Supplemental Materials on

Eigenlanes: Data-Driven Lane Descriptors for Structurally Diverse Lanes

Dongkwon Jin[1], Wonhui Park[1], Seong-Gyun Jeong[2], Heeyeon Kwon[2], Chang-Su Kim[1]
Korea University[1], 42dot.ai[2]
{dongkwonjin, whpark}@mcl.korea.ac.kr, {seonggyun.jeong, heeyeon.kwon}@42dot.ai, changsukim@korea.ac.kr

A. Implementation Details

Let us describe how to configure training data for SIIC-Net. Then, we present the training process and the architecture of SIIC-Net in more detail.

A.1. Data configuration

SI module: Let $\bar{\mathbf{l}}$ and $\bar{\mathbf{c}}$ denote a ground-truth lane and its coefficient vector in the eigenlane space. For a lane candidate $\mathbf{l}_i$, the l_1-distance $d_i = \|\mathbf{l}_i - \bar{\mathbf{l}}\|_1$ is computed from the ground-truth $\bar{\mathbf{l}}$. If the distance is lower than a threshold, it is declared as a positive lane. In such a case, the probability $\bar{P}_i$ is annotated to be proportional to $e^{-d_i^2}$, and the offset vector $\bar{O}_i$ is annotated to be $\bar{\mathbf{c}} - \mathbf{c}_i$, respectively. Also, let $\mathcal{H} = \{h_1, h_2, \ldots, h_e\}$ denote the set of height classes of lanes, which are obtained by clustering the y-coordinates of the ending points of all training lanes. Then, the height probability $\bar{H}_i$ is annotated as one-hot vector, in which the jth element is set to 1 if h_j is the closest one to the ending point of $\bar{\mathbf{l}}$.

IC module: For an index set of lane candidates $\mathcal{S} = \{s_1, s_2, \ldots, s_T\}$, a ground-truth relation matrix $\mathbf{R}$ of size $T \times T$ should be generated to train the IC module. To compose the index set $\mathcal{S}$, each element can be selected randomly from $[1, K]$. However, this approach fails to train the IC module reliably due to many redundant negative lane candidates. To overcome this issue, we construct $\mathcal{S}$ by exploiting the NMS output, which informs of the top T reliable lanes. Specifically, for each ground-truth lane, a positive lane candidate becomes an element in $\mathcal{S}$. Then, randomly selected lanes from the NMS output become the remaining ones in $\mathcal{S}$. The relation score $\bar{\mathbf{R}}(i,j)$ is annotated as $(\bar{P}_{s_i} + \bar{P}_{s_j})/2$ if two candidates are positive lanes corresponding to different ground-truth ones, where $\bar{P}_{s_i}$ is the ground-truth probability for lane candidate $\mathbf{l}_{s_i}$. Otherwise, $\bar{\mathbf{R}}(i,j)$ is annotated to be zero.

A.2. Training details

We define the loss for training the SI module as

$$\ell_{\text{SI}} = \ell_{\text{cls}}(P_i, \bar{P}_i) + \ell_{\text{cls}}(H_i, \bar{H}_i) + \ell_{\text{reg}}(O_i, \bar{O}_i), \tag{1}$$

where P_i, H_i, O_i are the output of the SI module, corresponding to the annotations $\bar{P}_i$, $\bar{H}_i$, $\bar{O}_i$, respectively. Also, ℓ_{cls} is the focal loss over the classes, and ℓ_{reg} is the mean squared error (MSE). The loss function for training the IC module is defined as $\ell_{\text{IC}} = \|\mathbf{R} - \bar{\mathbf{R}}\|_F^2$ between the estimated relation matrix $\mathbf{R}$ and the ground-truth $\bar{\mathbf{R}}$. Finally, to train the decoder, the cross entropy loss [25] is used.

We use the Adam optimizer with an initial learning rate of 10^{-5} and halve it after every 30 epochs five times. Also, we use a batch size of four for 400,000 iterations and augment training images by randomly flipping them horizontally. For the TuSimple, CULane, and SDLane datasets, we resize training images to 384×640, 384×800, and 384×640, respectively. We will make the source codes publicly available.

A.3. Network details

SIIC-Net consists of encoder-decoder part, SI module, and IC module. Based on ResNet50 [9], we implement the encoder, which extracts an aggregated feature map X_{a} of size $H \times W \times 384$ and a squeezed feature map X_{s} of size $H \times W \times 128$. We also employ the auxiliary branch [25] to design the decoder to produce a binary segmentation map of size $H \times W \times 1$. The number R of pre-defined heights is 2 and the number M of eigenlanes is 4 in all tests. Thus, in the SI module, the size of

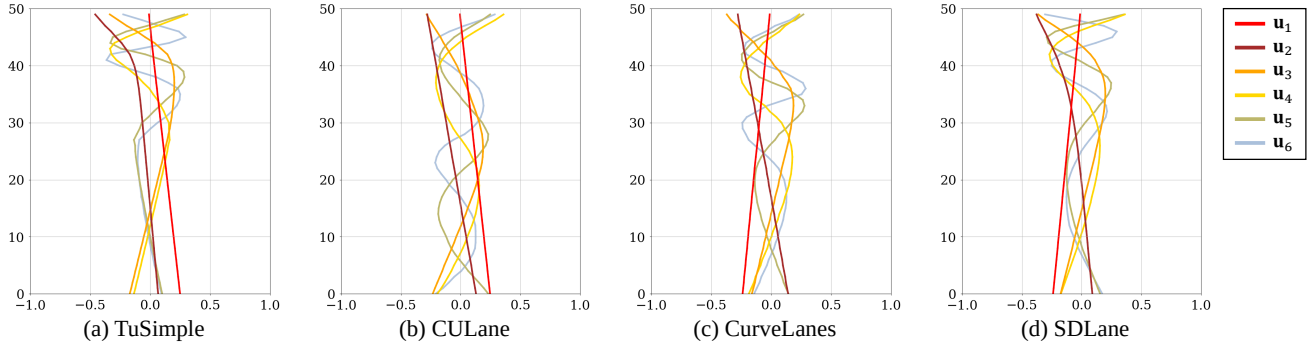
the last fully-connected layers of the height classification layer and the regression layer are 128×2 and 128×4, respectively. In the IC module, we halve the number of channels of a lane feature map Y_{a} of size 10×384 by employing the feature transforms ϕ_1 and ϕ_2 in Eq. (12) in the main paper, which contain a series of 1D convolution layers and ReLU activation functions. Then, $\phi_1(Y_{\mathrm{a}})$ and $\phi_2(Y_{\mathrm{a}})$ of size 10×192 are l_2-normalized, respectively. Also, in the default mode, $K = 500$ and $T = 10$.

B. Dataset Analysis

We analyze and compare four lane detection datasets: TuSimple [1], CULane [22], CurveLanes [33], and SDLane. Except for CurveLanes, the three datasets are described in Section 4.1. CurveLanes is the largest lane detection dataset, consisting of 100K training, 20K validation, 30K testing images, but the annotations for test images are not publicly available yet.

B.1. Eigenlanes - Image Examples

As in Section 3.2, we construct the eigenlane spaces for the four lane detection datasets. Figure 1 shows the first six eigenlanes $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_6$ for each dataset in the image space. For TuSimple and SDLane, $\mathbf{u}_1$ is a straight line and $\mathbf{u}_2$ is slightly curved at the top side (far from the cameras). For CULane and CurveLanes, the two eigenlanes are straight lines with different slopes. As m increases, $\mathbf{u}_m$ represents a more complicated curve. While the eigenlanes for TuSimple exhibit high fluctuations at the top side, those for the other datasets oscillate more regularly. This indicates the differences in curved lane distributions among the four datasets.



(a) TuSimple (b) CULane (c) CurveLanes (d) SDLane

Figure 1. The first six eigenlanes $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_6$ for the four lane detection datasets.

B.2. Structural diversity

The four lane detection datasets have different characteristics. We analyze the structural diversity of lanes within each dataset. Let $\mathbf{A} = [\mathbf{x}_1, \cdots, \mathbf{x}_L]$ be a dataset and $\mathbf{A}_M = [\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_L]$ be its rank-M approximation in Eq. (2) in the main paper. Then, we define the error-to-signal (E2S) ratio of the rank-M approximation as

$$\begin{aligned} \mathrm{E2S}(M) &= \frac{\|\mathbf{A} - \mathbf{A}_M\|_F^2}{\|\mathbf{A}\|_F^2} = \frac{\sum_{i=1}^{L} \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2}{\sum_{i=1}^{L} \|\mathbf{x}_i\|^2} \\ &= 1 - \frac{\sum_{i=1}^{M} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2} \end{aligned} \tag{2}$$

where the last equality holds because of Eq. (3) in the main paper. Figure 2 plots the E2S graph of each dataset. E2S approaches zero quickly as M gets higher.

For CULane and CurveLanes, E2S tends to decrease similarly before $M = 4$. Especially, the scores drop significantly between $M = 1$ and $M = 2$. It means that two eigenlanes are enough to represent most lanes in CULane and CurveLanes, which are straight ones. Note that both $\mathbf{u}_1$ and $\mathbf{u}_2$ are straight lines as shown in Figure 1 (b) and (c). Also, even though some lanes in the 'Curve' category in CULane have high curvatures, there are too few such curved lanes. Even though over 90% of images contain curved lanes in CurveLanes, those lanes are slightly curved near the ending points. For TuSimple, E2S(2) is higher than CurveLanes, but E2S(4) is the lowest. This means that it contains curved lanes whose shapes are simple and similar to one another. Finally, SDLane yields significantly higher E2S ratios than other datasets after $M = 1$. In other words, among the four datasets, the proposed SDLane is the most diverse one.

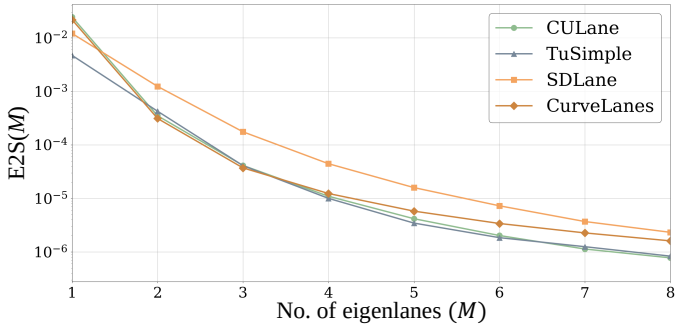


Figure 2. The E2S graphs of the benchmark lane detection datasets according to the number M of eigenlanes. The y-axis is in a logarithmic scale.

B.3. More details on SDLane

Let us describe the proposed SDLane dataset in more detail. We collected 45K road images with a high resolution of 1208×1920 in several cities in Korea, such as Seoul, Seongnam, Gwacheon, and Hwaseong. In an image, there are up to 7 road lanes, and the lanes in the opposite direction are excluded. Each lane is annotated by a series of 2D points.

Table 1. The distribution of lanes according to the maximal curvatures in the SDLane dataset.

Curvature	$0^\circ \sim 10^\circ$	$11^\circ \sim 20^\circ$	$21^\circ \sim 30^\circ$	$31^\circ \sim 45^\circ$	$46^\circ \sim 89^\circ$
Percentage (%)	38.7	17.8	11.9	9.6	22.0

We analyze the lane distribution of the SDLane dataset in terms of curvatures. We divide each lane into line segments. Then, we measure the maximal curvature, which is the maximum angle between adjacent segments. Table 1 reports the percentage of lanes in SDLane that have maximal curvatures in the five ranges: $0^\circ \sim 10^\circ$, $11^\circ \sim 20^\circ$, $21^\circ \sim 30^\circ$, $31^\circ \sim 45^\circ$, and $46^\circ \sim 89^\circ$. About 40% of lanes are in the range of $0^\circ \sim 10^\circ$, which are almost straight ones. On the other hand, over 40% have maximal curvatures higher than 20°. It means that SDLane contains more curved lanes than straight ones.

In Figure 3, the images are grouped according to the maximum of the maximal curvatures of all lanes in an image. Note that, in $0^\circ \sim 10^\circ$, lanes are straight or slightly curved. As the curvatures get higher, lanes are more curved. Especially, in $45^\circ \sim 89^\circ$, some lanes for left or right turns are highly curved and implicit.

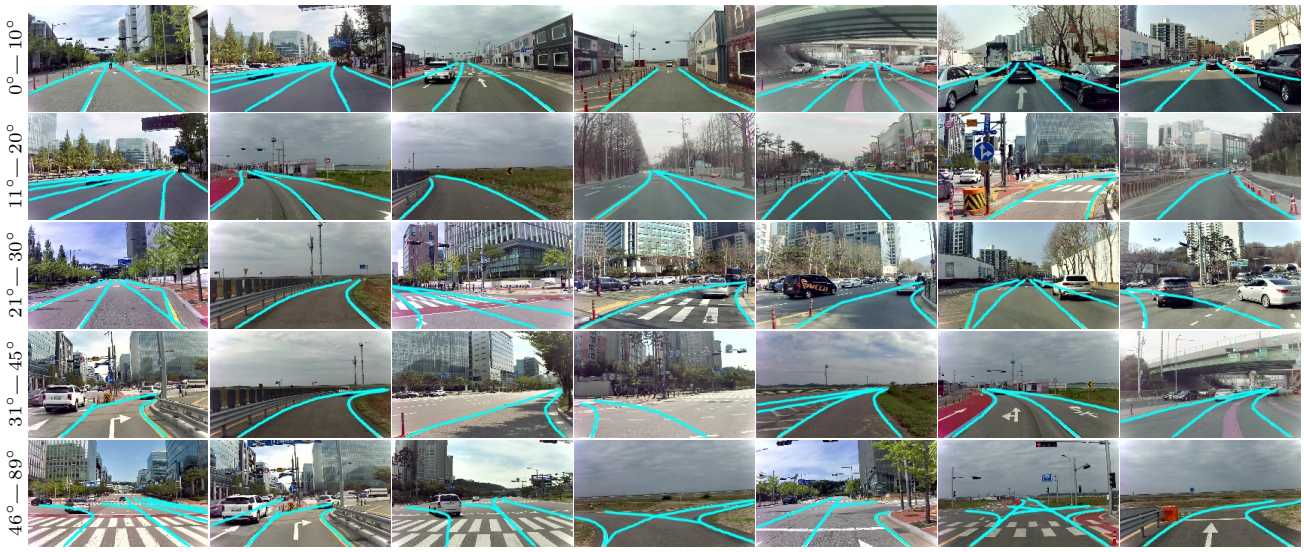
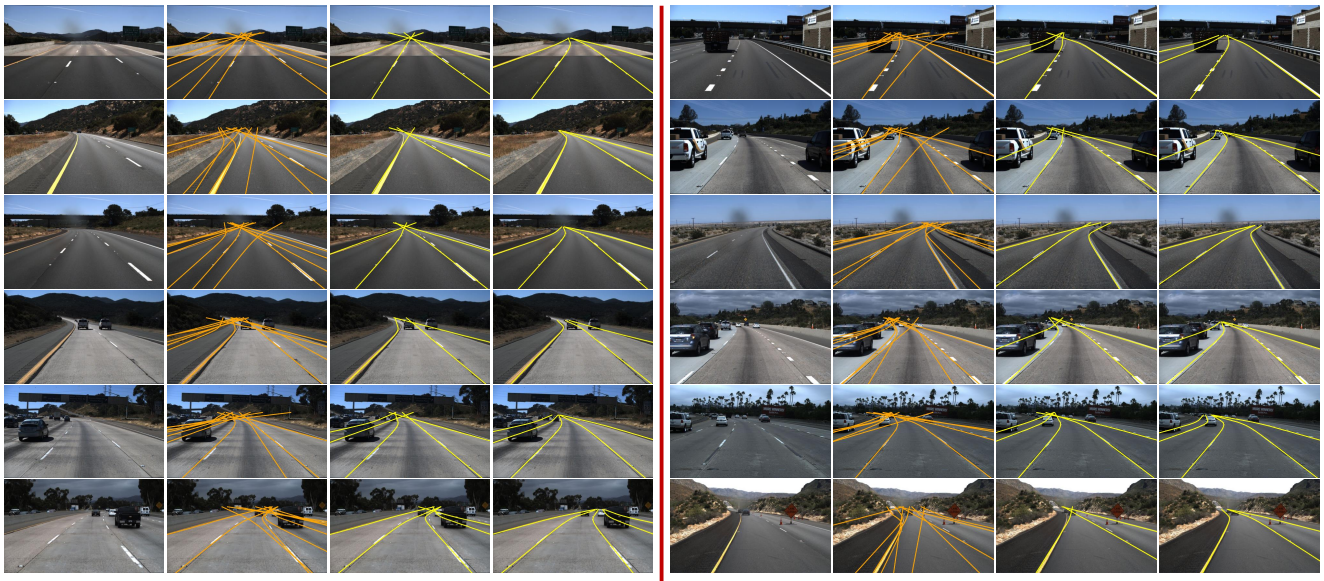


Figure 3. Example images with the ground-truth lanes in the SDLane dataset, according to the maximal curvature of all lanes in an image.

C. Experimental Results

C.1. More examples of lane detection by SIIC-Net

Figure 4 illustrates how the proposed SIIC-Net detects lanes.



(a) (b) (c) (d) (a) (b) (c) (d)

Figure 4. Lane detection examples by SIIC-Net: (a) input image, (b) 10 selected lanes after NMS, (c) optimal lanes determined by MWCS, and (d) refined lanes using the regression offsets from the SI module.

C.2. More detection results on TuSimple

The proposed algorithm detects even curved or occluded lanes precisely, except for too short lanes far from the cameras.

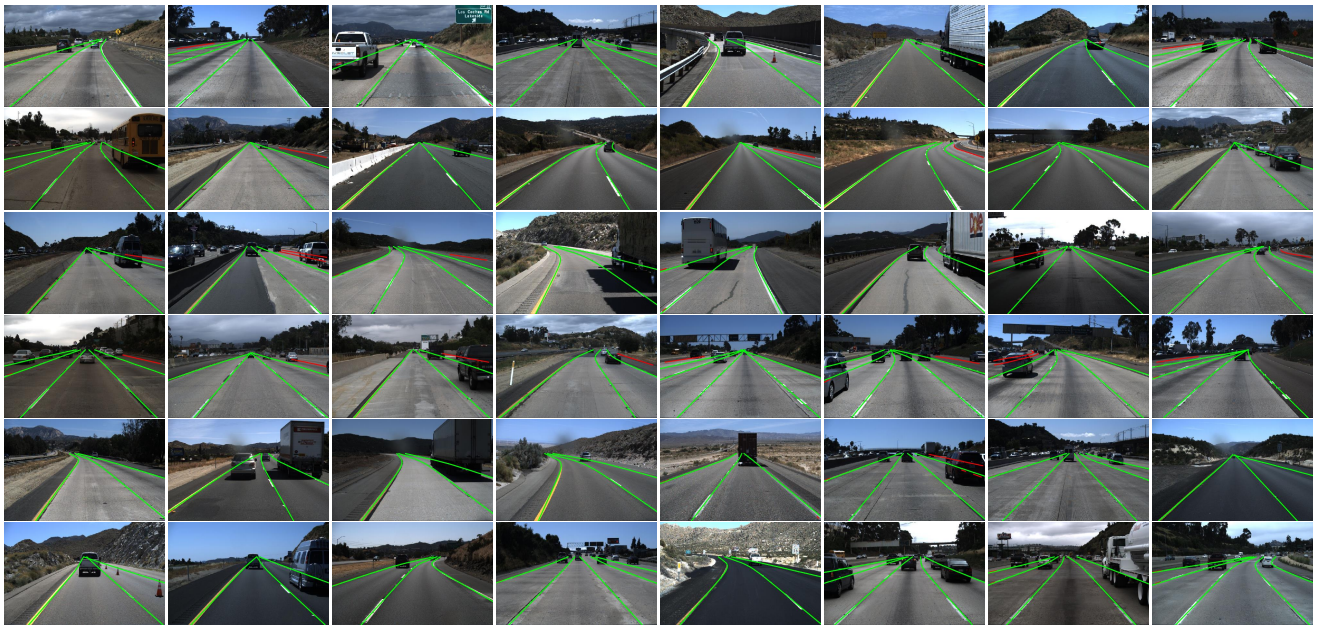


Figure 5. Detection results of the proposed algorithm on the TuSimple dataset. Detected lanes are depicted in green, while false negatives are in red.

C.3. Comparison on CULane

Figure 6 compares the proposed algorithm with the conventional road lane detectors [22, 25, 34] on the CULane dataset. Since the conventional detectors are based on the segmentation framework, they cannot preserve the continuous lane structure in detection results. Also, they poorly recall highly implied lanes or fail to localize each lane precisely. In contrast, the proposed algorithm detects lanes reliably.

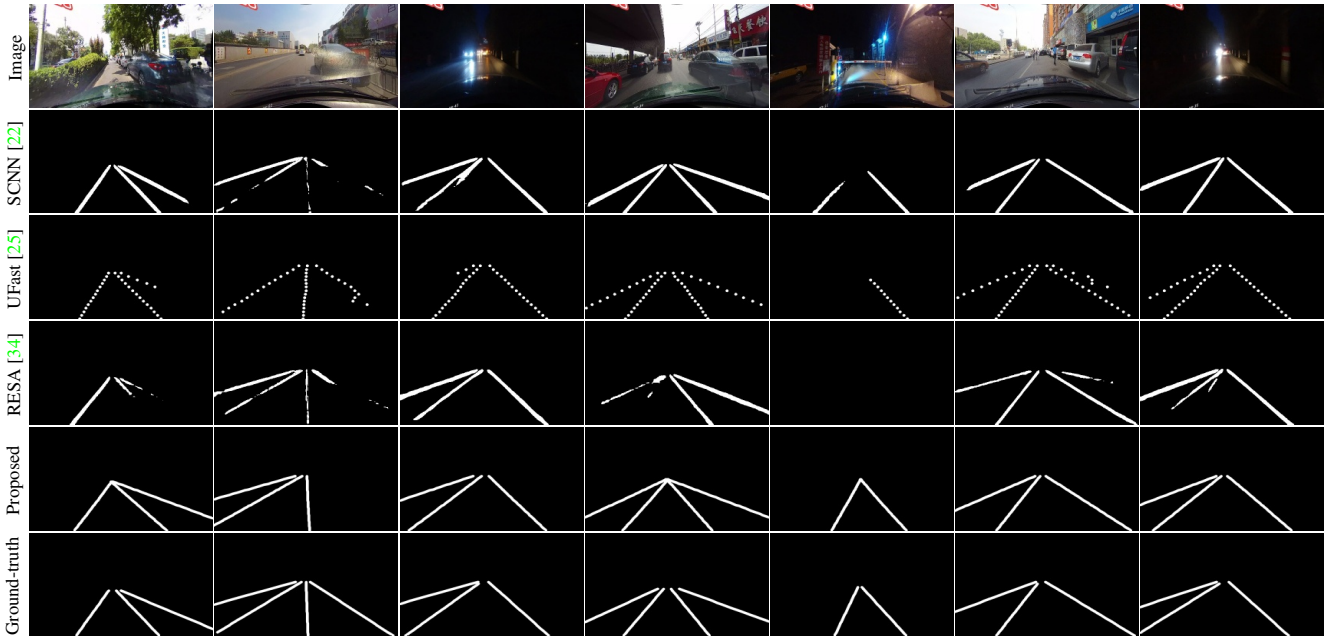


Figure 6. Comparison of lane detection results on the CULane dataset.

C.4. Comparison on SDLane

Figure 7 compares the proposed algorithm with [19, 30, 34], which are the state-of-the-art lane detectors. LaneATT [30] is an anchor-based detector that considers straight lines as anchors, while RESA [34] is based on the semantic segmentation framework. Recently, CondLaneNet [19] was proposed, yielding an F-measure of 79.48% on CULane. LaneATT fails to detect highly curved lanes. RESA detects such curved lanes better than LaneATT does. However, for invisible lanes, it does not preserve their structures faithfully. CondLaneNet extracts some lanes precisely but fails to detect highly implied lanes. In contrast, the proposed algorithm detects both straight and curved lanes precisely and reliably, by generating structurally diverse lane candidates and then localizing lanes effectively in the eigenlane space.

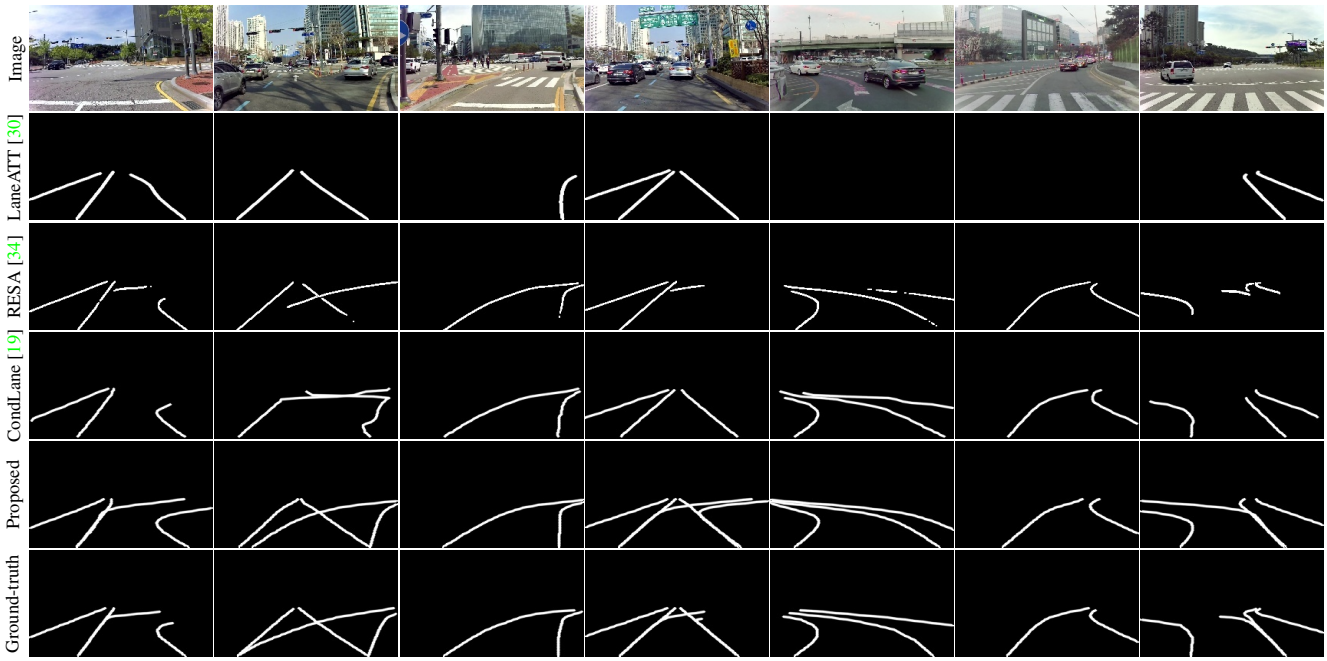


Figure 7. Comparison of lane detection results on the SDLane dataset.