BN-DRISHTI: Bangla Document Recognition Through Instance-Level Segmentation of Handwritten Text Images

Sheikh Mohammad Jubaer[1], Nazifa Tabassum[1], Md Ataur Rahman[1(✉)], and Mohammad Khairul Islam[2]

[1] Department of CSE, Premier University, Chittagong, Bangladesh
jubaer.puc@gmail.com, nazifa.puc@gmail.com, ataur.cse@puc.ac.bd
[2] Department of CSE, University of Chittagong, Chittagong, Bangladesh
mkislam@cu.ac.bd

Abstract. Handwriting recognition remains challenging for some of the most spoken languages, like Bangla, due to the complexity of line and word segmentation brought by the curvilinear nature of writing and lack of quality datasets. This paper solves the segmentation problem by introducing a state-of-the-art method (`BN-DRISHTI` (**Code and Demo:** https://github.com/crusnic-corp/BN-DRISHTI)) that combines a deep learning-based object detection framework (YOLO) with Hough and Affine transformation for skew correction. However, training deep learning models requires a massive amount of data. Thus, we also present an extended version of the BN-HTRd dataset comprising 786 full-page handwritten Bangla document images, line and word-level annotation for segmentation, and corresponding ground truths for word recognition. Evaluation on the test portion of our dataset resulted in an F-score of 99.97% for line and 98% for word segmentation. For comparative analysis, we used three external Bangla handwritten datasets, namely BanglaWriting, WBSUBNdb_text, and ICDAR 2013, where our system outperformed by a significant margin, further justifying the performance of our approach on completely unseen samples

Keywords: Handwritten Text Recognition (HTR) · Data Annotation · Image Segmentation · Computer Vision · Deep Learning

1 Introduction

Line and word segmentation are one of the most fundamental parts of handwritten document image recognition. As the field of deep learning is maturing at an unprecedented speed, the choice for solving this sort of task employing off-the-shelf deep learning frameworks is getting popular nowadays for its efficiency. However, few attempts have been made to utilize this approach for Bangla handwritten recognition task due to the scarcity of datasets in this domain. Our previous endeavors involved an initial dataset-making process named BN-HTRd (v1.0), comprising of Bangla handwritten document images and only line-level

M. Coustaty and A. Fornés (Eds.): ICDAR 2023 Workshops, LNCS 14194, pp. 195–212, 2023.
https://doi.org/10.1007/978-3-031-41501-2_14

annotations and ground truths for word recognition. However, that dataset was incomplete due to the missing word-level annotation. Therefore, to have a more comprehensive and useable handwritten recognition dataset, we have extended the BN-HTRd (v4.0) dataset[1] by integrating word-level annotations and necessary improvements in the ground truths for the word recognition task.

As segmentation plays a vital role in recognizing handwritten documents, another pivotal *contribution* of this paper is the conglomeration of a state-of-the-art method for segmenting lines and words from transcribed images. Our approach treats the segmentation task as an object detection problem by identifying the distinct instances of similar objects (i.e., lines, words) and demarcating their boundaries. Thus in a way, we are performing `instance-level segmentation` as it is particularly useful when homogeneous objects are required to be considered separately. To do so, we partially rely on the YOLO (You Only Look Once) framework. However, the success of our method is more than just the training of the YOLO algorithm. In order to get the perfect words segmented from possibly complex curvilinear text lines, we had to improvise our approach to retrieve the main handwritten text lines correctly by removing other unnecessary elements. For that, we used a combination of the Hough and Affine transform methods. The Hough transform predicts the skew angles of the main handwritten text lines, and the Affine transform rotates them according to the expected gradients, making them straight horizontally. Therefore, the word segmentation approach provides much better results compared to the segmentation on skewed lines. Thus, the main contributions of this paper are threefold:

1. Introducing a straightforward `novel hybrid approach`, for instance-level handwritten document segmentation into corresponding lines and words.
2. Achieved *state-of-the-art* (`SOTA`) scores on three different prominent Bangla handwriting datasets for line/word segmentation tasks.
3. Set a new `benchmark` for the BN-HTRd dataset. Also, `extended`[2] it to be one of the largest and the most comprehensive Bangla handwritten document image segmentation and recognition dataset by adding `200k+` annotations.

2 Related Work

CMATERdb [21] is one of the oldest character-level datasets consisting of 150 Bangla handwritten document images distributed among two versions. Another prominent character-level dataset having 2000 handwritten samples named **BanglaLekha-Isolated** [5] contains 166105 handwritten characters written by an age group of 6 to 28. **Ekush** [15], which is a multipurpose dataset, contains 367,018 isolated handwritten characters written by 3086 individual writers. The authors also benchmarked the dataset using a multilayer CNN model (**EkushNet**) for character classification, achieving an accuracy of 97.73% on their dataset while scoring 95.01% in the external **CMATERdb** dataset.

[1] **Extended Dataset:** https://data.mendeley.com/datasets/743k6dm543.
[2] **Changes:** https://data.mendeley.com/v1/datasets/compare/743k6dm543/4/1.

A paragraph-level dataset that resembles our dataset in terms of word-level annotation is the **BanglaWriting** [12] dataset, which includes single-page handwriting comprising 32,787 characters, 21,234 words, and 5,470 unique words produced by 260 writers of different ages and personalities. Another paragraph-level unannotated dataset **WBSUBNdb_text** [10], consisting of 1383 handwritten Bangla scripts having around 100k words, was collected from 190 transcribers for the writer identification task. While in terms of a document-level dataset, mostly resembling our own, **ICDAR 2013** [22] handwriting segmentation contests dataset comes with 2649 lines and 23525 word-level annotations for 50 handwritten document images on Bangla.

Segmenting handwritten document images in terms of lines and words is the most crucial part when it comes to end-to-end handwritten document image recognition. In ***Projection-based*** methods [8,9,13,14], the handwritten lines are obtained by computing the average distance between the peaks of the projected histogram. A method based on the skew normalization process is proposed in [3]. ***Hough-based*** methods [9] represent geometric shapes such as straight lines, circles, and ellipses in terms of parameters to determine geometric locations that suggest the existence of the desired shape. The author of [8] presented a skew correction technique for handwritten Arabic document images using their optimized randomized Hough transform, followed by resolving the primary line for segmentation. For layout analysis, ***Morphology-based*** approaches [7,9] have been used along with piece-wise painting (PPA) algorithms [2], to segment script independent handwritten text lines. In contrast, ***Graph-based*** approaches [9,11,23] compactly represent the image structure by keeping the relevant information on the arrangement of text lines. ***Learning-based*** techniques recently became popular for segmenting handwritten text instances. The authors of [4,19,20,24] used a Fully Convolutional Network (FCN) for this purpose. A model based on the modified multidimensional long short-term memory recurrent neural networks (**MDLSTM RNNs**) was proposed in [6]. An unsupervised ***clustering*** approach [16] was utilized for line segmentation which achieved an F-score of 81.57% on the BN-HTRd dataset.

A series of consistent recent works on **Bangla handwriting segmentation** [1,17,18] is carried out by a common research team that also developed the WBSUBNdb_text dataset. Their technique predominantly relies on the projection profile method and connected component analysis. They initially worked on a tri-level (line/word/character) segmentation [17] while their latest works are focused solely on word [1] and line segmentation [18]. Moreover, in [18], the method serves the line segmentation on multi-script handwritten documents while the other two research only work for the Bangla scripts.

Our work can be categorized as a **Hybrid Approach** for segmenting lines and words. Our supervised models employ YOLO deep learning framework to predict lines and words from handwritten document images. We used the Hough Line Transform to measure the segmented line's skew angle, then corrected it with Affine Transform. These combinations were never used in the literature for Bangla handwritten recognition tasks.

3 Dataset

Data annotation is one of the most crucial parts of the dataset curation process where supervised learning is concerned. As a primary text source, we considered the BBC Bangla News platform since it does not require any restrictions and has an open access policy. Hence, we downloaded various categories of news content as files in TEXT and PDF format, renamed files according to the sequence of 1 to 237, and put them in separate folders. We distributed those 237 folders among 237 writers of different ages, disciplines, and genders. They were instructed to write down the text file's contents in their natural writing style and to take pictures of the pages afterward. This resulted in 1,591 handwritten images in total. Due to the complexity of the task, we were only able to recruit a total of 75 individuals to annotate lines of assigned handwritten images using an annotation tool called `LabelImg`. As a result, we were only able to annotate a maximum of 150 folders. The resultant annotation produced YOLO and PASCAL VOC formatted ground truth for line segmentation. These 150 folders of handwritten images and their line annotations were included in the first version of the BN-HTRd dataset [16]. For the purpose of word segmentation, we have extended the dataset (v4.0) by adding bounding-box annotations of individual words for all the annotated lines. We also organized each word of the text file into separate rows in Excel in order to create the ground truth Unicode representation of the corresponding word's images for recognition purposes in the future.

We used this extended BN-HTRd dataset containing annotations in 150 folders to develop and test our system. It contains a total of 786 handwritten images comprising 14,383 lines and 1,08,181 words. The rest of the unannotated 87 folders were automatically annotated using our system, resulting in an additional 14,836 lines and 1,06,135 words, which we denoted as Automatic Annotations. For the purpose of experimental evaluation, we split the 150 folders into two subsets and took one image from each of the folders for either validation or testing (resulting in 75 images for each subset). The rest of the 636 images were used for training purposes. Table 1 below shows this subdivision.

Table 1. Distribution of extended BN-HTRd (v4.0) dataset for experimentation. (**Splitted Dataset:** https://doi.org/10.57967/hf/0546)

Type	Purpose	Train	Valid	Test	Total
Doc. Images	Line Segmentation	636	75	75	786
Line Images	Word Segmentation	11,471	1,515	1,397	14,383
Word Images	Word Recognition	86,055	11,712	10,414	1,08,181

4 Proposed Methodology

We have broken down our overall system architecture in Fig. 1, which consists of six parts. Those six parts cover the overall process of how our system functions.

Before dissecting those parts in detail in the later sections (4.1–4.5), we will provide a brief overview in the following:

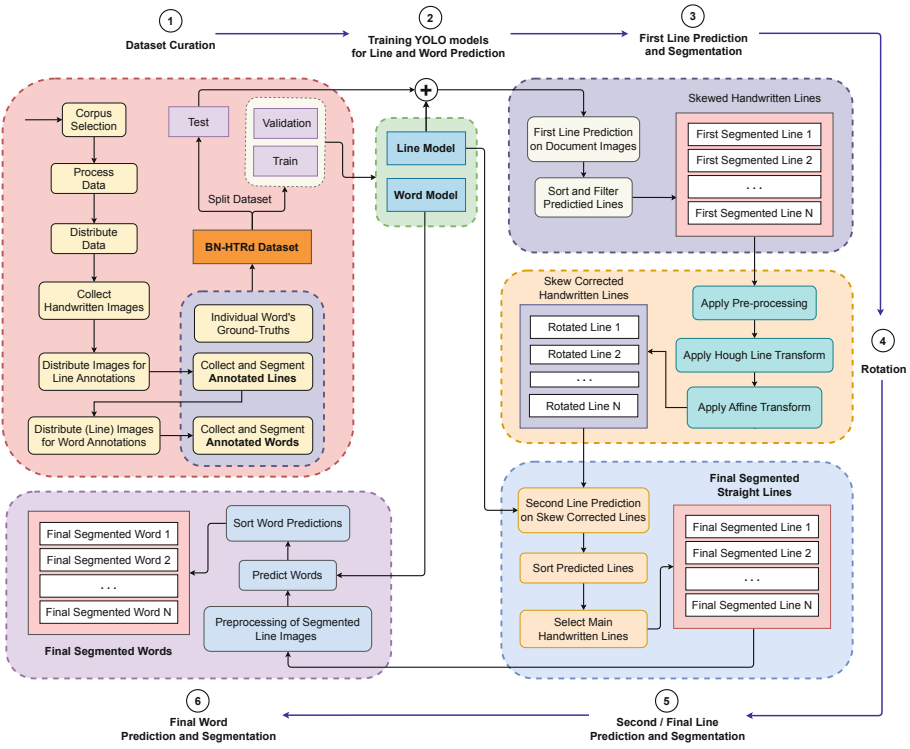

Fig. 1. Overall System Architecture for BN-DRISHTI.

- Our efforts in making and extending the BN-HTRd dataset involved various development processes such as distributing the data to the writers, manual annotations, and making it compatible with supervised learning methods such as ours (details in Sect. 3).
- Although training the models is a crucial part of any supervised system, it was not enough in our case despite YOLO being one of the best frameworks. It was predicting redundant lines, which we had to eliminate in order to get better segmentation scores (details in Sects. 4.1 and 4.2).
- As we were also getting some unnecessary lines along with the target line, a better line segmentation method is essential to segment the words correctly. To remove them, we rotated the curvilinear lines using the Hough-Affine transformation and corrected their skewness (details in Sect. 4.3).
- We applied the final/second YOLO line prediction on the skew-corrected lines, followed by some post-processing in order to extract the main handwritten line (details in Sect. 4.4).

– Finally, word prediction and segmentation are performed on skew-corrected final segmented line images using the word model (details in Sect. 4.5).

4.1 Training Models

YOLOv5x (XLarge) model architecture having a default SGD optimizer was used to train both our Line and Word models for 300 epochs. We used document images with line annotations to train the initial line segmentation model. In contrast, line images and their word annotations were used to train the Word model. The training was done using an NVIDIA RTX 3060 Laptop GPU containing 6 GB GDDR6 memory and 3840 CUDA cores.

4.2 First-Line Prediction and Segmentation

The line detection is performed on document images without pre-processing or resizing; some output samples are shown in Fig. 2a. YOLO generates a TEXT file for each document image, representing each predicted line as *<class_id, x, y, width, height, confidence>* without particular order. The confidence threshold during prediction is set to 0.3 to include lines with few words or a single word that was initially missing. However, this approach resulted in both unnecessary line predictions and correct ones with confidence below 0.5. To address this, the output is sorted based on the y-axis attribute, and unnecessary bounding boxes having unusual heights but lower confidence that encompasses or overlaps with one or more boxes are filtered out, resulting in filtered first-line predictions (Fig. 2b). The filtered predicted lines are then extracted using their YOLO attributes: *< x, y, width, height >*. Figure 2c illustrates the process of first-line detection, filtering, and corresponding segmentation.

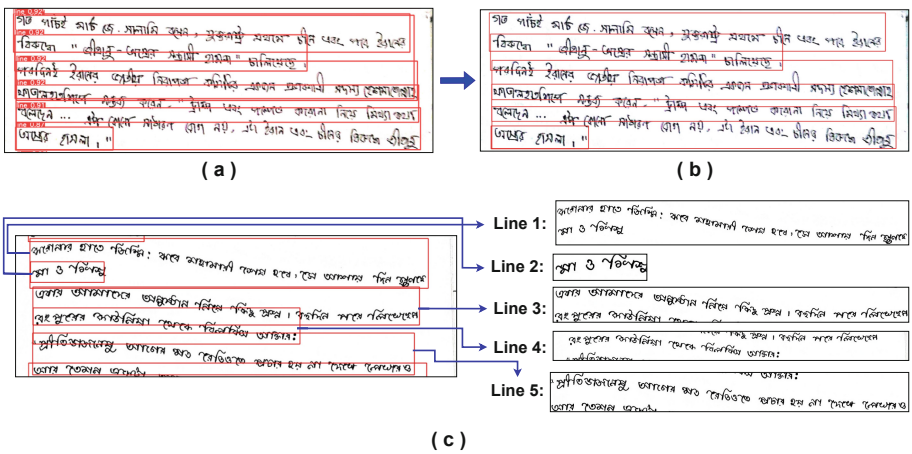

Fig. 2. Representation of First-line prediction and segmentation, where a) sample image with first-line prediction containing multiple unnecessary predictions, b) filtered first-line prediction, and c) another sample image with filtered first-line prediction and segmentation for curvilinear handwriting.

4.3 Rotation (skew Estimation and Correction)

After analyzing our first segmented line images, we found out that, with the main handwritten line, we are also getting some unwanted lines at the top or bottom due to the skewness of the lines and the rectangular shape of the predicted bounding box. Therefore, the skew correction over the first line prediction is important in order to retrieve the main handwritten line. We denoted this process as *Rotation*, which is performed by applying the Hough line and Affine transform. We have represented the overall rotation process in Fig. 3.

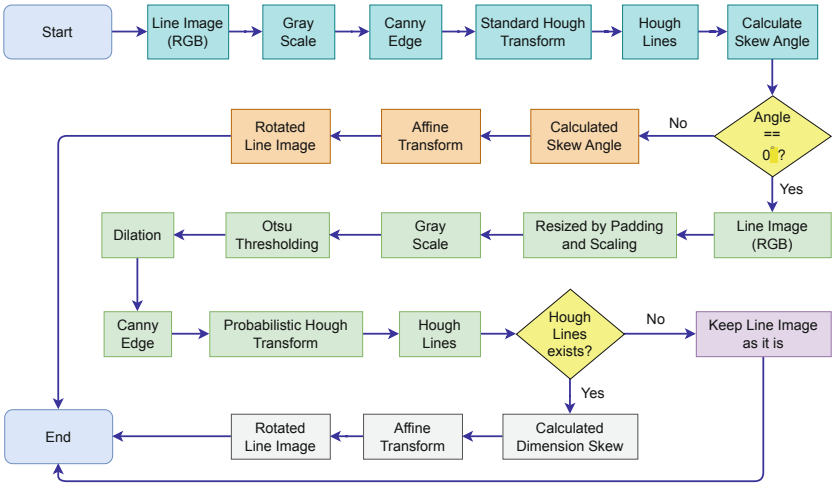

Fig. 3. Flowchart of skew estimation and correction over the first predicted lines.

4.3.1 Skew Estimation: We categorized handwritten lines' skew into two types: Positive and Negative (shown in Fig. 4). The skew angle estimation is performed in two phases:

1. Line Skew (LSkew) Estimation: where we applied the Standard Hough Transform (SHT).
2. Dimension Skew (DSkew) Estimation: where we applied the Probabilistic Hough Transform (PHT).

LSkew: In the Bangla writings, each word consists of letters and the letters are often connected by a horizontal line called 'mātrā'. By connecting those horizontal lines above the words using SHT, we construct straight lines, which we denote as Hough lines. Using those Hough lines, we estimate the skew angle of the main handwritten line. In terms of the representation of LSkew (Fig. 4), if the detected Hough lines have positive skew, the estimated skew angle will be negative; otherwise positive. We illustrate this LSkew estimation process in

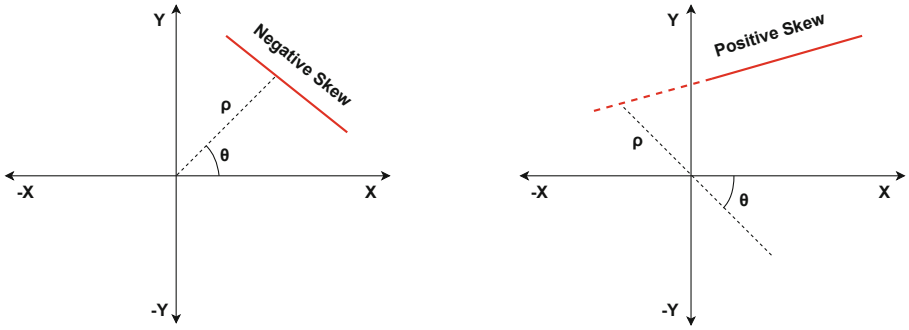

Fig. 4. Representation of Hough lines using equation $\rho = x^*\cos\theta + y^*\sin\theta$; where θ is the angle of the detected line and ρ is the distance from x-axis.

Fig. 5 by taking two samples of segmented line images, where one got positive skew, and the other got negative skew.

The SHT is applied to get the Hough lines by connecting the adjacent edge points of the main handwritten line's words, represented in Fig. 5 (top). Consequently, we calculated the average of all the detected Hough lines' parameters and considered this value to be the best detected Hough line. Figure 5 (bottom) represents the average skew angle (θ_{avg}), which is the optimal skew angle of our best detected Hough line.

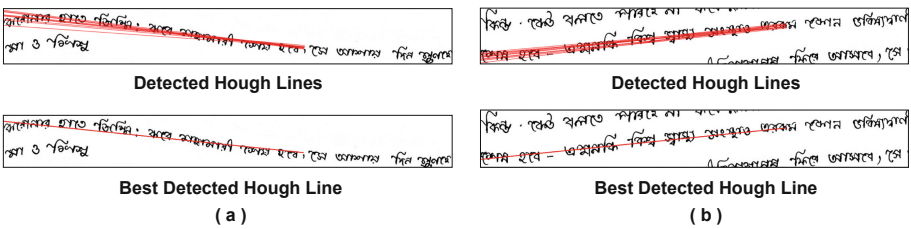

Fig. 5. Detected (top) and the Best Detected (bottom) Hough Lines, where the main handwritten line contains, a) Positive Skew, and b) Negative Skew.

DSkew: In some cases, SHT fails to detect the Hough lines, despite the main handwritten line on those segmented images being well skewed. We identified that the dimension of those failed images is too small compared to the standard dimension of the line images where SHT works. Moreover, in most cases, those line images contain only a few words, in such cases, not requiring any skew correction. Therefore, we opt for the DSkew process by applying PHT. We perform up-scaling on those failed images by preserving the aspect ratio before applying PHT (shown in Fig. 6).

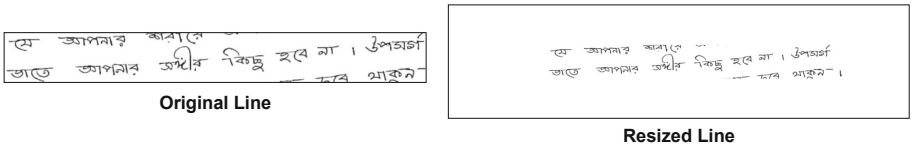

Fig. 6. Changing the dimension of nonstandard line image before applying PHT.

We apply some preprocessing steps such as image binarization and morphological operation with a 3×3 kernel to make the objects' lines and overall shape thicker and sharper. Finally, the canny edge detection method is applied before we can use the PHT. The output of preprocessing steps can be seen in Fig. 7.

Fig. 7. Preprocessing and Hough line detection of sample resized line image represented in Fig. 6; where a) Binarization, b) Morphological Dilation, c) Canny edge detection, and d) Detected Hough lines using PHT.

The PHT not only joins the 'matra' of words but also connects any subsets of the points of each word edges individually if there is any potential Hough line. We named it dimension skew or DSkew, as each word component in the image takes part in the skew estimation process. Like SHT, we also get the typical Hough line parameters such as (x_1, y_1), (x_2, y_2), and (ρ, θ) in PHT. Hence, we applied the PHT in the edge detected image (of Fig. 7c) and got the Hough lines detected, shown in Fig. 7d. As the process detects multiple Hough lines for almost every word, therefore, each line has many θ, which we denote as *Degree*. To obtain the optimal skew angle of that image, we perform a voting process by dividing the xy space into six cases to determine where the maximum detected Hough lines had fallen. We then take an average of those lines' parameters to find the average of Degree ($Degree_{avg}$) and consider this as the skew angle of the detected Hough lines by PHT. The six cases of the voting process and their outcomes are given in Table 2:

Table 2. Voting process of DSkew with their categories and outcomes.

Based On	Voting Categories	Detected Hough Line Types	Final outcome as an average of degrees
Coordinates	x_1 equals x_2	Vertical	Return $Degree_{avg}$ as 90°
	y_1 equals y_2	Straight	Return $Degree_{avg}$ as 0°
Quadrants	$-45° \leq$ Degree $\leq 0°$	Positive Skew	Return $Degree_{avg}$
	$-90° \leq$ Degree $< -45°$	Negative Skew	Return $Degree_{avg}$
	$0° <$Degree $\leq 45°$	Negative Skew	Return $Degree_{avg}$
	$45° <$Degree $\leq 90°$	Positive Skew	Return $Degree_{avg}$

4.3.2 Skew Correction: In order to correct the estimated skew of our segmented lines, we rotate them using the Affine Transform (AT) relative to the center of the image. The rotation process for LSkew and DSkew is as follows:
LSkew: After estimating the optimal skew angle (θ_{avg}) using LSkew, we rotate the image with that skew angle through AT using the following two conditions:

1. If the value of θ_{avg} is Negative, we rotate the image Clockwise.
2. If the value of θ_{avg} is Positive, we rotate the image Anti-Clockwise.

Figure 8 illustrates the skew-correction for the segmented lines of Fig. 5.

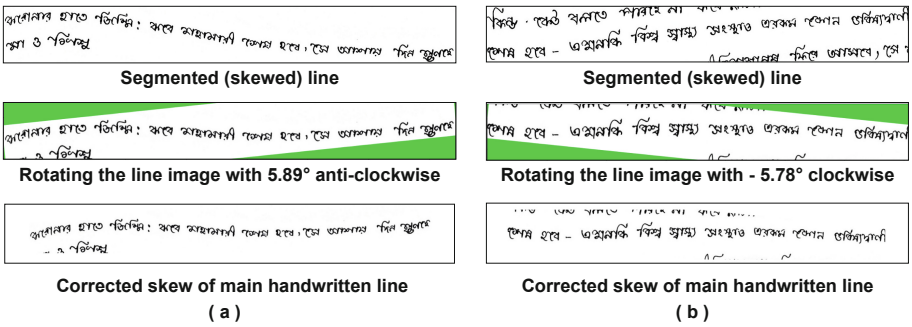

Fig. 8. Skew correction of segmented lines using AT where the original line was, a) Negative Skewed (rotated anti-clockwise); and b) Positive Skewed (rotated clockwise).

DSkew: The rotation for DSkew correction is similar to the rotation for LSkew correction, but the process of finding the optimal degree for rotation is different. Here, we calculate the optimal skew angle (θ_{avg}) based on the estimated $Degree_{avg}$ from DSkew. Then according to θ_{avg}, we rotate the image using AT by following the four conditions listed in Table 3:

Table 3. Conditions for skew correction for the process of DSkew.

No.	Conditions	Optimal Skew (θ_{avg})	Rotation
1	$-45^{\circ} \leq Degree_{avg} \leq 0^{\circ}$	$\theta_{avg} = Degree_{avg}$	Clockwise
2	$-90^{\circ} \leq Degree_{avg} < -45^{\circ}$	$\theta_{avg} = Degree_{avg} + 90^{\circ}$	Anti-clockwise
3	$0^{\circ} < Degree_{avg} \leq 45^{\circ}$	$\theta_{avg} = Degree_{avg}$	Anti-clockwise
4	$45^{\circ} < Degree_{avg} \leq 90^{\circ}$	$\theta_{avg} = Degree_{avg} - 90^{\circ}$	Clockwise

4.4 Final/Second Line Prediction and Segmentation

Final or second line prediction is applied on the skew-corrected lines to retrieve the main handwritten lines by eliminating the unwanted lines. Before that, we trim down each side of the DSkewed line image by a little portion to avoid unnecessary word prediction. Here, we consider a confidence threshold of 0.5. We also follow a selection process when we have multiple lines even after the second line prediction, as described below:

1. **The number of line predictions is one:** In this case, we segment the line with the given bounding box attributes, like in Fig. 9. If the width of the predicted line is less than 40% of the image width, we keep it as it is.

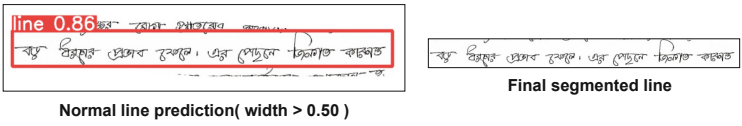

Fig. 9. Line image with single line prediction and segmentation.

2. **The number of line predictions is two:** In this case, normally, we segment the line prediction with maximum widths, like in Fig. 10. But, if both the predicted line's width is less than 50% of the image width, then we check their confidence and segment the line with maximum confidence. Otherwise, we keep the image as it is.

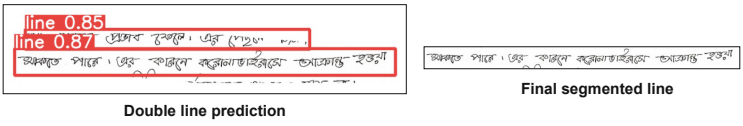

Fig. 10. Line image with two line prediction and segmentation (usual case).

3. **The number of line predictions is three:** In this case, we segment the line which stays in the middle like in Fig. 11.

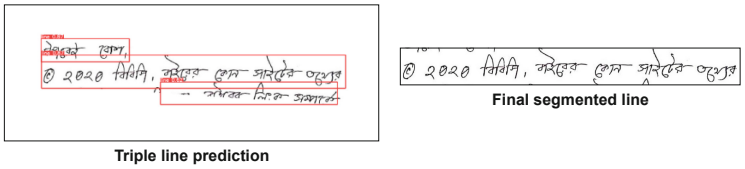

Fig. 11. Line image with two line prediction and segmentation.

4. **The number of line predictions is more than three:** Unseen cases where we select and segment the line having maximum width.

As the segmented lines have passed through the pre-processing, rotation, and final line segmentation process, we now have our final lines segmented from the handwritten document images. Note that, we also keep track of the predicted *line numbers* within the document for future recognition purposes. Figure 12 illustrates the resultant final line segmentation of the lines represented in Fig. 5.

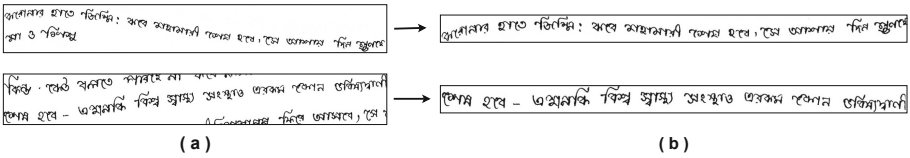

Fig. 12. a) Initial line segmentation by YOLO containing mostly curvilinear or skewed handwritten lines with noises, and b) Final segmented lines by our line segmentation approach, which are straight and without any unnecessary lines.

4.5 Word Prediction and Segmentation

We perform word prediction on the Final segmented lines by directly employing our custom YOLO word model, where we set the confidence threshold to be 0.4. We also sort the predictions based on the horizontal axis of the lines in order to get the position of a particular word in that line for future recognition purposes. Figure 13 illustrates word prediction and segmentation from the running example.

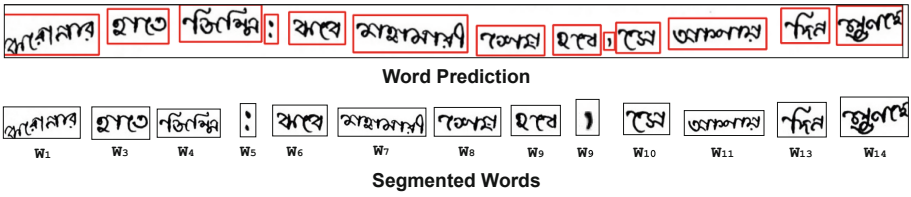

Fig. 13. Word prediction and segmentation on skew corrected final segmented lines; where W_i is the i^{th} word within the line.

5 Experimental Results

In this section, we evaluate the efficiency of our line and word segmentation approach on the BN-HTRd dataset. We will also compare our results with an unsupervised line segmentation approach of BN-HTR_LS system [16].

5.1 Evaluation Matrices

Two bounding boxes (lines) are considered a one-to-one match if the total matching pixels exceed or equal the evaluator's approved threshold (T_a). Let N be the number of ground-truth elements, M be the count of detected components, and $o2o$ be the number of one-to-one matches between N and M; the Detection Rate (DR) and Recognition Accuracy (RA) are equivalent of Recall and Precision. Combining these, we can get the final performance metric FM (similar to F-score) using the equation below:

$$DR = \frac{o2o}{N}, \quad RA = \frac{o2o}{M}, \quad FM = \frac{2DR * RA}{DR + RA} \tag{1}$$

5.2 Line Segmentation

For the evaluation of our BN-DRISHTI line segmentation approach, we first did the **Quantitative analysis** on the test set of 75 handwritten document images from the BN-HTRd dataset containing 1397 (N) manually annotated ground truth lines. Our segmentation approach's final line prediction was 1396 (M). Among those, the number of $o2o$ matches was 1314. However, by using only YOLO trained model, we got 1433 (M) which implies that YOLO predicted 37 more redundant lines as compared to our approach, making our approach much superior. These results are listed in rows 2–3 of Table 4.

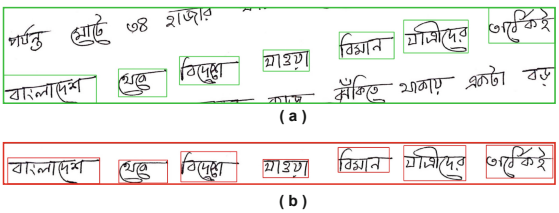

Fig. 14. a) Ground truth annotation on skewed line; Vs. b) Prediction on straight line.

After analyzing the line's ground truth and prediction bounding boxes visually (see Fig. 14), we came to the conclusion that the overlap between them for each line is not quite accurate since we performed skew correction before segmenting the line images. Thus, in automatic or quantitative evaluation the results we are getting are not as significant as we were expecting, since almost

every line of the document images was segmented perfectly. Hence, we decided to do a **Qualitative evaluation** by going through all the ground truth and predictions manually to find the *o2o* for each handwritten document. And the overall *o2o* match was 1396, which is equal to the final line predictions we were getting. In Table 4 we put together the relative performance of our line segmentation approach's (`BN-DRISHTI`) quantitative and qualitative analysis as compared to the unsupervised approach of `BN-HTR_LS` system[3] [16] where they only performed line segmentation on the same dataset.

Table 4. Comparison of line segmentation results on BN-HTRd test sets.

Approaches	N	M	o2o	DR(%)	RA(%)	FM(%)
BN-HTR_LS [16]	2915	3437	2591	88.88	75.38	81.57
YOLO line model	1397	1433	1314	94.06	91.7	92.86
BN-DRISHTI (Quantitative)	1397	1396	1314	94.06	94.13	94.09
BN-DRISHTI (Qualitative)	1397	1396	1396	**99.93**	**1.00**	**99.97**

5.3 Word Segmentation

For this experiment, we used 10,414 manually annotated ground truth words within the line images of the test set's 75 handwritten documents. Our word model predicted 10,348 words. Table 5 shows the score of **Quantitative** analysis.

Table 5. Quantitative evaluation of our word segmentation on BN-HTRd test sets.

Ground Truths	Prediction	DR (%)	RA (%)	FM (%)
10,414	10,348	15.2	17.7	16.0

Again for the same aforementioned reason, the quantitative evaluation does not do justice to our approach's true word segmentation capabilities. Hence, we visually compared the ground truths against our predictions and found that the position of the words bounding box has changed drastically due to the changes in image dimension during our line segmentation approach, as illustrated in Fig. 14. This occurred because the original ground truth annotation was on the skewed lines, and our word prediction was done on the skew-corrected straight lines. Thus, after analyzing the ground truth and prediction bounding boxes, we came to the conclusion that the evaluation will not be fair if done automatically. Therefore, we again opt for a manual **Qualitative** analysis. We show both the quantitative and qualitative results in Table 6.

[3] **BN-HTR_LS Codebase:** https://github.com/shaoncsecu/BN-HTR_LS.

Table 6. Results of our word segmentation approach on the original ground truth (skewed) vs. skew-corrected (straight) lines from the BN-HTRd test sets.

Analysis	Word prediction on	N	M	DR	RA	FM
Quantitative	First segmented (Skewed) line	10,414	10,383	0.39	0.45	0.42
Quantitative	Final segmented (Straight) line	10,414	10,348	0.15	0.17	0.16
Qualitative	Final segmented (Straight) line	10,414	10,348	0.98	0.98	**0.98**

In Table 6, the qualitative analysis results perfectly justify our systems word segmentation capabilities. We also emphasize that word segmentation is far more precise when combined with our skew correction strategy.

5.4 Comparative Analysis

ICDAR 2013 Dataset[22]**:** This handwriting segmentation contests dataset contains 50 images for Bangla. As ground truth (N), we got 879 lines and 6,711 words; against which our system segmented 874 lines and 6,667 words (M). We choose team Golestan-a, Golestan-b, and INMC for performance comparison, as the Golestan method outperforms all other contestants with an overall score (SM) of 94.17%. And for Line segmentation, the INMC method was on the top with a 98.66% FM score. The comparison in Table 7 indicates that our system outperforms Golestan and INMC team's SM scores by a good margin. While our word segmentation results absolutely smashed the competitors, the line segmentation score was only second to INMC by a narrow margin.

Table 7. Comparison among top teams of ICDAR 2013 and our BN-DRISHTI system.

Systems	Class	N	M	o2o	DR (%)	RA (%)	FM (%)	SM (%)
Golestan-a	Lines	2649	2646	2602	98.23	98.34	98.28	94.17
	Words	23525	23322	21093	89.66	90.44	90.05	
Golestan-b	Lines	2649	2646	2602	98.23	98.34	98.23	90.06
	Words	23525	23400	21077	89.59	90.07	89.83	
INMC	Lines	2649	2650	2614	98.68	98.64	**98.66**	93.96
	Words	23525	22957	20745	88.18	90.36	89.26	
BN-DRISHTI	Lines	879	874	863	98.18	98.74	98.46	**96.65**
	Words	6711	6677	6348	98.74	95.07	**94.83**	

BanglaWriting Dataset [12]**:** It comprises 260 full-page Bangla handwritten documents and only the words ground truth. We manually evaluated the word segmentation results using randomly selected 50 document images from this dataset, as the word annotation was done directly over the document without

any intermediate line annotation. Those selected 50 images contain 4409 words, and our system correctly segmented 4186 words against them. Table 8 indicates how our system performed on the BanglaWriting dataset.

Table 8. Word segmentation results on fifty images of BanglaWriting dataset.

Task	N	M	o2o	DR (%)	RA (%)	FM (%)
Word Segmentation	4409	4219	4186	94.9	99.2	97.0

WBSUBNdb_text Dataset [10]**:** This publicly available dataset has been used by two of the most prominent line [18] and word [1] segmentation methods for evaluation. As it contains 1352 Bangla handwriting without any ground truth, we only performed a qualitative analysis similar to the settings mentioned in those papers. We positioned our approach against these systems in Table 9.

Table 9. Comparison of segmentation results based on WBSUBNdb_text dataset.

Systems	Class	DR (%)	RA (%)	FM (%)
WBSUBNdb	Lines [18]	96.99	97.07	97.02
	Words [1]	86.96	93.25	90.0
BN-DRISHTI	Lines	99.27	99.44	**99.35**
	Words	96.85	97.18	**97.01**

6 Conclusions

The main contribution of this research is the significant improvement in line and word segmentation for Bangla handwritten scripts, which lays the foundation of our envisioned Bangla Handwritten Text Recognition (HTR). To alleviate the shortage of Bangle document-level handwritten datasets for future researchers, we have extended our BN-HTRd dataset. Currently, it is the largest dataset of its type with line and word-level annotation. Moreover, keeping the recognition task in mind, we have stored the words' Unicode representation against their position in the ground truth text. The main recipe behind our approach's overwhelming success is a two-layer line segmentation technique combined with an intricate skew correction in the middle. Our proposed line segmentation approach has achieved a near-perfect benchmark evaluation score in terms of F measure (99.97%) compared to the unsupervised approach (81.57%) of BN-HTR_LS [16]. The word segmentation technique also achieved an impressive score (98%) on the skew-corrected lines by our system compared to the skewed lines. Furthermore, we have compared our method against the previous SOTA systems on three of the most prominent Bangla handwriting datasets. Our approach outperformed

all those methods by a significant margin, making our BN-DRISHTI system a new state-of-the-art for Bangla handwritten segmentation task. We aim to expand our work by integrating supervised word recognition to build an "End-To-End Bangla Handwritten Image Recognition system".

References

1. Agarwal, K., Mantry, A., Halder, C.: Word segmentation of offline handwritten Bangla text lines. In: Mandal, J.K., Buyya, R., De, D. (eds.) Proceedings of International Conference on Advanced Computing Applications. AISC, vol. 1406, pp. 551–560. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-5207-3_46
2. Alaei, A., Pal, U., Nagabhushan, P.: A new scheme for unconstrained handwritten text-line segmentation. Pattern Recogn. **44**(4), 917–928 (2011)
3. Bal, A., Saha, R.: An improved method for text segmentation and skew normalization of handwriting image. In: Sa, P.K., Sahoo, M.N., Murugappan, M., Wu, Y., Majhi, B. (eds.) Progress in Intelligent Computing Techniques: Theory, Practice, and Applications. AISC, vol. 518, pp. 181–196. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-3373-5_18
4. Barakat, B., Droby, A., Kassis, M., El-Sana, J.: Text line segmentation for challenging handwritten document images using fully convolutional network. In: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 374–379. IEEE (2018)
5. Biswas, M., et al.: Banglalekha-isolated: a multi-purpose comprehensive dataset of handwritten Bangla isolated characters. Data Brief **12**, 103–107 (2017)
6. Bluche, T.: Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
7. Boudraa, O., Hidouci, W.K., Michelucci, D.: An improved skew angle detection and correction technique for historical scanned documents using morphological skeleton and progressive probabilistic hough transform. In: 2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B), pp. 1–6. IEEE (2017)
8. Boukharouba, A.: A new algorithm for skew correction and baseline detection based on the randomized Hough transform. J. King Saud Univ. Comput. Inf. Sci. **29**(1), 29–38 (2017)
9. Fernández-Mota, D., Lladós, J., Fornés, A.: A graph-based approach for segmenting touching lines in historical handwritten documents. Int. J. Doc. Anal. Recog. (IJDAR) **17**(3), 293–312 (2014). https://doi.org/10.1007/s10032-014-0220-0
10. Halder, C., Obaidullah, S.M., Santosh, K., Roy, K.: Content independent writer identification on Bangla script: a document level approach. Int. J. Pattern Recogn. Artif. Intell. **32**(09), 1856011 (2018)
11. Kumar, J., Kang, L., Doermann, D., Abd-Almageed, W.: Segmentation of handwritten textlines in presence of touching components. In: 2011 International Conference on Document Analysis and Recognition, pp. 109–113. IEEE (2011)
12. Mridha, M.F., Ohi, A.Q., Ali, M.A., Emon, M.I., Kabir, M.M.: Banglawriting: a multi-purpose offline Bangla handwriting dataset. Data Brief **34**, 106633 (2021)
13. Mullick, K., Banerjee, S., Bhattacharya, U.: An efficient line segmentation approach for handwritten Bangla document image. In: 2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR), pp. 1–6. IEEE (2015)

14. Nicolaou, A., Gatos, B.: Handwritten text line segmentation by shredding text into its lines. In: 2009 10th International Conference on Document Analysis and Recognition, pp. 626–630. IEEE (2009)
15. Rabby, A.K.M.S.A., Haque, S., Islam, M.S., Abujar, S., Hossain, S.A.: Ekush: a multipurpose and multitype comprehensive database for online off-line Bangla handwritten characters. In: Santosh, K.C., Hegadi, R.S. (eds.) RTIP2R 2018. CCIS, vol. 1037, pp. 149–158. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-9187-3_14
16. Rahman, M.A., Tabassum, N., Paul, M., Pal, R., Islam, M.K.: BN-HTRd: A Benchmark Dataset for Document Level Offline Bangla Handwritten Text Recognition (HTR) and Line Segmentation. In: Computer Vision and Image Analysis for Industry 4.0, pp. 1–16. CRC Press 6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487–2742 (2023)
17. Rakshit, P., Halder, C., Ghosh, S., Roy, K.: Line, word, and character segmentation from Bangla handwritten text-a precursor toward Bangla HOCR. Adv. Comput. Syst. Secur. **5**, 109–120 (2018)
18. Rakshit, P., Halder, C., Sk, M.O., Roy, K.: A generalized line segmentation method for multi-script handwritten text documents. Expert Syst. Appl. **212**, 118498 (2023)
19. Renton, G., Chatelain, C., Adam, S., Kermorvant, C., Paquet, T.: Handwritten text line segmentation using fully convolutional network. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 5, pp. 5–9. IEEE (2017)
20. Renton, G., Soullard, Y., Chatelain, C., Adam, S., Kermorvant, C., Paquet, T.: Fully convolutional network with dilated convolutions for handwritten text line segmentation. Int. J. Docu. Anal. Recogn. (IJDAR) **21**(3), 177–186 (2018). https://doi.org/10.1007/s10032-018-0304-3
21. Sarkar, R., Das, N., Basu, S., Kundu, M., Nasipuri, M., Basu, D.K.: Cmaterdb1: a database of unconstrained handwritten Bangla and Bangla-English mixed script document image. Int. J. Docu. Anal. Recogn. (IJDAR) **15**, 71–83 (2012)
22. Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., Alaei, A.: ICDAR 2013 handwriting segmentation contest. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1402–1406. IEEE (2013)
23. Surinta, O., Holtkamp, M., Karabaa, F., Van Oosten, J.P., Schomaker, L., Wiering, M.: A path planning for line segmentation of handwritten documents. In: 2014 14th International Conference on Frontiers in Handwriting Recognition, pp. 175–180. IEEE (2014)
24. Vo, Q.N., Lee, G.: Dense prediction for text line segmentation in handwritten document images. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 3264–3268. IEEE (2016)