Multimedia Tools and Applications (2021) 80:27323–27342
https://doi.org/10.1007/s11042-021-10822-2

A novel skew correction methodology for handwritten words in multilingual multi-oriented documents

Rahul Pramanik[1] · Soumen Bag[2]

Received: 25 October 2019 / Revised: 13 January 2021 / Accepted: 10 March 2021 /
Published online: 27 May 2021

Abstract

Multi-oriented handwritten documents require additional preprocessing for segmentation and subsequent phases to work accurately in handwritten recognition systems. Skew correction is one such additional phase. Appearance of skew in multi-oriented Indian language based handwritten document is higher due to the presence of cursive nature. In the current work, we utilise a salient feature present in Indian scripts called *mātrā* (also known as headline), extract a group of eligible pixels, and employ linear curve fitting for detecting and correcting skew in handwritten words. The proposed method is capable of correcting skew in four distinct Indian languages, viz. Bangla, Hindi, Marathi, and Punjabi. It is capable of efficiently handling skewed word images to an extent of ±55° and delivers precise result even when the *mātrā* is mostly absent or discontinuous.

Keywords Curve fitting · Indian regional language · Multilingual · Skew detection · Skew estimation

1 Introduction

Image processing domain has come a long way and as time has passed each of its various research domains [18, 36, 37, 40] brought novel challenges [19, 38, 39]. Handwriting recognition is one such domain. Handwriting recognition has become popular from the past decade due to the need of automation of handwritten forms and texts that are being filled everyday at railway counters, banks, post offices, educational institutions, and for application of jobs. The drastic need for performing research and development in various

✉ Rahul Pramanik
rahul@dubai.bits-pilani.ac.in

Soumen Bag
soumen@iitism.ac.in

[1] Department of Computer Science, BITS Pilani, Dubai Campus, PO Box 345055, Dubai, UAE

[2] Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, Dhanbad - 826004, India

aspects of a handwriting recognition system is severe, especially for a developing and multilingual country like India. A handwritten character recognition system encompasses four components, viz. preprocessing, segmentation, feature detection and extraction, and classification [8, 26]. The preprocessing component hones the text, utilising various strategies in order to increase the accuracy at the segmentation level. Due to different reasons, like feeding document into a scanner, or sometimes individuals tend to add skew in text (downwards or upwards) while writing, etc., skew present in most documents can vary between ±30°. Most of the segmentation algorithms work best in ideal cases, i.e., when no skew is present. Presence of a skew estimation and correction strategy at the preprocessing phase helps segmentation algorithms to achieve better result. An example of presence of skew in handwritten text is shown in Fig. 1.

Skew estimation and correction is addressed at two levels:

- For correcting skew present in text lines of a handwritten text document,
- For correcting skew present in words of a text line extracted from a multi-oriented handwritten text document.

A diverse amount of methodologies have been presented in the literature to estimate and correct skew present at text line level of document images [2, 5, 6, 15, 22, 31, 33, 41]. But, the same has not reflected at word level. Presence of cursive nature makes the task difficult for Indian regional languages [1]. Also, the stroke styles of characters are different for different scripts, as such the script itself arrives as a very challenging obstacle [14]. Bhowmik et al. [4] have used modified Hough transform based method for correcting skew in handwritten Bangla word images. Roy et al. [29] have illustrated a methodology that utilises the angle created between two successive characters to estimate and correct skew present in handwritten Bangla words. Jayadevan et al. [11] have used Radon transform for correcting skew present in handwritten words. They have experimented on handwritten Devanagari legal words. Malakar et al. [20] have delineated a generalized Hough transform based method for skew detection and correction in handwritten Bangla words. Ghosh and Mandal [7] have defined a method where they divide a word based on its bounding box. This is followed by calculation of the center of gravity at both parts of word. The slope

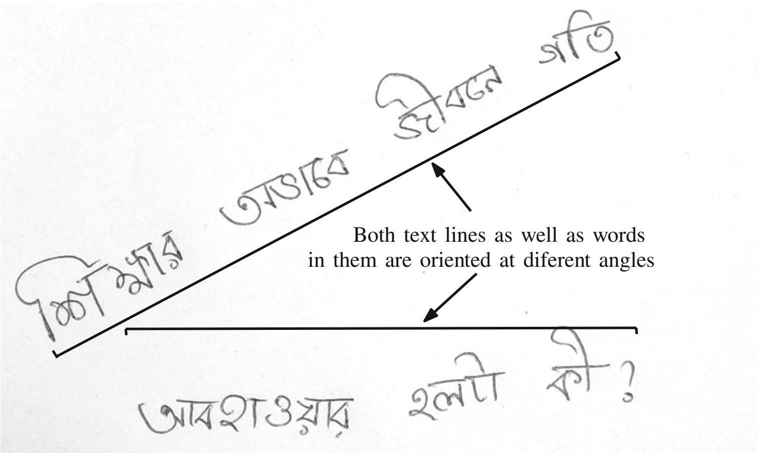

Both text lines as well as words
in them are oriented at diferent angles
আবহাওয়ার হলটা কী ?

Fig. 1 An example of multiple skew present in text

created by the centers of gravity is considered as the skew present in the word. They experimented on Bangla handwritten words. Guru et al. [9] have proposed a methodology that uses an ellipse for encapsulating a handwritten word image. The slope of the major axis of the ellipse is considered as the skew present in the word. They experimented on multilingual Indian documents. Jundale and Hegadi [12] have delineated the use of Hough transform for performing skew correction of handwritten Devanagari words. Further, they [13] have used a parallel axes based linear regression based strategy for correcting skew in handwritten Devanagari words. Guru et al. [10] have extracted the contour of a word image. This is followed by extraction of line segments based on small eigen values. They have further applied K-means clustering for extracting the head line present in Devanagari words. Finally, they estimate skew based on the extracted head line. Pramanik and Bag [25] have proposed a linear regression based strategy to correct skew in Bangla and Devanagari handwritten words but with limited capability. Kar et al. [14] have used a rectangular mask to detect the core region of a word, followed by linear regression and skew correction. A detailed overview of the aforementioned works is provided in Table 1.

Based on our observation from the literature, we found the following limitations or challenges on skew correction of handwritten words:

M1: In present literature, works have been carried out by directly imposing linear curve fitting on words. We observed that, if we use the structural properties of the handwritten words before employing linear curve fitting, then it may provide better results. To the best of our knowledge, none of these types of works have utilized the structural knowledge of handwritten words. (e.g. [13]).

M2: Most reported works fail when the skew orientation is greater than ±25°. (e.g. [4, 14, 29]).

M3: Few other works fail when the *mātr*ā (also known as headline) is absent or discontinuous. (e.g. [20]).

M4: Absence of skew detection limits a method's capability of handling skewed text. (e.g. [16, 32]).

M5: Pramanik and Bag [25] fails at times to distinguish eligible from ineligible pixels required to estimate the regression line.

Keeping the aforementioned limitations in mind, we present a method that provides the following solutions:

C1: Based on the structural knowledge of the languages under consideration, the proposed method extracts a group of eligible pixels and employs linear curve fitting for estimation and correction of skew present in Bangla, Hindi, Marathi, and Punjabi handwritten words.

C2: The proposed method is capable of handling high skew to an extent of ±55°.

C3: The proposed method works even when the *mātr*ā is mostly absent or discontinuous.

C4: Moreover, we have used an existing segmentation-based classification strategy to show the efficacy of the proposed method when compared with non-skew-corrected words as baseline.

C5: The proposed work is able to distinguish efficiently between eligible and ineligible pixels which is a limitation for [25].

In the past few years, Convolution Neural Networks (CNNs) have widely evolved and being used towards improving the performance of various image processing and multimedia research domains. However, CNNs require considerable training data and energy consumption. In addition, when the training data is limited, the methods may cause a cross-learning

Table 1 Existing works on skew estimation and correction of handwritten words for Indian languages

Research Works	Language	Preprocessing	Method	Dataset	Result (%)	Year
Bhowmik et al. [4]	Bangla	Median filtering, binarization	Hough transform	Local dataset	—	2005
Roy et al. [29]	Bangla	Median filtering, binarization, smoothing	Morphological	Locally collected 150 words	—	2005
Jayadevan et al. [11]	Hindi, Marathi	Not mentioned	Radon transform	DB1, DB2, and DB3 with 26,720 words	—	2011
Malakar et al. [20]	Bangla	Not mentioned	Hough transform	Local dataset	—	2012
Ghosh and Mandal [7]	Bangla	Binarization, quadratic filtering	Morpho- logical	Locally collected 3839 words	92.22	2012
Guru et al. [9]	Kannada, English, Hindi	Not mentioned	Morphological, clustering	Locally collected 114 documents	—	2013
Jundale and Hegadi [12]	Hindi, Marathi	Noise removal, binarization	Hough transform	Locally collected 1050 words	92.88	2015
Jundale and Hegadi [13]	Hindi, Marathi	Noise removal, binarization	Morphological, linear regression	Locally collected 500 words	89.00	2015
Guru et al. [10]	Hindi	Thinning	Morphological	Locally collected 400 words	92.00	2015
Pramanik and Bag [25]	Bangla, Hindi, Marathi, Panjabi	Binarization, noise removal	Morphological, linear regression	Cmaterdb 1.1.1 [3], 1.5.1 [3, 23]	92.37	2018
Kar et al. [14]	Bangla, Hindi, Roman	—	Core region detection, linear regression	Locally collected 400 words for each script	—	2019

problem. So, there is still research space for model-based approaches for image segmentation, which are more cost efficient, energy efficient and space efficient. As such, to identify the effectiveness of our proposed method, we have also focussed towards the feature extraction and recognition part. For correct recognition, selection of correct feature techniques plays a crucial part [17, 21, 35, 42]. We used structural features delineated in [28] and MLP and SVM for classification.

The paper is systematised as follows: A detailed description of the proposed methodology is provided in Section 2. A delineation of the experimental results and their corresponding analysis is presented in Section 3, followed by a brief conclusion in Section 4.

Algorithm 1 Admissible Pixel Identification.

```
Input  : $(I_q)_{m\times n}, \mathcal{L} = \{c_1, c_2, \cdots, c_l\}, \mathcal{P} = \mathcal{A} = \mathcal{A}' = \mathcal{T} = \phi$, change = 1
Output: $\mathcal{P}$: Consisting of admissible pixels
while change $\neq$ 0 do
    for $i \in \{1, ..., |\mathcal{L}|\}$ do
        if First foreground pixel $p_i$ is encountered in column $c_i$ then
            Store $p_i$ in $\mathcal{P}$;
        end
    end
    for $i \in \{1, ..., |\mathcal{P}|\}$ do
        if ($\angle p_i p_{i+1} p_{i+2} \leq 165^\circ$) and ($\mathcal{T}$ is empty or $\mathcal{T}$ does not already contain the
        pair) then
            Store the consecutive pairs with higher difference value in $\mathcal{T}$;
        end
    end
    for Every two consecutive pairs $(p_{i_1}, p_{i_2})$ and $(p_{i_3}, p_{i_4}) \in \mathcal{T}$ do
        if $p_{i_2}(r) = p_{i_3}(r)$ then
            $\mathcal{A}' \leftarrow \mathcal{A}' \cup p_{i_2}; \mathcal{T} \leftarrow \mathcal{T} \setminus \{(p_{i_1}, p_{i_2}), (p_{i_3}, p_{i_4})\}$;
        end
    end
    Modify change to zero if $\mathcal{T}$ and $\mathcal{A}'$ are empty;
    $\mathcal{A} \leftarrow \mathcal{P} \setminus (\mathcal{T} \cup \mathcal{A}')$;
    for $i \in \{1, ..., |\mathcal{T}|\}$ do
        for $j \in \{1, ..., |\mathcal{A}|\}$ do
            $d_{p_{i_1}} \leftarrow p_{i_1}(r) - p_j(r); d_{p_{i_2}} \leftarrow p_{i_2}(r) - p_j(r)$;
            Increment $\max_{d_1}$ or $\max_{d_2}$ by 1 based on the outcome of the difference
            between $d_{p_{i_1}}$ and $d_{p_{i_2}}$;
        end
        if $\max_{d_1} > \max_{d_2}$ then
            $\mathcal{A}' \leftarrow \mathcal{A}' \cup p_{i_1}; \mathcal{T} \leftarrow \mathcal{T} \setminus \{(p_{i_1}, p_{i_2})\}$;
        end
        else
            $\mathcal{A}' \leftarrow \mathcal{A}' \cup p_{i_2}; \mathcal{T} \leftarrow \mathcal{T} \setminus \{(p_{i_1}, p_{i_2})\}$;
        end
    end
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{A}'; \mathcal{T} \leftarrow \phi; \mathcal{A} \leftarrow \phi; \mathcal{A}' \leftarrow \phi$;
end
```

2 Proposed methodology

We propose a strategy that detects and corrects skew present in Bangla, Hindi, Marathi, and Punjabi handwritten words. A complete working model of the entire approach is shown in Fig. 2.

2.1 Preprocessing

Given a *mātrā*-based multilingual document image, we utilise the method depicted in [30] to extract the text lines. This is followed by word extraction (Fig. 2a). For each extracted word image (I_q), we perform component labelling and noise cleaning on I_q and remove all

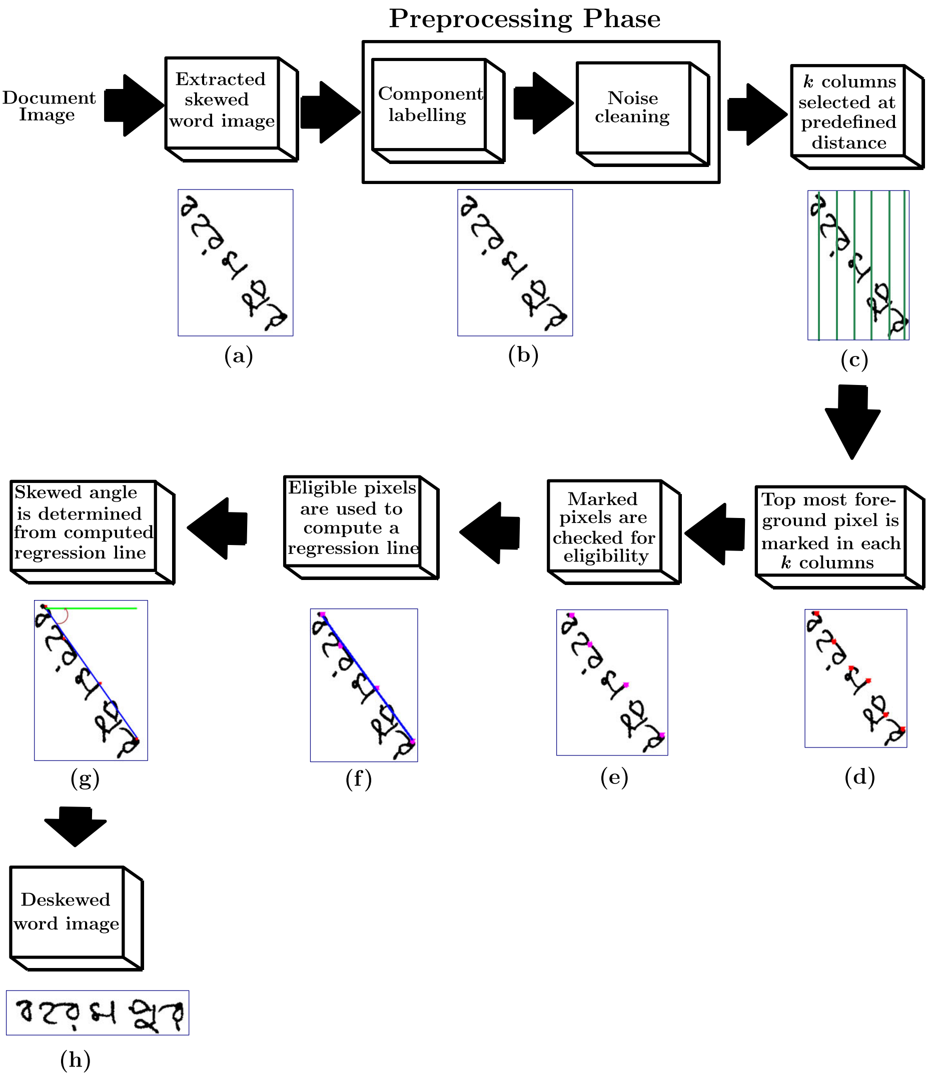

Preprocessing Phase
Document Image
Extracted skewed word image
Component labelling
Noise cleaning
k columns selected at predefined distance
(a)
(b)
(c)
Top most foreground pixel is marked in each k columns
Marked pixels are checked for eligibility
Eligible pixels are used to compute a regression line
Skewed angle is determined from computed regression line
(d)
(e)
(f)
(g)
Deskewed word image
(h)

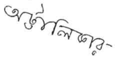
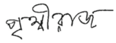
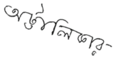
Fig. 2 Work flow of the entire proposed methodology with an example

components below a certain threshold, υ. Here, we consider υ as 30 [24]. We crop this noise cleaned image I_q to its boundaries (Fig. 2b).

2.2 Skew estimation and correction

Here, I_q having dimension $m \times n$ is taken as input. The k of the n columns in I_q are chosen as 12% of the width of I_q consecutively (Fig. 2c). We experimented on a set of 500 randomly chosen images and surmised 12 to be the optimal percentage of selecting columns in order to obtain the best result. A validation for the aforementioned conclusion is provided in Section 3.2. These k selected columns are denoted as $\mathcal{L} = < c_1, c_2, \cdots, c_k >$. Algorithm 1 is used for admissible pixel identification for skew estimation and correction. Each column in $\mathcal{L}$ is consecutively traversed from top to bottom and the first encountered foreground pixel is stored in $\mathcal{P}$ (Alg. 1, Steps: 2–6) (Fig. 2d). The stored pixels in $\mathcal{P}$ are denoted as $\mathcal{P} = < p_1, p_2, \cdots, p_l >$, where $l \leq k$. Each pixel p_j in $\mathcal{P}$ associates with a row number and a column number $(p_j(r), p_j(c))$. Three sets viz., $\mathcal{A}$, $\mathcal{A}'$, and $\mathcal{T}$ are used to represent pixels in $\mathcal{P}$ as admissible, inadmissible, and transitional, respectively. $\mathcal{A}$ constitutes pixels that are admissible for further skew correction. $\mathcal{A}'$ constitutes pixels that are inadmissible for further computation and will be removed from $\mathcal{P}$. $\mathcal{T}$ constitutes pairs of pixels that are provisionally kept here before examining their belongingness in $\mathcal{A}$ or $\mathcal{A}'$. For every three successive pixels p_j, p_{j+1}, and p_{j+2} in $\mathcal{P}$, the angle $\angle p_j p_{j+1} p_{j+2}$ (denoted as θ_l) is computed. If $\theta_l \leq 165°$, then $|p_j(r) - p_{j+1}(r)|$ and $|p_{j+1}(r) - p_{j+2}(r)|$ are evaluated. A validation for choosing 165° as optimal value is provided in Section 3.2. If $|p_j(r) - p_{j+1}(r)| > |p_{j+1}(r) - p_{j+2}(r)|$, then (p_j, p_{j+1}) is considered as transitional pixel pair and stored in $\mathcal{T}$. Otherwise, (p_{j+1}, p_{j+2}) is considered as transitional pixel pair and stored in $\mathcal{T}$ instead (Alg. 1, Steps: 7–11). If a pixel p_j in $\mathcal{P}$ appears twice in two consecutive pairs in $\mathcal{T}$ in a single iteration, then p_j is transferred from the set $\mathcal{T}$ to $\mathcal{A}'$, while the pixels paired with p_j in $\mathcal{T}$ are removed from $\mathcal{T}$ (Alg. 1, Steps: 12–16). Once all the intermediate pixels get marked in a single iteration, the admissible pixels in $\mathcal{A}$ are computed as $\mathcal{A} = \mathcal{P} \setminus (\mathcal{T} \cup \mathcal{A}')$ (Alg. 1, Step: 18).

For each pixel pair (p_j, p_{j+1}) in $\mathcal{T}$, the row-wise difference, d_{p_j} and $d_{p_{j+1}}$ of p_j and p_{j+1} with every pixel in $\mathcal{A}$ is computed. The maximum of the two differences d_{p_j} and $d_{p_{j+1}}$ is evaluated as max_d. The pixel in the pair (p_j, p_{j+1}) that is associated with the maximum number of max_d is sent from $\mathcal{T}$ to $\mathcal{A}'$, while the other is sent to $\mathcal{A}$ (Alg. 1, Steps: 19–30). Once all the pixel pairs in $\mathcal{T}$ are examined, the pixels belonging to $\mathcal{A}'$ are deleted from $\mathcal{P}$ and $\mathcal{A}$, $\mathcal{T}$, and $\mathcal{A}'$ are all emptied (Alg. 1, Step: 31). This approach is carried out until no pixel gets represented in $\mathcal{T}$ or $\mathcal{A}'$. The inadmissible pixels are removed from $\mathcal{P}$ to ensure that the skew estimation does not get affected due to the presence of certain consonants and upper modifiers that appear above the headline and at places where a rift is formed due to the absence of the headline in a word (Fig. 2e). , , ,, and are few examples of consonants and upper modifiers that appear above the headline in Bangla script.

To illustrate the proposed approach, the skewed words  and  are used as working examples. The two words illustrate the two different conditions as explained above. Currently, we consider the word  for illustrating the first condition. During first iteration, for the first three pixels p_1, p_2, and p_3, the computed angle, i.e., $\angle p_1 p_2 p_3$, is lesser than the specified threshold 165°, i.e., $\angle p_1 p_2 p_3$ (θ_l) is $\leq$ 165° and $|p_1(r)$ - $p_2(r)| > |p_2(r)$ - $p_3(r)|$. So, the pixel pair (p_1, p_2) is sent to $\mathcal{T}$ (Fig. 3). Similarly,

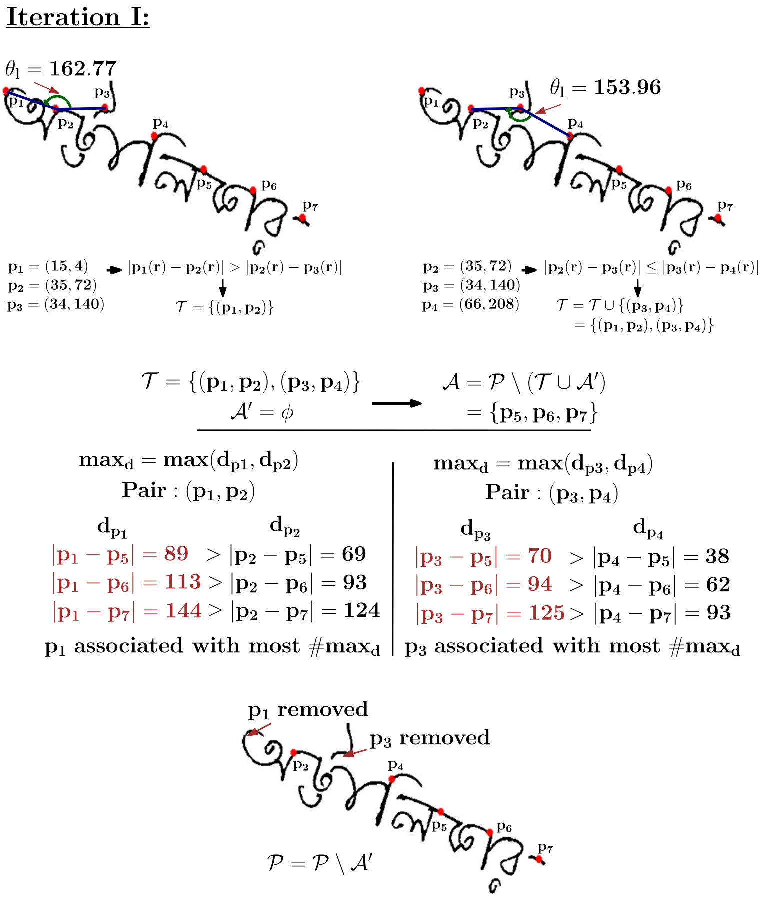

Iteration I:
$\theta_l = 162.77$
p_1
p_2
p_3
p_4
p_5
p_6
p_7
$p_1 = (15,4)$
$p_2 = (35,72)$
$p_3 = (34,140)$
$|p_1(r) - p_2(r)| > |p_2(r) - p_3(r)|$
$\mathcal{T} = \{(p_1,p_2)\}$
$\theta_l = 153.96$
$p_2 = (35,72)$
$p_3 = (34,140)$
$p_4 = (66,208)$
$|p_2(r) - p_3(r)| \leq |p_3(r) - p_4(r)|$
$\mathcal{T} = \mathcal{T} \cup \{(p_3,p_4)\} = \{(p_1,p_2),(p_3,p_4)\}$
$\mathcal{T} = \{(p_1,p_2),(p_3,p_4)\}$
$\mathcal{A}' = \phi$
$\mathcal{A} = \mathcal{P} \setminus (\mathcal{T} \cup \mathcal{A}') = \{p_5,p_6,p_7\}$
$\max_d = \max(d_{p1}, d_{p2})$
Pair : (p_1,p_2)
d_{p_1} d_{p_2}
$|p_1 - p_5| = 89 > |p_2 - p_5| = 69$
$|p_1 - p_6| = 113 > |p_2 - p_6| = 93$
$|p_1 - p_7| = 144 > |p_2 - p_7| = 124$
p_1 associated with most $\#\max_d$
$\max_d = \max(d_{p3}, d_{p4})$
Pair : (p_3,p_4)
d_{p_3} d_{p_4}
$|p_3 - p_5| = 70 > |p_4 - p_5| = 38$
$|p_3 - p_6| = 94 > |p_4 - p_6| = 62$
$|p_3 - p_7| = 125 > |p_4 - p_7| = 93$
p_3 associated with most $\#\max_d$
p_1 removed
p_3 removed
$\mathcal{P} = \mathcal{P} \setminus \mathcal{A}'$

Fig. 3 Stepwise illustration of inadmissible pixel removal delineating the first iteration of the first condition for skew correction

for the consecutive pixels p_2, p_3, and p_4, the angle $\angle p_2p_3p_4$, is lesser than the specified threshold and $|p_2(r)$ - $p_3(r)| \leq |p_3(r)$ - $p_4(r)|$. As a result, pixel pair (p_3, p_4) is sent to $\mathcal{T}$ (Fig. 3). Once all the elements in $\mathcal{P}$ are examined and the first iteration completes, the elements in $\mathcal{A}$ is computed as $\mathcal{A}$=$\mathcal{P} \setminus (\mathcal{T} \cup \mathcal{A}')$. d_{p_1} and d_{p_2} of pixel pair (p_1, p_2) is computed with elements in $\mathcal{A}$ and correspondingly, d_{p_3} and d_{p_4} of pixel pair (p_3, p_4) is computed with elements in $\mathcal{A}$. It is concluded that for the first pair, p_1 is associated with most # $\max_d$ and for the second pair, p_3 is associated with most # $\max_d$ (Fig. 3). As a result, p_1 and p_3 are transferred to $\mathcal{A}'$ from $\mathcal{T}$ and subsequently, p_1 and p_3 are removed from $\mathcal{P}$ (Fig. 3). All the elements in $\mathcal{A}$, $\mathcal{A}'$, and $\mathcal{T}$ are removed.

During the second iteration, for the three pixels p_2, p_4, and p_5, the computed angle, i.e., $\angle p_2p_4p_5$, is lesser than the specified threshold 165°, i.e., $\angle p_2p_4p_5$ (θ_l) is $\leq$ 165° and $|p_2(r)$ - $p_4(r)| \leq |p_4(r)$ - $p_5(r)|$. So, the pixel pair (p_4, p_5) is sent to $\mathcal{T}$ (Fig. 4). Once all the elements in $\mathcal{P}$ are examined and the second iteration completes, the elements in $\mathcal{A}$ is computed as $\mathcal{A}$=$\mathcal{P} \setminus (\mathcal{T} \cup \mathcal{A}')$. d_{p_4} and d_{p_5} of pixel pair (p_4, p_5) is computed with elements in $\mathcal{A}$ and is concluded that p_4 is associated with most # $\max_d$. As a result, p_4 is transferred

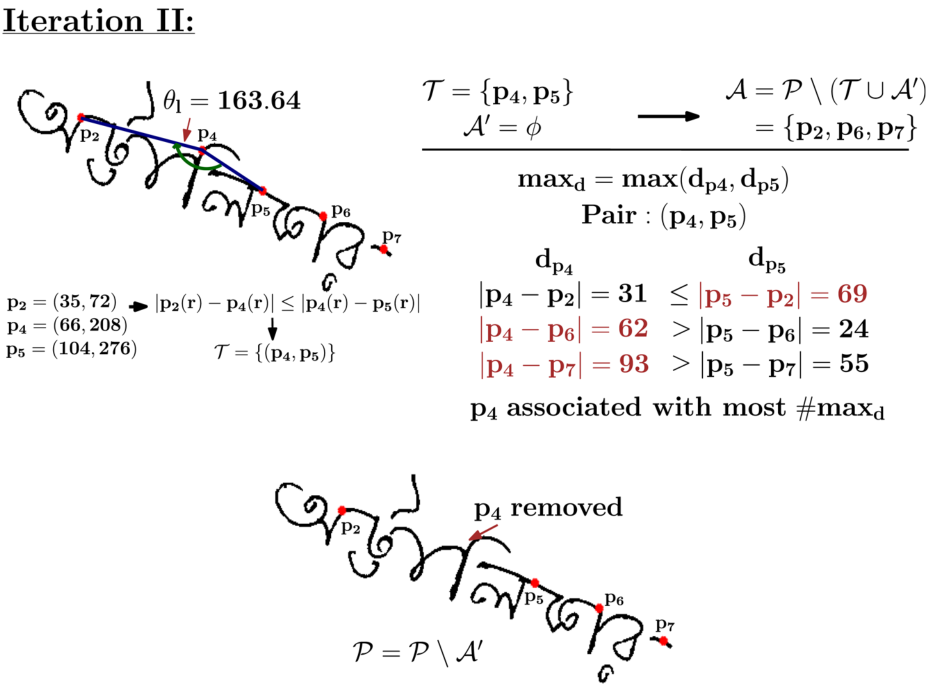

Iteration II:
$\theta_l = 163.64$
p_2
p_4
p_5
p_6
p_7
$p_2 = (35, 72)$
$p_4 = (66, 208)$
$p_5 = (104, 276)$
$|p_2(r) - p_4(r)| \leq |p_4(r) - p_5(r)|$
$\mathcal{T} = \{(p_4, p_5)\}$
$\mathcal{T} = \{p_4, p_5\}$
$\mathcal{A}' = \phi$
$\mathcal{A} = \mathcal{P} \setminus (\mathcal{T} \cup \mathcal{A}')$
$= \{p_2, p_6, p_7\}$
$\max_d = \max(d_{p4}, d_{p5})$
Pair : (p_4, p_5)
d_{p_4}
d_{p_5}
$|p_4 - p_2| = 31 \leq |p_5 - p_2| = 69$
$|p_4 - p_6| = 62 > |p_5 - p_6| = 24$
$|p_4 - p_7| = 93 > |p_5 - p_7| = 55$
p_4 associated with most $\#\max_d$
p_4 removed
$\mathcal{P} = \mathcal{P} \setminus \mathcal{A}'$

Fig. 4 Stepwise illustration of inadmissible pixel removal delineating the second iteration of the first condition for skew correction

to $\mathcal{A}'$ from $\mathcal{T}$ and subsequently, p_4 is removed from $\mathcal{P}$ (Fig. 4). All the elements in $\mathcal{A}$, $\mathcal{A}'$, and $\mathcal{T}$ are removed. During third iteration, no consecutive pixel with angle lesser than the specified threshold 165° is found (Fig. 5).

To illustrate the second condition, the word পৃথ্বীরাজ is used. Presence of the upper modifier ী in পৃথ্বীরাজ, puts the third encountered foreground pixel p_3 much higher than the rest of the pixels in $\mathcal{P}$ (Fig. 6a). As a result, during first iteration, when the angle for the pixels p_1, p_2, and p_3 is computed, i.e., $\angle p_1 p_2 p_3$, the angle is lesser than the specified threshold 165°, i.e., $\angle p_1 p_2 p_3$ (θ_l) is $\leq$ 165° and $|p_1(r)$ - $p_2(r)| \leq |p_2(r)$ - $p_3(r)|$. So, the pixel pair (p_2, p_3) is sent to $\mathcal{T}$ (Fig. 6a). Similarly, the angle computed for the next

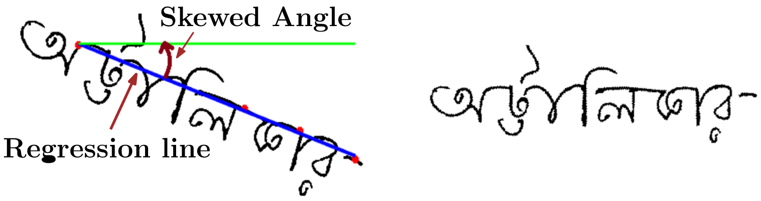


Fig. 5 Stepwise illustration of inadmissible pixel removal delineating the third iteration of the first condition for skew correction

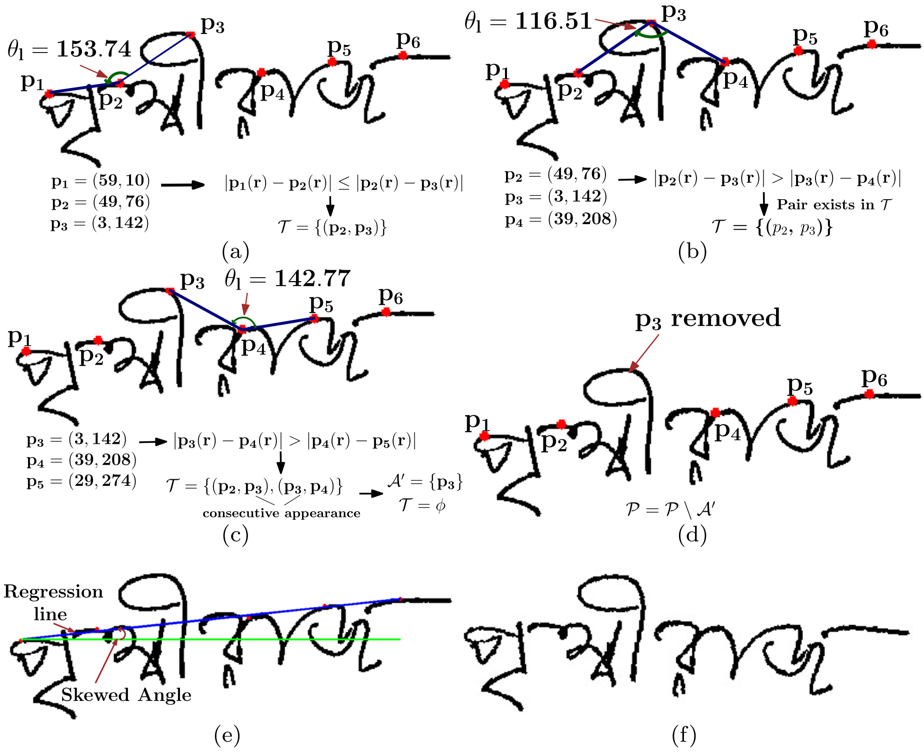


Fig. 6 Stepwise illustration of inadmissible pixel removal delineating the second condition for skew correction

three consecutive pixels p_2, p_3, and p_4, i.e., $\angle p_2 p_3 p_4$, is lesser than the specified threshold 165° and $|p_2(r)$ - $p_3(r)| > |p_3(r)$ - $p_4(r)|$. So, the pixel pair (p_2, p_3) is sent to $\mathcal{T}$. As the pair already exists in $\mathcal{T}$, the current pair is discarded (Fig. 6b). Identically, the angle created by the next three subsequent pixels p_3, p_4, and p_5 is lesser than 165°. As, $|p_3(r)$ - $p_4(r)| > |p_4(r)$ - $p_5(r)|$, the pixel pair (p_3, p_4) is sent to $\mathcal{T}$. Pixel p_3 appears in two consecutive pairs in $\mathcal{T}$. As a consequence, p_3 is transferred to $\mathcal{A}'$, while the two pixels associated with it, i.e., p_2 and p_4, are removed from $\mathcal{T}$ (Fig. 6c). Eventually, p_3 is removed from $\mathcal{P}$ (Fig. 6d) and $\mathcal{A}$, $\mathcal{A}'$, and $\mathcal{T}$ are emptied. During second iteration, no consecutive pixel with angle lesser than the specified threshold 165° is found.

Next, predictive $\hat{p}_i(r)$ values are computed utilising Algorithm 2 and Matlab's polyfit function based on the pixels in $\mathcal{P}$. For this purpose, we compute the mean values of rows ($p_i(r)$) and columns ($p_i(c)$) (Alg. 2, Step: 2–5). We compute two coefficients b_0 and b_1. b_0 is the intercept that determines where the regression line to be computed intercepts the y-axis and b_1 is the slope of the aforementioned line (Alg. 2, Step: 6–9). Finally, we compute the predictive $\hat{p}_i(r)$ values utilising the aforementioned coefficients (Alg. 2, Step: 10–12). These predicted $\hat{p}_i(r)$ values and $p_i(c)$ values (where i$\in\{1 \cdots |\mathcal{P}|\}$) are used to compute the regression line (Fig. 2f, 5, and 6e). The skewed angle is determined from the computed regression line (Figs. 2g). Finally, the image is rotated according to the estimated angle for de-skewing (Figs. 2h, 5, and 6f).

Algorithm 2 Pixel Prediction for Regression Line Computation.

```
$C_{sum} \leftarrow 0$; $R_{sum} \leftarrow 0$; $b_1 \leftarrow 0$;
for $i \in \{1, ..., |\mathcal{P}|\}$ do
    $C_{sum} \leftarrow C_{sum} + p_i(c)$; $R_{sum} \leftarrow R_{sum} + p_i(r)$;
end
$C_m \leftarrow C_{sum}/|\mathcal{P}|$; $R_m \leftarrow R_{sum}/|\mathcal{P}|$;
for $i \in \{1, ..., |\mathcal{P}|\}$ do
    $t_1 \leftarrow p_i(c) - C_m$; $t_2 \leftarrow p_i(r) - R_m$; $b_1 \leftarrow b_1 + ((t_1 \times t_2)/{t_1}^2)$;
end
$b_0 \leftarrow R_{sum} - b_1 \times C_{sum}$;
for $i \in \{1, ..., |\mathcal{P}|\}$ do
    $\hat{p}_i(r) \leftarrow b_0 + b_1 \times p_i(c)$;
end
```

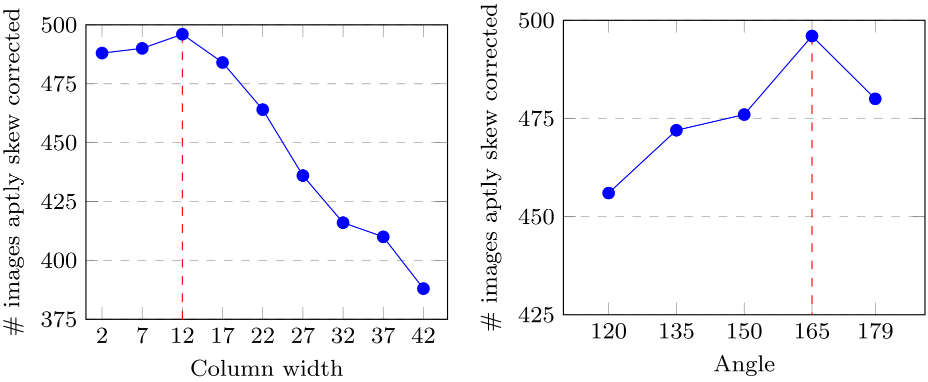
3 Experimental results and analysis

3.1 Dataset

We have employed ICDAR 2013 Segmentation Dataset [34] and PHDIndic_11 [23] dataset for performing skew correction of the word images. A total of 10,000 images are used to carry out the experiment. Apart from this, we have also used another set of 1000 word images for performing validation. All modules and codes used for implementation are written in MATLAB environment.

3.2 Parameter validation

Figure 7a delineates the validation of choosing 12% of the width of I_q as the optimal percentage for selecting columns in order to traverse uniformly distributed columns for skew estimation and correction. We have experimented on a set of 500 randomly chosen images



(a)

(b)

Fig. 7 **a** Validation delineating 12 as the optimal column width selection percentage for I_q; **b** Validation depicting 165 as the optimal value for examining if θ_l is eligible or not and observed the surmised 12 to be the optimal percentage of selecting columns in order to obtain the best result.

For every three successive pixels p_j, p_{j+1}, and p_{j+2} in $\mathcal{P}$, it is validated if the angle $\angle p_j p_{j+1} p_{j+2}$ (denoted as θ_l) $\leq$ 165°. We have randomly chosen 500 word images and performed validation and surmised 165 to be the optimal value (Fig. 7b).

3.3 Time complexity

The time complexity of Algorithm 1 depends on three main factors, namely *change*, $|\mathcal{T}|$, and $|\mathcal{A}'|$. *change* is the number of iterations the algorithm runs until the sets $\mathcal{T}$ and $\mathcal{A}'$ become empty. $\mathcal{A}$ comprises of pixels that are eligible for further skew correction and $\mathcal{T}$ comprises pairs of pixels that are provisionally stored before examining their belongingness in $\mathcal{A}$ or $\mathcal{A}'$. So, the time complexity of Algorithm 1 can be approximated to $O(change \times |\mathcal{T}| \times |\mathcal{A}|)$. Similarly, the time complexity of Algorithm 2 can be approximated as $O(|\mathcal{P}|)$.

3.4 Experimental analysis

As the actual skew angle estimation of a particular word in an unconstrained environment is only possible by human beings, we have employed a semi-automated process to find the ground truth images and the corresponding angles of each word. We have used a recently developed method [25] to estimate the slope of each handwritten word. In case of erroneous cases, we have used traditional rotation operations to find the actual skew. The ground truth angle of each word is compared with the corresponding result of our proposed method as well as with other state-of-the-art methods. The root mean square (RMS) value of absolute error is determined by the following equation:

$$E_s = \sqrt{\frac{\sum_{i=1}^{t} |g - x|^2}{t}} \tag{1}$$

where, g is the ground truth angle, x is the skew angle determined by the proposed method and t is the total number of words.

The proposed methodology has the lowest RMS error of 0.95 for Punjabi words while the highest error of 2.47 is recorded for Bangla words. In an ideal case, a word belonging to any of the four languages under consideration will contain a headline that connects the entire word. In such an ideal scenario, all the four languages will provide almost similar results for the proposed method. But, in general case, we have observed based on the writing of different individuals that the headlines are completely absent or very frequently disconnected in Bangla words. This is not the case for other languages. Our dataset similarly reflects the aforementioned general case. Due to the aforementioned characteristic of Bangla words, the performance of Bangla is not as good as other languages. The performance of our proposed method is delineated in Table 2. To show the efficiency of the proposed skew estimation and correction technique, we have synthetically rotated few handwritten word images and provided a visual and statistical analysis in Table 3. We have compared our proposed method with six recent works, namely Malakar et al. [20], Ghosh and Mandal [7], Jundale and Hegadi [12], Jundale and Hegadi [13], Pramanik and Bag [25], and Roy et al. [29]. Malakar et al. [20] have used generalized Hough transform for skew correcting handwritten Bangla word images. Ghosh and Mandal [7] have divided the word images

Table 2 Skew estimation and correction performance of our proposed method for different languages

Language	Total Words	Measurement Parameter	RMS Error Value
Bangla	2500	E_s	**2.47**
Hindi			**1.25**
Marathi			**0.95**
Punjabi			**1.13**

morphologically and computed the centers of gravity of both part for skew angle detection. Jundale and Hegadi [12] have used a modified Hough transform based method for skew correcting Hindi handwritten words. Later, they [13] have used a basic parallel axes based linear regression based strategy for correcting skew in handwritten Devanagari words. Pramanik and Bag [25] is similar to the current method at initial stage but differs considerably at the latter stage. This method examines if the angle computed between three consecutive pixels is greater than or equal to a certain threshold and removes the middle pixel if the condition matches. Due to this rigid condition, several eligible pixels get removed and several ineligible pixels remain. Roy et al. [29] have used the angle between two successive characters to estimate and correct skewness. This technique cannot handle skew beyond ±10°. In the current method, we use three different sets as well as few other conditions to ensure that an eligible pixel does not get removed and handles skewed word images to an extent of ±55°. This mechanism shows improvement in accuracy when compared with other recent works. A comparison with [20], [7], [12], [13], and [25] is reported in Table 4. We have also reported and compared the average execution time required for each of the aforementionaed method to completely process each image. In terms of execution time, our proposed method is relatively slow when compared with other methodologies. We have also delineated a visual comparison of few word images with [25] and [29] in Tables 5 and 6 respectively to validate that the current method provides better skew correction.

Apart from the aforementioned analysis, we have also used a segmentation-based classification methodology for delineating the effectiveness of our proposed skew correction strategy. For segmentation of the words into pseudo-characters, we used the methodology depicted in [27]. We have extracted the structural features using the methodology delineated in [28]. We have used MLP and SVM for classification. For comparison purpose, we have used non-skew-corrected word images as baseline. We have applied the same segmentation and feature set extraction strategy on these word images. The comparison delineates that the segmentation methodology works better on skew corrected words and thereby provides better classification accuracy (Table 7).

3.5 Failure cases

The proposed methodology is very robust if single skew is present in the word images. But, it fails if a particular word is associated with multiple skews. If the size of upper modifier is too large w.r.t. the actual size of the word image, or the characters in a word are too much separated, then the proposed method fails at times, as most eligible pixels get deleted. Complete absence of headline is also a reason for some failures. Few such examples are provided in Fig 8.

Table 3 Few skew estimation and correction results of handwritten words

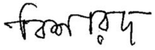
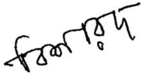
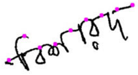
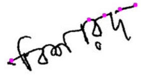
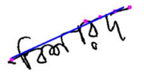
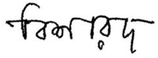
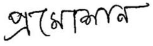
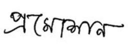
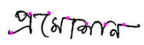
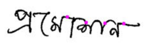
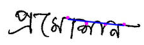
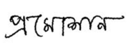
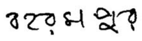
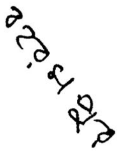
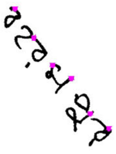
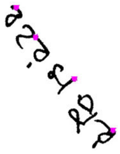
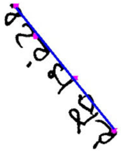
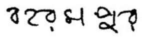
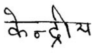
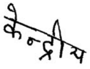
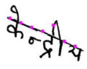
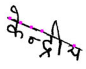
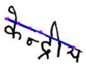
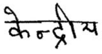
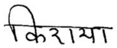
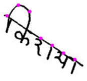
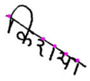
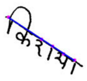
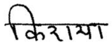
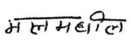
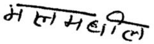
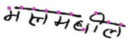
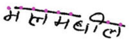
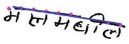
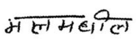
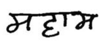
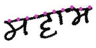
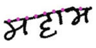
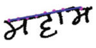
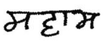
Binarized image	Synthetically rotated image	Rotation angle (θ)	Initial marked pixels in $\mathcal{P}$	After removal of ineligible pixels	Regression line drawn	Skew corrected output	Corrected angle (θ')
		-20.25					16.39
		10.27					-6.35
		53.76					-51.19
		22.57					-23.19
		31.44					-33.04
		10.00					-8.90
		-10.35					10.52

Table 4 Skew estimation and correction performance comparison of our proposed method with other recent works

Language	Method	Model Used	Total # words	Measurement Parameter	RMS Error Value	Avg. Execution Time (in seconds)
Bangla	Malakar et al. [20]	Generalized Hough Transform	2500	E_s	3.09	0.68
	Ghosh and Mandal [7]	Morphological based			3.44	0.73
	Pramanik and Bag [25]	Linear Regression based			3.58	0.65
	Proposed Method	Linear Regression based			**2.47**	0.69
Hindi	Jundale and Hegadi [12]	Modified Hough Transform	2500	E_s	2.52	0.48
	Jundale and Hegadi [13]	Linear Regression based			3.14	0.55
	Pramanik and Bag [25]	Linear Regression based			2.29	0.60
	Proposed Method	Linear Regression based			**1.25**	0.65
Marathi	Jundale and Hegadi [12]	Modified Hough Transform	2500	E_s	2.88	0.45
	Jundale and Hegadi [13]	Linear Regression based			3.35	0.54
	Pramanik and Bag [25]	Linear Regression based			1.46	0.58
	Proposed Method	Linear Regression based			**0.95**	0.62
Punjabi	Pramanik and Bag [25]	Linear Regression based	2500	E_s	1.64	0.60
	Proposed Method	Linear Regression based			**1.13**	0.64

Table 5 Comparison of skew correction on few word images of current method with Pramanik and Bag [25]

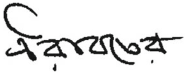
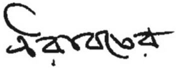
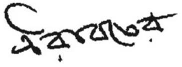
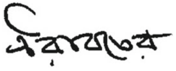
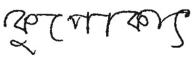
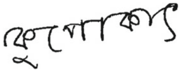
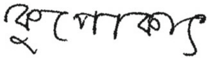
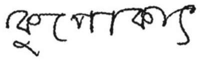
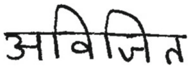
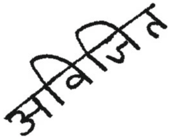
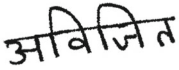
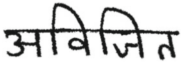
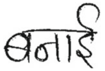
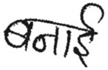
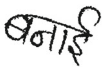
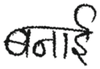
Binarized image	Synthetically rotated image	Rotation angle (θ)	Deskewed output of [25]	Skew corrected (θ') [25]	Deskewed output of proposed method	Skew corrected (θ'')
		-1.29		-6.37		**1.12**
		-8.56		12.08		**7.07**
		-38.25		29.43		**37.80**
		10.43		6.49		**-7.33**

Table 6 Comparison of skew correction on few word images of current method with Roy et al. [29]

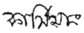
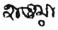
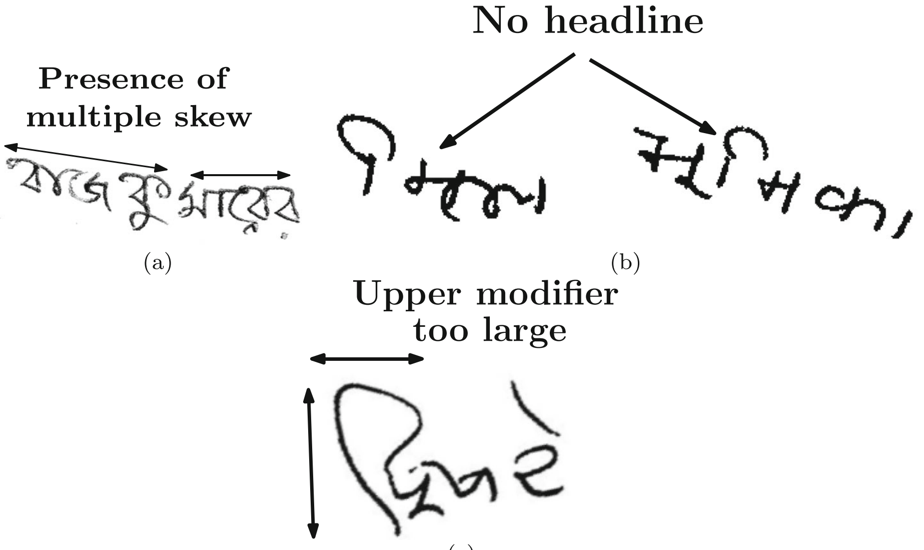
Input image	Skew present (θ)	Deskewed output of [29]	Skew corrected (θ') [29]	Deskewed output of proposed method	Skew corrected (θ'')
	8.37		-4.90		**-6.02**
	-7.00		1.15		**5.02**

Table 7 Accuracy comparison with different classifiers

Handwritten Words	Classifier	Recognition Accuracy (%)
Non skew corrected	MLP	77.34
	SVM	74.53
Skew corrected	MLP	**86.51**
	SVM	**82.95**



(a) (b) (c)

Fig. 8 Failure cases. **a** A Bangla word with multiple skew present; **b** Headline is completely absent in Hindi and Marathi word; **c** Size of upper modifier is too large w.r.t. the actual size of the Punjabi word image

4 Conclusion

Segmentation algorithms in multi-oriented handwritten document often fail due to the presence of skew in text. So, it is important to deskew the text images in the preprocessing phase before applying segmentation algorithms. Appearance of skew in multi-oriented Indian language based handwritten document is higher due to the presence of cursive nature. In the current work, we utilise a salient feature present in Indian scripts called *mātr*ā, extract a group of eligible pixels and employ linear curve fitting for detecting and correcting skew in handwritten words. This method is capable in correcting skew in four different Indian languages, viz. Bangla, Hindi, Marathi, and Punjabi and handles skewed word images to an extent of ±55°.

Declarations

Conflict of Interests Rahul Pramanik declares that he has no conflict of interest. Soumen Bag declares that he has no conflict of interest.

References

1. Bag S, Harit G (2013) A survey on optical character recognition for Bangla and Devanagari scripts. Sadhana 38(1):133–168
2. Bagdanov A, Kanai J (1997) Projection profile based skew estimation algorithm for JBIG compressed images. In: Proceedings of the international conference on document analysis and recognition, vol 1, pp 401–405. IEEE
3. Basu S, Chaudhuri C, Kundu M, Nasipuri M, Basu DK (2007) Text line extraction from multi-skewed handwritten documents. Pattern Recogn 40(6):1825–1839
4. Bhowmik TK, Roy A, Roy U (2005) Character segmentation for handwritten Bangla words using artificial neural network. In: Proceedings of the IAPR TC3 NNLDAR
5. Boukharouba A (2017) A new algorithm for skew correction and baseline detection based on the randomized Hough Transform. Journal of King Saud University-Computer and Information Sciences 29(1):29–38
6. Brodić D, Milivojević ZN (2012) Estimation of the handwritten text skew based on binary moments. Radioengineering 21(1):162–169
7. Ghosh R, Mandal G (2012) Skew detection and correction of online Bangla handwritten word. Int J Comp Sci Issues 9(4):202
8. Gupta D, Bag S (2019) Handwritten multilingual word segmentation using polygonal approximation of digital curves for Indian languages. Multi Tools App 78(14):1–26
9. Guru DS, Ravikumar M, Manjunath S (2013) Multiple skew estimation in multilingual handwritten documents. Int J Comp Sci Issues 10(5):65
10. Guru DS, Suhil M, Ravikumar M, Manjunath S (2015) Small eigenvalue based skew estimation of handwritten Devanagari words. In: International conference on mining intelligence and knowledge exploration, pp 216–225. Springer
11. Jayadevan R, Kolhe SR, Patil PM, Pal U (2011) Database development and recognition of handwritten Devanagari legal amount words. In: Proceedings of the international conference on document analysis and recognition, pp 304–308. IEEE
12. Jundale TA, Hegadi RS (2015) Skew detection and correction of Devanagari script using Hough Transform. Proc Comp Sci 45:305–311
13. Jundale TA, Hegadi RS (2015) Skew detection of Devanagari script using pixels of axes-parallel rectangle and linear regression. In: Proceedings of the international conference on energy systems and applications, pp 480–484. IEEE
14. Kar R, Saha S, Bera SK, Kavallieratou E, Bhateja V, Sarkar R (2019) Novel approaches towards slope and slant correction for tri-script handwritten word images. The Imaging Sci J 67(3):159–170
15. Kavallieratou E, Fakotakis N, Kokkinakis G (2002) Skew angle estimation for printed and handwritten documents using the Wigner–Ville distribution. Image Vis Comput 20(11):813–824

16. Kumar R, Singh A (2010) Detection and segmentation of lines and words in Gurmukhi handwritten text. In: Proceedings of the international conference on advance computing conference, pp 353–356. IEEE
17. Liang Y, He F, Zeng X (2020) 3D mesh simplification with feature preservation based on whale optimization algorithm and differential evolution. Integrated Computer-Aided Engineering Preprint, pp 1–19
18. Liu S, Li M, Li M, Xu Q (2020) Research of animals image semantic segmentation based on deep learning. Concurrency and Computation: Practice and Experience 32(1):e4892
19. Liu S, Yu M, Li M, Xu Q (2019) The research of virtual face based on deep convolutional generative adversarial networks using tensorflow. Physica A: Statistical Mechanics and its Applications 521:667–680
20. Malakar S, Seraogi B, Sarkar R, Das N, Basu S, Nasipuri M (2012) Two-stage skew correction of handwritten Bangla document images. In: Proceedings of the international conference on emerging applications of information technology, pp 303–306. IEEE
21. Mei M, Zhong Y, He F, Xu C (2020) An innovative multi-label learning based algorithm for city data computing. GeoInformatica 24(1):221–245
22. Mello CarlosAB, Sánchez A, Cavalcanti GeorgeDC (2011) Multiple line skew estimation of handwritten images of documents based on a visual perception approach. In: Proceedings of the international conference on computer analysis of images and patterns, pp 138–145. Springer
23. Obaidullah SM, Halder C, Santosh KC, Das N, Roy K (2018) PHDIndic_11: page-level handwritten document image dataset of 11 official Indic scripts for script identification. Multi Tools App 77(2):1643–1678
24. Pramanik R, Bag S (2017) Linear curve fitting-based headline estimation in handwritten words for Indian scripts. In: Proceedings of the international conference on pattern recognition and machine intelligence, pp 116–123. Springer
25. Pramanik R, Bag S (2018) Linear regression-based skew correction of handwritten words in Indian languages. In: Proceedings of the international conference on computer vision & image processing, pp 129–139. Springer
26. Pramanik R, Bag S (2018) Shape decomposition-based handwritten compound character recognition for Bangla OCR. J Vis Commun Image Represent 50:123–134
27. Pramanik R, Bag S, Kumar R (2018) A fuzzy and contour-based segmentation methodology for handwritten Hindi words in legal documents. In: Proceedings of the international conference on recent advances in information technology, pp 1–6. IEEE
28. Pramanik R, Raj V, Bag S (2018) Finding the optimum classifier: Classification of segmentable components in offline handwritten Devanagari words. In: Proceedings of the international conference on recent advances in information technology, pp 1–5. IEEE
29. Roy A, Bhowmik TK, Parui SK, Roy U (2005) A novel approach to skew detection and character segmentation for handwritten Bangla words. In: Proceedings of the international conference on digital image computing: techniques and applications, pp 30–30. IEEE
30. Roy K, Roy K, Pal U (2006) Segmentation of unconstrained handwritten text based on RLSA algorithm. In: Proceedings of the national conference on recent trends in information systems, pp 196–199
31. Sharma MK, Dhaka VP (2016) Segmentation of English offline handwritten cursive scripts using a feedforward neural network. Neural Comput & Applic 27(5):1369–1379
32. Shaw B, Parui SK (2010) A two stage recognition scheme for offline handwritten Devanagari words. In: Machine interpretation of patterns: image analysis and data mining, World Scientific, pp 145–165
33. Shi Z, Govindaraju V (2003) Skew detection for complex document images using fuzzy runlength. In: Proceedings of the international conference on document analysis and recognition, p 715. IEEE
34. Stamatopoulos N, Gatos B, Louloudis G, Pal U, Alaei A (2013) ICDAR 2013 handwriting segmentation contest. In: Proceedings of the international conference on document analysis and recognition, pp 1402–1406. IEEE
35. Wu Y, He F, Zhang D, Li X (2015) Service-oriented feature-based data exchange for cloud-based design and manufacturing. IEEE Transactions on Services Computing 11(2):341–353
36. Xu Q, Huang G, Yu M, Guo Y (2020) Fall prediction based on key points of human bones. Physica A: Statistical Mechanics and its Applications 540:123205
37. Xu Q, Li M, Li M, Liu S (2018) Energy spectrum CT image detection based dimensionality reduction with phase congruency. J Medical Systems 42(3):49
38. Xu Q, Wang F, Gong Y, Wang Z, Zeng K, Li Q, Luo X (2019) A novel edge-oriented framework for saliency detection enhancement. Image Vis Comput 87:1–12
39. Xu Q, Wang Z, Wang F, Gong Y (2019) Multi-feature fusion CNNs for Drosophila embryo of interest detection. Physica A: Statistical Mechanics and its Applications 531:121808

40. Xu Q, Wang Z, Wang F, Li J (2018) Thermal comfort research on human CT data modeling. Multi Tools App 77(5):6311–6326
41. Yu H, He F, Pan Y (2020) A scalable region-based level set method using adaptive bilateral filter for noisy image segmentation. Multi Tools App 79(9):5743–5765
42. Zhang DJ, He FZ, Han SH, Li XX (2016) Quantitative optimization of interoperability during feature-based data exchange. Integrated Computer-Aided Engineering 23(1):31–50

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.