International Journal on Document Analysis and Recognition (IJDAR) (2021) 24:299–313
https://doi.org/10.1007/s10032-021-00370-8

ORIGINAL PAPER

# Segmentation of text lines using multi-scale CNN from warped printed and handwritten document images

Arpita Dutta[1] · Arpan Garai[2] · Samit Biswas[1] · Amit Kumar Das[1]

Received: 27 November 2019 / Revised: 14 April 2021 / Accepted: 28 April 2021 / Published online: 21 May 2021

**Abstract**
Paper documents are ideal sources of useful information and have a profound impact on every aspect of human lives. These documents may be printed or handwritten and contain information as combinations of texts, figures, tables, charts, etc. This paper proposes a method to segment text lines from both flatbed scanned/camera-captured heavily warped printed and handwritten documents. This work uses the concept of semantic segmentation with the help of a multi-scale convolutional neural network. The results of line segmentation using the proposed method outperform a number of similar proposals already reported in the literature. The performance and efficacy of the proposed method have been corroborated by the test result on a variety of publicly available datasets, including ICDAR, Alireza, IUPR, cBAD, Tobacco-800, IAM, and our dataset.

**Keywords** Text-line segmentation · CNN · Multi-scale context · Semantic segmentation

## 1 Introduction

Text-line segmentation is one of the essential prerequisites for document image analysis tasks such as alignment of texts [16], spotting of words [43] and OCR [20]. The digital camera captured document images may suffer from different types of warping; this is due to the camera angles and/or shape of the surface of the document. For flatbed scanned document images, the distortions are less. The non-uniform and/or cylindrical surfaces are often used to paste the documents (posters, advertisements, and notices). These documents are captured in different camera angles, and it causes skew and warping. Due to the well-structured form of printed documents, warping makes line segmentation more complex. On the other hand, handwritten documents, even without warping and other distortions, impose a more difficult challenge because font, skew, presence of touching and overlapping lines, writing styles, and decoration vary widely even in the same document. Figure 1 shows a few examples of camera captured/flatbed scanned printed/handwritten document images with a varying degree of warping. The images may be skewed as well as warped. In these images, the text portion appears to be narrower as the distance of the text from the camera gets increased. The text lines lie in an interleaved manner with a varying inter-line distance. In a nutshell, the presence of such types of warping in document images causes a huge variation.

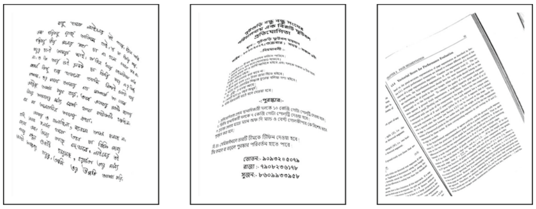

**Fig. 1** Various forms of Handwritten/Printed *Bengali/English* document images

Line segmentation is one of the preliminary steps for any further processing of document images. For example, the success of the OCR primarily depends on the de-warped lines fed

✉ Arpita Dutta
arpita_dutta.rs2018@cs.iiests.ac.in

Arpan Garai
ag.rs2016@cs.iiests.ac.in

Samit Biswas
samit@cs.iiests.ac.in

Amit Kumar Das
amit@cs.iiests.ac.in

[1] Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, India

[2] Department of Computer Science and Engineering, University of Engineering and Management, Kolkata, India

to it. Several methods for dewarping of document images are available in the literature [7,15,21,22,29]. These dewarping methods are based on the individual text-line information like text-line flow, the curvature of the text lines, etc., of those documents. Therefore, text-line segmentation is one of the essential requirements for the dewarping of document images.

The approaches for text-line segmentation reported in recent literature are classified into two categories [42]: ad hoc methods and learning-based methods. The ad hoc methods depend on the projection profile, smearing techniques, morphological techniques, etc. In contrast, learning-based methods rely on automatic feature learning based on the concept of a neural network. Though deep learning-based methods have been used in several computer vision applications [25,51], application of these methods in text-line segmentation has not been fully explored.

**Contribution**

Our approach depends on the learning-based technique. The main contribution of this paper is to propose a deep neural network that can efficiently locate each line from both printed and handwritten document images along with a wide variety of complex structures. In our proposed method, the text lines are represented through the set of pixels in textual components. We considered the text-line segmentation problem as a semantic segmentation problem, i.e., pixel-wise labeling problem. Inspired by various kinds of deep learning approaches [26], we have explored those methods to develop our algorithm and decided that the multi-scale encoder-decoder-based convolutional neural network (CNN) would be the best choice to tackle our problem. Another contribution is the development of a semi-automatic approach for the generation of the ground truth of text lines to train the proposed CNN model. In developing the ground truth, our semi-automatic approach saves a lot of time in terms of time and cost. However, one of the main constraints of using CNN in document image processing is its higher resolution. The convolution operations on images with high resolution require an extensive amount of down-sampling of images, which causes the loss of many discriminating details within the images. To resolve this issue, we have used the idea of patch-based learning [11,59]. The designed network has been trained using scanned and single folded warped images from Warped Document Image Dataset (WDID) [15] (See Fig. 1), scanned images from BESUS dataset [6,46], IAM dataset [33], and Alireza dataset [3]. To test the robustness of our model, we have tested it on different kinds of publicly available datasets having a wide variety of document images, and it achieves significant accuracy. The policy adopted for the development of the training dataset improves the performance of the learning model (See Sect. 3.2). Also, the structuring of the proposed network plays a significant role to segment text line efficiently from distorted document images such as multiple folded warped document images of printed Bangla scripts, warped and skewed handwritten Bangla scripts, and skewed English document images based on the concept of semantic segmentation (see Sect. 3). Interesting results have also been obtained on various kinds of publicly available benchmark datasets such as IUPR dataset [9], Tobacco-800 dataset [55], ICDAR 2013 Handwritten Segmentation Contest [53] dataset, and cBAD datasets [10]. However, these images have not been used during training. Considering and comparing the results with other approaches signify the efficient application of semi-supervised learning with better performance than the contemporary approaches reported in the literature. The rest of the paper is organized as follows: Sect. 2 presents the motivation and related work. The entire architecture, along with its implementation details, is described in Sect. 3; experimental results and evaluation metrics are presented in Sect. 4. Finally, concluding remarks are given in Sect. 5.

## 2 Related work

Several methods for line segmentation were proposed in [5,24,28,40,41,48], etc. Table 1 gives a brief description of previous state-of-the-art techniques. An in-depth analysis of existing text-line segmentation methods was described in [14]. Most of the text-line detection schemes relied on analysis of projection profile [4,52,57] or profile generated through Hough transform [32]. He and Downton [19] and Shi et al. [49] proposed methods based on *RXY cuts* [39] and smearing, respectively. Shi and Govindaraju [47] used the patterns of text lines based on the concept of a fuzzy run length matrix. Liwicki et. al. [31] proposed an algorithm based on the dynamic programming approach. Shivakumar et. al. [50] utilized the skeletons of the connected components of the text region for line segmentation. Bhukari et. al. [8] proposed a method that used a set of line filters. Another novel approach was proposed by Alaei et. al. [2] for handwritten document images. The algorithm, proposed by Sabbni et. al. [45], was based on the energy map from the input text of historical document images. Yin et. al. [58] proposed an algorithm with the amalgamation of minimum spanning tree and distance metric learning. Gatos et. al. [18] proposed a novel approach for segmentation of text lines from historical document images based on vertical lines and connected components. Recently, researchers have focused on learning-based techniques with the amalgamation of different deep learning techniques for the segmentation of text lines. Moysset et. al. [34–36,38] used deep learning-based method for text-line segmentation. In [36], a LSTM recurrent neural network was proposed to segment paragraphs, whereas in [38], the authors proposed an architecture which was made of both CNN and LSTM network. Vo et al. [56] proposed an approach on text-line segmentation of different types of handwritten document images based on fully convolutional neural network (FCNN). Renton et al. [42] presented a method for handwritten text-line segmentation based on FCNN with dilated convolutions. Though there are plenty of existing approaches for text line, it is still the hotbed for new thinking and innovation from warped and perspective distorted images.

## 3 Proposed method

Most of the previous state-of-the-art algorithms for text-line segmentation are based on bounding box prediction. In the case of complex structured document images with distortions in the form of skew, warping, multiple folds, etc., the prediction of boundaries across text lines is a challenging problem. Moreover, these methods cannot achieve significant accuracy on a complex layout like handwritten and/or warped document images due to touching characters. Actually, in the case of complex structured images as well as handwritten images, the representation of text lines through polygons cannot separate those lines from each other perfectly. Therefore, we decided to represent the text lines through a set of pixels in textual components. In reality, there exist plenty of connected textual components within a single text line. As our goal is to locate the position of each text line, we manually labeled each text line so that each of them is represented through a single connected component by eliminating the non-connectives among words. As a consequence, in our current approach, each text line is expressed through a set of pixels of that single connected component. In contrast, the pixels, which reside within the inner gap between two consecutive text lines, are represented as background pixels.

In our method, the problem of text-line segmentation is considered as a semantic segmentation problem, i.e., pixel-wise classification task. The main objective of semantic segmentation is to assign each pixel to its appropriate label. In our method, we have two class labels: text line and background. Therefore, after partitioning all the pixels of an image into their respective parts (i.e., text line and background), our aim is to detect whether a pixel belongs to text line or background. We propose an encoder–decoder-based CNN architecture where there exist no densely connected layers in the proposed model. The encoder part, which consists of multiple hierarchical convolutional and max-pooling layers, is used for the extraction of features that are required for describing semantics. In a nutshell, this encoder part plays a crucial role in representing an image into a more abstract level guided by semantics. The decoder part focuses on upsampling and is trained to perform pixel-wise classification. In order to get a dense pixel-wise prediction, the main objective of the decoder part is to project discriminative features obtained from the encoder onto the pixel space of the input image. The upsampling branches of the decoder part help the model to achieve better performance by combining the intermediate representation obtained from the encoder along with the decoding activation. The final output layer produces the exact dimension of input size and computes a dense pixel-wise probability map that belongs to one of the predefined class labels, i.e., text line or background.

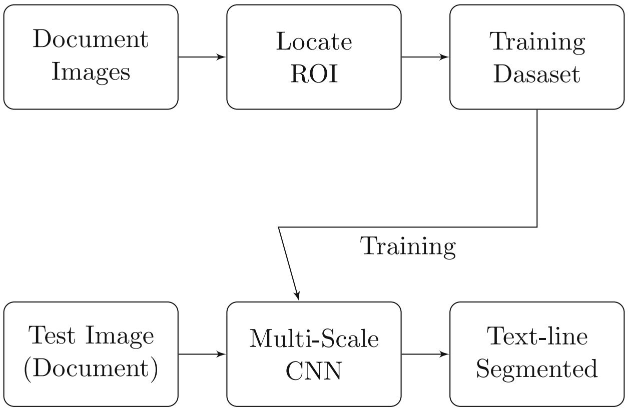

**Fig. 2** Overview of the Proposed Method

Another essential task is the development of the training dataset (See 3.2 for details) to train the proposed CNN architecture. Those images are annotated at pixel level and are used to train the proposed model for performing segmentation on two semantic labels, i.e., text line and background. During training image annotation at pixel level, the images are manually labeled for locating text pixels as well as background pixels. However, in the presence of touching/overlapping characters in handwritten documents, only manual labeling cannot perfectly detect every text line, resulting in faulty ground truth annotations of text lines. To overcome this difficulty, some prepossessing steps are applied before performing manual labeling in case of ground truth annotations for handwritten document images. Also, due to the availability of very limited resources of complex structure document images, we have synthesized warped document images from scanned documents to train the proposed model correctly.

Document images are mostly generated by flatbed scanners or captured using digital cameras. A shadow may arise at the end of the margin of these document images. The reasons behind this marginal noise are poor illumination of cameras while capturing images or placing documents in a slanted way within scanners. Moreover, the historical handwritten documents cause a huge amount of marginal noise after binarization. In the proposed approach, we have taken a color image as input and then binarized and de-noised those images using [13]. Next, those images are fed into the designed network. Figure 2 shows the overview of the proposed approach.

**Table 1** Summary of the related state-of-the-art methods

| Author | Dataset Language | Features Used | Document type |
|---|---|---|---|
| He et. al [19] | Pyraloidea dataset | Projection Profile | Handwritten |
| Karatzas [4] | Japanese | Projection Profile | Printed |
| Shi et. al. [47] | English | Fuzzy Run Length | Handwritten |
| Alaei et al. [2] | English, Oriya and Bangla | Morphological Operation | Handwritten |
| Sabbni et al. [45] | English, Arabic and Spanish | seams and energy map | Historical |
| Gatos et al. [18] | English | vertical lines, connected components analysis | Historical handwritten |
| Moysset et al. [36] | Maurdor Dataset (Arabic, English, French) | LSTM Recurrent Neural Network | Handwritten; Printed |
| Moysset et al. [38] | Maurdor Dataset (Arabic, English, French) | CNN and LSTM network | Handwritten, Printed |
| Stewart et al. [54] | English | CNN | Handwritten; Printed |
| Vo et al. [56] | English, Greek and Bengali | FCNN and LAG | Handwritten |

**Table 2** Layers of convolutional neural network architecture. Here $N$-convolution denotes $N$ no. of successive convolution operations followed by ReLU

| Layer No | Layer Operation | Input (Out of $L$#) | Filter size/stride | No.of filters | Output Dimension |
|---|---|---|---|---|---|
| $L_0$ | 1- Convolution + ReLU | Input Image | $5 \times 5 / 1$ | 8 | $256 \times 256 \times 8$ |
| $L_1$ | Max- Pooling | $L_0$ | $2 \times 2/2$ | 8 | $128 \times 128 \times 8$ |
| $L_2$ | 1- Convolution+ReLU | $L_1$ | $5 \times 5/1$ | 16 | $128 \times 128 \times 16$ |
| $L_3$ | Max-pooling | $L_2$ | $2 \times 2/2$ | 16 | $64 \times 64 \times 16$ |
| $L_4$ | 4- Convolution+ReLU | $L_3$ | $5 \times 5/1$ | 32 | $64 \times 64 \times 32$ |
| $L_5$ | 5- Convolution+ReLU | $L_2$ | $5 \times 5/1$ | 16 | $128 \times 128 \times 16$ |
| $L_6$ | 6- Convolution+ReLU | $L_0$ | $5 \times 5/1$ | 8 | $256 \times 256 \times 8$ |
| $L_7$ | Upsampling | $L_4$ | $5 \times 5/1$ | 32 | $128 \times 128 \times 32$ |
| $L_8$ | Upsampling | $L_5$ | $5 \times 5/1$ | 16 | $256 \times 256 \times 16$ |
| $L_9$ | Concatenation | $L_6, L_8$ | – | – | $256 \times 256 \times 24$ |
| $L_{10}$ | Concatenation | $L_5, L_7$ | – | – | $128 \times 128 \times 48$ |
| $L_{11}$ | 1-Convolution + ReLU | $L_{10}$ | $5 \times 5/1$ | 16 | $128 \times 128 \times 16$ |
| $L_{12}$ | 1-Convolution + ReLU | $L_9$ | $5 \times 5/1$ | 8 | $256 \times 256 \times 8$ |
| $L_{13}$ | Upsampling | $L_{11}$ | $5 \times 5/1$ | 16 | $256 \times 256 \times 16$ |
| $L_{14}$ | Concatenation | $L_{12}, L_{13}$ | – | – | $256 \times 256 \times 24$ |
| $L_{15}$ | 1- Convolution + ReLU | $L_{14}$ | $5 \times 5/1$ | 2 | $256 \times 256 \times 2$ |
| $L_{16}$ | Sigmoid | $L_{15}$ | – | – | $256 \times 256 \times 1$ |

### 3.1 CNN architecture

Convolutional neural networks (CNNs) are a class of deep, feed-forward artificial neural networks successfully applied to analyze visual imagery. The core idea behind using CNN is to obtain more dense feature mapping from an image. This performs well in the image classification problem because the obtained feature maps are converted into a vector that is used for classification. However, in the image segmentation problem, the challenging part is to reconstruct the image from the vector, which is obtained from the feature maps learned from CNN.

**(A) Intuition behind the proposed architecture** This work proposes a multi-scale encoder-decoder-based CNN model to segment text lines from scanned or camera-captured document images. The encoder part of the architecture encodes the input image into feature representations at multiple resolutions. More specifically, the higher encoding level signifies less spatial resolution along with more dense semantics and fine-grained details of an image. That is why this part of the architecture is also called as contraction part of the network. However, in the case of semantic segmentation, the objective of the proposed model is not only to learn the discrimination at pixel level but also to project those features obtained at various stages of the encoder onto pixel space. The decoder part, which consists of upsampling and concatenation operations, is dedicated to achieve this goal, i.e., conversion of the feature vectors learned by the encoder to

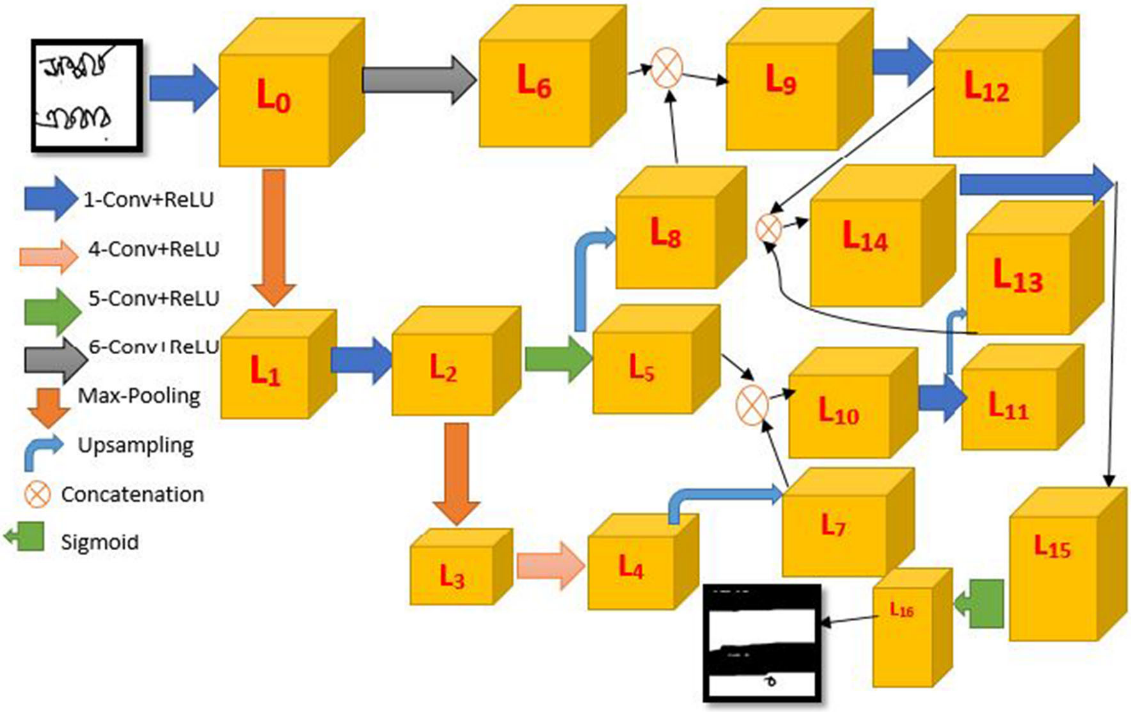

**Fig. 3** Multi-scale convolutional neural network architecture

segmented images. The main intuition behind the upsampling operation is to restore the dense feature map to the original resolution of the input image. In our method, upsampling is performed through transposed convolution [12]. However, the upsampling operation is sparse in nature, requiring good prior information from the early stage to obtain better localization. Therefore, after each upsampling step, the upsamples of internal representations are concatenated with the feature maps obtained from the corresponding encoding layer. This operation ensures that the same feature maps, which are learned while encoding the images, are used to reconstruct the segmented image from feature vectors. Also, the structural integrity of the input image is preserved. To maintain the symmetry in the proposed network, one convolution operation is performed on the augmented representations with the number of filters equal to the number of filters of the corresponding encoding layer. The last step of the decoding part is to apply the sigmoid activation function on the last convolutional layer of the network, which is 256 × 256 representations of the floating values in 0,1). This action finally predicts the probability of every pixel belonging to a text line. Therefore, the decoder part results in a pixel-based probability map about the location of text lines. These obtained prediction maps are compared with our annotated images of text lines.

**(B) Building block of the architecture** The input to the CNN model architecture is a patch $P$ of size $a \times b \times c$ where $a$, $b$ and $c$ denote height, width and number of color channels (for RGB image, $c = 3$), respectively. Let $p_{ij}$ be the pixel of patch $P$ at position $(i, j)$, where $i \in (1, 2, \ldots, a)$ and $j \in (1, 2, \ldots, b)$. The CNN is trained using these patches, and the deep features are extracted. Each of these pixels is labeled as either text line $L_T$ or background $L_B$. The proposed CNN identifies the text lines using the class probability $(C_{ij})$ of each pixel located at $(i, j)$ where $C_{ij} \in [0, 1]$. The mathematical formulation is - $p \in R^{c\times a\times b} \implies C \in R^{a\times b}$. The CNN consists of a series of convolution layers having element-wise nonlinearities. Each element in the $k^{th}$ layer of convolution operation is followed by a nonlinear activation function RELU. It is defined as $p_k = RELU(w_k \times p_{k-1} + b_k)$. Here, $1 \leq k \leq L$ and $L$ denote the total number of layers, $RELU(x) = max(0, x)$ is a nonlinear activation function, $p_k \in R^{c_k\times a\times b}$ is the output of layer $k$ and $b_k \in R^{c_k}$ denotes the bias term for each filter. $w_k \in R^{c_k\times f_k\times f_k\times c_{k-1}}$ is the weight of 'learnable' filters where each $w_k$ is obtained by applying $c_k$ distinct convolutions, $f_k$ is the size of filters at$k^{th}$ layer, $c_{k-1}$ signifies the number of filters appeared in previous layer. Therefore, after each convolution operation on patch size $P$ padded with $p$ number of pixels, the output dimension is denoted by - $X = \frac{P-f+2p}{S} + 1$, where $f$ signifies the filter size and $S$ signifies the stride of pixels, i.e., the sliding of kernels. The *pooling* operation is performed to reduce the dimension of the output of the previous layer. It also introduces invariability to the small transformations of images. After applying max-pooling operation on the output

$X$ of the convolutional neural network with filter size $E$ and stride $S$, the output dimension will be - $Y = \frac{X-E}{S} + 1$.

**(C) Architecture Details** The details about the layers of CNN Architecture are shown in Table 2. The input fed to the network is a patch of size 256 × 256. During each convolutional operation, padding is done so that the pixels on the border get an opportunity to participate in the computation of the feature map by interacting with the filters. In our experiment, we have set stride value as 1, filter size as 5 × 5, and padding as 2 at each convolution layer to preserve the input dimension after each convolution operation. The features are also encoded at a different scale to learn a more condensed feature map. In the encoder part of our architecture, the features are computed at scales $\frac{1}{2}$ and $\frac{1}{4}$ along with the original input. Therefore, several convolution operations are applied on each of these scaled layers, i.e., ($L_1$ and $L_3$) for encoding features from those layers. Moreover, instead of directly down-scaling the input, one convolution operation is performed on the input of each scale for extraction of features at that corresponding scale before performing downsampling.

Consequently, one convolution operation is first performed on the input image for extraction of lower-level features such as edge, contours, etc., and $L_0$ is obtained. Then, $L_0$ is downsampled by a max-pooling operation to $\frac{1}{2}$ the input image size, and $L_1$ is obtained. During each max-pooling operation, the filter size is set to 2 × 2 along with a stride value of 2 to obtain non-overlapping patches. Next, $L_2$ is obtained by one convolution operation on $L_1$ to learn features at scale $\frac{1}{2}$. $L_2$ is downsampled by a max-pooling operation to $\frac{1}{4}$ the input image size, and $L_3$ is obtained. Now, 4 successive convolution operations are performed on $L_3$ to encode the feature maps at scale $\frac{1}{4}$. Actually, the number of convolution operations increases with the increment of the scale factor due to the expansion of the pixel space. Therefore, at scale $\frac{1}{2}$ and 1, 5 and 6 number of convolution operations are performed on layer $L_2$ and $L_0$ , respectively. However, the number of convolutional feature maps doubles with the downscaling of the input image by a factor of $\frac{1}{2}$ to encode more complex and fine-grained features effectively. In the decoder part, upsampling and concatenation operations are performed to project the discriminative features learned by the encoder onto the pixel space of the input image. At scales $\frac{1}{2}$ and $\frac{1}{4}$, after encoding features by performing convolution operation, upsampling is performed on those scaled layers by using transposed convolution operation to get back the original input size. Therefore, $L_4$ is first upsampled to $\frac{1}{2}$ the input image size and $L_7$ is obtained. However, at each scale, the previous upsampled layer features are concatenated with the encoded features of that corresponding scale to reconstruct the segmented image. Therefore, at scale $\frac{1}{2}$ , layer $L_5$ and $L_7$ are augmented and $L_{10}$ is obtained. To main symmetry, after each concatenation operation, one convolution operation is performed on the augmented features with the same number of filters of that corresponding scale. Therefore, one convolution operation is performed on $L_{10}$ with 16 number of filters as the number of filters at scale $\frac{1}{2}$ is 16. However, at scale 1, the concatenated layer, i.e., $L_{14}$, is passed through a convolution layer with two feature maps as the number of segments is two (text line and background) and $L_{15}$ is obtained. Finally, a sigmoid activation function is applied on $L_{15}$ in order to predict the probability of a pixel belonging to the text line. The details of our proposed architecture are shown in Fig. 3.

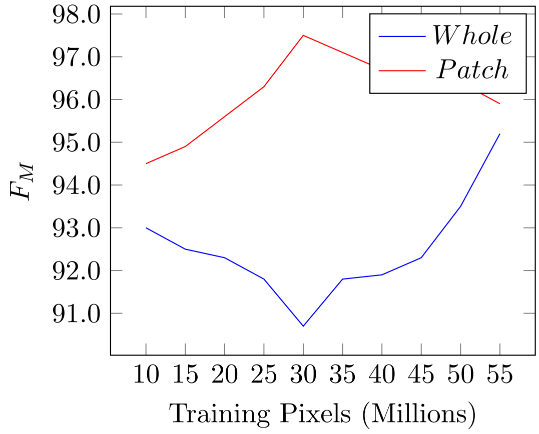

**Fig. 4** Learning curve during training data

## 3.2 Development of training dataset

One of the main constraints of using a CNN for the analysis of document images is the preparation of the training dataset. Training with more data plays a significant role in achieving the better performance of the learning model. However, eventually adding more data may lead to the problem of overfitting, which causes the degradation of the performance of the learning model in achieving good performance for unseen data. The greater diversity in the entire training dataset plays a significant role in improving the performance of the learning model. Actually, most of the pixels of document images are locally similar due to the homogeneity in handwriting style and noise content. Therefore, in this paper, we have focused on creating more training data with great diversity. Instead of training with fully annotated large images, the patches of size 256 × 256 are cropped from the original images into overlapping manner, excluding the area near the boundary. A window size of 256 × 256 is moved from left to right and top to bottom to generate image patches by cropping the portion of the image under this window (see Fig. 5). Figure 4 depicts the amount of data needed to train our proposed model by plotting the learning curve.

It is evident that instead of using whole images, training with image patches improves the performance of the

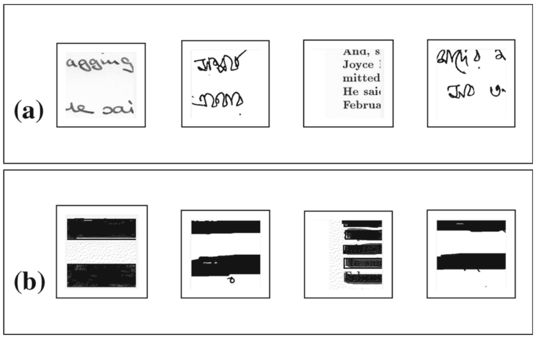

**Fig. 5** Sample images from Real Document Image Dataset: **a** Input images; **b** corresponding ground truth images

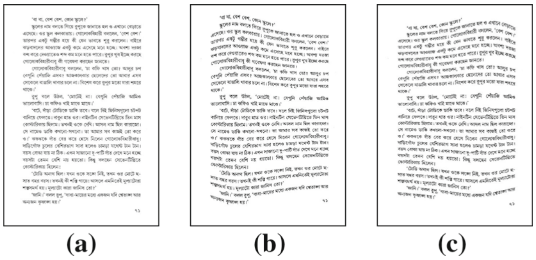

**Fig. 6** **a** Preprocessed scanned image; corresponding generated image. **b** Warping at the left side; **c** warping at the right side

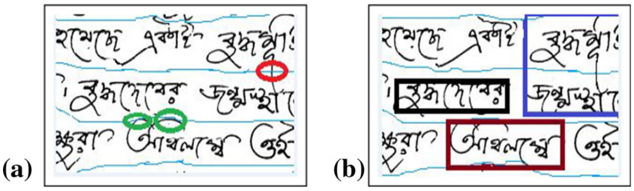

**Fig. 7** **a** Touching, overlapping components and splitting lines are marked with red, green circles and blue lines, respectively; **b** the height of the black rectangle signifies the height of the minimum bounding box of the components that do not have intersection with splitting line. The height of the blue rectangle denotes $T_c$ and the height of the brown rectangle signifies $O_c$

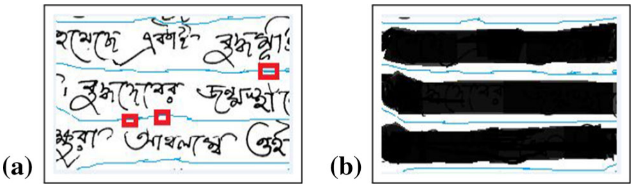

**Fig. 8** **a** The width of $\frac{H_C}{7}$ pixels from both upper side and lower side of the splitting line around intersecting points are marked with red rectangular boxes; **b** generated ground truth images

proposed model with the reduced number of training pixels. Also, these image patches help to create a huge number of training data with lots of diversity. However, some preprocessing steps are applied in handwritten document images due to deal with the presence of touching/overlapping components. Moreover, due to the lack of an available dataset of warped document images to train a deep neural network, we develop an algorithm to synthesize warped document images from scanned document images. After dealing with those constraints, image patches of size 256 × 256 are generated from those images in the same way as mentioned earlier.

**(a) Camera captured/Scanned Document Image** No preprocessing steps are needed for real camera captured/ scanned document images. After annotating those images, image patches of size 256 × 256 are generating by sliding the window of size 256 × 256 from left to right and top to bottom (Fig. 5).

**(b) Synthetically Warped Document Image** The dataset containing warped document images is not enough to train most of the deep neural networks. So, we have synthesized warped document images from any scanned document image. These synthesized warped images look like any document having a curved surface. The process is based on specifying warping factors at each pixel of the flatbed scanned document image. These warping factors are generated using 2 warping position parameters (WPP) and 8 warping control parameters (WCP) [17]. In our experiment, we have chosen the WPP ($P_1$ and $P_2$) such that the peak of the book surface is either at the left side (Fig. 6b) or right side (Fig. 6c) but not diagonally oriented or not along the middle of the document. Here, the parameters $P_1$ and $P_2$ are used to estimate the position of knot points at the top most row and bottom-most row in the image, respectively. For left side $m_l^t$ and $m_l^b$ numbers of possible values are taken for $P_1$ and $P_2$, respectively. Similarly, for right side $m_r^t$ and $m_r^b$ numbers of possible values are taken for $P_1$ and $P_2$, respectively. As a result, a total of $m_l^t \times m_l^t$ and $m_r^t \times m_r^t$ variations can be generated for different values of $P_1$ and $P_2$. Let $n$ numbers of possible values are used for each WCP. So, from a single scanned image a total number of $n^8 \times (m_l^t \times m_l^t + m_r^t \times m_r^t)$ images can be produced. In our experiment, we have taken $n = 5, m_l^t = m_l^t = m_l^t = m_l^t = 3$. Some examples are shown in Fig. 6. The value of WCP ($P_{3,\ldots,10}$) is set as a fraction of diagonal of the image, whereas for $P_{1,2}$ the value ranges from 0.1 to 0.3 for warping at the left side and from 0.7 to 0.9 for warping at the right side [17].

**(c) Dealing with the touching components** One of the most challenging parts of any text-line segmentation algorithm is to deal with handwritten document images having touching/overlapping characters. In our proposed method, the ground truth for those images is generated in a semi-automated way. The touching or overlapping of components in neighboring text lines of handwriting document images occurs due to the ascenders or descenders of text components (see Fig. 7a). When two neighboring text lines touch with each other, then at least two different components from those two lines meet each other and generate a single component. Therefore, both touching and overlapping components must intersect with the splitting line. Next, the components, which do not have an intersection with the splitting line, are detected. Then, heights of the minimum bounding boxes from each detected component are calculated (i.e., the height of the black rectangular box in Fig. 7b). However, the distributions of those calculated heights contain both smaller and larger values too. In our experiment, the heights of the bounding boxes, which are less than 100, are considered as noise. In contrast, the heights greater than 550 are considered as non-textual components, i.e., graphics or logos, etc. Therefore, the heights of the bounding boxes, which have values greater than 100 and less than 550, are considered as a valid set of distributions. Let $H_c$ denote the mean of all these heights from that valid distribution set. The height of a touching component is denoted by $T_c$ (i.e., the height of the blue rectangular box in Fig. 7b), and then, the relation between them can be defined as $T_c \geq 2 \times H_c$. On the other hand, if $O_c$ (i.e. the height of the brown rectangular box in Fig. 7b) denotes the height of an overlapping component, then it satisfies the inequality $H_c < O_c < 2 \times H_c$. Based on these equations, the touching and overlapping components within the document are traced, respectively. After searching those components, the points where splitting lines intersect with those touching/overlapping components are detected. These intersecting points are used to form the ground truth images for training.

In the presence of touching/overlapping components, components obtained by the manual labeling process from two or more consecutive text lines are merged together and generate a larger single component. Now, if we use those annotated images directly for training our network, the network will not be able to assign each pixel to its correct label. Therefore, these components in training images need to be correctly labeled as either within the line (text line) or between-line (background). This issue is fixed using those intersection points, i.e., the point where the touched/overlapped components meet with the split-lines. The width of $\frac{H_C}{7}$ pixels from both the upper side and lower side of the splitting line is assigned as the label of background pixels (i.e., white pixels) (see Fig. 8a). This value is obtained based on the experiment conducted on 1500 images from various handwritten datasets. We also generate patches of size 256 × 256 after annotating those images at pixel level and use those images for the training of the proposed network (See Fig. 8b).

### 3.3 Loss function

In the proposed architecture, the binary cross-entropy is considered as loss function. Actually, the more the loss value increases, the more the predicted class probability of every pixel diverges from the actual label. As there are two labels, i.e., text line $(L_T)$ and background $(L_B)$ , each pixel is assigned to either label $L_T$ or $L_B$ depending on its corresponding probability. Therefore, binary cross-entropy loss $(B_{cross})$ is defined as $B_{cross} = -[t_T \times \log(e_T) + t_B \times \log(e_B)]$. Here, for $L_T$, $t_T \in \{0, 1\}$ and $e_T \in \{0, 1\}$ signify the ground truth and the predicted class probability, respectively. Similarly, for $L_B$, $t_B = (1 - t_T)$ and $e_B = (1 - e_T)$ denote the ground truth and the predicted class probability, respectively.

### 3.4 Training details

In our experiment, we have used Adaptive Moment Estimation (Adam) optimizer [23] to optimize the loss function. The proposed CNN is implemented using Python Image Library (PIL) and TensorFlow. We have also used MATLAB 2017$a$ to locate the region of interest (ROI) from input images before feeding them to the network. The proposed CNN model is trained with both the camera captured/flatbed scanned document images and synthetically generated warped document images. We have considered both warped [15] and flatbed scanned document images [3,6,15,33] for training. Among considered flatbed scanned document image datasets, handwritten document images are present in [6,33] and printed document images are present in [3,15]. 60%, 20%, and remaining 20% images are used for training, cross-validation, and testing, respectively. For checking pair-wise cross-validation, two visually similar but different kinds of images are taken. From these pair of images, training is performed on one image along with its ground truth image. Then, the other image is used to test its generalization accuracy, which is termed as cross-validation loss. To prevent overfitting of training data, CNN architecture, features, etc., are selected based on cross-validation loss (see Table-3). Back-propagation algorithm is used during the training process of the proposed CNN mode. Therefore, the computing time for the proposed model depends on the number of parameters of the proposed CNN. The total number of updations of derivatives for $L_{th}$ layer of the proposed CNN is –$Cout_L * (Ow_L * Oh_L) * (Fw_L * Fh_L * Cin_L)$, where $Ow_L$ and $Oh_L$ represent the width and height of output dimension, $Fw_L$ and $Fh_L$ signify width and height of filters, and $Cout_L$ and $Cin_L$ denote the number of output and input channels, respectively, for $L_{th}$ layer of the proposed CNN model. Figure (see Fig. 9a) demonstrates the performance of the proposed method based on the improvement of detection rate $(D_L)$ per millisecond (ms). It is evident that the proposed model achieves the highest detection rate $(D_L)$ per millisecond (ms) with a batch size of 10 images. So, the batch size is set as 10 during training.

The learning rate is set as 0.001. Figure 9 depicts the modification of loss and accuracy with the number of epochs. At iteration 20000, the training procedure is stopped as the training and validation losses are 0.000001 and 0.000085 (see

**Table 3** Cross-validation on printed and handwritten document images

| Method | Dataset | Training | Cross-validation Loss |
|---|---|---|---|
| Stewart et al. [54] | BESUS Dataset [6] | 0.4700000 | 0.6875 |
| | IAM [30] | 0.7654310 | 0.9900 |
| | Scanned [15] | 0.3845210 | 0.5500 |
| | Warped [15] | 0.6502100 | 0.8000 |
| | Alireza [3] | 2.8502100 | 3.5000 |
| Proposed Method | BESUS Dataset [6] | 0.0000001 | 0.0005 |
| | IAM [30] | 0.0000010 | 0.0010 |
| | Scanned [15] | 0.0000010 | 0.0900 |
| | Warped [15] | 0.0000100 | 0.1000 |
| | Alireza [3] | 0.010000 | 0.140000 |

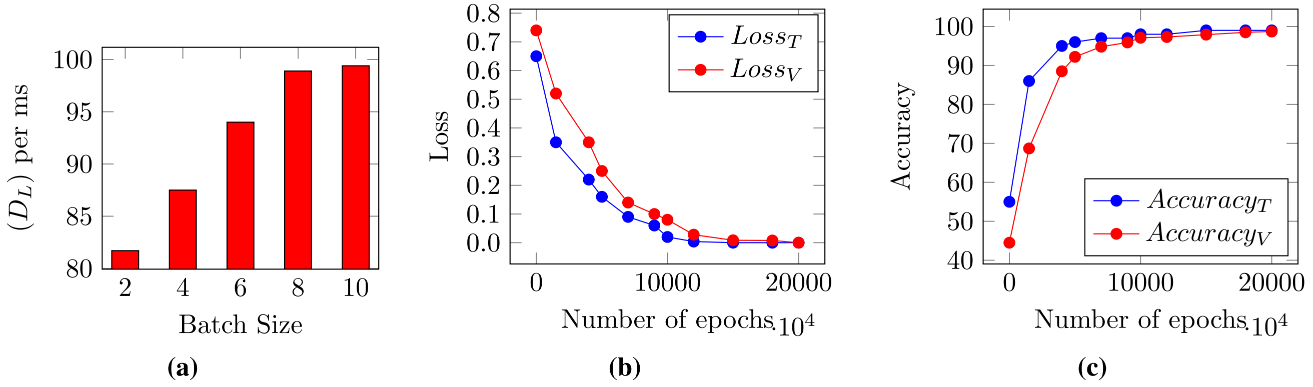

**Fig. 9** Performance analysis of the proposed model: **a** changes in detection rate $(D_L)$ per ms with the batch size; **b** changes in loss with number of epochs; **c** changes in accuracy with number of epochs. Here, $Accuracy_T$ and $Accuracy_V$ denote training and validation accuracy and $Loss_T$ and $Loss_v$

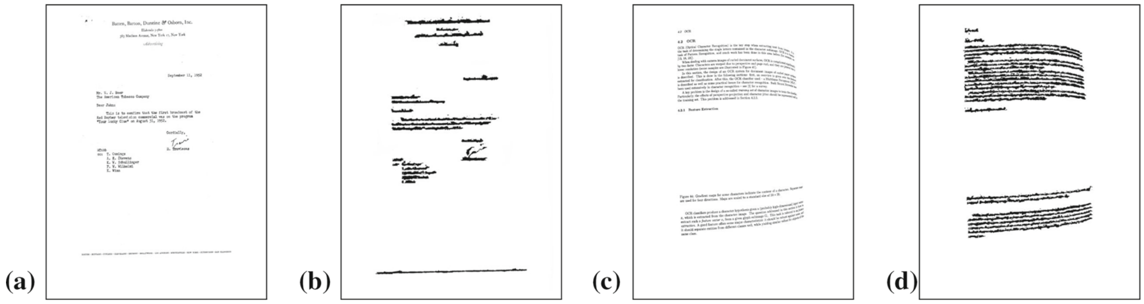

**Fig. 10** Result using proposed method on English printed document image: **a** input from Tobacco-800 dataset [55]; **b** corresponding output; **c** input from IUPR dataset [9]; **d** corresponding output

Fig. 9b) and the training and validation accuracies are 99% and 98.7% (see Fig. 9c), respectively.

## 4 Experimental results and evaluation

We compare the proposed method with other state-of-the-art methods on various publicly available datasets along with our developed dataset of warped document images WDID [15]. We use the standard evaluation metrics for fair comparisons and include the results reported in the published methods only. In the following three subsections, we discuss dataset, evaluation metrics, and obtained results, respectively.

### 4.1 Dataset

This work has been evaluated using both the flatbed scanned/camera captured printed and handwritten document images. For training, images from WDID [15], Alireza, IAM, and Besu [46] have been used. We have been selective in using the images for the training dataset. For example, there exist images (WDID [15], and Besus dataset [6,46]) containing a large amount of perspective distortion, multiple folds, etc., but they were not used during training. Note in the case of WDID [15], flatbed scanned images and single folded warped images with less skew (absolute value of skew angle $< 3^o$) have been used only for training. However, testing is done using warped with a large amount of skew (absolute value of skew angle ranges between $6^o$and $15^o$), multiple folded images, and perspective distorted images. Similarly, only the scanned images from Besus dataset [6] have been used for training, but the proposed approach has been tested on warped images of the Besus dataset [6,46] as well. To show the relevance of the proposed approach, it has been tested on four different types of publicly available datasets that were not used for training. They are (i) IUPR dataset [9], (ii) Tobacco-800 dataset [55], (ii) ICDAR 2013 Handwritten Segmentation Contest [53] dataset and (iv) cBAD datasets [10]. IUPR dataset [9] consists of warped printed document images having *English* script only. Tobacco-800 dataset [55] contains a total of 800 scanned printed English document images [27]. ICDAR 2013 Handwritten Segmentation Contest [53] dataset is a benchmark dataset that contains 150 images with English and Greek texts and 50 images with text in Bangla. The cBAD dataset with a total of 775 images [10] contains historical document images. The dataset has been used to signify the applicability of the proposed approach on historical document images.

## 4.2 Evaluation metric

The performance of the proposed approach is measured through (1) text-line detection rate, (2) precision, (3) recall, (4) *F-score* and (5) *IoU*. The rate of detection of text lines $D_L$ identifies the numbers of text lines that are correctly segmented with respect to the ground truth. The metric ($D_L = \frac{L_1}{L_0}$) can efficiently measure the penalty of misclassification of text lines, where $L_0$ is the total number of lines in the ground truth image and $L_1$ denotes the total number of text lines in the output images. On the other hand, two standard metrics, precision and recall, are defined as follows. Let, $P_R$ and $R_C$ denote precision and recall, respectively. Here, $P_R$ can be defined as -$P_R = \frac{t_p}{t_p+f_p}$. Similarly, $R_C$ is defined as -$R_C = \frac{t_p}{t_p+f_n}$. In our method, $t_p$, i.e., true-positives are those pixels which are correctly predicted as text lines, $f_p$, i.e., false-positives are those pixels which are originally background but are predicted as text line and $f_n$, i.e., false-negatives are those pixels that originally belong to text lines but are incorrectly predicted as background. Actually, in our problem, the prediction of false positives causes a big penalty between two consecutive text lines, especially where the inner gap between two consecutive lines is very less (for example, heavily warped text, warped and skewed images, etc.).

Beside this, the F-score ($F_M$), defined by -$F_M = 2 \times \frac{P_R \times R_C}{P_R + R_C}$, is also considered to determine the test accuracy. Moreover, intersection over union (IoU) is also considered as another metric where $IoU$ is defined as $IoU = \frac{Groundtruth \cap Prediction}{Groundtruth \cup Prediction}$. Here, $(Groundtruth \cap Prediction)$ signifies pixels found in both the prediction mask and the ground truth mask. On the other hand $(Groundtruth \cup Prediction)$ represents pixels found in either the prediction or ground truth mask. More specifically, this metric can be defined for each class at pixel level as -$IoU = 100 \times \frac{tp}{tp+fp+fn}$.

## 4.3 Result analysis

Results on various kinds of handwritten and printed document image datasets are evaluated through standard evaluation metrics. Encouraging results are obtained for both printed English and Bengali document images and are discussed separately.

**(a) Printed Document Images** Figure 10 shows the outputs of the proposed method tested on printed English document images from Tobacco-800 dataset [55] and IUPR dataset [9]. On the other hand, Fig. 11 shows the corresponding outputs of the proposed method tested on printed Bengali document images from WDID [15]. The text-line detection rate ($D_L$), precision ($P_R$), recall ($R_C$), F-score ($F_M$) and $IoU$ for printed document images are shown in Table 4.

**(b) Handwritten Document Images :** The text-line detection rate ($D_L$), precision ($P_R$), recall ($R_C$), F-score ($F_M$) and $IoU$ for handwritten document images are shown in Table 5. The outputs of the proposed method tested on handwritten Document Images from Besus dataset [6,46] and IAM dataset [33] are shown in Fig. 12.

To test the efficiency of our model on historical documents, we have tested our method on the cBAD dataset [10]. Renton et al. [42] presented different methods to evaluate the performance of text line segmentation algorithms using cBAD dataset [10]. Besides considering the result obtained from Renton et al. [42], we have also implemented methods presented by Moysset et al. [34], and Ahn et al. [1]. These methods are tested on cBAD dataset [10]. Table 6 shows the results of different text-line segmentation methods evaluated on cBAD datasets [10] along with the proposed algorithm. Besides performing well on previously unseen images of the training dataset, the proposed method has achieved remarkable results on those datasets which are not used for training. Figure 13 shows the results of the proposed method tested on Alireza dataset [3], cBAD datasets [10]. We have achieved significant accuracy while tested our method on the ICDAR2013 Handwritten Segmentation Con-

**Table 4** Performance of text-line detection for printed document images using Scanned [15],Warped [15], Perspective Distorted [15], Tobacco [55] and IUPR [9]

| Method | Dataset | $D_L$ % | $P_R$ % | $R_C$ % | $F_M$ % | $IoU$ % |
|---|---|---|---|---|---|---|
| Gatos et al. [18] | Scanned | 70 | 56.8 | 53.4 | 55 | 53.8 |
| | Warped | 55 | 42.7 | 49.9 | 46 | 43.7 |
| | Perspective Distorted | 40 | 33.1 | 37.2 | 35 | 32 |
| | Tobacco | 64 | 55.7 | 48.8 | 52 | 50 |
| | IUPR | 46 | 41.7 | 46.6 | 44 | 41.2 |
| Moysset et al. [37] | Scanned | 73 | 74.5 | 70.1 | 72.2 | 71 |
| | Warped | 55 | 56.2 | 53.1 | 54.6 | 55 |
| | Perspective Distorted | 47 | 49.7 | 46.8 | 48.2 | 49 |
| | Tobacco | 74 | 72.1 | 69.7 | 70.9 | 70 |
| | IUPR | 68 | 67.6 | 65.2 | 66.4 | 65 |
| Moysset et al. [35] | Scanned | 85 | 86.1 | 84.5 | 85.3 | 85 |
| | Warped | 74 | 75.7 | 74.2 | 80.0 | 79 |
| | Perspective Distorted | 69 | 67.6 | 63.7 | 68.2 | 67 |
| | Tobacco | 83 | 84.1 | 80.0 | 82.0 | 83 |
| | IUPR | 79 | 71.1 | 73.2 | 72.1 | 73 |
| Vo et al. [56] | Scanned | 77 | 63.7 | 58.6 | 61 | 58.8 |
| | Warped | 59 | 49.1 | 51.0 | 40 | 49 |
| | Perspective Distorted | 43 | 35.7 | 40.7 | 38 | 35.1 |
| | Tobacco | 69 | 62.1 | 56.2 | 59 | 56.2 |
| | IUPR | 52 | 46.7 | 51.6 | 49 | 46.8 |
| Proposed Method | Scanned | 100 | 100 | 100 | 100 | 98.8 |
| | Warped | 97 | 98.7 | 97.4 | 98 | 96.5 |
| | Perspective Distorted | 92 | 97.4 | 94.7 | 96 | 95.2 |
| | Tobacco | 95 | 98.4 | 95.7 | 97 | 96.1 |
| | IUPR | 92 | 95.1 | 93.0 | 94 | 92.4 |

**Table 5** Text-line detection rate on Handwritten script using Scanned [6], Warped [6], IAM [30] and Alireza [3]Dataset

| Method | Dataset | $D_L$ % | $P_R$ % | $R_C$ % | $F_M$ % | $IoU$ % |
|---|---|---|---|---|---|---|
| Ryu et al. [44] | Scanned | 89 | 88.2 | 85.3 | 86.7 | 85 |
| | Warped | 82 | 82.7 | 80.7 | 81.6 | 83 |
| | IAM | 88 | 89.6 | 85.7 | 87.6 | 86 |
| | Alireza | 79 | 77.2 | 71.3 | 74.1 | 75 |
| Moysset et al. [38] | Scanned | 50 | 47.1 | 51.0 | 49 | 47 |
| | Warped | 25 | 19.2 | 25.9 | 22 | 19.5 |
| | IAM | 55 | 55.4 | 49 | 52 | 46.2 |
| | Alireza | 44 | 37.2 | 33.1 | 35 | 37 |
| Vo et al. [56] | Scanned | 68 | 67.4 | 62.8 | 65 | 64 |
| | Warped | 43 | 45.2 | 39.3 | 42 | 40.5 |
| | IAM | 70 | 63.4 | 71.1 | 67 | 65.2 |
| | Alireza | 58 | 49 | 55.4 | 52 | 50.1 |
| Renton et al. [42] | Scanned | 79 | 80.5 | 75.7 | 78 | 77 |
| | Warped | 52 | 44.7 | 54.3 | 49 | 47 |
| | IAM | 77 | 74.7 | 77.4 | 76 | 74 |
| | Alireza | 62 | 63.7 | 66.4 | 65 | 63 |
| Proposed Method | Scanned | 99 | 96.1 | 94 | 95 | 94 |
| | Warped | 95 | 92.7 | 91.4 | 92 | 90.5 |
| | IAM | 97 | 94.5 | 93.6 | 94 | 92.8 |
| | Alireza | 91 | 94.2 | 91.9 | 92 | 90.1 |

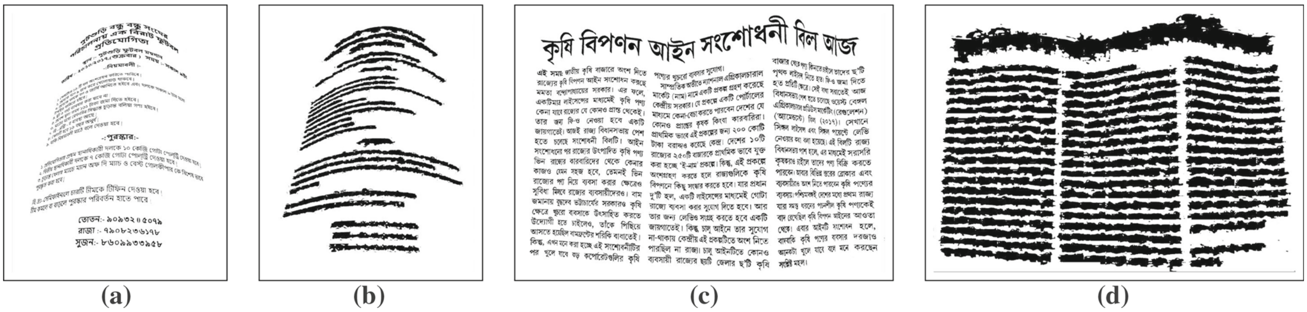

**Fig. 11** Result using proposed method on Bengali printed document images from WDID dataset: **a** perspective distorted image; **b** corresponding output; **c** multiple folded image; **d** corresponding output

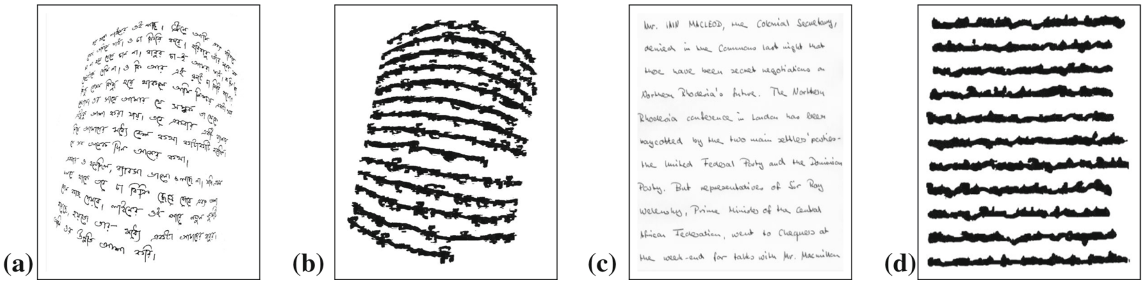

**Fig. 12** Result using proposed method on handwritten document images: **a** input image from Besus dataset [6]; **b** corresponding output; **c** input image from IAM dataset [30]; **d** corresponding output

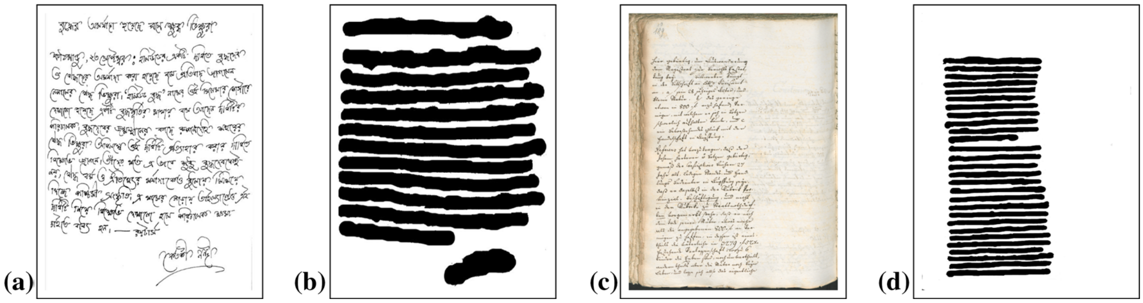

**Fig. 13** Result using proposed method on complex structured handwritten document images: **a** input image from Alireza dataset [3]; **b** corresponding output; **c** input image from cBAD datasets [10]; **d** corresponding output

**Table 6** Performance of text-line detection for historical document images

| Method | Precision % | Recall % | F-measure % |
|---|---|---|---|
| DMRZ | 97.3 | 97.0 | 97.1 |
| Renton et al. [42] 11 layers | 94.9 | 88.1 | 91.3 |
| UPVLC | 93.7 | 85.5 | 89.4 |
| BYU | 87.8 | 90.7 | 89.2 |
| IRISA | 88.3 | 87.7 | 88.0 |
| Moysset et al. [34] | 82.5 | 81.7 | 82.1 |
| Ahn et al. [1] | 77.2 | 76.1 | 76.64 |
| Proposed method | 99.7 | 99.2 | 99.49 |

**Table 7** Result evaluated on ICDAR 2013 Handwritten Segmentation Contest [53]

| Algorithm | M | o2o | DR % | RA % | FM % |
|---|---|---|---|---|---|
| CUBS | 2677 | 2595 | 97.96 | 96.64 | 97.45 |
| GOLESTAN-a | 2646 | 2602 | 98.23 | 98.34 | 98.28 |
| GOLESTAN-b | 2646 | 2602 | 98.23 | 98.34 | 98.28 |
| INMC | 2650 | 2614 | 98.68 | 98.64 | 98.66 |
| LRDE | 2632 | 2568 | 96.94 | 97.57 | 97.25 |
| MSHK | 2696 | 2428 | 91.66 | 90.06 | 90.85 |
| NUS | 2645 | 2605 | 98.34 | 98.49 | 98.41 |
| QATAR-a | 2626 | 2404 | 90.75 | 91.55 | 91.15 |
| QATAR-b | 2609 | 2430 | 91.73 | 93.14 | 92.43 |
| NCSR(SoA) | 2646 | 2477 | 92.37 | 92.48 | 92.43 |
| ILSP(SOa) | 2685 | 2546 | 96.11 | 94.82 | 95.46 |
| TEI(SoA) | 2675 | 2590 | 97.77 | 96.82 | 97.30 |
| Ryu et al. [44] | 2646 | 2612 | 98.64 | 98.68 | 98.66 |
| LAG-horizontal [56] | 2643 | 2583 | 97.51 | 97.73 | 97.62 |
| LAG-vertical [56] | 2643 | 2608 | 98.45 | 98.68 | 98.56 |
| Proposed | 2648 | 2640 | 99.67 | 99.7 | 99.68 |

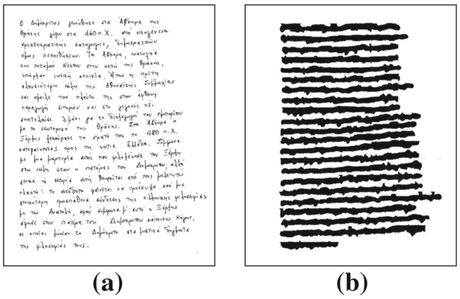

**Fig. 14** **a** Sample handwritten document images from ICDAR 2013 Handwritten Segmentation Contest [53]; **b** corresponding results using proposed method

test dataset [53]. To evaluate the performance of our method, we have used the metrics described in [53].

Table 7 shows the results of different segmentation algorithms presented in ICDAR 2013 Handwritten Segmentation Contest [53] along with some other algorithms and the proposed method. Figure 14 shows the output of the proposed method tested on Handwritten Document Images from ICDAR 2013 Handwritten Segmentation Contest [53].

## 5 Conclusion

In this paper, a very efficient method is presented for text line segmentation from both flatbed scanned or warped camera-captured document images. The method works for both handwritten and printed document images based on the concept of semantic segmentation. The demonstration of the relevance of the proposed work is given by testing using a wide variety of handwritten or printed document images. Though segmentation of text lines from warped document images is still a challenging field for researchers, the proposed method has achieved significant accuracy compared to the results already reported in the literature. Moreover, we have also tested our method with four different kinds of document image datasets that have not been used for training, and the results are equally encouraging. In the future, the proposed method can be fine-tuned for more accuracy even with highly warped and folded handwritten document images.

## References

1. Ahn, B., Ryu, J., Koo, H.I., Cho, N.I.: Textline detection in degraded historical document images. EURASIP J. Image Video Process. **2017**(1), 82 (2017)
2. Alaei, A., Pal, U., Nagabhushan, P.: A new scheme for unconstrained handwritten text-line segmentation. Pattern Recogn. **44**(4), 917–928 (2011)
3. Alaei, A., Pal, U., Nagabhushan, P.: Dataset and ground truth for handwritten text in four different scripts. Int. J. Pattern Recogn. Artif. Intell. **26**, 2012 (2012)
4. Antonacopoulos, A., Karatzas, D.: Document image analysis for world war ii personal records. In: Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on, pp. 336–341. IEEE (2004)
5. Asi, A., Rabaev, I., Kedem, K., El-Sana, J.: User-assisted alignment of arabic historical manuscripts. In: Proceedings of the 2011 Workshop on Historical Document Imaging and Processing, pp. 22–28. ACM (2011)
6. Biswas, S., Das, A.K.: Writer identification of bangla handwritings by radon transform projection profile. In: 2012 10th IAPR International Workshop on Document Analysis Systems (DAS), pp. 215–219. IEEE (2012)
7. Bukhari, S.S., Shafait, F., Breuel, T.M.: T.m.: Dewarping of document images using coupled-snakes. In: In: Proceedings of Third International Workshop on Camera-Based Document Analysis and Recognition, pp. 34–41 (2009)
8. Bukhari, S.S., Shafait, F., Breuel, T.M.: Text-line extraction using a convolution of isotropic gaussian filter with a set of line filters. In: 2011 International Conference on Document Analysis and Recognition (ICDAR), pp. 579–583. IEEE (2011)
9. Bukhari, S.S., Shafait, F., Breuel, T.M.: The IUPR Dataset of Camera-Captured Document Images, pp. 164–171. Springer, Berlin (2012)
10. cBAD: Scriptnet / icdar 2017 competition on baseline detection (cbad). https://scriptnet.iit.demokritos.gr/competitions/5/1/. (Accessed on 03/14/2019)
11. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: European conference on computer vision, pp. 184–199. Springer (2014)
12. Dumoulin, V., Visin, F.: A guide to convolution arithmetic for deep learning, 2016. arXiv preprint arXiv:1603.07285 (2016)
13. Dutta, A., Garai, A., Biswas, S.: Segmentation of meaningful text-regions from camera captured document images. In: 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT), pp. 1–4. IEEE (2018)
14. Eskenazi, S., Gomez-Krämer, P., Ogier, J.M.: A comprehensive survey of mostly textual document segmentation algorithms since 2008. Pattern Recogn. **64**, 1–14 (2017)

15. Garai, A., Biswas, S., Mandal, S.: A theoretical justification of warping generation for dewarping using cnn. Pattern Recognition **109**, 107621
16. Garai, A., Biswas, S., Mandal, S., Chaudhuri, B.B.: Automatic rectification of warped bangla document images. IET Image Processing (2019)
17. Garai, A., Biswas, S., Mandal, S., Chaudhuri, B.B.: A method to generate synthetically warped document image. arXiv preprint arXiv:1910.06621 (2019)
18. Gatos, B., Louloudis, G., Stamatopoulos, N.: Segmentation of historical handwritten documents into text zones and text lines. In: 2014 14th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 464–469. IEEE (2014)
19. He, J., Downton, A.C.: User-assisted archive document image analysis for digital library construction. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition, 2003, pp. 498–502. IEEE
20. Hendry, R.C.: Automatic license plate recognition via sliding-window darknet-yolo deep learning. Image Vis. Comput. **87**, 47–56 (2019)
21. Kil, T., Seo, W., Koo, H.I., Cho, N.I.: Robust document image dewarping method using text-lines and line segments. In: 2017 14th IAPR International Conference on Document Analysis and Recognition, vol. 01, pp. 865–870
22. Kim, B.S., Koo, H.I., Cho, N.I.: Document dewarping via text-line based optimization. Pattern Recogn. **48**(11), 3600–3614 (2015)
23. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
24. Kornfield, E.M., Manmatha, R., Allan, J.: Text alignment with handwritten documents. In: Proceedings of the First International Workshop on Document Image Analysis for Libraries, 2004, pp. 195–209. IEEE (2004)
25. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 (2012)
26. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
27. Lewis, D., Agam, G., Argamon, S., Frieder, O., Grossman, D., J.Heard: Building a test collection for complex document information processing. In: Proceedings of the 29th Annual International ACM SIGIR Conference, pp. 665–666 (2006)
28. Li, Y., Zheng, Y., Doermann, D., Jaeger, S.: Script-independent text line segmentation in freestyle handwritten documents. IEEE Trans. Pattern Anal. Mach. Intell. **30**(8), 1313–1329 (2008)
29. Liu, C., Zhang, Y., Wang, B., Ding, X.: Restoring camera-captured distorted document images. IJDAR **18**(2), 111–124 (2015)
30. Liwicki, M., Bunke, H.: Iam-ondb - an on-line english sentence database acquired from handwritten text on a whiteboard. In: Eighth International Conference on Document Analysis and Recognition, pp. 956–961 Vol. 2 (2005)
31. Liwicki, M., Indermuhle, E., Bunke, H.: On-line handwritten text line detection using dynamic programming. In: Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on, vol. 1, pp. 447–451. IEEE (2007)
32. Louloudis, G., Gatos, B., Pratikakis, I., Halatsis, C.: Text line and word segmentation of handwritten documents. Pattern Recogn. **42**(12), 3169–3183 (2009)
33. Marti, U.V., Bunke, H.: The iam-database: an english sentence database for offline handwriting recognition. Int. J. Doc. Anal. Recogn. **5**(1), 39–46 (2002)
34. Moysset, B., Adam, P., Wolf, C., Louradour, J.: Space displacement localization neural networks to locate origin points of handwritten text lines in historical documents. In: Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing, HIP '15, pp. 1–8. ACM, New York, NY, USA (2015)
35. Moysset, B., Kermorvant, C., Wolf, C.: Full-page text recognition: Learning where to start and when to stop. In: 2017 14th IAPR International Conference on Document Analysis and Recognition, vol. 01, pp. 871–876 (2017)
36. Moysset, B., Kermorvant, C., Wolf, C., Louradour, J.: Paragraph text segmentation into lines with recurrent neural networks. In: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on, pp. 456–460. IEEE (2015)
37. Moysset, B., Kermorvant, C., Wolf, C., Louradour, J.: Paragraph text segmentation into lines with recurrent neural networks. In: 2015 13th International Conference on Document Analysis and Recognition, pp. 456–460 (2015)
38. Moysset, B., Louradour, J., Kermorvant, C., Wolf, C.: Learning text-line localization with shared and local regression neural networks. In: Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on, pp. 1–6. IEEE (2016)
39. Nagy, G., Seth, S.: Hierarchical representation of optically scanned documents (1984)
40. Rath, T.M., Manmatha, R.: Word image matching using dynamic time warping. In: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, vol. 2, pp. II–II. IEEE (2003)
41. Rath, T.M., Manmatha, R., Lavrenko, V.: A search engine for historical manuscript images. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 369–376. ACM (2004)
42. Renton, G., Soullard, Y., Chatelain, C., Adam, S., Kermorvant, C., Paquet, T.: Fully convolutional network with dilated convolutions for handwritten text line segmentation. IJDAR **21**(3), 177–186 (2018)
43. Roy, P.P., Rayar, F., Ramel, J.Y.: Word spotting in historical documents using primitive codebook and dynamic programming. Image Vis. Comput. **44**, 15–28 (2015)
44. Ryu, J., Koo, H.I., Cho, N.I.: Language-independent text-line extraction algorithm for handwritten documents. IEEE Signal Process. Lett. **21**(9), 1115–1119 (2014)
45. Saabni, R., Asi, A., El-Sana, J.: Text line extraction for historical document images. Pattern Recogn. Lett. **35**, 23–33 (2014)
46. Samit, B.: Department of computer science and technology. https://oldwww.iiests.ac.in/index.php/researchsamitbiswas-cst-menuitem
47. Shi, Z., Govindaraju, V.: Line separation for complex document images using fuzzy runlength. In: Proceedings of the First International Workshop on Document Image Analysis for Libraries, 2004. pp. 306–312. IEEE (2004)
48. Shi, Z., Setlur, S., Govindaraju, V.: Text extraction from gray scale historical document images using adaptive local connectivity map. In: Proceedings of the Eighth International Conference on Document Analysis and Recognition, 2005, pp. 794–798. IEEE (2005)
49. Shi, Z., Setlur, S., Govindaraju, V.: A steerable directional local profile technique for extraction of handwritten arabic text lines. In: 10th International Conference on Document Analysis and Recognition, 2009. ICDAR'09, pp. 176–180. IEEE (2009)
50. Shivakumara, P., Phan, T.Q., Tan, C.L.: A laplacian approach to multi-oriented text detection in video. IEEE Trans. Pattern Anal. Mach. Intell. **33**(2), 412–419 (2011)
51. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
52. Song, Y., Liu, A., Pang, L., Lin, S., Zhang, Y., Tang, S.: A novel image text extraction method based on k-means clustering. ICIS **08**, 185–190 (2008)
53. Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., Alaei, A.: Icdar 2013 handwriting segmentation contest. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1402–1406. IEEE (2013)

54. Stewart, S., Barrett, B.: Document image page segmentation and character recognition as semantic segmentation. In: Proceedings of the 4th International Workshop on Historical Document Imaging and Processing, pp. 101–106. ACM (2017)
55. Tobacco: The Legacy Tobacco Document Library ($LTDL$). http://legacy.library.ucsf.edu/
56. Vo, Q.N., Kim, S.H., Yang, H.J., Lee, G.S.: Text line segmentation using a fully convolutional network in handwritten document images. IET Image Process. **12**(3), 438–446 (2017)
57. Ye, Q., Gao, W., Wang, W., Zeng, W.: A robust text detection algorithm in images and video frames. In: Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on, vol. 2, pp. 802–806. IEEE (2003)
58. Yin, F., Liu, C.L.: Handwritten chinese text line segmentation by clustering with distance metric learning. Pattern Recogn. **42**(12), 3146–3157 (2009)
59. Zhu, X., Qian, Y., Zhao, X., Sun, B., Sun, Y.: A deep learning approach to patch-based image inpainting forensics. Signal Process. Image Commun. **67**, 90–99 (2018)

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