

Experiences of Integration and Performance Testing of Multilingual OCR for Printed Indian Scripts

Deepak Arya
CDAC, Noida

deepakarya@cdacnoida.in

Tushar Patnaik
CDAC Noida

tusharpatnaik@cdacnoida.in

Santanu Chaudhury
IIT Delhi

santanuc@ee.iitd.ac.in

C V Jawahar
IIIT,Hyderabad

jawahar@iiit.ac.in

B.B.Chaudhuri
ISI,Kolkata

bbc@isical.ac.in

A.G.Ramakrishna
IISC,Bangalore

ramkiag@ee.iisc.ernet.in

Chakravorty Bhagvati
University of Hyderabad

chakcs@uohyd.ernet.in

G. S. Lehal
Punjabi University, Patiala

gslehal@gmail.com

ABSTRACT

This paper presents integration and testing scheme for managing a large Multilingual OCR Project. The project is an attempt to implement an integrated platform for OCR of different Indian languages. Software engineering, workflow management and testing processes have been discussed in this paper. The OCR has now been experimentally deployed for some specific applications and currently is being enhanced for handling the space and time constraints, achieving higher recognition accuracies and adding new functionalities.

Keywords

Multilingual, Feature Extraction, Hierarchical classification, Template matching, Shape analysis

1. INTRODUCTION

Optical Character Recognition (OCR) is the electronic translation of images of printed text (usually captured by a scanner) into machine-editable text. OCR is being studied for about sixty years, but Indian script OCR is being seriously studied since only past decade. OCR systems for English are readily available but OCR's for different Indian languages have still not reached their full

maturity [6]. There have been many attempts in development of OCRs for Indian Scripts like Devanagari, Malayalam[10], Telugu, Tamil[7], Bangla[14], Gurmukhi[15] and Kannada[8]. These individual OCRs do not take care of various script-independent document image analysis issues. In this paper, we describe an integrated OCR which exploits common characteristics and common solutions across scripts to generate a robust system. This OCR is the outcome of a consortium mode project executed at various institutes for the development of printed Indian script OCR systems, sponsored by Department of Information Technology, Govt. of India.

Each center had the liberty to choose the approaches of preprocessing including symbol and parts segmentation, feature selection, Choice of Recognition engine and combining the outputs into words and sentences. However, for uniformity in the overall system an integrated platform was agreed upon.

Key challenges involved in the development of the integrated OCR platform were the following:

- Developing specification scheme for each functional module so that modules can be independently developed, tested and subsequently integrated, thereby minimizing coupling between modules.
- Maintaining the coding standards decided upon during the initial stages of the project (C++, Fedora core6 compatible, Doxygen, etc).
- Representation scheme for rendering and editing the electronic version of the documents in different scripts.
- Developing, testing and performance evaluation strategies.

The paper is organized into the following sections: Section 2 gives the overall architecture of the Multilingual OCR system. Section 3 talks about the challenges faced during system integration and testing. Section 4 presents the strategy for performance evaluation of the OCR system as a whole and evaluating the individual modules [2]. Section 5 presents OCR Performance Result.

Key challenges involved in development of OCR System

Bangla and Devanagari are among the most symbol-rich since originally about 2000 shapes need to be recognized for a complete OCR system[14]. However, if the zone above the headline (shirorekha) as well as that below the baseline are separated and the symbols and parts of symbols are segmented and recognized, then the number of classes comes down to about 500. This approach was adopted in the current OCR systems.

There is a trade-off of OCR accuracy because of the above step. Reduction of classes from 2000 to 500 resulted in higher symbol recognition accuracy. On the other hand, segmentation of the upper zone and lower zone as well as combining the output of the classifiers into character and words resulted in additional errors. We have noted that if this error can be substantially reduced, the overall OCR accuracy could be substantially improved. It is more difficult to do this for old letterpress printing and that is the main reason of higher error rates of Devanagari OCR.

Devanagari and Bangla OCR system was a two stage. In the first stage we clubbed similar characters into groups and classified the test character into one of the groups. At the second stage we classified into characters belonging to the group. In both cases we sub-divided the character bounding box into 5 x 5 windows and calculated the cumulative value of moves in four directions (Up-down, Left-right, Right slant at 45 degree, Left slant at 45 degree) while traversal of the border. Thus we got 5 x 5 x 4 = 100 dimensional features in both stages

We initially worked on minimum distance and K-NN classifier. Later on, it was noted that SVM classifier yielded more consistent and accurate results. For the mid-zone symbols, it was a two stage classifier. In the first stage, several characters having high shape similarity and are highly confusable among them according to single character SVM, were put in a single group. In this way, several groups were formed. SVM classifiers for such groups were trained using prototypes. Then for the second stage, another set of SVM classifiers were designed for the symbols of the individual groups. Since the symbols of the upper zone and lower zone of the text line are few in number, single stage classifier system was designed for each zone.

Combination of the recognition results needed knowledge of the script grammar. We had to discover the rules used for combining to form the Ortho-syllables (Akshara) using the symbols and part of symbols in the upper, middle and lower zone. Making it exhaustive is still a problem and the erroneous and spurious outputs of three zones sometimes created difficulty in word formation.

However, the systems were not too bad, at least for Bangla OCR, though we need improvements for Devanagari, which we plan to take in the second phase of the project

Tamil is one of the most ancient languages and has recently been recognized as a classical language by Government of India. Tamil has 18 consonants and 12 vowels[7]. In addition, there are 5 more graphemes commonly used to represent the consonants borrowed from Sanskrit. There is another character, called /Aytam/. The current version of the Tamil OCR deals with about 200 classes, including all the punctuation marks, special symbols and Indo-Arabic numerals. Based on the high performance of the Tamil OCR, it is being regularly used by Worth Trust, a not-for-profit organization in Chennai to convert printed Tamil books to Braille books. Over the past one year period, about 150 school, college and other books, involving around 25,000 pages have been converted to Braille books, which are already being used by about 100 persons with visual disability. The OCR has been found to work with any arbitrary font, without any major degradation of performance.

The main challenges of segmentation of merged and broken characters have already been handled to quite an extent, and thus the OCR has delivered an average performance of about 93% in testing about 1500 pages. It is to be noted that this performance is based on raw recognition, without incorporating any language model or dictionary based spell correction. Thus, the potential exists for further good improvement of the recognition accuracy.

The Malayalam writing system is mostly syllabic. The predominant orthographic unit is a vowel ending syllable with the canonical structure (C)V. The obligatory V represents a short or long vowel. The optional C represents one or more consonants Except in a few instances the system follows the principles of phonology and mostly corresponds to the pronunciation Each consonant letter represents a single consonant sound. There are 56 letters in Malayalam, 15 vowels and 36 consonants in addition to the many conjugated and miscellaneous letters[10]. The conjugated letters are combinations of two consonants, but they are written distinctly.

Script Change: By the arrival of modern word-processors, which can generate any complex shape, most of the old lipi characters again came into picture. Also, among the word processors and fonts, there is no standardization followed. Nowadays, a mixture of old and new lipi characters are used by different word-processors[11].

Similar Characters: There are a set of characters which look similar to each other. The variation between these characters are so small that, even human reads the text usually only by its context[10].

ക ക ഹ ഹ ഹ ഹ ഹ ഹ ഹ ഹ ഹ ഹ

Glyph Variation: As the font or style changes, the glyph of a character also changes considerably, which makes the recognition difficult.

ര ര ര ര ര
സ സ സ സ സ

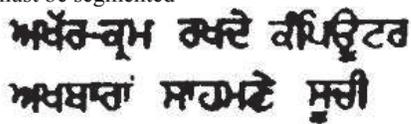
Malayalam character recognition engine uses an SVM classifier inside. Pairwise SVM classifiers are arranged in a Directed Acyclic Graph (DAG) architecture. At present all the pairwise classifiers are linear. All the pairwise classifiers use the same feature description. Simple statistical features like PCA & random projection give satisfactory results. However for wider support (for multifont system) HOG (Histogram Of Gradient) features are getting tried. Initial results are promising.

Character segmentation is based on connected component analysis. These characters/symbols are recognized and the class labels are converted to UNICODE with the help of a loop up table. Additional language specific clues are used at this stage. Initial experiments with sub symbol language models show that a post processing can significantly improve the recognition results.

Gurmukhi syllabary initially consisted of thirty two consonants, three vowel bearers, ten vowel modifiers (including mukt having no sign) and three auxiliary signs. Later on, six more consonants have been added to this script. These six consonants are multi component characters that can be decomposed into isolated parts. Besides these, some characters modify the consonants once they are appended just below them. These are called half characters or subjoined characters.

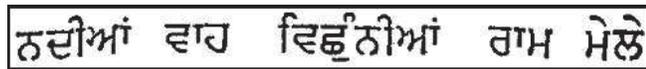
Touching Characters[15]

This is the most commonly found degradation in printed Gurmukhi script. In this category of degraded text, two neighboring characters touch each other. The important issue involved in recognition of the touching characters is to segment them correctly, i.e., identifying the position at which the touching pair of characters must be segmented



Multiple Skew in documents

Another typical problem found in old printed Gurmukhi text is existence of multiple skew on same page. Each word or line could be skewed differently, which calls for development of skew detection and correction algorithms at global and local level.



Kannada script is distinctly different from Devanagari related scripts such as Bangla and Gujarati, as well as the classical Dravidian script of Tamil[8]. Modern kannada script has 48 characters, called varnamale. Consonants are divided into grouped consonants and ungrouped consonants. There are 14 vowels 34 consonants and 10 numerals. Vowels along with consonants constitute basic character. Vowel modifiers can appear to the right on the top or at the bottom of a base consonant. In addition, consonant clusters in Kannada have a two-dimensional structure[8], as shown in Table 1.

Table 1. Some examples of CCV and CCCV

CCV Combinations	ಕೃ ಕ್ಕಾ ಕ್ಯಾ ಕ್ಕಾ ಣ್ಣಿ
CCCV Combinations	ಸ್ತ್ರ ಸ್ತ್ರಳ ಸ್ತ್ರಳ

Combinations in Kannada.

The Kannada OCR [9] handles all the above basic and compound characters, the Kannada numerals and most other symbols that can be keyed in from a QWERTY keyboard. Thus, there are totally about 300 classes to be recognized. Karhunen Leuve transform is used to extract features from the normalized images of the segmented components. SVM is used as the classifier. The characters, which have multiple distinct connected components that appear one above the other are segmented as a single unit, by a judicious combination of connected component analysis and vertical projection. Thus, the component segmentation accuracy is very high, except when there are merged or broken characters. Thus, the real challenge in developing a good Kannada (or for that matter, any other script) OCR is in obtaining a high quality segmentation of the primitives.

Telugu is a phonetic language with characters roughly representing spoken sounds (often syllables). Telugu script contains rounded characters of complex shapes with no vertical lines there are 16 vowels (12), 36 consonants (35) and 2 vowel Modifiers.

Complex shapes very localized structures (e.g., **ಫ ಭ ರ ದ**)

Subset/Superset relationships (e.g., **ವ ಮ ಋ ಋ**)

Visually similar and confusing sets (e.g., **ಲ ನ ನ ಪ ಟ ಲು ಅ ಆ ನು ಮ**)

Vowel modifiers connect to consonants and consonant conjuncts modifying their shapes.

2. ARCHITECTURE OF MULTILINGUAL OCR SYSTEM

The user provides input to the OCR system either a scanned document image or selects an image from database. The user has control over each module of the system to change the parameters and view the effects. Each input image undergoes few of the preprocessing steps for making it fit for recognition engine.

Pre-processing routines available are:

- Skew correction
- Noise cleaning
- Binarization
- Orientation detection
- Block segmentation
- Text non-text separation.

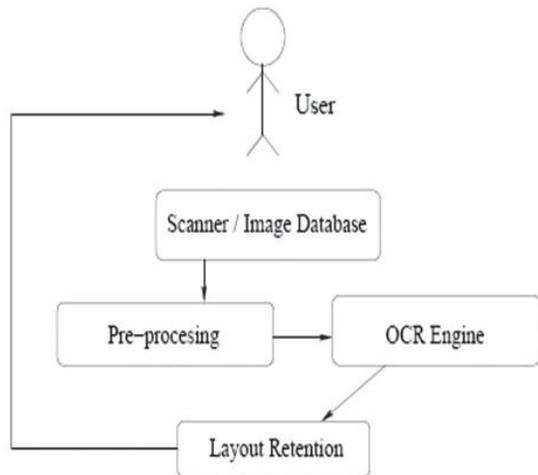


Figure 1: Overall Architecture of Multilingual OCR System

The above routines listed come under the category of script-independent processes. The segmentation routine identifies each segmented block/document image component as text, picture or graphics at the coarser level. Further a semantic label is also attached to each text block as paragraph, column, heading, section, sub-heading etc.

After pre-processing the image is passed to the recognition engine. Here each text block is further segmented to get Line and Word boundaries, followed by character and symbol level segmentation. The processes involved in recognition engine are script-dependent. This engine is capable of recognizing text images of Devanagari, Bangla, Tamil, Kannada, Telugu, Malayalam, Gurumukhi scripts.

The last module is layout retention, which involves rendering of the recognized text and presenting it in a editable format maintaining the layout structure for the end-user. This module is also capable of representing the electronic document in various formats (odt, doc, html etc.).

XML has been used as architecture specification language and enables handling huge amount of data in such large projects [1]. XML based Input/Output interface has been adopted for smooth interaction between all the modules. All modules developed as part of this project are expected to be consistent with this extensible specification of the architectural model.

3. SYSTEM INTEGRATION AND TESTING

Earlier OCR's for each Indian language were available as separate stand-alone package. This is the first complete software package where OCR's for seven Indian Scripts along with different pre-processing and post-processing algorithms are integrated together. The major challenge faced in integrating each of the modules was that all the modules were developed by different people having different software development practices. Below we shall discuss the challenges faced and the remedies for them. In order to handle the modules coming from different members, a website was created by CDAC, Noida coupled with SVN. SVN facilitated updation of repositories. Log-in IDs were provided to each member for uploading & downloading modules. The website works on the concepts of black board architecture for sharing information among the other consortia members. The codes coming from

different consortia members were checked against a set of code acceptance parameters listed below:

- Codes submitted should be strictly in C++ and Fedora 6 compatible.
- Use of OpenCV library only.
- It should support all image formats (PGM, TIFF, BMP and JPG).
- Namespace should be used.
- It should follow the Input/Output XML scheme specified for the project [1].
- It should be Doxygen compatible.
- It should follow the specified directory structure.
- It should have a Makefile and Readme.

Each accepted module is tested and re-engineered to handle the following:

- Taking care of memory leaks.
- Exception handling wherever required.
- Avoid intermixing of new and malloc for memory allocation.
- Optimization with respect to speed and memory size, without tampering the logical meaning of the module.

The project undergoes the following testing phases:

- Unit testing
- Integration testing
- System testing.

Unit testing was done for testing the functionality of individual modules using the dataset provided by the respective member and test dataset created by CDAC, Noida. The modules are tested against specified parameters and the identified bugs are reported to respective consortia member. The modified codes received after removing bugs are re-tested. Once the code is accepted for integration shared library (.so) of each module is created. Further individual modules are combined and tested to evaluate their interaction against the design parameters. We follow the bottom-up approach wherein we test small software elements first and keep on integrating and testing the bigger module. After each module has been integrated the bigger module undergoes regression testing to verify that modifications have not caused unintended affects on the integrated system and the performance. System is also tested to verify

- Abnormal behavior(system crashing, abrupt termination)
- Handling invalid input and large volume of data.
- Analyzing processing time and memory utilization.
- Evaluating Human Computer Interface

A Graphical user interface (see Figure 2) was developed over the integrated system. The GUI provides various options for the user to play with the integrated system.

- Using basic image enhancement and editing tools (cropping, rotation, zoom in/zoom out, orientation, binarization, noise removal etc.).
- Running individual modules successively for obtaining final OCR'ed output.
- End-to-End OCR (Figure 2).
- Defining workflows (Figure 3).
- Text editing tool coupled with dictionary.

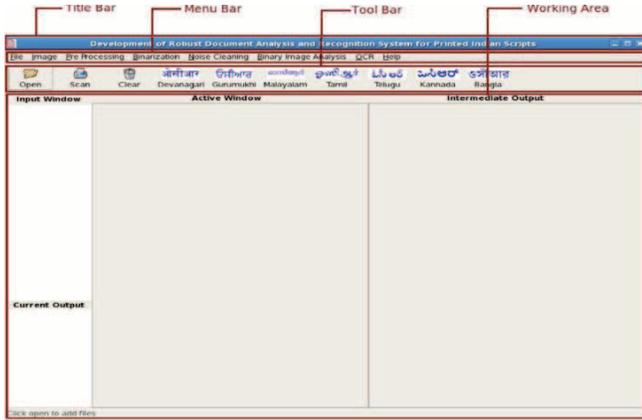


Figure 2: Multilingual OCR GUI

User has the facility to run the end-to-end OCR, wherein the best possible combination of the pre-processing routines and the script specific recognition engine are packaged. Also the user can choose from a set of pre-processing routines suitable for a set of input images jointly with the script specific recognition engine. This can be saved as a user defined workflow which can be used in future for batch processes like OCRing complete books.



Figure 3: Multilingual OCR Workflow

মধুচন্দ্রিমা ১১
 দিয়েছে। কোথায় চলেছে তারা জানে না। এই অকারণ চলা কার জন্য তাও জানে না। তবু এই চলাতেই আনন্দ। গড্ডলিকা প্রবাহে চলার মধ্যেও বোধ হয় এক ধরনের আনন্দ আছে। সেই আনন্দে রূপনারায়ণ বিভোর হয়ে আছে। নীলকণ্ঠ ও

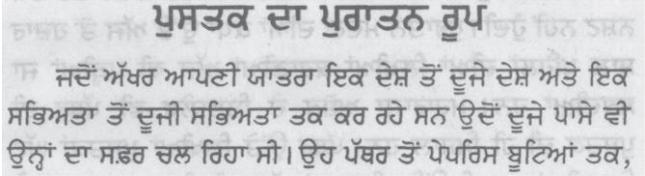
মবুচন্দ্রিমা এ ১১
 দয়িছে। কোথায় চলেছে তারা জানে না। এই অকারণ চলা কার জন্য তাও জানে না। তবু এই চলাতেই আনন্দ। গড্ডলিকা প্রবাহে চলার মধ্যেও বোধ হয় এক ধরনের আনন্দ আছে। সেই আনন্দে রূপনারায়ণ বিভোর হয়ে আছে। নীলকণ্ঠ ও

Figure 3a: Bangla OCR Input and Output

उपेक्षा और अपमान की पीड़ा ढोये जैसे-तैसे वह बाबा के आश्रम में पहुँच गया। बाबा मानो उसी की प्रतीक्षा में बैठे थे। वह ज्योंही दण्डवत की मुद्रा में हुआ त्योंही बाबा का गंभीर स्वर उसके कानों से टकराया 'आओ, मैं तुम्हारे लिए ही बैठा हूँ।'

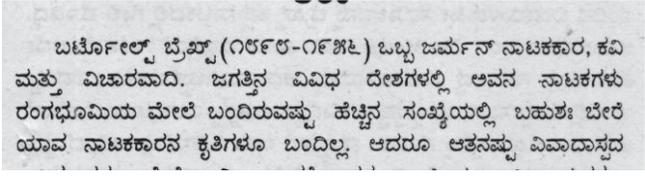
उपेक्षा और अपमान की पीड़ा ढोये जैसेतैसे वह बाबाके आश्रम में पहुँच गया। बाबा मानो उसी की प्रतीक्षा में बैठे थे। वह ज्योंही दण्डवत की मुद्रा में हुआ त्योंही बाबा का गंभीर स्वर उसके कानों से टकराया आओ, मैं तुम्हारे लिए ही बैठा हूँ।

Figure 3b: Devanagari OCR Output



ਪੁਸਤਕ ਦਾ ਪੁਰਾਤਨ ਰੂਪ
 ਜਦੋਂ ਅੱਖਰ ਆਪਣੀ ਯਾਤਰਾ ਇਕ ਦੇਸ਼ ਤੋਂ ਦੂਜੇ ਦੇਸ਼ ਅਤੇ ਇਕ ਸਭਿਅਤਾ ਤੋਂ ਦੂਜੀ ਸਭਿਅਤਾ ਤਕ ਕਰ ਰਹੇ ਸਨ ਉਦੋਂ ਦੂਜੇ ਪਾਸੇ ਵੀ ਉਨ੍ਹਾਂ ਦਾ ਸਫ਼ਰ ਚਲ ਰਿਹਾ ਸੀ। ਉਹ ਪੱਥਰ ਤੋਂ ਪੇਪਰਿਸ ਬੁਟਿਆਂ ਤਕ ,

Figure 3c: Gurumukhi OCR Output



ಬರೋಲ್ಡ್ ಬೈಬ್ಲ್ (೧೮೯೮-೧೯೫೬) ಒಬ್ಬ ಜರ್ಮನ್ ನಾಟಕಕಾರ, ಕವಿ ಮತ್ತು ವಿಚಾರವಾದಿ. ಜಗತ್ತಿನ ವಿವಿಧ ದೇಶಗಳಲ್ಲಿ ಅವನ ನಾಟಕಗಳು ರಂಗಭೂಮಿಯ ಮೇಲೆ ಬಂದಿರುವಷ್ಟು ಹೆಚ್ಚಿನ ಸಂಖ್ಯೆಯಲ್ಲಿ ಬಹುಶಃ ಬೇರೆ ಯಾವ ನಾಟಕಕಾರನ ಕೃತಿಗಳೂ ಬಂದಿಲ್ಲ. ಆದರೂ ಆತನನ್ನು ವಿವಾದಾಸ್ಪದ

Figure 3d: Kannada OCR Output

ദ്യാഹസ്മിണിത്ത ശ്രീ മാരാർ അവാർകൾ പറയുന്നില്ലോ. പക്ഷേ, ഈ പറഞ്ഞ പരാമയ്മകളെ സംഗ്രഹത്തിനുള്ളൊരുമെങ്കിലും ആവശ്യക്കാരനെ മൂലത്തിലേക്കുപിടിക്കുന്ന ഒരേ ഒരാകർഷണവും സംഗ്രഹങ്ങളാകുന്നു.

ദ്യാഹസ്മിണിത്ത ശ്രീ മാരാർ അവാർകളെ പറയുന്നില്ലോ. പക്ഷേ, ഈ പറഞ്ഞ പരാമയ്മകളെ സംഗ്രഹത്തിനുള്ളൊരുമെങ്കിലും ആവശ്യക്കാരനെ മൂലത്തിലേക്കുപിടിക്കുന്ന ഒരേ ഒരാകർഷണവും സംഗ്രഹങ്ങളാകുന്നു. |

Figure 3e: Malayalam OCR Output

பத்தாண்டுகள், இருபதாண்டுகளுக்கு முன் வெளிவந்த எந்த ஒரு தமிழ்நூலும் எவரிடத்திலும் எந்த நூலகத்திலும் கிடைப்பதில்லை. பெரும்பாலான நூல்கள் ஒரே பிறப்பில் முக்தி அடைந்து விடுகின்றன. நாவல் வளர்ச்சியை ஆராயப்

பத்தாண்டுகள் . இருபதாண்டுகளுக்கு முன் வெளிவந்த எந்த ஒரு தமிழ்நூலும் எவரிடத்திலும் எந்த நூலகத்திலும் கிடைப்பதில்லை. பெரும்பாலான நூல்கள் ஒரே பிறப்பில் முக்தி அடைந்து . விடுகின்றன நாவல் வளர்ச்சியை ஆராயப்

Figure 3f: Tamil OCR Output

தெலுగులో సామాజిక స్పృహతో, సంఘ సంస్కరణ లక్ష్యంతో రచనా వ్యాసంగాన్ని సాగించిన మొట్టమొదటి తెలుగు సాహిత్య కర్త ఆయనే. చిలకమర్తి లక్ష్మీనరసింహం మొదలు చెల దాకా ఆయన తన సమకాలీన, భావి రచయితలకు స్ఫూర్తినిచ్చాడు, ఆదర్శం ఆయ్యాడు. తొలుగులో సామాజిక స్పృహతో, సంఘ సంస్కరణ లక్ష్యంతో రచనా వ్యాసంగాన్ని సాగించిన మొట్టమొదటి తెలుగు సాహిత్య కర్త ఆయనే. చిలకమర్తి లక్ష్మీనరసింహం మొదలు చెల దాకా ఆయన తన సమకాలీన, భావి రచయితలకు స్ఫూర్తినిచ్చాడు, ఆదర్శం ఆయ్యాడు.

Figure 3g: Telugu OCR Output

4. PERFORMANCE EVALUATIONS

Performance Evaluation Tool has been developed by CDAC, Noida [2] to analyze the error statistics of OCR output with ground truth data. Figure 4 shows the performance evaluation testing flowchart.

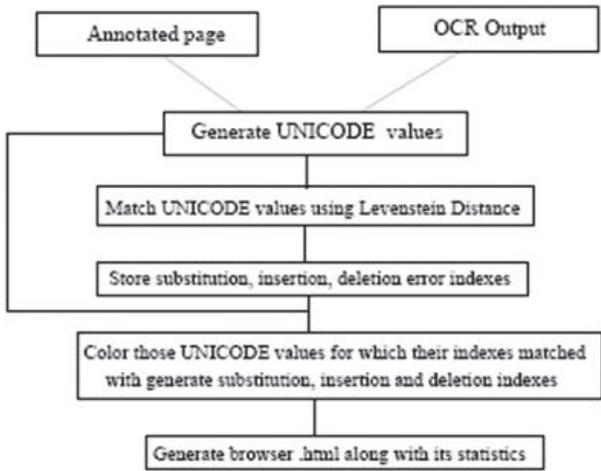


Figure 4: Performance Evaluation Flowchart

The Error rate in OCR output with respect to Ground truth is calculated using Levenshtein distance [2]. It gives the measure of in-equality in terms of insertion, deletion, or substitution at character level. To show the recognition errors a browser window has been created on which ground truth data and OCR output with substitution, insertion and deleted characters are highlighted with

different colors. Match characters have been shown with black color. The characters which are substituted have been shown with red color in both ground truth and OCR output. Insertion error represents those characters which are generated by Recognition engine and have been shown with blue color in OCR output. Deletion error represents those characters which are not recognized by Recognition engine and are shown with green color in Ground truth data.

AnnotationSize	OCRoutputSize	Substitution Error	insertScore	deleteScore	Substitution Error-Rate	Total Error-Rate
1560	1560	23	10	10	1.474%	2.756%

Figure 5: Character Level Testing Browser

Using the Levenshtein distance[2] we get count of match & mismatch of each character in OCR output with respect to ground truth. These counts are used to generate a confusion matrix of n x n size for every script, where n is the number of UNICODE values in a particular script. The diagonal elements represent the number of times a recognized character matches with ground truth character.

The elements in red color represent the number of character mismatch in OCR output as compared to ground truth data. This tool has helped in identifying the problems in various routines which can be worked upon thereby improving the accuracy of OCRs. Figure 6 shows the snapshot of Confusion matrix.

A\O	౧	౨	౩	౪	౫	౬	౭	౮	౯	౧౦	౧౧	౧౨	౧౩	౧౪	౧౫	౧౬	౧౭	౧౮	౧౯	౨౦	
౧	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
౨	246	8733	5	0	9	1	16	0	3	0	0	0	0	2	0	0	0	0	2	19	14
౩	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
౪	0	2	2	2017	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
౫	0	0	0	10	2782	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
౬	0	0	1	0	0	1547	2	3	0	0	0	0	0	0	0	0	0	0	0	0	1
౭	0	3	0	0	0	240	777	0	0	0	0	0	0	0	0	0	0	0	0	0	1
౮	0	4	0	0	0	5	0	1995	5	0	0	0	0	0	0	0	0	0	0	0	1
౯	0	1	0	0	0	0	0	0	308	0	0	0	0	0	0	0	0	0	0	0	0
౧౦	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0

Figure 6: Confusion Matrix

Word level comparison of the final output Unicode sequence with the ground truth using Edit distance based measure. We can define word error rate (WER) as:-
 WER = WordEditDistance(Wtrue,Wocr)/Wtrue
 Where Wtrue is the aligned word in the ground truth data. Edit distance is expected weigh equally insertion, deletion and substitution errors.

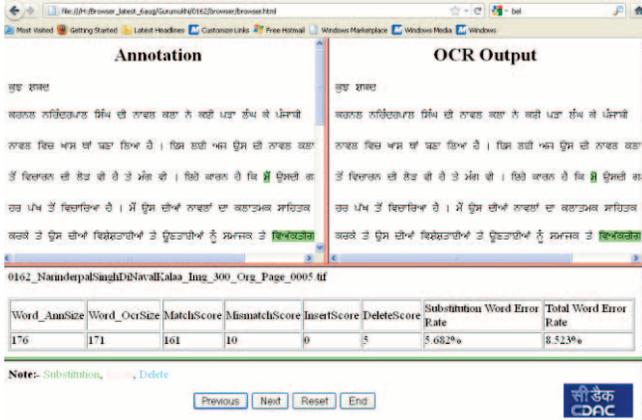


Figure 7: Word Level Testing Browser

5. OCR RESULTS

OCR Performance was also tested by Third Party Tester. We have defined different quality types of documents images:-

Quality B: Books printed on normal paper. Computer typesetting. Without degradations. Higher text density

Table2: QualityB Documents

OCRs	Tested Pages	Overall Total Error Rate	Overall SubstitutionError Rate
Bangla	652	2.95%	1.37%
Devnagari	1037	11.94%	6.80%
Gurumukhi	948	4.11%	2.38%
Kannada	216	8.43%	4.77%
Malayalam (600 dpi)	797	4.61%	2.04%
Malayalam (300 dpi)	797	4.66%	2.35%
Tamil	1192	9.14%	4.14%
Telugu	1569	18.22%	11.84%

Quality C: Average or inferior quality print documents with degradations,.

- (i) Partly non-computer typeset but normal paper/ink.
- (ii) Documents with complex manhattan layout.
- (iii) Computer typeset with poorer paper, degradations, some backside reflections

Table3: QualityC Documents

OCRs	Tested Pages	Overall Total Error Rate	Overall SubstitutionError Rate
Bangla	598	4.03%	1.99%
Devnagari	2097	13.36%	8.37%
Gurumukhi	2284	6.96%	4.09%
Kannada	1386	13.62%	7.62%
Malayalam (600 dpi)	1214	5.29%	2.27%
Malayalam (300 dpi)	1214	6.09%	3.00%
Tamil	1014	9.58%	4.02%
Telugu	1511	26.31%	16.52%

Quality D: Bad quality print, low quality paper, low quality ink, noise, merges, cuts and breaks, blobs, etc.

Table4: QualityD Documents

OCRs	Tested Pages	Overall Total Error Rate	Overall SubstitutionError Rate
Bangla	0	0	0
Devnagari	468	18.02%	11.27%
Gurumukhi	283	7.14%	3.95%
Kannada	145	7.73%	4.49%
Malayalam (600 dpi)	0	0	0
Malayalam (300 dpi)	0	0	0
Tamil	253	15.57%	8.33%
Telugu	0	0	0

6. CONCLUSION

Currently we have developed shared linked libraries of each module (script-independent or script-dependent module) received by other consortia members and integrated with GUI which is packaged into RPM. We are testing each language OCR for character level accuracy and word level accuracy. We are in the process of testing for 5000 pages atleast of each script for character and word level accuracy.

As a future prospect of the OCR project we are trying to tune the recognition to respond well to a variety of fonts and font/point sizes. Also multilingual OCR could be integrated with Braille interface for all Indian scripts addressed here. We can aim of developing large Document Management Systems.

The integration and testing of complex software like Multilingual OCR System is a difficult phase. It was required to verify each module, identify problems/issues and coordinate with the developer.

7. ACKNOWLEDGMENT

This work is funded by the grant from Ministry of Communication and Information Technology, Govt. of India. We thank all the consortium members for their active participation in the project and making it a success.

8. REFERENCES

- [1] Gaurav Harit, K. J. Jinesh, Ritu Garg C.V Jawahar and Santanu Chaudhury Managing Multilingual OCR Project using XML Proc. of International Workshop on Multilingual OCR 2009 Barcelona, Spain.
- [2] Tushar Patnaik, Shalu Gupta and Gaurav K. Rai. Performance evaluation for Indian Languages in Consortia based OCR. ASCNT 2009, CDAC, Noida.
- [3] A. Lear, "XML seen as integral to application integration," IT Professional, vol. 1, no. 5, pp. 12–16, Sep/Oct 1999.
- [4] S Rice, J Kanai and T Nartker, An evaluation of OCR accuracy, UNLV Annual Report, pp 9-33, 1993.
- [5] J Esakov, D. P. Lopresti and J. S Sandberg, Classification and distribution of Optical Character Recognition errors, SPIE Vol. 2181, Document Recognition, 1994.
- [6] P. B. Pati and A. G. Ramakrishnan, " OCR in Indian scripts: A Survey," IETE Technical Review, May-Jun 2005, 22(3):217-227.
- [7] K. G. Aparna and A. G. Ramakrishnan, " A complete Tamil Optical Character Recognition System," Proc. Fifth IAPR Workshop on Document Analysis Systems DAS-02, Princeton, NJ, August 19-21, 2002, pp. 53-57.
- [8] B. Vijay Kumar and A. G. Ramakrishnan, " Machine Recognition of Printed Kannada Text," Proc. Fifth IAPR Workshop on Document Analysis Systems (DAS-02), August 19-21, 2002, Springer Verlag, Berlin. pp. 37-48.
- [9] R S Umesh, Peeta Basa Pati and A G Ramakrishnan, Design of a bilingual Kannada-English OCR, in the book "Guide to OCR for Indic Scripts: Document Recognition and Retrieval" Springer, 2009 in the Advances in Pattern Recognition Series. Ed: Venu Govindaraju and Setlur Srirangaraj. pp. 97-124. ISBN: 978-1-84800-330-9
- [10] Karthika Mohan and C.V.Jawahar A Post-Processing Scheme for Malayalam using Statistical Sub-character Language Models *Proceedings of Ninth IAPR International Workshop on Document Analysis Systems (DAS'10)*, pp.493-500, 9-11 June, 2010, Boston, MA, USA.
- [11] C.V. Jawahar and Anand Kumar Content-level Annotation of Large Collection of Printed Document Images *Proc of 9th International Conference on Document Analysis and Recognition, Brazil, 23-26 September, 2007*.
- [12] U. Pal, B. B. Chaudhuri: Indian script character recognition: a survey. *Pattern Recognition* 37(9): 1887-1899 (2004)
- [13] V. Govindaraju and S. Setlur (Editors), "Guide to OCR for Indic Scripts", Springer, Sep 2009.
- [14] B. B. Chaudhuri and U. Pal, "A Complete Printed Bangla OCR System", *Pattern Recognition*, vol. 31, pp. 531-549, 1998.
- [15] M K Jindal, G S Lehal and R K Sharma, "On Segmentation of Touching Characters and Overlapping Lines in Degraded Printed Gurmukhi Script", *International Journal of Image and Graphics*, Volume 9, No. 3, pp. 321-353 (July 2009).