Efficient Road Lane Marking Detection with Deep Learning

Ping-Rong Chen*, **Shao-Yuan Lo***, Hsueh-Ming Hang, Sheng-Wei Chan, Jing-Jhih Lin

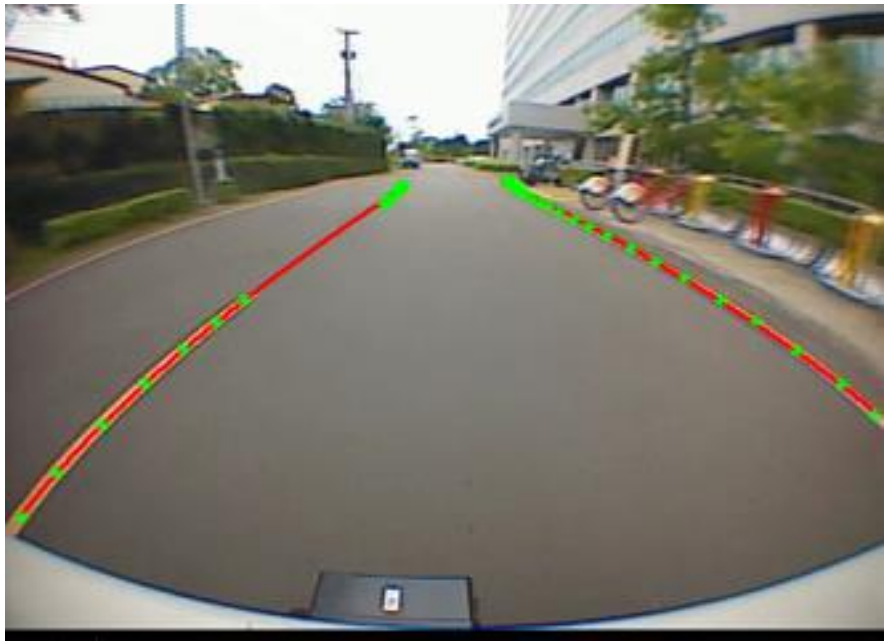
National Chiao Tung University

DSP 2018 Oral

November 21, 2018

Introduction

Lane
Marking
Detector

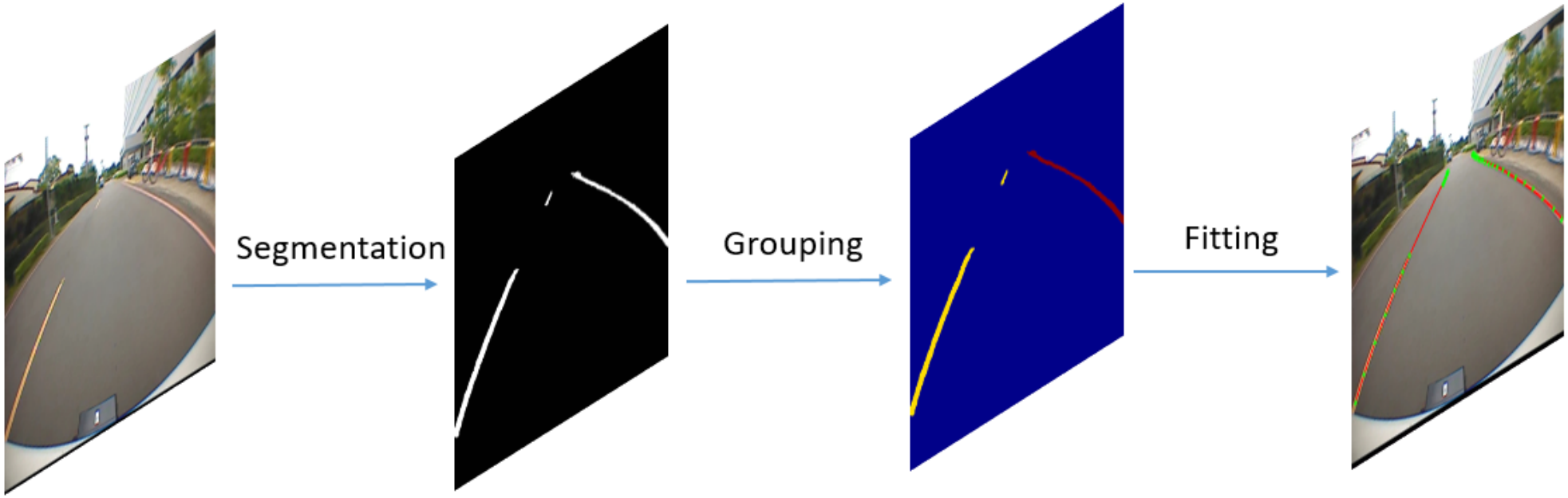


Input: RGB image

Output: Lane Marking

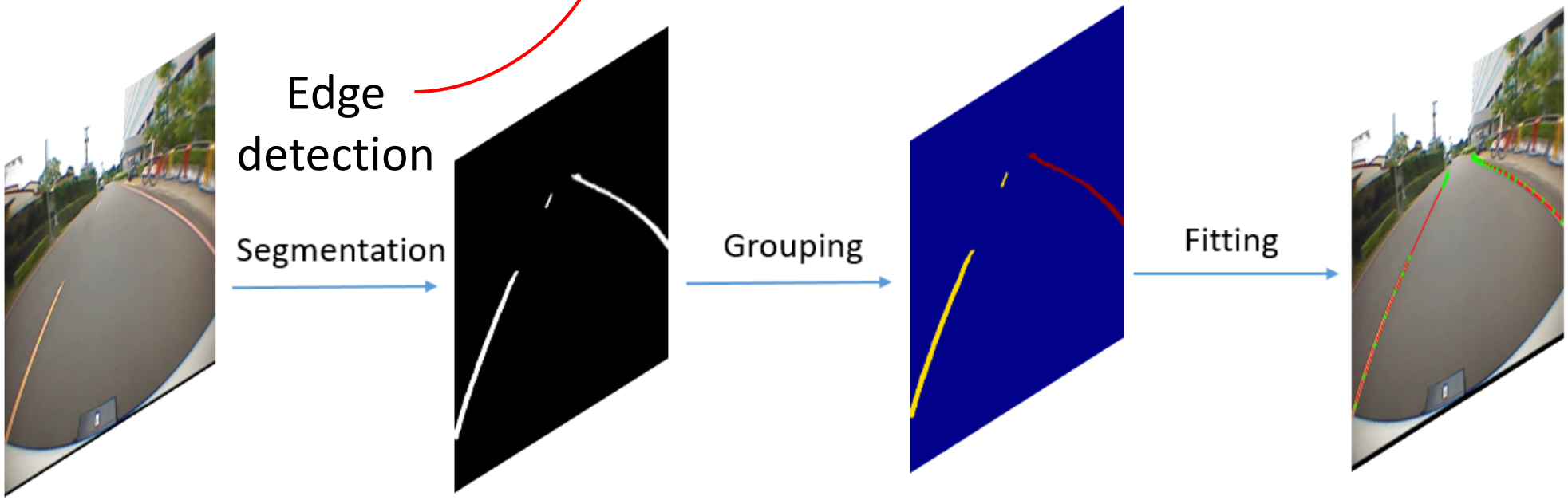
Introduction

- **Step 1:** Segmentation
- **Step 2:** Grouping
- **Step 3:** Fitting



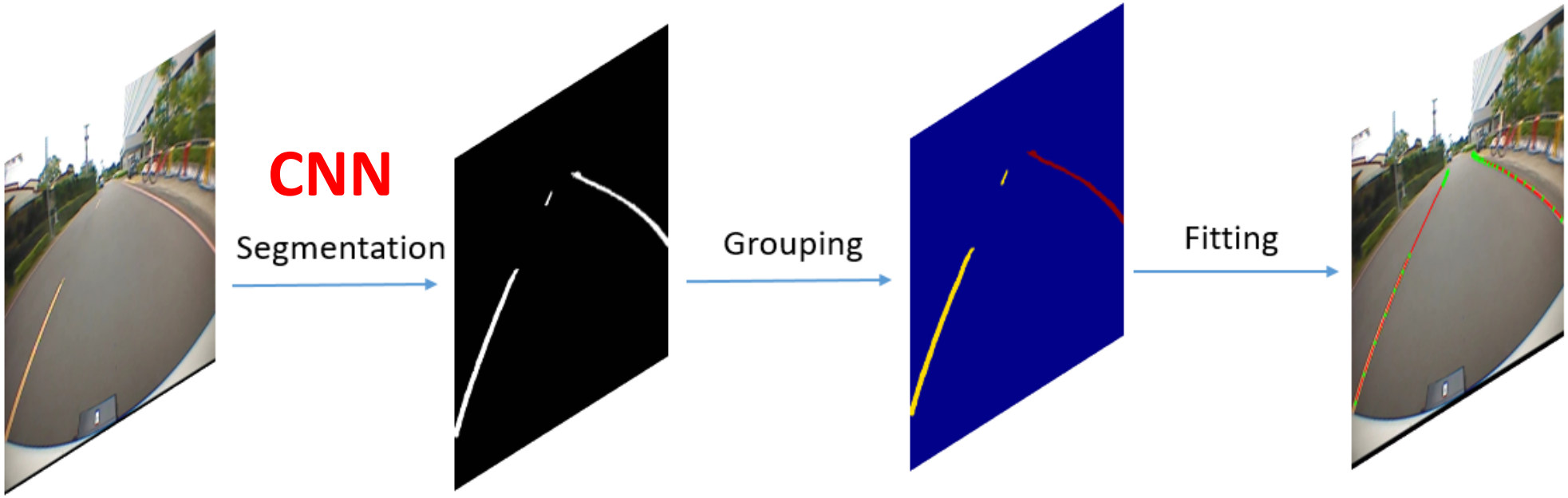
Introduction

- **Step 1:** Segmentation
- **Step 2:** Grouping
- **Step 3:** Fitting

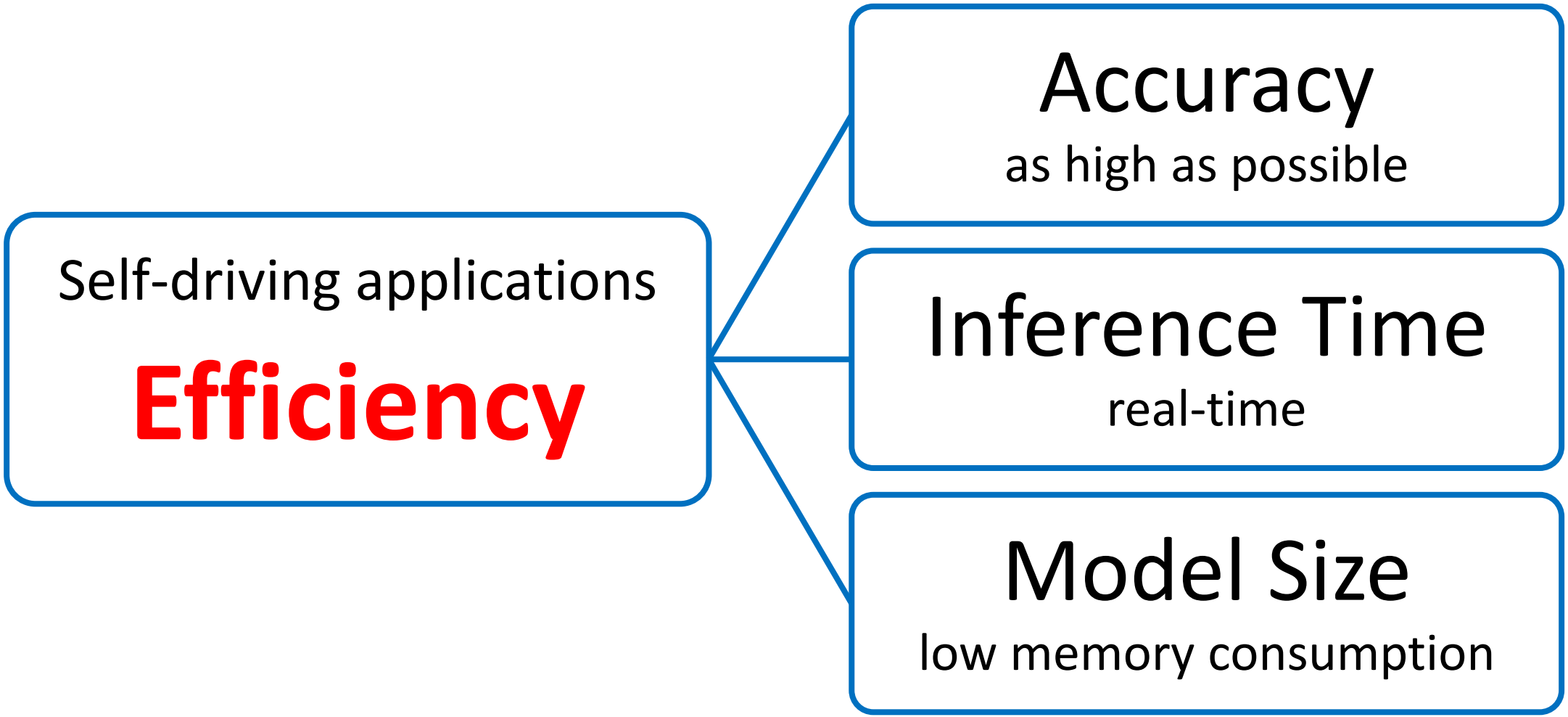


Introduction

- **Step 1:** Segmentation (by deep learning)
- **Step 2:** Grouping (by post-processing scheme)
- **Step 3:** Fitting (by post-processing scheme)

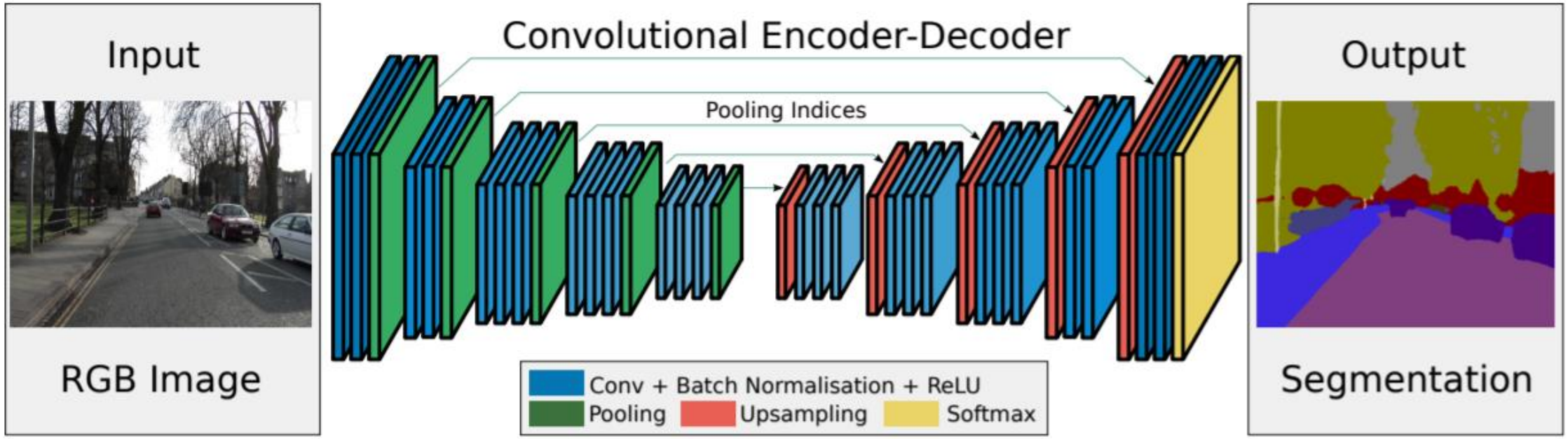


Deep Learning Method



SegNet

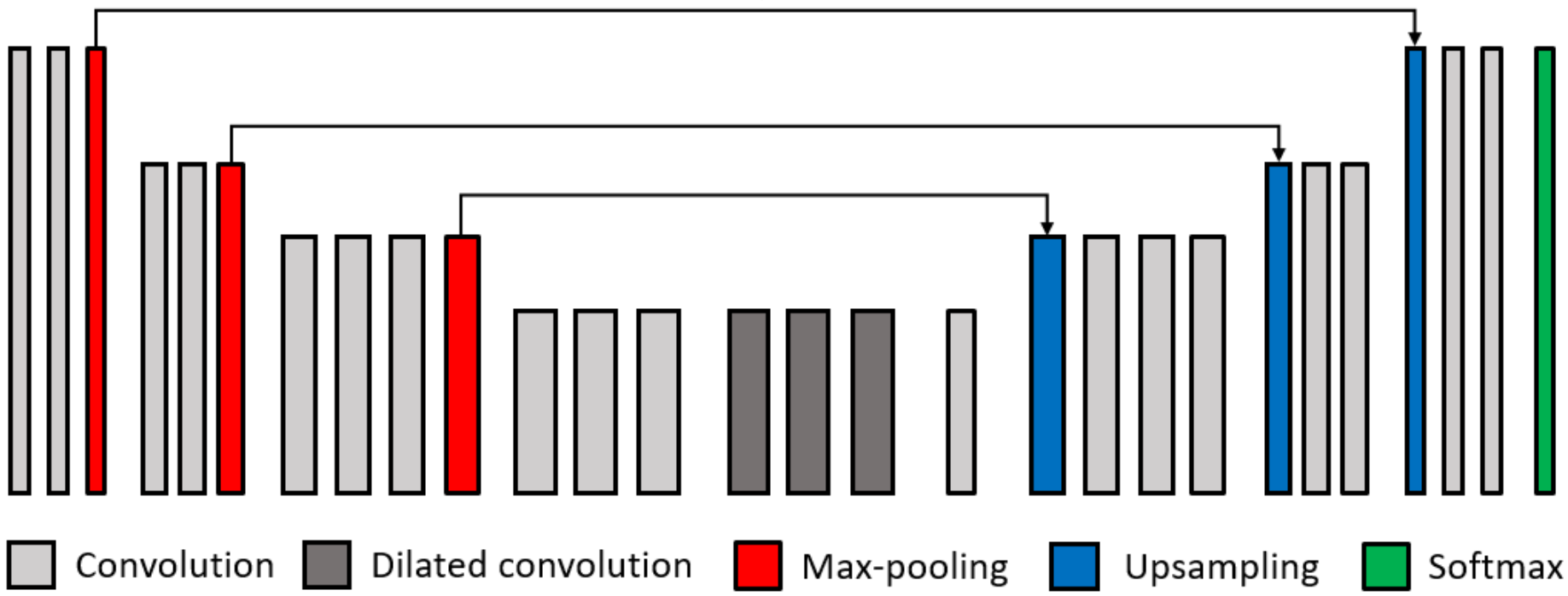
- **Encoder:** Based on a VGG16.
- **Decoder:** Symmetric to the encoder.



[Simonyan and Zisserman], [Badrinarayanan et al.]

LMD

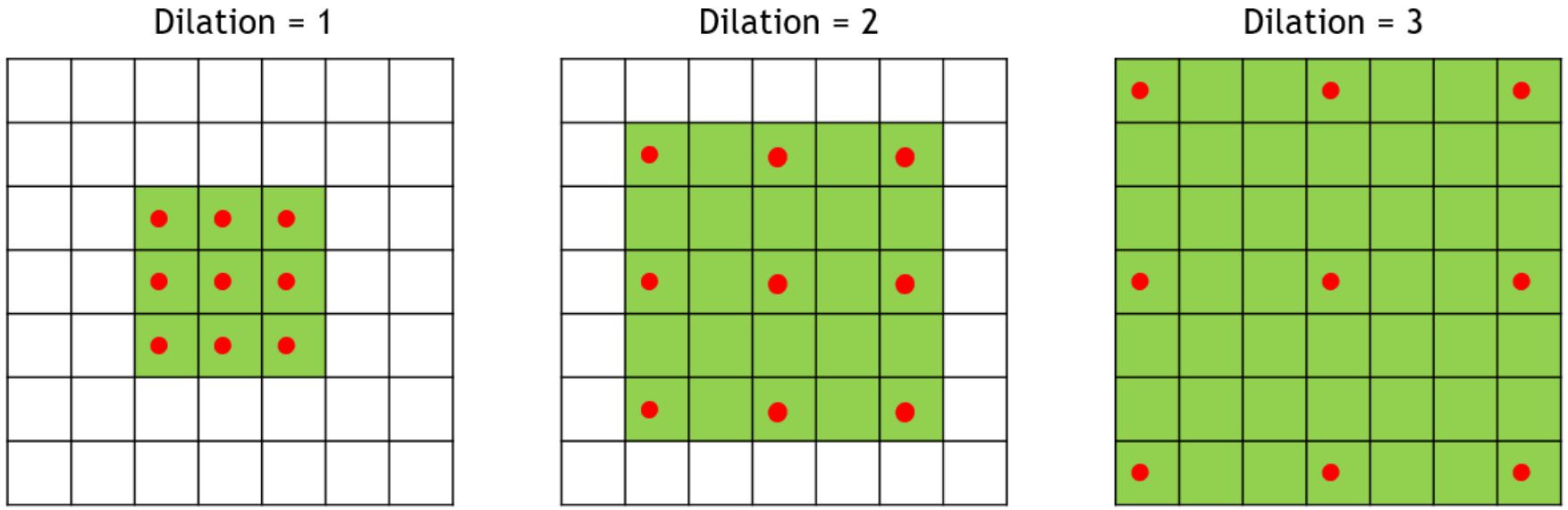
- **LMD:** Lane Marking Detector, the proposed convolutional network for semantic segmentation.



LMD

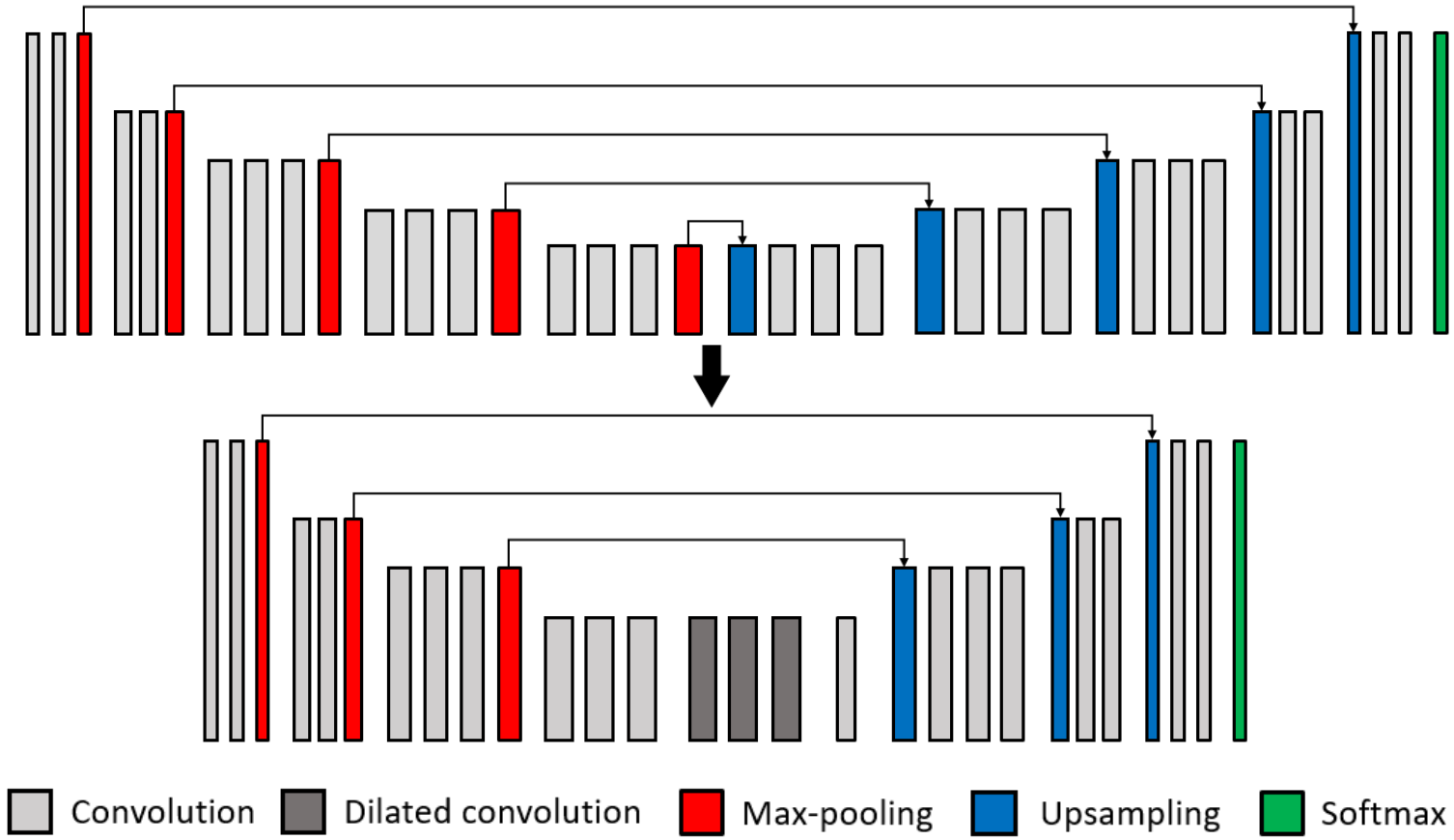
- **Dilated convolution:** Insert zeros between two consecutive kernel values along each dimension.
- The effective size of a *n*x*n* convolution with dilation rate *r*:

$$[r(n\text{-}1)+1]\times[r(n\text{-}1)+1]$$



LMD

- Remove the last two downsampling layers to get larger feature maps.
- Apply the dilated convolution to keep the receptive fields without the increases of the number of parameters and computational cost.
- Simplify the decoder structure.



Class Balancing

- Median frequency balancing：

$$w_c = \frac{median\ freq}{freq(c)}$$

*freq(c): the number of pixels of class *c* divided by the number of total pixels in images

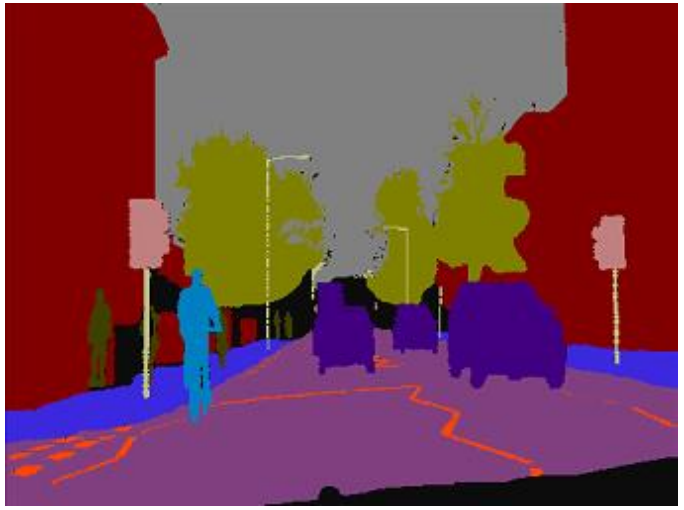
*median freq: the median of these frequencies

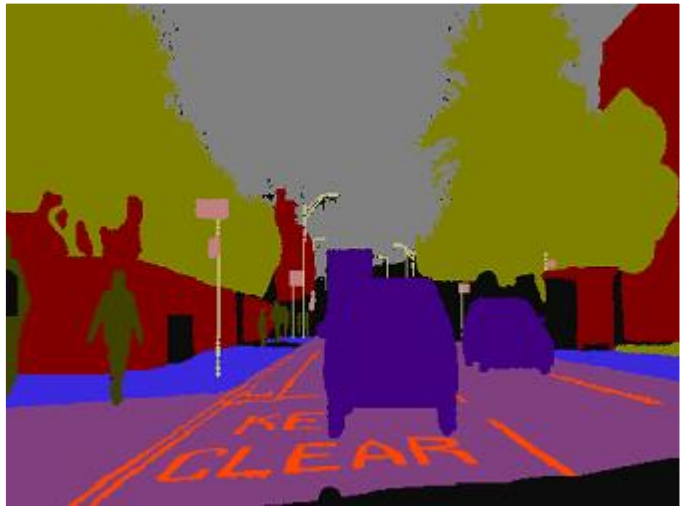
- Goal：$\min \sum w_c \times (loss\ function)_c$

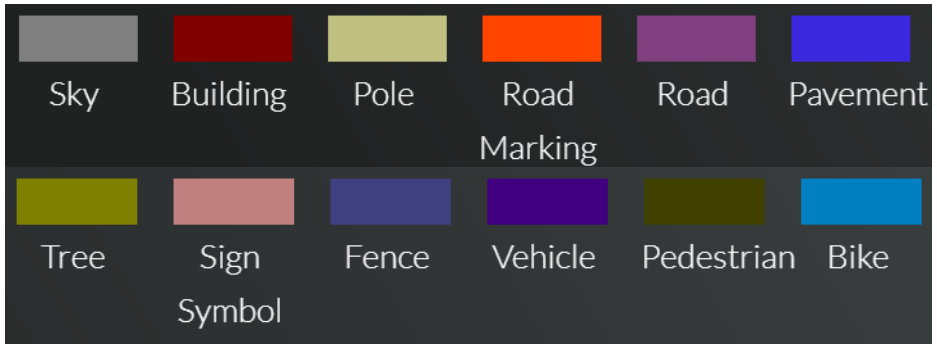
CamVid Dataset

- Classes: 12
- Training data: 367
- Test data: 233
- Resolution: 360 x 480

[Brostow et al.]



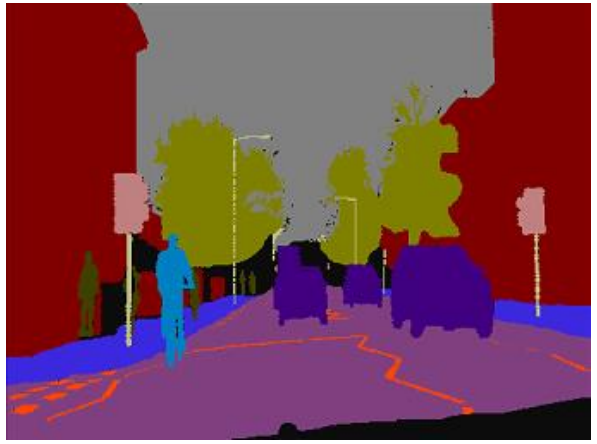




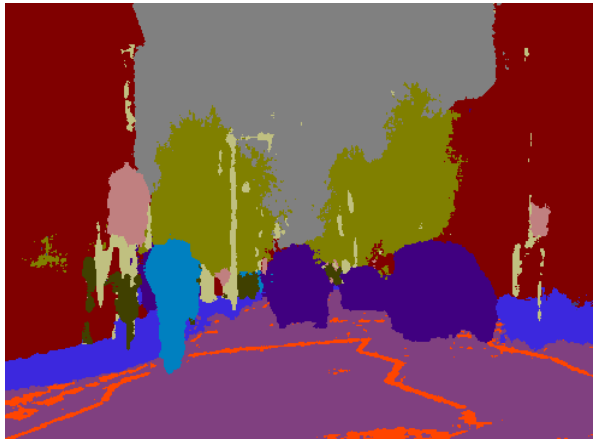
Class Balancing Experiment

Input

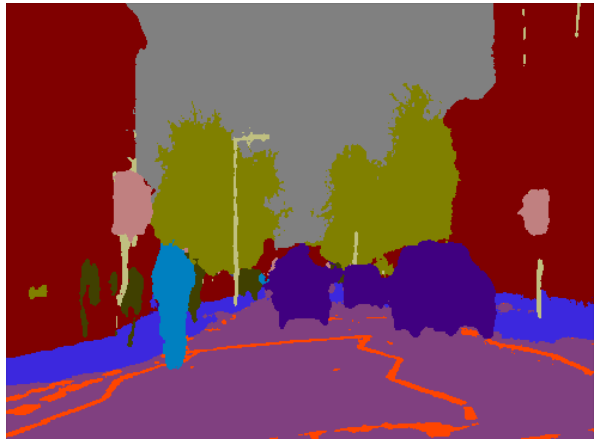
Ground truth



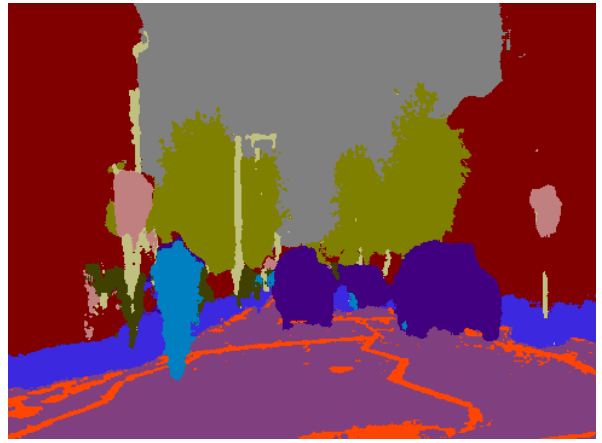
Balanced x 0.6



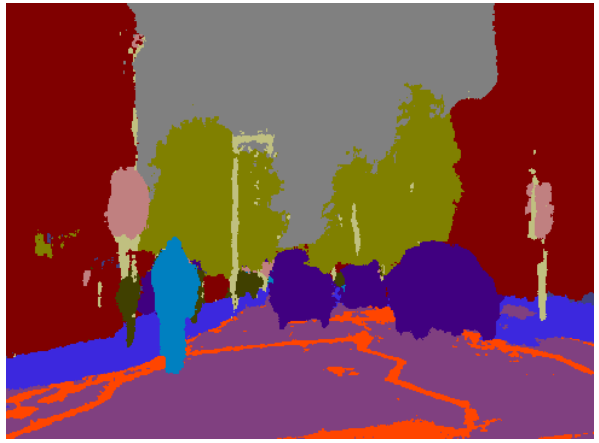
Balanced



Balanced x 2



Balanced x 5



Class Balancing Experiment

The class weight of lane	Class accuracy	IoU
Balanced x 0.6	80.7	53.6
Balanced	83.9	52.6
Balanced x 2	82.9	51.3
Balanced x 5	88.7	47.3

IoU: intersection over union

Comparison

- Resolution: 360 x 480
- GPU: GTX 1080
- Framework: Caffe

Network	Frame rate (fps)	Model size (MB)
SegNet [Badrinarayanan et al.]	28.1	117
LMD (ours)	34.4	66
	+22.4%	save 43.6%

Comparison

Network	Buil.	Tree	Sky	Car	Sign	Road	Pede.	Fenc.	Pole	Side.	Bike	Lane	Class avg.	mIoU
SegNet [1]	88.8	**87.3**	92.4	82.1	50.5	**97.2**	57.1	49.3	27.5	84.4	30.7	-	65.2	55.6
ENet [2]	74.7	77.8	**95.1**	82.4	51.0	95.1	67.2	51.7	35.4	86.7	34.1	-	68.3	51.3
LMD	**89.2**	86.4	93.7	**83.8**	**58.1**	95.4	**79.3**	**52.7**	**48.6**	**90.5**	**61.6**	-	**76.3**	**63.5**
LMD-12	88.1	86.8	94.0	84.3	55.4	90.1	80.1	51.9	48.4	92.3	64.7	83.9	76.7	62.2
LMD-12*	89.3	87.9	94.1	87.0	63.7	91.2	86.0	55.2	54.8	93.9	67.0	85.4	79.6	65.2

* with ImageNet pretrained model

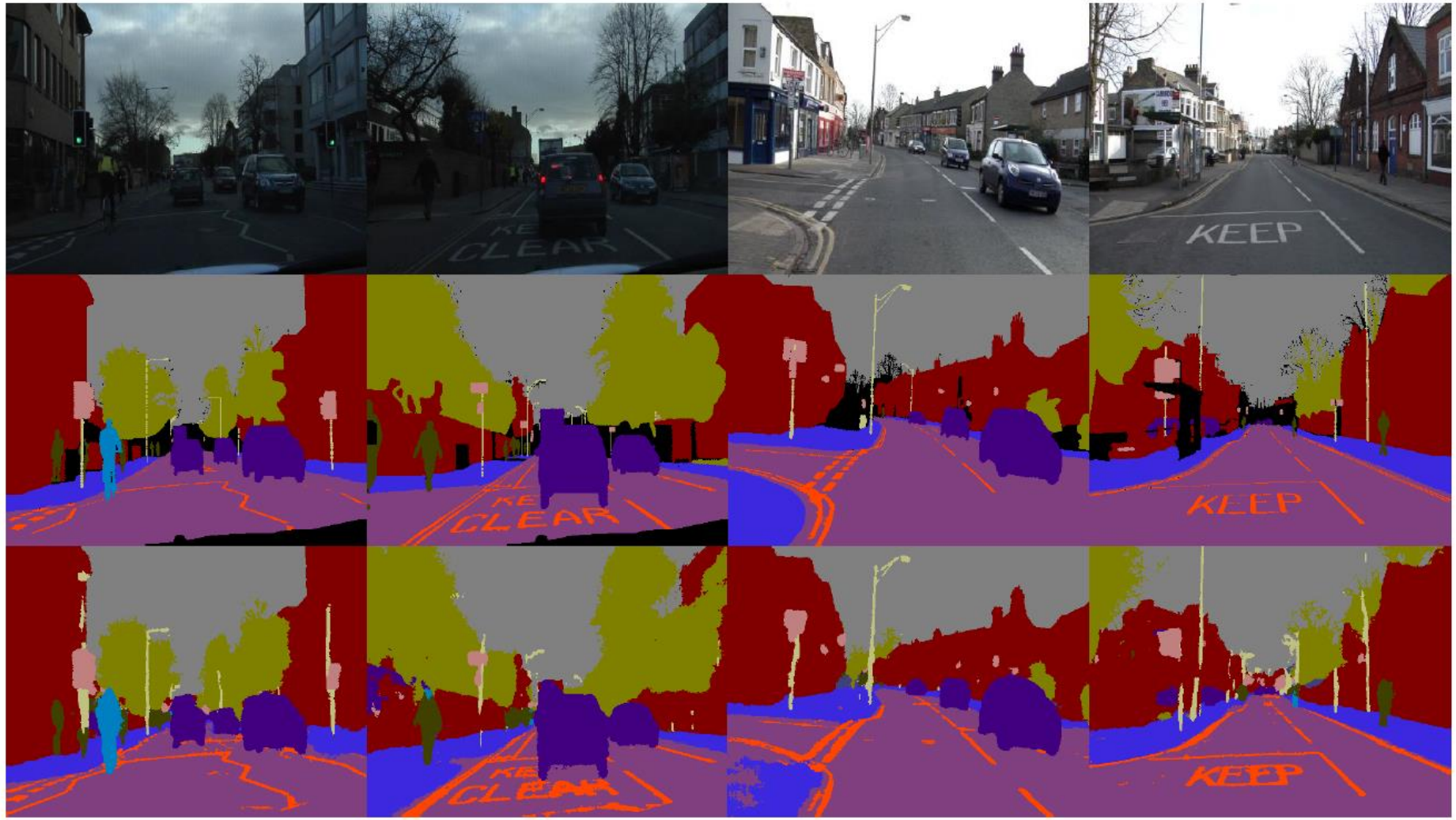
[1] Badrinarayanan et al. [2] Paszke et al.

Results

Input

Ground truth

LMD output



Lane Segmentation

- Network: LMD
- Classes: 2 (lane and non-lane)

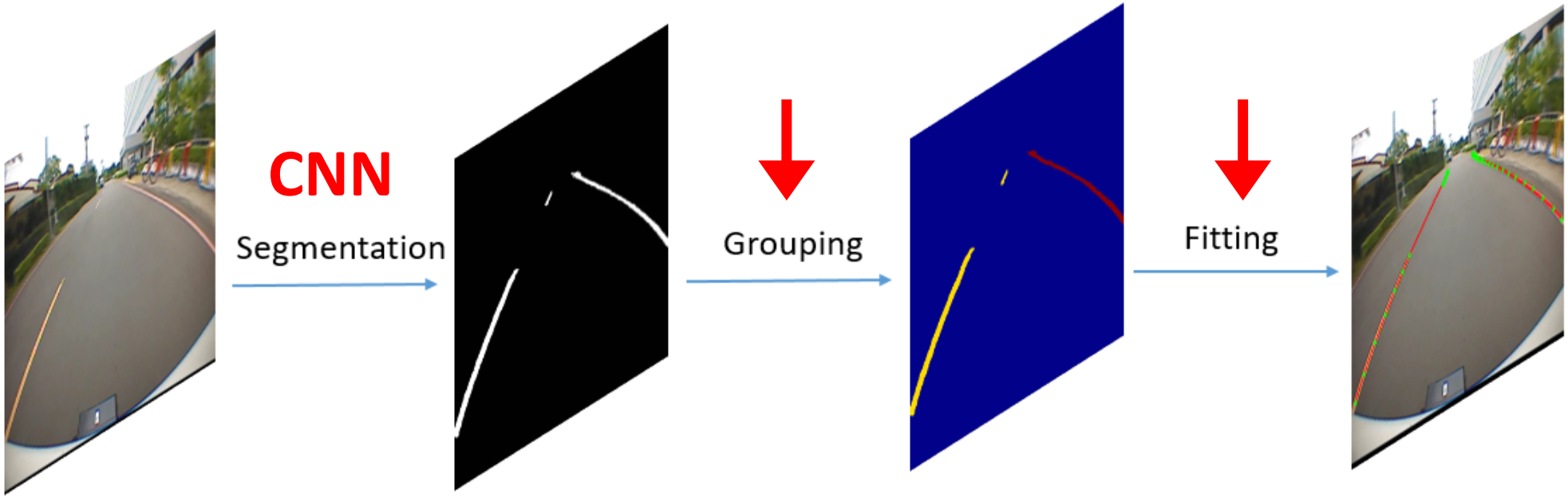
Input: RGB image

LMD

Output: Segmentation

Post-Processing Scheme

- **Step 1:** Segmentation (by deep learning)
- **Step 2:** Grouping (by post-processing scheme)
- **Step 3:** Fitting (by post-processing scheme)



Grouping

- Connected Component Labeling (CCL)

0	1	0	0	0	1
1	1	0	0	0	1
0	1	0	1	1	1
0	1	0	0	0	1
0	0	0	0	0	0
1	1	1	1	0	0

0	1	0	0	0	2
3	1	0	0	0	2
0	1	0	4	4	2
0	1	0	0	0	2
0	0	0	0	0	0
5	5	5	5	0	0

Equal Label :

{1, 3}

{2, 4}

{5}

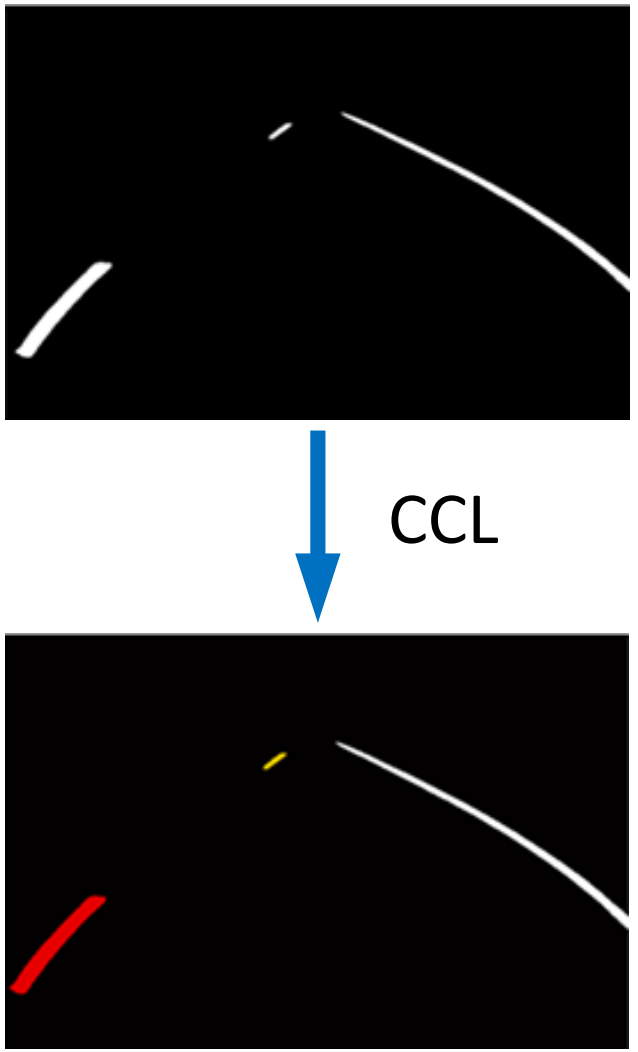
0	1	0	0	0	2
1	1	0	0	0	2
0	1	0	2	2	2
0	1	0	0	0	2
0	0	0	0	0	0
5	5	5	5	0	0

Re-label :

{1, 3} → {1}

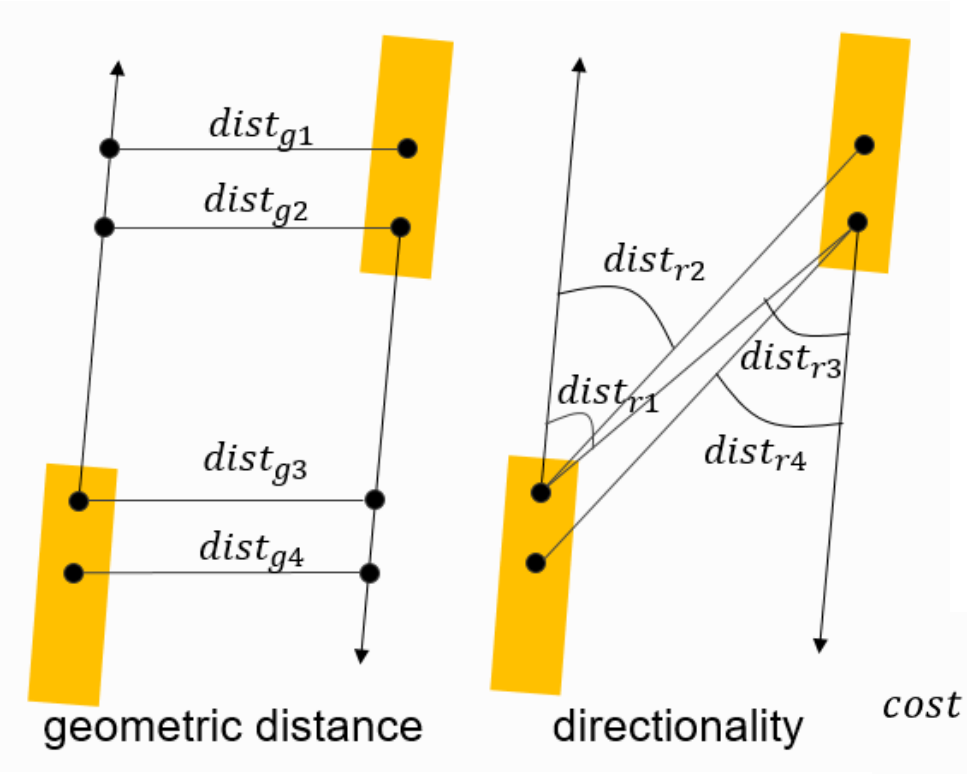
{2, 4} → {2}

{5} → {5}

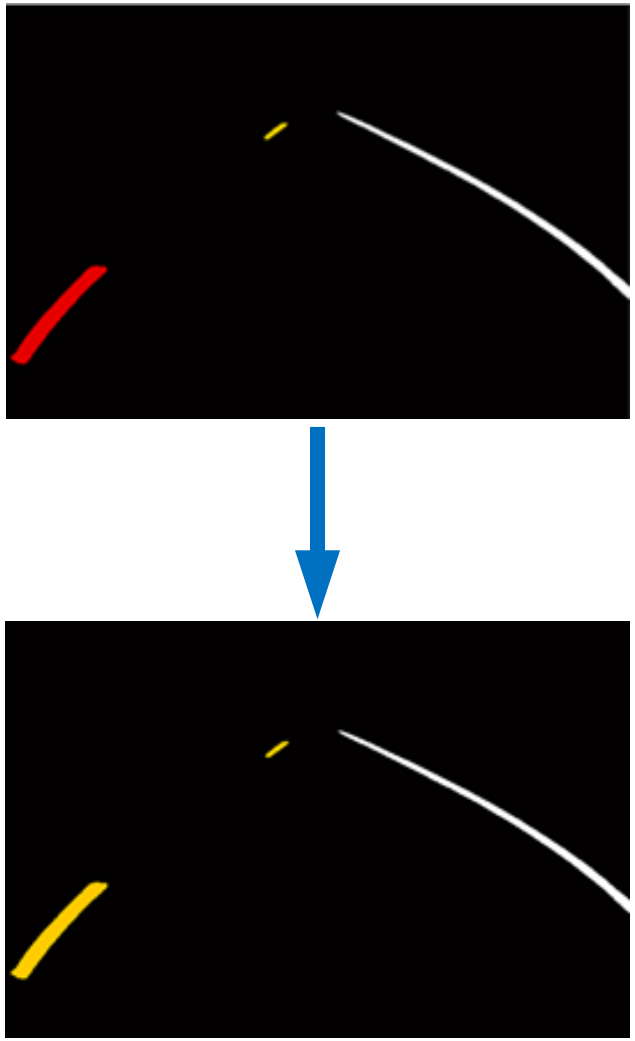


Grouping

- Measurement function design

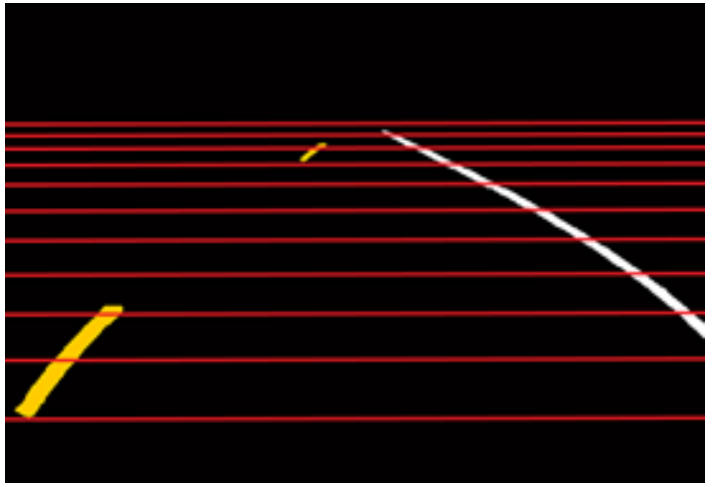


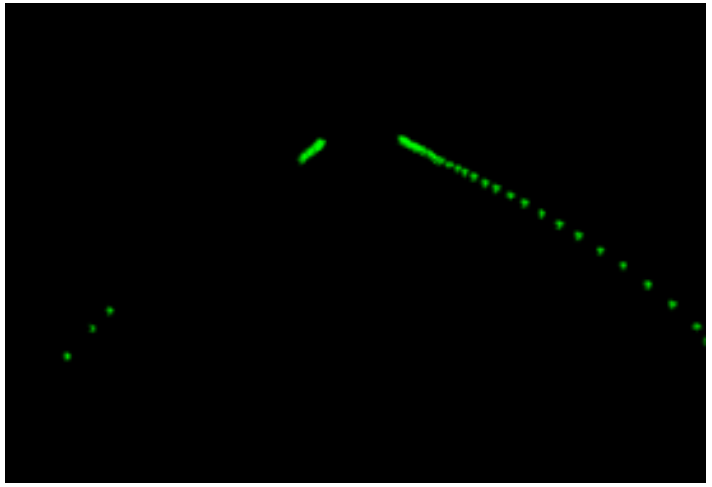
$$cost = \frac{1}{4}\sum_{i=1}^{4}\left(\frac{dist_{gi} + dist'_{ri}}{2}\right)$$



Fitting

- Fitting points selection

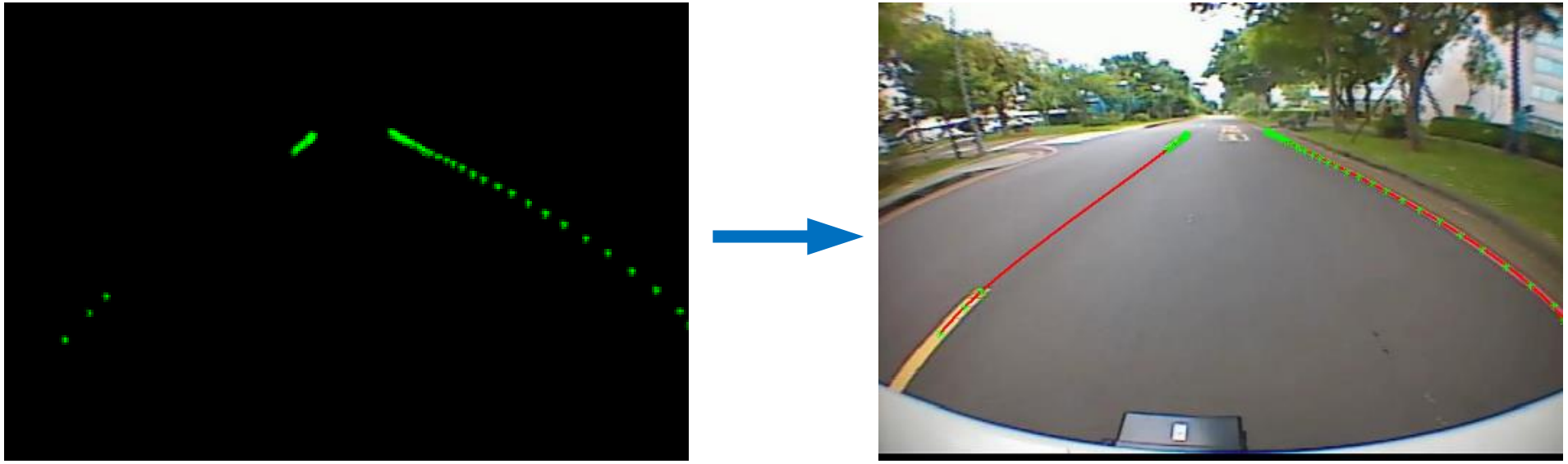




Fitting

- A 3rd order polynomial for curve fitting:

$$y = ax^3 + bx^2 + cx + d$$



Results

Results

- Video demo

Input

Segmentation

Lane detection

Conclusion

- We proposed a new network for semantic segmentation, LMD, which is faster, smaller, and more accurate than SegNet, and can achieve real-time for self-driving applications.

- We designed a post-processing scheme for the final lane detection.