Electronic Letters on Computer Vision and Image Analysis 20(1):55-69; 2021

Recognition of Devanagari Scene Text Using Autoencoder CNN

S. S. Shiravale* R. Jayadevan+ and S. S. Sannakki++

* *Department of Computer Engineering, MMCOE, Pune, India*
+ *Department of Computer Engineering, AIT, Pune, India*
++ *Department of Computer Science and Engineering, G.I.T., Belagavi, India.*

Received 19th of September 2020; accepted 13th of January 2021

Abstract

Scene text recognition is a well-rooted research domain covering a diverse application area. Recognition of scene text is challenging due to the complex nature of scene images. Various structural characteristics of the script also influence the recognition process. Text and background segmentation is a mandatory step in the scene text recognition process. A text recognition system produces the most accurate results if the structural and contextual information is preserved by the segmentation technique. Therefore, an attempt is made here to develop a robust foreground/background segmentation(separation) technique that produces the highest recognition results. A ground-truth dataset containing Devanagari scene text images is prepared for the experimentation. An encoder-decoder convolutional neural network model is used for text/background segmentation. The model is trained with Devanagari scene text images for pixel-wise classification of text and background. The segmented text is then recognized using an existing OCR engine (Tesseract). The word and character-level recognition rates are computed and compared with other existing segmentation techniques to establish the effectiveness of the proposed technique.

Key Words: scene text recognition; Devanagari script; OCR; segmentation technique; encoder-decoder CNN

1 Introduction

Text recognition systems are becoming more efficient due to the increasing availability of multimedia data, low-cost image capturing devices, and high-performance computing devices. Understanding the text present in a scene image like nameplates, instructional boards, navigation boards, banners, wall paintings, etc. is essential for effective communication. But, understanding the text written in an unknown language or script is a massive challenge in scene text recognition. A solution can be provided by developing a smartphone-based system that can process, detect, recognise and translate the text present in a scene image from one language to a known language. Detection and recognition of the text are the two major steps in such applications. The process of localization and extraction of text regions from the image is called text detection. The output of the text detection is always in the image format. The process of converting that text image into the corresponding digital format (Unicode) is called text recognition. This paper presents a technique to recognize Devanagari text from natural scene images. In the last few decades, various methods have been reported regarding the recognition of text present on document images. But scene text recognition is still a challenging task compared to the document image recognition [1, 2]. Foreground and background

Correspondence to: sankirtishiravale@mmcoe.edu.in

Recommended for acceptance by Angel D. Sappa
https://doi.org/10.5565/rev/elcvia.1344
ELCVIA ISSN: 1577-5097
Published by Computer Vision Center / Universitat Autonoma de Barcelona, Barcelona, Spain

segmentation (separation) is the first step in the scene text recognition process. It becomes critical due to complex background, uneven lighting, shadow effects, climatic conditions etc. Some challenges in scene text recognition along with some challenges specific to the Devanagari script, are shown in Fig. 1.

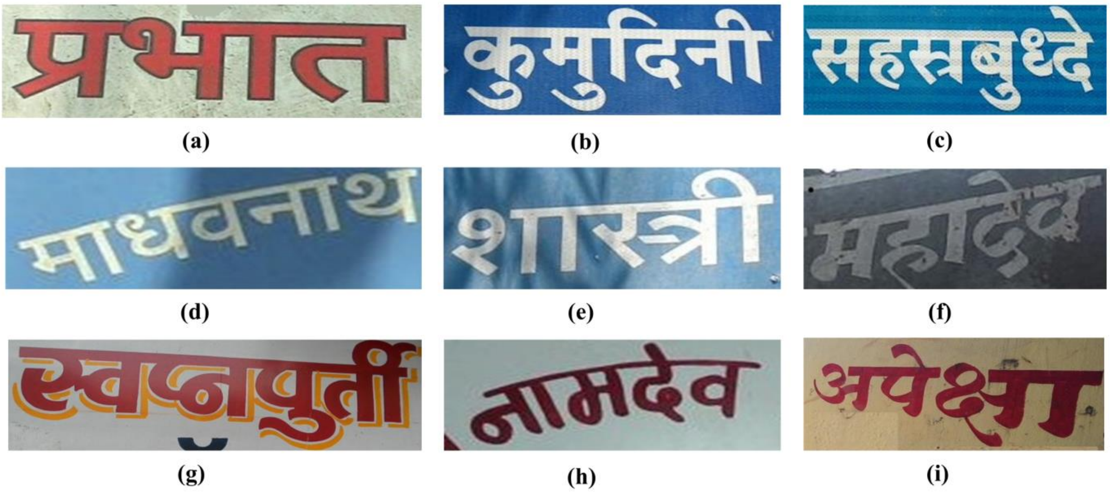

Figure 1. Various challenges in Devanagari scene text recognition (a) Simple text with rough background (b) Text with modifiers (c) Text with conjunct (d) Uneven illumination (e) Shadow effect (f) Poor quality text (g) Artistic text (h) Curved text (i) Perspective distortion.

Devanagari script is one of the popular Indic scripts and the languages like Nepali, Hindi, Marathi, Konkani etc. are written (documented) using it. Hindi is the national language of India and is extensively used for writing official documents, instructional boards, wall paintings, banners, hoardings etc. Devanagari script has a distinctive feature called as 'shirorekha' (Header-line running across the word). Sometimes header-line connects all the alphabets to form a word (refer Fig.1 (b)), or it may join few alphabets (refer Fig.1 (d)). The script has an exclusive set of consonants, vowels and modifiers (e.g. matras). Modifiers are associated with the consonants and are placed at the upper or lower part by following specific composition rules, as shown in Fig.1 (b). The script also contains a few special characters and conjuncts. Conjunct Characters are formed by connecting half part of one alphabet to any other alphabet (refer to Fig.1 (c)). Detailed information about the Devanagari script is presented in [3]. Devanagari text recognition is a challenging task due to the presence of modifiers and conjunct characters. An efficient character segmentation technique that preserves the shape and contextual information is required to recognize text written in the Devanagari script accurately.

The challenges mentioned in Fig.1 depict the importance of text (foreground) and background separation. Another computational challenge in pre-processing/segmentation is the colour of text and background. The image with lighter text and darker background produces binary segmentation results with text in white and background in black colour. For the opposite scenario, it produces black text with white background. It is difficult to apply a common pre-processing step that covers both scenarios. For example, in the character segmentation process, the detection and removal of header-lines based on horizontal projection profiles (i.e., counting the number of black or white pixels in the horizontal direction) are problematic. Thus, a competent segmentation technique is required to tackle all these challenges, and it should also be capable of converting the segmentation results of various scene images into a uniform format. This paper presents an efficient segmentation technique that addresses these problems and results in better recognition rate.

The flow of the proposed work and the organization of the paper is shown in Fig. 2. A camera captured Devanagari scene text image is given as an input to the system. The input image is then segmented into text (foreground) and background for better recognition results. Foreground/background separation (segmentation) based on a convolution neural network (CNN) is proposed in section 3. Experimental results and comparative analysis are presented in section 4. The segmented text is then recognized by an existing OCR engine. The proposed foreground separation technique is evaluated using performance measures like word recognition rate and character recognition rate. The recognition results are mentioned in section 5.

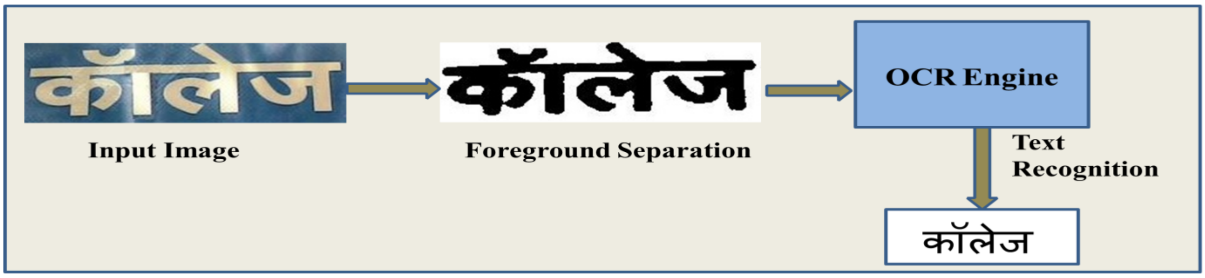

Figure 2. The flow of proposed work

2 Related Work

Scene text recognition process is challenging, and the contribution to research in this area is really encouraging. The benchmarking datasets for Latin script (English language) are available and hence great research is being done in this area. The process of text recognition is script dependant and it is difficult to apply the same method to other scripts to evaluate the recognition results on different scripts. A survey of recent developments in other scripts and existing Devanagari text recognition methods are covered in this section. Different scene text recognition methods are explored to reach out to a new method by which the Devanagari scene text recognition shall be successfully completed.

2.1 Review of scene text recognition methods

A paradigm of scene text recognition is shifted from segmentation-based to segmentation-free approach (also known as holistic word recognition). The conventional text recognition process relies on a segmentation-based approach where the word is segmented into individual characters [4]. Characters are then recognized from left to right using OCR systems and words are predicted. The work entirely relies on character classification and grouping of recognized characters to form words. Nowadays, a holistic word recognition guided by supervised algorithms is used by researchers. Instead of extracting features per character, the ordered sequence of features is extracted to train the classifiers. Emerging memory-efficient deep neural network architectures are capable of storing contextual information and process the sequence of features more efficiently. In [5], CNN based encoder/decoder architecture is proposed to extract and recognize the ordered feature sequence. Text is recognized by the Bidirectional long short-term memory (Bi-LSTM) network. The basic idea of an encoder is to extract the character-wise features by a set of convolutional layers and map it to a set of ordered features using any memory-efficient algorithms. The decoder recognizes sequences of character with the help of an additional attention mechanism [6]. Prediction of words is more accurate due to the consideration of bidirectional features. The similar concept of encoder/decoder is enhanced further for the perspective distortion and curve text line correction in [7].

The holistic word recognition minimizes the necessity of post-processing, i.e. grouping of characters for word construction. The different models like HMM [8], convolution neural network (CNN) [9] and recurrent neural networks (RNN) [10] have been proven efficient for holistic word recognition. HMM has a special capability to explore the contextual information derived from the spatial description that helps the word recognition process. In [11], three different CNN models are designed for word recognition. The first layer is proposed for word classification using a pre-defined language knowledge dictionary. The second layer is trained for sequence prediction and the third layer constructs words as a bag of n_grams. The synthetic data is generated for the training. The technique is simple and cost-effective. The variants of RNN such as LSTM and BLSTM [5, 6, 7, 10] are memory efficient and capable to store contextual information for a longer duration. In any word recognition problem, the contextual information of the previous and next character is equally important and hence BLSTM architectures are gaining more popularity. Highly accurate results and the ability to retrieve contextual information of RNN architectures are attaining more success compared to the HMM model in word recognition problems [12]. CNN or RNN models may suffer from the problem of overfitting in case if an inadequate amount of data trains models. The problem of overfitting can be avoided by generating synthetic data for the training [13, 11, 14]. Intelligent Algorithms, synthetic data generation and high computing resources facilitate holistic word recognition.

Various text recognition methods are compared in Table 1 using word recognition rate (WRR) and character recognition rate (CRR). Different benchmarking datasets for English scene text recognition are available for experimentation. Methods evaluated using ICDAR 2013 dataset of scene words are only considered and results are mentioned. In some cases, evaluation parameters (i.e. WRR/CRR) are not clearly mentioned; in such cases, recognition accuracy is considered CRR. It is observed that great success is achieved in scene text recognition in English and some Asian languages (e.g., Chinese). Script specific constraints, lack of benchmarking datasets and evaluation parameters are the major causes of less attainment in research outcomes for scene text recognition in Indic scripts [15].

Techniques	Script	Features	WRR	CRR
Minesh et.al [36], 2017	Hindi	Hybrid approach (CNN-RNN)	42.9	75.6
	Telugu		57.2	86.2
	Malayalam		73.4	92.8
Bhunia et.al [39], 2018	Devanagari	PHOG, HMM	72.87	-
	English		82.31	90.87
Jain et.al [14], 2017	Arabic	Hybrid approach (CNN-RNN)	39.43	75.05
Zuo et.al [5], 2019	English	CNN, CTC, BLSTM, encoder/decoder	-	91
B. Shi et.al [6], 2016	English	Automatic Rectification Network	-	88.6
F. Zhan and S. Lu [7], 2019	English	Iterative Rectification Network	-	91.3
Su B. and Lu S. [10], 2015	English	HOG, RNN-LSTM	-	84
Jaderberg et. Al [11], 2014	English	CNN	-	90.8
S. Tian et.al [26], 2016	Chinese	COHOG, CNN	-	71.2
	Bengali		-	92.2
X. Ren et.al [13], 2017	Chinese	Stroke features, TSCD, CNN	38	-
		ABBYY OCR	40	-

Table 1. Experimental results of some scene text recognition methods.

2.2 Review of Devanagari document text recognition methods

Structural characteristics of Devanagari script make the recognition process more complicated. Even though adequate research has been reported on printed and handwritten Devanagari text recognition, only a few works have been reported on scene text detection or recognition. The traditional Devanagari text recognition process relies on a segmentation-based approach in which the Devanagari word is segmented into characters and character recognition is carried out using either existing OCR engines or building a new OCR system. Survey of different OCR systems, various features and classifiers used for Devanagari and other Indic scripts recognition are mentioned in [3, 16- 19]. A common way of character segmentation is to subdivide the Devanagari word into different zones like upper zone with upper modifiers, middle zone with alphabets and lower zone with lower modifiers [20-24]. The simplest and popular technique to achieve character segmentation is detecting and removing header-line or horizontal/vertical projections (i.e. the number of black/white pixels). Horizontal and vertical projections are script independent and frequently used for segmenting due to its simplicity. In [25], the concept of the water reservoir is proposed for character segmentation. Features such as structural, contour, topological, templates [18], histogram of gradients [28, 26, 27], water reservoirs [20, 25, 29], writing strokes [22, 30] are the most frequently used features for character recognition. A different approach is proposed in [31] where a stroke feature of segmented character and the whole word is derived and results are compared. It has experimentally proven that character-level features produce highly accurate results than word-level features due to its capacity to capture more detailed information. The training of classifiers for the recognition purpose is followed after the feature extraction process. Smart and intelligent machine learning algorithms such as K-NN [17, 23], Decision Tree, SVM [20, 27, 47] are the commonly used classifiers for character recognition. Apart from these classifiers, architecturally advanced classifiers like deep neural networks such as CNN and RNN [32, 33, 48, 49] are extensively used due to their robustness and accuracy. In [50], a novel capsule CNN architecture is proposed for unconstrained recognition of handwritten Devanagari Numerals. The additional capsule layer ensures spatial relationships and helps to enhance the precision of recognition.

The main advantage of the segmentation-based approach is to minimize the number of recognition classes. On the other side, most recognition errors occur due to incorrect segmentation [19]. Inappropriate segmentation may distort structural shapes, which ultimately hampers the overall recognition result [34]. Segmentation of conjunct characters is again a challenging task compared to simple characters with modifiers [35]. The overall process of Devanagari word segmentation is precarious and the complexity of segmentation increases for the scene text. The criticality in segmentation driven processing is one reason that affects the overall attainment in Devanagari scene text recognition. The segmentation-free Devanagari text recognition process is comparatively simpler than the segmentation-based approach. The promising outcome in the holistic word recognition process in other scripts such as English attracts researchers to incorporate this methodology in the domain of Indic script recognition. Hence remarkable improvement is observed in scene text recognition for other scripts. In [36], synthetic data of Devanagari scene text is generated for the training of CNN, where a hybrid model of CNN-RNN is proposed for the recognition of the whole word. The segmentation-free approach is computationally simple and less error-prone for scene text recognition [36] due to the omission of the character segmentation step.

Foreground/background separation is an essential step in segmentation-based and segmentation-free approaches. The accuracy of scene text recognition is directly influenced by text/background segmentation results. For example, noisy segmentation may lead to incorrect character segmentation/recognition. Binarization of natural scene images is a complex task due to the challenges like complex background, shadows, uneven lighting, etc. As conventional binarization methods fail to accommodate all these challenges, new methodologies need to be designed. Roy et al. [37] have proposed a technique that combines wavelet and gradient features for video text binarization. In [38], Bayesian classifier is used for the text and background separation for English video text. The prior probability of each pixel estimates the text as well as background information. Though the method is computationally simple, character recognition rate (mentioned 39%) is poor. Almost all the techniques apply binarization as a pre-processing step. A different approach is found in [39], Pyramidal Histogram of Oriented Gradient features are extracted from different colour channels. The text is recognized by HMM using extracted features. The received results are encouraging, but additional computation is essential for the selection of the proper colour channel.

To summarize, the overall process of developing an OCR system follows general steps like pre-processing, character segmentation, feature extraction, training of the classifier, character recognition and word formation [4]. Building an OCR engine is a very complicated and computationally expensive task. Hence many researchers have preferred existing OCR engines like Tesseract or ABBYY for experimentation/recognition purposes [40]. Numerous researchers have developed their own OCR systems by training their model with their dataset. In such cases, the results may decline for a new dataset. The accuracy of any OCR system is subject to vary for a variety of font sizes, and styles used for the training. Considering the complexity of natural scene images and adaptability towards generalising recognition, this paper proposes recognition based on Tesseract OCR engine. Many OCR engines produce better results for black and white images (text in black and background in white); hence designing a robust segmentation/binarization technique is the primary task while acquiring functionality of existing OCR. Most of the work found in the literature survey is on the English language. Devanagari text recognition is more sensitive to noise and false edges as they may recognize as a modifier (e.g. anuswar). Thus, there is a necessity for a robust text/background segmentation technique to achieve the highest accurate recognition results for the Devanagari text.

3 CNN for Foreground Segmentation

The researchers are delighted to analyse and investigate the domain of deep neural networks (DNNs) due to its self-learning ability and great success in the classification and recognition domain. Incredible architectural evolution in DNN and its high attainment for solving complex problems have been observed in this decade. Encoder-decoder, a newly evolved deep neural network architecture, is being widely used in the domain of sequence detection (speech recognition, NLP problems etc.)[41], image segmentation [42] and many more. Architectural flexibility and strong mapping between input and output is the key virtue of encoder-decoder architecture. The basic functionality of the encoder is to produce an abstract representation of the input, whereas the decoder reproduces the target output. Encoder-decoder is compatible with any of DNN models. In [41], LSTM recurrent neural network is used successfully for language translation, whereas convolutional neural network-based architecture was proposed in [42] for scene image segmentation. Highly accurate semantic pixel-wise segmentation is possible due to the high strength of DNNs. This paper presents an encoder-decoder convolutional neural network for text and background segmentation. The symmetric architecture proposed here comprises identical layers in the encoder and decoder, as shown in Fig. 3. The encoder is composed of three convolutional layers. The decoder is symmetrical to the encoder, consisting of three de-convolution layers in reverse order.

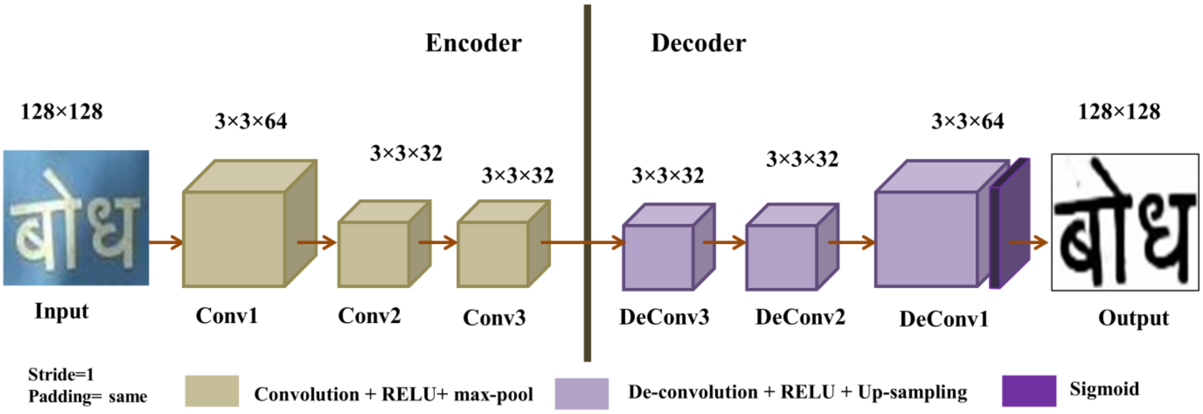

Figure 3. Encoder-decoder architecture

Input image of a fixed size (128×128) is convolved with filter banks for feature extraction. The feature maps generated are directly proportional to the number of filters present in each convolutional layer. Each convolution layer is associated with the Rectified Linear unit (ReLU) activation function. The function Max_pooling is applied on each feature map. The Max_pooling operation reduces spatial information and generates a translation-invariant feature map. In the entire process, three convolution layers followed by Max-pooling are present in the encoder network. The filter banks at each layer with the sizes 64, 32 and 32 respectively and each of 3×3 kernel size are present. Encoded output is passed as an input to the decoder. The process of deconvolution is carried out at the decoder end. Three layers of convolution layers with ReLU followed by up-sampling are constructed. The correspondent de-convolution layer possesses 32, 32 and 64 filter banks with 3×3 kernel size. Unlike pooling, up-sampling is performed to regain the segmented image in the original size. An additional convolution layer of the single filter with ‘sigmoid’ activation function is attached at the end. The sigmoid layer is added to get the output in the normalized range of [0, 1] as it was in the input layer. Stride and padding are the two control parameters of dimensionality reduction. The rate of slide of kernel over the input is defined by stride. The non-overlapping one-pixel shift of kernel (i.e. stride=1) is set at each layer. A higher stride rate produces a higher reduction. Padding helps kernel to fit into the input window. The padding of the same size is set at each layer. The detailed description of layers and architectural flow of the encoder-decoder convolution neural network is shown in Table 2:

Encoder layers	Decoder layers
Input: 128×128 **Padding:** same **Stride:** 1	**Output:** 128×128 **Padding:** same **Stride:** 1
Conv1: Kernel size: 3×3 No. of filters: 64 ReLU Pool size: 2×2 max-pooling	**DeConv1:** Kernel size: 3×3 No. of filters: 64 ReLU Up-sample: 2×2 Kernel size: 3×3 Sigmoid No. of filters: 1
Conv2: Kernel size: 3×3 No. of filters: 32 ReLU Pool size: 2×2 max-pooling	**DeConv2:** Kernel size: 3×3 No. of filters: 32 ReLU Up-sample: 2×2
Conv3: Kernel size: 3×3 No. of filters: 32 ReLU Pool size: 2×2 max-pooling	**DeConv3:** Kernel size: 3×3 No. of filters: 32 ReLU Up-sample: 2×2

Table 2. Layers of encoder-decoder CNN model

The model is trained with 1052 Devanagari scene text images and correspondent binary images. The feature maps of input images and feature maps of expected segmented images are trained together. The network is trained for 1000 epochs and checkpoints are maintained. Error is back-propagated throughout the network by integrating the Adadelta optimization technique. Adaptive delta (Adadelta) is a simple and architectural independent technique in which weights are updated with the delta factor [43]. Delta is the difference between current and updated weights.

$$\Delta w_t = w_t - w_{t-1} \tag{1}$$

In Adadelta, the learning rate is replaced by ‘D’ moving average of the squared delta, computed as shown in the equation.

$$D_t = \beta D_{t-1} + (1 - \beta)\,(\Delta w_t)^2 \tag{2}$$

Weights are updated by factor D and ‘υ’ exponential moving average of squared gradient ‘gt’ at timestamp t, computed as shown in equations. Control parameter β is set to 0.9 and small floating value ε is used to avoid division by zero error.

$$v_t = \beta v_{t-1} + (1 - \beta)\,(gt)^2 \tag{3}$$

$$w_{t+1} = w_t - \frac{\sqrt{D_{t-1} + \varepsilon}}{\sqrt{v_t + \varepsilon}}\, gt \tag{4}$$

Training is optimized and loss is measured with binary cross-entropy function. Optimization curve in terms of loss versus epoch is plotted in Fig. 4. The two optimization curves indicate training and validation (hold-out) loss. The curve depicts that loss is decreasing with an increasing number of epochs. A good fit of any model is stated as reducing training loss and validation loss to the stability point. Training loss is lesser than validation loss with a minimum gap in between. Hence it is depicted from the plot that the model is evaluated as good fit and stable with minimum error loss. The plot shows the training and validation error-loss against the number of epochs. Checkpoint with minimum validation loss is preserved as the best fit model.

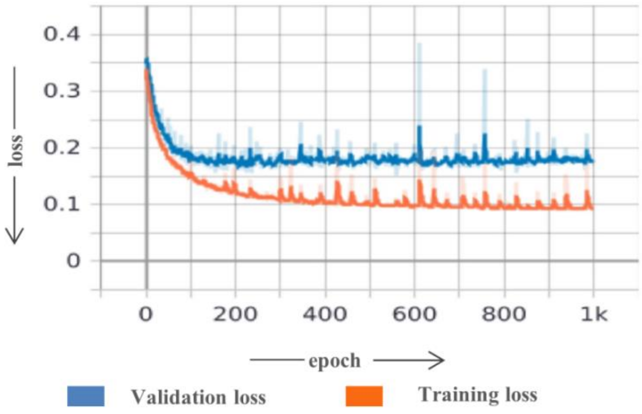

Figure 4. Optimization curve

4 Experimentation

A dataset containing Devanagari scene text images of wall paintings, banners, instructional boards, navigation boards are prepared for the experimentation. Scene text images with different font colours, styles and sizes are captured by a cell phone camera at varying viewing angles and light conditions. Foreground segmentation becomes challenging due to the various challenges mentioned earlier. Considering the complex nature of the problem, text and background segmentation is treated as a bi-level classification problem and an encoder-decoder CNN model is proposed for the segmentation. Rigorous experimentation is carried out and the segmentation results of the proposed technique are compared with the existing methods mentioned below:

i) Otsu's technique for binarization: It is one of the popular and conventional binarization technique, frequently found as a pre-processing technique in the domain of text recognition. Otsu's technique works on a global threshold value. This threshold is computed by observing the co-variance of two classes [44]. The technique is suitable for bi-level classification of foreground and background of high contrast images.

ii) Adaptive thresholding: Instead of relying on a single global threshold, adaptive thresholding works with local thresholds changing over the image [45]. The input image is divided into several blocks of uniform size and mean value of intensities of each block is set as a local threshold of that respective block. After the empirical study, the block size is set to 10×10. The capability of handling intensity variations is an advantage of local adaptive thresholding.

iii) Colour-based clustering: High contrast of text and background colour helps the segmentation process. Different colour models such as RGB, La*b*, YCbCr and HSV are adopted by various researchers for segmentation. Processing in YCbCr colour space produces highly accurate results and can handle various text recognition challenges like shadow effect, uneven illumination, etc. [46]. Y is luminance and Cb, Cr is chroma blue and chroma red colour components, respectively. The variation for black and white colour intensities does not exist in the chroma component. They are present in the Y component only. Different colour shades are formed due to the combination of Y with the chroma component. Hence the effect of shadow, uneven lighting, poor quality can be avoided by separating Y component from CbCr components. The details are described in [46].

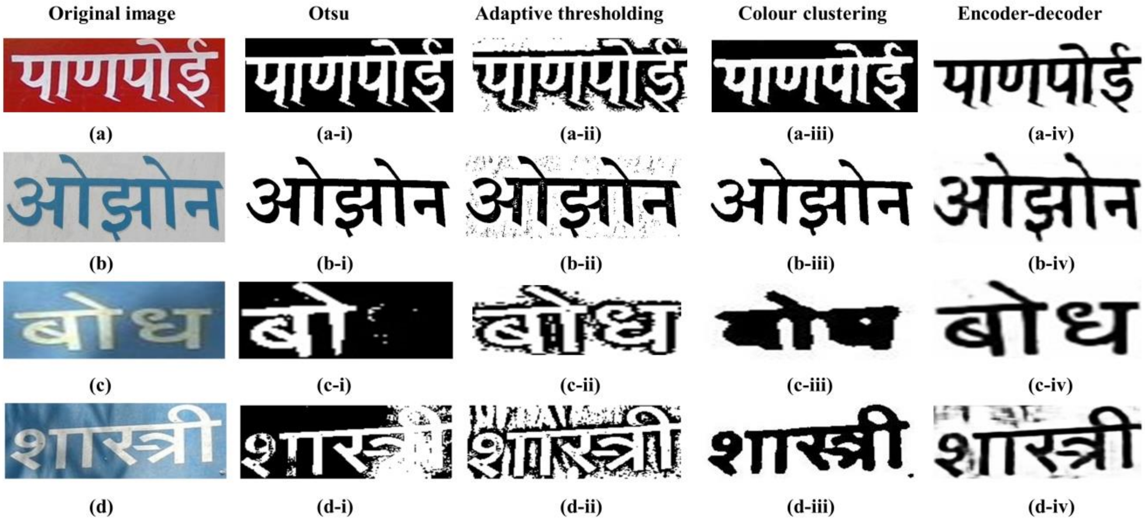

Figure 5. Comparison of outputs from different pre-processing methods for different images (a) image with light text on dark background (b) image with dark text on a light background (c) uneven illumination (d) shadow effect; (i) corresponding binary image (Otsu) (ii) segmentation with adaptive thresholding iii) segmentation by colour clustering iv) segmentation by the encoder-decoder method

Experimental results of different methods are presented in Fig.5. Results of various algorithms on simple images with darker text and lighter background or vice versa, images with shadow and uneven illumination are compared. Otsu's method is computationally simple for the implementation and produces clean segmentation for simple images. For a darker text on a lighter background, it segments text in black with white background and for the opposite scenario, it produces inverted segmentation as shown in Fig 5. (a-i) and (b-i). Otsu's method is suitable for simple images but inefficient to handle shadow and uneven lighting conditions due to the single threshold value (observe Fig.5.(c-i) and (d-i)). To avert the impact, adaptive local threshold values based on blocks are set. Image is divided into equal-sized blocks and the block-wise threshold is set. Thus adaptive thresholding is capable to address the challenges like shadow effect and uneven illumination. But setting the same block size for all kinds of images may not work well always. For example, in Fig. 5(b-ii), the output of adaptive thresholding is poorer than Otsu's. In colour-based clustering, image is segmented into two colour clusters only. The impact of intensity variations caused due to shadows, dust and uneven lighting are eliminated by separating Y and chroma components. Therefore, the segmentation output is noise-free and clear [46]. Colour-based clustering is capable of addressing most of the challenges and produces cleaned segmentation. But, the original shape can not get retained for small font size words as observed in Fig.5.(c-iii). Loss of shape and sharpness of character boundaries is a significant challenge. The proposed encoder-decoder CNN model is trained with Devanagari scene text images and with its expected output binary image. The model has strong learning ability and generated mapping relations are competent enough to handle most of the text recognition challenges. The model retains the shapes and structural characteristics of the script due to its power of pixel-wise classification. Compared to existing segmentation techniques, the proposed technique produces highly accurate segmentation. Irrespective of text and background colour, the proposed technique has an output in the uniform format of black text on white background.

Experimentation is performed using python open source libraries such as TensorFlow and Keras. These libraries are enriched with highly abstracted building blocks of deep learning algorithms. The proposed encoder-decoder architecture is developed using Keras on the top of TensorFlow platform. Different foreground and background segmentation algorithms and several image processing functionalities are implemented using OpenCV. As CNN models are computationally expensive, training of encoder-decoder CNN model is performed on accelerated Tesla K80 GPUs provided by Google Colab. Tesseract version 4.3 engine is used for Devanagari text recognition.

5 Recognition by OCR engine

To meet a common objective of generalized text recognition (i.e., independent of font style, size, etc.) existing OCR engine is used for the recognition purpose. Most of the OCR engines are intended for document character reading [28]. Wider usage and success in printed and handwritten character recognition transformed an OCR system into a commercially available product. Different OCR engines are available in the market and some of them are freely available for experimentation. Adaptive development, inclusive of smart algorithms and extensive support for several languages, made them popular in the domain of recognition. As per the literature survey, Tesseract and ABBYY are the most popular OCR engines [40]. Tesseract OCR with Marathi/Hindi language trained model is used for the experimentation. The model is trained using LSTM, a highly efficient machine learning algorithm in the recognition domain. Tesseract is capable to recognize the regular, special characters, numerals and conjunct characters too. It produces highly accurate results only on a noise-free simple binary image. Thus a more efficient pre-processing technique is required for precise segmentation to produce highly precise recognition results. Text and background segmentation of complex scene text images are tedious compared to plain document images.

A ground-truth dataset consisting of 1052 Devanagari scene text (word) images containing simple characters, composite characters, special characters, and modifiers is prepared and used for experimentation. Images of banners, instruction boards, wall paintings, bus route details, nameplates written in Devanagari script were captured by a smartphone (Lenovo K900 with 13MP rear camera). The dataset covers horizontal, non-horizontal and curved text images. Images were captured from different angles at varying light conditions at different locations. Ground-truth labelling is done manually for each word present in the image. The bounding-box details of each word such as height, width and location are mentioned in the annotation file. The Unicode corresponding to each word present in the bounding box is also stored in the annotation file. Word is represented as a sequence of Unicode for the automatic evaluation of the recognition result [31]. Word recognition rate (WRR) and character recognition rate (CRR) are two important accuracy measures in the domain of text recognition. WRR is the ratio of the number of correctly recognized words to the total number of words present in the ground truth dataset.

$$WRR = \frac{c\_words}{total\_words} \qquad 5$$

Similarly, CRR is the ratio of correctly recognized characters to the total number of characters present in the ground_truth dataset. For the segmentation-free approach, correctly recognized characters are computed by Levenshtein distance formula. Levenshtein distance between two words is the minimum edit distance between sequences of characters. In short, it shows a number of characters by which given two words differ.

$$CRR = \frac{total\_chars - \sum_{i=1}^{n} levenshtein\_distance(c\_words_i - total\_words_i)}{total\_chars} \qquad 6$$

Where n is the total number of words in ground-truth. WRR and CRR are used for the evaluation of the proposed Devanagari text recognition method. Segmentation results of different existing techniques and proposed encoder-decoder CNN technique are passed as an input to the Tesseract for recognition. Tesseract possesses some inbuilt functionality such as contrast enhancement and slant/skew correction. Recognition rates in terms of word and character of different segmentation techniques are expressed in Table 3. It is clear from Table 3 that Tesseract produces low recognition rate for non pre-processed input images. Results are improved by applying elementary pre-processing techniques like grey-scaling and Otsu's technique. Though the colour-based clustering can handle many text detection and recognition challenges, it produces poor recognition results because the shape of characters does not get preserved in the segmentation process. OCR engines produce highly accurate results if spatial and contextual information gets retained by the segmentation technique. So along with noise-free segmentation, retention of the original shape of character strokes is equally important. Compared to other techniques, the adaptive thresholding technique produces better results but manually setting of the tuning parameter is a hurdle. The proposed encoder-decoder CNN technique is more efficient and has highly accurate results compared to the existing one. The three main advantages of the proposed technique are i) Capability to handle most of the text recognition challenges like shadow effects, effects of climatic conditions and uneven illumination ii) Retention of shapes and contextual information iii) It produces segmentation results in a uniform format with black text on white background irrespective of text and background colour.

Segmentation methods	CRR	WRR
Orignal	33.99	14.16
Grayscale	48.24	24.24
Otsu	48.89	26.52
Adaptive Thresholding	66.32	37.45
Colour-based Clustering	42.73	23.76
Proposed technique	**68.92**	**39.06**

Table 3. Recognition rates of different segmentation-techniques.

Devanagari word recognition is more tedious due to its structural characteristics. Recognition becomes complex due to the presence of some isolated modifiers (e.g., Anuswara). Recognition is also highly sensitive to noise as it may falsely get recognized as one of the modifiers. As a result, there is a massive difference between CRR and WRR. For example, वंदन and वदन are two different words with a difference of one modifier. Recognition accuracy gets affected by false recognition of noise as 'Anuswara' or removal of 'Anuswara' by treating it as noise by mistake in the pre-processing step.

5.1 Significance of Padding

The ground-truth dataset contains Devanagri text with varying font sizes and styles. Few words are bigger in size and few are too small for accurate recognition. Sufficient size is required by the OCR engine to efficiently learn the features for correct recognition. To improve the efficiency of recognition, simple padding is applied to the segmented image. Padding helps in capturing suitable features that are being used for recognition. Deciding padding element (zero or one) is difficult in other pre-processing techniques due to light text on a darker background or dark text on a lighter background. Images with darker backgrounds must be padded with zeros and lighter backgrounds must be padded with ones. The problem can be solved by applying padding twice on the segmented image, initially with zeros and with ones after that. In fact, this is a computationally expensive step as recognition is carried out twice. But in the proposed approach, uniformity is maintained in the outputs of foreground segmentation, padding becomes easy. The padding of 25% of image size is carried out for each image and the recognition results are computed.

Results have significantly improved due to padding as enough space is provided for the OCR engine to learn character features. Sometimes the OCR engine produces false recognition of noise in the form of some punctuation symbols like a hyphen, commas and full stops etc., which are eliminated in the post-processing step. A pre-defined set of punctuation symbols is prepared for the post-processing step. Improved results of Adaptive thresholding and proposed technique after applying padding and post-processing are mentioned in Table 4. Higher accuracy depicts that the proposed encoder-decoder CNN technique is highly efficient for foreground separation and is very suitable for text recognition.

Segmentation methods	CRR	WRR
Adaptive Thresholding	67.1	52.38
Proposed technique	**82.66**	**61.69**

Table 4. Recognition results after padding

Word recognition rate may improve by incorporating post-processing techniques like usage of a word dictionary or any semantic modeling. Results can be enhanced further by customized training of different font styles present in the ground-truth dataset. To summarize, the result of recognition is affected due to complex backgrounds, effects of climatic conditions, varying light conditions, different font styles, sizes, colours, and structural pattern of Devanagari scene text image.

6 Conclusion

Various climatic conditions strongly influence the quality of scene text images. Thus efficient foreground segmentation technique is very much essential. A robust encoder-decoder CNN model is developed for the segmentation of Devanagari text from complex backgrounds. The in-depth learning ability of the model allows for pixel-wise precise classification. The designed model is well-organized, good-fitted and competent enough to handle various recognition challenges such as uneven illumination, complex background, and shadow effect. Uniformity is maintained in the foreground segmentation outputs (black text on white background) irrespective of font style, size, and colour. The retention of shape and structural information is the biggest advantage of the proposed model, leading to a good recognition rate. Segmentation results of the proposed technique are compared with other popular techniques. Comparative results state that the proposed segmentation technique is most appropriate for word recognition. The Tesseract OCR engine is used for text recognition to achieve unconstrained text recognition independent of font style, size, and orientation. Comparative results depict the significance of an efficient foreground segmentation technique in the word recognition process. The proposed technique is highly recommended due to accurate and noise-free segmentation with retention of contextual and structural features of the script.

Acknowledgement

This work was supported by the Science and Engineering Research Board (SERB), New Delhi, India for non-commercial research under the Young Scientist Scheme (Start-up Grant Y.S.S./2015/000812).

References

[1] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends", *Frontiers of Computer Science,* 10(1):19–36, 2016, https://doi.org/10.1007/s11704-015-4488-0.

[2] Shiravale S. S., Sannakki S. S. and Rajpurohit V. S., "Recent Advancements in Text Detection Methods from Natural Scene Images", *International Journal of Engineering Research and Technology,* 13(6): 1344-1352, 2020.

[3] R. Jayadevan, S. R. Kolhe, P. M. Patil, U. Pal, "Offline Recognition of Devanagari Script: A Survey", *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 41(6):782-796, 2011, doi: 10.1109/TSMCC.2010.2095841.

[4] B. Chaudhuri, U. Pal, "A complete printed Bangla OCR system", Pattern Recognition, 3(5):531-549, 1998, https://doi.org/10.1016/S0031-3203(97)00078-2.

[5] L. Zuo, H. Sun, Q. Mao, R. Qi and R. Jia, "Natural Scene Text Recognition Based on Encoder-Decoder Framework," *in IEEE Access,* 7: 62616-62623, 2019, doi: 10.1109/ACCESS.2019.2916616.

[6] B. Shi, X. Wang, P. Lyu, C. Yao and X. Bai, "Robust Scene Text Recognition with Automatic Rectification," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Las Vegas, 4168-4176, 2016, doi: 10.1109/CVPR.2016.452.

[7] F. Zhan and S. Lu, "ESIR: End-To-End Scene Text Recognition via Iterative Image Rectification," */CVF Conference on Computer Vision and Pattern Recognition (CVPR),* CA, USA, 2054-2063, 2019, doi: 10.1109/CVPR.2019.00216.

[8] K. S. Raghunandan, P. Shivakumara, S. Roy, G. H. Kumar, U. Pal and T. Lu, "Multi-Script-Oriented Text Detection and Recognition in Video/Scene/Born Digital Images", *in IEEE Transactions on Circuits and Systems for Video Technology,* 29(4):1145-1162, 2019, doi: 10.1109/TCSVT.2018.281764

[9] X. Yin, Z. Zuo, S. Tian and C. Liu, "Text Detection, Tracking and Recognition in Video: A Comprehensive Survey," *in IEEE Transactions on Image Processing*, 25(6): 2752-2773, 2016, doi: 10.1109/TIP.2016.255432.

[10] B. Su and S. Lu , "Accurate Scene Text Recognition Based on Recurrent Neural Network", *Computer Vision -- ACCV 2014, Lecture Notes in Computer Science*, Springer, Cham., 9003, 2015, https://doi.org/10.1007/978-3-319-16865-4_3.

[11] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition", *In NIPS Deep Learning Workshop*, 2014.

[12] Rajib. Ghosh, Chirumavila. Vamshi, Prabhat. Kumar, "RNN based online handwritten word recognition in Devanagari and Bengali scripts using horizontal zoning", *Pattern Recognition*, 92: 203-218, 2019, https://doi.org/10.1016/j.patcog.2019.03.030.

[13] X. Ren, Y. Zhou, Z. Huang, J. Sun, X. Yang and K. Chen, "A Novel Text Structure Feature Extractor for Chinese Scene Text Detection and Recognition," *in IEEE Access,* 5: 3193-3204, 2017, doi: 10.1109/ACCESS.2017.2676158.

[14] M. Jain, M. Mathew and C. V. Jawahar, "Unconstrained scene text and video text recognition for Arabic script," *International Workshop on Arabic Script Analysis and Recognition (ASAR),* Nancy, 26-30, 2017, doi: 10.1109/ASAR.2017.8067754.

[15] K. Tulsyan , N. Srivastava , A. Mondal , C. V. Jawahar, "A Benchmark System for Indian Language Text Recognition", *Document Analysis Systems. Lecture Notes in Computer Science,* Springer, Cham ,12116, 2020, https://doi.org/10.1007/978-3-030-57058-3_6.

[16] U. Pal, B.B. Chaudhuri, "Indian script character recognition: a survey", *Pattern Recognition,* 37(9):1887-1899, 2004, https://doi.org/10.1016/j.patcog.2004.02.003.

[17] U. Pal, T. Wakabayashi and F. Kimura, "Comparative Study of Devnagari Handwritten Character Recognition Using Different Feature and Classifiers," *10th International Conference on Document Analysis and Recognition*, Barcelona, 1111-1115, 2009, doi: 10.1109/ICDAR.2009.244.

[18] Soumen ba, gaurav harit, "A survey on optical character recognition for Bangla and Devanagari scripts", *Sadhana* , 38(1): 133–168, 2013, DOI: 10.1007/s12046-013-0121-9

[19] U. Pal, R. Jaydevan, N. Sharma, "Handwriting Recognition in Indian Regional Scripts: A Survey of Offline Techniques", *ACM Transactions on Asian Language Information Processing,* 11(1):1-35, 2012, doi: 10.1145/2090176.2090177.

[20] Partha Pratim Roy, Ayan Kumar Bhunia, Ayan Das, Prasenjit Dey, Umapada Pal, "HMM-based Indic handwritten word recognition using zone segmentation", *Pattern Recognition*, 60:1057-1075, 2016, doi:10.1016/j.patcog.2016.04.012.

[21] B. Thakral and M. Kumar, "Devanagari handwritten text segmentation for overlapping and conjunct characters- A proficient technique," *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization,* Noida, 1-4,2014.

[22] R. Ghosh and P. P. Roy, "Comparison of Zone-Features for Online Bengali and Devanagari Word Recognition Using HMM," *15th International Conference on Frontiers in Handwriting Recognition (ICFHR),* Shenzhen, 435-440, 2016, doi: 10.1109/ICFHR.2016.0087.

[23] P. Sahare and S. B. Dhok, "Multilingual Character Segmentation and Recognition Schemes for Indian Document Images," *in IEEE Access*, 6:10603-10617, 2018, doi: 10.1109/ACCESS.2018.2795104.

[24] P. S. Deshpande, Latesh. Malik and Sandhya. Arora, "Handwritten devnagari character recognition using connected segments and minimum edit distance," *TENCON IEEE Region 10 Conference*, Taipei, 1-4, 2007.

[25] Umapada Pal, Partha Pratim Roy, Nilamadhaba Tripathy, Josep Lladós, “Multi-oriented Bangla and Devnagari text recognition”, *Pattern Recognition*, 43(12): 4124-4136, 2010, https://doi.org/10.1016/j.patcog.2010.06.017.

[26] S. Tian, U. Bhattacharya, S. Lu, B. Su, Q. Wang, X. Wei, Y. Lu, C. L. Tan, “Multilingual scene character recognition with co-occurrence of histogram of oriented gradients”, *Pattern Recognition*, 51:125-134, 2016, https://doi.org/10.1016/j.patcog.2015.07.009.

[27] Parshuram M. Kamble, Ravinda S. Hegadi, “Handwritten Marathi Character Recognition Using R-HOG Feature”, *Procedia Computer Science*, 45: 266-274, 2015, https://doi.org/10.1016/j.procs.2015.03.137.

[28] A. Gupta, R. Sarkhel, N. Das, M. Kundu, “Multiobjective optimization for recognition of isolated handwritten Indic scripts”, *Pattern Recognition Letters*, 128: 318-325, 2019, https://doi.org/10.1016/j.patrec.2019.09.019.

[29] U. Pal and B. B. Chaudhuri, "Automatic identification of English, Chinese, Arabic, Devnagari and Bangla script line," *Proceedings of Sixth International Conference on Document Analysis and Recognition*, Seattle U.S.A., ,790-794, 2001.

[30] Bhattacharya, Nilanjana , Roy, Partha , Pal, “Sub-Stroke-Wise Relative Feature for Online Indic Handwriting Recognition”, *ACM Transactions on Asian and Low-Resource Language Information Processing*,18:1-16,2018.

[31] N. Sankaran, A. Neelappa and C. V. Jawahar, "Devanagari Text Recognition: A Transcription Based Formulation," *12th International Conference on Document Analysis and Recognition,* Washington, DC, 678-682, 2013, doi: 10.1109/ICDAR.2013.139.

[32] N. Sankaran and C. V. Jawahar, "Recognition of printed Devanagari text using BLSTM Neural Network," *Proceedings of the 21st International Conference on Pattern Recognition,* Tsukuba, 322-325, 2012.

[33] P. Keshri, P. Kumar and R. Ghosh, "RNN Based Online Handwritten Word Recognition in Devanagari Script," *16th International Conference on Frontiers in Handwriting Recognition (ICFHR*), Niagara Falls, NY, 517-522, 2018, doi: 10.1109/ICFHR-2018.2018.00096.

[34] K. Jindal, R. Kumar, “A new method for segmentation of pre-detected Devanagari words from the scene images: Pihu method”, *Computers & Electrical Engineering*, 70: 754-763, 2018.

[35] V. Bansal, R.M.K. Sinha, “Segmentation of touching and fused Devanagari characters”, *Pattern Recognition*, 35(4): 875-893, 2002, https://doi.org/10.1016/S0031-3203(01)00081-4.

[36] M. Mathew, M. Jain and C. V. Jawahar, "Benchmarking Scene Text Recognition in Devanagari, Telugu and Malayalam," *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, 42-46, 2017, doi: 10.1109/ICDAR.2017.364.

[37] S. Roy, P. Shivakumara, P. P. Roy, & C. L. Tan, “Wavelet-gradient-fusion for video text binarization” *In Proceedings of the ICPR*, 3300–3303, 2012.

[38] S. Roy, S. Palaiahnakote, P. P. Roy, U. Pal, C. L. Tan, T. Lu, “Bayesian classifier for multi-oriented video text recognition system”, *Expert Systems with Applications*, 42(13):5554-5566, 2015.

[39] A. K. Bhunia, G. Kumar, P. P. Roy, “Text recognition in scene image and video frame using Color Channel selection” *Multimed Tools Appl* ,77:8551–8578, 2018, https://doi.org/10.1007/s11042-017-4750-6.

[40] T. Q. Phan, P. Shivakumara, S. Bhowmick, S. Li, C. L. Tan and U. Pal, "Semiautomatic Ground Truth Generation for Text Detection and Recognition in Video Images," *IEEE Transactions on Circuits and Systems for Video Technology*, 24(8):1277-1287, 2014, doi: 10.1109/TCSVT.2014.2305515.

[41] I. Sutskever, O. Vinyals, and V. Quoc, “Sequence to sequence learning with neural networks”, *27th International Conference on Neural Information Processing Systems - 2 (NIPS’14). MIT Press*, Cambridge, U.S.A., 3104–3112, 2014.

[42] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481-2495, 2017.

[43] M. D. Zeiler, “ADADELTA: An Adaptive Learning Rate Method”, *CoRR, abs/1212.5701*, 2012.

[44] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Trans. Systems, Man, and Cybernetics*, 9(1): 62–66, 1979.

[45] B. Bataineh, SNHS. Abdullah, K. Omar, M. Faidzul, “ Adaptive Thresholding Methods for Documents Image Binarization”, *In Mexican Conference on Pattern Recognition (MCPR), Springer*, Berlin, Heidelberg, 230-239, 2011, https://doi.org/10.1007/978-3-642-21587-2_25.

[46] S. S. Shiravale, R. Jayadevan, & S.S. Sannakki, “Devanagari Text Detection From Natural Scene Images”, *International Journal of Computer Vision and Image Processing (IJCVIP),* 10(3):44-59, 2020, doi:10.4018/IJCVIP.2020070104.

[47] S. R. Narang, M. K. Jindal, S. Ahuja, et al. “On the recognition of Devanagari ancient handwritten characters using SIFT and Gabor features”, *Soft Computing*, 24:17279–17289, 2020.

[48] SP Deore, A. Pravin, “Devanagari Handwritten Character Recognition using fine-tuned Deep Convolutional Neural Network on trivial dataset” *Sādhanā*, Springer, 45:1-13, 2020, https://doi.org/10.1007/s12046-020-01484-1.

[49] T. Kundaikar, J. D. Pawar, “Multi-font Devanagari Text Recognition Using LSTM Neural Networks”, In: Luhach A., Kosa J., Poonia R., Gao XZ., Singh D. (eds) *First International Conference on Sustainable Technologies for Computational Intelligence, Advances in Intelligent Systems and Computing*, 1045:495-506,2020, https://doi.org/10.1007/978-981-15-0029-9_39.

[50] SP Deore, A. Pravin, “Real-time Devanagari Numeral Recognition using Capsule Neural Network”, *International Journal of Advanced Science and Technology*, 29(7):2817-2825, 2020.

Sankirti S. Shiravale has completed BE degree from Shivaji University, Kolhapur in 2003, M. E. degree from University of Pune, Pune in 2012. Currently, she is with Department of Computer Engineering, Marathwada Mitra Mandal’s College of Engineering, Pune as an Assistant Professor and pursuing her research from V.T.U., Belagavi. Her research area is Image Processing and Pattern Recognition.

Jayadevan R received the B. Tech degree from Cochin University of Science and Technology, Kochi in 2002, the ME degree from University of Pune, Pune in 2006, both in Computer Science and Engineering, and the Ph.D. degree in Computer Engineering from North Maharashtra University, Jalgaon, in 2013. He is currently an Associate Professor with the Department of Computer Engineering, Army Institute of Technology, Pune, India. His research interests include Image Processing and Pattern Recognition.

Sanjeev S Sannakki, has completed his Ph.D. degree in Image processing & Data Mining from VTU. Belagavi. His career spans over a period of two decades is in the field of teaching, research and other diversified in-depth experience in academics. He is currently working as a Professor in the Department of C.S.E., Gogte Institute of Technology, Belgaum. Currently, he is shouldering the responsibility of Head of the Research centre. He has published several papers in reputed national/international conferences and journals. He is also guiding the research scholars & UG/PG students of VTU.