Application-Oriented Segmentation of Printed Sindhi Text for Document Recognition and Natural Language Processing Systems

Pir Bakhsh Khokhar[1,*], Shahnawaz Talpur[1], Muhammad Ismail[2], Hassan Abbas[3], Muhammad Asif Khan[2]

[1]Institute of Information and Communication Technologies, MUET, Jamshoro, Pakistan.
[2]Department of Computer Science, Sukkur IBA University, Sukkur, Pakistan.
[3]Department of Business Administration, Sukkur IBA University, Sukkur, Pakistan.
[*]Corresponding author: kpirbakhsh@gmail.com

Abstract

Text segmentation of printed Sindhi documents serves as a fundamental requirement for constructing competent OCR and NLP systems for the Sindhi language. The cursive format with complex linking elements in the Sindhi script creates difficulties in automatically identifying lines and characters. This paper presents a new application-driven method to precisely separate lines and ligatures in printed images of Sindhi text. Adaptive thresholding acts first to process noisy and skewed images robustly, and then pixel counting detects the top and bottom lines for subsequent line segmentation. Ligature segmentation is performed through a vertical profile technique on the extracted lines. New techniques within the improved skew correction algorithm target resolve two main text-related problems: unaligned text lines and inconsistent spacing between lines. The system operated on 400 printed Sindhi text images, yielding a line segmentation accuracy of 98.3%. The segmented ligatures obtained through this process form the basis of Sindhi OCR development, which also enables applications in speech recognition, text mining, and other digital language processing operations.

Index Terms—Document Image Analysis, Printed Sindhi Script, Pixel-Based Segmentation, Vertical Projection Profiles, Skew Detection and Correction, OCR Preprocessing, Cursive Script Processing

1 Introduction

Applied linguists and sociolinguists have extensively studied media languages because they influence both public communication and second-language education. Media platforms supply readily available authentic linguistic information that serves research requirements and supports learning activities for both scholars and learners. The media defines linguistic standards through its constant display of authentic language practices occurring across different dialects, along with sociolects. Second-language learners from minority communities often depend on media channels to gain contact with native speakers, because they are their main or exclusive language sources. Linguists study language stylization in advertisements together with tabloid media dialect mimicry and radio-television performance while utilizing these data extensively for their research [1], [2]. Media organizations condense social structures with their political and cultural values while becoming notable shapers of national and linguistic identities. The linguistic significance of media content faces significant challenges because digital and offline data creation generates diverse and extensive information on a daily basis. Under the broad definition of literature, we find two categories: artistic texts that include novels, plays, and poems, while utilitarian content makes up the second group consisting of newspaper reports and other print media. The original unprocessed data format of this content remains difficult for computers to extract from existing physical and scanned forms. Digital transformation processes have dramatically increased in recent years because global data creation amounts to 2.5 quintillion bytes daily, and digital data presently represent more than 90% of all information created during the last two years [3]. Social media generates millions of new entries

ISSN: 2523-0379 (Online), ISSN: 1605-8607 (Print)
DOI: https://doi.org/10.52584/QRJ.2302.03

every day as YouTube users watch five billion videos per day, and Twitter users send hundreds of thousands of tweets per minute. Machine-readable formats such as ASCII, XML, and JSON house digital forms that programming tools can effortlessly decode for processing.

Large pools of information continue to remain inaccessible because they exist as handwritten documents combined with printed books, along with scanned newspaper archives and legacy manuscripts. Both PDF formats and word processing programs are readable by humans; however, they usually do not contain machine-processable metadata. Complex document analysis and optical character recognition (OCR) techniques are needed to transform these data so that computers can perform searches, apply tags, and perform semantic link functions. The development of OCR systems has been rapid for Latin scripts; however, it shows limited progress when operating on cursive writing and non-Latin scripts, especially those that contain complex ligature systems, including Arabic, Urdu, and Sindhi [4]. Recent work has also addressed complex document layout analysis using deep learning, particularly for historical and degraded documents, where segmentation plays a critical role in OCR accuracy [5].

The Perso-Arabic script writing system in Sindhi demonstrates the challenging conditions that result from trying to develop OCR recognition for underrepresented languages. The official language status of Sindhi in Pakistan does not translate into sufficient digital resources to support reliable document analysis because it lacks the necessary tools, including annotated corpora, labeled datasets, lexicons, and segmentation options. Only limited Sindhi OCR activities exist at the character recognition level without being extended to perform word-level or sentence-level segmentation operations. The principal obstacles in Sindhi OCR, according to [6], stem from lines that deviate from vertical alignment as well as double letters along with mispositioned diacritics and character shapes that alter with context.

Research in Arabic and Urdu OCR shows how adaptive thresholding and edge detection preprocessing merged with machine learning or statistical models achieve successful segmentation according to [7], [8]. Research on Sindhi retains a complete lack of investigation, even though it represents a substantial portion of the urdu-speaking community. Standard segmentation methods fail when used on cursive scripts because their compound ligatures render them ineffective in both noisy and scanned documents.

This study adopts a robust mechanism to preprocess and segment printed Sindhi text as a solution to this deficiency. The technique utilizes adaptive threshold control for coping with skew and noise, as well as pixel-count techniques to find the top and bottom baselines for line segmentation combined with vertical profile processing for ligature segmentation. This method stands out because it operates without relying on script-specific heuristics that enable it to work with differently printed Sindhi documents with various fonts and layouts. The framework was evaluated on multiple printed Sindhi text datasets with diverse layouts. The ligature extraction process creates data that enable subsequent text-to-speech synthesis encounters, along with document indexing systems and information retrieval functions.

OCR development in Sindhi remains essential, both technically and as an important cultural and educational requirement. These technological tools promote digital inclusion for under-resourced linguistic groups by enabling the digital conversion of Sindhi newspapers and books along with archival texts. The study aligns itself with international developments in both inclusive artificial intelligence and multilingual technological frameworks, while fulfilling the necessary digital technology accessibility throughout all language groups.

The subsequent portion of this study presents an overview of the Sindhi script and comparable OCR literature starting with traditional methods and continuing to contemporary approaches in **Section 2**. The methodology outlined in **Section 3** consists of fundamental processing steps before introducing correction for skew and line detection and methods for separating ligatures into individual characters. The experimental design section is derived from **Section 4**, which displays the evaluation process and performance indicators alongside outcome assessments through a diverse collection of printed Sindhi data. **Section 5** provides a summary of the contributions along with an evaluation of the current limitations and research plans to enhance full-script recognition and language digitization. Finally, **Section 6** concludes the study with the outcomes of the overall study.

2 Related Work

Optical character recognition (OCR) for cursive and dot-differentiated scripts such as Arabic, Urdu, and Sindhi has gained increasing attention due to the growing demand for digitization of non-Latin textual resources. These scripts present inherent challenges for OCR systems because of their cursive structure,

overlapping characters, contextual shape variations, and the presence of multiple diacritical marks. As a result, segmentation into lines, ligatures, and characters remains a fundamental difficulty, particularly for printed and scanned documents affected by noise and skew.

Early research efforts in Arabic OCR explored statistical sequence modeling techniques. Ahmad et al. [7] proposed a multi-font Arabic OCR framework based on Hidden Markov Models (HMMs), employing sliding-window feature extraction and a two-stage recognition strategy. Font identification was first performed, followed by the application of font-specific recognizers. Their results demonstrated that font-aware recognition significantly improves accuracy, an approach relevant to Sindhi OCR where font variability is common.

Assistive OCR systems have also been explored in constrained environments. Prajapati and Shah [9] developed a real-time wearable OCR system for visually impaired users that integrated video capture, zone-based OCR, and text-to-speech synthesis. Although limited to English scripts, the work highlighted the feasibility of segmentation-driven OCR pipelines for resource-limited and mobile deployments.

Beyond recognition, several studies have emphasized the importance of structuring textual data after digitization. Chakraborty and Shima [10] introduced EduBD, a semantic web-based framework that converts unstructured educational text into RDF-based representations. While not an OCR system itself, the study illustrates how effective segmentation and recognition form the foundation for higher-level semantic processing tasks such as classification and retrieval.

Sindhi-specific OCR research remains comparatively limited. Ali et al. [11] investigated handwritten Sindhi digit recognition using traditional machine learning classifiers, including k-NN, Decision Trees, Multilayer Perceptrons, and Random Forests, with Random Forests achieving the best performance. Similarly, Kumari et al. [12] developed an offline Sindhi character recognition system using feed-forward neural networks trained via back-propagation. Their experiments highlighted inter-writer variability and demonstrated the feasibility of neural models for isolated character recognition. Early machine learning-based efforts include handwritten Sindhi digit recognition using classical classifiers such as k-NN and Random Forests, demonstrating the feasibility of Sindhi script recognition at the symbol level [13].

A comprehensive analysis of technical challenges in Sindhi OCR was presented by Hakro et al. [6], who identified major obstacles such as inconsistent dot placement, contextual shape variation, and difficulties in segmenting interconnected characters under skewed and noisy conditions. Their findings emphasize the necessity of robust preprocessing, skew correction, and ligature-level segmentation in Sindhi OCR pipelines.

Advances in Urdu OCR have provided transferable insights due to script similarity. Ahmad et al. [14] proposed a sentence-level Urdu OCR system based on bidirectional LSTM networks, demonstrating that ligature-based modeling preserves contextual dependencies effectively. These approaches are directly applicable to Sindhi, which shares Nastaleeq writing characteristics and ligature structures.

More recent work has shifted toward deep learning-based segmentation and recognition. Ali et al. [11] introduced the Subword Guided Neural Word Segmenter (SGNWS) using BiLSTM-CRF architecture, achieving a high F1 score for Sindhi word segmentation without handcrafted features. Mahar et al. [15] proposed a multi-level tokenizer that segments Sindhi words into simple, compound, and complex forms, achieving competitive accuracy using contextual and whitespace cues.

Line segmentation remains a critical preprocessing step for cursive scripts. Goel and Lehal [8] developed an adaptive projection-profile-based algorithm capable of handling skew and noise across Indic scripts, demonstrating robustness under varied document conditions. Their work supports the applicability of adaptive thresholding techniques for printed Sindhi text.

Earlier projection-profile-based approaches were also explored for Sindhi line segmentation; however, these methods were often evaluated on limited datasets and struggled with skewed or irregularly spaced text lines [**?**].

Deep neural architectures have also been applied to Sindhi character recognition. The study in [16] employed a deep residual network with summation fusion to recognize isolated handwritten Sindhi characters, achieving improved performance through effective spatial feature representation. Benchmarking efforts by Awan et al. [17] further contributed to the field by releasing a comprehensive Sindhi ligature dataset covering multiple fonts and layouts, enabling more realistic OCR evaluation.

Recent transformer-based models have broadened OCR capabilities across scripts. Tan et al. [18] proposed TrOCR, a transformer-based OCR model pretrained on multilingual data, achieving strong results on both printed and handwritten text. Although not yet adapted specifically for Sindhi, such architectures show promise for future low-resource script recognition. Similarly, Mustafa et al. [19] evaluated large lan-

guage models, including GPT-4, for non-Latin OCR tasks and found that while zero-shot performance is limited for cursive scripts, fine-tuning with ligature-level annotations significantly improves results.

Institutional efforts have also supported Sindhi digitization. The Abdul Majid Bhurgri Institute of Language Engineering [20] has contributed to Unicode standardization, font development, and dataset creation for Sindhi, providing essential resources for OCR and language technology research.

Similar challenges for non-Latin scripts were addressed by Win et al. [21] for Myanmar printed documents.

Despite these advances, few studies address the combined challenge of skew correction, line detection, and ligature segmentation specifically for printed Sindhi text. Existing approaches often focus on isolated components or rely on script-specific heuristics. This gap motivates the present study, which proposes a unified, adaptive segmentation framework designed to operate across varied printed Sindhi documents without font-dependent assumptions.

3 Research Methodology

3.1 Dataset Selection

The research data stems from [18], and the researchers used their developed software to convert the collected data into text image formats. A specific set of parameters, such as font and font size, along with color and gray levels, are fed into software until the program transforms the information into digital images from books, newspapers, magazines, and websites. The software includes the functionality to cut printed text images into selected sizes. The developed database is described in the next section. Their database included various books published by many publishers, from poetry to fiction to history and novels. A special application was built to accept these books, which produced grayscale and binary images of various sizes.

They gathered many Sindhi magazines for scanning, and selected Saranga as their main focus, both as printed and digital copies. The website contains printed Sindhi text materials that researchers accessed through numerous websites on the Internet, where they also found electronic magazine PDF books and articles. The team provided content to the developed software while converting the data from these websites to text-based images. Their internet journey included visiting https://www.sindhica.org/ and https://sindhsalamat.com/ together with other pages.

A bulk collection of Sindhi printed text can be found on various newspaper websites. The analysts collected information through searches of Sindhi newspapers that operated online. The research team collected information by analyzing the daily Sobh, kawish, Sindh Hilal-e-Pakistan, Awami Awaz, etc. The team inserted these data into a software program that generated the printed text images.

The research team gathered naming trends for boys and girls by visiting educational hostels for boys and girls. The researchers obtained ordinary gender names in English and transferred them to Sindhi. The program converted 1,980 names into images from multiple font families, textual sizes, projection hues with numerous background color combinations across gray levels, and binary image formats. The researchers collected both the names and castes of the individuals and added the Sindhi versions. A specific application converts the commonly known castes of Sindh Province into formatted textual images. Government websites and voter lists serve as sources for compiling both district information and city names. A custom-built application designed by the researchers converted the gathered data into printed text images. The Sindhi text processing application served to gather names from international countries and their capital through manual input from internet sources. The repository received country and capital names that were transformed into a text image format to increase system flexibility. Figure 1 illustrates the overall block diagram of the proposed segmentation methodology. The following steps were performed to complete the operations:

- First, the above-listed printed Sindhi text database undergoes preprocessing.
- In the second step, the text-filled image requires segmentation into lines.
- The splitting operation occurs between lines before segmenting the individual ligatures or characters.
- The proposed methodology operates on a database of 400 text pictures (2800 lines) from the same Sindhi printed text image collection.
- A second approach was designed to solve specific problems in text-line segmentation, including differences in text size and irregular line spacing, together with text-line irregularities.
- The proposed algorithm demonstrated 98.3% success during the evaluation of the printed text data.
- The final part of this technique establishes a dataset for text recognition from Sindhi printed text images.

The proposed technique accurately retrieved lines

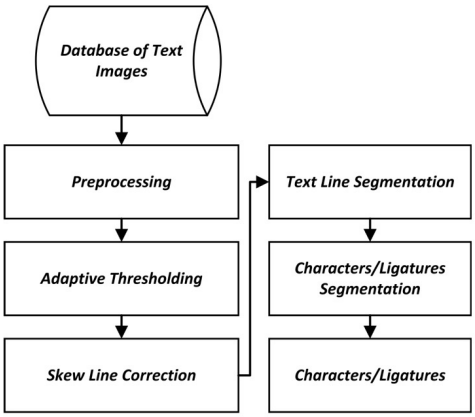

Fig. 1: Block Diagram of Proposed Methodology

and ligatures during the evaluation of the Sindhi text documents.

3.2 Data Preprocessing

Data preprocessing is essential for obtaining superior segmentation. The adaptive thresholding method assists in image pre-processing by supplying noisy data. The simplest version of adaptive thresholding produces a binary image output by operating on grayscale or color images. The threshold value must be calculated for each visual point in the presented image. The threshold data used for each pixel results from applying interpolation techniques to all sub-image outcomes. The RGB image requires conversion to the YCBCR color space, where Y represents the intensity range of black and white pixels and allowing an 8-bits grayscale depth and 24-bits RGB depth. Before adaptive thresholding begins, a grey-level conversion process takes place.

A binary image is developed from an RGB or grayscale image to process the white and black pixels in the image. Only two distinct colors exist in the text images, because they include the textual element color and background shade. Conversion to binary images creates an increased contrast, which simplifies the thresholding process using Otsu's method.

Non-uniform picture brightness requires adaptive thresholding to separate foreground objects from their background areas. An adaptive filter should be run first to emphasize the visual features before the Otsu threshold produces a binary image output. Otsu's thresholding method produces excellent results for textual information datasets. As a result of the improved preprocessing technique, pixels are converted into white when intensity values exceed the threshold level, but become black when intensity values fall below this threshold. This generated an image with black and white pixels. The calculation of adaptive thresholding contains the following mathematical elements:

$$\sigma_T^2 = \sum_{i=0}^{L-1} (i-\mu)^2 P_i \tag{1}$$

Probability is indicated by P in the algorithm, and I symbolizes the L bin intensity values. A specialist edge detector generated a complete edge border consisting of a single pixel thickness throughout the entire image. Our sources demonstrate that a Canny edge detector functions as a pre-processing tool. A low-pass filter implemented by the Canny edge detector was used to reduce dot noise. The detection process requires the application of a Sobel filter followed by non-maximal suppression through multiple local selections to obtain the best edge pixels.

Algorithm 1 Preprocessing of Printed Sindhi Text Image

Require: Input grayscale image I
Ensure: Preprocessed binary image I_b
1: Convert input image I to grayscale if required
2: Apply noise reduction using median filtering
3: Estimate image skew angle
4: Correct skew by rotating the image accordingly
5: Apply adaptive thresholding to obtain binary image I_b
6: Remove small connected components and background noise
7: Perform morphological operations to enhance text regions
8: Normalize image contrast and intensity
9: **return** I_b

3.3 Image Skew Correction

The skew correction method is the main component of the proposed line segmentation procedure. Line segmentation methods only succeed with unskewed images for proper performance results. Our algorithm for correcting image skew stands as a development using pixel intensity details from images which this work proposes. The text lines of the printed text images were used by this skew correction algorithm to create straight lines across the text lines.

The proposed algorithm targets white areas between the text lines and characters of Sindhi printed sentences to locate areas where the background contains white spaces. The search process examines white

sections located within the center of the text lines, without checking the content. The process moves from the middle of the first space line to the middle of the next line when the page finishes. This technique runs repeatedly across every page during the skew correction process. The algorithm determines the text-line angle before performing a rotational shift around the center point when skew detection occurs.

Algorithm 2 Skew Correction of Printed Text Image

Require: Noiseless skewed document image I
Ensure: Skew-corrected image I_c
1: Extract the Region of Interest (ROI) from I by scanning the document
2: Identify all pixels located in the space between text lines, excluding text pixels
3: Merge the identified pixels to form a reference line approximating the text slope
4: Generate a bottom reference line through the mid-point of the fitted line
5: Compute the skew angle θ between the reference line and the horizontal axis
6: Rotate the image I by angle $-\theta$ to correct skew
7: Obtain the corrected image I_c
8: **return** I_c

3.4 Line and Ligature Segmentation

Noiseless and straight images are sent to the line segmentation algorithm for line segmentation as seen in Figure 2. The supplied binary image is converted to generate an image of IHXW, which features a black inscription on a white background. Subsequently, the pixel strength denoted by P of the black text on a page was calculated.

$$P(x) = \sum_{y=1}^{H} X(x, y) \tag{2}$$

The threshold of a text picture is determined by pixel strength (P), which varies for each image. Dark pixels with a value greater than the standard deviation of the document are eliminated from the rows. Between the top line and footer, the text line comprises many rows received as pixels (P).

$$P_{x_i} : x_{i-1} < x_i > x_{i+1} \tag{3}$$

$$P_{x'_i} : x'_{i-1} > x'_i < x'_{i+1} \tag{4}$$

The equations above represent the requirements for determining the top and bottom lines of a line in a printed text image. Two lines are split by 'white gaps' whenever the adaptive threshold in rows of the overall document is lower than the black pixels in a row, as seen in Figure 3. This white pixel serves as a border between the two lines of the printed text.

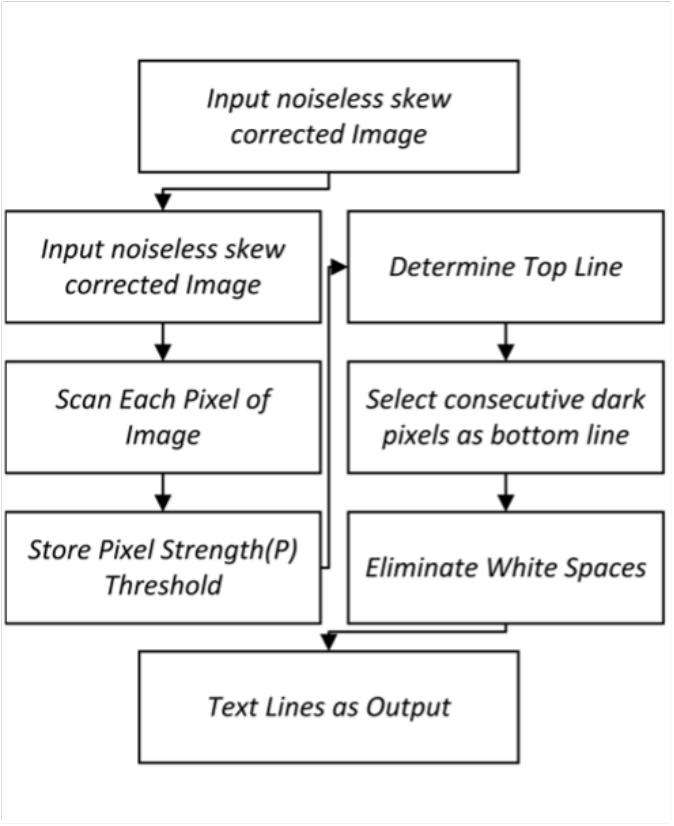

Fig. 2: Proposed Methodology for Line Segmentation

The adaptive page threshold computes the standard deviation rates of text pixels to determine page numerical diversity. The standard deviation increases in proportion to the space between the middle lines, according to this theory. The algorithm determines the line that contains the minimum number of straight text rows. Visible marks located above the lines require fewer pixels than the other text elements. The

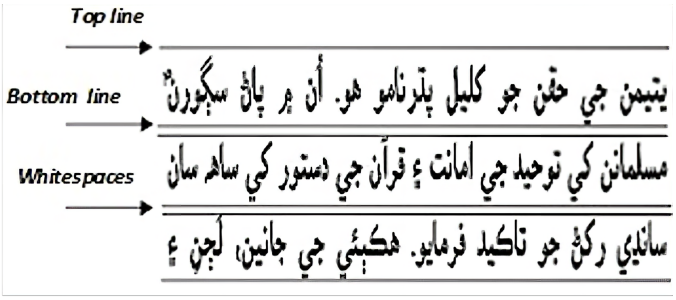

Fig. 3: Top-line, bottom line and white spaces representation in a text image

identification of text pixels within lines is challenging when a row contains pigmentation below the determined threshold level. This affects text-pixel recognition processes. The method chosen in this approach only utilizes pixel counting. The application of this method depends on how the page lines have more black pixels than the blank spaces between them. The length of the segmented line depended on the defined height threshold parameter. The set threshold value is obtained from the examined page. Page line spacing served as the basis for adjusting the threshold value using this tailored method. A constant threshold adjustment system must be integrated into the approach because frequent changes in line spacing occur. The text lines are segmented through ligature segmentation using the projection profile approach; however, the vertical profile method turns lines into words. The proposed system accepts a Sindhi printed text page as input. Each segmented text line receives words and ligatures using a segmentation method that minimizes all possible ligature units. Ligatures display sequence-based sorting behavior because the algorithm digitizes words step-by-step.

4 Results

This section evaluates the effectiveness of the proposed segmentation technique, which includes adaptive thresholding, skew correction, pixel-based lines, and ligature segmentation. The method was applied to a dataset of 400 printed Sindhi text images derived from books, newspapers, and digital platforms, representing a diverse collection of fonts, line spacing, and textual structures.

The dataset used for evaluation was compiled from various sources, including scanned pages from newspapers (e.g., Daily Kawish and Awami Awaz), books, and manually generated text samples [18]. Each image was processed to identify and label the ground-truth line and ligature boundaries. Table 1 provides an overview of the specifications of the dataset.

TABLE 1: Dataset Specifications

Parameter	Value
Total number of pages	400
Skewed pages	80
Average text lines per page	15
Skewed lines (approximate)	110
Average words per page	165
Average words per line	11
Total word count	66,000

Line segmentation was evaluated by comparing the number of correctly segmented lines against the labeled ground truth. The proposed algorithm achieved a line segmentation accuracy of 98.3%, with minor errors occurring in densely packed or overlapping lines. The adaptive thresholding and skew correction steps were critical for improving the success rate of line separation, especially for skewed or noisy inputs. Let R be the number of correctly segmented lines and L be the number of total ground truth lines:

$$\text{Accuracy} = \frac{\sum_{i=1}^{n} L_i - (\sum_{i=1}^{n} L_i - \sum_{i=1}^{n} R_i)}{\sum_{i=1}^{n} L_i} \times 100 \tag{5}$$

Using this formula, the performance was measured for all the 400 images. Of 6,000 labeled lines, 5,898 were accurately segmented, yielding

- Precision: 97.9%
- Recall: 98.7%
- F1-Score: 98.3%

Average processing time per image: 0.42 seconds

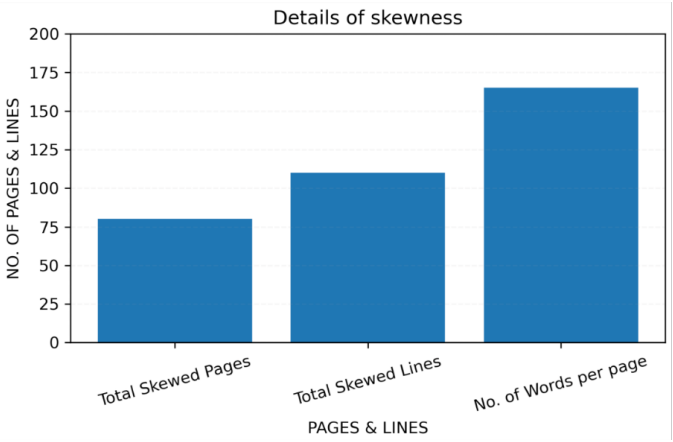

Fig. 4: Skew-related statistics across the dataset, including the number of skewed pages, skewed lines, and average words per page.

Document scanning and printing introduce skew, which remains one of the most detrimental elements for effective line segmentation. The proposed technique detects skew deviation through an intensity-based process that determines Regions of Interest between text lines to calculate the angular shift. The detection process triggers a rotation of the image that occurs at its central point to achieve horizontal positioning. The evaluation revealed that 80 out of 400 pages demonstrated perceptible skew problems because of the combination of scanning imperfections and text-alignment problems. From the 80 image cases that were detected as skewed, the skew correction module successfully corrected 73 images, resulting in a total skew correction accuracy of 91.25%.

Figure 5 shows how the skew affects projection profiles through graphical illustrations that presents uncorrected histogram comparison and post-correction

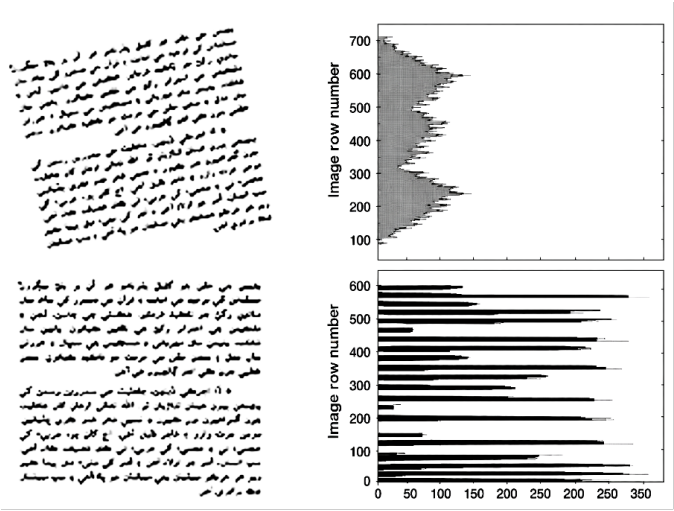

Fig. 5: Skewed vs. deskewed Sindhi text images and corresponding black pixel projection profiles.

results. Line detection becomes irregular because the black pixel distributions do not align properly on the vertical axis before correction occurs. The text rows become easier to detect through a more uniform distribution resulting from skew adjustment procedures. The improvements obtained create a base that supports the next stage of line segmentation.

The distribution details of skewness are shown in Figure 4 including the count of skewing documents and skewed lines and average word length across all records as presented in "Details of Skewness." The number of more than 100 skewed lines throughout the dataset confirms why this corrective step is essential.

The processing workflow included both correction of document skew and preprocessing before sending data to the line segmentation module. The algorithm adopts adaptive thresholding, pixel density evaluation, and projection profiling during its operation. The system detects the top and bottom edges of rows through the analysis of black pixel distributions and divides the images into separate lines at locations with white gaps between them.

The dataset received 6000 text lines of human annotation for use as reference points. The proposed algorithm executed a line segmentation of 5,898 lines leading to a total accuracy rate of 98.3%. The algorithm established its performance through a matching process between automatically detected lines and manually drawn ground truth references. The algorithm demonstrated 97.9% precision, which showed that almost all detected lines were accurate, and achieved 98.7% recall through the retrieval of most actual lines in the documents. The combined F1-score measurement reached 98.3%, which indicates superior performance of both detection and precision competency. The recall performance exceeds the precision by a small margin as shown in Figure 6, indicating that the detection method maintains a sensitive approach to avoid missing any valid lines.

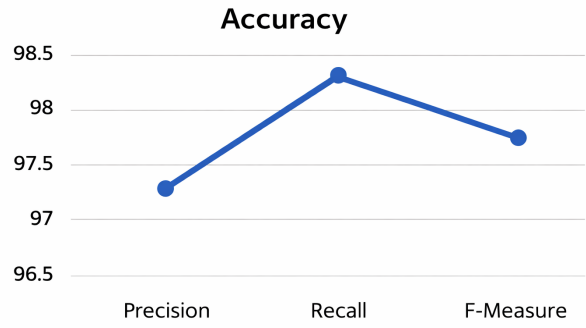

Fig. 6: Accuracy evaluation graph showing system performance in terms of Precision, Recall, and F-Measure. The Recall metric peaks highest, indicating the system's sensitivity to true line detection.

The proper alignment of lines during the segmentation process is essential for the Sindhi script, which must maintain baseline stability. Poor segmentation resulting from either oversegmenting or undersegmenting leads to distorted OCR outputs through broken lines and merged lines between texts. The algorithm demonstrates its operational strength in dealing with real-world printed Sindhi documents by maintaining stability when processing newspaper columns, along with stylized titles within dense layouts.

Line segmentation leads to ligature segmentation as an intricate process, because Sindhi contains cursive script features with context-dependent character shapes and compact ligature structures. Vertical projection profiling detects valleys (white gaps) between connected characters and ligature blocks, according to the algorithm. The analysis converts the segmented line into a vertical pixel profile to detect the spaces between each black pixel group.

The algorithm analyzed 66,000 words in the dataset, which led to the correct separation of 60,000 ligature units. The system achieved a 91.4% accuracy rate for ligature unit division. Figure 5 presents visual examples of ligature identification that show vertical white lines to separate successful boundaries within Sindhi text lines. The separation method demonstrated reliability, because the two results showed no significant variation. The extracted ligature blocks

display even width and spacing measurements, which proves the high resilience of segmentation algorithms even when processing compound or multi-stroke forms. Several misidentified segments occurred because specific sections had closely spaced ligature connections or dot patterns that crossed character boundaries.

Fig. 7: Vertical segmentation of a Sindhi text line showing ligature boundaries. Each vertical white line represents a successfully identified ligature break extracted through projection profiling.

Vertical segmentation of a Sindhi text line showing ligature boundaries as shown in Figure 7 . Each vertical white line represents a successfully identified ligature break extracted through projection profiling.

Most segmentation errors occurred as over-segmentation, which divided single ligatures because of internal gaps or noise, and under-segmentation, which combined two or more ligatures when spacing was inadequate. The method faced more problems when operating on scans of degraded newspapers or fonts with stylized designs. However, most boundary recoveries were successful because of the adaptive characteristics of the projection profile method.

The segmentation pipeline demonstrates its effectiveness, because it functions effectively through different document types without script-dependent heuristics. This text segmentation method demonstrates its main advantage of being usable on various documents owing to its lightweight design and minimal computational requirements, but also because it does not depend on deep learning models or large annotated dataset requirements. Lower resource language support is possible through succinct method design.

The accumulated figures demonstrate both the numerical outcomes and visual evidence that support the quality of segment-based analysis. Figure 7 displays the issues with skew correction, while Figure 6 demonstrates how the line segmentation algorithm achieves its high accuracy metrics of precision, recall, and F-measure, and Figures 2 and 3 show specific examples of ligature isolation success.

The proposed framework enables precise document segmentation regardless of difficult document conditions, including skew, noise, and dense font arrangements. The text processing system for printed Sindhi documents works optimally because it utilizes a combination of skew correction integration and adaptive preprocessing at two segmentation levels. The achieved results lay a solid foundation for complete Sindhi OCR development and significantly advance the digitization and archival activities of Sindhi scripts for educational and linguistic purposes.

The experimental results prove that the proposed segmentation method demonstrates both high effectiveness and flexibility when applied to printed Sindhi texts. The system displayed excellent performance throughout the preprocessing sequence, skew correction procedure, line segmentation process, and ligature extraction process. The method demonstrated resilience because it achieved 98.3% accuracy in line segmentation and 91.4% accuracy in ligature segmentation while successfully correcting skew errors in more than 91% of noisy and complex real-world documents. Both pixel-counting analysis with projection profiling and adaptive thresholding produced an efficient method that delivered precise results for handling Sindhi low-resource scripts. Visual graphic representations, together with statistical data analysis, display both the stability of the model and its capability to effectively segment different layouts. These findings create a vital base for developing complete Sindhi language digitalization systems, which will lead to further work regarding textual recognition, search functions, and voice application development.

5 Discussion

Comparison With Existing Work

Prior research on Sindhi script-processing methods must be used to understand the performance of the proposed segmentation system. The main focus of past efforts directed at Sindhi text processing was word-level segmentation through rule-based and neural methods; however, line processing across various document types has received less attention.

Mahar et al. [15] developed a hierarchical tokenizer for the Sindhi word segmentation. The method operated through three separate levels that used white space for basic word isolation in the first step, followed by simple and compound phrase identification in the second step, and complex term analysis in the third level. The evaluation of the tokenizer showed a segmentation accuracy of 91.76% across 2,792 Sindhi words. This methodology achieved good results in word-level separation for structured written material, but heavily depended on human-made rules together with whitespace recognition, which made it weak when processing tightly packed or irregularly printed content.

Ali et al. established a deep learning-based subword-guided neural word segmenter (SGNWS) that they applied to Sindhi Word Segmentation in

their study [11]. The authors implemented BiLSTM networks and added self-attention features and CRFs within a single model that automatically detected morphemic patterns and subword dependencies without requiring manual feature engineering. Rephrase the sentence to make it direct, flowing, and easy to understand, while normalizing verbalization when possible. The model isolated its focus on word segmentation while preventing the analysis of full-page printed documents through line and ligature segmentation until the present research.

Shaikh and Chandio [16] established a projection profile-based approach for performing line segmentation on Sindhi printed text. A solution to over- and under-segmentation problems was achieved using their method, which used adjusted pixel density thresholds combined with content structural segmentation boundary refinements. A total of 100 pages of Sindhi novels were evaluated, resulting in a line segmentation accuracy of 99.95% according to the proposed method. The reported results appear outstanding, whereas their testing data consist only of scanned high-quality book images. As the system failed to work with different document styles, the outcome remains restricted for general use. The proposed method was successfully applied to a wide range of document data types that contained both skewed and variable formatting elements, while maintaining an accuracy level of 98.3% line segmentation.

The field of Sindhi text processing has received significant contributions from previous approaches, although these approaches have mainly concentrated their efforts on single segmentation problems while focusing on limited document input types and structured document formats. This study focused on the research forward through its development of a thorough noise-tolerant system that performs line and ligature segmentation across different types of printed Sindhi documentation. The solution has both scalable preprocessing operations and a language-free design that enables its successful deployment in real-world Sindhi text applications, including OCR workflows, public record digitization, and accessible reading tool development.

Practical Implications for Sindhi OCR and Language Digitization

These study outcomes create substantial advancement in Sindhi text digitization methods specifically geared toward printed documents showing different and complicated formatting patterns. This approach provides sufficient segmentation precision levels of 98.3% for line segmentation combined with 91.4% for ligature extraction, which proves to be a suitable fundamental component for establishing complete Sindhi OCR systems. The system works well for each document type, ranging from books to newspapers to websites, because it fits various public archive programs, educational digitization projects, and cultural preservation initiatives. The system enables Sindhi language integration into the digital infrastructure through linguistic tools that work to reduce digital exclusion among minority languages.

Accurate text segmentation is also a prerequisite for higher-level semantic processing tasks, such as ontology querying and structured knowledge extraction, where errors introduced during OCR directly affect downstream NLP systems [22].

The methodology is different from deep learning methods because it requires limited GPU processing, simple training datasets, and standard computational resources for operation. The system is suitable for local institutions, libraries, and research centers in low-resource areas because of its usability. The framework maintains its validity across different scripts because it does not depend on heuristic programming, making it adaptable to more than one cursive language type.

Technical Reflections and Observed Limitations

The results from the framework confirm its operational strength; however, existing technical problems have emerged. The main obstacles in documents presenting severe deterioration features include inadequate scan resolution, ink bleeding, or substantial background disturbances. When such cases occur, the projection profile-based technique incorrectly identifies white spaces, which leads to over- and under-segmentation of ligatures and lines. A high number of space irregularities emerged in stylized headlines as well as in older newspaper archives owing to their frequent appearance.

The present approach assumes that layouts contain only one language type and structural design. The present system requires optimization before it can handle documents with multiple columns, integrated images, and Latin characters at different text levels. The segmentation pipeline operates best on layout materials with straight lines that run vertically and horizontally, yet remains vulnerable to the normal formatting deviations found in publications written in various languages and graphics-driven materials.

Linguistic information in the data is not important to the segmentation strategy. The system's lack of language awareness for outcome validation presents a

disadvantage owing to its generalized approach toward segmentation. The system produces occasional incorrect ligature separations that mostly occur when pixel patterns within characters match other pixel patterns in the text.

Opportunities for Improvement and Future Work

Researchers should address these disadvantages by using technical solutions and language development mechanisms. High-end technical improvements will occur by including deep learning segmentation models, such as CNNs and TrOCRs, to gain a better context for character edges. Sindhi-trained models using these abstract features instead of visual limits boost the segmentation outcomes from this system.

The implementation of language-aware post-processing should act as a module to improve the segmentation results. A simple Sindhi morphological analyzer or statistical language model analyzes text segments to check whether the proposed segmentation sequence follows grammatical rules, thus allowing automated error correction. Language integration into the system leads to better reading performance in crowded text documents and degrades visual reading conditions.

The next crucial step includes creating a publicly available Sindhi OCR dataset [23] that should contain annotations for lines, along with ligatures and characters. This dataset development would serve to enhance both the current methodology and stimulate research blending across similar script approaches. A stronger Sindhi script recognition environment can be created when benchmarking tools are combined with standard evaluation metrics and shared models.

The established system operates at an initial stage in larger processing systems, such as indexation tools, speech-to-text applications, machine translation systems, and reading aids. Sindhi script integration with mobile OCR apps, as well as government digitization projects along with educational repositories, will make a strong contribution toward enhancing accessibility along with information retrieval methods in Sindhi.

This research created and proved an adaptable comprehensive and inclusive segmentation system for processing printed Sindhi texts. The technology bridges an important void in low-resource OCR development while achieving effective performance for skewed multi-source printed documents. The fundamental work introduced here will serve as the basis for developing end-to-end Sindhi OCR systems and general script digitization solutions, even though additional improvements can be made by integrating deep models and linguistic understanding along with larger datasets.

6 Conclusion

This research established a complete and flexible segmentation framework for printed Sindhi text because low-resource languages require an effective script-aware text preprocessing solution. The proposed system delivers effective segmentation results for lines and ligatures through a combination of adaptive thresholding, skew correction, and pixel-based projection profiling on various document types with noise. The framework achieved 98.3% line segmentation accuracy and 91.4% accuracy in ligature segmentation during the implementation on pages with skew distortion, various font types, and degraded text inputs. This approach demonstrates reliable performance by processing various printed Sindhi materials, including books, newspapers, and magazines, along with digital content. Owing to its high processing speed and ability to work without learning data while maintaining wide applications, this system excels in areas lacking sufficient OCR tools for native languages. The system's basic design combined with its high accuracy makes it a good selection for use in large OCR systems and assistive technology frameworks that support language processing and text-to-speech accessibility for blind users. The research states both advantages and disadvantages, where the system reacts strongly to intense noise, shows constraint adapting to handwriting, and does not incorporate language-specific processing. Research opportunities exist in these fields. Improving the framework with language-specific enhancements and network-based ligature identification requires the building of uniform datasets for future development. This research provides a hands-on ready solution and conceptual mapping for Sindhi OCR development that will advance in the future. Through this development, the Sindhi language has gained digital representation in modern computational systems, thus supporting language diversity by enhancing the accessibility of technological tools.

References

[1] A. Bell, *The Language of News Media*. Oxford: Blackwell, 1991.

[2] J. Androutsopoulos, "Potentials and limitations of discourse-centered online ethnography," *Language@Internet*, vol. 5, 2008.

[3] G. Press, "The next stage in the digital transformation of museums." https://www.forbes.com/sites/gilpress, 2021. Accessed: 2026.

[4] N. Schenk and R. Mihalcea, "Building and evaluating a complex annotated corpus for arabic ocr correction," in *Proc. Int. Conf. Language Resources and Evaluation (LREC)*, 2012.

[5] L. Suteu, D. Sitar-Taut, and A. Andreica, "A deep learning approach for complex document layout analysis and ocr of historical books," *Journal of Imaging*, vol. 6, no. 10, p. 96, 2020.

[6] D. N. Hakro, A. R. Memon, and M. A. Chandio, "Issues and challenges in sindhi ocr," *Sindh University Research Journal (Science Series)*, vol. 46, no. 2, pp. 231–238, 2014.

[7] I. Ahmad, S. A. Mahmoud, and G. A. Fink, "Open vocabulary recognition of machine-printed arabic text using hidden markov models," *Pattern Recognition*, vol. 51, pp. 97–111, 2016.

[8] S. Goel and G. S. Lehal, "A projection-profile-based line segmentation method for indic scripts," *IEEE Access*, vol. 10, pp. 78065–78076, 2022.

[9] R. Prajapati and P. Shah, "Design and testing algorithm for real-time text images: Rehabilitation aid for blind," *International Journal of Science Technology and Engineering*, vol. 2, no. 11, pp. 275–278, 2016.

[10] Chakraborty and Shima, "Edubd: A machine understandable approach to integrate information of educational institutions of bangladesh." Unpublished manuscript.

[11] W. Ali, M. Bhatti, and M. Afzal, "A subword guided neural word segmentation model for sindhi," *arXiv preprint arXiv:2012.15079*, 2020.

[12] A. K. et al., "Off-line sindhi handwritten character identification," *International Journal of Information Technology and Computer Science*, vol. 11, no. 6, pp. 9–17, 2019.

[13] M. A. Ali, D. N. Hakro, and M. A. Chandio, "Machine learning-based sindhi handwritten digit recognition," *International Journal of Computer Science and Network Security*, vol. 19, no. 10, pp. 195–202, 2019.

[14] I. Ahmad, X. Wang, Y. Mao, G. Liu, H. Ahmad, and R. Ullah, "Ligature-based urdu nastaleeq sentence recognition using gated bidirectional lstm," *Cluster Computing*, vol. 21, no. 1, pp. 703–714, 2018.

[15] J. A. Mahar, H. Shaikh, and G. Q. Memon, "A model for sindhi text segmentation into word tokens," *Sindh University Research Journal (Science Series)*, vol. 44, no. 1, 2012.

[16] M. A. Chandio, D. N. Hakro, and R. A. Memon, "Summation fusion-based deep residual network for sindhi handwritten character recognition," *Journal of King Saud University – Computer and Information Sciences*, 2021.

[17] A. H. A. et al., "Font-diverse sindhi ligature dataset for ocr benchmarking," *Journal of Low Resource Languages and Applications*, vol. 3, no. 2, pp. 35–44, 2020.

[18] M. Tan, J. Wang, Y. Zhang, V. Bapna, and W. T. Yih, "Trocr: Transformer-based ocr with pre-trained vision and language models," *arXiv preprint arXiv:2109.10282*, 2021.

[19] K. Mustafa, F. Rehman, and N. Khan, "Evaluating zero-shot llms for low-resource script ocr: A case study on sindhi and pashto," in *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. Forthcoming.

[20] Abdul Majid Bhurgri Institute of Language Engineering, "Resources for sindhi computing." http://www.amblesindhi.org.pk, 2023. Government of Sindh.

[21] H. P. P. Win, P. T. T. Khine, and K. N. N. Tun, "Converting myanmar printed document image into machine understandable text format," in *Int. Conf. Digital Information Management*, IEEE, 2011.

[22] E. Kaufmann, A. Bernstein, and R. Zumstein, "Querix: A natural language interface to query ontologies based on clarification dialogs," in *Proc. Int. Semantic Web Conf. (ISWC)*, (Athens, GA), 2006.

[23] D. N. Hakro and A. Z. Talib, "Printed text image database for sindhi ocr," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 15, no. 4, pp. 1–18, 2016.