PreSTU: Pre-Training for Scene-Text Understanding

Jihyung Kil[1*] Soravit Changpinyo[2]
Xi Chen[2] Hexiang Hu[2] Sebastian Goodman[2] Wei-Lun Chao[1] Radu Soricut[2]
[1]The Ohio State University [2]Google Research
{kil.5,chao.209}@osu.edu
{schangpi,chillxichen,hexiang,seabass,rsoricut}@google.com

Abstract

The ability to recognize and reason about text embedded in visual inputs is often lacking in vision-and-language (V&L) models, perhaps because V&L pre-training methods have often failed to include such an ability in their training objective. In this paper, we propose PRESTU*, a novel pre-training recipe dedicated to scene-text understanding (STU).* PRESTU *introduces OCR-aware pre-training objectives that encourage the model to recognize text from an image and connect it to the rest of the image content. We implement* PRESTU *using a simple transformer-based encoder-decoder architecture, combined with large-scale image-text datasets with scene text obtained from an off-the-shelf OCR system. We empirically demonstrate the effectiveness of this pre-training approach on eight visual question answering and four image captioning benchmarks.*

1. Introduction

Understanding the role of text as it appears in the context of a visual scene is important in various real-world applications, *e.g.*, from automatically organizing images of receipts, to assisting visually-impaired users in overcoming challenges related to comprehension of non-Braille writing in their surroundings, to enabling autonomous robots to make safe decisions in environments designed for humans. As a result, scene-text understanding (STU) has received increased attention in vision-and-language (V&L) understanding tasks, such as visual question answering (VQA) [46, 5, 40, 55, 38, 37, 36] or image captioning [45, 16, 30]. Please see Figure 1 for an illustration.

We identify two distinct capabilities that models targeting STU must address: (i) *recognizing* text in a visual scene and (ii) *connecting* the text to its context in the scene. Previous solutions that target STU tasks [46, 45, 19, 59] often delegate scene-text recognition to off-the-shelf OCR (Optical Character Recognition) systems [45, 7] and model the visual context using pre-computed object-detection features. These two streams of information (noisy OCR strings and visual features on detected objects) are used as input into a V&L model. While achieving decent results, these methods heavily rely on the quality of the upstream OCR system and lack a direct connection between the text being recognized and a high-fidelity representation of its context.

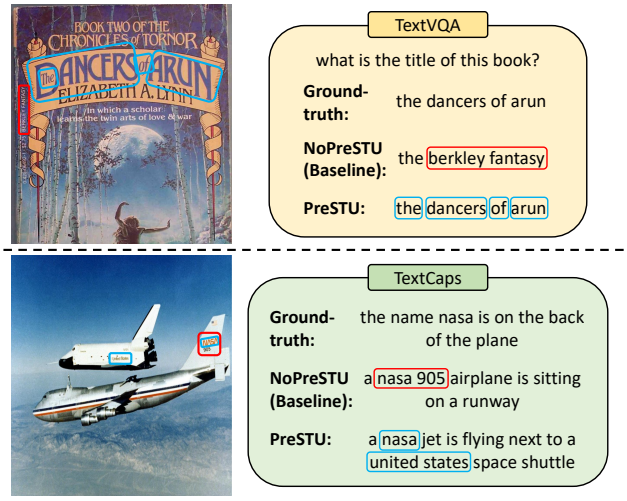


Figure 1: **Example of scene-text understanding (STU) tasks.** NOPRESTU (baseline) and PRESTU share the same V&L model, but PRESTU is pre-trained on our proposed pre-training objectives. Scene texts are highlighted by bounding boxes. Unlike the baseline, PRESTU correctly predicts the title of the book on scene-text VQA (TextVQA [46]) and even generates a more detailed scene-text caption (*e.g.*, "united states space shuttle") than the ground-truth annotated by humans (TextCaps [45]).

More concretely, previous methods have not fully explored pre-training objectives that specifically target STU. In general, V&L pre-training objectives (*e.g.*, masked language modeling, image-text matching [33], etc.) have been proven effective for learning and became the go-to approach in V&L research. However, these objectives typically do

[*] Work done at Google Research.

not require a model to understand the role of text embedded in a visual context. For instance, LaTr [4] ignores the visual context during pre-training and instead focuses on modeling the co-occurrence statistics of layout-aware text-only OCR tokens. Even in systems that do perform STU pre-training, such as TAP [59], their models are built upon the aforementioned pipeline. Specifically, TAP represents the visual input by a set of object features detected and extracted by FRCNN [43]. As a result, it may lose some visual contexts that cannot be captured by objectness (*e.g.*, activities) but are relevant to understand the role of recognized text.

In this paper, we address such a challenge by incorporating an OCR-aware learning objective in the context of a high-fidelity representation of the image context. We adopt a Transformer-based [48] encoder-decoder V&L architecture, using a T5 [42] backbone. The model takes both image and text inputs. For the former, we extract fine-tunable visual features directly from image *pixels* using a ViT [12] encoder, rather than adopting frozen visual features from pre-detected objects [43]. For the latter, we concatenate task-specific text tokens (*e.g.*, task prompts) with tokens extracted from an off-the-shelf OCR system, in a manner that allows the model to interpret (via the prompt) the OCR tokens in the context of the image.

Building upon this model, we propose PRESTU, a novel recipe for **Pre**-training for **S**cene-**T**ext **U**nderstanding (Figure 2). PRESTU consists of two main steps. First, it teaches the model to recognize scene text from image pixels[1] and at the same time connect scene text to the visual context. Specifically, given an image and the "part" of the scene texts in the image, the model is pre-trained to predict the "rest" of the scene texts. We call this step SPLITOCR. Second, it teaches the model to further strengthen the connection between scene text and visual context by pre-training with OCR-aware downstream tasks (*e.g.*, VQA and CAP). For pre-training, we leverage large-scale image-text resources [44, 8, 5], with the (noisy) scene text extracted by the off-the-shelf OCR system (Google Cloud OCR[2]).

We validate PRESTU on eight VQA (ST-VQA [5], TextVQA [46], VizWiz-VQA [15], VQAv2 [14], OCR-VQA [40], DocVQA [38], ChartQA [36], AI2D [26]) and four image captioning (TextCaps [45], VizWiz-Captions [16], WidgetCap [30], Screen2Words [51]) benchmarks. Our OCR-aware objectives SPLITOCR, VQA, and CAP are significantly beneficial. For instance, compared with strong baselines which take OCR signals as input, we observe more than 10% absolute gain on TextVQA and 42 CIDEr point gains on TextCaps (Figure 1). Finally, we conduct comprehensive experiments to understand which factors contribute to effective STU pre-training. In summary, our contributions are as follows:

- We propose PRESTU, a simple and effective pre-training recipe with OCR-aware objectives designed for scene-text understanding (§2).
- We show that our objectives consistently lead to improved scene-text understanding on twelve diverse downstream VQA / image captioning tasks (§3.1) and even on cases when OCR signals are absent during downstream tasks (§3.2).
- We perform detailed analyses to understand the effect of our design choices on STU performance (§3.2).

2. PreSTU: Pre-Training for Scene-Text Understanding

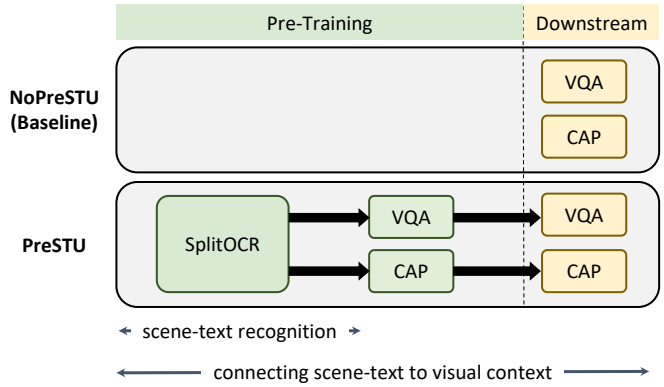
Figure 2 provides an overview of PRESTU OCR-aware objectives and their input-output format. In what follows, we first describe our starting point: model architecture and OCR signals (§2.1). Then, we describe our recipe for pre-training (§2.2), including the objectives, **SPLITOCR**, **VQA**, and **CAP** (§2.2.1), and data sources (§2.2.2). Finally, we describe the fine-tuning stage and target benchmarks (§2.3).

2.1. Setup

V&L model architecture. Our main architecture is illustrated in Figure 3. We start from an encoder-decoder V&L architecture which unifies image-to-text (e.g., image captioning) and image+text-to-text (e.g., VQA) tasks. The pre-trained vision encoder is ViT-B/16 [12], and the pre-trained language encoder-decoder is mT5-Base [58]. Specifically, ViT is a transformer-based encoder that takes a sequence of image patches as input, pre-trained on an image classification task. mT5 is a multilingual variant of text-to-text transformers T5 [42], pre-trained on a massive multilingual text corpus with the span corruption objective. See more details in the supplementary material.

As mentioned in LaTr [4], this starting point leads to modeling advantages over existing model architectures for STU tasks. First, we believe that understanding the role of OCR text in the visual context is much easier from image pixels, making ViT a natural choice. Second, mT5 uses wordpiece vocab to encode and decode text tokens; thus a certain level of robustness to the noise in the input OCR texts comes with it by default. On the other hand, M4C [19] and TAP [59] resort to a more complicated solution of using fastText [6] and Pyramidal Histogram of Characters features [2]. Third, mT5 is an encoder-decoder model which enables to generate the open-ended text. This is suitable for general image captioning and scene-text VQA where the answers tend to be out-of-vocab. In contrast, most prior works [46, 19, 59, 52, 34] treat VQA as answer vocab-based classification. Lastly, our model is built upon well-developed vanilla unimodal building blocks in vision and NLP. We deliberately choose this general encoder-decoder

[1] This makes our model more robust to the quality of OCR systems.
[2] https://cloud.google.com/vision/docs/ocr



Objective	Text Input	Output
SplitOCR	Generate ocr_text in en: $<OCR_1>$ $<OCR_2>$...$<OCR_m>$	$<OCR_{m+1}>$...$<OCR_N>$
VQA	Answer in en: <Question> $<OCR_1>$ $<OCR_2>$...$<OCR_N>$	<Answer>
CAP	Generate alt_text in en: $<OCR_1>$ $<OCR_2>$...$<OCR_N>$	<Caption>

Figure 2: **Our proposed pipeline.** Left: Comparison between PRESTU and NOPRESTU (baseline) we want to compare against. Green denotes the PRESTU pre-training phase and yellow the downstream/fine-tuning phase. SPLITOCR encourages scene-text recognition as well as the learning of the connection between scene text and its visual context; VQA and CAP further strengthen that connection. Right: The text input and output for each objective. All objectives utilize OCR signals. See Figure 3 for the architecture of PRESTU.

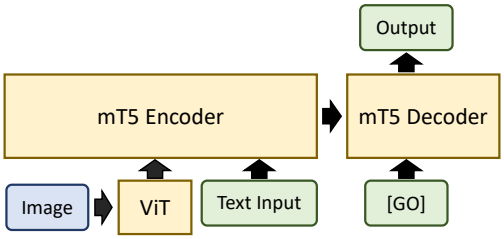


Figure 3: **V&L model architecture used in all of our experiments.** We use a simple transformer-based encoder-decoder (pre-trained ViT [12] + mT5 [58]) transforming image and text inputs to the text output. Green box: text input/output. Blue box: visual input. Yellow box: model blocks. See Figure 2 for the input-output pairs for different objectives.

architecture to push for the applicability of our objectives. Such a design choice allows us to develop less model-dependent pre-training objectives.

Image resolution. Unless stated otherwise, we use the image resolution of 640x640 in all of our experiments.

OCR signals. We obtain OCR signals from Google Cloud OCR for all pre-training and downstream datasets in our experiments. They come in the form of a set of texts and their corresponding box coordinates in the image (*i.e.*, object detection-like). We order OCR texts based on their locations, top-left to bottom-right and concatenate them with the T5 separator </s>. This allows models to implicitly learn the scene text's spatial information and standarize the target output sequence during training. Unless stated otherwise, we use these sorted *silver* OCR texts in all of our experiments.

2.2. Pre-Training Stage

2.2.1 PreSTU Objectives

We consider two sets of OCR-aware pre-training objectives for scene-text understanding.

Task-agnostic objective: SplitOCR. Inspired by the impressive performance of the visual language modeling pre-training objective for image+text-to-text downstream tasks [56], we propose an OCR-aware pre-training objective called SPLITOCR. This objective is designed to be downstream task-agnostic, focusing on teaching the two core capabilities for STU: recognizing scene text and connecting it to the visual context.

We randomly split the OCR texts into two parts and use the first part as additional input and the second part as a target. Recall that we have ordered the OCR texts based on their locations such that the model can recognize them in a consistent manner. Note that if the splitting point is right at the beginning of the OCR sequence, the model performs a simplified version of the traditional Optical Character Recognition task (*i.e.*, predicting the whole OCR tokens). We denote this by OCR in Table 6 and also compare it with SPLITOCR in our ablation studies.

Why SPLITOCR? SPLITOCR equips the model with the abilities to recognize scene text and connect it to the visual context in a unified, seamless manner. Specifically, operating SPLITOCR upon the "first part" of OCR tokens and the image pixels (not pre-extracted global or object detection features) and predicting the "second part" of OCR tokens requires the model to (i) identify which scene text in the image *still* needs to be recognized, inherently connecting the input scene text to its visual context; (ii) perform the OCR *task*, inherently acquiring the scene-text recognition skill.

Task-specific objectives: VQA and CAP. We propose

OCR-aware downstream-task-specific pre-training objectives on top of SPLITOCR. We consider two objectives based on our downstream tasks: (i) VQA which predicts the target answer from the question prompt, the visual question, and OCR texts and (ii) CAP which predicts the target caption from the caption prompt and OCR texts. This is similar to previous approaches to STU, except that we encode the image pixels, not features from pre-detected regions.

Why VQA or CAP? Task-specific objectives aim to achieve two goals. First, they further encourage the learning of the relationship between scene text and its visual context through direct interaction between input image pixels and input OCR texts. Second, it eases the knowledge transfer from pre-training to fine-tuning since task-specific objectives share the same input format as that of the downstream tasks (§2.3). See Figure 2 for more details.

2.2.2 Pre-Training Data

Our main pre-training data is CC15M, the union of two popular image-text datasets: Conceptual Captions (CC3M) [44] and Conceptual 12M (CC12M) [8].[3] CC3M consists of 3.3M $\langle image, caption \rangle$ pairs, obtained by processing raw alt-text descriptions from the Web. CC12M extends CC3M by relaxing its over-restrictive filtering pipeline. We use CC15M for SPLITOCR and CAP pre-training. Note that the captions of CC15M are not used for SPLITOCR and their images are not necessarily scene text-related. See more details in the supplementary material.

Since CC15M does not have data in the form of visual questions and their answers for us to leverage, we resort to ST-VQA [5]. It is a scene-text VQA dataset whose images are collected from 6 diverse data sources (COCO-Text [50], Visual Genome [27], VizWiz [15], ICDAR [25, 24], ImageNet [10], IIIT-STR [39]). We use its training set for pre-training. We use ST-VQA as pre-training data for other VQA benchmarks as well as a downstream benchmark for testing SPLITOCR (§2.3).

2.3. Fine-tuning Stage

In all of our downstream scene-text V&L tasks, the input-output pairs follow the same format as either VQA or CAP (with OCR text tokens as input.) The only difference from the task-specific pre-training is the training data.

We validate PRESTU on twelve datasets related to VQA and image captioning tasks. ST-VQA, TextVQA, and TextCaps are the main benchmarks for STU. We also consider other scene-text domains, including book (OCR-VQA), document (DocVQA), illustration (ChartQA), diagram (AI2D), and screenshot domains (WidgetCap and Screen2Words). VizWiz-VQA and VizWiz-Captions are for the blind and heavily involve STU. VQAv2 is a general VQA dataset. See complete details in the supplementary material.

[3] Due to expired URLs, only 13M $\langle image, caption \rangle$ pairs are used in our experiments.

2.4. Discussion

We compare PRESTU with two well-known prior STU works TAP [59] and LaTr [4]. In terms of modeling, TAP leverages two conventional V&L objectives: visual-region masked language modeling and image-text matching, as well as the objective of learning the relative spatial position of two OCR text detections. TAP models the image using object-based features [43], which we believe is a suboptimal visual context. Besides, TAP adopts vocab-based classification, less suitable for some STU tasks which are full of out-of-vocab words. LaTr overcomes those weaknesses by adopting a similar V&L architecture to ours (ViT-B/16 / $T5_{large}$). However, its pre-training objective does not involve the visual component (ViT). Instead, it only pre-trains its language component to learn the co-occurrence statistics of layout-aware OCR tokens. As the visual component is distorted or absent during pre-training, these models do not inherently learn the two essential STU capabilities, and would likely suffer in a case when OCR signals are absent during downstream tasks. In contrast, PRESTU fully embraces the visual component. As shown in §3.2, this brings a huge benefit especially when OCR signals are not available. See a more detailed comparison in §3.1.4.

In terms of pre-training data, TAP aggregates scene-text *dedicated* downstream data, including ST-VQA, TextVQA, TextCaps, and OCR-CC. Thus, while it aligns well with the corresponding downstream tasks, it is less generalizable to other V&L tasks. In contrast, PRESTU adopts *general* pre-training data (*i.e.*, CC15M), providing a more flexible interface for V&L tasks. Besides, LaTr argues that pre-training on document images is a better choice since acquiring large quantities of natural images with scene text for pre-training is challenging and hard to scale, and the amount of text is often sparse. Our work challenges this assumption and shows that one can pre-train effectively for STU on natural images with minimal preprocessing. (*i.e.*, nothing beyond extracting OCR signals).

Finally, in terms of evaluation as we will show next, our experiments are done on a much wider range of benchmarks than before. This is in stark contrast to existing works which often focus on three benchmarks at most.

3. Experimental Results

Baselines. We denote by NOPRESTU our main baseline. It is the same pre-trained V&L model as PRESTU (*i.e.*, ViT-B/16 / mT5) but *not* pre-trained with any of our pre-training objectives.

Metrics. For VQA tasks, we use standard VQA accuracy

Model	Pre-training Objective	Test Benchmark			
		ST-VQA ANLS	TextVQA Acc	VizWiz-VQA Acc	VQAv2 Acc
NoPreSTU	-	56.7	44.8	57.7 / 57.2	74.8 / 75.2
PreSTU	VQA	N/A	48.3	58.3 / 57.6	75.0 / 75.0
	SplitOCR	**65.5**	55.2	61.9 / 61.3	**76.0 / 76.2**
	SplitOCR→VQA	N/A	**56.3**	**62.5 / 62.0**	**76.1 / 76.1**

Table 1: **Effectiveness of PreSTU objectives on VQA.** Our pre-training objectives (VQA, SplitOCR, SplitOCR→VQA) show consistent gains over the baseline on all VQA benchmarks. We use CC15M for SplitOCR pre-training and ST-VQA for VQA pre-training. Since ST-VQA for VQA pre-training, we mark VQA and SplitOCR→VQA as "N/A". Results are reported on the test set for ST-VQA, test-std for TextVQA, and test-dev/test-std for VizWiz-VQA and VQAv2.

Model	Pre-training Objective	TextCaps test-std					VizWiz-Captions test-std				
		B	M	R	S	C	B	M	R	S	C
NoPreSTU	-	23.4	21.0	45.0	13.6	96.9	29.4	22.6	49.9	18.5	87.2
PreSTU	Cap	31.6	25.6	51.5	18.7	133.1	33.7	24.5	52.8	20.8	103.1
	SplitOCR	28.5	23.9	48.9	16.3	126.1	29.8	22.6	50.3	18.6	90.2
	SplitOCR→Cap	**32.8**	**26.2**	**52.2**	**19.1**	**139.1**	**34.3**	**24.7**	**53.4**	**21.1**	**105.6**

Table 2: **Effectiveness of PreSTU objectives on image captioning.** Our pre-training objectives (Cap, SplitOCR, SplitOCR→Cap) show significant gains over the baseline on all image captioning benchmarks, with SplitOCR→Cap performing best. We use CC15M for both SplitOCR and Cap pre-training. B: BLEU@4, M: METEOR, R: ROUGE-L, S: SPICE, C: CIDEr.

following [46, 59, 53]. It is the average score over nine subsets of the ground-truth ten answers, where each score is: $min(\frac{\#answer\ occurrences}{3}, 1)$. For ST-VQA/DocVQA, we use Average Normalized Levenshtein Similarity (ANLS), *softly* penalizing the model's mistakes on scene-text recognition. For ChartQA, we report its official metric, a relaxed accuracy that allows a minor inaccuracy for numeric answers. For image captioning tasks, we use their standard evaluation metrics, including BLEU [41], METEOR [11], ROUGE-L [31], SPICE [3], and CIDEr [49].

3.1. Main Results

The main goal of our experiments is to assess the utility of our pre-training objectives SplitOCR and VQA/Cap in VQA (§3.1.1) and image captioning (§3.1.2) tasks.

3.1.1 VQA

Table 1 summarizes our main results on VQA tasks, including ST-VQA, TextVQA, VizWiz-VQA, and VQAv2. SplitOCR outperforms the baseline (*i.e.*, without our STU pre-training) by a large margin on scene-text-heavy VQA tasks, more than +8.8 ANLS on ST-VQA, +10.4% on TextVQA, and +4.1% on VizWiz-VQA. With SplitOCR→VQA, we slightly but significantly improve the performance further on TextVQA and VizWiz-VQA, +1.1% and 0.7%, respectively. These results show the utility and applicability of our pre-training objectives for improving scene-text understanding.

SplitOCR and VQA are complementary on scene-text-heavy VQA tasks (TextVQA/VizWiz-VQA), where each of them alone underperforms SplitOCR→VQA. Additionally, we observe the first-stage pre-training via SplitOCR is more beneficial than the second-stage task-specific pre-training VQA. This could be due to the superiority of SplitOCR or the lack of large-scale scene-text VQA pre-training data, or both. We identify data development for scene-text VQA as an open research question.

Our results also highlight the importance of STU in general real-world VQA (*i.e.*, not specially designed for STU). We observe a slight but significant improvement over the baseline on VQAv2 and a more significant improvement on VizWiz-VQA for blind people. We attribute this to a subset of questions that require text recognition and reasoning skills [60]. We believe this is an important step since these questions are considered "hard to learn" or even "outliers" that work against VQA algorithms [47, 23].

3.1.2 Image Captioning

Table 2 summarizes our main results on image captioning tasks, TextCaps and VizWiz-Captions. Aligned with the VQA results, SplitOCR significantly improves over the baseline across all evaluation metrics, with SplitOCR→Cap performing best. The gain is notably 42.2 CIDEr points on TextCaps, and 18.4 on VizWiz-Captions. Overall, we highlight the usefulness of SplitOCR across V&L tasks with different input-output formats.

Model	OCR VQA %Acc	Doc VQA %ANLS	Chart QA %RelaxedAcc	AI2D %Acc	Widget Cap CIDEr	Screen2 Words CIDEr
NoPreSTU	71.5	47.5	40.5	64.5	63.9	98.5
PreSTU-SplitOCR	72.2	50.1	50.7	69.3	125.6	113.8

Table 3: **PreSTU on other scene-text domains (Val split).** See §3.1.3 for a detailed discussion.

Similar to the VQA results, SPLITOCR and CAP are complementary. However, CAP alone is more beneficial than SPLITOCR alone. We attribute this to our large-scale web-based image-text data that is already suitable for CAP pre-training. Despite such a strong CAP model, SPLITOCR still provides an additional benefit.

3.1.3 Applicability to Other Scene-Text Domains

Unlike prior STU literature [59, 52, 34, 4, 54, 13], we further explore other scene-text domains (Table 3). We show that PreSTU is also effective on book (OCR-VQA), document (DocVQA), illustration (ChartQA), diagram (AI2D), and screenshot domains (WidgetCap & Screen2Words). This demonstrates the applicability of PRESTU to many different real-world STU problems.

3.1.4 Comparison to Prior Works

So far our results provide strong evidence for the benefit of our proposed objectives. In this section, we provide a comparison to prior works as further context. While apples-to-apples comparison has become increasingly difficult, we make our best attempt to analyze our results in the context of these works. For example, TAP's objective has coupled the use of object detection signals, which we do not resort to. More importantly, many prior works [4, 1, 53] do not release code, rely on private data, and/or require too large-scale pre-training that is prohibitively costly to reproduce.

We first compare PRESTU to recent works focusing on STU tasks (Rows Non-TAP to LaTr in Table 4). Overall, PRESTU establishes strong results on all tasks. Concretely, PreSTU achieves better results than all prior smaller-scale works (*i.e.*, TAP, TAG, LOGOS). More interestingly, with much less data, we even outperform two larger models CoCa/UniTNT (139.1 vs. 105.6/109.4 in CIDEr) on TextCaps and (56.3% vs. 55.4%) on TextVQA.

PreSTU, however, performs worse than another larger model LaTr on TextVQA/ST-VQA. We attribute this to the superiority of LaTr's V&L backbones. As shown in Table 5, $\text{LaTr}_{\text{base}}$ with no pre-training significantly outperforms our baseline (NOPRESTU) on TextVQA (52.3% vs. 45.2%). LaTr and PRESTU use different scene-text pre-training data: LaTr uses five times larger data than PRESTU (64M vs. 13M in Table 4), which covers more *diverse* scene text. This is particularly beneficial to TextVQA/ST-VQA, which contain scene text from multiple domains (*e.g.*, brand, vehicle, etc.) and may explain why LaTr outperforms PRESTU.

In contrast, OCR-VQA [40] only covers book-related scene text. Thus, pre-training data becomes less important than pre-training approaches, and PRESTU outperforms LaTr (72.2% vs. 67.5% in Table 5). Moreover, while LaTr only shows its effectiveness on VQA tasks, PreSTU shows on both VQA and image captioning tasks.

We further compare PRESTU to extremely large-scale V&L models pre-trained on more than 2B $\langle image, text \rangle$ pairs. Interestingly, our best model even outperforms two much larger models Flamingo [1] and GIT2 [53] on some tasks; using much less data, we achieve better results than Flamingo (56.3% vs. 54.1%, Table 4) on TextVQA and than GIT2 (72.2% vs. 69.9%, Table 5) on OCR-VQA.

Recently, PaLI [9], a large-scale V&L model (ViT-e/mT5-XXL) pre-trained on 10B $\langle image, text \rangle$ pairs, reports SOTA results on all major V&L tasks, except for VizWiz-Captions (Table 4). It is worth noting that PRESTU (specifically, our OCR) was an ingredient in the pre-training objective of PaLI to tackle OCR and STU tasks, demonstrating OCR's utility in large-scale SOTA models.

The closest to PRESTU in terms of model/data sizes is GIT_{L}, a smaller-scale version of GIT2 (347M parameters and 20M $\langle image, text \rangle$ pairs). As shown in Table 5, PRESTU outperforms (or is on par with) GIT_{L} on all tasks, demonstrating efficiency with respect to model/data sizes. See more comparisons in the supplementary material.

3.2. Analysis

We aim to understand PRESTU in detail. We show (a) the importance of different components of our design choice, (b) its zero-shot transferability, (c) the effect of pre-training image resolution, (d) the effect of pre-training data size, and (e) the effect of downstream OCR quality.

Detailed ablation. As shown in Figure 2, our PRESTU consists of two (optional) pre-training stages, followed by fine-tuning on downstream tasks. Here, we aim to understand the gain brought by each component. We consider different combinations of the design choices at each stage and organize the results stage-by-stage into Table 6. We have the following three major observations.

First, SPLITOCR is significantly and consistently better than OCR (Rows with SPLITOCR vs. Rows with OCR in their Stage-1). OCR is a **"pure"** OCR prediction task, a variant of our main SPLITOCR (OCR-conditioned OCR prediction) in which the splitting point is always at the beginning. At first glance, such a result may seem counterintuitive: predicting the entire scene text is strictly harder than predicting part of the OCR text given the other part. When thought of carefully, this result indicates that OCR may put too much emphasis on *recognizing* scene text, at the expense of *connecting* scene text to its visual context. In other words, this

Model	Vision / Language	Model Size	Data Size	Pre-training Objective	Test Benchmark					
					TextCaps CIDEr	VizWiz-Captions CIDEr	ST-VQA ANLS	TextVQA Acc	VizWiz-VQA Acc	VQAv2 Acc
NoPreSTU	ViT-B/16 / $mT5_{base}$	473M	0	-	96.9	87.2	56.7	44.8	57.2	75.2
PreSTU	ViT-B/16 / $mT5_{base}$	473M	13M	SplitOCR	126.1	90.2	65.5	55.2	61.3	76.2
				SplitOCR→VQA/CAP	139.1	105.6	N/A	56.3	62.0	76.1
Non-TAP [59]	FRCNN / $BERT_{base}$	146M	0	-	93.4	-	51.7	44.8	-	-
TAP [59]			1.5M*	MLM+ITM+RPP	109.7	-	59.7	54.0	-	-
TAG [52]			88K*	MLM+ITM+RPP	-	-	60.2	53.7	-	-
LOGOS [34]			88K*	RoiLocal	-	-	57.9	51.1	-	-
ConCap [54]	BLIP	559M	129M	VLM+ITM+ITC	105.6	-	-	-	-	-
UniTNT [13]					109.4	-	66.0	55.4	-	80.1
LaTr [4]	ViT-B/16 / $T5_{large}$	831M	64M	MLM	-	-	69.6	61.6	-	-
Flamingo [1]	NFNet / Chinchilla	80B	2.3B	VLM	-	-	-	54.1	65.4	82.1
GIT2 [53]	DaViT / TransDec	5B	12.9B	VLM	145.0	**120.8**	75.8	67.3	70.1	81.9
PaLI [9]†	ViT-e / mT5-XXL	16B	10B	our OCR w/ others	**160.4**	-	**79.9**	**73.1**	**73.3**	**84.3**

Table 4: **Comparison to prior works.** See §3.1.4 for a detailed discussion. FRCNN: Faster R-CNN, TransDec: 6-layer transformer decoder, MLM: Masked Language (visual region) Modeling, ITM: Image-Text Matching, RPP: Relative Position Prediction, VLM: Visual Language Modeling, ITC: Image-Text Contrastive Loss, RoiLocal: ROI localization. *: dedicated scene-text understanding data, including ST-VQA, TextVQA, TextCaps, and OCR-CC. †: our objective OCR is an ingredient in their pre-training objectives.

Model	Vision / Language	Model Size	Data Size	Pre-training Objective	Val or test-dev Benchmark						
					TextCaps CIDEr	VizWiz-Captions CIDEr	ST-VQA ANLS	TextVQA Acc	VizWiz-VQA Acc	VQAv2 Acc	OCR-VQA Acc
NoPreSTU	ViT-B/16 / $mT5_{base}$	473M	0	-	100.0	87.7	55.6	45.2	57.7	74.8	71.5
PreSTU	ViT-B/16 / $mT5_{base}$	473M	13M	SplitOCR	134.6	90.3	**62.7**	55.6	61.9	76.0	**72.2**
				SplitOCR→VQA/CAP	**141.7**	**105.6**	N/A	**56.7**	**62.5**	**76.1**	-
$LaTr_{base}$ [4]	ViT-B/16 / $T5_{base}$	281M	0	-	-	-	-	52.3	-	-	-
$LaTr_{base}$ [4]	ViT-B/16 / $T5_{base}$	281M	64M	MLM	-	-	67.5	58.0	-	-	67.5
GIT_L [53]	ViT-L/14 / TransDec	347M	20M	VLM	106.3	96.1	44.6	37.5	**62.5**	75.5	62.4
GIT2 [53]	DaViT / TransDec	5B	12.9B	VLM	148.6	119.4	75.1	68.4	71.0	81.7	69.9

Table 5: **Comparison to GIT_L (similar model/data sizes to PreSTU).** PreSTU outperforms (or is on par with) GIT_L on all tasks. GIT2/$LaTr_{base}$-64M are for reference to show that PreSTU even outperforms these large-scale works on OCR-VQA.

highlights how SplitOCR is able to balance the two capabilities that we identify as important for STU (§1).

Second, SplitOCR (or OCR) makes the visual component (ViT) *inherently* better at recognizing text (gap between "Yes" and "No" Rows with Stage-1 pre-training vs. gap between "Yes" and "No" Rows without Stage-1 pre-training). Without Stage-1 (*e.g.*, VQA/CAP), removing OCR signals during fine-tuning leads to more than a 33% drop on TextVQA and a 49 CIDEr point drop on TextCaps. With Stage-1, these drops become less than 17% and 26 CIDEr points, respectively. For TextCaps, SplitOCR with "No" OCR input tokens during fine-tuning even outperforms the baseline *with* OCR input (116.6 vs. 100.0 in CIDEr). In summary, *recognizing* scene text via Stage-1 pre-training is important (*i.e.*, cannot be achieved via VQA or CAP alone).

Third, having two sources of OCR signals is beneficial. OCR signals by pre-trained ViT (Row SplitOCR→VQA/CAP with "No") and OCR signals by the off-the-shelf system (Row NoPreSTU "Yes") are complementary; we achieve the best result when leveraging both OCR signal sources (Row SplitOCR→VQA/CAP with "Yes"). See more ablation studies in the supplementary material.

Zero-shot transferability on scene-text VQA. Table 7 shows zero-shot transferability of SplitOCR on TextVQA. We observe that performing SplitOCR and then fine-tuning on ST-VQA (SplitOCR→VQA) already leads to a strong model; SplitOCR→VQA *without* fine-tuning (44.3%) is competitive to NoPreSTU *with* fine-tuning on TextVQA training set (45.2%), while ST-VQA alone (VQA) only achieves 35.7%. This suggests that SplitOCR enables generalization for STU and may remove the need to collect TextVQA data entirely!

Effect of image resolutions during pre-training. We hypothesize that pre-training with high-resolution images is important for scene-text recognition; Table 8 supports this argument. Further, pre-training with the 224x224 image resolution (standard resolution for many vision tasks) almost does not help; it achieves the accuracy of 47.1%, close to 45.2% of NoPreSTU baseline (Table 6 Row 2), suggesting non-standard resolution must be considered to reap the benefit of STU pre-training.

Pre-training Stage-1	Pre-training Stage-2	Fine-tuning OCR input	TextVQA Val Acc	TextCaps Val CIDEr
-	-	No	19.5	40.1
		Yes	45.2	100.0
-	VQA/CAP	No	13.7	81.1
		Yes	47.2	130.2
OCR	-	No	35.8	110.4
		Yes	49.9	126.7
OCR	VQA/CAP	No	38.6	108.9
		Yes	51.9	134.4
SPLITOCR	-	No	39.4	116.6
		Yes	55.6	134.6
SPLITOCR	VQA/CAP	No	44.3	118.4
		Yes	**56.7**	**141.7**

Table 6: **Main ablation studies** for validating the importance of our main components: SPLITOCR, VQA/CAP, and having OCR input during fine-tuning. See §3.2 for a detailed discussion. OCR refers to predicting the entire OCR text.

Model	Pre-training Objective	Fine-tuning	TextVQA Val Acc
NOPRESTU	-	-	0.04
		TextVQA	45.2
PRESTU	VQA	-	35.7
	SPLITOCR→VQA	-	44.3

Table 7: **Zero-shot transferability on TextVQA.** Our zero-shot SPLITOCR→VQA (*without* fine-tuning on TextVQA) is competitive to supervised NOPRESTU (*with* fine-tuning on TextVQA).

Model	Pre-training		Fine-tuning Resolution	TextVQA Val Acc
	Objective	Resolution		
PRESTU	SPLITOCR	224	640	47.1
		384		50.2
		480		53.1
		640		**55.6**

Table 8: **Effects of image resolutions.** TextVQA accuracy goes up as the pre-training image resolution increases, emphasizing the necessity of high-resolution images during pre-training.

Effect of pre-training data scale. How much data do we need to learn to recognize text? Table 9 shows the performance of TextVQA given checkpoints pre-trained on 1%, 3%, 10%, and 30% subsets of CC15M. We find that the TextVQA performance goes up as more pre-training data is included. This highlights the importance of data scale in acquiring *transferable* scene-text recognition skills.

Effect of downstream OCR systems. We study the effect of different OCR systems during fine-tuning (Table 10). We observe that the SPLITOCR-pre-trained model is more robust to the change in downstream OCR systems than

Model	Pre-training			TextVQA Val Acc
	Objective	Proportion	# of Data	
PRESTU	SPLITOCR	1%	130K	42.3
		3%	390K	45.4
		10%	1.3M	50.6
		30%	3.9M	53.0
		100%	13M	**55.6**

Table 9: **Importance of pre-training data scale.** TextVQA performance improves as more pre-training data, showing the importance of data scale in learning *transferable* scene-text recognition.

Model	Pre-training Objective	Fine-tuning OCR System	TextVQA Val Acc
NOPRESTU	-	TextOCR [45]	44.0
		Rosetta [7]	36.7
		gOCR	45.2
PRESTU	SPLITOCR	TextOCR [45]	54.8
		Rosetta [7]	50.7
		gOCR	**55.6**

Table 10: **Effect of downstream OCR systems on TextVQA.** SPLITOCR makes the model more robust to the change in OCR systems during fine-tuning.

NOPRESTU. Indeed, SPLITOCR + Rosetta can even perform better than NOPRESTU + gOCR. This result is consistent with Table 6, where we experiment with removing OCR texts entirely during fine-tuning. We also find that gOCR is the most effective. Interestingly, it is even better than human-annotated TextOCR; we hypothesize this is because TextOCR only provides word-level annotation whereas gOCR provides some grouping.

4. Related Work

Scene-Text Understanding. Most early STU works [20, 21, 29, 7, 18, 32] have merely focused on Optical Character Recognition (OCR). We instead focus on scene-text understanding (STU) in the context of V&L tasks: VQA [46, 5] and image captioning [45]. The most common approach for these STU tasks is to fuse pre-extracted object detection features with off-the-shelf OCR signals as additional input [46, 19, 45, 4, 17, 22, 52, 57, 35, 28]. These works often focus on specific challenges in downstream STU tasks, including dealing with noisy OCR signals, enabling the generation of rare words, or incorporating geometric information of OCR texts. In contrast, our work focuses on pre-training general-purpose STU models and shows the effectiveness of our objectives on multiple downstream STU tasks (§3.1).

V&L Pre-Training for STU. One line of works incorporates OCR signals explicitly for pre-training [59, 4, 34]. TAP proposes an objective to learn the relative spatial position of two OCR texts. LOGOS [34] localizes a region that

is most related to a given task and relies on its OCR text to complete the task. LaTr [4] models the co-occurrence statistics of layout-aware OCR tokens. Our pre-training objectives, on the other hand, focus on learning both scene-text recognition and the role of scene-text in its visual context.

The other line of works is OCR-free. Recently, extremely large image-text models have shown promising results on STU tasks, despite having no explicit STU objectives (*e.g.*, GIT2 [53], Flamingo [1]). However, it would require an analysis of their private data and a prohibitive amount of resources to pinpoint what contributes to such strong results. Our study offers a complementary perspective to this OCR-free approach by pushing the limit of the OCR-heavy approach further than before and conducting more thorough experiments at a smaller scale.

5. Conclusion

We introduce a simple recipe for scene-text understanding, consisting of OCR-aware pre-training objectives operating from image pixels. Our task-agnostic objective SPLITOCR teaches the model to recognize scene text and to connect scene text to its visual context. Our task-specific objectives VQA and CAP further strengthen that connection. We conduct comprehensive experiments to demonstrate the utility of this recipe.

Acknowledgments. We would like to thank Bo Pang, Xiao Wang, Kenton Lee, and Tania Bedrax-Weiss for their thoughtful feedback and discussions. J. Kil and W. Chao are supported in part by grants from the National Science Foundation (IIS-2107077, OAC-2118240, and OAC-2112606) and Cisco Systems, Inc.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 6, 7, 9

[2] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny. Word spotting and recognition with embedded attributes. *In TPAMI*, 2014. 2

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. 5

[4] Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R Manmatha. Latr: Layout-aware transformer for scene-text vqa. In *CVPR*, 2022. 2, 4, 6, 7, 8, 9

[5] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, C.V. Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *ICCV*, 2019. 1, 2, 4, 8

[6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *In TACL*, 2017. 2

[7] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018. 1, 8

[8] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021. 2, 4

[9] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Alexander Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A jointly-scaled multilingual language-image model. In *ICLR*, 2023. 6, 7

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4

[11] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, 2014. 5

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 3

[13] Roy Ganz, Oren Nuriel, Aviad Aberdam, Yair Kittenplon, Shai Mazor, and Ron Litman. Towards models that can see and read. *arXiv preprint arXiv:2301.07389*, 2023. 6, 7

[14] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 2

[15] Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. VizWiz Grand Challenge: Answering visual questions from blind people. In *CVPR*, 2018. 2, 4

[16] Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning images taken by people who are blind. In *ECCV*, 2020. 1, 2

[17] Wei Han, Hantao Huang, and Tao Han. Finding the evidence: Localization-aware answer prediction for text visual question answering. *arXiv preprint arXiv:2010.02582*, 2020. 8

[18] Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, and Changming Sun. An end-to-end textspotter with explicit alignment and attention. In *CVPR*, 2018. 8

[19] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *CVPR*, 2020. 1, 2, 8

[20] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014. 8

[21] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *In IJCV*, 2016. 8

[22] Yash Kant, Dhruv Batra, Peter Anderson, Alexander Schwing, Devi Parikh, Jiasen Lu, and Harsh Agrawal. Spatially aware multimodal transformers for TextVQA. In *ECCV*, 2020. 8

[23] Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher D. Manning. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. In *ACL*, 2021. 5

[24] 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 *ICDAR*, 2015. 4

[25] 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 *ICDAR*, 2013. 4

[26] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016. 2

[27] 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. *In IJCV*, 2017. 4

[28] Bingjia Li, Jie Wang, Minyi Zhao, and Shuigeng Zhou. Two-stage multimodality fusion for high-performance text-based visual question answering. In *ACCV*, 2022. 8

[29] Hui Li, Peng Wang, and Chunhua Shen. Towards end-to-end text spotting with convolutional recurrent neural networks. In *ICCV*, 2017. 8

[30] Yang Li, Gang Li, Luheng He, Jingjie Zheng, Hong Li, and Zhiwei Guan. Widget captioning: generating natural language description for mobile user interface elements. *arXiv preprint arXiv:2010.04295*, 2020. 1, 2

[31] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 2004. 5

[32] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *CVPR*, 2018. 8

[33] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 1

[34] Xiaopeng Lu, Zhen Fan, Yansen Wang, Jean Oh, and Carolyn P Rosé. Localize, group, and select: Boosting text-vqa by scene text modeling. In *ICCV*, 2021. 2, 6, 7, 8

[35] Siwen Luo, Feiqi Cao, Felipe Nunez, Zean Wen, Josiah Poon, and Caren Han. Scenegate: Scene-graph based co-attention networks for text visual question answering. *arXiv preprint arXiv:2212.08283*, 2022. 8

[36] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 1, 2

[37] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *WACV*, 2022. 1

[38] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *WACV*, 2021. 1, 2

[39] Anand Mishra, Karteek Alahari, and C.V. Jawahar. Image retrieval using textual cues. In *ICCV*, 2013. 4

[40] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, 2019. 1, 2, 6

[41] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 5

[42] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *In JMLR*, 2020. 2

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2, 4

[44] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 2, 4

[45] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. TextCaps: a dataset for image captioning with reading comprehension. In *ECCV*, 2020. 1, 2, 8

[46] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *CVPR*, 2019. 1, 2, 5, 8

[47] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *EMNLP*, 2020. 5

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[49] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *CVPR*, 2015. 5

[50] 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. 4

[51] Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. Screen2words: Automatic mobile ui summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, 2021. 2

[52] Jun Wang, Mingfei Gao, Yuqian Hu, Ramprasaath R Selvaraju, Chetan Ramaiah, Ran Xu, Joseph F JaJa, and Larry S Davis. Tag: Boosting text-vqa via text-aware visual question-answer generation. *arXiv preprint arXiv:2208.01813*, 2022. 2, 6, 7, 8

[53] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. GIT: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022. 5, 6, 7, 9

[54] Ning Wang, Jiahao Xie, Jihao Wu, Mingbo Jia, and Linlin Li. Controllable image captioning via prompting. *arXiv preprint arXiv:2212.01803*, 2022. 6, 7

[55] 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 *CVPR*, 2020. 1

[56] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple visual language model pretraining with weak supervision. In *ICLR*, 2022. 3

[57] Dongsheng Xu, Qingbao Huang, and Yi Cai. Device: Depth and visual concepts aware transformer for textcaps. *arXiv preprint arXiv:2302.01540*, 2023. 8

[58] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *NAACL*, 2021. 2, 3

[59] Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. TAP: Text-aware pre-training for text-vqa and text-caption. In *CVPR*, 2021. 1, 2, 4, 5, 6, 7, 8

[60] Xiaoyu Zeng, Yanan Wang, Tai-Yin Chiu, Nilavra Bhattacharya, and Danna Gurari. Vision skills needed to answer visual questions. *Proceedings of the ACM on Human-Computer Interaction*, 2020. 5