Bharat Scene Text: A Novel Comprehensive Dataset and Benchmark for Indian Language Scene Text Understanding

Anik De, Abhirama Subramanyam Penamakuri, Rajeev Yadav, Aditya Rathore, Harshiv Shah, Devesh Sharma, Sagar Agarwal, Pravin Kumar, Anand Mishra

[1]Indian Institute of Technology Jodhpur, Jodhpur, 342030, Rajasthan, India
https://vl2g.github.io/projects/IndicPhotoOCR/.

*Corresponding author(s). E-mail(s): mishra@iitj.ac.in;

Abstract

Reading scene text, that is, text appearing in images, has numerous application areas, including assistive technology, search, and e-commerce. Although scene text recognition in English has advanced significantly and is often considered nearly a solved problem, Indian language scene text recognition remains an open challenge. This is due to script diversity, non-standard fonts, and varying writing styles, and, more importantly, the lack of high-quality datasets and open-source models.

To address these gaps, we introduce the Bharat Scene Text Dataset (BSTD) – a large-scale and comprehensive benchmark for studying Indian Language Scene Text Recognition. It comprises more than 100K words that span 11 Indian languages and English, sourced from over 6,500 scene images captured across various linguistic regions of India. The dataset is meticulously annotated and supports multiple scene text tasks, including: (i) Scene Text Detection, (ii) Script Identification, (iii) Cropped Word Recognition, and (iv) End-to-End Scene Text Recognition. We evaluated state-of-the-art models originally developed for English by adapting (fine-tuning) them for Indian languages. Our results highlight the challenges and opportunities in Indian language scene text recognition. We believe that this dataset represents a significant step toward advancing research in this domain. All our models and data are open source.

Keywords: Multilingual Scene Text Recognition, Bharat Scene Text Dataset, IndicPhotoOCR

1 Introduction

Scene Text Recognition, especially for English, has advanced significantly in the last decade, thanks to a series of efforts on dataset creation [1–10] and model development [11–25]. In contrast, scene text recognition for Indian languages has been extremely underexplored and possibly more challenging (refer to Fig. 1). Note that Indian languages are used by more than 1.4 billion people, which is approximately 18% of the global population. Despite its usage and popularity, we observe that scene text recognition in Indian languages lacks a comprehensive benchmark and open-source models, except for some early efforts [26–33] that laid important groundwork, but did not aim for a comprehensive coverage of tasks or languages. We aim to fill this gap through our work.

To accelerate advancements in Indian Language Scene Text Recognition, we present a comprehensive benchmark called the *Bharat Scene*

Fig. 1: **Scene Text Understanding: Case of India.** This figure showcases typical street scenes from northern, eastern, western, and southern parts of India, highlighting the country's vast linguistic diversity. With 11 languages (five of them: Hindi, English, Bengali Gujarati and Tamil are shown here) including English commonly featured on signboards, scene text understanding in India presents unique challenges. While significant progress has been made in English Scene Text Recognition, *open-source comprehensive effort* for Indian language scene text understanding, including large public datasets and comprehensive models, are still limited. Our work seeks to address this gap and advance the field.

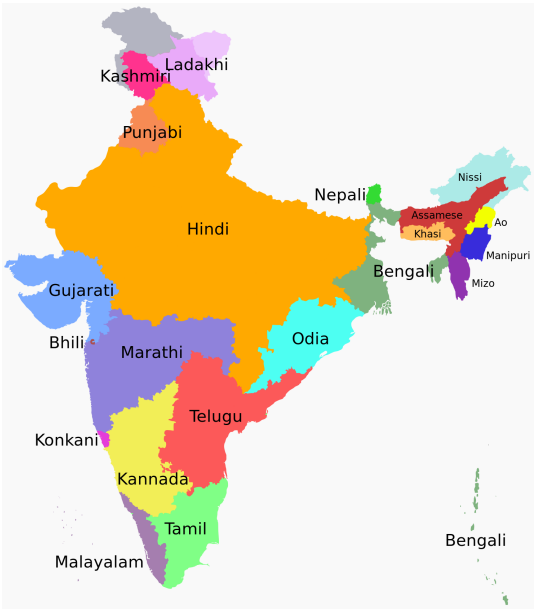

Fig. 2: Language map of India (Source: Wikimedia). The BSTD covers scene text images from 11 prominent languages namely Assamese, Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Odia, Punjabi, Tamil and, Telugu. Additionally, it contains English, as English is often included as one of the languages as signboard in India.

Text Dataset (BSTD)[1]. The BSTD comprises 6.5K scene images that feature more than 100K words in 11 Indian languages, along with English. These images, sourced from Wikimedia[2], span various signboards and accidental background text from different Indian cities and towns, ensuring that the dataset captures the linguistic diversity of India. We obtain a polygon-level bounding box for each visible text in the image and the text it contains using rigorous manual annotation. The dataset is publicly available for download here[3].

The BSTD can be used for the following scene text tasks: (i) scene text detection, (ii) script identification, (iii) cropped word recognition, and (iv) end-to-end scene text recognition. In this work, we benchmark the BSTD using our new pipeline, IndicPhotoOCR, which integrates state-of-the-art architectures across its modules. We firmly believe that these benchmarks will help accelerate advancements in Indian language scene text recognition.

We list our contributions here:

1. We introduce the Bharat Scene Text Dataset (BSTD) – a large-scale benchmark for Indian language scene text recognition, consisting of 6,582 scene images featuring 1,06,478 words in 11 Indian languages and English. Each image is manually annotated with polygon-level bounding boxes and corresponding text and script, ensuring high-quality data for research and applications. (Section 2)
2. We evaluated BSTD using state-of-the-art techniques, including off-the-shelf methods and fine-tuned models on synthetic and/or BSTD train data (as applicable). Detailed benchmark results are provided for all key tasks, i.e., text detection, script identification, cropped word

[1]India, traditionally known as *Bharat*, reflects the nation's rich cultural and linguistic diversity, which inspired the naming of our dataset.

[2]https://www.wikimedia.org/

[3]https://github.com/Bhashini-IITJ/BharatSceneTextDataset

Dataset	# Images	# Words	# Languages	Tasks Supported		
				Detection	Script Identification	Recognition
ICDAR 2003 [1]	509	2,268	1	✓	✗	✓
SVT [2]	350	725	1	✓	✗	✓
IIIT-5K [3]	-	5,000	1	✗	✗	✓
ICDAR 2013 [5]	462	5,003	1	✓	✗	✓
ICDAR 2015 [6]	1,500	6,545	1	✓	✗	✓
CUTE80 [7]	80	288	1	✓	✗	✓
Total-Text [8]	1,555	11,459	1	✓	✗	✓
IIIT-TL-STR [30]	440	3,035	2	✗	✗	✓
IIIT-ILST [27]	-	3,168	3	✗	✗	✓
MLT-17* [34]	18,000	96,000	9	✓	✓	✓
MLT-19* [10]	20,000	191,000	10	✓	✓	✓
IIIT-indicSTR12** [31]	-	27,000	12	✗	✗	✓
BSTD (This Work)	6,582	1,06,478	12	✓	✓	✓

Table 1: Comparison of the Bharat Scene Text Dataset (BSTD) with existing scene text datasets. BSTD supports all three core tasks—text detection, script identification, and recognition (and, thereby end-to-end evaluation) across 12 Indian languages, including English. It is the **only publicly available dataset for Indian languages that enables comprehensive end-to-end scene text understanding**. *: *MLT-17 and MLT-19 are not focused on Indian languages, they only include Hindi and Bengali.* **: IIIT-IndicSTR12 is primarily annotated for cropped word recognition and can not be studied to evaluate end-to-end scene text recognition as ours.

recognition, and end-to-end scene text recognition. These evaluations establish a strong baseline for future research and advancements in multilingual scene text recognition, particularly for Indian languages. (Section 4)

3. We introduce an open source toolkit, namely IndicPhotoOCR, designed to detect, identify, and recognize text in English and 11 Indian languages. The toolkit is easy to install and replicate, making it accessible to researchers and developers. (Section 5)

2 Bharat Scene Text Dataset

India, often referred to as Bharat (the Indian name for the country), represents a rich and diverse cultural and linguistic landscape, as illustrated in Fig. 2. In the context of scene text understanding, the vast multilingual environment poses unique challenges, as the scripts, writing styles, and text formats vary significantly between different regions. In addition, the identification of scripts becomes extremely important. Although there has been some effort in collecting and benchmarking Indian language scene text understanding tasks, it falls short compared to efforts for scene text understanding in English [1–10]. Please refer to Table 1 for a detailed comparison of the Bharat Scene Text Dataset compared to other related scene text datasets.

To address this gap, we present the Bharat Scene Text Dataset, a novel and comprehensive dataset specifically curated to represent the linguistic diversity of India. By using the term "Bharat", we not only acknowledge the nation's deep-rooted cultural identity but also highlight our focus on supporting scene text understanding across 11 major Indian languages. This dataset contains 6,582 scene images and 1,06,478 word images in total. This dataset and associated annotations can be downloaded from our project website[4]. In this section, we give details of dataset curation and annotation, supported tasks, train-test splits followed by dataset analysis.

2.1 Dataset Curation and Annotation

We illustrate the pipeline for constructing the Bharat Scene Text Dataset from Wikimedia Commons in Fig. 3. The dataset is curated and annotated in the following three stages:

[4] https://vl2g.github.io/projects/IndicPhotoOCR/

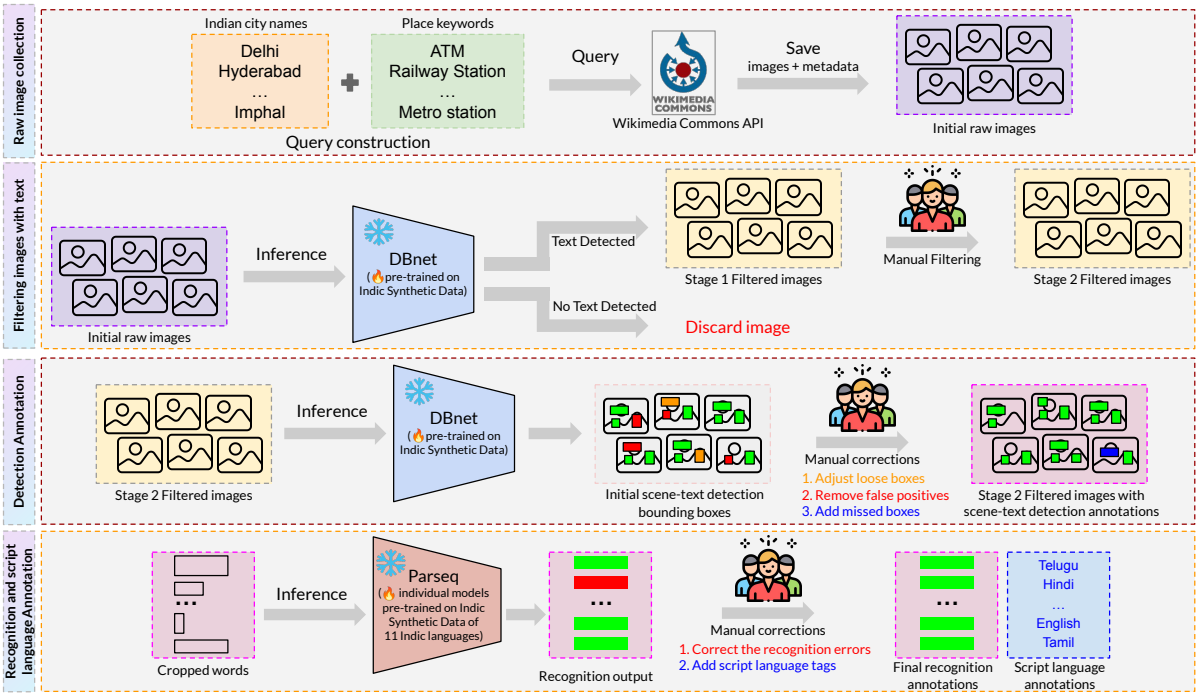

Fig. 3: Pipeline for constructing a multilingual scene-text dataset from Wikimedia Commons images. The process begins with query construction using combinations of Indian city names and place-related keywords to retrieve relevant images via the Wikimedia Commons API. A two-stage filtering process using DBnet [19] (pre-trained on Indic Synthetic Data) removes non-textual images and manually filters relevant ones. Scene-text detection annotations are generated using DBnet and refined through manual corrections. Finally, cropped word images are recognized using the PARSeq [22] model (trained on 11 Indic languages using synthetic data), with recognition errors corrected and script language tags added to produce the final annotated dataset.

2.1.1 Stage-1: Image collection

We compile a comprehensive list of cities and towns in various regions of India where 11 selected languages are spoken predominantly. Furthermore, we curate a catalog of public places, such as 'bus stations', 'banks', 'schools', 'railway stations', 'hospitals', 'ATMs', 'metro stations', 'street signs', 'airport', 'shop entrance', 'billboards', 'press conference', etc. By combining the city and place names as search queries, e.g. 'Delhi Railway Station', we retrieved images along with their metadata from the Wikimedia Commons API[5]. It should be noted that not all retrieved images contain scene text at this stage, which requires subsequent filtering. At this stage, we obtain 50K images.

2.1.2 Stage-2: Image filtering

We filter the images harvested in the previous step using the pre-trained scene text detection method [19]. To this end, we run this detector and obtain text even with low confidence. We retain images containing at least one word, while others were discarded. In addition, we ask human annotators to review the filtered set to eliminate false positives. The final data set comprises 6,582 images that contain scene text in 11 Indian languages and English.

2.1.3 Stage-3: Semi-automatic annotation

We annotate the final filtered images for scene text detection in two steps to reduce human annotation efforts. We obtain bounding box annotation using

[5] https://commons.wikimedia.org/w/api.php


```
{
"annotations": {
        "polygon_0":
            {
               "coordinates": [[687,259],[755,244],[759,289],[700,303] ],
               "text": "TUNI",
                "script_language": "english"
            },
        "polygon_1":
            {
               "coordinates": [[595,282],[617,281],[632,268],...[602,328]]
               "text": "तुनी",
               "script_language": "hindi"
            },
        "polygon_2":
            {
               "coordinates: [[487,265],[513,259],[541,277], ..., [481,313]],
               "text": "తుని",
              "script_language": "telugu"
            },
"split": "test",
"url": "https://upload.wikimedia.org/wikipedia/commons/b/b6/Tuni_railway_station_view_02.jpg",

}
```


Fig. 4: An example image from BSTD along with the corresponding JSON annotation.

a fine-tuned DBnet model on synthetic Indian language data[6]. Since these automatically obtained bounding boxes are not always perfect, they are refined by our hired annotators using an in-house tool to edit, draw, or delete polygons around the scene text words.

Once the bounding boxes were obtained, they were assigned to the annotators and assigned to label each box with the script it contains. These annotators were well-versed in script identification and were also informed about the geographical origin of each image. With this context, they were able to accurately label each bounding box with the corresponding script. We then use PARSeq [22], fine-tuned on synthetic data for Indian languages, to read the text within each bounding box. The bounding boxes, along with the recognized text, are then provided to language expert annotators. These annotators review the text and correct any recognition errors or retain the text as is if it is accurate.

2.2 Annotator Details

We employed two distinct groups of annotators for this work. Initially, three human annotators for image filtering (Section 2.1.2) and bounding box annotation (Section 2.1.3). For the next two steps, text correction and script assignment (refer 2.1.3), we hired eleven language-specific experts, one for each target language. The annotators were proficient in reading, understanding, and writing their native language, as well as English. They were shown cropped words and asked to correct the pseudo-recognition annotations and assign the script. All final annotations underwent an additional quality check to eliminate potential errors and noise. The annotators were compensated at rates aligned with standard wages for a tier-2 city in India. The entire process–pseudo-annotation, manual correction, and quality verification took 10 months. Fig 4 shows the fine grained nature of the annotations done in this work.

2.3 Supported Tasks and Evaluation Protocols

The Bharat Scene Text Dataset (BSTD) enables us to study the following four tasks:

1. **Scene Text Detection**: where given an image, the task is to detect all the text instances in the image along with their bounding boxes. We provide polygon-level tight bounding boxes for each word image in the dataset. The popular metrics such as Precision, Recall and F1-score using TedEval evaluation [35] can be used to assess the performance of the method.
2. **Script Identification**: This task involves identifying the language of the script present in a given cropped word image. The performance of the model is evaluated using accuracy.

[6]Refer to Section-3.3.1 for details of synthetic data generation

Additionally, we include a confusion matrix to analyze where the model's predictions fail.

3. **Cropped Word Recognition**: where the task is to recognize and extract the text given a cropped word image provided the script/language of text is known. For cropped word recognition, the word recognition rate (WRR) is used as a performance metric.
4. **End-to-End Scene Text Recognition**: This task is the closest to the natural setting. Here, given a scene image, the task is to recognize all the instances of text. Solving end-to-end Scene Text Recognition for Indian languages, often requires a pipeline of scene text detection, script identification followed by cropped word recognition. We use mean word recognition rate (WRR) and character recognition rate (CRR) over all the test images in the dataset. The WRR and CRR for an image are defined as follows:

$$\text{WRR} = \left(1 - \frac{S_w + D_w + I_w}{N_w^{\text{total}}}\right) \times 100 \quad (1)$$

$$\text{CRR} = \left(1 - \frac{S_c + D_c + I_c}{N_c^{\text{total}}}\right) \times 100 \quad (2)$$

where S, D, and I with subscript w and c denote the number of substitutions, deletions, and insertions, respectively, at the word or character level. Further, N_w^{total} and N_c^{total} denote the total number of words and characters present in the ground-truth transcription in the image, respectively. It should be noted that due to false alarms (i.e., excessive insertions of words or characters in the recognized text) or in some rare cases of any missing annotation, the computed WRR or CRR values can become negative for few images. To keep the interpretation simple, we truncate such values to 0. Furthermore, standard WRR and CRR computations assume a fixed reading order; however, since reading order in scene text images can be subjective, we relax this constraint in our evaluation. Further, to make interpretation of results more clean, we also use standard precision, recall, and F1-score to evaluate end-to-end performance. Note that precision is defined as the number of correctly recognized words over the total number of recognized words, while recall is defined as the number of correctly recognized words over the total number of words in the ground truth.

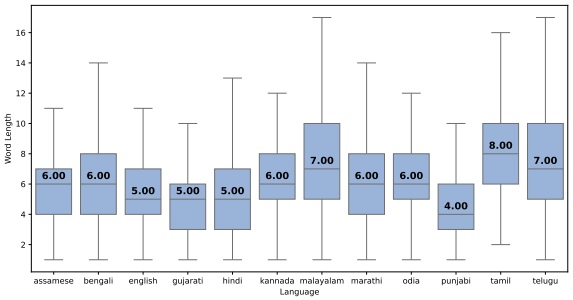

Fig. 5: Box plot highlighting the word length range across all the 12 languages.

Set	Images	Bounding Box Annotations
Total	6,582	1,26,292
Train	5,263	94,128
Test	1,319	32,164

Table 2: Summary of image and detection annotation counts in the BSTD dataset, detailing the total, training, and test set distributions

We randomly divided the 6,582 images into an 80-20% train-test split. As a result, the training set consists of 5,263 scene images and 77,726 cropped words, each with corresponding language and text annotations. The test set contains 1,319 scene images and 28,752 cropped words with their respective language and text annotations. The image-wise and cropped world-level train/test split is summarized in Table 2 and Table 3, respectively. For the script identification task, we randomly selected an equal number of word images from each of the 12 languages. The split comprises 1,800 training images and 478 testing images per language.

2.4 Dataset Analysis

In this section, we present a comprehensive analysis of the Bharat Scene Text Dataset (BSTD) to evaluate its quality and the challenges it poses. Toward this end, we first report the language-wise word count in Table 3. We observe that it has a good coverage of all the major Indian

Language	#Word images	Train	Test
Assamese	4,132	2,627	1,505
Bengali	6,304	4,936	1,368
English	41,696	29,123	12,573
Gujarati	2,899	1,884	1,015
Hindi	19,773	14,927	4,846
Kannada	2,928	2,208	720
Malayalam	2,940	2,393	547
Marathi	5,113	3,917	1,196
Odia	4,192	3,148	1,044
Punjabi	11,199	8,319	2,880
Tamil	2,542	2,029	513
Telugu	2,760	2,215	545
Others	19,814	-	-
Total	1,26,292	77,726	28,752

Table 3: Cropped word image data split for the BSTD dataset, showing the number of annotated instances, train images, and test images across multiple languages. Here, Others correspond to those words which are not from the listed languages here (e.g., Meitei and Urdu). We plan to include their annotation in subsequent versions of our dataset.

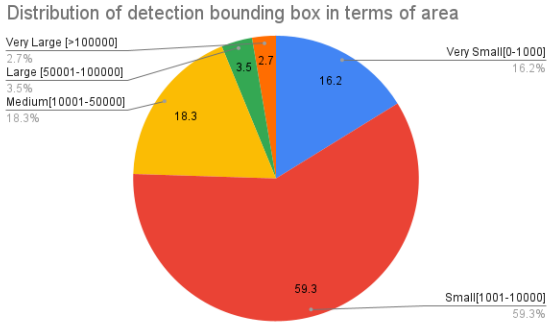

Fig. 6: Distribution of the bounding box size in our dataset, in terms of pixels.

languages with minimum 2,542 word images for Tamil. This makes the dataset linguistically rich. We further analyze the average word length for each language in our dataset in Fig. 5). The plot reveals that Malayalam, Tamil, and Telugu contain longer words on average, whereas English and Hindi have shorter word lengths. This observation, though potentially intuitive, holds practical implications for the development of specialized language models in the future.

Fig. 7: **Word cloud for recognition annotations in our BSTD dataset**: Highlighting the diversity of vocabulary across 12 languages. The tiles are arranged row-wise from left to right as follows: the first row contains Hindi, Assamese, and Gujarati; the second row shows English, Malayalam, and Kannada; the third row corresponds to Odia, Panjabi, and Tamil; and the fourth row includes Telugu, Marathi, and Bengali.

Moving further, we studied the distribution of the bounding box area in Fig. 6. It indicates that most text regions (59.4%) are small (1001–10,000 pixels), while only 2.6% of text instances fall into the largest category (above 100,000 pixels). These findings highlight the diverse textual characteristics within BSTD and it contains a mix of small and large text regions within images making it challenging for detection. Furthermore, the high percentage of small text regions suggests that many images contain *incidental text* rather than just focused text, which adds to the dataset's difficulty implying the real word scenario.

Finally, to illustrate diversity in the dataset, we show word clouds for all 12 languages in our data set in Fig. 7. As can be seen, the most frequent words (represented by the largest font size) across all languages point to text commonly found on public signs, government buildings, and commercial establishments. In addition, the vocabulary is dominated by nouns that are staples of scene text recognition tasks. These are words that one would expect to see "in the wild". Some of the cropped word samples from BSTD samples are shown in Fig. 8.

Fig. 8: **A selection of challenging cropped words from our dataset.** Few key challenges include eroded text (c, d, k), handwritten text (f), occluded text (e, h, l), ambiguous text (a, b, i), text in different orientations (g, j).

In short, the analysis of the Bharat Scene Text Dataset suggests that it has diverse linguistic and geographical coverage, with a clear focus on text from public, commercial and transport-related signage, and has a mix of focused and incidental text, making it a valuable resource for developing robust real-world text recognition models for Indian languages. Further, while widely spoken languages such as Hindi, English, Bengali, and Marathi are well represented in the dataset, low-resource languages including Assamese, Gujarati, Kannada, Malayalam, Odia, Punjabi, Tamil, and Telugu also have a substantial presence, with at least 2,542 word images for Tamil (see Table 3). This shows that the dataset is not only strong in high-resource languages but also sufficiently representative of low-resource languages, making it suitable for comprehensive multilingual text recognition research.

3 IndicPhotoOCR: Proposed Baseline

Scene text recognition in a multilingual environment typically involves three key modules: (i) Scene Text Detection, (ii) Script Identification, and (iii) Cropped Word Recognition. In the absence of a dedicated architecture for Indian language scene text recognition in the literature, we develop a strong baseline by adapting or fine-tuning (as needed) state-of-the-art architectures across various modules. We describe each of these modules below and collectively refer to this pipeline as IndicPhotoOCR.

3.1 Scene Text Detection

Our systematic analysis of modern scene text detectors revealed that these models are largely language-agnostic. Hence, we directly incorporate them into our pipeline without language-specific modifications. In particular, we adopt TextBPN++ [24] for scene text detection, as it demonstrated superior performance compared to other detectors in our evaluations. Detailed results for this stage are presented in the experimental section 4.1.

3.2 Script Identification

For script classification, we used the `google/vit-base-patch16-224-in21k` Vision Transformer (ViT) model, which consists of 12 transformer layers and uses 16×16 patch embeddings. Pre-trained on the large-scale ImageNet-21k dataset, this model provides rich visual representations that are well-suited for transfer learning for the task of script identification.

Language	Train Size	Test Size	Vocabulary Size	#Fonts
Assamese	8,092,205	899,134	636,312	44
Bengali	8,475,805	941,757	4,447,622	26
Gujarati	6,431,896	714,656	4,255,185	57
Hindi	8,357,906	811,976	4,195,486	101
Kannada	4,002,547	444,728	8,288,718	22
Malayalam	7,199,420	400,000	4,857,185	32
Marathi	3,168,386	352,043	3,739,734	101
Odia	4,041,852	449,095	1,144,724	35
Punjabi	5,479,762	608,863	2,083,909	40
Tamil	5,652,718	628,080	7,842,880	70
Telugu	3,485,570	387,286	5,705,371	42

Table 4: Statistics for Synthetic Dataset generated using [36] for 11 Indian languages.

To adapt the model for our task, we appended a custom classification head that projects the learned visual features into output logits, corresponding to either 12 classes (for full multilingual classification) or 3 classes (a practical setting with Hindi, English, and one regional language at a time).

We fine-tuned the model, including both the transformer backbone and the classification head, using the AdamW optimizer with a learning rate of 2×10^{-4} and a batch size of 16 using BSTD-train set. This ViT-based approach consistently outperformed other configurations, demonstrating strong generalization in both script classification settings of the 3-way and 12-way scripts.

3.3 Cropped Word Recognition

Once text instances are detected and their scripts identified, the next step is text recognition. Unlike detection, this task relies heavily on language-specific modeling and requires large-scale training data. *Off-the-shelf models for English text recognition cannot be directly applied as they are not designed for this task.* Instead, we fine-tune a successful scene text recognition architecture (namely PARSeq [22]) for the 11 Indian languages in our study. A key challenge in this setting is the scarcity of real-world scene text data for low-resource languages, in contrast to English, where recognizers are trained on millions of samples. To address this, we generated synthetic word images to augment the training data and bridge this gap as described in the following section.

3.3.1 Synthetic Data Generation

To train models for each of the 11 languages, we required a large-scale dataset of scene text images. However, such datasets for Indian languages were not readily available. This lack of data led us to adopt an existing method for generating synthetic scene text images. We used SynthText [36] to generate large-scale data, with statistics summarized in Table 4. SynthText [36] provided 8000 background images, which we leveraged as the background image for our synthetic data.

To generate scene text in different Indian languages, we needed appropriate vocabulary and fonts for each language. We sourced vocabulary from the AI4Bharat-IndicNLP Dataset[7] introduced in the IndicNLPSuite [29]. SynthText also accounts for local 3D geometry, enabling varied orientation of text placement within the scene. Although the generated images lacked complete realism, the scale of data generation offered significant value for our training needs.

This synthetically generated data is then used to train the PARSeq model for each language. PARSeq requires accurate character sets, which vary by language (e.g. Assamese: 91, Bengali: 91, Gujarati: 89; Hindi: 125, Kannada: 85, Malayalam: 116, Marathi: 125, Odia: 90, Punjabi:75, Tamil: 64, Telugu: 98). During this stage, we used an input image size of 32×128, a maximum label length of 25, and random augmentation. Following synthetic training, the best model is then fine-tuned on real data (BSTD) using decaying learning rate applied layer-wise.

4 Experimental Analysis

In this section, we provide a detailed evaluation of the proposed baseline, namely IndicPhotoOCR, and its individual modules on the newly introduced Bharat Scene Text Dataset. We compare it with several open-source OCR systems, including Tesseract [37], PaddleOCR [38], EasyOCR [39], and SuryaOCR [40]. For end-to-end evaluation, we also compare it with commercial closed-source systems such as Google OCR [41] and GPT-4 [42]. In addition, we include appropriate competitive approaches to evaluate individual modules. Our results reveal both the strengths and limitations

[7] https://github.com/AI4Bharat/indicnlp_corpus

Method	P	R	F1
EAST [14]	0.55	0.10	0.17
CRAFT [16]	0.51	0.22	0.19
DBNet [19]	0.71	0.50	0.59
Hi-SAM [43]	**0.75**	0.19	0.46
TextBPN++ [24]	**0.75**	**0.78**	**0.77**

Table 5: Scene Text Detection results on BSTD-Test Set using techniques prevalent in English scene text datasets.

	Trained on BSTD			
Language	AlexNet [44]	CRNN [12]	CLIP [46]	Proposed
Assamese	53.1	72	75.6	**92.2**
Bengali	54	75.7	77.5	**93.3**
Kannada	54.7	73.8	82.7	**94.5**
Malayalam	54.3	71.6	84.1	**90.8**
Marathi	45.9	62.3	70.1	**79.6**
Odia	50.1	87.2	86.7	**93.5**
Tamil	51.3	75.3	79.6	**93.6**
Telugu	54.9	82.9	86.9	**95.0**
Gujarati	50.7	70.2	79.1	**83.1**
Punjabi	51.3	81.1	79.7	**92.3**

Table 6: **Script Identification Accuracy in 3-way classification setting** on Hindi vs English vs third regional Indian language shown in the language column.

of existing systems in handling diverse multilingual Indian scene text. They also demonstrate the effectiveness of IndicPhotoOCR in a challenging setting. Please note that all the results are reported on the Bharat Scene Text Dataset (test set).

4.1 Results on Scene-text detection

Our proposed baseline, IndicPhotoOCR, employs TextBPN++ [24] for scene text detection. We compare its performance with several widely used scene text detection methods, including EAST [14], CRAFT [16], DBNet [19], and Hi-SAM [43]. The comparison results are summarized in Table 5. We observe that TextBPN++ significantly outperforms the other detectors, achieving higher F1-score and recall. Although Hi-SAM is also equally precise, it misses out many in detecting many Indian language words. Please note that we use TedEval [35] evaluation for reporting these numbers.

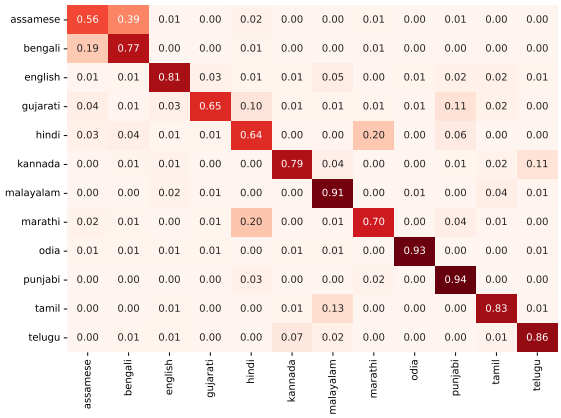

Fig. 9: Normalized confusion matrix illustrating the performance of script identification model across 12 Indian. **(Best viewed in color)**.

4.2 Results on Script Identification

For script identification, we evaluate performance under two settings: (i) **3-way classification**, where the model distinguishes among a regional language, Hindi and English. This setup reflects a common real-world scenario in India, where signboards often include all three languages. (ii) **12-way classification**, where the model selects from the English and 11 Indian languages.

We use the following baselines: (i) AlexNet [44]: a popular CNN-based Imagenet-pretrained image classification model finetuned on trainset of our script identification data, (ii) CRNN [45]: a CNN-based scene text recognition model with the ability to encode sequence information, adapted for script identification, followed by state-of-the-art transformer architectures for image classification; (iii) CLIP [46] and (iv) ViT [47], respectively. All the results reported in Table 6 correspond to the models trained on real images.

For the 3-way classification, the results are reported in Table 6. We observe that VIT-based proposed script identification clearly outperforms other baselines. For 12-way script classification, the CLIP-based baseline achieves an accuracy of 67.7%, while our proposed ViT-based approach achieves 80.5%. The corresponding normalized confusion matrix is presented in Fig. 9. While the overall results are satisfactory, the model

frequently confuses Assamese with Bengali and Hindi with Marathi. This is primarily due to their sharing Unicode patterns. Also, since they share majority of the Unicodes, these two confusion does not affect significantly our end-to-end performance. We have also noticed that other languages frequently confuse with English, likely due to some data bias or weak context for small words (such as '&', 'ON', etc.). Our end-to-end evaluation suggests that further improving script identification can significantly boost overall scene text recognition performance.

4.3 Results on Cropped Word Recognition

For the cropped-word recognition baselines, there are no models specifically designed for Indian languages. As a result, we must rely on English scene-text recognition approaches such as [22, 25, 45, 48]. However, adopting all these methods still demands substantial computational resources, especially when running recognition models across 12 languages. Therefore, we only choose PARSeq [22], one of the recent and effective recognition models, as our baseline. In addition, we compare the PARSeq-based baseline with the already available, CRNN [12] by AI4Bharat [49] and off-the-shelf open-source models, namely Tesseract [37], PaddleOCR [38], and EasyOCR [39].

ParSeq is trained on a combination of synthetic data and the BSTD dataset. Our baseline for cropped word recognition is based on the existing text recognition framework PARSeq [22]. We report results under two settings: (i) training PARSeq from scratch using synthetic data, and (ii) fine-tuning it on the BSTD training set. The results are presented in Table 7. Fine-tuning on real data leads to substantial improvements in recognition accuracy. This highlights the critical role of real-world data in scene text recognition. Also, as can be seen, Word Recognition Rates for cropped English scene text are relatively high (around 92%). However, performance drops noticeably for other languages: Marathi (86%), Bengali (82%), Tamil (80%), Aasamease (79%) Punjabi (75%), Odia (72%), and Hindi (71%) form the next tier of top-performing languages. The gap widens further for low-resource languages such as Telugu and Malayalam, whose recognition accuracies remain in the 50% range. These results clearly indicate substantial room for improving recognition performance across Indian languages.

In the next section we will be evaluating recognition performance in even more challenging end-to-end setting.

4.4 End-to-End Evaluation

We now present end-to-end evaluation results on the BSTD benchmark. For this, we compare our baseline IndicPhotoOCR with the following methods:

1. Closed-Source Commercial Baselines: We evaluated two commercial systems, namely GPT-4 with vision [42] and Google OCR [41] on the BSTD benchmark[8]. Although these comparisons are not entirely fair due to the opaque nature of their training data and model architectures, they offer a useful reference point to gage the current state of the art in Indian language scene text recognition.

We leveraged OpenAI's GPT-4 with Vision (accessed via the `gpt-4o` model) to perform zero-shot scene text recognition directly. We use the following structured prompt: *"Extract all text from this image. Return all words in a list format like this: ['word1', 'word2']"*, with a system message defining the model's role as: *"You are an OCR system that extracts text from images."* The interaction was handled through OpenAI's `ChatCompletion.create()` API, where both the instruction and the encoded image were passed as part of the message parameter. The response of the model was restricted using `max tokens = 500`. Although with more detailed prompt such as also locating text or chain-of-thoughts, this baseline may be improved. However, such prompt engineering is beyond the scope of this paper.

Further, we also utilized the Google Cloud Vision API to perform OCR on BSTD (Test Set).

2. Open-Source Baselines: Currently, apart from our own, no open-source system is explicitly designed for Indian language scene text recognition. However, we include comparisons with general-purpose OCR tools such as Tesseract [37]

[8] We obtained their results using their best available API on October 31, 2025.

Model	Train Data	Assamese	Bengali	English	Gujarati	Hindi	Kannada	Malayalam	Marathi	Odia	Punjabi	Tamil	Telugu	Average
PARSeq [22]	Synth	0.47	0.44	0.92	0.32	0.43	0.47	0.52	0.44	0.42	0.4	0.37	0.44	0.47
	BSTD Train	0.79	0.82	0.92	0.61	0.71	0.6	0.58	0.86	0.72	0.75	0.8	0.56	0.73
CRNN [12]	AI4Bharat	NA	NA	0.27	NA	0.13	NA	0.06	NA	NA	0.27	0.16	0.05	0.16
Off-the-shelf open source models														
Tesseract [37]	-	0.15	0.18	0.33	0.12	0.2	0.12	0.03	0.21	0.11	0.19	0.13	0.09	0.15
PaddleOCR [38]	-	NA	NA	0.61	NA	0.31	0.03	NA	0.36	NA	NA	0.28	0.15	0.29
EasyOCR [39]	-	0.12	0.24	0.29	NA	0.18	0.06	NA	0.25	NA	NA	NA	0.11	0.18

Table 7: Baseline Results on Cropped Word Scene Text Recognition for 12 languages on BSTD. Please note that PARSeq in used in our proposed baseline – IndicPhotoOCR.

		WRR on BSTD-Test										
Method	🔒/🔓	Bengali*	English	Gujarati	Devanagari*	Kannada	Malayalam	Odia	Punjabi	Tamil	Telugu	Average
ChatGPT [42]	🔒	0.08	0.36	0.09	0.2	0.03	0.05	0.04	0.12	0.21	0.1	0.13
Google OCR [41]	🔒	**0.43**	**0.37**	**0.32**	**0.55**	**0.24**	**0.37**	**0.43**	**0.46**	**0.52**	**0.42**	**0.41**
Tesseract [37]	🔓	0.04	0.02	0.02	0.03	0	0	0.02	0.02	0	0.02	0.02
SuryaOCR [40]	🔓	0.18	0.15	0.09	0.19	0.07	0.12	0.15	0.21	0.24	0.05	0.14
IndicPhotoOCR	🔓	0.45	0.37	0.3	0.39	0.29	0.31	0.38	0.41	0.42	0.31	0.36
IndicPhotoOCR (+oracle TD)	🔓	0.75	0.74	0.54	0.63	0.49	0.54	0.68	0.7	0.69	0.49	0.62
IndicPhotoOCR (+oracle TD+SI)	🔓	0.8	0.92	0.61	0.74	0.6	0.58	0.72	0.75	0.8	0.56	0.71
		CRR on BSTD-Test										
Method	🔒/🔓	Bengali*	English	Gujarati	Devanagari*	Kannada	Malayalam	Odia	Punjabi	Tamil	Telugu	Average
ChatGPT [42]	🔒	0.13	**0.46**	0.15	0.28	0.1	0.14	0.11	0.2	0.31	0.21	0.21
Google OCR [41]	🔒	**0.57**	0.42	**0.36**	**0.66**	**0.45**	**0.55**	**0.57**	**0.59**	**0.69**	**0.6**	**0.55**
Tesseract [37]	🔓	0.12	0.07	0.06	0.09	0.05	0.09	0.09	0.09	0.04	0.06	0.08
SuryaOCR [40]	🔓	0.29	0.24	0.15	0.33	0.19	0.26	0.29	0.31	0.45	0.26	0.28
IndicPhotoOCR	🔓	0.59	0.47	0.43	0.54	0.48	0.55	0.59	0.53	0.62	0.57	0.54
IndicPhotoOCR (+oracle TD)	🔓	0.87	0.78	0.64	0.74	0.68	0.8	0.84	0.84	0.78	0.73	0.77
IndicPhotoOCR (+oracle TD+SI)	🔓	0.93	0.97	0.75	0.89	0.86	0.86	0.89	0.9	0.94	0.83	0.88

Table 8: We conduct a comparative evaluation of commercial systems, including Google OCR and GPT-4, alongside our proposed baseline namely IndicPhotoOCR for **end-to-end scene text recognition** on the BSTD dataset. The evaluation measures script-wise performance using Word Recognition Rate (WRR) and Character Recognition Rate (CRR). Additionally, we present baseline results under two scenarios: (i) when text detection is provided as ground truth (denoted as oracle TD) and (ii) when both text detection and script identification are given as ground truth (denoted as TD+SI). " 🔒/🔓" indicates whether the tool is open-source (🔓) or closed-source (🔒). Bold and underline number shows best performing results in non-oracle setting for closed and open-source models respectively. IndicPhotoOCR is the best open-source model for all the Indian languages. The asterisks (*) on Bengali and Devanagari indicate that Assamese and Bengali languages were merged into Bengali, and Hindi and Marathi languages were merged into Devanagari script for evaluation, owing to their shared scripts.

and SuryaOCR [40], which are widely used in practical settings.

To further analyze the performance of IndicPhotoOCR, we also introduce the following oracle variants: (i) *IndicPhotoOCR (+Oracle TD)*: Uses ground-truth text detection. (ii) *IndicPhotoOCR (+Oracle TD + Oracle SI)*: Uses ground-truth text detection and script identification. These variants help estimate the upper bound on performance in scenarios where text detection or both text detection and script identification are solved reliably.

Comparison with the aforementioned competitive baselines is presented in Table 8 and Table 9 using WRR/CRR-based metric and Precision-Recall-based metric, respectively. We observe that IndicPhotoOCR consistently outperforms all existing open-source models across all the evaluated

Method	🔒/🔓	Metric	Languages										Avg.
			Bengali*	English	Gujarati	Devanagari*	Kannada	Malayalam	Odia	Punjabi	Tamil	Telugu	
ChatGPT [42]	🔒	P	0.11	0.51	0.15	0.28	0.05	0.07	0.06	0.17	0.23	0.11	0.17
		R	0.09	0.42	0.12	0.23	0.05	0.06	0.06	0.15	0.22	0.1	0.15
		F	0.09	0.44	0.13	0.24	0.05	0.06	0.05	0.15	0.22	0.1	0.15
Google OCR [41]	🔒	P	**0.52**	**0.54**	**0.42**	**0.64**	**0.34**	**0.45**	**0.54**	**0.59**	**0.58**	**0.49**	**0.51**
		R	**0.55**	**0.71**	**0.44**	**0.65**	**0.3**	**0.42**	**0.61**	**0.6**	**0.64**	**0.53**	**0.54**
		F	**0.52**	**0.59**	**0.42**	**0.63**	**0.31**	**0.43**	**0.56**	**0.58**	**0.6**	**0.5**	**0.51**
Tesseract [37]	🔓	P	0.08	0.05	0.05	0.06	0.01	0	0.04	0.08	0	0.03	0.04
		R	0.08	0.11	0.03	0.05	0.02	0.01	0.04	0.06	0.01	0.02	0.04
		F	0.07	0.06	0.03	0.05	0.01	0.01	0.03	0.05	0	0.02	0.03
SuryaOCR [40]	🔓	P	0.27	0.3	0.19	0.29	0.11	0.17	0.23	0.38	0.34	0.1	0.24
		R	0.24	0.36	0.1	0.23	0.09	0.15	0.18	0.24	0.29	0.13	0.20
		F	0.23	0.29	0.12	0.25	0.09	0.15	0.19	0.28	0.29	0.11	0.2
IndicPhotoOCR	🔓	P	0.59	0.55	0.43	0.53	0.41	0.39	0.47	0.56	0.55	0.38	0.49
		R	0.53	0.46	0.4	0.43	0.33	0.38	0.47	0.49	0.52	0.37	0.44
		F	0.55	0.48	0.4	0.46	0.36	0.38	0.47	0.51	0.52	0.37	0.45
IndicPhotoOCR (+oracle TD)	🔓	P	0.72	0.83	0.65	0.66	0.56	0.57	0.65	0.69	0.78	0.54	0.66
		R	0.69	0.75	0.62	0.61	0.5	0.57	0.63	0.68	0.73	0.53	0.63
		F	0.7	0.78	0.62	0.62	0.52	0.57	0.63	0.68	0.75	0.53	0.64
IndicPhotoOCR (+oracle TD+SI)	🔓	P	0.75	0.89	0.71	0.71	0.57	0.6	0.66	0.72	0.8	0.55	0.70
		R	0.75	0.9	0.72	0.71	0.56	0.59	0.66	0.73	0.81	0.56	0.70
		F	0.75	0.89	0.71	0.71	0.56	0.6	0.66	0.72	0.8	0.56	0.70

Table 9: Performance comparison across 10 Indic scripts on the BSTD dataset for **end-to-end scene text recognition**. We report Precision (P), Recall (R), and F1-score (F) for each script and the average performance. The proposed baseline model, IndicPhotoOCR, significantly outperforms available open-source models and achieves competitive performance compared to commercial models. The inclusion of oracle Text Detection (TD) and Script Identification (SI) shows the potential for further improvement. The best scores for **closed-source** and open-source are shown in bold and underlined, respectively. The asterisks (*) on Bengali and Devanagari indicate that Assamese and Bengali languages were merged into Bengali, and Hindi and Marathi languages were merged into Devanagari script for evaluation, owing to their shared scripts.

languages. Remarkably, it also surpasses the popular commercial vision-language model GPT-4, which may not be optimized for Indian scripts. However, our approach still lags behind Google OCR in some cases, highlighting a performance gap that we aim to address in future work. Encouragingly, the variant IndicPhotoOCR (+oracle TD + SI) achieves results that are competitive, infact 30% (WRR) superior to Google OCR on an absolute scale on average, indicating the potential of our pipeline with improved text detection and script identification. In Table 9, on P/R/F metric, IndicPhotoOCR significantly surpasses other open-source baselines. The proposed baseline also not significantly behind the commercial state-of-the-art Google OCR. In fact, when combined with oracle text detection and oracle script identification, our baseline can significantly outperform Google OCR, highlighting its potential for further improvement.

A somewhat surprising observation is the noticeable performance drop for English in the end-to-end setting. Although the proposed baseline achieves a cropped word recognition accuracy of 92%, its end-to-end performance declines sharply. This drop is primarily attributable to script identification and text detection errors introduced in the full pipeline. Notably, this pattern is also reflected in commercial OCR systems such as Google's OCR.

We also performed a comprehensive qualitative analysis of the proposed baseline. A selection of results is shown in Fig. 10. We observe that IndicPhotoOCR demonstrates strong multilingual scene text recognition across diverse real-world settings - from street signs and transportation hubs to institutional boards and cultural heritage sites. The model performs well even on slanted or perspective-distorted text (e.g., directional signs, railway nameplates), as well as on low-resolution or distant text, indicating robustness in challenging visual conditions.

On the BSTD test set, we additionally asked annotators to label each image as easy or hard based on the readability of the scene text. In the easy subset, the proposed IndicPhotoOCR

Fig. 10: A selection of results using IndicPhotoOCR. Detected bounding boxes are shown in green, with the recognized text displayed in red on a white background adjacent to each box. **(Best viewed in color)**.

achieves 60% WRR, whereas in the hard subset the performance drops to 31% across all 10 scripts. Script-wise word recognition rates are presented in Fig. 12. This clear separation between easy and hard cases highlights the sensitivity of current methods to challenging visual conditions and calls for more robust recognition models.

4.5 Error Analysis

The proposed baseline is not flawless and leaves several avenues for improvement. We performed an extensive error analysis of each module of our pipeline (Section 4.4), summarized in Fig. 11, and highlight the main failure modes below.

Text Detection: Common issues in text detection include (i) *false positives* where background edges resemble text, (ii) *missed detections* in low-contrast regions, (iii) *merged detections* when nearby words are grouped together, and (iv) difficulty with *extremely large text*, suggesting the model is tuned more for small or medium size words.

Script Identification: (i) *Misclassifications* occur among visually similar scripts (e.g. Gujarati-Hindi, Bengali-Assamese). (ii) *Multilingual overlap* leads to errors when multiple scripts co-occur, as the model assumes a single script per region. (iii) *Degraded or stylized text* often

Fig. 11: Failure cases in end-to-end scene text processing: The images showcase representative errors observed across the pipeline. (1) Text detection failures from TextBPN++, including false positives, missed detections, merged text, and large text misdetections. (2) Script identification errors from the ViT-based model, highlighting misclassified scripts, multi-script confusion, and challenges with degraded text. (3) Recognition failures from PARSeq, demonstrating incorrect predictions, missing or hallucinated characters, and cross-script misinterpretations. **(Best viewed in color)**.

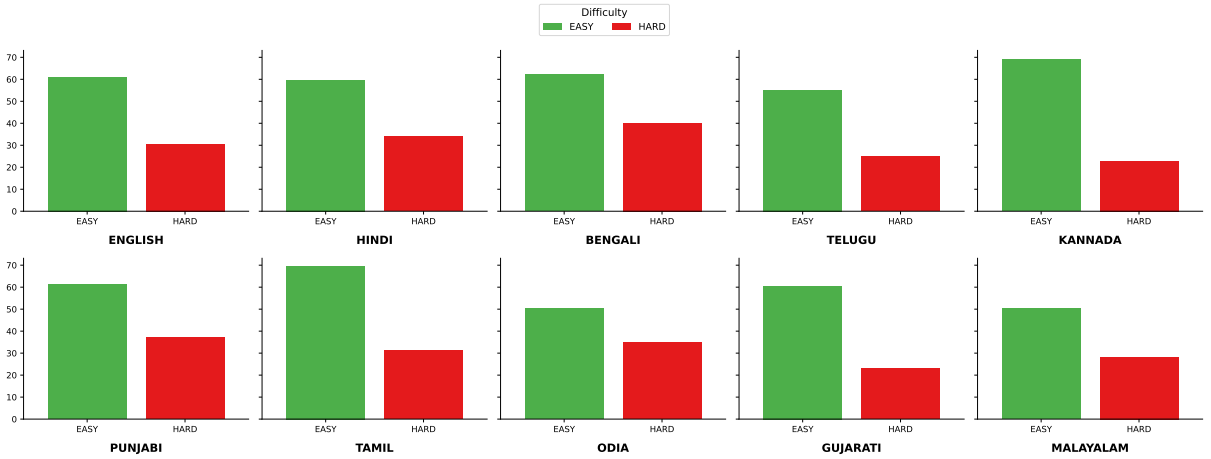

Fig. 12: Average Word Recognition Rate (WRR) based on difficulty level.**(Best viewed in color)**.

confuses the classifier due to missing or distorted characters.

Text Recognition: Errors include (i) *incorrect predictions* in complex scripts like Telugu or

Fig. 13: Recognition errors (words shown in red) from Google Vision, GPT-4, and IndicPhotoOCR in comparison with ground truth annotations on images from BSTD test set. **(Best viewed in color)**.

Tamil, (ii) *hallucinated or missing characters*, and (iii) failures on *eroded/unclear text*.

These findings emphasize the need for cross-module optimization and feedback-driven refinement between detection, script identification, and recognition. In addition, these qualitative analysis, as well as the quantitative analysis in the earlier section, suggest that improving each module in the pipeline can significantly improve overall performance. These observations also motivated a visual performance comparison of IndicPhotoOCR with closed-source applications relative to the ground truth, as shown in Fig. 13.

5 IndicPhotoOCR: Installation and Usage

We open-sourced IndicPhotoOCR, a toolkit for recognizing text in English and 11 Indian languages: Assamese, Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Odia, Punjabi, Tamil, and Telugu.[9] It is designed to handle the unique scripts and complex layouts of Indian languages. By releasing this toolkit publicly, we enable reproducible research, provide a common benchmark for multilingual scene text, and support the development of real-world applications across diverse Indian scripts.

We provide a user-friendly installation and usage platform, allowing our proposed baseline to be easily replicated. We firmly believe that this open-source effort will accelerate future research in Indian language scene text recognition. The installation and usage instructions are provided below.

5.1 Installation

```
$ git clone https://github.com/Bhashini-IITJ/
    IndicPhotoOCR.git
$ cd IndicPhotoOCR
$ chmod +x setup.sh
$ ./setup.sh
```

5.2 Usage

Detection

```
>>> from IndicPhotoOCR.ocr import OCR

# Initialize the OCR system
>>> ocr_system = OCR(verbose=True)

# Define input image path
>>> image_path = "path/to/image.jpg"

# Perform text detection
>>> detections = ocr_system.detect(image_path)

# Save and visualize the detection results
```

[9] Future versions will include Urdu and Meitei.

```
>>> ocr_system.visualize_detection(image_path,
    detections)
# Output image saved
```

Script Identification

```
>>> from IndicPhotoOCR.ocr import OCR

# Initialize the OCR system with automatic
    language identification model
>>> ocr_system = OCR(identifier_lang="auto")
>>> image_path = "path/to/cropped_image.jpg"

# Perform script identification
>>> recognized_lang = ocr_system.identify(
    image_path)

# Output recognized words
>>> print(recognized_lang)
```

Cropped Word Recognition

```
>>> from IndicPhotoOCR.ocr import OCR
>>> ocr_system = OCR(verbose=True)

>>> image_path = "path/to/cropped_image.jpg"

>>> language = "hindi"

# Perform text recognition
>>> recognized_text = ocr_system.recognise(
    image_path, language)

# Output recognized text
>>> print(recognized_text)
```

End-to-End Scene Text Recognition

```
>>> from IndicPhotoOCR.ocr import OCR

# Initialize the OCR system with automatic
    language identification
>>> ocr_system = OCR(identifier_lang="auto")

# Define input image path
>>> image_path = "path/to/image.jpg"

# Perform end-to-end OCR
>>> recognized_words = ocr_system.ocr(image_path
    )

# Output recognized words
>>> print(recognized_words)
```

Detailed documentation for usage is also available here[10].

6 Conclusion and Future Work

In this work, we introduced the Bharat Scene Text Dataset (BSTD), a large-scale benchmark specifically designed to address the gap in Indian language scene text recognition. BSTD comprises 6,582 scene images containing 1,06,478 words across 11 Indian languages and English, making it the most comprehensive dataset of its kind. Through rigorous manual annotation, we ensured high-quality polygon-level bounding boxes and text transcriptions. We also provided extensive benchmarks using state-of-the-art text detection, script identification, cropped word recognition, and end-to-end scene text recognition methods, offering a strong foundation for future research.

Our evaluations highlight the challenges posed by Indian scripts, including complex character structures, diverse vocabulary, and script variations. The results also demonstrate that existing models, primarily designed for English and Latin-based scripts, struggle with Indian languages, emphasizing the need for improved recognition techniques. Despite our extensive efforts, several challenges remain open for exploration in the future. We list them here:

Dataset Expansion: While BSTD provides a strong foundation, future extensions could include more languages (especially Urdu and Meiti) and larger-scale data to improve generalization.

Improved Recognition Models: Developing novel architectures that better capture the complexities of Indian scripts, including joint modeling of script identification and recognition, remains an open research avenue.

Benchmarking and Community Involvement: We encourage the research community to contribute to BSTD by proposing new models, sharing additional annotations, and utilizing the dataset for novel applications such as scene text translation, Indian language scene text aware captioning and question answering, and cross-lingual scene text-aware image retrieval.

We firmly believe that the BSTD and open-source toolkit IndicPhotoOCR will serve as catalyst for future advancements in the Indian language scene-text recognition and associated problems, inspiring researchers to build robust and scalable solutions for multilingual photoOCR.

Acknowledgments

This work is supported by the Ministry of Electronics and Information Technology, Govt. of India, under the NLTM-Bhashini Project.

References

[1] Lucas, S.M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R., Ashida, K., Nagai, H., Okamoto, M., Yamamoto, H., *et al.*: Icdar

[10] https://bhashini-iitj.github.io/IndicPhotoOCR/

2003 robust reading competitions: entries, results, and future directions. International Journal of Document Analysis and Recognition (IJDAR) **7**(2), 105–122 (2005)

[2] Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: 2011 International Conference on Computer Vision, pp. 1457–1464 (2011). IEEE

[3] Mishra, A., Alahari, K., Jawahar, C.: Scene text recognition using higher order language priors. In: BMVC-British Machine Vision Conference (2012). BMVA

[4] Yao, C., Bai, X., Liu, W., Ma, Y., Tu, Z.: Detecting texts of arbitrary orientations in natural images. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1083–1090 (2012). IEEE

[5] Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P.: Icdar 2013 robust reading competition. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1484–1493 (2013). IEEE

[6] Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., *et al.*: Icdar 2015 competition on robust reading. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 1156–1160 (2015)

[7] Risnumawan, A., Shivakumara, P., Chan, C.S., Tan, C.L.: A robust arbitrary text detection system for natural scene images. Expert Systems with Applications **41**(18), 8027–8048 (2014)

[8] Ch'ng, C.K., Chan, C.S.: Total-text: A comprehensive dataset for scene text detection and recognition. In: 14th International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 935–942 (2017). IEEE

[9] Phan, T.Q., Shivakumara, P., Tian, S., Tan, C.L.: Recognizing text with perspective distortion in natural scenes. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 569–576 (2013)

[10] Nayef, N., Patel, Y., Busta, M., Chowdhury, P.N., Karatzas, D., Khlif, W., Matas, J., Pal, U., Burie, J.-C., Liu, C.-l., *et al.*: Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1582–1587 (2019). IEEE

[11] Huang, W., Qiao, Y., Tang, X.: Robust scene text detection with convolution neural network induced mser trees. In: European Conference on Computer Vision, pp. 497–511 (2014). Springer

[12] Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE transactions on pattern analysis and machine intelligence **39**(11), 2298–2304 (2016)

[13] Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. International journal of computer vision **116**(1), 1–20 (2016)

[14] Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: East: an efficient and accurate scene text detector. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5551–5560 (2017)

[15] He, T., Tian, Z., Huang, W., Shen, C., Qiao, Y., Sun, C.: An end-to-end textspotter with explicit alignment and attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5020–5029 (2018)

[16] Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9365–9374 (2019)

[17] Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X.: Aster: An attentional scene text recognizer with flexible rectification. IEEE transactions on pattern analysis and machine intelligence **41**(9), 2035–2048 (2018)

[18] Liu, Y., Chen, H., Shen, C., He, T., Jin, L., Wang, L.: Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9809–9818 (2020)

[19] Liao, M., Wan, Z., Yao, C., Chen, K., Bai, X.: Real-time scene text detection with differentiable binarization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11474–11481 (2020)

[20] Wang, Y., Xie, H., Fang, S., Wang, J., Zhu, S., Zhang, Y.: From two to one: A new scene text recognizer with visual language modeling network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14194–14203 (2021)

[21] Fang, S., Xie, H., Wang, Y., Mao, Z., Zhang, Y.: Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7098–7107 (2021)

[22] Bautista, D., Atienza, R.: Scene text recognition with permuted autoregressive sequence models. In: European Conference on Computer Vision, pp. 178–196 (2022). Springer

[23] Fang, S., Mao, Z., Xie, H., Wang, Y., Yan, C., Zhang, Y.: Abinet++: Autonomous, bidirectional and iterative language modeling for scene text spotting. IEEE transactions on pattern analysis and machine intelligence **45**(6), 7123–7141 (2022)

[24] Zhang, S.-X., Yang, C., Zhu, X., Yin, X.-C.: Arbitrary shape text detection via boundary transformer. IEEE Transactions on Multimedia **26**, 1747–1760 (2023)

[25] Zhao, S., Quan, R., Zhu, L., Yang, Y.: Clip4str: A simple baseline for scene text recognition with pre-trained vision-language model. IEEE transactions on Image Processing (2024)

[26] Pal, U., Roy, P.P., Tripathy, N., Lladós, J.: Multi-oriented bangla and devnagari text recognition. Pattern Recognition **43**(12), 4124–4136 (2010)

[27] Mathew, M., Jain, M., Jawahar, C.: Benchmarking scene text recognition in devanagari, telugu and malayalam. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 7, pp. 42–46 (2017). IEEE

[28] Dey, S., Shivakumara, P., Raghunandan, K., Pal, U., Lu, T., Kumar, G.H., Chan, C.S.: Script independent approach for multi-oriented text detection in scene image. Neurocomputing **242**, 96–112 (2017)

[29] Kakwani, D., Kunchukuttan, A., Golla, S., NC, G., Bhattacharyya, A., Khapra, M.M., Kumar, P.: Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 4948–4961 (2020)

[30] Gunna, S., Saluja, R., Jawahar, C.: Transfer learning for scene text recognition in indian languages. In: International Conference on Document Analysis and Recognition, pp. 182–197 (2021). Springer

[31] Lunia, H., Mondal, A., Jawahar, C.: Indicstr12: a dataset for indic scene text recognition. In: International Conference on Document Analysis and Recognition, pp. 233–250 (2023). Springer

[32] Vijayan, V.P., Chanda, S., Doermann, D., Krishnan, N.C.: Scene text recognition: an indic perspective. International Journal on Document Analysis and Recognition (IJDAR) **28**(1), 31–40 (2025)

[33] Vijayan, V.P., Chanda, S., Doermann, D., Krishnan, N.C.: Scene text recognition:

an indic perspective. International Journal on Document Analysis and Recognition (IJDAR) **28**(1), 31–40 (2025)

[34] Nayef, N., Yin, F., Bizid, I., Choi, H., Feng, Y., Karatzas, D., Luo, Z., Pal, U., Rigaud, C., Chazalon, J., *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), vol. 1, pp. 1454–1459 (2017). IEEE

[35] Lee, C.Y., Baek, Y., Lee, H.: Tedeval: A fair evaluation metric for scene text detectors. In: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), vol. 7, pp. 14–17 (2019). IEEE

[36] Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2315–2324 (2016)

[37] Smith, R.: An overview of the tesseract ocr engine. In: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), vol. 2, pp. 629–633 (2007). IEEE

[38] Cui, C., Sun, T., Lin, M., Gao, T., Zhang, Y., Liu, J., Wang, X., Zhang, Z., Zhou, C., Liu, H., et al.: Paddleocr 3.0 technical report. arXiv preprint arXiv:2507.05595 (2025)

[39] JaidedAI: EasyOCR: Ready-to-use OCR with 80+ supported languages. https://github.com/JaidedAI/EasyOCR. Accessed: 2025-07-26 (2020)

[40] Paruchuri, V., Team, D.: Surya: A lightweight document OCR and analysis toolkit. https://github.com/VikParuchuri/surya. GitHub repository (2025)

[41] Google Cloud: Cloud Vision API: Optical Character Recognition (OCR). https://cloud.google.com/vision/docs/ocr. Accessed: 2025-07-26 (2025)

[42] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)

[43] Ye, M., Zhang, J., Liu, J., Liu, C., Yin, B., Liu, C., Du, B., Tao, D.: Hi-sam: Marrying segment anything model for hierarchical text segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)

[44] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)

[45] Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE transactions on pattern analysis and machine intelligence **39**(11), 2298–2304 (2016)

[46] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., *et al.*: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763 (2021)

[47] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021 (2021)

[48] Aberdam, A., Bensaïd, D., Golts, A., Ganz, R., Nuriel, O., Tichauer, R., Mazor, S., Litman, R.: Clipter: Looking at the bigger picture in scene text recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 21706–21717 (2023)

[49] Karthik, G.: Indian-Scene-Text-Recognition. https://github.com/gokulkarthik/Indian-Scene-Text-Recognition. Accessed: 2025-07-26 (2023)