Contextual Text Block Detection towards Scene Text Understanding

Chuhui Xue, Jiaxing Huang, Wenqing Zhang,
Shijian Lu, Changhu Wang, and Song Bai

Supplementary Material

1 Attention Preliminaries

1.1 Attention Layers:

We adopt the attention layers [3] which are defined by:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V,$$

where Q, K and V refer to input *Queries*, *Keys* and *Values*, respectively. d_k is the dimension of the Q and K.

1.2 Multi-Head Attention Layers:

The multi-head attention layer [3] linearly projects queries, keys and values M times by using different learnt linear projections. It is a concatenation of several single attention heads which is defined by:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, ..., head_M)W^O,$$
$$\text{where } head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V),$$

where W_i^Q, W_i^K and W_i^V are learnt projections of queries, keys and values in head i, respectively.

2 Dataset Details

Besides contextual text annotation, we additionally label each integral to 'Normal', 'Hard' or 'Ignore' category. Mostly, an integral text unit is labelled with 'Normal' if it belongs to a contextual text block. In some special cases, a contextual text blocks (e.g. 'POPYEYES CHICKEN & BISCUITS in Fig. 1) can be split into multiple sub-blocks (e.g. 'POPYEYES' and 'CHICKEN & BISCUITS' in Fig. 1) where each sub-block conveys a textual message and is significant to be an independent contextual text block. Here the last integral text (e.g. 'POPYEYES' in Fig. 1) in each sub-block (except the last sub-block) is label to 'Hard'. Furthermore, if an integral text or its reading order is hardly recognized, it will be labeled as 'Ignore' (e.g. '®' of the second sample in Fig. 1).

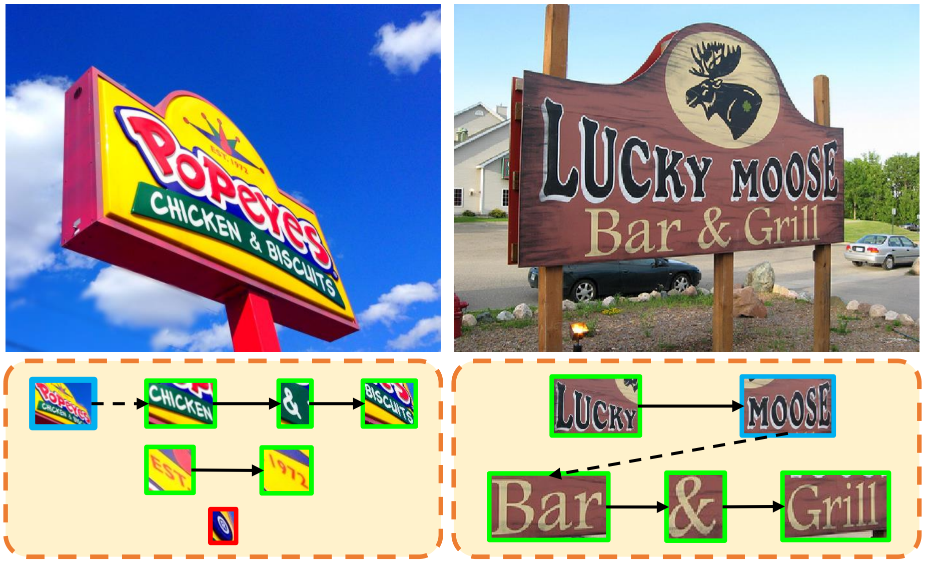

Fig. 1. Illustration of the proposed SCUT-CTW-Context dataset with different annotations: Integral texts that belong to 'Normal', 'Hard' or 'Ignore' classes are highlighted in green, blue and red, respectively. In 'Hard' cases, a contextual text block can be split into multiple sub-blocks (links shown in dotted) each of which conveys a significant textual message.

3 Implementation Details

The training of the proposed CUTE consists of two stages. First, we train the integral text detector over ReCTS and SCUT-CTW datasets, respectively which is pre-trained on COCO [2] dataset for fast convergence. We optimize the integral text detector by AdamW optimizer with batch size of 8. The learning rate is set to 10^{-5} for backbone and 10^{-4} for transformers. The numbers of the transformer encoder/decoder layers are set to 6 and the number of heads in multi-head attention is set to 8. We apply dropout of 0.1 for every multi-head attention and FFN layers before the normalization layers. For data augmentation, we apply random scaling and cropping on the input images during training. The training loss follows [1].

Second, we train the overall CUTE by freezing the parameters in backbone. We optimize the integral embedding extractor and the contextual text block generator by AdamW optimizer with batch size of 16. The learning rate is set to 10^{-4} with. The number of the mutlti-head attention layer is set to 6 and the number of heads in multi-head attention is set to 8. The dimension for constituent tokens, spatial embeddings and indexing embeddings is set to 128 and the maximum number of indices is 200. For data augmentation, we apply random scaling on the input images during training. The training loss is defined by:

$$\mathcal{L} = -\frac{1}{r} \cdot \sum_{i=0}^{r-1} y_i \cdot \log(\hat{y_i}), \tag{1}$$

where r, y_i and $\hat{y_i}$ refer to the number of integral text in image, ground-truth indices and predicted indices, respectively.

4 Additional Experiments

4.1 Integral Detectors

The proposed CUTE can adopt most of the advanced scene text detectors as the integral text detector. We build CUTE on several advanced scene text detectors and evaluate their performances on SCUT-CTW-Context dataset. As shown in Table 1, the CUTE achieves the best contextual text detection performance by adopting DETR as the integral detector, although the original DETR is not the best in scene text detection. This is DETR learns the interactions among feature elements in network backbone and so the the relationship between each pair of integral texts can be modelled well in CUTE.

Note that state-of-the-art scene text detectors are specifically designed for detecting texts with arbitrary shapes which usually achieve better performances as compared with generic object detector DETR. We directly apply original DETR in our CUTE without any modification and so the DETR doesn't achieve state-of-the-art in scene text detection. We conjecture better performances will be achieved by adopting Transformer-based scene text detectors in our CUTE.

Table 1. Integral text grouping and ordering performance of CUTE by using different numbers of attention layers.

Detector	mAP	LA	LC	GA
PSENet [4]	52.30	35.14	25.06	22.64
MSR [6]	60.07	40.41	32.52	25.17
LINK [5]	**62.03**	50.49	36.81	29.28
DETR [1]	56.11	**54.01**	**39.19**	**30.65**

4.2 Hyper-parameters

Two hyper-parameters are adopted which are commonly used in transformer-based applications.

Dimension of Embeddings: We introduce a hyper-parameter d which refers to the dimension of feature, indexing and spatial embeddings. We study the influence of hyper-parameter d on the integral text grouping and ordering task as it doesn't affect the integral detector. As Table 2 shows, the increase on d will lead to improvement on GA but also increase of parameter number consistently. By considering LA, LC, GA and number of parameters, we finally adopt $d = 128$ in our experiment.

Table 2. Integral text grouping and ordering performance of CUTE by using embeddings with different dimensions.

Dim	#param	LA	LC	GA
64	**47.97M**	70.48	57.19	48.21
128	55.67M	**71.48**	**58.53**	49.67
256	76.37M	70.64	56.38	**50.18**

Number of Attention Layers: We also introduce a hyper-parameter which refers to the number of attention layers. As Table 3 shows, we both model size and performances vary depending on the number of attention layers. We choose 6 as the layer number in all our experiments by considering both model size and performances.

Table 3. Integral text grouping and ordering performance of CUTE by using different numbers of attention layers.

#layers	#param	LA	LC	GA
1	**44.83M**	64.26	50.14	44.78
3	49.16M	70.63	56.76	47.97
6	55.67M	**71.48**	**58.53**	**49.67**
9	62.17M	71.16	56.18	48.19

4.3 Comparing with Single-Transformer Model

A single-transformer can simultaneously predict text bounding boxes and sequences as a transformer-based detector can model all interactions between elements. However, the single-transformer learns interactions among all object queries many of which are noisy or unconfident. This often introduces lots of noises in training. Differently, CUTE detects integral text units first and then learns interactions among the detected text units only (instead of all object queries), leading to more effective training and better detection performance. We conducted a new experiment by training a single transformer to detect both integral text units and contextual text blocks simultaneously (the rest remain the same as in CUTE). The trained model performs much worse than CUTE (mAP drops from 56.11 to 39.9, LA from 54.01 to 33.83, LC from 30.65 to 13.91, and GA from 30.65 to 10.19).

5 Qualitative results

Fig. 2 demonstrates the qualitative results of the proposed CUTE on ReCTS-Context and SCUT-CTW-Context datasets. As Fig. 2 shows, the proposed CUTE successfully detects the contextual text blocks from input images, demonstrating its effectiveness.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
2. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
3. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
4. Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., Shao, S.: Shape robust text detection with progressive scale expansion network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9336–9345 (2019)
5. Xue, C., Lu, S., Hoi, S.: Detection and rectification of arbitrary shaped scene texts by using text keypoints and links. arXiv preprint arXiv:2103.00785 (2021)
6. Xue, C., Lu, S., Zhang, W.: Msr: Multi-scale shape regression for scene text detection. arXiv preprint arXiv:1901.02596 (2019)

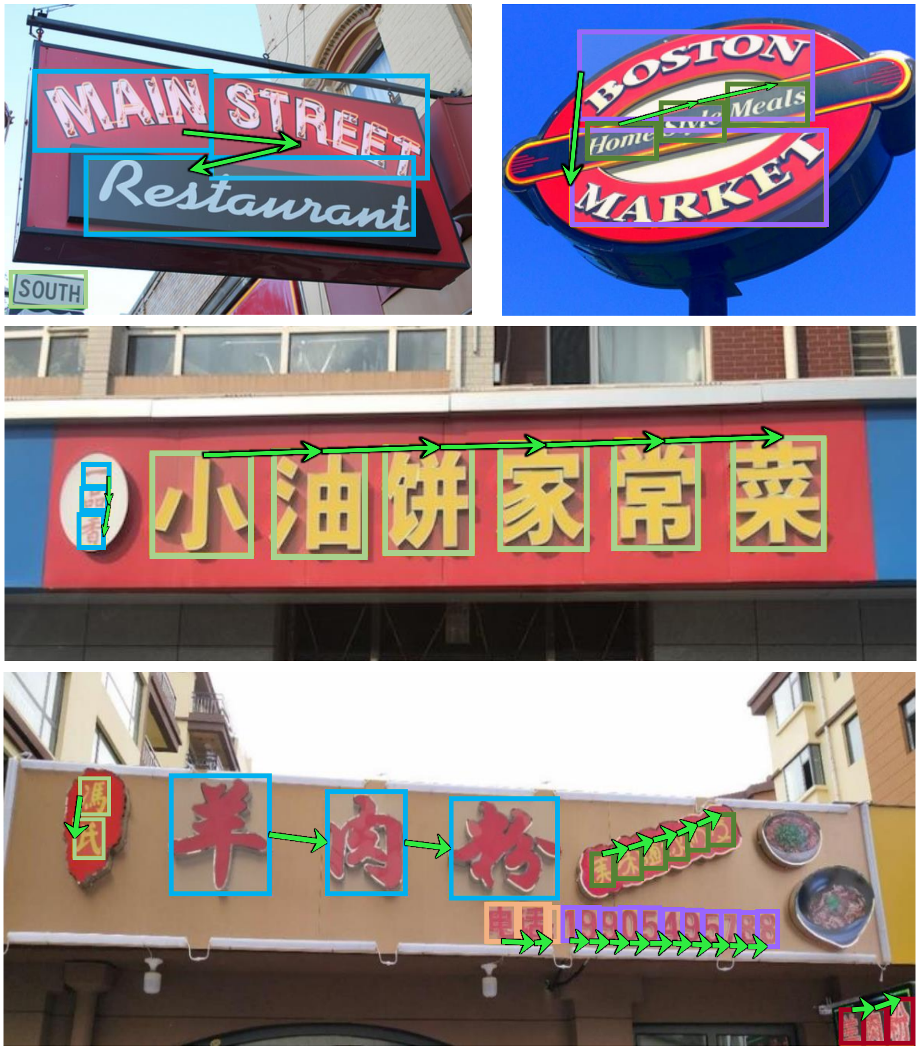

Fig. 2. Sample detection results by the proposed CUTE: Given input images from ReCTS-Context and SCUT-CTW-Context datasets, the proposed CUTE successfully detects the contextual text blocks in which each integral texts and their orders are shown by boxes with the same color and green arrows, respectively.