Harnessing the Power of Multi-Lingual Datasets for Pre-training: Towards Enhancing Text Spotting Performance

Alloy Das [1] Sanket Biswas[2] Ayan Banerjee[2] Josep Lladós[2] Umapada Pal[1] Saumik Bhattacharya[3]

[1]CVPR Unit, Indian Statistical Institute, Kolkata [2]CVC, Universitat Autònoma de Barcelona [3]ECE, Indian Institute of Technology, Kharagpur

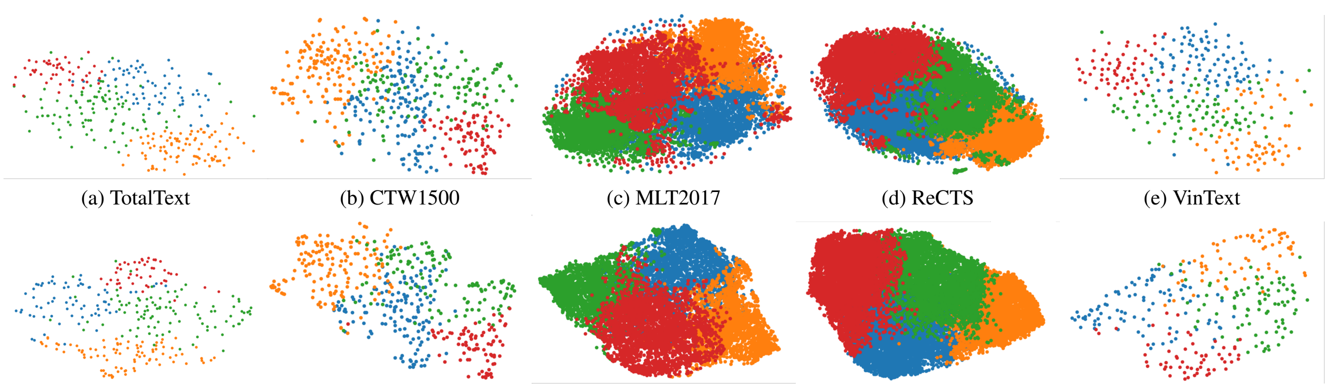
Abstract

The adaptation capability to a wide range of domains is crucial for scene text spotting models when deployed to real-world conditions. However, existing SOTA approaches usually incorporate scene text detection and recognition simply by pretraining on natural scene text datasets, which do not directly exploit the intermediate feature representations between multiple domains. Here, we investigate the problem of domain-adaptive scene text spotting, i.e., training a model on multi-domain source data such that it can directly adapt to target domains rather than being specialized for a specific domain or scenario. Further, we investigate a transformer baseline called ***Swin-TESTR*** *to focus on solving scene-text spotting for both regular and arbitrary-shaped text along with an exhaustive evaluation. The results demonstrate the potential of intermediate representations to gain significant performance on text spotting benchmarks across multiple domains (e.g. language, synth-to-real, and documents). both in terms of accuracy and efficiency.*

1. Introduction

End-to-end scene text spotting has been an active research problem in the computer vision community owing to its immense utility in real-world application scenarios like autonomous driving [14, 52], intelligent navigation systems [44], image retrieval [21] and so on. It can be defined as the joint optimization of scene-text detection and recognition tasks in a unified model pipeline. Research efforts in recent years on text spotting [13, 19, 28, 35, 46, 54] have demonstrated the superiority of deep CNNs and transformer-based approaches on natural scene image benchmarks containing regular text [22], and arbitrarily-shaped text [8, 27]. However, it is essential for a reading system (OCR) to learn and adapt to new complexities in unseen tasks (domains) without forgetting to read previous tasks (domains) [48], hence overcoming the phenomenon of *catastrophic forgetting* for neural networks [15].

In recent years, multimodal foundation models pre-trained on multiple source data, have demonstrated impressive performance on several tasks, including text recognition [1, 6, 49], entity extraction [20, 43], and document layout analysis [5, 20, 23, 32]. Given the growing practi-



(a) TotalText (b) CTW1500 (c) MLT2017 (d) ReCTS (e) VinText

(f) TotalText (g) CTW1500 (h) MLT2017 (i) ReCTS (j) VinText

Figure 1. **Text Spotting Benchmark Diversity.** We employ a t-SNE (top row) and UMAP (bottom row) to visualize the different clusters of similar images corresponding to text orientation characteristics where Vertical, Circular, Wavy, and Horizontal Curve.

cal significance of pre-training on diverse domains containing both real [30] and synthetic data [26], there is a surging interest in analyzing its empirical performance. Recent explorations undertaken in [3] and [4] have delved into different aspects of model design choices and the impact of training and evaluation settings for benchmarking Scene Text Recognition (STR). However, there are open research questions that need to be addressed for a deeper understanding of the performance gain in the SOTA text spotting models. **Firstly: Given the exuberant amount of data available for pre-training having different sets of text instance properties: which of them are most empirically useful?** This is really difficult to assess given the current training and evaluation scheme followed by every recent SOTA approach. **Secondly: Why is the multilingual MLT17 [33] most extensively used dataset for pretraining?** As MLT contains samples in multiple languages (6 in total) and scripts (8 in total) in a wide variety of real-world challenging conditions with diverse text instance properties (see Fig. 1), there are hopes that the intermediate pre-trained representation learned for this task will generalize well to other downstream tasks such as text detection and end-to-end spotting for both arbitrary-shaped text and regular text benchmarks. However, the existing SOTA has not made sufficient attempts towards empirically analyzing this hypothesis. **Thirdly: Why do we need the domain adaptation settings for scene text spotting?** Since the SOTA text spotting models mostly focus on domain-specific fine-tuning for getting the best results for the target evaluation benchmarks, domain adaptation settings from synthetic (source) to real (target) datasets [11,47] have yet to be investigated more in-depth. The existing domain adaptation approaches utilize adversarial training and discriminative feature alignment to learn some domain-invariant feature representations for text. But these methods are more complex as they are generative and our key objective is to pay attention to the model capacity and computation cost. **Fourthly: Are the model representations robust enough to be used as an OCR for document layout Analysis (DLA)?** When it comes to measuring model robustness and generalizability, the effectiveness of the text spotting model to detect and read from text regions could be indeed beneficial for DLA tasks without the need for a commercial OCR engine.

Given the bottleneck that annotated real data is expensive and fewer in proportion to the available synthetic data [16], it further motivates us to study the domain adaptation settings for text spotting task. Although purely relying on synthetic data doesn't help to get the best model performance, they provide a better initialization point with learnt structural and contextual cues to improve the overall performance with further addition of real data training with far lesser number of iterations. Additionally, as shown in [41] a model should inherently learn different degraded scene complexities (like blurs, distortions, occluded objects, ...) to obtain some robustness in it's pre-trained feature representations. In this work, we introduce a simple text spotting baseline called *Swin-TESTR* which performs visual feature extraction using a Swin-Transformer [29] backbone, coupled with a Transformer-based unified single encoder-dual decoder framework as proposed in TESTR [54]. On account of it's higher tranferability [55] compared to simple ViTs and CNNs and more interpretable hierarchical feature maps obtained during the feature extraction phase, Swin-TESTR helps to better analyse and establish strong benchmark performances for both domain adaptation and SOTA end-to-end spotting settings.

The overall contributions of this work can be summarized in four folds:1) To boost the text spotting performance by simply utilizing intermediate representations learned from more diverse and complex datasets (like the multilingual MLT17 [33], Chinese ReCTS [53] or English Total-Text [8]), we *analyze the trade-offs for different combinations of tasks (domains) in a domain adaptation setting*. 2) With extensive experimentation, we analyze the *significance of multilingual MLT17 as a strong intermediate representation* after pre-training on synthetic data helps to obtain a fully competitive model for tackling the data distribution shift [42]. 3) A flexible Swin-TESTR baseline has been proposed to show an exhaustive experimental study to address the large domain gap between synthetic and real scene datasets and *explain how diverse intermediate representations could play a role to minimize this gap*. 4) Moreover, we make a *first attempt towards investigating the generalization capability of the Swin-TESTR baseline by evaluating its OCR performance on the DLA task*. Our baseline shows promising results as it outperforms the commercial Microsoft-OCR engine for "text" and "math" regions.

2. Challenges and Limitations in SOTA

In this section, we examine the different benchmarks that have been adapted for training and evaluation by prior works and address their pitfalls. Through this investigation, we explore the performance inconsistencies across different benchmarks and find shreds of evidence through experimentation and intuitive discussions.

2.1. Pitfalls in Pretraining Strategies

When pre-training text spotting models, the most common practice is to pre-train with the Curved SynthText data containing 150K samples first introduced in [26]for detecting arbitrary-shaped scene text. The dataset was curated from the 800K SynText dataset [16] containing quadrilateral bounding boxes. Such synthetic dataset curations have been also used in scene text editing task [10, 39]. Since then, the most influential SOTA text spotting literature [13, 19, 26, 28, 35, 54] has persisted with large-scale pretraining with the Curved SynthText. MLT2017 [33] and TotalText [8] are the most used real-world datasets that have been used in pre-training the SOTA models. Although the richness and sample diversity of these datasets has been considered to be the key reasons, there have not been enough justifications to prove why they have been chosen. In this work, we try to explore and understand why and how they prove to be beneficial during the model pre-training. Note that the essential pre-training protocols used by the recent methods are mainly of two different kinds. While methods like ABINet++ [13], ABCNetv2 [28] and TESTR [54] mix both synthetic and real data together for pre-training, others like SwinTextSpotter [19] and MANGO [35] prefer to generate a synthetic data initialized model checkpoint and then pre-train all the real-world data together on top of it for the rest of the iterations. The problem of how to use real datasets for STR tasks has been previously addressed in [3]. Also, the same authors addressed the pitfalls of STR datasets and model evaluation in [2]. Contrary to the aforementioned works, we delve deep into understanding (see sec. 4.2) the current text spotting practices and finding the balance between the number of datasets used and the model complexities for the overall performance gain. In this regard, a domain adaptation setting for evaluation has been introduced to actually validate the aforementioned open questions.

2.2. Fairness in Evaluating Real-world datasets

Text spotting has been commonly fragmented into two different sub-tasks in the literature: text detection and recognition. Past approaches [18, 24, 25, 45] were mainly evaluated on regular-shaped text spotting benchmarks like ICDAR15 [22]. Later, arbitrary-shaped text benchmarks like Total-Text [8] and CTW1500 [27] became the standard benchmarks for evaluation protocols. Recent SOTA text spotting methods ranging from CNN-based [26, 28, 35–37, 46] to recent transformer-based [13, 19, 50, 54] techniques have mainly done their evaluations with the aforementioned arbitrary-shaped English text benchmarks. Recently, Vintext [34] and ReCTS [53] have emerged as more challenging non-English arbitrary-shaped benchmarks for further tough evaluation. One major issue plaguing all the SOTA methods is they rely heavily upon fine-tuning their models on the corresponding evaluation benchmark to achieve superior performance gain. In this work, we investigate a domain adaptation setting to evaluate the proposed baseline on several combinations of intermediate pretraining and evaluation steps to gain some fairness in the model selection. A few prior works [9, 47] have studied such domain distribution mismatches for scene text recognition. Also, we have added model generalization capability when transferred to another task from a different domain as a further evaluation criterion. The closest work related to this task is [51] and [1] which harnesses the power of CLIP [38] model for scene text detection and recognition task respectively.

3. Preliminaries

The primary goal of our proposed baseline is to learn some domain-agnostic feature representation through an improved intermediate representation of combining real data with synthetic data in an effective way and address the domain shift problem. Additionally, we also introduce the technical details of our proposed text-spotting architecture with an efficient self-attention unit.

3.1. Problem Formulation

In order to formulate the problem statement, let us assume, we have an input space χ and output space ξ. We formulate domain adaptation as a learning problem that is specified by two parameters: a distribution D over χ (i.e. the domain) and a relation function $R : \chi \rightarrow \xi$ which maps instances to features. Basically, it is a representation that induces the distribution D over ξ to perform the same task in multiple domains. Now what are our source (input space χ) and target domains (output space ξ)? We define three pairs of source-target domains towards text spotting.

1. Language to Language: We take the source domain as the English language and perform training over it and adapt it to the Vietnamese and Chinese text domains.

2. Synth to Real: We initialize training on a synthetic dataset and adapt it to the real text domain. Here we observed that, a potential complex intermediate real dataset (e.g. MLT 2017) boosts up the domain adaptation process.

3. Scene Text to Document: Several previous works [20, 40] used OCR tokens in order to boost the performance of Document Layout Analysis. We replace those OCR tokens with the extracted information by the text spotting framework and evaluate it using the same baseline.

In order to formulate the problem of domain adaptation, Let D_s denote the feature distribution of the source domain, Y_s denote the labels of the source domain distribution over the input space χ and D_t denote the feature distribution of the target domain, Y_t denote the labels of the target domain distribution over ξ. The goal of domain adaptation is to learn a labeling rule $f : (D_s, Y_s) \rightarrow (D_t, Y_t)$ that can generalize well on the target domain using labeled data from the source domain and potentially some labeled data from the target domain. We can formulate it as finding a mapping function f that minimizes the distribution mismatch between the source and target domains while ensuring good generalization on the target domain as defined in eq. 1.

$$\min_f [\mathcal{L}_s(f) + \lambda \cdot \mathcal{L}_t(f)] \tag{1}$$

where, $\mathcal{L}_s(f)$ is the loss on the source domain data, encouraging the model to fit the source domain, $\mathcal{L}_t(f)$ is a dis-

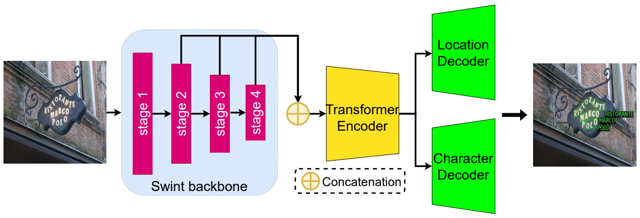


Figure 2. Schematic overview of the model Swin-TESTR.

tribution discrepancy or adaptation loss between the source and target domains, capturing the differences between the two domains, and λ is a balancing parameter that controls the trade-off between fitting the source domain and adapting to the target domain. The goal of solving this eq. 1 is to learn a model f that can generalize effectively on the target domain even when the distribution of the source and target domains is different. The focus is on mitigating the negative effects of domain shift and improving the transferability of the learned knowledge from the source to the target domain.

3.2. Model Architecture

The overall architectural pipeline as illustrated in Fig.2 consists of three major components: (1) a visual feature extraction unit based on a Swin-Transformer [29] backbone to extract multi-scale features; (2) a text spotting unit consisting of a Transformer encoder to encode the image features into positional object queries and then two Transformer decoder units to predict the location of text instances and recognize the corresponding characters respectively.

Visual Feature Extraction Unit. It is hard to connect remote features with vanilla convolutions since they operate locally at fixed sizes (e.g., 3×3). Text spotting requires capturing the relationships between different texts since scene text from the same image has a similar representation with respect to the text background, style and texture. For our backbone, we chose to use a small and efficient Swin-Transformer [29] unit denoted as Swin-tiny for extraction of more fine-grained image features.

Text Spotting Unit. The text-spotting unit is mainly composed of a transformer encoder and two transformer decoders for text detection and recognition following a similar schema as proposed in the TESTR framework [54]. Accordingly, we formulate our problem as a set prediction problem as in DETR [7], to predict a set consisting of point-character tuples, for a particular image. We formulate it as $X = \{(S^{(i)}, R^{(i)})\}_{i=1}^{K}$. Where i is the index of each instances, $S^{(i)} = (s_1^{(i)}, ..., s_M^{(i)})$ is the coordinates of M control points, and $R^{(i)} = (r_1^{(i)}, ..., r_M^{(i)})$ is the M characters of the text. The text location decoder (TLD) will detect (predict $S^{(i)}$) while the text recognition decoder (TRD) will recognize (predict $R^{(i)}$) the text in a unified manner.

4. Experiments and Analysis

For the purpose of validation, we have considered some important benchmark dataset e.g. the ICDAR 2015 [22], Total-Text [8], CTW1500 [27], VinText [34], Curved SynthText 150K [26], ICDAR 2017 MLT [33], and ReCTS [53] with different text orientation categorizations. Our experimental evaluation suggests that the domain adaptation through intermediate representation advances the state-of-the-art. Moreover, extensive ablation studies have been performed on Total-Text [8] to show the contribution of some important elements of Swin-TESTR. For more dataset details please refer to the supplementary materials. The code is be publicly available on github.

4.1. Implementation Details

The hyper-parameters for the deformable transformer [56] have been kept similar to the original work, with the number of heads = 8 and using 4 sampling points for deformable attention. The number of encoder and decoder layers is initialized to 6.

Data Augmentation. During pre-training, we augment the data with a random resize with the shorter edge (the length of the shorter edge is from 480 to 896, and the longest edge is kept within 1600 (i.e. the length of the shorter edge (either width or height) of the input images is randomly resized to a value between 480 pixels and 896 pixels, and while resizing the shorter edge, it's ensured that the length of the longest edge (either width or height) does not exceed 1600 pixels.). This constraint is applied to prevent excessively large images, which can be computationally expensive to process. An instance-aware random crop (cropping operation takes into consideration the objects or instances within the image) has also been used to make the model more robust to text variations (e.g. curved, wavy, and so on) and positions within the images.

Training Details. To compare with the state-of-the-art methods, shown in Table 3 we pre-trained Swin-TESTR with synthtext150k [26], MLT 2017 [33] for 4000K iteration with a learning rate of 2×10^{-5} and is decayed at 3000-Kth iteration by a factor of 0.1. We use AdamW [31] optimizer, with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a weight decay of 10^{-5}. We use $Q = 100$ composite queries. The number of control points is 20 and the maximum text length is set to 25.

4.2. Domain Adaptation Experiments

Through our Domain Adaptation experiments, we have demonstrated the Swin-TESTR framework's ability to seamlessly transition between synthetic and real domains, as well as adapt within real domains, all without the need for re-training or fine-tuning on specific datasets. While it's well-known that fine-tuning typically results in supe-

Table 1. **Domain Adaptation on Total-Text and CTW-1500.** Method implies pre-trained on the first dataset and then trained on the next dataset, Mix pre-train means pre-trained on SynthText, ICDAR MLT17, ICDAR15, and Total-Text. Results style: **best**, second best.

Method	Total-Text					CTW-1500				
	Detection			End-to-End		Detection			End-to-End	
	P	R	F	None	Full	P	R	F	None	Full
Synth-Text	76.79	28.09	41.13	27.54	38.54	50.37	15.25	23.42	12.45	19.25
Synth-Text → ICDAR MLT17	90.02	59.17	82.74	62.97	77.98	38.48	49.00	43.10	28.24	33.84
Synth-Text → ICDAR15	90.24	60.16	72.20	55.76	69.74	36.20	28.53	31.91	19.64	24.3
Synth-Text → ICDAR MLT17 → ICDAR15	90.52	67.71	77.47	58.14	74.48	39.94	31.10	34.97	23.47	27.23
Synth-Text → ICDAR MLT17 → Total-Text	92.01	**85.82**	**88.81**	71.33	83.17	39.76	56.85	46.7	28.23	35.46
Synth-Text → ICDAR MLT17 → CTW1500	66.23	36.77	47.28	24.11	41.54	92.61	82.55	87.29	54.88	81.94
Synth-Text → Total-Text	92.19	78.95	85.06	70.51	81.06	43.86	51.04	47.18	29.33	35.63
Synth-Text → CTW1500	62.14	33.29	43.35	21.03	37.3	**94.11**	77.27	84.86	47.56	80.2
Mix pre-train	**93.59**	75.2	83.4	67.06	80.51	39.19	36.35	37.71	25.83	29.57
ICDAR MLT17	79.35	55.37	65.23	15.44	36.02	33.46	32.28	32.81	0.077	20.03
ICDAR MLT17 → Synth-Text	77.98	26.56	39.62	21.09	34.45	59.27	26.41	36.54	17.81	30.37
Synth-Text → ReCTS	87.08	36.22	51.16	33.17	0.02	41.89	23.36	30.00	19.11	0.04
Mix pre-train → Fine-tune	90.58	85.46	87.95	**75.14**	**86.0**	91.45	**85.16**	**88.19**	**56.02**	**82.91**

Table 2. **Domain Adaptation Experiments on Total-Text and CTW-1500.** Method implies pre-trained on the first dataset and then trained on the next dataset, Mix pre-train means pre-trained on SynthText, ICDAR MLT17, ICDAR15, and Total-Text.

Method	ICDAR 2015							ReCTS			
	Detection			End-to-End				Detection			End-to-End
	P	R	F	S	W	G	None	P	R	F	1-NED
Synth-Text	76.81	26.62	39.54	37.19	32.78	26.6	20.02	61.43	12.85	21.25	05.98
Synth-Text → ICDAR MLT17	89.49	84.02	86.67	82.95	75.15	67.69	57.36	78.62	**73.78**	76.12	14.77
Synth-Text → ICDAR15	89.83	74.82	81.64	77.07	70.91	62.51	52.82	47.35	21.51	29.58	10.24
Synth-Text → ICDAR MLT17 → ICDAR15	93.42	79.25	85.75	81.48	75.88	68.14	57.17	45.34	32.59	37.64	11.01
Synth-Text → ICDAR MLT17 → Total-Text	84.27	77.61	80.8	54.99	72.01	64.81	54.09	60.96	29.17	39.46	14.56
Synth-Text → ICDAR MLT17 → CTW1500	69.84	41.69	52.22	24.78	42.95	35.69	24.78	52.16	30.91	38.81	12.94
Synth-Text → Total-Text	77.27	77.56	77.41	46.9	66.59	58.41	46.95	45.38	32.65	37.98	11.43
Synth-Text → CTW1500	57.93	39.58	47.03	19.22	37.09	30.61	19.22	41.87	23.34	29.98	09.13
Mix pre-train	**95.25**	76.26	84.71	81.39	75.36	68.04	57.29	73.32	64.46	68.61	13.30
ICDAR MLT17	75.27	70.49	72.8	46.16	33.93	21.68	10.94	65.85	59.49	62.51	06.62
ICDAR MLT17 → Synth-Text	79.68	33.41	47.08	43.42	37.23	30.55	21.76	60.92	19.65	29.72	07.22
Synth-Text → ReCTS	84.48	44.29	58.12	1.54	0.05	0.05	18.19	**82.95**	72.67	**77.47**	**48.75**
Mix pre-train → Fine-tune	95.03	**85.70**	**90.13**	**86.63**	**81.67**	**75.44**	**66.46**	79.24	56.39	66.19	38.12

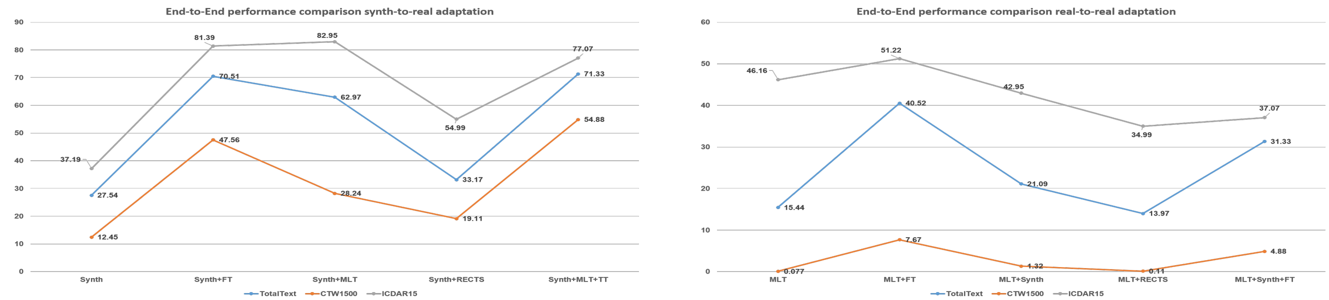
rior performance compared to generalization, our findings indicate that the performance gap is remarkably narrow.

In Table 1 and Table 2, we present our results using SynthText as the base, with re-training involving ICDAR 2017 MLT on the top of SynthText pre-trained weights. Traditionally, it has been common practice to pre-train models using a combination of synthetic and real data. However, in Table 1, we delve into the impact of pre-training with individual datasets, shedding light on how this approach affects our model's performance.

Some valuable insights gleaned from our experimental analysis include:

MLT dataset gives the maximum performance boost during pretraining. The existing SOTA approaches do not highlight the impact of individual datasets during pre-training. MLT dataset has quite a lot of variability among the real scene-text benchmarks as demonstrated in Fig. 1 and could be highly beneficial when it comes to domain adaptation performance as exhibited in both Table 1 and Table 2. This approach also helps us to *outperform SwinTextSpotter [19] on TotalText (with fine-tuning) with 88.8% detection performance* as reported. Not only that, the same pre-training strategy with MLT helps us to get state-of-the-art results on the ICDAR15 benchmark with a detection performance of 90.13% as shown in Table 3.

Domain Adaptation could be a great alternative to fine-tuning. Without fine-tuning we achieve really competitive results on both the word-level benchmarks, arbitrary-



(a) Synth to Real

(b) Real-to-Real

Figure 3. **Different Adaptation Settings:** Illustration of our baseline when combined with intermediate representations. Zoom in for better visualization.

Table 3. Scene text spotting results on Total-Text and CTW1500 and ICDAR 2015. "None" refers to recognition without any lexicon. The "Full" lexicon contains all the words in the test set. "S", "W", and "G" represents recognition with "Strong", "Weak", "Generic", lexica, respectively. For the models, which do not perform detection we denote it by N/A.

Methods	Total-Text			CTW1500			ICDAR 15			
	Detection	End-to-end		Detection	End-to-end		Detection	End-to-end		
	H-mean	None	Full	H-mean	None	Full	H-mean	S	W	G
Text Perceptron [36]	85.2	69.7	78.3	84.6	57.0	N/A	87.5	83.4	79.9	68.0
ABCNet v2 [28]	87.0	70.4	78.1	84.7	57.5	77.2	88.1	82.7	78.5	73.0
MANGO [35]	N/A	72.9	83.6	N/A	58.9	78.7	N/A	81.8	78.9	67.3
TESTR [54]	86.90	73.25	83.9	86.3	53.3	79.9	90.0	85.2	79.4	73.6
Swintextspotter [19]	88.0	74.3	84.1	88.0	51.8	77.0	N/A	83.9	77.3	70.5
Abinet++ [13]	N/A	**79.4**	85.4	N/A	**61.5**	81.2	N/A	86.1	**81.9**	**77.8**
Swin-TESTR	87.95	75.14	**86.0**	**88.19**	56.02	**82.91**	**90.13**	**86.63**	81.67	75.44

Table 4. Performance evaluation Vintext datasets.

Methods	H-mean
ABCNet [26]	54.2
ABCNet + D [34]	57.4
Mask Textspotter v3 [34]	53.4
Mask Textspotter v3 + D [34]	68.5
Swintextspotter [19]	71.1
Swin-TESTR(w/o fine-tune)	21.54
Swin-TESTR	**73.20**

shaped TotalText, and regular-shaped ICDAR 15. The difference between the performances for detection and end-to-end recognition (with and without fine-tuning) in the above benchmarks is quite marginal. This shows we could indeed have an efficient and robust model which do not require any fine-tuning on the corresponding evaluation dataset. Also, Fig. 3 demonstrates how domain adaptation settings gain a performance boost under synthetic-to-real when compared to real-to-real scenario.

More variable text orientations make the dataset a better pre-training candidate. As inferred from Fig. 1, the more the separability in terms of text orientations, corresponds to learning better representations during pretraining like as in Total-Text. However, even more dense and complex candidates like MLT or ReCTS as shown can help the model to learn more richer representations in terms of text orientation. As the representations of the MLT17 and ReCTS both are quite dense and diverse in Fig. 1, we need to analyze further to understand which one provides better intermediate representations. In order to do that, we observe the perceptual and structural similarity scores with the FID, FSIM, and SSIM metrics(See Table 6). Results demonstrate that MLT17 has the closest perceptual similarity with Total-Text, which is why they are the most preferred choices for SOTA pretraining. Structurally with SSIM, Total-Text is mostly similar to CTW1500, which helps immensely to improve the performance for CTW1500 evaluation.

Can Swin-TESTR representations help to read and benefit in document layout analysis task? As shown in Table 5, we compare the reading ability of our Swin-TESTR model for document layout analysis as we evaluate the performance of detecting several layout regions in documents

Table 5. Domain Adaptation performance from scene text to document

	Text	Image	Table	Math	Separator	other	AP	AP@0.5	AP@0.75
Layout Parser [40]	83.1	73.6	95.4	75.6	20.6	39.7	64.7	77.6	71.6
Layout Parser (our OCR)	**85.2**	64.7	90.2	**77.1**	11.2	28.1	59.8	68.1	61.7
LayoutLMv3 [20]	70.8	50.1	42.5	46.5	9.6	17.4	40.3	49.4	42.7
LayoutLMv3 (our OCR)	**72.1**	47.8	**43.5**	**47.2**	1.8	20.2	38.7	45.2	40.8

Table 6. The dataset diversion. Results style: **best**, second best

Source Dataset	Target Dataset	FID ↓	FSIM ↑	SSIM ↑
MLT2017	Total-Text	**0.82**	**25.23**	21.22
	CTW1500	6.97	18.12	16.98
	ICDAR15	14.78	14.20	13.25
	ReCTS	11.23	6.24	5.17
	VinText	2.22	21.48	19.07
Total-Text	CTW1500	1.29	24.27	**23.32**
	ICDAR15	9.74	16.67	15.12
	ReCTS	12.14	4.37	3.54
	VinText	2.43	19.32	19.14
CTW1500	ICDAR15	0.97	23.12	22.87
	ReCTS	16.32	2.22	1.46
	VinText	0.92	20.27	20.12
ReCTS	ICDAR15	20.20	1.05	1.03
	VinText	12.32	6.98	4.42

by utilizing the OCR from our model. The performance demonstrates that it is quite comparable to the original document layout analysis [20, 40] OCR and it outperforms the original model for text, math and table regions. This further justifies the information extraction capability of Swin-TESTR, although it gives a considerably poor performance on the separator, other and image regions due to its initialization with the text spotting weights which has never seen those kinds of instances.

4.3. Comparison with State-of-the-art

Table 3 shows the results summary of Swin-TESTR when compared to other related text spotting approaches. We outperform the existing state-of-the-art Abinet++ [13] on both word-level Total-Text [8] and sentence-level CTW1500 [27] benchmarks in end-to-end recognition task. On the other hand for scene text detection, we outperform SwinTextSpotter [19] on CTW1500 while having the second-best results for Total-Text in terms of F-score. It is worth mentioning that our model performs the best in the recall metrics for both benchmarks. Qualitative example studies have been further shown in Fig. 4.

Regular-shaped Scene Text Spotting. We evaluate our method on ICDAR15 benchmark [22] and display our results compared with the state-of-the-art in Table 3. For Text detection, we achieve a 5% gain in precision over TESTR [54] while beating the state-of-the-art F-measure marginally. For the text spotting task, our approach gives the *best result in the most challenging "Strong" type* as in this setting every image contains a lexicon of only 100 words. We outperformed TESTR, Swintspotter, and Abinet++ by almost 1.5%, 2.5%, and 0.5% respectively. In Fig 4 we show how our method performs on this dataset in the third column.

Low-Resource Text Spotting. We have also evaluated Vintext [34], and ReCTS [53] a low-resource scene text detection benchmark for Vietnamese and Chinese text to show the high generalizability of our model. We can see the results in Table 4 showing that our methods outperform the state-of-the-art by almost 2%. In Fig 4 we show the performance of our model in Vintext and ReCTS in the last column (**NOTE:** due to the space limitations we shift the Table of the ReCTS in the supplementary materials (Table 3). It has been observed that Swin-TESTR outperforms the SOTA approaches with ChineseSynthTexT pre-training).

We further throw light on the following insights after some intuitive analysis:

Why ABINet++ has better performance in End-to-End None performance? Since ABINet++ [13] uses linguistic information for the text spotting framework, they highly improve in this metric. A similar method MANGO [35] also follows this trend. Our approach outperforms both these approaches with a purely visual approach.

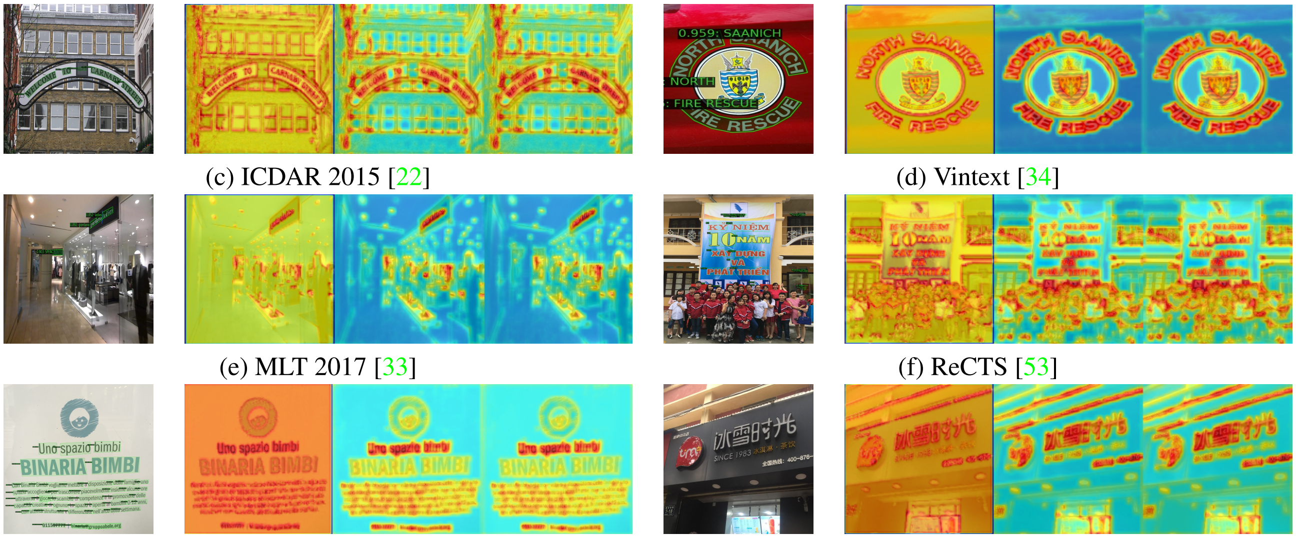
4.4. Ablation Studies

To understand the significance of the different components of the Swin-TESTR framework, we conduct ablation studies and deduce the following insights.

Table 7. Ablation for different feature extraction backbones.

Methods	Detection			End-to-end
	P	R	F	None
Resnet-50 [17]	88.87	76.47	82.20	60.06
ViT-T [12]	90.17	72.90	80.62	59.40
Swin-T [29]	**93.59**	**75.20**	**83.40**	**67.06**

Swin Transformer is a strong visual backbone for text spotting. The Swin-Tiny [29] feature extraction backbone in the Swin-TESTR framework gives a significant perfor-



(a) Total-Text [8] (b) CTW 1500 [27] (c) ICDAR 2015 [22] (d) Vintext [34] (e) MLT 2017 [33] (f) ReCTS [53]

Figure 4. Some illustration of our method on different datasets and their feature maps from the last three layers of backbone respectively. Zoom in for better visualization.

mance gain over ViT-Tiny [12] and Resnet-50 [17] backbones in both detection and recognition metrics. Almost a 4% change in detection precision and a 1% change in the detection F-measure over Resnet-50 is observed. We get almost a massive 7% improvement in the End-to-End recognition results when it is run on the Total-Text [8] dataset as shown in Table 7. This justifies that the window-based local attention computed with Swin-T can be really effective for special text regions with smaller local boundaries.

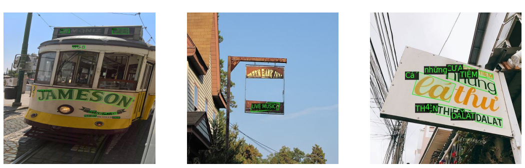


Figure 5. Some failure cases of our method on Swin-TESTR. Zoom in for better visualization.

Semi-Supervised Training Strategy. To assess the data dependency of Swin-TESTR, we conducted a series of experiments involving different subsets of the dataset. These subsets consisted of 25%, 50%, 75%, and the entire dataset, respectively. The objective was to investigate how Swin-TESTR's performance varied with the amount of training data available. Our findings provided compelling evidence of a strong data dependency exhibited by Swin-TESTR. Notably, as we increased the proportion of data used for training, the model's performance consistently improved. Training with 25% and 50% of the dataset resulted in noticeable enhancements, indicating the reliance on available data.

Table 8. Performance Evaluation with different % of labeled data

% Labels	Detection			End-to-end
	P	R	F	None
25%	89.88	81.8	85.65	65.71
50%	90.56	82.75	86.48	70.38
75%	89.61	85.73	87.63	71.17
All	90.58	**85.46**	**87.95**	**74.13**

5. Conclusion and Future Work

In this work, we have explored a new direction towards benchmarking of text-spotting models in domain adaptation settings. Comprehensive experiments on different benchmarks exhibit how the proposed baseline achieves competitive performance against the SOTA approaches under different settings. Moreover, diverse multilingual datasets could help the model in getting better intermediate representations during pre-training. This opens further scope for more diverse linguistic and visually-rich datasets to be introduced for aggravating further research.

Acknowledgement

This work acknowledges the Spanish projects GRAIL PID2021-126808OB-I00, DocAI 2021-SGR-01559, the CERCA Program / Generalitat de Catalunya, the FCT-19-15244, and PhD Scholarship from AGAUR 2023 FI-3-00223 and special thanks to Subhankar Ghosh, Prasun Roy and Kunal Biswas for the insightful discussions.

References

[1] Aviad Aberdam, David Bensaïd, Alona Golts, Roy Ganz, Oren Nuriel, Royee Tichauer, Shai Mazor, and Ron Litman. Clipter: Looking at the bigger picture in scene text recognition. *arXiv preprint arXiv:2301.07464*, 2023. 1, 3

[2] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4715–4723, 2019. 3

[3] Jeonghun Baek, Yusuke Matsui, and Kiyoharu Aizawa. What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3113–3122, 2021. 2, 3

[4] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9365–9374, 2019. 2

[5] Ayan Banerjee, Sanket Biswas, Josep Lladós, and Umapada Pal. Swindocsegmenter: An end-to-end unified domain adaptive transformer for document instance segmentation. *arXiv preprint arXiv:2305.04609*, 2023. 1

[6] Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R Manmatha. Latr: Layout-aware transformer for scene-text vqa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16548–16558, 2022. 1

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 4

[8] Chee-Kheng Ch'ng, Chee Seng Chan, and Cheng-Lin Liu. Total-text: toward orientation robustness in scene text detection. *International Journal on Document Analysis and Recognition (IJDAR)*, 23(1):31–52, 2020. 1, 2, 3, 4, 7, 8

[9] Alloy Das, Sanket Biswas, Umapada Pal, and Josep Lladós. Diving into the depths of spotting text in multi-domain noisy scenes. *arXiv preprint arXiv:2310.00558*, 2023. 3

[10] Alloy Das, Prasun Roy, Saumik Bhattacharya, Subhankar Ghosh, Umapada Pal, and Michael Blumenstein. Fast: Font-agnostic scene text editing. *arXiv preprint arXiv:2308.02905*, 2023. 2

[11] Jinhong Deng, Xiulian Luo, Jiawen Zheng, Wanli Dang, and Wen Li. Text enhancement network for cross-domain scene text detection. *IEEE Signal Processing Letters*, 29:2203–2207, 2022. 2

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 7, 8

[13] Shancheng Fang, Zhendong Mao, Hongtao Xie, Yuxin Wang, Chenggang Yan, and Yongdong Zhang. Abinet++: Autonomous, bidirectional and iterative language modeling for scene text spotting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1, 2, 3, 6, 7

[14] Sergi Garcia-Bordils, George Tom, Sangeeth Reddy, Minesh Mathew, Marçal Rusiñol, CV Jawahar, and Dimosthenis Karatzas. Read while you drive-multilingual text tracking on the road. In *Document Analysis Systems: 15th IAPR International Workshop, DAS 2022, La Rochelle, France, May 22–25, 2022, Proceedings*, pages 756–770. Springer, 2022. 1

[15] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013. 1

[16] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2315–2324, 2016. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7, 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5020–5029, 2018. 3

[19] Mingxin Huang, Yuliang Liu, Zhenghao Peng, Chongyu Liu, Dahua Lin, Shenggao Zhu, Nicholas Yuan, Kai Ding, and Lianwen Jin. Swintextspotter: Scene text spotting via better synergy between text detection and text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4593–4603, 2022. 1, 2, 3, 5, 6, 7

[20] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091, 2022. 1, 3, 7

[21] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International journal of computer vision*, 116:1–20, 2016. 1

[22] 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, 3, 4, 7, 8

[23] Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. Dit: Self-supervised pre-training for document image transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3530–3539, 2022. 1

[24] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Thirty-first AAAI conference on artificial intelligence*, 2017. 3

[25] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5676–5685, 2018. 3

[26] 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 *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9809–9818, 2020. 2, 3, 4, 6

[27] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Canjie Luo, and Sheng Zhang. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recognition*, 90:337–345, 2019. 1, 3, 4, 7, 8

[28] Yuliang Liu, Chunhua Shen, Lianwen Jin, Tong He, Peng Chen, Chongyu Liu, and Hao Chen. Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. *arXiv preprint arXiv:2105.03620*, 2021. 1, 2, 3, 6

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2, 4, 7

[30] Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. Towards end-to-end unified scene text detection and layout analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1049–1059, 2022. 2

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[32] Subhajit Maity, Sanket Biswas, Siladittya Manna, Ayan Banerjee, Josep Lladós, Saumik Bhattacharya, and Umapada Pal. Selfdocseg: A self-supervised vision-based approach towards document segmentation. *arXiv preprint arXiv:2305.00795*, 2023. 1

[33] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 1454–1459. IEEE, 2017. 2, 3, 4, 8

[34] Nguyen Nguyen, Thu Nguyen, Vinh Tran, Minh-Triet Tran, Thanh Duc Ngo, Thien Huu Nguyen, and Minh Hoai. Dictionary-guided scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7383–7392, 2021. 3, 4, 6, 7, 8

[35] Liang Qiao, Ying Chen, Zhanzhan Cheng, Yunlu Xu, Yi Niu, Shiliang Pu, and Fei Wu. Mango: A mask attention guided one-stage scene text spotter. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2467–2476, 2021. 1, 2, 3, 6, 7

[36] Liang Qiao, Sanli Tang, Zhanzhan Cheng, Yunlu Xu, Yi Niu, Shiliang Pu, and Fei Wu. Text perceptron: Towards end-to-end arbitrary-shaped text spotting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11899–11907, 2020. 3, 6

[37] Siyang Qin, Alessandro Bissacco, Michalis Raptis, Yasuhisa Fujii, and Ying Xiao. Towards unconstrained end-to-end text spotting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4704–4714, 2019. 3

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[39] Prasun Roy, Saumik Bhattacharya, Subhankar Ghosh, and Umapada Pal. Stefann: scene text editor using font adaptive neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2

[40] Zejiang Shen, Ruochen Zhang, Melissa Dell, Benjamin Charles Germain Lee, Jacob Carlson, and Weining Li. Layoutparser: A unified toolkit for deep learning based document image analysis. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I 16*, pages 131–146. Springer, 2021. 3, 7

[41] Mohamed Ali Souibgui, Sanket Biswas, Andres Mafla, Ali Furkan Biten, Alicia Fornés, Yousri Kessentini, Josep Lladós, Lluis Gomez, and Dimosthenis Karatzas. Text-diae: Degradation invariant autoencoders for text recognition and document enhancement. *arXiv preprint arXiv:2203.04814*, 2022. 2

[42] Adarsh Subbaswamy, Roy Adams, and Suchi Saria. Evaluating model robustness and stability to dataset shift. In *International conference on artificial intelligence and statistics*, pages 2611–2619. PMLR, 2021. 2

[43] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. Unifying vision, text, and layout for universal document processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19254–19264, 2023. 1

[44] Hsueh-Cheng Wang, Chelsea Finn, Liam Paull, Michael Kaess, Ruth Rosenholtz, Seth Teller, and John Leonard. Bridging text spotting and slam with junction features. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3701–3708. IEEE, 2015. 1

[45] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *2011 International conference on computer vision*, pages 1457–1464. IEEE, 2011. 3

[46] Wenhai Wang, Enze Xie, Xiang Li, Xuebo Liu, Ding Liang, Yang Zhibo, Tong Lu, and Chunhua Shen. Pan++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1, 3

[47] Weijia Wu, Ning Lu, Enze Xie, Yuxing Wang, Wenwen Yu, Cheng Yang, and Hong Zhou. Synthetic-to-real unsuper-

vised domain adaptation for scene text detection in the wild. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 2, 3

[48] Ying Xu, Xu Zhong, Antonio Jose Jimeno Yepes, and Jey Han Lau. Forget me not: Reducing catastrophic forgetting for domain adaptation in reading comprehension. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 1

[49] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8751–8761, 2021. 1

[50] Maoyuan Ye, Jing Zhang, Shanshan Zhao, Juhua Liu, Tongliang Liu, Bo Du, and Dacheng Tao. Deepsolo: Let transformer decoder with explicit points solo for text spotting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19348–19357, 2023. 3

[51] Wenwen Yu, Yuliang Liu, Wei Hua, Deqiang Jiang, Bo Ren, and Xiang Bai. Turning a clip model into a scene text detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6978–6988, 2023. 3

[52] Chongsheng Zhang, Yuefeng Tao, Kai Du, Weiping Ding, Bin Wang, Ji Liu, and Wei Wang. Character-level street view text spotting based on deep multisegmentation network for smarter autonomous driving. *IEEE Transactions on Artificial Intelligence*, 3(2):297–308, 2021. 1

[53] Rui Zhang, Yongsheng Zhou, Qianyi Jiang, Qi Song, Nan Li, Kai Zhou, Lei Wang, Dong Wang, Minghui Liao, Mingkun Yang, et al. Icdar 2019 robust reading challenge on reading chinese text on signboard. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 1577–1581. IEEE, 2019. 2, 3, 4, 7, 8

[54] Xiang Zhang, Yongwen Su, Subarna Tripathi, and Zhuowen Tu. Text spotting transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9519–9528, 2022. 1, 2, 3, 4, 6, 7

[55] Hong-Yu Zhou, Chixiang Lu, Sibei Yang, and Yizhou Yu. Convnets vs. transformers: Whose visual representations are more transferable? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2230–2238, 2021. 2

[56] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 4