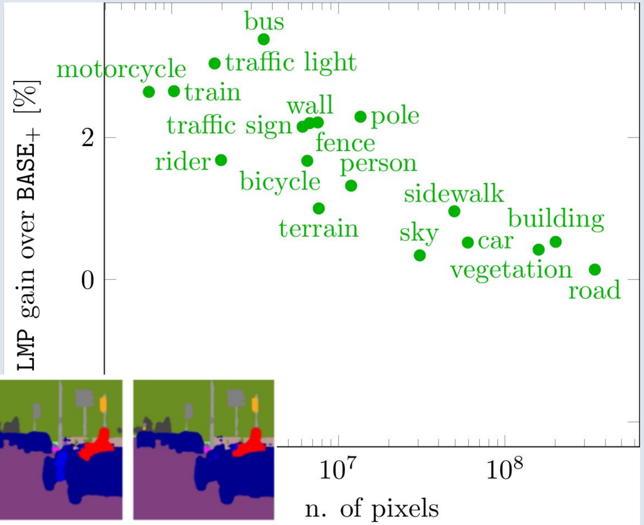
STANDARD LOSS

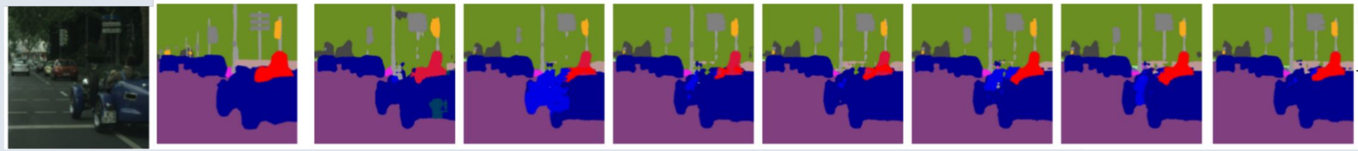
$$L(\hat{y}, y) = \frac{1}{|\mathcal{I}|} \sum_{u \in \mathcal{I}} \ell_{\hat{y}y}(u)$$

LOSS MAX-POOLING

$$L_{\mathcal{W}}(\hat{y}, y) = \max \left\{ \sum_{u \in \mathcal{I}} w(u) \ell_{\hat{y}y}(u) : w \in \mathcal{W} \right\}$$







Loss Max-Pooling for Semantic Segmentation. **CVPR'17**

Semantic Segmentation Results

Experimental Results

TASKS & DATASETS

Image Classification on ImageNet
Semantic Segmentation on Mapillary Vistas, Cityscapes, COCO-Stuff, Kitti, WildDash, ScanNet

NETWORKS

ResNeXt-101/152
WideResNet-38
DenseNet-264
+ DeepLabV3 head

TYPES

Fixed crop, max batch size
Fixed batch size, max input res.
With or w/o synchronized BN

Image Classification

ImageNet (val)

Network		224^2 center		224^2 10-crops		320^2 center	
	batch size	top-1	top-5	top-1	top-5	top-1	top-5
ResNeXt-101, STD-BN	256	77.04	93.50	78.72	94.47	77.92	94.28
ResNeXt-101, INPLACE-ABN	512	78.08	93.79	79.52	94.66	79.38	94.67
ResNeXt-152, INPLACE-ABN	256	78.28	94.04	79.73	94.82	79.56	94.67
WideResNet-38, INPLACE-ABN	256	79.72	94.78	81.03	95.43	80.69	95.27
DenseNet-264, INPLACE-ABN	256	78.57	94.17	79.72	94.93	79.49	94.89
ResNeXt-101, INPLACE-ABN^{sync}	256	77.70	93.78	79.18	94.60	78.98	94.56

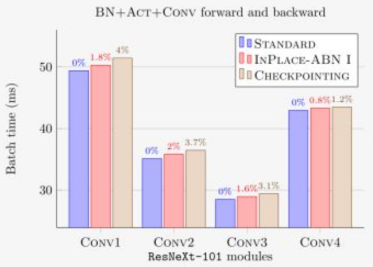
Effect of RELU vs LEAKYRELU on ImageNet (val)

Network	activation		224^2 center		224^2 10-crops		320^2 center	
	training	validation	top-1	top-5	top-1	top-5	top-1	top-5
ResNeXt-101	RELU	RELU	77.74	93.86	79.21	94.67	79.17	94.67
ResNeXt-101	RELU	LEAKY RELU	76.88	93.42	78.74	94.46	78.37	94.25
ResNeXt-101	LEAKY RELU	LEAKY RELU	77.04	93.50	78.72	94.47	77.92	94.28
ResNeXt-101	LEAKY RELU	RELU	76.81	93.53	78.46	94.38	77.84	94.20

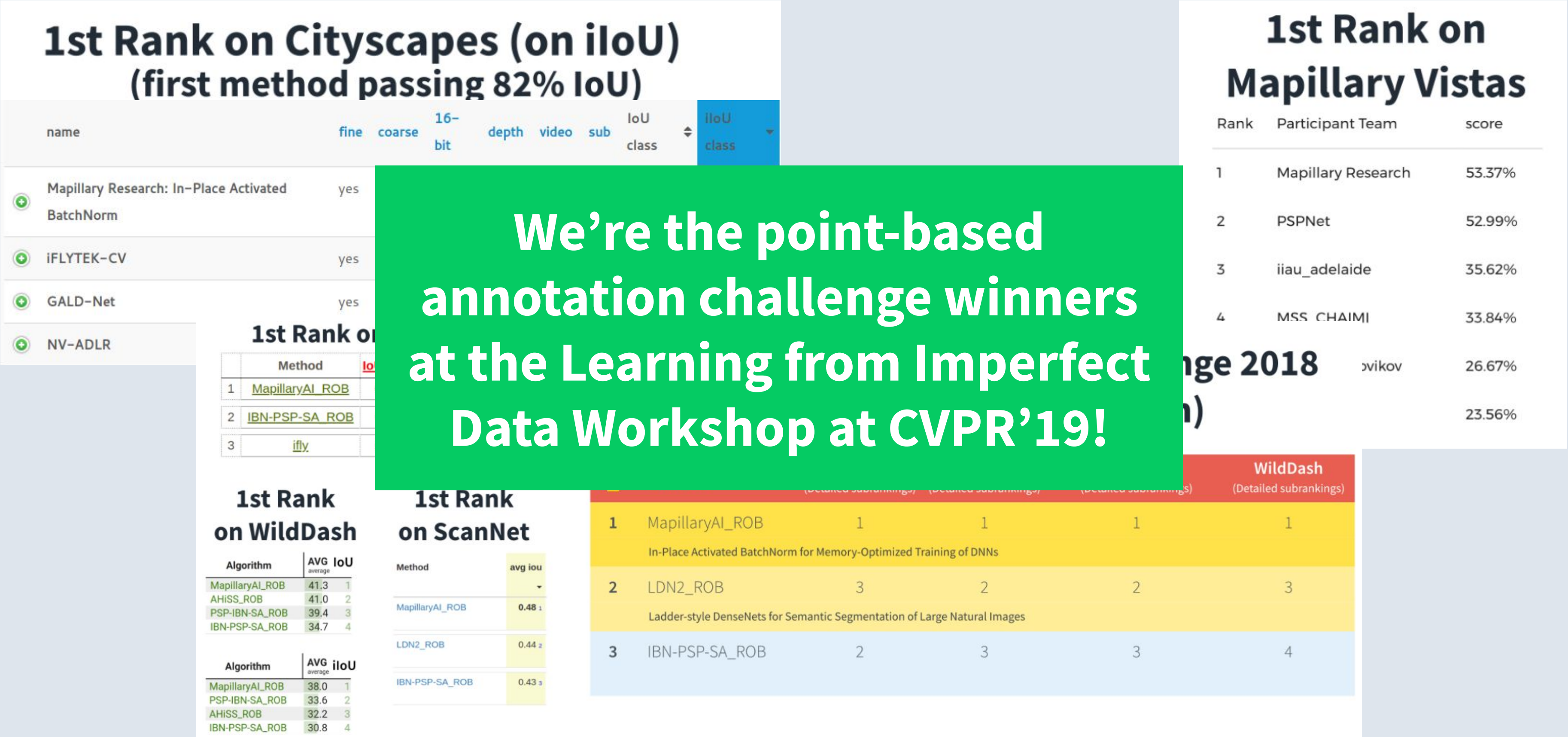
Semantic Segmentation

BATCHNORM	ResNeXt-101				WideResNet-38			
	Cityscapes		COCO-Stuff		Cityscapes		COCO-Stuff	
STD-BN + LEAKY RELU	16×512^2	74.42	16×480^2	20.30	20×512^2	75.82	20×496^2	22.44
INPLACE-ABN, FIXED CROP	28×512^2 [+75%]	75.80	24×480^2 [+50%]	22.63	28×512^2 [+40%]	77.75	28×496^2 [+40%]	22.96
INPLACE-ABN, FIXED BATCH	16×672^2 [+72%]	77.04	16×600^2 [+56%]	23.35	20×640^2 [+56%]	**78.31**	20×576^2 [+35%]	24.10
INPLACE-ABN^{sync}, FIXED BATCH	16×672^2 [+72%]	**77.58**	16×600^2 [+56%]	**24.91**	20×640^2 [+56%]	78.06	20×576^2 [+35%]	**25.11**
Cityscapes **val** (single model & scale)	12×872^2	79.16	Cityscapes **val** (single model & scale) + CLASS-UNIFORM SAMPLING				12×872^2	**79.40**
Cityscapes **test** (single Vistas pre-trained model, 5 scales + horizontal flipping, fine + coarse label data) + CLASS-UNIFORM SAMPLING							12×872^2	**82.03**
Mapillary Vistas **val** (single model & scale, no horizontal flipping) + CLASS-UNIFORM SAMPLING							12×776^2	**53.12**
Mapillary Vistas **test** (single model & scale, no horizontal flipping) + CLASS-UNIFORM SAMPLING							12×776^2	**53.37**

Computation Time



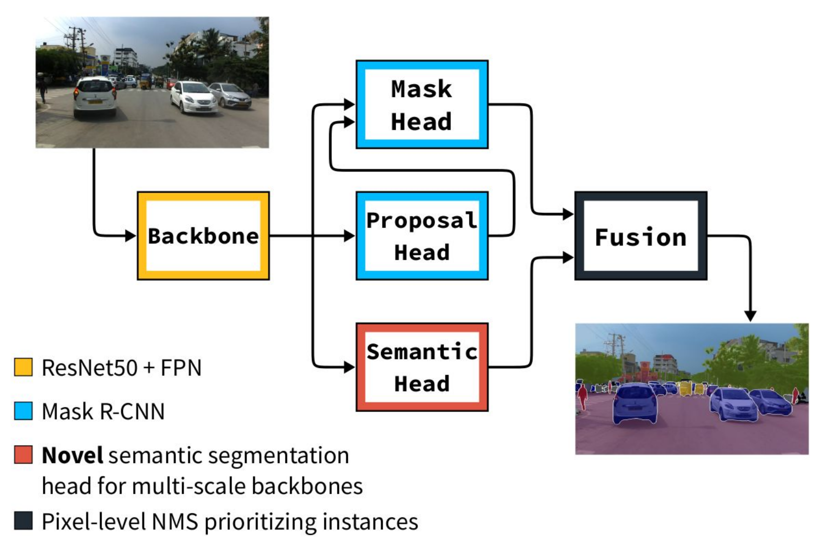
Semantic Segmentation Results

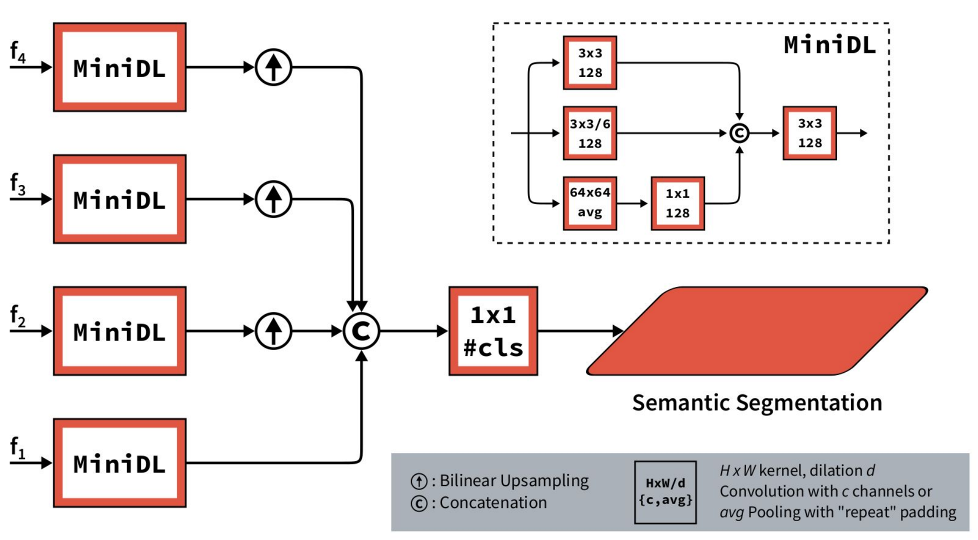


We're the point-based annotation challenge winners at the Learning from Imperfect Data Workshop at CVPR'19!

Seamless Scene Segmentation

Unified approach for semantic & instance-specific segmentation





Join us at our Poster on Wednesday (#42, Session 2.2)!

Panoptic Segmentation Results

Method	Body	Data	Cityscapes PQ	Cityscapes PQ_{St}	Cityscapes PQ_{Th}	Cityscapes $PQ^{\dagger}$	Cityscapes AP_M	Cityscapes IoU	Vistas PQ	Vistas PQ_{St}	Vistas PQ_{Th}	Vistas $PQ^{\dagger}$	Vistas AP_M	Vistas IoU
de Geus *et al.* [1]	R50	I	-	-	-	-	-	-	17.6	27.5	10.0	-	-	34.7
Supervised in [2]	R101	I	47.3	52.9	39.6	-	24.3	71.6	-	-	-	-	-	-
FPN-Panoptic [3]	R50	I	57.7	62.2	51.6	-	32.0	75.0	-	-	-	-	-	-
TASCNet [4]	R50	I+C	59.2	61.5	56.0	-	37.6	77.8	32.6	34.4	31.1	-	18.5	-
UPSNet [5]	R50	I	59.3	62.7	54.6	-	33.3	75.2	-	-	-	-	-	-
DeeperLab [6]	X71	I	56.3	-	-	-	-	-	32.0	-	-	-	-	55.3
Ours independent	R50	I	59.8	64.5	53.4	59.0	31.9	75.4	37.2	42.5	33.2	38.6	16.3	50.2
Ours combined	R50	I	60.3	63.3	56.1	59.6	33.6	77.5	37.7	42.9	33.8	39.0	16.4	50.4

Method	Body	Data	Indian Driving Dataset PQ	PQ_{St}	PQ_{Th}	$PQ^{\dagger}$	AP_M	IoU
Ours independent	R50	I	47.2	46.6	48.3	48.8	29.8	67.2
Ours combined	R50	I	46.9	45.9	48.7	48.5	29.8	67.5

CURRENT BEST PQ!
(among comparable backbones)


WC

3D Object Recognition

Monocular, Single RGB Image-based 3D Detection

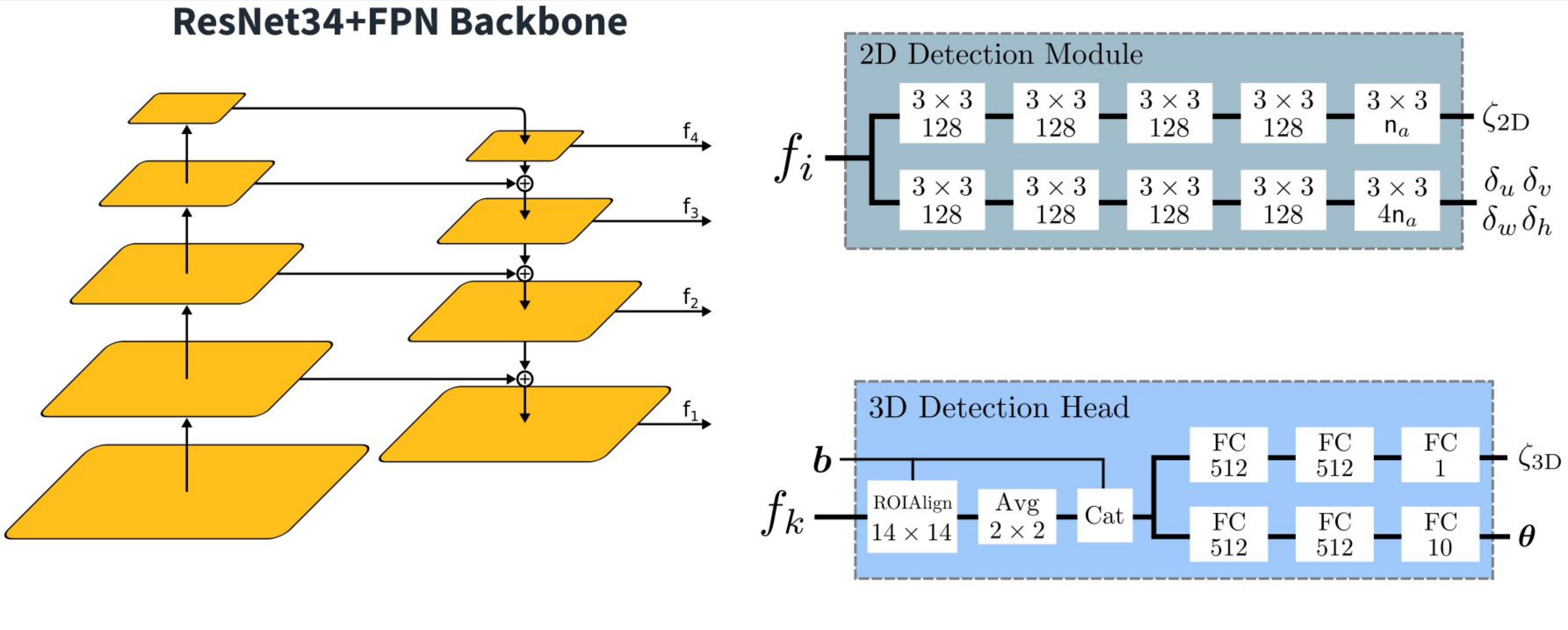
Given a single RGB image, provide 3D object detection (box) predictions in camera coordinates for each relevant object category

Disentangling Monocular 3D Object Detection

Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Manuel Lopez-Antequera, Peter Kontschieder

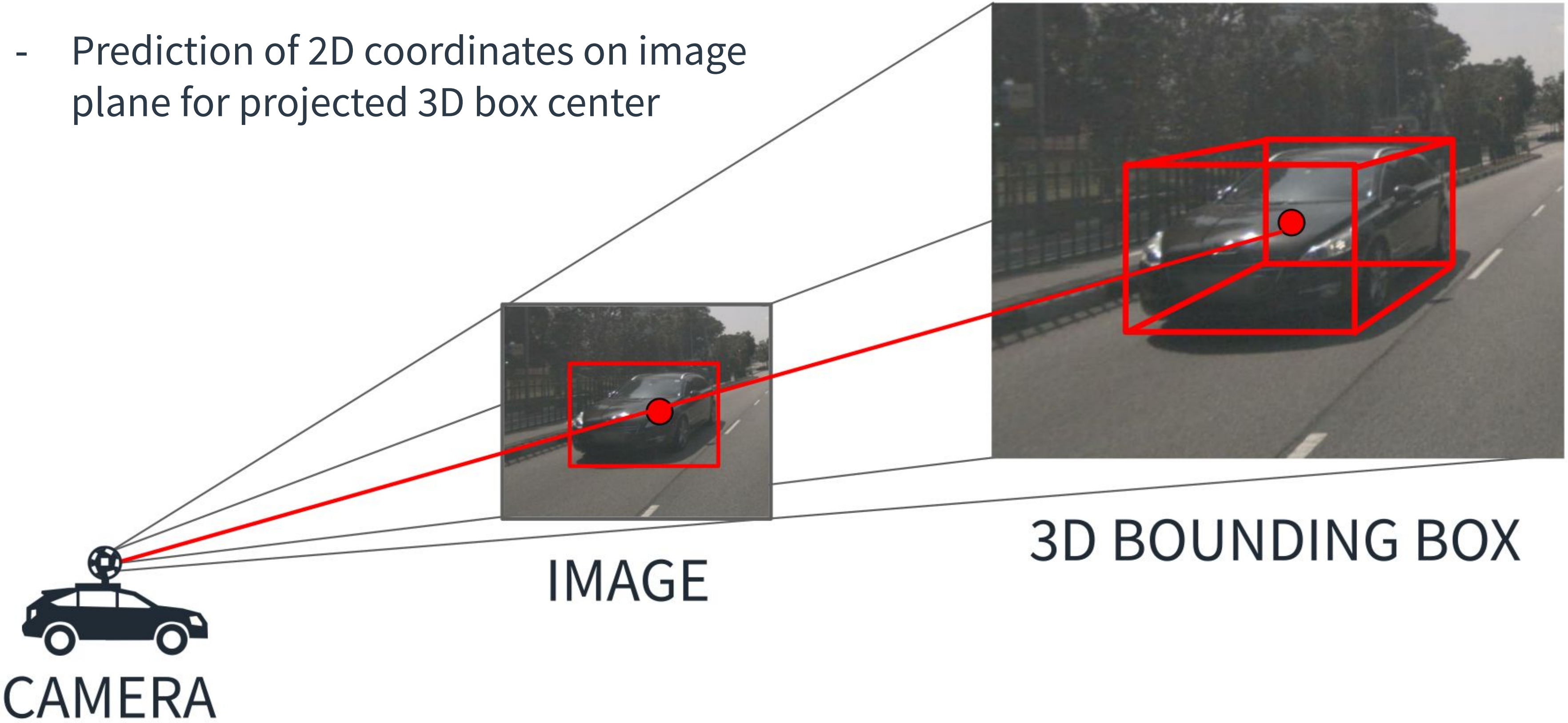
Mapillary Research

Network Architecture

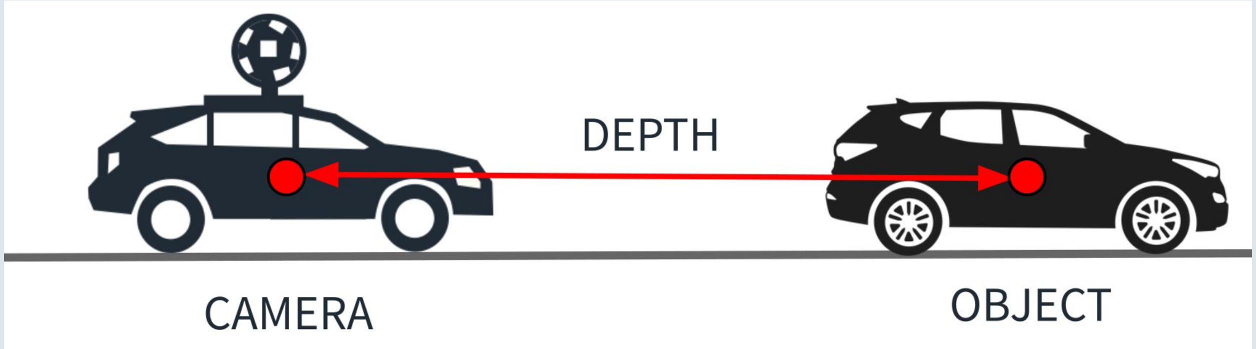


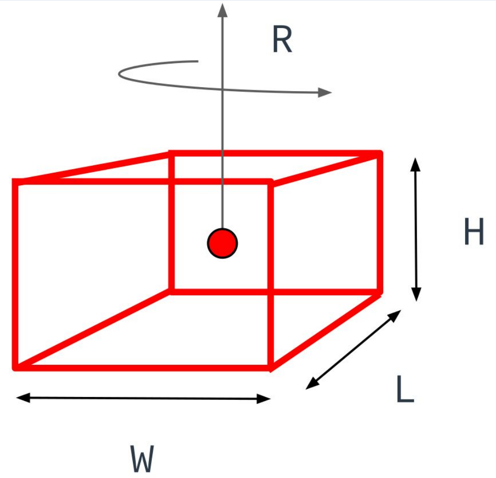
Predictions per Detection Hypothesis

- Prediction of 2D coordinates on image plane for projected 3D box center



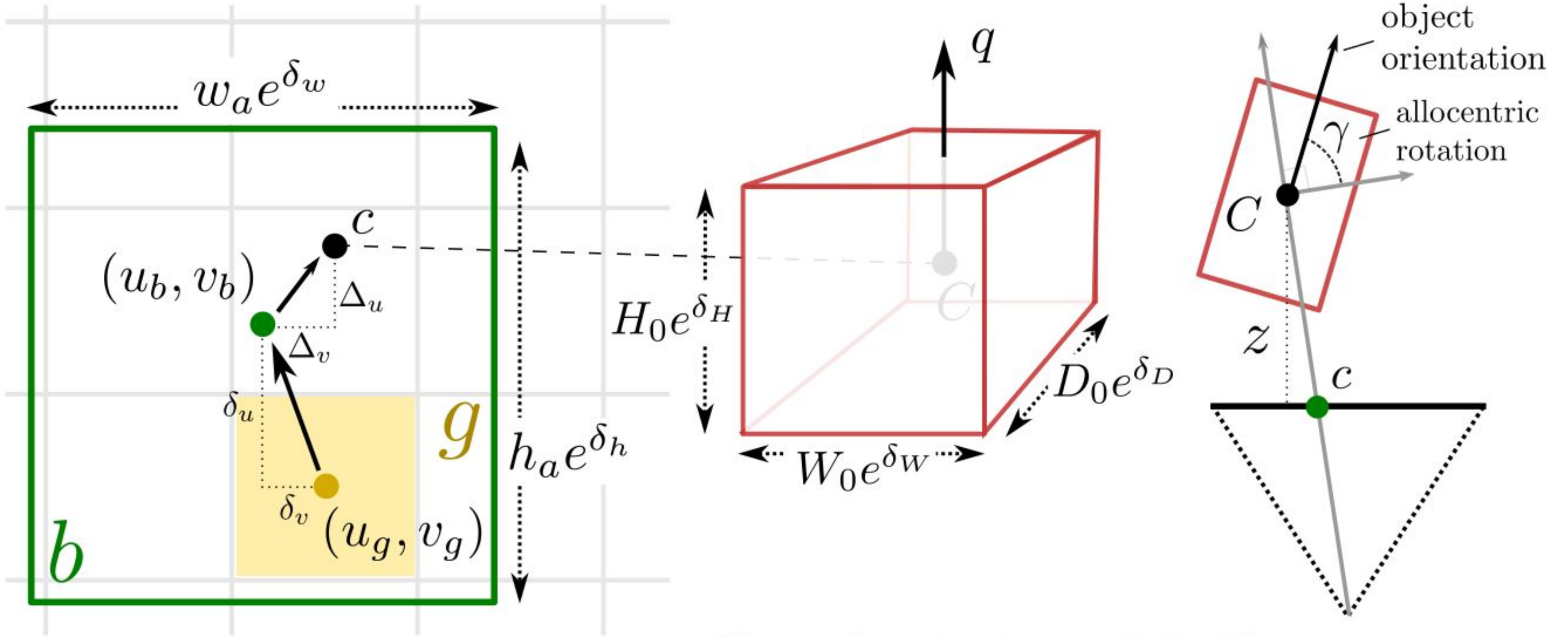
Predictions per Detection Hypothesis (ct’d)





- Allocentric rotation quaternion R of 3D bounding box
- 3D bounding box size (H/W/L)
- Object depth (distance to camera)

Parameterization of Outputs

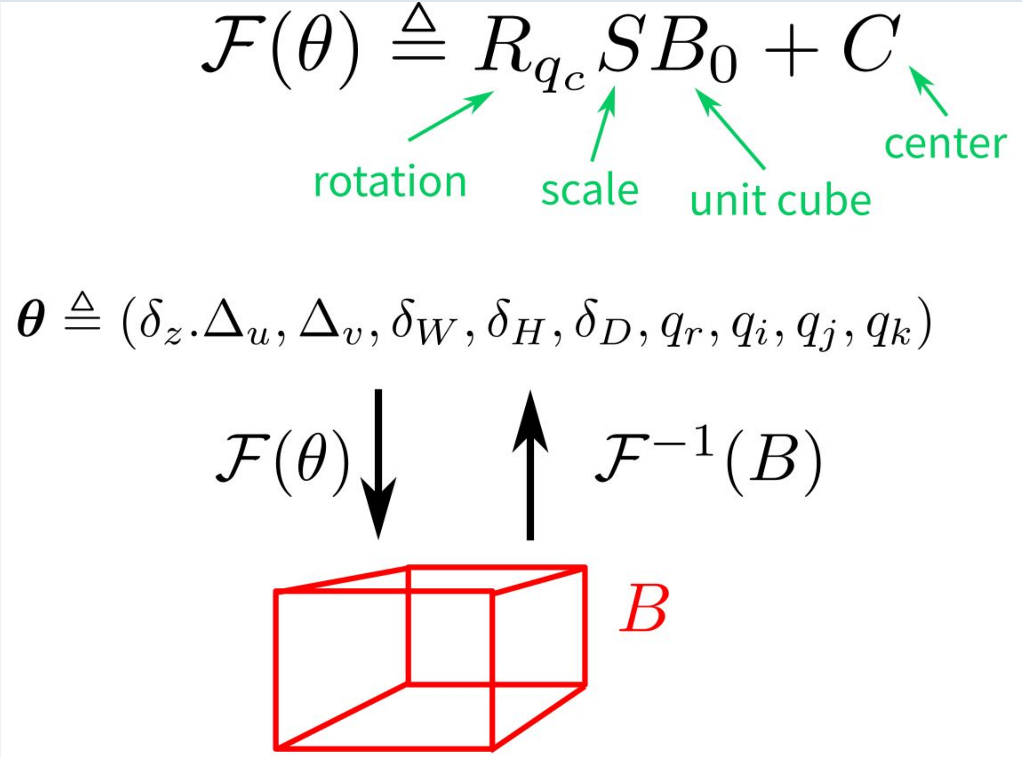


$$\gamma \triangleq q_r + q_i \mathbf{i} + q_j \mathbf{j} + q_k \mathbf{k}$$
$$z \triangleq \mu_z + \sigma_x \delta_z$$

Network outputs per detection

$$\boldsymbol{\theta} \triangleq (\delta_z . \Delta_u, \Delta_v, \delta_W, \delta_H, \delta_D, q_r, q_i, q_j, q_k)$$

Lifting Transform



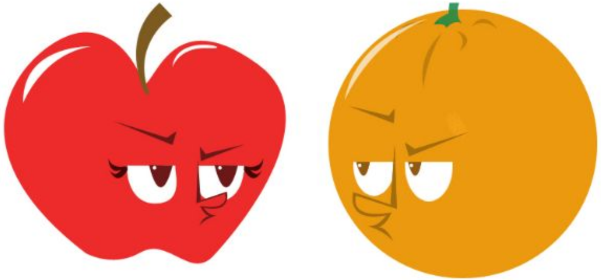
Manhardt et al., ROI-10D: Monocular Lifting of 2D Detection to 6D Pose and Metric Shape. In **CVPR**, 2019

Network Output Regression Loss

Ground-truth bounding box B

$$\theta^* \triangleq \mathcal{F}^{-1}(B)$$

Network output θ



$$\boldsymbol{\theta} \triangleq (\delta_z . \Delta_u, \Delta_v, \delta_W, \delta_H, \delta_D, q_r, q_i, q_j, q_k)$$

$\sum$ indepedent regression losses

$$\boldsymbol{\theta}^* \triangleq (\delta_z^* . \Delta_u^*, \Delta_v^*, \delta_W^*, \delta_H^*, \delta_D^*, q_r^*, q_i^*, q_j^*, q_k^*)$$

Not directly comparable

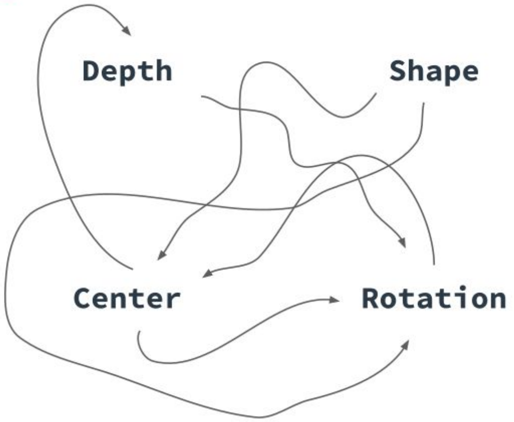
Directly Optimizing 3D Box Coordinates

3D Bounding Box Loss

$$L^{\mathrm{bb}}_{\mathrm{3D}}(B, \hat{B}) \triangleq \frac{1}{8}\|B - \hat{B}\|_H$$

prediction ground truth

Mutual dependences



Proposed Disentangling Transformation

Output space $\mathcal{Y}$ (*e.g.* 3D bounding boxes)

Loss function $L \in \mathbb{R}_+^{\mathcal{Y} \times \mathcal{Y}}$

$\psi \in \mathcal{Y}^{\mathbb{R}^d}$: 1-to-1 map from the set of network outputs $\Theta \subset \mathbb{R}^d$ to $\mathcal{Y}$

Outputs divided into groups: $\boldsymbol{\theta} \triangleq (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k)$

$$L_{\text{dis}}(y, \hat{y}) \triangleq \sum_{j=1}^{k} L(\psi(\boldsymbol{\theta}_j, \hat{\boldsymbol{\theta}}_{-j}), \hat{y})$$

ground truth

$$\hat{\boldsymbol{\theta}} = \psi^{-1}(\hat{y})$$

$$\boldsymbol{\theta} = \psi^{-1}(y)$$

Toy Example

Experimental Results

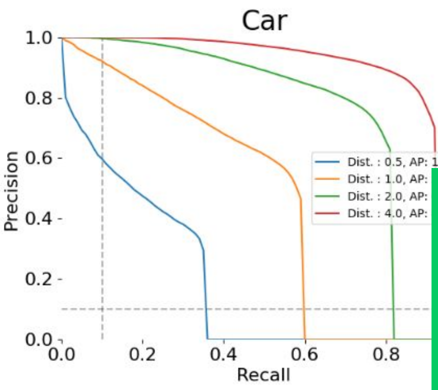
KITTI3D Cars

Method	2D detection Easy	2D detection Moderate	2D detection Hard	3D detection Easy	3D detection Moderate	3D detection Hard	Bird's eye view Easy	Bird's eye view Moderate	Bird's eye view Hard
Regression	66.50	72.30	66.00	1.60	1.50	1.20	2.70	2.10	2.30
3D BB	70.80	77.10	66.50	4.70	3.00	2.90	7.80	5.40	5.80
Regression w/ IoUDIS, 3DConf	67.20	73.60	65.50	3.20	2.90	2.00	5.80	4.80	4.30
3D BB w/ IoUDIS, 3DConf	90.20	88.40	78.40	15.40	13.60	12.00	20.50	16.20	15.70
3D BB w/ disentangling	76.40	80.30	73.20	4.90	3.40	3.10	7.30	5.70	6.30
MonoDIS	90.23	88.64	79.10	**18.05**	**14.98**	**13.42**	**24.26**	**18.43**	**16.95**
Single correct hypothesis per difficulty	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09
OFTNet [33]	–	–	–	4.07	3.27	3.29	11.06	8.79	8.91
Xu *et al*. [42]	–	–	–	7.85	5.39	4.73	19.20	12.17	10.89
FQNet [20]	–	–	–	5.98	5.50	4.75	9.50	8.02	7.71
Mono3D [4]	**93.89**	**88.67**	**79.68**	2.53	2.31	2.31	5.22	5.19	4.13
Mono3D++ [11]	–	–	–	10.60	7.90	5.70	16.70	11.50	10.10
ROI-10D [23]	78.57	73.44	63.69	10.12	1.76	1.30	14.04	3.69	3.56
ROI-10D w/ Depth [23]	89.04	88.39	78.77	7.79	5.16	3.95	10.74	7.46	7.06
ROI-10D w/ Depth, Synthetic [23]	85.32	77.32	69.70	9.61	6.63	6.29	14.50	9.91	8.73
MonoGRNet [29]	–	–	–	13.88	10.19	7.62	–	–	–
Best in [1]	–	–	–	13.96	7.37	4.54	–	–	–

Table 5: $AP|_{R_{11}}$ scores on KITTI3D (0.7 IoU threshold): Ablation results (white background), val set results of SOTA (grey background).

Experimental Results (ct’d)

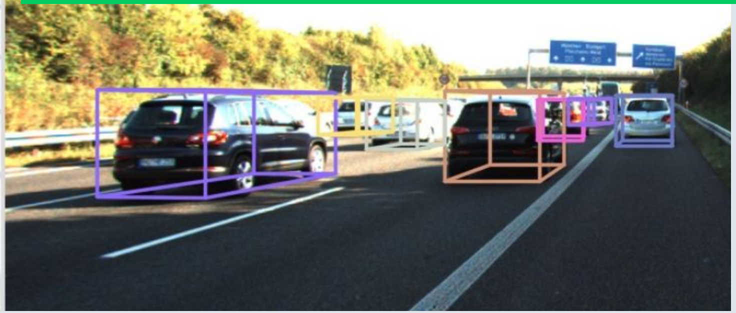
nuScenes Cars

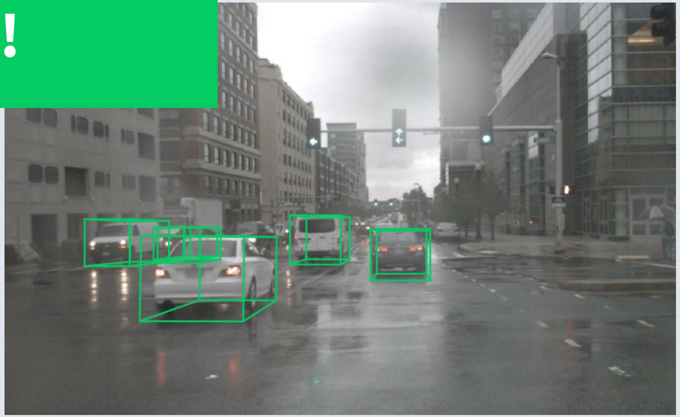


Method	AP_{Car} ↑ [%] 0.5m	1.0m	**2.0m**	4.0m	TP_{Car} ↓ ATE [m]	ASE [1-IoU]	AOE [rad]
PointPillar	55.5	71.8	76.1	78.6	0.27	0.17	0.19
OFTNet	–	–	27.0	–	0.65	0.16	0.18
				85.7	**0.61**	**0.15**	**0.08**

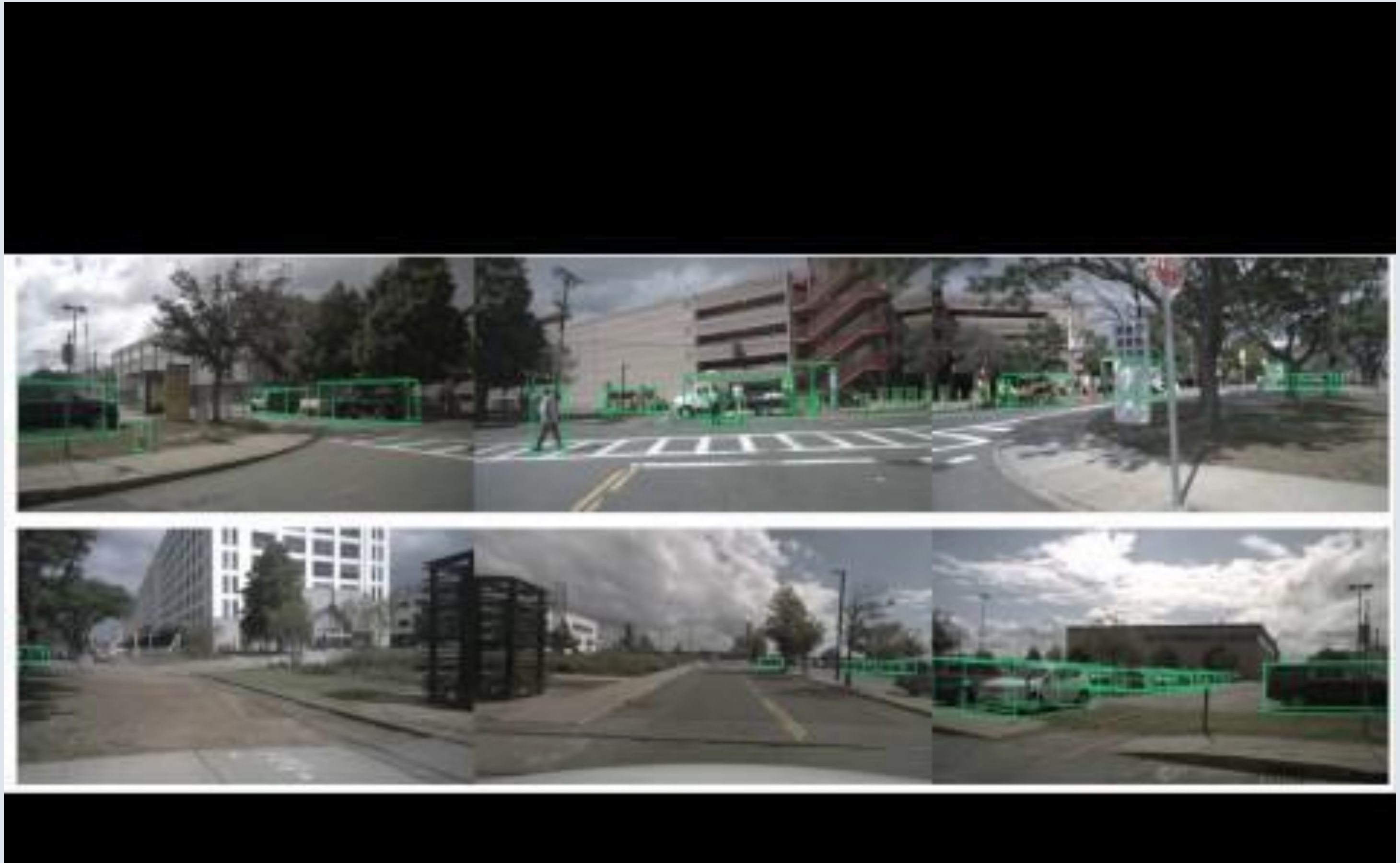
omparison for results on category 2]. Top row: LiDAR-based Point- ompleteness). Bottom: Available DIS.

We’re the nuScenes Vision-only challenge winners at the Workshop on Autonomous Driving at CVPR’19!





nuScenes Test Results



Embedding Semantics in 3D

[Demo link]

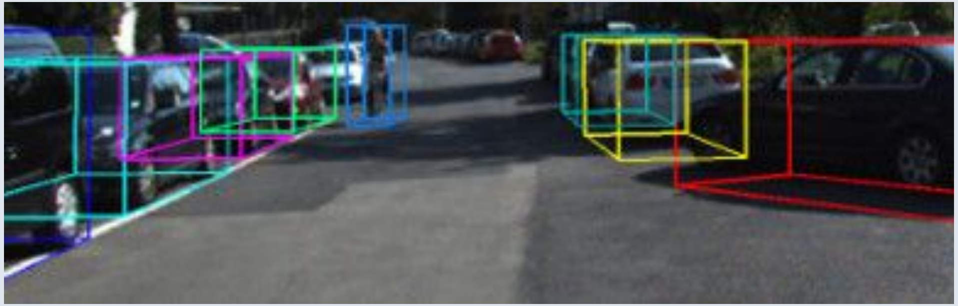
Metrics

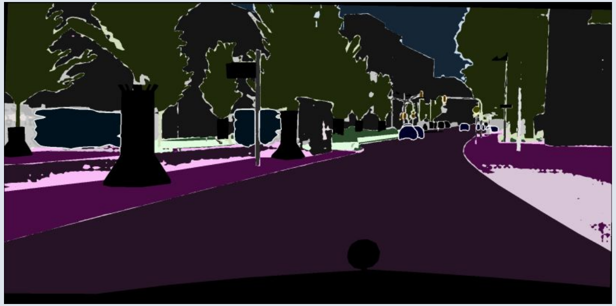
Exemplary Issues with Metrics

Can we adequately measure the performance on the tasks we want to solve?

11-Point Interpolated AP

PQ Metric





3D Object Detection on KITTI3D: Metric Issues

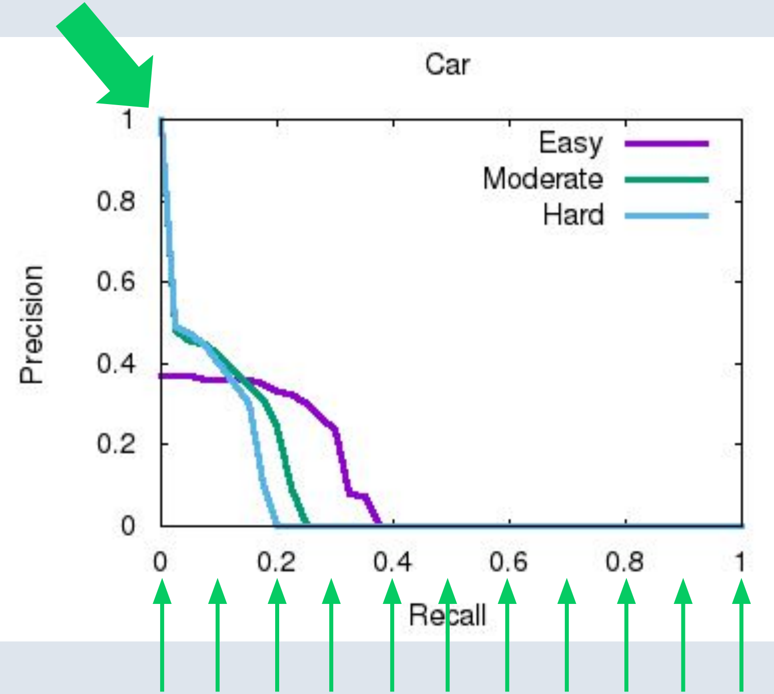
P/R curves for MonoDIS, generated from KITTI3D test server

$$\mathrm{AP}|_R = \frac{1}{|R|} \sum_{r \in R} \rho_{interp}(r)$$

$$\rho_{interp}(r) = \max_{r' : r' \geq r} \rho(r')$$

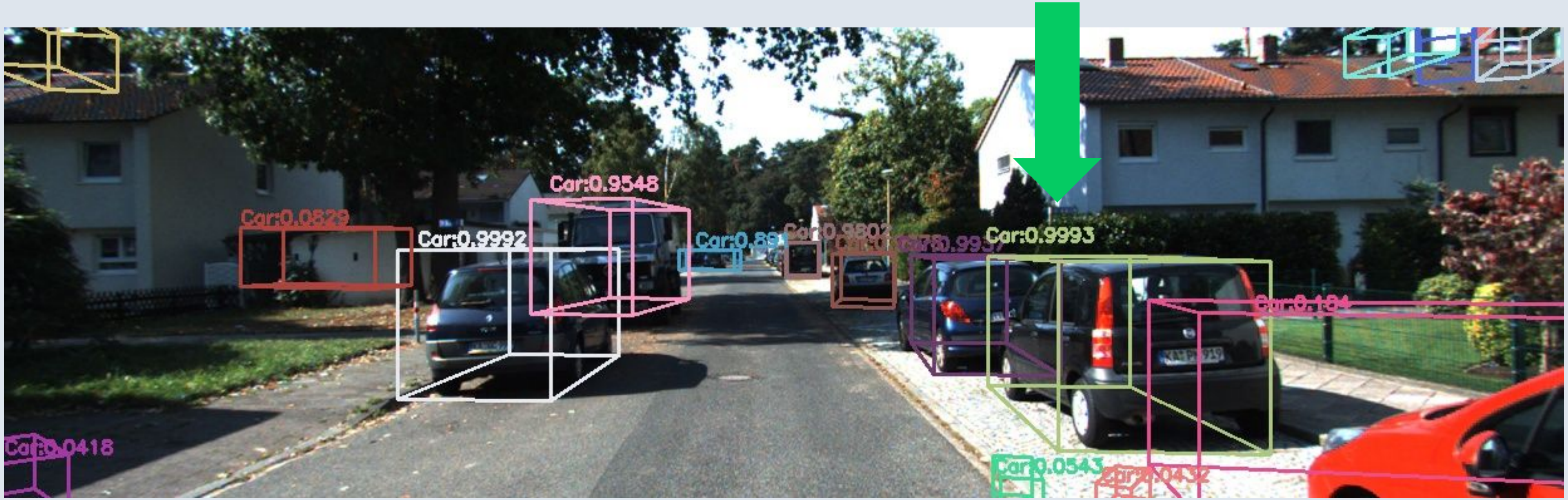
$\rho(r)$ gives the precision at recall r

$$R_{11} = \{0, 0.1, 0.2, \ldots, 1\}$$



Beating SOTA with a single detection!

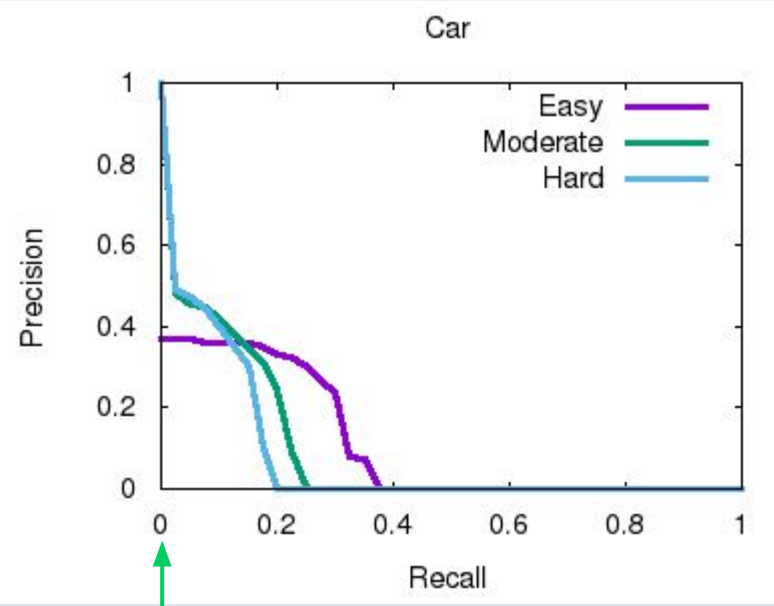
On KITTI3D: Assume we **keep only the single, best detection** per difficulty (among thousands of gt ones)



Beating SOTA with a single detection! (ct'd)

AP = 1/11 ~9.09%
(evaluating only on recall at 0)

$$R_{11} = \{0, \cancel{0.1, 0.2, \ldots, 1}\}$$



Results on KITTI3D (again)

Method	2D detection Easy	2D detection Moderate	2D detection Hard	3D detection Easy	3D detection Moderate	3D detection Hard	Bird's eye view Easy	Bird's eye view Moderate	Bird's eye view Hard
Regression	66.50	72.30	66.00	1.60	1.50	1.20	2.70	2.10	2.30
3D BB	70.80	77.10	66.50	4.70	3.00	2.90	7.80	5.40	5.80
Regression w/ IoUDIS, 3DConf	67.20	73.60	65.50	3.20	2.90	2.00	5.80	4.80	4.30
3D BB w/ IoUDIS, 3DConf	90.20	88.40	78.40	15.40	13.60	12.00	20.50	16.20	15.70
3D BB w/ disentangling	76.40	80.30	73.20	4.90	3.40	3.10	7.30	5.70	6.30
MonoDIS	90.23	88.64	79.10	**18.05**	**14.98**	**13.42**	**24.26**	**18.43**	**16.95**
Single correct hypothesis per difficulty	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09
OFTNet [33]	–	–	–	4.07	3.27	3.29	11.06	8.79	8.91
Xu *et al.* [42]	–	–	–	7.85	5.39	4.73	19.20	12.17	10.89
FQNet [20]	–	–	–	5.98	5.50	4.75	9.50	8.02	7.71
Mono3D [4]	**93.89**	**88.67**	**79.68**	2.53	2.31	2.31	5.22	5.19	4.13
Mono3D++ [11]	–	–	–	10.60	7.90	5.70	16.70	11.50	10.10
ROI-10D [23]	78.57	73.44	63.69	10.12	1.76	1.30	14.04	3.69	3.56
ROI-10D w/ Depth [23]	89.04	88.39	78.77	7.79	5.16	3.95	10.74	7.46	7.06
ROI-10D w/ Depth, Synthetic [23]	85.32	77.32	69.70	9.61	6.63	6.29	14.50	9.91	8.73
MonoGRNet [29]	–	–	–	13.88	10.19	7.62	–	–	–
Best in [1]	–	–	–	13.96	7.37	4.54	–	–	–

Panoptic Segmentation: PQ Metric Issues

Segment-specific assessment of segmentation quality.

Matching class-agnostic segments with IoU>0.5

Ich liebe PQ nicht

TOYOTA RESEARCH INSTITUTE

$$PQ = \frac{\sum_{(p,g)\in TP} IoU(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}$$

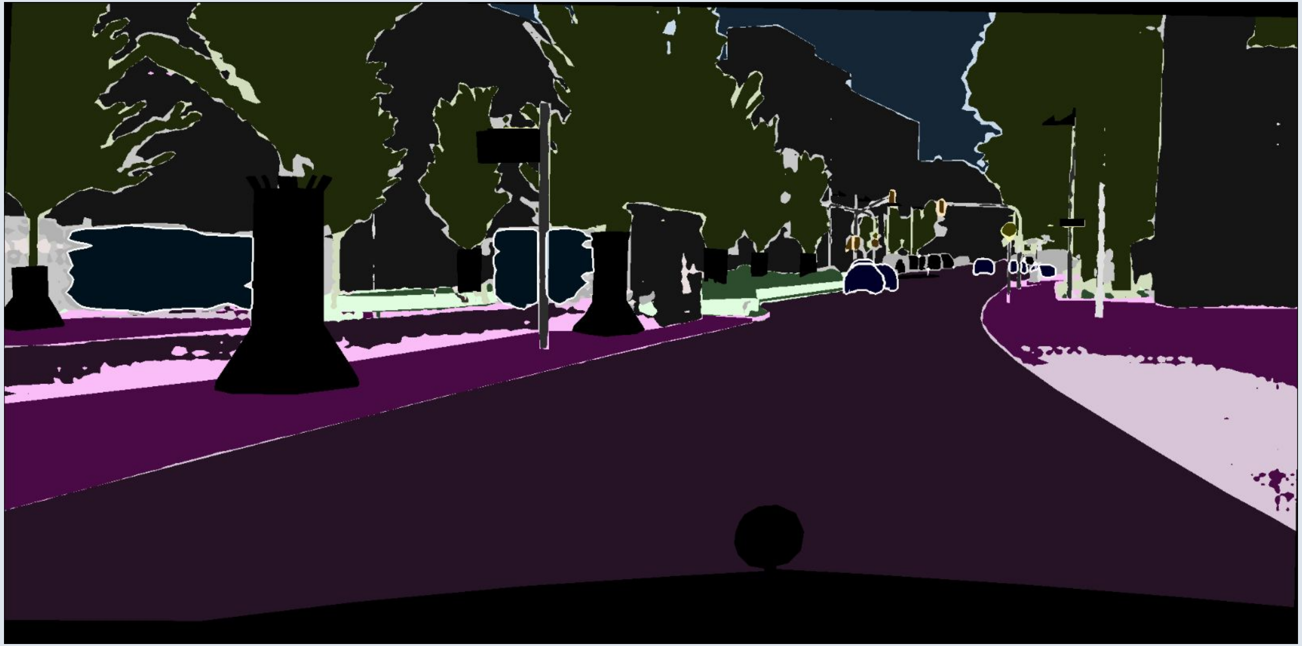
$$PQ = \underbrace{\frac{\sum_{(p,g)\in TP} IoU(p,g)}{|TP|}}_{\text{Segmentation quality (SQ)}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{Recognition quality (RQ)}}$$

DeeperLab: Single-Shot Image Parser

Tien-Ju Yang[1], Maxwell D. Collins[2], Yukun Zhu[2], Jyh-Jing Hwang[2,3], Ting Liu[2], Xiao Zhang[2], Vivienne Sze[1], George Papandreou[2], Liang-Chieh Chen[2]
MIT[1], Google Inc.[2], UC Berkeley [3]

“PQ is sensitive to false positives with small regions … suitable in applications ... with instances irrespective of their sizes.”

PQ Issue Demonstration



Several classes, e.g. pole (IoU 0.49) and traffic light (IoU 0.46), are just below the PQ acceptance threshold, while the sidewalk class (IoU 0.62) is just above it.

Proposed Variant of PQ

Keep using IoU>0.5 overlap criterion only for *thing* segments

Conventional, pixel-based IoU computation on *stuff* segments, as there is at most *one* segment for both, gt and prediction of stuff classes

$$\mathrm{PQ}_c^\dagger = \begin{cases} \frac{1}{|\mathcal{S}_c|} \sum_{(s,\hat{s})\in\mathcal{M}_c} \mathrm{IoU}(s,\hat{s})\,, & \text{if } c \text{ is stuff class} \\ \mathrm{PQ}_c\,, & \text{otherwise.} \end{cases}$$

$$\mathrm{PQ}_c = \frac{\sum_{(s,\hat{s})\in\mathrm{TP}_c} \mathrm{IoU}(s,\hat{s})}{|\mathrm{TP}_c| + \frac{1}{2}|\mathrm{FP}_c| + \frac{1}{2}|\mathrm{FN}_c|}$$

$$\mathrm{PQ}^\dagger = \frac{1}{N_{\text{classes}}} \sum_{c\in\mathcal{Y}} \mathrm{PQ}_c^\dagger$$

where

$$\mathrm{TP}_c = \{(s,\hat{s}) \in \mathcal{S}_c \times \hat{\mathcal{S}}_c \,:\, \mathrm{IoU}(s,\hat{s}) > 0.5\}$$

Summary & Conclusions

- Generating map data at scale requires thoroughly understood and designed machine learning solutions
- Mapillary's object recognition comprises of state-of-the-art
  - Semantic & panoptic segmentation
  - 3D object recognition
- It requires efficient & accurate 3D modeling (not part of today's talk)

[We have not touched potential issues of available metrics]

We have not touched issues arising from images captured in the wild

We have not touched the lack of benchmarks at scale

We are hiring!

Send me an email to **research@mapillary.com**

Mapillary Research