The Visual Computer (2021) 37:291–305
https://doi.org/10.1007/s00371-020-01799-4

ORIGINAL ARTICLE

Meitei Mayek handwritten dataset: compilation, segmentation, and character recognition

Sanasam Inunganbi[1] · Prakash Choudhary[2] · Khumanthem Manglem[1]

Published online: 31 January 2020

Abstract

A peculiar Indian Script Meitei Mayek has experienced a resurgence in the last few years and gets very little attention in handwriting research due to recently insurgence and limited sources. The objective of this paper is two folds; firstly, develop two different datasets: *Mayek27* having 4900 isolated Meitei Mayek alphabets and MM (Meitei Mayek) dataset of 189 full-length handwritten text page. Secondly, develop a recognition system on the Mayek27 dataset using convolutional neural network and segmentation algorithms (text-lines, words, and characters) on the full-length Meitei Mayek handwritten text. A recognition rate of 99.02% is achieved using three layers of convolutional layers with a filter size of 3×3 with 16, 32, and 96 kernels. In MM text dataset, the text-line and word segmentation are performed concurrently on 809 lines by tracking space between lines in a novel approach based on horizontal projection histogram and monitoring vertical projection histogram along the run-length of segmentation. Various constraints like skew, curve, close, and touching text-lines are incorporated, and the segmentation algorithm results are 91.84% and 88.96% for text-line and word, respectively. Furthermore, characters are segmented by headline removal, and connected component analysis achieves an accuracy of 91.12%.

Keywords Meitei Mayek · Indian language · Character recognition · Segmentation

1 Introduction

Identifying and recognizing handwritten characters and digits are one of the core problem and prominent task in computer vision society. It has achieved an abundance of interest for its diverse applications in helping to recognize characters from images. The applications stretch from zip code identification to writer recognition, from recognizing numerals and alphabets in the number plate of traffics to bank check processing, etc. Moreover, it can be practically used to convert handwritten characters to ASCII or Unicode, which can serve as a standard in further processing, such as translation to other language or speech synthesis. However, constructing a design for handwritten character recognition establishes various challenges to the researchers due to the problems that prevail in data acquisition and its nature of unconstrained handwritten characters or text. The shape of the same character that may differ depending upon the writers, some may write with large structure, while others may complete in small-scale version. Hence, the overlap area of ink trace of the character may be very less.

Further, depending on the acquisition device, pen width and ink color may impose variation on writing style. Moreover, handwritten Manipuri characters are complicated due to their structure and shape. They include a significant character set with more curves, loops, and other details in the characters. Many character pairs are quite similar in shape. All these issues demand attention and solution with the help of an efficient recognition system. The recognition process requires the documents to decompose into identifiable elements. In a simple form, the recognizable elements can be isolated character which can be applied directly for recognition. If sentences or paragraphs are taken into consideration, then the segmentation of line, word, and character is required to attain the level of identifiable elements. Segmentation is

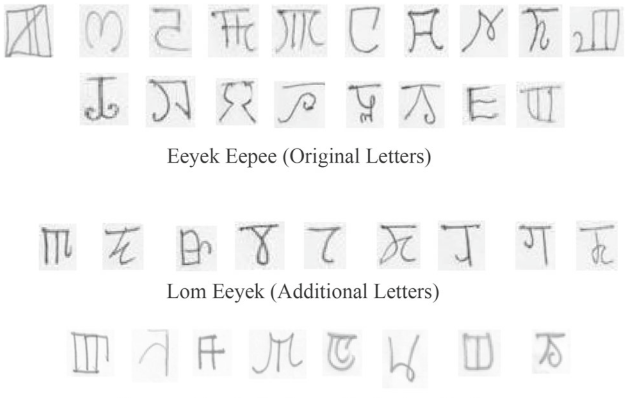
✉ Sanasam Inunganbi
inung.sam@gmail.com

Prakash Choudhary
choudharyprakash87@gmail.com

Khumanthem Manglem
manglem@gmail.com

[1] National Institute of Technology, Langol, Imphal, Manipur 795004, India

[2] National Institute of Technology, Hamirpur, Himachal Pradesh, India



Eeyek Eepee (Original Letters)

Lom Eeyek (Additional Letters)

Lonsum Eeyek (Letters with short ending)

Fig. 1 Meitei Mayek alphabets

the elementary constituent for a handwritten text recognition system, and therefore, it has drawn the attention of numerous researchers [1–5]. In this paper, we have introduced self-collected Meitei Mayek datasets for handwritten isolated characters and unconstrained written text. Moreover, to validate the collected datasets, character recognition on isolated character dataset and segmentation (lines, words, and character) on text dataset is presented.

Manipuri (Meitei Mayek) is a scheduled language of the Indian constitution and the official language of Manipur state. It belongs to the Tibeto-Burman branch of the Sino-Tibetan language family and is the primary communicating language of Manipur, which has spread in the northeast and some parts of Bangladesh and Myanmar. The current script is the reconstruction of the ancient Meitei Mayek script, and it consists of rich set of characters which involve 10 numerals called *Cheising Eeyek*, 27 basic alphabets (which is further comprised of 18 original letters called *Eeyek Eepee* and 9 additional called *Lom Eeyek*), 8 derived letters called *Lonsum Eeyek* and 8 associating symbol called *cheitap Eeyek*. Their architecture is given in Fig. 1. In this paper, we have considered the 27 basic alphabet and 8 derived letters only, for the experiment in this paper. The script has been reinstated recently, and there is no standard database available for research. Unavailability of a conventional database on this script has made a hindrance to research work on this script.

Keeping in view the prevailing problem of unavailability of a dataset, we have attempted to develop a Meitei Mayek database of isolated characters (Mayek27) and unconstrained full-length Meitei Mayek (MM) handwritten text. The database of Meitei Mayek (Mayek27) characters will form a basis for recognition of this script. Most of the previous studies on Meitei Mayek have been carried out on isolated numerals and characters only. Therefore, the segmentation method on the full-text MM handwritten text dataset may form a benchmarking procedure. Further, to validate the proposed databases, we have performed some experiments on the databases. In the Mayek27 database, a convolutional neural network-based character recognition system is proposed on 4900 image. On the dataset MM, we have performed text-line segmentation based on the morphological operation to identify midpoints between consecutive lines and accordingly trail the gap block by block till the entire line is separated. Word segmentation is not considered a separate problem but goes effectively alongside the run-length of line segmentation by identifying the optimal column to separate words. Moreover, the connected component-based technique is applied to the segmentation of individual characters from the words. The proposed segmentation algorithm works well on curve, touching, and skew lines.

The remainder of the paper is organized as follows: Sect. 2 reviews numerous existing methods and techniques available in the literature for the character recognition system and various segmentation algorithms. Section 3 presents the acquisition of databases: Mayek27 and MM in detail. Section 4 presents multiple experiments performed on the databases, character recognition algorithms on Mayek27 dataset and text-line, word and character segmentation on MM dataset, and their results. Section 5 concludes the paper stating the achievement and future work.

2 Related work

This section has provided with an extensive survey on various character recognition algorithms that exist in the literature and methods for segmentation to text documents to identifiable elements such as text-line, words or characters.

2.1 Review on character recognition

Various character recognition systems have been developed for different languages in the world. Numerous feature extraction techniques existed in the literature are based on the statistical attribute of the image in the spatial domain, while some are based on the transformed field. The spatial feature deals with the same spatial variables as the original image, where each pixel holds information of the image. The transformed domain converts an image to a different domain such as frequency or time that facilitates the feature description that may not be conveniently exposed in the spatial area. The extraction of a distinctive feature is a vital stage in the recognition process. Feature selection should promote increase interclass differences and decrease the intraclass gap.

The spatial feature explores the demographic attribute and structural characteristics of an image individual or combine. Pixel values are manipulated, or their correlation is estimated to extract discriminant properties, and this can be extracted globally or locally, depending on the application. Pixel-based methods have been explored for its essential characteristics for recognition in the literature [6–8]. Either the direct binary ink trace (pixel pattern) or its density has been used for recognition problems. The prominent texture descriptor, Local Binary Pattern (LBP), had been reported in the literature for various pattern recognition system [9–11]. Many different forms of LBP can be obtained based on the distance and orientation of sampling points. The effectiveness of the LBP in various types had been explored, and their results had been discussed. Further, the classification of an image by fusion of global feature using wavelet transform of local ternary pattern and local features such as speed up robust feature descriptor and bag of words has been proposed in [12]. Shape features like chain code [13] and histogram feature for gradient were exploited in [14] for character recognition. The decomposition of complex objects into elementary shapes has been explored in [15] for pattern recognition. The analysis of the shapes will help in finding the relevant primitives in recognizing an object. The 2D SIFT algorithm [16] has been extended to a perfectly scale-invariant feature selection algorithm of a 3D mesh for 3D object recognition in [17]. Recognition of character by air-writing through hand or fingertip is an interesting mechanism. One such work has been proposed in [18], where the color and depth images of the Kinect sensor are used to identify the writing. The proposed method utilizes slope variation detection to extract features from the trajectory to identify Persian numbers.

The spatial domain feature extraction misses the frequency component of the image. Therefore, researchers were inspired to exploit the transformed domain for feature extraction. Low-frequency components relate basic shape, and high-frequency components of the transform domain describe details of an image. That is the reason for considering the coefficient of Fourier transform (FT) for feature extraction in [19,20]. Despite being a robust descriptor, the FT captures the overall properties of an image and often excludes local features. So, to achieve fine details, wavelet transform (WT) has been examined in [21]. With a valid numerical basis, the WT can analyze an image extensively by constructing a ground wavelet. On account of the strong response for localized target along a particular direction, orientation-based Gabor filters have been explored for character recognition [22,23].

A dynamic transformation called Stockwell transform [24] has been proposed to acquire the time-frequency domain description of an image, and these properties have been considered for various pattern recognitions by researchers [25–28]. A mathematical model called Markov model-based [29] and probabilistic and fuzzy feature-based [30] pattern recognition were reported in the literature as well for character recognition.

Image zoning is another technique in the literature that was used in recognition of handwritten characters as it can address variation in writing patterns. The entire image is broken down into numerous sub-images called zones that each can provide useful information related to a specific part of an image. The optimal selection of the zone from where the localized feature was exploited had been proposed in [27], which was based on a bio-inspired process. The zoning process iterates itself until error had been minimized. Another zoning method combined with membership function had been proposed in [31,32]. The role of zone membership functions had been elaborated by defining the influence that different zones had contributed to the overall feature. It had been proposed to select the best-suited membership function for each zone of an image to exploit the characteristics of feature distribution of that zone. Their experimental results had claimed to be superior to other traditional zoning methods.

In contemporary years, deep learning architecture [33,34] has acquired much attention for various applications of pattern recognition and computer vision problems. The anticipation of deep learning helps in the commencement of the convolutional neural network (CNN) in machine learning algorithms. A multilayer artificial neural network has been introduced for character recognition and various other computer vision problems in [35]. A classic network popularly known as Lenet has been proposed by Lecun et al. [36], which incorporates gradient-based learning methods to the convolutional neural network (CNN). Further, similar to Lenet network but quite bigger and powerful has been introduced in [37]. Experiments have been conducted on the ImageNet dataset for classification. The network used ReLu, and multiple GPUs with a special layer called local response normalization (LRN). A remarkable scheme is proposed in [38] where instead of having as many hyperparameters, the focus has been made on the evaluation of simpler networks where convolution layer of (3 × 3) filters is fixated with increasing depth.

2.2 Review on document segmentation

Segmentation of a transcribed document image into lines, word, and character is a significant problem to solve due to the complication occurs in the handwritten document, such as the irregular spacing between lines and words and touching of characters across words and text-lines. Although many algorithms have been proposed and enormous effort devoted to the segmentation of text-lines and words for unconstrained handwritten documents, but still there is plenty of room for improvement.

Methods for text-line detection and segmentation of printed document are relatively easy and have explored [13,39] as they have approximately straight with parallel text-line, and global projection profile can segment them. But, handwritten documents are often non-uniformly spaced and associated with skew and curve. Many efforts have been devoted to solving the challenging task of the handwritten text-line and word segmentation. The approaches can be categorized broadly as projection profile analysis [7,40–49], connected component grouping [50–52], and level set method [53].

The $X - Y$ cut algorithm [44] is a projection-based top-down segmentation method but performs well only on documents of the parallel text-line and a large gap. The partial-projection profile [41] has been proposed to deal with curve lines. The level set method [53] is an effective top-down approach for the segmentation of unconstrained handwritten documents, but it has high computation complexity. The Docstrums method [50] is a bottom-up approach that merges neighboring connected components but fails to detect some lines. The piece-wise projection [47] segmentation method is highly sensitive to variation in the size of characters.

In [54], line and word segmentation detection techniques using midpoint had been proposed. The midpoint detection-based approach is based on the recognition of spaces that separates two lines or words. Another text-line and word segmentation algorithm were proposed in [52], where the text-lines are segmented and normalized, and then words are segmented. In their technique, the distance between connected components is measures, and lines and words are segmented based on a threshold. A hough transformation-based lines and word segmentation from document images were presented in [55,56]. The proposed technique not only applied on document images but also on a dataset for the business card reader system and license plate recognition system. Although the algorithm has low under segmentation results, it sometimes fails to segment closely spaced lines.

In [57], the segmentation of words was formulated as a binary quadratic assignment problem that considered the pairwise correlation between the gaps as well as the likelihood of individual gaps. The parameters are estimated on a structured SVM framework so that it is independent of language and writing style. Segmentation of lines, words, and characters was presented in [58]. In this paper, the authors had extracted a horizontal projection profile from the document, and using local minima points line segmentation was performed. Further, simultaneous word and character segmentation are proposed by popping out column runs from each row in an intelligent manner.

3 Compilation of Meitei Mayek dataset

Data acquisition plays a significant role in the research area. It accounts for gathering and estimating relevant information to develop a target system, here handwritten character recognition system. There is no publicly available dataset for handwritten Meitei Mayek characters. Therefore, we have manually collected isolated characters from various people who can read and write Meitei Mayek for the development and evaluation of efficient character recognition. Previously, an isolated handwritten Meitei Mayek dataset has been proposed in [59], but it consists of the only 27 classes of *Eeyek eepee*. In this paper, we have included the 8 letters called *Lonsum Mayek*, which are derived from distinct *Eeyek Eepee*. So, in total, there are 35 classes of Meitei Mayek characters considered for recognition in this paper. The derived characters are very similar to their respective original, which further adds to the challenge in recognizing them as highlighted in Fig. 2. Figure 3 illustrates an instance of a filled form of Meitei Mayek dataset having four sections: a unique label, demographic information holder, printed character or text, and empty slot(s) for various individuals to inscribe the written character in their writing style.

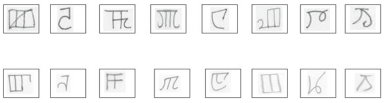


Fig. 2 The original and derived letters of Meitei Mayek alphabets, images in the first row are the original letters and the second row signifies the respective derived lonsum Eeyek

3.1 Mayek27 dataset

It consists of 35 letters, and they have been collected in a set of 4 in 140 pages of the A4 sheet. The isolated characters are raised in a tabular format where a cell is occupied by one handwritten character sample. Every page has been provided with 35 empty slots for various individuals to inscribe the printed character in their writing style. Since every character has been sampled 35 instances in a set, a sub-total of 1225 (35 × 35) isolated characters are collected for a set. Therefore, considering all the four sets, there are a total of 4900 Meitei Mayek characters available for experimentation in this work. To complete the data acquisition process, 90 people have contributed to their writing habits. These people have a different educational background and have a mixed age group between 6 and 40 years. The writers also record their demographic information in the dataset form such as name, address, occupation, qualification signature so that other applications like signature verification can utilize the data. The preprocessing methods performed in this paper are similar to the approach described in our previous work in [59,60] and are illustrated in Figs. 4 and 5.

3.2 The Meitei Mayek (MM) dataset

It is devoted to the text document database having words comprising of Meitei Mayek and English. The MM dataset has been developed because there exist English words whose Meitei Mayek equivalent does not avail. The text documents have been incorporated with various challenges to make the

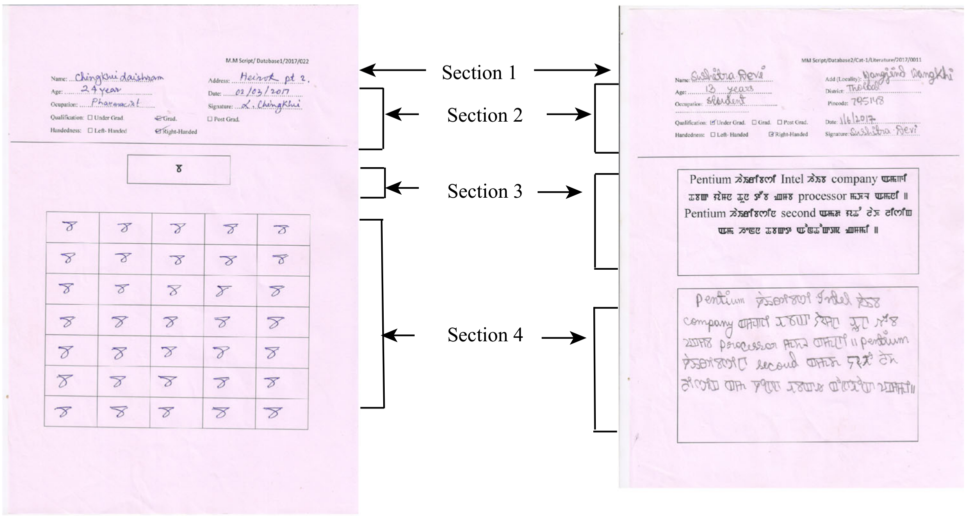

Section 1
Section 2
Section 3
Section 4

Fig. 3 Sample filled forms of Mayek27 and MM datasets

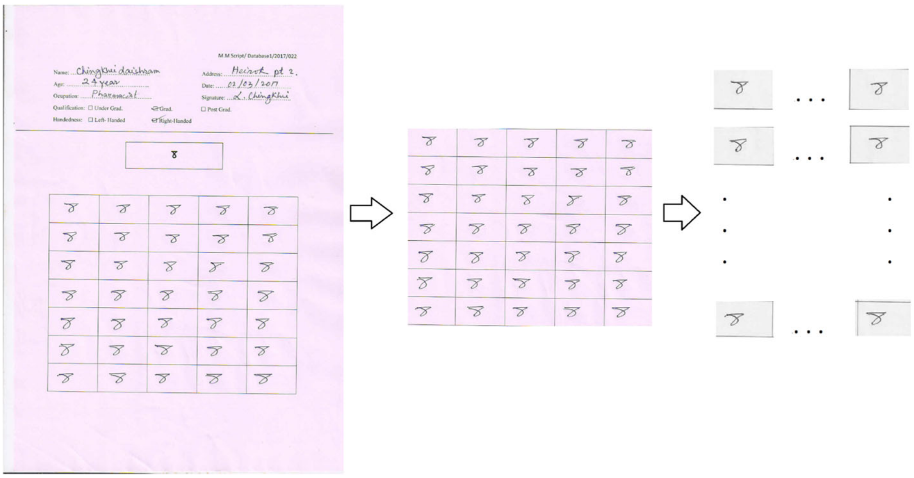


Fig. 4 Preprocessing performed in Mayek27 dataset

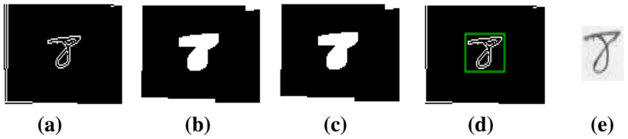


Fig. 5 Process for obtaining minimum bounding rectangle of a character in Mayek27 **a** edge image, **b** dilate image, **c** filled image, **d** minimally bounded box, **e** cropped image

segmentation problem more interesting such as skew, curve, close, and touching lines. In total, 189 documents pages have been collected from 114 peoples of varied age group consisting 809 lines.

Besides the handwritten characters and text for experimental evaluation, we have also collected the demographic information of the writer. This information can be made available for other applications, such as signature verification. A sample format of filled Meitei Mayek dataset (isolated character and text documents) is illustrated in Fig. 3 consisting of four different sections.

The developed database will be made available to the public for research. A researcher working on Meitei Mayek can easily download them and use it for their work. The generated dataset contained samples of both printed and handwritten characters and text. So, it can also be used for separate recognition of printed as well as handwritten character. Identification of printed and handwritten text can also be performed. Moreover, demographic information collected can be used in other applications such as signature verification, writer identification from the names, pincode identification from the address. Besides, the Meitei Mayek has been reinstated recently, and full focus is being made so that the proposed datasets will be beneficial in the future for various digital applications research work. The Mayek27 and MM dataset have been scanned at 300 dpi and stored in the "png" format for further processing.

4 Experimentation on datasets

For the technical validation and contribution of the datasets as a standard benchmark platform for linguistic research on

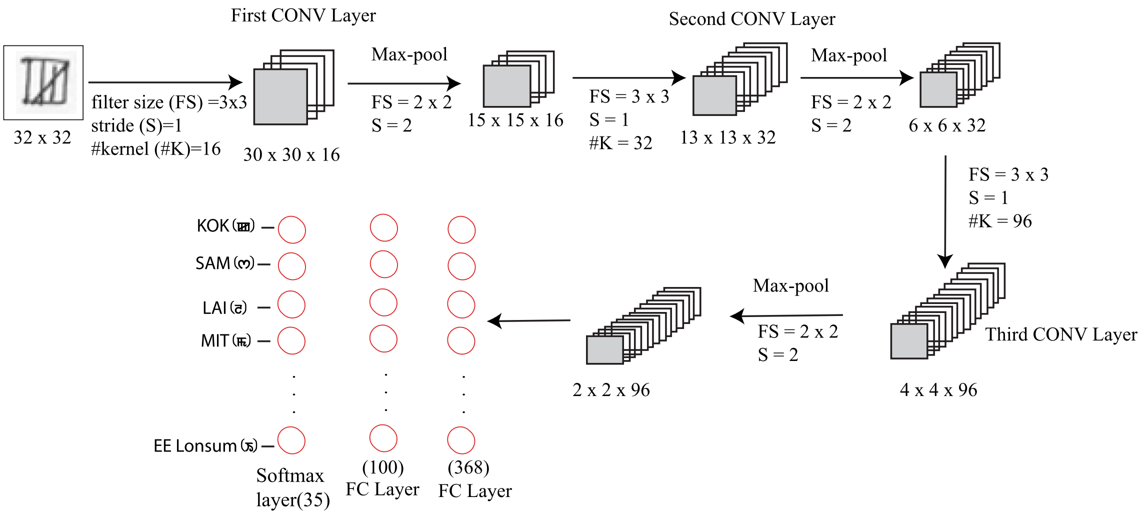


Fig. 6 A sample architecture of the proposed convolutional neural network for Meitei Mayek recognition

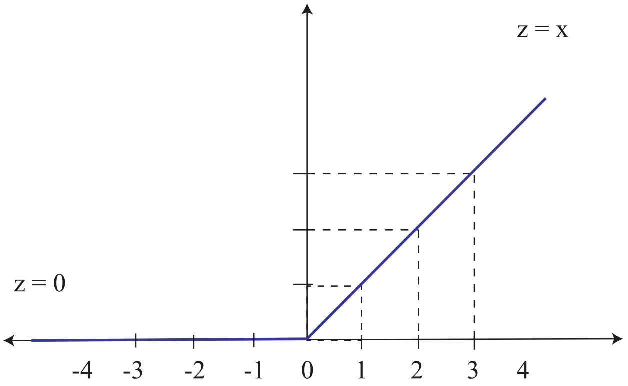


Fig. 7 Visual illustration of ReLU activation function

Meitei Mayek script, we have applied character recognition technique on the Mayek27 dataset and text-line, word, and character segmentation on the MM dataset and present the experimental results.

4.1 Character recognition on Mayek27 dataset by CNN

The experiment has been carried out on the collected samples of Mayek27 Dataset, having 4900 samples. All the images are isolated individually and normalized to a fixed size of 32 × 32 to complete the recognition system.

Convolutional neural networks (CNN) are a genre of deep, feed-forward artificial neural networks. Deep learning provides a powerful set of techniques for learning in neural networks, which are successfully adapted to various applications of investigating visual imagery. Convolutional neural networks are designed that enables a computer to learn from observational data. The CNN is commonly sequenced by a set of layers that can be aggregated by their functionalities as illustrated in Fig. 6 and interpreted as follows:

4.1.1 Convolution layer

The convolution operation is one of the fundamental building blocks of a CNN. It performed 2D convolution to the input image with the filters' weight and harmonized them across the channels. The filter has an equal number of layers as the input image channels, while the output volume has the same depth as the number of filters. The advantage of having a convolution layer over having only FC layers is parameter sharing and sparsity of connections. Parameter sharing defines that if a feature detector is useful in one part of the image is probably useful in another part of the image as well. Sparsity signifies that in each layer, each output value depends only on a small number of inputs.

An activation function introduces the nonlinearity mapping between the input and the output. It also promotes robustness and strength to the network to determine something complicated and productive from the image. Another significant characteristic of an activation function is the differentiability to perform backpropagation escalation procedures. The backward propagation in the network computes gradients of error(loss) concerning weights and then accordingly optimize weights using gradient descent. Therefore, the activation layer increased the nonlinearity of the network without affecting the respective fields of the convolution layer.

$$R(z) = \max(0, z) \tag{1}$$

Some of the popular types of activation functions are the sigmoid, tanh, and rectify linear unit (ReLu). In this approach, we have used the rectify direct unit (ReLu), which is given by Eq. 1 and visually illustrated in Fig. 7. This method solves the vanishing gradient problem in a rational approach. The function returns zero for all negative values and preserves the linearity for the positive values. Therefore, the ReLU is sparsely activated and is more likely to process a meaningful aspect of the problem. However, it should only be used within the hidden layers of a neural network model. Hence for the output layer, a special kind of activation function called the softmax layer is used at the end of fully connected layer output to compute the probabilities for the classes (for classification). For a given sample vector input x and weight vectors w_i, the predicted probability of $y = j$ is given by Eq. 2.

$$P(y = i|x) = \frac{e^{x^T w_j}}{\sum_{k=1}^{K} e^{x^T w_k}} \tag{2}$$

4.1.2 Pooling layer

It can be inferred from the previous section that the convolution layers provide activation mapping between input and output variables, pooling layers employ nonlinear downsampling on activation maps. This layer basically takes a filter (normally of sizes 2 × 2) and a stride of the same length. It then applies it to the input volume and outputs the maximum number (in case of max-pooling) in every subregion that the filter convolves around.

The pooling operation preserves any feature detected in any quadrant of the image in the output of the processed. The popular type of pooling method is average and max-pooling. In this paper, we have used max-pooling where no zero paddings on the image are required. It uses two hyper-parameters, filter size FS and stride S. For an image of size, $M \times N \times D$, pooling results in $\left(\lfloor \frac{M-f}{s} + 1 \rfloor \times \lfloor \frac{N-f}{s} + 1 \rfloor \times D\right)$ size output. The pooling operation requires no parameter to learn. We have used filter size, $FS = 2 \times 2$ and stride, $S = 2$ for all the max-pooling operation in this approach.

4.1.3 Fully connected layer

It can be perceived as a regular neural network with ultimate learning acquiring all possible visual features to relate to the appropriate output labels. Fully connected (FC) layers are usually adaptive to classification or encoding tasks with a common output of a vector, which, when the Softmax layer displays the confidence level for classification.

The model of CNN architecture in this approach consists of three convolution and pooling layers, one pooling later applied after each convolution layer for downsizing the sampling to half. Finally, there are three fully connected layers, with the last being the Softmax layer (Eq. 3). As described in Eq. 2, this layer postulates a probability distribution over a fixed number of categories and selects the category that has the maximum probability designated by the network. Adding more layers without having a large training set may likely introduce irregularities in the data, causing over-fitting and in turn, reduce accuracy on the test data. Therefore, in this approach, we have limits to only three convolution layers as we do not large dataset.

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \tag{3}$$

The first convolution layer accepts a character image of size 32 × 32 for the start of distinctive feature extraction for recognition. Every convolution layer has the same filter size of 3 × 3, but the number of kernels for every convolution layer varies. In the first layer, we have employed 16 kernels, followed by 32 and 96. The increase in the number of kernels helps in growing a network in volume, thereby boosting the representational power of the network as the number of units per layer becomes larger. Each convolution layer is passed through the ReLu activation layer, and the ReLu layer output is downsized to half by a max-pooling layer of size 2 × 2, as illustrated in Fig. 6. After the final pooling layer, the image size is reduced to 2 × 2 with 96 channels from the number of kernels employed. Then, the image is vectorized and flatten to be passed to a fully connected convolution layer, having dimension 368 and then to another FC layer of size 100 and lastly to the Softmax layer, which generates a probability distribution over the classes for a given input. In this architecture, we have applied the regularization technique, batch normalization on every layer to facilitate network training and diminish the sensitivity to network initialization.

4.1.4 Experimental results and comparison

The CNN model is tested on a self-collected handwritten Meitei Mayek isolated character dataset of size 4900 sample images all normalized to 32 × 32. This CNN model works well even with not so large number of sample images. A validation accuracy of 99.02% of correct recognition is obtained on 20th epoch. It took about 146 s to reach until 20th epoch and reached final iteration. The accuracy has been observed to be above 90% as it reaches second epoch and is consistently going toward higher accuracy.

Further, as the network goes more in-depth with the higher number of convolution layers and filters, complex and detail information is gained; however, to a certain limit only, beyond that, there is a possibility of over-fitting. It can be illustrated visually in Fig. 8; more meaningful information can be perceived as the number of convolution layer proceeds toward the Softmax layer with the expense of complex computation. The CONV layer one can be seen as pure black

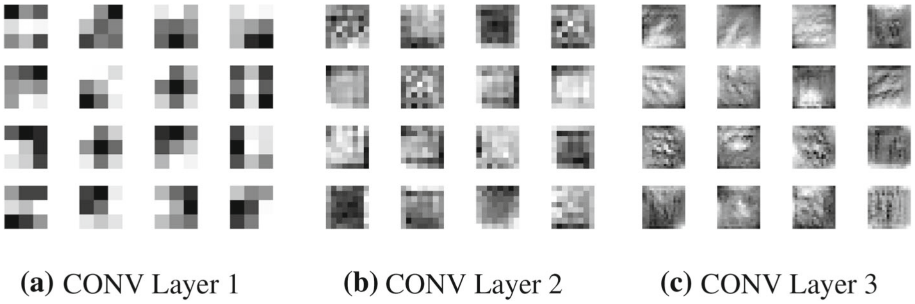


Fig. 8 Images that activate the channels within the network through the layers (for simplicity 16 images from each CONV layer are shown)

Table 1 Comparison of recognition rate (RR) with the existing methods in the literature

Papers	Recognition methods	RR
[30]	Probabilistic and fuzzy feature with ANN	90.3%
[7]	Binary pattern as vector and NN with back propagation	80%
[61]	Distance profile & BDD	98%
Proposed method	CNN	**99.02%**

The bold entry signifies the best results

and white blocks stack together. However, in the process, as the network advances toward CONV Layer 3, it can be realized that meaningful image can be seen, which can be used for classification. These images are focused on the features cultivated by the network.

We have also experimented on the additional fourth layer and fifth layer to analyze the change in recognition accuracy. However, it has been observed that it is not necessarily true to improve accuracy as the number of layers increases. The experimental analysis shows a decline in recognition accuracy, however, slightly as the additional layer has been added. This is because of the fact that increasing layer definitely extracts more feature but up to a certain extent only. There is a possibility of over-fitting the data and may results in false-positive if training data are not large enough.

The proposed character recognition work has been compared with the previous work in the literature, and the results are summarized in Table 1. It can be observed from the table that the proposed CNN model has provided with higher recognition rate as compared to the other neural network methods and techniques existing in the literature.

4.2 Text-line segmentation on MM dataset

The approach for text-line segmentation is the modification of the existing partial-projection profile technique and calculates the projection histogram of only the first 100 to 200 columns from the left side of the document, and based on that number of lines and transition points are estimated. Generally, the text is written from the left side of the document, and most of the lines in a document are covered in the first 100 to 100 columns. Therefore, no further division of the text document takes place. From the transition points, we get the midpoints between two consecutive lines. The method for text-line segmentation keeps track of the space between lines by calculating the projection profile for i rows above and below of the midpoint to j columns forward. The horizontal projection histogram is divided into three parts based on the fact that a line can proceed straight or either upward or downward. Then, various cases are analyzed to identify the optimal row among the three parts and advances in the region, which has the lowest value of the projection histogram. The process continues until it covers all lines and throughout the column. The whole procedure is explained in detail in the subsequent section, and it is illustrated in Fig. 9.

The first and foremost processing step followed by our algorithm for text-line segmentation after extracting the handwritten document is to find the midpoints between adjacent lines. The procedure for computing the midpoints (mp) is given in Algorithm 1. The first process is to convert the given document into grayscale, and the edge of the whole text document is estimated. But for some text edges are not well defined due to the uneven distribution of stroke by the writers. Therefore, Gamma correction has been performed before edge detection so that the stroke is even and not broken. One such instance is illustrated in Fig. 10. The sudden change in intensity value, or where the gradient is maximum represents an edge. We have used Sobel operator having window size 3 × 3 for this process. The sample edge image is illustrated in Fig. 9c. Further, morphological image operation dilation is performed for adding foreground pixels to the boundary so that the text is filled. To conduct dilation, we have used structural element disk having size three, such that the pixels covered by the structural elements will be changed to 1 if the origin is a hit operation. Similarly, we have also performed erosion operation on the dilated image with the same structural element to preserve the core shape of text if lines are too close or touching, as shown in Fig. 9d. Histogram projection

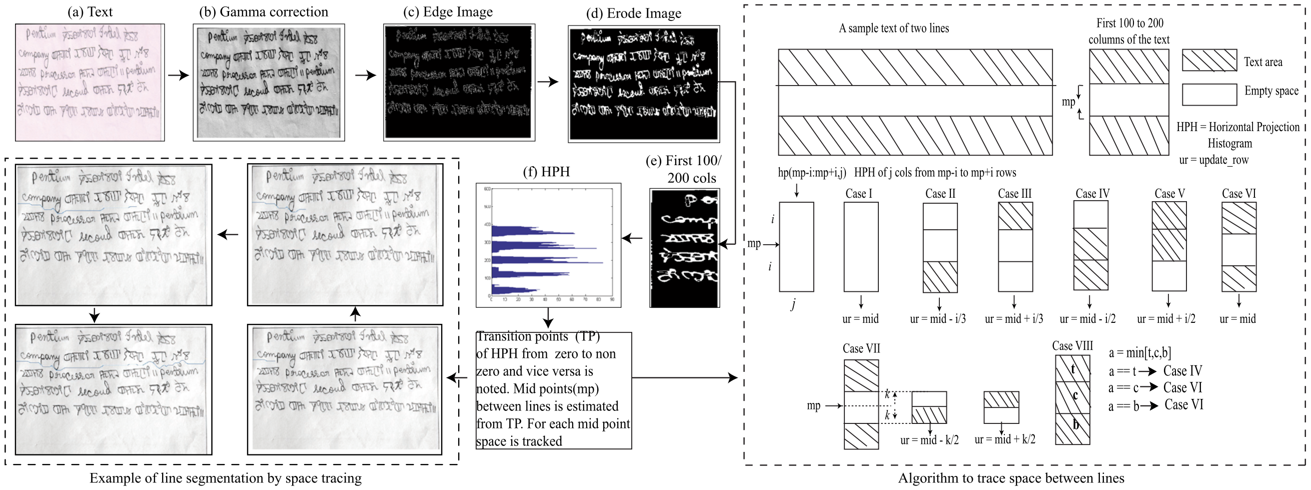

(a) Text
(b) Gamma correction
(c) Edge Image
(d) Erode Image
(e) First 100/ 200 cols
(f) HPH
Transition points (TP) of HPH from zero to non zero and vice versa is noted. Mid points(mp) between lines is estimated from TP. For each mid point space is tracked
Example of line segmentation by space tracing
A sample text of two lines
First 100 to 200 columns of the text
mp
Text area
Empty space
HPH = Horizontal Projection Histogram
ur = update_row
hp(mp-i:mp+i,j) HPH of j cols from mp-i to mp+i rows
i
mp
i
j
Case I
ur = mid
Case II
ur = mid - i/3
Case III
ur = mid + i/3
Case IV
ur = mid - i/2
Case V
ur = mid + i/2
Case VI
ur = mid
Case VII
mp
k
k
ur = mid - k/2
ur = mid + k/2
Case VIII
t
c
b
a = min[t,c,b]
a == t → Case IV
a == c → Case VI
a == b → Case VI
Algorithm to trace space between lines

Fig. 9 The procedure for text-line segmentation

Algorithm 1 $findMidPoint(HWT)$

```
Input: $HWT$: Handwritten text image of size row × col
1: $HWT \longrightarrow gray_{img} \longrightarrow gammaCorrected_{img}(255 * (gray_{img}/255)^{\gamma}) \longrightarrow edge_{img}(Sobel\, operator)$
2: $d = egde_{img} \oplus s \longrightarrow g = d \ominus s,$
3: $stripe_{im}$= g(1:row,1:n) n=first 100 to 200 columns of col
4: $proj_{stripe[j]} = \sum_{j=1}^{row} \sum_{i=1}^{n} if(stripe_{im}(i,j) == 1)$
5: $proj_{stripe}[j] = \frac{1}{3} \sum_{j=j-1}^{j=j+1} proj_{stripe[j]}$
6: if $proj_{stripe[j]} < 10$ then
7:     $proj_{stripe[j]} = 0$
8: end if
9: lines[k] = j if $(proj_{stripe}[j-1] == 0 \cap proj_{stripe}[j] == 0 \cap proj_{stripe}[j+1] > 0) or (proj_{stripe}[j-1] > 0 \cap proj_{stripe}[j] == 0 \cap proj_{stripe}[j+1] == 0)$
10: mp[l] = (lines[r]+lines[r+1])/2 where r ∈ 2,3,…k-1
```

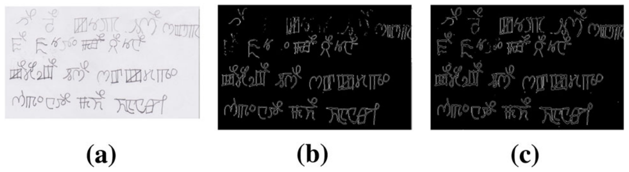


Fig. 10 **a** A text image, **b** edge image without gamma correction, **c** edge image after gamma correction

profile of a stripe of the eroded image (Fig. 9e) has been calculated and smoothened by a simple moving average filter of window size three. These operations can be summarized from statement 1 to 5 of Algorithm 1. To remove continuity in projection histogram, if present, the HPH whose value is less than around 10 have been set to 0, which is depicted in Fig. 11. This procedure takes care of the lines that are too close, and their projection profile coincides so that a possible mid-index between lines can be obtained.

After obtaining the projection histogram of the stripe, we find the points (rows) where the value changes from zero to nonzero and vice versa using statement 9 of Algorithm 1 and stored them in an array, lines [k], where k ranges from 1 to (2 × number of line). As illustrated in Algorithm 2, except for the first and last element of the array (which represent above-border of the first line and below-border of the last line) we took the average of the consecutive elements of array lines[k] and the midpoint (mp) of two adjacent lines are calculated. For each mid-index point, we keep track of the space between lines. Then, starting from n column to a fixed distance of j, the horizontal projection profile is calculated for mp $- i$ to mp $+ i$ rows. For our algorithm, we have taken $j = 20$ and $i = 15$. The projection profile is divided into three parts based on the fact that a line can proceed straight or either upward or downward. Various cases have been considered to find the optimal row to proceed, and then it has advanced on the part, which has the lowest value of the projection profile. The process is repeated until it covers all mp and throughout the column. The cases examine for finding the optimal row for line segmentation is presented in Algorithm 2.

The total number of lines considered for our experiment is 809. There are 197 skews, 114 curved, 145 close, 77 touching lines, and 276 are straight lines. Out of the total, 746 lines are segmented, and 66 are not segmented, giving an accuracy of 91.84%. The performance of the text-line segmentation algorithm on various constraints is summarized in Table 2. It can be realized from the table that the proposed segmentation algorithm significantly segments lines considering various constraints. The order of the constraint in which segmentation accuracy decreases is as follows: normal, skew, curve, close, and touching lines. The proposed algorithm can segment normal and skew lines with more accuracy than the other challenges imposed. A few images are illustrated in Fig. 12 that depicts text-line segmentation on various constraints.

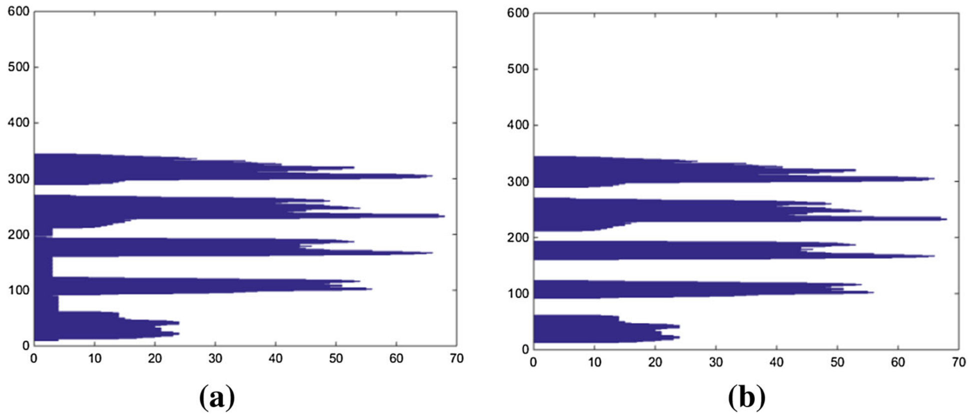


Fig. 11 **a** A Sample continuous horizontal projection histogram, **b** rectified projection histogram

Algorithm 2 *traceSpace*(*mp*[*l*])

```
Input: mp: Mid points between lines
while col1 < col, ∀ mp do
    hp(mp-i:mp+i,j), row=mp, col1=n // HPH of ith rows above and
    below mp for j cols
    top = hp(1:2*i+1/3) ; center = (2*i+1/3)+1:2* (2*i+1/3);
    down= 2*(2*i+1/3)+1 :3*(2*i+1/3)
    t = nnz(top) ; c= nnz(center); d=nnz(down)
    a=min[t,c,b]
    if unique(hp) == 0 then
        update-row=row;
    else if unique(hp(1:2*(2*i+1/3)))==0 then
        update-row= row-i/3;
    else if unique(hp((2*i+1/3):3*(2*i+1/3)))==0 then
        update-row= row+i/3;
    else if unique(top)==0 or a==t then
         update-row=row-i/2;
     else if unique(down)==0 or a==b then
         update-row=row+i/2;
     else if unique(center)==0 or a==c then
         update-row=row;
     else if unique(hp(center))==0 or a==c then
         update-row=row;
     else if unique(hp(mid-5:mid))==0 then
         update-row=row-2;
     else if unique(hp(mid+1:mid+5))==0 then
         update-row=row-2;
     end if
     line([col1,col1+j],[row,update-row])
     col1=col1+j; row=update-row;
end while
```

The normal lines are more or less straight, which have adequate space between consecutive lines. That is why it has achieved the highest accuracy of 96.1% among the other constraints. However, due to a large gap between words of a line, sometimes the separator goes wrongly upward or downward if an extension of a character (from above or below line) is present that is aligned with the gap. The unconstrained handwritten text-lines are generally skewed (either upward or downward) in nature. The procedure for estimating the slopes and assisting the in-between text-lines gap is an important stage in document segmentation. Further, it may also happen that a greater degree of skewness results in touching or overlapping characters between text-line, which ultimately produce inefficient results. However, in this approach, we do not have a separate algorithm for slope detection, rather the gap between consecutive lines are traced block by block. An accuracy of 94.92% is achieved for skew line segmentation. The lesser the gap, the inaccuracy of segmentation has been reported. Therefore, the touching lines have the least accuracy of 84.41%, while close lines have slightly improved the accuracy of 85.51%.

4.3 Word segmentation

In this approach, segmentation of words is not treated as a separate problem after line segmentation. It efficiently goes alongside the line segmentation. As the tracking of the optimal row between neighboring lines goes on using Algorithm 2, we monitor the vertical projection histogram (VPH)

Table 2 Results of segmentation of line having various constraints

No. of lines and constraint	Segment	Under segment	Over segment	Accuracy (in %)
197-skew	187	10	–	94.92
114-curve	102	12	–	89.47
145-close	124	21	–	85.51
77-touching	65	12	–	84.41
276-normal	265	11	1	96.01
809-total	743	66	1	91.84

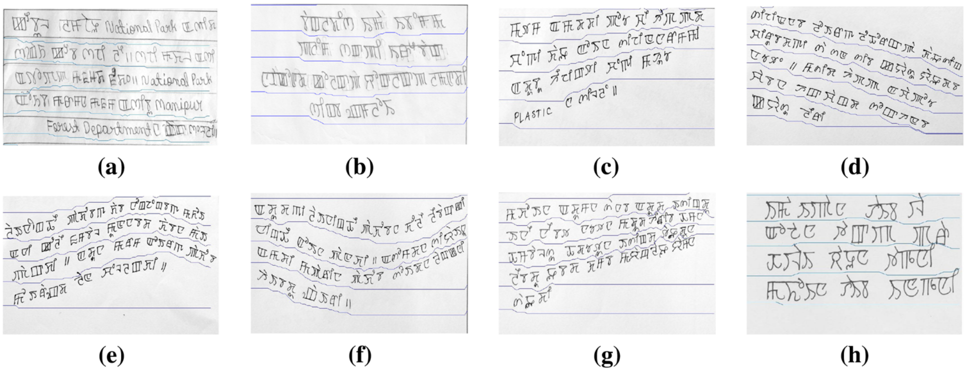


Fig. 12 Few text-line-segmented documents **a**, **b** are straight lines, **c**, **d** are skew lines, **e**, **f** are curve lines, **g**, **h** are close and touching lines

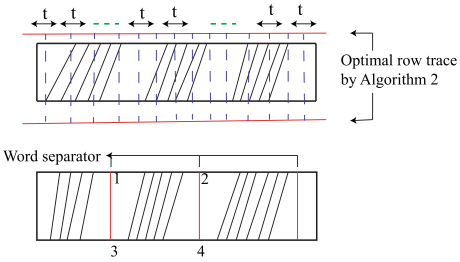


Fig. 13 Process for word segmentation

of t columns between the above and below-border a line as given in Fig. 13. The operation is repeated for the run-length of line segmentation so that every word that exists in a line is detected.

A word is detected by identifying the column when the VPH of t columns are found to be all zeros, i.e., no text is present. If two consecutive *t* columns are found to be all zeros, then only one is considered as only one column is required to separate and detect a word. All the separating points of words, i.e., rows and columns of the beginning and end of each word are identified and obtained. A word is identified by four points, two points above the word, and two below (for example, in Fig. 13, points marked as 1, 2, 3, and 4). These series of points are stored in two arrays, one for above-border and the other for below-border of aline. Then, a word is extracted by using the four points. It may also happen that a column is not detected as in the case of the first word in Fig. 13, but the word is extracted using the information available. If a text document contains almost straight lines, then the rows representing a word do not differ by 20 to 30 levels. If that is the case, then the word can be extracted by taking the respective average of the above and below line rows. It may be noted that the lines are not always straight, as discussed in the previous section. So, just taking average is not enough to give a clear separation of words. For skew lines, we have extended our word extraction algorithm. A sample of a skew word is given with the identification of rows and columns given in Fig. 14.

Algorithm 3 illustrates the extraction of word procedure, for straight line or minimally skew/curve line words are extracted by taking an average of either the above word or below words rows (for instance, row1 and row1' in Fig. 14). Otherwise, we identified the clockwise or counterclockwise skewness or cursiveness, if (floor($\frac{\text{col2}-\text{col1}}{\text{row1}-\text{row1'}}$)< 0) clockwise otherwise counterclockwise.

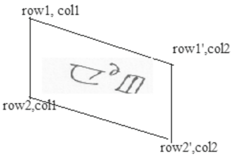


Fig. 14 A sample skew word with rows and columns identification

Algorithm 3 $extractWord(row1, row1', row2, row2', col1, col2)$

```
Assume image as text document under segmentation
if abs(row1-row1')<30 then
    row ← avg(row1,row1') to avg(row2,row2')
    col ← col1 to col2
    eword ← image (row, col)
end if
if abs(row1-row2) > 30 then
    base = row1-row1', perp = col2-col1, curvey = floor($\frac{perp}{base}$)
    if curvey < 0, id = 1 otherwise id = -1
     inc = abs(curvey)
     col ← col1 to col2
     row ← row1+inc to row2+inc, for every inc iteration, row is added with id
     eword ← image (row, col)
end if
if (nnz(∼ eword)∼ = 0) then
     extracted word is kept otheriwse discarded
end if
```

$$\text{base} = \text{abs(row1-row1')}$$

$$\text{perp} = \text{abs(col1-col2)}$$

For, clockwise skew of lines, the row is incremented by 1 for every floor($\frac{\text{perp}}{\text{base}}$). Similarly, for counterclockwise skew of lines, the row is decremented by 1 for every floor($\frac{\text{perp}}{\text{base}}$), to perform the word extraction operation. False detection of blank space as a word is discarded by the word extraction algorithm.

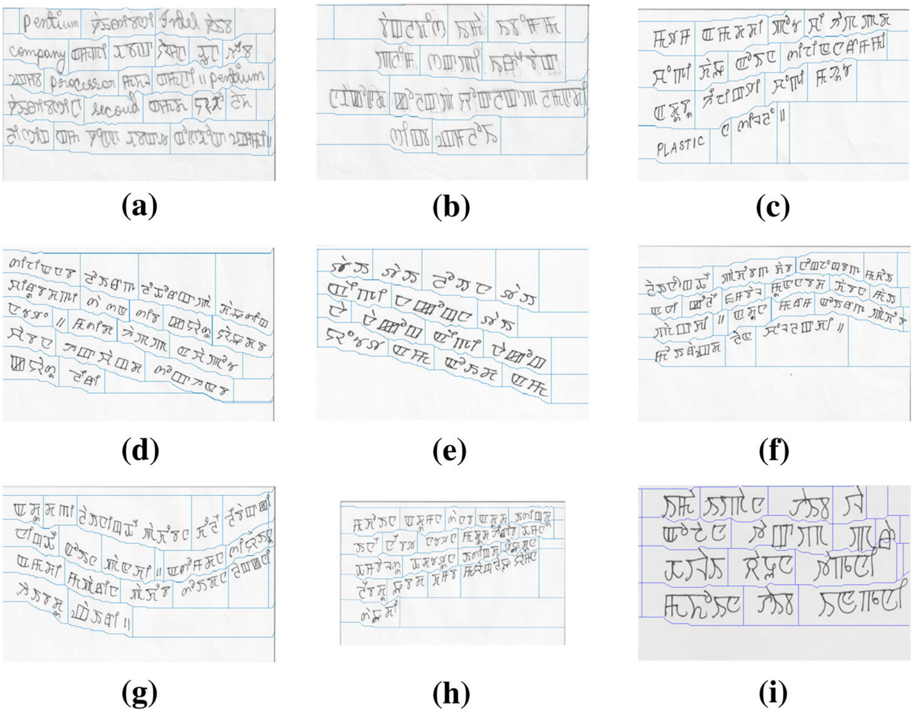

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 15 Few word segmented documents **a**, **b** are straight lines, **c**, **d**, **e** are skew lines, **f**, **g** are curve lines, **h**, **i** are close and touching lines

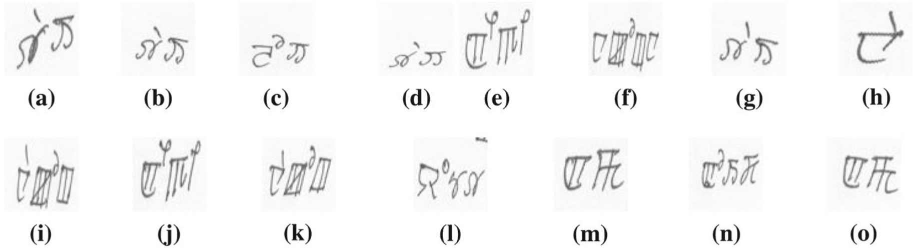

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m) (n) (o)

Fig. 16 Segmented words of document image in Fig. 15e

The total number of words considered for word segmentation is 2917, and segmented words among them are 2595 giving an accuracy of 88.96%. The post-processing method needs to be performed to eliminate the extra space around the word and additional text from neighboring lines or words. Figure 15 illustrates the results of word segmentation, and Fig. 16 depicts the segmented words of a sample document.

The words that appear in a text document are extracted using the word extraction algorithm and save as individual images for further processing of character segmentation. The words in an accurately segmented normal text-line documents are extracted with no loss of characters. The documents which are not largely skewed have also been extracted correctly. However, those documents in which skewness is very high, then the few characters of words are lost, or irrelevant character is added.

Algorithm 4 $segmentCharacter(word)$

```
Input: $wim$: word image
$bin_{im} \longrightarrow binarized(wim)$
$com_{bin} \longrightarrow complement(bin_{im})$
ncc = no. of connected component(cc) of $bin_{im}$
$i = 1$
for i<ncc do
    pixels of $i$th cc is turned to white and $(i-1)^{t}h$ to black
    a $\longrightarrow (bin_{im}(cc(PixelIdList(i)) = 1))$
    $(bin_{im}(cc(PixelIdList(i-1)) = 0))(i > 1)$
    $sub_{im} = com_{bin} - a'$ // where a' = complement(a)
    $sub_{im}$ contain the $i$th character
    Minimum bounding rectangle property is applied on $sub_{im}$ and
    statistics are used to crop the character from $wim$ image
end for
```

4.4 Character segmentation

Segmentation of documents to the identifiable element is a necessary step for building a recognition system. In this section, character segmentation from word images using connected component analysis is explained in detailed by Algorithm 4. In the first step, a word image is taken, and its binary counterpart is computed. Further, the complement of the binary image is estimated, and no. of the connected component is calculated. Then, all the pixels identified by every connected component are turned to background pix-

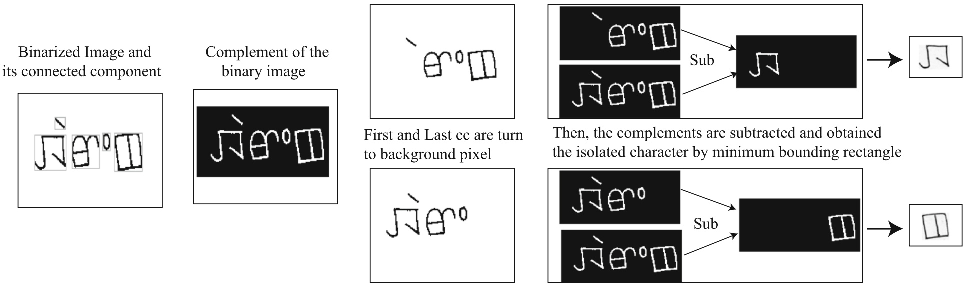

Binarized Image and its connected component
Complement of the binary image
First and Last cc are turn to background pixel
Then, the complements are subtracted and obtained the isolated character by minimum bounding rectangle
Sub
Sub

Fig. 17 The process for character segmentation

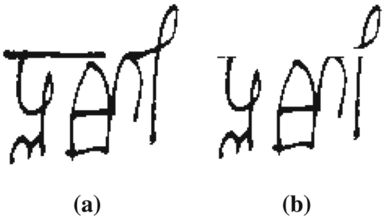


Fig. 18 **a** A word image with headline, **b** same image after headline removed

els, and its complement is subtracted from the complement of the binary image. Finally, the minimum bounding rectangle of the subtracted image is obtained, and the character is segmented. The whole procedure is depicted in Fig. 17.

The above algorithm work for words without a headline. Generally, Meitei Mayek is written without a headline, but at the time of database acquisition, some writer writes sentences with headlines. For those words, the above algorithm fails and has to be dealt with a different approach. First, the headline is removed so that each character in a word is isolated, and they can be detected and separated by a vertical projection histogram. The headline is removed by finding the maximum horizontal projection profile row within the half size of the image and deleting it (converting to background pixel). Then, the characters are separated by a vertical projection histogram, as shown in Fig. 18.

For the segmentation algorithm, we have considered 4932 characters and out of which 4494 have been isolated, giving an accuracy of 91.12%. After segmenting words using Algorithm 4, few of them are found to have characters that are joined due to the presence of headline; they are further processed by removing headline and isolating by vertical projection histogram. Still, the segmentation algorithm fails to segment some touching characters and those attached with symbols.

5 Conclusions and future work

The work presented in this paper addresses two problems: firstly, developed Meitei Mayek databases (Mayek27 and MM datasets) for research work and secondly, performing experimentation on the acquired dataset. We have conducted character recognition and segmentation (line, word, and character) on the collected databases to validate the proposed databases. The Mayek27 dataset has been compiled to form a basis for the recognition model on Meitei Mayek. The MM dataset has been developed because there exist English words whose Meitei Mayek equivalent does not avail.

Character recognition and text-line segmentation have been performed on the large and challenging dataset that we have developed called Mayek27 and MM database, respectively. Character recognition is performed using CNN with three convolutional layers employing with 3 × 3 filter size on the images. The analysis and extensive experimental study reveal the efficiency and effectiveness of this method for character recognition. The comparative evaluation for character recognition suggests that this work has accomplished superior performance than the existing methods in the literature.

On the MM dataset, text-line, word, and character segmentation have been performed, and various constraints have been considered in our experiment. The proposed text-line segmentation can separate skew, curved, and touching text-lines. Lines in the document could not be detected if it has began written moving toward the center instead of the standard left side of the text. But, the overall percentage of accurate segmentation of the text-line of our proposed method is 91.84%. Further, the accuracy obtained in word and character segmentation is 88.96% and 91.12%, respectively, .

Compliance with ethical standards

Conflict of interest All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for relevant intellectual content; and (c) approval of the final version. The stated authors have written the article, and the work is original which is not published elsewhere. This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue. The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript. The below authors have affiliations with organizations with a direct or indirect financial interest in the subject matter discussed in the manuscript: Sanasam Inunganbi (National Institute of Technology, Manipur) Prakash Choudhary (National Institute of Technology, Hamirpur) Khumanthem Manglem Singh (National Institute of Technology, Manipur)

References

1. Wang, D.-H., Liu, C.-L., Zhou, X.-D.: An approach for real-time recognition of online Chinese handwritten sentences. Pattern Recognit. **45**(10), 3661–3675 (2012)
2. Wang, Q.-F., Yin, F., Liu, C.-L.: Handwritten Chinese text recognition by integrating multiple contexts. IEEE Trans. Pattern Anal. Mach. Intell. **34**(8), 1469–1481 (2011)
3. Zhou, X.-D., Wang, D.-H., Tian, F., Liu, C.-L., Nakagawa, M.: Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields. IEEE Trans. Pattern Anal. Mach. Intell. **35**(10), 2413–2426 (2013)
4. Zhou, X.-D., Zhang, Y.-M., Tian, F., Wang, H.-A., Liu, C.-L.: Minimum-risk training for semi-Markov conditional random fields with application to handwritten Chinese/Japanese text recognition. Pattern Recognit. **47**(5), 1904–1916 (2014)
5. Wu, Y.-C., Yin, F., Liu, C.-L.: Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models. Pattern Recognit. **65**, 251–264 (2017)
6. Laishram, R., Singh, A.U., Singh, N.C., Singh, A.S., James, H.: Simulation and modeling of handwritten Meitei Mayek digits using neural network approach. In: Proceedings from the International Conference Advances in Electronics, Electrical and Computer Science Engineering-EEC, pp. 355–358 (2012)
7. Laishram, R., Singh, P.B., Singh, T.S.D., Anilkumar, S., Singh, A.U.: A neural network based handwritten Meitei Mayek alphabet optical character recognition system. In: 2014 IEEE International Conference on Computational Intelligence and Computing Research, pp. 1–5. IEEE (2014)
8. Surinta, O., Schomaker, L., Wiering, M.: A comparison of feature and pixel-based methods for recognizing handwritten bangla digits. In: 2013 12th International Conference on Document Analysis and Recognition. IEEE (2013)
9. Hassan, T., Khan, H.A.: Handwritten bangla numeral recognition using local binary pattern. In: 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), pp. 1–4. IEEE (2015)
10. Ilmi, N., Budi, W.T.A., Nur, R.K.: Handwriting digit recognition using local binary pattern variance and K-nearest neighbor classification. In: 2016 4th International Conference on Information and Communication Technology (ICoICT), pp. 1–5. IEEE (2016)
11. Inunganbi, S., Seal, A., Khanna, P.: Classification of food images through interactive image segmentation. In: Asian Conference on Intelligent Information and Database Systems, pp. 519–528. Springer, Cham (2018)
12. Kabbai, L., Abdellaoui, M., Douik, A.: Image classification by combining local and global features. Vis. Comput. **35**(5), 679–693 (2019)
13. Ghosh, S., Barman, U., Bora, P.K., Singh, T.H., Chaudhuri, B.B.: An OCR system for the Meetei Mayek script. In: 2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), pp. 1–4. IEEE (2013)
14. Kumar, C.J., Kalita, S.K.: Recognition of handwritten numerals of Manipuri script. Int. J. Comput. Appl. **84**(17), 1–5 (2013)
15. Hammouda, G., Sellami, D., Hammouda, A.: Pattern recognition based on compound complex shape-invariant Radon transform. Vis. Comput. (2018). https://doi.org/10.1007/s00371-018-1604-9
16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
17. Lim, J., Lee, K.: 3D object recognition using scale-invariant features. Vis. Comput. **35**(1), 71–84 (2019)
18. Mohammadi, S., Maleki, R.: Air-writing recognition system for Persian numbers with a novel classifier. Vis. Comput. (2019). https://doi.org/10.1007/s00371-019-01717-3
19. Mahmoud, S.A.: Arabic character recognition using Fourier descriptors and character contour encoding. Pattern Recognit. **27**(6), 815–824 (1994)
20. Shridhar, M., Badreldin, A.: High accuracy character recognition algorithm using Fourier and topological descriptors. Pattern Recognit. **17**(5), 515–524 (1984)
21. Mowlaei, A., Faez, K., Haghighat, A.T.: Feature extraction with wavelet transform for recognition of isolated handwritten Farsi/Arabic characters and numerals. In: 2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No. 02TH8628), vol. 2, pp. 923–926. IEEE (2002)
22. Liu, C.-L., Koga, M., Fujisawa, H.: Gabor feature extraction for character recognition: comparison with gradient feature. In: Eighth International Conference on Document Analysis and Recognition (ICDAR'05), pp. 121–125. IEEE (2005)
23. Maring, K.A., Dhir, R.: Recognition of cheising iyek/eeyek-Manipuri digits using support vector machines. Ijcsit **1**(2) (2014)
24. Stockwell, R.G., Mansinha, L., Lowe, R.P.: Localization of the complex spectrum: the S transform. IEEE Trans. Signal Process. **44**(4), 998–1001 (1996)
25. Mansinha, L., Stockwell, R.G., Lowe, R.P.: Pattern analysis with two-dimensional spectral localisation: applications of two-dimensional S transforms. Phys. A Stat. Mech. Appl. **239**(1–3), 286–295 (1997)
26. Badrinath, G.S., Gupta, P.: Stockwell transform based palm-print recognition. Appl. Soft Comput. **11**(7), 4267–4281 (2011)
27. Dash, K.S., Puhan, N.B., Panda, G.: Handwritten numeral recognition using non-redundant Stockwell transform and bio-inspired optimal zoning. IET Image Process. **9**(10), 874–882 (2015)
28. Drabycz, S., Stockwell, R.G., Mitchell, J.R.: Image texture characterization using the discrete orthonormal S-transform. J. Digit. Imaging **22**(6), 696 (2009)
29. Bianne-Bernard, A.-L., Menasri, F., Mohamad, R.A.-H., Mokbel, C., Kermorvant, C., Likforman-Sulem, L.: Dynamic and contextual information in HMM modeling for handwritten word recognition. IEEE Trans. Pattern Anal. Mach. Intell. **33**(10), 2066–2080 (2011)
30. Thokchom, T., Bansal, P.K., Vig, R., Bawa, S.: Recognition of handwritten character of manipuri script. JCP **5**(10), 1570–1574 (2010)
31. Pirlo, G., Impedovo, D.: Fuzzy-zoning-based classification for handwritten characters. IEEE Trans. Fuzzy Syst. **19**(4), 780–785 (2011)
32. Pirlo, G., Impedovo, D.: Adaptive membership functions for handwritten character recognition by voronoi-based image zoning. IEEE Trans. Image Process. **21**(9), 3827–3837 (2012)
33. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015)
34. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
35. Fukushima, K.: Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. **36**(4), 193–202 (1980)
36. LeCun, Y., et al.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
37. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **25**, 1106–1114 (2012)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
39. Kahan, S., Pavlidis, T., Baird, H.S.: On the recognition of printed characters of any font and size. IEEE Trans. Pattern Anal. Mach. Intell. **2**, 274–288 (1987)

40. dos Santos, R.P., Clemente, G.S., Ren, T.I., Cavalcanti, G.D.C.: Text line segmentation based on morphology and histogram projection. In: 2009 10th International Conference on Document Analysis and Recognition, pp. 651–655. IEEE (2009)
41. Zahour, A., Taconet, B., Mercy, P., Ramdane, S.: Arabic handwritten text-line extraction. In: Proceedings of Sixth International Conference on Document Analysis and Recognition, pp. 281–285. IEEE (2001)
42. Nguyen, K.C., Nakagawa, M.: Text-line and character segmentation for offline recognition of handwritten japanese text. IEICE Tech. Rep. **115**(517), 53–58 (2016)
43. Zahour, A., Likforman-Sulem, L., Boussellaa, W., Taconet, B.: Text line segmentation of historical arabic documents. In: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), vol. 1, pp. 138–142. IEEE (2007)
44. Nagy, G., Seth, S., Viswanathan, M.: A prototype document image analysis system for technical journals. Computer **25**(7), 10–22 (1992)
45. He, J., Downton, A.C.: User-assisted archive document image analysis for digital library construction. In: Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings, pp. 498–502. IEEE (2003)
46. Pal, U., Datta, S.: Segmentation of Bangla unconstrained handwritten text. In: Null, p. 1128. IEEE (2003)
47. Arivazhagan, M., Srinivasan, H., Srihari, S.: A statistical approach to line segmentation in handwritten documents. In: Document Recognition and Retrieval XIV, vol. 6500, p. 65000T. International Society for Optics and Photonics (2007)
48. Su, T.-H., Zhang, T.-W., Huang, H.-J., Zhou, Y.: Skew detection for Chinese handwriting by horizontal stroke histogram. In: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), vol. 2, pp. 899–903. IEEE (2007)
49. Weliwitage, C., Harvey, A.L., Jennings, A.B.: Handwritten document offline text line segmentation. In: Digital Image Computing: Techniques and Applications (DICTA'05), pp. 27–27. IEEE (2005)
50. O'Gorman, L.: The document spectrum for page layout analysis. IEEE Trans. Pattern Anal. Mach. Intell. **15**(11), 1162–1173 (1993)
51. Kise, K., Sato, A., Iwata, M.: Segmentation of page images using the area Voronoi diagram. Comput. Vis. Image Underst. **70**(3), 370–382 (1998)
52. Marti, U.-V., Bunke, H.: Text line segmentation and word recognition in a system for general writer independent handwriting recognition. In: Proceedings of Sixth International Conference on Document Analysis and Recognition, pp. 159–163. IEEE (2001)
53. 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)
54. Jindal, P., Jindal, B.: Line and word segmentation of handwritten text documents written in Gurmukhi script using mid point detection technique. In: 2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS), pp. 1–6. IEEE (2015)
55. Saha, S., Basu, S., Nasipuri, M., Basu, D.K.: A Hough transform based technique for text segmentation. arXiv preprint arXiv:1002.4048 (2010)
56. Louloudis, G., Gatos, B., Pratikakis, I., Halatsis, C.: Text line and word segmentation of handwritten documents. Pattern Recognit. **42**(12), 3169–3183 (2009)
57. Ryu, J., Koo, H.I., Cho, N.I.: Word segmentation method for handwritten documents based on structured learning. IEEE Signal Process. Lett. **22**(8), 1161–1165 (2015)
58. Javed, M., Nagabhushan, P., Chaudhuri, B.B.: Extraction of line-word-character segments directly from run-length compressed printed text-documents. In: 2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), pp. 1–4. IEEE (2013)
59. Inunganbi, S., Choudhary, P.: Recognition of handwritten Meitei Mayek script based on texture feature. Int. J. Natural Lang. Comput. (IJNLC) **7**(5), 99–108 (2018)
60. Inunganbi, S.C., Choudhary, P.: Recognition of handwritten Meitei Mayek and English alphabets using combination of spatial features. In: International Conference on Intelligent Systems Design and Applications. Springer, Cham (2018)
61. Kumar, C.J., Kalita, S.K., Sharma, U.: Recognition of Meetei Mayek characters using hybrid feature generated from distance profile and background directional distribution with support vector machine classifier. In: 2015 Communication, Control and Intelligent Systems (CCIS), pp. 186–189. IEEE (2015)

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

Sanasam Inunganbi is currently pursuing Ph.D in Computer Science and Engineering Department of NIT Manipur. She obtained her M. Tech degree from PDPM IIITDM Jabalpur after completing DOEACC B Level from NIELIT, Delhi. Her area of interest includes, Pattern Recognition, Image Processing, handwritten character recognition.

Prakash Choudhary obtained his Ph.D. in Computer Science and Engineering from MNIT Jaipur. He received his M.Tech. in Computer Science and Engineering from VNIT Nagpur and B.E. in Information Technology from Rajasthan University. He is currently working as Assistant Professor in NIT Hamirpur. His area of research interests are Biomedical Image analysis, Pattern Recognition, Annotation and retrieval of Images.

Khumanthem Manglem is currently working as a Professor in NIT Manipur. He received his Ph.D. from Indian Institute of Technology, Guwahati (IITG), Assam in 2007. He completed his MS from BITS, Pilani Rajasthan in Software System. His area of interest includes digital image and video processing, digital watermarking and information hiding, Hashing Function, Cyber Forensics.