2017 14th IAPR International Conference on Document Analysis and Recognition

Benchmarking Scene Text Recognition in Devanagari, Telugu and Malayalam

Minesh Mathew, Mohit Jain and C. V. Jawahar
Center for Visual Information Technology, IIIT Hyderabad, India.
minesh.mathew@research.iiit.ac.in, mohit.jain@research.iiit.ac.in, jawahar@iiit.ac.in

***Abstract*—Inspired by the success of Deep Learning based approaches to English scene text recognition, we pose and benchmark scene text recognition for three Indic scripts - Devanagari, Telugu and Malayalam. Synthetic word images rendered from Unicode fonts are used for training the recognition system. And the performance is bench-marked on a new - *IIIT-ILST* dataset comprising of hundreds of real scene images containing text in the above mentioned scripts. We use a segmentation free, hybrid but end-to-end trainable CNN-RNN deep neural network for transcribing the word images to the corresponding texts. The cropped word images need not be segmented into the sub-word units and the error is calculated and backpropagated for the the given word image at once. The network is trained using CTC loss, which is proven quite effective for sequence-to-sequence transcription tasks. The CNN layers in the network learn to extract robust feature representations from word images. The sequence of features learnt by the convolutional block is transcribed to a sequence of labels by the RNN+CTC block. The transcription is not bound by word length or a lexicon and is ideal for Indian languages which are highly inflectional.**

***Keywords*—Indian Languages, Scene Text, Indic Scripts, Synthetic data, CNN-RNN, Text Recognition, OCR**

I. Introduction

A. Scene Text Recognition

The problem of scene text recognition deals with recognizing text in natural scene images. Traditionally text recognition was focused on recognizing printed text in documents. Such systems expected the images to be black and white, and in a document style layout comprising of text lines. The text in natural scenes in contrast appear in huge varieties in terms of layout, fonts and style. The traditional OCR systems do not generalize well to such a setting where inconsistent lighting, occlusion, background noise, and higher order distortions add to the problem complexity. Most works in this area, treat the scene text recognition problem as two sub-problems - detecting bounding boxes of words in an image and then recognizing the individual, cropped word images. Our work deals with the second problem, where a cropped word image need to be recognized.

Recognizing text appearing in natural scenes has become increasingly relevant today with the proliferation of mobile imaging devices. Text appearing in natural scenes provide a great deal of information helpful in understanding what the whole image is about. To be able to recognize text in natural scenes will be quite useful in scenarios like autonomous navigation, assistive technologies for the visually impaired, traffic surveillance systems, mobile transliteration/translation technologies, mobile document scanners etc. Hence building a robust scene text recognition system will have a significant impact on many other problems involving computer vision.

Fig. 1: Natural scene images, having text in Indic scripts; Telugu, Malayalam and Devanagari in the clock wise order

Deep Learning based methods significantly improved the scene text recognition accuracies for English. A Convolutional Neural Network (CNN) was used in [23], [24] to recognize individual characters in a word image and predicted characters are combined for a word to output the transcription for the given word image. These methods required a good character segmentation algorithm to segment the word image into sub-word units. Such methods would not be suitable for Indic scripts where sub word segmentation is often difficult. The problem was modelled as image classification in [12] where each word image was classified into word classes, drawn from a fixed size lexicon. This method, bounded by a lexicon is inherently not suitable for highly inflectional Indian languages. Another set of solutions learn common representations for word-label pairs and retrieval and recognition is performed on the learnt representations. In [25] word image and the text are embedded into a vector-subspace. This setting, enabled to model the scene text recognition problem as a retrieval problem - to retrieve the most satiable text from the vector-subspace once a word image is given.

The segmentation-free, transcription approach has been proven quite effective for Optical Character Recognition (OCR) in Indic Scripts [18], [19] and Arabic [11] where segmentation is often problematic. Similar approach was followed for English scene text recognition in [9]. Handcrafted features derived from image gradients were used with a Bidirectional Long Short Term Memory (LSTM) to map the sequence of features to a sequence of labels. Then Connectionist Temporal

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.364

Classification (CTC) [16] loss was used to computer the loss for the entire sequence of image features at once. Unlike the problem of OCR, scene text recognition required more robust features to yield results comparable to the transcription based solutions for OCR. A novel approach combining the robust convolutional features and transcription abilities of RNN was introduced in [26]. Here a 7 layer CNN stack is used at the head of an RNN + CTC transcription network. The network named as CRNN is end-to-end trainable and yielded better results than transcription using handcrafted features.

B. Recognizing Indic Scripts

Research in text recognition for Indic scripts has mostly been centred around the problem of printed text recognition; popularly known as OCR. Lack of annotated data, and inherent complexities of the script and language were the major challenges faced by the community. Most of the early machine learning approaches which were effective for English OCR, were not easily adaptable to Indian languages setting for this reason. Even today, when modern machine learning methods could be used in a language/script agnostic manner, lack of annotated data remains as major challenge in case of Indian languages. There had been few works, which addressed the data scarcity by using synthetic data. A synthetic dataset comprising of 28K word images was used in [1] for training a nearest neighbour based Telugu OCR. The images were rendered from a vocabulary of 1000 words, by varying font, font size, kerning and other rendering parameters.

Early attempts to recognize text in Indian scripts, often required a segmentation module which could segment word into sub-word units like characters or *aksharas*. Lately the works in OCR started following segmentation-free approaches. There have been works using Hidden Markov Models (HMMs) [2] and Recurrent Neural Networks (RNNs) [19]. Among these RNNs became quite popular choice for transcribing text words or lines directly into a sequence of class labels. LSTM Networks used along with (CTC) loss enabled end-to-end training of a network which can transcribe from a sequence of image features to a sequence of characters. This approach did not require segmenting words or line images into sub-units, and could handle variable length images and output label sequences. It has been proven quite effective for complex scripts like Indic Scripts and Arabic were segmentation of words into sub-word units is often difficult [18], [19], [11]. Use of a Bidirectional LSTM (BLSTM) enabled modelling past and future contextual dependencies. This significantly helped in accurately predicting certain characters, particularly vowel modifiers (*matras*) in Indic scripts. The challenges posed by Indic scripts and how the transcription approach helped to overcome those are discussed in detail in [19]. OCR using RNN+CTC has been using either raw pixels of the input image [11], [19] or handcrafted features like profile based features [18] as the input representation. We shall refer to such methods using RNN+CTC on handcrafted features as RNN-OCR hereafter.

C. Scene Text Recognition for Indic Scripts

Inspired form the success of RNN-OCR, we attempt to extend it to the problem of scene text recognition in Indian languages. Unlike English , there has been no works in scene text recognition in Indic scripts to the best of our knowledge. Three major contributions of this paper can be listed down as

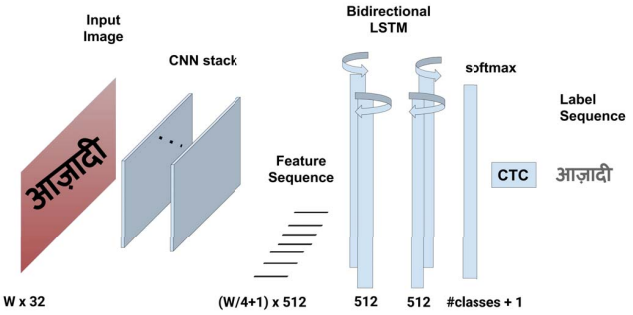

Fig. 2: Visualization of the Hybrid CNN-RNN network architecture used in this work.

- Synthetic scene text data - Word images are rendered using Unicode fonts for indic scripts, from a large vocabulary. The foreground and background texture, colors and other rendering parameters are varied to make the rendered images look similar to the real scene images. Fig. 3 shows sample synthetic images rendered
- Real scene text data - A new, real scene text dataset - *IIIT-ILST* was curated to benchamrk the recognition performance. This dataset comprises of hundreds of word images, for each script
- Benchmarking Scene text recognition - A segmentation free, end-to-end trainable system is employed. The hybrid CNN-RNN network used here, is trained purely on syntheticcally rendered word images. The trained network is then tested on the real scene text data.

The rest of the paper is organized as follows; Section II describes the hybrid CNN-RNN architecture we use. Section III presents details of the rendering process used to generate the synthetic dataset, a brief summary of the *IIIT-ILST* dataset and the network architecture used. Quantitative and qualitative results, and a discussion on the results are presented in section IV Section V concludes with the findings of our work.

II. CNN-RNN Hybrid Architecture for Transcription

The hybrid CNN-RNN network consists of three components. The initial convolutional layers, middle recurrent layers and a final transcription layer. This network can be trained in an end-to-end fashion using the CTC loss and is not constrained by any language-specific lexicon. Consequently, any possible combination of the script's character-set can be recognized by the model.

The convolutional layers follow a VGG [10] style architecture where the fully-connected layers have been removed. These layers obtain robust feature representations from the input images. The sequence of features are then passed on to the recurrent layers which transcribe them into an output

sequence of labels representing the script's characters/glyphs. Transcription on to the output sequence of labels is performed using a CTC loss layer at the output.

All the images are scaled to a fixed height before being fed to the convolutional layers. The convolutional components then create a sequence of feature vectors from the feature maps by splitting them column-wise, which then act as inputs to the recurrent layers (Fig. 2). The convolutional features learnt by the network, are much more robust than the handcrafted features. The layers of convolution, max-pool and element-wise activation functions are translation invariant as they operate on local regions. Hence, each column feature from the feature map corresponds to a rectangle region of the original image. These rectangular regions are themselves in the same order to their corresponding columns on the feature maps from left to right and hence can be considered as an image descriptor for that region. The recurrent layers following the convolutional layers, take each frame from these column-feature sequences and make predictions.

The recurrent layers consist of deep BLSTM nets. Since the number of parameters of an RNN is independent of the length of its input sequence, RNNs are capable of handling variable length sequences. The network can be unrolled as many times as the number of time-steps in the input sequence and hence, any sequence of characters/glyphs derived from the root script's character set can be predicted. This in our case helps to perform unconstrained recognition as the predicted output can be any sequence of labels derived from the entire label set. Traditional RNN units (*vanilla* RNN*s*) faced the problem of vanishing gradients [13] and hence LSTM units are used which by design tackle the vanishing-gradients problem [14] and remembers contextual dependencies for longer time.

For a typical text recognition problem, the transcription accuracy is benefitted if context from both directions (left-to-right and right-to-left) are available. A BLSTM is hence used to combine contexts from forward and backward orientations. Multiple such BLSTM layers can be stacked to make the network deeper and gain higher levels of abstractions over the image-sequences as shown in [15].

The transcription layer at the top of the network is used to translate the predictions generated by the recurrent layers into label sequences for the target language. The CTC layer's conditional probability is used in the objective function as shown in [16]. The complete network configuration used for the experiments can be seen in Fig. 2. The objective is to minimize the negative log-likelihood of conditional probability of ground truth. This objective function calculates a cost value directly from an image and its ground truth label sequence, eliminating the need of manually label all the individual components in sequence.

III. Transcribing Scene Text in Indic Scripts

Since scene text recogniton in Indic scripts has not been attempted before, we introduce two new datasets. A synthetic dataset, comprising of around 4 million images for each script and the *IIIT-ILST* dataset comprising of real scene images for benchmarking the performance of the scene text recognition. The details are presented in the following two subsections.

Fig. 3: Sample images for Hindi, Malayalam and Telugu from the synthetic dataset.

A. Synthetic Scene Text Dataset

Synthetic data is now widely used within the computer vision community for problems where it is often difficult to acquire large amounts of training data. This approach to deal with the data scarcity issue has become popular with the advent of Deep Learning based methods, which are ever more data hungry [4], [5], [6]. The trend was prevalent in the area of text recognition, even before Deep learning based methods became popular. For low resource languages like Indian languages and Arabic, synthetically rendered text images have been in use for quite sometime [1], [28], [27]. It has been widely accepted when [4] trained an English scene text recognition system trained purely on a synthetic dataset comprising of 8 million word images called *MJSynth*. Since then all the works in English scene text recognition have been using this synthetic dataset alone as the training data. The use of such a huge corpus of word images along with state-of-the art deep learning methods yielded superior results in English scene text recognition. Recently, similar approach was followed in making synthetic datasets for scene text and video text recognition for Arabic [29]

In this work, the CNN-RNN hybrid architecture is purely trained on synthetically generated word images rendered from a large vocabulary - around 100K words for each script crawled from Wikipedia. The words are rendered into images using freely available Unicode fonts. Each word from the vocabulary is first rendered into the foreground layer of the image by varying the font, font size, stroke color, stroke thickness, kerning, skew and rotation along the horizontal line. Later a random perspective projective transformation is applied to the foreground image, which is then optionally blended with a random crop from a natural scene image. Finally the foreground layer is alpha composed with a background layer, which is either an image with uniform background color, or a random crop from a natural image. Details of the rendering process are presented here [17].

B. IIIT-ILST Datasaet

A new dataset for benchmarking the performance of scene text recogniton for Indic scripts was curated by capturing images with text in Indic scripts and compiling freely avaibale images from Google Images. The dataset consists of hundreds

Fig. 4: Qualitative results of scene text recognition for Indic scripts. For each script, the top-row shows images for which correct predictions were made while the bottom-row shows images with incorrect predictions. The number on top-right corner of each incorrectly predicted image is the *Levenshtein distance* between its ground-truth and the prediction made by our model.

of scene images occuring in various scenrios like local markets, billboards, navigation and traffic signs, banners, graffiti. etc, and spans a large variety of naturally occuring image-noises and distortions. There are around 1000 scene text word images for each script extracted from these scene images. Each scene image is annotated by marking the bounding boxes of each word in the image, and typing the corresponding text in Unicode. Hence the dataset can also be used for scene text detection task. For the problem of scene text recognition addressed in this paper, we only deal with recognizing the individual word images. Hence in all the results discussed below, it is assumed that cropped word images are provided.

C. Implementation Details

The hybrid CNN-RNN network has it's convolutional stack inspired from the VGG-style architecture with minor modifications made to the layers to better fit a script recognition setting. In the 3rd and 4th max-pooling layers, the pooling windows used are rectangular instead of the usual square windows used in VGG. This helps us obtain wider features after the convolutions and hence more time-steps for the recurrent layers to unfold on. All input images are converted to grey scale and re-scaled to a fixed height of 32 pixels, while keeping the aspect ratio the same. The convolutional stack is followed by two BLSTM layers each of size 512. The second BLSTM layer is connected to a fully connected layer of size equivalent to the number of labels + 1 (extra label for *blank*) . Lables in our case are Unicode points of the script for which the transcription is learnt. The label set include all the unique Unicode points found in the vocabulary used to render the synthetic dataset and basic punctuation symbols. Finally Softmax activation is applied to the outputs at the last year and the CTC loss is computed between the output probabilities and the expected, target label sequence.

To accelerate the training process, two batch-normalization [20] is performed after the 3rd and 4th convolutional layers. We have observed that applying batch normalization after each convolutional layer yielded slightly poorer accuracies. To automate the process of setting optimization parameters, ADADELTA optimization [22] is used while training the network using stochastic gradient descent (SGD). The transcription layers' error differentials are backpropagated with the forward-backward algorithm. While in the recurrent layers, the Backpropagation Through Time (BPTT) [21] algorithm is applied to calculate the error differentials.

TABLE I: Performance Evaluation on *IIIT-ILST* Dataset

IL SceneText Script	No. of Images	Method			
		RNN-OCR		HYBRID CNN-RNN	
		WRR (%)	CRR (%)	WRR (%)	CRR (%)
Hindi	1150	29.7	58.1	42.9	75.6
Telugu	1211	33.6	61	57.2	86.2
Malayalam	807	40.2	73.2	73.4	92.8

Comparing performance of the hybrid CNN-RNN model against the RNN-OCR style approach in [19]

IV. Results and Discussion

We compare the transcription capabilities of our hybrid CNN-RNN network against an RNN-OCR style model. RNN-OCR could yield superior results for printed text recognition in Indic scripts [19] even without a convolutional feature extraction block, since the printed text recognition is relatively an easier problem compared to the scene text recognition. We compare the performance of the hybrid architecture against RNN-OCR style approach, where raw image pixels are directly fed to the RNN. The RNN-OCR style approach used here is same as the one used for OCR in [19]. Table I presents a comparison of both the approaches.

The performance has been evaluated using the following metrics; CRR - *Character Recognition Rate* and WRR - *Word Recognition Rate*. In the below equations, *RT* and *GT* stand for recognized text and ground truth respectively.

$$CRR = \frac{(nCharacters - \sum LevenshteinDistance(RT, GT))}{nCharacters}$$

$$WRR = \frac{nWordsCorrectlyRecognized}{nWords}$$

The tabular results further reinforce our hypothesis that addition of convolutional layers before the transcription block,

improves the performance, since the convolutional layers output much robust representations than the raw pixel values of the input images or other handcrafted features. A qualitative analysis of the trained models' transcription capabilities can be seen in Fig. 4.

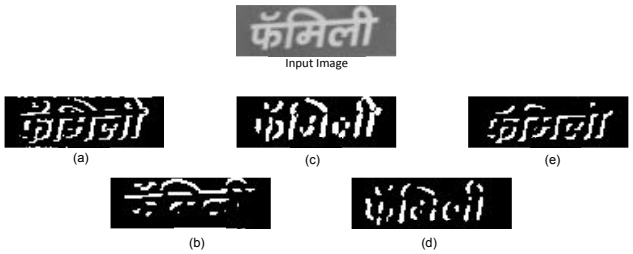

Fig. 5: Visualization of activations learnt by the hybrid CNN-RNN model's convolutional layers. We observe that convolutional layers learn to detect edges and diacritics in the text and derive directional edge information from the source image. Notice how (a) detects all text edges while (b) focuses on *matras* and *top-connector-line* of Devanagiri script. Right-edges, left-edges and right-bottom edges are detected in (c), (d) and (e) respectively.

To gain further insights into the workings of the convolutional layers, we visualize the activations of these layers generated by performing a forward pass on the model using sample images (Fig. 5). The model seems to be learning orientation-specific edge detectors in the initial convolutional layers. Additionally, the convolutional layers also grasp the ability to differentiate between main text body and the diacritics appearing in contiguation.

V. Conclusion

We demonstrate that state-of-the-art deep learning based methods can be successfully adapted to some rather challenging tasks like scene text recognition in Indic scripts. The newer script and language agnostic approaches are well suited for low resource languages like Indian languages where the traditional methods often involved language specific modules. The success of RNNs in sequence learning problems has been instrumental in the recent advances in speech recognition and image to text transcription problems. This came as a boon for Indic scripts where segmentation of words into sub word units are often troublesome. The sequence learning approach could directly transcribe the images and also model the context in both forward and backward directions. With better feature representations and learning algorithms available, we believe the focus should now shift to harder problems like scene text recognition. We hope the introduction of a new dataset and the initial results would instill an interest among the researchers to pursue this field of research further. In future we hope to increase the size of the real scene dataset and to add images in other indic scripts too. With a reasonably big real scene text corpus, some percentage of the images shall be used to fine-tune the network which now is purely trained on synthetic images. We believe that the fine-tuning would considerably improve the performance.

References

[1] Pramod Sankar, C.V. Jawahar and Raghavan Manmatha, *Nearest neighbor based collection OCR*. DAS 2010.

[2] Prem Natarajan, Ehry MacRostie and Michael Decerbo, *The BBN Byblos Hindi OCR System*. DRR 2005.

[3] B. Taskar, C. Guestrin, and D. Koller, *Max-margin markovnetworks*. NIPS 2004.

[4] M. Jaderberg, K. Simonyan, A. Vedaldi and A. Zisserman, *Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition*. arXiv 2014.

[5] A. Gupta, A. Vedaldi and A. Zisserman, *Synthetic Data for Text Localisation in Natural Images*. CVPR 2016.

[6] A. Gaidon, Q. Wang, Y. Cabon and E. Vig, *Virtual Worlds as Proxy for Multi-Object Tracking Analysis*. CVPR 2016.

[7] Y. LeCun, L. Bottou, Y. Bengio and P.Haffiner, *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 1998.

[8] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke and J. Schidhuber, *A novel connectionist system for unconstrained handwriting recognition*. TPAMI, 2009.

[9] B. Su and S. Lu, *Accuracte scene text recognition based on recurrent neural network*. ACCV 2014.

[10] K. Simonyan and A. Zisserman. *Very deep convolutional networks for large-scale image recognition*. CoRR 2014.

[11] Ul-Hasan, Adnan, et al, *Offline printed Urdu Nastaleeq script recognition with bidirectional LSTM networks*, ICDAR 2013.

[12] M. Jaderberg, K. Simoyan, A. Vedaldi and A. Zisserman, *Reading text in the wild with convolutional neural networks*. IJCV 2015.

[13] Y. Bengio, P.Y. Simard and P. Frasconi, *Learning long-term dependencies with gradient descent is difficult*. NN 1994.

[14] S. Hochreiter and J. Schmidhuber, *Long short-term memory*. Neural Computation 1997.

[15] A. Graves, A. Mohamed and G.E. Hinton, *Speech recognition with deep recurrent neural networks*. ICASSP 2013.

[16] A. Graves, S. Fernandez, F.J. Gomez and J. Schmidhuber, *Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks*. ICML 2006.

[17] Praveen Krishnan and C.V. Jawahar, *Generating Synthetic Data for Text Recognition*. arXiv 2016.

[18] Naveen Sankaran and C.V. Jawahar, *Recognition of printed Devanagari text using BLSTM Neural Network*. ICPR 2012.

[19] Minesh Mathew, Ajeet Kumar Singh and C.V. Jawahar, *Multilingual OCR for Indic Scripts*. DAS 2016.

[20] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. ICML 2015.

[21] P.J. Werbos, *Backpropagation through time: what it does and how to do it*. Proceedings of the IEEE 1990.

[22] M.D. Zeiler, *ADADELTA: an adaptive learning rate method*. CoRR 2012.

[23] T. Wang, D.J. Wu, A. Coates and A.Y. Ng, *End-to-end text recognition with convolutional neural networks*. CVPR 2014.

[24] A. Bissacco, M. Cummins, Y. Netzer and H. Neven, *Photoocr: Reading text in uncontrolled conditions*. ICCV 2013.

[25] J. Almazan, A. Gordo, A. Fornes and E. Valveny, *Word spotting and recognition with embedded attributes*. TPAMI 2014.

[26] 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*. arXiv 2015.

[27] N. Sabbour and F. Shafait, *A segmentation-free approach to Arabic and Urdu OCR*. DRR 2013.

[28] F. Slimane, R. Ingold, S. Kanoun, A.M. Alimi and J. Hennebert, *A New Arabic Printed Text Image Database and Evaluation Protocols*. DAS 2009.

[29] Mohit Jain, Minesh Mathew and C.V. Jawahar, *Unconstrained Scene Text and Video Text Recognition for Arabic Script*. ASAR 2017.