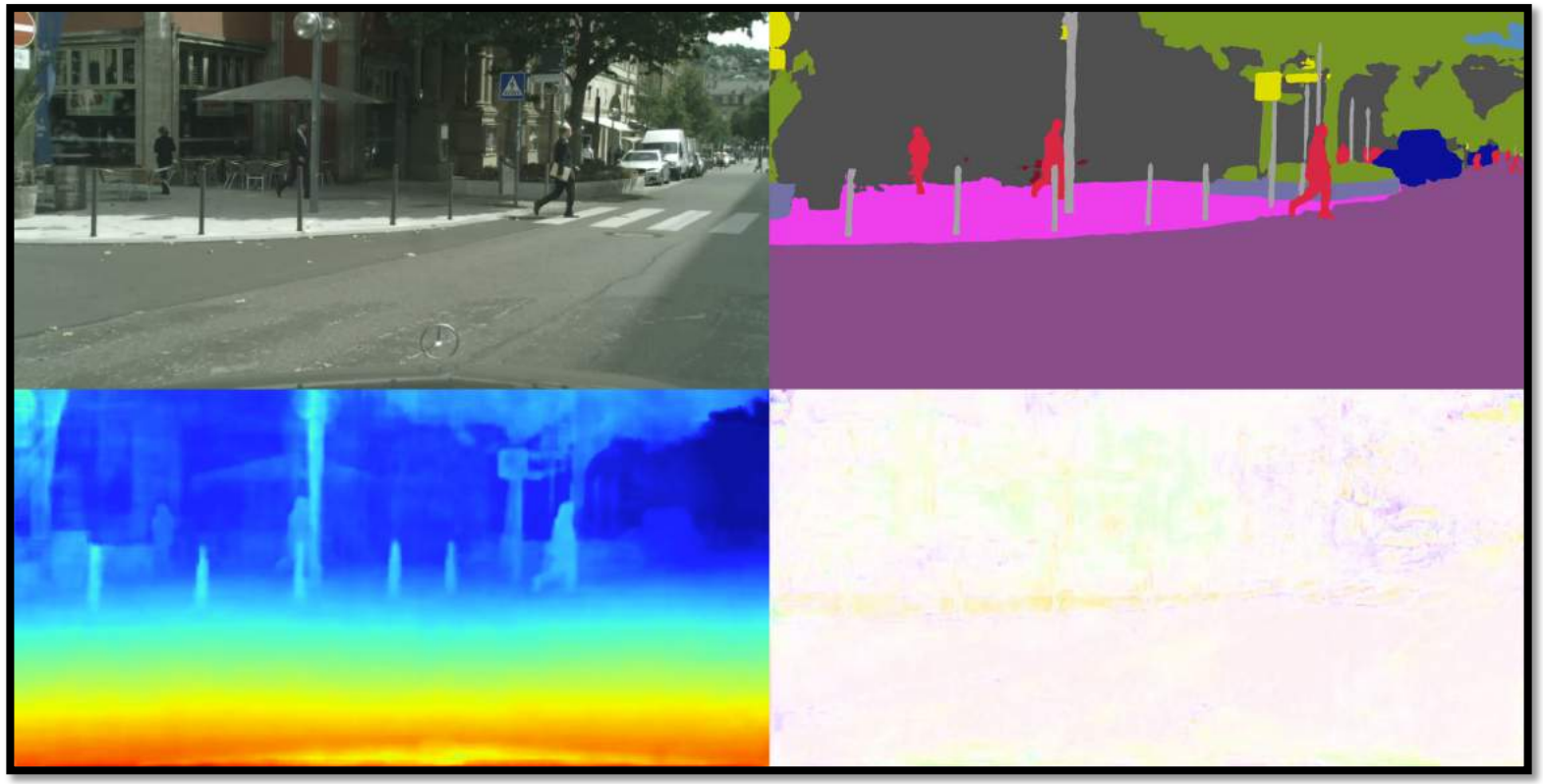
Scene Understanding with Deep Learning

Alex Kendall @ ICVSS, Sicily, July 2019

WAYVE

UNIVERSITY OF CAMBRIDGE



Progression of scene understanding from

2015

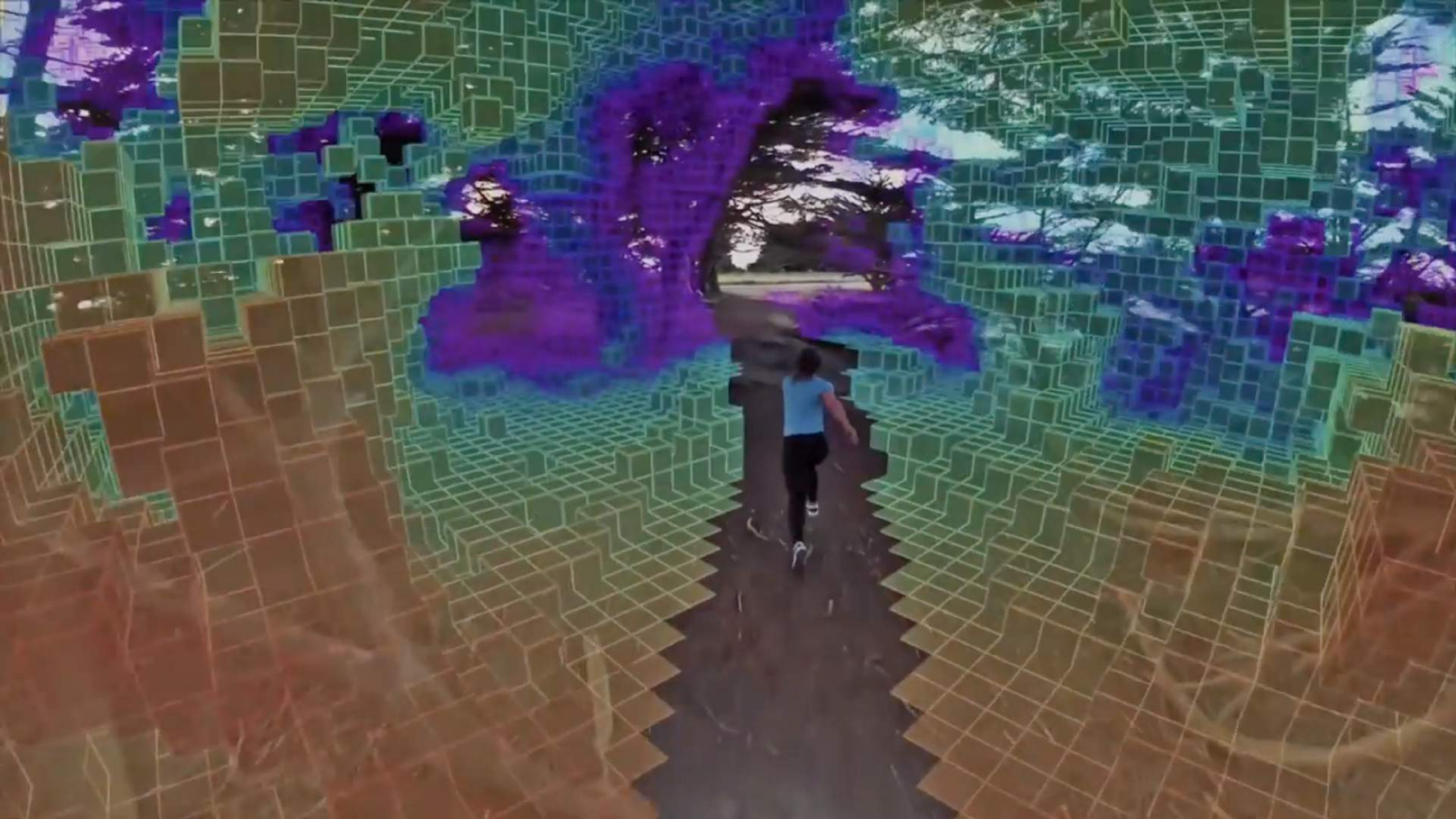
… to 2018

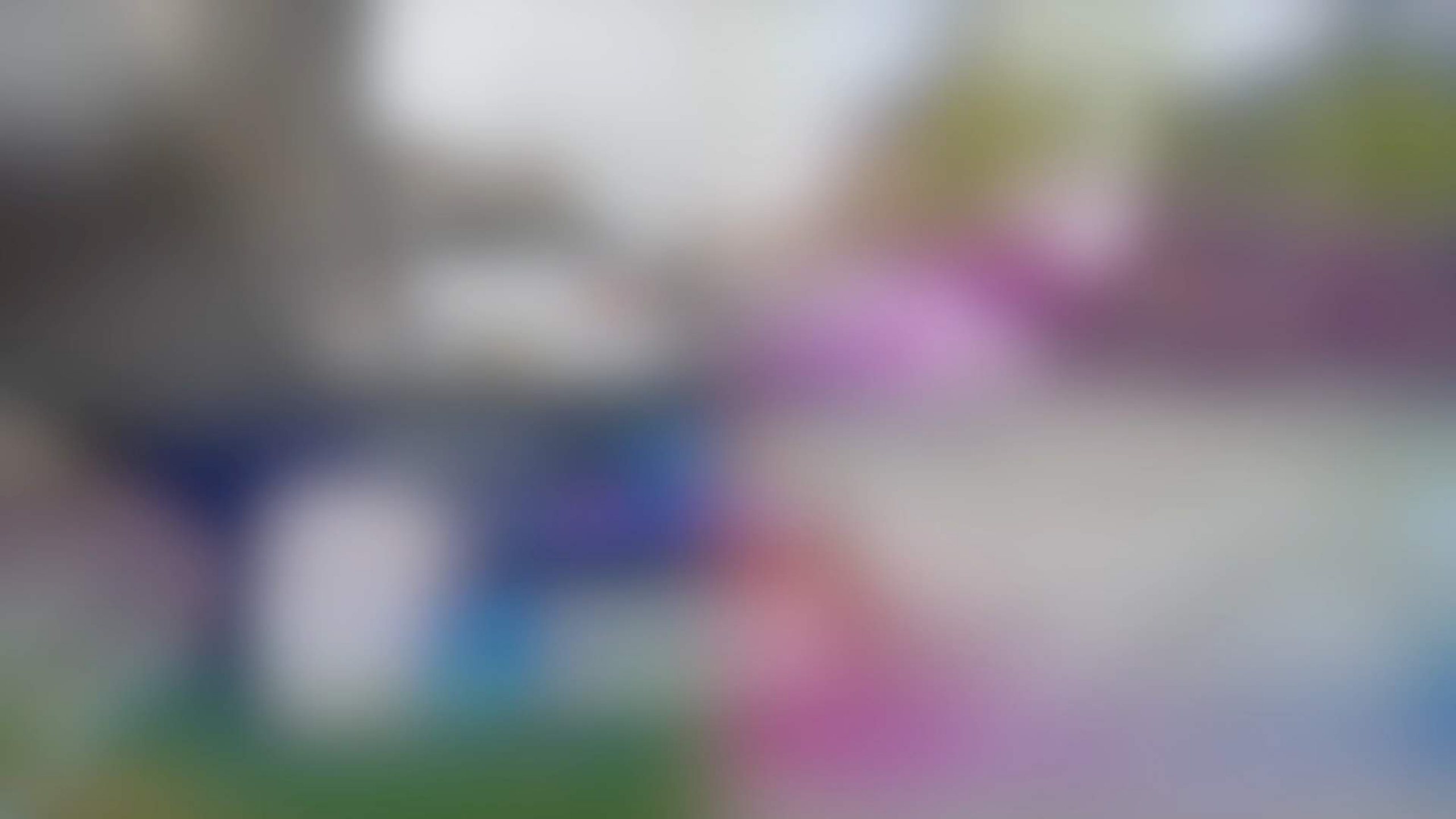
Badrinarayanan, Kendall, Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. PAMI, 2015.
Kendall, Gal and Cipolla. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. CVPR, 2018.

Scene understanding is beginning to work out in the wild..

Skydio

WAYVE






20
WX16 FDY

Outline of lecture

1. Motivate and define scene understanding
2. Learning representations of semantics, motion and geometry
3. Application to mobile robotics and autonomous driving

Part 1: What is ‘scene understanding’?

A possible computer vision definition?

- "Holistic scene understanding … reasons jointly about regions, location, class and spatial extent of objects, presence of a class in the image, as well as the scene type."

  Yao, Fidler and Urtasun "Describing the Scene as a Whole: Joint Object Detection, Scene Classification and Semantic Segmentation", CVPR, 2012.

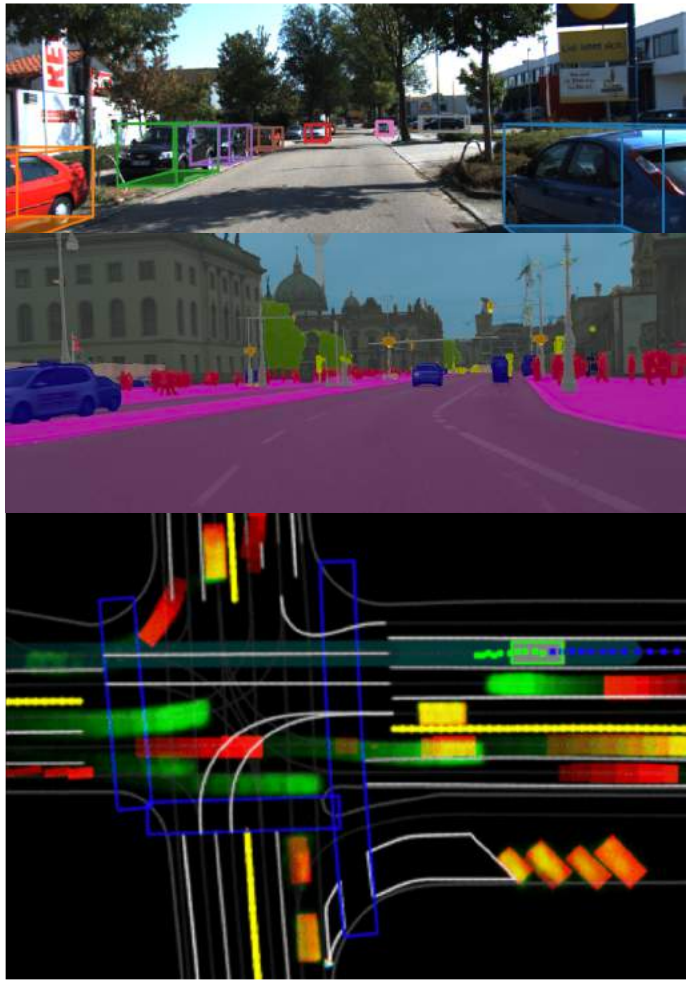
- "Scene understanding, in contrast to object recognition, attempts to analyze objects in context with respect to the 3D structure of the scene, its layout, and the spatial, functional, and semantic relationships between objects."

  Max Planck Institute for Intelligent Systems, Perceiving Systems homepage, retrieved August 2018.

- "Scene understanding is to analyze a scene by considering the geometric and semantic context of its contents and the intrinsic relationships between them."

  Indoor Scene Understanding in 2.5/3D: A Survey. Naseer et al. 2018.

Scene Understanding for Autonomous Driving



3D Object Detection

Semantic Segmentation

Agent Prediction

Turning indicator detector

HD Map

Driving Affordability Prediction

Traffic sign detection

.....

Autonomous Driving State Representation

Is this the best approach?

- Many KITTI metrics are at 90%+
- Are these metrics a good proxy for improving autonomous driving performance?
- Can we enumerate a priori the information that is required for a task?
- Are these tasks the best intermediate representation?
- Can we consider scene understanding independently of control?
- Do we care about in-domain test data?

Stereo:	98.26%
Flow:	95.27%
Odometry:	99.45%
2D Object:	91.97%
3D Object	77.86%
Tracking	90.77%
Segmentation:	72.82%

Geiger, Andreas, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the KITTI vision benchmark suite." CVPR, 2012.

A possible neuroscientist's definition

- "Studies in scene perception have shown that observers recognize a real-world scene at a single glance. During this expeditious process of seeing, the visual system forms a spatial representation of the outside world that is rich enough to grasp the meaning of the scene, recognizing a few objects and other salient information in the image, to facilitate object detection and the deployment of attention."

  Oliva. "Gist of the scene." Neurobiology of attention. 2005.

- "Scene understanding exists on a continuum. At one end is a very fast and seemingly effortless extraction of the scene's gist—often just its category name. At the other end is the slower and often effortful attachment of deeper meaning to the scene. I will adopt the lay person's definition of scene understanding—what is the scene about? What is the story that it is trying to tell?" Understanding scene understanding."

  Zelinsky. "Understanding scene understanding". Frontiers in psychology, 2013.

- "There is little evidence suggesting any bias toward either scene-level or object-level recognition."

  Fei-Fei, Iyer, Koch, Perona. "What do we perceive in a glance of a real-world scene?". Journal of Vision, 2007.

A possible neuroscientist's definition

- "It is possible to know every object and action in a scene and still not know what the scene is about—knowledge of these elements is, quite literally, not the whole story. Minimally, true understanding requires a more extensive filtering and ordering of this list to capture only those objects, actions, and events that are important to a viewer's interpretation."

Zelinsky. "Understanding scene understanding". Frontiers in psychology, 2013.

“We have a brain for one reason and one reason only -- that’s to produce adaptable and complex movements. Movement is the only way we have affecting the world around us… I believe that to understand movement is to understand the whole brain. And therefore it’s important to remember when you are studying memory, cognition, sensory processing, they’re there for a reason, and that reason is action.”

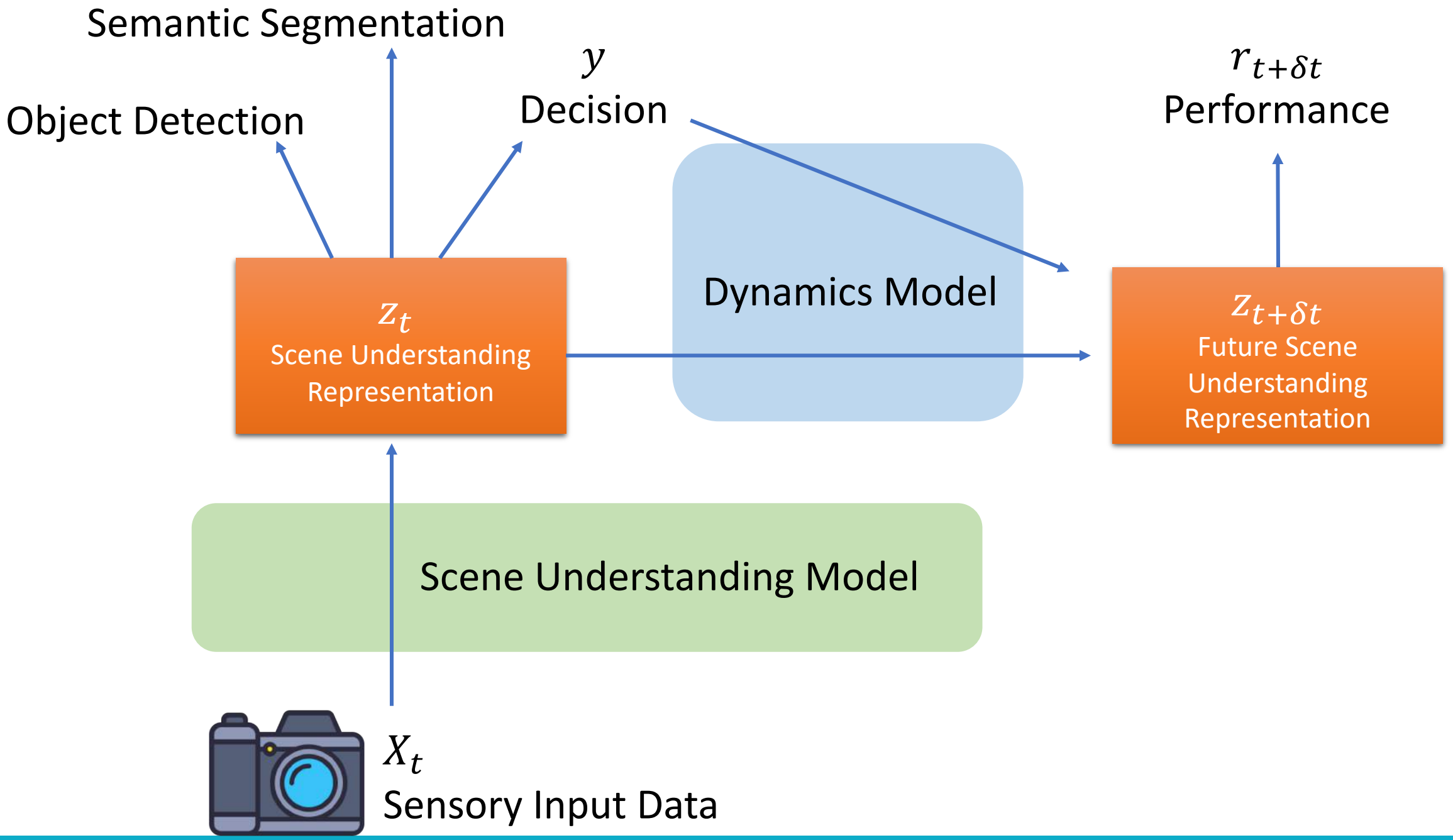
Prof Daniel Wolpert, TED 2011

How I think about scene understanding:

Scene understanding is to extract a minimal representation of the world which can be used to evaluate action.

Scene Understanding Topology

- Input sensory data, x
- Scene understanding model, $f(\cdot)$, to learn representation, $\mathbf{z} = f(x)$
- Policy model, $\pi(\cdot)$, to learn output(s) such as decision, action, auxiliary representations, $y = \pi(z)$
- Dynamics/transition/prediction model, $t(\cdot)$, to predict future states, $z_{t+1} = t(y, z)$



A representation learning perspective

1. Form a world model, to model the global dynamics and explain the world:

- Ha and Schmidhuber. Recurrent world models facilitate policy evolution. NeurIPS, 2018.
- Srinivas, Jabri, Abbeel, Levine, and Finn. Universal planning networks. ICML, 2018.
- Burda, et al. Large-scale study of curiosity-driven learning. arXiv 2018.

2. Or, we can learn a task-specific representation:

- Ghosh et al. Learning actionable representations with goal-conditioned policies. ICLR 2019
- Dwibedi et al. Learning actionable representations from visual observations. IROS, 2018.

A recipe for a good representation

- Contain the hand-specified information which is believed to be necessary (but not sufficient) for the task. For driving, this includes semantics, motion and geometry,

- Be optimised with respect to the end task to learn the information sufficient for the task,

- Contain an excellent signal-to-noise ratio to observe the data required to make the decision. Therefore we need the right sensor configuration and transform the signal into a compressed, nuisance free & invariant representation.

- Eliminate any spurious correlations in the data. Disentangle the data and ensure the correct causal information is used.

Part 2: Scene Understanding

- Learning semantics, motion and geometry

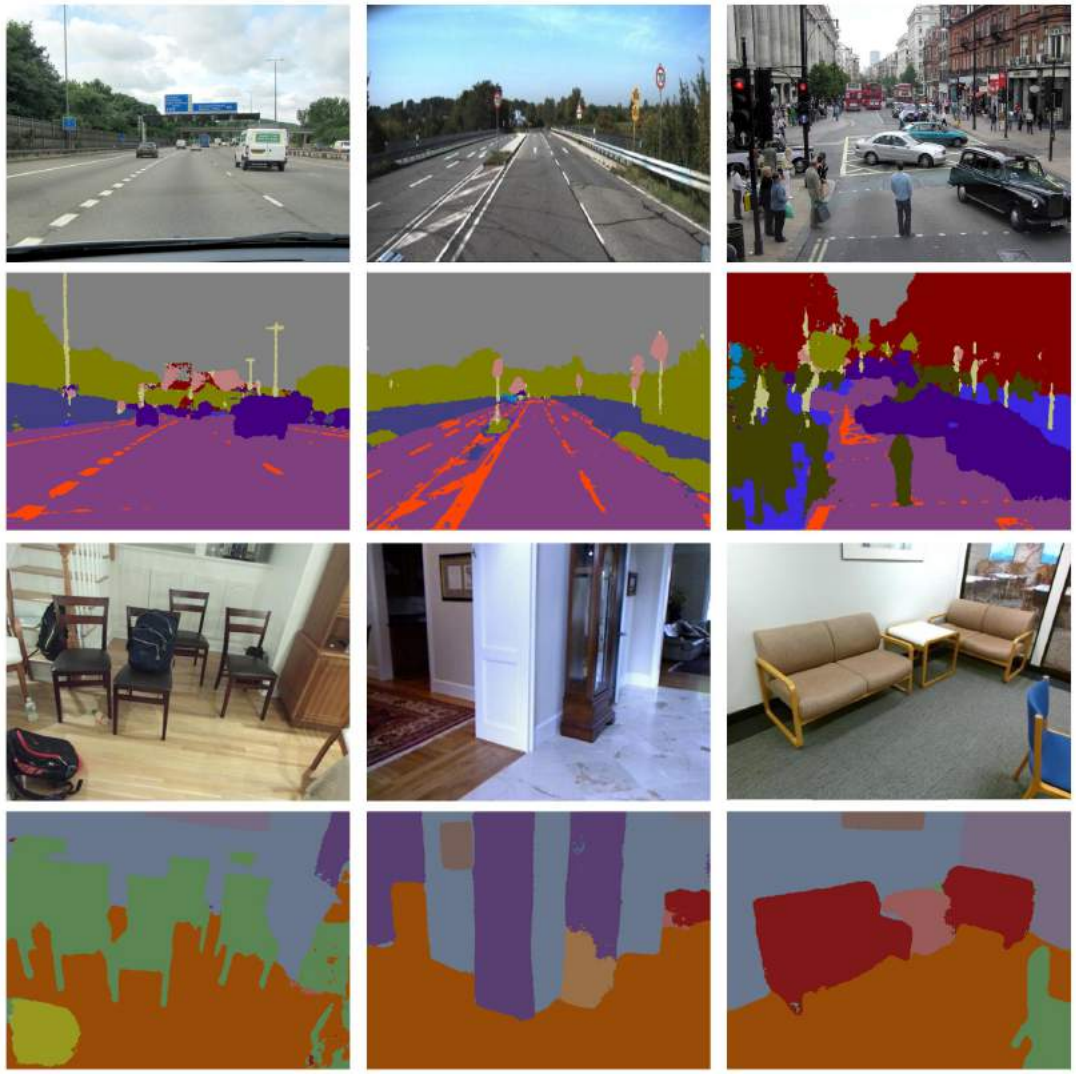
Learning Semantics

With Semantic Segmentation

Understanding Semantics

Many representations

- Image classification
- 2D & 3D detection
- Semantic segmentation
- Instance segmentation
- Panoptic segmentation
- Semantic embedding



Semantic Segmentation Datasets

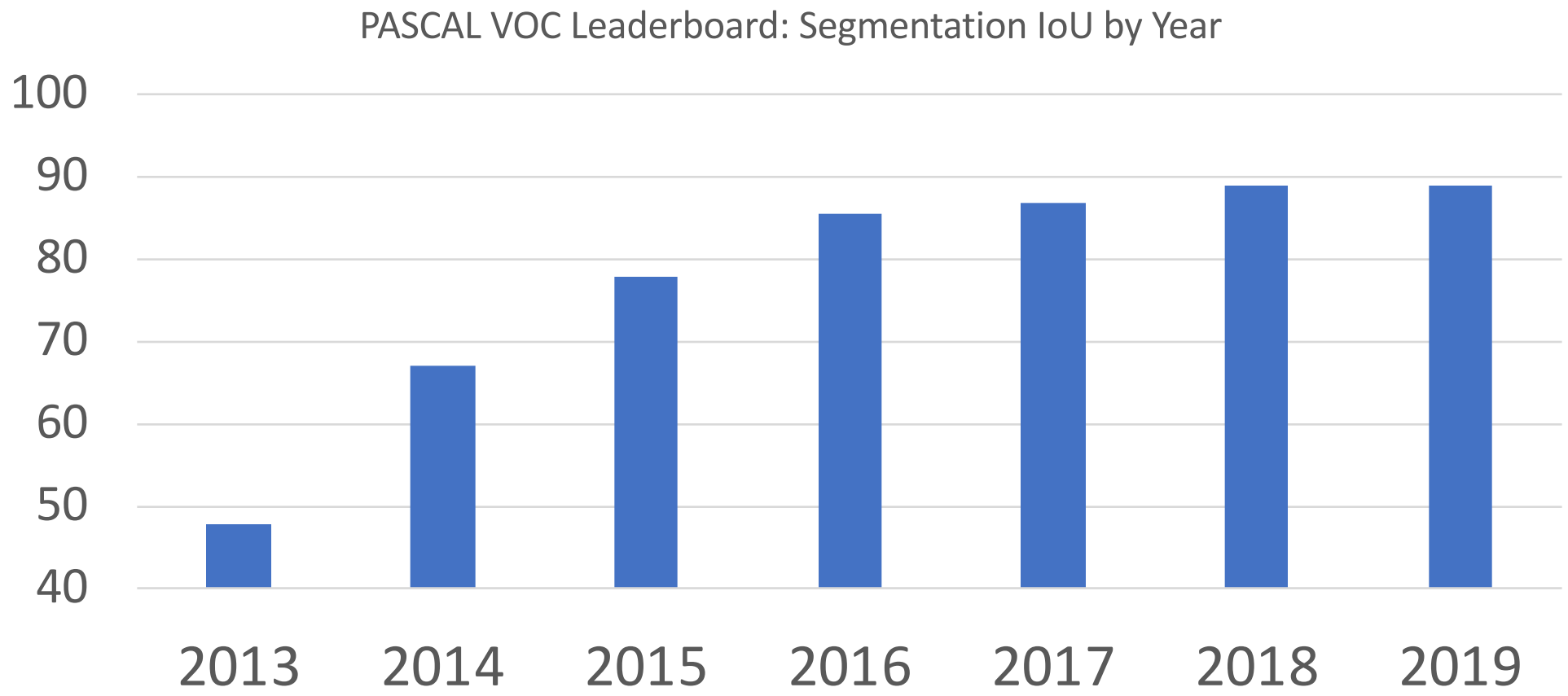
Difficulty tends to scale with diversity and number of classes.

- CamVid (small scale driving), 300 images, 12 classes
- CityScapes (clean driving scenes), 5k images, 20 classes, 84% mIoU
- Mapillary Vistas (diverse driving scenes), 25k images, 66 classes, 52% mIoU

- SUN RGB-D (indoor scenes), 5k images
- NYUv2 (indoor Kinect data), 2k images

- Pascal VOC (object segmentation), 10k images, 20 classes, 85% mIoU
- MSCOCO (object segmentation), 200k images, 150 classes, 56% mIoU

Summary of core ideas in semantic segmentation from 2015-2019



Source: http://host.robots.ox.ac.uk/pascal/VOC/

‘Fully’ Convolutional Neural Networks

- Prior deep learning approaches to semantic segmentation used CNNs to classify each image patch
- FCN proposed to interpret densely connected layers as 1x1 convolutions
- Fine-tuned ImageNet classification models for pixel-wise semantic segmentation

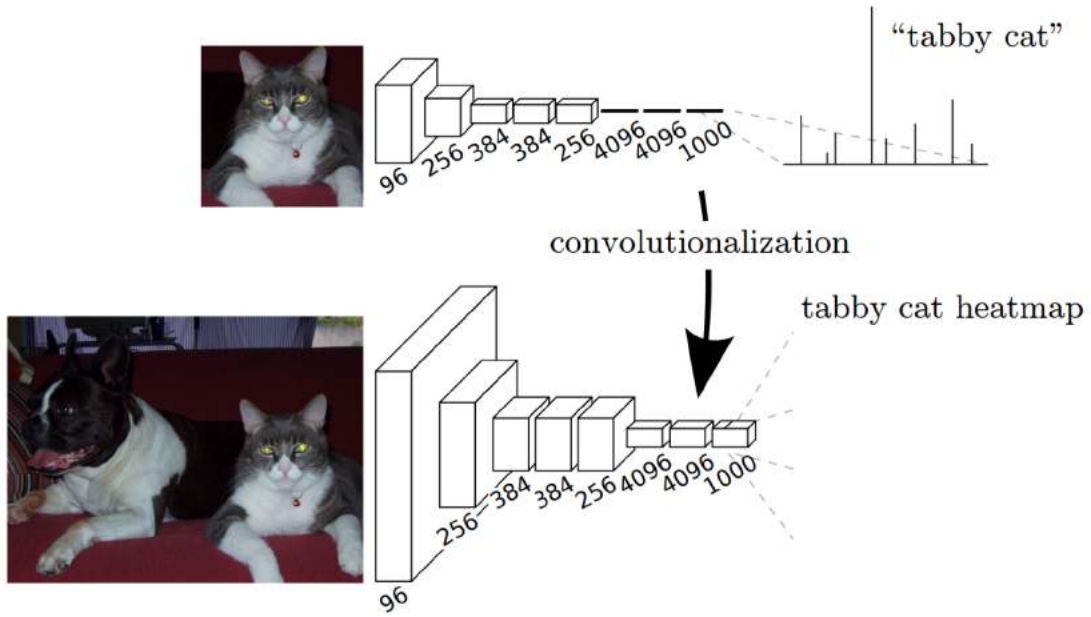
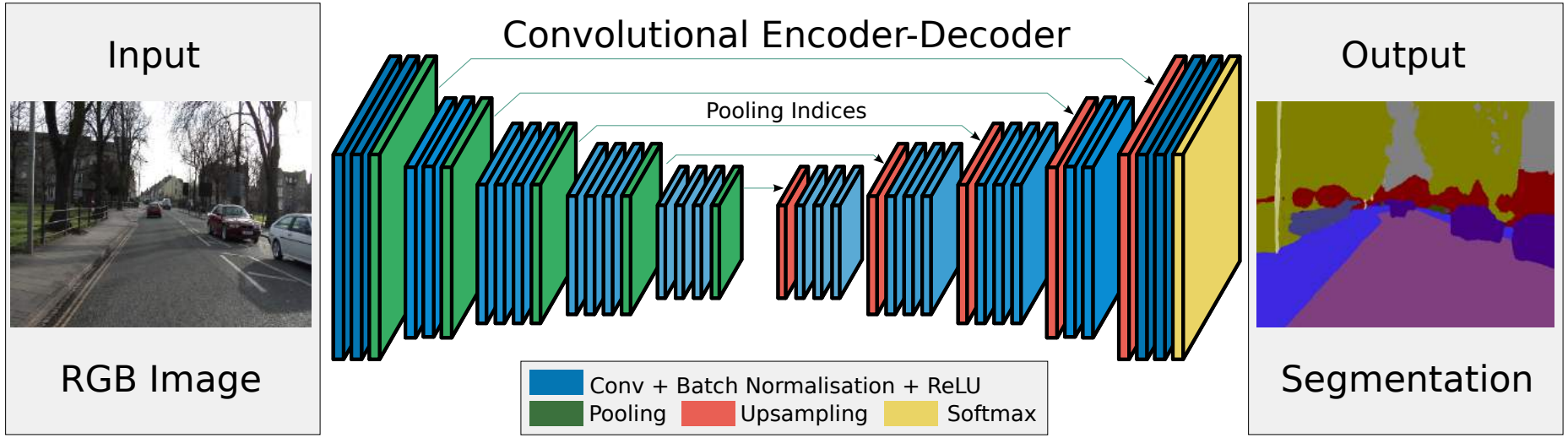


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." *CVPR*. 2015.

Encoder-Decoder Architectures

Encoder reduces spatial dimensions,
enhancing feature complexity

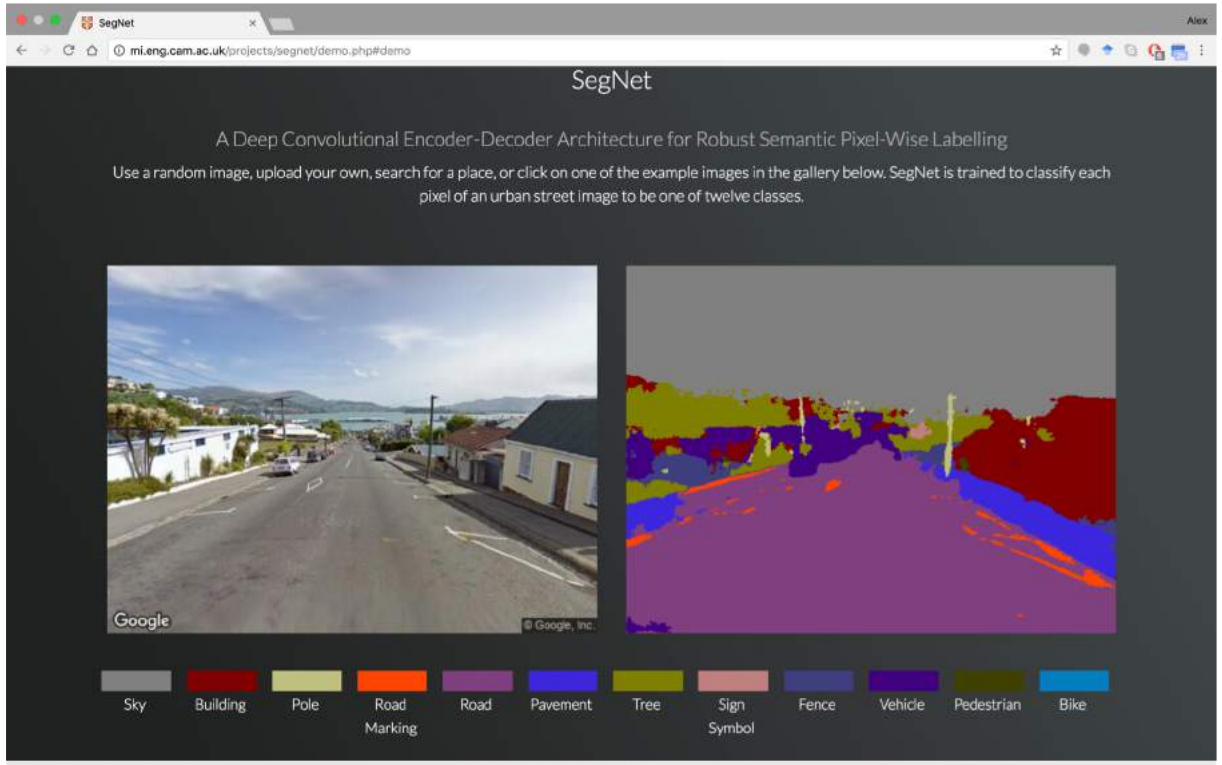


Decoder recovers spatial dimensions

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A deep convolutional encoder-decoder architecture for image segmentation." *PAMI 2015*

Encoder-Decoder Architectures

- SegNet first trained a custom semantic segmentation architecture end-to-end
- ‘Invert’ classification networks like VGG to construct encoder-decoder
- Encoder downsamples spatial dimensions with max-pooling, building more depth in feature dimensions
- Decoder upsamples spatially with unpooling
- Efficient real-time performance and webdemo



2015 Webdemo: http://mi.eng.cam.ac.uk/projects/segnet/

Skip-Connections

- Big improvement by introducing residual connections and skip connections
- Train deeper models and ensure information is aggregated from all stages in hierarchy, and from all stages of spatial subsampling

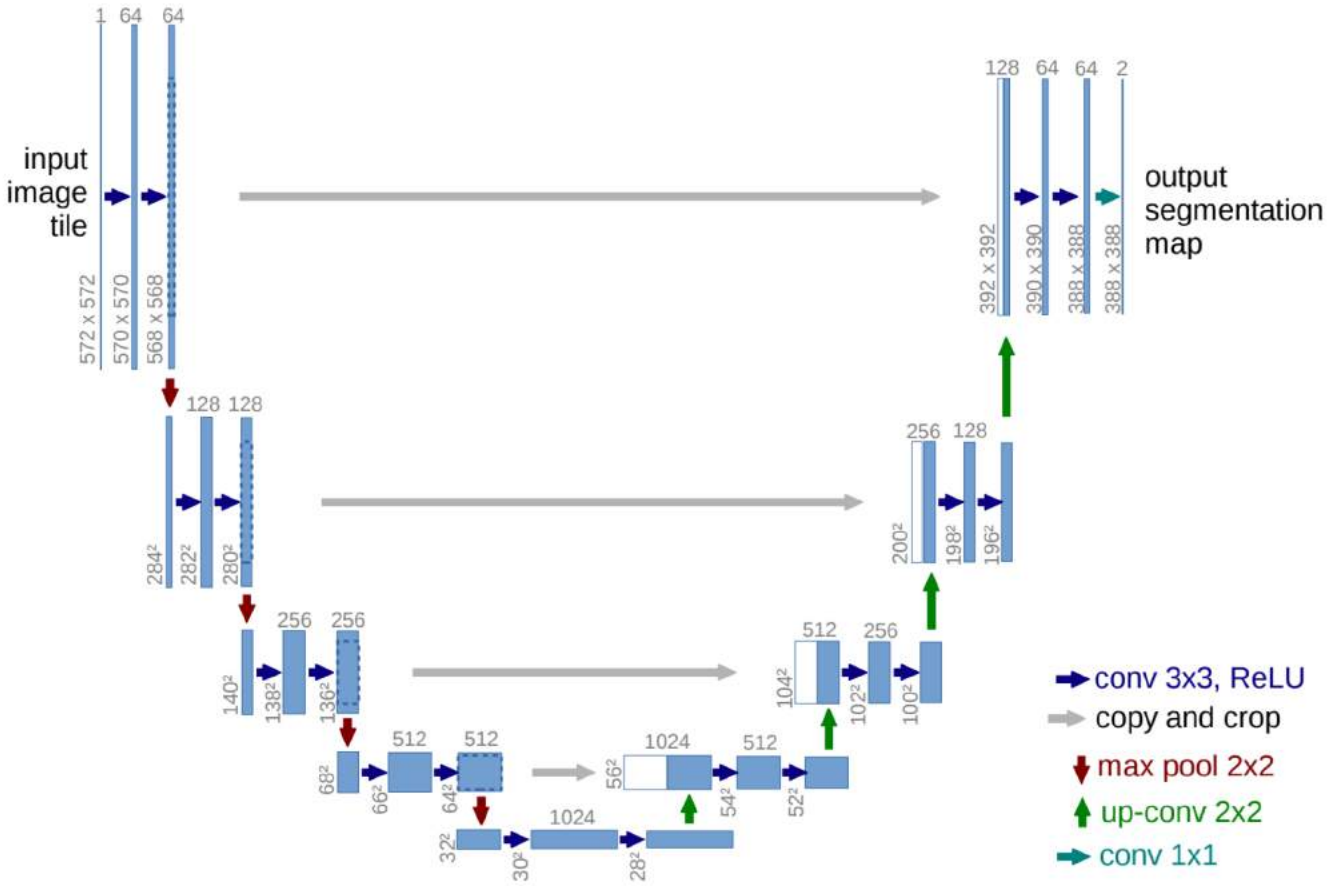
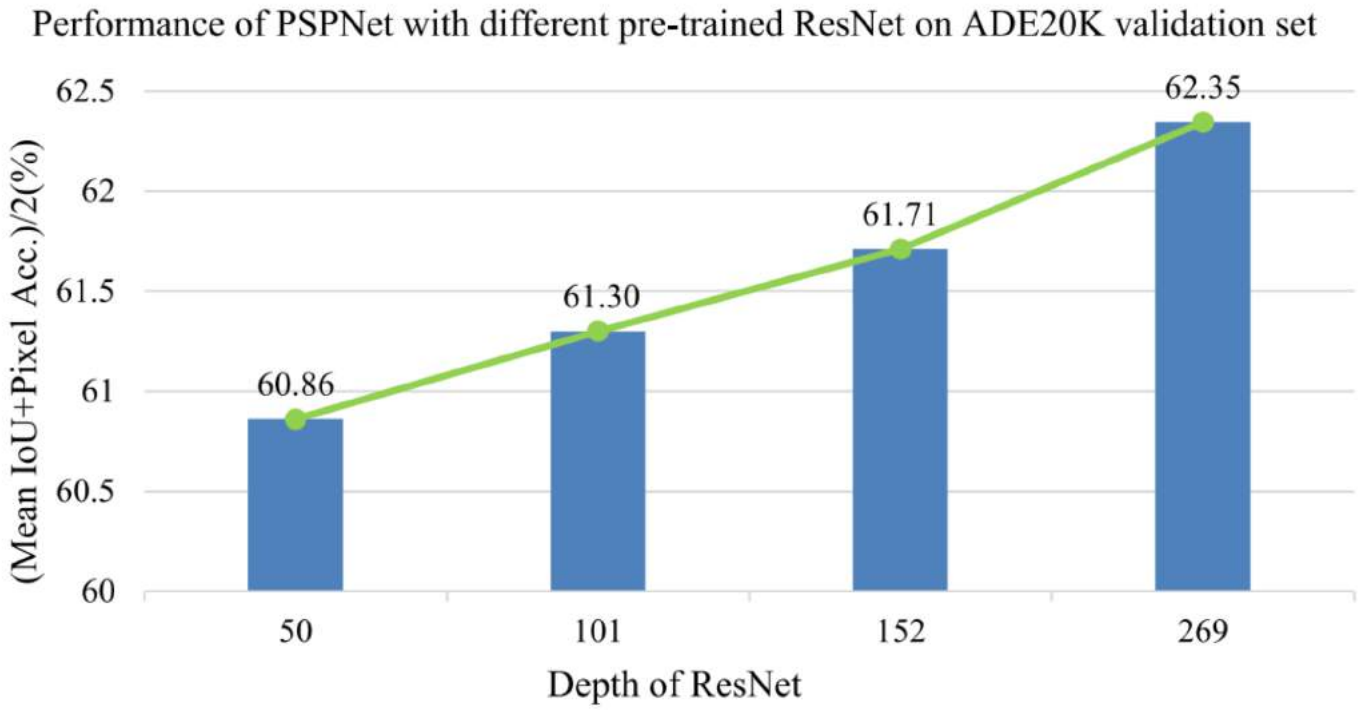


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *MICCAI*, 2015.

Improving the recognition front-end

- Improves semantic segmentation performance from AlexNet -> VGG -> Resnet



He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
Zhao, Hengshuang, et al. "Pyramid scene parsing network." *CVPR*. 2017.

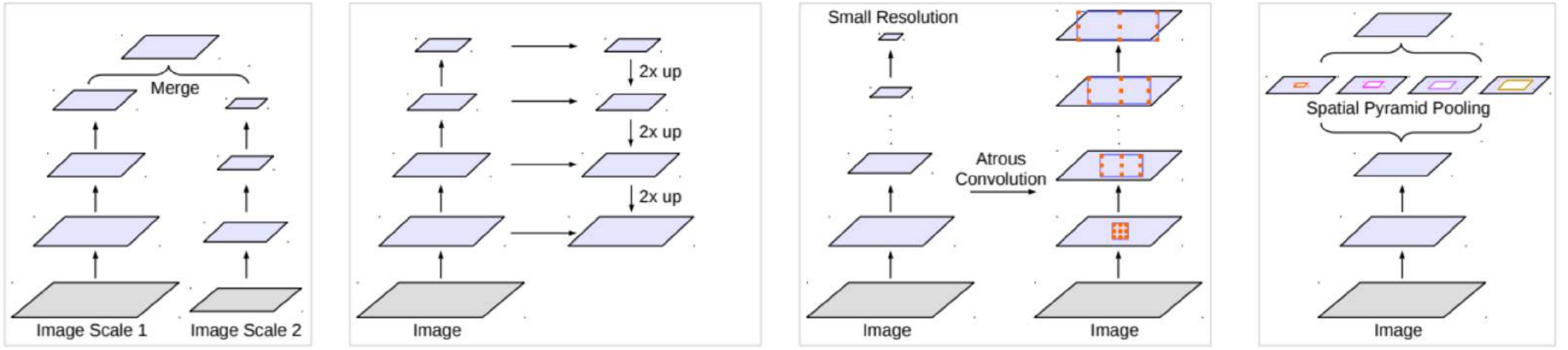
Increasing Context

- With larger context, improve segmentation performance
- Better consistency across homogenous regions like sky, road
- Include more semantic meaning across scene and disambiguate challenging appearance
- Examples include:
  - Dilated convolutions
  - Pyramid pooling module
  - Atrous Spatial Pyramid Pooling

Yu, Fisher, and Vladlen Koltun. "Multi-scale context aggregation by dilated convolutions." ICLR (2016).
Zhao, Hengshuang, et al. "Pyramid scene parsing network." *CVPR*. 2017.
Chen, Liang-Chieh, et al. "Rethinking atrous convolution for semantic image segmentation." *arXiv preprint arXiv:1706.05587* (2017).

Increasing Context

- Approaches to aggregating context



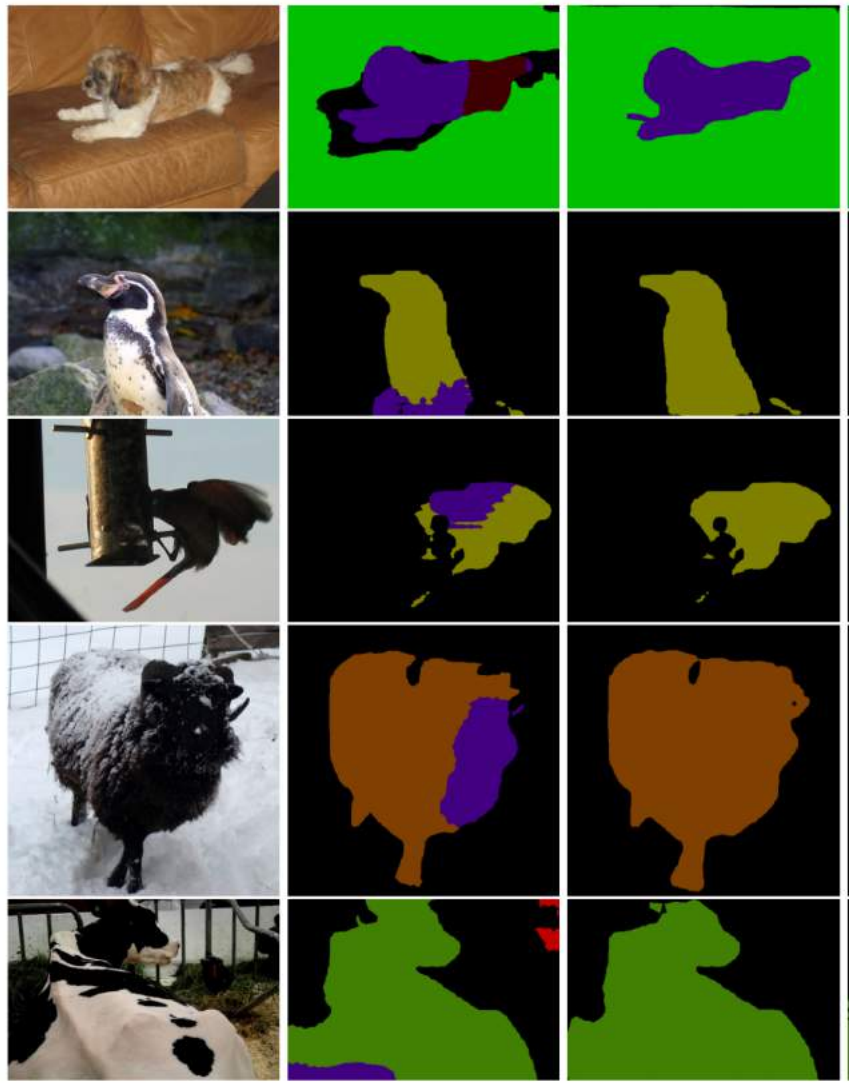
(a) Image Pyramid (b) Encoder-Decoder (c) Deeper w. Atrous Convolution (d) Spatial Pyramid Pooling

Figure 2. Alternative architectures to capture multi-scale context.

Chen, Liang-Chieh, et al. "Rethinking atrous convolution for semantic image segmentation." *arXiv preprint arXiv:1706.05587* (2017).

Increasing Context

- Qualitative results show improved class consistency



Yu, Fisher, and Vladlen Koltun. "Multi-scale context aggregation by dilated convolutions." ICLR (2016).

Qualitative results perform well on all scales

Chen, Liang-Chieh, et al. "Rethinking atrous convolution for semantic image segmentation." *arXiv preprint arXiv:1706.05587* (2017).

Increasing Context

- Atrous Spatial Pyramid Pooling module

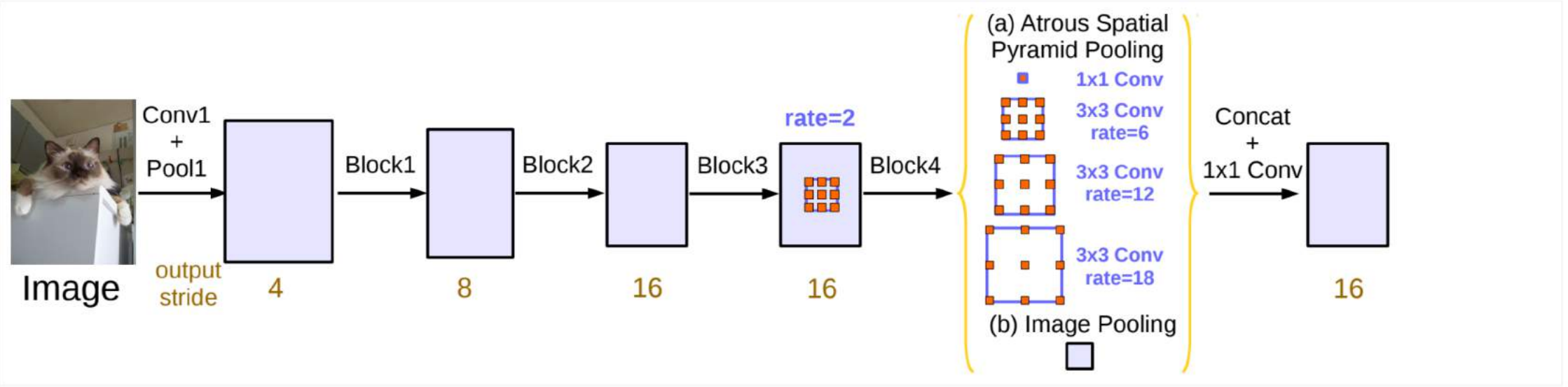


Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.

Chen, Liang-Chieh, et al. "Rethinking atrous convolution for semantic image segmentation." *arXiv preprint arXiv:1706.05587* (2017).

Increasing Context with PSPNet

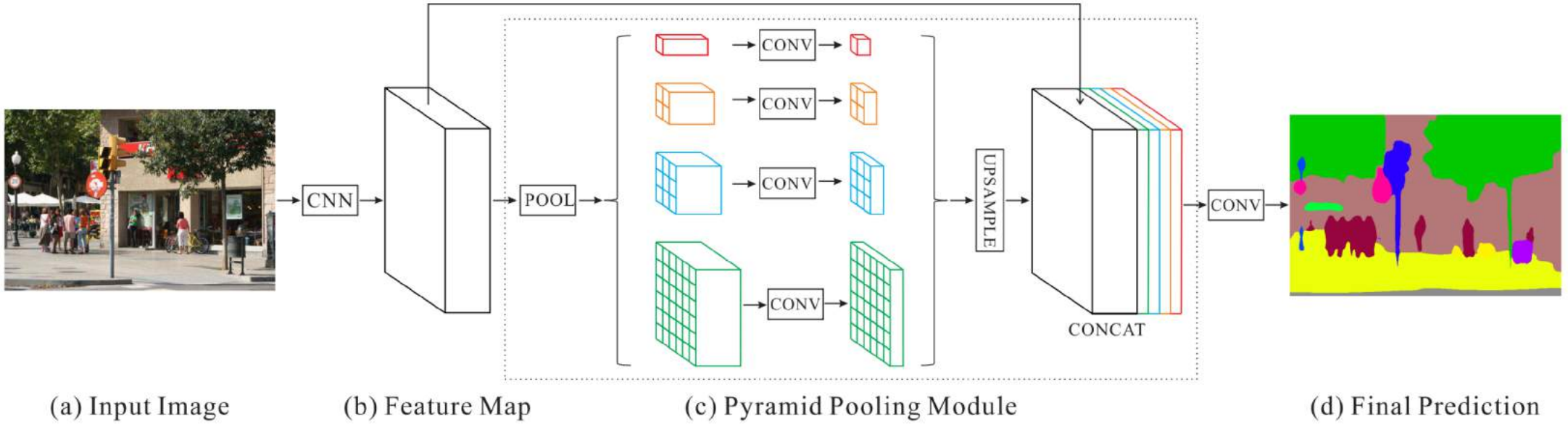
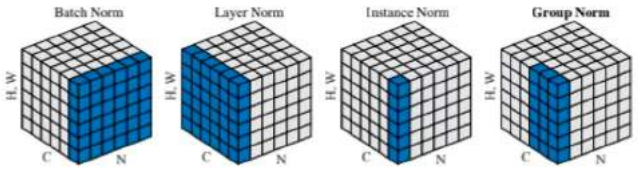


Figure 3. Overview of our proposed PSPNet. Given an input image (a), we first use CNN to get the feature map of the last convolutional layer (b), then a pyramid parsing module is applied to harvest different sub-region representations, followed by upsampling and concatenation layers to form the final feature representation, which carries both local and global context information in (c). Finally, the representation is fed into a convolution layer to get the final per-pixel prediction (d).

Zhao, Hengshuang, et al. "Pyramid scene parsing network." *CVPR*. 2017.

Hyper-parameters are important!

- Learning rate schedule commonly used:
  - $LR = LR_0 \left(1 - \frac{iter}{maxiter}\right)^{power}$, $LR_0 = 0.01, power = 0.9$
- Batch size and normalisation:
  - In-place activated batchnorm
  - Instance normalisation
  - Very big GPU cluster



batch size	mIOU
4	64.43
8	75.76
12	76.49
16	77.21

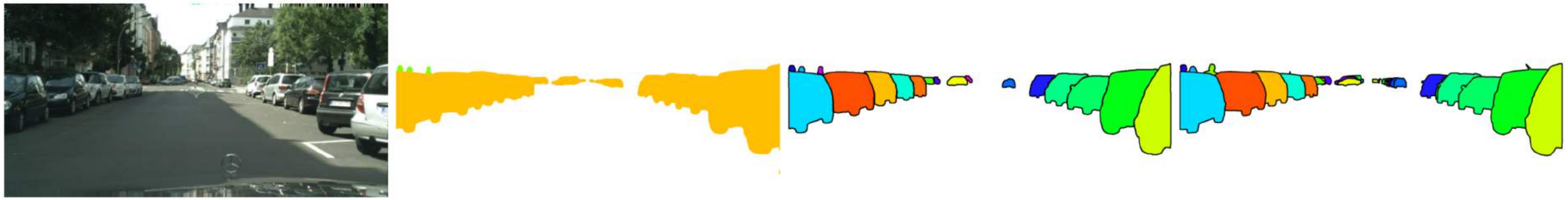
Table 9. Effect of batch size on PASCAL VOC 2012 *val* set. We employ *output_stride=16* during both training and evaluation. Large batch size is required while training the model with fine-tuning the batch normalization parameters.

Rota Bulò, Samuel, Lorenzo Porzi, and Peter Kontschieder. "In-place activated batchnorm for memory-optimized training of dnns." *CVPR*. 2018.

Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky. "Instance normalization: The missing ingredient for fast stylization." *arXiv preprint arXiv:1607.08022* (2016).

Instance Segmentation

- Embedding / clustering [kendall et al., De Brabandere et al.]
- Region proposals [mask-rcnn, He et al.]
- Edge detection [deep watershed, Bai et al.]



(a) Input Image (b) Semantic Segmentation [34] (c) Our Instance Segmentation (d) GT Instance Segmentation

Summary of segmentation & open problems

- Performs very well with sufficient labelled data
- Large context and receptive fields help
  (but why do practical << theoretical receptive fields?)
- Rare classes are still hard (zero / few shot learning)
- Open set / unknown set classification is interesting
- Can we learn embeddings that reduce reliance on supervision?
- Robustness and modelling uncertainty
- Understanding reasoning and causality

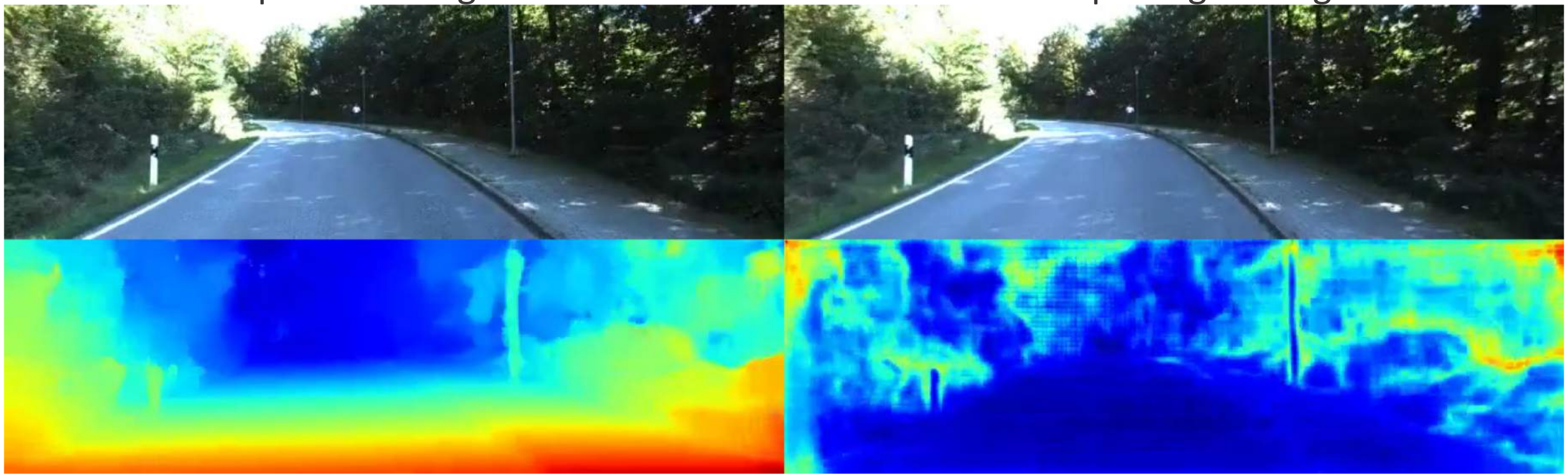
Learning Geometry

Depth and Shape Estimation

Deep Learning for Stereo Vision

Input Left Image

Input Right Image



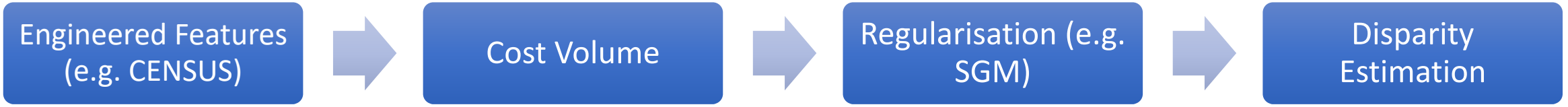
Depth Prediction

Depth Prediction Uncertainty

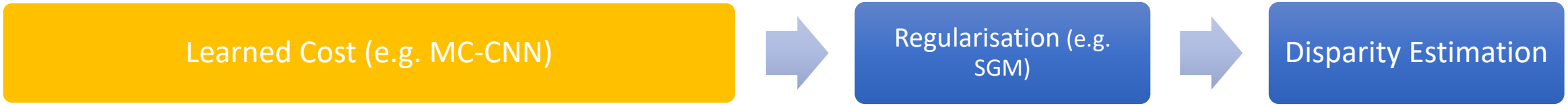
Alex Kendall et al. **End-to-End Learning of Geometry and Context for Deep Stereo Regression.** ICCV, 2017.

Alex Kendall and Roberto Cipolla. **Uncertainty and Unsupervised Learning for Stereo Vision with Probabilistic Deep Learning.** *Under Review*, 2017.

Brief History of Stereo Vision



H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. CVPR 2005



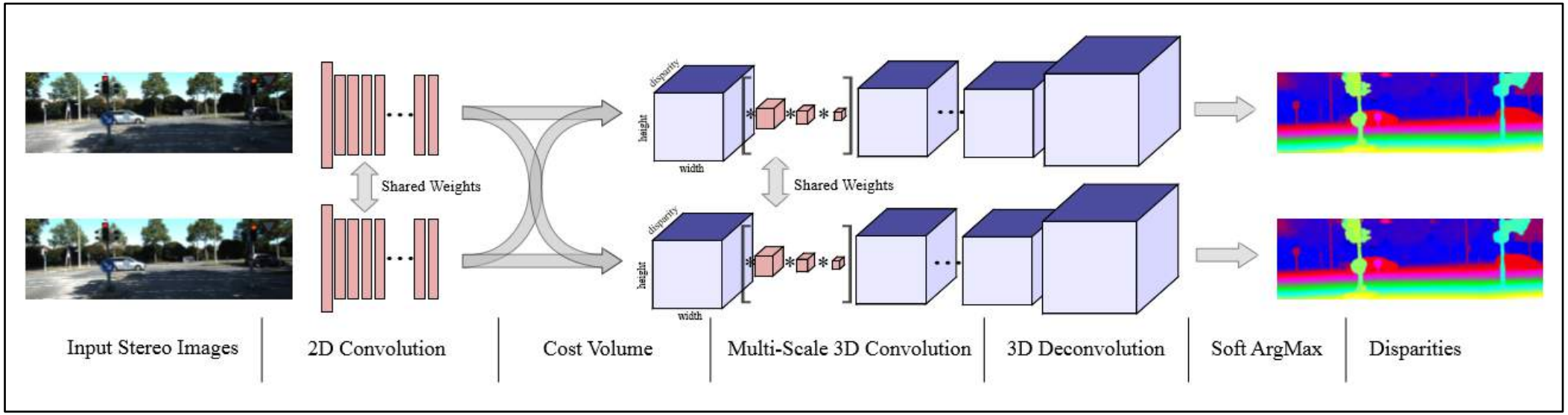
J. Zbontar and Y. LeCun. Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. JMLR 2016.

N. Mayer et al. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. CVPR 2016.

Alex Kendall et al. End-to-End Learning of Geometry and Context for Deep Stereo Regression. ICCV, 2017.

GC-Net: end to end deep learning for stereo

- Form differentiable cost volume using stereo geometry
- Sub-pixel disparity regression with soft ArgMax function
- Use 3-D convolutions to learn features with large context

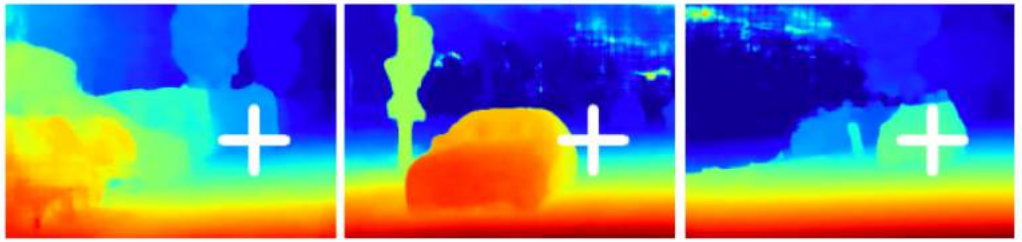


Alex Kendall et al. **End-to-End Learning of Geometry and Context for Deep Stereo Regression.** ICCV, 2017.

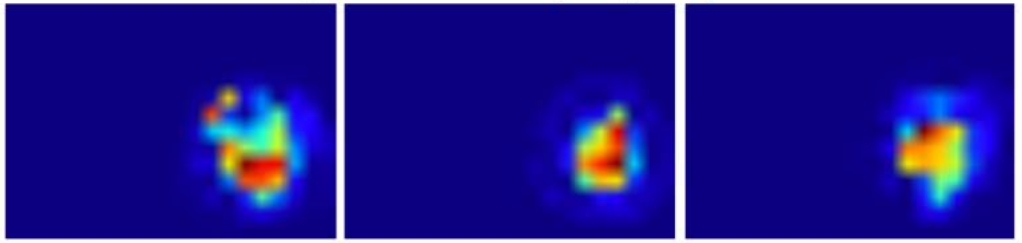
Context-aware

- Saliency shows which part of the input signal affects output prediction
- Demonstrates the model has a large receptive field to learn disparity with context

(a) Left stereo input image



(b) Predicted disparity map



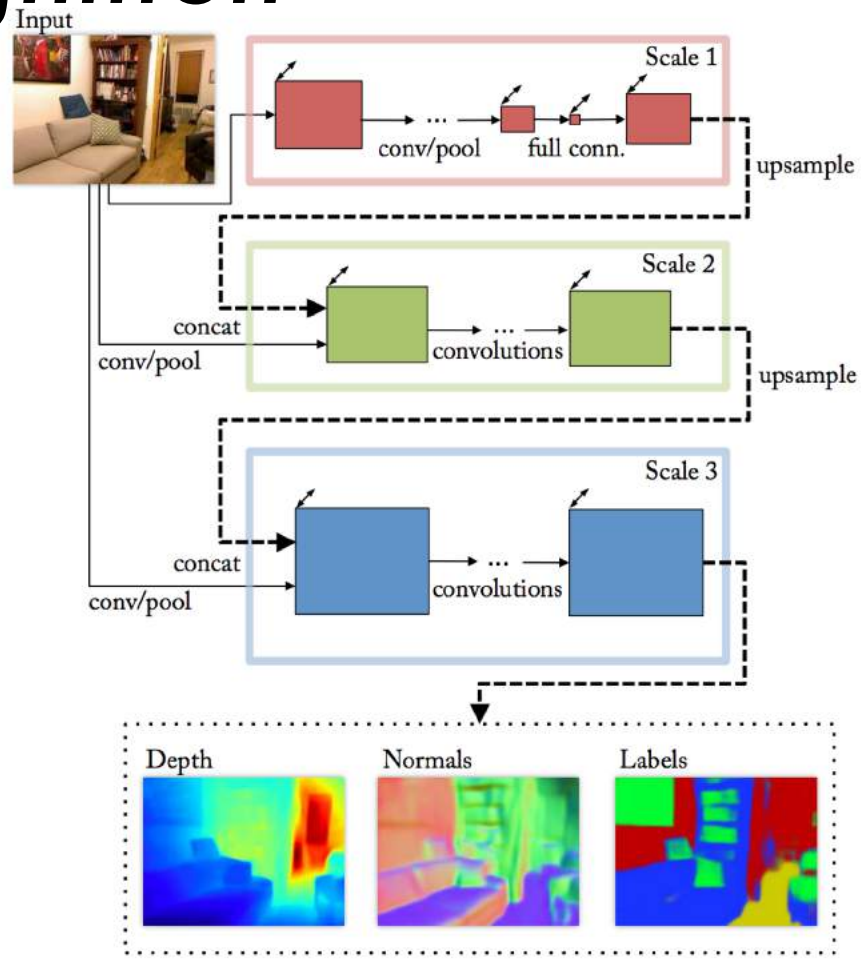
(c) Saliency map (red = stronger saliency)

(d) What the network sees (input attenuated by saliency)

Alex Kendall et al. **End-to-End Learning of Geometry and Context for Deep Stereo Regression.** ICCV, 2017.

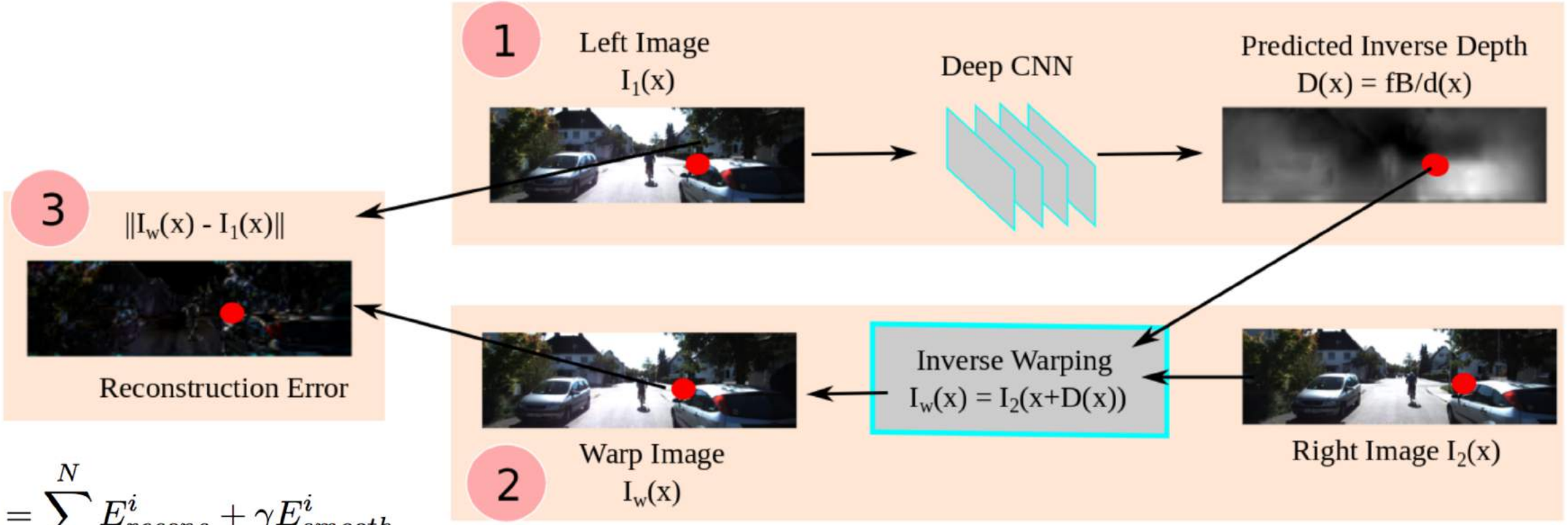
Monocular *Depth from Recognition*

- Eigen & Fergus regressed depth and surface normals from a CNN
- Showed that networks could learn regression tasks
- Estimating depth based on semantic and geometric cues
- Results on challenging indoor datasets



Eigen, and Fergus. "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture." *ICCV*. 2015.
Eigen, Puhrsch, and Fergus. "Depth map prediction from a single image using a multi-scale deep network." *NeurIPS*. 2014.

Self-Supervised Depth Estimation



$$E = \sum_{i=1}^{N} E^i_{recons} + \gamma E^i_{smooth}$$

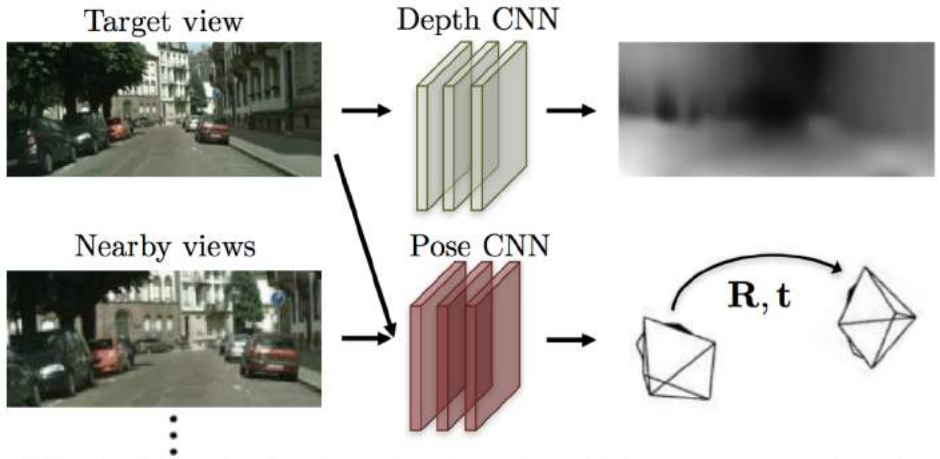
$$E^i_{smooth} = \|\nabla D^i(x)\|^2$$

$$E^i_{recons} = \int_\Omega \|I^i_w(x) - I^i_1(x)\|^2 dx = \int_\Omega \|I^i_2(x + \underbrace{D^i(x)}_{fB/d^i(x)}) - I^i_1(x)\|^2 dx$$

Garg, Ravi, et al. "Unsupervised CNN for single view depth estimation: Geometry to the rescue." ECCV, 2016.

Monocular Self-Supervised Depth Estimation

(a) Training: unlabeled video clips.



(b) Testing: single-view depth and multi-view pose estimation.

Zhou, Tinghui, et al. "Unsupervised learning of depth and ego-motion from video." *CVPR*. 2017.

Self-Supervised Learning of Geometry

We can learn depth and ego-motion using multi-view geometry with the camera intrinsics, K, predicted depth, y_{depth}, and egomotion, T.

Camera projection

$$X = \mathbf{y}_{depth,t}(p_{ijt})K^{-1}p_{ijt}.$$

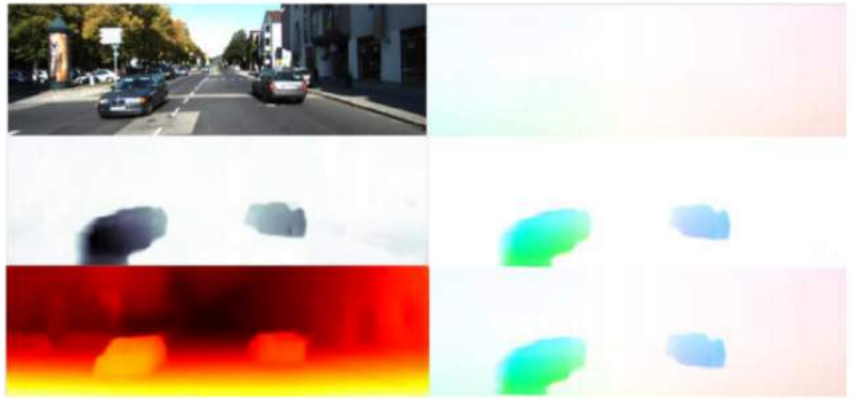
$$\hat{p}_{ij(t-1)} = K\hat{T}_{t\rightarrow(t-1)}X.$$

Photometric reconstruction loss

$$\mathcal{L}_{mono\ depth,\ t} = \frac{1}{N}\sum_{i,j}|I_t(p_{ijt}) - I_{(t-1)}(\hat{p}_{ij(t-1)})|,$$

Zhou, Tinghui, et al. "Unsupervised learning of depth and ego-motion from video." *CVPR 2017*.

Remaining Challenges



- Occlusion, aperture problem, ambiguity, dynamic objects…

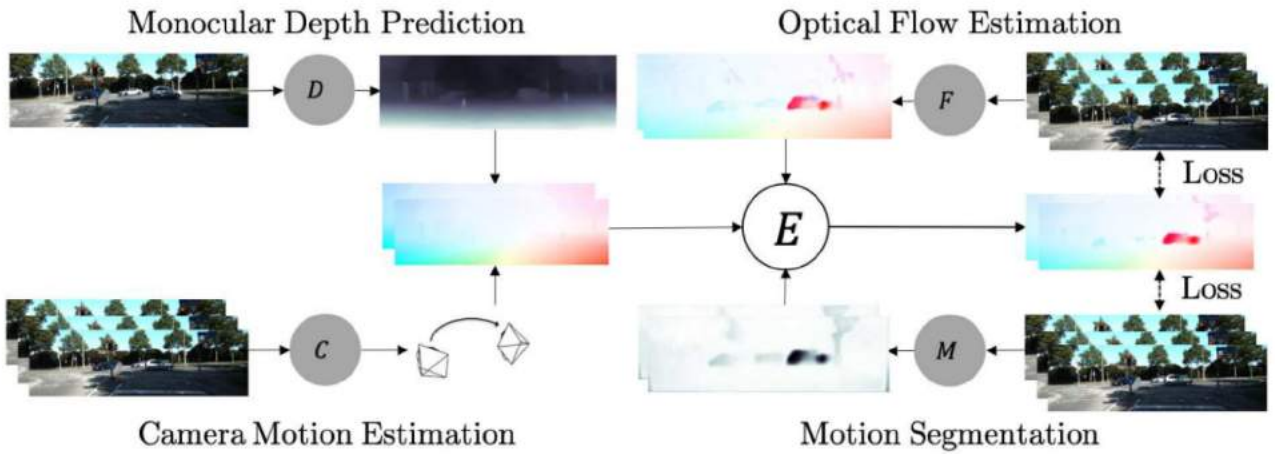
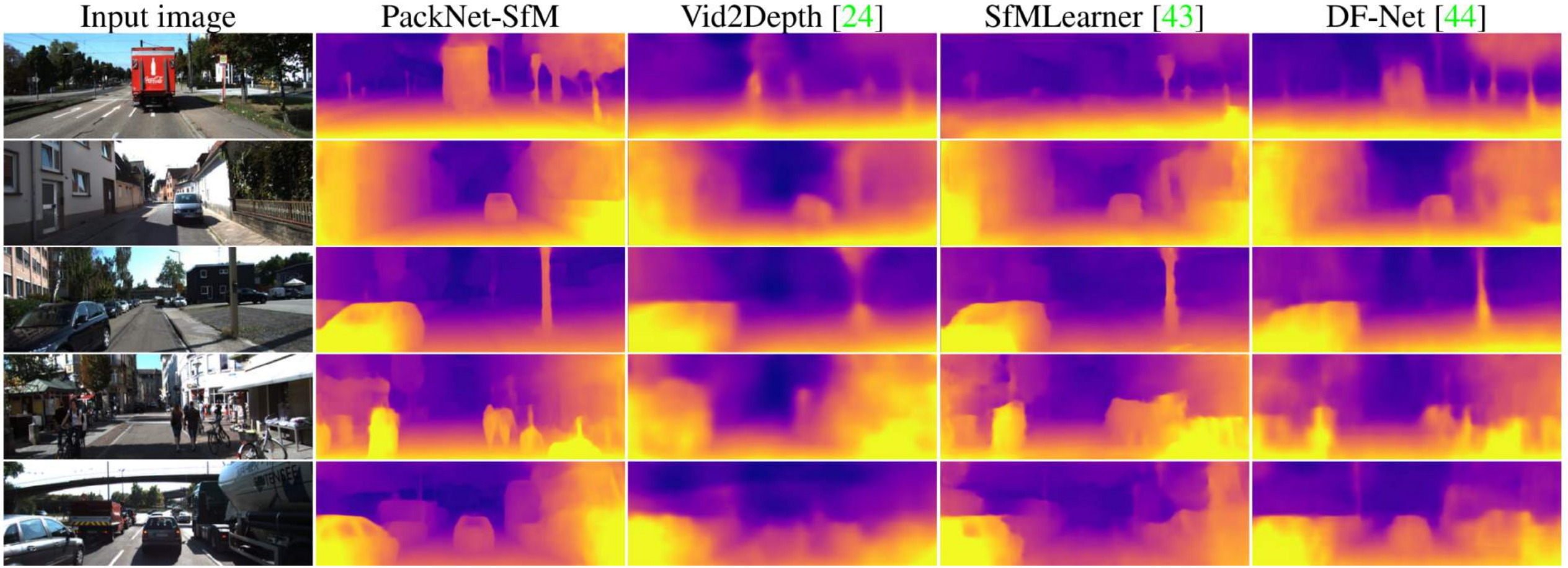


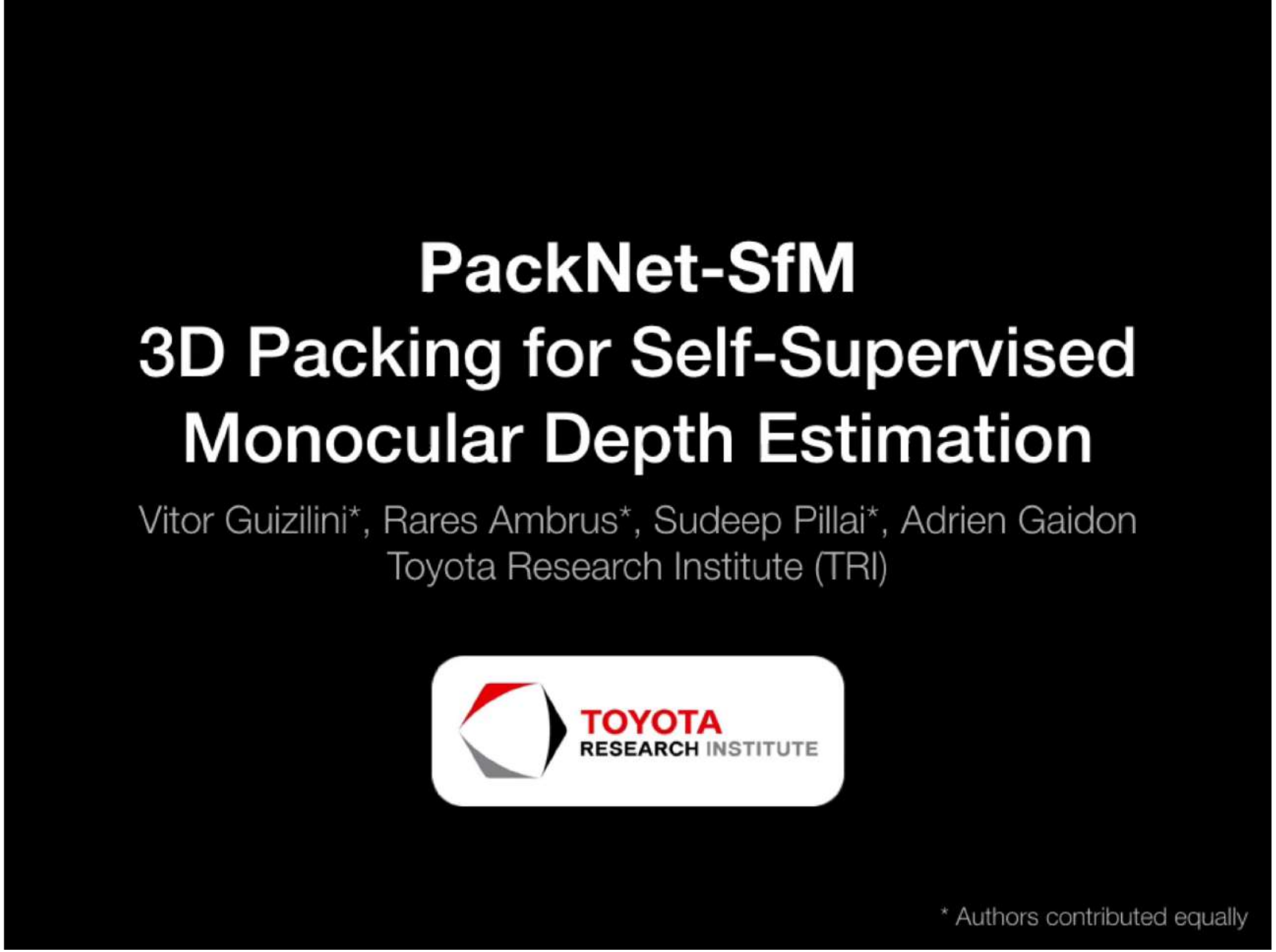
Figure 2: The network $R = (D, C)$ reasons about the scene by estimating optical flow over static regions using depth, D, and camera motion, C. The optical flow network F estimates flow over the whole image. The motion segmentation network, M, masks out static scene pixels from F to produce composite optical flow over the full image. A loss, E, using the composite flow is applied over neighboring frames to train all these models jointly.

Ranjan, Anurag, et al. "Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation." *CVPR*. 2019.

State of the art self-supervised depth out performs supervised learning!



Guizilini, Vitor, et al. "PackNet-SfM: 3D Packing for Self-Supervised Monocular Depth Estimation." *CVPR* (2019).

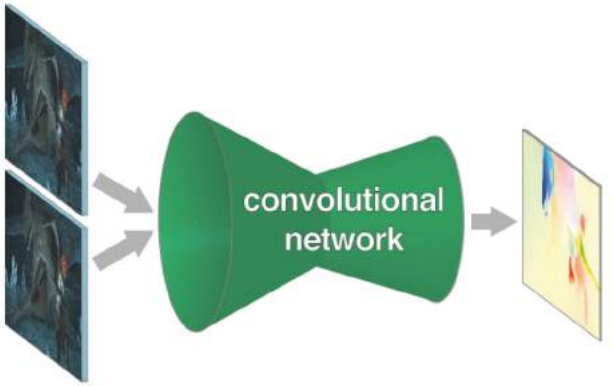


Guizilini, Vitor, et al. "PackNet-SfM: 3D Packing for Self-Supervised Monocular Depth Estimation." *CVPR* (2019).

Learning Motion

With Optical Flow and Ego-Motion Representations

Optical Flow with Deep Learning



- Learning dense correspondence between images
- No ground truth sensor -> use large synthetic datasets

Figure 5. **Flying Chairs.** Generated image pair and color coded flow field (first three columns), augmented image pair and corresponding color coded flow field respectively (last three columns).

Dosovitskiy, Alexey, et al. "Flownet: Learning optical flow with convolutional networks." *ICCV*. 2015.

Improving FlowNets with Hierarchy & Geometry

- Use learnable cost-volumes
- Hierarchical refinement to reduce disparity domain

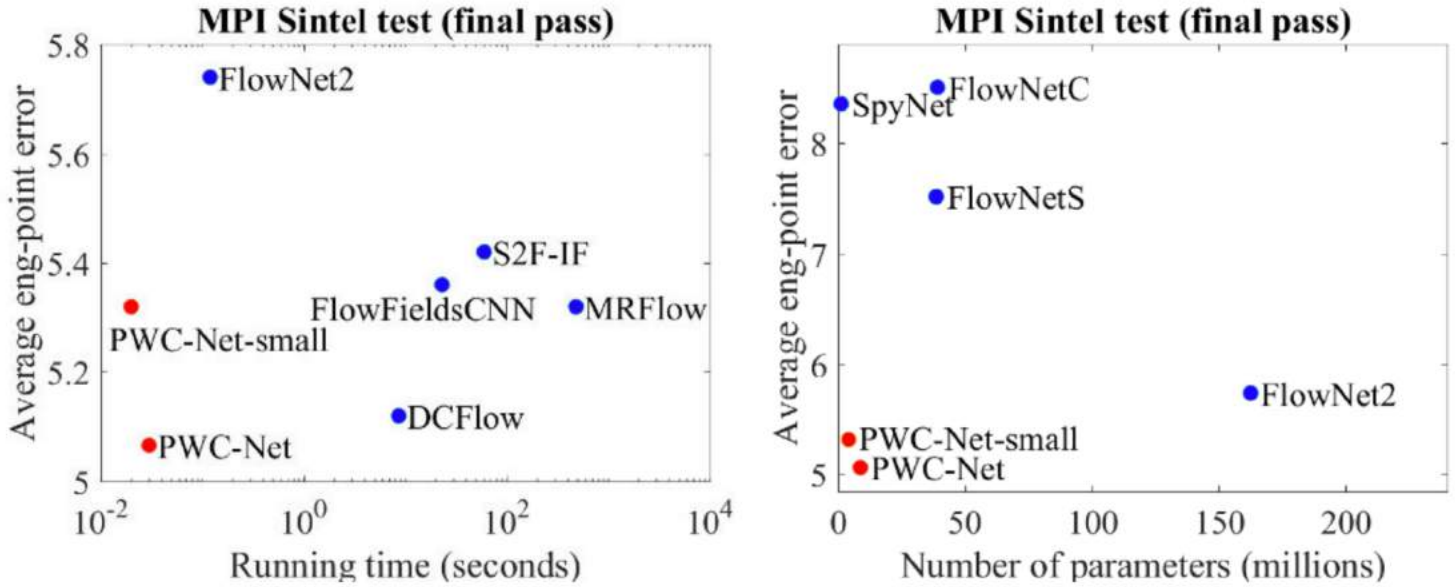
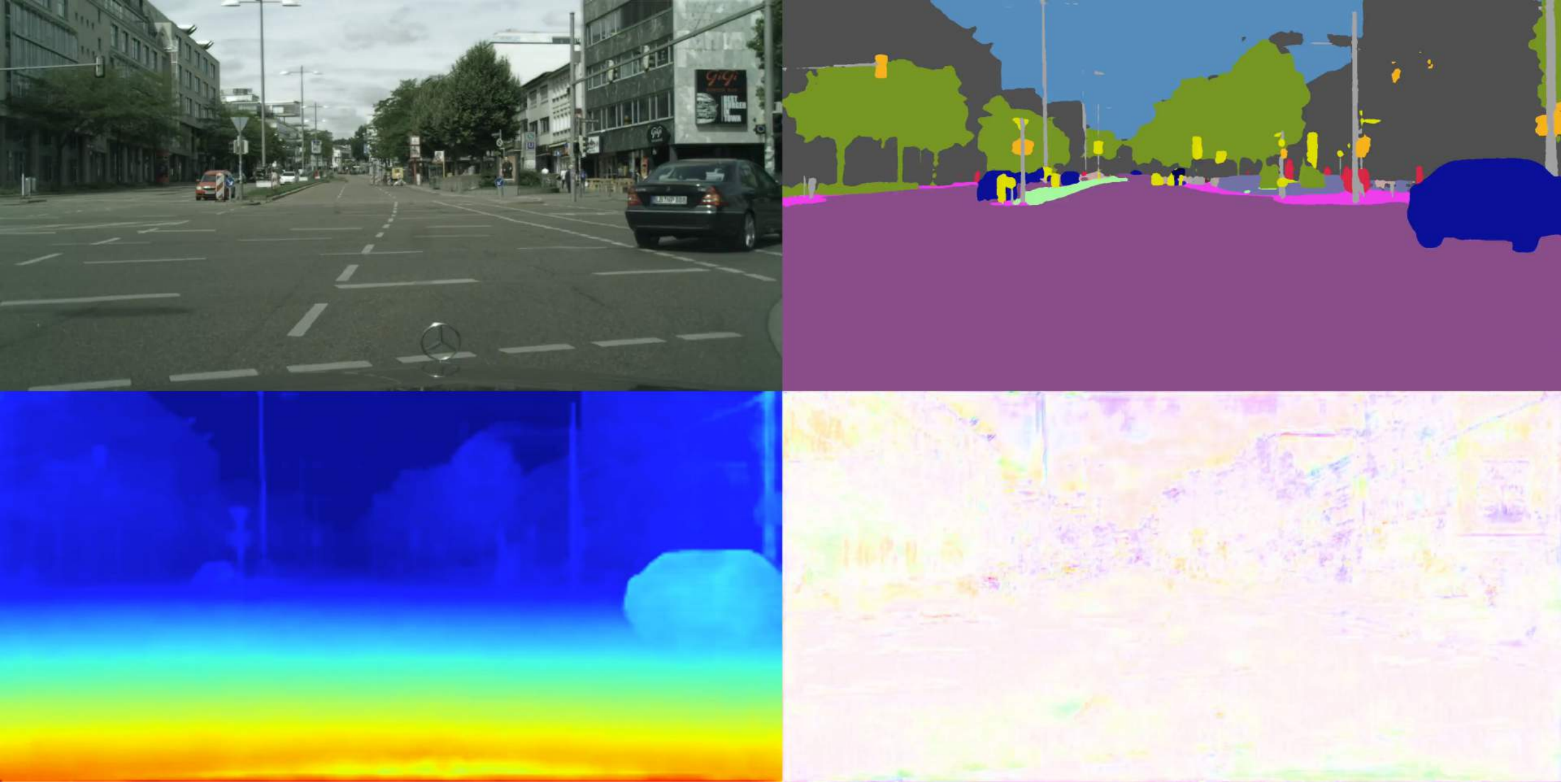


Figure 1. Left: PWC-Net outperforms all published methods on the MPI Sintel final pass benchmark in both accuracy and running time. Right: among existing end-to-end CNN models for flow, PWC-Net reaches the best balance between accuracy and size.

Sun, Deqing, et al. "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume." *CVPR*. 2018.

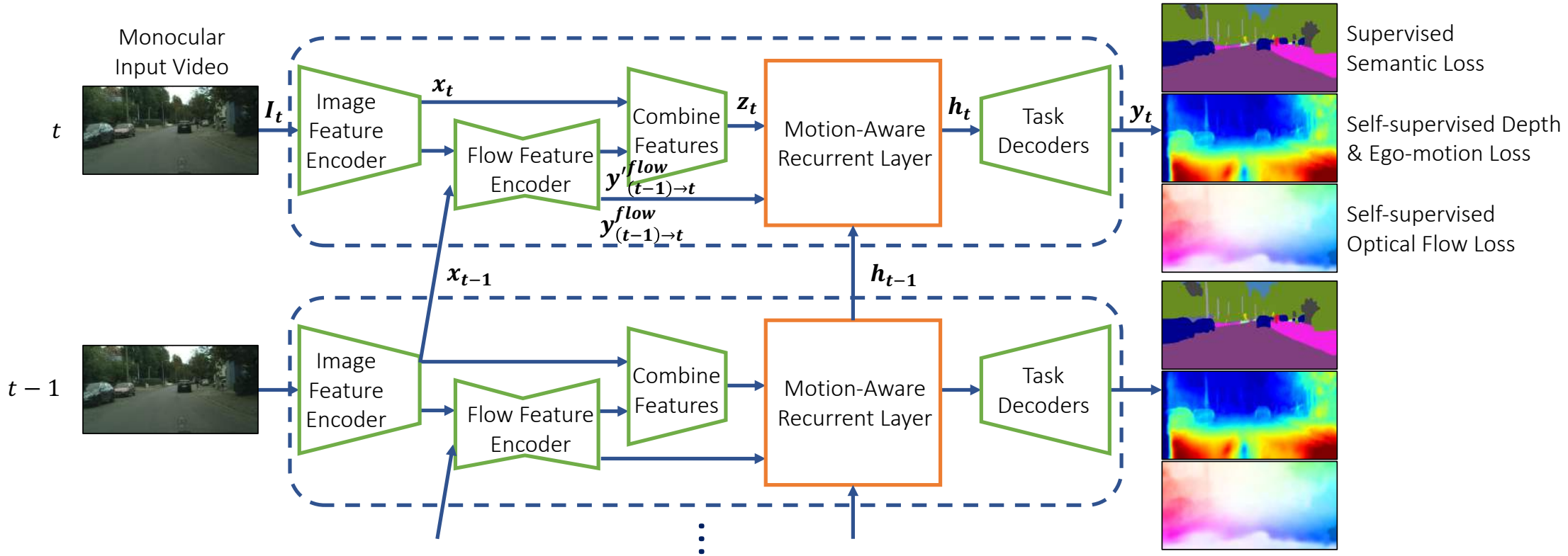


Alex Kendall. **Geometry and Uncertainty in Deep Learning for Computer Vision.** PhD Thesis, University of Cambridge, 2018.

Video scene understanding literature

- Accurate non-real-time graphical models: Tripathi et al., 2015; Kundu et al. 2016; Budvytis et al. 2010
- Focusing on real-time performance with conditional computation: Shelhamer et al. 2016; Zhu et al. 2017
- One shot mask propagation: Tokmakov et al. 2017; Tsai et al. 2016; Vertens et al. 2017
- Only two-frame: Gadde et al., 2017; Zhu et al., 2017; Zhou et al. 2018
- RNN models which perform worse than single frame models! Patraucean et al. 2015; Valipour et al. 2017

VideoSegNet Architecture



Alex Kendall. **Geometry and Uncertainty in Deep Learning for Computer Vision.** PhD Thesis, University of Cambridge, 2018.

Three tricks to enable Video SegNet

1. Account for motion and geometry when propagating features
   - Unlike RNNs for vectors, convolutional RNN features are not aligned spatially over time due to motion
2. Provide a loss at each timestep
   - Semantic are expensive to label
   - We leverage self supervised learning for motion and geometry – for free at each timestep
3. Use temporal data augmentation
   - Learn stable features by augmenting sequence length and label position
   - Challenging to fit in memory!

Alex Kendall. **Geometry and Uncertainty in Deep Learning for Computer Vision.** PhD Thesis, University of Cambridge, 2018.

Self-Supervised Learning of Motion

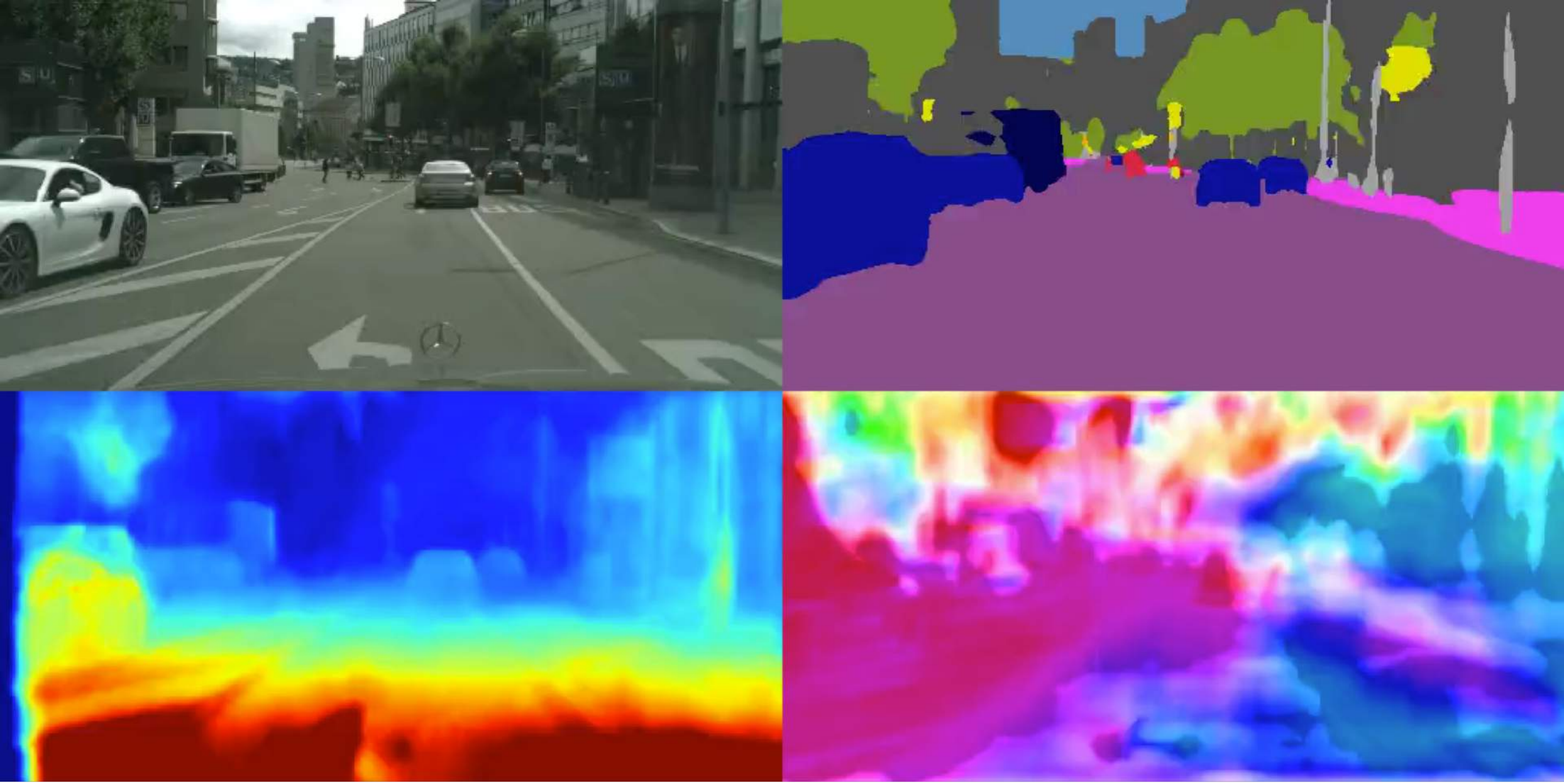
We can learn optical flow, depth and ego-motion using self-supervised losses based on photometric reprojection error.

Key idea is to learn to warp image to next timestep using spatial transformer network

e.g. for optical flow,

$$\mathcal{L}_{flow,\ t} = \frac{1}{N} \sum_{i,j} |I_t(i,j) - I_{(t-1)}(i + \mathbf{y}_{flow\ i,t}(i,j), j + \mathbf{y}_{flow\ j,t}(i,j)))|.$$

Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "**Spatial transformer networks.**" *NIPS*. 2015.



Alex Kendall. **Geometry and Uncertainty in Deep Learning for Computer Vision.** PhD Thesis, University of Cambridge, 2018.

Motion-GRU

- Align features temporally with a motion gated recurrent unit.
- Significantly improves performance and temporal consistency.

$$\mathbf{g}_t = \text{sigmoid}(W_g * \mathbf{z}_t + U_g * \mathbf{h}_{t-1}^{warped} + b_g)$$
$$\mathbf{r}_t = \text{sigmoid}(W_r * \mathbf{z}_t + U_r * \mathbf{h}_{t-1}^{warped} + b_r)$$
$$\widetilde{\mathbf{h}}_\mathbf{t} = \tanh(W_h * \mathbf{z}_t + U_h * \mathbf{r}_t \cdot \mathbf{h}_{t-1}^{warped})$$
$$\mathbf{h}_t = (1 - \mathbf{g}_t) \cdot \mathbf{h}_{t-1}^{warped} + \mathbf{g}_t \cdot \widetilde{\mathbf{h}}_t,$$

Recurrent Model	Segmentation IoU	Segmentation Consistency	Depth Err. (px)	Flow Err. (px)
per-frame baseline (no motion)	63.9%	82.3%	11.2	-
GRU	63.5%	87.4%	9.3	14.7
motion-GRU	65.6%	91.9%	**9.1**	12.3
motion-GRU + consistency loss	**65.9%**	**94.2%**	9.4	**12.1**

Alex Kendall. **Geometry and Uncertainty in Deep Learning for Computer Vision.** PhD Thesis, University of Cambridge, 2018.

Using multi-task self-supervision

Tasks	Segmentation IoU	Segmentation Consistency	Depth Err. (*px*)	Flow Err. (*px*)	Egomotion Err. (*m*)
segmentation	63.8%	82.7%	-	-	-
segmentation+flow	65.1%	91.3%	-	14.1	-
segmentation+flow+mono depth	65.6%	93.8%	21.8**	12.3	0.39**
segmentation+flow+stereo depth	**65.9%**	**94.2%**	**9.4**	**12.1**	-

Alex Kendall. **Geometry and Uncertainty in Deep Learning for Computer Vision.** PhD Thesis, University of Cambridge, 2018.

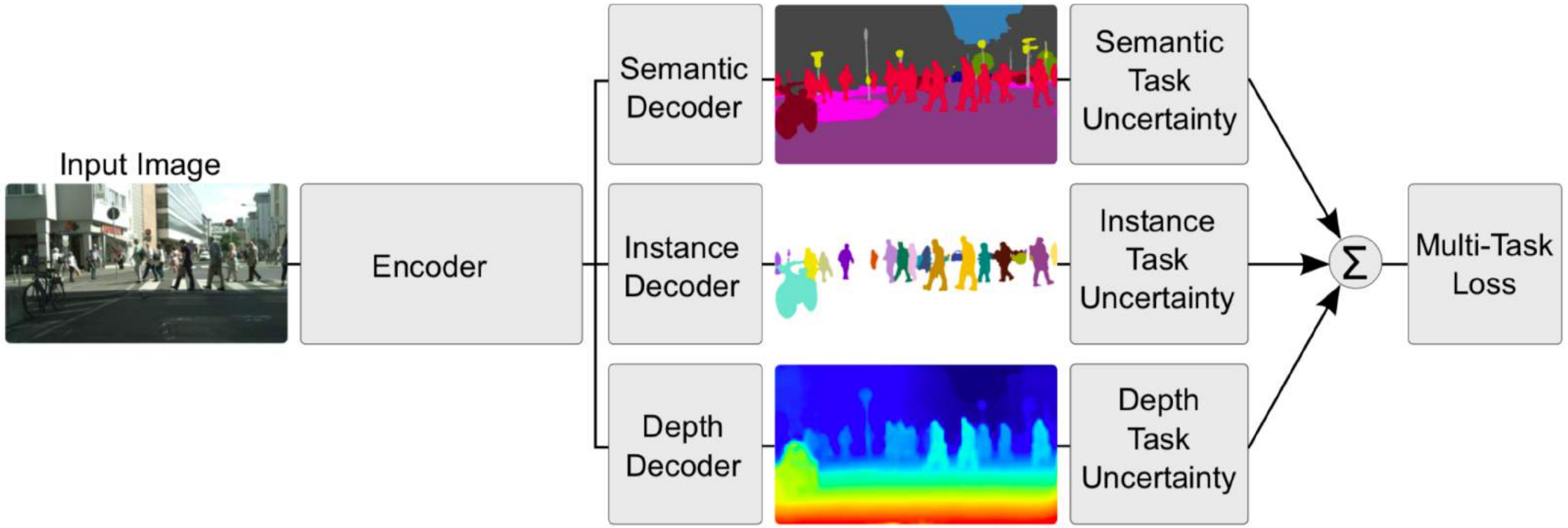
Holistic Scene Understanding

With Semantics, Motion and Geometry

Multi-task Deep Learning

- We now want to learn a representation which contains the union of all the information we need
- We also want to use information from one task to benefit the performance on another task and vice-versa

Multitask Scene Understanding Architecture

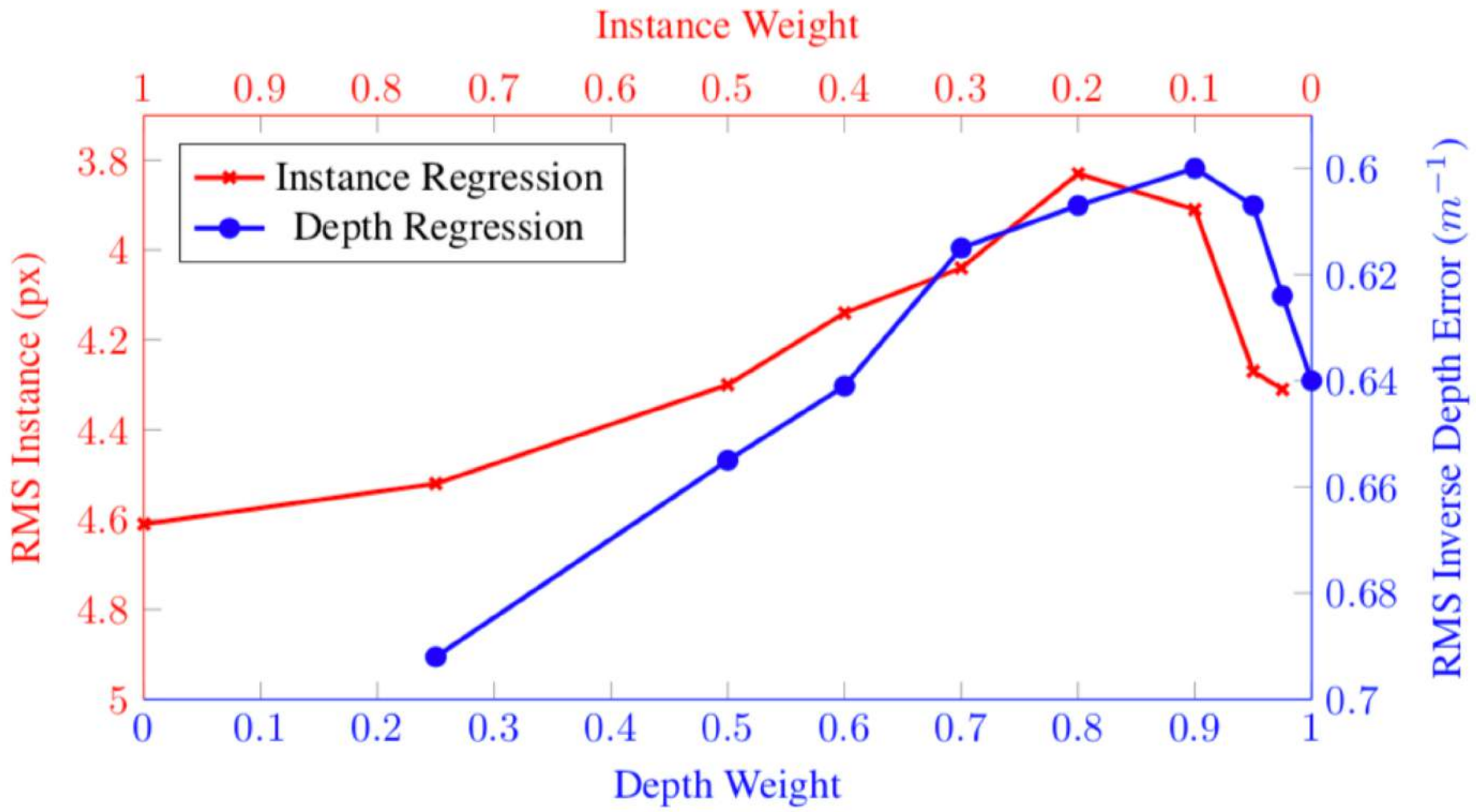


Alex Kendall, Yarin Gal and Roberto Cipolla. **Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics.** CVPR, 2018.

Multi-task deep learning literature

- **Machine Learning:** Caruana. Multitask learning. Learning to learn, 1998
- **Computer Vision:** Kokkinos. UberNet: Training a universal convolutional neural network for low, mid, and high-level vision using diverse datasets and limited memory. CVPR, 2017.
- **Medical Imaging:** SpineNet: automatically pinpointing classification evidence in spinal MRIs." MICCAI, 2016.
- **Natural Language Processing:** Collobert and Weston. A unified architecture for natural language processing. ICML, 2008.
- **Speech Recognition:** Huang et al. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. ICASSP, 2013.

All previous methods use uniform or manually tuned weights



Alex Kendall, Yarin Gal and Roberto Cipolla. **Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics.** CVPR, 2018.

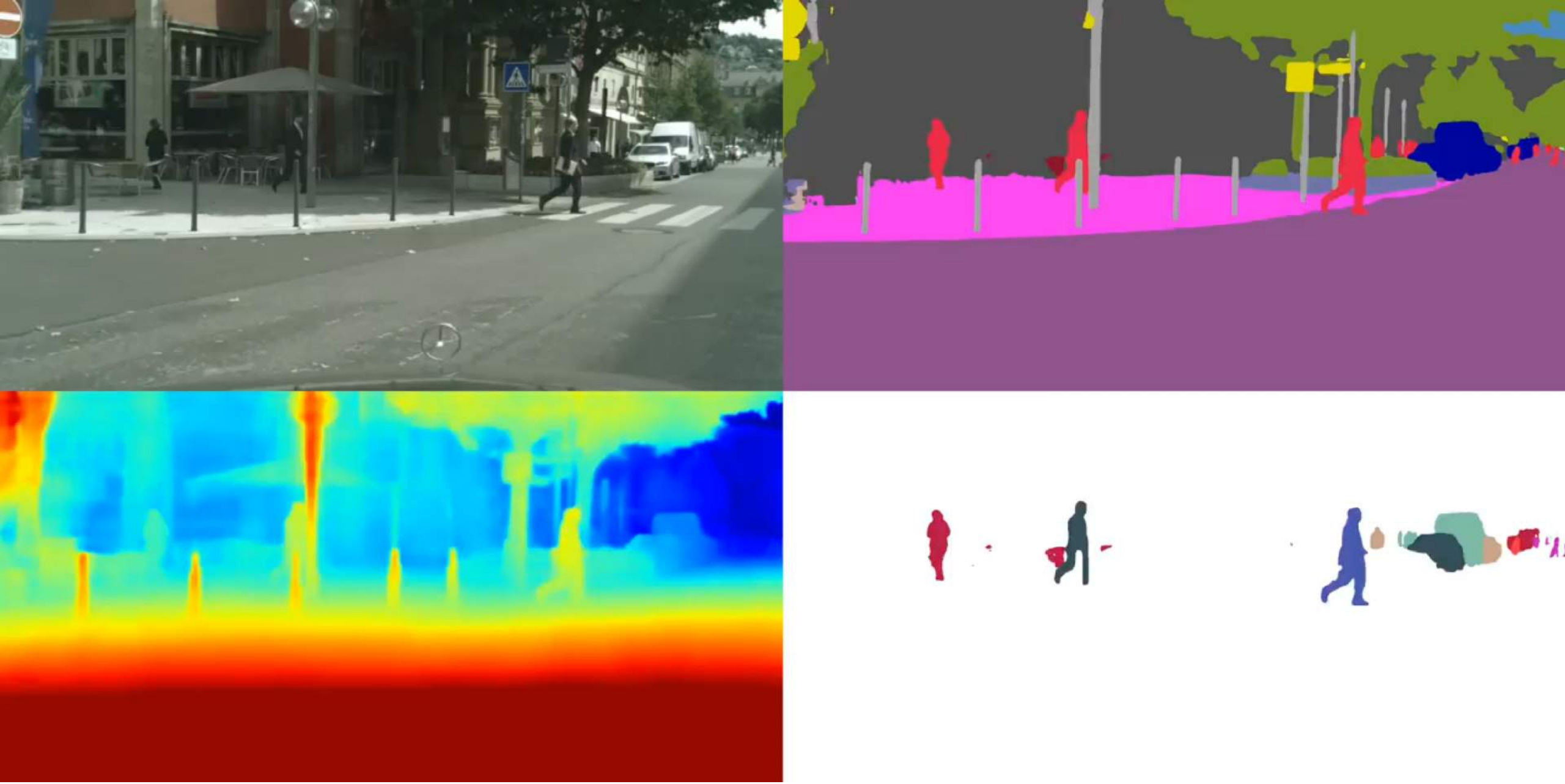
Using Task Uncertainty as Weighting

- Weighting should vary with magnitude and difficulty of task
- A measure of uncertainty is a good proxy
- We consider homoscedastic uncertainty as task uncertainty as it does not vary with input data
- Formulate as maximum likelihood estimation
- Multitask model outperforms equivalent single trained models

For example, for two regression losses the loss is given by:

$$= \frac{1}{2\sigma_1^2}\mathcal{L}_1(\mathbf{W}) + \frac{1}{2\sigma_2^2}\mathcal{L}_2(\mathbf{W}) + \log \sigma_1\sigma_2$$

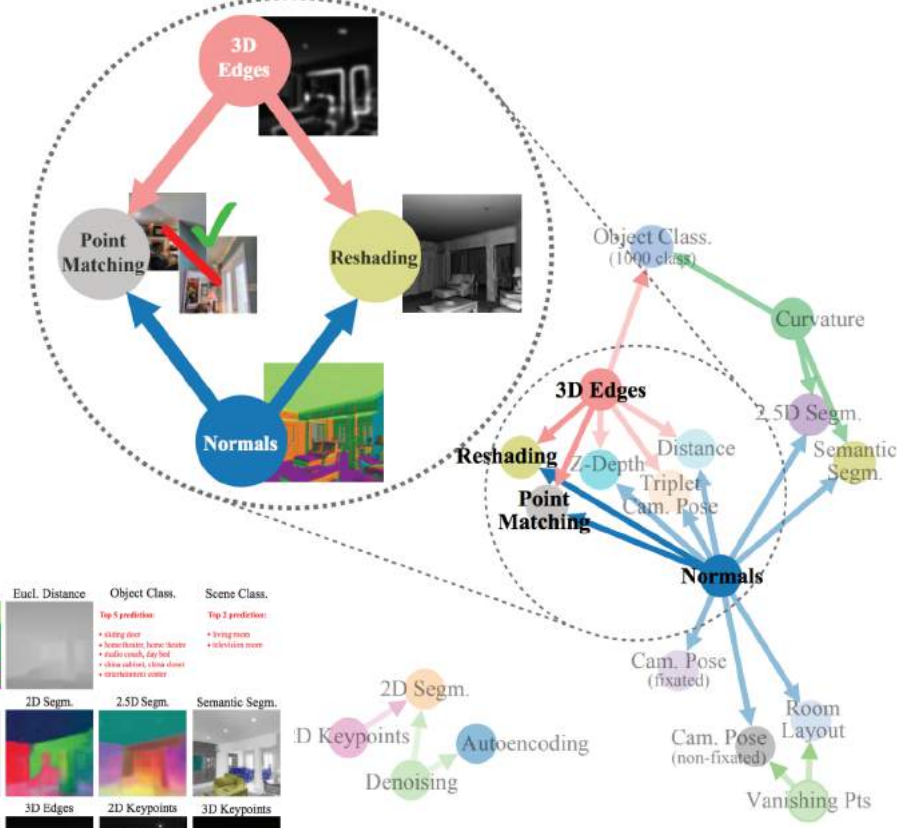
Alex Kendall, Yarin Gal and Roberto Cipolla. **Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics.** CVPR, 2018.



Alex Kendall, Yarin Gal and Roberto Cipolla. **Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics.** CVPR, 2018.

Multi-task Deep Learning

- Other solutions now proposed:
  - GradNorm: normalise gradients between various tasks
  - Multi-objective normalisation to a Pareto optimal solution
- Taskonomy: empirically measuring the related-ness or distance between tasks and how representations should relate



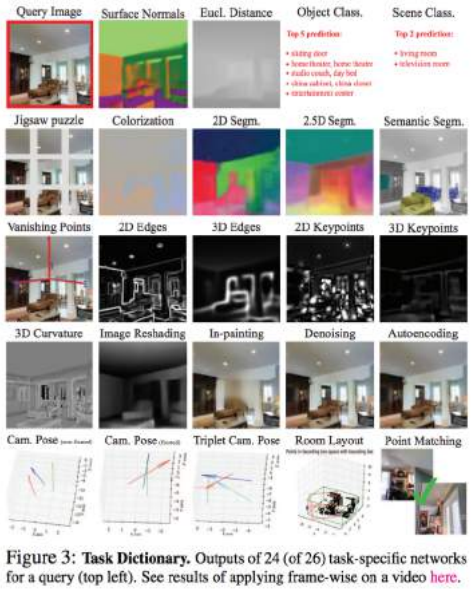


Figure 3: **Task Dictionary.** Outputs of 24 (of 26) task-specific networks for a query (top left). See results of applying frame-wise on a video here.

Chen, Zhao, et al. "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks." *arXiv preprint arXiv:1711.02257* (2017).

Sener, Ozan, and Vladlen Koltun. "Multi-task learning as multi-objective optimization." *Advances in Neural Information Processing Systems*. 2018.

Zamir, Amir R., et al. "Taskonomy: Disentangling task transfer learning." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

Let’s take a break!

Part 3: Autonomous Driving

- Some Historical Background
- Why do we need computer vision when we can learn end-to-end for action?
- Can we understanding what we don’t know?
- How do we get enough data?
- How do we interpret and debug deep learning representations?

Machine Learning for Autonomous Driving

Some Historical Background

1989 ALVINN: End-to-End Imitation Learning

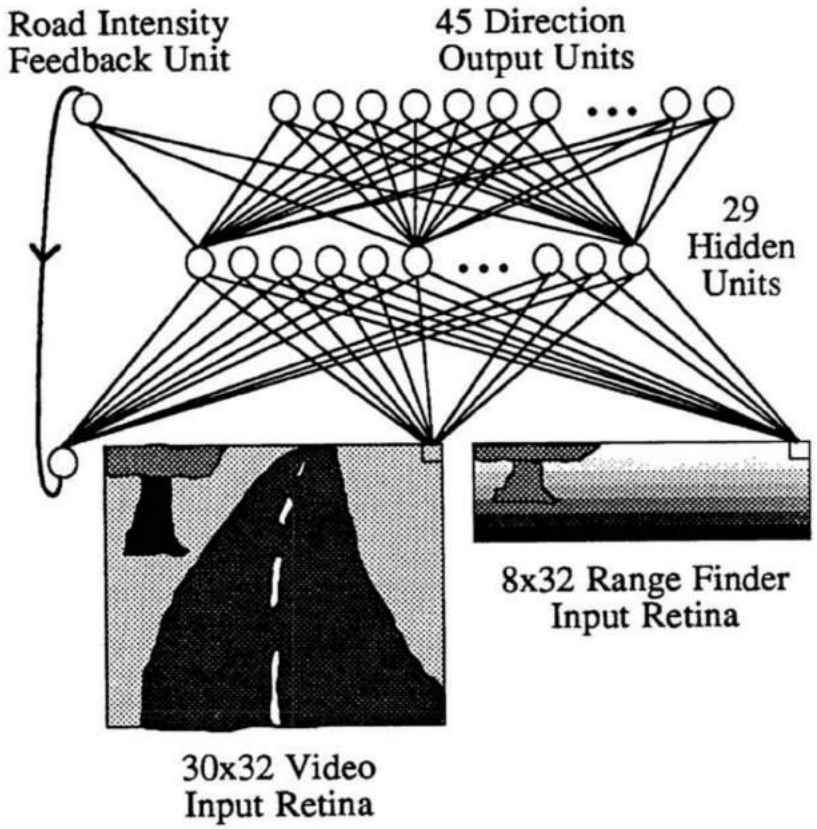


Figure 1: ALVINN Architecture

What's Hidden in the Hidden Layers?

The contents can be easy to find with a geometrical problem, but the hidden layers have yet to give up all their secrets

David S. Touretzky and Dean A. Pomerleau

tions, we fed the network road images taken under a wide variety of viewing angles and lighting conditions. It would be impractical to try to collect thousands of real road images for such a data set. Instead, we developed a synthetic road-image generator that can create as many training examples as we need.

To train the network, 1200 simulated road images are presented 40 times each, while the weights are adjusted using the back-propagation learning algorithm. This takes about 30 minutes on Carnegie Mellon's Warp systolic-array supercomputer. (This machine was designed at Carnegie Mellon and is built by General Electric. It has a peak rate of 100 million floating-point operations per second and can compute weight adjustments for back-propagation networks at a rate of 20 million connections per second.)

Once it is trained, ALVINN can accurately drive the NAVLAB vehicle at about 3½ miles per hour along a path through a wooded area adjoining the Carnegie Mellon campus, under a variety of weather and lighting conditions. This speed is nearly twice as fast as that achieved by non-neural-network algorithms running on the same vehicle. Part of the reason for this is that the forward pass of a back-propagation network can be computed quickly. It takes about 200 milliseconds on the Sun-3/160 workstation installed on the NAVLAB.

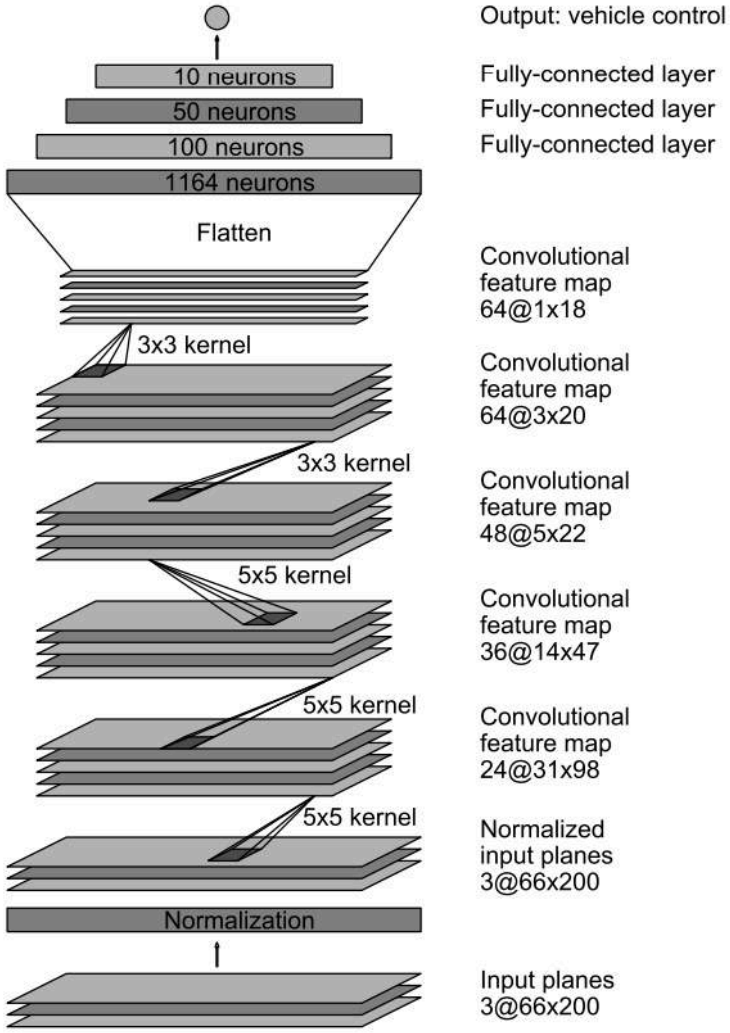
The hidden-layer representations ALVINN develops are interesting. When trained on roads of a fixed width, the network chooses a representation in which hidden units act as detectors for complete roads at various positions and orientations. When trained on roads of variable

continued

Photo 1: *The NAVLAB autonomous navigation test-bed vehicle and the road used for trial runs.*

Pomerleau, Dean A. "ALVINN: An autonomous land vehicle in a neural network." Advances in neural information processing systems. 1989.

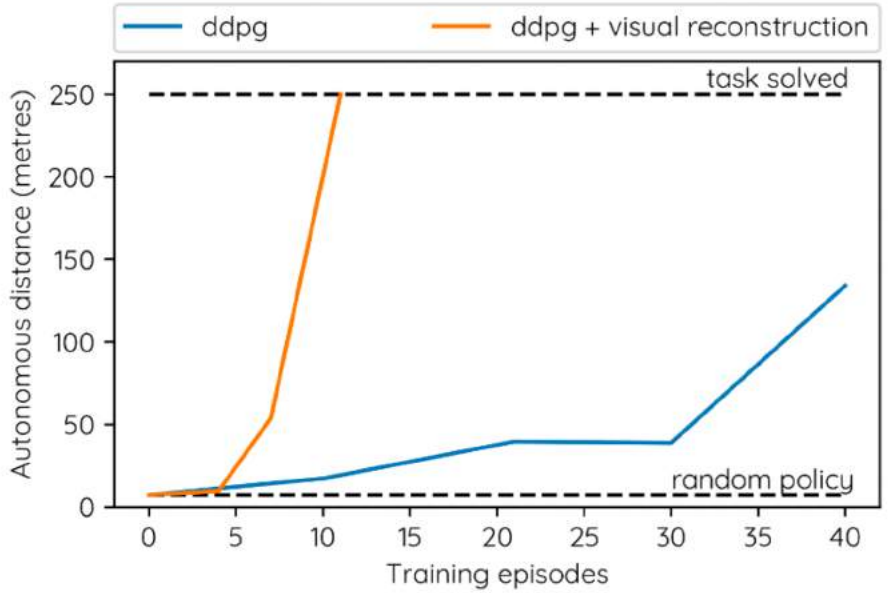
2016 NVIDIA: Lane Following on Highways



Bojarski, Mariusz, et al. "End to end learning for self-driving cars." arXiv preprint arXiv:1604.07316 (2016).

Deep reinforcement learning on a self driving car is possible!

- Data-efficiency: learning to lane follow from 11 training episodes (15 mins)
- Sparse reward: drive as far as possible without safety-driver intervention
- End-to-end deep learning from image input
- All optimisation using on-board computer



	Training			Test	
Model	Episodes	Distance	Time	Meters per Disengagement	# Disengagements
Random Policy	-	-	-	7.35	34
Zero Policy	-	-	-	22.7	11
Imitation Learning [18] †	-	250 m	2 min	41.7	6
Imitation Learning [18] †	-	12,000 m	60 min	-	0
Deep RL from Pixels	35	298.8 m	37 min	143.2	1
Deep RL from VAE	11	195.5 m	15 min	-	0

Alex Kendall et al. "Learning to Drive in a Day" International Conference on Robotics and Automation. 2019.

Urban driving with end-to-end machine learning

Inputs: camera video and a sat-nav

End-to-end deep learning

Uncertainty propagation from *sensing* to *action*

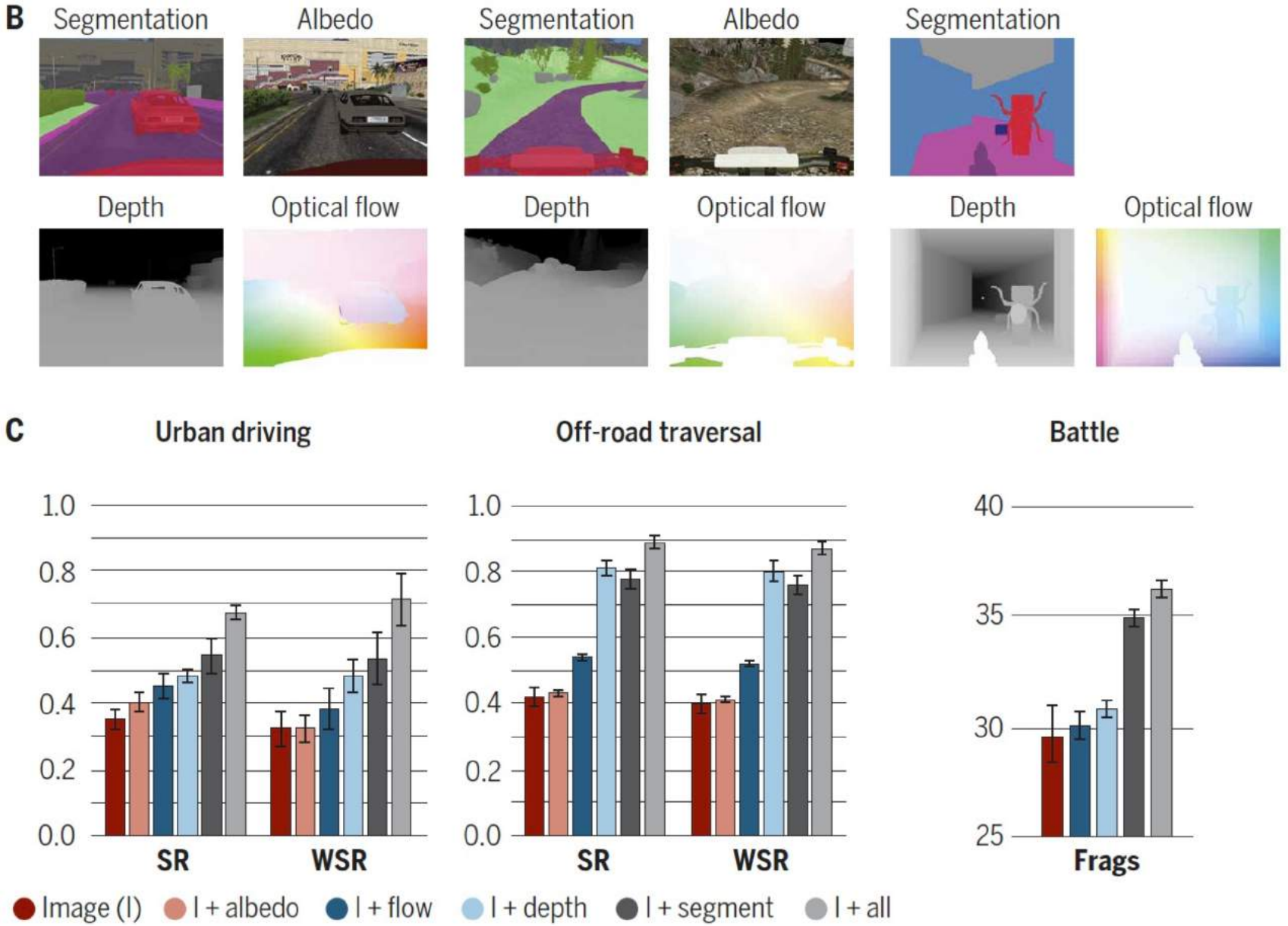
Outputs: driving commands

WAYVE

Wayve et al. Urban Driving with Conditional Imitation Learning, Under Review, (2019)

Why do we need computer vision when we can learn end-to-end for action?

- Effect of computer vision on performance for control in simulation



Brady Zhou, Philipp Krähenbühl, and Vladlen Koltun. Does Computer Vision Matter for Action? Science Robotics, 4(30), 2019

Urban driving with end-to-end machine learning

Inputs: camera video and a sat-nav

End-to-end deep learning

Uncertainty propagation from *sensing* to *action*

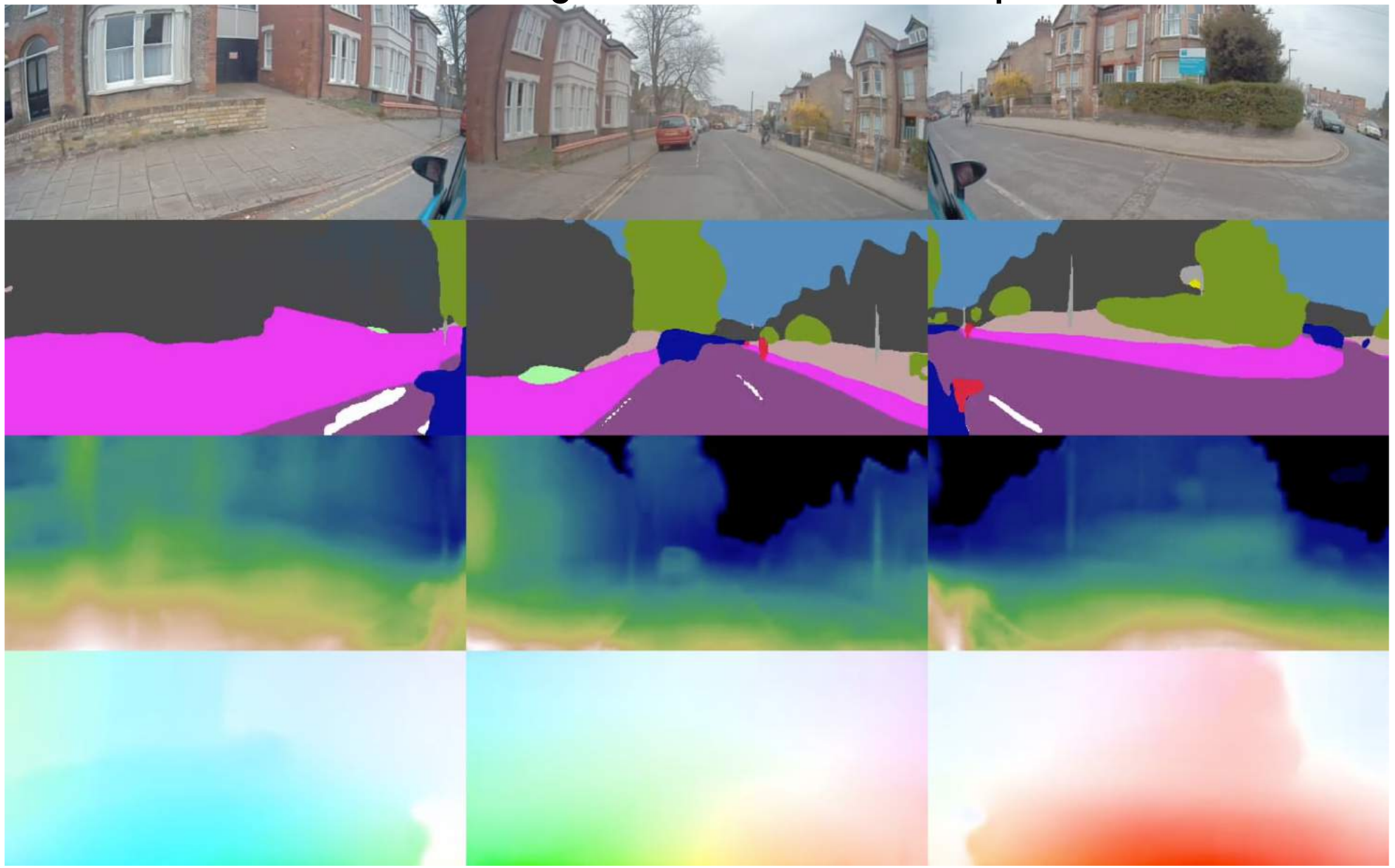
Outputs: driving commands

WAYVE

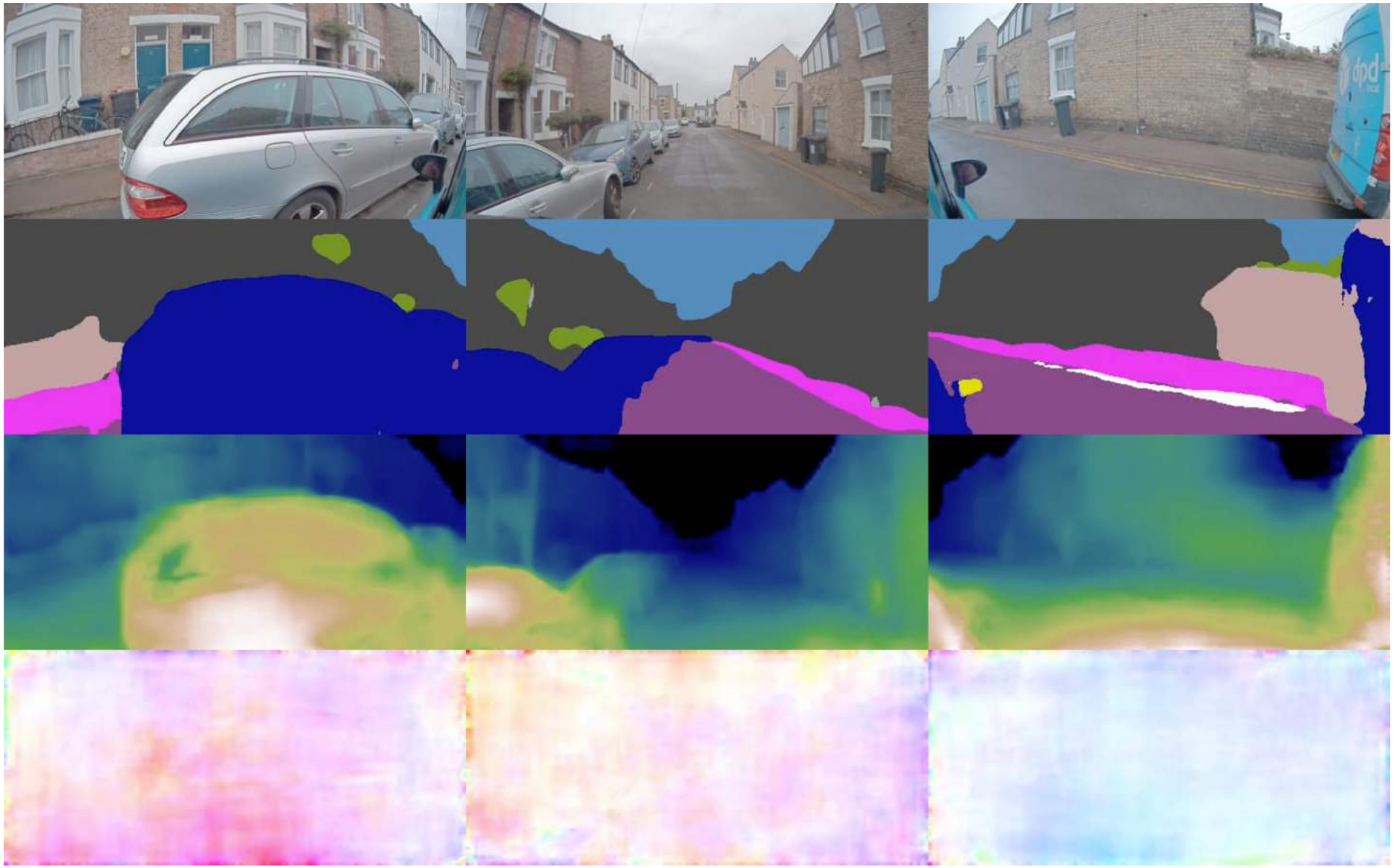
Wayve et al. Urban Driving with Conditional Imitation Learning, Under Review, (2019)

Data Collection from Human Demonstration

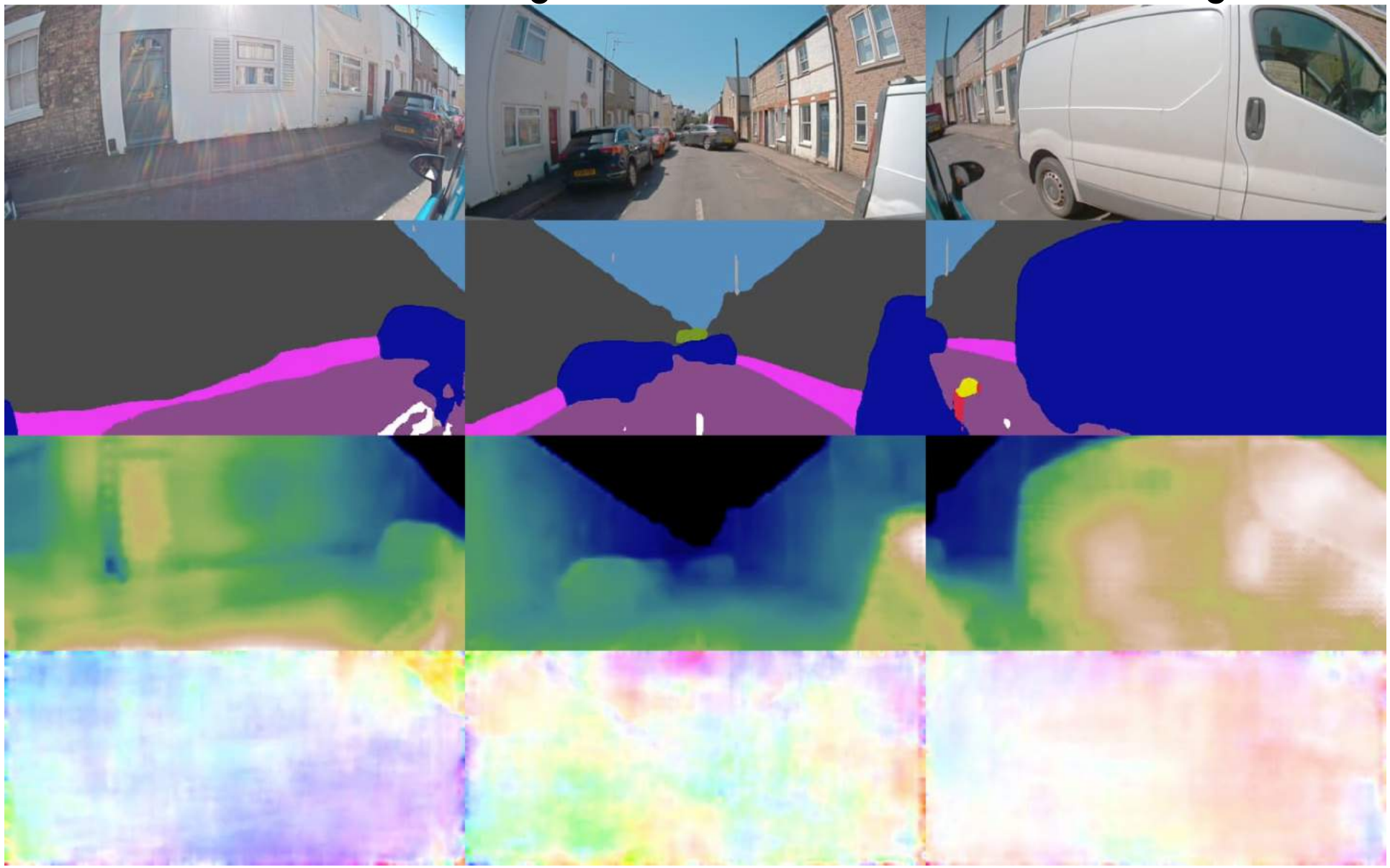
Autonomous Driving Demonstration – Complex Roads



Autonomous Driving Demonstration – Interaction with Traffic



Autonomous Driving Demonstration – Vehicle Following

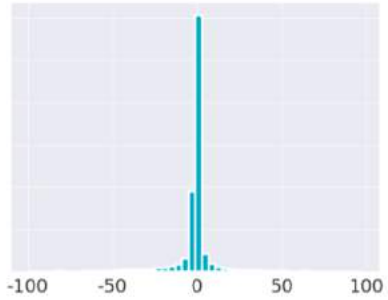


Modelling Uncertainty

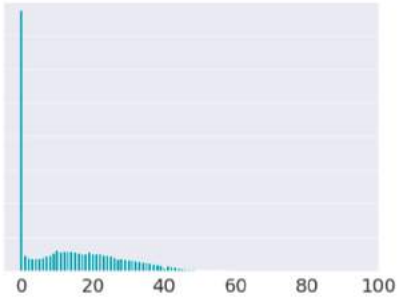
Understanding what we don't know

Why do we need to model uncertainty?

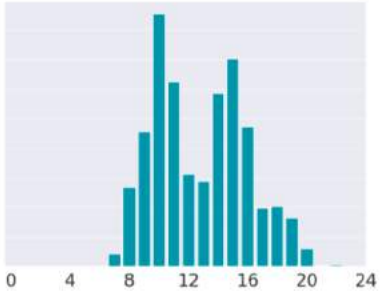
- Active learning: important to generate a learning curriculum, data is very biased with long tail distribution problems
- For safety: important to know when our model isn't confident
- For representation learning: to fuse sensory information



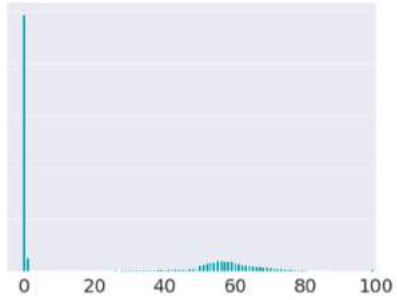
(a) Steering (± %)



(b) Speed (km/h)



(c) Time of day (hour)



(d) Throttle (%)

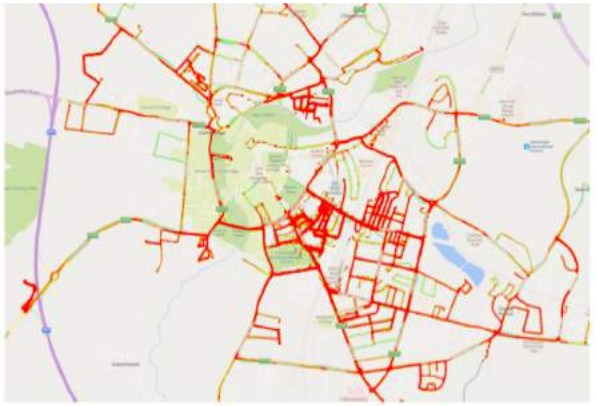


Figure 5: We collect training data driving across a European city (see 3a). The data, as is typical, is imbalanced, with the majority driving straight (5a), and a significant portion stationary (5b).

What kind of uncertainty can we model?

Epistemic uncertainty

- Measures what your model doesn't know
- Can be explained away by unlimited data

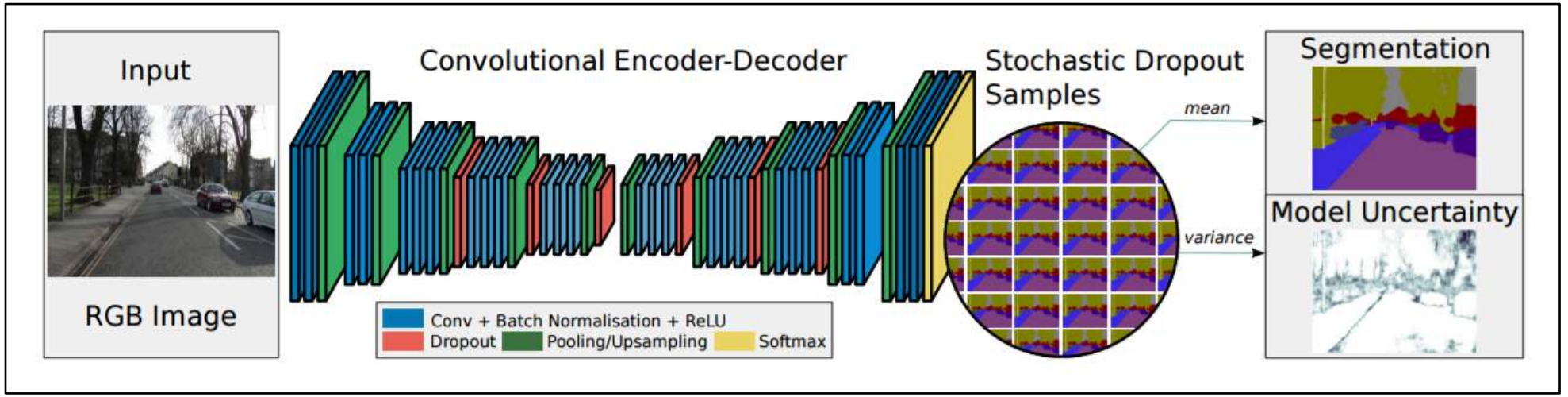
Aleatoric uncertainty

- Measures what you can't understand from the data
- Can be explained away by unlimited sensing

Der Kiureghian, A. and Ditlevsen, O. (2009). Aleatory or epistemic? does it matter? Structural Safety, 31(2):105–112.

Modeling Epistemic Uncertainty with Bayesian Deep Learning

- We can model epistemic uncertainty in deep learning models using Monte Carlo dropout sampling at test time.
- Dropout sampling can be interpreted as sampling from a distribution over models.



Alex Kendall and Yarin Gal. **What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?** NeurIPS, 2017.

Aleatoric Uncertainty with Probabilistic Deep Learning

	Deterministic Deep Learning	Probabilistic Deep Learning
Model	$[\hat{y}\,] = f(x)$	$[\hat{y}, \hat{\sigma}^2] = f(x)$
Regression	$Loss = \|y - \hat{y}\|^2$	$Loss = \frac{\|y - \hat{y}\|^2}{2\hat{\sigma}^2} + \log \hat{\sigma}$
Classification	$Loss = SoftmaxCrossEntropy(\hat{y}_t)$	$\hat{y}_t = \hat{y} + \epsilon_t \quad \epsilon_t \sim N(0, \hat{\sigma}^2)$ $Loss = \frac{1}{T}\sum_t SoftmaxCrossEntropy(\hat{y}_t)$

Alex Kendall and Yarin Gal. **What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?** NeurIPS, 2017.

Probabilistic Loss Derivation

$$Gaussian\ PDF = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

We want to directly regress this probability distribution function.

Therefore, forming a negative log likelihood loss:

$$Loss = -\log\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\right) = \frac{(x-\mu)^2}{2\sigma^2} + \log(\sigma) + constant$$

We can ignore the constant term. We want to constrain σ to be positive real, therefore we regress $\mathbf{s} := \log(\sigma^2)$.

$$Loss = \frac{1}{2} e^{-s} \left|\left|y - \hat{y}\right|\right|_2 + \frac{1}{2} s$$

Train/Test Distribution Shift

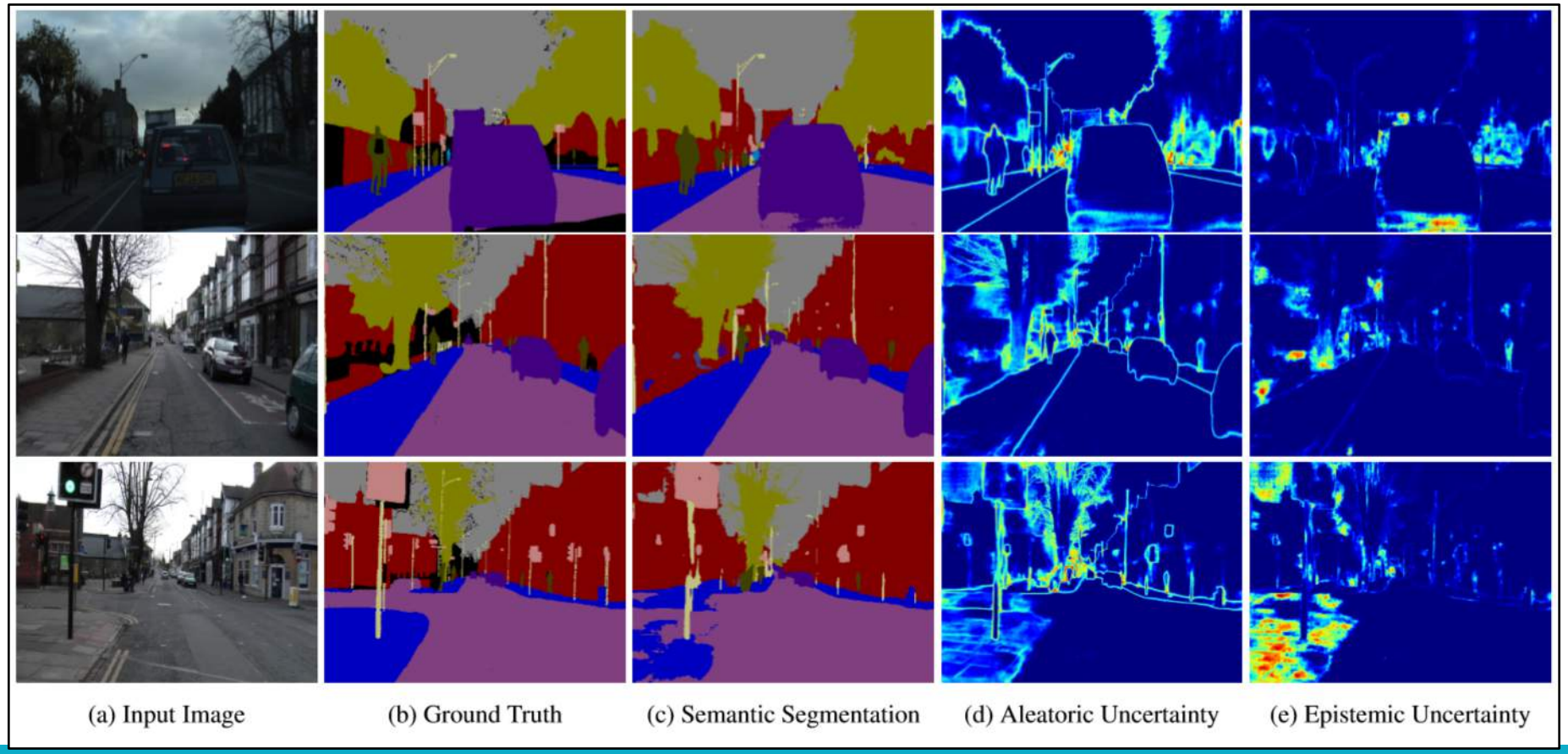
- Aleatoric uncertainty remains constant while epistemic uncertainty increases for out of dataset examples!

Train dataset	Test dataset	RMS	Aleatoric variance	Epistemic variance
Make3D / 4	Make3D	5.76	0.506	7.73
Make3D / 2	Make3D	4.62	0.521	4.38
Make3D	Make3D	3.87	0.485	2.78
Make3D / 4	NYUv2	-	0.388	15.0
Make3D	NYUv2	-	0.461	4.87

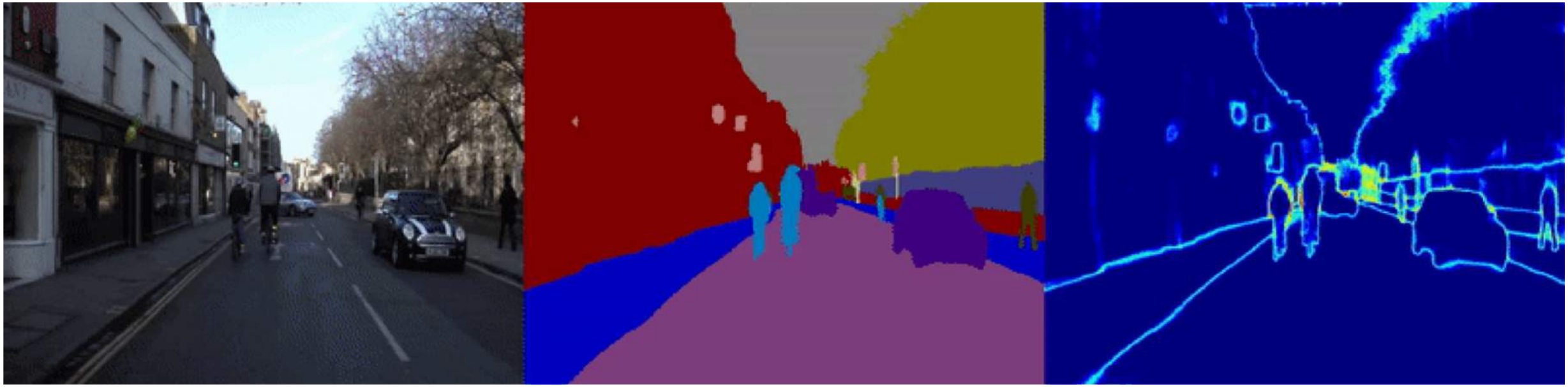
Alex Kendall and Yarin Gal. **What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?** NeurIPS, 2017.

Qualitative comparison

- Epistemic uncertainty is modeling uncertainty
- Aleatoric uncertainty is sensing uncertainty



Bayesian Deep Learning for Segmentation



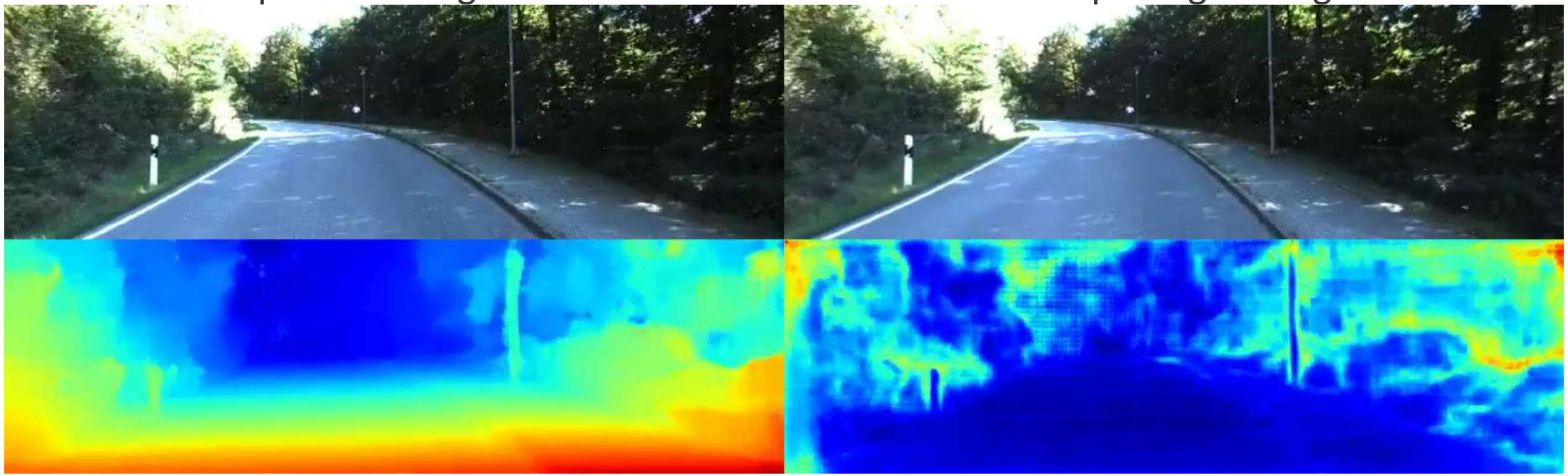
Input Image | Semantic Segmentation | Uncertainty

Alex Kendall et al. Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. BMVC 2017

Bayesian Deep Learning for Stereo Vision

Input Left Image

Input Right Image



Depth Prediction

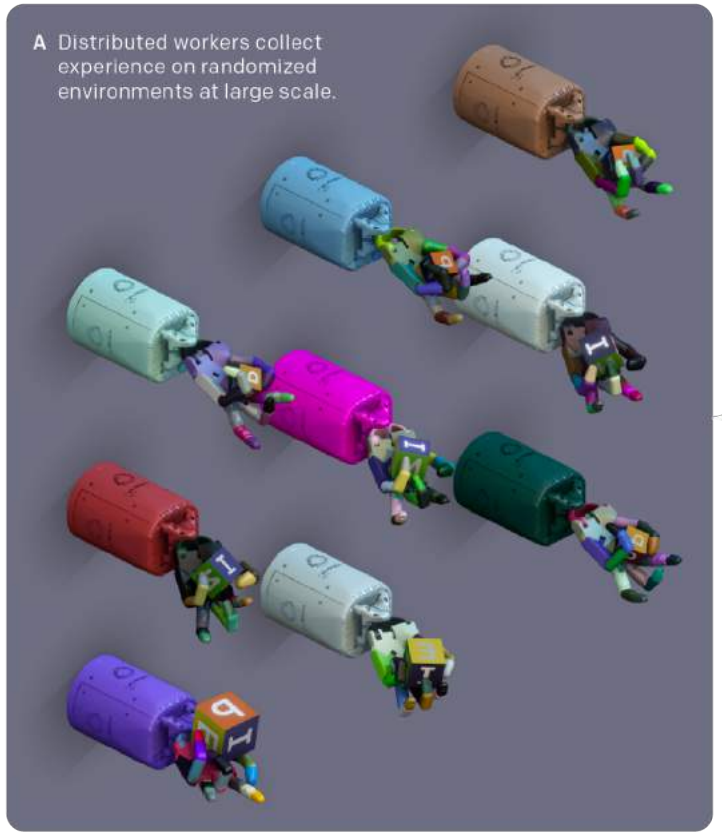
Depth Prediction Uncertainty

Alex Kendall et al. **End-to-End Learning of Geometry and Context for Deep Stereo Regression.** ICCV, 2017.

How do we get enough data?

Learning to Drive in Simulation With no Real-World Labels

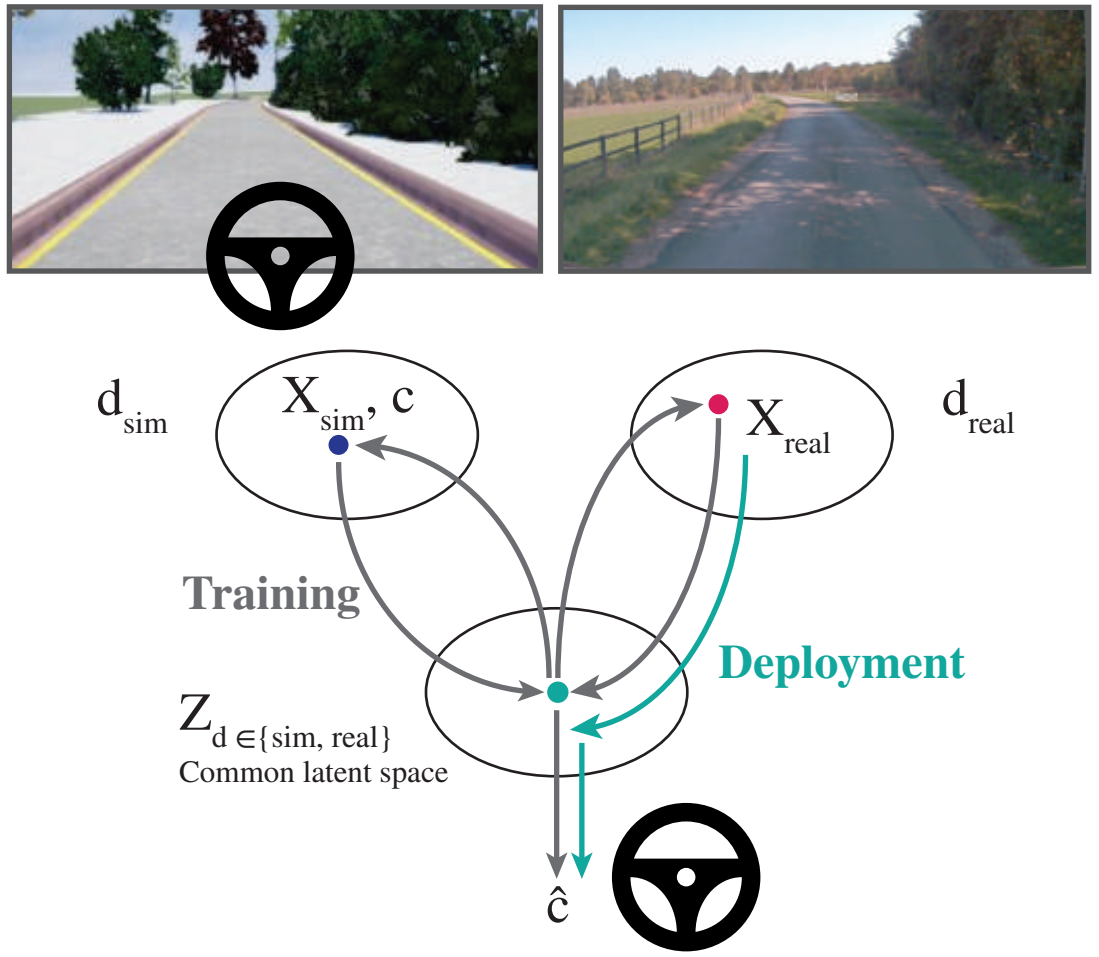
Prior Approaches to Sim2Real



- Photo-realistic simulation
- Transfer learning and fine-tuning
- Intermediate representation like segmentation
  - M. Mueller et al. Driving Policy Transfer via Modularity and Abstraction. CoRL, 2018.
- Domain randomisation
  - Andrychowicz et al. Learning dexterous in-hand manipulation. *arXiv* 2018.
  - Tobin et al. Domain randomization and generative models for robotic grasping. IROS, 2018.
  - Peng et al. Sim-to-real transfer of robotic control with dynamics randomization. ICRA, 2018.

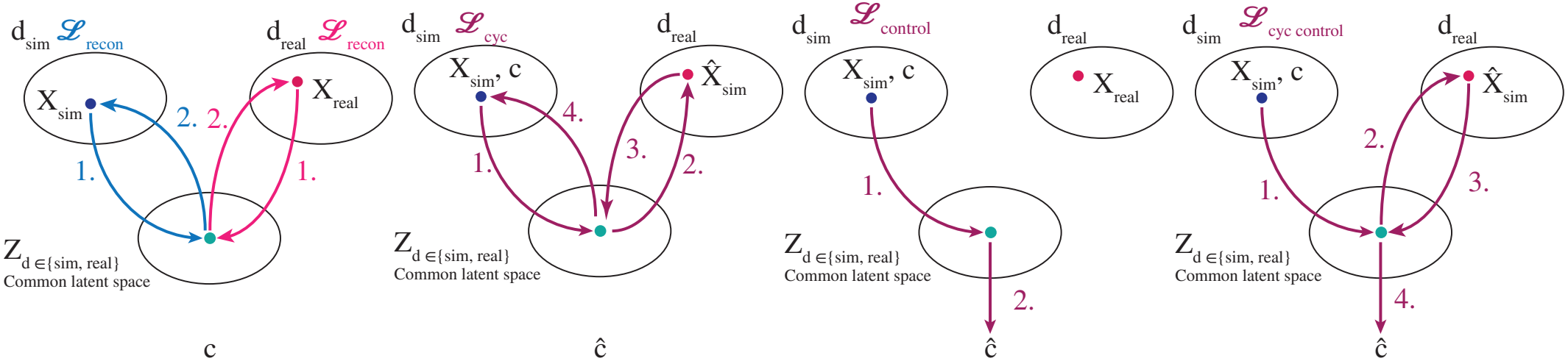
Can we train real-world models in simulated worlds?

- Zero shot sim2real
- Learn to project to a latent space for domain translation and control jointly
- Demonstrate this method can drive 3km+ on public UK roads



Alex Bewley et al. Learning to Drive from Simulation without Real World Labels. ICRA, 2019.

Learning to Drive from Simulation without Real World Labels



(a) Reconstruction loss (b) Cyclic recon. loss (c) Control loss (d) Cyclic control loss

Reconstruction Loss

$$X_d^{recon} = G_d(E_d(X_d))$$

Cyclic Reconstruction Loss

$$X_d^{cyc} = G_d(E_{d'}(G_{d'}(E_d(X_d))))$$

Control Loss

$$\hat{c} = C(E_d(X_d))$$

Cyclic Control Loss

$$\hat{c}^{cyc} = C(E_{d'}(G_{d'}(E_d(X_d))))$$

Not shown: adversarial LSGAN loss, latent reconstruction loss, perceptual loss.

Alex Bewley et al. Learning to Drive from Simulation without Real World Labels. ICRA, 2019.

Comparison to Baseline Methods

	Simulation		Real		
	MAE	Bal-MAE	MAE	Bal-MAE	DPI (metres)
Drive-Straight	0.043	0.087	**0.019**	0.093	$23^{\dagger}$
Simple Transfer	0.055	0.056	0.265	0.272	$9^{\dagger}$
Real-to-Sim Translation	-	-	0.261	0.234	$10^{\dagger}$
Sim-to-Real Translation	-	-	0.059	**0.045**	$28^{\dagger}$
Latent Feature ADA [3]	0.040	0.047	0.032	0.071	$15^{\dagger}$
Ours	**0.017**	**0.018**	0.081	0.087	$>$**3000**

Alex Bewley et al. Learning to Drive from Simulation without Real World Labels. ICRA, 2019.


B&B
Building Contractors
GB

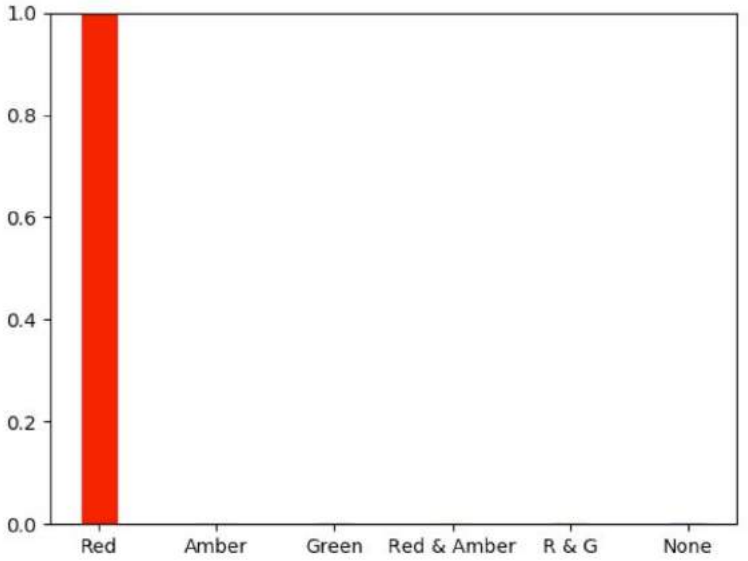
How do we interpret and debug deep learning representations?

Inspecting the state for traffic light signal

$x_{input} \rightarrow z \rightarrow y_{control}$

$z \rightarrow y_{aux}$

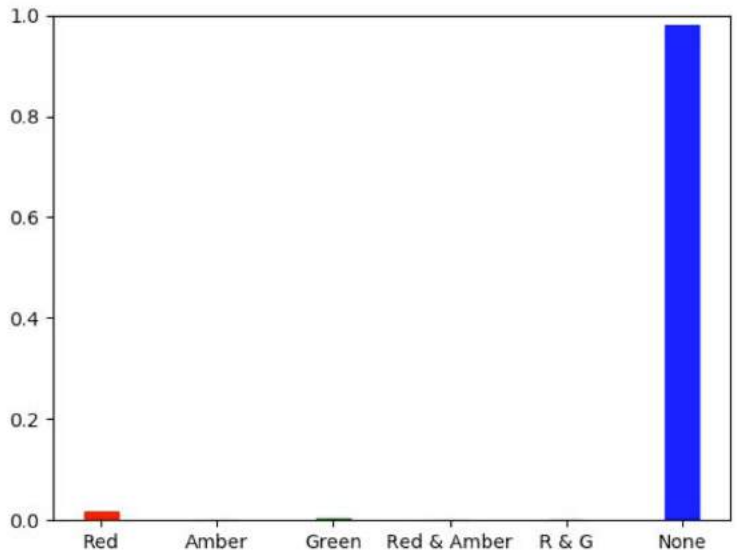
Learn to decode the high dimensional state

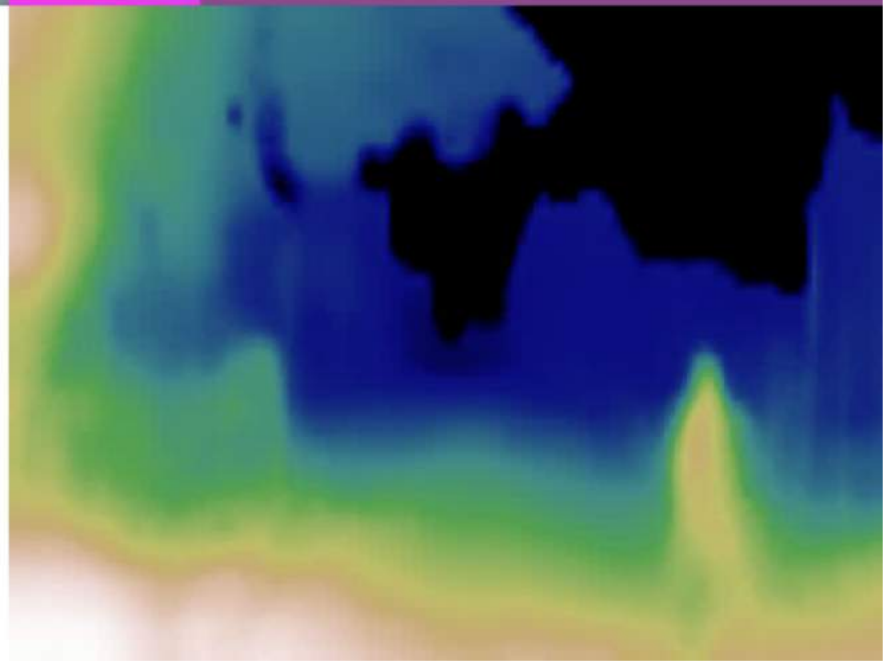


Perception Encoder Architecture	Accuracy	Mean Class Accuracy
Naive Convolutional Encoder	55%	23%
Naive Convolutional Encoder Fine-Tuned for Traffic Lights	92%	46%
Self-Attention Traffic Light Encoder	83%	47%

Table 3: Accuracy of various models classifying an image into six traffic light states (see Figure 3). This technique informs us about the efficacy of each model to extract this information.

Inspecting the state for traffic light signal, semantics and depth





Model-Based Saliency

- Faster than input-perturbation analysis and more accurate than gradient based saliency methods.

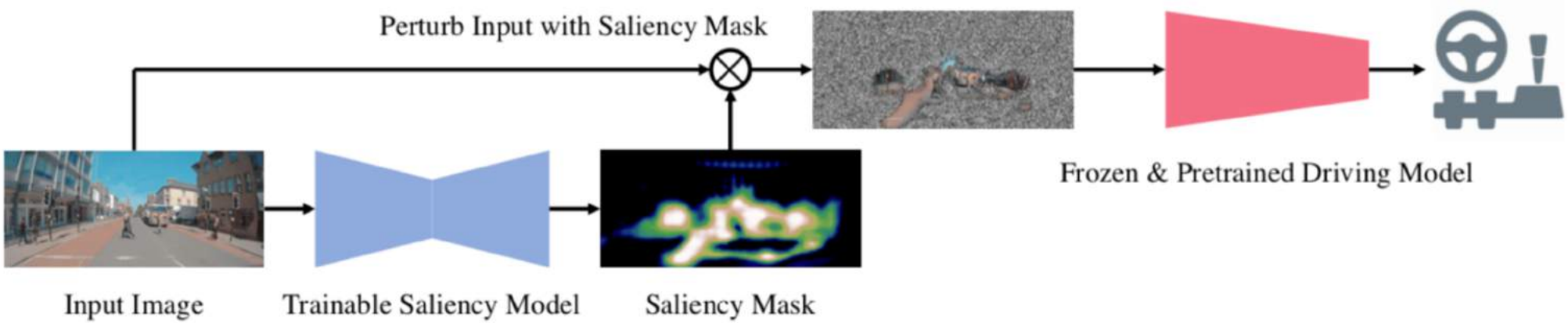


Figure 4: **Model-based saliency architecture.** We analyse a pretrained policy model by freezing the weights and learning to perturb the input with a saliency model using the loss given in Equation (1)

Dabkowski and Gal. "Real time image saliency for black box classifiers." NeurIPS. 2017.
Fong and Vedaldi. "Interpretable explanations of black boxes by meaningful perturbation." ICCV. 2017.

Model-Based Saliency

Suppose $f(\cdot)$ is our driving model and $m(\cdot)$ is our saliency model and $L(\cdot)$ is our loss function for the driving model and the operator $x \cdot m$ degrades the image with noise.

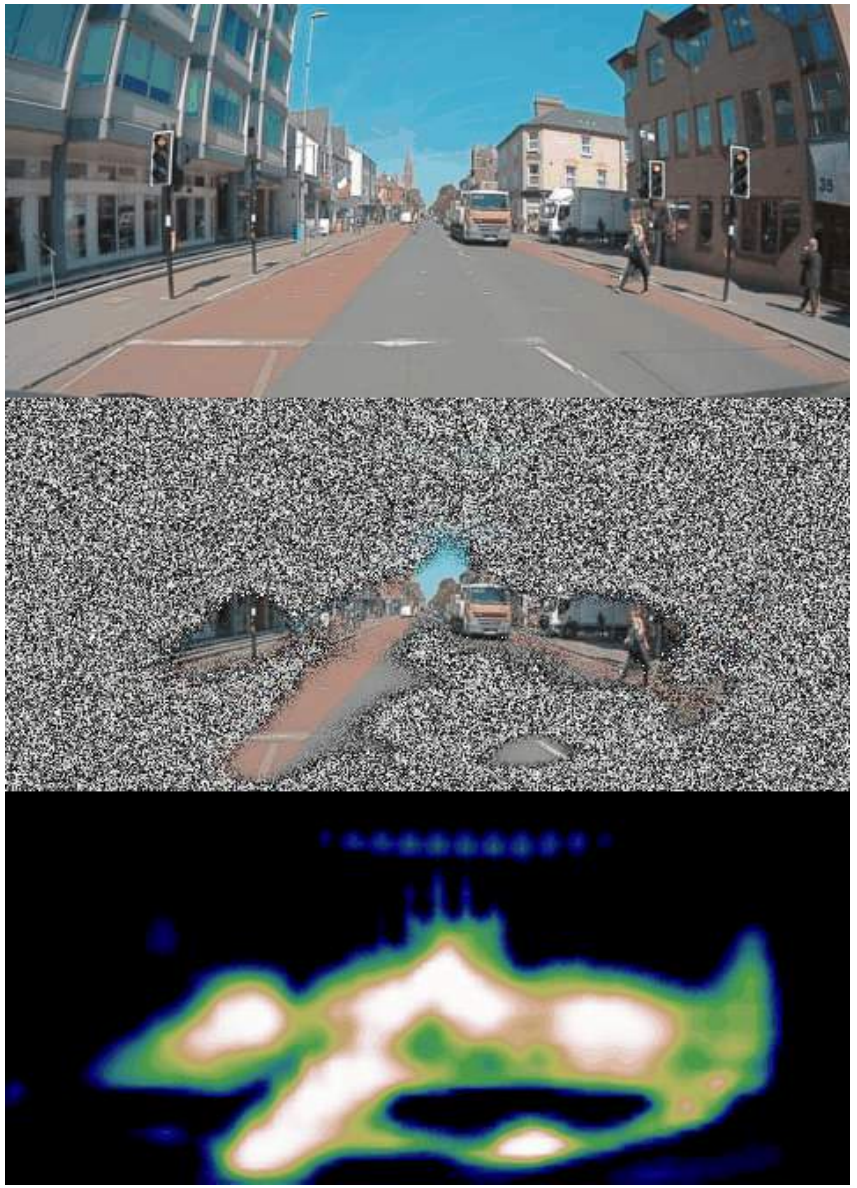
$$L = \lambda_1 |m(x)| + \lambda_2 |\nabla m(x)| + \lambda_3 L_0\left(f\left(x \cdot m(x)\right)\right) + \lambda_4 L_0\left(f\left(x \cdot \left(1 - m(x)\right)\right)\right)^{-\lambda_5}$$

Sparse saliency mask

Smooth saliency mask

Informative saliency mask

Uninformative inverse saliency mask



Dabkowski and Gal. "Real time image saliency for black box classifiers." NeurIPS. 2017.
Fong and Vedaldi. "Interpretable explanations of black boxes by meaningful perturbation." ICCV. 2017.

Model-Based Saliency

Suppose $f(\cdot)$ is our driving model and $m(\cdot)$ is our saliency model and $L(\cdot)$ is our loss function for the driving model and the operator $x \cdot m$ degrades the image with noise.

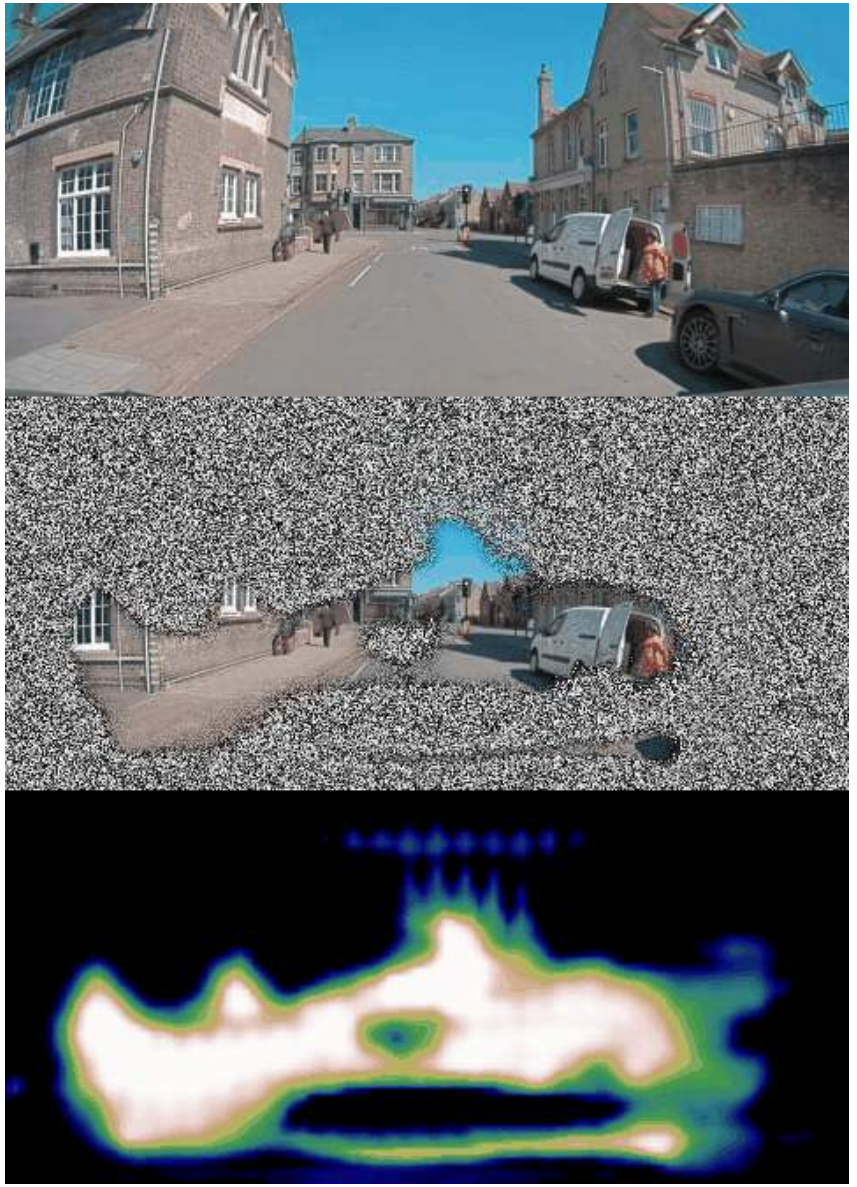
$$L = \lambda_1 |m(x)| + \lambda_2 |\nabla m(x)| + \lambda_3 L_0\left(f\left(x \cdot m(x)\right)\right) + \lambda_4 L_0\left(f\left(x \cdot \left(1 - m(x)\right)\right)\right)^{-\lambda_5}$$

Sparse saliency mask

Smooth saliency mask

Informative saliency mask

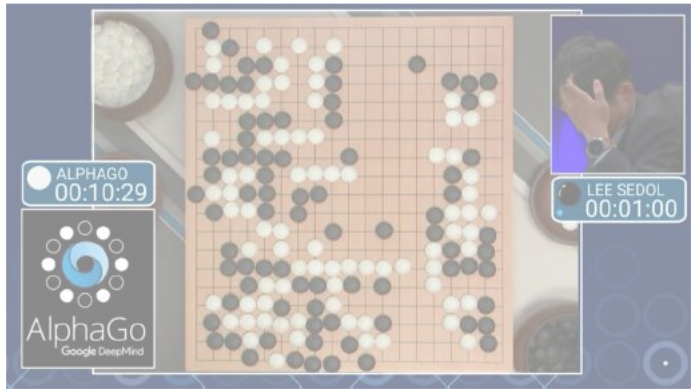
Uninformative inverse saliency mask



Dabkowski and Gal. "Real time image saliency for black box classifiers." NeurIPS. 2017.
Fong and Vedaldi. "Interpretable explanations of black boxes by meaningful perturbation." ICCV. 2017.

Conclusions

Games like Go & DOTA



DeepMind OpenAI

- Incredibly difficult action space: long term strategy, cooperation
- Very basic state space, often discrete, fully observable and noise-free

Autonomous Driving

- Quite easy action space: stop, go, left, right motion primitives
- Super challenging state space: manifold of natural images!

This needs to be solved by the computer vision community!

Future challenges for scene understanding

Deploying deep learning to robotics

- Efficient, robust and multi-task representations
- Metrics need to reflect end-to-end system performance
- Jointly training scene understanding with control policies

Future challenges for scene understanding

Structural Building Blocks:

- What happens next? Develop prediction and dynamics models (but not in RGB space!)
- Jointly training scene understanding with control policies
- Learning memory and longer term temporal features

Future challenges for scene understanding

Giving feedback and learning:

- Surpassing human demonstration learning, reward design
- Hierarchical or natural language interfaces
- Safety and understanding causal decision making factors in these models

A complete paradigm shift for AVs

- Low vehicle compute and sensor requirements
- Large training compute and data requirements
- Increased vehicle intelligence
- No reliance on HD-maps
- Ability to leverage simulation for training
- Abundance of open and interesting research questions!

Come work with our team wayve.ai/careers

WAYVE

Thank you & References

- Slides and publications: alexgkendall.com
- Technology demonstration videos: wayve.ai/blog

Thank you to the amazing people who made this work possible:

Roberto Cipolla, Vijay Badrinarayanan, Yani Ioannou, Yarin Gal, Tom Roddick, Matthew Grimes, Anthony Hu, Adrian Weller, Amar Shah, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abe Bachrach, Adam Bry, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, Corina Gurau, Richard Shen, Nikolay Nikolov, Siddharth Sharma, Sean Micklethwaite, Yuxuan Liu

UNIVERSITY OF CAMBRIDGE

WAYVE