Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2019.DOI

Cursive Text Recognition in Natural Scene Images using Deep Convolutional Recurrent Neural Network

ASGHAR ALI CHANDIO[1,2], MD. ASIKUZZAMAN[2], MARK R. PICKERING[2], MEHWISH LEGHARI[1]
[1]Department of Information Technology, Quaid-e-Awam University, Pakistan (e-mail: asghar.ali@quest.edu.pk)
[2]School of Engineering and Information Technology, The University of New South Wales, Canberra, Australia
Corresponding author: Asghar Ali Chandio (e-mail: asghar.ali@quest.edu.pk, z5122075@zmail.unsw.edu.au).

ABSTRACT Text recognition in natural scene images is a challenging problem in computer vision. Different than the optical character recognition (OCR), text recognition in natural scene images is more complex due to variations in text size, colors, fonts, orientations, complex backgrounds, occlusion, illuminations and uneven lighting conditions. In this paper, we propose a segmentation-free method based on a deep convolutional recurrent neural network to solve the problem of cursive text recognition, particularly focusing on Urdu text in natural scenes. Compared to the non-cursive scripts, Urdu text recognition is more complex due to variations in the writing styles, several shapes of the same character, connected text, ligature overlapping, stretched, diagonal and condensed text. The proposed model gets a whole word image as an input without pre-segmenting into individual characters, and then transforms into the sequence of the relevant features. Our model is based on three components: a deep convolutional neural network (CNN) with shortcut connections to extract and encode the features, a recurrent neural network (RNN) to decode the convolutional features, and a connectionist temporal classification (CTC) to map the predicted sequences into the target labels. To increase the text recognition accuracy further, we explore deeper CNN architectures like VGG-16, VGG-19, ResNet-18 and ResNet-34 to extract more appropriate Urdu text features, and compare the recognition results. To conduct the experiments, a new large-scale benchmark dataset of cropped Urdu word images in natural scenes is developed. The experimental results show that the proposed deep CRNN network with shortcut connections outperform than other network architectures. The dataset is publicly available and can be downloaded from https://data.mendeley.com/datasets/k5fz57zd9z/1.

INDEX TERMS Cursive text recognition in natural images, Urdu scene text recognition, natural scene text recognition, convolutional recurrent neural network, segmentation-free scene text recognition

I. INTRODUCTION

Text in natural scene images contains rich and valuable information that has great importance with several real-world applications, such as automatic license plate recognition, content-based image or video retrieval, geo-location, assisting visually impaired people, robot navigation, street and road signs recognition and helps in image understanding [1]–[3]. Regardless of remarkable improvements in natural scene text recognition, it still remains a challenging task due to complex backgrounds, variations in text size, colors, orientations, low resolution, occlusion, environmental noise and blur [4]. In addition, several non-text objects such as leaves, bricks, fences and other patterns as illustrated in Figure 1 that resemble with text decrease the recognition accuracy when present in natural scene images.

Recently, deep learning methods have been developed to address the above challenges in natural scene text recognition [5]. It is noteworthy that most of these developments have focused on the Latin scripts [6], [7]. However, for cursive scripts such as Arabic and Urdu, text detection and recognition in natural scene images is an emerging field. In recent years, segmentation-free methods have been demonstrated for the handwritten, printed or artificial Arabic and Urdu text recognition in the scanned documents and

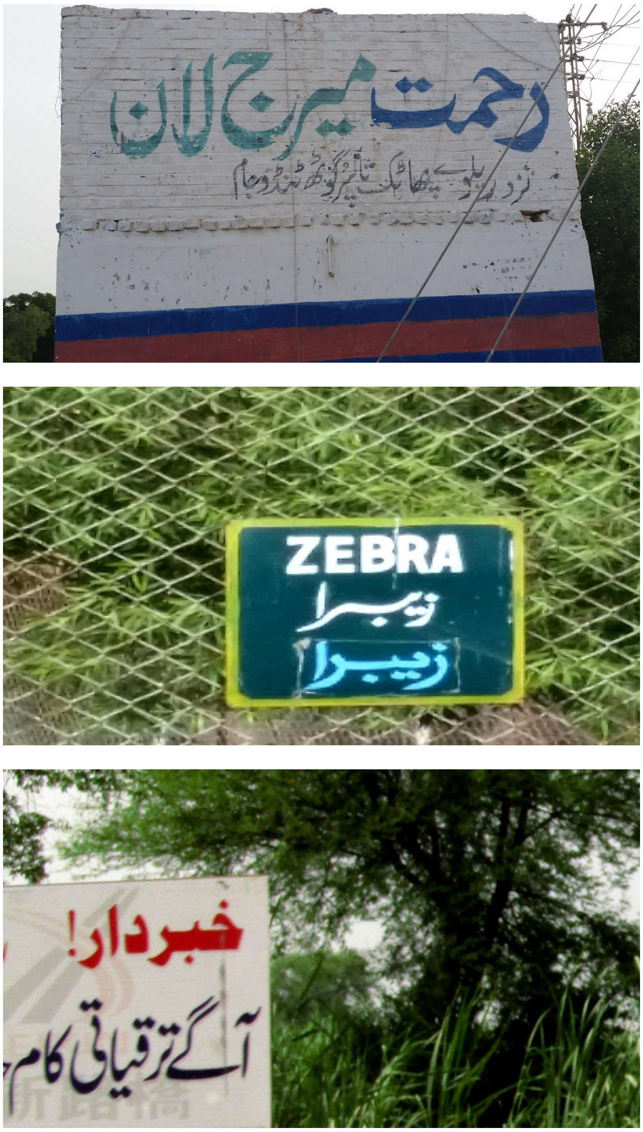


FIGURE 1: Non-text objects such as bricks, fences and leaves resembling with text in natural scene images.

video images [8]–[11]. The state-of-the-art techniques such as a convolutional recurrent neural network (CRNN) have been applied that achieved remarkable results for handwritten and printed Urdu text recognition [12], [13]. Due to plain background, unique font color, size and style, the printed Arabic or Urdu text is not as complex as natural scene text. Although significant work has been performed for the handwritten, printed or artificial text in Arabic or Urdu scripts, the recognition of Arabic and Urdu text in natural scene images has not demonstrated significant results yet [14], [15]. Moreover, most existing research works reported in the literature are limited to the isolated Arabic and Urdu character recognition [16]–[20].

Urdu is a type of cursive script written in right to left direction. It is the official language of Pakistan and is widely spoken in western India [21]. The Urdu script has joiner and non-joiner characters. The joiner characters can appear in one of the four positions in a word (initial, middle, final and isolated) as demonstrated in Fig. 2, while the non-joiner characters have two forms (isolated and final). The initial form connects another character at its left position, the middle form connects two other characters at its left and right positions, the final form connects another character at its right position and the isolated form does not connect any character at its either positions. Moreover, in Urdu script, the isolated characters have no meaning unless they are connected to other characters as demonstrated in Fig. 2. Due to different positions of the same character in a word, the implicit segmentation becomes very complex. Similarly, more than one shapes of the same character increase the complexity of text recognition. Several additional characteristics such as inter and intra ligature overlapping, character diagonality, context-sensitivity and stretched text associated with the cursive script like Urdu make the text recognition problem further complex. Fig. 3 demonstrates some complexities associated with the Urdu text in natural scene images. The multilingual OCR system [22] that have been used for the text recognition in scanned documents, failed when applied for the Urdu text recognition in natural scene images. This is mainly due to the complex structure of the Urdu script and the challenges related to the text in natural images.

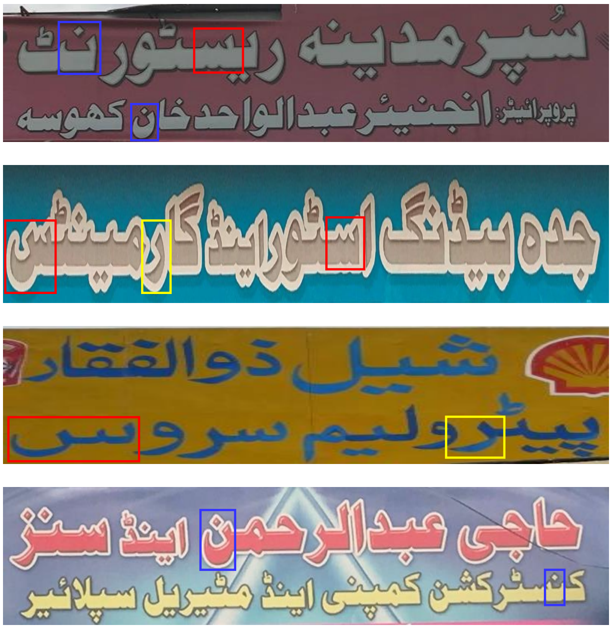
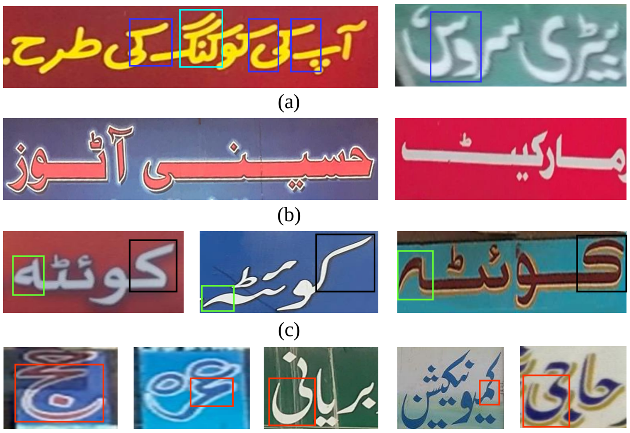


FIGURE 2: Urdu text in natural images. The red and blue rectangles show two Urdu characters ‘seen’ and ‘noon’ in their different positions. On the other hand, the yellow rectangles show Urdu character ‘ray’ in its isolated and last form.



(a)

(b)

(c)

(d)

FIGURE 3: Characteristics of Urdu text in natural scene images. The blue, cyan, green and red rectangles show the intra-ligature overlapping, inter-ligature overlapping, context sensitivity and character diagonality respectively.

In this paper, we propose a segmentation-free deep CRNN to recognise the cropped Urdu word image text in natural scene images. We combine the CNN and RNN, where the CNN part is used to extract the relevant features from the Urdu word images and encode them into feature sequences. On the contrary, the RNN part separately implementing a bi-directional long short term memory (BLSTM) and a bi-directional gated recurrent unit (BiGRU) is used to decode the corresponding feature sequences into the predicted labels. A connectionist text component (CTC) cost function is applied to map the predicted labels with the target labels. In the experiments, deep CNN structures such as VGG-16 [23] and VGG-19 [23] with different depths are exploited for encoding robust image features. However, the text recognition accuracy of the deep CNN structures like VGG-19 [23] is decreased due to the vanishing gradient problem. To overcome the vanishing gradient problem, ResNet [24] networks are utilised. Further, a new deep VGG-16 [23] architecture with shortcut connections is proposed to deal with the degradation problem and improve the text recognition accuracy. A large-scale dataset of cropped Urdu word images in natural scenes is developed to evaluate the proposed models. To the best of our knowledge, this is the first dataset developed for the Urdu text recognition in natural scene images. The main contributions of this paper are summarised as follows:

1) Several deep structures of the CNN including VGG-16, VGG-19, ResNet-18 and ResNet-34 are explored and modified for the challenging problem of cursive text recognition in natural scene images.
2) Several structures of the RNN including LSTM, BLSTM and BiGRU are exploited for better feature decoding and label predictions. A CTC cost function is used with the RNN structures for mapping predicted sequences into the target labels.
3) To overcome the gradient vanishing problem, a new VGG-16 architecture with shortcut connections is proposed that outperformed than the original VGG-16 and VGG-19 architectures.
4) A large-scale dataset of cropped Urdu word images in natural scenes is proposed. This is the first dataset developed for the Urdu text in natural scene images.

The remainder of this paper is organised as follows. The existing state-of-the-art deep learning algorithms developed for the cropped word text recognition from natural scene images are presented in Section II. Section III describes the proposed cropped Urdu word image text recognition framework. The network training process is given in Section IV. Section V demonstrates the experimental results and analysis. Finally, Section VI summarises and concludes this paper.

II. RELATED WORK

The traditional methods developed for text recognition in natural scene images are either based on the isolated character or whole word recognition. Character-based methods used sliding window [25], [26], connected component [27], part-based tree structure [28] or stroke width transform [29] to localise the individual characters. The character classifiers with a combination of different feature descriptors such as CNN [26], a histogram of oriented gradient (HOG) with random ferns [25], and combination of multi-scale mid-level features with random forest [30] were applied. The individually recognised characters were grouped into the words by applying some fixed lexicon-based clustering methods. Jaderberg et al. [26] used a CNN to generate text/non-text, a case-sensitive and case-insensitive character and bi-gram saliency maps for detecting and recognising text in natural scene images. They trained a supervised character classifier to generate the abundant features and used the intermediate layers of the network as features for text detection, character and bi-gram classifications. Further, they used a Viterbi algorithm to recognise the whole word from a fixed lexicon. These methods only consider the individual character recognition, which in some cases may reduce the performance of the text recognition system due to the large number of inter-character and intra-character variations as well as character classes. Additionally, variations in the shape of the same character, ligature overlapping, stretched characters and different writing styles will further reduce the performance of cursive scene text recognition systems.

To overcome the problem of false character detection and recognition, word-level text recognition methods are proposed that directly map the word string from the entire image. Almazán et al. [31] embedded both word images and label strings into the common subspace vectors, which were then used to match the images with their labels. Jaderberg et al. [32] gave whole word image as input to the CNN model that generated fixed representations. They generated a dictionary of 90k words and considered the text recognition problem as a 90k class problem, wherein each class corresponded to a word. Although impressive results are reported with this model, it is limited to recognise the words available in the predefined lexicon file.

Recently, the scene text recognition problem has been treated as a sequence-to-sequence problem. She et al. [33] integrated the feature encoding, sequence modeling and text transcription methods into an end-to-end trainable framework for text recognition in natural scene images. The network is able to handle text with arbitrary length without using character level segmentation or horizontal scale normalization. The network achieves remarkable results on English scene text datasets using both lexicon-based and lexicon-free text recognition tasks. Zhang et al. [34] proposed an attention-based sequence-to-sequence network to recognise text in natural scene images. The attention-based encoder-decoder automatically concentrated on the regions which were most relevant to the text. Lei et al. [35] considered the text recognition problem as a sequence-to-sequence recognition and combined convolutional and recurrent neural networks. They explored several feature extraction and sequence labelling architectures. Sheng et al. [36] used a stacked self-attention sequence-to-sequence encoder and decoder model. Further, they implemented a modality-transform method that effectively transformed 2D natural scene image features into the 1D feature sequences.

Some research works recently presented have focused on Arabic text recognition in video images [8], [14], [37] and natural scenes [14]. Zayene et al. [8] used a segmentation-free method based on multidimensional LSTM and a CTC to recognise Arabic text in news video images. Yousfi et al. [37] used an RNN to persist long-range Arabic video image text sequences. Further, they combined the language models with the LSTM to improve text recognition accuracy. To decode the LSTM sequences, they proposed a beam search method that used both OCR and the language models to predict the probabilities at each time step. Jain et al. [14] demonstrated a hybrid segmentation-free model for Arabic text recognition in video and natural images. To make the model an end-to-end trainable as in [33], they combined the CNN and RNN with a CTC cost function. For the Arabic text in natural scene images, they rendered artificial Arabic text on the backgrounds of real images downloaded from the Google search engine.

In this paper, we propose a segmentation-free method that transforms the text recognition problem into a sequence-based temporal classification task. We show that the deep CNN architectures with shortcut connections extract more robust features. When combining CNN with bidirectional recurrent structures, the network is able to learn long-range contextual information in both forward and backward directions. This contextual information obtained in both directions is important in making more accurate predictions when considering cursive scripts, in which many characters have similar shapes. Further, integrating with a CTC cost function, recurrent structures are able to perform text transcriptions without any prior information about text elements or their complex structures.

III. PROPOSED METHODOLOGY

The general architecture of the proposed cropped Urdu word image text recognition framework is illustrated in Fig. 4. The framework is based on three components: (1) the CNN component for feature extraction, (2) the RNN component to decode the feature sequences into per-frame predictions and (3) the transcription component to map the per-frame predictions into the target labels. This framework combines two different networks (CNN and RNN) and is end-to-end trainable with a single loss function. Combining CNN with RNN for English text recognition in natural scenes was first proposed in [33]. The proposed framework is inspired by this network and applies several modifications to handle the problem of Urdu text recognition. The framework proposed in [33] used seven convolutional layers, four max pooling layers and two batch normalisation layers [38] after the fifth and sixth convolutional layers. However, the frameworks proposed for cropped Urdu word image text recognition use VGG-16 [23] model without fully connected layers, ResNet [24] and its variant models, and a new proposed network similar to [23] but with shortcut connections for feature extraction. A bidirectional recurrent layer such as BLSTM or a BiGRU is applied on top of the feature extraction module to predict per-frame label sequences. Finally, a CTC layer is used to map the predicted sequences into their target labels. Same as [33], we use VGG-16 network without fully connected layers and BLSTM with 256 hidden units in model 2 as shown in Table 1. Model 2 gives 84.26% and 90.82% WRR and WRR$_{1F}$ respectively. The configuration of the model proposed in [33] is almost same as Model 2 however, the size of input in [33] is 100 x 32, whereas in our case it is 100 x 64. If we compare the results obtained with our proposed networks i.e., VGG-16 with skip connection and ResNet-18 with v2 residual block the WRR and WRR$_{1F}$ are 87.13%, 94.21%, 84.42% and 92.30%, respectively as shown in Table 5. These results demonstrate that VGG-16 with skip connection performs much better than the original model of VGG-16 network as used in [33] and Model 2 as shown in Table 1. Moreover, the model proposed in [33] is trained only on the synthetic images with English text only, however, the proposed model is trained on the real natural scene images of cursive text. Compared to [33] the proposed model uses strided convolutional layers with ResNet models and provides a detailed implementation of various networks and compares their results.

A. CNN FOR FEATURE EXTRACTION

In the experiments, deep networks with different depths such as VGG-16 [23], ResNet-18 [24] and ResNet-34 [24] are exploited to better encode the image information in the CNN part of the proposed framework. To further analyse the effect of deep network depths for feature extraction on the recognition ability of the model, a deeper network such as VGG-

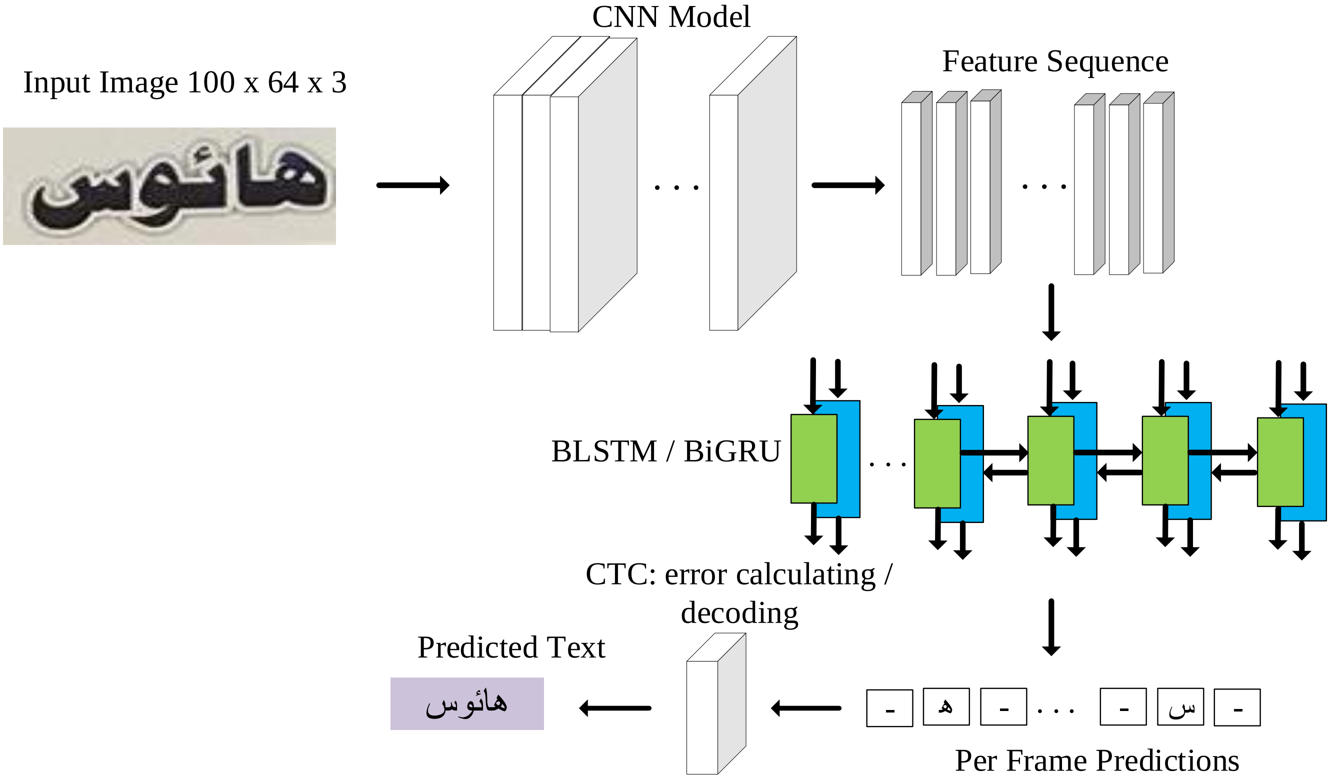


FIGURE 4: General architecture of the proposed cropped Urdu word image text recognition system.

19 [23], is used. However, VGG-19 did not perform better than VGG-16 and decreased recognition accuracy due to the problem of gradient vanishing. To overcome this problem, a novel CNN architecture is proposed by introducing shortcut connections within a VGG-16 network. The performance of each feature extraction network is evaluated separately on the test dataset. It is assumed that text in the images is a sequence of characters; therefore, the purpose of the feature extraction models is to find the best representations of the sequential patterns from the given images and preserve the critical information at different levels (characters or words).

As the direction of text in the dataset images is mostly horizontal, the feature maps can be down-sampled in the vertical direction several times until their height is reduced to 1. However, down-sampling the feature maps excessively in the horizontal direction may cause the problem of overlapping of two adjacent characters. Therefore, depending upon the maximum length of text instances, the feature maps can be reduced slightly in the horizontal direction. The output of the last CNN layer can be a 1D feature map with a variable width and a height of 1. In the proposed framework, the output of the last CNN layer is $1 \times 25 \times C$, where 1, 25 and C are the height, width and depth of the feature map respectively. This feature map is then split column-wise to make a feature vector as a time step and passed to the recurrent layers. In CNNs, the convolutional, max pooling and activation functions operate on a small region and are translation invariant, such that they can recognise an object regardless of its position within the feature map. Each column in the feature map corresponds to a rectangular region in the input image or the previous layer. These rectangular regions are called receptive fields.

Each feature sequence not only contains the character information, but also includes adequate contextual information. In the proposed networks, the feature sequences are denoted as $x = \{x_1, x_2, \cdots, x_N\}$, where $x_t \in R^{512}$ and N is the length of the feature sequences. As this paper exploits the VGG-16 model and VGG-16 model with the shortcut connections as well as ResNet model, the modified architectures of these models are described below.

1) VGG-16 Network

VGG-16 is the first model used to extract feature sequences from cropped Urdu word images in natural scenes. The architecture of the convolutional layers is the same as in a VGG-16 network; however, an additional block with two convolutional layers and a max pooling layer is used to extract more abstract features and down-sample the height of the feature map to 1. Fig. 5 illustrates the VGG-16 model with additional convolutional block. This figure shows that the network has 15 convolutional layers, except for the input, which takes a cropped word image of the Urdu text with a fixed size of $64 \times 100 \times 3$ pixels. To down-sample the feature maps, each convolutional layer is followed by a max

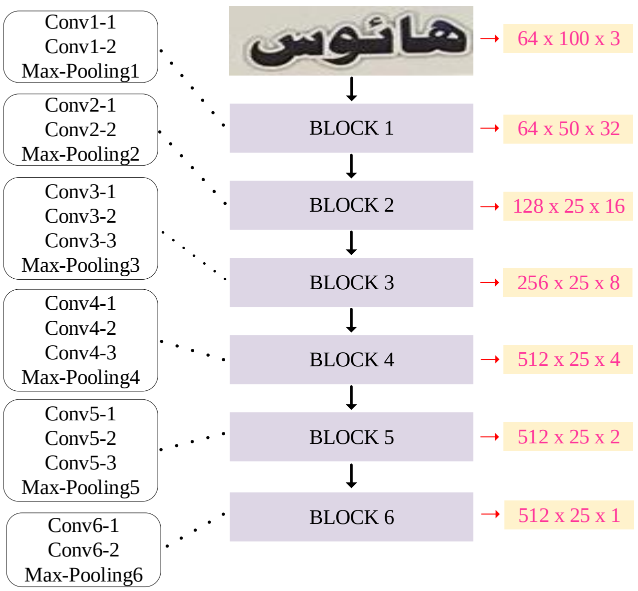


FIGURE 5: VGG-16 network with additional convolutional block implemented for cropped Urdu word recognition in natural images.

pooling layer. The standard VGG-16 network uses a 3 × 3 kernel size in the convolutional layers and a 2 × 2 max pooling throughout the whole network. However, the VGG-16 network implemented for cropped Urdu word recognition uses different max pooling 2 × 2 and 2 × 1) windows.

2) VGG-16 Network with Shortcut Connections

VGG-16 is a sequential feature extraction network that uses a stack of convolutional and max pooling layers to extract the features sequentially from the top layer to the bottom layer. As previously explained, increasing the depth of the VGG-16 network causes the problem of gradient vanishing and degrades the network performance while it is training. This is due to performing repeated multiplications on the gradients, which make their values very small when back-propagated to the earlier layers. To handle the degradation problem, a novel VGG-16 architecture is proposed that incorporates shortcut connections. The differences between the structures of the convolutional layers of the VGG-16 network and the proposed VGG-16 network with shortcut connections are shown in Fig. 6. The output feature vector o of the standard VGG-16 network is defined as

$$o = f\Big(x, W\Big) \tag{1}$$

where x is the input vector of the previous layer and W is the weight parameters of the learned features. f is a mapping function that learns the best values of the W and maps x into o. Further, x can be added after the f operation as described in [39]. Hence, in the proposed VGG-16 network, the output feature vector o is defined as

$$o = F\Big(x, \{W_i\}\Big) + x_l \tag{2}$$

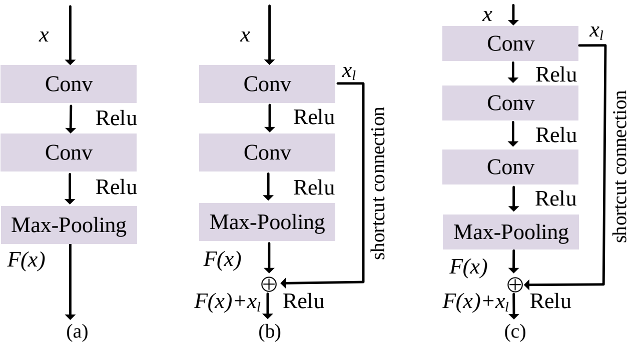


FIGURE 6: Different structures of the convolutional layers in the VGG-16 networks. (a) Structure of convolutional layers in a standard VGG-16 network, (b) proposed structure of the convolutional layers in the first two VGG-16 blocks and (c) proposed structure of the convolutional layers in the last four VGG-16 blocks.

where W_i is the weight parameters of the i^{th} convolutional layer and x_l is the output feature vector of first convolutional layer in the l^{th} block.

3) Residual Networks

The core idea of the ResNet is to introduce an identity shortcut connection in the network that can skip one or more layers, as illustrated in Fig. 7. This connection adds the output of previous layer(s) to the outputs of the next layer without increasing network parameters. The ResNets use two types of shortcut connections. First, the identity mapping is directly performed when the dimensions of the input and output are equal, as

$$o = F\Big(x, \{W_i\}\Big) + x \tag{3}$$

where o and x are the output and input feature vectors of the layers considered. Second, when the input and output have different dimensions, the shortcut connections still perform identity mappings either by padding extra zeros to equal the dimensions or by a linear projection W_s of the shortcut connections to make the dimensions equal using 1 × 1 convolutions as

$$o = F\Big(x, \{W_i\}\Big) + W_s x \tag{4}$$

This second type of the shortcut connection using 1 × 1 convolution adds extra parameters in the form of W_s. It is possible to represent multiple convolutional layers in the function $F\Big(x, \{W_i\}\Big)$. However, the element-wise addition is performed on two feature maps, i.e., the output of the previous layer and the next layer. Therefore, the spatial dimensions of both feature maps must be the same. A structure of the residual block using two convolutional layers proposed for ResNet-18 and ResNet-34 architectures is shown in Fig. 7(a), while Fig. 7(b) shows the typical structure of the residual block with three convolutional

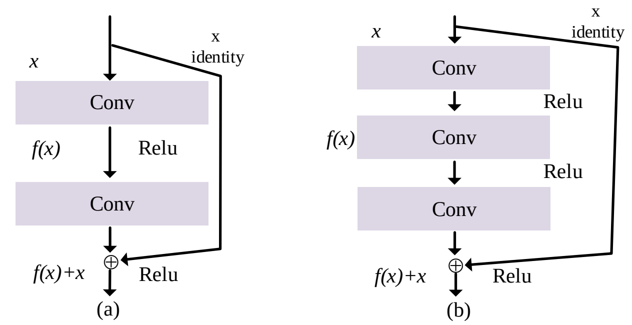


FIGURE 7: The difference between the structures of residual blocks in ResNet models. (a) ResNet-18 and ResNet-34 with two convolutional layers, and (b) ResNet-50, ResNet-101 and ResNet-152 with three convolutional layers.

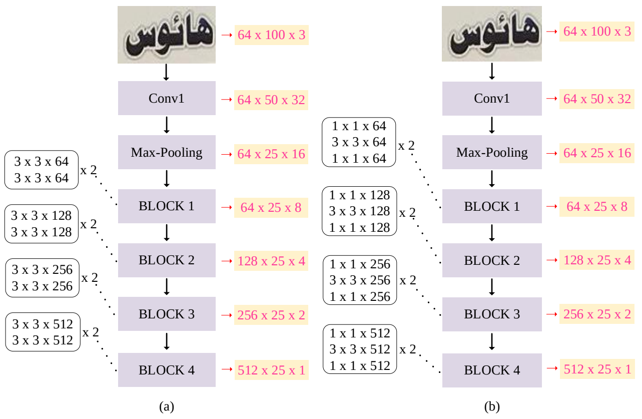


FIGURE 8: (a) ResNet-18 architecture with two convolutional layers in each residual block and (b) the proposed ResNet-18 architecture with three convolutional layers in each residual block and a max pooling layer of 2×1 following every block to reduce the feature maps vertically.

layers proposed for ResNet-50, ResNet-101 and ResNet-152 architectures.

Fig. 8(a) and Fig. 8(b) show the original and proposed architectures of the ResNet-18 network respectively. The original ResNet-18 architecture uses eight residual blocks, each with two convolutional layers of 3×3 kernel size, followed by rectified linear unit (ReLU) activation and batch normalisation layers. The proposed ResNet-18 uses three convolutional layers of 1×1, 3×3 and 1×1 kernel sizes in the residual blocks. The original ResNet-18 uses only one convolutional layer of 7×7 kernel size and a stride of 2, followed by a max pooling layer of 2×2. The proposed ResNet-18 uses the first convolutional and max pooling layers as in ResNet-18, but also uses an additional max pooling layer with 2×1 after every residual block to reduce only the height of the feature map and retain a fixed width.

Following the baseline ResNet [24] model, different types of shortcut connections are proposed by changing the arrangements of the batch normalisation, ReLU and convolutional layers [40]. In the baseline model, the activations are applied after the convolutional layers, while in [40], the activations are placed at different positions, including before convolutional layers, as shown in Fig. 9(b) and Fig. 9(c). Moreover, the post-activation residual block works effectively when the network depth is small, such as in ResNet-18 and ResNet-34, while the pre-activation residual block yields better results when the network is deeper, as in ResNet-101, ResNet-152 or ResNet-1001 [40]. Fig. 9 (a) shows the structure of the residual block used in ResNet-18, while Fig. 9(b) and Fig. 9(c) show the structures of residual blocks with pre- and post-activations implemented for cropped Urdu scene text recognition. During experiments, the effect of both pre- and post-activation structures is examined. The proposed ResNet-18 network with post-activation units achieves the best recognition accuracy than pre-activation units. Therefore, the residual block with post-activation units, as shown in Fig. 9(c), is selected in all the experiments regarding cropped Urdu scene text recognition.

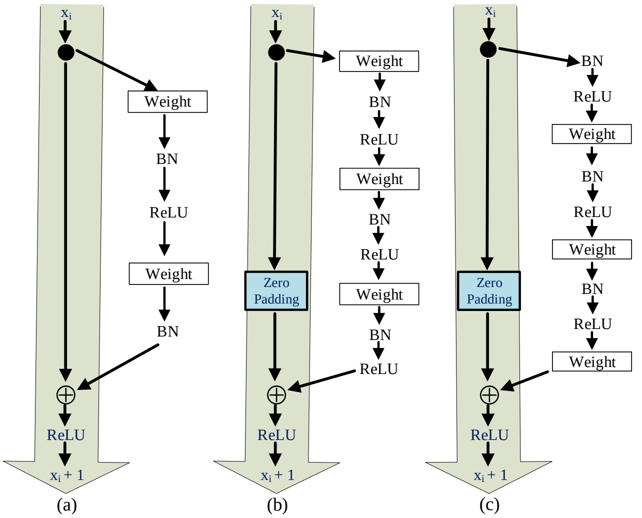


FIGURE 9: Different architectures of the residual blocks used for cropped Urdu scene text recognition. (a) Original residual block used in ResNet-18, (b) residual block with batch normalisation and ReLU activations used after the convolutional layers (post–activation units) and (c) Residual block with batch normalisation and ReLU activations used before the convolutional layers (pre–activation units).

B. SEQUENCE LABELLING

The recurrent layers in the RNN consist of a set of hidden units with cyclic connections—that is, the activations a_k^t of hidden units k at time step t depend on the state of the current input x_j^t at time t and the activations of the state of the previous hidden units at time $t-1$, stated as a_i^{t-1}. Therefore, for an RNN layer with N inputs and M hidden units, the activations a_k^t are calculated as

$$a_k^t = \sigma\left(\sum_{j=1}^{N} w_{jk}x_i + \sum_{i=1}^{M} w_{ik}a_i^{t-1}\right) \tag{5}$$

where σ is an activation function and w_{jk} and w_{ik} are the weights of the current input and the previous state, respectively. This type of cyclic connection in the structure of the network allows it to persist its previous internal state. The outputs of the network at time step t are calculated in parallel by obtaining the activations of the hidden layer a_k^t, $k \in \{1, 2, \cdots, M\}$ as input. This shows that the network's outputs are influenced implicitly by its current inputs and previous states.

Although RNNs use a feedback loop within the recurrent layers to persist information in the memory, when trained on long-range sequence problems, they suffer from gradient vanishing and exploding problems [41]. This is due to the exponential decrease in the gradients of the loss function over a time period. To overcome these problems, LSTMs [42] were introduced, which used gates to store the current and previous state of the memory cell. LSTMs can store long-range information; however, they can store it only in one direction—from the past—as they receive the input from the previous state. In some tasks, such as text recognition, speech recognition and natural language processing, both the future and past information is required to make accurate predictions. To tackle this problem, [43] proposed a special LSTM architecture: BLSTM. In BLSTM, the hidden recurrent layer was replaced with two hidden layers: a forward layer that processed the input sequences from the past to future time steps and a backward layer that processed the sequences from the future to past-time steps in the opposite direction. The forward and backward layers were not directly connected to each other, but their outputs were connected to the same activation function in the output layer. At each time step, the state h_t of the cell was updated by taking the current features x_t and the previous state of the cell h_{t-1} or h_{t+1} as inputs, such that

$$\begin{cases} h_t^{(f)} = LSTM_1(x_t, h_{t-1}^{(f)}) \\ h_t^{(b)} = LSTM_2(x_t, h_{t+1}^{(b)}) \end{cases} \tag{6}$$

where (f) and (b) are the forward and backward recurrent layers.

Fig. 10 shows the architecture of a BLSTM and BiGRU cell used in this paper. The features extracted either by the VGG-16 or ResNet models from the given images are passed to the RNN layer to be decoded into feature sequences. Both the BLSTM and BiGRU are implemented as a part of the RNN. Since the BLSTM and BiGRU take the feature sequences in both forward and backward time steps, each character of scene text recognition in the predicted sequence considers the context before and after time step x_t. This results in a decreasing text recognition error rate. The BLSTM and BiGRU are similar and use gates, but they differ in terms of the number of gates; the BiGRU has two gates, while the LSTM has three. A BiGRU does not use an output gate and has fewer parameters, which makes it faster to train. On some sequence recognition problems, it may outperform the LSTM when the training data samples are small. The experimental results in Section V-E compare the effectiveness of LSTM, BLSTM and a BiGRU for cropped Urdu text recognition in natural scene images. The VGG-16 and ResNet-34 networks with BLSTM have word recognition rates better than those with the LSTM and BiGRU, respectively. Therefore, BLSTM is used as part of the RNN network in the experiments.

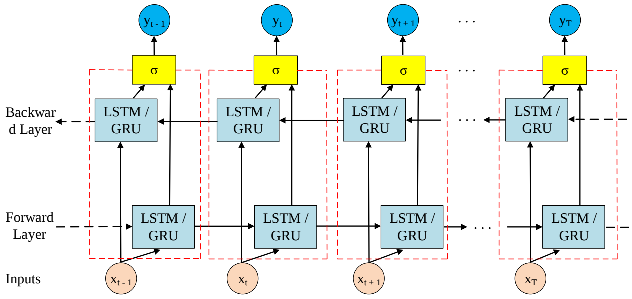


FIGURE 10: Architecture of the BLSTM and BiGRU implemented for cropped Urdu text recognition in natural scene images.

Unlike the network model in [33], the cropped Urdu text recognition framework uses a single BLSTM layer on top of the feature extraction component to decode the feature sequences. Using a single BLSTM layer performs better on scene text recognition and reduces the training time and the network parameters. To select the proper number of hidden units in the BLSTM, the proposed models are trained with different numbers of hidden units such as 128, 256, 512 and 1024. The comparative analysis of the different number of hidden units in the test set is described in Section V-D. Finally, the number of hidden units in the BLSTM is set to 512 in all the experiments. A Softmax function is applied on the output states of the BLSTMs to transform them into the probability distributions of 96 character classes as

$$p_t(c = c_j|x_t) = \text{Softmax}\left(\left[h_t^{(f)}, h_t^{(b)}\right]\right) \tag{7}$$

where $h_t^{(f)}$ and $h_t^{(b)}$ are the forward and backward hidden states of LSTMs at time step t, $j = 1, 2, \cdots, 96$ and $t = 1, 2, \cdots, N$. A Softmax function concatenates these hidden states together and transforms them into the probability distributions of $p = \{p_1, p_2, \cdots, p_N\}$.

C. TEXT TRANSCRIPTION

Although LSTM networks are sufficiently powerful to perform classification tasks on sequential data, the major limitation of these networks is that they require pre-segmented training samples and a post-processing operation to transform the output predictions of the network into the sequence of labels. Since the output of the BLSTM is a score for each time step at a horizontal position in an image, it is, therefore, necessary to specify the position of each of the characters of the ground truth text in an image while network training. For cursive scripts such as Arabic and Urdu, it is more difficult to segment each character of the ground truth text in an image due to the connected text and overlapping ligatures. For example, if an image contains a piece of three-character text صدف it is necessary to specify where the character صـ starts and ends (e.g., starts from pixel 20 and ends at pixel 35). The same process is performed for the remaining characters in the ground truth text. This becomes more complex when an image contains long sequences of characters such as کمیونیکیشن.

Another problem related to LSTM network-based sequence classification of cursive text is the length of time steps for each character; that is, if a character is horizontally stretched (which is common in Urdu text, see Section I), it increases the width of the character. Hence, each character occupies multiple time steps. In this case, transforming the output scores of the LSTM for each character at every time step will probably yield more incorrect results. Further, if the ground truth text contains consecutive duplicate characters in the same word, removing all the duplicate values while decoding the LSTM outputs also leads to an incorrect result. For example, the word سستی has a duplicate character سسـ appearing consecutively. If one of them is removed during decoding, the resultant text transcription would be ستی, which is an incorrect output.

To overcome these problems, a temporal classification method called CTC [44] is used for cropped Urdu text recognition. The CTC has been commonly used in several sequence-to-sequence recognition problems including speech recognition [45]–[47], handwritten text recognition [48] and natural scene and video image text recognition [37], [49], [50]. The purpose of CTC is to label unsegmented data sequences through the LSTM or RNNs without requiring pre-segmented data to train the network or a post-processing operation to merge the individual recognised characters into the complete output sequences. The network is trained from the pairs (I, G) without specifying the relative position of each character or the width of text in the ground truth by using a CTC cost function. The CTC cost function uses an additional special character called 'blank', denoted as '-' in the sequences, to specify that no character exists at the specified time step. Thus, the output layer of the CTC has $n+1$ the number of nodes for n character classes. The ground truth text G is then modified to $G^{'}$ by inserting the blanks. As the length of the ground truth label sequences should be less than the input feature sequences, there are many possible ways to repeat the characters into their correct label sequences. This way, each character may occupy several time steps in the image. The blanks are also inserted between the characters that occur repeatedly, eliminating the problem of removing repeated characters in the ground truth text. For example, one possible way to align the word سستی could be -- ﺳ -- ﺳ ﺳ ﺳ ﺳ -- ﺗ ﺗ ﺗ ﺗ --- ی ی -- ; when removing all the duplicate characters and then inserting the blanks, the decoded output will be a correct ground truth text.

The CTC interprets the per-frame predictions of the LSTM as a probability distribution score for every possible $G^{'}$ text, then sums overall scores, which yields the loss for the pair (I, G), conditioned on a given input sequence. This probability distribution is then used to directly draw a cost function to maximise the probabilities of the correct label alignments. Since the cost function is independent of the neural network architecture, a CTC layer can then be added in any network and trained with the back-propagation through time (BPTT) [51] algorithm.

The cost function C in the CTC is defined as the negative log probability of the network correctly labelling the entire training set, such that

$$C = -\sum_{(x,g)\in S} \ln P\Big(g|x\Big) \tag{8}$$

where S is the entire training set consisting of input sequences x and target sequences g. $P\Big(g|x\Big)$ is the conditional probability of achieving the target sequences g through the input sequences x. The objective of the cost function is to minimise C, which is equivalent to maximising $P\Big(g|x\Big)$.

Since the cost function is differentiable from the inputs, the probabilities p of the output activations of the BLSTM are directly given as the inputs to the cost function C, such that:

$$P(g|x) = \sum_{\pi:\beta(\pi)=g} P\Big(\pi|p\Big) \tag{9}$$

where π is the path of each sequence in $G^{'}$ and β is an operator that decodes the sequences and removes the duplicates and all the blanks from the sequence path. This is a many-to-one mapping from $G^{'}$ to the set of sequences with lengths less than or equal to G. For example, $\beta($ سستی $) = \beta($ -- ﺳ -- ﺳ ﺳ ﺳ ﺳ -- ﺗ ﺗ ﺗ ﺗ --- ی ی -- $) =$ $\beta($ -- ﺳ -- ﺳ -- ﺗ --- ی -- $) =$ سستی.

This part of the network is then trained with a gradient descent and the BPTT algorithm. Once the network is trained, the goal of the sequence decoding is to find the best path π that has the maximum probability of replicating the ground truth sequence labelling though the BLSTM sequence outputs as

$$h(x) = \beta(\pi^{*}) \tag{10}$$

where $\pi^{*} = \underset{\pi\in N^{t}}{arg\,max}\, P\Big(\pi|x\Big)$, which is a concatenation of the most active outputs at each time step t. The CTC loss is visually represented in Fig. 11. The top illustration shows the probability computation of the CTC for the BLSTM output sequences at time t for an Urdu word عائزہ.

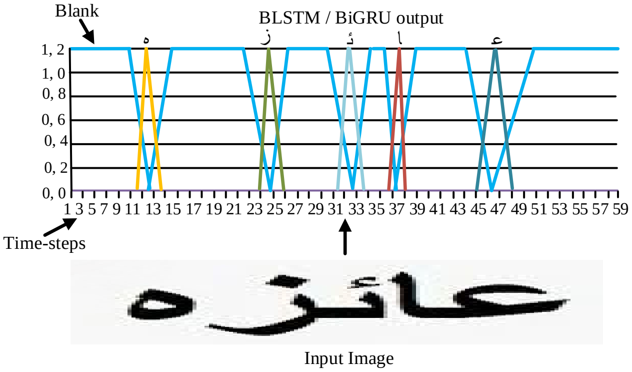


FIGURE 11: Visual representation of the CTC loss. The probabilities of the BLSTM and BiGRU output sequences are computed as a sum over all the possible alignments of the input sequences that can be mapped to the output sequences by considering that the ground truth labels may occupy several time steps due to character stretching.

In addition to training RNN networks like BLSTMs without pre-segmented data, the CTC also facilitates the network in searching correct labels. These models, referred to as discriminative models, have many advantages over generative models such as Hidden Markov Model (HMM) [52]. Moreover, the RNN- and CTC-based discriminative models are able to directly calculate the posterior class probabilities $P\Big(class|x\Big)$ over the entire input sequences, whereas the generative models first define the class conditional probability densities $P\Big(x|class\Big)$ of each observation only on their current states and then apply Baye's theorem to imply this to posterior probabilities.

IV. NETWORK TRAINING

In cropped Urdu scene text recognition, the BLSTM and BiGRU networks with a CTC loss function are implemented separately. The output feature vector of the convolutional network has a dimension of 1×25, which is given as input to the BLSTM network. This input is fully connected to the forward and backward hidden layers of the BLSTM of 256 cells each. The size of the cells in the hidden layers is set empirically by performing several experiments. The output of both the forward and backward hidden layers is connected with the bidirectional dynamic recurrent layer that uses the Softmax classifier to predict the probabilities of each character class at each time step. Every character at each position of the word is considered a class or a label. Hence, for cropped Urdu text recognition, there are 96 different classes (including characters, digits and some symbols) and an additional special class for the 'blank'. The network is trained with an SGD using momentum of 0.9 and an initial learning rate of 0.005. The learning rate is exponentially decreased after every 7000 iterations by using an exponential decay method. The batch size was set to 32 and the network was trained upto 25000 epochs. The network uses BPTT to calculate the error differentials in the BLSTM part of the model. While network training, the value of cost and the edit distance between the ground truth labels and the validation labels is measured, so that the network generalisation can be observed and the over-fitting problem avoided.

Once the network is trained, a CTC-based the best path decoding technique is applied to the output of the BLSTM Softmax sequence predictions. The decoder concatenates the most probable characters at every time step and removes the duplicate characters and all the blanks to yield the final recognised text.

V. EXPERIMENTAL SETUP AND RESULTS

To demonstrate the effectiveness of the proposed methods, different experiments were conducted on a cropped Urdu natural scene text recognition dataset. The experiments were implemented on an NVIDIA GeForce GTX 1080 Ti with 12 GB of GPU memory using an open-source TensorFlow library in python language.

A. IMPLEMENTATION DETAILS

The implementation of the proposed networks was based on the VGG-16 [23], ResNet-18 and ResNet-34 [24] networks. The architecture of the convolutional layers in the first proposed network is similar to that of VGG-16, with additional convolutional and max pooling layers. Moreover, in the proposed network, a batch normalisation [38] layer is added after every convolutional layer. The VGG-16 network does not change the dimensions of the feature maps in the convolutional layers and uses max pooling layers with 2×2 kernel size and 2×2 stride to reduce the dimensions. In the cropped Urdu word image dataset, the dimensions of the input images are $64 \times 100 \times 3$. Therefore, the proposed network uses six max pooling layers, wherein the first two pooling layers are unchanged and use the same 2×2 kernel size with 2×2 stride for pooling. The remaining four max pooling layers use a 2×1 kernel size with 2×1 stride to ensure that the width of the feature maps is not down-sampled. Thereby, the width of the final feature map is reduced to 25 pixels, which is one-quarter of the actual image width and the height is reduced to 1. The purpose of using the first two max pooling layers with a 2×2 kernel size and 2×2 stride is to reduce the dimensions of the feature maps in the early layers, which will affect the reduction of model computations. The final feature map of the network outputs has $1 \times 25 \times 512$ dimensions.

In addition to the VGG-16 network being used as part of the CNN, a deeper network VGG-19 is tested, under the assumption that a deeper network will improve the recognition accuracy of the model. The architecture of the VGG-19 is similar to the VGG-16; however, it uses an additional convolutional layer in the last three blocks. Compared to the VGG-16, the VGG-19 did not improve recognition accuracy—in fact, the accuracy slightly decreased. This proves that increasing the depth or number of layers in the network is not guaranteed to improve its feature extraction performance. However, deeper networks may take more time to train and learn from the data, causing a gradient vanishing problem. To avoid gradient vanishing and improve the recognition accuracy of the network, a new VGG-16 architecture with residual connections as implemented in [24] is proposed for cropped Urdu scene text recognition. In this architecture, shortcut connections are implemented to reuse the activations of the previous layers. The shortcut connections help the network to avoid the problem of gradient vanishing.

To further analyse the problem of cropped Urdu scene text recognition, two residual network architectures are proposed. In the experiments, ResNet-18, ResNet-34 and the modified ResNet architectures are implemented. Similarly to the proposed VGG-16 model, the ResNet models down-sample the width and height of the feature map by applying max pooling layers. The first convolutional layer in the ResNet models uses a 7×7 kernel size with a stride of 2 to halve the width and height of the input image. A max pooling layer with a 2×2 kernel size and a 2×2 stride value follows the convolutional layer to further down-sample the feature map by half. In the subsequent residual blocks, a max pooling layer with a 2×1 kernel size and a 2×1 stride value is used after every block to down-sample only the height of the feature map, while keeping the width unchanged. The final output obtained by the last ResNet block after applying the max pooling layer has the dimensions $1 \times 25 \times 512$, the same as the output of the VGG-16 network.

After the convolutional layers, a single BLSTM layer with forward and backward layers is used to decode the feature sequences. The output sequences of the BLSTM are concatenated and a Softmax layer follows, which transforms the BLSTM output sequences into the probability distributions over 96 classes. Finally, a CTC layer is used to transform the probability distributions into the sequence of characters.

As the ResNet architectures do not use max pooling layers except after the first convolutional layer, in this research work, the max pooling layers used after the residual blocks are replaced with strided convolutional layers. This increases the number of network-trainable parameters and improves the expressiveness of the model. The max pooling layers are replaced with the strided convolutions in various image recognition benchmarks without decreasing the recognition accuracy [53].

B. DATASET

To train the proposed models, we photographed more than 2500 natural scene images and developed a new dataset of 14100 cropped Urdu word images. Some samples of photographed images are illustrated in Fig. 1. Due to the inter and intra character overlapping, all the word images were manually segmented and resized to 100×64 pixels. Some examples of the segmented word images from the Urdu natural scene images are illustrated in Fig. 12. The dataset consists of a huge number of Urdu word images with the possible number of text variations. The dataset also has several images with handwritten Urdu text written on the walls and signboards. Several cropped word images have stretched, intra-lingature, inter-ligature and diagonal text. The stretched, overlapping, diagonal and handwritten text when available in the natural scene images is more complex to recognize than the plain and typewritten (superimposed) text. This is the first dataset that contains large number of cropped Urdu word natural scene images, hence it can be used as a benchmark. Moreover, the dataset can be used for the natural scene text recognition of other cursive languages such as Arabic, Persian and Sindhi. The dataset was divided into 12600 training and 1500 testing samples. To speed up the network training and network convergence, the input features are normalised to 0 and 1 values. As deep networks require more training samples to provide better accuracy, the dataset is further increased by applying a data augmentation technique to rotate the images at random angles (not more than 10 degrees). The cropped Urdu word images generated using data augmentation were used only in the training set.

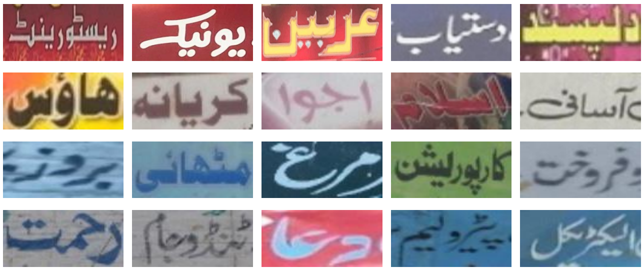


FIGURE 12: Some examples of segmented word images of Urdu text in natural scene images in our proposed dataset. The top two rows demonstrate machine printed Urdu text with variant font styles, while the bottom two rows demonstrate handwritten Urdu text written on the walls and signboards in natural scene images.

C. EVALUATION METRICS

Generally, two evaluation metrics have been used to measure the performance of scene text recognition systems: character-level evaluation metric measured as character recognition rate (CRR) and a word-level evaluation metric measured as word recognition rate (WRR). The latter evaluation metric is more rigorous, as it recognises the predicted word as a correct when each character in the ground truth label is identified correctly. However, for CRR, the evaluation metric measures the distance between the predicted text and the ground truth text, where the least distance is considered to be the best. As the proposed methods for cropped Urdu text recognition are segmentation-free, the performance of the cropped word recognition was evaluated on the basis of the CRR as

$$\text{CRR} = \frac{N_{\text{char}} - \sum ED(P_T, G_T)}{N_{\text{char}}} \times 100\% \tag{11}$$

where N_{char} is the number of characters, ED is the edit distance, P_T and G_T are the predicted text labels and the ground truth text labels, respectively.

The performance of the cropped word recognition was evaluated in terms of the WRR as

$$\text{WRR} = \frac{N_{\text{cword}}}{N_{\text{word}}} \times 100\% \tag{12}$$

where N_{cword} is the number of correctly recognised words and N_{word} is the total words in a test set. While evaluating the proposed method, it was observed that the WRR does not perform a fair evaluation, since it considered a large number of recognised words false if a single character among them is not recognised correctly. Therefore, in this paper, an additional evaluation metric for WRR was considered as

$$\text{WRR}_{\text{1F}} = \frac{N_{\text{cword}} + N_{\text{1F}}}{N_{\text{word}}} \times 100\% \tag{13}$$

where N_{1F} is the number of correctly recognised words with one false character. In this way, if the evaluation metric incorrectly recognises one character in a word, it is considered as a correct word. If it recognises more than one character as wrong, the whole word text is considered incorrect.

D. SELECTING NUMBER OF RNN HIDDEN UNITS IN BLSTM

As described in Section III-B, the modified models were trained with different numbers of hidden units in the BLSTM network. The accuracy of cropped Urdu scene text recognition with VGG-16 using four different numbers of hidden units 128, 256, 512 and 1024 is shown in Table 1. The number of hidden units in the BLSTM affected the accuracy of the model. This table shows that the VGG-16 Model 4 used 1024 hidden units and achieved the highest CRR and WRR of 94.63% and 86.47%, whereas Model used 128 hidden units and produced the lowest CRR and WRR of 91.35% and 73.40%, respectively. Model 3 used 512 hidden units and achieved CRR and WRR of 93.83% and 86.05%. The WRR of Model 3 is slightly less than the Model 4. The test accuracy of Models 3 and 4 are almost equal, and in this case, increasing the number of hidden units in the BLSTM does not much improve the performance of the model. However, increasing the number of hidden units in the BLSTM does increase the number of network parameters and computation. Therefore, the number of hidden units in the BLSTM for all the experiments were set to 512.

Due to the various shapes of the same character and similarity in the baseline structure of several characters, the WRR performed unfair evaluations, i.e., if a single character was not recognized correctly, the whole word text was considered as incorrect. When evaluated with a new metric as presented in eq. 13, all the models improved the performance of WRR. As shown in the last column in Table 1, the accuracy of Models 1, 2, 3 and 4 improved by 11.99%, 6.56%, 6.36% and 5.83%, respectively. This improvement in the WRR indicates that the recognition performance of the network models can be improved by using a language model in a post-processing step.

E. SELECTING CNN MODELS

After selecting the use of 512 hidden units for the BLSTM, four more models were implemented as shown in Table 2. Models 5 and 6 used VGG-16 and VGG-19 network, while Models 7 and 8 used ResNet-18 and ResNet-34 with additional residual blocks. Each model was followed by a BLSTM layer. The results in Table 2 indicate that the recognition accuracy of the models decreases as the number of convolutional layers are increased. In Model 6, the recognition accuracy slightly decreased when three more layers were added to Model 5. Moreover, when the residual networks were added as a part of the CNN, the accuracy of Models 7 and 8 was decreased as compared to Models 5 and 6. Both Models 7 and 8 used the structure of the residual block with two convolutional layers, as illustrated in Fig. 7. For the cropped Urdu scene text recognition problem, simply increasing the number of layers or residual blocks in the ResNet is not effective. One possible reason for this could be the lower amount of training samples, since the proposed networks are trained on $25,200$ samples only.

To further analyse the effect of different RNN structures, different experiments were conducted on the LSTM, BLSTM and BiGRU cells. A comparative analysis of these RNN structures is shown in Table 3. Models 10, 11, 13, 14, 16 and 17 contained a BiGRU and LSTM after the CNN part of the network. The BiGRU is a bidirectional type of RNN with similar performance to the BLSTM, whereas the LSTM uses the contextual information in one direction, and preserves only past information. Compared to the BLSTM and BiGRU models, the performance of the LSTM models was the worst. Moreover, the LSTM-based models took more time to converge than the BLSTM or BiGRU models. Therefore, LSTM was not considered a part of the RNN in the experiments.

Typically, ResNet architectures apply only one max pooling layer after the first convolutional layer. However, in the experiments, both max pooling and convolutional layers with strides were used after every residual block to down-sample the width and height of the feature map. The kernel size and stride values in the max pooling and convolutional layers were set to 2×1, so that the width of the feature map became consistent and only its height was down-sampled. The performance accuracy of applying max pooling and convolutional layers with strides is shown in Table 4. Model 19 and Model 21 used the strided convolutional layers and have an accuracy improvement of 0.32% and 1.04% in terms of the WRR than the Model 18 and Model 20, respectively. The last column in Table 4

TABLE 1: Text recognition accuracy comparison between different numbers of RNN hidden units in the BLSTM network.

Model	CNN Type	RNN Structure	No. of RNN Hidden Units	CRR (%)	WRR (%)	WRR_{1F} (%)
1	VGG-16	BLSTM	128	91.35	73.40	85.39
2	VGG-16	BLSTM	256	93.61	84.26	90.82
3	VGG-16	BLSTM	512	93.83	86.05	92.41
4	VGG-16	BLSTM	1024	94.63	86.47	92.30

TABLE 2: Text recognition accuracy comparison between different CNN models.

Model	CNN Type	RNN Structure	No. of RNN Hidden Units	CRR (%)	WRR (%)	WRR_{1F} (%)
5	VGG-16	BLSTM	512	93.83	86.05	92.41
6	VGG-19	BLSTM	512	93.37	85.73	91.79
7	ResNet-18	BLSTM	512	92.32	80.04	89.56
8	ResNet-34	BLSTM	512	91.27	83.00	90.08

TABLE 3: Text recognition accuracy comparison between different RNN architectures.

Model	CNN Type	RNN Structure	No. of RNN Hidden Units	CRR (%)	WRR (%)	WRR_{1F} (%)
9	VGG-16	BLSTM	512	93.83	86.05	92.41
10	VGG-16	BiGRU	512	93.37	84.71	91.79
11	VGG-16	LSTM	512	89.72	79.53	87.13
12	ResNet-18	BLSTM	512	92.32	80.04	89.56
13	ResNet-18	BiGRU	512	92.59	80.92	89.17
14	ResNet-18	LSTM	512	88.75	77.19	85.57
15	ResNet-34	BLSTM	512	91.27	83.00	90.08
16	ResNet-34	BiGRU	512	91.00	82.66	88.76
17	ResNet-34	LSTM	512	85.56	77.12	83.32

TABLE 4: Text recognition accuracy comparison between max pooling and strides convolutions.

Model	CNN Type	RNN Structure	No. of RNN Hidden Units	CRR (%)	WRR (%)	WRR_{1F} (%)
18	ResNet-18 + Max Pooling	BLSTM	512	90.32	79.72	87.80
19	ResNet-18 + Conv with strides	BLSTM	512	92.32	80.04	89.56
20	ResNet-34 + Max Pooling	BLSTM	512	92.96	81.96	89.94
21	ResNet-34 + Conv with strides	BLSTM	512	91.27	83.00	90.08

TABLE 5: Text recognition accuracy of the proposed VGG-16 with skip connection and ResNet-18 with v2 residual block.

Model	CNN Type	RNN Structure	No. of RNN Hidden Units	CRR (%)	WRR (%)	WRR_{1F} (%)
Proposed I	VGG-16 with skip connections	BLSTM	512	95.75	87.13	94.21
Proposed II	ResNet-18 with v2 residual block	BLSTM	512	94.03	84.42	92.30

shows the performance of the models in terms of the WRR_{1F} evaluation metric.

F. TEXT RECOGNITION RESULTS OF THE PROPOSED MODELS

To further improve the text recognition accuracy, two additional models were proposed. The first model proposed a new VGG-16 network with residual connections, as shown in Fig. 6(b) and Fig. 6(c). The features of the first convolutional layer in every block were added with the features of the last convolutional layer in the same block. These features were then passed to the next block. The accuracy of the proposed new VGG-16 model (proposed I) is summarized in Table 5. This model improved the CRR, WRR and WRR_{1F} to 1.92%, 1.08% and 1.80% respectively over the standard VGG-16 model as shown in Table 2. Similarly, in the second proposed model (proposed II), the standard ResNet-18 architecture was modified with the residual block containing three convolutional layers. Table 5 shows that this modified architecture (proposed II model) improved the CRR, WRR and WRR_{1F} to 1.71%, 4.38% and 2.74% respectively over the standard ResNet-18 model as shown in Table 2. Although the use of residual connections with deep VGG-16 network improved the recognition accuracy

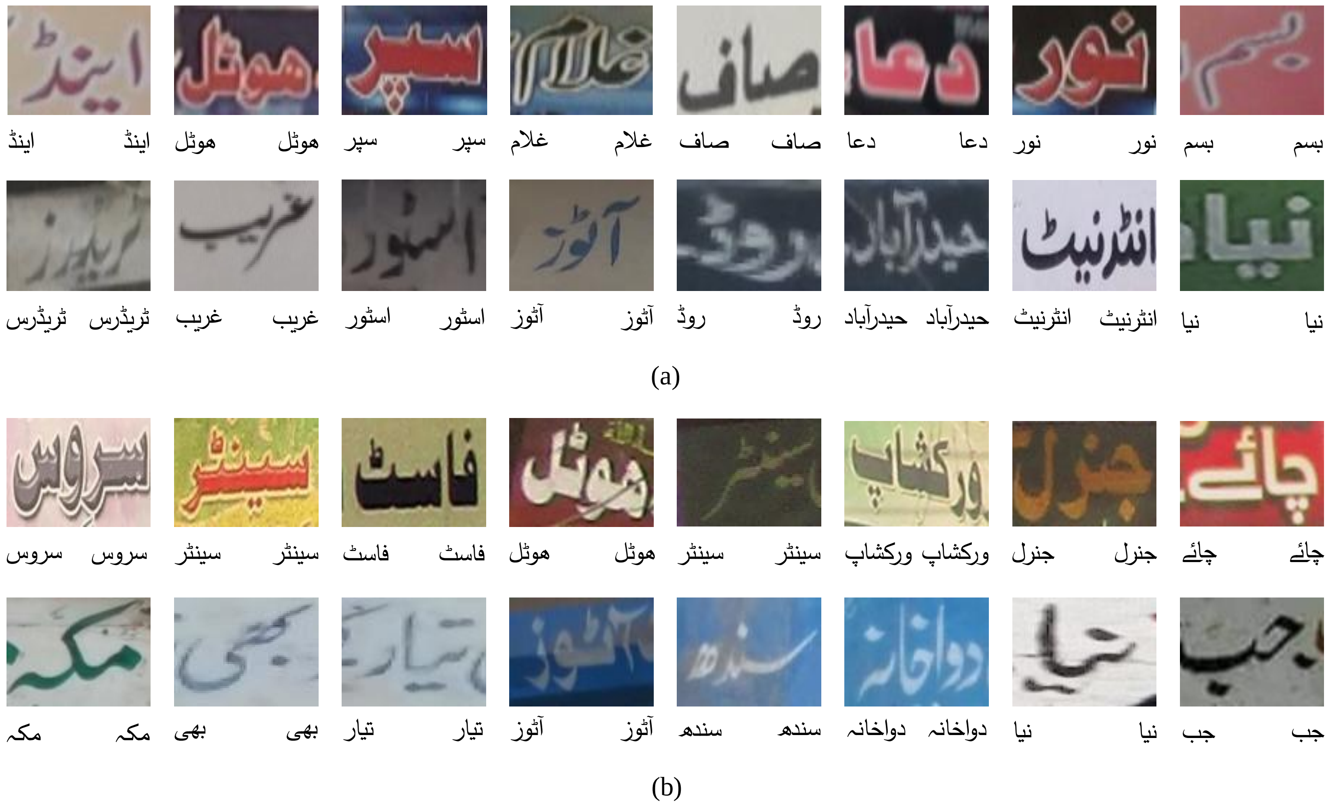

اینڈ اینڈ ہوٹل ہوٹل سپر سپر غلام غلام صاف صاف دعا دعا نور نور بسم بسم
ٹریڈرس ٹریڈرس غریب غریب اسٹور اسٹور آٹوز آٹوز روڈ روڈ حیدرآباد حیدرآباد انٹرنیٹ انٹرنیٹ نیا نیا
(a)
سروس سروس سینٹر سینٹر فاسٹ فاسٹ ہوٹل ہوٹل سینٹر سینٹر ورکشاپ ورکشاپ جنرل جنرل چائے چائے
مکہ مکہ بھی بھی تیار تیار آٹوز آٹوز سندھ سندھ دواخانہ دواخانہ نیا نیا جب جب
(b)

FIGURE 13: Qualitative results of correctly recognised cropped Urdu scene text using the (a) proposed I model and (b) proposed II model. For each word image, the annotations at the bottom left are the ground truths, while those at the bottom right are the predicted text.

than the standard model, increasing the number of layers in the network reduced the recognition accuracy as shown in the second row of Table 5. Moreover, all the network models have been trained on the augmented data. We only compare the results of proposed model I and II as shown in Table 5 with and without data augmentation. The WRR and WRR_{1F} obtained without data augmentation using proposed model I were 74.24% and 84.36%, respectively, while with data augmentation, WRR and WRR_{1F} were 87.13% and 94.21%, respectively. Similary, the WRR and WRR_{1F} without data augmentation using proposed model II were 72.88% and 84.22%, respectively, while with data augmentation, the WRR and WRR_{1F} were 84.42% and 92.30%, respectively.

A qualitative analysis of the correctly and incorrectly recognised words using the proposed models are illustrated in Fig. 13 and Fig. 14 respectively. In Fig. 13, the annotations in the bottom-left corner of each word image are the ground truths, while those at the bottom-right corners are the predicted text. The first rows in Fig. 13(a) and Fig. 13(b) show correctly recognised machine-printed cropped Urdu scene text word images with the proposed I and proposed II methods, respectively. The second rows in Fig. 13(a) and Fig. 13(b) show correctly recognised handwritten Urdu text in natural images (e.g., on walls or signboards) using the proposed I and proposed II models, respectively. As shown in the first rows in Fig. 13(a) and Fig. 13(b), the images contain Urdu text with variations in font size, colour, alignment, writing style and background, which is difficult to recognise correctly. The handwritten text in the second rows in Fig. 13(a) and Fig. 13(b) includes additional complexities introduced by different handwritten styles.

The first rows in Fig. 14(a) and Fig. 14(b) show qualitative results of incorrectly recognised cropped Urdu scene text word images using the proposed I and proposed II methods, respectively. The green text at the bottom-left corners of images shows the characters that are not recognised by the proposed methods, while the red text at the bottom-right corners shows incorrectly recognised text. Fig. 14 demonstrates that Urdu word images contain high contrast, blurred and noisy text, and complex backgrounds. Some images contain stretched and very small text, which also makes Urdu scene text recognition more complex.

G. TEXT RECOGNITION PERFORMANCE COMPARISON

The following sections compare the performance of the proposed model I and model II with Arabic text in natural and video images, and the commercial multilingual OCR systems.

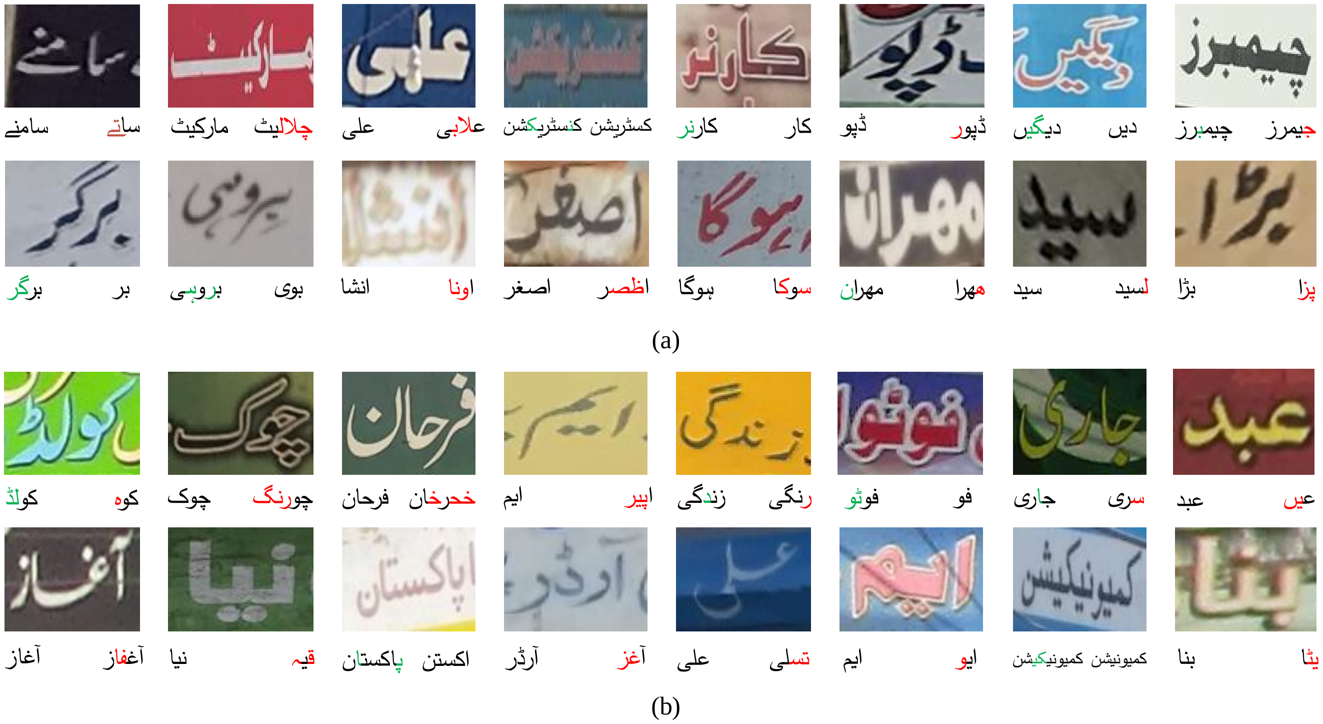


FIGURE 14: Qualitative results of incorrectly recognised cropped Urdu scene text using the (a) proposed I model and (b) proposed II model. Green text at the bottom left of a word image indicates the missing characters in the predicted text, while red text at the bottom right indicates the incorrectly recognised characters.

TABLE 6: Text recognition accuracy comparison of the proposed models with state-of-the-art methods developed for the Arabic text recognition in natural scenes and video images.

Model	Method	RNN Structure	Dataset	CRR (%)	WRR (%)	WRR_{1F} (%)
Yousfi et al. [54]	ConvNets	BLSTM	ALIF_Test	94.36	71.26	86.77
Yousfi et al. [55]	ConvNets	BLSTM	ALIF_Test2	90.71	65.67	–
Jain et al. [14]	ConvNets	BLSTM	Synthetic	75.05	39.43	–
Proposed I	ConvNets	BLSTM	Urdu-Text	95.75	87.13	94.21
Proposed II	ConvNets	BLSTM	Urdu-Text	94.03	84.42	92.30

1) Accuracy Compare with State-of-the-Art Methods

Since there is no existing work examining word-level Urdu text recognition in natural scene images, the performance of the proposed methods was compared with recently proposed state-of-the-art methods proposed for the Arabic text recognition in the natural scene and video images. Several samples in the ALIF train and test datasets [55] contain two or more text words or a sentence in a single image. These datasets have not been developed with isolated cropped word images. However, the authors have reported text recognition accuracy in terms of CRR, WRR and line recognition rate (LRR). Similarly, the synthetic Arabic text image dataset as reported in [14] contains full images with bounding box annotations and does not provide separate cropped word images. We, therefore, have compared the CRR and WRR of the proposed methods with the CRR and WRR of Arabic scene and video images as reported in [14], [54], [55]. Table 6 shows the comparison of text recognition accuracy of the proposed methods and Arabic text recognition in the natural scene and video images.

Yousfi et al. [54] proposed a CNN and BLSTM network to recognise Arabic video text. They evaluated their model on two test datasets: the ALIF_Test1 and the ALIF_Test2 [55]. A word recognition rate of 71.26% was reported for the ALIF_Test1, whereas the word recognition rate for the ALIF_Test2 was 65.67%. The performance of the model on the ALIF_Test2 was low because this dataset contains Arabic video images with more variety in terms of font size, text style, background complexity and colour. Next, Jain et al. [14] proposed a CNN and BLSTM network

TABLE 7: Text recognition accuracy comparison of the proposed models with off-the-shelf Tesseract OCR engine.

Method	CRR (%)	WRR (%)
Model 5	93.83	86.05
Model 6	93.37	85.73
Model 7	92.32	80.04
Model 8	91.27	83.00
Proposed I	95.75	87.13
Proposed II	94.03	84.42
Tesseract OCR	15.48	6.81

for synthetic Arabic text recognition in natural images. They created a synthetic dataset of four million Arabic scene text word images, using a list of commonly used Arabic words available over the internet. They evaluated the performance of the model by calculating CRR (75.05%) and WRR (39.43%). The model performed poorly in terms of WRR, due to the cursive nature and complexities of Arabic language. The proposed I and II models yield a WRR of 87.13% and 84.42% respectively for cropped Urdu scene text recognition, which is state-of-the-art than the method proposed for the Arabic text recognition in natural scenes and video images. Further, the CRR of the proposed I model is 95.75%, which is higher than all the methods, while the CRR of the proposed II model is slightly less than the model evaluated on ALIF_Test dataset as reported in [54].

2) Accuracy Compare with Commercial OCR System

A comparative study of the performance of the proposed methods against an off-the-shelf OCR system was also performed. A well-known OCR engine—the Google Tesseract [22] was tested on the cropped Urdu word image test dataset. The Tesseract OCR supports UTF-8 encoding and can recognise text in more than 100 languages, including Arabic and Urdu scripts. The results of the Tesseract OCR were evaluated according to CRR and WRR. The performance of the proposed methods and the VGG-16, VGG-19 and ResNets was much higher than that of the Tesseract OCR. Table 7 illustrates the performance of the proposed methods and the Tesseract OCR on the cropped Urdu natural scene text recognition dataset.

VI. CONCLUSIONS

This paper presented cropped Urdu word image text recognition solutions. A segmentation-free method was proposed that eliminated the problem of individually segmenting each character in a word image. The proposed framework was based on three components: feature extraction, sequence labelling and text transcription. For sequence extraction, two methods based on VGG-16 and ResNet networks were proposed. Further, a new VGG-16 architecture using shortcut connections was proposed that extracts more robust text features. Two RNN-based structures—a BLSTM and a BiGRU—were applied to decode the feature sequences into probability distributions. Finally, a CTC cost function was applied on top of the BLSTM or BiGRU sequences to transform per-frame predictions into the target sequence of labels. A BLSTM and CTC–based network learned temporal classification sequences without requiring pre-segmented data and the length of the ground truth text to train the network. It did not need a post-processing operation to merge the individual recognised characters into complete strings. The proposed methods were extensively evaluated on the new cropped Urdu scene text dataset and yielded high character and word recognition results. Since no research has been conducted regarding Urdu scene text recognition, the performance of the proposed methods was compared with that of existing Arabic natural scene and video text recognition methods. The method proposed with shortcut connections outperformed all other models in terms of CRR, WRR and $\mathrm{WRR}_{\mathrm{1F}}$. The proposed methods still produce some incorrect results due to the language complexities present in cursive scripts. These recognition errors can be improved by using a language model with linguistic information.

ACKNOWLEDGMENTS

The first author is thankful to the University of New South Wales, Australia, for supporting his Ph.D. candidature with a scholarship.

REFERENCES

[1] R.-C. Chen et al., “Automatic license plate recognition via sliding-window darknet-yolo deep learning,” Image and Vision Computing, vol. 87, pp. 47–56, 2019.

[2] C. Kang, G. Kim, and S. I. Yoo, “Detection and recognition of text embedded in online images via neural context models,” in Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[3] Y. Zhu, M. Liao, M. Yang, and W. Liu, “Cascaded segmentation-detection networks for text-based traffic sign detection,” IEEE transactions on intelligent transportation systems, vol. 19, no. 1, pp. 209–219, 2017.

[4] W. Feng, F. Yin, X.-Y. Zhang, and C.-L. Liu, “Semantic-aware video text detection,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1695–1705.

[5] S. Long, X. He, and C. Yao, “Scene text detection and recognition: The deep learning era,” arXiv preprint arXiv:1811.04256, 2018. [Online]. Available: https://arxiv.org/abs/1811.04256

[6] H. Lin, P. Yang, and F. Zhang, “Review of scene text detection and recognition,” Archives of Computational Methods in Engineering, pp. 1–22, 2019.

[7] X. Liu, G. Meng, and C. Pan, “Scene text detection and recognition with advances in deep learning: a survey,” International Journal on Document Analysis and Recognition (IJDAR), vol. 22, no. 2, pp. 143–162, 2019.

[8] O. Zayene, S. M. Touj, J. Hennebert, R. Ingold, and N. E. B. Amara, “Multi-dimensional long short-term memory networks for artificial arabic text recognition in news video,” IET Computer Vision, vol. 12, no. 5, pp. 710–719, 2018.

[9] I. U. Din, I. Siddiqi, S. Khalid, and T. Azam, “Segmentation-free optical character recognition for printed urdu text,” EURASIP Journal on Image and Video Processing, vol. 2017, no. 1, p. 62, 2017.

[10] N. H. Khan and A. Adnan, “Urdu optical character recognition systems: Present contributions and future directions,” IEEE Access, vol. 6, pp. 46 019–46 046, 2018.

[11] M. Badry, M. Hassanin, A. Chandio, and N. Moustafa, “Quranic script optical text recognition using deep learning in iot systems,” CMC-COMPUTERS MATERIALS & CONTINUA, vol. 68, no. 2, pp. 1847–1858, 2021.

[12] S. Naz, A. I. Umar, R. Ahmad, I. Siddiqi, S. B. Ahmed, M. I. Razzak, and F. Shafait, “Urdu nastaliq recognition using convolutional–recursive deep learning,” Neurocomputing, vol. 243, pp. 80–87, 2017.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3144844, IEEE Access

[13] S. Y. Arafat and M. J. Iqbal, "Two stream deep neural network for sequence-based urdu ligature recognition," IEEE Access, vol. 7, pp. 159 090–159 099, 2019.

[14] M. Jain, M. Mathew, and C. Jawahar, "Unconstrained scene text and video text recognition for arabic script," in 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR). IEEE, 2017, pp. 26–30.

[15] A. Ali and M. Pickering, "A hybrid deep neural network for urdu text recognition in natural images," in 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC). IEEE, 2019, pp. 321–325.

[16] S. B. Ahmed, S. Naz, M. I. Razzak, and R. Yousaf, "Deep learning based isolated arabic scene character recognition," in 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR). IEEE, 2017, pp. 46–51.

[17] M. Tounsi, I. Moalla, A. M. Alimi, and F. Lebouregois, "Arabic characters recognition in natural scenes using sparse coding for feature representations," in 2015 13th International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2015, pp. 1036–1040.

[18] A. A. Chandio, M. Pickering, and K. Shafi, "Character classification and recognition for urdu texts in natural scene images," in 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET). IEEE, 2018, pp. 1–6.

[19] A. Ali, M. Pickering, and K. Shafi, "Urdu natural scene character recognition using convolutional neural networks," in 2018 IEEE 2nd international workshop on Arabic and derived script analysis and recognition (ASAR). IEEE, 2018, pp. 29–34.

[20] 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. 109 054–109 070, 2020.

[21] O. S. Nag, "What Languages Are Spoken In Pakistan? WolrdAtlas," https://www.worldatlas.com/articles/what-languages-are-spoken-in-pakistan.html/, 2019, [Online; accessed 10-April-2020].

[22] R. Smith, "Tesseract Open Source OCR Engine," https://github.com/tesseract-ocr/tesseract/, 2020, [Online; accessed 8-April-2020].

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[25] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in 2011 International Conference on Computer Vision. IEEE, 2011, pp. 1457–1464.

[26] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in European conference on computer vision. Springer, 2014, pp. 512–528.

[27] A. Mishra, K. Alahari, and C. Jawahar, "Scene text recognition using higher order language priors," 2012.

[28] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang, "Scene text recognition using part-based tree-structured character detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2961–2968.

[29] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010, pp. 2963–2970.

[30] C. Yao, X. Bai, B. Shi, and W. Liu, "Strokelets: A learned multi-scale representation for scene text recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 4042–4049.

[31] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," IEEE transactions on pattern analysis and machine intelligence, vol. 36, no. 12, pp. 2552–2566, 2014.

[32] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," International Journal of Computer Vision, vol. 116, no. 1, pp. 1–20, 2016.

[33] 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.

[34] Y. Zhang, S. Nie, W. Liu, X. Xu, D. Zhang, and H. T. Shen, "Sequence-to-sequence domain adaptation network for robust text image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2740–2749.

[35] Z. Lei, S. Zhao, H. Song, and J. Shen, "Scene text recognition using residual convolutional recurrent neural network," Machine Vision and Applications, vol. 29, no. 5, pp. 861–871, 2018.

[36] F. Sheng, Z. Chen, and B. Xu, "Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition," in 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019, pp. 781–786.

[37] S. Yousfi, S.-A. Berrani, and C. Garcia, "Contribution of recurrent connectionist language models in improving lstm-based arabic text recognition in videos," Pattern Recognition, vol. 64, pp. 245–254, 2017.

[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015. [Online]. Available: https://arxiv.org/abs/1502.03167

[39] T. He, W. Huang, Y. Qiao, and J. Yao, "Text-attentional convolutional neural network for scene text detection," IEEE transactions on image processing, vol. 25, no. 6, pp. 2529–2541, 2016.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in European conference on computer vision. Springer, 2016, pp. 630–645.

[41] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE transactions on neural networks, vol. 5, no. 2, pp. 157–166, 1994.

[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[43] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," Neural networks, vol. 18, no. 5-6, pp. 602–610, 2005.

[44] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in Proceedings of the 23rd international conference on Machine learning, 2006, pp. 369–376.

[45] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013, pp. 6645–6649.

[46] J. Salazar, K. Kirchhoff, and Z. Huang, "Self-attention networks for connectionist temporal classification in speech recognition," in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 7115–7119.

[47] L. Wu, T. Li, L. Wang, and Y. Yan, "Improving hybrid ctc/attention architecture with time-restricted self-attention ctc for end-to-end speech recognition," Applied Sciences, vol. 9, no. 21, p. 4639, 2019.

[48] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," IEEE transactions on pattern analysis and machine intelligence, vol. 31, no. 5, pp. 855–868, 2008.

[49] 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. 62 616–62 623, 2019.

[50] Z. Cheng, J. Lu, Y. Niu, S. Pu, F. Wu, and S. Zhou, "You only recognize once: Towards fast video text spotting," in Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 855–863.

[51] P. J. Werbos, "Backpropagation through time: what it does and how to do it," Proceedings of the IEEE, vol. 78, no. 10, pp. 1550–1560, 1990.

[52] L. Rabiner and B. Juang, "An introduction to hidden markov models," ieee assp magazine, vol. 3, no. 1, pp. 4–16, 1986.

[53] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," arXiv preprint arXiv:1412.6806, 2014. [Online]. Available: https://arxiv.org/abs/1412.6806

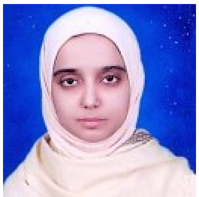
[54] S. Yousfi, S.-A. Berrani, and C. Garcia, "Deep learning and recurrent connectionist-based approaches for arabic text recognition in videos," in 2015 13th International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2015, pp. 1026–1030.

[55] S. Yousfi and S.-A. Berrani, "Alif: A dataset for arabic embedded text recognition in tv broadcast," in 2015 13th International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2015, pp. 1221–1225.

ASGHAR ALI CHANDIO was a Lecturer from 2010 to 2015 and is an Assistant Professor from 2016 till today at Department of Information Technology, Quaid-e-Awam University of Engineering, Science and Technology (QUEST), Pakistan. He received the BS degree in Information Technology from the Institute of Information Technology, University of Sindh, Pakistan in 2008, MS degree in Information Technology, from QUEST in 2014 and Ph.D. from the School of Engineering and Information Technology, University of New South Wales, at Canberra, Australia in 2020. His major research interests include machine learning, deep learning, handwritten text recognition, text extraction in natural scene images, document analysis and semantic text similarity matching.

MD. ASIKUZZAMAN received the B.Sc. degree in electronics and telecommunication engineering from the Rajshahi University of Engineering & Technology, Rajshahi, Bangladesh, in 2010, and the Ph.D. degree in electrical engineering from the University of New South Wales, Canberra, Australia, in 2015, under a very competitive University International Postgraduate Award Scholarship. He is currently a Research Associate with the School of Engineering and Information Technology, The University of New South Wales. He was the Technical Program Chair for the 2018 International Conference on Digital Image Computing: Techniques and Applications. He is currently serving as an Associate Editor for the multidisciplinary journal IEEE Access. His current research interests include 2D and 3D video watermarking, 3D modelling, deep learning, medical imaging, and video coding.

MARK R. PICKERING was born in Biloela, Australia, in 1966. He received the B.Eng. degree from the Capricornia Institute of Advanced Education, Rockhampton, Australia, in 1988, and the M.Eng. and Ph.D. degrees from The University of New South Wales, Canberra, Australia, in 1991 and 1995, respectively, all in electrical engineering. He was a Lecturer from 1996 to 1999, a Senior Lecturer from 2000 to 2009 and an Associate Professor from 2010 to 2017 with the School of Electrical Engineering and Information Technology, The University of New South Wales, where he is currently a Professor. His research interests include video and audio coding, medical imaging, data compression, information security, data networks and error-resilient data transmission.



MEHWISH LEGHARI was a Lecturer from July 2012 to June 2021 and is an Assistant Professor from July 2021 till today at Department of Information Technology, Quaid-e-Awam University of Engineering, Science and Technology (QUEST), Pakistan. Before joining QUEST, she was working as a Software Engineer at Isra University, Hyderabad, Pakistan. She received the BS degree in Information Technology from the Institute of Information Technology, University of Sindh, Pakistan in 2008, MS degree in Information Technology, from QUEST in 2015, and Ph.D. from the Dr. A.H.S Bukhari Institute of Information and Communication Technology, University of Sindh, Jamshoro in 2021. Her major research interests include biometrics recognition, multimodal biometrics systems, machine learning, handwritten text recognition and deep learning.

• • •