

# Supplemental Materials on Harmonious Semantic Line Detection via Maximal Weight Clique Selection

Dongkwon Jin, Wonhui Park  
Korea University

dongkwonjin, whpark@mcl.korea.ac.kr

Seong-Gyun Jeong  
42dot.ai

seonggyun.jeong@42dot.ai

Chang-Su Kim  
Korea University

changasukim@korea.ac.kr

## A1. Implementation Details

### A1.1. Data configuration and loss functions

**S-Net:** S-Net computes classification probabilities  $P$  and regression offsets  $O$  of line candidates; for the  $i$ th line candidate  $\mathbf{l}_i = (\rho_i, \varphi_i)$ ,  $P_i$  indicates the probability that it is semantic, and  $O_i = \Delta \mathbf{l}_i = (\Delta \rho_i, \Delta \varphi_i)$  is the offset vector for line refinement. We configure training data for S-Net as follows. Let  $\mathbf{l}^* = (\rho^*, \varphi^*)$  be a ground-truth semantic line in an image. Since line candidates are generated by quantizing line parameters, the most overlapping line with the ground-truth  $\mathbf{l}^*$  is declared as semantic. If line candidate  $\mathbf{l}_i$  is declared to be semantic, the probability  $\bar{P}_i$  is annotated as 1 and the offset vector  $\bar{O}_i$  is annotated as  $(\rho^* - \rho_i, \varphi^* - \varphi_i)$ . Otherwise,  $\bar{P}_i$  and  $\bar{O}_i$  are annotated as 0 and  $(0, 0)$ , respectively. However, there are only a few semantic lines in each image. To alleviate the class imbalance, we disturb the line location of each semantic line and annotate the corresponding probability  $\bar{P}_i$  to be proportional to  $e^{-d_i^2}$ , where the disturbance  $d_i$  is defined as the positional distance between line  $i$  and the semantic line in the Hough space [10, 22]. The disturbance  $d_i$  is constrained to be less than 1.5. To train S-Net, we minimize the loss

$$\ell(P_i, \bar{P}_i, O_i, \bar{O}_i) = \ell_{\text{cls}}(P_i, \bar{P}_i) + \ell_{\text{reg}}(O_i, \bar{O}_i) \quad (1)$$

where  $P_i$  and  $O_i$  are the output of S-Net,  $\bar{P}_i$  and  $\bar{O}_i$  are the corresponding annotations,  $\ell_{\text{cls}}$  is the cross entropy loss over the two classes (semantic or not), and  $\ell_{\text{reg}}$  is the mean squared error (MSE).

**H-Net:** H-Net computes a harmony score between a pair of lines. The harmony score  $\bar{h}_{ij}$  is annotated to be proportional to  $e^{-(d_i^2 + d_j^2)}$  or 0, depending on whether the pair  $(i, j)$  is positive or not. Here,  $d_i$  and  $d_j$  denote the disturbances of lines  $i$  and  $j$ , respectively, and are constrained to be less than 3.0. However, if an image contains only a single ground-truth semantic line, there is no positive pair. Thus, we train H-Net with the pair  $(i, i)$ , which has a self-harmony score  $\bar{h}_{ii}$ . We annotate the self-harmony score  $\bar{h}_{ii}$  similarly to the line probability  $\bar{P}_i$ . Also, the loss function is defined as the MSE between  $h_{ij}$  and  $\bar{h}_{ij}$ .

### A1.2. Network details

**S-Net:** By employing the 13 convolution layers of VGG16 [26] as the backbone, we implement S-Net to exploit multi-scale features. Specifically, S-Net extracts two feature maps after Conv10 and Conv13, and then squeezes their channels using a convolution filter of size  $1 \times 1 \times 512 \times 32$ , respectively. Each squeezed map is used to extract a line feature map. Then, the two line feature maps are concatenated and fed into the fully connected layers for classification and regression. Thus, the sizes of fully connected layers are  $64 \times 1$  and  $64 \times 2$  for classification and regression, respectively.

**H-Net:** H-Net uses the same backbone as S-Net. H-Net also extracts multi-scale feature maps after Conv10 and Conv13, but does not perform the channel squeeze. From each single-scale feature map after Conv10 or Conv13, the line pooling layers extract local features of size  $512 \times 1$ . Also, the IRC feature of size  $256 \times 1$  is extracted using a fully connected layer of size  $2048 \times 256$ , which takes the concatenated regional feature vector  $R$  in Eq. (5) as input. Thus, in Figure 4(b), the sizes of Reg1 and Reg2 are  $256 \times 1$  and  $512 \times 1$ , respectively. Last, we yield the harmony score by averaging two harmony scores, which are computed separately from the two single-scale feature maps.







### A1.3. Learning rate and data augmentation

To train S-Net and H-Net, we set an initial learning rate to  $10^{-5}$  and halve it after every 60 epochs four times. Also, for each training image of size  $400 \times 400$ , we randomly flip it horizontally. We will make the source codes publicly available.

## A2. More Experimental Results

### A2.1. Comparison of metrics

As described in Section 4.2, the proposed HIoU metric assesses the overall harmony of semantic lines more accurately than the conventional metrics mIoU and EA-score. Figure A-1 shows semantic line detection results on four scenes. For each scene, the ground-truth and two detection results are provided. In result I, the position of each detected line is different from the ground-truth, but the detected lines convey the scene composition relatively well. In result II, two or three detected lines match the ground-truth exactly, but they are not harmonious with the remaining one. As a group, they are inferior to result I. Since mIoU and EA-score consider only the accuracy of each individual line, they do not differentiate these two results and provide only marginally different scores (except on the right two scenes, mIoU properly assigns a higher score to result I). In contrast, HIoU quantifies the superiority of result I correctly on all four scenes.

							
Ground-truth	Detection result I	Detection result II	Ground-truth	Detection result I	Detection result II		
<b>mIoU</b>	0.94	≈	0.92	<b>mIoU</b>	0.95	>	0.84
<b>EA-score</b>	0.81	≈	0.81	<b>EA-score</b>	0.86	≈	0.88
<b>HIoU</b>	0.77	>	0.64	<b>HIoU</b>	0.76	>	0.60







							
Ground-truth	Detection result I	Detection result II	Ground-truth	Detection result I	Detection result II		
<b>mIoU</b>	0.93	≈	0.91	<b>mIoU</b>	0.94	>	0.89
<b>EA-score</b>	0.84	≈	0.84	<b>EA-score</b>	0.79	<	0.84
<b>HIoU</b>	0.75	>	0.59	<b>HIoU</b>	0.76	>	0.59

Figure A-1: Comparison of mIoU [19], EA-score [10], and the proposed HIoU metric.

### A2.2. Model analysis

**Comparison of running times:** Semantic line detection is performed in two stages: line detection and refinement. In the line detection stage, deep line features are extracted to classify line candidates. In the line refinement stage, redundant lines are removed from the detected lines. Table A-1 compares the running times at each stage. We use a PC with an Intel Core i5-8500 CPU and an NVIDIA RTX 2080 ti GPU. In the detection stage, SLNet and DRM take a lot of time to extract discriminative line features, especially DRM to perform mirror attention. Compared to these algorithms, the proposed algorithm and DHT are much faster. In the line refinement stage, SLNet, DRM, and the proposed algorithm take 20~30 milliseconds to remove redundant lines based on NMS, pairwise comparison, and MWCS, respectively. On the other hand, DRM takes only about 1 millisecond, because it simply computes the centroid of each connected component in the probability map. Therefore, DHT is the fastest overall. However, the recall performance of DHT is not competitive as shown in Table 1 in the paper.

Table A-1: Analysis of running times of the proposed algorithm and the conventional algorithms. The processing times in seconds per frame (SPF) are reported.

	Detection	Refinement	Total
SLNet [19]	0.108s	0.028s	0.136s
DHT [10]	0.032s	0.001s	0.033s
DRM [16]	0.921s	0.031s	0.952s
Proposed	0.022s	0.024s	0.046s

**Robustness of H-Net and MWCS:** Table A-2 compares the AUC and HIoU scores according to the number  $K$  of selected lines in the selection-and-removal process. As more lines are selected, AUC\_P is lowered and AUC\_R is improved. However, note that AUC\_F and HIoU vary only slightly, as long as  $K \geq 6$ . These results that H-Net estimates edge weights between

nodes accurately and MWCS finds a maximal weight clique in a complete graph robustly, regardless of the number of nodes. Therefore, the overall performance is not sensitive to  $K$ .

Table A-2: AUC and HIoU scores (%) on the SEL dataset, according to the number  $K$  of selected lines.

$K$	4	6	8	10	12
AUC_P	89.90	89.90	89.61	89.54	89.51
AUC_R	82.09	83.89	84.21	84.15	84.42
AUC_F	85.82	86.80	86.83	86.76	86.89
HIoU	80.83	81.08	81.03	81.02	81.02

### A2.3. Comparative assessment

**Comparison on the CULane dataset:** We compare the proposed algorithm with the conventional road lane detectors [14, 25] on the CULane dataset [24]. CULane is a dataset for road lane detection, in which pixel-wise masks for up to 4 road lanes are provided for each image. To obtain ground-truth semantic lines corresponding to road lanes in an image, we determine the most overlapping line with the segmentation mask of each lane among line candidates and declare it as a semantic line. To this end, we first measure the edge score for each line candidate. The edge score is the ratio of segmented lane pixels over all pixels on the line candidate. Then, the line candidate with the highest edge score is regarded as the semantic line. Also, we filter out images containing curved lanes. Then, using the ground-truth semantic lines in the training images, S-Net and H-Net are trained, as described in Section A1. Meanwhile, the conventional techniques [14, 25] are based on the segmentation framework. For comparison, we also generate the semantic lines from the segmentation results of the conventional techniques. Figure A-2 shows some semantic lines generated from ground-truth mask and segmentation results. Figure A-3 compares detection results on the CULane dataset.

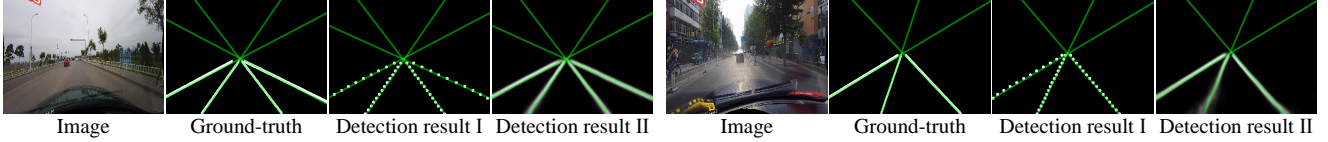


Figure A-2: Examples of generated semantic lines from the ground-truth binary mask and detection results on the CULane dataset: result I is the output of [25] and result II is the output of [14].

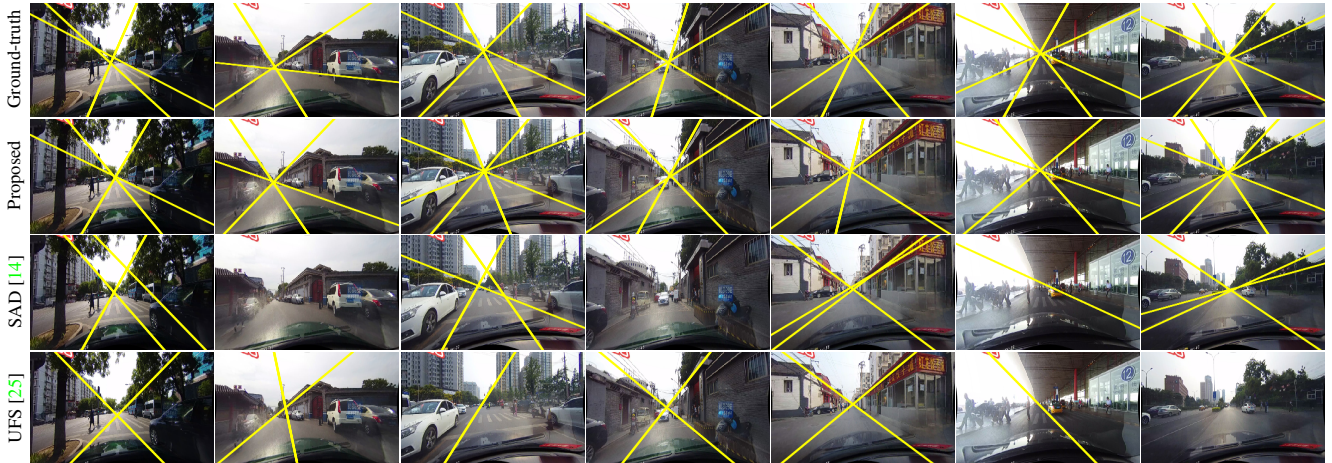


Figure A-3: Comparison of semantic line detection results on the CULane dataset ('no lane' category).



**Comparison of EA-scores on SEL and SEL\_Hard:** Table A-3 reports EA-scores on the SEL and SEL\_Hard datasets. The proposed algorithm provides poorer EA-scores (F-measure) than DRM or DHT. However, it does not imply that the proposed algorithm is inferior in terms of the overall harmony of detected lines. As shown in Figure A-4, the detected lines of the proposed algorithm are more harmonious even when those EA-scores (F-measure) are lower than those of DRM or DHT. Note that the EA-score metric focuses on measuring the positional accuracy of each detected line, and the scores may be higher although detected lines are less harmonious, as illustrated in Figure A-1. In Figure A-4, we also specify HIoU scores, which assess the harmony between the detected lines accurately. Figure A-5 compares more detection results on SEL and SEL\_Hard, which also confirm that the proposed algorithm detects harmonious semantic lines comparably to or more effectively than the existing methods.

Table A-3: Comparison of the EA-scores (%) on the SEL and SEL\_Hard datasets.

	SEL			SEL_Hard		
	Precision	Recall	F-measure	Precision	Recall	F-measure
SLNet [19]	85.75	84.05	84.87	78.63	67.57	72.62
DHT [10]	<b>89.85</b>	82.36	85.92	<b>84.94</b>	68.34	75.70
DRM [16]	85.40	<b>86.67</b>	<b>86.03</b>	82.76	<b>75.75</b>	<b>79.08</b>
Proposed	87.84	83.37	85.53	81.83	70.40	75.65

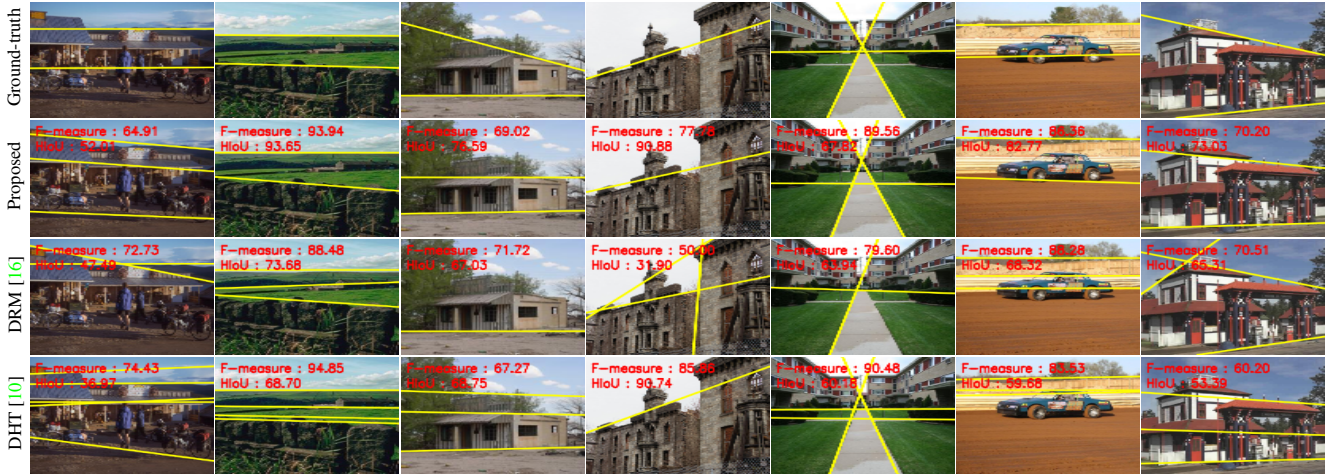


Figure A-4: Comparison of semantic line detection results on the SEL and SEL\_Hard datasets. For each result, the EA-score (F-measure) and HIoU score are specified.

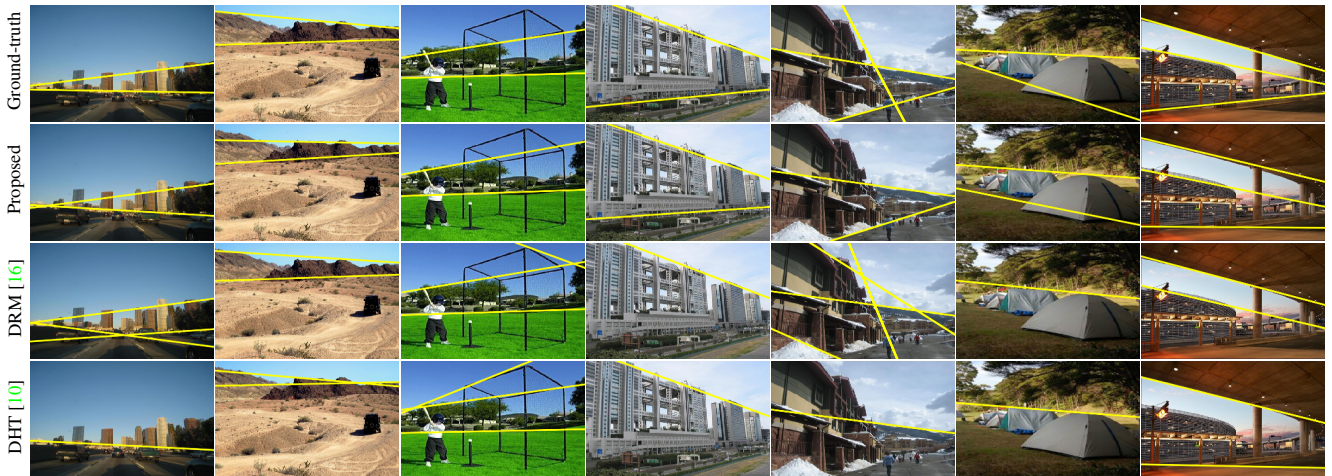


Figure A-5: Comparison of semantic line detection results on the SEL and SEL\_Hard datasets.



## A2.4. More detection results on SL5K

We provide more detection results of the proposed algorithm on the SL5K dataset.

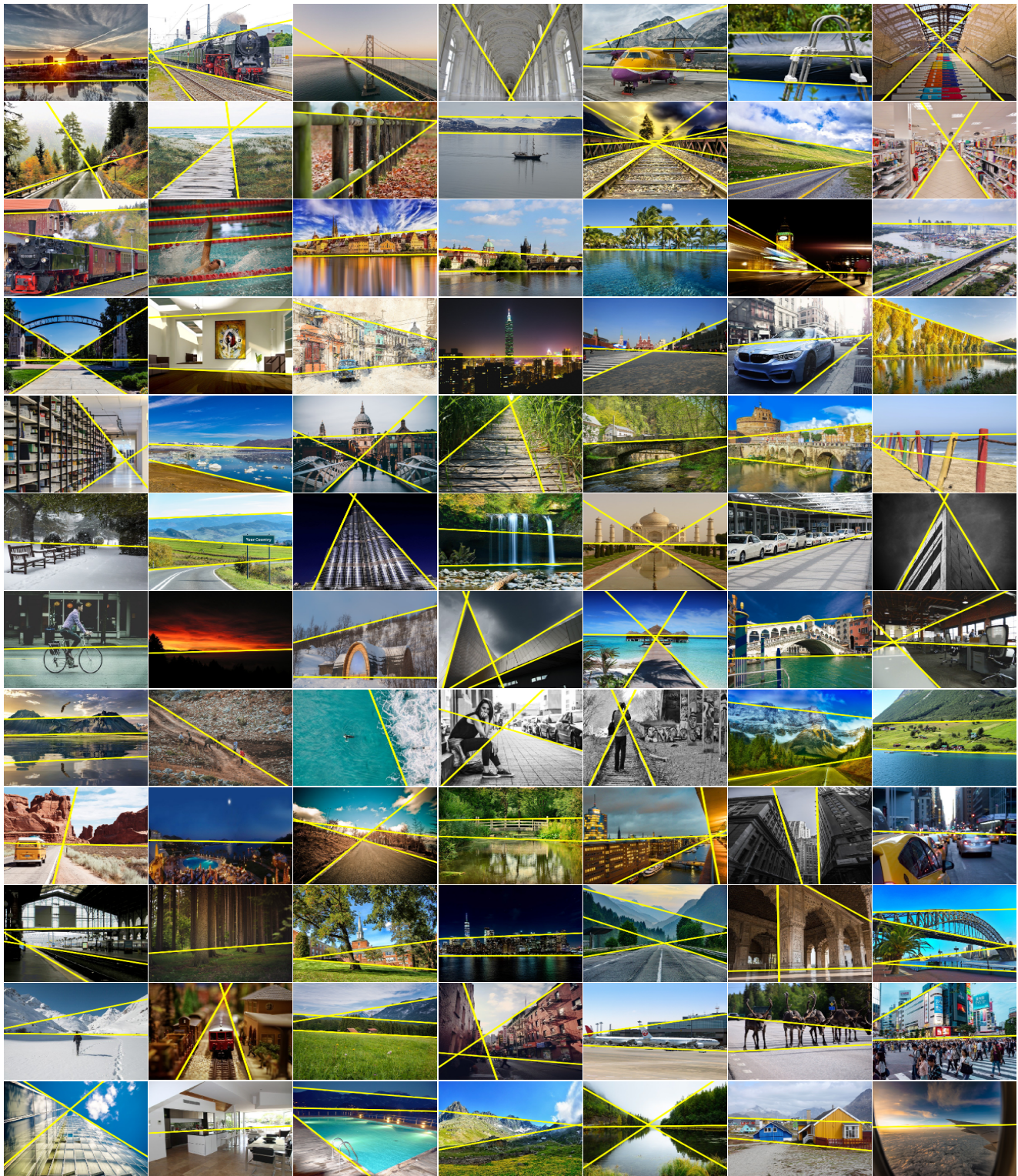


Figure A-6: Semantic line detection results of the proposed algorithm on the SL5K dataset.