Text Line Segmentation from Struck-Out Handwritten Document Images

[1]Palaiahnakote Shivakumara, [2]Tanmay Jain, [2]Umapada Pal, [2]Nitish Surana, [3]Apostolos Antonacopoulos and [4]Tong Lu

[1]Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur-50603, Malaysia. shiva@um.edu.my.

[2]Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata, India.

Email: tanmayj000@gmail.com, umapada@isical.ac.in, surananitish@gmail.com

[3]Pattern Recognition and Image Analysis (PRImA) Research Lab, University of Salford, Greater Manchester, UK.
Email: a.antonacopoulos@primaresearch.org

[4]National Key Lab for Novel Software Technology, Nanjing University, Nanjing, China. Email: lutong@nju.edu.cn

Abstract

In the case of freestyle everyday handwritten documents, writing, erasing, striking out, and overwriting are common behaviors of the writers. This not cleanly-written text poses significant challenges for text line segmentation. Accurate text line segmentation in handwritten documents is essential to the success of several real-world applications, such as answer script evaluation, fraud document identification, writer identification, document age estimation and writer gender classification, to name a few. This paper proposes the first, to the authors' best knowledge, text line segmentation approach that is applicable in the presence of both cleanly-written and struck-out text. The approach consists of three steps. In the first step, components - at the word level - are detected in the input handwritten document images (containing both cleanly-written and struck-out text) based on stroke width information estimation, filtering of noise, and morphological operations. In the second step, the struck-out components are identified using the DenseNet deep learning model and treated differently to clean text in further analysis. In the third step, geometrical spatial features, the direction between candidate components and the overall text line, and the common overlapping region between adjacent components are evaluated to progressively form text lines. To evaluate the proposed steps and compare the proposed method to the state-of-the-art, experiments have been conducted on a new problem-focused dataset containing instances of struck-out text in handwritten documents, as well as on two standard datasets (ICDAR2013 text line segmentation contest dataset and ICDAR2019 HDRC dataset) to show the proposed steps are effective and useful, with superior performance compared to existing methods.

Keywords: Handwriting recognition, Writer identification, Connected Component Analysis, Deep learning, Struck-out words, Text line segmentation.

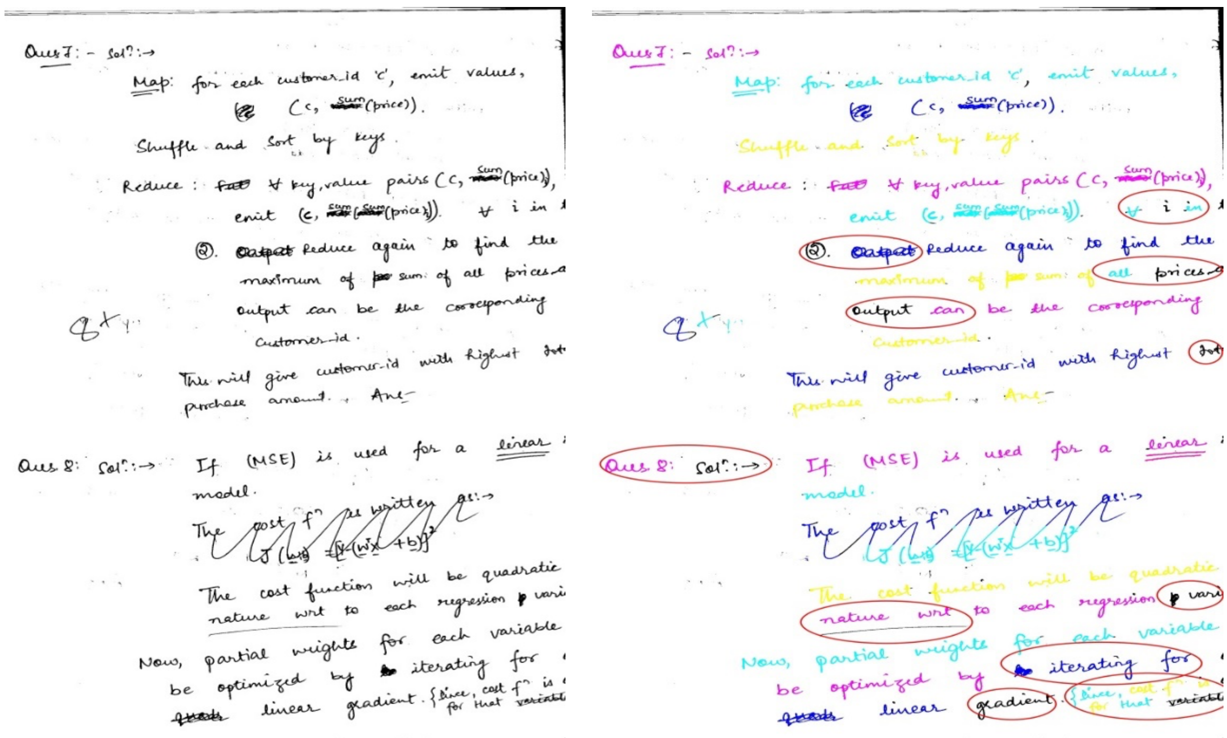
1. Introduction

Despite the considerable progress and solutions that have been developed for document digitization, several challenging problems remain in real-world applications that involve the analysis and recognition of every day handwritten text (e.g. as opposed to formal/historical manuscripts, say), such as the reading of handwritten notes, student answers for automated grading, person/writer identification, person behavior identification, form recognition, detection of forgeries, document age estimation, person age estimation and gender identification (Chaudhuri & Adak (2017); Brink et al. (2008); Shivakumara et al. (2019); Navya et al. (2018); Raghunandan et al. (2016); Basavaraj et al. (2019); Nag et al. (2018); Kundu et al. (2019)). When writing such everyday (freestyle) text, it is a natural human tendency to make some errors and to attempt to correct them by erasing, overwriting, and cross striking the wrong text. As a result, one can expect every day handwritten documents to contain evidence of the above-mentioned corrective actions.

The presence of such handwriting error corrections (struck-out text) in document images poses several challenges throughout the analysis and recognition pipeline, starting with text line segmentation and progressively affecting all subsequent steps. Text line segmentation is a particularly important step since any early errors there propagate and significantly affect the accuracy of the steps depending on it and, ultimately of the whole system. Therefore, text line segmentation from struck-out handwritten document images is a significant challenge, especially due to its intrinsic complexity. This challenge motivated the authors to develop a new approach for addressing the issues of text line segmentation from struck-out handwritten document images, as described in this paper.

Handwritten text line segmentation is not a new problem in document image analysis. Several methods can be found in the literature applied to a wide range of application domains e.g. (Li et al. (2019); Gader et al. (2020)). However, it is noted that existing methods have been developed for segmenting text lines from cleaned (with artifacts removed) document images, degraded document images, formal historical manuscript images, non-structured layout document images etc., but not for struck-out handwritten document images.

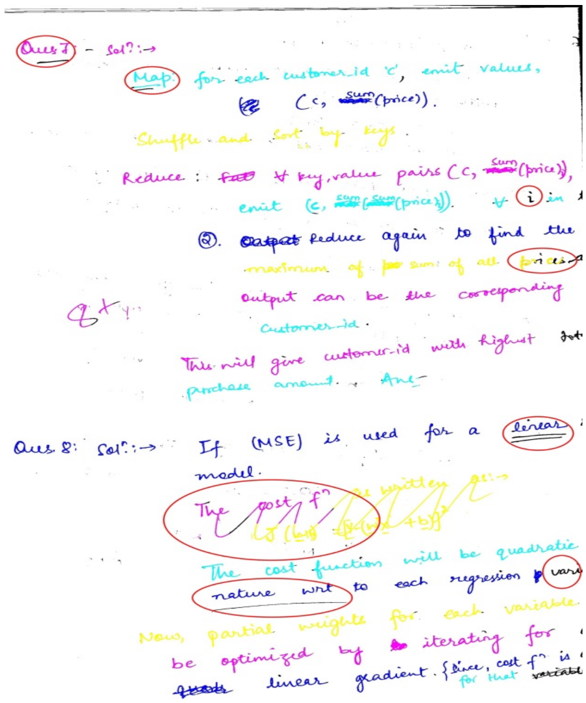
To illustrate visually the inadequacy of existing methods, an example of a typical freestyle handwritten document with struck-out text is shown in Fig. 1(a). Line segmentation results (with each text line shown in a different color and errors highlighted with red circles) from two prominent existing methods (Li et al. (2019); Gader et al. (2020)) are shown in Fig. 1(b) and Fig. 2(a). It is observed that those methods, which explore the connected component analysis approach and deep learning, respectively (two major categories of approach), do not segment text lines properly as they miss certain words. This is due to the presence of struck-out handwritten words in the input image, which affects the direction of text lines, creates touching text lines and changes the shape of the characters/words.



(a) Input freestyle handwritten document image with struck-out words (b) Text line segmentation by existing method (Li et al. (2019))

Fig. 1. Performance of existing method (Li et al. (2019)) for text line segmentation from struck-out handwritten document image (incorrect segmentations in (b) are marked by ellipse).

The approach proposed in this paper addresses this open research problem. As a preview example of the capabilities of the proposed method, it can be seen in Fig. 2(b) that, in contrast to the existing methods, it segments text lines accurately for the input image in Fig. 1(a), irrespective of the type of struck-out words. The different types of struck-out words considered in this work are listed in Table 1 and corresponding example images are shown in Fig. 3. It can be observed that it is not feasible to anticipate the nature of the struck-out text in freestyle handwritten documents. The reason for selecting the specific list of possible struck-out word types is that those types were observed in the answer scripts collected from the college students (this work is part of an automatic descriptive answer evaluation project). The variety is considered adequate and representative of struck-out word types in general.



(a) Text line segmentation by existing method (Gader et al. (2020)) (b) Text line segmentation by the proposed method.

Fig. 2. Performance of existing method (Gader et al. (2020)) and the proposed method for text line segmentation from the struck-out handwritten document image shown Fig. 1(a). (Incorrect segmentation results in (a) are marked by red ellipse).

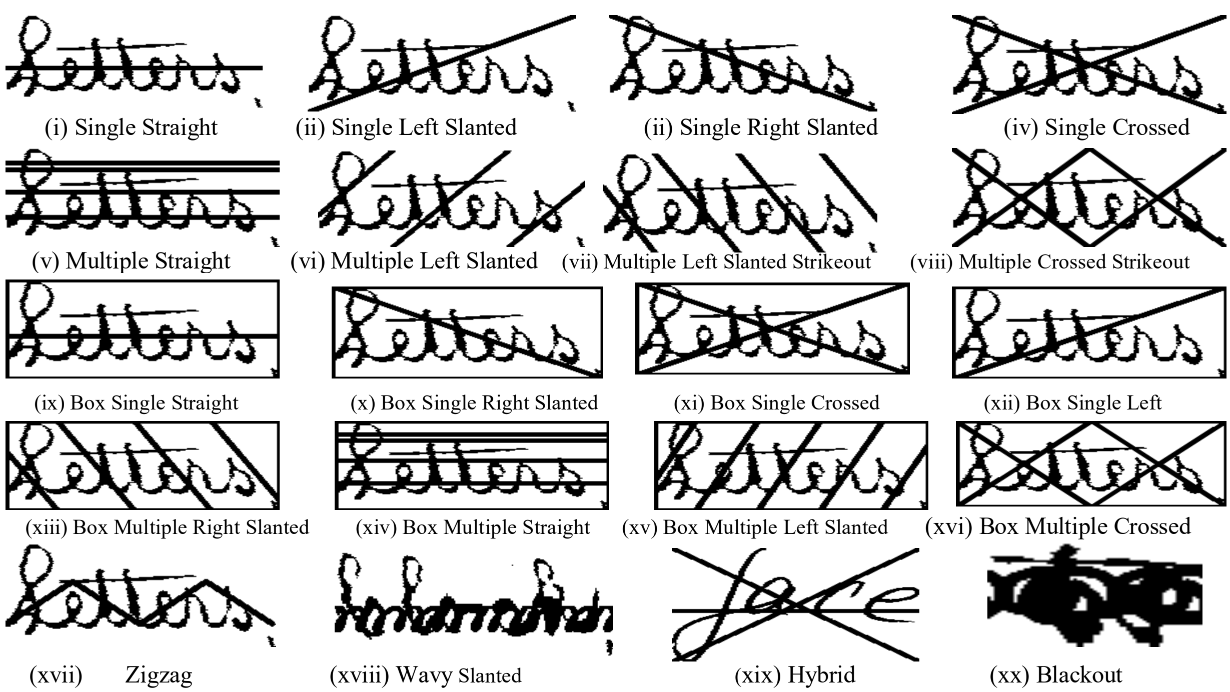


Fig. 3. Illustrations of the multiple types of struck-out handwritten words.

Table 1. Category of multiple types of struck-out handwritten words.

(i)	Single Straight Strikeout	(xi)	Box Single Slanted Right Strikeout
(ii)	Single Slanted Left Strikeout	(xii)	Box Single Crossed Strikeout
(iii)	Single Slanted Right Strikeout	(xiii)	Box Multiple Straight Strikeout
(iv)	Single Crossed Strikeout	(xiv)	Box Multiple Slanted Left Strikeout
(v)	Multiple Straight Strikeout	(xv)	Box Multiple Slanted Right Strikeout
(vi)	Multiple Slanted Left Strikeout	(xvi)	Box Multiple Crossed Strikeout
(vii)	Multiple Slanted Right Strikeout	(xvii)	Zigzag Strikeout
(viii)	Multiple Crossed Strikeout	(xviii)	Wavy Strikeout
(ix)	Box Single Straight Strikeout	(xix)	Hybrid Strikeout
(x)	Box Single Slanted Left Strikeout	(xx)	Blackout Strikeout

To the best of the authors' knowledge this is the first work which addresses the challenges of text line segmentation in both struck-out and normal handwritten documents, therefore making it a highly effective and practical approach in the real world. The key contributions of the proposed method are the following: (i) Exploiting the combination of hand-crafted features with a deep learning model is novel for component detection (components can be either struck-out or cleanly written words) and struck-out component classification. (ii) The use of geometric spatial relationships between word-level components and the general direction of the text lines for text line segmentation, in the presence of struck-out components of multiple types is novel compared to existing methods.

The structure of the remainder of this paper is as follows. Existing methods in the related fields of struck-out word classification and regular text line segmentation from handwritten documents are reviewed in Section 2. Section 3 describes the proposed approach by detailing its steps for component detection, struck-out component classification and text line segmentation. Experiments to validate the proposed steps and methods along with comparing them against existing methods on a new focused dataset and on standard datasets are presented in Section 4. Section 5 concludes the proposed work by summarizing and discussing the contributions of the proposed approach.

2. Related Work

Word detection and line segmentation are well-known problems in document analysis. Several connected component-based approaches, for instance, have been proposed to address the challenges in both topics. However, there are comparatively very few methods in the literature which have been developed for struck-out component detection and classification. It is also noted that none of the methods developed for handwritten text line segmentation consider struck-out text/components during line segmentation. Therefore, the literature review below focuses on methods for struck-out component detection and on methods for text line segmentation from handwritten document images.

2.1. Struck-Out Words Detection and Classification

Since the field of struck-out and non-struck out handwritten word classification is in its infancy, the literature contains hardly any such methods. Brink et al. (2008) proposed a method for the automated removal of crossed-out handwritten text and studied its effects on writer verification and identification. Their method used connected component analysis, involving branching-size features of the text, and a classifier for determining the crossed-out words in the document images. However, the method does not focus on multiple types of struck-out words as considered in the approach proposed in this paper. Adak et al. (2014) used a connected component analysis approach and constructed graphs using the intersection points to study the degree of straightness in order to identify the struck-out words in the handwritten document images. However, that method is not robust to noise and generally degraded documents because performance is dependent on finding correct intersection points. Chaudhuri et al. (2017) focused on the cleaning of struck-out handwritten words in the document images based on a combination of pattern classification and graph-based approaches. Their method first uses feature-based binary classification for detecting struck-out words in the images. Then a graph-based approach is used for identifying the pixels that represent strike lines over the text. The success of that method depends on the success of several stages and the on validity of a considerable number of heuristics; it is therefore reasonable to conclude that the overall approach may not perform well for varying kinds of struck-out words. Adak et al. (2017) proposed a hybrid method which combines an SVM classifier and a convolutional neural network for classifying struck-out handwritten words. In addition, the method examines the impact of struck-out words on writer identification. However, it is not clear whether the method will be applicable to a wide range of types of struck-out words, or it is rather limited to a particular type of struck-out words assumed in that work.

The methods discussed above use either handcrafted features or the combination of deep learning and handcrafted features for struck-out words classification. It should be noted that the main objective of those methods is to improve the recognition of handwritten text by removing struck-out words from the document image. However, Nisa et al. (2019) proposed a deep learning-based method for recognizing handwritten text in the presence of struck-out text. In their method, a convolutional recurrent neural network-based approach has been explored. That method works well in its overall goal of improving recognition performance, but it is not designed to benefit and improve the performance of other handwritten text based applications, such as answer script evaluation, gender identification and fraudulent document identification. Similarly, Qi et al. (2020) proposed a deep learning-based method for weakly supervised link artifact removal in document images. That method is based on the fact that text regions can have random ink smudges or spurious strokes. For the purpose of removing such ink occurrences, a deep learning-based approach was proposed, named DeepErase. However, the scope of that method is confined to simple struck-out handwritten words and does not expand to complex struck-out words in freestyle documents such as the example shown in Fig. 1(a).

Overall, although previous methods have been developed for struck-out handwritten word detection and classification, none of those methods are applicable to the classification the wide variety of types of struck-out words that can be found in everyday handwritten documents. Existing methods are mostly designed to work well only for text affected by horizontal strike-out lines. This particular assumption in those methods is not true in reality because the type of strike-out lines present depends on the individual writer who may have a particular style of crossing out text. In addition, none of the existing methods considers both struck-out and clean handwritten documents, making them impractical in several real-life application scenarios, as mentioned earlier. Hence, as part of the overall approach proposed in this paper, a novel and practical method will be described for classifying struck-out handwritten words irrespective of the type of strike-out line.

2.2. Text Line Segmentation

Romero et al. (2015) examined the influence of text line segmentation on handwritten text recognition. That paper discussed various segmentation methods to addresses challenges of historical documents to improve handwritten text recognition. Ryu et al. (2015) proposed a word segmentation method for handwritten documents based on structured learning, which considers the gap between words and text lines for segmentation. Mullick et al. (2015) proposed an efficient line segmentation method for handwritten Bangla documents, which uses blurred images for finding seed points that represent spaces between text lines. Kesiman et al. (2016) proposed a new scheme for segmenting text lines and characters from grayscale images of palm leaf manuscripts, which splits documents into vertical zones in order to examine holes and spacing between characters or text lines. Garz et al. (2017) proposed a user-centered method for segmenting complex historical manuscripts based on document graphs, which captures a sparse representation for line segmentation. Nhat and Lee (2016) proposed a method which uses dense predictions for text line segmentation in handwritten document images and uses convolutional neural networks and line adjacency graphs. Zhu et al. (2017) proposed text segmentation using super-pixel clustering, which utilizes stroke pixels and density based spatial clustering. Baig et al. (2018) proposed automatic segmentation and reconstruction of historical manuscripts in the gradient domain. Their method focuses on separating text from background in a noisy environment. Vo et al. (2018) proposed text line segmentation in handwritten document images using a fully convolutional network. The method explores the use of a CNN for segmenting text lines by learning character information and constructing adjacency graphs. Choudhury et al. (2019) proposed exploiting the force alignment of time–reversed data for improving the HMM-based handwriting segmentation. The method uses a Hidden Markov Model for segmenting text lines from handwritten documents. However, the scope of the method is to improve text recognition and does not provide information for other application scenarios, such as fraudulent document identification. Li et al. (2019) proposed a novel method for text line segmentation in historical documents written in Uchen Tibetan script. The method explores the idea of baseline detection for text line segmentation. However, the scope of the method is limited to the particular type of documents it is targeted at. Gadar et al. (2020) proposed text line segmentation based on deep learning models for Arabic handwritten document images. The main focus of their method is to address the challenges of Arabic text line segmentation and not any other scripts. Kundu et al. (2020) proposed text line extraction from handwritten document images using a GAN-based approach. The method explores a deep learning model for addressing complex issues like touching line, non-uniformly spaced text and multi-skewed text lines. However, it is not clear whether that method is applicable to documents affected by other issues. Renton et al. (2018) proposed a fully convolutional network with dilated convolutions for handwritten text line segmentation. The method explores deep learning models for learning features and text line segmentation. It is noted that the method may not work well for arbitrarily oriented text in degraded images.

Overall, most of the existing methods mentioned above use direct or indirect (e.g. through connected components) projection profiles and zoning (dividing the text line into upper, lower and middle zones) for segmenting text lines from handwritten documents. There are also methods which rely on deep learning models to overcome the problems of hand-crafted features-based methods. However, the scope of existing methods does not include handwritten text words containing instances of struck-out text. Struck-out text adversely affects both projection profiles and zoning as the regularity of the words and text lines is significantly altered. Moreover, the presence of such text causes touching lines, changes the shape of the text components and affects the direction of the text lines. It can therefore be concluded that existing methods may not perform well in handwritten documents affected by struck-out text. It is the authors' view therefore, that text line segmentation from struck-out handwritten document images has remained an open problem and that there is a corresponding need for developing a new text line segmentation method as a practical solution that will be applicable both to documents with clean and with struck-out handwritten text.

3. Proposed Methodology

To be practical in the real world, initial input for the overall proposed text line segmentation approach is considered to be images both of cleanly handwritten documents and of freestyle handwritten documents containing struck-out text. In other words, the proposed approach is agnostic of the input in terms of whether struck-out text is present or not.

It can be observed that for any handwritten text line, the distance and the direction between words in the same line share a unique spatial relationship irrespective of the script, orientation (e.g. whether the text is written predominantly horizontally or vertically), and of the type of struck-out words. The proposed approach, therefore, examines such reliable features at the word level (as opposed to the character level) in order to successfully address the challenges posed by struck-out text within the handwritten documents.

In order to capitalize on the above observations, the proposed approach is organized into three steps, namely: (i) *component detection* by exploring stroke width information, a component being either a struck-out or a cleanly-written word, (ii) *classification between struck-out and cleanly-written components* by exploring a deep learning model, and (iii) *text line segmentation* by exploring spatial relationship between the components.

The combination of stroke width information and connected component analysis enables the first step to detect components (struck-out and cleanly-written words. Inspired by the deep learning models that have had considerable success in solving complex classification problems (Nisa et al. (2019)), DenseNet is employed then for the classification of struck-out and non-struck-out components. Finally, the spatial relationship between (word-level) components within text lines is estimated by evaluating the distance between centers of gravity of adjacent components, the direction of the notional line segments connecting the centers of gravity of a given component with its two neighbors either side, and the projected region overlap between adjacent components. The latter process combines neighboring word components into text lines. It should be noted that, since those three extracted features represent global information at the word level, the line segmentation step has the ability to accurately handle the challenges of both clean handwriting and that containing struck-out text.

The block diagram illustrating the three steps (dashed line rectangles) and flow of the proposed approach can be seen in Fig. 4, where the individual operations in each step are also indicated.

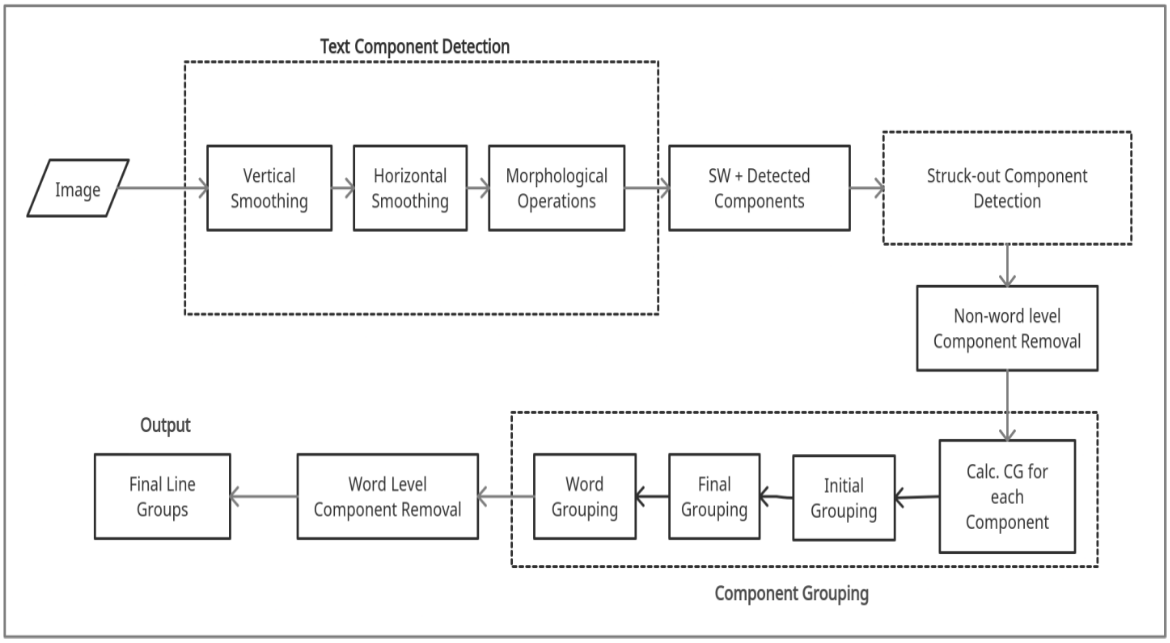


Fig. 4. Block diagram of proposed approach. Here SW denotes Stroke Width and CG denoted center of gravity.

3.1. Component Detection

For a given input handwritten document image, the proposed method first estimates the Stroke Width (SW) by identifying runs of (consecutive) black pixels across each row of the image, counting their length, and calculating the mode of those run-lengths of black pixels. This (mode) is used to represent the stroke width of the handwritten text on the page. This stroke width is used as a threshold (referred to as SW below) for performing vertical and horizontal smoothing operations (mitigating the effects of noise), resulting in word-level components (agnostic of whether the corresponding words were struck-out or cleanly-written in the input image) as described below.

Vertical Smoothing: If the length of *black* (value is 0) pixel runs present in the image when traversing it *vertically* is found to be less than or equal to the stroke width (SW), those pixels are considered as noisy pixels and hence are discarded by replacing the pixels in those runs with white (value is 1) pixels which represent the background of the image as defined in Equation (1). Otherwise, the pixels are considered to be part of the foreground which is represented by black pixels as defined in Equation (1).

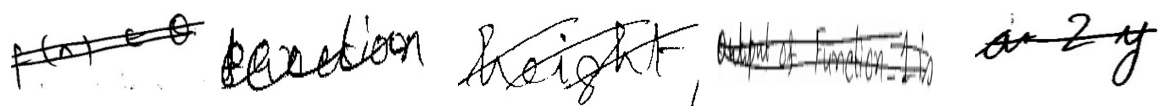
$$TextRegion\,(x) = \begin{cases} 1, & Run\_Black\_Pixels(x) \leq SW \\ 0, & Run_{Black_{Pixels(x)}} > SW \end{cases} \tag{1}$$

Horizontal Smoothing: If the length of *white* (value is 1) pixel runs present in the image when traversing it *horizontally* is less than or equal to five times the stroke width (SW), the pixels are considered as part of a text region and hence the pixels are replaced by black (value is 0) pixels, which represent foreground as defined in Equation (2). Otherwise, the pixels are considered as noisy pixels and the pixels are replaced by background pixels (white pixels, value 1) as defined in Equation (2). The value of the multiplier 5 is determined empirically by experimenting with the method on random samples.

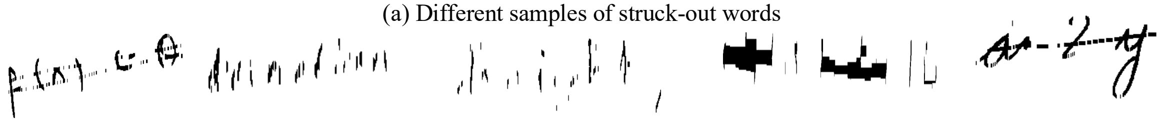
$$TextRegion(x) = \begin{cases} 0, & Run_{White_{Pixels}} \leq 5 \times SW \\ 1, & Run_{White_{Pixels}} > 5 \times SW \end{cases} \tag{2}$$

Morphological Operations: After the two operations above, a morphological opening operation is performed using a rectangular kernel of (SW, SW×6) to merge all the characters of each word into a single component. The value of 6 is estimated based on experimentation.

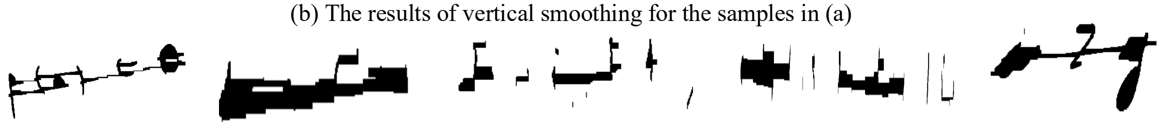
The effect of the above three operations is illustrated in Fig. 5 and Fig. 6 for non-struck-out and struck-out words, respectively. For the different struck-out and non-struck-out words shown in Fig. 5(a) and Fig. 6(a), the results of vertical, horizontal smoothing and morphological operations are shown in Fig. 5(b-d) and Fig. 6(b-d), respectively. The results in Fig. 5(d) and Fig. 6(d) indicate that the above operations work sufficiently well for both struck-out and non-struck-out words in terms of merging the characters of the words into single components. It can also observe from Fig. 5(d) and Fig. 6(d) that the three above operations emphasize the difference between the better preserved (compared to the input) pattern of the non-crossed out words and that of the struck-out words, which is almost lost. This differentiation effect facilitates the classification between struck-out and non-struck-out components (exploited in the following steps of the proposed approach).



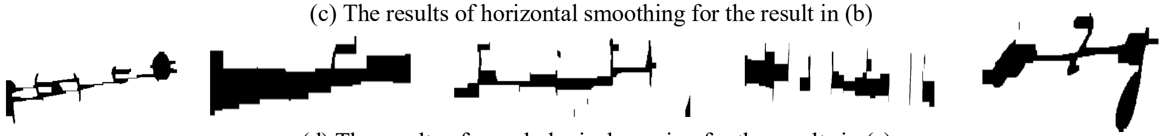
(a) Different samples of struck-out words



(b) The results of vertical smoothing for the samples in (a)

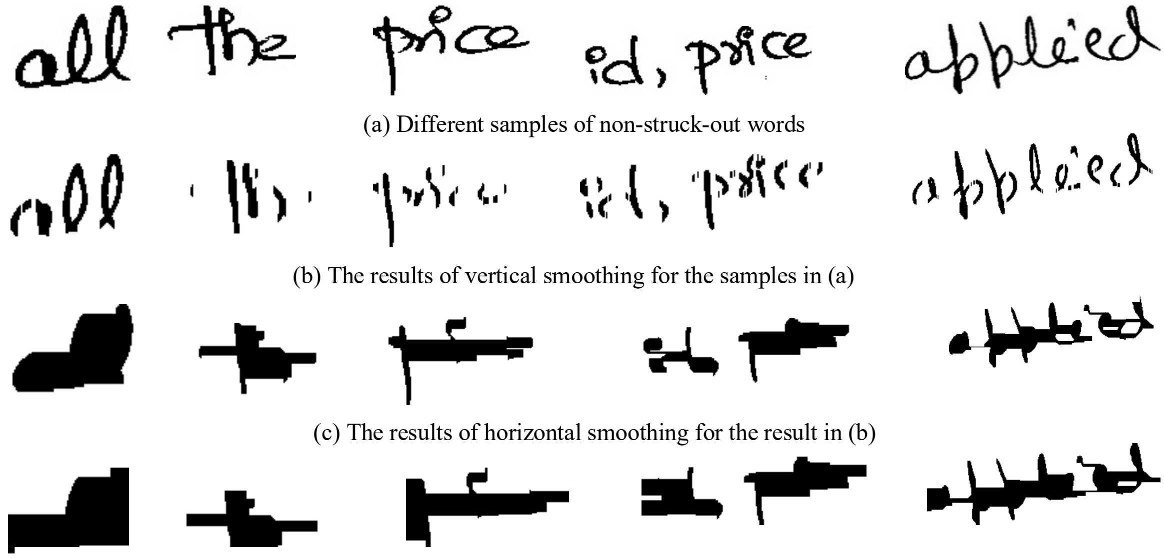


(c) The results of horizontal smoothing for the result in (b)



(d) The results of morphological opening for the results in (c)

Fig. 5. The effect of vertical and horizontal smoothing for struck-out components.



(a) Different samples of non-struck-out words

(b) The results of vertical smoothing for the samples in (a)

(c) The results of horizontal smoothing for the result in (b)

(d) The results of morphological opening for the results in (c)

Fig. 6. The effect of vertical and horizontal smoothing for non-struck-out components.

3.2. Struck-Out Component Detection

The step presented in the previous section outputs the coordinates of bounding boxes around components regardless of whether they are struck-out or not. In this step, a deep learning model is trained and used to detect struck-out components by classifying each component (referenced by its bounding box) as struck-out or cleanly-written.

As mentioned in the introduction of this paper, one cannot expect the occurrence of only a particular type of struck-out words, which makes the classification between struck-out components and cleanly-written ones challenging. Additionally, the handwritten documents from which the components were collected for the dataset had a far larger number of non-struck-out components compared to struck-out components. Hence, in order to balance the number of struck-out and non-struck-out components for training the deep learning model, a synthetic struck-out dataset has been created using the IAM Handwriting (Matri & Bunke (2002)) and Washington (Frinken et al. (2012)) manuscript datasets (more information is presented later in Section 4.1). In total, this new dataset contains nearly 100,000 samples, including struck-out and cleanly-written components for training the proposed deep learning model. The data was split into a ratio of 80:20, i.e. 80% of data was used for training the model, and the remaining 20% was used for testing/validating the model.

To create the labelled data required for training the model, the words are manually segmented in the document pages and different types of struck-out words are generated automatically. For testing, the proposed method considers a full page containing the struck-out and non-struck-out words as input and first employs the step of component detection to separate the words by identifying bounding boxes for each word. Therefore, the segmented words along with the bounding boxes are input to the model (described below) for struck-out word detection.

The classic DenseNet121 architecture was used with certain modifications as shown in Fig. 7, where the architecture diagram of the model is presented. More specifically, the last fully connected layers present in the classic DenseNet architecture have been replaced with a fully connected layer of 1024 units. This fully connected layer is followed by a single output unit using the sigmoid activation function that provides the probability of the component being struck-out or not. Weights were pre-trained on the ImageNet dataset, using the Stochastic Gradient Descent (SGD) as the optimizer with a learning rate of 0.001 and binary cross-entropy as the loss function. The details of the proposed DenseNet architecture are as follows.

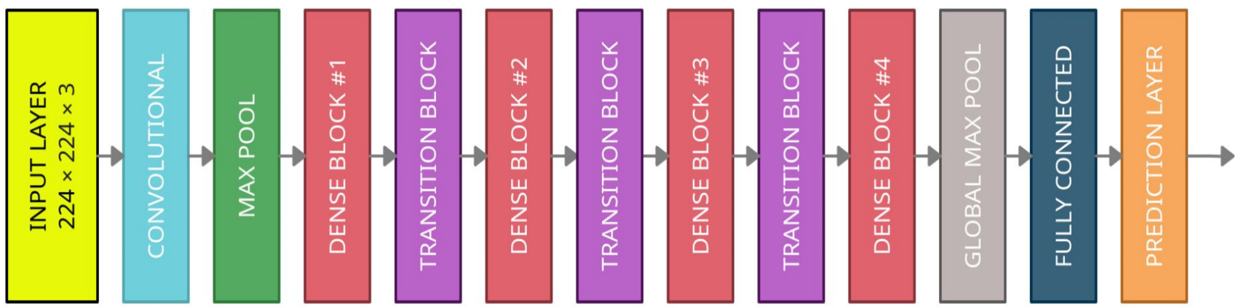
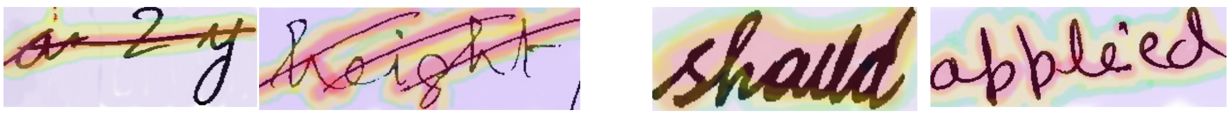


Fig. 7. The proposed deep learning DenseNet architecture for struck-out word classification.

Each of the dense blocks used in the architecture consists of a stack of densely connected convolutional layers. It can extract more efficient features than plain and residual convolutional networks. It is built around a batch normalization layer, a ReLU layer and a convolutional layer. Dense Block#1 contains the BN-ReLU-Conv(1x1)-BN-ReLU-Conv(3x3) architecture repeated 6 times, whereas the Dense Blocks #2, #3, and #4 contain the same architecture repeated 12, 24 and 16 times, respectively. The size of the feature maps after passing through each dense block remains the same. Moreover, since it is important to reduce the size of the feature map size in a convolutional network, a transition block is applied between two dense blocks to decrease the size. All the transition layers shown in the architecture in Fig. 7 consist of a batch normalization layer and a 1×1 convolutional layer, followed by an average pooling layer with a 2×2 kernel.

The input image is first passed through a convolutional layer containing a 7×7 kernel followed by a 3×3 max pooling layer. After this, the image is passed through multiple dense blocks and transition blocks before being passed through a 7×7 global max pooling layer. Next, the image is passed through a fully connected layer containing 1024 units with a ReLU activation function and finally sent to the prediction layer containing one unit with a sigmoid activation function that outputs the probability of whether the given connected component is struck-out or not. To demonstrate the effect of the DenseNet121 architecture in the classification of struck-out and non-struck-out components, the activation maps for representative examples of struck-out and non-struck-out words are presented in Fig. 8(a)-(b), respectively. It is evident from Fig. 8(a) that the strike-out strokes/regions received maximum activation when the model predicted the components to be struck-out. This means the classifier successfully learned to identify the presence of strike-outs on text. On the other hand, Fig. 8(b) shows when the model predicted components as non-struck out, the entire region covering the non-struck out component received activation. Sample results of struck-out and non-struck out word classification are shown in Fig. 9, where it can be seen that bounding boxes are fitted to all the struck-out words in the document images.



(a) Activation map for different struck-out components

(b) Activation map for non-struck-out components

Fig. 8. Activation maps of DenseNet121 architecture for struck-out and not-struck-out components.

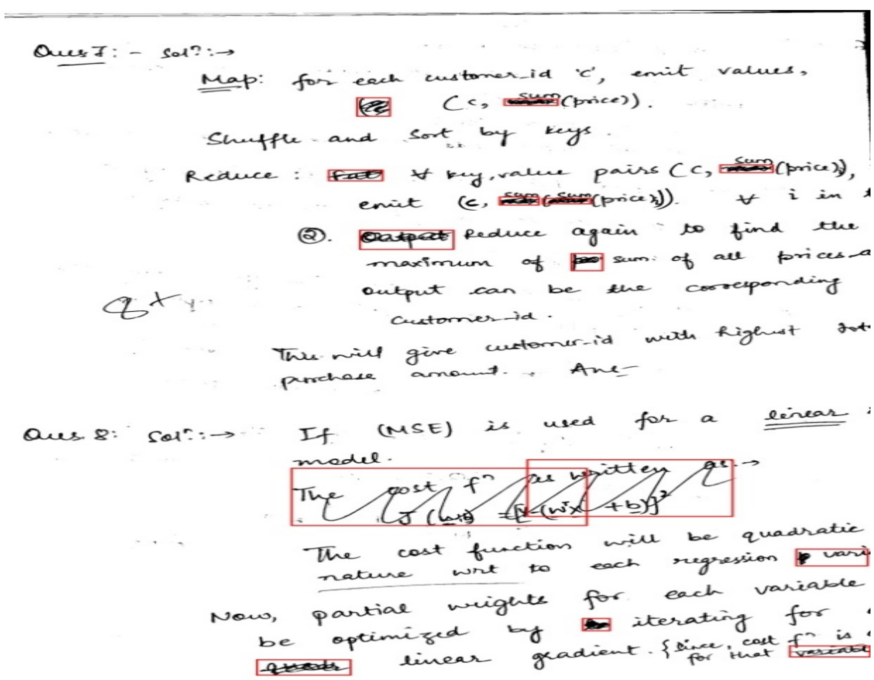
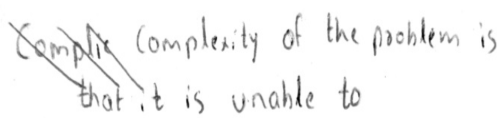


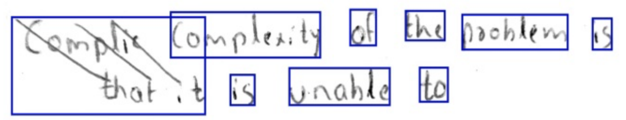
Fig. 9. Example results of the struck-out word detection step of the proposed method. Bounding boxes are shown around the struck-out text components.

One of the advantages of the proposed method is that it has ability to detect struck-out words accurately even when stuck-out words are touching words of another text line, as shown in the example of Fig 10(a).

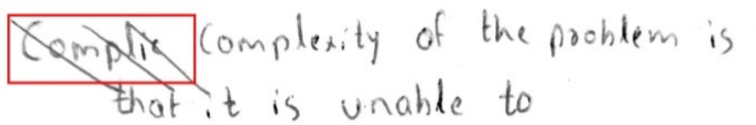
In this case, the component detection step initially fits a bounding box around both the struck-out word and the touching words, as shown in Fig. 10(b). However, the proposed model finally detects the struck-out words accurately without the other words as shown in Fig. 10(c). This is because the deep learning model is trained with clear ground truth of both struck-out and non-struck-out words, so it finds the words within merged components that have a good match with the ground truth. In this way, the struck-out component detection step reduces the complexity of text line segmentation, especially in the presence of struck-out words which are touching words in other text lines.



(a) Sample of touching lines cases



(b) The result of component detection



(c) The results of struck-out component detection

Fig. 10. The struck-out component detection in the presence touching lines.

3.3. Line Segmentation

If the input image contains struck-out words, with the help of the step presented in the previous section, the proposed method ignores struck-out words when extracting features for text line segmentation step.

For the set of non-struck-out components, the proposed method extracts the distances between component pairs and the direction/alignment between components in the following way.

For each non-struck-out component, the Center of Gravity (CG) is calculated as defined in Equation (3).

$$CG(x, y) = \left(\left(\frac{1}{n} \times \sum x_i\right), \left(\frac{1}{n} \times \sum y_i\right)\right) \quad (3)$$

where, n is the total number of black pixels in the bounding box, x_i *is the* x-coordinate of black pixel i in the bounding box, and y_i *is the* y-coordinate of black pixel i in the bounding box.

Then, using the CG, the proposed method finds the Left-most Point (LP) and the Right-most Point (RP) for each component as shown in Fig. 11. The LP and RP points are used to estimate the distance between the current and adjacent candidate components to be merged with it. For instance, the Euclidean distance between the RP of the current component and the LP of its right candidate component is computed to calculate the distance between the components. Similarly, the Euclidean distance between the LP of the current component and the RP of its left candidate component is computed to calculate the distance between the two components. It is expected that the distance between any two adjacent components within the same text line is very similar unlike the distance between them and other components belonging to different text lines. The slope of the notional line segment between the CG of the current component and the CG of its right candidate component is calculated to compute the slope difference between the two components. Similarly, the slope of the notional line segment between the CG of the current component and CG of its left candidate component is also calculated. The idea underpinning this angle-based feature is that the angle between a pair of adjacent components within the same text line must be close to the angle of the overall text line (zero if the text line is completely horizontal), while the angle between the components in different text lines should be very far in magnitude from the overall angle of their respective text lines.

In addition, to further evaluate whether a pair of components lies within the same text line, the proposed method divides each component into two halves by splitting it vertically at the CG as shown in Fig. 12(a). The region between the left half of the current component and the right half of the candidate component to its left is considered as a common region for the current and candidate components as shown in Fig. 12(b). Next, the percentage of pixels in the common region that are horizontally common to both the components is calculated. If all the three features in a component pair satisfy a certain threshold, the two components are included into one group.

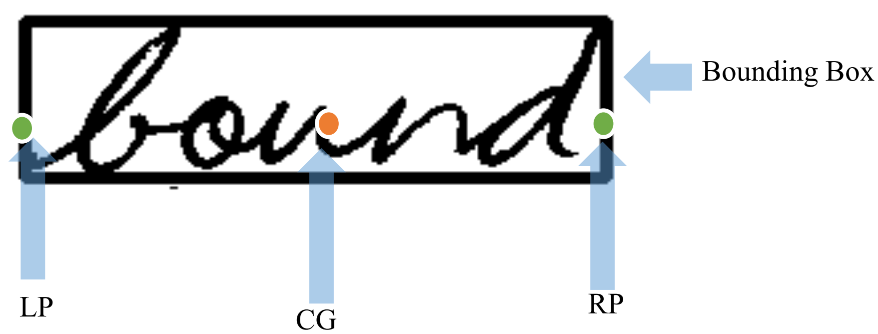
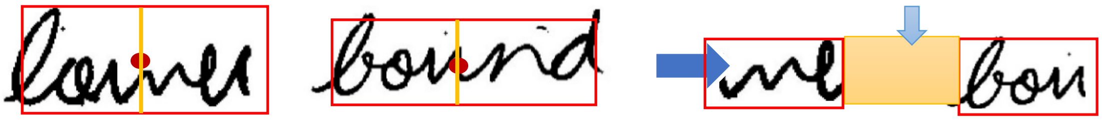


Fig. 11. Examples of the Left-most Point (LP) boundary, Center of Gravity (CG), and Right-most Point (RP) boundary in a component.



(a) Split the Bounding Box Vertically at CG (b) Common Horizontal Pixels

Fig. 12. Extracting the overlap feature for grouping the components of the same text line

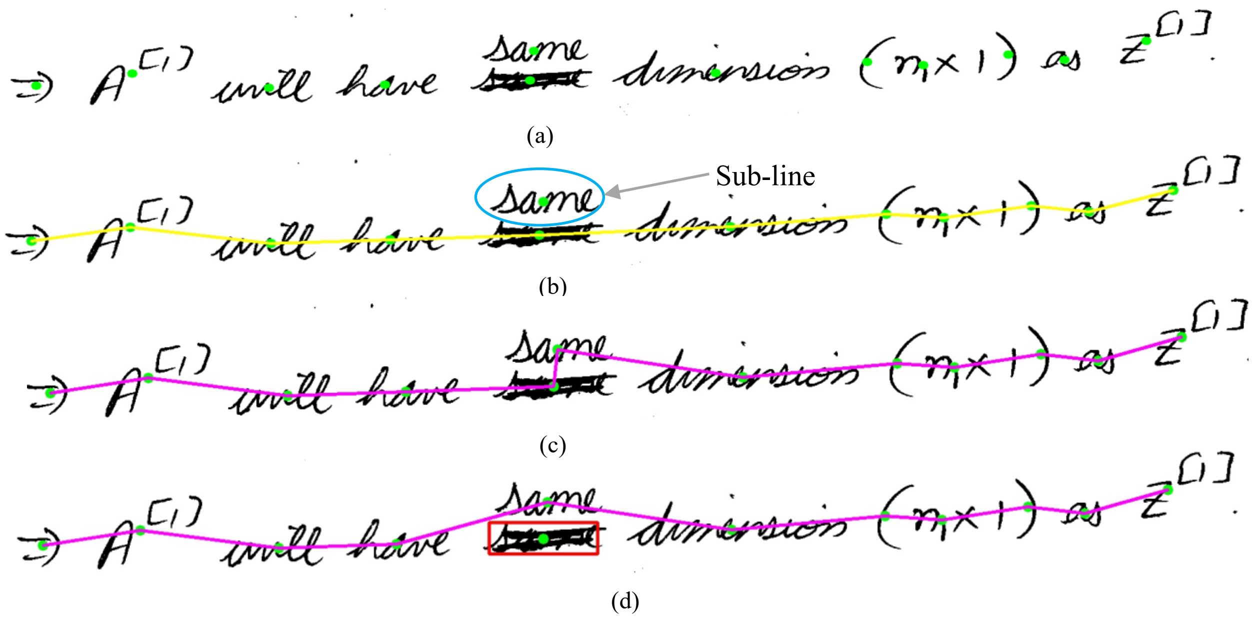


Fig. 13. The process of text line segmentation in the presence of struck-out components.

This process results in multiple groups of two components. To further group components which are within the same text line, the above three features are extracted for the components of a pair of groups. If the extracted features satisfy the criteria of mutual nearest neighbor groups, the components in the two groups are considered as components within the same text line. This process continues for all the groups in the document image as illustrated in Fig. 13(a)-(d), where we can see the CG for each word including a struck-out word in the text line as shown in Fig. 13(a). Multiple groups (the word "same" is considered as one sub-line/group) of the text line can be seen in Fig. 13(b). The results of the mutual nearest neighbor grouping can be seen in Fig. 13(c). Next, taking into account the results of the struck-out component classification step, the proposed method alters the grouping process to exclude the struck-out component and arrive at the final text line segmentation result, as shown in Fig. 13(d). This is the advantage of struck-out component classification for text line segmentation. The result of text line segmentation for the input image in the running example is shown in Fig. 14, where it can be observed that all the text lines are segmented correctly.

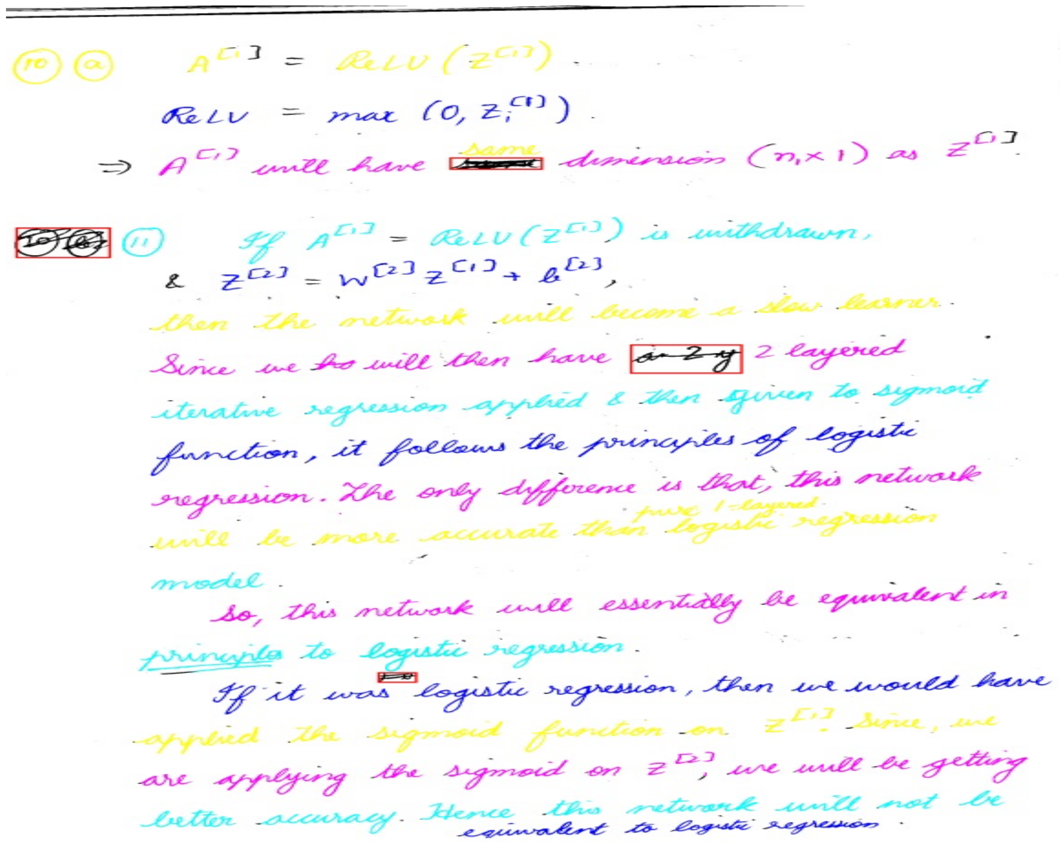
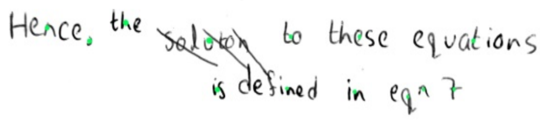


Fig.14. The results of text line segmentation of the proposed method for the input image

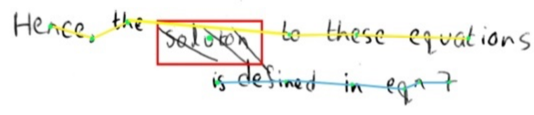
The effectiveness of the text line segmentation is shown Fig. 15(a)-(b) and Fig. 16(a)-(b), where Fig. 15 presents the successful text line segmentation for the touching components seen previously in Fig. 10 and Fig. 16 presents the successful text line segmentation in the presence of an entire crossed-out line. It is observed from Fig. 15(b) and Fig. 16(b), that the struck-out component detection step detects words accurately, irrespective of touching components and an entire line being crossed-out. When struck-out words are detected, the text line segmentation step ignores the struck-out words for extracting all the above features, and instead uses its spatial information to find the next nearest neighbor.

A key issue that must be taken into account in finding the next nearest neighbor is that in general, when we cross out the words, we write the correct words on the top of struck-out words, while writing answers to questions in an exam for instance. The struck-out words indicate the possible nearest neighbor on top of the word. With this information, the process of grouping components into lines is modified such that text lines are segmented accurately as shown in Fig. 15(b) and Fig. 16(b).

Overall, the advantage of detecting struck-out words first is that it helps the text line segmentation step to improve its overall performance, especially when the words are misaligned with the text lines (e.g. when correct words are written on top of the struck-out words) and when the struck-out words are connected to words of another text line.

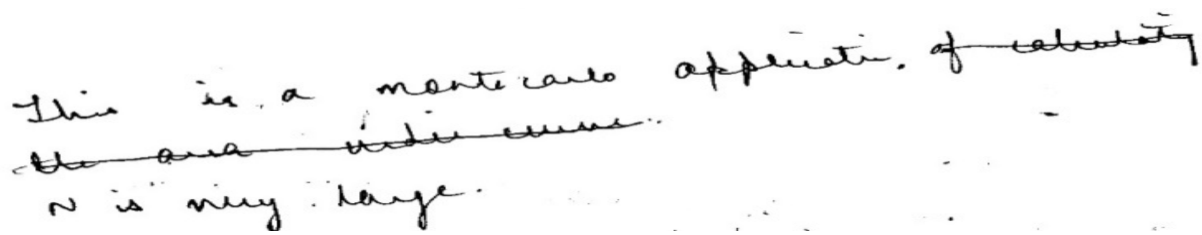


(a) Sample of struck-out word touches another word

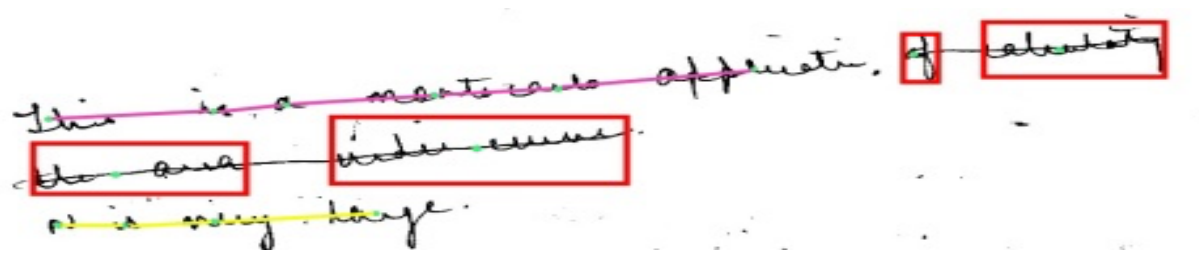


(b) The results of text line segmentation

Fig. 15. Text line segmentation in the presence of touching struck-out components.



(a) Sample of the whole crossed out line



(b) The results of text line segmentation

Fig. 16. Text line segmentation in the case where an entire text line has been struck-out.

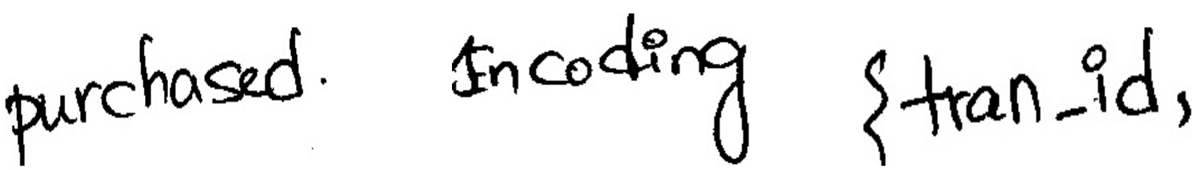
4. Experimental Results

Since text line segmentation from struck-out handwritten documents is a novel area of work, a suitable dataset for evaluating the proposed method is required. A new dataset was constructed for detailed and focused experimentation and evaluation of the work described in this paper. The dataset and the evaluation metrics are described in Section 4.1. The proposed approach was also evaluated on two benchmarking datasets, namely, the ICDAR2013 handwriting segmentation contest (Stamatopoulos et al. (2013)) and the ICDAR2019 HDRC-Chinese (Christlein et al. (2019)) datasets. To assess the effectiveness of the key steps proposed in the work, an ablation study was conducted, the observations of which are included in Section 4.2. In Section 4.3, the results of experiments are described to validate the component detection step from struck-out handwritten document images. Section 4.4 presents an evaluation of the struck-out and non-struck-out component classification step, contrasted with existing methods. Finally, the results of the experimental analysis for evaluating the proposed text line segmentation method are included in Section 4.5.

4.1. Dataset Construction and Evaluation

Since the main objective of the work described in this paper was to address the issues of text line segmentation in freestyle handwritten documents (which frequently contain struck-out text), 50 pages were chosen randomly from a set of student answer sheets, for experimentation. For generating ground truth annotations, bounding boxes were fitted manually to each word and each component was labeled as struck-out or non-struck-out in all 50 answers. Naturally, the number of struck-out components is lower than the number of non-struck-out components. To balance the size of struck-out and non-struck-out component classes, we generated a synthetic dataset to augment the struck-out component class using the IAM handwriting dataset (Marti & Bunke, (2002)) and the Washington manuscript dataset (Frinken et al. (2012)). The IAM handwriting dataset includes instances of handwritten English text that are widely used for training in text recognition applications. It contains over 1,500 pages of scanned text written by over 650 writers. The Washington manuscript dataset includes pages from the George Washington Papers found in the Library of Congress. It contains over 4000 word-level segmented text images. The created dataset will be made available to researchers to support research reproducibility.

(a) Sample struck-out word images created manually



(b) Sample non-struck-out word images created manually

Fig. 17. Labeling struck-out and non-struck-out words manually for constructing the dataset.

The main challenge during the creation of a synthetic dataset containing struck-out components was replicating how people perform strike-out lines on the actual answer sheet. Many factors were considered while creating the dataset, such as, the stroke width used by a person, different patterns of strike-out lines used, the placement of strikeout, etc. For example, in case of the single straight strikeout, a person does not always strike out the text across the center of the word; he/she generally strikes out the text such that most of the text which needs to be struck-out is covered. To synthetically generate such patterns in the clean text, factors such as the position of horizontal maxima of black pixels in the image are considered. In many cases, people use a continuous wave-like pattern / spiral-pattern to strikeout the text. Such patterns need to be synthetically created over the clean data as well. When a person strikes out the text in a multiple-slanted right/left pattern, all these lines in the image are generally non-parallel. Hence, an uncertainty factor also needs to be added while synthetically creating this pattern over clean data. All these factors are considered while creating a synthetic dataset for strikeouts as shown in sample images in Fig. 17, where we can see samples of both struck-out and non-struck-out components. A total of 1,000 images were used in the test dataset for experimentation comprising 50% images containing struck-out component and 50% images containing non-struck-out components only.

The general procedure mentioned in (Nisa et al. (2019)) was followed, in principle, for generating synthetic struck-out words. It should be noted, however, that the approach in (Nisa et al. (2019)), for the purposes of that particular text recognition study, considers relatively simple struck-out types such as words with single or double horizontal and diagonal cross-out lines. On the other hand, since the aim of the proposed work is to study the broader problem of segmenting text lines in the presence of struck-out words, there is a need to consider more varieties of struck-out word types, such as spiral strike-out, zig zag strike-out, wavy strike-out as shown in Fig. 3. When compared to (Nisa et al. (2019)), the synthetically generated dataset produced in the work described in this paper is more complex in two ways. First, a greater variety of struck-out text is used. Second, a degree of uncertainty was also introduced when imposing struck-out lines on text. This includes the random placement as well as the number of struck-out lines, non-parallel and crooked lines, for instance. In (Nisa et al. (2019)), only straight cross-out lines were used in creating that dataset. Therefore, the dataset described in this paper is a more realistic reflection of actual handwritten text containing struck-out instances and it is representatively complex for evaluating the performance of the methods.

As mentioned earlier, to further objectively evaluate the proposed approach and show its independence from scripts and from the focused dataset for segmenting text lines, two additional publicly available datasets have been used namely, the ICDAR 2019 HDRC-Chinese dataset and the ICDAR 2013 Handwriting Segmentation Contest dataset. The ICDAR 2019 HDRC dataset consists of 1,172 historical Chinese document images written in the traditional Han script. The dataset is quite challenging to test on as it comprises of document images taken from multiple books which have different layouts and text representations. On the other hand, the ICDAR 2013 Handwriting Segmentation contest dataset comprises of 150 document images out of which 100 images are a mixture of handwritten English and Greek text documents, and 50 images of documents handwritten in the Indian Bangla script. In addition, the images of the ICDAR 2019 HDRC dataset suffer from document ageing and degradations. Overall, the new focused dataset and the two benchmark datasets enabled comprehensive evaluation of the robustness, script independence and generalizability of the proposed approach.

In order to evaluate the step of component detection and the step of the classification of struck-out and cleanly-written components, the standard measures of Precision, Recall and F-Score were employed as defined in Equation (4)-Equation (6), where TP: True Positive, FP: False Positive, FN: False Negative. In addition, in the case of the classification of struck-out and non-struck components, we also considered the Average Classification Rate (ACR), which is the mean of the diagonal elements of the resultant confusion matrix.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{F1} - \text{score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6}$$

To evaluate whether a bounding box was correctly predicted, the IoU (Intersection over Union) measure as defined in Equation (7). It is defined as the ratio between the common area shared by the predicted bounding box and the actual bounding box present in the ground truth to the total area of both the predicted bounding box and actual bounding box combined together. In this work, IoU > 0.5 is considered, which is a standard threshold used for struck-out component classification for counting classified and misclassified components.

$$\text{IoU} = \frac{Overlapping\ Area\ between\ the\ prediction\ and\ ground\ truth}{Union\ of\ the\ Area\ between\ the\ predicition\ and\ ground\ truth} \tag{7}$$

For this measure, as well as for the evaluation of text line segmentation, the standard measures suggested in the ICFHR2010 handwriting segmentation contest (Gatos et al. (2010)) are used. This process considers pixel-level comparison to measure how accurate the entities are after prediction as defined in Equation (8).

$$Pixel\ Level\ Comparison\ (PC) = \frac{Number\ of\ pixels\ in\ the\ Entity\ after\ prediction}{Number\ of\ pixels\ in\ the\ Entity\ present\ in\ the\ Ground\ truth} \tag{8}$$

An entity produced by the proposed approach is considered as a one-to-one match with a ground truth entity if the pixel level comparison for the former is above T_a >0.95 (this is the standard threshold value set by the ICFHR2010 handwriting contest).

The following standard metrics are used to evaluate the performance of text line segmentation:

Detection Rate: It is the ratio of the total number of one-to-one matches of entities in the image over the number of entities in the ground truth as defined in Equation (9).

$$Detection\ Rate\ (DR) = \frac{Number\ of\ One\ to\ One\ Matches}{Number\ of\ Entities\ in\ the\ Ground\ Truth} = \frac{o2o}{M} \quad (9)$$

Recognition Accuracy: It is the ratio of the total number of one-to-one matches of entities in the image over the number of entities produced by the proposed method as defined in Equation (10).

$$Recognition\ Accuracy\ (RC) = \frac{Number\ of\ One\ to\ One\ Matches}{Number\ of\ Entities\ after\ Prediction} = \frac{o2o}{N} \quad (10)$$

Performance Metric: It is the harmonic mean of Detection Rate and Recognition Accuracy as defined in Equation (11).

$$Performance\ Metric\ (PM) = 2 * \frac{Detection\ Rate * Recognition\ Accuracy}{(Detection\ Rate + \ Recognition\ Accuracy)} \quad (11)$$

where *o2o* is the number of one-to-one matches, *M* is the number of entities in the ground truth and *N* is the number of entities after prediction.

For evaluating the performance standard measures are used. For example, for component detection Recall, Precision and F-measure were used, while for evaluating the text line segmentation performance the DR, RC and PM were used. Since the ground truth of components provides bounding boxes with gray values, unlike the bounding boxes with white pixels in the case of text line segmentation, the bounding boxes resulting from the proposed method were compared with the ground truth. On the other hand, one-to-one pixel comparisons were performed between the results of the proposed method and the ground truth in order to calculate the DR, RC and PM. Overall, the measures for component detection consider the boundary of the components (global information), while the measures for text line segmentation focus on the pixels within the bounding boxes (local information). Therefore, the measures for text line segmentation are more accurate than those for component detection.

To demonstrate comparatively the effectiveness of the proposed struck-out and cleanly-written component classification method, two state-of-the art methods were implemented and evaluated alongside the proposed. The first of the two existing approaches selected was that of Chaudhuri et al. (2017) which uses aspect ratios of the words as features and an SVM classifier for classification of normal (cleanly-written) and struck-out words. The second was that of Adak et al. (2017) which employs a convolutional neural network for classification of struck-out and cleanly-written words. It is noted that the former uses handcrafted features for classification while the latter uses a CNN. The main reason to choose the above two existing methods is to demonstrate that the either the handcrafted features or the CNN alone may not be effective for classification of multiple types of struck-out components, while the combination of the two methodologies (the guiding principle of the proposed method) is effective. Another reason is that both of those existing methods have been design to classify the components first (the goal of the proposed method) before proceeding to their different goal of cleaning the struck-out handwritten documents.

Similarly, to demonstrate the comparative effectiveness of the proposed text line segmentation method, the following state-of-the-art methods were implemented and evaluated alongside it: The first was that of Li et al. (2019) which utilises characteristics of scripts for text line segmentation in Uchen Tibetan historical documents. The second method was that of Gader et al. (2020) which proposes a system to segment text lines from images of Arabic texts based on a deep learning model. The first motivation for selecting the above two existing methods for a comparative study is that both methods face similar challenges of non-regularly handwritten (as opposed to clean Latin script) text as the proposed method. An additional motivation was to demonstrate that methods which have been designed only for a particular script are not effective for text line segmentation from struck-out handwritten document in general.

4.2. Ablation Study

Within the proposed approach, the two steps of component detection and struck-out component classification are key to achieving the most successful text line segmentation results. To evaluate the effectiveness of these two key steps, the following experiments were conducted using the new focused dataset.

(i) An evaluation of DR, RC and PM is conducted for text line segmentation without the proposed component detection step, i.e., a standard connected component labeling approach is used rather than the processes presented in Section 3.1, for text line segmentation. The resulting components detected using the standard connected component labeling approach are fed to the proposed struck-out component classification and the proposed text line segmentation steps to calculate the evaluation measures. This has been done to evaluate the effect of the absence of the key step of component detection in achieving successful text line segmentation results.

(ii) The components detected by the proposed step presented in Section 3.1, without struck-out component classification, are fed as input to the proposed text line segmentation process and the DR, RC, and PM measures are then calculated. This has been done to evaluate the effect of the absence of the key step of struck-out component classification in achieving successful text line segmentation results.

(iii) The evaluation measures are calculated for the complete proposed method which includes the key steps of component detection and struck-out classification that each had been omitted in (i) and (ii) for segmenting text lines.

The results of experiments (i) and (ii) and of the proposed method with all its steps (iii) are reported in Table 2. The results in Table 2 show that experiment (i) and experiment (ii) report lower results than the proposed method for achieving segmentation results. This shows that neither the combination of (i) nor that of (ii) scores higher than the proposed method in terms of any of the measures considered. This indicates that the individual proposed steps do not have by themselves the ability to perform better than the complete proposed method. Therefore, one can infer that a combination of both the steps is essential to achieve the best text line segmentation results.

The main reason for the drop in performance in experiments (i) and (ii) is that in freestyle handwriting, large variations and stroke disconnections occur. When a conventional connected component labelling approach is used for component detection, it is very likely that a single word will be detected as split into multiple sub-components for both non-struck-out and struck-out words. This results in loss of the overall shape of handwritten words and hence affects the classification of struck-out and non-struck-out words, which in turn affects the text line segmentation in the presence of struck-out words.

Table 2. Evaluating the effectiveness of the key steps of the proposed method for text line segmentation on the focused dataset.

#	Experiments	DR	RC	PM
(i)	Proposed method without component detection (connected component labeling approach for component detection +struck-out classification +text line segmentation)	0.79	0.76	0.77
(ii)	Proposed method without struck-out classification (component detection + text line segmentation)	0.84	0.82	0.83
(iii)	Proposed method (all proposed steps)	**0.94**	**0.90**	**0.92**

4.3. Experiments for Component Detection

Qualitative results of the proposed component detection step on the focused dataset are presented in Fig. 18, where it can be seen that the proposed method fits bounding boxes correctly for both struck-out and non-struck-out components. Similarly, qualitative results of the proposed method for examples from the ICDAR2013 and ICDAR2019 HDRC datasets are shown in Fig. 19, where it can be seen that the proposed method works well with different scripts. In other words, the proposed component detection method is script-independent.

Experimental results of proposed component detection step on the focused dataset and on the two standard datasets, namely, ICDAR 2013 and ICDAR 2019 HDRC datasets are reported in Table 3. Table 3 shows that the proposed component detection step is effective and has the ability to work on different datasets. However, it can be noticed in the results that for the focused dataset, the proposed component detection method reports better results than for the other two datasets. The first reason is that the focused dataset is more uniform, containing only English documents while the ICDAR2013 dataset contains images of multiple scripts, namely English, Greek and Bangla. The second reason, moreover, is that the ICDAR2019 dataset contains Chinese historical document images, the quality of which is degraded as compared to the images of both the focused dataset and the ICDAR2013 dataset. Therefore, the

ICDAR2019 dataset is more complex in that respect compared to the other two datasets. Hence, the proposed component detection method reports lower results for it compared to the other two datasets. However, the margin of performance is not large and thus the proposed method is considered valid and useful in practice.

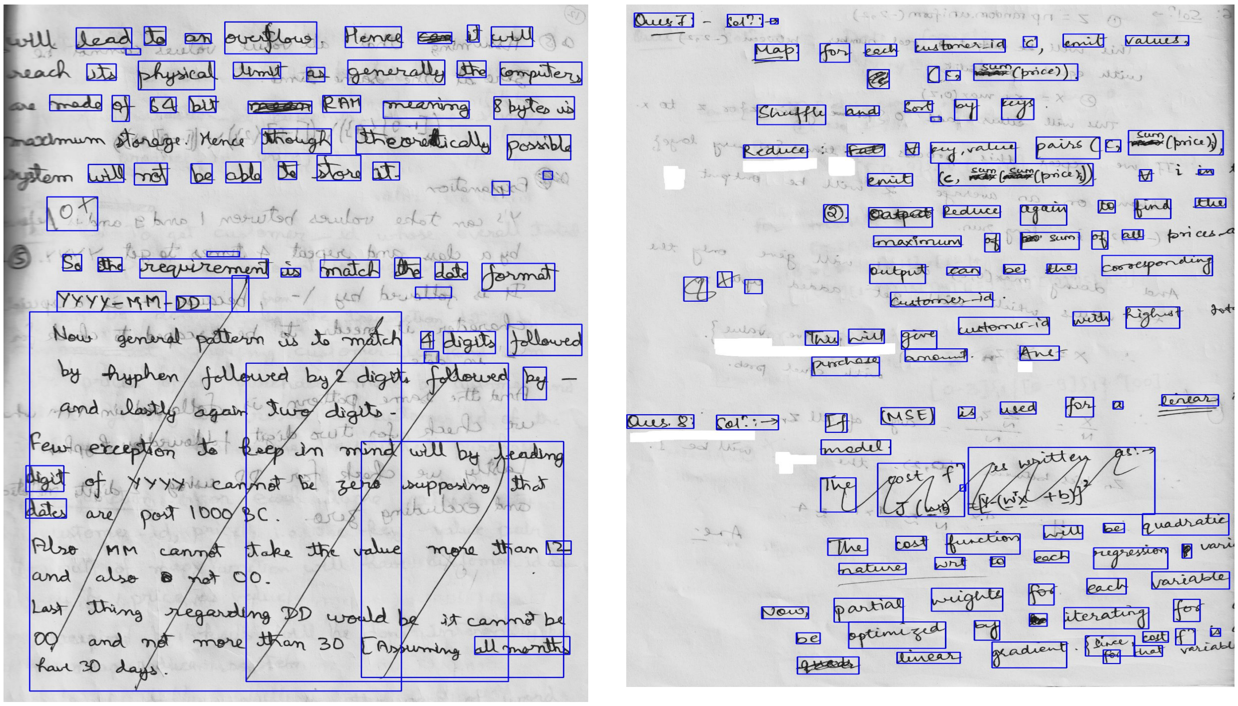


Fig. 18. Results of the proposed component detection method on examples from the focused dataset.

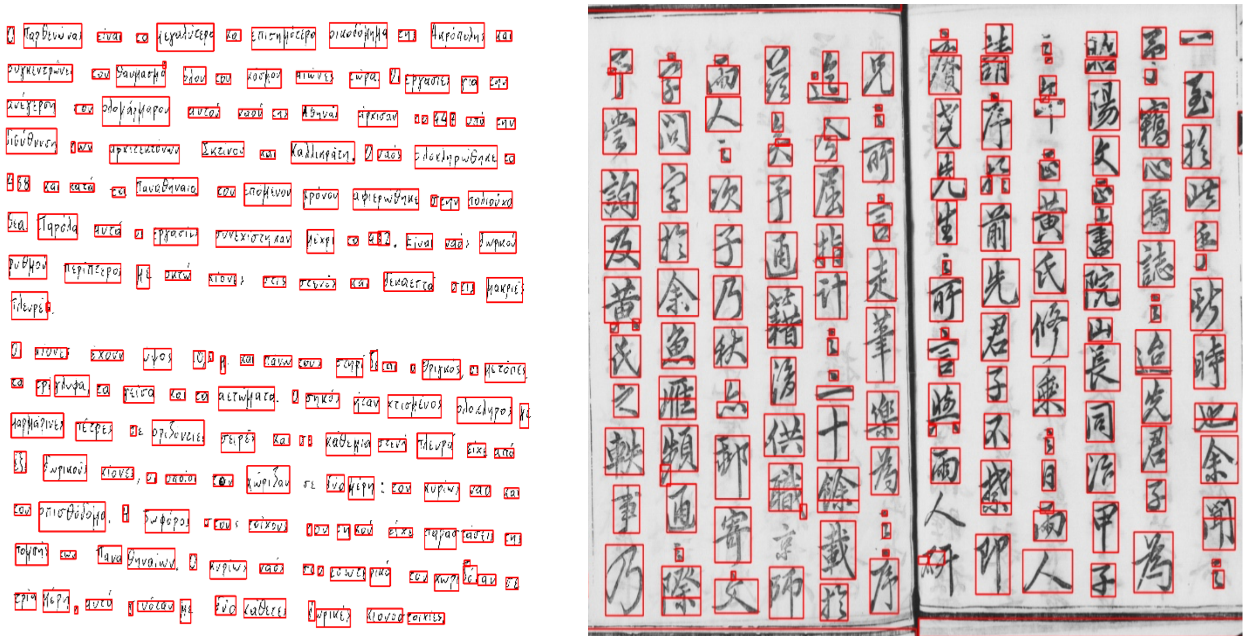


Fig. 19. The results of component detection of the proposed method on ICDAR 2013 handwriting dataset (left image) and ICDAR 2019 HDRC dataset (right).

Table 3. Evaluating the component detection method of the proposed approach on the new focused dataset and the two other standard datasets.

Datasets	Precision	Recall	F-score
Focused dataset	0.96	0.93	0.94
ICDAR2013 (Stamatopoulos et al. (2013))	0.92	0.90	0.91
ICDAR2019 HDRC (Christlein et al. (2019))	0.87	0.84	0.85

4.4. Experiments for Struck-Out and Non-Struck-Out Component Classification

For struck-out component classification, qualitative results of the proposed method on the focused dataset can be seen in Fig. 20, where one can observe that the correct bounding boxes have been fitted in red color for all the struck-out components and green bounding boxes for non-struck-out components. It can be observed that even though the type of strike-out differs between components, the proposed method works well. It can, therefore, reasonably be concluded that the proposed method is independent of the difference in the mannerism of crossing out text employed by different writers.

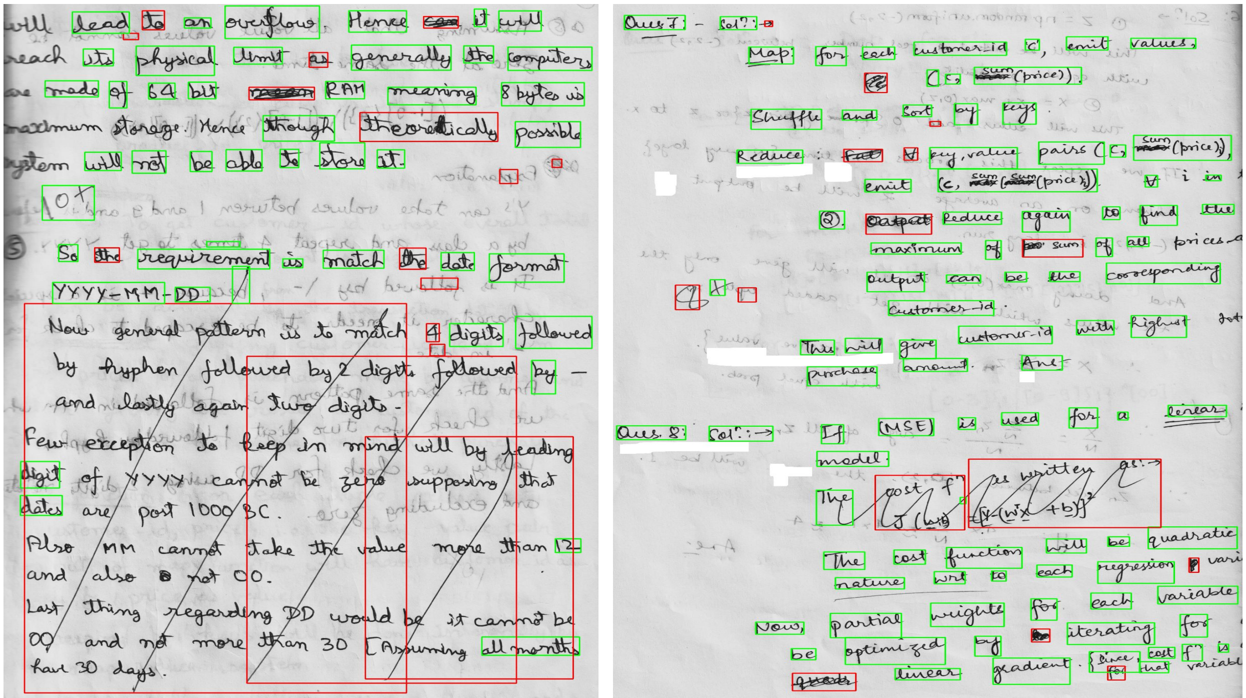


Fig. 20. Results of the proposed classification method for struck-out and non-struck-out components on the example images of Fig 18.

This can further be verified from the quantitative results of the proposed struck-out classification step on the focused dataset reported in Table 4, where the proposed method achieves the best F-score and ACR (Average Classification Rate) compared to the two state-of-the-art methods. Comparing the results of those existing methods, the Adak et al. (2017) method performs better than that of Chaudhuri et al. (2017). This indicates that the use of the CNN is more effective than the application of handcrafted features for the classification of struck-out and non-struck-out components. However, the results of Adak et al. (2017) are inferior to the proposed method in terms of F-score and ACR. The main reason for the poor results given by the two existing methods is that their scopes are limited to a particular type of struck-out component classification, particularly for Latin scripts. On the other hand, the proposed classification method is script-independent and uses a combination of hand-crafted features and deep learning, which is responsible for the difference in performance compared to the two existing methods.

Table 4. Comparative study with the existing methods for struck-out and non-struck-out handwritten word classification. ACR: Average Classification Rate in (%) on the focused dataset.

Classes	Proposed			Chaudhuri et al. (2017)			Adak et al. (2017)		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-Score
Non-Struck-Out	0.86	0.89	0.87	0.72	0.74	0.73	0.86	0.76	0.80
Struck-Out	0.88	0.85	0.87	0.75	0.72	0.73	0.79	0.88	0.83
ACR	**0.87**	**0.87**	**0.87**	0.73	0.73	0.73	0.82	0.82	0.81

4.5. Experiments for Text Line Segmentation

Qualitative results of the proposed text line segmentation method on representative examples from the focused dataset and from the two benchmark datasets are shown in Fig. 21 and Fig. 22, respectively. It can be seen in Fig. 21 that the proposed text line segmentation method works well even when there are instances of struck-out components present on the page. Similarly, Fig. 22 shows that the proposed text line segmentation works well also for documents that do not contain struck-out components. Therefore, one can reasonably argue that the proposed text line segmentation is script independent and of high practical value as it segments text lines correctly, irrespective of the presence of struck-out components in the images.

Experimental results of the proposed text line segmentation method and the two state-of-the-art methods on the focused dataset and on the two standard datasets are reported in Table 5. It can be observed that the proposed text line segmentation method performs better than the two existing ones in terms of DA, RC, and PM. A significant reason for the lower results of the existing methods is that those approaches were developed to addresses the challenges of a particular script while the proposed text line segmentation method is script independent. This is because the features considered in the proposed method are not language specific as it is evident from the results on ICDAR 2019 handwritten text of Chinese.

Overall, one can infer from the experimental results that the proposed method is capable of segmenting text lines in the handwritten document images in spite of the presence of struck-out words and differences in scripts. In addition, the proposed text line segmentation method is robust against ageing and quality degradations in document images, which can be observed from its performance on the ICDAR 2019 dataset which contains images of documents suffering from both ageing and degradations.

Table 5. Performance evaluation results of the proposed and existing methods for text line segmentation on the focused dataset and the two standard datasets.

Datasets	Proposed			Li et al. (2019)			Gader et al. (2020)		
	DA	RC	PM	DA	RC	PM	DA	RC	PM
Focused dataset	**0.94**	**0.90**	**0.92**	0.74	0.70	0.72	0.85	0.82	0.83
ICDAR2013 (Stamatopoulos et al. (2013))	**0.86**	**0.79**	**0.82**	0.77	0.73	0.75	0.81	0.73	0.75
ICDAR2019 HDRC (Christlein et al. (2019))	**0.83**	**0.78**	**0.80**	0.74	0.69	0.71	0.80	0.75	0.77

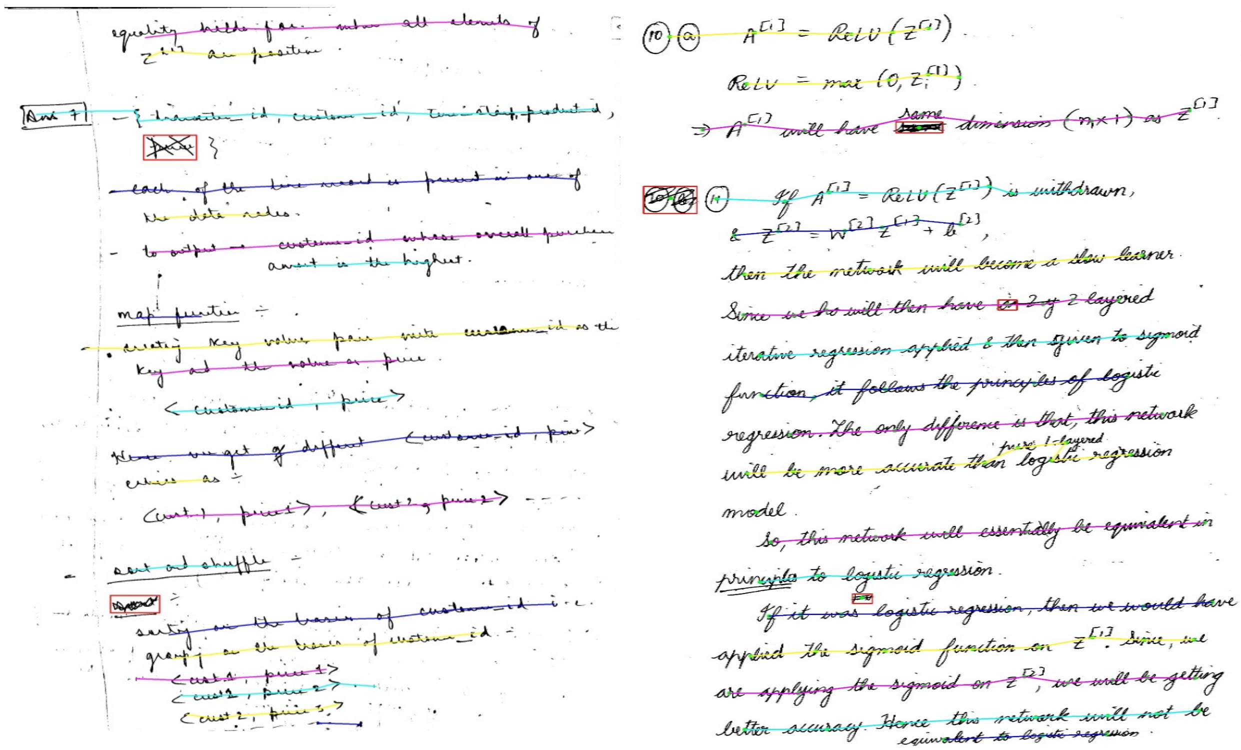
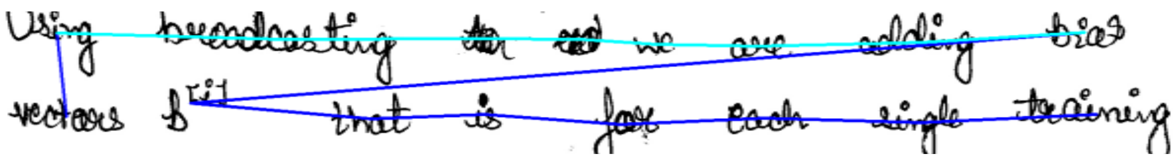


Fig. 21. Text line segmentation results of the proposed method on our dataset

Fig. 22. Text line segmentation results of the proposed method on ICDAR2013 handwriting segmentation dataset (left) and ICDAR2019 handwriting dataset (right).

(a) Misclassification of non-struck-out component as struck-out words by the proposed method



(b) Incorrect text line segmentation results by the proposed method

Fig. 23. Examples of error cases of the proposed method.

4.6. Discussion of Erroneous Cases and Limitations

In some cases, in the classification step when non-struck-out words contain certain characters with some noise as shown in the examples in Fig. 23(a), there is a significant chance of misclassification. This is due to the loss of context information. To overcome this problem, there is a need to extract context information at the pixel level and not at the character or word levels. This is beyond the scope of the work described in this paper and it can be considered as future work. Similarly, in the text line segmentation step when the spacing between words varies largely, as shown in Fig. 23(b), the features extracted based on assumptions of regularity of the distance between words and common pitch may lead to incorrect groupings of components. Therefore, there is scope for improvement of the proposed text line segmentation method in the near future. One such improvement would be to introduce natural language processing to group the words based also on semantics rather than just on the characteristics of handwritten text. The rationale is that when words from different text lines are wrongly combined, the meaning of the predicted text line changes from that of actual text line (the line most likely becomes incomprehensible). Naturally, such an approach would also depend on incorporating a robust handwriting recognition step. This is beyond the scope of the proposed method and can be considered as future work.

Furthermore, while the proposed method is applicable to a realistically quite comprehensive number of common cross-out types as already described, there could be some exceptional types of cross-out strokes which may cause the proposed method to underperform. Since the proposed method involves learning for achieving the best classification results, if a similar type of the struck-out words is not used in training it may not be identified correctly as such. Therefore, it may not be claimed that the proposed method can handle all possible types of cross-out. An exhaustive study of cross-out types and their effect on the performance of the proposed method is outside the scope of this paper.

Finally, since the proposed method involves a connected component labeling approach for fitting bounding boxes to the detected components, it requires sufficient computational power to complete such image-based operations. Such operations can be further optimized but, since the primary goal of this study is to address the challenges of text line segmentation in the presence of struck-out words, achieving maximum efficiency is left as work to be considered in the near future.

5. Concluding Remarks

In this paper a novel approach for text line segmentation from struck-out handwritten document images has been proposed based on the combination of handcrafted features and a deep learning model. To achieve this, the proposed method is divided into three steps. In the first step, components are detected from the input image by extracting and using stroke width information and a connected component identification approach. In the second step, a deep learning model has been introduced to classify the struck-out and cleanly-written components among the detected components. In the third step, text lines are segmented by exploring spatial and angular-based features of adjacent components based on the results of the struck-out component classification step.

To the best of the authors' knowledge, this is the first work proposed for segmenting text lines from handwritten document images containing instances of struck-out text. Experimental results on a newly constructed focused (on the problem characteristics) dataset and on two standard datasets show that the proposed method performs better for text component detection, struck-out component classification and text line segmentation compared to the most representative state-of-the-art methods.

There are some cases however, as discussed in the experimental section, when a word contains certain characters with noise, where the performance of the proposed component classification method degrades. Similarly, when the spacing between words in the same text line varies arbitrarily, the proposed text line segmentation method may not work well. Therefore, there is scope for future work and the authors plan to explore natural language processing for grouping words of the same text line based on semantics rather than only on extracting features which represent characteristics of the handwritten text.

Acknowledgement

The author of this work received partial support from Faculty Grant (GPF096A-2020, GPF096B-2020 and GPF096C-2020), University of Malaya, Malaysia.

References

Adak, C and Chaudhuri, B. B. (2014). An approach of strike-out text identification from handwritten documents. *In Proc. ICFHR*, 643-648.

Adak, C., Chaudhuri, B. B and Blumestein, B. (2017). Impact of struck-out text on writer identification. *In Proc. IJCNN*, 1465-1471.

Baig, A., Al-Maadeed, S. A. S., Bouridane, A and Cheriet, M. (2018). Automatic segmentation and reconstruction of historical manuscripts in gradient domain. *IET Image Processing*, 502-512.

Basavaraj, V., Shivakumara, P., Guru, D. S., Pal, U., Lu, T and Blumenstein, M. (2019). Age estimation using disconnectedness features in handwriting. *In Proc. ICDAR*, 1131-1136.

Brink, Klauw, H. V. D and Schomaker, L. (2008). Automatic removal of crossed-out handwritten text and the effect on writer verification and identification. *In Proc. SPIE.*

Chaudhuri, B. B and Adak, C. (2017). An approach for detecting and cleaning of struck-out handwritten text. *Pattern Recognition*, 82-294.

Choudhury, H., Mandal, S and Prasanna, S. R. M. (2019). Exploiting forced alignment of time-reversed data for improving HMM-based handwriting segmentation. *Expert Systems with Applications*, 121, 158-169.

Christlein, V., Nicolaou, A., Seuret, M., Stutzmann, D and Maier, A. (2019). ICDAR 2019 Competition on Image Retrieval for Historical Handwritten Documents. *In Proc. ICDAR*, 1505-1509.

Frinken, VB., Fischer, A., Manmatha, R and Bunke, H. (2012). A Novel Word Spotting Method Based on Recurrent Neural Networks. *IEEE Trans. PAMI*, 211-224.

Gader, T. B. A and Echi, A. K. (2020). Unconstrained Handwritten Arabic Text-lines Segmentation based on AR2U-Net. *In Proc. ICFHR*, 349-354.

Garz, A., Seuret, M., Fischer, A and Ingold, R. (2017). A user centered segmentation method for complex historical manuscripts based on document graphs. *IEEE Trans. HMS*, 181-193, 2017.

Gatos, B., Stamatopoulos, N and Louloudis, G. (2010). ICFHR 2010 Handwriting Segmentation Contest. *In Proc. ICFHR*, 737-742.

Kesiman, M. W. A., Buric, J. C and Ogier, J. M. (2016). A new scheme for text line and character segmentation from gray scale images of Palm leaf manuscript. *In Proc. ICFHR*, 325-330.

Kundu, S., Paul, S., Bera, S. K., Abraham, A and Sarkar, R. (2020). Text line extraction from handwritten document images using GAN. *Expert Systems with Applications*, 140.

Kundu, S., Shivakumara, P., Grouver, A., Pal, U., Lu, T and Blumenstein, B. (2019). A new forged handwriting detection method based on Fourier spectral density and variation. *In Proc. ACPR*, 136-150.

Li, Z., Wang, W., Chen, Y and Hao, Y. (2019). A novel method of text line segmentation for historical document image of the uchen Tibetan. *Journal of Visual Communication and Image Representation*, 61, 23-32.

Marti, U and Bunke, H. (2002). The IAM-database: An English Sentence Database for Off-line Handwriting Recognition. *IJDAR*, 39 – 46.

Mullick, K., Banerjee, S and Bhattacharya, U. (2015). An efficient line segmentation approach for handwritten Bangla document image. *In Proc. ICAPR.*

Nag, S., Shivakumara, P., Wu, Y., Pal, U and Lu, T. (2018). New COLD feature based handwriting analysis for ethnicity/nationality identification. *In Proc. ICFHR*, 523-527.

Nandanwar, L., Shivakumara, P., Pal, U., Lu, T., Lopresti, D., Seraogi, B and Chaudhuri, B. B. (2020). A new method for detecting altered text in document images. *In Proc. ICPRAI*, 93-108.

Navya, B. J., Swetha, G. C., Shivakumara, P., Roy, S., Guru, D. S., Pal, U and Lu, T. (2018). Multi-gradient directional features for gender identification. *In Proc. ICPR*, 3657-3662.

Nhat, Q and Lee, G. (2016). Dense prediction for text line segmentation in handwritten document images. *In Proc. ICIP*, 3264-3268.

Nisa, H., Thom, J. A., Ciesielski, V and Tennakoon, R. (2019). A deep learning approach to handwritten text recognition in the presence of struck-out text. *In Proc. IVCNZ.*

Qi, Y., Huang, W. R., Li, W and DeGange, J. L. (2020). DeepErase: Weakly supervised ink artifact removal in document text images. *In Proc. WACV*, 3511-3519.

Raghunandan, K. S., Shivakumara, P., Navya, B. J., Pooja, G., Prakash, N., Kumar, G. H., Pal, U and Lu, T. (2016). Fourier coefficients for fraud handwritten document identification through age analysis. *In Proc. ICFHR*, 25-30.

Renton, G., Soullard, Y., Chatelain, C., Adam. S., Kermorvant, C and Paquet, T. (2018). Fully convolutional network with dilated convolutions for handwritten text line segmentation. *IJDAR*, 177-186.

Romero, R., Sanchez, J. A., Bosch, V., Depuydt, K and Does, J. D. (2015). Influence of text line segmentation in handwritten text recognition. *In Proc. ICDAR*, 536-540.

Ryu, J., Koo, H and Cho, N. I. (2015). Word segmentation method for handwritten documents based on structural learning. *IEEE. SPL*, 161-1165.

Shivakumara, P., Pal, U., Lu, T., Chakarborti, T and Blumenstein, M. (2019). A New Roadmap for Evaluating Descriptive Handwritten Answer Script. *In Proc. ICPRAI*, 83-96.

Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U and Alaei, A. (2013). ICDAR 2013 Handwriting Segmentation Contest. *In Proc. ICDAR*, 1402-1406.

Vo, Q. N., Kim, S. H., Yang, H. J and Lee, G. S. (2018). Text line segmentation using a fully convolutional network in handwritten document images. *IET Image Processing*, 438-446.

Zhu, Y and Zhang, K. (2017). Text segmentation using superpixel clustering. *IET Image Processing*, 455-464.