A Benchmark System for Indian Language Text Recognition

Krishna Tulsyan, Nimisha Srivastava, Ajoy Mondal, and C V Jawahar

Centre for Visual Information Technology,
International Institute of Information Technology, Hyderabad, India
krishna.tulsyan@research.iiit.ac.in and
{nimisha.srivastava,ajoy.mondal,jawahar}@iiit.ac.in

Abstract. The performance various academic and commercial text recognition solutions for many languages world-wide has been satisfactory. Many projects now use the OCR as a reliable module. As of now, Indian languages are far away from this state, which is unfortunate. Beyond many challenges due to script and language, this space is adversely affected by the scattered nature of research, lack of systematic evaluation, and poor resource dissemination. In this work, we aim to design and implement a web-based system that could indirectly address some of these aspects that hinder the development of OCR for Indian languages. We hope that such an attempt will help in (i) providing and establishing a consolidated view of state-of-the-art performances for character and word recognition at one place (ii) sharing resources and practices (iii) establishing standard benchmarks that clearly explain the capabilities and limitations of the recognition methods (iv) bringing research attempts from a wide variety of languages, scripts, and modalities into a common forum. We believe the proposed system will play a critical role in further promoting the research in the Indian language text recognition domain.

Keywords: Indian language · text detection and recognition · ground truth · evaluation platform · online benchmark system.

1 Introduction and Related Work

Text recognition solutions are becoming more and more data driven in recent years [18]. Machine learning algorithms have emerged to be the central component of the OCR systems [41, 42]. This is true in most areas of perception and language processing. The quality of solutions is often measured based on the empirical performance on popular benchmarks. It is observed in the history that: (i) establishing proper benchmarks has brought a community together to solve a specific problem with the objective performance systematically improving with time and with growth in the community; (ii) with the performance of the solution becoming *"satisfactory"*, newer challenges are thrown to the community. This trend has been true in some of the OCR problems, such as scene text recognition. However, Indian language OCR research has not yet adapted to this well-known model of research and development. Beyond technical, there are many social challenges still left out in this space.

There has been a convergence of methods for recognizing text in printed, handwritten, natural scenes. Due to the success of deep learning-based formulations [1, 16, 29], advances in one modality of input (e.g., printed) influence the formulations in other modalities. We believe that this has possibly been the most impactful technical trend that can unite and advance the research and developments in Indian languages. Research groups that worked on OCR alone had a specific focus on a language or script. We observe it the world over, that research groups often work only on one of the modalities, i.e., only on one of the scene text, handwritten or printed. Given that there are more than 20 official languages and hundreds of unofficial languages, the number of research groups that work in this area is clearly deficient.

There has been significant research in developing highly accurate OCR solutions [28, 37]. Most of these techniques are driven by the availability of a large amount of data. Unfortunately, creating standard datasets and sharing them across this community has yet not penetrated. This work is also an attempt to bring data and representations into a common format for future use.

Having dynamic leader boards or performance stats has been a way to keep up-to-date on the status of research, and know the harder challenges to be focused. Many open platforms have emerged in document image analysis and also in general machine learning. Some of the available open platforms related to this domain are EU's catch-all repository[1], Github[2] for code sharing and Kaggle[3] for hosting research related contests. The robust reading competition platform [19–21, 26, 36] has been a driving force behind many of the big challenges in ICDARs. However, RRC platform is too broad for our purpose.

This paper proposes a novel benchmark system for Indian language text recognition. Using this system, we hope to tackle some of the inherent challenges evident in the domain of Indian language text recognition. Though this paper does not propose any algorithm, the proposed system could be very important for solving many of the open problems and furthering the research and development of this domain.

2 Indian Language Text Recognition: Practical Challenges

In recent years, there has been significant research in the domain of text recognition in Indian languages [17]. There have been many attempts to create OCRs for recognition in Indian languages like Bangla [5,31,34], Hindi [3,4,14], Tamil [1,30], and Kannada [2, 37]. As there are many languages and numerous scripts in India, we have several challenges in developing state-of-the-art text recognition platform across all these use cases.

[1] https://www.openaire.eu/faqs

[2] https://github.com/

[3] https://www.kaggle.com/

2.1 Lack of People

India has as many as 23 official languages [7]. Though many of these languages share common linguistic and gramma+++tical structures, their underlying scripts remain very different. Furthermore, the non-standardization of Indian language fonts and their rendering scheme has made the development of a multilingual OCR very challenging.

Moreover, only a limited number of researchers are working on text segmentation and recognition tasks in Indian languages. This small group is not sufficient for exploring text recognition across Indian languages due to the diversity in languages (e.g., Hindi, Bangla, Tamil, Urdu, etc.), modalities (e.g., printed, historical, scene text, etc.), and tasks (e.g., text localization, word recognition, etc.).

2.2 Lack of Data

Researchers report results on their own datasets, which in most cases, are not available publicly. Text recognition of Indian languages is an emerging domain that it only recently gained the much-needed traction. Hence, there is a distinct shortage of standard datasets. Indian language consists of many scripts, among of them, only two scripts of Devanagari [24,35], and Bangla [8,24] have any substantial dataset associated with them. This lack of dataset is a serious concern as it results in sub-par performance in most of the modern machine learning techniques like Neural networks (RNN, CNN) [16], Long short-term Memory (LSTM) [6] and Support Vector Machines (SVM) [10] because most of these modern techniques are heavily data-driven.

The vast scope of this domain further compounds the issue. There exist multiple modalities for each of the languages like scanned documents, born-digital images, natural scene images, and text in videos. Also, most of the datasets in this domain are not available publicly, and those that do, are scattered and are individual attempts. Another significant issue is that there is no central community-driven attempt to track and benchmark the different datasets in this domain.

2.3 Challenges in Evaluation

Most of the modern OCR techniques use two primary evaluation criteria to evaluate text recognition tasks. Character error rate (CER) is a character metric that is based upon the Levenshtein distance [25], which is the minimum number of single-character edit operations (insertions, deletions, and substitutions) required to change the given the word to another. Word error rate (WER) [22] is a word metric that is also based on Levenshtein distance the same as the character metric but at the word level, i.e., a minimum number of single-word operations required to change one text to another.

In the case of Indian language, CER and WER fail to accurately represent and evaluate all aspects of the text recognition method in Indic script.

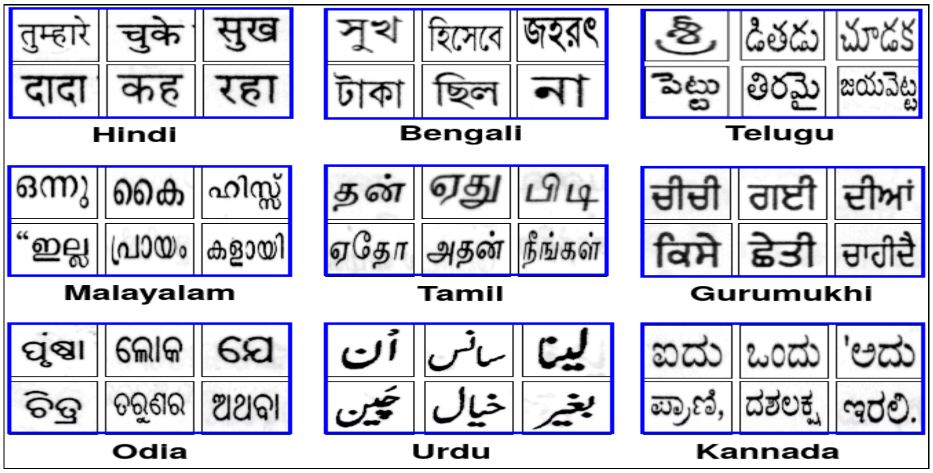


Fig. 1: A visual illustration some Indian scripts.

2.4 Challenges from Language and Scripts

Large variations are observed in the Indic scripts when compared to the Latin scripts, that resulted in many challenges in the development of general text recognition for Indian Languages. There are as many as 23 official languages [7] spoken and written in India and 12 different underlying scripts [11] which results in a significant variation, further increasing the complexity of the task. Furthermore, the lack of any standard Indian language fonts and differences in their rendering schemes has made the development of multilingual OCR very challenging.

Let us consider Bangla script [32] as an example. It is primarily used for Bengali, Assamese, and Manipuri languages. Bangla script contains 11 vowels, whereas the number of consonants is 39. As depicted in Figure 1, Bengali language shows a horizontal line running across the characters and words, which is commonly referred to as *Shirorekha*. In some cases, a set of consonants is followed by another consonant, which results in the formation of a new character that has an orthographic shape and is called a *compound* character.

Due to the presence of compound and *modified* characters (the shape of a consonant is changed when followed by a vowel, hence, a modified character), the number of distinct characters possible in Bangla [40] (roughly 400) is far higher than Latin scripts (62 different characters in English) hence, making text recognition for Bangla script is challenging when compared to English.

3 Contributions

The key factors behind building this benchmarking system are:

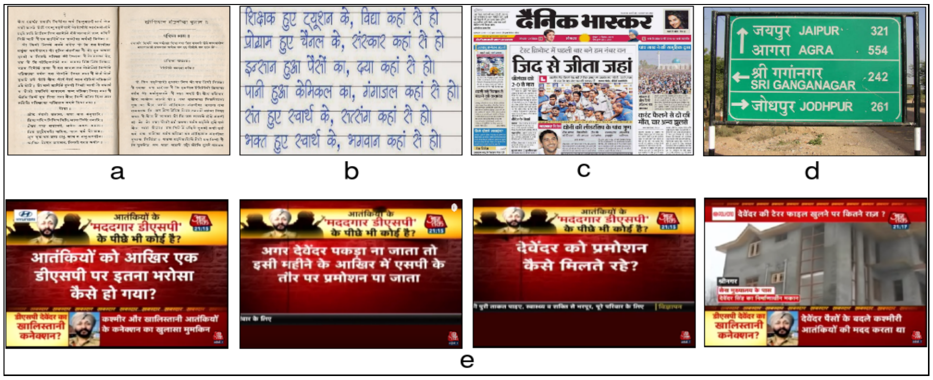


Fig. 2: A visual illustration of different modalities of text in Hindi language — (a) printed text, (b) handwritten text, (c) newspaper, (d) scene text and (e) text in few frames of a video.

(i) Introduce some amount of standardization in the research in Indian language text recognition and bring the research community together on a single platform.

(ii) Given the lack of structured resources, the portal aims to give a platform to the research community to collaborate, share, and standardize datasets to different benchmark tasks, thereby hosting challenges, workshops, and community events.

We hope that this system brings a unification in the research done in silos till now, and the community collaboratively grows. With this background, we have developed a system that is:

(a) Scalability with respect to the task, language, modality, and dataset by offering flexibility to the community to propose new tasks or new evaluation methods for existing tasks.

(b) Verification of over-fitting of the methods to the specific datasets through online testing.

(c) Common platform for logging research outcomes against the datasets/tasks and comparison of results to the state-of-the-art.

Integration of Large Community: Due to large variations in scripts in Indian languages [38] (twenty-three major scripts; Figure 1 illustrates few), the algorithm designed for one does not work for the other. The communities of researchers in a particular language produce results on their datasets, which in most cases, are not available publicly. At the same time only a limited number of datasets [8, 24, 35] are available and various other research communities which are working on the detection, segmentation and recognition tasks in different modalities of documents (e.g., scanned [23, 33], scene text [27, 29], video text [12], camera captured [15], etc.) produce independent results that cannot be benchmarked and compared. Figure 2 visually illustrates the various modalities of Hindi text documents. Our system addresses these challenges by providing a common systematic evaluation and a benchmarking platform to quantify and track the progress in the domain of text detection, segmentation, and recognition from a variety of sources. The sources are crafted for the different tasks and approved by our panel of eminent researchers in Indian Languages, thus providing the desired standardization.

The variations in task, image modality and language specific to the needs of Indian Languages are limited in the existing systems such as such as Kaggle, Robust Reading Competition [19–21, 26, 36], Pascal VOC[4] [13], Cityscapes[5] [9] when compared to our system.

Flexibility in Design: Due to the scale and the diversity of the research, it is becoming increasingly challenging for one owner to develop and maintain the system, introduce new datasets and challenges, organize workshops for all the different languages and scripts. We have designed our system in a way such that the fraternity gets the flexibility to contribute to the evolution of the system. The existing systems for other languages are more rigid in their design. We provide access to the full system or a small designated part of the system to any research community on this field to update/improve the system. The research community can propose to (i) modify existing tasks, (ii) integrate new tasks, evaluation metrics and datasets, (iii) host a new challenge, and (iv)organize a workshop. All the proposals will be reviewed by a panel and included in the system upon approval. This flexibility makes our system scalable as the research community grows and fosters constructive discussions and collaborations.

Online Evaluation: The existing popular web-based evaluation tools such as Kaggle, Robust Reading Competition, Pascal VOC, Cityscapes, and much more deal with only offline evaluation on the submitted results. They have no control over the overfitting[6] of submitted methods on the test datasets. Our system is different from the existing systems since we added the capability to evaluate the results online in real-time.

Since our system provides the test datasets to the users, there is a chance to overfit the submitted methods to the specified dataset. The online evaluation feature is incorporated into our portal to minimize the possibility of overfitting any submitted method on the test dataset. With this feature, we can detect and alert the users, if the results for a particular submitted method overfits the test dataset by maintaining and checking the presented technique against other random sample images that are not part of the offline test dataset. With this facility, any registered user can establish a connection to our server for a specified duration by sending a request to the server. The server sends test images

[4] http://host.robots.ox.ac.uk:8080/

[5] https://www.cityscapes-dataset.com/

[6] Overfitting is a modeling error that occurs when a function is too closely fit a limited set of data points.

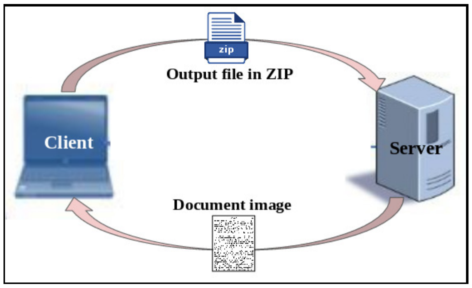


Fig. 3: A visual illustration of online evaluation. The server sets a connection to the client for a duration based on user requests. The server sends test images one at a time to the user and receives the corresponding result.

to the user, one at a time, and receives the result for the individual image after running the method on that image based on the user's request. Figure 3 is a brief illustration of the working of the online testing facility.

After the process is done, i.e., all the images from online test dataset are processed and evaluated using corresponding evaluation measure for the task, the system attempts to detect the overfitting of the submitted method by comparing the evaluation results for online and offline datasets. As specified, the online dataset is a super-set of an offline dataset with some random variations of the same type added.

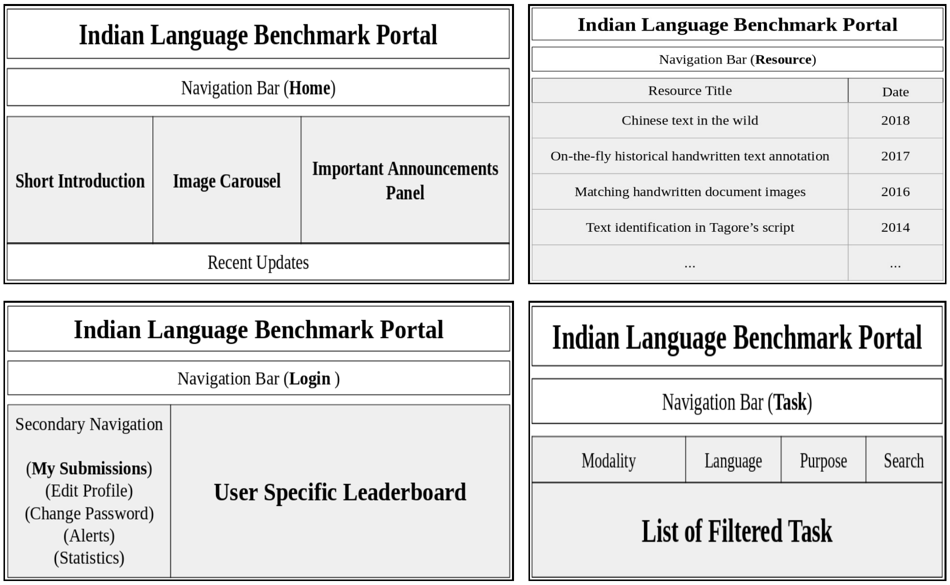
The system calculates the probability of overfitting by measuring the response of the submitted method on the random images of the online dataset compared to the result of the specified technique on the offline dataset. Ideally, if the method is viable (not overfitting), then the difference between the result accuracy of the offline and online datasets should be minimal. On the other hand, this difference is more significant than a specified threshold. For example, in the offline dataset, the accuracy (mAP) of the method is reported to be 94%. In the case of an online dataset, the same approach has the accuracy (mAP) dropped to 62%. Then it is clear that this method is overfitting on the offline dataset.

Other than overfitting, this system can also be used as an intermediary, that gathers and compiles various stats related to processing, submission, and evaluation of the methods on a particular dataset. For example, with this online testing system, we can gather the submission method's time to process each test image. This data can be interpreted by the user to do some analysis, which offers further insights into the workings and efficiency of the submitted method.

4 Design, Functionality and Implementation

4.1 Design

The benchmark system portal has the following pages to navigate to — **Home**, **Login**, **Resource**, **Workshop/Challenge** and **Task**. Figure 4 highlights the functionalities of these heads. We subsequently explain the details available, capabilities and workflow of the various components in the system through these heads.



Resource Title	Date
Chinese text in the wild	2018
On-the-fly historical handwritten text annotation	2017
Matching handwritten document images	2016
Text identification in Tagore's script	2014
...	...

Fig. 4: Visual illustration of the functionalities of our benchmark system.

Home: This is the launch page for the benchmark system that provides consolidated information and statistics around the key indicators and trends (most used datasets, the highest number of submissions, etc.). It also highlights news flashes, information on upcoming events, challenges, and any other essential information that the research community may need to know. The underlying idea is to eventually make the landing page as the go-to page for the community to get information on all the research related trends, any recent breakthroughs, and key information in the space.

Login: Users need to register themselves on the benchmark system. After logging into the system, the users can download datasets, test their algorithms, and participate in the workshops/challenges. The user profile page gives all the information on the activities done by the user on the system like datasets downloads, submissions made, the relative position of the users submission on the leaderboard, etc.

Resource: Here, we refer to most of the available literature on segmentation, detection, and recognition tasks in Indian languages. The users can search the extensive collection of resources based on keywords and filters. Going forward, we intend to crowd-source all the available literature or links to the online support for the same in the Indian Language research space.

Workshop/Challenge: This page hosts all the information on workshops and challenges that the community plans to host. Registered users can participate in these workshops and challenges and present their work.

Task: This is the key module of our system, which the registered users will most frequently visit to check for available tasks/datasets and submit their methods. In the subsections to follow, we explain **Task** module and the functions available in detail.

4.2 Functionality

Resource Module: This serves as a warehouse of all resources(mostly research papers) for text segmentation, detection, and recognition in Indic languages. We store the title, abstract, authors, publication date, URL and BibTex for the paper. An anonymous user can search/browse through our database of resources. Still, only registered users can submit a request (with the details of the resource) to include any resource that is missing from the collection. Once the administrator approves the request, the requested resource is added to the collection.

Task Module: This is one of the most important modules of the system. A task is defined as a combination of **Language** (e.g., Hindi, Bengali, Telugu), **Modality** (e.g., camera captured documents, scanned printed books) and **Purpose** (text localization, text recognition) and represents a problem for which the user designs an appropriate method to achieve the goal. Each task in the system represents a unique research problem (e.g., word detection in Hindi scanned book, line recognition in Tamil scene images). Every task may have one or more associated test datasets available.

The benchmark system supports either multiple versions of the same dataset or an entirely new dataset for each task. Each submission in the task is evaluated against performance measures indicated for the particular task, and the result obtained is stored in the database. The users have the option to request new evaluation metrics or to make changes to the existing ones.

Submission Module: The user can use the associated dataset as a test/benchmarking set and submit the obtained results against any task. A user can have multiple submissions against a single task. Each submission is identified by a unique identifier (ID). Every time a user uploads his/her results for a *submission*, the system evaluates the results against the ground truth for the dataset as per the evaluation criterion defined for the task. The evaluation results are then displayed at an appropriate position in the leaderboard. The user can choose to keep his submission either **public** or **private**. In case, submission is marked as **public**. The submission results are available for visitors to see on the leaderboard. In case of submission being marked as **private**, the results are displayed only for this particular user.

To check for overfitting of an algorithm developed by the user, **Online** evaluation has been proposed. The Online verification system allows the user to verify his/her algorithm dynamically by calling the interfacing function (API) to get one test image at a time and submit the output generated immediately back into the system. Once the entire dataset is iterated upon, and the system receives the output. The results are evaluated as per the metrics defined for the task and displayed at an appropriate position in the leaderboard. It is noted that additionally, the test dataset in the **online** mode is a super-set of the dataset in the **offline** mode for the same task. Figure 3 displays the **online** submission process.

The user gets to see the relative position of his result in the public leaderboard on his/her profile page under the **My Submission** tab. The user can also see the statistics of his/her submissions under the **Stats** tab on his profile page.

Evaluation of Results: As the number and variations of the tasks hosted on the system continue to grow, we faced a different challenge in managing and organizing the evaluation metrics for each task. We aim to design the system so that each task can have multiple evaluation measures and are easily configured/modified by the administrator. The algorithm developed for a particular task is evaluated by comparing the results submitted with the corresponding ground truths stored in the system. For each task, multiple standard measures are considered and may differ from one task to the other. The user can propose other evaluation measures that may be relevant to the task by writing to the system administrator. The proposal goes to a review board before getting added to the task.

Leaderboard: It shows the relative performance of the method for a particular task and evaluation criteria against a dataset. Every task has its separate leaderboard, where ordered results are shown for every evaluation criteria, thereby ranking the methods based on their performance. Users can reorder the displayed ordered list by selecting from the evaluation measures. The user can choose to see the leaderboard corresponding to the **online** or **offline** way of evaluation (as described in the submission module) or for the different dataset versions available for a task.

4.3 System Implementation

The benchmark system uses Django[7]/Python for back-end functionality and HTML/CSS for front-end. The system's dynamic components include resource search, Task display page, Leaderboard, Online/Offline Submissions, and User Authentication. The system is planned in such a way that every individual module of the system can act as an optional app/plugin separately. The system is designed with a standard model, view, and controller (MVC) architecture in mind. There are five major models for the project as follows:

[7] https://www.djangoproject.com/

- **User Model:**, All the corresponding details of the registered users, are stored in this model (e.g., being full name, username, email, date of birth, affiliation, etc.).
- **Task Model:** This is a central model that stores the details of each defined task. Some of the tasks' attributes include the name of the task, description, modality, language, the purpose of the task, etc. The task model is connected to the dataset model by a one-to-many relationship.
- **Dataset Model:** The following model is used to store the details of the available test datasets. Each dataset is inexplicably linked to a single corresponding task model. The model also saves the ground truth zip/file for the dataset.
- **Submission Model:** This model is used to store the details of the user's submissions.
- **Resource Model:** This model is used to store the metadata for the essential resources (like articles published in various journals and conferences).

Our designed system is highly modular and consists of a collection of various tasks. Each of these tasks is independent of others and can be added or removed by the supervisor of the system independently or based on the user or an administration panel's suggestions without affecting the rest of the system. The supervisor or admin has the freedom to add a new task/dataset or modify/delete the existing tasks/datasets using the admin portal with the approval from an advisory committee. It will comprise of eminent researchers from the fraternity. The administration panel is only accessible to the supervisor and a selected number of staff accounts that correspond to the administration panel.

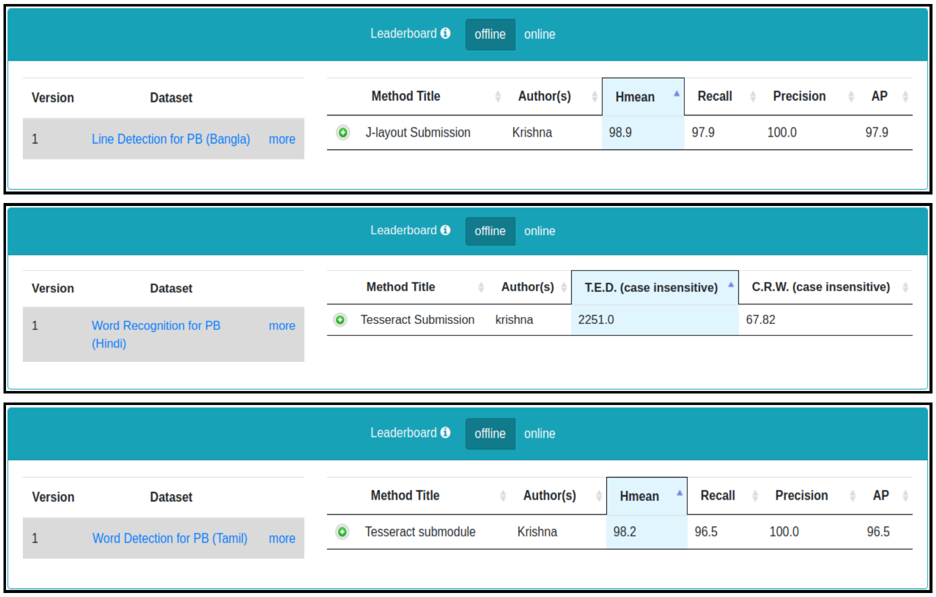
Workflow: Any anonymous user (without registration) can only view the latest news, upcoming events, overall statistics flushed on the Homepage, resource search page, and public leaderboard corresponding to a Task. They are not allowed to download the test datasets for any task. On the other hand, registered users have access to a lot more features. Currently, the system supports both username and email-based registration, and the user alone can login into the website using username.

- Request addition of new resources by contacting the supervisor or a staff member.
- Download the dataset of any available task.
- Upload result corresponds to the dataset of a particular task, in case of offline submission.
- Upload result of a particular task through API, in case of online submission.
- View the submitted results on either public or private leaderboard.
- View the submitted results on his/her profile page.
- Edit and view the profile associated with the user.
- Modify/delete the submitted results.

5 Current Status

Task/Dataset results: Currently, the system has the following 5 different tasks/datasets against which any user can upload the submissions.

(i) **Line detection for Bangla printed books**: The objective of this task is to localize individual text lines presents on a page. We use the Tesseract [39] line detector as a baseline. Quantitative scores of the results obtained using this method are: (Hmean=98.9%, Recall=97.9%, Precision=100%, AP=97.9%). Figure 5 (First Row) displays the evaluation result for this task on our leaderboard.



Version	Dataset	
1	Line Detection for PB (Bangla)	more

Method Title	Author(s)	Hmean	Recall	Precision	AP
J-layout Submission	Krishna	98.9	97.9	100.0	97.9

Version	Dataset	
1	Word Recognition for PB (Hindi)	more

Method Title	Author(s)	T.E.D. (case insensitive)	C.R.W. (case insensitive)
Tesseract Submission	krishna	2251.0	67.82

Version	Dataset	
1	Word Detection for PB (Tamil)	more

Method Title	Author(s)	Hmean	Recall	Precision	AP
Tesseract submodule	Krishna	98.2	96.5	100.0	96.5

Fig. 5: A visual illustration of Leaderboard containing results of various tasks in Indian languages. **First Row:** result of text line detection for Bangla printed books. **Second Row:** result of word recognition for Hindi printed books. **Third Row:** result of word detection for Tamil printed books.

(ii) **Word recognition for Hindi printed books**: Given a set of cropped images of Hindi words, the objective of this task is to recognize those words correctly. We use the Tesseract OCR [39] as a baseline for this task. The evaluation results for this method are: (Correctly Recognized Words = 67.82%, Total Edit distance = 2251). Figure 5 (Second Row) displays the evaluation result for this task on our leaderboard.

(iii) **Word detection for Tamil printed books**: The objective of this task is to localize each word on a page. Again we use a sub-module of Tesseract

OCR [39] as a baseline to evaluate this task. The quantitative scores of results obtained using this method are: (Hmean=98.2%, Recall=96.5%, Precision=100%, AP=96.5%). Figure 5 (Third Row) displays the evaluation result for this task on our leaderboard.

(iv) **Block detection for Telugu printed books**: The objective of this task is to predict the bounding box of each block present on a page.

(v) **End-to-End recognition for Hindi printed books**: The objective of the task is to localize and recognize the words present in a page of Hindi printed books.

The evaluation measures available for the detection tasks are Recall, Precision, Hmean (F-score), and Average Precision (AP). In the case of recognition task, two evaluation measures are used — (i) Total Edit Distance and (ii) Correctly Recognized Words. Both of these measures are evaluated in a case insensitive manner. For the end-to-end recognition task, three evaluation measures (as a combination of detection and recognition measures) are used - (i) Average Precision (AP), (ii) Total Edit Distance, and (iii) Correctly Recognized Words.

6 Conclusions and Future Work

This paper presents a Benchmark System for Indian Language Text Recognition through which the researchers can benchmark their method for detection, recognition, and end-to-end recognition tasks in document images for Indian languages. We believe that this system will bring all the researchers working in the Indian language domain into a common space.

We expect that this system will foster a sense of cooperation and healthy competition in the field by allowing the users to compare their results against a well-known standard. Our target is to eventually produce meaningful insights on the state-of-the-art based on the statistics collected over time. We are hoping to organize periodic workshops to bring the community on a common platform to share ideas and collaborate constructively. We are also continuously working towards improving and adding exciting new functionalities in the system. We target to introduce challenges to the system.

References

1. Achanta, R., Hastie, T.J.: Telugu OCR framework using deep learning. ArXiv (2015)
2. Ashwin, T.V., Sastry, P.S.: A font and size-independent OCR system for printed Kannada documents using Support Vector Machines. Sadhana (2002)
3. Bansal, V., Sinha, R.: A complete OCR for printed Hindi text in Devanagari script. In: ICDAR (2001)
4. Bansal, V., Sinha, R.M.K.: A complete OCR for printed Hindi text in Devanagari script. ICDAR (2001)
5. Basu, S., Das, N., Sarkar, R., Kundu, M., Nasipuri, M., Basu, D.K.: Handwritten Bangla alphabet recognition using an MLP based classifier. CoRR (2012)

6. Breuel, T.M.: High performance text recognition using a hybrid convolutional-LSTM implementation. In: ICDAR (2017)
7. Chandramouli, C., General, R.: Census of India 2011. Provisional Population Totals. New Delhi: Government of India (2011)
8. Chaudhuri, B.B.: A complete handwritten numeral database of Bangla a major Indic script. In: IWFHR (2006)
9. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
10. Das, N., Das, B., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M.: Handwritten Bangla basic and compound character recognition using MLP and SVM classifier. ArXiv (2010)
11. Datta, A.K.: A generalized formal approach for description and analysis of major Indian scripts. IETE Journal of Research (1984)
12. Dutta, K., Mathew, M., Krishnan, P., Jawahar, C.V.: Localizing and recognizing text in lecture videos. In: ICFHR (2018)
13. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The Pascal visual object classes (voc) challenge. IJCV (2010)
14. Gaur, A., Yadav, S.: Handwritten Hindi character recognition using k-means clustering and SVM. ISETTLIS (2015)
15. Gupta, V., G N, R., Ramakrishnan, K.: Automatic Kannada text extraction from camera captured images. In: MCDES, IISc Centenary Conference (2008)
16. Jain, M., Mathew, M., Jawahar, C.V.: Unconstrained OCR for Urdu using deep CNN-RNN hybrid networks. In: ACPR (2017)
17. Jomy, J., Pramod, K.V., Kannan, B.: Handwritten character recognition of south Indian scripts: A review. CoRR (2011)
18. Jordan, M.I., Mitchell, T.M.: Machine learning: Trends, perspectives, and prospects. Science (2015)
19. Karatzas, D., Gómez, L., Nicolaou, A., Rusiñol, M.: The robust reading competition annotation and evaluation platform. In: DAS (2018)
20. 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: ICDAR
21. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i 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: ICDAR (2013)
22. Klakow, D., Peters, J.: Testing the correlation of word error rate and perplexity. Speech Communication (2002)
23. Krishnan, P., Sankaran, N., Singh, A.K., Jawahar, C.: Towards a robust OCR system for Indic scripts. In: DAS (2014)
24. Kumar, A., Jawahar, C.V.: Content-level annotation of large collection of printed document images. In: ICDAR (2007)
25. Levenshtein, V.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady (1966)
26. Lucas, S.M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R.: ICDAR 2003 robust reading competitions. In: ICDAR (2003)
27. Mathew, M., Jain, M., Jawahar, C.V.: Benchmarking scene text recognition in Devanagari, Telugu and Malayalam (2017)
28. Mathew, M., Singh, A.K., Jawahar, C.V.: Multilingual OCR for Indic scripts. DAS (2016)

29. Nag, S., Ganguly, P.K., Roy, S., Jha, S., Bose, K., Jha, A., Dasgupta, K.: Offline extraction of Indic regional language from natural scene image using text segmentation and deep convolutional sequence. ArXiv (2018)
30. Negi, A., Bhagvati, C., Krishna, B.: An OCR system for Telugu. In: ICDAR (2001)
31. Omee, F.Y., Himel, S.S., Bikas, M.A.N.: A complete workflow for development of Bangla OCR. CoRR (2012)
32. Pal, U., Chaudhuri, B.: Indian script character recognition: a survey. Pattern Recognition (2004)
33. Sankaran, N., Jawahar, C.V.: Recognition of printed Devanagari text using BLSTM neural network (2012)
34. Sarkar, R., Das, N., Basu, S., Kundu, M., Nasipuri, M., Basu, D.K.: Word level script identification from Bangla and Devanagri handwritten texts mixed with Roman script. CoRR (2010)
35. Setlur, S., Kompalli, S., Ramanaprasad, V., Govindaraju, V.: Creation of data resources and design of an evaluation test bed for Devanagari script recognition. In: WPDS (2003)
36. Shahab, A., Shafait, F., Dengel, A.: ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. In: ICDAR (2011)
37. Sheshadri, K., Ambekar, P.K.T., Prasad, D.P., Kumar, R.P.: An OCR system for printed Kannada using k-means clustering. In: ICIT (2010)
38. Sinha, R.M.K.: A journey from indian scripts processing to Indian language processing. IEEE Annals of the History of Computing (2009)
39. Smith, R.: An overview of the Tesseract OCR engine. In: ICDAR (2007)
40. Stiehl, U.: Sanskrit-kompendium. Economica Verlag (2002)
41. Ye, Q., Doermann, D.S.: Text detection and recognition in imagery: A survey. IEEE Trans. on Pattern Analysis and Machine Intelligence (2015)
42. Zhu, Y., Yao, C., Bai, X.: Scene text detection and recognition: recent advances and future trends. Frontiers of Computer Science (2015)