Multimedia Tools and Applications (2024) 83:7135–7158
https://doi.org/10.1007/s11042-023-15633-1

Product identification in retail stores by combining faster r-cnn and recurrent neural network

Rajib Ghosh[1]

Received: 10 March 2022 / Revised: 14 July 2022 / Accepted: 22 April 2023 /
Published online: 6 June 2023

Abstract
Identifying various products on the racks of supermarkets is a very easy task for human beings. But, when the same identification task is given to a computer vision based system, it poses a huge challenge for it. This article proposes a method to identify various products on the racks of supermarkets by detecting the text blocks in product labels using Faster R-CNN with more than one region proposal networks (RPNs) and then recognizing the text using Recurrent Neural Network (RNN) classifier. To detect the varying sized text blocks in product labels, several diverse sized RPNs have been proposed in this investigation. The traditional Faster R-CNN creates regions-of-interest (ROIs) using a sole RPN and so remains unable to detect the labels with diverse sized text blocks accurately. The novelty of this work lies in proposing more than one diverse sized RPNs in the traditional Faster R-CNN to detect the text blocks in the product labels and recognizing the text using RNN classifier. Three different public datasets, namely GroZi-120, Grocery Products, and Grocery Dataset have been used to assess the performance of this work and it outperforms state-of-the-art results on text block detection. The proposed system has provided the text recognition accuracies of 99.18%, 99.21%, and 99.12% for GroZi-120, Grocery Products, and Grocery Dataset respectively.

Keywords Shopping mall product identification · Text block detection · Text recognition · Faster R-CNN · Several RPNs · RNN.

1 Introduction

Investigations on developing computer vision based automated systems for identifying various products on the racks of supermarkets/retail stores have gained momentum in the recent past due to the vast applications of these systems in this digital era. The main purposes of these machine vision based systems are to create an inventory of the products in the supermarket/retail store from the images of the racks and to verify that the products are displayed as

✉ Rajib Ghosh
rajib.ghosh@nitp.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology Patna, Patna 800005, India

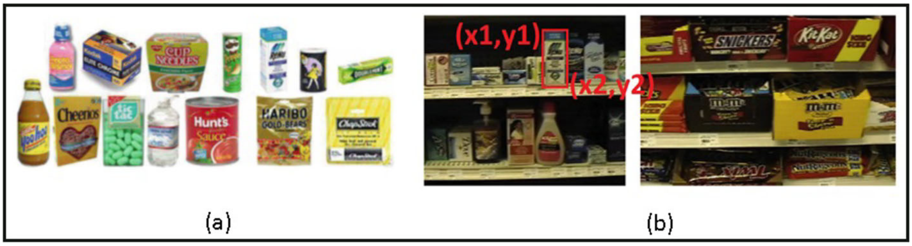


Fig. 1 Product and rack images from GroZi-120 dataset [10]: (a) few product images generally used for marketing, (b) few rack images in which products are to be located and recognized. $(x1, y1)$ and $(x2, y2)$ denote the spatial coordinates of upper-left and bottom-right corners of the detected bounding box respectively.

per the plan of merchandise or not. These systems also provide the shopping assistance to the users. With the availability of more advanced and powerful computer vision tools, researchers are developing more powerful systems in this domain. But, this identification task poses a huge challenge for any computer vision based system due to the following factors:

1. Multiple orientations of logo text, complicated ways of writing the logo text, etc.
2. The supermarket racks generally remain messy and are not arranged in an uniform manner.
3. The product images fed to the vision systems are most often captured using separate cameras emanating in varying distributions of image intensities.
4. Due to varying parameters of images, aspect ratio of the product packet is mapped with separate pixel resolutions for product and rack images. This difference is illustrated in Fig. 1.
5. The text in product labels come with various shapes and sizes.

A few works on machine vision based product identification systems are already present in the literature and these studies have relied upon both machine learning (ML) [1–3, 7] and non-ML [10–12] based methods. In ML based methods, the use of different shallow (discriminative random forests [1], support vector machine (SVM) [2, 3], Naive Bayes classifier [4]) as well as deep (deep convolutional neural network (CNN) [6], AlexNet [7]) ML methods have been reported in the literature. But, the utilization of Faster R-CNN model with multiple region proposal networks (RPNs) to detect the product labels and then recognizing the text in product labels using Recurrent Neural Network (RNN) are still unexplored in this problem area. The Faster R-CNN architecture was first published by Ren et al. [8] to detect objects from the scene images. Traditional Faster R-CNN can detect the objects more accurately in comparison to other CNN based architectures due to employing the region proposal algorithm. Regions-of-interest (ROIs) are produced in the traditional Faster R-CNN by a sole RPN and so it is not always able to correctly detect the diverse sized texts present

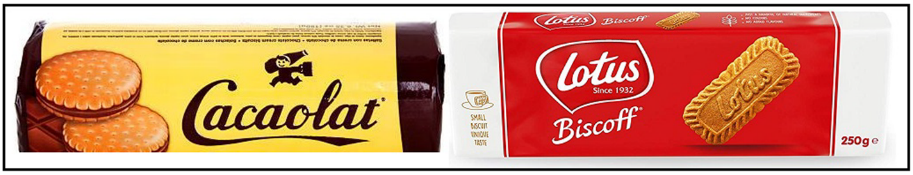


Fig. 2 Diverse sized texts present in the labels of two different products.

in the product labels as shown in Fig. 2. This article proposes a novel approach of creation of ROIs in Faster R-CNN by incorporating more than one RPNs in the traditional Faster R-CNN architecture. The dimensions of these RPNs are varying in nature and it allows this system to detect the varying sized text blocks in the product labels. The text present in the detected text blocks have then been recognized using RNN classifier, a deep learning (DL) network. Recognition of text in the product labels using RNN classifier is another novelty of the proposed system. RNN has two different versions—long-short term memory (LSTM) [9] and bidirectional long-short term memory (BLSTM) [9]. RNN can store a long sequence of symbols over a longer duration through these two versions. Apart from this, BLSTM can store any sequence in forward as well as backward directions. Text in product labels consists of a sequence of few characters. This work can recognize any character in the text after having the knowledge of characters occurring both before and after it through exploiting these storage capabilities of LSTM and BLSTM versions of RNN. Other shallow ML techniques do not have this power and due to which the present system produces better text recognition results (Section 5 may be referred) in comparison to various shallow ML methods.

The major contributions of this work are as follows:

1. The use of several RPNs in the Faster R-CNN to detect the product labels from the rack images of retail stores. The novelty of this work lies in proposing more than one diverse sized RPNs in the traditional Faster R-CNN to detect the text blocks in the product labels.
2. The recognition of text in the product labels.
3. The use of RNN classifier to recognize the text in the detected text blocks in various products. Recognition of text in the product labels using RNN classifier is another novelty of the proposed system.

The rest of the paper is organized as follows: Section 2 discusses the related works. The details of the datasets used in this work are discussed in section 3. Section 4 discusses the proposed method of product identification in retail stores. Experimental results and analysis of the results are discussed in Section 5. Section 6 presents the conclusion of this paper and possible research works that can be carried out in future in this regard.

2 Literature Survey

Few studies in this problem area are available in the literature. Some of the important studies are discussed below in brief.

George et al. [1] presented one computer vision based product identification system utilizing the ML technique discriminative random forests. A multi-label image classification approach using dense SIFT features [20] of product images was proposed in this study. The method subdivided the rack image into several blocks. Varol et al. [2] presented a SVM based machine vision system to identify various products on the racks of supermarket. The products were detected on the racks using an object detector and then recognized using SVM classifier after extracting the SIFT features from the product images. In another [3] use of SVM classifier for product recognition on the racks, two separate modules were implemented. In the first module, the brand name was predicted by detecting the text region of product label and then recognizing those detected texts using OCR. Cleveland et al. [4] developed a navigating robot to assist visually impaired shoppers. Shelves were located using morphological operations. 3D point cloud of rack image was utilized to locate the products. Finally, products were recognized using Naive Bayes classifier. Karlinsky et al. [5] presented a three phase method for product identification. In the first phase, primary detecting boxes were generated using a probabilistic inference model based on the SIFT features extracted from the product images. In the second phase, the primary detecting boxes from the first phase were classified using a re-trained CNN model. In the third and last phase, first two phases were integrated using the KLT tracker [21] to determine the final detecting boxes. Zientara et al. [6] presented a smart glass based product detection system that locates the products in the rack images through deriving the saliency map of the rack images. The products were then recognized using deep CNN. The use of the DL network AlexNet to recognize products from the rack images has also been reported in the literature [7]. In this study, a scheme using Harris corner detector [22] and 3D color histograms of rack images was presented to find out the the possible locations of the products in the rack images before recognizing them. Merler et al. [10] presented a machine vision based system on product recognition from rack images that has compared three different product localization approaches. One of those was the sliding window based approach that determined the possible locations of the products in the rack images using sliding window of different scales. Marder et al. [11] presented a vision based two phase method for product identification on the supermarket shelves. In the first phase, products were detected in the rack images using point based vote map [23], sliding window based Histogram of Oriented Gradients (HOG) [24], and Bag of Words (BOW) [25] methods. In the second phase, the detected products were recognized by solving the visual ambiguity using saliency map. Saran et al. [12] proposed a method that first detected the shelves in the images using Sobel operator followed by Hough transform. The products were then located in the detected shelves and recognized in two successive steps—(i) SURF feature [26] correspondence and (ii) false positive elimination. Liu et al. [13] presented pattern based features recurring patterns for product identification in rack images. In another investigation [14], saliency maps were utilized to locate the products in the rack images. Feature vector of any image was generated by combining all SURF descriptors and color histograms. Yörük et al. [15] presented a method of detecting various products in the rack images by estimating the poses of the products. The pose was determined using Hough voting scheme. The products were recognized by constructing a k-d tree of SURF features of the products. Zhang et al. [16] presented a dual layer density estimation scheme for product identification. In one recent investigation [17], probable matches of product images were performed using SIFT and Hough transform. Sub-graph isomorphism was then applied to eliminate the false matches in observed output. In another recent work [18], Hu et al. proposed the deep CNN architecture DiffNet for product identification. Umer et al. [19] proposed a cosmetic product recognition method using various ML techniques. Several feature extraction methods such as structural and statistical texture analysis have been employed to extract the discriminating features from the product images.

Table 1 discusses the advantages and limitations of the aforementioned existing works. The limitations presented in Table 1 of various existing studies can be overcome by the proposed system as it employs Faster R-CNN with several diverse sized RPNs to detect the text blocks in the product labels. In general, Faster R-CNN can detect any object more accurately in comparison to other object detection methods because it employs the region proposal algorithm to detect the objects.

Table 1 Advantages and limitations of the existing related works

Reference	Advantages	Limitations
George et al. [1]	Block based method—simple and easy to implement	Selecting the number and size of overlapping or non-overlapping blocks may be wrong
Varol et al. [2]	Can generate the detecting box for partially occluded products	May fail to locate the product if intensity distributions of train and test images differ
George et al. [3]	Block based method—simple and easy to implement	Selecting the number and size of overlapping or non-overlapping blocks may be wrong
Cleveland et al. [4]	Geometric transformation based method –fast and appropriate for real time environment	Dependent on assumptions of number of key points
Karlinsky et al. [5]	Can generate the detecting box for partially occluded products	May fail to locate the product if intensity distributions of train and test images differ
Zientara et al. [6]	Saliency based method—minimizes false detection, pays attention to rotation and scaling of products	Usually fails for partially-occluded products in the image
Franco et al. [7]	Saliency based method—minimizes false detection, pays attention to rotation and scaling of products	Usually fails for partially-occluded products in the image
Marder et al. [11]	Block based method—simple and easy to implement	Selecting the number and size of overlapping or non-overlapping blocks may be wrong
Saran et al. [12]	Block based method—simple and easy to implement	Selecting the number and size of overlapping or non-overlapping blocks may be wrong
Liu et al. [13]	User-in-the-loop method— able to detect a novel product	Only classification or recognition performances are judged using this approach
Winlock et al. [14]	Saliency based method—minimizes false detection, pays attention to rotation and scaling of products	Usually fails for partially-occluded products in the image
Yörök et al. [15]	Geometric transformation based method —fast and appropriate for real time environment	Dependent on assumptions of number of key points
Tonioni et al. [17]	Geometric transformation based method —fast and appropriate for real time environment	Dependent on assumptions of number of key points

3 Dataset Description

Three different public datasets, namely GroZi-120, Grocery Products, and Grocery Dataset have been used to test the performance of the present system. The datasets are discussed below briefly.

1. GroZi-120: GroZi-120 dataset is the very first published benchmark dataset of 120 grocery products. In this dataset, the product and rack images have been collected in completely different setups. In the present investigation, only the rack images have been considered. Rack images were captured from retail stores through videos. A total of 29 videos with overall duration of 30 minutes are present in this dataset. In this dataset, the rack images have been called as in situ images. The dataset contains 11194 in situ images. The total images have been fragmented into train and test sets in 3:1 ratio.
2. Grocery Products [1]: This dataset was developed for classification and detection of objects. Product images were collected from the internet. The rack images were grabbed by a mobile phone from the real-life retail store environments from different viewing angles and under varying lighting conditions. The present investigation has considered only the rack images. This dataset contains 680 rack images. These 680 images have been fragmented into train and test sets in 3:1 ratio. A sample image of racks from this dataset is shown in Fig. 3.
3. Grocery Dataset [2]: In Grocery dataset, the product images were captured with four different types of cameras. Rack images were captured from 40 different grocery stores with these cameras at varying distances from the racks. The number of products in a rack image varies from 2 to 137. A sample image of racks from this dataset is shown in Fig. 4. This dataset contains 354 rack images. The total images have been fragmented into train and test sets in 3:1 ratio.

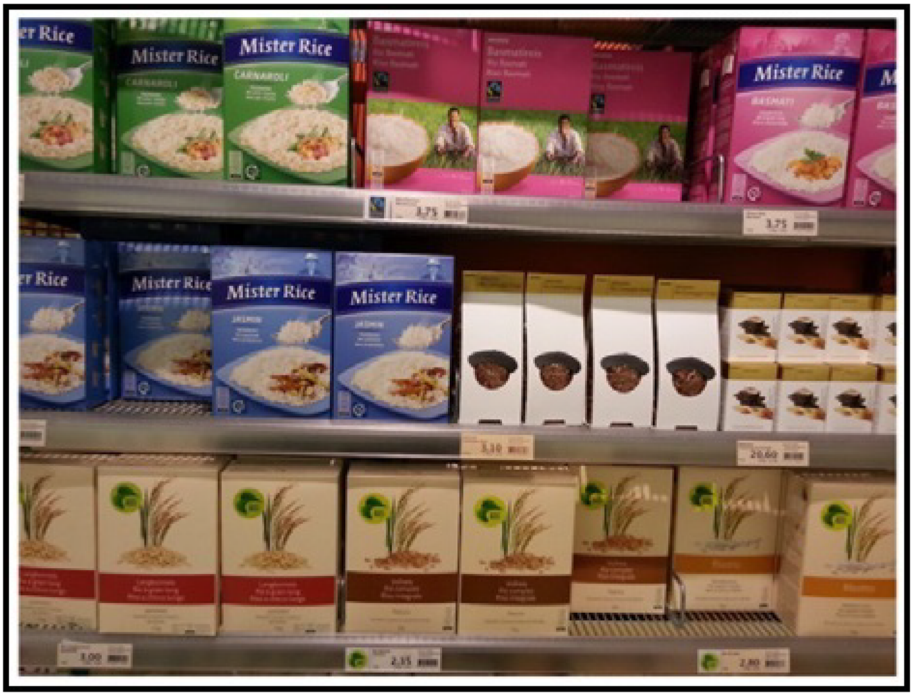


Fig. 3 A sample image of racks from Grocery Products dataset.

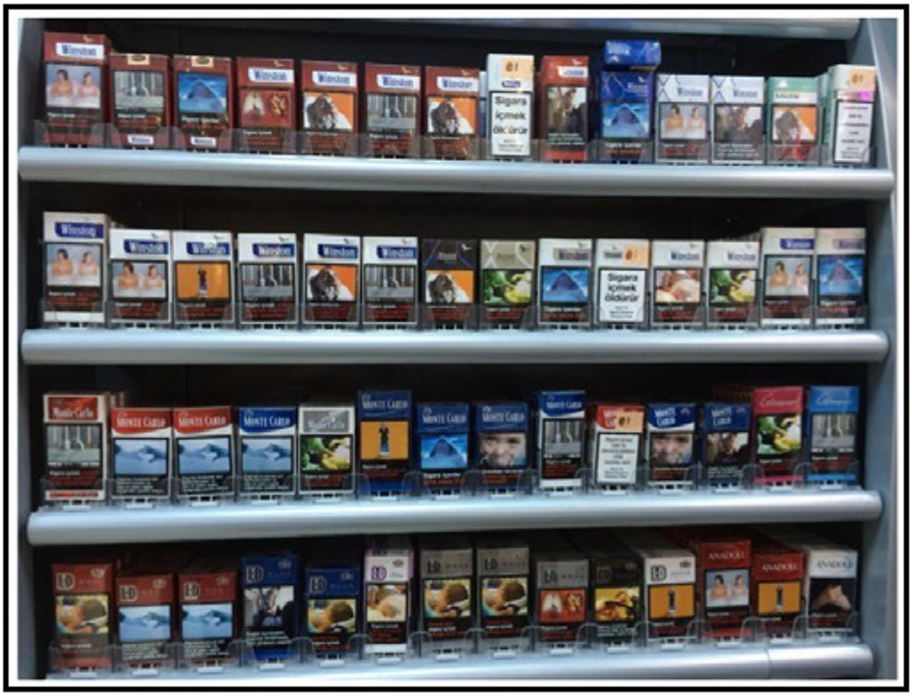
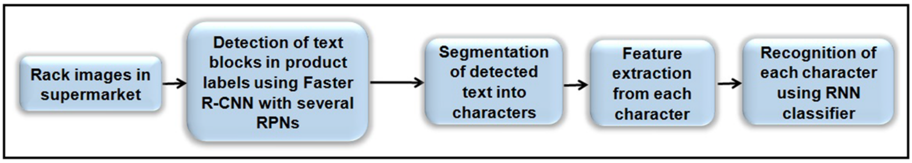


Fig. 4 A sample image of racks from Grocery Dataset.

In the present work, the classifier has been trained with the feature vectors of numerous samples of each character in Latin script. A total of 150 different printed samples of varying sizes for each character in Latin script have been considered to train the classifier.

4 Proposed Method

In the proposed method, the text blocks in the product labels have been first detected through an modified version of the conventional Faster R-CNN. The detected text have then been segmented into character level before extracting various features from each character. Finally, each segmented character has been recognized using RNN classifier. The detailed framework of the proposed method is shown in Fig. 5.



Rack images in supermarket → Detection of text blocks in product labels using Faster R-CNN with several RPNs → Segmentation of detected text into characters → Feature extraction from each character → Recognition of each character using RNN classifier

Fig. 5 The detailed framework of the proposed method.

4.1 Text detection in product labels

This article proposes a new approach of detecting text in the product labels using an alternative version of the traditional Faster R-CNN architecture. ROIs are produced in the traditional Faster R-CNN by a sole RPN. The architecture of Faster R-CNN proposed in this work contains more than one diverse sized RPNs [28] and an extra merger layer to merge morev than one RPNs [28]. The traditional Faster R-CNN is not able to detect always the text portions of varying sizes present in the product labels as it contains a single RPN. But the present system overcomes this drawback as it introduces multiple, varying sized RPNs in the traditional Faster R-CNN architecture. The alternative architecture of Faster R-CNN employing several RPNs proposed in this work is shown in Fig. 6. Different steps of the proposed method are discussed below in brief.

1. **Creation of several RPNs**: Any RPN uses a small network where a single window slides over the convolutional feature matrix obtained from a designated layer to generate the region proposals. The RPN predicts multiple region proposals for each location of the feature matrix simultaneously. Each region proposal is known as anchor box. As many anchors are there in one sliding window, that many region proposals are generated in each location of the feature matrix. The center point of the anchor box coincides with the centre point of each sliding window. As far as the text blocks detection in the product labels is concerned, text portions in the product labels come up with varying sizes and in this scenario the single RPN of conventional Faster R-CNN remains incapable to detect these diverse sized texts. In this work, four varying sized RPNs (illustrated in Fig. 6) have been included in the Faster R-CNN architecture. The optimal value of number of RPNs has been obtained by applying *Bayesian optimization technique*. Various values for the number of RPNs have been passed to the optimization algorithm, but the algorithm has returned the optimal value as "4" after checking the detection results. Compared to other optimization techniques, *Bayesian optimization technique* allows us to tune more parameters jointly with lesser number of experiments and obtain better values. The dimension of the input image has been scaled into 1200x1200 dimension, so the varying dimensional receptive fields (illustrated in Fig. 6) of distinct RPNs can contain the features of the text blocks of all dimensions in the product labels.

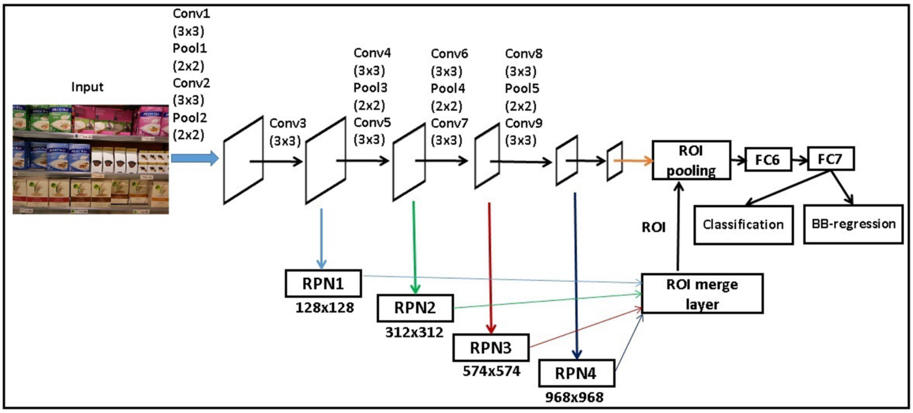

Input
Conv1 (3x3) Pool1 (2x2) Conv2 (3x3) Pool2 (2x2)
Conv3 (3x3)
Conv4 (3x3) Pool3 (2x2) Conv5 (3x3)
Conv6 (3x3) Pool4 (2x2) Conv7 (3x3)
Conv8 (3x3) Pool5 (2x2) Conv9 (3x3)
ROI pooling
FC6
FC7
Classification
BB-regression
ROI
ROI merge layer
RPN1 128x128
RPN2 312x312
RPN3 574x574
RPN4 968x968

Fig. 6 The proposed architecture of Faster R-CNN employing several RPNs.

Table 2 Statistics of various anchors

Scale	HeightxWidth			
1:1	128x128	257x257	515x515	1031x1031
1:2	105x210	211x422	423x846	847x1694
2:1	200x100	370x185	670x335	1250x625
1.5:1	135x90	270x180	540x360	1080x720

2. **Merging of RPNs**: Each RPN produces more than one distinct ROI arrays. All of these distinct arrays have been merged to a single array. This merging is performed by including an independent ROI merger layer (illustrated in Fig. 6) in the Faster R-CNN architecture. ROIs having the intersection over union (IoU) value more than 0.85 have been considered as the candidate ROIs for the detection purpose. All candidate ROIs have been sorted in descending order of their IoU values and then the first 80 ROIs have been chosen for the next step. The optimal value for the number of ROIs is obtained by applying *Bayesian optimization technique*.
3. **Setting of anchors**: Various anchors with different scales and aspect ratios have been incorporated in each RPN. In fact, anchor is a rectangular area indicating the location of the goal object. The total number of anchor boxes for the convolutional feature matrix of size $H*W$ is $H*W*k$, where (H,W) denotes the total number of rows and columns of the convolutional feature matrix respectively and k is the total number of anchors generated for each location of the feature matrix. The statistics of various anchors used in this work are shown in Table 2. For efficient localization of text blocks of varying sizes, smaller anchors were incorporated in RPN1, RPN2 contains larger anchors and the largest anchors were incorporated in RPN3 and RPN4.
4. **Training of RPNs**: In this work, separate training sets were prepared to train four different RPNs. The ground-truth (GT) of each training sample has been developed manually. Small rectangles have been used in the GT to train RPN1. On the other hand, large rectangles in the ground truth have been used to train RPN3 and RPN4. The learning rate was assigned to 0.05 during training of each RPN. The classification loss and regression loss calculated during the training of the proposed architecture are presented in Fig.7. The classification loss measures the classification correctness of each detecting box. The regression loss measures the closeness of the coordinates of the detecting box around a detected text block.

4.2 Text recognition using DL technique

To recognize the detected text blocks in the product labels, initially, each text block has been segmented into character level using vertical projection profile method. Different features have then been extracted from each printed character (written in Latin script) present in the text blocks and then each character present in the text blocks has been recognized using RNN classifier. Thus, the entire text block has been recognized.

4.2.1 Feature extraction

In this work, four structural and direction based features have been extracted from each sample of various characters. The four features are as follows: change of trajectory direction (CTD),

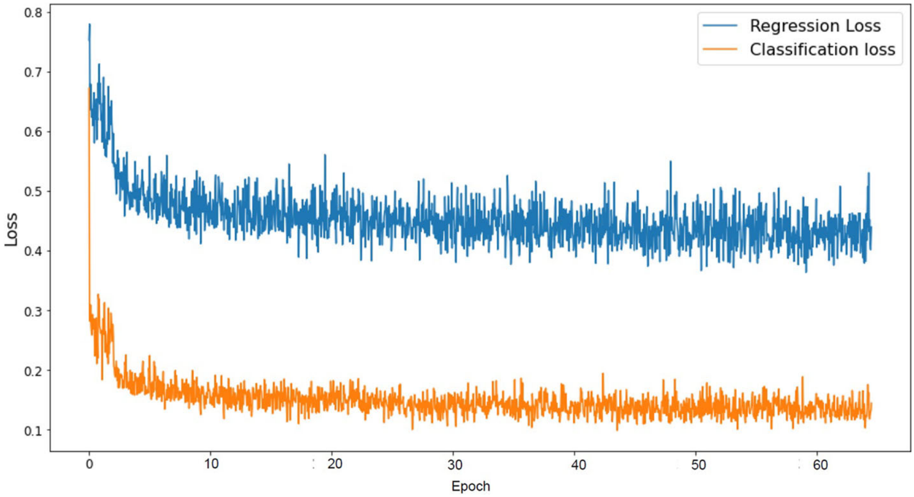


Fig. 7 Regression and classification losses during training.

trajectory slope (TS), trajectory waviness (TW), and centre-of-mass (COM). For signature verification and recognition, Ghosh [29] has used the aforementioned four features. To extract the discriminating features from various characters, the same four aforementioned features have been used in this work. Fig. 8 illustrates the idea of extracting CTD feature from one

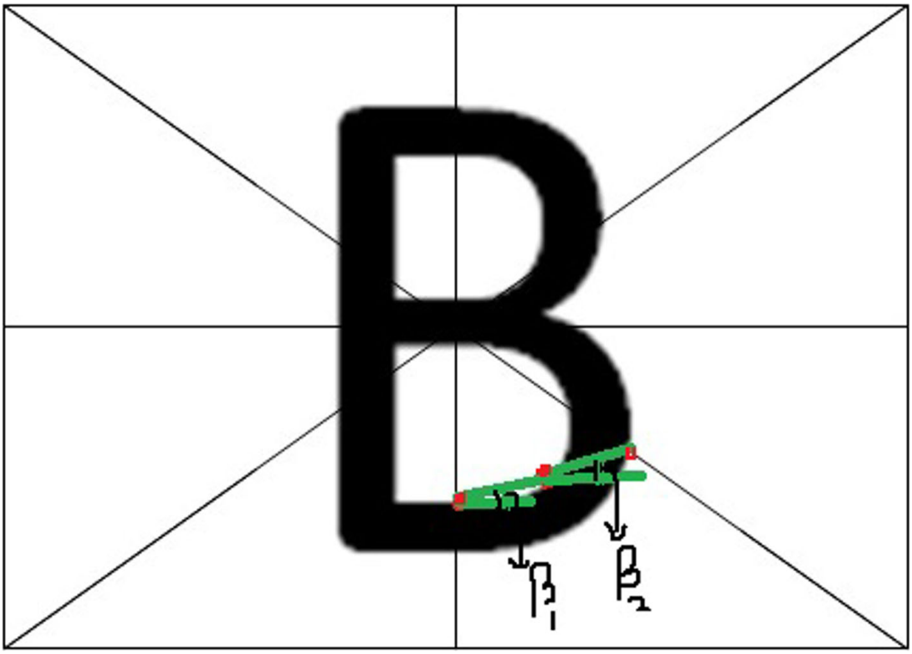


Fig. 8 Illustration of extracting "change of trajectory direction" feature in one printed character sample in Latin script.

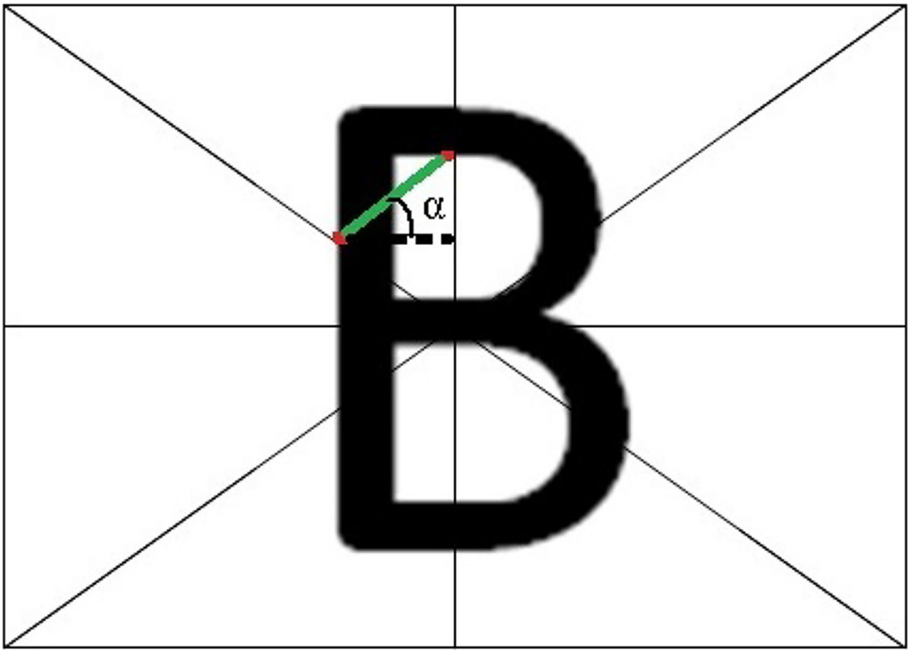


Fig. 9 Illustration of extracting "trajectory slope" feature in one printed character sample in Latin script.

printed character sample in Latin script. Fig. 9 illustrates the idea of extracting TS feature from one printed character sample in Latin script. A 64 dimensional feature vector is generated for each character sample from the aforementioned four features.

4.2.2 Classification and recognition

In this work, the classification of each character in the text blocks is performed using LSTM and BLSTM models of RNN classifier. The internal states of RNN can remember the inputs of several past timestamps due to the existence of recurrently-connected nodes in the hidden layers. RNNs are models that consist of standard recurrent cells, shown in Fig. 10. The typical feature of the RNN cell is a cyclic (or loop) connection, which enables the model to update the current state based on past states and current input data. Formally, the standard recurrent cell is defined as follows:

$$h_j = \phi(W_h h_{j-1} + W_z z_j + b) \tag{1}$$

$$o_j = h_j \tag{2}$$

where z_j=$(x, y, t)_j$ denotes the j^{th} vector of the input signal z=$(x, y, t)_{j=1,...,|z|}$ at timestep j, h_j is the hidden state of the cell, and o_j denotes the cell output, respectively; W_h and W_z are the weight matrices; b is the bias of the neurons; and ϕ is an activation function. Standard recurrent cells have achieved success in many sequence learning problems. However, the standard recurrent cells are not capable of handling long-term dependencies. To solve this issue, the LSTM cells were developed. LSTM cells improve the capacity of the standard recurrent cell by introducing different gates, which are briefly described below.

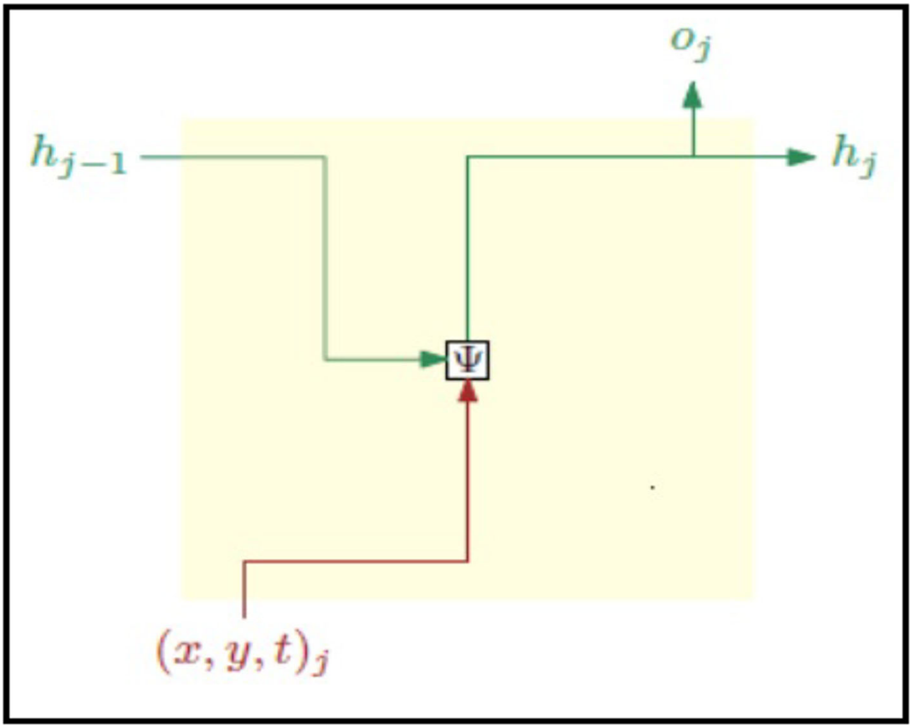


Fig. 10 A simple RNN cell.

The LSTM cell is defined as follows:

$$G_{ip} = \sigma(W_{ud}[h_{j-1}, z_j] + b_{ip}) \tag{3}$$

$$G_{fg} = \sigma(W_{fg}[h_{j-1}, z_j] + b_{fg}) \tag{4}$$

$$G_{op} = \sigma(W_{op}[h_{j-1}, z_j] + b_{op}) \tag{5}$$

$$c_j = G_{ud} \circ \tilde{c}_j + G_{fg} \circ c_{j-1} \tag{6}$$

$$\tilde{c}_j = \phi(W_c[h_{j-1}, z_j] + b_c) \tag{7}$$

$$h_j = G_{op} \circ \phi(c_j) \tag{8}$$

where c_j is an additional hidden state, W_* are weight matrices, b_* are biases, G_* denote cell gates (ip: input, fg: forget, op: output, ud: update), and ϕ and σ are activation functions (hyperbolic tangent and sigmoid, respectively). The operator $\circ$ denotes the Hadamard (element-wise) product. Fig. 11 shows the organization of one LSTM cell.

It may be noted that the LSTM has two kinds of hidden states: a "slow" state c_j that keeps long-term memory, and a "fast" state h_j that makes decisions over short periods of time. The forget gate decides which information will be kept in the cell state and which information will be thrown away from the cell state. Apart from hyperbolic tangent and sigmoid activation functions, there are various other non-linear activation functions have been promoted in the research literature. In the present work, as RNN receives the input from the convolutional layers, so it is already receiving complex features extracted by the convolutional layers.

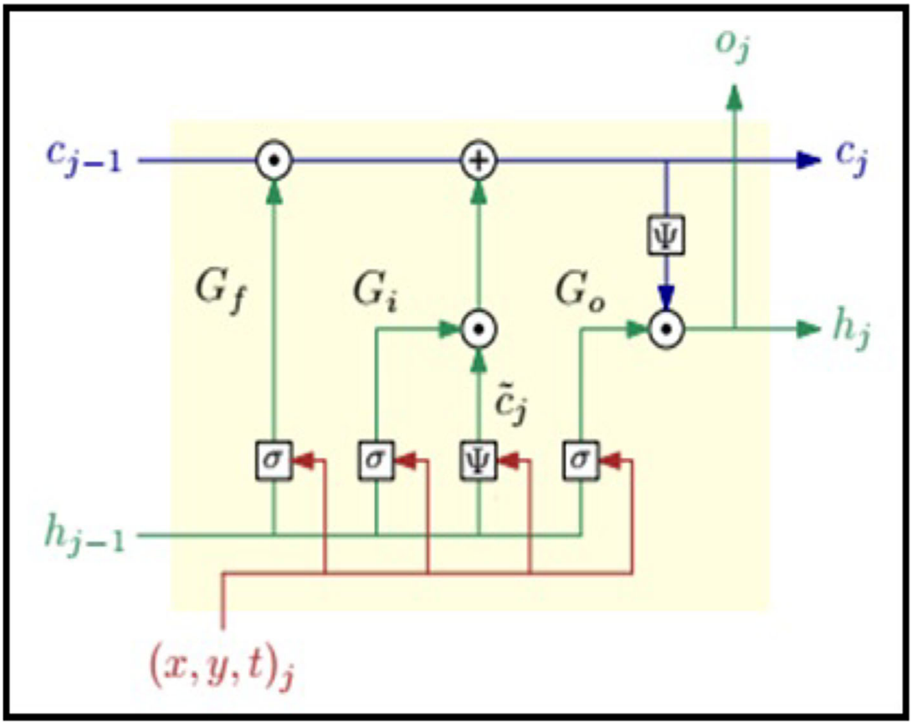


Fig. 11 A LSTM cell.

Accessing of both future and past contexts are required in several tasks to predict the correct class label of the sample. For example, if the class label of any character is predicted by knowing both preceding and succeeding characters, then the prediction will be more accurate. Bidirectional RNNs (BRNNs) [30] are capable to access context in forward as well as backward directions along the sequence. BRNNs have two separate hidden layers—one processes the sequence in forward direction and the second one processes it in backward direction. The same output layer remains connected with both of these hidden layers and it enables the accessing of both past and future contexts of the sequence. Combination of BRNNs and LSTM gives rise to BLSTM variant of RNN.

For both LSTM and BLSTM implementation, the *theano* toolkit[1] has been used. Both LSTM and BLSTM versions have been trained with the feature vectors of numerous samples of each character in Latin script. The 64-dimensional feature vectors of different character samples have been labelled with appropriate class label in the training set. One unique class label has been assigned to the feature vectors corresponding to each character (case-insensitive) in Latin script. For example, all the feature vectors corresponding to the character "A" or "a" have been labelled with class label "1" in the training set, all the feature vectors corresponding to the character "B" or "b" have been labelled with class label "2", and so on. During testing, the feature vectors of the characters in detected text blocks in the product labels have been fed to LSTM and BLSTM models of RNN classifier to know the class label

[1] http://deeplearning.net/software/theano/

Table 3 Accuracy, precision, recall, and F1-Score of the text block detection using the proposed architecture of Faster R-CNN

Dataset	Accuracy	Precision	Recall	F1-Score
GroZi-120	95.74%	95.52%	95.58%	95.54%
Grocery Products	95.67%	95.46%	95.53%	95.50%
Grocery Dataset	95.69%	95.51%	95.55%	95.52%

of each character present in the text. Thus, the entire text block in the product label has been recognized.

5 Results and Analysis

For detecting the text blocks in the product labels, the experiments have been carried out using both traditional and proposed architecture of Faster R-CNN.

5.1 Text block detection results using the proposed architecture

The performance of detecting the text blocks in the product labels has been measured through various metrics, namely, accuracy, precision, recall, and F1-Score. The values of these metrics have been computed by matching the detecting bounding boxes with the ground-truth bounding boxes. If the detecting bounding box overlaps more than 85% of the ground-truth bounding box, then the detecting bounding box has been accepted. The text block detection results using Faster R-CNN with more than one RPNs in terms of accuracy, precision, recall, and F1-Score are listed in Table 3. Table 4 presents the execution speed of the text block detection in terms of frames per second (fps). This work has been executed on the Titan Xp GPU. The precisions of the present system against various orders of receptive field dimensions of four RPNs are shown in Table 5. The optimal set of values of various hyper-parameters in the proposed architecture is presented in Table 6. This optimal set has been obtained using *Bayesian optimization technique*. Fig. 12 illustrates the correct detection of text blocks in few test images from all the three datasets. Detected text blocks have been enclosed within green colored boxes in this figure.

5.2 Text block detection results using traditional Faster R-CNN

The text block detection experiment has been conducted using traditional Faster R-CNN as well so that a comparative performance analysis can be done with the proposed Faster R-CNN architecture. Table 7 lists the results of text block detection using traditional Faster R-CNN.

Table 4 Execution speed for text block detection using the proposed architecture of Faster R-CNN

Dataset	speed (in fps)
GroZi-120	20
Grocery Products	18
Grocery Dataset	18

Table 5 Precision of the text block detection against various orders of receptive field dimensions of four different RPNs

Dataset	Precision	Order of receptive field dimensions of four RPNs
GroZi-120	95.52%	128x128-312x312-574x574-968x968
	89.62%	968x968-574x574-312x312-128x128
Grocery Products	95.46%	128x128-312x312-574x574-968x968
	89.52%	968x968-574x574-312x312-128x128
Grocery Dataset	95.51%	128x128-312x312-574x574-968x968
	89.56%	968x968-574x574-312x312-128x128

Table 6 Optimal values of different hyper-parameters in the proposed architecture

Hyper-parameters	Value
Batch size	512
Overlap threshold for ROI	0.85
Number of RPNs	4
Number of ROIs	80
Learning Rate	0.05
Weight decay for regularization	0.005

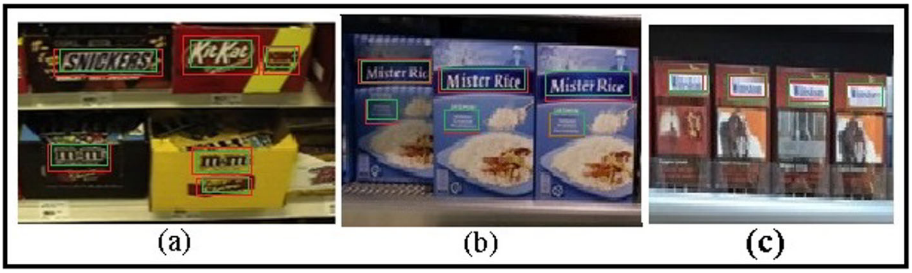


Fig. 12 Correct detection of text blocks using the proposed Faster R-CNN architecture on few test images from (a) GroZi-120, (b) Grocery Products, and (c) Grocery dataset. The red bounding box indicates the ground-truth and the green one indicates the detecting box.

Table 7 Text block detection results using traditional Faster R-CNN with a sole RPN

Dataset	Accuracy	Precision	Recall
GroZi-120	88.67%	88.43%	88.51%
Grocery Products	88.62%	88.41%	88.49%
Grocery Dataset	88.52%	88.34%	88.39%

Table 8 Text block detection results using YOLOv3 and Cascade R-CNN

Dataset	YOLOv3		Cascade R-CNN	
	Accuracy	**Precision**	**Accuracy**	**Precision**
GroZi-120	87.34%	87.29%	90.42%	90.39%
Grocery Products	87.31%	87.27%	90.34%	90.30%
Grocery Dataset	87.28%	87.24%	90.32%	90.27%

The results shown in Tables 3 and 7 clearly state that the performance of the proposed Faster R-CNN architecture is superior than the conventional Faster R-CNN model.

5.3 Text block detection results using YOLOv3 and Cascade R-CNN

The text block detection experiments have also been conducted using two widely used object detection models known as YOLOv3 [31] and Cascade R-CNN [31] to get a picture of com-

Table 9 Results of one-way ANOVA test on four different text block detection models

Anova: Single Factor						
SUMMARY						
Groups	0Count	Sum	Average	**Variance**		
Proposed Faster R-CNN model	2	1.8718	0.9359	0.00089		
Conventional Faster R-CNN model	2	1.7338	0.8669	0.00073		
YOLOv3 model	2	1.7152	0.8576	0.891		
Cascade R-CNN model	2	1.782	0.891	0.000318		
ANOVA						
Source of variation	**SS**	**df**	**MS**	**F**	**P-value**	**F crit**
Between Groups	0.00734538	3	0.0.0024484	4.050288	0.105023	6.591382
Within Groups	0.00241806	4	0.000604			
Total	0.00976343	7				

parative performances with the proposed as well as conventional Faster R-CNN architectures. Table 8 lists the text block detection results using YOLOv3 as well as Cascade R-CNN.

The results shown in Tables 3, 7, and 8 clearly state that the proposed Faster R-CNN architecture has shown the best text block detection performance among itself, conventional Faster R-CNN, YOLOv3, and Cascade R-CNN models.

5.4 Statistical significance test results

A statistical significance test using one-way ANOVA has also been conducted to validate the performance of the proposed text block detection strategy. In this test, four different groups have been created—(i) for the proposed Faster R-CNN model, (ii) for the Conventional Faster R-CNN model, (iii) for the YOLOv3 model, and (iv) for the Cascade R-CNN model. Top two choices (by setting the IoU threshold values at 0.85 and 0.80 for Top 1 and Top 2 choices respectively) have been included in each group. Results of the one-way ANOVA test are presented in Table 9.

5.5 Text recognition results

For recognition of text, LSTM model of RNN contains a single forward hidden layer, whereas BLSTM contains two separate hidden layers—one for processing the input sequence in forward direction and the other for processing it in backward direction. The input layer of both LSTM and BLSTM versions receives a 64 dimensional feature vector. 45 recurrently connected memory blocks have been placed in the hidden layer of LSTM. On the other hand, 45 LSTM memory blocks have been placed in both forward and backward hidden layers of BLSTM. This number has been obtained by applying the *Bayesian optimization technique*. Sigmoid function was used as gate activation function. The number of neurons in the output layer has been set to 26 as Latin script contains 26 basic characters and in the proposed system character wise class labelling scheme has been followed. The softmax activation function has been used to activate each output neuron.

The experiments for recognition have been carried out using both LSTM and BLSTM versions of RNN with several combinations of blocks, epoch, and batch size [9]. The text recognition accuracy results of the present system are shown in Table 10. The results in this table show that the maximum recognition accuracy has been obtained for the combination "block=45, epoch=40, and batch size=8" for all of the three datasets. Table 10 **also shows that BLSTM model provides better recognition accuracy over LSTM model and lower batch size has provided higher recognition accuracy.** Fig. 13 shows different top choices of text recognition accuracy.

Apart from this, the text recognition accuracy has also been evaluated for various feature subsets for the combination "block=45, epoch=40, and batch size=8" using both LSTM and BLSTM versions. The recognition accuracies using some of those subsets using BLSTM version are presented in Table 11. **Few combinations of features in this table are showing lower recognition accuracy due to increase in inter-class similarity from the resultant feature vectors generated through these combinations.** The ***Precision***, ***Recall***, *F1-Score*, and *Receiver Operating Characteristic* (*ROC*) curve analysis of text recognition in the present system **for the combination "block=45, epoch=40, and batch size=8"** are illustrated in Fig. 14, Fig. 15, Fig. 16, and Fig. 17 respectively.

Table 10 Text recognition accuracy (RA) of the proposed system

Dataset	Blocks	Epochs	Batch size	Accuracy LSTM	BLSTM
GroZi-120	20	40	20	43.58%	49.64%
	20	40	8	52.21%	59.81%
	45	40	20	79.67%	87.57%
	45	40	8	89.12%	99.18%
Grocery Products	20	40	20	43.62%	49.69%
	20	40	8	52.26%	59.86%
	45	40	20	79.69%	87.61%
	45	40	8	89.15%	99.21%
Grocery Dataset	20	40	20	43.42%	49.51%
	20	40	8	52.12%	59.72%
	45	40	20	79.56%	87.50%
	45	40	8	89.04%	99.12%

5.6 Comparison with state-of-the-art methods

Initially, a few existing product detection methods that have used the same three public datasets (GroZi-120, Grocery Products, and Grocery Dataset) as used in the present study have been considered for performance comparison with that of the proposed system. Later, few other existing works that have not used the aforementioned three public datasets have been considered and evaluated on these datasets and the performances are compared with the

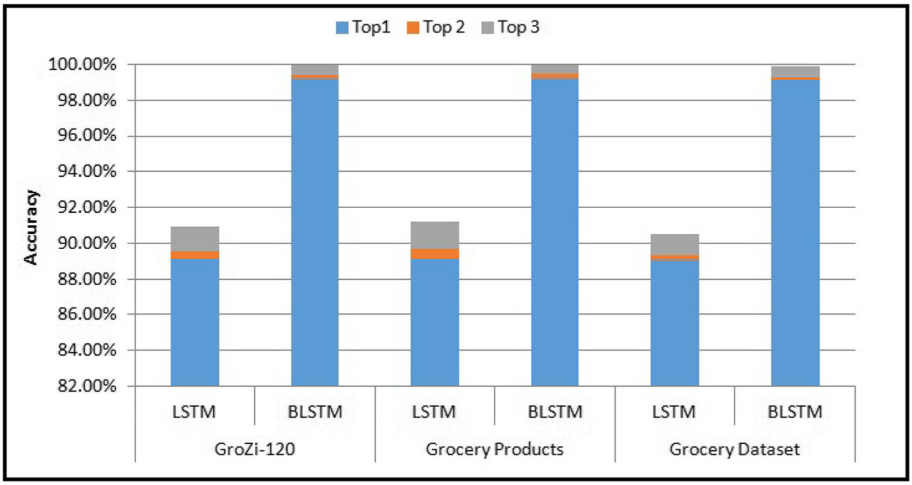


Fig. 13 Text recognition accuracy of various top choices using LSTM and BLSTM (the proposed approach).

Table 11 Text recognition accuracy using BLSTM for the combination “block=45, epoch=40, and batch size=8" with some of the feature subsets

Dataset	Features	Accuracy
GroZi-120	CTD, TS, TW, COM	99.18%
	CTD, TS, TW	92.28%
	CTD, TS, COM	89.36%
	CTD, TW, COM	85.89%
Grocery Products	CTD, TS, TW, COM	99.21%
	CTD, TS, TW	x 92.35%
	CTD, TS, COM	89.41%
	CTD, TW, COM	85.94%
Grocery Dataset	CTD, TS, TW, COM	99.12%
	CTD, TS, TW	x 92.25%
	CTD, TS, COM	89.31%
	CTD, TW, COM	85.82%

present system. The performance comparisons are shown in Table 12. The results presented in Table 12 demonstrate that the proposed text block detection strategy outperforms the state-of-the-art methods on text block detection in the product labels. As far as recognition of text in the detected text blocks is concerned, performance comparison cannot be carried out as no significant study is found in the literature that has attempted for recognition of text present in the product label.

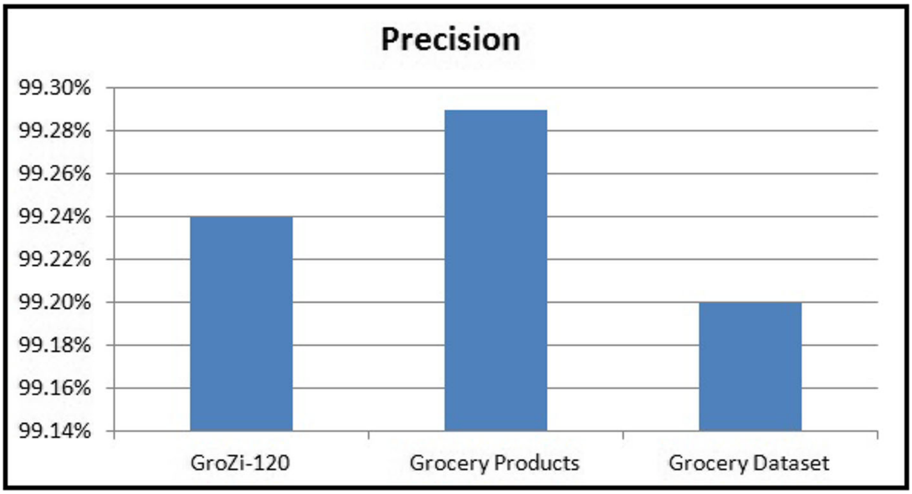


Fig. 14 ***Precision* of text recognition in the present system for the combination “block=45, epoch=40, and batch size=8”.**

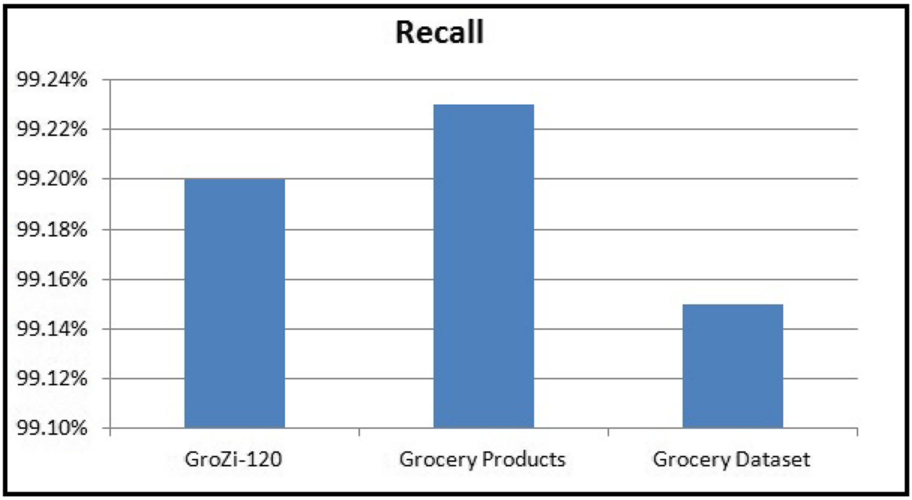


Fig. 15 ***Recall* of text recognition in the present system for the combination "block=45, epoch=40, and batch size=8".**

5.7 Strengths of the present system

1. The proposed system can detect varying sized text blocks on the product packets more successfully as compared to the state-of-the-art methods as more than one diverse dimensional RPNs have been included in the proposed Faster R-CNN architecture.

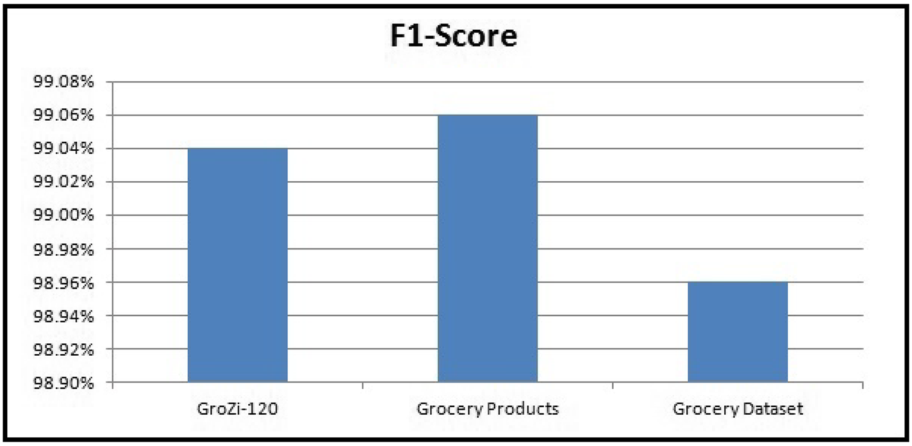


Fig. 16 *F1-Score* of text recognition in the present system **for the combination "block=45, epoch=40, and batch size=8".**

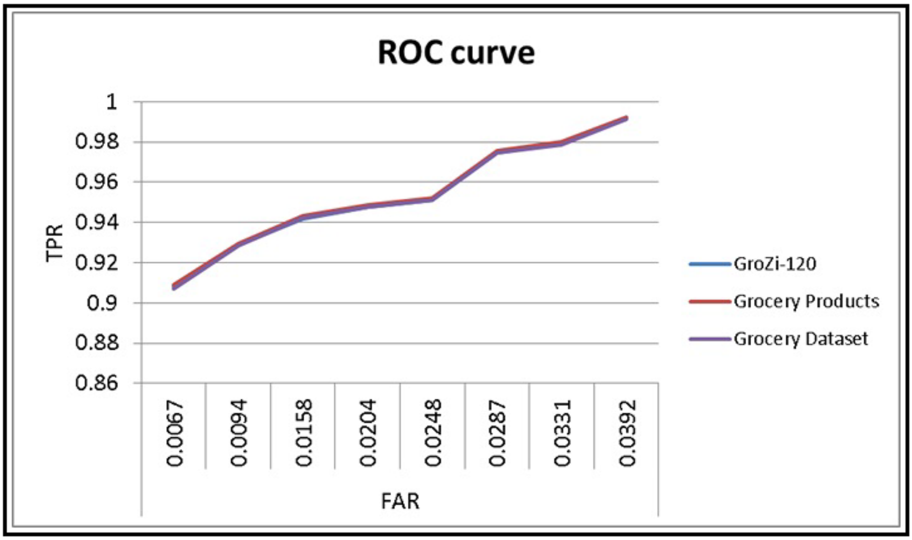


Fig. 17 *ROC curve* analysis of text recognition in the present system **for the combination "block=45, epoch=40, and batch size=8".**

2. The proposed system exploits the sequence memorizing power of LSTM and BLSTM versions of RNN and so it recognizes any character in the text after having the knowledge of the characters occurring both before and after it. Other shallow ML techniques do not have this capability and due to which the present system produces better text recognition results in comparison to various shallow ML based methods [9].

6 Conclusion and Future Scope

This article proposes a novel method of product label detection as well as identification from the supermarket rack images by incorporating several RPNs in conventional Faster R-CNN structure and using RNN classifier. The proposed text block detection strategy has provided 95.52%, 95.46%, and 95.51% precisions for GroZi-120, Grocery Products, and Grocery Dataset respectively. The text block detection results presented in Table 3 and Table 7 demonstrate that the proposed Faster R-CNN architecture is more efficient than the conventional Faster R-CNN architecture in detecting the varying sized text blocks in the product labels. The results **presented in Table** 12 establish that execution speed of the proposed text block detection strategy is more than the existing systems due to the use of Faster R-CNN and the proposed text block detection strategy outperforms the existing methods **on text block detection in the product labels**. The text recognition strategy in the proposed method has provided the accuracies of 99.18%, 99.21%, and 99.12% for GroZi-120, Grocery Products, and Grocery Dataset respectively. In future, the attempt will be made to improve the text block detection performance of the present system. Increasing the execution speed of the system is another future direction of this research work.

Table 12 Comparison with state-of-the-art results on text block detection on three benchmark datasets

Dataset	Reference	Method	Precision	Execution speed (in fps)
GroZi-120	George et al. [1]	Dense SIFT features	13.21%	0.5
	Karlinsky et al. [5]	Three phase method	49.70%	—
	Zientara et al. [6]	Saliency map based method	62.38%	10
	Franco et al. [7]	BOW	45.70%	—
	Merler et al. [10]	Sliding window based method	56.28%	10
	Marder et al. [11]	Sliding window based HOG	67.34%	09
	Saran et al. [12]	Sobel operator and Hough transform	72.18%	12
	Hu et al. [18]	Deep CNN	92.21%	18
	Umer et al. [19]	ML techniques	86.54%	18
	Proposed method	Faster R-CNN with several RPNs	**95.52%**	20
Grocery Products	George et al. [1]	Dense SIFT features	30.70%	0.5
	Karlinsky et al. [5]	Three phase method	44.72%	—
	Zientara et al. [6]	Saliency map based method	61.52%	10
	Franco et al. [7]	BOW	77.70%	—
	Merler et al. [10]	Sliding window based method	55.48%	10
	Marder et al. [11]	Sliding window based HOG	67.58%	09
	Saran et al. [12]	Sobel operator and Hough transform	72.54%	12
	Tonioni et al. [17]	SIFT and Hough transformX	90.40%	—
	Hu et al. [18]	Deep CNN	92.08%	18
	Umer et al. [19]	ML techniques	86.38%	18
	Proposed method	Faster R-CNN with several RPNs	**95.46%**	18
Grocery Dataset	George et al. [1]	Dense SIFT features	42.68%	0.5
	Varol et al. [2]	Object detector	88.00%	—
	Zientara et al. [6]	Saliency map based method	60.42%	09
	Merler et al. [10]	Sliding window based method	56.12%	09
	Marder et al. [11]	Sliding window based HOG	67.62%	08
	Saran et al. [12]	Sobel operator and Hough transform	73.28%	11
	Varol et al. [27]	Canny edge detector and Hough transform	81.00%	—
	Hu et al. [18]	Deep CNN	92.13%	18
	Umer et al. [19]	ML techniques	86.44%	18
	Proposed method	Faster R-CNN with several RPNs	**95.51%**	18

Data Availability Data sharing not applicable to this article as no datasets were generated during the current study.

Declarations

Funding and/or Conflicts of interests/Competing interests The author has no conflict of interest/competing interest to declare that are relevant to the content of this article.

References

1. George M, Floerkemeier C (2014) “Recognizing products: a per-exemplar multi-label image classification approach”, Proceedings of the European Conference on Computer Vision, pp. 440–455
2. Varol G, Kuzu RS, Akgiil YS (2014) “Product placement detection based on image processing”, Proceedings of the Signal Processing and Communications Applications Conference, pp. 1031–1034
3. George M, Mircic D, Soros G, Floerkemeier C, Mattern F (2015) “Fine-grained product class recognition for assisted shopping”, Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 154–162
4. Cleveland J, Thakur D, Dames P, Phillips C, Kientz T, Daniilidis K, Bergstrom J, Kumar V (2017) Automated system for semantic object labeling with soft-object recognition and dynamic programming segmentation. IEEE Trans Autom Sci Eng 14(2):820–833
5. Karlinsky L, Shtok J, Tzur Y, Tzadok A (2017) “Fine-grained recognition of thousands of object categories with single-example training”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4113–4122
6. Zientara P, Advani S, Shukla N, Okafor I, Irick K, Sampson J, Datta S (2017) “A multitask grocery assistance system for the visually impaired smart glasses, gloves, and shopping carts provide auditory and tactile feedback. IEEE Consum Electron Mag 6(1):73–81
7. Franco A, Maltoni D, Papi S (2017) Grocery product detection and recognition. Expert Syst Appl 81:163–176
8. Ren S, He K, Girshick R, Sun J (2015) “Faster R-CNN: Towards real-time object detection with region proposal networks”, Proceedings of the NIPS
9. Ghosh R, Vamshi C, Kumar P (2019) RNN based online handwritten word recognition in Devanagari and Bengali scripts using horizontal zoning. Pattern Recog 92:203–218
10. Merler M, Galleguillos C, Belongie S (2007) “Recognizing groceries in situ using in vitro training data”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8
11. Marder M, Harary S, Ribak A, Tzur Y, Alpert S, Tzadok A (2015) Using image analytics to monitor retail store shelves. IBM J Res Dev 59(2/3):1–3
12. Saran A, Hassan E, Maurya AK (2015) “Robust visual analysis for planogram compliance problem”, Proceedings of the 14th IAPR International Conference on Machine Vision Applications, pp. 576–579
13. Liu S, Tian H (2015) “Planogram compliance checking using recurring patterns”, Proceedings of the 2015 IEEE International Symposium on Multimedia, pp. 27–32
14. Winlock T, Christiansen E, Belongie S (2010) “Toward real-time grocery detection for the visually impaired”, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 49–56
15. Yörük E, Öner KT, Akgül CB (2016) “An efficient Hough transform for multi-instance object recognition and pose estimation”, Proceedings of the 23rd International Conference on Pattern Recognition, pp. 1352–1357
16. Zhang Q, Qu D, Xu F, Jia K, Sun X (2016) “Dual-layer density estimation for multiple object instance detection”, J Sensors, pp. 1–13
17. Tonioni A, Stefano LD (2017) “Product recognition in store shelves as a subgraph isomorphism problem”, Proceedings of the International Conference on Image Analysis and Processing, pp. 682–693
18. Hu B, Zhou N, Zhou Q, Wang X, Liu W (202) “DiffNet: A Learning to Compare Deep Network for Product Recognition”, IEEE Access, Volume 8, pp. 19336–19344
19. Umer S, Mohanta PP, Rout RK, Pande HM (2020) Machine learning method for cosmetic product recognition: a visual searching approach. Multimed Tools Appl. https://doi.org/10.1007/s11042-020-09079-y

20. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vision 60(2):91–110
21. Shi Tomasi J (1994) “Good features to track”, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 593–600
22. Harris C, Stephens M (1988) “A combined corner and edge detector”, Proceedings of the Alvey Vision Conference, pp. 10–5244
23. Fritz M, Leibe B, Caputo B, Schiele B (2005) “Integrating representative and discriminant models for object category detection”, Proceedings of the Tenth IEEE International Conference on Computer Vision, pp. 1363–1370
24. Dalal N, Triggs B (2005) “Histograms of oriented gradients for human detection”, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 886–893
25. Lazebnik S, Schmid C, Ponce J (2006) “Beyond bags of features: spatial pyramid matching for recognizing natural scene categories”, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2169– 2178
26. Bay H, Ess A, Tuytelaars T, Gool LV (2008) Speeded-up robust features (SURF). Comput Vision Image underst 110(3):346–359
27. Varol G, Kuzu RS (2014) “Toward retail product recognition on grocery shelves”, Proceedings of the Sixth International Conference on Graphic and Image Processing, pp. 944309–944309
28. Ghosh R (2021) On-road Vehicle Detection in Varying Weather Conditions using Faster R-CNN with Several Region Proposal Networks. Multimed Tools Appl 80:25985–25999
29. Ghosh R (2021) A Recurrent Neural Network based Deep Learning Model for Offline Signature Verification and Recognition System. Expert Systems With Applications 168. https://doi.org/10.1016/j.eswa.2020.114249
30. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45:2673–2681
31. Ghosh R (2022) A Faster R-CNN and recurrent neural network based approach of gait recognition with and without carried objects. Expert Systems With Applications 205. https://doi.org/10.1016/j.eswa.2022.117730

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