Character Region Awareness for Text Detection

CVPR 2019

Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee∗ Clova AI Research, NAVER Corp.

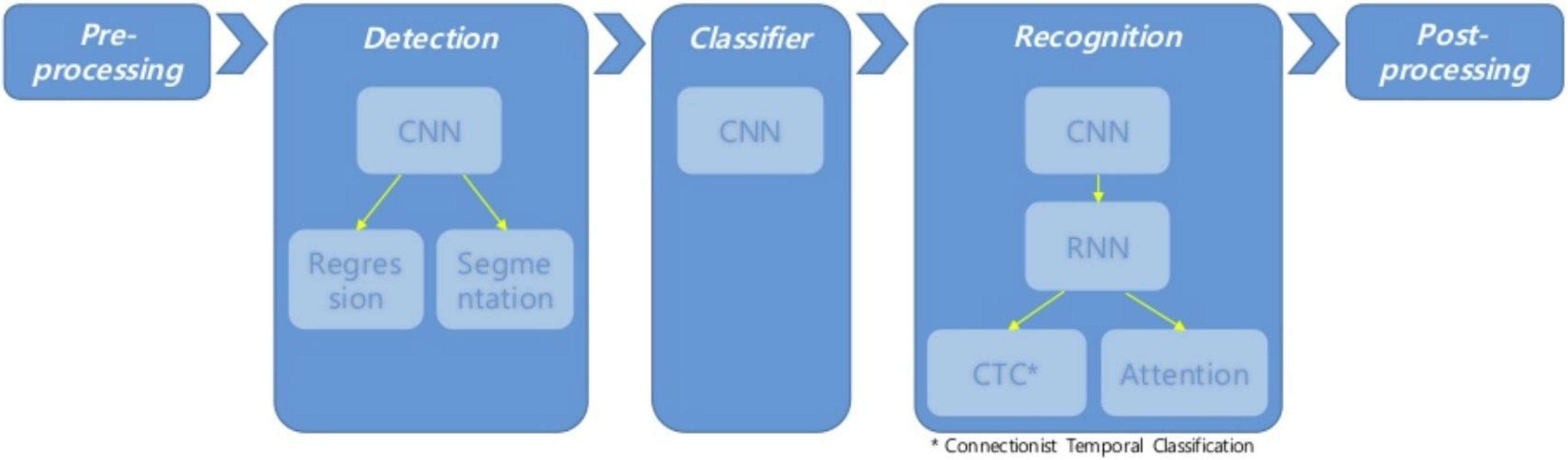
Bochan Kim

kmc2048@gmail.com

2020. 04. 06 Mon

Before Start..

- OCR Pipeline (with DL)

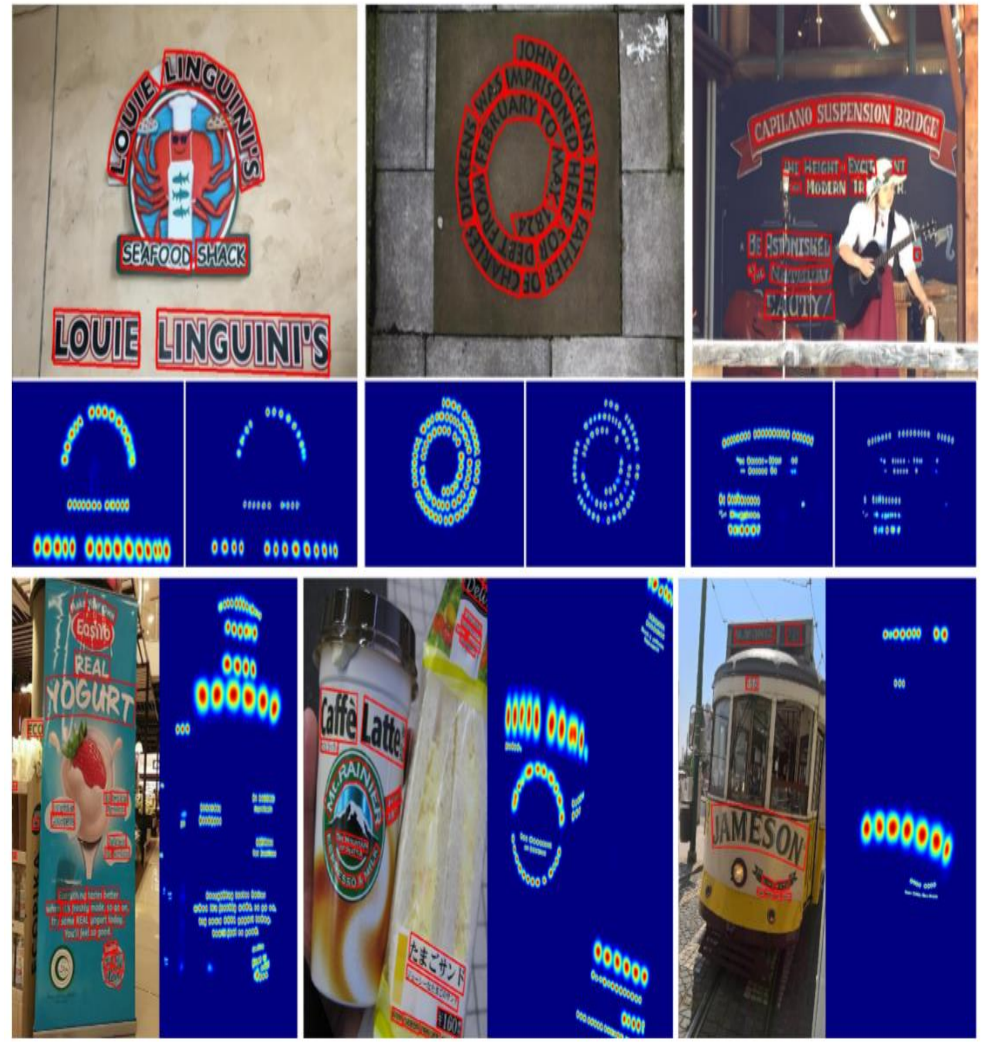


- Detection ✔

- Recognition

Results


Horizontal
Curved
Arbitrary
RIVERSIDE WALK
CROCODILES
NO SWIMMING
THE KEEN
ANKYLOSAURUS
CREATION MUSEUM
EXIT 11

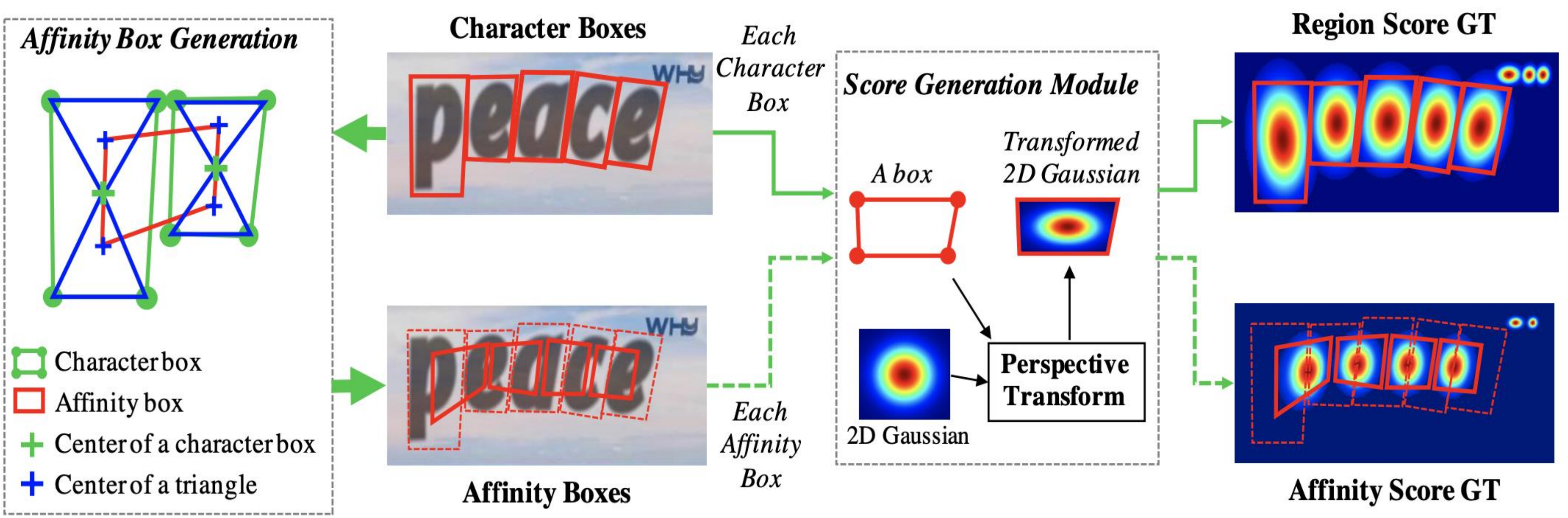

LOUIE LINGUINI'S
SEAFOOD SHACK
LOUIE LINGUINI'S
CAPILANO SUSPENSION BRIDGE
REAL
YOGURT
Caffè Latte
JAMESON

Main Idea

Region score + Affinity Score

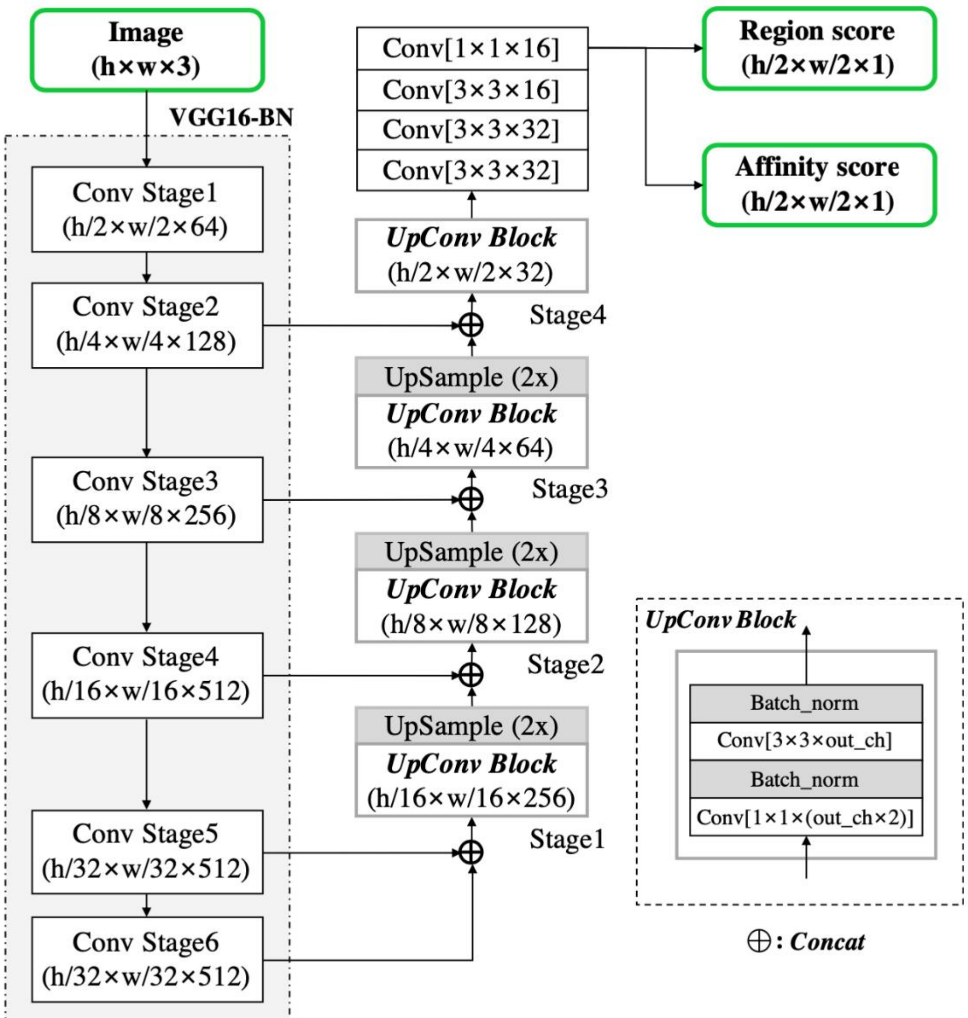
Main Idea

Ground Truth Label Generation (if character-level annotation exists)



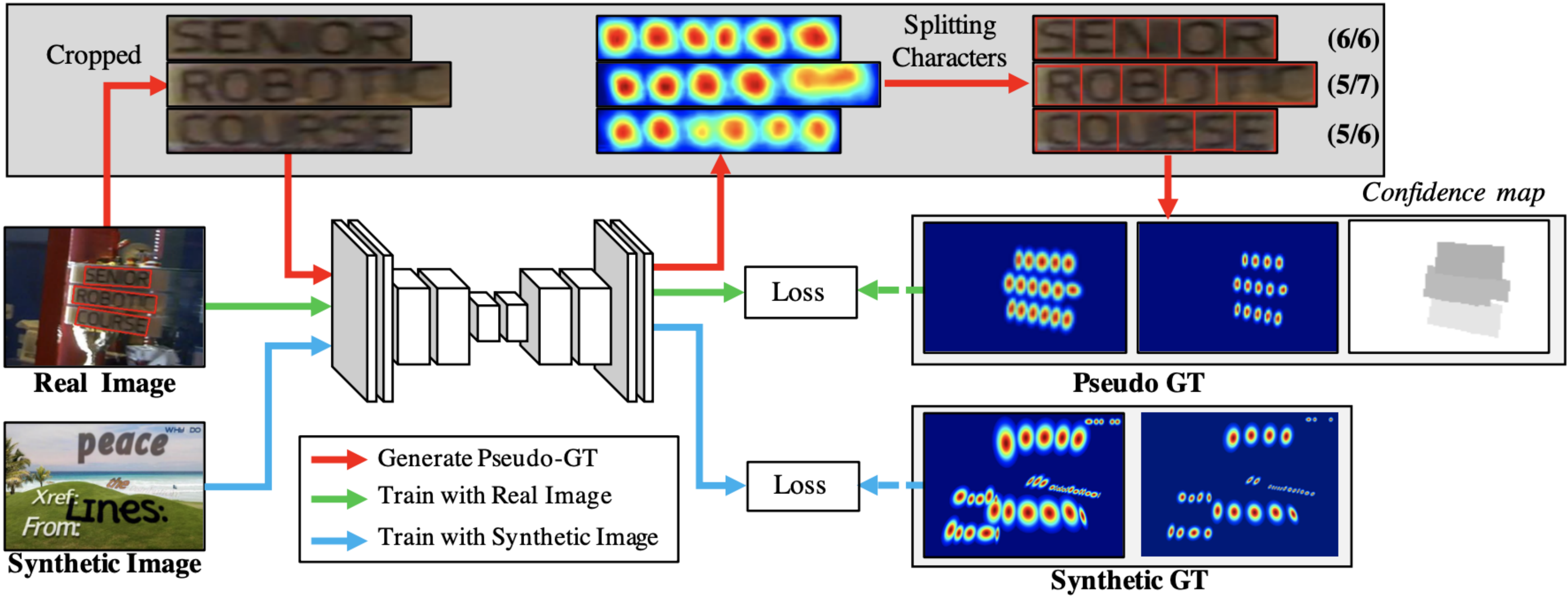
1. Using character boxes(from annotation), generate affinity boxes
2. Perspective transform 2D isotropic Gaussian map to region and affi nity boxes
3. Map 2D Gaussian to region and affinity score

Architecture



1. Backbone as VGG-16
2. U-net shape decoding part for aggregating low-level features
3. Output has two channels (region score, affinity score)

Weakly Supervised Learning



1. Pretrain a model (using SynthText in paper)
2. Crop the word in real image using word-level annotation
3. Predict the region score
4. Split the character regions using watershed algorithm
5. Comparing the length of predicted words, calculate confidence.

Objective Function (with weakly-supervised learning)

Calculating confidecne

$$s_{conf}(w) = \frac{l(w) - \min(l(w), |l(w) - l^c(w)|)}{l(w)}$$

- $l(w)$: length of sample word w
- $l^c(w)$: estimated length of sample word w

- If confidence is less than 0.5, just assume that word is evenly separated and set confidence to 0.5

- This is for learning unseen appearances of texts.

Objective Function (with weakly-supervised learning)

$$S_c(p) = \begin{cases} s_{conf}(w) & p \in R(w), \\ 1 & \text{otherwise}, \end{cases} \tag{2}$$

where p denotes the pixel in the region $R(w)$. The objective L is defined as,

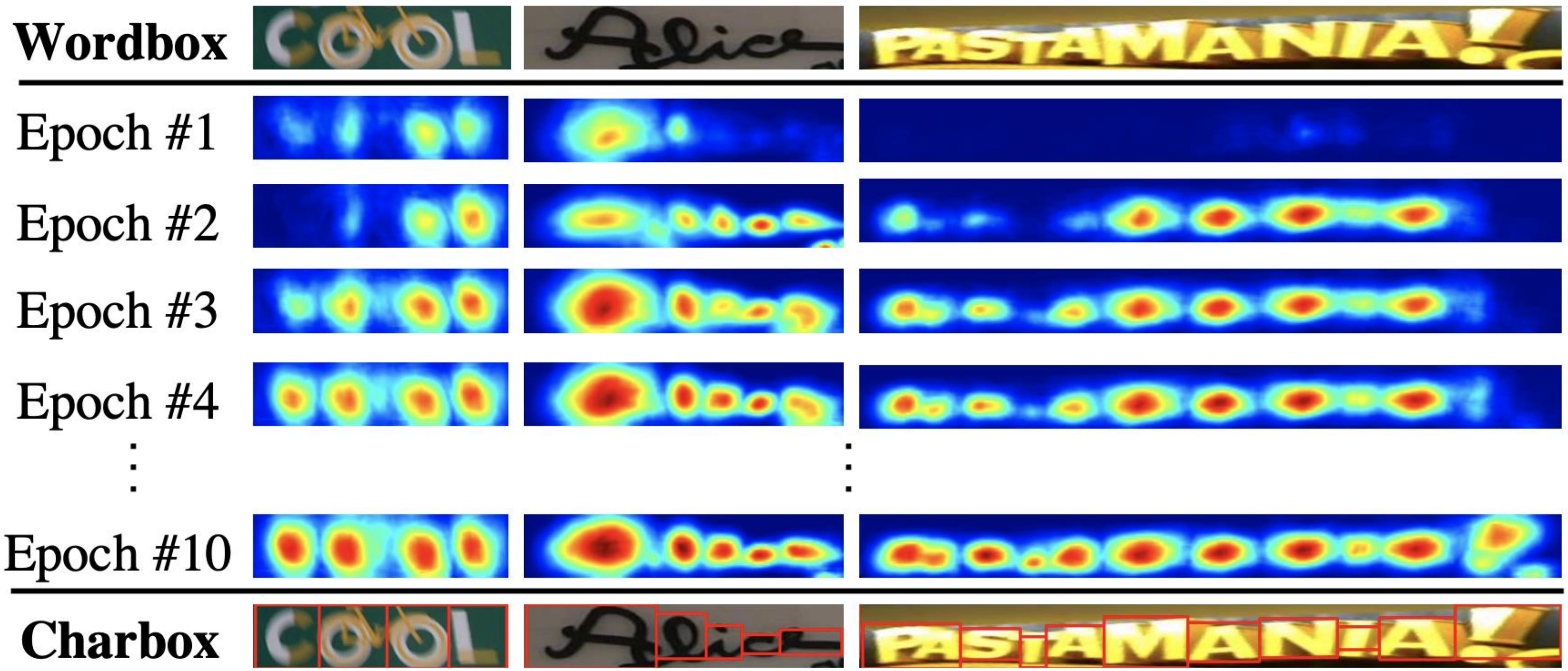
$$L = \sum_p S_c(p) \cdot \left(||S_r(p) - S_r^*(p)||_2^2 + ||S_a(p) - S_a^*(p)||_2^2 \right),$$

- R(w) : bounding box region of word w
- $S_r(p)$: region score for pixel p
- $S_a(p)$: affinity score for pixel p

- When training with synthetic data, $S_c(p)$ is set to 1.

- During training real data, synthetic data is still for the exact char-level annotation (synth : real = 1: 5)

Character region score maps during training

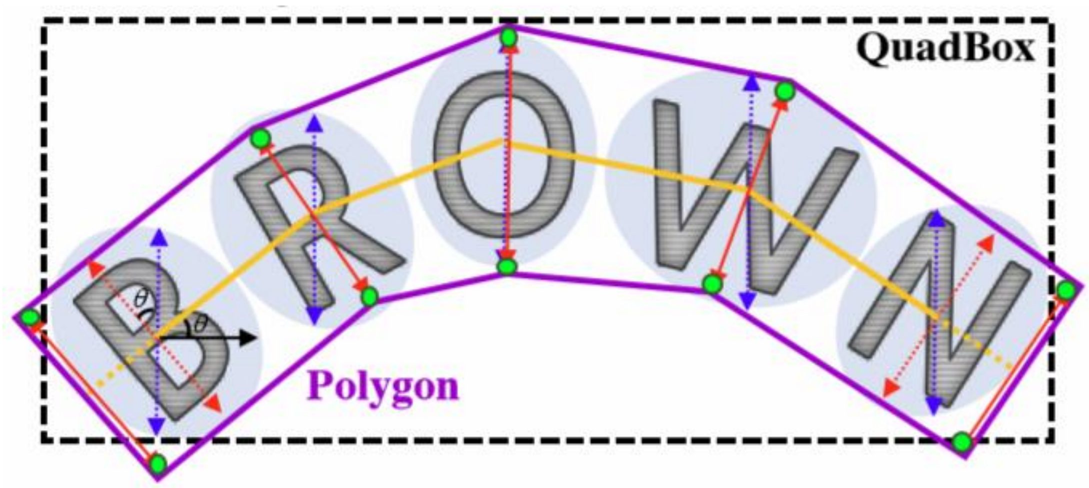


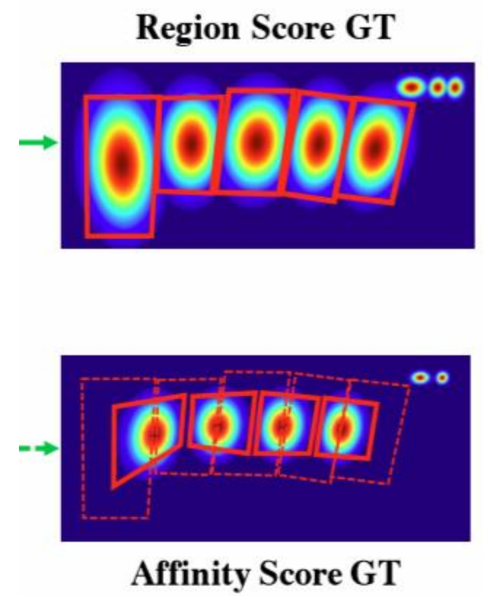
- As training is performed, model can predict characters more accurately, and the confidence scores are gradually increased as well

- The model learns the appearances of new texts, such as irregular fonts, and synthesized texts that have a different data distribution against that of the SynthText dataset.

Post-processing

1. For QuadBox Generation





1. Initialize binary map M with 0 and set 1 if $S_r(p) > \tau_r$ or $S_a(p) > \tau_a$
2. Connected Component Labeling(CCL) on M (cv2.connectedComponents)
3. Find a rectangle of minimum area enclosing the connected components (cv2.minAreaRect)

Post-processing

2. For Polygon Generation

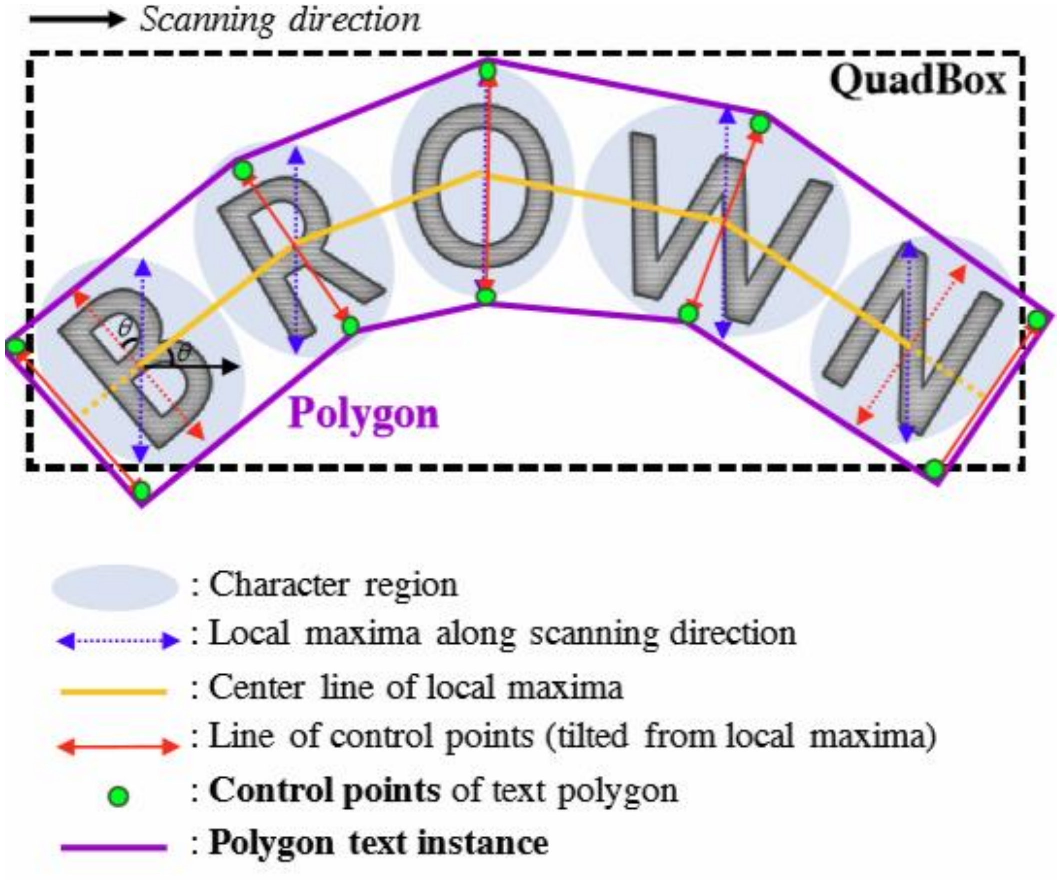


Figure 7. Polygon generation for arbitrarily-shaped texts.

1. Blue line
2. Yellow line
3. Red line
4. Green dots

Experiments

1. Results on quadrilateral-type dataset

Method	IC13(DetEval)			IC15			IC17			MSRA-TD500			FPS
	R	P	H	R	P	H	R	P	H	R	P	H	
Zhang et al. [39]	78	88	83	43	71	54	-	-	-	67	83	74	0.48
Yao et al. [37]	80.2	88.8	84.3	58.7	72.3	64.8	-	-	-	75.3	76.5	75.9	1.61
SegLink [32]	83.0	87.7	85.3	76.8	73.1	75.0	-	-	-	70	86	77	20.6
SSTD [8]	86	89	88	73	80	77	-	-	-	-	-	-	7.7
Wordsup [12]	87.5	93.3	90.3	77.0	79.3	78.2	-	-	-	-	-	-	1.9
EAST* [40]	-	-	-	78.3	83.3	80.7	-	-	-	67.4	87.3	76.1	13.2
He et al. [11]	81	92	86	80	82	81	-	-	-	70	77	74	1.1
R2CNN [13]	82.6	93.6	87.7	79.7	85.6	82.5	-	-	-	-	-	-	0.4
TextSnake [24]	-	-	-	80.4	84.9	82.6	-	-	-	73.9	83.2	78.3	1.1
TextBoxes++* [17]	86	92	89	78.5	87.8	82.9	-	-	-	-	-	-	2.3
EAA [10]	*87*	*88*	*88*	*83*	*84*	*83*	-	-	-	-	-	-	-
Mask TextSpotter [25]	*88.1*	*94.1*	*91.0*	*81.2*	*85.8*	*83.4*	-	-	-	-	-	-	4.8
PixelLink* [4]	87.5	88.6	88.1	82.0	85.5	83.7	-	-	-	73.2	83.0	77.8	3.0
RRD* [19]	86	92	89	80.0	88.0	83.8	-	-	-	73	87	79	10
Lyu et al.* [26]	84.4	92.0	88.0	79.7	89.5	84.3	**70.6**	74.3	72.4	76.2	87.6	81.5	5.7
FOTS [21]	-	-	*87.3*	*82.0*	*88.8*	*85.3*	*57.5*	*79.5*	*66.7*	-	-	-	23.9
CRAFT(ours)	**93.1**	**97.4**	**95.2**	**84.3**	**89.8**	**86.9**	68.2	**80.6**	**73.9**	**78.2**	**88.2**	**82.9**	8.6

Table 1. Results on quadrilateral-type datasets, such as ICDAR and MSRA-TD500. * denote the results based on multi-scale tests. Methods in *italic* are results solely from the detection of end-to-end models for a fair comparison. R, P, and H refer to recall, precision and H-mean,

Experiments

2. Results on polygon-type dataset

Method	TotalText			CTW-1500		
	R	**P**	**H**	**R**	**P**	**H**
CTD+TLOC [38]	-	-	-	69.8	77.4	73.4
MaskSpotter [25]	55.0	69.0	61.3	-	-	-
TextSnake [24]	74.5	82.7	78.4	**85.3**	67.9	75.6
CRAFT(ours)	**79.9**	**87.6**	**83.6**	81.1	**86.0**	**83.5**

Table 2. Results on polygon-type datasets, such as TotalText and CTW-1500. R, P and H refer to recall, precision and H-mean, respectively. The best score is highlighted in **bold**.

Discussions

1. Robustness to Scale Variance
   - Solely performed single scale experiments on all the datasets

2. Multi-language issue
   - Some language which is difficult to separate words into characters exists such as Bangla, Arabic

3. Generalization ability
   - Achieved SOTA performances on 3 different datasets without additional fine-tuning

4. Comparison with End-to-end methods
   - Though trained only with detection GT boxes, still very comparable with other end-to-end methods

Method	IC13	IC15	IC17
Mask TextSpotter [25]	91.7	86.0	-
EAA [10]	90	87	-
FOTS [21]	92.8	**89.8**	70.8
CRAFT(ours)	**95.2**	86.9	**73.9**

Q & A

감사합니다.