Hindawi
Computational Intelligence and Neuroscience
Volume 2021, Article ID 2676780, 11 pages
https://doi.org/10.1155/2021/2676780

Research Article

Scene Text Recognition Based on Bidirectional LSTM and Deep Neural Network

MVV Prasad Kantipudi,[1] **Sandeep Kumar**,[2] **and Ashish Kumar Jha**[3]

[1] *Department of E&TC, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, India*
[2] *Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India*
[3] *Nepal Engineering College, Kathmandu, Nepal*

Correspondence should be addressed to Ashish Kumar Jha; ashishkj@nec.edu.np

Received 26 July 2021; Accepted 3 November 2021; Published 23 November 2021

Academic Editor: Gaurav Singal

Deep learning is a subfield of artificial intelligence that allows the computer to adopt and learn some new rules. Deep learning algorithms can identify images, objects, observations, texts, and other structures. In recent years, scene text recognition has inspired many researchers from the computer vision community, and still, it needs improvement because of the poor performance of existing scene recognition algorithms. This research paper proposed a novel approach for scene text recognition that integrates bidirectional LSTM and deep convolution neural networks. In the proposed method, first, the contour of the image is identified and then it is fed into the CNN. CNN is used to generate the ordered sequence of the features from the contoured image. The sequence of features is now coded using the Bi-LSTM. Bi-LSTM is a handy tool for extracting the features from the sequence of words. Hence, this paper combines the two powerful mechanisms for extracting the features from the image, and contour-based input image makes the recognition process faster, which makes this technique better compared to existing methods. The results of the proposed methodology are evaluated on MSRATD 50 dataset, SVHN dataset, vehicle number plate dataset, SVT dataset, and random datasets, and the accuracy is 95.22%, 92.25%, 96.69%, 94.58%, and 98.12%, respectively. According to quantitative and qualitative analysis, this approach is more promising in terms of accuracy and precision rate.

1. Introduction

Understanding the visual scene is an active research area for the computer vision community. It needs enormous research in the field of computer vision and its subfields. Visual scene understanding includes the processing of both image and text, and it is always a difficult task to understand the scene and read the text written in the image. This research area is increasing gradually because it is helpful for many applications such as content-based image retrieval systems, assistance for blind people, automatic navigation systems in vehicles, and digitization of textbooks. OCR [1] is a traditional technique of recognizing the text from the documents, and the accuracy of this technique is good in the scanned documents; but when the same technique is applied to scene images, the performance of this method was not up to the mark [2]. The recognition of text from the scene needs special features because the character present in the scene may differ in size, shape, color, writing style, orientation, aspect ratio, quality of the image due to different lighting conditions, and blurred and complex background. These are the various challenges of text detection and text recognition. Generally, text detection identifies the location where exactly text is present in the image and creates a bounding box for each word or letter or line of text, and it also improves the accuracy of text recognition. The sample example of text detection is shown in the Figure 1.

Text recognition allows the computer to understand and predict the text in the given input scene image and convert it into the computer's understandable format. Text recognition is the most popular method for converting old printed documents into digitized forms. This process looks simple and more accessible because most of the image proceeding techniques follow the same. According to the literature

Figure 1: Sample image of text-involved scenes of SVT dataset.

survey, most of the text detection and recognition techniques are influenced by machine learning and deep learning techniques. Given the problems mentioned above and their solution, the main goal of this paper is to give a novel approach to text detection and text recognition based on deep neural networks and bidirectional LSTM. We proposed a scene text recognition method, and the proposed system is divided into three steps: in the first step, adaptive binarization technique is applied to the image so that the noise can be removed from the image, and it helps to extract the features from the blurred and complex background. In the second step, the contour detection technique is applied to the image, and it detects the meaningful area of the image, which makes the detection process easier and faster. In the third step, CNN-based architecture is designed in such a way that it can locate the text region and create a bounding box on each letter and also predict the characters. Here, CNN is combined with Bi-LSTM to make the classifier more powerful, and it is a handy tool for extracting the features from the sequence of words. This paper combines the two powerful mechanisms for extracting the features from the image and contour-based input image making the recognition process faster, which makes this technique better compared to existing methods. The complete detail of the proposed method is discussed in Section 3. The rest of the paper is structured as follows: Section 2 describes the related work. Section 3 discusses the proposed work. Section 4 presents the experimental results and their comparison with existing methods. Section 5 shows the conclusion and future scope.

2. Literature Work

Many researchers have worked on the various techniques of detection of the text in images. Some researchers explored the texture-based approach for locating the text information in the image and used the sliding window concept to analyze the unique texture present in the input image [3–7]. Some researchers focused on sparse-based text detection methods used for computer vision applications [8–12]. It is proposed by Zhao et al. [8]. These methods work to transform the image into edge maps. A further sliding window is used to extract the text patches present in the image, and then classification has been performed. Most researchers focus on deep learning-based methods for scene text detection and recognition, and a detailed comparative analysis has been done in Table 1.

A tremendous amount of work has been done in scene text recognition, and results are also satisfactory [27–32]. However, these algorithms cannot give better results if the background is complex, blurry, and has different lighting conditions. The computation cost is very high when the algorithms are applied to the real dataset. Therefore, it remains a challenge.

3. Proposed Work

The proposed method is divided into three steps: firstly, finding the contour of the image; secondly, the text detection is using CNN; and thirdly, the text recognition using combined RNN and Bi-LSTM. The detailed description is discussed further, and the flow chart of the proposed method is shown in Figure 2.

3.1. Contour Detection. The scene text recognition-based method is essential to identify the region where exactly text is present in the image. Rather than working on the whole image, only the object's boundary is sufficient for further processing. Considering the same, in the proposed method, the first contour of the image is identified [32]. Contour is used to find the boundary of the objects which is present in the image. These boundaries can be identified in different ways, such as finding the edges of the objects and finding the intensities of objects which are present in the image.

In the proposed method, we used the wireframe-based boundary detection method in which the whole image is traced using structuring elements, and the first pixel of the object is identified. This first pixel represents the component of the object. Identifying the first pixel in the image always depends on how the tracing has to begin in the input image.

The preferred direction of the tracing is the left most corner of the image and then towards the right direction of the image. The tracing of the image is continued until it will not find the contour of the whole image. Finally, all the boundaries of the objects are integrated, and the algorithm displays the contour of the image. The results of the contour detection are shown in Figure 3.

3.2. Text Detection Using CNN. The performance of any model always depends on the ability to discriminate the various features. An image-text can be arranged as a sequence of letters. A sequence of convolution and max-pooling layers are used to detect the text in an image. In the proposed method, four layers of CNN classify that the image's patch contains a character. The configuration of CNN is represented in Table 2 and Figure 4. First, the CNN classifier is trained with 62 classes in which 26 classes are used for uppercase letters, 26 classes are used for lower case letters, 10 classes are used for digits (0–9), and 1 for spacing. The image patches are directly classified as letters or digits; therefore, here, binary classifier is not required. The learned

Table 1: Literature study on existing methodology.

S. no.	Author & year	Methodology	Dataset	Performance
1	S. Yasser Arafat et al. [12], 2020	Faster RCNN + two stream deep neural network (TSDNN)	UPTI dataset	Avg. precision = 98% R. R. = 95.20%

2	Asghar Ali Chandio et al. [13], 2020	Multiscale and multilevel features	Chars74 K and ICDAR03 datasets	Precision = 90%
				Recall = 91%
				F-score = 91%

3	Yao Qin et al. [14], 2020	Faster RCNN + BLSTM	ICDAR 2015 datasets	Precision = 89.8%
				Recall = 84.3%
				F-score = 86.9%

4	Jheng-Long Wu et al. [15], 2020	BLSTM + CNN	Corpus dataset	Macro-F1 = 72%
				Micro-F1 = 71%

5	S. Yasser Arafat et al. [16], 2020	(AlexNet and Vgg16) + BLSTM	UPTI dataset	Accuracy = 97%

6	Sardar Jaf et al. [17], 2019	Recurrent neural network (RNN) + BLSTM	English web treebank universal dependencies dataset	Precision = 91.43%
				Recall = 94.52%
				F-score = 92.20%

7	M. A. Panhwar et al. [18], 2019	ANN	Self-dataset	Accuracy = 85%

8	Yen-Min Su et al. [19], 2019	Contour + morphological operation + ROI	ICDAR datasets	Accuracy = 93.44%
				Recall = 79.16%
				F-score = 85.71%

9	Ling-Qun Zuo Su et al. [20], 2019	CNN + BLSTM	SVT dataset, IIIT5K dataset, ICDAR 2003 and 2015 dataset	Accuracy = 95.96%
				Accuracy = 98%
				Accuracy = 98.2%
				Accuracy = 91%

10	Baoguang Shi et al. [21], 2017	CRNN	SVT dataset, IIIT5K dataset, ICDAR dataset	Accuracy = 97.5%
				Accuracy = 97.8%
				Accuracy = 98.7%
				Accuracy = 89.6%

11	Xiaohang Ren et al. [22], 2017	Text structure component detector (TSCD)	Ren’s dataset, Zhou’s dataset, Pan’s dataset	Precision = 82%
				Recall = 72%
				F-score = 77%

12	Xiang Bai et al. [23], 2016	Bag of strokelets + HOG	SVT dataset, IIIT5K dataset, ICDAR 2003 dataset	Accuracy = 80.99%
				Accuracy = 85.6%
				Accuracy = 82.64%

13	Mingkun Yang et al. [24], 2021	CAPTCHA system	IIIT5K, SVT, IC03 IC13, IC15, SVTP CUTE	Accuracy = 92.9%, 89.6%, 92.5%, 92.2%, 76.8%, 80%, 77.1%

14	Anna Zhu et al. [25], 2021	Anchor selection-based region proposal network	ICDAR2013, ICDAR2015, and MSRA-TD500	Precision = 90.18%, 83.34%, 84.67%
				Recall = 91.16%, 79.99%, 80.37%
				F-score = 90.62%, 81.63%, 82.49%

15	ZiLing Hu et al. [26], 2021	Text contour attention text detector	ICDAR2015, CTW1500	Precision = 88.9%, 86.5%
				Recall = 85.2%, 80%
				F-score = 87%, 83.1%

features are more specific and easy to discriminate from each other, making the learning process more accurate and speedy. The bounding box needs to be generated for each text present to detect the text in the image. The input image of this step is the contour image. There is a possibility that the input image can differ in size; therefore, to make it uniform, the size of the input image is 24×24, and each image is a greyscale image. First, the input image is padded from each side because if any character is near to the boundary of the image, then it can be detected through a sliding window. Using the sliding window, each row of the image is traced, NMS is performed to noise if it is present in the image, and the mean deviation and standard deviation of spacing are calculated. If the spacing value is lower than the threshold value, then it is considered that neighbour pixels are connected. Now, finally, the bounding box is identified for each character using a connected component analysis algorithm.

3.3. Scene Text Recognition Using Combined RNN and Bi-LSTM.

This step is used to recognize the characters that are present in the image. Generally, the recognition system’s performance depends on the segmentation techniques, but sometimes good segmentation will also lead to poor recognition because of noise, different lighting conditions, different sizes of text, rotation and illumination, etc. Deep learning-based methods are used, and in this paper, to

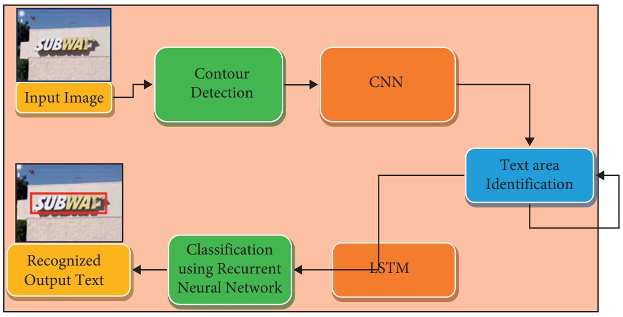


Figure 2: Block diagram of proposed work.

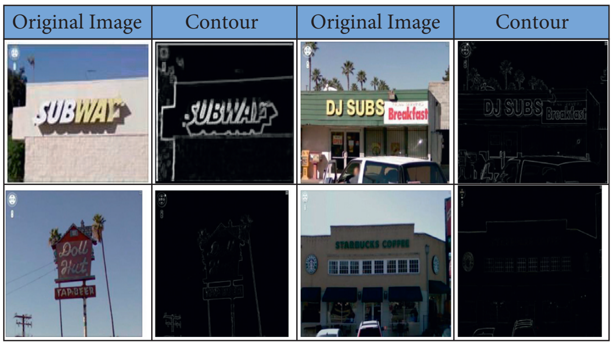


Figure 3: Original image and contour of the image on SVT dataset.

Table 2: Hyperparameters used in the proposed work.

Parameter	Value
Epochs	50
Validation_split = 0.1	0.1
Drop out	0.2
Filters	16
Batch_size	64 × 64
Learning rate	0.00001

overcome these problems, we combined RNN and LSTM to improve the recognition rate. The first features are extracted from the image. The CNN classifier is used for the sequential feature extraction from the image, and training is done for all 63 classes mentioned in Section 3.2. The feature extraction is done through the sliding window concept. The images which are already detected are the input for this step. The first padding of 12 pixels is done on the image, and the new image's size is now 24 × 94. For the partition of the padded image, a subwindow is used with size 24 × 24. Each portioned patch of the image is fed into the trained CNN, and this trained CNN extracts the features from the image with size 4 × 4 × 256 and 1000 features which are the output of the 4th convolution layer and the first FC layer. These two feature vectors are combined, and it forms a one-dimensional feature vector of size 5096. PCA and normalization technique is applied to reduce the size of the feature vector. Now, the new feature vector is the size of 256-d, and these are the local and global features of the image. After extraction of local and global features from the image, the next step is feature labeling. For labeling of the feature, RNN is used in the proposed method. RNN is a unique neural network that

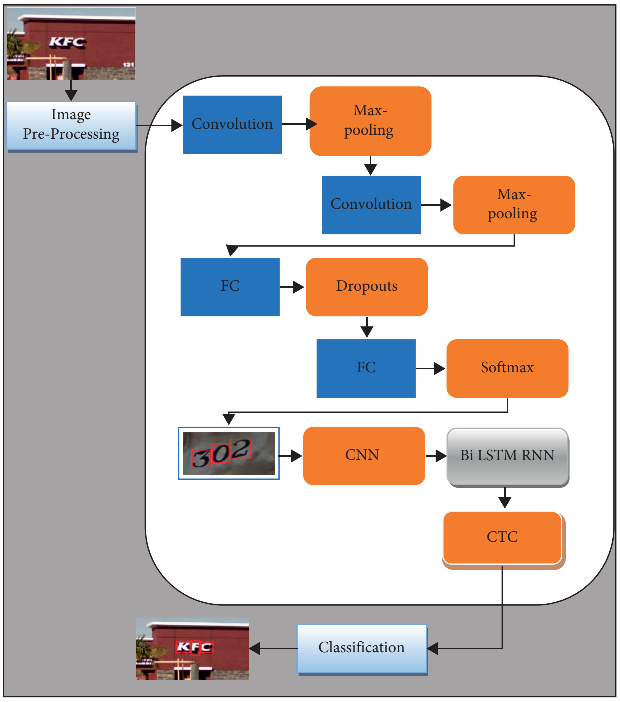


Figure 4: Block diagram of RCNN combined with Bi-LSTM.

can make use of past feature information, and it can also process the sequential inputs. To make the RNN more powerful, LSTM is combined here. LSTM has the capability of memorizing contextual information for a long time. LSTM consists of the memory cell and connection to itself and three gates that control the flow of information. The pictorial representation of the LSTM is shown in Figure 5.

i_t is the gate; C_{t-1} is the status at the last cell, and it is hidden; f_t is the forget gate; H_t is the final state of the latest C_t; W is the weight of each connection; and O_t is the output gate. The following equations compute the values of the previous parameters:

$$\begin{aligned}
i_t &= \sigma\left(W_{xi} * X_t + W_{hi}H_{t-1} + W_{ci\circ}C_{t-1} + b_i\right),\\
f_t &= \sigma\left(W_{xf} * X_t + W_{hf}H_{t-1} + W_{cf\circ}C_{t-1} + b_f\right),\\
C_t &= f_{t\circ}C_{t-1} + i_{t\circ}\tanh\left(W_{xc} * X_t + W_{hc}H_{t-1} + b_c\right),\\
O_t &= \sigma\left(W_{xo} * X_t + W_{ho}H_{t-1} + W_{co\circ}C_t + b_o\right),\\
H_t &= O_{t\circ}\tanh\left(C_t\right).
\end{aligned} \tag{1}$$

It is better to access the past and future contextual information to recognize the text string properly. Bi-LSTM consists of two hidden layers in which one hidden layer is used to process the features in the forward direction and the other is used to process the features in the backward direction. Both the hidden layers have produced the output using the same output layer. Bi-LSTM is applied recursively for each feature present in the feature sequence in the sequence labeling process. According to the computation (mentioned in the above equation), it takes input as the current state and neighborhood state; every time, H_t is updated. After that, a softmax layer is used to distribute the state of Bi-LSTM into a probability distribution for 62

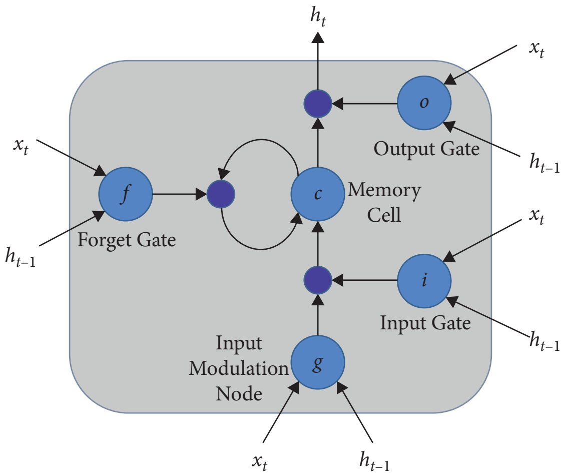


Figure 5: Block diagram of LSTM.

classes. The one extra class is used for finding the space between two words. Finally, the feature sequence is transformed into the sequence of probability P.

Now, finally, the sequence of probability P must be transferred into a text string. In the proposed method, a CTC-based decoder is used for this purpose, and it is used for the classification of sequential text. For each time t, CTC calculates the probability distribution over the alphabet of possible characters, and it gives the characters which have the highest probability as output. The CTC object function is defined as follows:

$$O = -\sum_{(i_s,t_s)\varepsilon S} \ln P(t_s|i_s). \tag{2}$$

It is the negative log probability function of the network which correctly labels the training dataset. Here, D represents the training dataset which consists of input and target sequence, and it is represented by $(i_s,\ t_s)$. Conditional probability is represented by $P(t_s|i_s)$. The target O must be minimized, and it is equivalent to maximize the $P(t_s|i_s)$. The object function is directly connected to the output of the Bi-LSTM layer, and it is defined as

$$P(i_s|t_s) = \sum_{\pi:B(\pi)=ts} P(\pi|p). \tag{3}$$

The model is trained using gradient descent and backpropagation. In the above equation, B is used for removing the repeated and space labels. Suppose the sequence is B (c–c–f-), then the final output will be B(ccf). Once the model training is done, sequencing labeling aims to find the optimal path with max probability using Bi-LSTM.

4. Results

The experiments are performed on MSRATD 50 dataset, SVHN dataset, vehicle number plate dataset, SVT dataset, and random datasets to verify the performance of the proposed methodology. The experiments are performed on NVIDIA GTX 1650/60 Hz, 16 GB RAM, core-i7 10^{th} generation processor. 80% of the dataset images are used for training purposes to train and test the model, and 20% of the images are used for testing purposes. Hyperparameters used in the architecture are described in Table 2.

Existing methods are compared with the proposed method's accuracy. We used accuracy, recall, precision, and F1-score in evaluating the proposed method. The accuracy is defined as the percentage of correctly classified instances. It is used to calculate the proportion of true positive and true negative for multiclass classification problems. The formula for calculating accuracy, precision, recall, and F1-score is given as follows:

$$\begin{aligned} \text{accuracy} &= \frac{(\text{TP}+\text{TN})}{(\text{TP}+\text{TN}+\text{FP}+\text{FN})}, \\ \text{precision} &= \frac{\text{TP}}{(\text{TP}+\text{FP})}, \\ \text{recall} &= \frac{\text{TP}}{(\text{TP}+\text{FN})}, \\ F1_{\text{Score}} &= 2.\left(\frac{\text{precision.recall}}{\text{precision}+\text{recall}}\right). \end{aligned} \tag{4}$$

Here, TP = true positive, TN = true negative, FP = false positive, and FN = false negative.

4.1. MSRATD 50 Dataset. MSRA TD dataset is one of the benchmark datasets for text recognition, and this dataset contains 3000 images of 32 × 32 sizes. The dataset is challenging and noisy, and it contains English and Chinese text. The images in the dataset have blur and noisy background. The sample input images and the recognized text and its bounded box x-coordinate, y-coordinate, width, and height are listed in Figure 6. The proposed system had shown the accuracy of 95.22% and recall of 85.73%. The precision is 94.15%, and F-score is 87.09%. The metrics of the SVHN dataset are shown in Table 3.

4.2. SVHN Dataset. SVHN dataset (Street View House Numbers) is the dataset that contains 600,000 digital numbers captured from various angles from various houses of Google street view. All images are of size 32 × 32. The images are blurred and have images captured from a different angle. The obtained accuracy is 92.25%, and recall, precision, and F-score are, 79.03%, 92.49%, and 89.80%, respectively. The sample input images and the recognized text along with its bounded box x-coordinate, y-coordinate, width, and height are listed in Figure 7. The metrics of the SVHN dataset are shown in Table 4.

4.3. Vehicle Number Plate Dataset. We collected sample images from UFPR-ALPR dataset and tested them on our proposed method. The proposed method has shown an accuracy of 96.69%. The recall, precision, and F-score values are 93.11%, 86.77%, and 90.01%, respectively. The sample input images and the recognized text along with its bounded

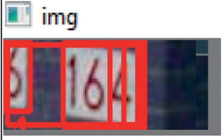
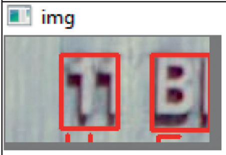
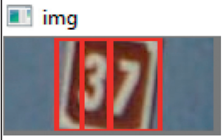
Input Image	Status of Recognition	Result
	Accurate	['1', '0', '10', '15', '46', '0'] ['6', '34', '2', '78', '45', '0'] ['1', '2', '3', '77', '44', '0'] [‘6', '69', '0', '81', '46', '0']
	Accurate	['N', '34', '9', '69', '55', '0'] ['B', '90', '8', '126', '55', '0']
	Accurate	['3', '30', '0', '61', '56', '0'] ['7', '62', '0', '91', '56', '0']

Figure 6: Text recognition on MSRA dataset.

Table 3: Metrics of MSRATD 50 dataset.

Sr. No.	Parameters	Output
1	Precision	94.15
2	Recall	85.73
3	F-score	87.09
4	Accuracy	95.22

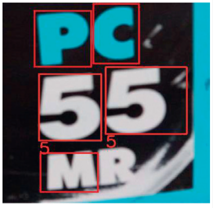
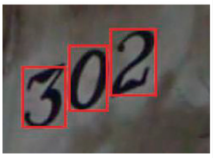
Input Image	Status of Recognition	Result
	Partially Accurate	['P', '1034', '838', '1074', '864', '0'] ['C', '1179', '797', '1281', '864', '0'] ['5', '610', '325', '722', '454', '0'] ['5', '730', '338', '876', '467', '0'] ['M', '401', '227', '414', '240', '0']
	Accurate	['3', '906', '412', '1130', '539', '0'] ['0', '906', '412', '1130', '539', '0'] ['2', '584', '401', '964', '540', '0']

Figure 7: Text recognition on SVHN dataset.

Table 4: Metrics of SVHN dataset.

S. no.	Parameters	Output
1	Precision	92.49
2	Recall	79.03
3	F-score	89.80
4	Accuracy	92.25

box *x*-coordinate, *y*-coordinate, width, and height are listed in Figure 8. The metrics of the SVHN dataset are shown in Table 5.

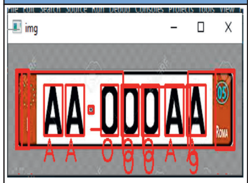
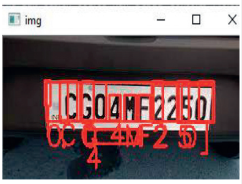
Input Image	Status of Recognition	Result
	Accurate	['A', '59', '39', '90', '89', '0'] ['A', '95', '39', '126', '89', '0'] ['O', '154', '39', '183', '89', '0'] ['O', '188', '39', '217', '89', '0'] ['O', '222', '39', '251', '89', '0'] ['A', '261', '39', '292', '89', '0'] ['A', '297', '39', '329', '89', '0']
	Accurate	['C', '72', '74', '94', '114', '0'] ['G', '100', '72', '115', '114', '0'] ['O', '98', '74', '131', '113', '0'] ['4', '135', '74', '149', '100', '0'] ['M', '154', '73', '169', '100', '0'] ['F', '172', '73', '187', '100', '0'] ['2', '189', '72', '203', '114', '0'] ['2', '191', '73', '223', '111', '0'] ['5', '221', '72', '236', '114', '0'] ['0', '227', '72', '268', '110', '0']

Figure 8: Text recognition on vehicle number plates dataset.

Table 5: Metrics of UFPR-ALPR dataset.

S. no.	Parameters	Output
1	Precision	93.11
2	Recall	86.77
3	F-score	90.01
4	Accuracy	96.69

4.4. SVT Dataset. SVT dataset is one of the challenging datasets where the images were taken from Google street view. The images in the dataset are high variability and meager resolution. The proposed method has shown an accuracy of 94.58%. The recall, precision, and F-score values are 84027%, 91.86%, and 88.49%, respectively. The sample input images and the recognized text along with its bounded box *x*-coordinate, *y*-coordinate, width, and height are listed in Figure 9. The metrics of the SVT dataset are shown in Table 6. The dataset results are partially accurate as the images are the 3D projections of original image.

4.5. Random Dataset. We collected random text from the Internet to check the proposed method accuracy. The images were collected with a plain background and colored background. The accuracy of the proposed work is 98.12% as the samples contained the text and numbers without any noise in them. The obtained recall, precision, and F-score are 98.19%, 90.18%, and 97.07%, respectively. The sample input images and the recognized text along with its bounded box *x*-coordinate, *y*-coordinate, width, and height are listed in Figure 10. The metrics of the random dataset are shown in Table 7.

4.6. Comparison Analysis of the Proposed Work. We analyzed the proposed method on four benchmark datasets MSRA TD dataset, SVHN dataset, UFPR-ALPR dataset, SVT dataset, and random text collected from the Internet

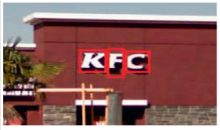
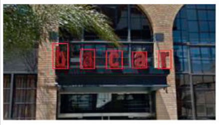
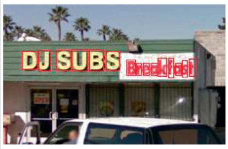
Output Image	Status of Recognition	Result
	Accurate	['K', '112', '499', '135', '542', '0'] ['F', '136', '467', '175', '541', '0'] ['C', '156', '463', '171', '546', '0']
	Accurate	['b', '515', '375', '546', '419', '0'] ['a', '550', '374', '617', '419', '0'] ['c', '634', '374', '669', '419', '0'] ['a', '652', '344', '679', '450', '0'] ['r', '676', '374', '714', '418', '0']
	Accurate	['D', '42', '469', '63', '500', '0'] ['J', '65', '470', '87', '500', '0'] ['S', '89', '470', '112', '501', '0'] ['U', '121', '511', '123', '514', '0'] ['B', '1', '447', '3', '450', '0'] ['S', '23', '446', '34', '461', '0'] ['B', '36', '446', '47', '461', '0'] ['r', '46', '446', '56', '461', '0'] ['e', '58', '447', '67', '462', '0'] ['a', '69', '447', '79', '462', '0'] ['k', '652', '344', '679', '450', '0'] ['f, '676', '374', '714', '418' , '0'] ['a', '722', '373', '756', '417', '0'] ['s', '796', '268', '826', '450', '0'] ['t', '1255', '367', '1285', '411', '0']

FIGURE 9: Text recognition on SVT dataset.

TABLE 6: Metrics of SVT dataset.

S. no.	Parameters	Output
1	Precision	91.86
2	Recall	84.27
3	F-score	88.49
4	Accuracy	94.58

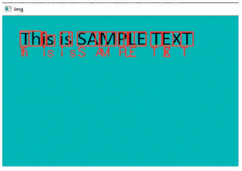
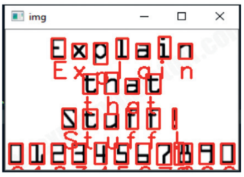
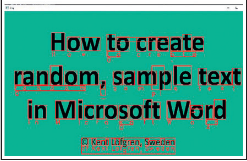
Input Image	Status of Recognition	Result
 This is SAMPLE TEXT	Accurate	['T', '42', '671', '62', '707', '0'] ['h', '42', '671', '90', '707', '0'] ['i', '93', '671', '101', '705', '0'] ['s', '103', '671', '121', '697', '0'] ['i', '134', '671', '161', '705', '0'] ['s', '154', '671', '161', '705', '0'] ['S', '173', '671', '223', '705', '0'] ['A', '215', '671', '239', '705', '0']
 Explain that Stuff! 0123456789	Accurate	['E', '55', '143', '71', '169', '0'] ['x', '80', '143', '97', '162', '0'] ['p', '106', '136', '122', '163', '0'] ['l', '133', '143', '147', '169', '0'] ['a', '157', '143', '173', '162', '0'] ['i', '184', '143', '198', '171', '0'] ['n', '208', '143', '224', '162', '0'] ['t', '93', '98', '110', '123', '0'] ['h', '119', '98', '135', '125', '0'] ['a', '144', '98', '160', '117', '0'] ['t', '169', '98', '186', '123', '0']
 How to create random, sample text in Microsoft Word © Kent Löfgren, Sweden	Accurate	['H', '245', '527', '313', '615', '0'] ['o', '328', '525', '393', '594', '0'] ['w', '401', '527', '499', '592', '0'] ['t', '534', '526', '577', '608', '0'] ['o', '585', '525', '649', '594', '0']

FIGURE 10: Text recognition on random/self-dataset.

Table 7: Metrics of random/self-dataset.

S. no.	Parameters	Output
1	Precision	90.18
2	Recall	98.19
3	F-score	97.07
4	Accuracy	98.12

Table 8: Metrics of various datasets used in the proposed system.

S. no.	Parameters	MSRATD 50	UFPR-ALPR	SVHN	SVT	Random/self
1	Precision	94.15	93.11	92.49	91.86	90.18
2	Recall	85.73	86.77	79.03	84.27	98.19
3	F-score	87.09	90.01	89.80	88.49	97.07
4	Accuracy	95.22	96.69	92.25	94.58	98.12

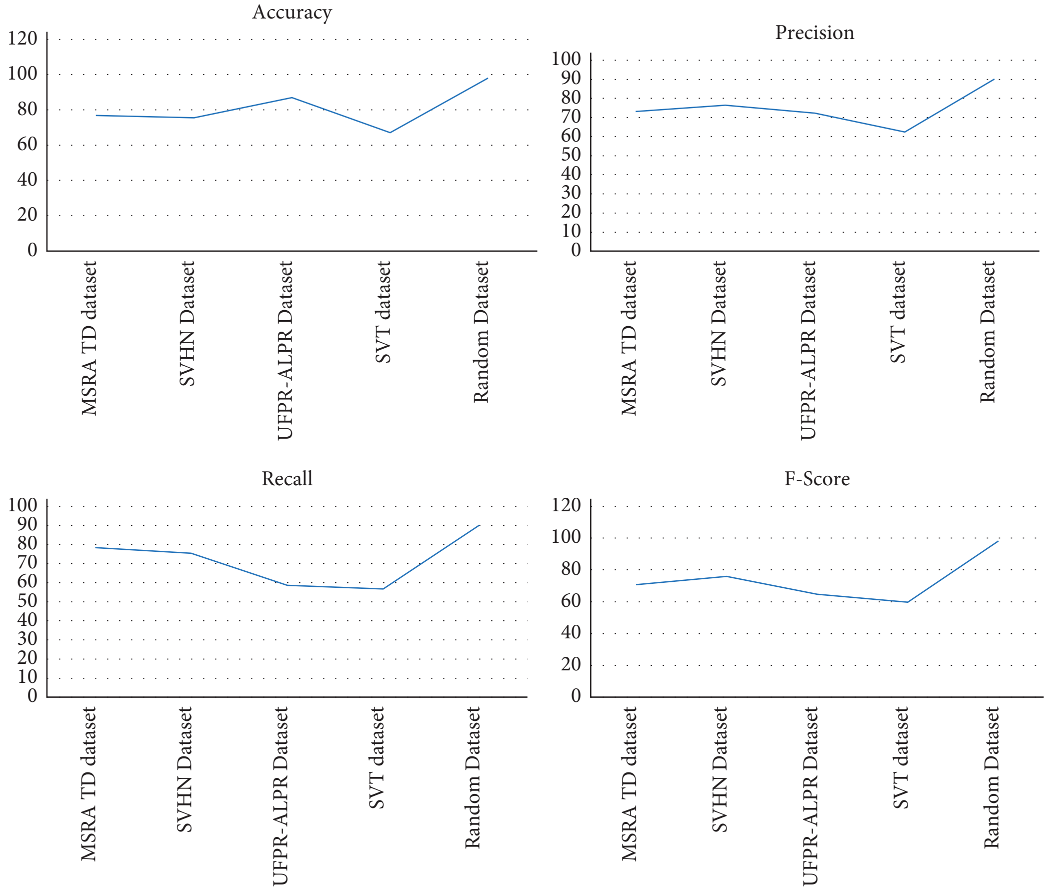


Figure 11: Overall text recognition on all dataset.

and phone camera. The datasets are challenging datasets in various aspects. MSRA TD dataset is a tiny dataset of street view door numbers dataset that contains only images of house numbers. The dataset contains a blurred dataset; the proposed system can recognize the number with 95.22% accuracy. The SVHN dataset is another challenging blurred dataset containing both text and numbers with different backgrounds and fonts, and the proposed system has shown an accuracy of 92.25%. The UFPR-ALPR is a vehicle number plate dataset with different backgrounds. The proposed system has shown an accuracy of 96.69%. We considered the SVT dataset, a Google street view dataset with heavy background fluctuations and unclear text with various fonts and 3D reflections. The proposed system has shown accurate results with 94.58% of accuracy. We collected random datasets from the Internet and few images captured from Samsung mobile phones with minimal resolution. The proposed

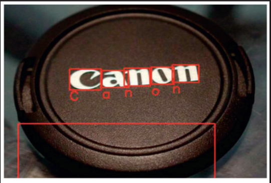
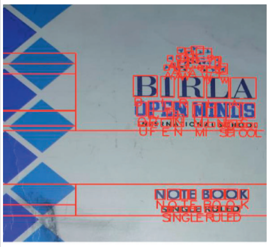
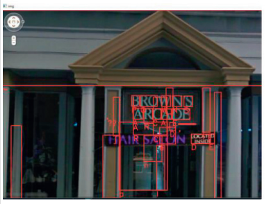
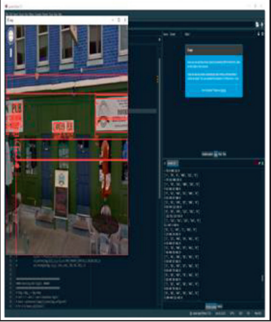
	Partially Inaccurate	['C', '505', '421', '576', '470', '0'] ['n', '628', '430', '686', '472', '0'] ['o', '689', '434', '740', '475', '0'] ['n', '736', '439', '796', '480', '0'] ['~', '390', '0', '832', '340', '0']
	Partially Inaccurate	['E', '324', '230', '343', '249', '0'] ['V', '346', '229', '373', '249', '0'] ['E', '376', '228', '396', '248', '0'] ['R', '401', '228', '426', '273', '0'] ['Y', '422', '227', '450', '268', '0'] ['D', '453', '227', '481', '272', '0'] ['A', '473', '227', '510', '264', '0'] ['Y', '510', '226', '535', '246', '0'] ['P', '548', '218', '572', '246', '0'] ['R', '573', '227', '596', '246', '0'] ['O', '597', '228', '625', '256', '0'] ['T', '626', '230', '651', '249', '0']
	Partially Accurate	['P', '515', '375', '546', '419', '0'] ['O', '550', '374', '617', '419', '0'] ['W', '634', '374', '669', '419', '0'] ['T', '652', '344', '679', '450', '0'] ['N', '676', '374', '714', '418', '0'] ['S', '722', '373', '756', '417', '0'] ['|', '796', '268', '826', '450', '0'] ['i', '1255', '367', '1285', '411', '0'] ['M', '1324', '402', '1327', '410', '0']
	Inaccurate	['P', '42', '469', '63', '500', '0'] ['U', '65', '470', '87', '500', '0'] ['B', '89', '470', '112', '501', '0'] ['|', '121', '511', '123', '514', '0'] ['&', '1', '447', '3', '450', '0'] ['C', '23', '446', '34', '461', '0'] ['A', '36', '446', '47', '461', '0'] ['T', '46', '446', '56', '461', '0'] ['E', '58', '447', '67', '462', '0'] ['R', '69', '447', '79', '462', '0']

Figure 12: Incorrect text recognition on all dataset.

Table 9: Metrics of various datasets used in the proposed system.

S. no.	Parameters	Ref. [22]	Ref. [19]	Ref. [18]	Ref. [14]	Proposed work (average)
1	Precision	82	—	91.43	90	92.15
2	Recall	72	79.16	94.52	91	83.50
3	F-score	77	85.71	92.20	91	88.56
4	Accuracy	—	93.44	—	—	93.83

system has shown an accuracy of 98.12%. The analysis is given in Table 8, and the corresponding graphs are plotted in Figure 11.

For some of the images, results are partially accurate. As we can see in Figure 12, the images which were captured from the long distance or the orientation of the image are different. In those cases, model is able to detect the text partially. Proposed work is compared with the existing state of art methods, and according to the analysis, precision and accuracy is improved. The average

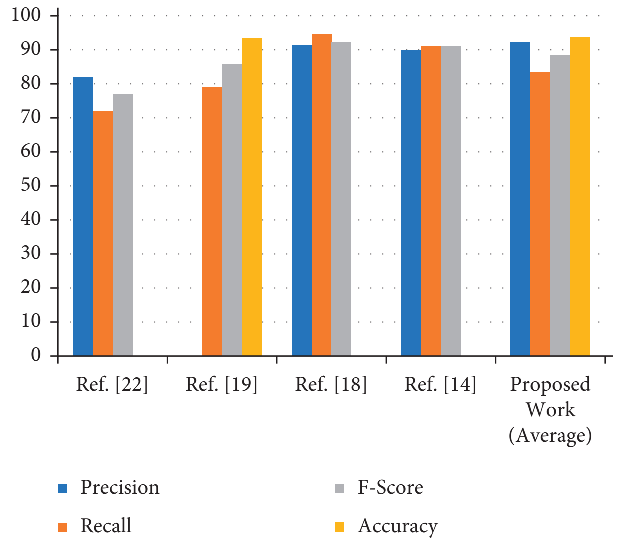


Figure 13: Comparative analysis of the proposed work with existing techniques.

recognition rate of proposed methodology and the comparison with state of art methods are shown in the Table 9 and Figure 13.

5. Conclusion and Future Scope

This research paper proposed a novel approach for scene text recognition that integrates bidirectional LSTM and deep convolution neural networks. In the proposed method, first, the contour of the image is identified, and then, it is feed into the CNN. CNN is used to generate the ordered sequence of the features from the contoured image. The sequence of features now coded using the Bi-LSTM. Bi-LSTM is a handy tool for extracting the features from the sequence of words. Thus, this paper combines the two powerful mechanisms for extracting the features from the image and contour-based input image making the recognition process faster, which makes this technique better compared to existing methods. The proposed method is evaluated on four benchmark datasets MSRA TD dataset, SVHN dataset, UFPR-ALPR dataset, SVT dataset, and random text collected from the Internet and phone camera. According to the quantitative and qualitative analysis, this approach is more promising in terms of accuracy and precision rate. The datasets are challenging datasets in various aspects. The proposed method can able to detect the text from the different backgrounds, unclear text, blurred images, different font size, and different orientation. In future, a better approach can be introduced which can deal with heavy background fluctuations and different 3D reflections.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] K. Hamad and M. Kaya, “Detailed analysis of optical character recognition technology,” *International Journal of Applied Mathematics, Electronics and Computers*, vol. 4, no. Special Issue-1, p. 244, 2016.

[2] Q. Ye and D. Doermann, “Text detection and recognition in imagery: a survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1480–1500, 2015.

[3] X. Chen and A. L. Yuille, “Detecting and reading text in natural scenes,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 366–373, Washington, DC, USA, June 2004.

[4] J. Gllavata, R. Ewerth, and B. Freisleben, “Text detection in images based on unsupervised classification of high-frequency wavelet coefficients,” in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004*, vol. 01, pp. 425–428, Cambridge, UK, August 2004.

[5] O. Alsharif and J. Pineau, “End-to-end text recognition with hybrid HMM max out models,” in *Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014*, Workshop Track Proceedings, Ban, AB, Canada, April 2014.

[6] R. Minetto, N. Thome, M. Cord et al., “Text detection and recognition in urban scenes,” in *Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 227–234, Barcelona, November 2011.

[7] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, “End-to-end text recognition with convolutional neural networks,” in *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012*, pp. 3304–3308, Tsukuba, Japan, November 2012.

[8] M. Zhao, S. Li, and J. Kwok, “Text detection in images using sparse representation with discriminative dictionaries,” *Image and Vision Computing*, vol. 28, no. 12, pp. 1590–1599, 2010.

[9] R. Walha, F. Drira, F. Lebourgeois, C. Garcia, and A. M. Alimi, “Sparse coding with a coupled dictionary learning approach for textual image super-resolution,” in *Proceedings of the ICPR*, pp. 4459–4464, Washington, DC, USA, August 2014.

[10] R. Walha, F. Drira, F. Lebourgeois, C. Garcia, and A. M. Alimi, “Resolution enhancement of textual images via multiple coupled dictionaries and adaptive sparse representation selection,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 18, no. 1, pp. 87–107, 2015.

[11] R. Walha, F. Drira, A. M. Alimi, F. Lebourgeois, and C. Garcia, “A sparse coding-based approach for the resolution enhancement and restoration of printed and handwritten textual images,” in *Proceedings of the ICFHR*, pp. 696–701, Crete, Greece, September 2014.

[12] S. Y. Arafat and M. J. Iqbal, “Urdu-text detection and recognition in natural scene images using deep learning,” *IEEE Access*, vol. 8, pp. 96787–96803, 2020.

[13] A. A. Chandio, M. Asikuzzaman, and M. R. Pickering, “Cursive character recognition in natural scene images using a multilevel convolutional neural network fusion,” *IEEE Access*, vol. 8, pp. 109054–109070, 2020.

[14] Y. Qin and Z. Zhang, “November. Summary of scene text detection and recognition,” in *Proceedings of the 2020 15th IEEE Conference on Industrial Electronics and Applications*

(ICIEA), pp. 85–89, IEEE, Kristiansand, Norway, November 2020.

[15] J.-L. Wu, Y. He, L.-C. Yu, and K. R. Lai, "Identifying emotion labels from psychiatric social texts using a bi-directional LSTM-CNN model," *IEEE Access*, vol. 8, pp. 66638–66646, 2020.

[16] S. Y. Arafat and M. J. Iqbal, "Two stream deep neural network for sequence-based Urdu ligature recognition," *IEEE Access*, vol. 7, pp. 159090–159099, 2019.

[17] S. Jaf and C. Calder, "Deep learning for natural language parsing," *IEEE Access*, vol. 7, pp. 131363–131373, 2019.

[18] M. A. Panhwar, K. A. Memon, A. Abro, Z. Deng, S. A. Khuhro, and S. Memon, "Signboard detection and text recognition using artificial neural networks," in *Proceedings of the 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pp. 16–19, IEEE, Beijing, China, July 2019.

[19] Y. M. Su, H. W. Peng, K. W. Huang, and C. S. Yang, "November. Image processing technology for text recognition," in *Proceedings of the 2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pp. 1–5, IEEE, Kaohsiung, Taiwan, November 2019.

[20] L.-Q. Zuo, H.-M. Sun, Q.-C. Mao, R. Qi, and R.-S. Jia, "Natural scene text recognition based on encoder-decoder framework," *IEEE Access*, vol. 7, pp. 62616–62623, 2019.

[21] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016.

[22] 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," *IEEE Access*, vol. 5, pp. 3193–3204, 2017.

[23] X. Bai, C. Yao, and W. Liu, "Strokelets: a learned multi-scale mid-level representation for scene text recognition," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2789–2802, 2016.

[24] M. Yang, H. Zheng, B. Xiang, and J. Luo, "Cost-effective adversarial attacks against scene text recognition," in *Proceedings of the 25th IEEE International Conference on Pattern Recognition (ICPR)*, pp. 2368–2374, Milan, Italy, January 2021.

[25] A. Zhu, H. Du, and S. Xiong, "Scene text detection with selected anchors," in *Proceedings of the 25th IEEE International Conference on Pattern Recognition (ICPR)*, pp. 6608–6615, Milan, Italy, January 2021.

[26] ZiL. Hu, X. Wu, and J. Yang, "TCATD: text contour attention for scene text detection," in *Proceedings of the 25th IEEE International Conference on Pattern Recognition (ICPR)*, pp. 1083–1088, 2021.

[27] S. Rani, K. Lakhwani, and S. Kumar, "Three dimensional wireframe model of medical and complex images using cellular logic array processing techniques," in *Proceedings of the 12th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2020). SoCPaR 2020*, A. Abraham, Y. Ohsawa, N. Gandhi et al., Eds., vol. 1383, Springer, Berlin, Germany, 2021, Advances in Intelligent Systems and Computing.

[28] A. Jain, A. Kumar, and S. Sharma, "Comparative design and analysis of mesh, torus and ring NoC," *Procedia Computer Science*, vol. 48, pp. 330–337, 2015.

[29] D. Ghai, H. K. Gianey, A. Jain, and R. S. Uppal, "Quantum and dual-tree complex wavelet transform-based image watermarking," *International Journal of Modern Physics B*, vol. 34, no. 04, Article ID 2050009, 2020.

[30] N. Agrawal, A. Jain, and A. Agarwal, "Simulation of network on chip for 3D router architecture," *International Journal of Recent Technology and Engineering*, vol. 8, pp. 58–62, 2019.

[31] A. K. Agarwal and A. Jain, "Synthesis of 2D and 3D NoC mesh router architecture in HDL environment," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 11, no. 4, pp. 2573–2581, 2019.

[32] A. Jain, A. K. Gahlot, R. Dwivedi, A. Kumar, and S. K. Sharma, "Fat tree NoC design and synthesis," in *Intelligent Communication, Control and Devices*, pp. 1749–1756, Springer, Berlin, Germany, 2018.