

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

Facebook AI Research

<https://textvqa.org/textocr>

Abstract

A crucial component for the scene text based reasoning required for TextVQA and TextCaps datasets involve detecting and recognizing text present in the images using an optical character recognition (OCR) system. The current systems are crippled by the unavailability of ground truth text annotations for these datasets as well as lack of scene text detection and recognition datasets on real images disallowing the progress in the field of OCR and evaluation of scene text based reasoning in isolation from OCR systems. In this work, we propose TextOCR, an arbitrary-shaped scene text detection and recognition with 900k annotated words collected on real images from TextVQA dataset. We show that current state-of-the-art text-recognition (OCR) models fail to perform well on TextOCR and that training on TextOCR helps achieve state-of-the-art performance on multiple other OCR datasets as well. We use a TextOCR trained OCR model to create PixelM4C model which can do scene text based reasoning on an image in an end-to-end fashion, allowing us to revisit several design choices to achieve new state-of-the-art performance on TextVQA dataset.

1. Introduction

The computer vision community has recently seen a surge in interest to understand and reason on the text present in the images (scene text) beyond the OCR extraction. In particular, multiple datasets have been introduced that focus on **visual question answering (VQA)** [55, 4, 41] and **image captioning** [53] but in the context of scene text. These tasks involve understanding the objects and text in the image and then reasoning over the spatial and semantic relations between these along with a textual input (*e.g.* question). Though the

OCR systems have matured, they still don't work well on pictures involving real-life scenarios given the lack of large annotated real scene text OCR datasets. The text extracted by the OCR systems doesn't mean anything in itself until it is used to solve a task which involves using the scene text. Other than VQA and image captioning, the potential use cases include several impactful and interesting tasks from hate speech and misinformation detection [27] to the study of cultural heritage [17, 18].

Although, the field has witnessed success and progress in datasets on downstream OCR applications, the performance of state-of-the-art models on these datasets are nowhere close to human accuracy due to multiple factors which includes the quality of the OCR extracted from existing OCR systems, unavailability of ground-truth text annotations for the real-world images, and no feedback to OCR system to improve detection or extraction based on the errors in the downstream application *i.e.* no end-to-end training.

In this paper, we introduce a new dataset, TextOCR, which aims to bridge these gaps by providing (i) high quality and large quantity text annotations on TextVQA images (ii) allowing end-to-end training of downstream application models with OCR systems and thus allowing fine-tuning of OCR pipeline based on the task involved. Prior to TextOCR, many OCR datasets exist [40, 61, 37, 26, 25, 47, 50, 8, 63, 36, 59, 43, 52, 42, 9, 56] that propelled the field's development, but many of these are either relatively small, or focus mostly on outdoor or store-front scenes. As a result, OCR models trained on these datasets usually don't perform well on downstream tasks from other scene types. Moreover, existing datasets usually have a low number of words per image, making them less dense, diverse and ideal to train OCR models for tasks commonly having a high text density. As a solution, we present the TextOCR dataset that contains more than 28k images and 903k words in total, averaging 32 words per image. Jointly with existing TextVQA [55]

*Equal Contribution. Correspondence to textvqa@fb.com

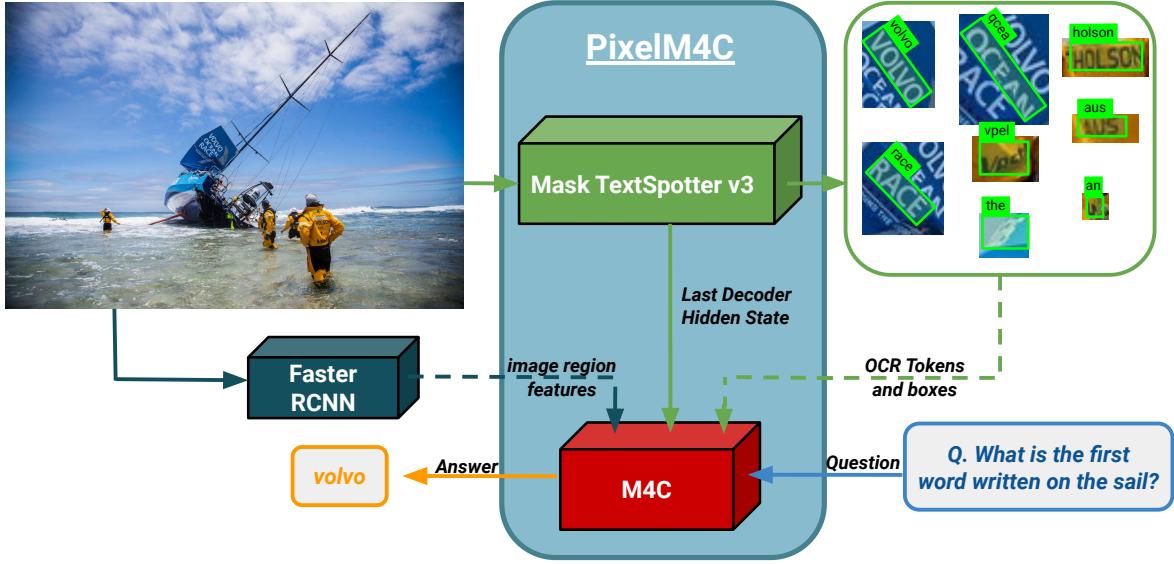


Figure 1: **PixelM4C - An end-to-end TextVQA model.** In this work, we bridge the gap between arbitrary scene-text detection/recognition and scene-text based reasoning in TextVQA [55] and TextCaps [53]. We introduce TextOCR, largest real scene-text detection and recognition dataset with 900k annotated arbitrary-shaped words collected on TextVQA images. Further, we build PixelM4C, an end-to-end TextVQA model which uses TextOCR trained Mask TextSpotter v3 [32] and M4C [19] models to do text-based reasoning directly on the images unlike previous works which rely on pre-extracted OCR text and features. The solid lines in the figure show backpropagable paths.

and TextCaps [53] datasets, it can also serve as an OCR upper bound for researchers working on them to evaluate their methods’ reasoning capabilities on a fair ground.

In addition to TextOCR, we present a novel architecture, PixelM4C, that connects an OCR model, Mask TextSpotter (MTS) v3 [32] with downstream TextVQA model, M4C [19], in an end-to-end trainable fashion, as illustrated in Figure 1. Through extensive analysis and ablations possible with end-to-end PixelM4C, we revisit and improve design choices from prior work to achieve new state-of-the-art performance on TextVQA dataset [55] under comparable settings and show TextOCR’s performance impact on new TextCaps dataset [53].

In summary, our main contributions include:

- A large and diverse OCR dataset with $\sim 1M$ arbitrary-shaped word annotations (3x larger than existing datasets), with high density of ~ 32 words per image.
- Extensive experiments to evaluate TextOCR showing that it is effective both as (i) a training dataset to push OCR state-of-the-art on multiple datasets and (ii) a testing dataset to offer a new challenge to the community.
- A new end-to-end novel architecture, PixelM4C for TextVQA and TextCaps, which connects Mask TextSpotter (MTS) v3 [32] to M4C [19] allowing extensive analysis and revisiting prior work’s design decisions.
- State-of-the-art on TextVQA [55] using OCR tokens generated from TextOCR trained OCR models and insights

from PixelM4C ablations under comparable settings.

2. Related work

2.1. OCR datasets

Recognition. The text recognition datasets which involve recognizing text from cropped words can be categorized as regular or irregular. The regular datasets like IIIT5K-Words (IIIT) [40], Street View Text (SVT) [61], ICDAR2003 (IC03) [37], ICDAR2013 (IC13) [26] have horizontally aligned words while irregular datasets like ICDAR2015 (IC15) [25], SVT Perspective (SVTP) [47], CUTE80 [50], and Total Text (TT) [8] are more challenging as they involve various transformations, such as arbitrary-oriented or curved.

Detection. Compared to older OCR datasets, which only allowed recognition as they came with pre-cropped words, the newer datasets can be used for either detection or end-to-end task as they have full images with labeled instances. The examples IC13[26], IC15[25], and TT[8] can use different word location formats, horizontal box, quadrilateral box, and curved polygon respectively. Additionally, datasets like MSRA-TD500 [63] and CTW1500 [36] with line-level labels are commonly only used for detection task.

Multi-Lingual. In recent years, there has been a surge in large-scale multi-lingual datasets containing (i) up to 7 or 8 different scripts (*e.g.* ICDAR17-MLT [43] and ICDAR19-MLT [43]), (ii) Chinese and English due to large character set (*e.g.* ICDAR17-RCTW [52], ICDAR19-ArT [9] and



Figure 2: **TextOCR visualizations.** The figure shows diversity and density in TextOCR images.¹

ICDAR19-LSVT [56]). These usually use test set for a challenge not releasing the labels and being multi-lingual, the amount of data distributed in each language is smaller.

Synthetic. Synth90k [21] with 9M word-level crops, and SynthText [14] with 800K images (6M words) are the most common in OCR research. As these are synthetic and contain a lot of data, they are typically used for model pretraining or joint training with real datasets.

2.2. Downstream OCR applications

In recent years, multiple datasets have been introduced for scene-text applications to study and reason about the text present in an image w.r.t. the objects in it. TextVQA [55] contains 28K images from OpenImages [30] with 45K questions, each with 10 human-annotated answers, which require reading and reasoning over scene-text to answer them. Similarly, ST-VQA [4] contains 32k questions on images from 6 different sources (IC13 [26], IC15 [25], ImageNet [10], VizWiz [3], IIIT Scene Text Retrieval, Visual Genome [29], and COCO-Text [59]). A series of datasets were introduced following these which focused on specific aspects of text-based VQA including OCR-VQA [41], STE-VQA [62], DocVQA [38], PlotQA [39], and LEAF-QA [7].

TextCaps dataset [53] requires reading comprehension with images and contains 143K captions on TextVQA images [55]. TextCaps requires understanding how OCR words interact with each other and objects to build a caption which is coherent while tackling challenges like parts-of-speech,

¹All images are licensed under CC BY 2.0. See appendix for full attributions.

OCR and fixed vocabulary switching. VizWiz-Captions [15] also contains similar captions on VizWiz images [3] but doesn't explicitly require scene-text reasoning.

2.3. Downstream application models

The state-of-the-art on TextVQA and TextCaps use the pre-extracted text tokens from a standard OCR system as additional input [23]. As the OCR text can be any string, for word embeddings, we use a system that allows out-of-vocabulary words via character-level modeling or piece-based modeling [23]. The other textual input (*e.g.* question) is encoded using a pretrained word-embedding (BERT, GloVe [11, 46]) and fused with image's object features and OCR embeddings. The joint embedding passes through a classifier or decoder to generate the output. The state-of-the-art TextVQA model, M4C [19], uses transformers [57] to model the fusion via self and inter-modality attention to achieve 40% on TextVQA compared to 86% human accuracy. On TextCaps, M4C can be adapted to generate a sentence by taking previously generated words as text inputs at each time step. Multiple models have been introduced recently which ablate various components of M4C for better accuracy [24, 13, 16, 22]. Contrary to M4C and derivative works which treat OCR as a black box, in PixelM4C, we train an end-to-end model and use this capability to apply new design choices in a more informed way. To test our hypothesis, we build and compare PixelM4C with M4C as our base because of its simplicity and modular design.

3. TextOCR dataset

3.1. Annotation Details

For collecting arbitrary shaped scene text, all words within an image are annotated with polygon annotation for detection. For recognition, only Latin words are annotated. All non-Latin and illegible words are then annotated with a “.”. Similar to COCOText, a word is defined as an uninterrupted sequence of letters.

The annotations are performed by a set of annotators familiar with polygon word annotation. We provided the team with annotation guidelines, and a quality control training and performed rigorous checks on their work. The annotators first annotate the bounding box around a word. If the word is near-horizontal, annotators are encouraged to draw a rectangle whenever possible. If the word is curved, then the annotators draw polygon as annotations with multiple points while preserving reading direction from the first annotated point to the second annotated point. The annotators are also encouraged to annotate the bounding box with as little background space as possible. To ensure accuracy of predictions, our annotation pipeline includes an audit procedure, where expert annotators (authors) provide feedback to individual annotators for re-annotation. Please see appendix for more details on our annotation user interface.

3.2. Statistics and Visualizations

Figure 3 shows that TextOCR is diverse both in terms of words per image (left) as well as the word locations (right). Figure 3 (a) compares and shows high density of word annotations in TextOCR compared with COCOText [59] and ICDAR15 [25]. Figure 3 (b) and (c) compare the density of word bounding boxes in TextOCR and COCOText depicting more uniform, regular and heavy density in TextOCR suggesting that TextOCR is more precisely, uniformly and carefully annotated.

Table 1 shows the statistics of TextOCR compared to other public datasets. TextOCR has more images than most existing public datasets except for ICDAR19-LSVT [56], a bilingual dataset focusing more on street view images in Chinese. TextOCR has much larger number of annotated words than any existing public datasets, with at least 3x more words than ICDAR19-LSVT and 10x more than the rest. ICDAR19-LSVT contains only 44K words in English, while TextOCR contains predominantly English, with 20x English words. As a result, TextOCR contains on average 32.1 words per images, 3x more than any existing datasets, making it a great source for both word-level text recognition task and image-level text spotting task in text heavy scenes.

Table 2 offers more detailed statistics into TextOCR. There are a total of 1.32M labeled instances in TextOCR if including empty word annotations where the word box or polygon is labeled but the text is not transcribed (due to

#	Dataset	# Images		# Words		Words per Image
		Train	Test	Train	Test	
1	Synth90k [21] [†]	–	–	8.9M	–	–
2	SynthText [14] [†]	800k	–	5.5M	–	6.9
3	IIT5K [40]	–	–	2000	3000	–
4	SVT [61]	–	–	257	647	–
5	ICDAR2003 [37]	–	–	1156	1110	–
6	ICDAR2013 [26]	229	233	848	1095	4.2
7	ICDAR2015 [25]	1000	500	4468	2077	4.4
8	SVTP [47]	–	–	–	645	–
9	CUTE80 [50]	–	–	–	288	–
10	Total-Text [8]	1255	300	9276	2215	7.4
11	MSRA-TD500 [63]	300	200	–	–	–
12	CTW-1500 [36]	1000	500	–	–	–
13	COCO-Text [59]	18895	4416	61793	13910	3.2
14	ICDAR17-MLT [43] ^{*‡}	9000	9000	85094	n/a	9.5
15	ICDAR17-RCTW [52] [*]	8034	4229	47107	n/a	5.9
16	ICDAR19-MLT [43] [*]	10000	10000	89407	n/a	8.9
17	ICDAR19-ArT [9] [*]	5603	4563	50042	n/a	8.9
18	ICDAR19-LSVT [56] [*]	30000	20000	243385	n/a	9.1
19	TextOCR (ours) [‡]	24902	3232	822572	80497	32.1

Table 1: **TextOCR vs other datasets.** We only count non-empty words and images with at least one instance). For non end-to-end and test-only datasets, unavailable fields are left blank. * ⇒ multi-lingual datasets with no test labels and small English annotations, † ⇒ synthetic datasets. ‡ ⇒ val set counted in the train set.

Count Type	Train	Val	Test	Total
Images	21749	3153	3232	28134
Labeled instances	1052001	150338	117725	1320064
Empty words	337815	41952	37228	416995
Non-empty words	714186	108386	80497	903069
Non-alphanumeric	102744	15595	11596	129935
Less than 3 chars	197100	28726	24643	250469
Alphanumeric & 3+ chars	414342	64065	44258	522665
Rotated (degree > 20)	118547	18548	13102	150197
Curved (points > 4)	14368	3099	1843	19310

Table 2: **TextOCR statistics.** Details on instance types

illegibility or language). If we remove words that are non-alphanumeric (e.g. symbols) or have fewer than 3 characters (a standard in some datasets), TextOCR still contains 523k words. Among these, 150k are rotated words ($> 20^\circ$ angle) and 19.3k are curved words (more than 4 points used to draw the polygon), almost twice the total words in Total-Text [8], a dataset focusing on curved text.

4. OCR Experiments

In this section, we evaluate the TextOCR dataset and the challenge it presents, then exhibit its usefulness and empirically show how it can be used for both training su-

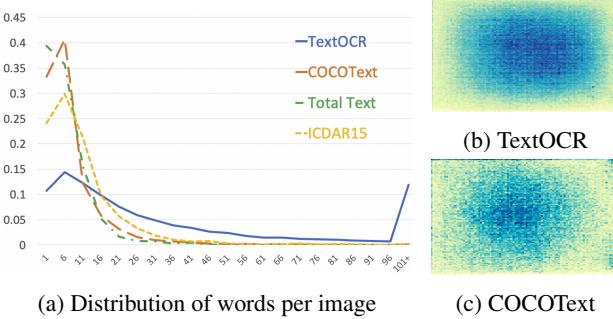


Figure 3: TextOCR distributions. (left) Comparison of words per image showing TextOCR’s higher text density, with $> 10\%$ images containing 100+ instances. (right) Word locations’ heatmaps across images with blue indicating higher density. TextOCR is more uniformly annotated/distributed than COCOText [59] which is more uniform than IC13 and IC15.

perior models and surpassing existing baselines on other text-recognition benchmark. We demonstrate this through three types of experiments: (i) cross-dataset empirical analysis, (ii) achieving state-of-the-art on public benchmarks, and (iii) evaluating state-of-the-art methods on TextOCR. Please refer to supplementary material for implementation details.

4.1. Cross-dataset empirical analysis

COCOText [59], one of the largest fully-annotated English OCR dataset, has images from COCO which were originally collected for object detection purpose, resulting in sparse text occurrences.

As this setting is different from usual OCR applications and benchmarks, COCOText is not ideal to train or test upon. On the other hand, TextVQA dataset (image source of TextOCR) is designed for visual question answering based on text in the image leading to more prominent text, making it also a great OCR data source.

In our text recognition experiments, shown in Table 3, we use the TPS-ResNet-BiLSTM-Attn model [1], and train it on COCOText (row #1) and TextOCR (row #2) separately from scratch for 100k iterations keeping all other settings the same. We evaluate and compare the results on TextOCR, COCOText and other common text recognition benchmarks. The model trained on TextOCR is 22.45% better than COCOText on the TextOCR test set, and 10.56% better even on the COCOText’s test set. On the other benchmarks, TextOCR-trained model is consistently better with 10% or more gap. The superior performance can be attributed to the sheer amount of difference in number of words in these datasets compared to TextOCR. Note that training on COCOText alone (w/o synthetic data) only achieves 64.05% word accuracy on ICDAR 2015 [25], signaling it is not a

good representative of oriented scene text. Comparatively, training on TextOCR alone can achieve near state-of-the-art performance of 80.07%. Besides its large scale, we also show TextOCR has good quality compared to previous datasets, by experiments on the same number of instances as ICDAR15 and COCO-Text. Results show TextOCR is 2.5% better than ICDAR15 on average in recognition benchmarks, and 0.3% better than COCO-Text, thanks to its good quality and diversity. Please refer to supplementary experiment details.

Table 4 shows results on end-to-end recognition evaluating TextOCR’s usefulness on the image-level task. We use the latest Mask TextSpotter (MTS) V3 [32]² and train it from scratch (with ResNet50 trunk pretrained on ImageNet) on COCOText (row #1) and TextOCR (row #2) separately. We can see model fine-tuned on TextOCR again has a 0.2% lead over COCOText on its own test set, and 10%+ lead on the TextOCR and Total-Text test sets. This demonstrates the advantage of using TextOCR as a training data as it is more generalizable on other datasets. Since the number of images in TextOCR is comparable to COCO-Text (21749 vs 18895), this result is another evidence of TextOCR’s good quality.

4.2. State-of-the-art on public benchmarks

In this section, using text recognition and end-to-end experiments again, we demonstrate that TextOCR is complementary to existing datasets, and training with it can improve model accuracy significantly on existing public benchmarks, and even outperform state-of-the-art.

For text recognition, we evaluate the state-of-the-art models based upon Baek et al. [1]³, as well as fine-tune them on TextOCR’s train set in Table 3 (rows #3-10). For each method, fine-tuning on TextOCR brings a significant increase on almost all datasets. The irregular datasets (*e.g.* ICDAR2015 [25], SVT Perspective [47] and CUTE80 [50]), gain most thanks to the rich diversity in TextOCR.

For end-to-end recognition, we fine-tuned the official weights by Mask TextSpotter V3 on TextOCR and Total Text. Table 4 (rows #3-8) again shows that adding TextOCR can further improve the F-measure on Total Text test set by 3.3% and 3.2% with none and weak lexicon respectively. Figure 4(a) shows qualitative examples of the results.

4.3. The challenges of TextOCR

Following others [12], we show the challenges of TextOCR, by evaluating pre-trained and TextOCR fine-tuned state-of-the-art methods on TextOCR test set. The end-to-end recognition results on TextOCR are evaluated in the same protocol as described in [31] following ICDAR2015 with support for polygon representation. All experiments were performed with a input short side of 1000 for fair com-

²<https://github.com/MhLiao/MaskTextSpotterV3>

³<https://github.com/clovaai/deep-text-recognition-benchmark>

#	Method	PW	Train Dataset	Test Dataset (Word accuracy)								
				IIIT	SVT	IC03	IC13	IC15	SVTP	CUTE	COCOText	TextOCR
Cross Dataset Experiments												
1	TPS-ResNet-BiLSTM-Attn [1]		COCOText	70.73	73.57	85.58	82.73	64.05	60.31	50.87	53.47	43.20
2	TPS-ResNet-BiLSTM-Attn [1]		TextOCR	80.50	82.84	92.16	91.25	80.07	76.74	70.38	64.03	65.65
Benchmarking state-of-the-art models and fine-tuning on TextOCR												
3	CRNN [5]	✓	S90k+ST	82.63	82.07	92.96	90.55	68.85	71.01	62.37	49.00	43.07
4	CRNN [51] (ours)		S90k+ST+TextOCR	85.97	87.94	92.96	93.70	79.85	78.30	73.87	60.09	58.61
5	Rosetta [5]	✓	S90k+ST	84.00	84.08	92.39	91.13	70.29	74.73	67.60	49.66	43.16
6	Rosetta [5] (ours)		S90k+ST+TextOCR	87.50	89.80	93.77	94.52	81.28	81.40	77.35	62.61	60.85
7	STAR-Net [34]	✓	S90k+ST	86.26	86.09	94.39	91.48	75.81	76.43	72.47	53.56	48.23
8	STAR-Net [34] (ours)		S90k+ST+TextOCR	90.30	92.12	94.69	95.33	86.09	83.88	83.62	67.92	66.84
9	TPS-ResNet-BiLSTM-Attn [1]	✓	S90k+ST	87.37	87.33	95.12	93.00	78.24	80.16	74.22	56.34	50.37
10	TPS-ResNet-BiLSTM-Attn [1] (ours)		S90k+ST+TextOCR	86.70	91.50	94.23	94.63	85.15	84.19	79.44	69.15	69.49

Table 3: **Text recognition experiments on the TextOCR dataset.** PW means the model uses public available weights. **S90k** and **ST** refer to the Synth90k [21] and SynthText [14] datasets respectively. Row #1-2 show the cross-dataset comparison between COCOText [59] and TextOCR. Row #3-10 show the experiments on state-of-the-art methods that including TextOCR in training can improve their word accuracy on most public benchmarks, as well as their word accuracy on TextOCR test set.

#	Method	Official	Train Dataset				Test Dataset (F-measure)				
			SynthText	Public	COCOText	TextOCR	TT (None)	TT (Full)	COCOText	TextOCR	
Cross Dataset Experiments											
1	Mask TextSpotter v3 [32]				✓			54.2	65.6	52.2	32.5
2	Mask TextSpotter v3 [32]					✓		64.8	74.1	52.4	45.8
Benchmarking state-of-the-art models and fine-tuning on TextOCR											
3	Qin et al. Inc-Res [48]	✓	✓	✓	✓	✓		63.9	—	—	—
4	Mask TextSpotter v2 [31]	✓	✓	✓	✓			65.3	77.4	47.6	—
5	Boundary TextSpotter [60]	✓	✓	✓	✓			65.0	76.1	41.3	—
6	ABCNet [35]	✓	✓	✓	✓	✓		64.2	75.7	—	30.5
7	Mask TextSpotter v3 [32]	✓	✓	✓	✓			71.2	78.4	46.1	34.9
8	Mask TextSpotter v3 [32] (ours)	✓	✓	✓	✓		✓	74.5	81.6	57.9	50.8

Table 4: **End-to-end recognition experiments on the TextOCR dataset.** Official means either using official weights (for testing on TextOCR) or offical reported results (other test data). Public refers to the model is trained with public real datasets [26, 25, 8, 43] other than COCOText [59] or TextOCR. TT is short for Total Text [8]. Row #1-2 show the cross-dataset comparison between COCOText and TextOCR. Row #3-7 show results of state-of-the-art methods, where TextOCR tests are obtained with official weights. Row #8 show improvements after fine-tuning with TextOCR train data.

parison. Note that TextOCR can benefit from higher short sides due to its high resolution.

Table 3 rows #3-10 and Table 4 rows #3-8 “TextOCR” column shows performance of state-of-the-art methods on text and end-to-end recognition tasks, respectively. The results demonstrate TextOCR’s challenge; even after fine-tuning with its own large train set of 21k images, the numbers are still much lower than other popular OCR datasets [25, 8], indicating a difficult task with a large room for improvement.

5. TextVQA and TextCaps Experiments

To evaluate the effectiveness and quality of TextOCR for downstream tasks, we calculate various heuristics and

conduct experiments on TextVQA and TextCaps dataset using PixelM4C with TextOCR trained OCR module.

5.1. Upper Bounds and Heuristics

First, we set new precedents for the TextVQA dataset in Table 5 by recalculating the OCR-based upper bounds (UB) and heuristics for its val set presented in [55] using Rosetta [6] OCR-en namespace, OCR tokens from TextOCR trained MTS v3 [32, 20], and the annotated text present in TextOCR.

The **human** accuracy (row #1) [55] stays the same at 85.01%. For UB, unlike [55], inspired by M4C’s iterative answer prediction, we calculate the accuracy using multi-word match checking whether the answer can be built using single or multiple token(s) from the source in consideration



Figure 4: (a) **Examples of Mask TextSpotter V3** [32] improvement on Total Text after fine-tuning on TextOCR compared to official weights; (b) **Failure cases by MTS V3** on TextOCR test set; (c) **Failure cases by Baek et al. [1]** on TextOCR test set

#	Method	TextVQA val accuracy(%)		
		Rosetta	MTS v3	TextOCR
1	Human	85.01	85.01	85.01
2	OCR UB	44.98	53.34	66.90
3	Vocab UB	59.02	59.02	59.02
4	OCR + Vocab UB	79.72	80.64	87.22
5	OCR Biggest	12.06	13.60	16.78
6	OCR Max	9.26	7.50	10.94

Table 5: **TextVQA heuristics.** Val accuracy for various heuristics compared with numbers from [55]. The comparison shows that TextOCR leads to much higher numbers than the original OCR tokens used in the TextVQA.

to cover all possibilities allowing better estimates M4C-like models’ UB. **OCR UB** (row #2) shows the UB achievable by only using OCR tokens and no vocabulary which is 11% and 22% higher for TextOCR compared to MTS v3 and Rosetta justifying the requirement of a better OCR mechanism while suggesting that training OCR systems on TextOCR would be crucial for TextVQA [55] and TextCaps [53]. **Vocab UB** (row #3) shows the UB achievable by only using a fixed word vocabulary (M4C 5k vocab). **OCR+Vocab UB** (row #4) is UB achievable using both OCR and vocab inter-changeably wherever suitable for prediction. For TextOCR, this surpasses the human accuracy indicating TextOCR’s high quality and the downstream benefits of improved OCR models. **OCR Biggest** and **OCR Max** (row #5 and #6) show the UB obtained by choosing biggest OCR box and the most occurring word in the scene text as an answer respectively advocating TextVQA’s difficulty, TextOCR’s quality and improvement room in current OCR systems.

5.2. Improving the state-of-the art

Given the positive results in Section 5.1, we naturally expect that TextOCR will help with downstream tasks as

well, as we know from literature [13, 24] that OCR is indeed an important aspect. Using TextOCR annotations directly will allow us to evaluate the reasoning capabilities or shortcomings of the TextVQA/TextCaps models in isolation from OCR inconsistencies. Furthermore, this also makes it possible to train an end-to-end model that can take images directly as an input, extract OCR tokens from them and then jointly reason over the object features, OCR and input text with a possibility of backpropagating via the recognition model.

We propose an end-to-end model, PixelM4C shown in Figure 1, that works directly on the images allowing us to test our hypotheses. Specifically, we connect the Mask TextSpotter (MTS) v3 trained on TextOCR with M4C. We extract the OCR tokens and features on-the-fly from MTS v3 and pass them to M4C model allowing more fine-grained control on which features to extract and which specific parts to use based on the downstream task and model. We achieve new state-of-the-art on TextVQA using PixelM4C which allows easy testing of our various hypotheses.

Training. We train PixelM4C and PixelM4C-Captioner (similar to M4C-Captioner) in an end-to-end fashion by extracting OCR tokens from MTS v3 in real time. We use same hyper-parameters and 5k vocabulary as used by M4C [19] but we set batch size to 16 given that model is slow and hard to train on larger batch sizes. We train with Adam [28] optimizer with 1e-4 learning rate, a step schedule and a linear warmup of 1k iterations. We train PixelM4C and PixelM4C-Captioner for 24k and 12k iterations. We decrease the learning rate to 1/10th at 14k and 19k for PixelM4C and 10k and 11k for PixelM4C-Captioner. We freeze MTS v3 during training as our empirical results suggested that fine-tuning predictor heads hurt TextVQA accuracy. We hypothesize that this happens because MTS v3 is trained using character-level losses while M4C is trained using word-level losses. Unlike M4C, we conduct ablations on using > 50 tokens

#	Method	OCR Source	OCR Feature	TextVQA		TextCaps val metrics				
				val acc (%)	B-4	M	R	S	C	
1	Human	—	—	85.01	24.40	26.10	47.00	18.80	125.50	
2	M4C / M4C-Captioner	Rosetta (OCR-en)	Object detector fc7	39.40	23.30	22.00	46.20	15.60	89.60	
3	M4C w/ STVQA	Rosetta (OCR-en)	Object detector fc7	40.55	—	—	—	—	—	
4	PixelM4C / PixelM4C-Captioner	MTS v3 (COCO-Text+TT)	MTS v3 fc7	37.61	23.09	21.08	45.55	14.51	81.44	
5	PixelM4C / PixelM4C-Captioner	MTS v3 (COCO-Text+TT)	MTS v3 LH	38.24	23.05	20.88	45.45	14.23	81.55	
6	PixelM4C / PixelM4C-Captioner	MTS v3 (TextOCR-en)	MTS v3 fc7	39.69	23.11	21.37	45.57	14.68	84.54	
7	PixelM4C / PixelM4C-Captioner	MTS v3 (TextOCR-en)	MTS v3 LH	40.67	23.41	21.45	45.69	14.75	86.87	
8	PixelM4C / PixelM4C-Captioner	MTS v3 (TextOCR)	MTS v3 fc7	39.64	23.01	21.27	45.65	14.58	84.99	
9	PixelM4C / PixelM4C-Captioner	MTS v3 (TextOCR)	MTS v3 LH	41.23	23.33	21.30	45.71	14.62	85.32	
10	PixelM4C w/ STVQA	MTS v3 (TextOCR)	MTS v3 LH	42.12	—	—	—	—	—	
11	PixelM4C / PixelM4C-Captioner	TextOCR	MTS v3 fc7 (from 8)	46.28	23.76	21.86	46.38	15.14	91.44	
12	PixelM4C / PixelM4C-Captioner	TextOCR	MTS v3 LH (from 9)	46.36	24.10	21.98	46.65	15.08	91.99	
13	PixelM4C w/ STVQA	TextOCR	MTS v3 LH (from 9)	48.04	—	—	—	—	—	

B-4 = Bleu4 [44], M = METEOR [2], R = ROUGE_L [33], M = METEOR [2], C = CIDEr [58]

Table 6: **PixelM4C experiments on TextVQA/TextCaps.** Val accuracy for ablations compared with M4C [19]. We show that OCR tokens and features from TextOCR trained models and directly help TextVQA and TextCaps models significantly.

given high word density in TextOCR. We train PixelM4C in a distributed fashion on 16 Nvidia Volta V100-SXM2-32GB GPUs using PyTorch based MMF framework [45, 54]..

Experiments and Results. We compare PixelM4C with M4C for TextVQA and PixelM4C-Captioner with M4C-Captioner for TextCaps. Table 6 shows results for various experiments and ablations. First, by an extensive sweep (details in appendix), we confirm that batch size of 16 performs better than batch size of 128 used in [19]. We test PixelM4C and PixelM4C-Captioner with four different OCR sources: MTS v3 trained (i) on COCO-Text and Total-Text (row #4 and #5) (ii) on TextOCR but using alphanumeric English only vocabulary (row #6 and #7) (iii) on TextOCR using 240 characters Latin vocabulary (row #8, #9, and #10), (iv) using TextOCR annotations directly as the OCR source (row #11, #12, and #13), extracting features using annotation boxes as the proposals from (iii). Enabled by our end-to-end PixelM4C model, we revisit the choice of OCR feature in [19] and try other features from MTS v3. We found that using last hidden state from prediction decoder (“MTS v3 LH” in Table 6) for < EOS > token as the OCR representation improves performance. Finally, we add ST-VQA [4] as extra annotation data following M4C [19] (row #10 and #13).

Based on our ablations (see appendix), we use 200 tokens instead of 50 in all experiments. MTS v3’s fc7 features as OCR representation boost accuracy when compared to Visual Genome pretrained FRCNN [49, 29] ones (row #2 vs #8). Further, we achieve state-of-the-art performance using prediction decoder’s last hidden state (row #9 vs #3) when compared to fc7 (row #4 vs #5, #6 vs #7, #8 vs #9, and #11 vs #12) suggesting that MTS v3 representations including decoder’s hidden state contain more relevant information for TextVQA task. Comparing row #5 with #7 and #9, we observe that TextOCR trained models provide better OCR tokens for TextVQA compared to COCO-Text+TT trained

ones. Finally, adding STVQA as additional data boosts the performance to 42.12% setting new state-of-the-art over M4C. In TextCaps, unfortunately, we don’t see significant improvement in metrics except B4 using TextOCR trained model’s OCR tokens testifying TextCaps’s complexity.

Using TextOCR directly as the OCR source gives a significant boost in TextVQA accuracy (6%) and TextCaps metrics (3%) signaling that apart from the gap in reasoning capabilities, there is still a room for improvement in OCR capabilities of the OCR module (MTS v3)⁴.

6. Conclusion

In this work, we introduced the large arbitrary scene text recognition dataset, TextOCR, collected on TextVQA images along with an end-to-end model, PixelM4C, that can perform scene-text reasoning directly on images by incorporating text recognition model as a module. Training on TextOCR, provides better text-recognition models which outperforms state-of-the-art on most text-recognition benchmarks. Further, using TextOCR trained text-recognition module in PixelM4C allows us to use different features from it with a possibility of even providing feedback which results in PixelM4C surpassing existing state-of-the-art methods on TextVQA. Through TextOCR dataset and PixelM4C model, we take a step towards bridging the communities of OCR and downstream applications based on OCR and hope that research from community will advance both fields at the same time as there is a large room for improvement as evident from TextVQA results from training directly on TextOCR.

⁴We don’t claim this as state-of-the-art because it would be non-ideal for community to train directly on TextOCR except for understanding reasoning capabilities in isolation from OCR systems.

References

- [1] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyo 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. 5, 6, 7, 12, 13
- [2] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 8
- [3] Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, et al. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342, 2010. 3
- [4] 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. 1, 3, 8, 14
- [5] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *KDD*, 2018. 6, 12
- [6] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *KDD*, 2018. 6
- [7] Ritwick Chaudhry, Sumit Shekhar, Utkarsh Gupta, Pranav Maneriker, Prann Bansal, and Ajay Joshi. Leaf-qa: Locate, encode & attend for figure question answering. In *Winter Conf. on App. of Comput. Vision*, pages 3512–3521, 2020. 3
- [8] Chee Kheng Ch'ng, Chee Seng Chan, and Cheng-Lin Liu. Total-text: Towards orientation robustness in scene text detection. *Int. J. on Document Analysis and Recognition*, 2019. 1, 2, 4, 6
- [9] Chee-Kheng Chng, Yuliang Liu, Yipeng Sun, Chun Chet Ng, Canjie Luo, Zihan Ni, ChuanMing Fang, Shuaítao Zhang, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, Chee Seng Chan, and Lianwen Jin. Icdar2019 robust reading challenge on arbitrary-shaped text (rrc-art). In *Proc. Int. Conf. on Document Analysis and Recognition*, 2019. 1, 2, 4
- [10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2009. 3
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3
- [12] Eran Eidinger, Roee Enbar, and Tal Hassner. Age and gender estimation of unfiltered faces. *Trans. on Inform. Forensics and Security*, 9(12):2170–2179, 2014. 5
- [13] Chenyu Gao, Qi Zhu, Peng Wang, Hui Li, Y. Liu, A. V. D. Hengel, and Qi Wu. Structured multimodal attentions for textvqa. *ArXiv*, abs/2006.00753, 2020. 3, 7
- [14] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2016. 3, 4, 6
- [15] Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning images taken by people who are blind. *arXiv preprint arXiv:2002.08565*, 2020. 3
- [16] Wei Han, Hantao Huang, and T. Han. Finding the evidence: Localization-aware answer prediction for text visual question answering. *ArXiv*, abs/2010.02582, 2020. 3
- [17] Tal Hassner, Malte Rehbein, Peter A Stokes, and Lior Wolf. Computation and palaeography: potentials and limits. *Dagstuhl Reports*, 2(9):184–199, 2012. 1
- [18] Tal Hassner, Robert Sablatnig, Dominique Stutzmann, and Sérgolène Tarte. Digital palaeography: New machines and old texts (dagstuhl seminar 14302). *Dagstuhl Reports*, 4(7), 2014. 1
- [19] 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, 3, 7, 8, 14
- [20] Jing Huang, Guan Pang, Rama Kovvuri, Mandy Toh, Kevin J Liang, Praveen Krishnan, Xi Yin, and Tal Hassner. A multiplexed network for end-to-end, multilingual ocr. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 6
- [21] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. In *Neural Inform. Process. Syst.*, 2014. 3, 4, 6
- [22] Zan-Xia Jin, Heran Wu, C. Yang, Fang Zhou, Jingyan Qin, Lei Xiao, and XuCheng Yin. Ruart: A novel text-centered solution for text-based visual question answering. *ArXiv*, abs/2010.12917, 2020. 3
- [23] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomas Mikolov. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016. 3
- [24] Yash Kant, Dhruv Batra, Peter Anderson, A. Schwing, D. Parikh, Jiasen Lu, and Harsh Agrawal. Spatially aware multi-modal transformers for textvqa. In *European Conf. Comput. Vision*, 2020. 3, 7
- [25] 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. 1, 2, 3, 4, 5, 6, 13
- [26] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493. IEEE, 2013. 1, 2, 3, 4, 6

- [27] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. *arXiv preprint arXiv:2005.04790*, 2020. 1
- [28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 7
- [29] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 3, 8
- [30] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018. 3
- [31] 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. 5, 6, 12
- [32] 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. 2, 5, 6, 7, 12
- [33] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 8
- [34] 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. 6, 12
- [35] Yuliang Liu*, Hao Chen*, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2020. 6
- [36] Yuliang Liu, Lianwen Jin, Shuaifeng Zhang, and Sheng Zhang. Detecting curve text in the wild: New dataset and new solution. *arXiv preprint arXiv:1712.02170*, 2017. 1, 2, 4
- [37] Simon M Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, and Robert Young. Icdar 2003 robust reading competitions. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 682–687, 2003. 1, 2, 4
- [38] Minesh Mathew, Dimosthenis Karatzas, R. Manmatha, and C. V. Jawahar. DocVQA: A dataset for vqa on document images, 2020. 3
- [39] Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. Plotqa: Reasoning over scientific plots. In *Winter Conf. on App. of Comput. Vision*, pages 1527–1536, 2020. 3
- [40] Anand Mishra, Kartek Alahari, and C.V. Jawahar. Scene text recognition using higher order language priors. In *Proc. British Mach. Vision Conf.*, pages 1–11, 2012. 1, 2, 4
- [41] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 947–952. IEEE, 2019. 1, 3
- [42] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khelif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Chenglin Liu, and Jean-Marc Ogier. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition – rrc-mlt-2019. In *Proc. Int. Conf. on Document Analysis and Recognition*, 2019. 1
- [43] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon, W. Khelif, M. M. Luqman, J.-C. Burie, C.-L. Liu, and J.-M. Ogier. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification - rrc-mlt. In *Proc. Int. Conf. on Document Analysis and Recognition*, 2017. 1, 2, 4, 6
- [44] Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. Bleu: a method for automatic evaluation of machine translation. 10 2002. 8
- [45] 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. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Proceedings of NeurIPS*, pages 8024–8035. Curran Associates, Inc., 2019. 8, 12
- [46] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. pages 1532–1543, 2014. 3
- [47] Trung Quy Phan, Palaiahankote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *Proc. Int. Conf. Comput. Vision*, pages 569–576, 2013. 1, 2, 4, 5
- [48] Siyang Qin, Alessandro Bissacco, Michalis Raptis, Yasuhisa Fujii, and Ying Xiao. Towards unconstrained end-to-end text spotting. In *Proc. Int. Conf. Comput. Vision*, 2019. 6
- [49] Shaoting Ren, Kaiming He, Ross B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Trans. Pattern Anal. Mach. Intell.*, 39:1137–1149, 2015. 8
- [50] Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. *Expert Systems with Applications*, 41(18):8027–8048, 2014. 1, 2, 4, 5
- [51] 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. 6, 12
- [52] Baoguang Shi, Cong Yao, Minghui Liao, Mingkun Yang, Pei Xu, Linyan Cui, Serge Belongie, Shijian Lu, and Xiang Bai. Icdar2017 competition on reading chinese text in the wild (rctw-17). In *Proc. Int. Conf. on Document Analysis and Recognition*, 2017. 1, 2, 4
- [53] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with

- reading comprehension. *arXiv preprint arXiv:2003.12462*, 2020. 1, 2, 3, 7, 14
- [54] Amanpreet Singh, Vedanuj Goswami, Vivek Natarajan, Yu Jiang, Xinlei Chen, Meet Shah, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Mmf: A multimodal framework for vision and language research. <https://github.com/facebookresearch/mmf>, 2020. 8
- [55] 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. 1, 2, 3, 6, 7, 14
- [56] Yipeng Sun, Zihan Ni, Chee-Kheng Cheng, Yuliang Liu, Canjie Luo, Chun Chet Ng, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, Chee Seng Chan, and Lianwen Jin. Icdar 2019 competition on large-scale street view text with partial labeling – rrc-lsvt. In *Proc. Int. Conf. on Document Analysis and Recognition*, 2019. 1, 3, 4
- [57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3
- [58] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 4566–4575, 2015. 8
- [59] 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. 1, 3, 4, 5, 6, 13
- [60] Hao Wang*, Pu Lu*, Hui Zhang*, Mingkun Yang, Xiang Bai, Yongchao Xu, Mengchao He, Yongpan Wang, and Wenyu Liu. All you need is boundary: Toward arbitrary-shaped text spotting. In *AAAI Conf. on Artificial Intelligence*, 2020. 6
- [61] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *Proc. Int. Conf. Comput. Vision*, pages 1457–1464, 2011. 1, 2, 4
- [62] Xinyu Wang, Yuliang Liu, Chunhua Shen, Chun Chet Ng, Canjie Luo, Lianwen Jin, Chee Seng Chan, Anton van den Hengel, and Liangwei Wang. On the general value of evidence, and bilingual scene-text visual question answering. In *Proc. Conf. Comput. Vision Pattern Recognition*, June 2020. 3
- [63] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2012. 1, 2, 4

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

(Supplementary Material)



Figure A.1: **Annotation UI** used for TextOCR. UI allows annotating arbitrary shaped text as polygons.

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, and shows TextOCR is more uniformly annotated and distributed than existing datasets.

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]⁵ for

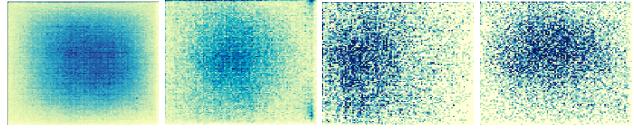


Figure B.1: **Word location heatmap** comparison. TextOCR has more uniform and dense distribution of text instances compared to other datasets.

text recognition task. We experimented with 4 models, including CRNN [51] (None-VGG-BiLSTM-CTC in [1]), Rosetta [5] (None-ResNet-None-CTC in [1]), STAR-Net [34] (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 pretrained 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. [32]⁶. 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 models using 8 Tesla Volta V100-SXM2-32GB GPUs in a distributed fashion using PyTorch [45]. During evaluation, we measure with the same protocol as described in [31] that follows ICDAR2015 with support for polygon representation, and the short side of input images resized to 1000.

⁵<https://github.com/clovaai/deep-text-recognition-benchmark>

⁶<https://github.com/MhLiao/MaskTextSpotterV3>



Figure C.1: More TextOCR annotation samples

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 [25] and COCO-Text [59]. 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 generaliza-

tion.

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

Table D.1: **Text recognition performance using same number of instances in TextOCR** as in ICDAR15 and COCO-Text. TextOCR achieves better performance indicating its superior quality.

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.
1	50 tokens	MTS v3 (TextOCR-en)	37.75
2	50 tokens	TextOCR	45.22
3	100 tokens	MTS v3 (TextOCR-en)	39.41
4	100 tokens	TextOCR	46.42
5	200 tokens	MTS v3 (TextOCR-en)	39.41
6	200 tokens	TextOCR	46.12
7	200 tokens	MTS v3 (TextOCR-en-LH)	40.31
8	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 [55] 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 [55] and TextCaps [53] 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 [4] train set and test on its validation set created in [19]. We get an accuracy of 38.49% and 47.89% ANLS better than that reported in [19] 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), “im 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 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