**2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)**
**Banff Center, Banff, Canada, October 5-8, 2017**

# Estimating High Definition Map Parameters with Convolutional Neural Networks

Sebastian Bittel
MBRDNA
Sunnyvale, USA

Timo Rehfeld
MBRDNA
Sunnyvale, USA

Michael Weber
FZI
Karlsruhe, Germany

J. Marius Zöllner
FZI
Karlsruhe, Germany

**Deep Learning – UFES 2017**
Frederico Damasceno Bortoloti

# I. INTRODUCTION

- Self-driving cars heavily rely on high definition (HD) map data
  - stay within the lane
  - stop at the right spot in an intersection

# I. INTRODUCTION

- High definition (HD) map data
  - Lane markers in centimeter precision
  - Stop lines
  - Landmarks for vehicle localization
  - Traffic rules
  - etc

✓ Expensive to generate!

✓ Needs to be updated on a regular basis!

# I. INTRODUCTION

- Train a neural network to estimate essential parameters of high definition maps based on data from laser scanners and cameras

# I. INTRODUCTION

- Pre-requisites:
  - Limited area with precise map information
  - Localization algorithm
- Inputs:
  - Map data + sensor inputs
- Outputs:
  - Validation of existing maps (changes)
  - Support localization

# II. RELATED WORK

- How the street layout or distances to the roadside can be estimated.
  - Vision-based highway border detection using Hough voting (Yu et al., 2015)
  - Highly accurate map to precise online localization (no GNSS) (Schreiber et al., 2013)
  - Grid map extracted from 3D lidar point clouds to recognize intersections (Zhu et al., 2012)
  - Street layout, (number and location of lanes, 3D location and orientation of other traffic participants), in a holistic optimization framework (Geiger et al., 2014)

# II. RELATED WORK

- Deep learning approaches
  - Benchmark like Cityscapes (Cordts et al., 2016)
  - CNN to detect other cars and lane markings based on images from a frontfacing camera (Huval et al., 2015)
  - CNN using an occupancy grid map as input in order to classify the grid map into four categories: freeway, highway, parking area and urban (Seeger et al., 2016)

# II. RELATED WORK

- Deep learning approaches
  - Multiple CNNs using Open Street Map based on Google Street View images in order to infer road layout attributes, such as: distance to intersection, possible heading angles, speed limit and number of lanes (Seff & Xiao, 2016)
  - CNNs to learn a subset of the environment (affordance indicators) such as distance from the ego-vehicle to different lanes, angle of the vehicle towards the road, and distance to leading vehicles (Chen et al., 2015)

# III. APPROACH

- Estimate high definition map parameters from sensor data
  - Validate existing high definition maps
  - Improve localization

# III. APPROACH

- Convolutional Neural Network (CNN)
  - A. Input modalities
    - 1) Dynamic Grid Map
    - 2) Lidar Intensity Map
    - 3) Semantic Map
- Existing method for localization generates ground truth samples for training
  - B. Ground Truth Extraction

# III. APPROACH

- Estimates
  - (1) the distance from the ego-vehicle to leftmost and rightmost road boundary + the orientation of the ego-vehicle with respect to the lanes (regression)
  - (2) the number of lanes (classification)
  - (3) the street topology type (straight road or intersection) (classification)

## III. APPROACH

Inputs overlaid by the high definition map (red, blue and white lines)

Input modalities


Green lines: prediction for the ’road boundary’ regression task. The prediction is perfect if it aligns exactly with the outer red lines of the reference map.

White box on the top: predictions (green) and ground truth (blue) for the ’street type’ and ’number of lanes’ classification task.

# A. Input Modalities

1) Dynamic Grid Map
2) Lidar Intensity Map
3) Semantic Map

# 1) Dynamic Grid Map

- Grid map (Nuss et al., 2016)
- Each cell has an occupancy probability that encodes how likely this grid cell is occupied by an obstacle
- Particle filter to track moving objects
- Encodes the three values (occupancy and 2D velocity)
  - occupancy is encoded with level of gray
  - velocity direction is encoded via a color wheel

# 1) Dynamic Grid Map

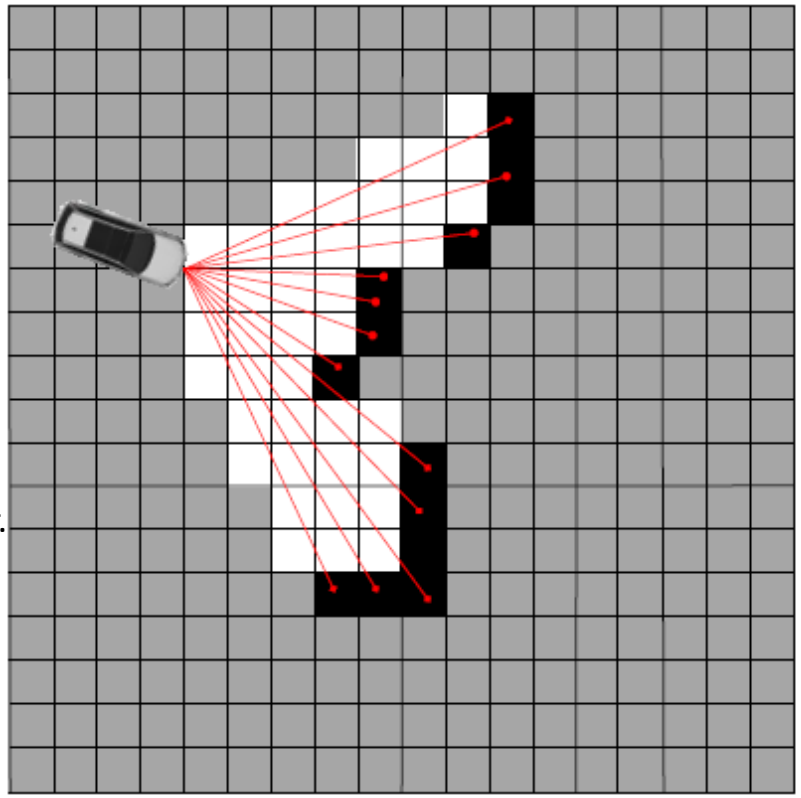
- D. Nuss, S. Reuter, M. Thom, T. Yuan, G. Krehl, M. Maile, A. Gern, and K. Dietmayer, “A Random Finite Set Approach for Dynamic Occupancy Grid Maps with Real-Time Application,” ArXiv e-prints, 2016.

# 1) Dynamic Grid Map

- Probability hypothesis density / multi-instance Bernoulli (PHD/MIB) filter is applied to estimate the dynamic state of grid cells.

Occupancy probabilities of two-dimensional grid cells, reasoning on a multi-beam laser range measurement.

Grid cells with a high probability of being occupied are colored black, free grid cells are marked with white color. Grid cells with an unknown state (same probability for both occupied and free) are displayed in gray color.



$p_{z_{k+1}}(O_{k+1}|z_{k+1}) = 0.95$

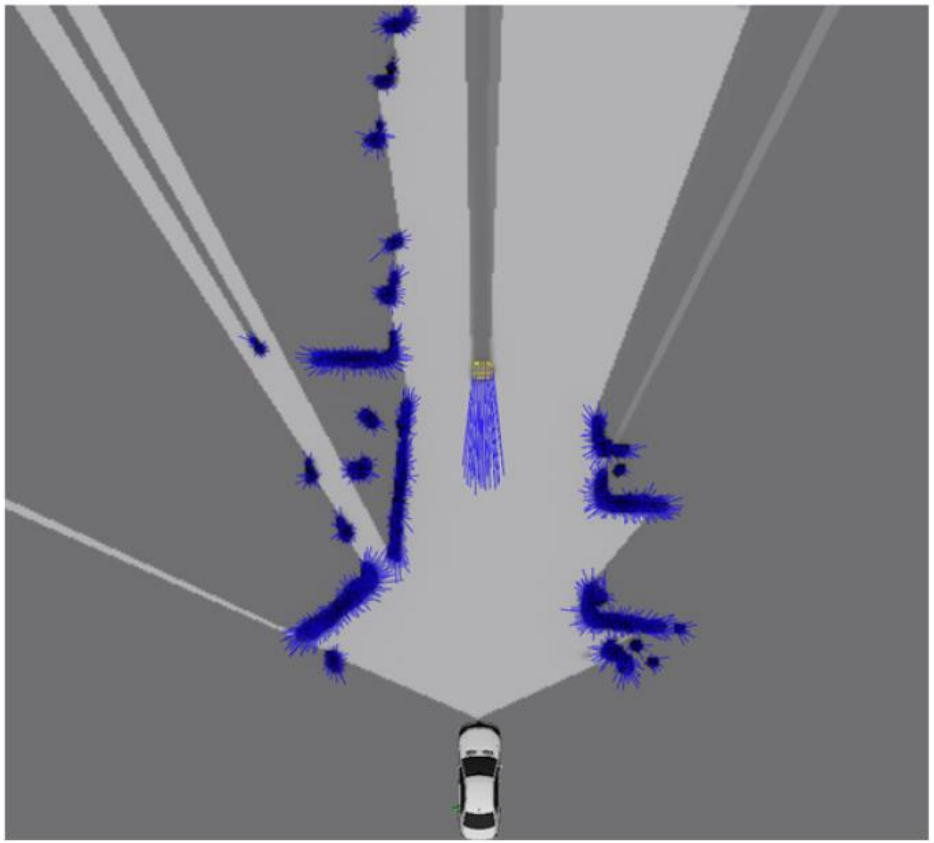
$p_{z_{k+1}}(O_{k+1}|z_{k+1}) = 0.05$

$p_{z_{k+1}}(O_{k+1}|z_{k+1}) = 0.5$

# 1) Dynamic Grid Map

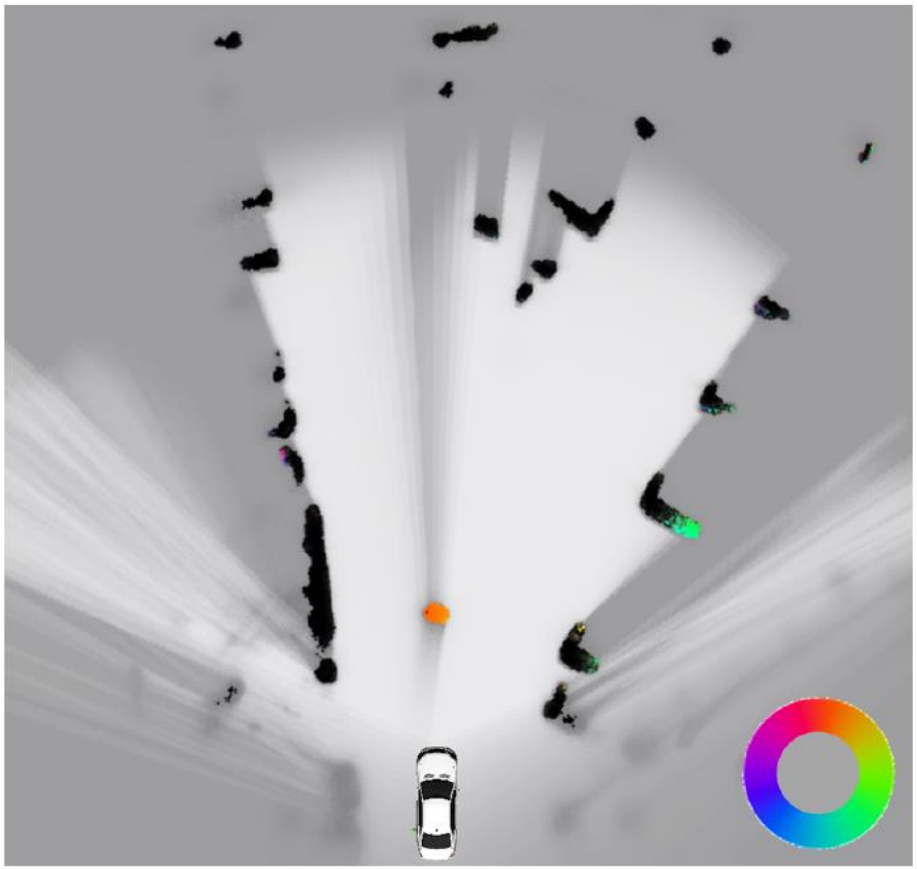
- Probability hypothesis density / multi-instance Bernoulli (PHD/MIB) filter is applied to estimate the dynamic state of grid cells.

Velocity estimation test scenario: A Segway approaches the test vehicle. The estimated mean velocity of every grid cell is visualized as a blue vector.



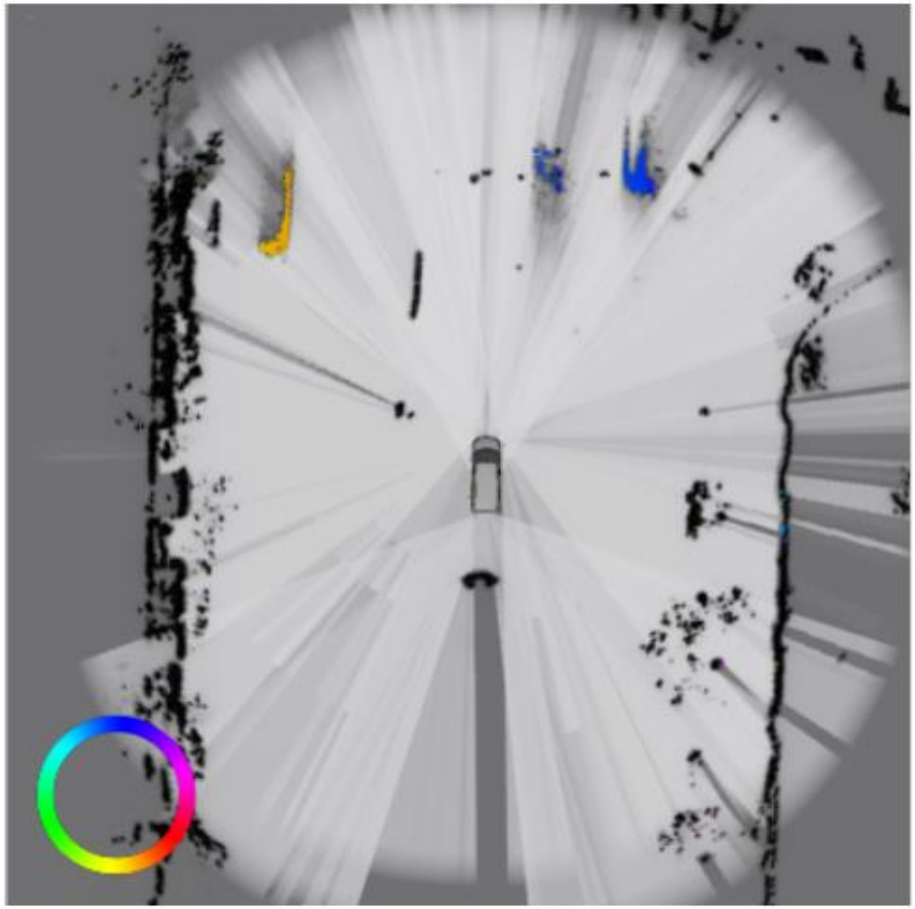
# 1) Dynamic Grid Map

- Probability hypothesis density / multi-instance Bernoulli (PHD/MIB) filter is applied to estimate the dynamic state of grid cells.

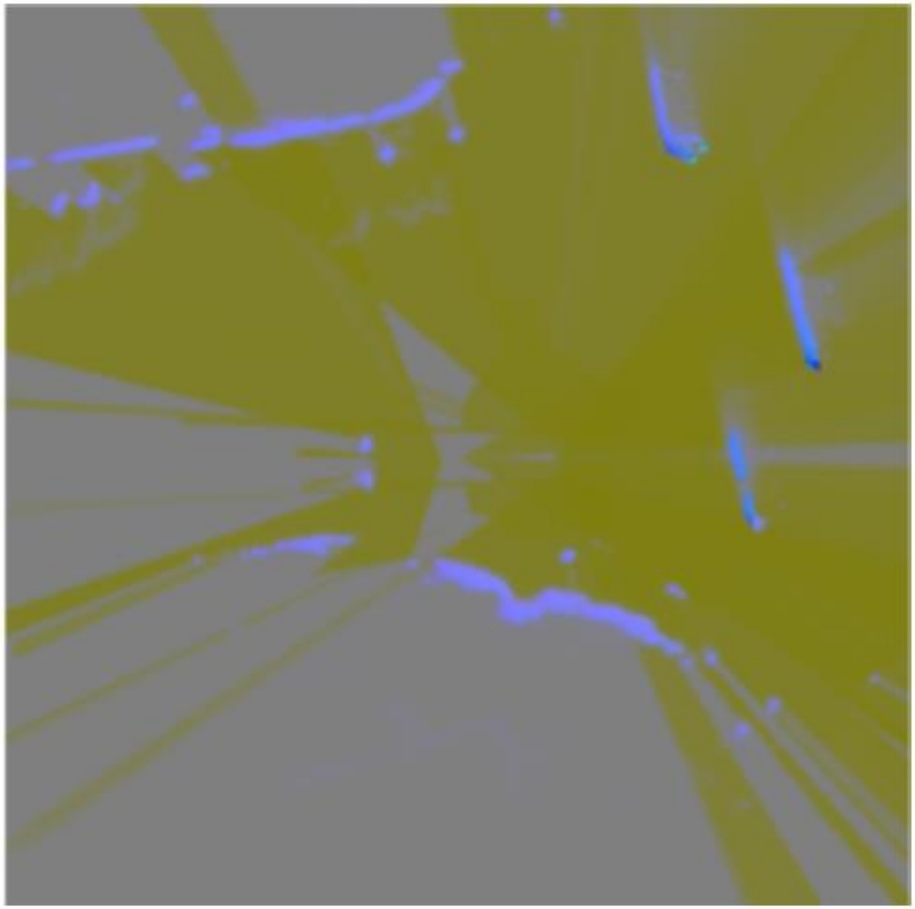


Test scenario for separation of static and dynamic grid cells. The color code represents the direction of movement, the color saturation is determined by the Mahalanobis distance between the velocity distribution and the velocity $v = 0$ in a grid cell.

## 1) Dynamic Grid Map

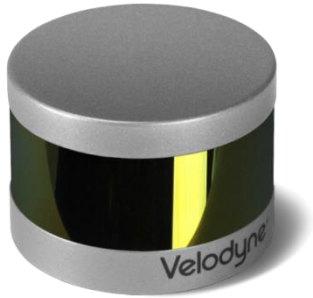


Occupancy of individual grid cells as gray scale value (white means 0% occupancy probability and black 100%). The colored cells visualize dynamic objects.



How the grid map values (occupancy and 2D velocity) are encoded for the network

# 2) Lidar Intensity Map



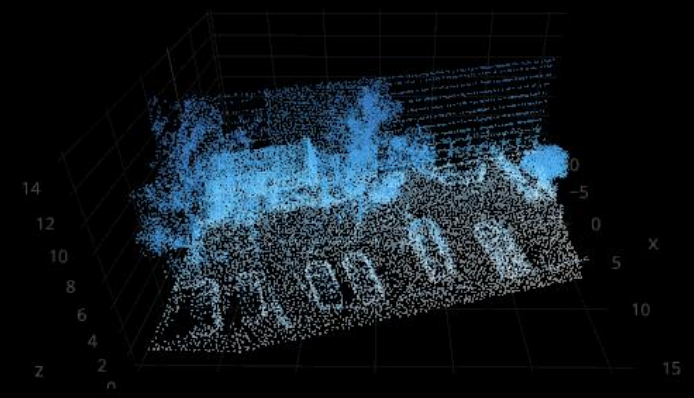

- Velodyne PUCK-16 laser scanner that provides an intensity value for each measured point
  - Strength of the material's reflection
- Only the reflective values of the ground plane are taken into account
- Grid map is obtained by integrating intensity values over time

# 2) Lidar Intensity Map

Velodyne LiDAR®

HOME PRODUCTS INDUSTRY DOWNLOADS PARTNERS MEDIA ABOUT CAREERS CONTACT US

## 3D - Real-Time - LiDAR



## Key Applications

AUTOMOTIVE

UAV

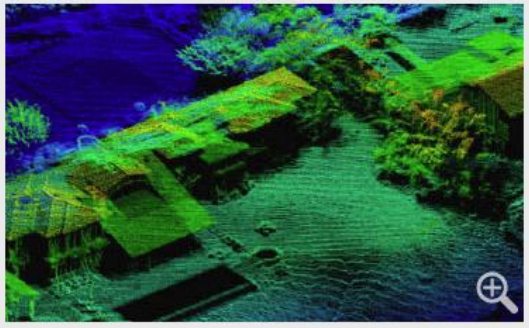
MAPPING

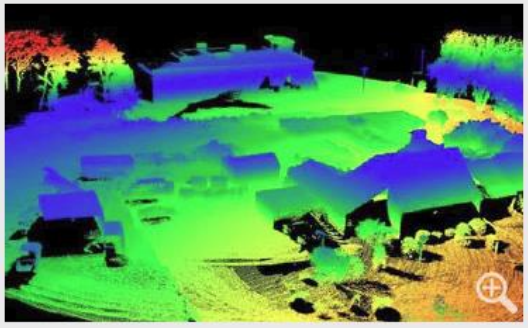
SECURITY

ROBOTICS

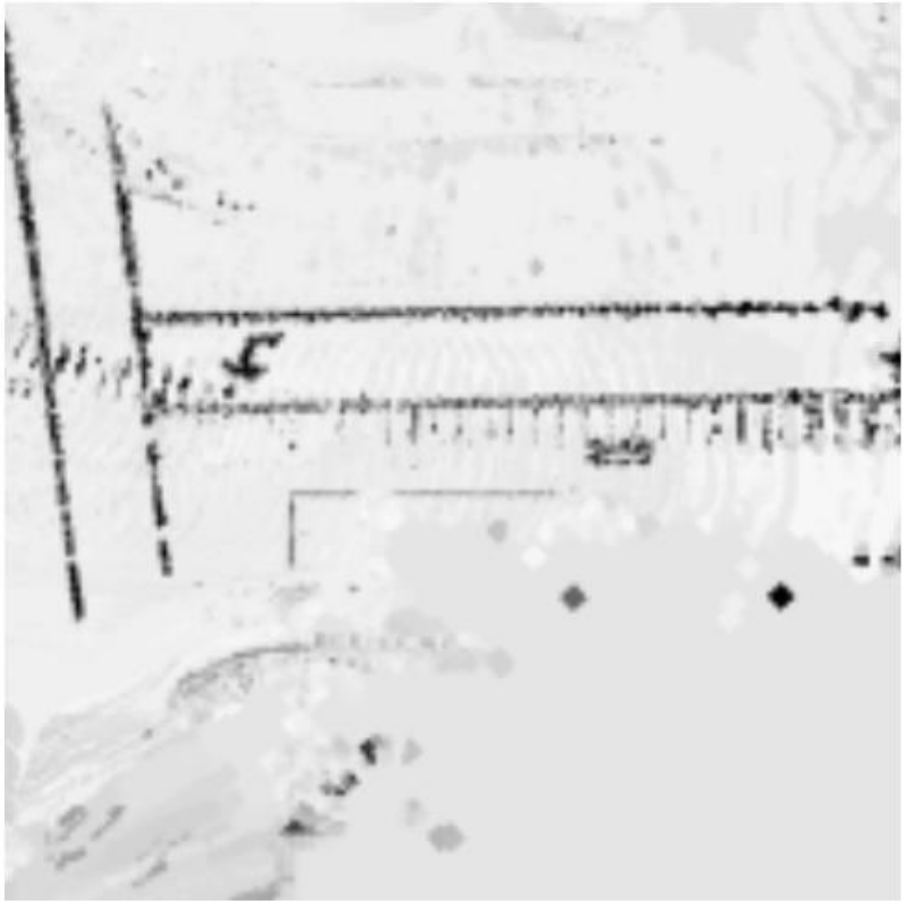
INDUSTRIAL

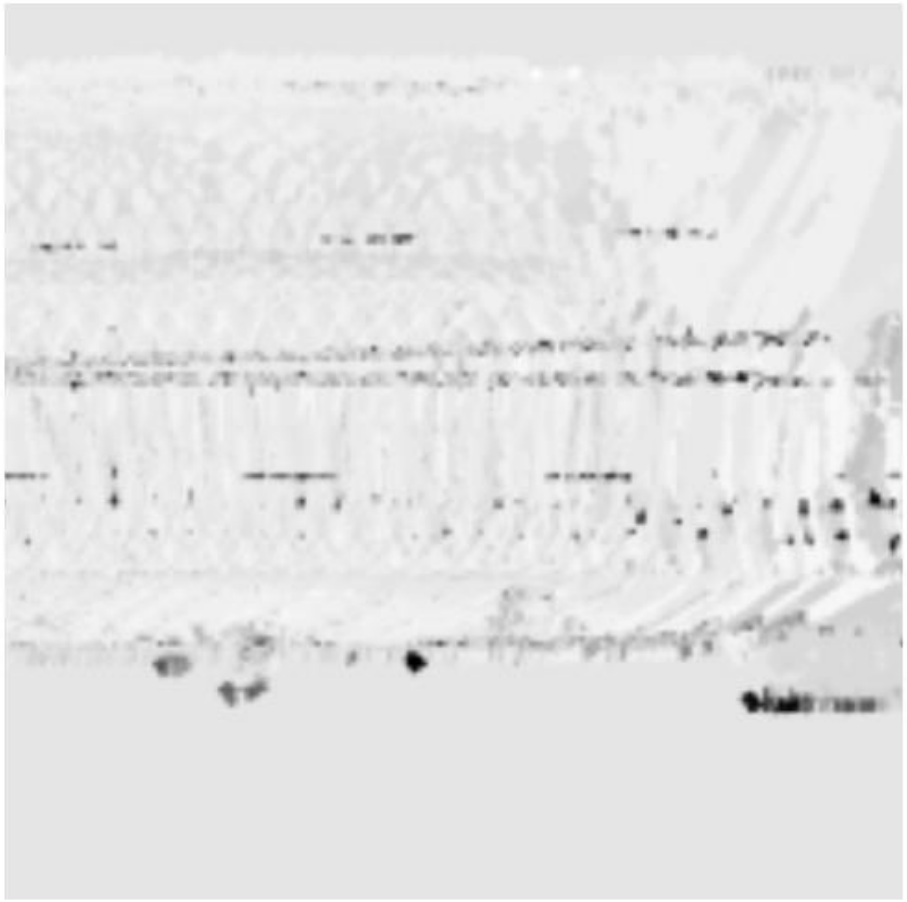
## USER DATA EXAMPLES





## 2) Lidar Intensity Map





The lane markings a represented by dark gray to black colors.

# 3) Semantic Map

- Based on the semantic stixels (Schneider et al., 2016)
  - Based on semantic scene labeling and stereo matching
  - Semantic scene labeling
    - Convolutional network
  - Stereo matching
    - Semi global matching
  - Joint stixel representation
    - Semantic class
    - Depth information
- The semantic map are formed by the semantic stixels are integrated over time
- Includes temporal information
- Stereo camera

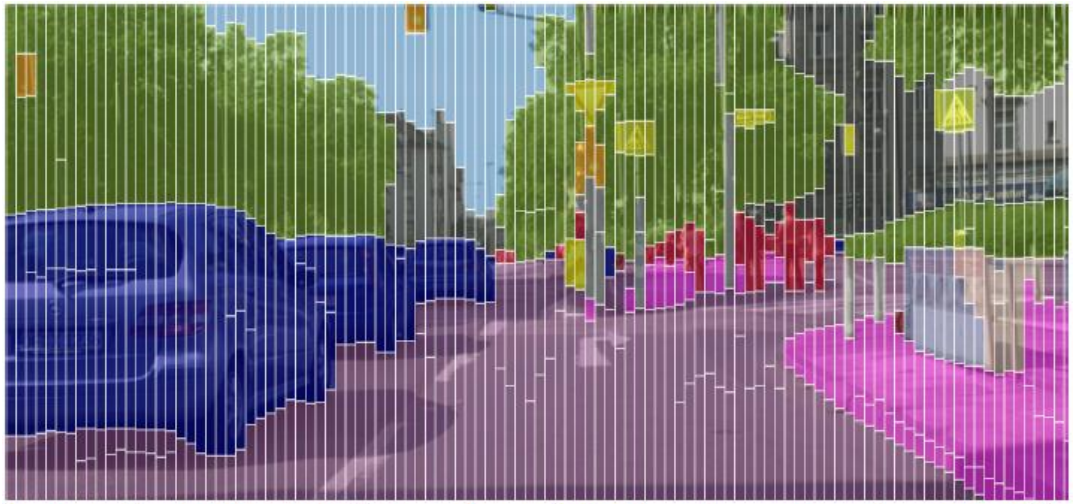
## 3) Semantic Map

Semantic and Depth in Stixel representation

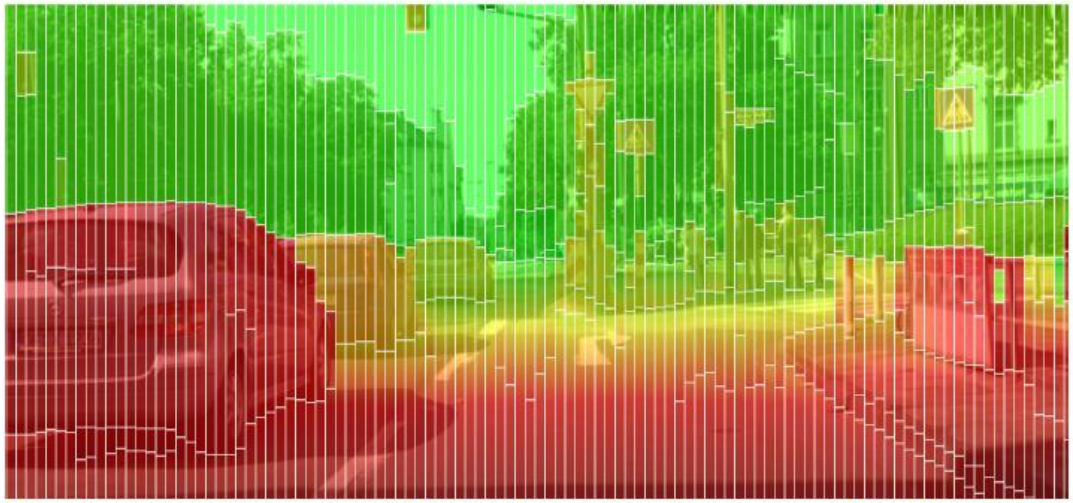
$s_i = (u; v_B; v_T; g; d; l)$

Narrow sticks with width *w*:

- column $u$,
- bottom and top coordinates $v_B$; $v_T$,
- geometric class $g$ (ground, object, sky),
- vertical displacement to the reference ground plane $d$, and
- semantic label $l$.

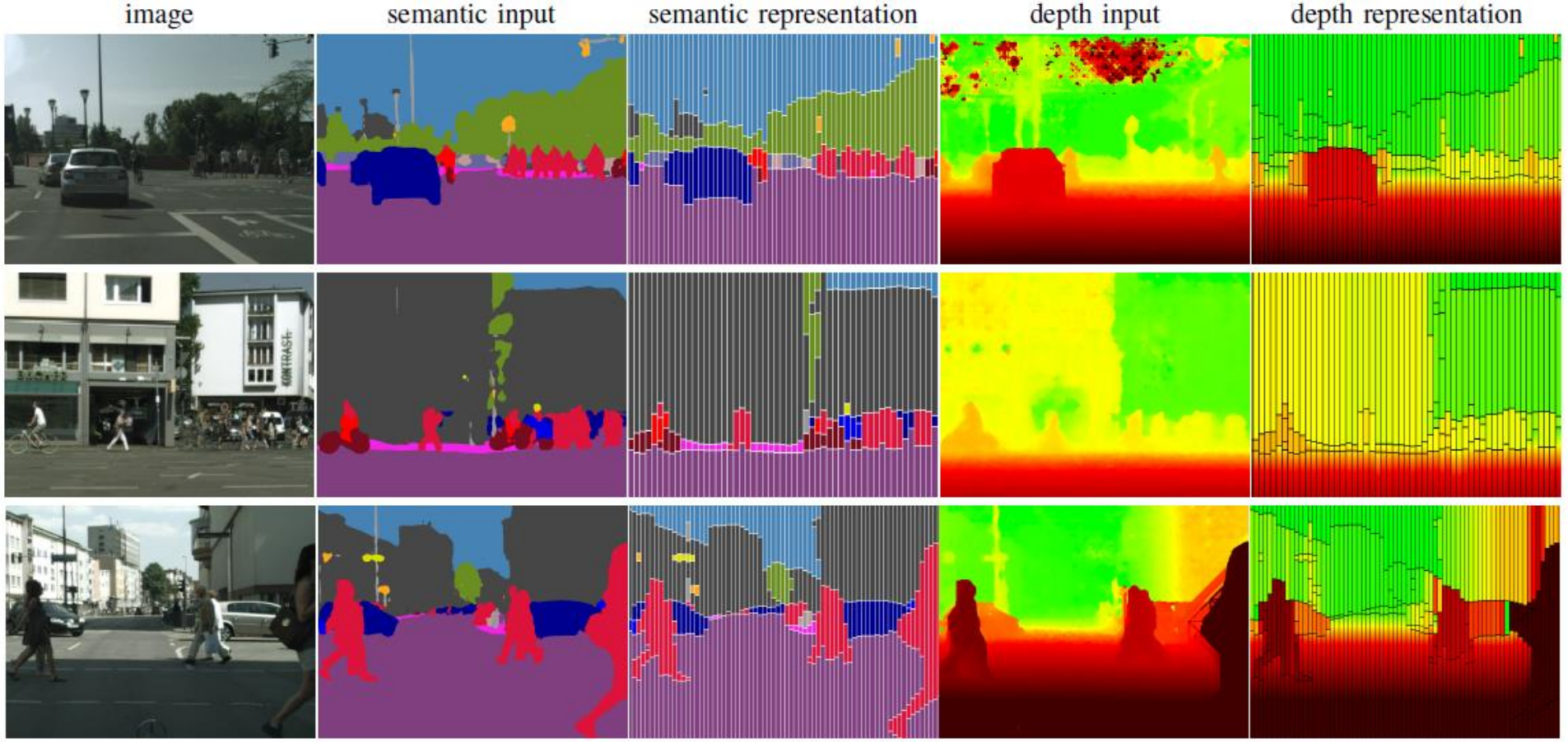


Semantic representation, where Stixel colors encode semantic classes following [10].



Depth representation, where Stixel colors encode disparities from close (red) to far (green).

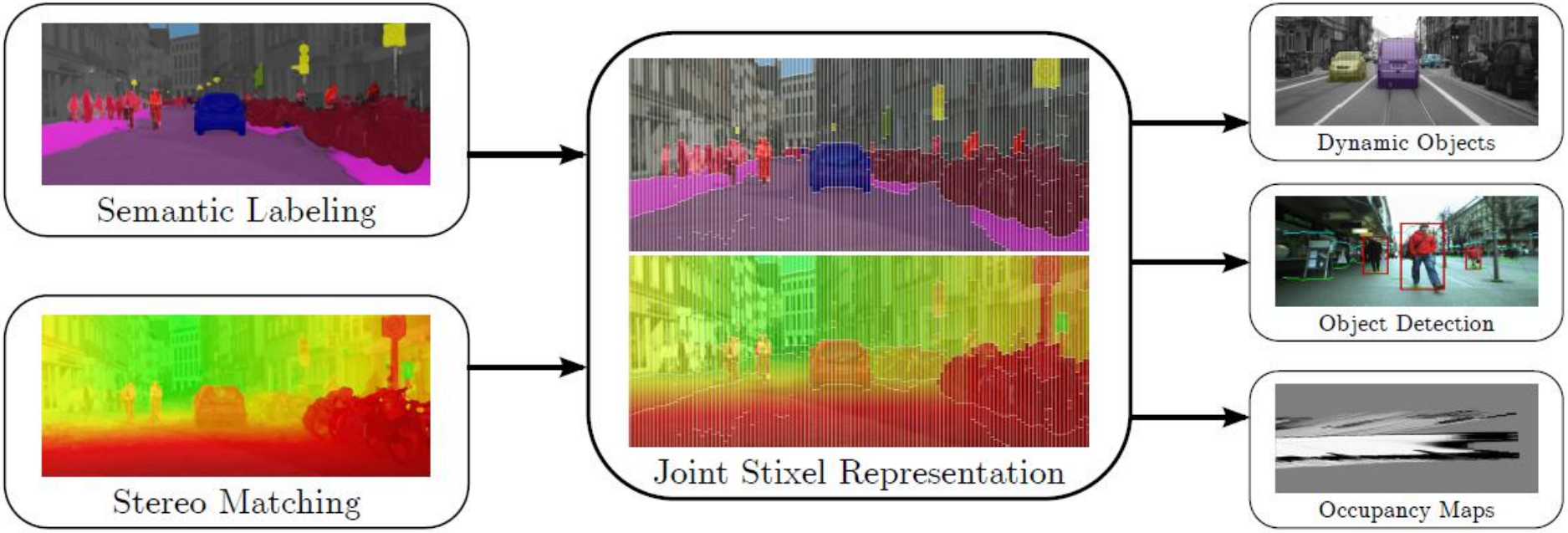
# 3) Semantic Map




L. Schneider, M. Cordts, T. Rehfeld, D. Pfeiffer, M. Enzweiler, U. Franke, M. Pollefeys, and S. Roth, “Semantic Stixels: Depth is not enough,” In IEEE Intelligent Vehicles Symposium (IV), pp. 110–117, 2016.

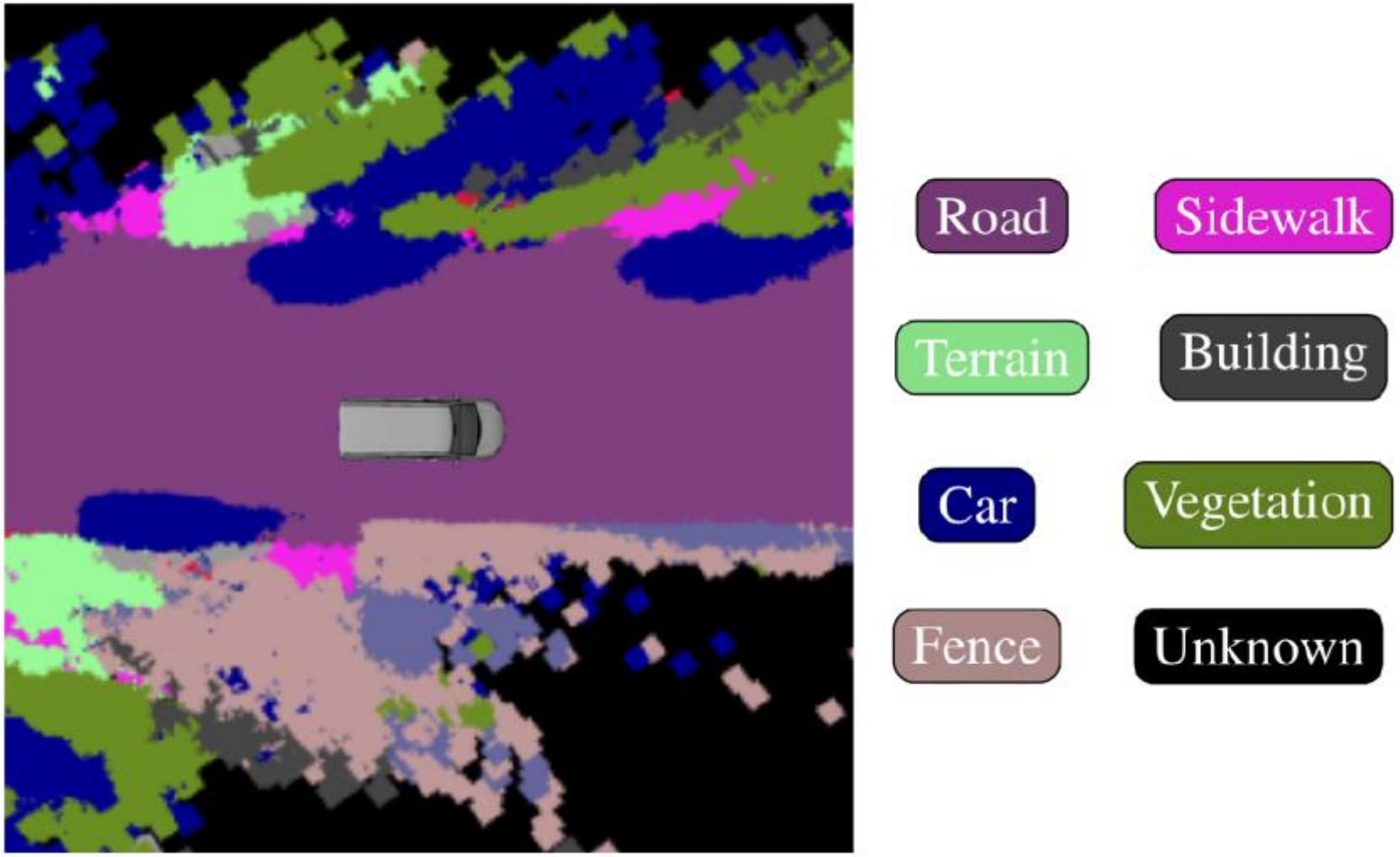
- Semantic Labeling: GoogLeNet as the underlying network architecture. A fully convolutional neural net (FCN)
- Stereo Matching: Semi Global Matching (SGM)

# 3) Semantic Map




Joint Stixel Representation is based on Semantic Labeling and Stereo Matching

## 3) Semantic Map

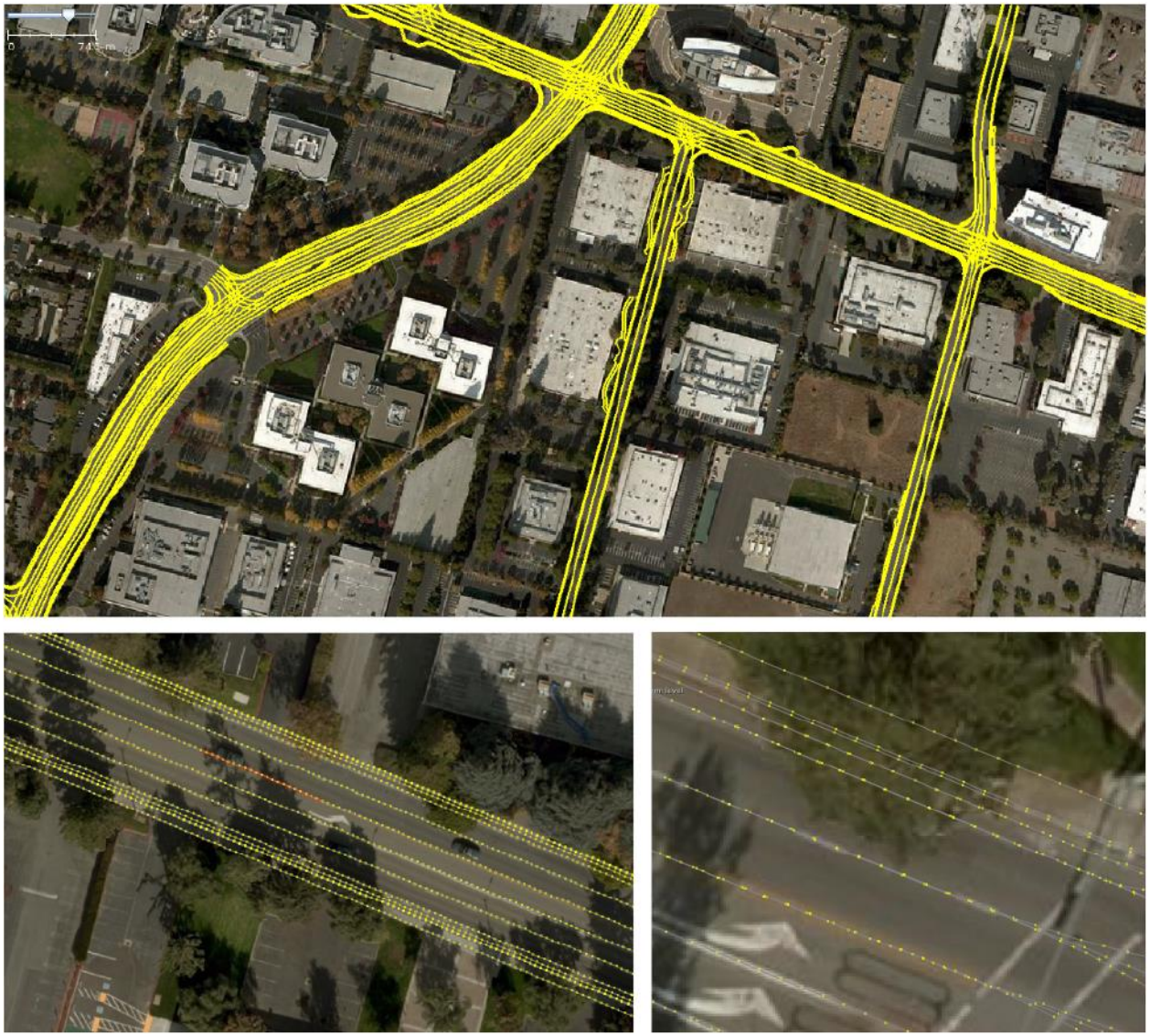



Stixel representation of objects and their corresponding classes.
Bird-eye view around the vehicle.

# B. Ground Truth Extraction

- High definition map of a limited area as source
- Represented by yellow points connected by lines
  - points consist of a position and a label
  - label yields information about the lane marking type, such as driving, roadside or border

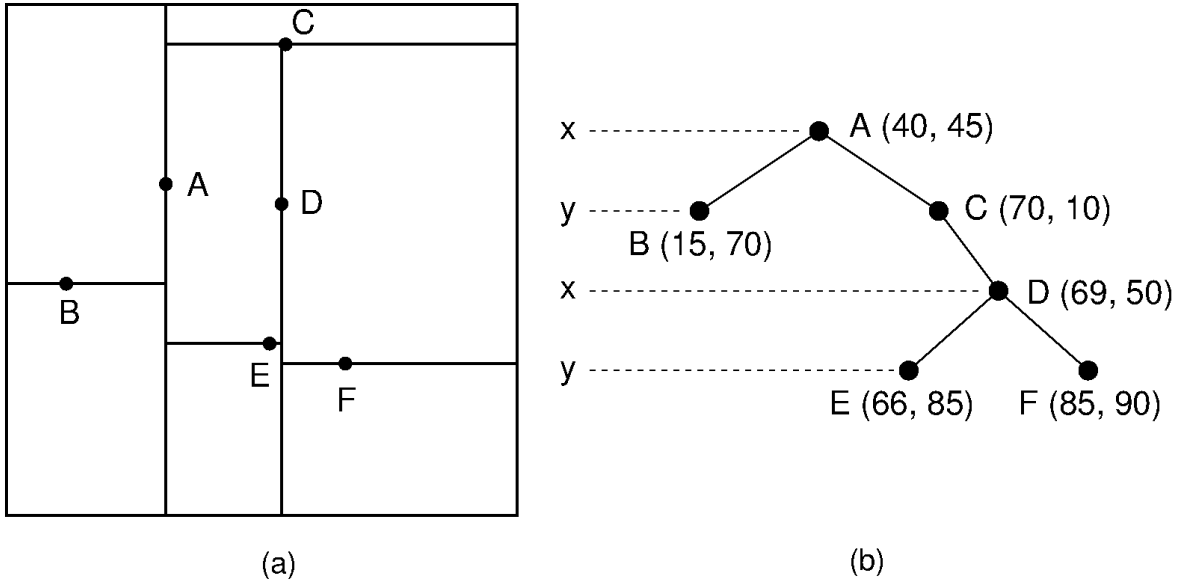
## B. Ground Truth Extraction



Different zoom levels of the high definition map (yellow points + lines) with the corresponding satellite images placed underneath

# B. Ground Truth Extraction

- Map horizon
  - Generated by parsing the entire map into a k-d tree
  - Allows to extract the labels for the tasks street type, number of lanes and the regression labels



# B. Ground Truth Extraction

- Map horizon
  - To align the map horizon with our grid map data, a precise localization of the vehicle is done by using an internal Mercedes-Benz localization method which is based on lidar grid maps and a feature based map matching method in combination with a global optimization technique.

# B. Ground Truth Extraction

- Label extraction
  - Road type
    - checking if the ego vehicle is inside a certain radius of an intersection
  - Number of lanes
    - counting the number of driving lane markings given by the map
  - Regression labels
    - simple trigonometry to calculate the distance between ego vehicle round boundaries and orientation with respect to the lanes

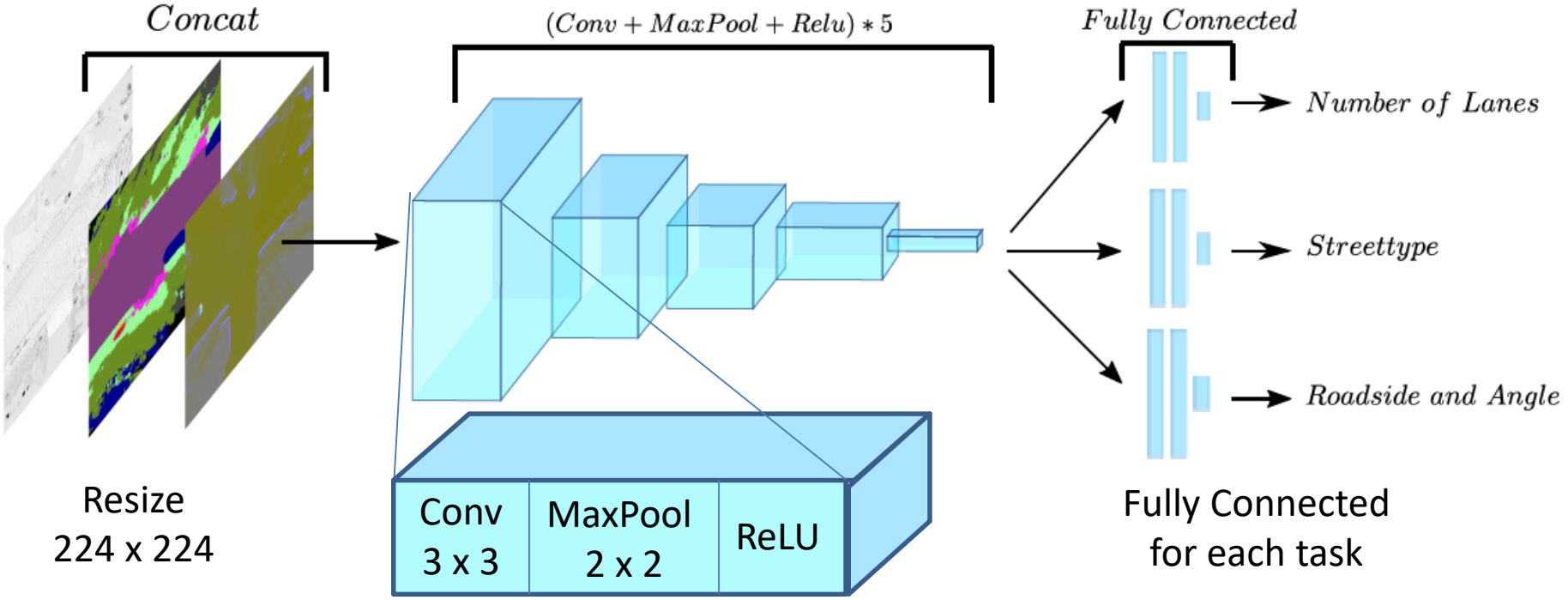
To ensure label correctness, recordings were removed parts where localization did not yield sufficiently good results*

# C. Network Architecture Design

- Influenced by successful CNN architectures such as (Krizhevsky et al., 2012), (Zeiler & Fergus, 2014), and (Simonyan et al., 2014).

- Each modality is scaled and cropped to a size of 224 x 224.
- For the filters in each convolutional layer, a kernel size of 3 x 3 is used.
- Max pooling with a size of 2 x 2 is performed and the output is passed through ReLU activation.
- Each task gets his own fully connected stack in the last part of the network.
- To be able to infer the different tasks by a single forward pass, a joint loss function, a combination of **Multi-class Cross-Entropy (MCE)** and **Least Squares Error (L2)** loss, is defined.

# C. Network Architecture Design




Designs:
- v1 = C1(36) - C2(36) - C3(48) - C4(64) - C5(64) - FC1abc(256) - FC2abc(256)
- v2 = C1(20) - C2(20) - C3(24) - C4(30) - C5(30) - FC1abc(96) - FC2abc(96)
- v1 3t = C1abc(12) - C2abc(12) - C3(48) - C4(64) - C5(64) - FC1abc(256) - FC2abc(256)

*T#(x)abc* — *T: C (convolutional layer), FC (fully connected layer) | #: layer position | x: number of filters | abc: three layers on the same level (for 3 outputs)*

Joint loss function, combining **Multi-class Cross-Entropy (MCE)** and **Least Squares Error (L2)**

$$L_{comb} = \gamma_1 MCE_{st-type} + \gamma_2 MCE_{\#lanes} + \gamma_3 L2$$

# IV. EVALUATION

- A. Data Set
- B. Experiments

*The results are not yet comparable to state-of-the-art

# A. Data Set

- Around one hour of driving data (36,000 frames) around Sunnyvale, CA
  - frames with insufficient localization discarded, resulting in 22,000 frames

- Number of lanes: 2 to 6 lanes
- Street type: intersection (25%) to straight road (75%)
- Distance vehicle - left roadside : 4-20 meters
- Distance vehicle - right roadside: 0-12 meters
- The vehicle's heading relative to the lane: -10° and 10°

# A. Data Set

- Labels are normalized to [0; 1]
- Training, validation and test set splits: 70%, 20% and 10%
- Training data augmented by applying random rotation and translations

# B. Experiments

- Weight initialization:
  - biases to 0
  - weights $W_{ij}$ at each layer with the following commonly used heuristic:

$$W_{ij} \sim U\left[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right]$$

    - $U[-a, a]$ is the uniform distribution in interval $(-a, a)$
    - $n$ is the size of the previous layer (the number of columns of $W$).
- Weight updates:
  - Adam optimizer

# B. Experiments

- Dropout:
  - During training after each fully connected layer

# B. Experiments

- Hyper-parameters
  - $\gamma_{1\text{-}3}$
  - dropout,
  - learning rate
- Randomized hyper-parameter optimization on the validation set

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. J. Machine Learning Res., 13, 281–305. 436, 437 ????

Define a marginal distribution for each hyperparameter, e.g.

- a Bernoulli or multinoulli for binary or discrete hyperparameters
- a uniform distribution on a log-scale for positive real-valued hyperparameters

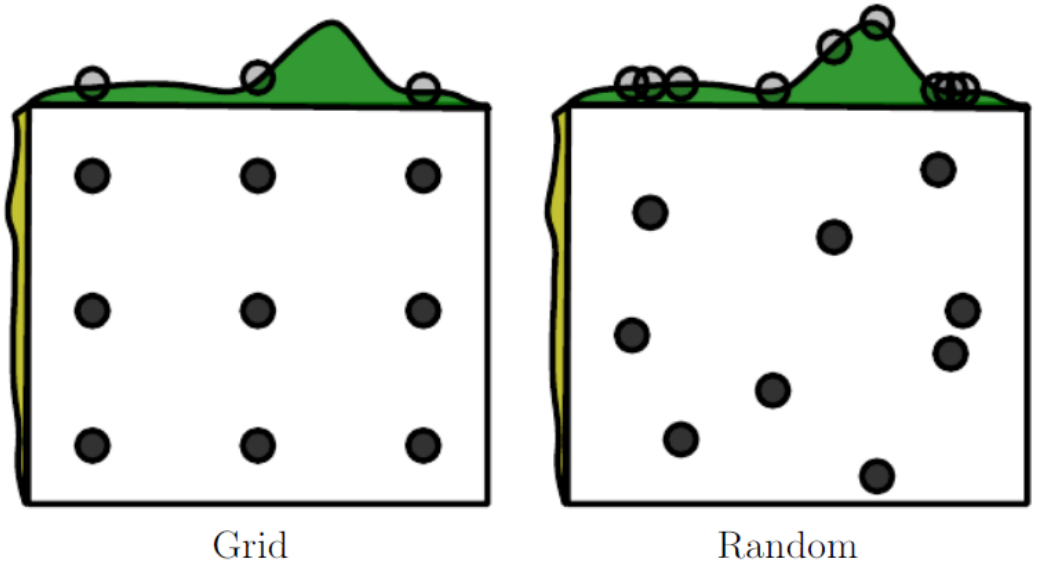
```
log_learning_rate ~ u(-1,-5)
```

```
log_number_of_hidden_units ~ u(log(50),log(2000))
```

# B. Experiments

- Randomized hyper-parameter optimization on the validation set

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. J. Machine Learning Res., 13, 281–305. 436, 437 ????




Define a marginal distribution for each hyperparameter, e.g.

- a Bernoulli or multinoulli for binary or discrete hyperparameters
- a uniform distribution on a log-scale for positive real-valued hyperparameters

**`log_learning_rate ~ u(-1,-5)`** `learning_rate = 10^log_learning_rate`

**`log_number_of_hidden_units ~ u(log(50),log(2000))`**

# B. Experiments

- Classification tasks

$$Acc = \frac{\#\text{correct classified samples}}{\#\text{all samples}}$$

- Regression tasks

$$Err = \frac{1}{N}\sum_{i=1}^{N}|p_i - l_i|$$

# B. Experiments

TABLE I

THE RESULTS FOR EACH NETWORK ARCHITECTURE AND THE BASELINE ON THE TEST SET WITH THE EPOCH SELECTED ON THE VALIDATION SET.

| Architecture | v1 | v1_3t | v2 | baseline |
|---|---|---|---|---|
| Street Type | 0.91 | 0.90 | 0.91 | - |
| Number of Lanes | 0.58 | 0.58 | 0.59 | - |
| Distance Left | 0.97 | 0.94 | 0.85 | 1.19 |
| Distance Right | 0.94 | 0.91 | 1.01 | 1.29 |
| Driving Angle | 1.68 | 1.60 | 1.54 | 2.47 |

All presented architectures clearly outperform the base line.

# B. Experiments

TABLE II

ONLY THE INPUTS (ABBREVIATED BY FIRST LETTER) NAMED IN THE TOP ROW ARE USED FOR INFERENCE OF THE CNN. THE RESULTS SHOW THE PERFORMANCE ON THE TEST SET WITH CNN V2.

| Input combination | I | S | D | I + S | I + D | S + D |
|---|---|---|---|---|---|---|
| Street Type | 0.68 | 0.31 | 0.31 | 0.89 | 0.73 | 0.31 |
| Number of Lanes | 0.29 | 0.31 | 0.31 | 0.57 | 0.36 | 0.31 |
| Distance Left | 2.18 | 2.52 | 3.47 | 1.04 | 1.95 | 2.45 |
| Distance Right | 1.21 | 2.61 | 4.39 | 1.02 | 1.40 | 3.68 |
| Driving Angle | 2.27 | 2.98 | 2.94 | 1.86 | 2.13 | 0.85 |

Intensity input by itself is able to reduce the error the most, followed by the semantic map and then the dynamic map.

The combination of intensity and semantic map yields the best performance.

# V. CONCLUSION

- Central idea is to use existing map data to bootstrap the approach with minimal human supervision
- Promising results
- Precision is currently not high enough to use it in a production system

# V. CONCLUSION

- Any localization algorithm we build around our network output cannot be more precise than the initial localization method used to generate training data
- A learned localization method can potentially generalize better and be more robust

# REFERENCES


- Bittel, Sebastian & Rehfeld, Timo & Weber, Michael & Zöllner, J. (2017). Estimating High Definition Map Parameters with Convolutional Neural Networks. 2017 IEEE International Conference on Systems, Man and Cybernetics, At Banff.
- D. Nuss, S. Reuter, M. Thom, T. Yuan, G. Krehl, M. Maile, A. Gern, and K. Dietmayer, “A Random Finite Set Approach for Dynamic Occupancy Grid Maps with Real-Time Application,” ArXiv e-prints, 2016.
- L. Schneider, M. Cordts, T. Rehfeld, D. Pfeiffer, M. Enzweiler, U. Franke, M. Pollefeys, and S. Roth, “Semantic Stixels: Depth is not enough,” In IEEE Intelligent Vehicles Symposium (IV), pp. 110–117, 2016.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” In Conference on Neural Information Processing Systems (NIPS), pp. 1097–1105, 2012.
- M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” In European Conference on Computer Vision (ECCV), pp. 818–833, 2014.
- K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps,” In International Conference on Learning Representations (ICLR), 2014.