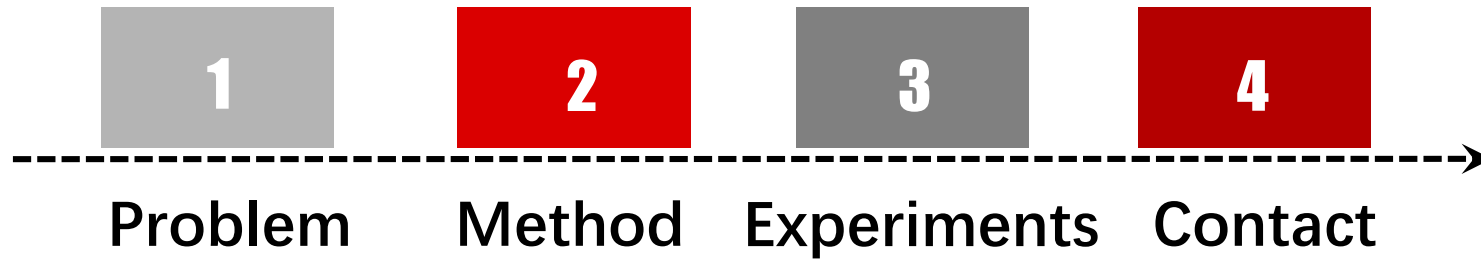


CONTENT



1. Problem

Text Spotting Pipeline



Detection



Crop



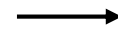
ROLAND

Recognition

Text Spotting Pipeline



Detection



Crop



ROLAND

Recognition

End-to-End Text Spotting: Global optimization/ Reduce error accumulation/ maintenance cost



Detection



- RoIAlign + TPS [1][2]
- RoIMasking [3]
- RoISlide [4]
- BezierAlign [5]
- ...



Recognition

ANTIQUE

Text Spotting Pipeline



Detection



Crop



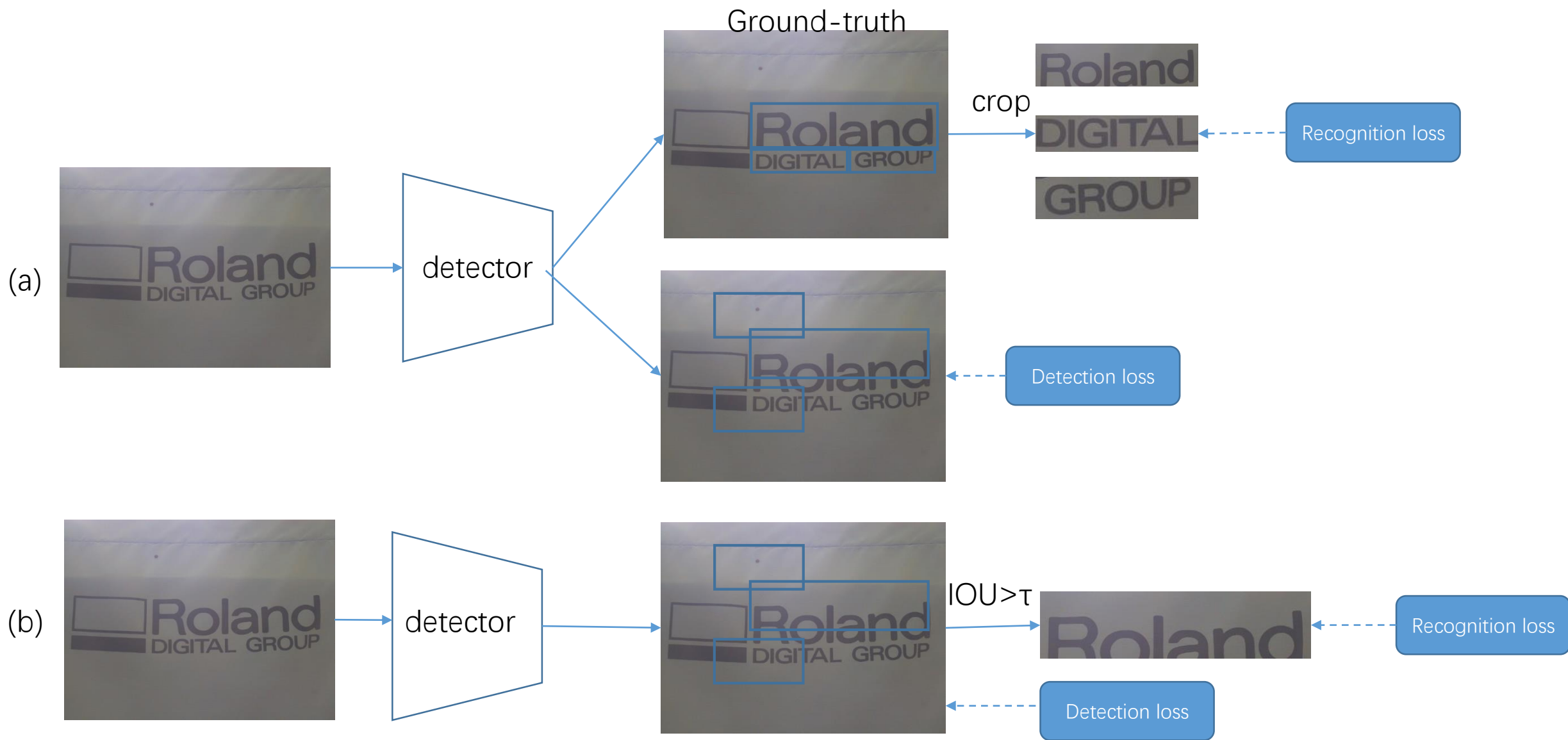
ROLAND

Recognition

End-to-End Text Spotting: Global optimization/ Reduce error accumulation/ maintenance cost



弯曲文本检测+差值采样



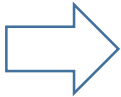
Problem: Recognition relies on detection — accurate boundaries required

Detection Annotations



Problem: Recognition relies on detection — accurate boundaries required

Detection Annotations



recognition

“WOODFORD”
“RESERVE”
“DISTILLERY”
...

Trade of expansion



2. Method

If there is only one text ...

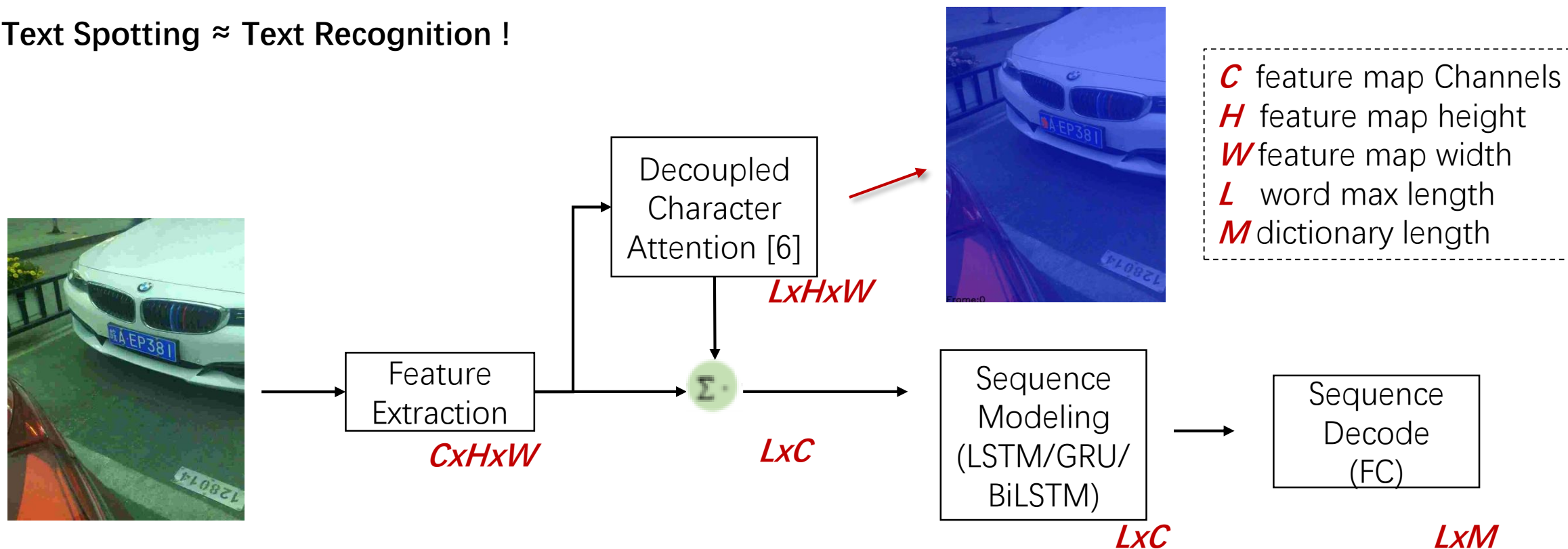
Industrial OCR scene (Industrial Printing Recognition / Meter Dial Recognition/ Plate License Recognition ...)

Text Spotting \approx Text Recognition !

If there is only one text ...

Industrial OCR scene (Industrial Printing Recognition / Meter Dial Recognition / Plate License Recognition ...)

Text Spotting \approx Text Recognition !



The workflow of single-text-based text spotting

When it comes to multiple text...

Q: How to define the output order (one-to-one matching) ?

When it comes to multiple text...

Q: How to define the output order (one-to-one matching) ?



- ① "WOODFORD"
- ② "RESERVE"
- ③ "DISTILLERY"
- ...

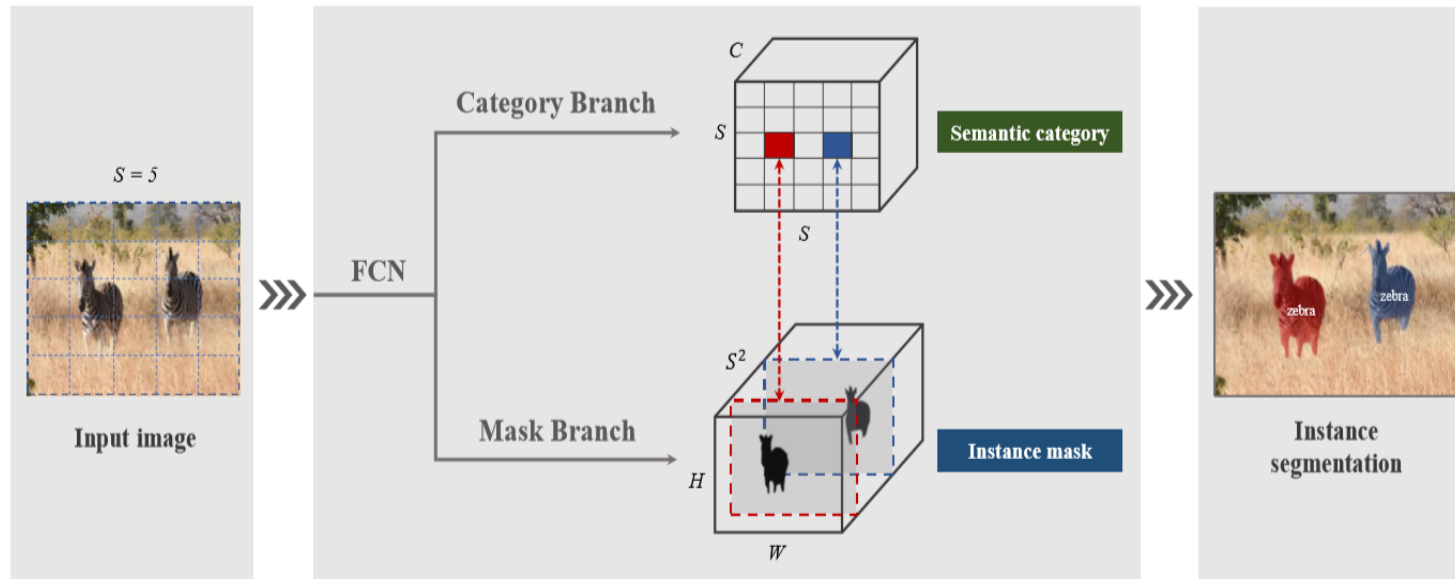
When it comes to multiple text...

Q: How to define the output order (one-to-one matching) ?



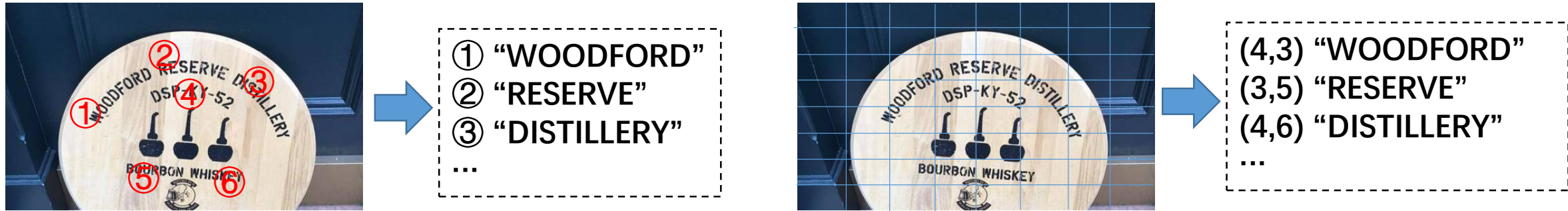
- ① "WOODFORD"
- ② "RESERVE"
- ③ "DISTILLERY"
- ...

SOLO [7][8]

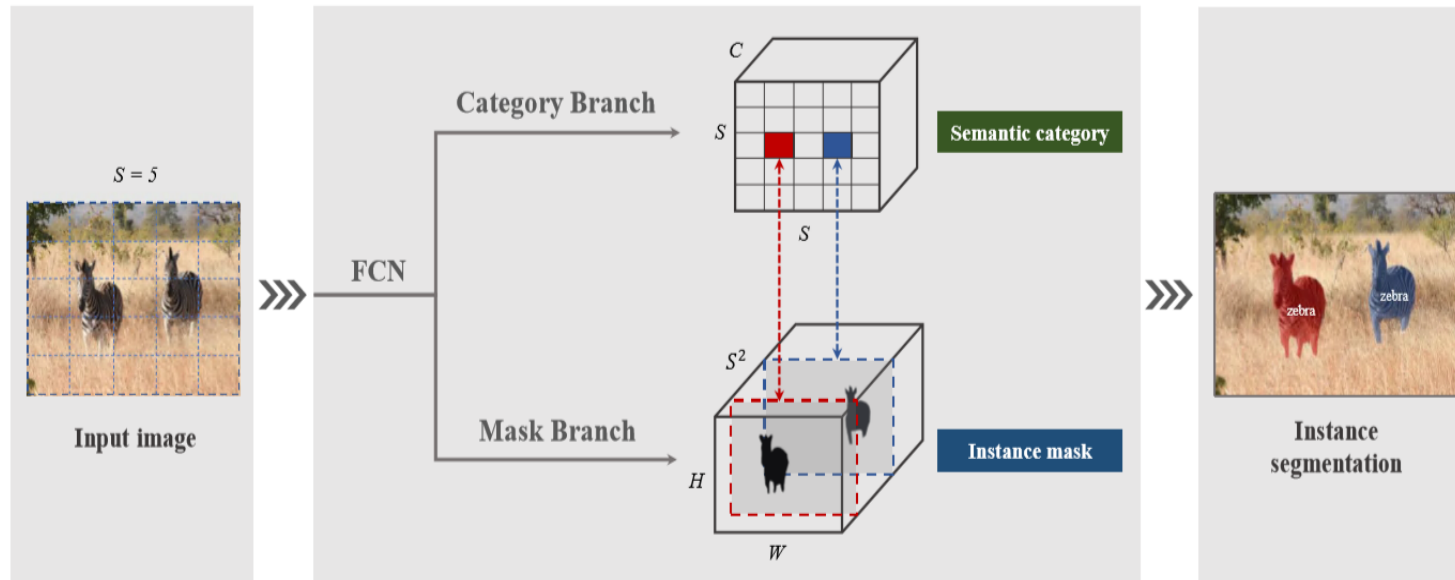


When it comes to multiple text...

Q: How to define the output order (one-to-one matching) ?



SOLO [7][8]



When it comes to multiple text...

Q: How to define the output order (one-to-one matching) ?

If more than one text *occupies* a grid. (1) increase grid numbers (2) define the priority

Occupy Ratio

$$o_{i,j} = \max\left(\frac{\text{Inter}(A(g_j), A(t_i))}{A(g_j)}, \frac{\text{Inter}(A(g_j), A(t_i))}{A(t_i)}\right)$$

When it comes to multiple text...

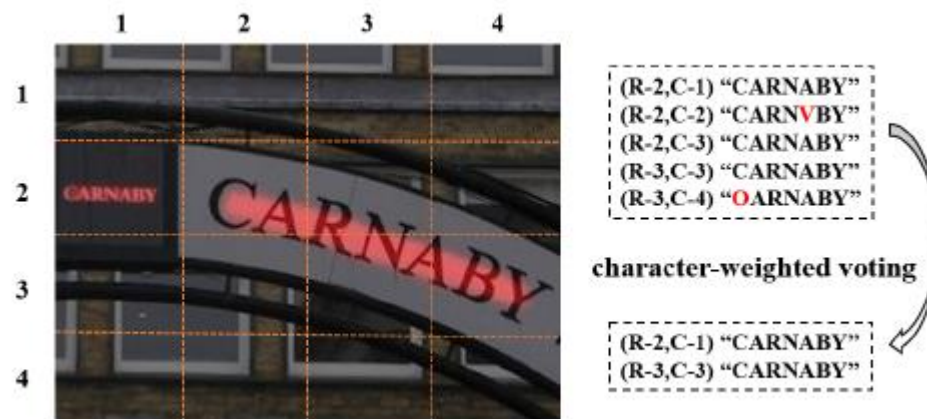
Q: How to define the output order (one-to-one matching) ?

If more than one text *occupies* a grid. (1) increase grid numbers (2) define the priority

Occupy Ratio

$$o_{i,j} = \max \left(\frac{\text{Inter}(A(g_j), A(t_i))}{A(g_j)}, \frac{\text{Inter}(A(g_j), A(t_i))}{A(t_i)} \right)$$

Inference Stage

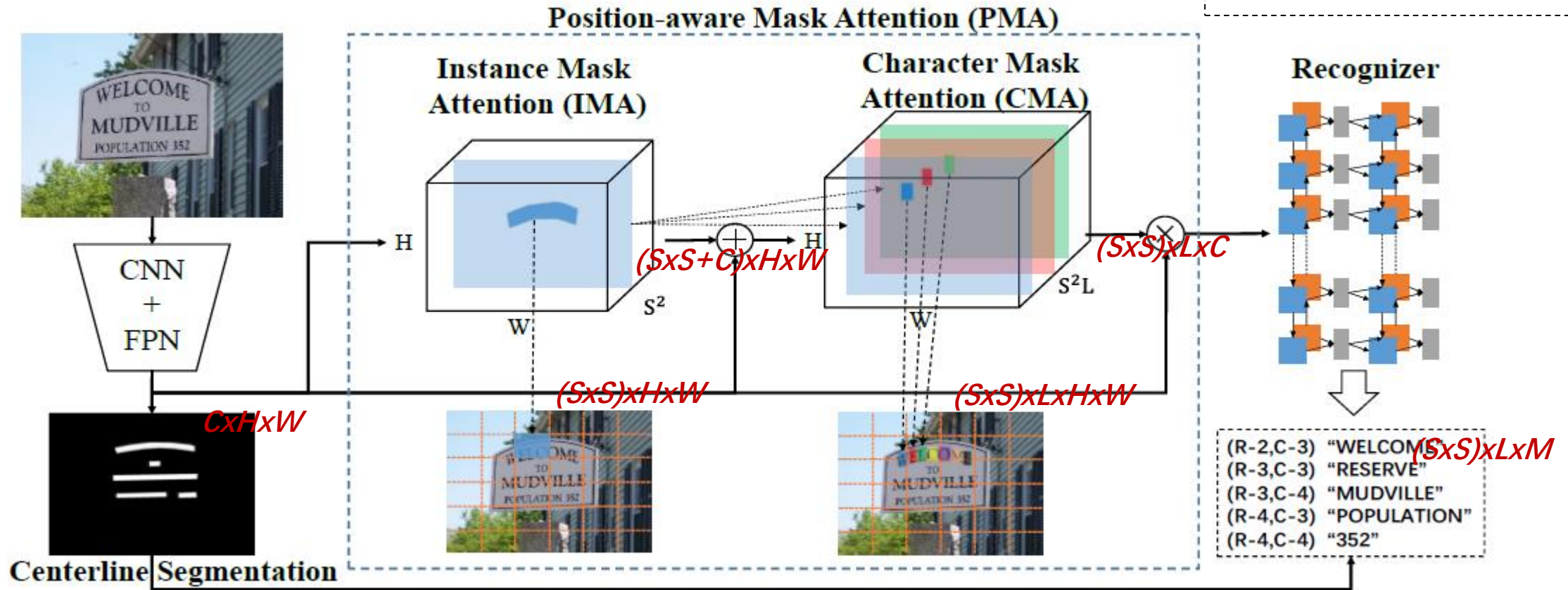


$$instance_i^{(k)} = \arg \max \left(\sum_{j \in (S \times S)} (o_{i,j} \cdot x_{recog}[j][k]) \right)$$

C feature map Channels
 H feature map height
 S^2 grid numbers
 W feature map width
 L word max length
 M dictionary length

When it comes to multiple text...

Mask(Segmentation) Supervision \approx Attention Guided



The workflow of MANGO

When it comes to multiple text...

Mask(Segmentation) Supervision \approx Attention Guided

- Training network in two phases:

Pre-training: $\mathcal{L} = \lambda_1 \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_I + \lambda_3 \mathcal{L}_C + \mathcal{L}_{recog}$

Fine-tuning: $\mathcal{L} = \lambda \mathcal{L}_{cls} + \mathcal{L}_{recog}$.

Model only need to be pre-trained once

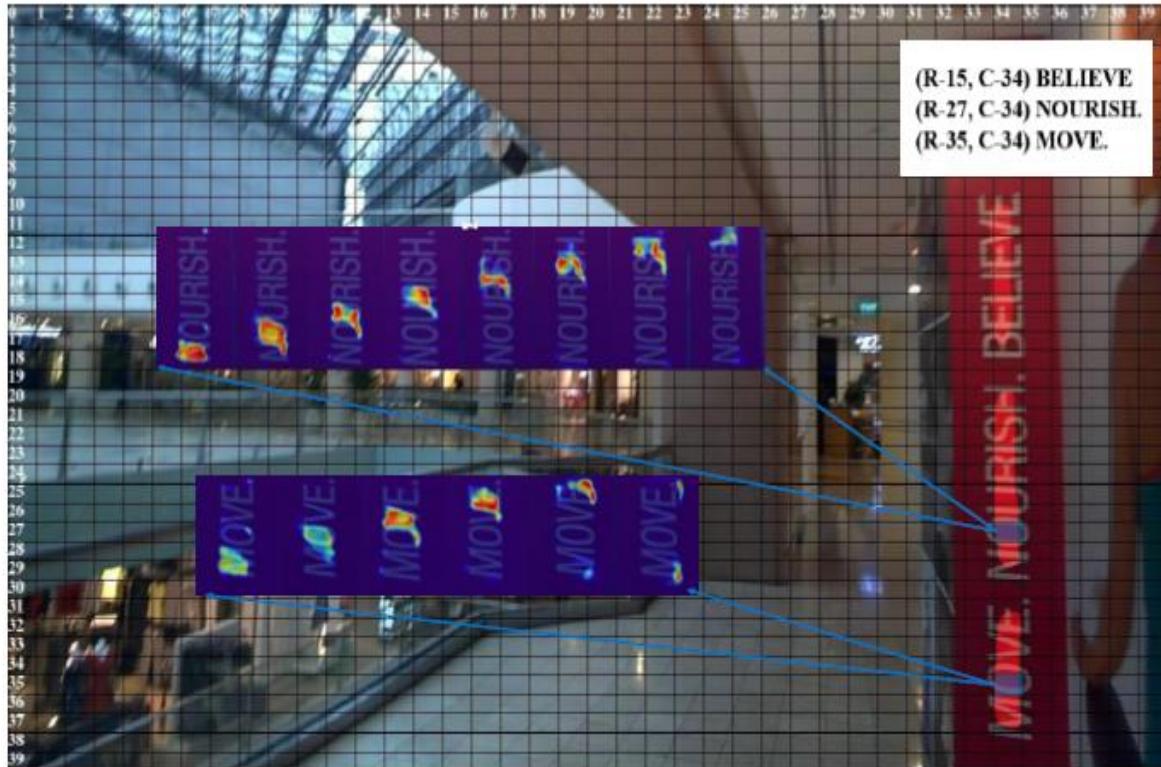
- Decouple detection and recognition
 - **Recognition:** rely on rough position (attention)
 - **Detection:** provide accurate position — used in inference

3. Experiment

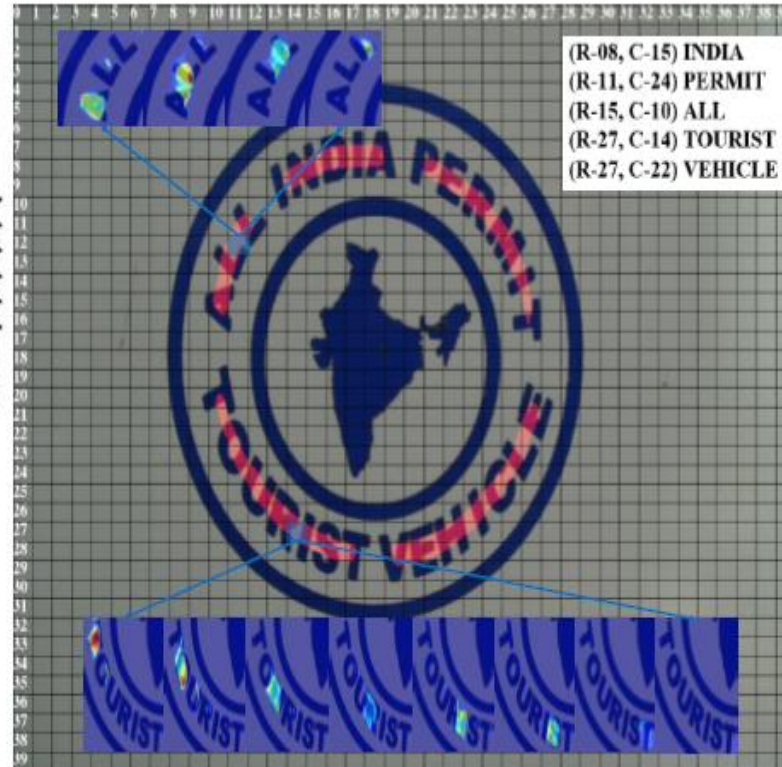
Implement Details:

- Backbone: ResNet-50 + FPN
- L=25, S=40 (IC13, Total-Text, SCUT-CTW1500) 60 (IC15)
- Pytorch, 8 32GB-Tesla-V100 GPUs
- Data augmentations: mutl-scale training, rotate, jitters, brightness...
- Single scale testing
- End-to-End metric: IoU > 0.1 (detected by centerline segmentation)
- **Sample K valid grids from SxS to save computing cost.**

Visualization Results:



(10,34)BE
 (11,34)BELIEVE
 (12,34)BELIEVE
 (13,34)BELIEVE
 (14,34)BELIEVE
 (15,34)BELIEVE
 (16,34)BELIEVE
 (20,34)NOURISH.
 (21,34)NOURISH.
 (22,34)NOURISH.
 (23,34)NOURISH.
 (24,34)NOURISH.
 (25,34)NOURISH.
 (26,34)NOURISH
 (27,34)NOURISH
 (28,34)NOURISH
 (32,34)MOVE.
 (33,34)MOVE.
 (34,34)MOVE.
 (35,34)MOVE
 (36,34)MOVE.
 (37,34)MOVE



(07,16)INDIA
 (07,17)INDIA
 (07,18)
 (08,14)INDIA
 (08,15)INDIA
 (08,16)INDIA
 (08,17)INDIA
 (08,18)
 (08,21)PERMIE
 (08,22)PERMIT
 (09,14)INDIA
 (09,22)PERMIT
 (09,23)PERMIT
 (10,23)PERMIT
 (10,24)PERMIT
 (11,11)ALL
 (11,24)PERMIT
 (12,11)ALL
 (12,25)PERMIT
 (13,10)ALL
 (13,11)ALL
 (13,25)PERMIT
 (14,10)ALL
 (14,26)T
 (15,10)ALL
 (15,26)T
 (20,10)TOURIST
 (21,10)TOURIST
 (21,26)EE
 (22,11)TOURIST

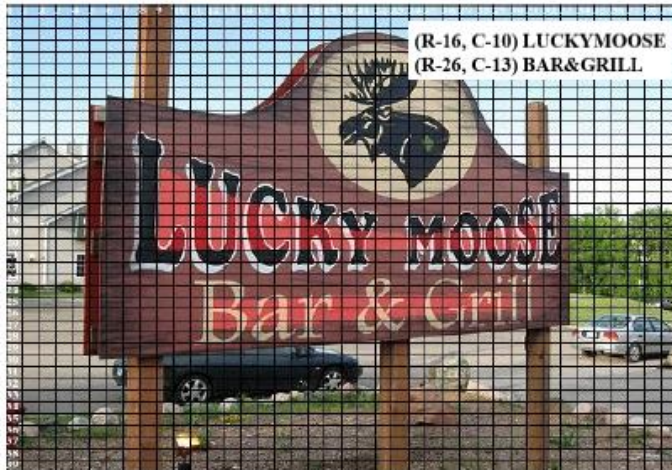
(22,26)EE
 (23,11)TOURIST
 (23,25)VEHICLE
 (24,12)TOURIST
 (24,25)VEHICLE
 (25,12)TOURIST
 (25,24)VEHICL
 (26,13)TOURIST
 (26,14)TOURIST
 (26,23)VEHICLE
 (27,14)TOURIST
 (27,15)TOURIST
 (27,21)VEHICLE
 (27,22)VEHICLE
 (27,23)VEHICLE
 (28,15)TOURIST
 (28,16)TOURIT
 (28,17)TOURIST
 (28,19)TOURIST
 (28,20)VEHICL
 (28,21)VEHICL

Visualization Results:



(R-07,C-15) WELCOMETOCOLYTON
(R-17,C-24) RESTAURANT
(R-21,C-18) SHOP
(R-24,C-15) TOILETSANDEXIT

07,14	WELCOME	(17,15) RESTAURANT	(24,19) AND
07,15	TO	(17,16) RESTAURANT	(24,20) AND
07,16	T	(17,17) RESTAURANT	(24,21) AND
07,17	TO	(17,18) RESTAURANT	(24,22) AND
07,18	TO	(17,19) RESTAURANT	(24,23) EXIT
07,19	TO	(17,20) RESTAURANT	(24,24) EXIT
07,20	TO	(17,21) RESTAURANT	(24,25) EXIT
07,21	TO	(17,22) RESTAURANT	(24,26) EXIT
07,22	TO	(17,23) RESTAURANT	(24,27) EXIT
08,12	WELCOME	(17,24) RESTAURANT	
08,13	WELCOME	(17,25) T	
08,23	CO	(20,17) SHOP	
08,24	CO	(20,18) SHOP	
09,11	WELCOME	(20,19) SHOP	
09,25	COLYTON	(20,20) SHOP	
09,26	COLYTON	(21,17) SHOP	
10,09	WELCOME	(21,18) SHOP	
10,10	WELCOME	(21,19) SHOP	
10,27	COLYTON	(21,20) SHOP	
10,28	COLYTON	(24,08) TOILETS	
11,08	WELCOME	(24,09) TOILETS	
11,09	WELCOME	(24,10) TOILETS	
11,28	COLYTON	(24,11) TOILETS	
11,29	COLYTON	(24,12) TOILETS	
12,08	WELCOME	(24,13) TOILETS	
12,30	WELCOME	(24,14) TO	
13,07	WELCOME	(24,15) TOILETS	
13,31	WELCOME	(24,16) TOILETS	
17,13	RE	(24,17) AND	
17,14	RE	(24,18) AND	



(R-16, C-10) LUCKYMOOSE
(R-26, C-13) BAR&GRILL

(14,09)	LUCKY	(19,14) LUCKY	(26,24) LUCE	(26,16) BAR
(15,09)	LUCKY	(19,15) LUCKY	(26,25) MOOSE	(26,17) BAR
(15,10)	LUCKY	(19,16) LUCKY	(26,26) MOOSE	(26,18) BAR
(15,11)	LUCKY	(19,17) LUCKY	(26,27) MOOSE	(26,19) BAR
(16,09)	LUCKY	(19,18) LUCKY	(26,28) MOOSE	(26,20) BAR
(16,10)	LUCKY	(19,19) KY	(26,29) MOOSE	(26,21) &
(16,11)	LUCKY	(19,20) LUCK	(26,30) MOOSE	(26,22) &
(16,12)	LUCKY	(19,21) LUCK	(26,31) MOOSE	(26,23) &
(16,13)	LUCKY	(19,22) LUCK	(25,12) BAR	(26,24) &
(17,09)	LUCKY	(19,26) MOOSE	(25,13) BAR	(26,25) GRILL
(17,10)	LUCKY	(19,27) MOOSE	(25,14) BAR	(26,26) GRILL
(17,11)	LUCKY	(19,28) MOOSE	(25,15) BAR	(27,12) BAR
(17,12)	LUCKY	(19,29) MOOSE	(25,16) BAR	(27,13) BAR
(17,13)	LUCKY	(19,30) MOOSE	(25,17) BAR	(27,14) BAR
(17,14)	LUCKY	(19,31) MOOSE	(25,18) BAR	(27,15) BAR
(17,15)	LUCKY	(20,09) LUCKY	(25,19) BAR	(27,16) BAR
(18,09)	LUCKY	(20,10) LUCKY	(25,20) BAR	
(18,10)	LUCKY	(20,11) LUCKY	(25,21) &	
(18,11)	LUCKY	(20,12) LUCKY	(25,22) &	
(18,12)	LUCKY	(20,13) LUCKY	(25,23) &	
(18,13)	LUCKY	(20,14) LUCKY	(25,24) &	
(18,14)	LUCKY	(20,15) LUCKY	(25,25) GRILL	
(18,15)	LUCKY	(20,16) LUCKY	(25,26) GRILL	
(18,16)	LUCKY	(20,17) LUCKY	(25,27) GRILL	
(18,17)	LUCKY	(20,18) LUCKY	(25,28) GRILL	
(19,09)	LUCKY	(20,19) KY	(25,29) GRILL	
(19,10)	LUCKY	(20,20) LUCK	(26,12) BAR	
(19,11)	LUCKY	(20,21) LUCK	(26,13) BAR	
(19,12)	LUCKY	(20,22) LUCK	(26,14) BAR	
(19,13)	LUCKY	(20,23) LUCK	(26,15) BAR	





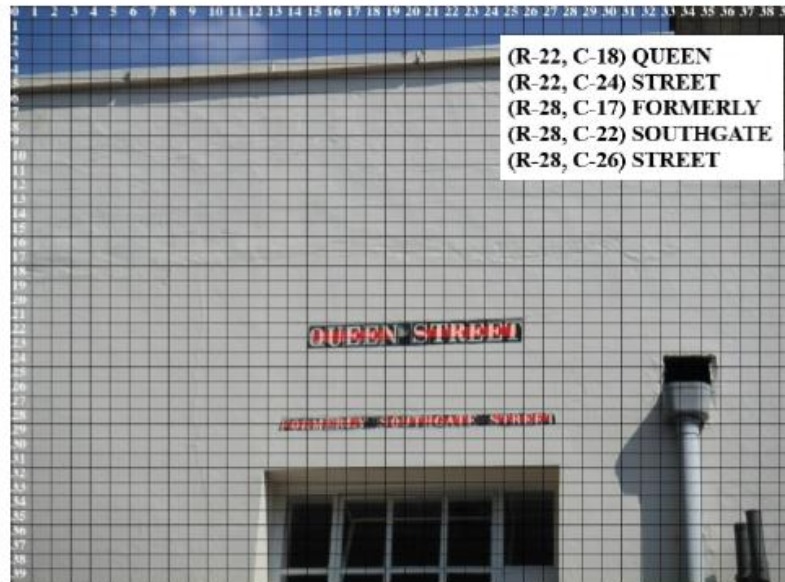
(R-14, C-19) CAFFE
(R-15, C-24) VELOCE
(R-22, C-25) BEANS
(R-23, C-17) WORLD



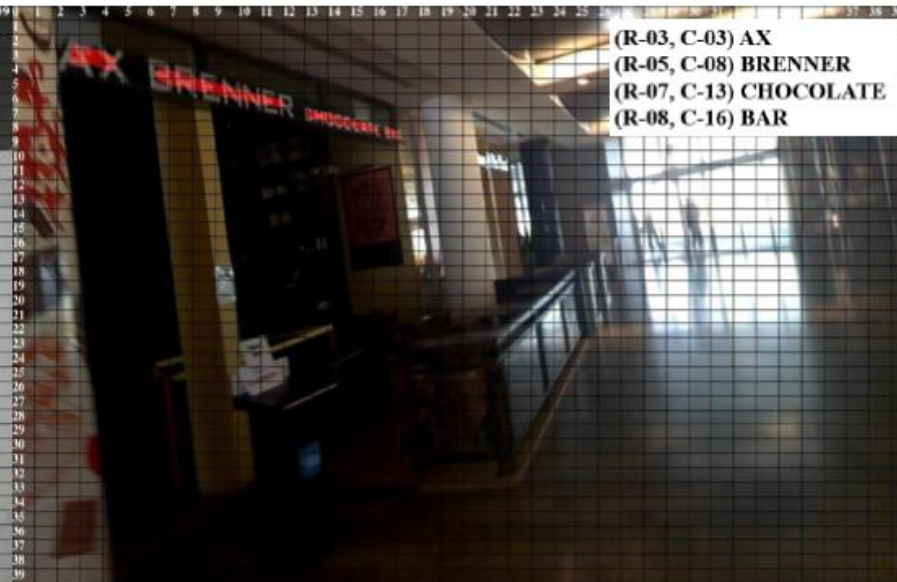
(R-02, C-04) CITY
(R-04, C-06) REMAKING
(R-07, C-03) SHOPPING
(R-07, C-08) COOSSSS
(R-10, C-07) AND
(R-11, C-09) DINING
(R-14, C-01) YOU
(R-14, C-03) FOR
(R-15, C-05) YOU
(R-15, C-07) SUPPORT



(R-14, C-18) CAFFE
(R-15, C-25) VELOCE
(R-23, C-17) WORLD
(R-23, C-24) BEANS



(R-22, C-18) QUEEN
(R-22, C-24) STREET
(R-28, C-17) FORMERLY
(R-28, C-22) SOUTHGATE
(R-28, C-26) STREET



(R-03, C-03) AX
(R-05, C-08) BRENNER
(R-07, C-13) CHOCOLATE
(R-08, C-16) BAR



(R-16, C-26) RISTORANTE
(R-19, C-22) MARCO
(R-23, C-22) POLO

Performance Evaluation:

Method	End-to-End		FPS
	None	Full	
Mask TextSpotter (Liao et al. 2019)	65.3	77.4	2.0
CharNet R-50 (Xing et al. 2019)	66.2	-	1.2
TextDragon (Feng et al. 2019)	48.8	74.8	-
Unconstrained (Qin et al. 2019)	67.8	-	-
Boundary (Wang et al. 2020a)	65.0	76.1	-
Text Perceptron (Qiao et al. 2020)	69.7	78.3	-
ABCNet (Liu et al. 2020)	64.2	75.7	17.9
MANGO (1280)	71.7	82.6	8.9
MANGO (1600)	72.9	83.6	4.3

Table 2: Results on Total-Text. ‘Full’ indicates lexicons of all images are combined. ‘None’ means lexicon-free. The number in brackets is the resized longer side of input image.

Method	End-to-End		FPS
	None	Full	
Text Perceptron (Qiao et al. 2020)	57.0	-	-
ABCNet (Liu et al. 2020)	45.2	74.1	-
MANGO (1080)	58.9	78.7	8.4

Table 3: Results on CTW1500. “Full” indicates lexicons of all images are combined. “None” means lexicon-free. The number in brackets is the resized longer side of input image.

Effect of Grid Numbers:

S	IC13				IC15				Total-Text		
	S	W	G	FPS	S	W	G	FPS	None	Full	FPS
20	83.2	82.5	78.7	6.58	33.8	33.0	29.1	5.12	46.9	58.5	4.49
30	88.8	88.3	85.9	6.32	69.4	67.1	57.8	4.57	69.8	80.6	4.37
40	90.5	90.0	86.9	6.25	80.4	77.3	66.8	4.43	72.9	83.6	4.28
50	90.3	89.8	86.7	6.12	81.6	78.8	67.8	4.38	73.1	83.0	4.23
60	89.9	89.3	85.7	6.07	81.8	78.9	67.3	4.27	72.2	82.9	4.21

Table 4: Evaluation results under different grid numbers.

Effect of Different Detection Supervisions:

Supervision Type	IC15			Total-Text	
	S	W	G	None	Full
Strong	81.8	78.9	67.3	72.9	83.6
Weak	81.8	78.3	64.0	69.7	80.6

Table 5: Results under different detection supervision types. ‘Strong’ means the original annotations, and ‘Weak’ means rectangular bounding box annotations.

Experiment on CCPD with no detection annotation:

Method	Base(100k)	DB	FN	Rotate	Tilt	Weather	Challenge	AP
SSD300 + HC	98.3	96.6	95.9	88.4	91.5	87.3	83.8	95.2
RPnet(Xu et al. 2018)	98.5	96.9	94.3	90.8	92.5	87.9	85.1	95.5
MANGO	99.0	97.1	95.5	95.0	96.5	95.9	83.1	96.9

Table 1: End-to-End recognition precision results on CCPD.



Contact Email:

qiaoliang6@hikvision.com

chengzhanzhan@hikvision.com

More works from Davar-Lab:

<https://davar-lab.github.io/>



Reference

- [1]** Qiao, L.; Tang, S.; Cheng, Z.; Xu, Y.; Niu, Y.; Pu, S.; and Wu, F. 2020. Text Perceptron: Towards End-to-End Arbitrary-Shaped Text Spotting. In AACL, 11899–11907.
- [2]** Wang, H.; Lu, P.; Zhang, H.; Yang, M.; Bai, X.; Xu, Y.; He, M.; Wang, Y.; and Liu, W. 2020a. All You Need Is Boundary: Toward Arbitrary-Shaped Text Spotting. In AACL, 12160–12167.
- [3]** Qin, S.; Bissacco, A.; Raptis, M.; Fujii, Y.; and Xiao, Y. 2019. Towards unconstrained end-to-end text spotting. In ICCV, 4704–4714.
- [4]** Feng, W.; He, W.; Yin, F.; Zhang, X.; and Liu, C. 2019. TextDragon: An End-to-End Framework for Arbitrary Shaped Text Spotting. In ICCV, 9075–9084.
- [5]** Liu, Y.; Chen, H.; Shen, C.; He, T.; Jin, L.; and Wang, L. 2020. ABCNet: Real-time Scene Text Spotting with Adaptive Bezier-Curve Network. In CVPR, 9809–9818.
- [6]** Wang, T.; Zhu, Y.; Jin, L.; Luo, C.; Chen, X.; Wu, Y.; Wang, Q.; and Cai, M. 2020b. Decoupled Attention Network for Text Recognition. In AACL, 12216–12224.
- [7]** Wang, X.; Kong, T.; Shen, C.; Jiang, Y.; and Li, L. 2019b. SOLO: Segmenting objects by locations. arXiv preprint arXiv:1912.04488.
- [8]** Wang, X.; Zhang, R.; Kong, T.; Li, L.; and Shen, C. 2020c. SOLOv2: Dynamic, Faster and Stronger. arXiv preprint arXiv: 2003.10152 .

THANKS !

MANGO: A Mask Attention Guided One-Stage Scene Text Spotter

Image to a batch of recognition results without RoI

C feature map Channels
 H feature map height
 S^2 grid numbers
 W feature map width
 L word max length
 M dictionary length

