Università degli Studi di Modena e Reggio Emilia

Dipartimento di Ingegneria "Enzo Ferrari"

Laurea Magistrale in Ingegneria Informatica

Elaborazione di un metodo di Deep Learning per Lane Line Detection basato su caratteristiche spazio-temporali

Relatore:

Prof. Lorenzo Baraldi

Relatore:

Prof. Alessandro Correâ Victorino

Correlatore:

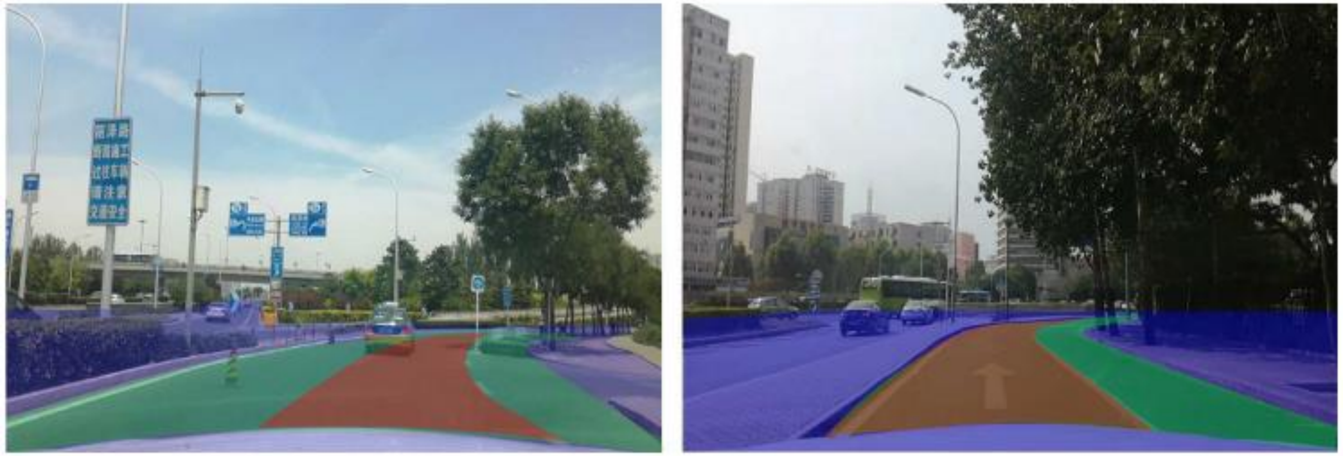
Hugo Pousseur

Candidato:

Antonio Palese

Lane Line Detection

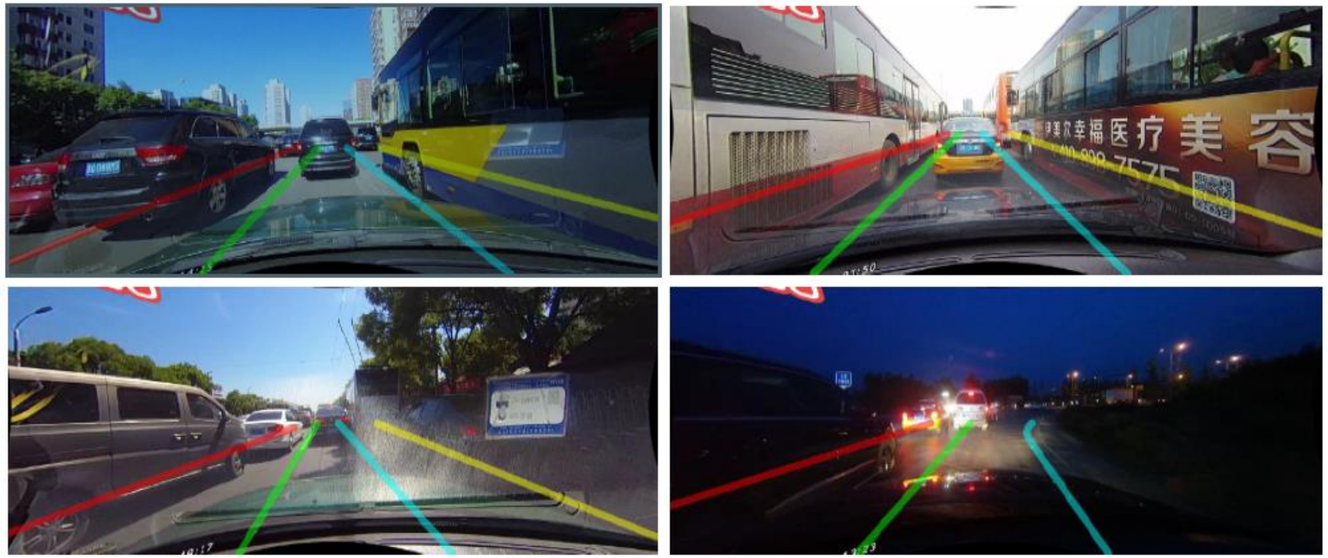
- While driving, we use our eyes to decide where to go. The lines on the road that show us where the lanes are serve as a constant reference.
- Of course, one of the priorities in developing a self-driving car is to automatically detect them.
- The field of Lane Line Detection addresses this issue with the goal of providing safe and responsive systems that can grant sufficiently accurate results to be installed on real systems.



Examples of Lane Detection [1].

[1] Dun Liang *et al.* Lane detection: A survey with new results. Journal of Computer Science and Technology, 2020.

No-visual-clue



No-visual-clue [1].

- The domain of self-driving cars offers a lot of challenging tasks due to the high variability of the driving environment

- A non-trivial issue is called *no-visual-clue*. Difficult scenarios with severe occlusion and extreme lighting conditions correspond to a key problem. In this case, lane detection needs high-level semantic analysis.

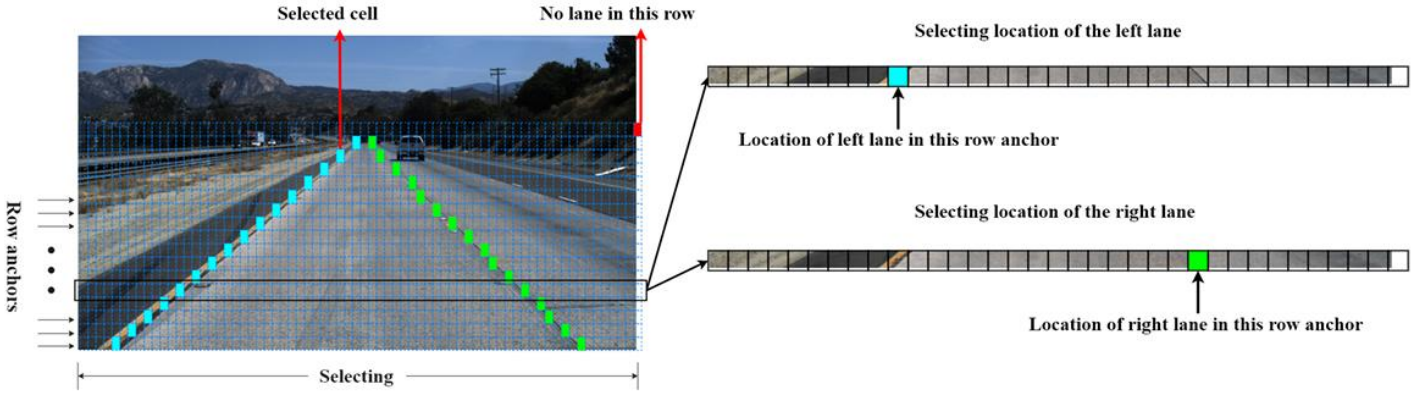
[1] Xingang Pan *et al.* Spatial as deep: Spatial CNN for traffic scene understanding. AAAI, 2018.

Row anchors method

- The approach from which my thesis work started was born by observing a method for lane detection presented in “Ultra Fast Structure-aware Deep Lane Detection”[1].

- The purpose is to regress a probability distribution of the existence of the line along the grid spaces, in each row.

- The cell with the highest value obtained is, with a good approximation, the correct one for the specific line, in the selected row anchor. Therefore, it is a case of classification. To handle the absence of a lane, an extra dimension is added.


Selected cell
No lane in this row
Selecting location of the left lane
Location of left lane in this row anchor
Selecting location of the right lane
Location of right lane in this row anchor
Row anchors
Selecting

Row anchors graphical explanation [1].

Computational cost = $C \cdot h \cdot (w+1)$

C → Number of lines to detect

h → Number of row anchors

w → Number of horizontal cells

[1] Zequn Qin *et al.* Ultra fast structure-aware deep lane detection. ECCV, 2020.

Adding temporal features

Two subsequent frames extracted from CULane [1].

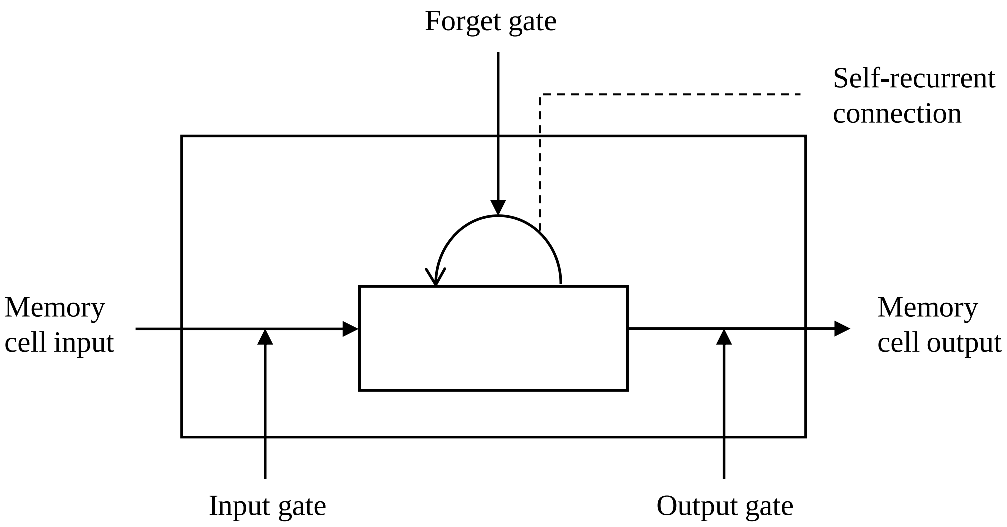
- Considering data where patterns are repeated more and more in time, focusing on temporal understanding could be effective.

- The main idea is to use temporal features to improve the detection of lines. In fact, each frame could be seen not in an absolute way, but as part of a chronological sequence.

- In practice, when a lane line in the current frame is covered by a foreign body, considering a time series of frames can overcome the problem of lack of visual information due to obstacles.

[1] Xingang Pan *et al.* Spatial as deep: Spatial CNN for traffic scene understanding. AAAI, 2018.

How to handle the concept of time



LSTM cell [1].

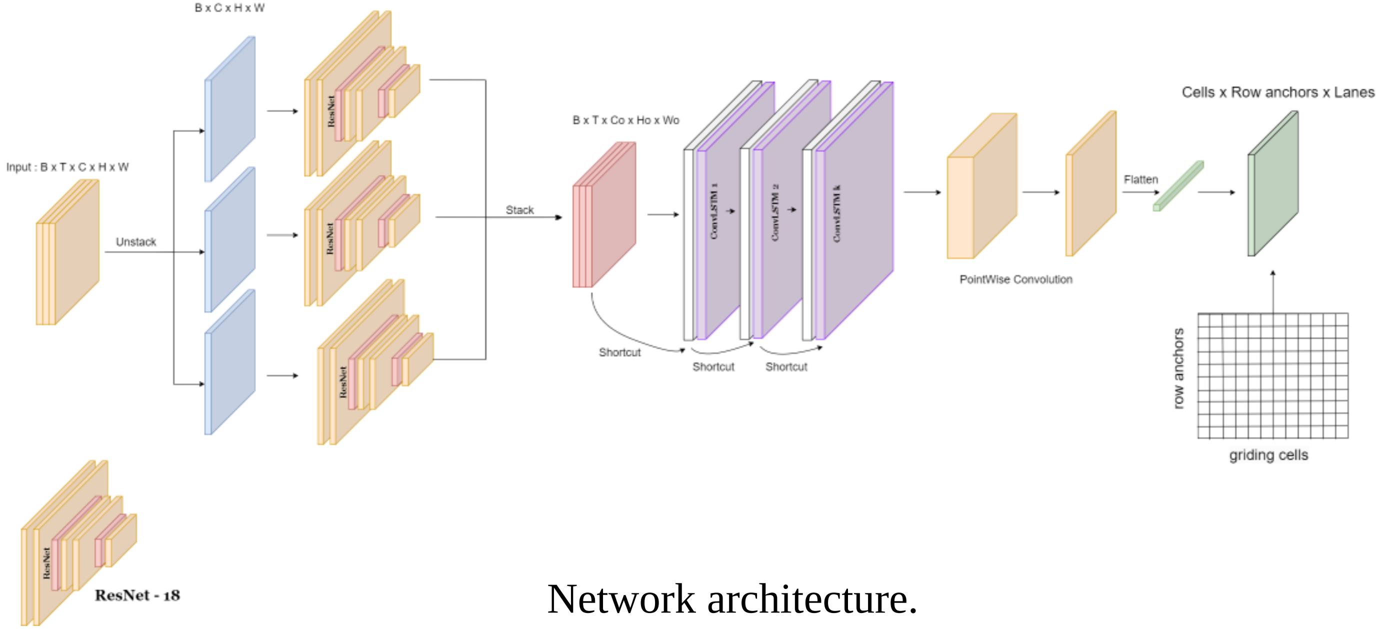
- For sequence prediction, the LSTM (Long-Short-Term-Memory) structure as a particular recurrent neural network has proven to be stable and powerful for modelling long-term dependencies [1].
- The main innovation of LSTM is its memory cell, which acts as an accumulator of state information. The cell is accessed, written to, and erased by various control gates.
- Each time a new input arrives, its information will be accumulated in the cell if the input gate is activated. Additionally, the past state can be “forgotten” if the forget gate is turned on. Whether the cell’s last output will be propagated to the final state is further controlled by the output gate.

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 1997.

Pipeline architecture

- This section describes the lane detection pipeline, regarding feature extraction and the temporal prediction phase. All of this is focused on the neural network designed for this purpose:
    - The backbone layer
    - The Convolutional Long Short Term Memory layer [1]
    - The final classifier

- The network has been trained end-to-end by using the **Focal Loss** [2].



Network architecture.

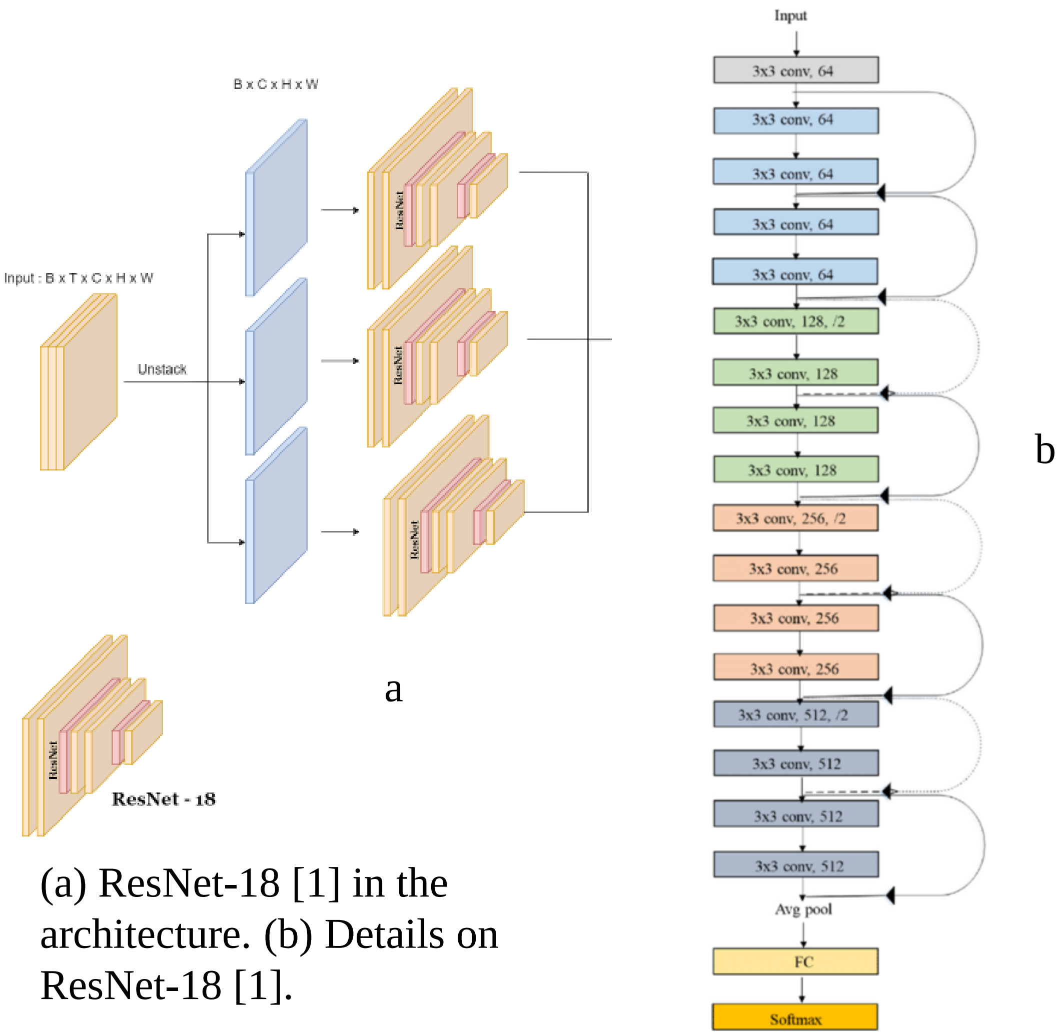
[1] Xingjian Shi *et al.* Convolutional LSTM network: A machine learning approach for precipitation nowcasting. NIPS, 2015.
[2] Tsung-Yi Lin *et al.* Focal loss for dense object detection. ICCV, 2017.

The backbone

- The input is a stacked sequence of three chronological frames fed into the backbone.

- The backbone has the role of feature extractor. Temporal prediction on input images cannot be considered using raw format images, but these must be processed to retain only the best spatial features. For this purpose, ResNet [1] is used.

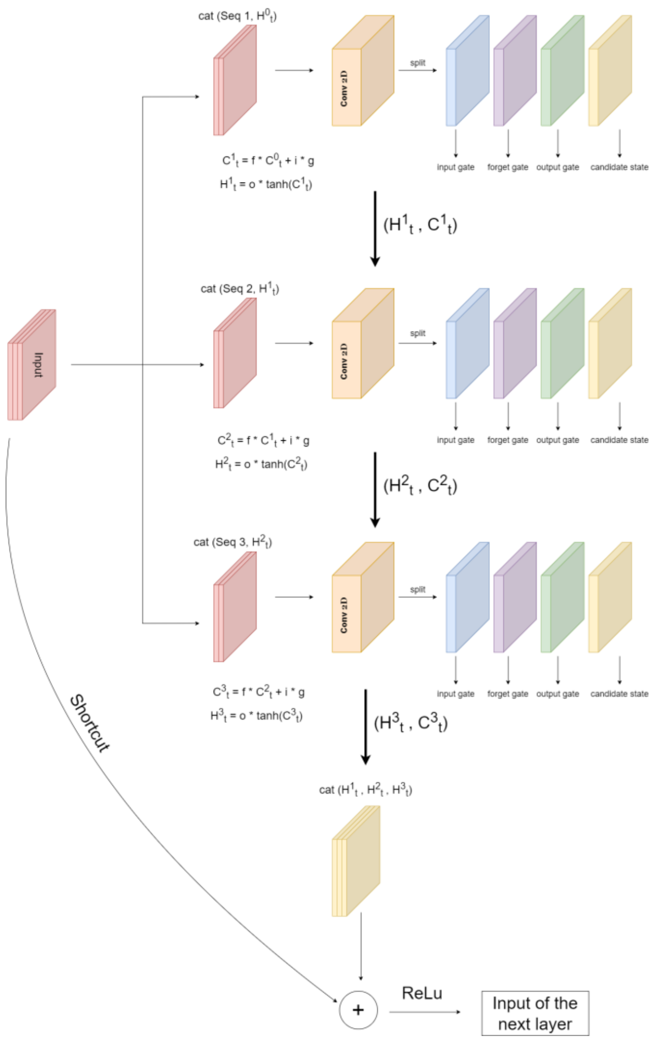
- ResNet [1], from which the fully connected layers were separated and only the convolutional architecture was retained, was used in the 18-layers version.



(a) ResNet-18 [1] in the architecture. (b) Details on ResNet-18 [1].

[1] Kaiming He *et al.* Deep residual learning for image recognition. CVPR, 2015.

ConvLSTM



ConvLSTM implementation.

- The proposed method combines the lane detection technique based on row anchors with a method of predicting temporal features, using Convolutional Long-Short-Term-Memory technology [1].

- The main disadvantage of standard LSTM in handling spatial-temporal data is the use of fully connected connections in input-to-state and state-to-state transitions, where no spatial information is encoded.

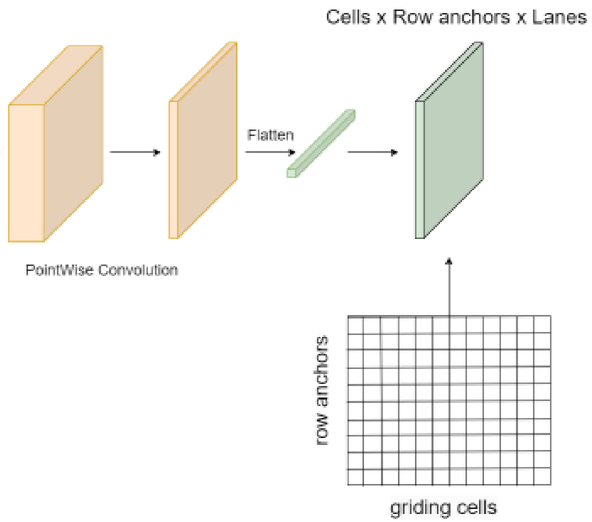
- ConvLSTM determines the future state of a certain cell on the inputs and past states of its local neighbours. This can be easily achieved by using a convolution operator in state-to-state and input-to-state transitions.

- Inspired by ResNet, the ConvLSTM [1] is enhanced by shortcut connections.

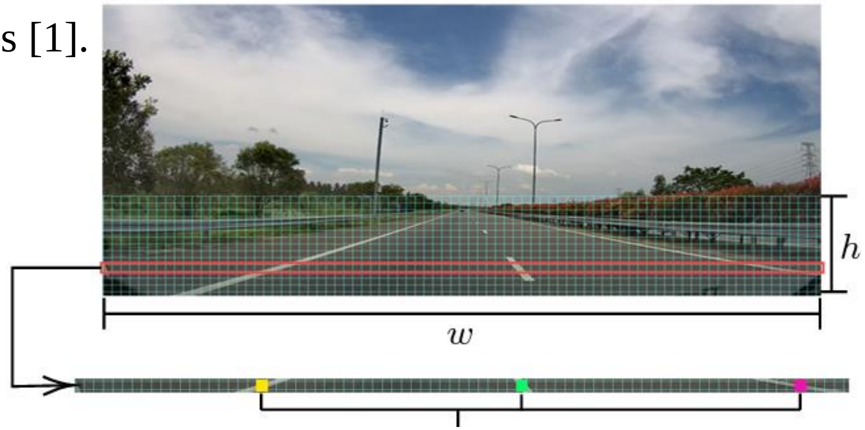
[1] Xingjian Shi *et al.* Convolutional LSTM network: A machine learning approach for precipitation nowcasting. NIPS, 2015.

Classifier

Classifier.



Lane locations [1].



- At the end of the model, the output of the ConvLSTM is resized along the channel dimension by a point-wise convolution and passed to a fully connected layer that flattens the feature map into a one-dimensional vector.

- The vector is arranged into a three-dimensional matrix for row-by-row classification of each cell position with a percentage of existence of a line segment.

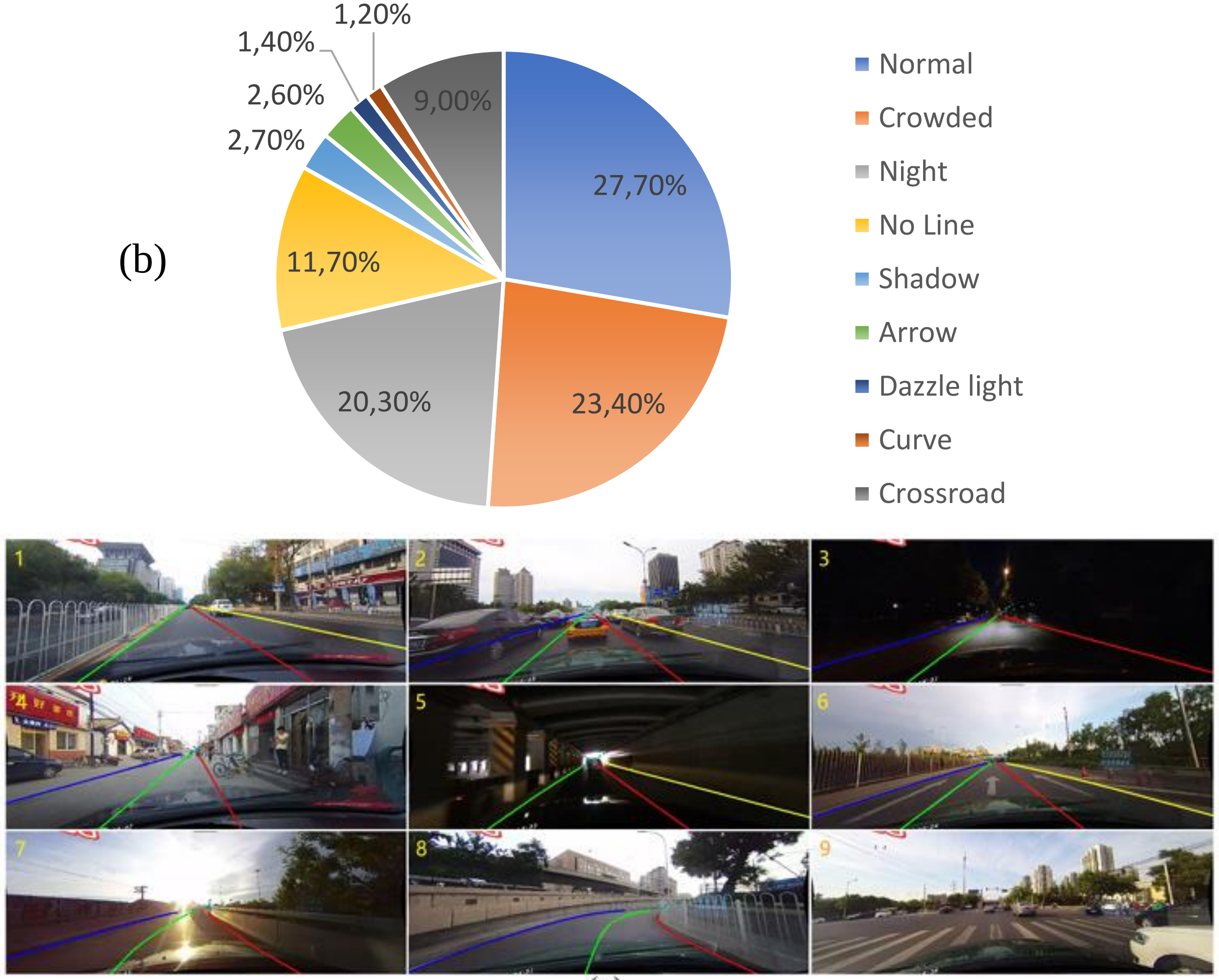
[1] Zequn Qin *et al.* Ultra fast structure-aware deep lane detection. ECCV, 2020.

CULane

- CULane is a large scale dataset for Lane Line Detection.

- More than 55 hours of video were collected and 133,235 images were extracted.

- The test set was divided into normal and eight complex categories, which correspond to the nine examples in the figure.

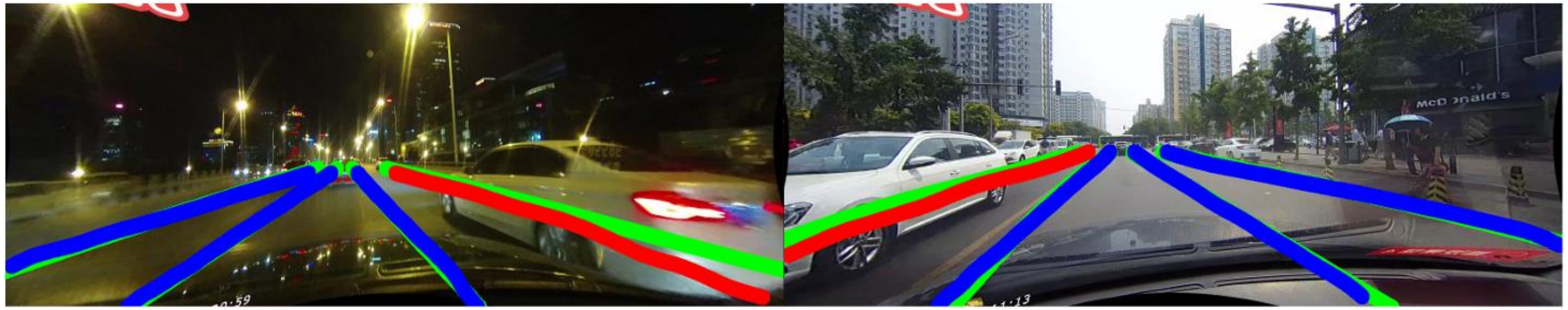


(a)

(a) CULane images. (b) Portions of the dataset [1].

[1] Xingang Pan *et al.* Spatial as deep: Spatial CNN for traffic scene understanding. AAAI, 2018.

Evaluation metric



Green lines denote ground truth, while blue and red lines denote TP and FP respectively [1].

$$\text{F1} - \text{measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

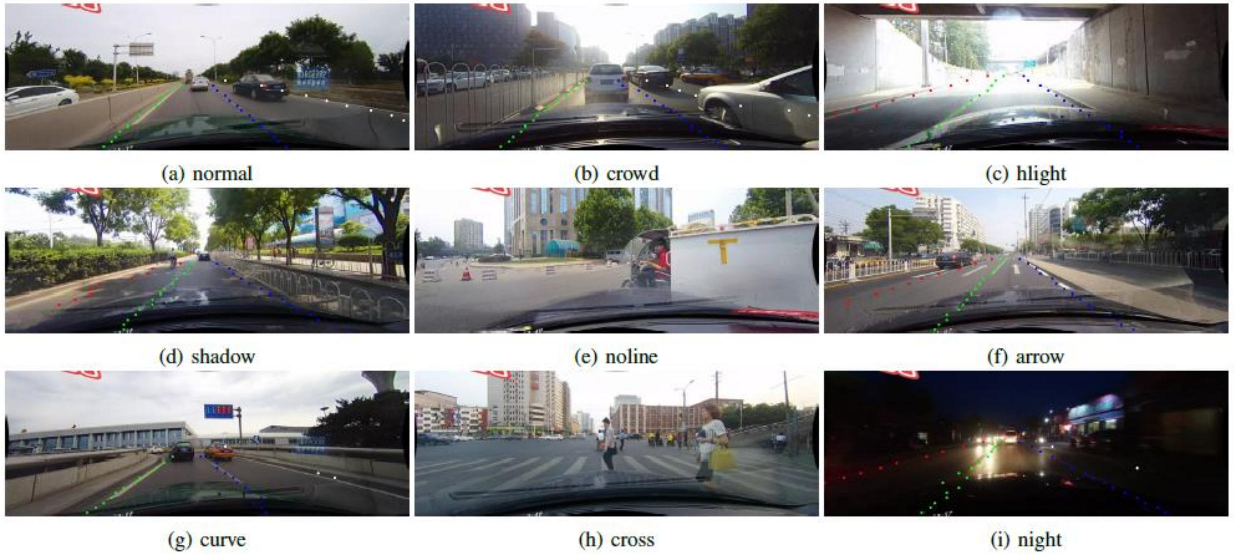
- As for the evaluation metric of CULane, each lane is treated as a line 30 pixels wide. Then the intersection over union (IoU) is calculated between the actual values and the predictions [1].

- Predictions with IoU greater than 0.5 are considered true positives [1].

[1] Xingang Pan *et al.* Spatial as deep: Spatial CNN for traffic scene understanding. AAAI, 2018.

Results

Results of the model applied to the CULane test set. The image shows the nine categories.

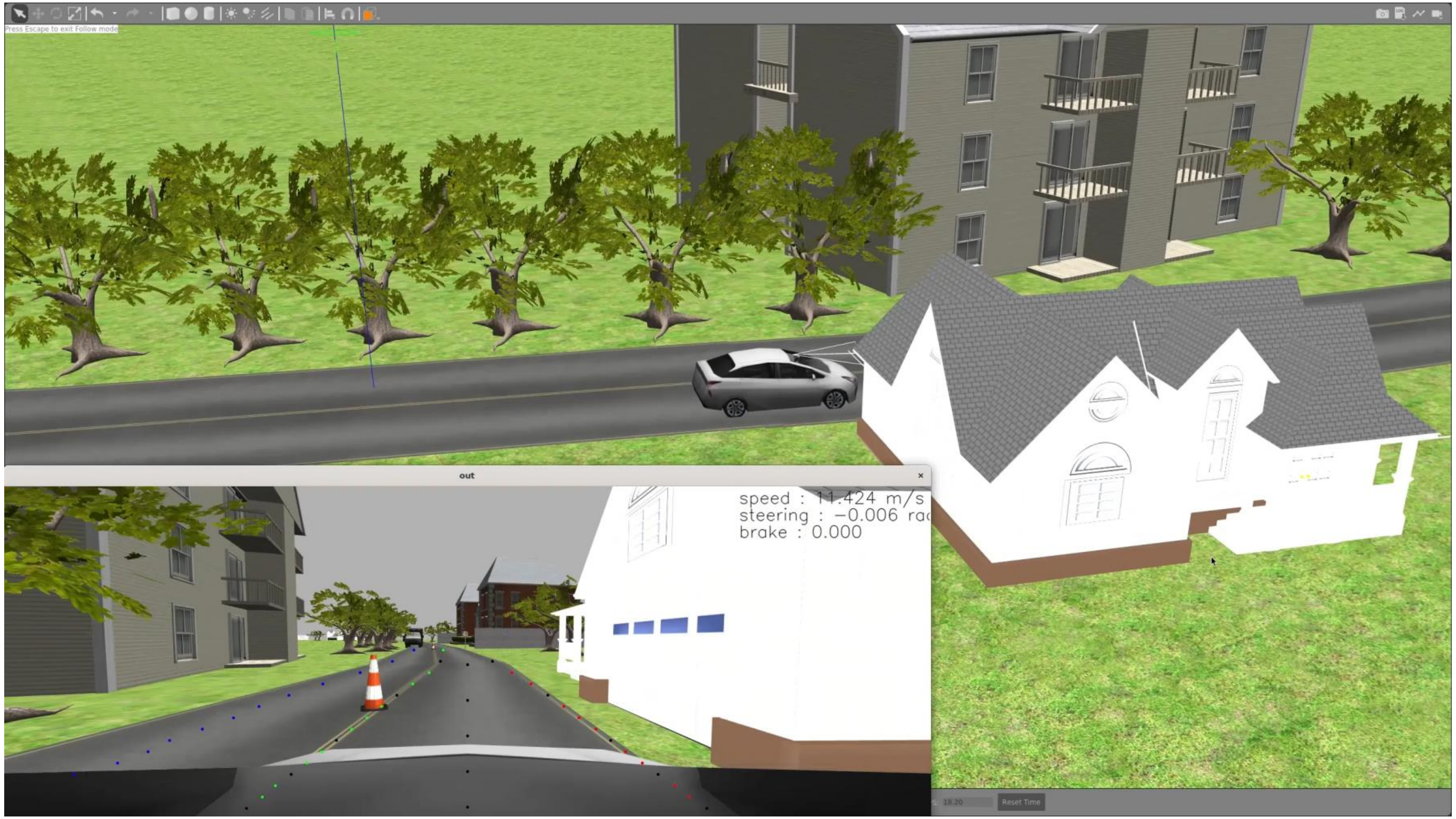

(a) normal
(b) crowd
(c) hlight
(d) shadow
(e) noline
(f) arrow
(g) curve
(h) cross
(i) night

Category	UltraFast Res-18 [1]	Ours Res-18
Normal	87.7	**89.44**
Crowded	66.0	**67.32**
Night	62.1	**63.81**
No-Line	40.2	**40.60**
Shadow	62.8	**63.44**
Arrow	81.0	**83.42**
Dazzle light	**58.4**	54.79
Curve	57.9	**65.07**
Crossroad	**1743**	2238
Total	68.4	**69.71**

Comparison with the competitor, with IoU threshold = 0.5. For crossroad, only FP are shown.

[1] Zequn Qin *et al.* Ultra fast structure-aware deep lane detection. ECCV, 2020.

ROS simulation



```
speed : 11.424 m/s
steering : −0.006 rac
brake : 0.000
```

https://www.osrfoundation.org/simulated-car-demo/

Thanks for your attention

Any question?