TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text

Amanpreet Singh*, Guan Pang*, Mandy Toh*,
Jing Huang, Wojciech Galuba, and Tal Hassner

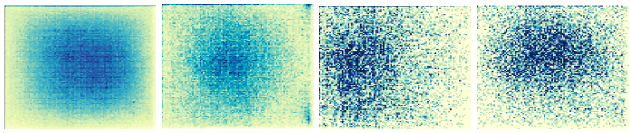
Figure A.1: Annotation UI used for TextOCR

A. Annotation UI Details

Figure A.1 shows the annotation UI that we used for the ground truth labeling of TextOCR. The annotators are able to draw any number of points to form a polygon around arbitrary-shaped word (although they are instructed to draw a quadrilateral whenever appropriate). Each polygon is displayed in a way that the edge between the first and second points is shown differently in a dotted line, to validate that the first point is at the top-left corner of the text, and the points are in clockwise order. Each polygon is then cropped out and displayed on the left screen, where annotators can transcribe the word in the polygon. The UI also has other standard functions such as zoom in/out, panning, delete polygon, and start over. Annotators are also able to re-annotate individual words within an image without needing to start over on the image by clicking 'x' on the cropped word. Annotated words are case sensitive. Figure C.1 contains more examples of annotated samples.

B. Dataset Instance Location Heatmap

Figure B.1 expands Fig.3 in main paper to compare the instance locations of TextOCR, COCO-Text, ICDAR15 and TotalText, an shows TextOCR is more uniformly annotated and distributed and existing datasets.



(a) TextOCR (b) COCOText (c) ICDAR15 (d) TotalText

Figure B.1: Word location heatmap comparison

C. OCR Model Implementation Details

We experimented with two types of OCR models in this work, text recognition, and end-to-end recognition.

We use the implementation by Baek et al. [1] [1] for text recognition task. We experimented with 4 models, including CRNN [10] (None-VGG-BiLSTM-CTC in [1]), Rosetta [3] (None-ResNet-None-CTC in [1]), STAR-Net [8] (TPS-ResNet-BiLSTM-CTC in [1]), and the TPS-ResNet-BiLSTM-Attn model proposed in [1]. For training hyper-parameters, we follow the same settings as in [1] to use AdaDelta optimizer with decay rate of 0.95. The batch size is set to 192, and gradient clipping is applied at a magnitude 5. For the cross-dataset experiments where we are training models from scratch, we train for a total of 200K iterations. For the rest of experiments that fine-tune pre-trained models on TextOCR train set, we train for 100K iterations using 4 Tesla Volta V100-SXM2-32GB GPUs. In evaluation, we measure the word accuracy by counting the rate of perfectly predicted words.

For the end-to-end recognition, we use the official implementation of Mask TextSpotter (MTS) V3 by Liao et al. [7] [2]. We use SGD with momentum of 0.9 and weight decay of 0.0001 for training. The initial learning rate is set to 0.001, and divided by 10 every 100K iterations, for a total of 300K iterations. The batch size is set to 8 and rotation augmentation is performed by randomly rotating input image with an angle between $-90°$ and $90°$. We also perform multi-scale training that resizes the short side of input image randomly to one of (800, 1000, 1200, 1400). We train our

*Equal Contribution. Correspondence to textvqa@fb.com

[1] https://github.com/clovaai/deep-text-recognition-benchmark
[2] https://github.com/MhLiao/MaskTextSpotterV3

models using 8 Tesla Volta V100-SXM2-32GB GPUs in a distributed fashion using PyTorch [9]. During evaluation, we measure with the same protocol as described in [6] that follows ICDAR2015 with support for polygon representation, and the short side of input images resized to 1000.

D. Experiments on same number of instances

To demonstrate that besides the large scale, TextOCR also has good quality compared to previous datasets, we experimented with the same number of instances as ICDAR15 [5] and COCO-Text [13]. We randomly sampled 4055 and 38839 word images from TextOCR for ICDAR15 and COCO-Text, respectively. All experiments fine-tune TPS-ResNet-BiLSTM-Attn [1] from a base pretrained on Synth90k+SynthText, same as paper. As shown in Table D.1, TextOCR-4055 outperforms ICDAR15 on all standard recognition benchmarks except ICDAR15 itself, proving TextOCR provides more diversity and generalizes better to other test sets than ICDAR15, which focuses on incidental scene text. TextOCR-38839 outperforms COCO-Text on 5 out of 7 benchmarks, indicating its superior quality and generalization.

Train Dataset	IIIT	SVT	IC03	IC13	IC15	SVTP	CUTE
ICDAR15	83.87	85.94	93.20	91.72	**79.46**	78.61	65.16
TextOCR-4055	**87.27**	**88.10**	**94.93**	**93.35**	78.25	**80.78**	**72.47**
COCO-Text	86.07	87.79	**93.66**	92.77	79.79	78.61	**74.91**
TextOCR-38839	**86.17**	**88.56**	92.85	**93.12**	**80.18**	**80.78**	74.56

Table D.1: Text recognition with same number of instances

E. PixelM4C: Number of OCR tokens

We conduct a sweep on number of OCR tokens used in PixelM4C to confirm that more tokens help when the OCR model is trained on TextOCR and the downstream model is using decoder's last hidden state. Table E.1

Experiment	OCR	TextVQA val acc.
50 tokens	MTS v3 (TextOCR-en)	37.75
50 tokens	TextOCR	45.22
100 tokens	MTS v3 (TextOCR-en)	39.41
100 tokens	TextOCR	46.42
200 tokens	MTS v3 (TextOCR-en)	39.41
200 tokens	TextOCR	46.12
200 tokens	MTS v3 (TextOCR-en-LH)	40.31
200 tokens	TextOCR-LH	45.49

Table E.1: **Ablation analysis on number of OCR tokens**. The results show that more OCR tokens are better for TextVQA [12] when the OCR model is trained on TextOCR.

F. PixelM4C: Hyper-parameters and ST-VQA

Table F.1 shows various hyper-parameter choices for PixelM4C and PixelM4C-Captioner used for training the models on TextVQA [12] and TextCaps [11] dataset. We compare the performance of the model on batch size 16 as well as 128 and found batch size 16 reasonably better or equal in performance to batch size 128. For the ease of training the model with less number of GPUs, we stick with batch size 16 for our experiments.

The confidence threshold for filtering of OCR tokens which works for the best OCR performance doesn't work as it is for PixelM4C suggesting one more motivation for fine-tuning and adjusting OCR models based on the downstream task. The OCR model (MTS v3) uses a of 0.2 confidence threshold on detection score and 0.8 on recognition score. For PixelM4C, the no threshold on detection score and 0.2 confidence threshold on recognition score works best which we confirm by a hyper-parameter sweep.

Hyper-parameter	PixelM4C	PixelM4C-Captioner
batch size	16	16
learning rate	1e-4	1e-4
learning schedule	step(14k, 19k)	step(10k, 11k)
warmup iterations	1000	1000
maximum iteration	24000	12000
Adam β_1	0.9	0.9
Adam β_2	0.999	0.99

Table F.1: **PixelM4C hyper-parameters.**

For completeness, we also trained PixelM4C with TextOCR trained Latin OCR model on ST-VQA [2] train set and test on its validation set created in [4]. We get an accuracy of 38.49% and 47.89% ANLS better than that reported in [4] again justifying that TextOCR leads to better downstream models.

G. Sources of the media used

- Figure 2 (row 1, column 1), "The What" by rjp licensed CC-BY-2.0.
- Figure 2 (row 1, column 2), "Washington D.C. Tour - African Land Forces Summit - 201005611" by US Army Africa licensed CC-BY-2.0
- Figure 2 (row 1, column 3), "slc camp" by Noah Sussman licensed CC-BY-2.0
- Figure 2 (row 1, column 4), "1945" by Homini:) licensed CC-BY-2.0
- Figure 2 (row 2, column 1), "iḿ watch" by shinji_w licensed CC-BY-2.0
- Figure 2 (row 2, column 2), "Cleansui CSP-801" by othree licensed CC-BY-2.0

Figure C.1: TextOCR Annotation Samples

- Figure 2 (row 2, column 3), "KA 003" by Kaja Avberšek licensed CC-BY-2.0
- Figure 2 (row 2, column 4), "Darwin Origin of Species exhibit at Huntington Library" by favouritethings licensed CC-BY-2.0
- Figure 2 (row 3, column 1), "Greetings from Tallahassee, Florida" by Boston Public Library licensed CC-BY-2.0
- Figure 2 (row 3, column 2), "Another design ready for our Print Party. In solidarity with a prisoner led-movement calling for the abolition of solitary confinement. prepping for a big rally and on Tuesday in Sacramento. #rinitempleton #abolishsolitary #art #artistactivism #phss" by dignidadrebelde licensed CC-BY-2.0
- Figure 2 (row 3, column 3), "Clock – 1319 F Street NW Washington (DC) July 2013,413654" by Ron Cogswell

licensed CC-BY-2.0

- Figure 2 (row 3, column 4), "Angry Man #Knock-out" by Phil Whitehouse licensed CC-BY-2.0
- Figure 4 (b) (left) "Ross Diploma" by Ross Housewright
- Figure 4 (b) (middle) "Clark's Big Top Restaurant, 1968" by Seattle Municipal Archives
- Figure 4 (b) (right) "Locomotive" by Duane Burdick
- Figure 4 (c) (top left) "Tienda de souvenirs en santiago" by compostelavirtual.com
- Figure 4 (c) (top right) "DSC00062" by Carlos Correa Loyola
- Figure 4 (c) (bottom left) "I REMEMBERS DAYS OF OLD" by marc falardeau
- Figure 4 (c) (bottom right) "DSC00062" by Carlos Correa Loyola
- Figure C1 (row 1, column 1), "ATRK" by BOMB THE SYSEM licensed CC-BY-2.0
- Figure C1 (row 1, column 2), "Lost Book" by Steve Bowbrick licensed CC-BY-2.0
- Figure C1 (row 1, column 3), "Big Helga and Bulmers" by James Dennes licensed CC-BY-2.0
- Figure C1 (row 1, column 4), "Bull Herzl fifty years to his death (original in Hebrew)" by zeevveez licensed CC-BY-2.0
- Figure C1 (row 2, column 1), "Good Grief Glasses" by brett jordan licensed CC-BY-2.0
- Figure C1 (row 2, column 2), "DSC_0092" by mlwilson1410 licensed CC-BY-2.0
- Figure C1 (row 2, column 3), "cien pesos 1977 4735" by Eric Golub licensed CC-BY-2.0
- Figure C1 (row 2, column 4), "Clock Squircle" by Gareth Simpson licensed CC-BY-2.0
- Figure C1 (row 3, column 1), "Every Woman Is At Risk" by Peter Galvin licensed CC-BY-2.0
- Figure C1 (row 3, column 2), "Spotted at Kinokunia Books, San Francisco @hollowlegs" by Gary Stevens licensed CC-BY-2.0
- Figure C1 (row 3, column 3), "Boozy from Bouzy is our favorite! #delectable #wine" by Dale Cruse licensed CC-BY-2.0
- Figure C1 (row 3, column 3), "Yurt Exhibit" by thekirbster licensed CC-BY-2.0

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. In *Proc. Int. Conf. Comput. Vision*, 2019. 1, 2

[2] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4291–4301, 2019. 2

[3] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *KDD*, 2018. 1

[4] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9992–10002, 2020. 2

[5] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE, 2015. 2

[6] Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 2

[7] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask textspotter v3: Segmentation proposalnetwork for robust scene text spotting. In *European Conf. Comput. Vision*, 2020. 1

[8] Wei Liu, Chaofeng Chen, Kwan-Yee K. Wong, Zhizhong Su, and Junyu Han. Star-net: A spatial attention residue network for scene text recognition. In *Proc. British Mach. Vision Conf.*, 2016. 1

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dÁlché-Buc, E. Fox, and R. Garnett, editors, *Proceedings of NeurIPS*, pages 8024–8035. Curran Associates, Inc., 2019. 2

[10] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *Trans. Pattern Anal. Mach. Intell.*, 39:2298–2304, 2017. 1

[11] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. *arXiv preprint arXiv:2003.12462*, 2020. 2

[12] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019. 2

[13] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016. 2