CLEval: Character-Level Evaluation for Text Detection and Recognition Tasks – Supplementary Material –

Youngmin Baek[1], Daehyun Nam[1], Sungrae Park[1], Junyeop Lee[1],
Seung Shin[1], Jeonghun Baek[1], Chae Young Lee[2] and Hwalsuk Lee[*1]

[1]Clova AI Research, NAVER Corp.
[2]Yale University

A. Toy-example experiment on ICDAR2015.

To show the stability of our metric, we additionally performed experiments on ICDAR2015 dataset. For detection evaluation, toy-set was produced in the same way as it was made using the ICDAR2013 dataset, and for end-to-end evaluation we constructed another set based on detection toy-examples. Note that we evaluated the end-to-end result using the best model reported in [1]. The result of the experiment and their attributes from CLEval are shown in table 1,2, and the line graphs in Figure 1 show results performed under different conditions.

The results on ICDAR2015 dataset show similar tendency when compared with the evalution results on ICDAR2013 dataset. Due to the trait of the IoU metric using a threshold value of 0.5, the metric assigns zero score to the cropped area less than 50 percent. Also, the zero scores on split cases are caused by the absence of handling granularity issues. One-to-many or many-to-one match cases frequently occur, but the IoU metric only considers one-to-one matching cases. Multiple box predictions could cover a single ground truth box, but zero scores are given if the overlapping region does not meet a predefined threshold.

The same holds for the IoU+CRW metric on end-to-end evaluation. Using a predefined threshold, a one-to-one match is first made to filter out valid box candidates, then CRW is performed to identify matching transcripts. In the transcript matching process, CRW requires ground truth and predicted text to be matched perfectly. Otherwise, a zero score is assigned to the matched box candidates. For this reason, we observe meaningful comparison was difficult with the IoU+CRW.

The proposed metric provides stable scores under various cases by performing evaluations at the character-level. Table 2 shows recall, precision scores of partially corrected detections.

Case	Attributes from CLEval				
	instance-level		character-level		
	Split	Merge	Miss	Overlap	FP
Original	18	15	5	45	0
Crop 80%	11	10	1723	26	0
Crop 60%	9	9	4592	17	0
Crop 40%	8	8	6246	10	0
Split by 2	1990	18	0	38	88
Split by 3	1988	19	0	36	178
Split by 4	1994	21	0	35	633
Overlap 10%	2073	21	0	1574	1
Overlap 20%	2074	23	0	2431	0
Overlap 30%	2074	23	0	4157	0

Table 1: Detection attributes from CLEval on ICDAR2015 dataset.

B. Evaluation of text detectors

In this section, we compare CLEval with other commonly used evaluation metrics using the state-of-the-art text detectors. We requested the authors of various scene text detectors to provide their test results on public datasets and organized the results in Table 3, 4 for ICDAR2013 and ICDAR2015, respectively.

The strength of using the CLEval metric is in its use of additional instance-level and character-level information to calculate recall, precision, and hmean values. As shown in Table 3, 4, even without the knowledge of recall, precision, and hmean values, we could examine the quality of the detection models by observing the attributes produced by the CLEval metric.

C. Evaluation on real end-to-end results

In this experiment, we take a close look into the end-to-end performance of various detector and recognizer combinations. We used the well-known detectors such

*Corresponding author.

Case	Detection Metrics									E2E Metrics					
	DetEval*			IoU			CLEval			IoU+CRW			CLEval		
	R	P	H	R	P	H	R	P	H	R	P	H	R	P	H
Original	98.1	99.9	99.0	100	100	100	99.8	99.4	99.6	72.4	72.4	72.4	88.6	91.1	89.8
Crop 80%	98.0	98.1	98.0	100	100	100	84.4	99.6	91.4	50.7	50.7	50.7	80.6	87.7	84.0
Crop 60%	0.7	1.8	1.0	100	100	100	58.6	99.6	73.8	8.0	8.0	8.0	54.8	77.0	64.0
Crop 40%	0.0	0.1	0.1	0.0	0.0	0.0	43.7	99.6	60.8	0.0	0.0	0.0	35.2	69.4	46.7
Split by 2	69.8	74.1	71.9	97.9	49.0	65.3	81.9	98.7	89.5	0.1	0.1	0.1	63.4	76.6	69.4
Split by 3	65.9	70.3	68.0	0.0	0.0	0.0	64.0	97.9	77.4	0.0	0.0	0.0	38.4	62.3	47.5
Split by 4	65.2	69.3	67.2	0.0	0.0	0.0	49.2	94.1	64.6	0.0	0.0	0.0	15.2	48.1	23.1
Overlap 10%	73.9	78.2	76.0	100	50.0	66.7	81.1	87.4	84.1	0.1	0.1	0.1	66.3	73.6	69.8
Overlap 20%	74.3	78.6	76.4	100	50.0	66.7	81.1	81.9	81.5	0.7	0.3	0.4	68.3	69.4	68.8
Overlap 30%	74.8	78.8	76.8	100	50.0	66.7	81.1	72.6	76.6	1.1	0.5	0.7	69.4.7	65.2	67.2

Table 2: Comparison of evaluation metrics on toy-set from ICDAR2015 dataset. Some scores are highlighted: Red above 95, Blue below 5. *denotes our implemented code since official evaluation does not exist.

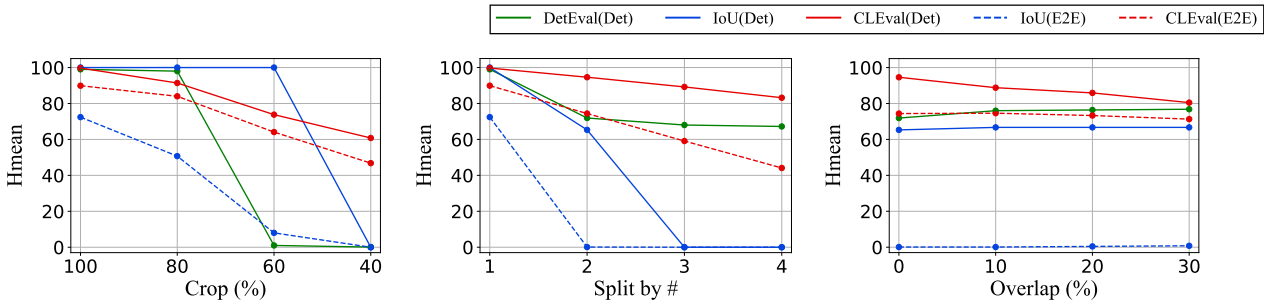

Figure 1: Line graph for H-mean of each evaluation metric according to different crop, split, overlap condition. Solid line indicates the detection evaluation while dashed line indicates the end-to-end evaluation results.

Detector	Metrics						Attributes				
	IoU			CLEval			instance-level		character-level		
	R	P	H	R	P	H	Split	Merge	Miss	Overlap	FP
CTPN [10]	83.0	93.0	87.7	82.5	84.1	83.3	14	231	1015	56	411
SegLink [9]	60.0	73.9	66.2	74.0	95.4	83.3	93	28	1293	46	116
EAST [11]	70.7	81.6	75.8	84.7	94.2	89.2	52	47	827	51	200
RRPN [8]	87.3	95.2	91.1	90.2	95.3	92.7	43	35	521	75	140
TextBoxes++ [4]	85.6	91.9	88.6	92.7	94.1	93.4	26	29	374	41	246
FOTS [6]	90.4	95.4	92.8	94.0	96.4	95.2	57	34	289	99	62
MaskTextSpotter [7]	88.6	95.0	91.7	93.9	97.7	95.7	26	23	325	24	69
CRAFT [2]	93.1	97.4	95.2	96.3	96.6	96.4	34	37	177	84	60
PMTD [5]	92.2	95.1	93.6	96.1	97.6	96.8	24	28	199	28	76

Table 3: Comparison of evaluation metrics & additional detection attributes for different detectors on **ICDAR2013** dataset. R, P, and H refer to recall, precision, and H-mean. Detectors are sorted from the highest score on DetEval metric.

Detector	Metrics						Attributes				
	IoU			CLEval			instance-level		character-level		
	R	P	H	R	P	H	Split	Merge	Miss	Overlap	FP
CTPN [10]	51.6	74.2	60.9	63.2	93.6	75.4	75	103	3842	31	302
SegLink [9]	72.9	80.2	76.4	79.4	95.1	86.5	130	117	2123	107	224
RRPN [8]	77.1	83.5	80.2	81.8	94.6	87.7	60	77	1938	49	377
EAST [11]	77.2	84.6	80.8	84.8	93.9	89.1	66	124	1607	65	401
TextBoxes++ [4]	80.8	89.1	84.8	84.2	95.0	89.3	35	52	1713	33	397
MaskTextSpotter [7]	79.5	89.0	84.0	83.7	96.2	89.5	52	63	1751	62	224
FOTS [6]	87.9	91.9	89.8	90.0	97.1	93.4	74	69	1033	67	160
CRAFT [2]	84.3	89.8	86.9	90.0	97.4	93.5	33	73	1076	19	171
PMTD [5]	87.4	91.3	89.3	90.8	97.0	93.8	38	47	982	33	232

Table 4: Comparison of evaluation metrics & additional detection attributes for different detectors on **ICDAR2015** dataset. R, P, and H refer to recall, precision, and H-mean. Detectors are sorted from the highest score on IoU metric.

as CRAFT[2], EAST[11], RRPN[8], PixelLink[3], and TextBoxes++[4]. We recognized the texts of those detectors with three types of recognizers provided in [1]. CLEval results are listed in the Table 5. *High* indicates recognizer with TPS+ResNet+BiLSTM+Attn moduels, *Mid* indicates recognizer with None+VGG+BiLSTM+CTC modules, and *Low* indicates recognizer with None+VGG+None+CTC modules. We observe that RS scores in each *High*, *Mid*, and *Low* recognition combination are similar. This infers that RS can be used to evaluate recognition performance regardless of the detection module.

D. PCC generation in polygon annotation

Most of the text bounding boxes in public datasets are represented using four quadrilateral points. However, there exist polygon-type datasets that use multiple vertexes to tightly bound the text regions. For polygon datasets, we could acquire the center information by splitting the polygon into a sub-groups of quadrilaterals. Algorithm 1 describes the detailed procedure for generating PCCs in polygon-type dataset. By extending PCC generation to polygon datasets, CLEval can be used to evaluate on a variety of datasets represented by both rectangles and polygons.

Algorithm 1: PCC generation process from polygon annotation

$\{\boldsymbol{p_1}, \boldsymbol{p_2}, \ldots, \boldsymbol{p_{2n}}\} \leftarrow$ a set of even-number annotation points from left-top, clockwise order
$\boldsymbol{LEN_{transcription}} \leftarrow$ length of transcription
def *PCC_polygon(Points, Length):*
  initialize *PCCs* as array
  $PTS_{top} \leftarrow$ a set of points above center $(= \{Points_1, \ldots, Points_n\})$
  $PTS_{bottom} \leftarrow$ a set of points below center $(= \{Points_{n+1}, \ldots, Points_{2n}\})$
  $CharSize = len(PTS_{top}) - 1$
  # to make order from left to right, reverse order of element
  $reverseOrder(PTS_{bottom})$
  NEW_{top}, NEW_{bottom} = Interpolate(PTS_{top}, PTS_{bottom}, $Length$)
  for *k* **from** 1 **to** L_{trans}
    $char_{tl} = New_{top}^{CharSize \times (k-1)+1}$
    $char_{tr} = New_{top}^{CharSize \times (k)+1}$
    $char_{bl} = New_{bottom}^{CharSize \times (k-1)+1}$
    $char_{br} = New_{bottom}^{CharSize \times (k)+1}$
    PCCs.append($mean(char_{tl}, char_{tr}, char_{bl}, char_{br})$)
  end
  return PCCs
end
def *Interpolate(P_{top}, P_{bottom}, L_{trans}):*
  initialize New_{top}, New_{bottom} as array
  for *i* **from** 1 **to** $len(P_{top}) - 1$
    p^{tl}, p^{tr} =ith, $(i+1)^{th}$ point of P_{top}
    p^{bl}, p^{br} =ith, $(i+1)^{th}$ point of P_{bottom}
    New_{top}.append(p^{tl})
    New_{bottom}.append(p^{bl})
    for *k* **from** 1 **to** $L_{trans} - 1$
      $n_{top} = \left(1 - \frac{k}{L_{trans}}\right) p^{tl} + \left(\frac{k}{L_{trans}}\right) p^{tr}$
      $n_{bottom} = \left(1 - \frac{k}{L_{trans}}\right) p^{bl} + \left(\frac{k}{L_{trans}}\right) p^{br}$
      New_{top}.append(n_{top})
      New_{bottom}.append(n_{bottom})
    end
  end
  New_{top}.append(last element of P_{top})
  New_{bottom}.append(last element of P_{bottom})
  return New_{top}, New_{bottom}
end

References

[1] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. *arXiv preprint arXiv:1904.01906*, 2019. 1, 3

[2] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *CVPR*, pages 4321–4330. IEEE, 2019. 2, 3, 4

[3] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. Pixellink: Detecting scene text via instance segmentation. In *AAAI*, 2018. 3, 4

[4] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *Image Processing*, 27(8):3676–3690, 2018. 2, 3, 4

[5] Jingchao Liu, Xuebo Liu, Jie Sheng, Ding Liang, Xin Li, and Qingjie Liu. Pyramid mask text detector. *arXiv preprint arXiv:1903.11800*, 2019. 2

[6] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *CVPR*, pages 5676–5685, 2018. 2

[7] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *arXiv preprint arXiv:1807.02242*, 2018. 2

[8] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 20(11):3111–3122, 2018. 2, 3, 4

[9] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *CVPR*, pages 3482–3490. IEEE, 2017. 2

[10] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *ECCV*, pages 56–72. Springer, 2016. 2

[11] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *CVPR*, pages 2642–2651, 2017. 2, 3, 4

Detector	CLEval Det			Recognizer	CLEval E2E			E2E Rec
	R	P	H		R	P	H	RS
RRPN[8]	81.8	94.6	87.7	High	76.7	84.3	79.9	89.0
				Mid	74.0	82.9	78.2	86.4
				Low	70.4	82.4	75.9	83.0
EAST[11]	84.8	93.9	89.1	High	78.4	83.7	81.0	88.4
				Mid	75.2	83.6	79.2	85.5
				Low	72.0	82.5	76.9	82.2
TextBoxes++[4]	84.2	95.0	89.3	High	78.1	86.4	82.0	90.0
				Mid	72.2	84.0	77.6	84.0
				Low	67.8	82.8	74.6	79.0
PixelLink[3]	89.0	96.7	92.7	High	80.8	88.4	84.5	87.8
				Mid	78.1	87.3	82.5	85.3
				Low	73.8	86.1	79.4	81.0
CRAFT[2]	89.9	97.3	93.5	High	81.6	88.9	85.1	88.2
				Mid	78.4	87.3	82.6	85.1
				Low	74.4	86.3	79.9	81.1

Table 5: End-to-end evaluation using CLEval for state-of-the art text detectors and recognizers.