Transfer Learning for Scene Text Recognition in Indian Languages

Sanjana Gunna(✉) , Rohit Saluja , and C. V. Jawahar

Centre for Vision Information Technology, International Institute of Information Technology, Hyderabad 500032, India
{sanjana.gunna,rohit.saluja}@research.iiit.ac.in, jawahar@iiit.ac.in,
https://github.com/firesans/STRforIndicLanguages

Abstract. Scene text recognition in low-resource Indian languages is challenging because of complexities like multiple scripts, fonts, text size, and orientations. In this work, we investigate the power of transfer learning for all the layers of deep scene text recognition networks from English to two common Indian languages. We perform experiments on the conventional CRNN model and STAR-Net to ensure generalisability. To study the effect of change in different scripts, we initially run our experiments on synthetic word images rendered using Unicode fonts. We show that the transfer of English models to simple synthetic datasets of Indian languages is not practical. Instead, we propose to apply transfer learning techniques among Indian languages due to similarity in their n-gram distributions and visual features like the vowels and conjunct characters. We then study the transfer learning among six Indian languages with varying complexities in fonts and word length statistics. We also demonstrate that the learned features of the models transferred from other Indian languages are visually closer (and sometimes even better) to the individual model features than those transferred from English. We finally set new benchmarks for scene-text recognition on Hindi, Telugu, and Malayalam datasets from IIIT-ILST and Bangla dataset from MLT-17 by achieving 6%, 5%, 2%, and 23% gains in Word Recognition Rates (WRRs) compared to previous works. We further improve the MLT-17 Bangla results by plugging in a novel correction BiLSTM into our model. We additionally release a dataset of around 440 scene images containing 500 Gujarati and 2535 Tamil words. WRRs improve over the baselines by 8%, 4%, 5%, and 3% on the MLT-19 Hindi and Bangla datasets and the Gujarati and Tamil datasets.

Keywords: Scene text recognition · Transfer learning · Photo OCR · Multi-lingual OCR · Indian languages · Indic OCR · Synthetic data

1 Introduction

Scene-text recognition or Photo-Optical Character Recognition (Photo-OCR) aims to read scene-text in natural images. It is an essential step for a wide

Fig. 1. Clockwise from top-left; "**Top:** Annotated Scene-text images, **Bottom:** Baselines' predictions (row-1) and Transfer Learning models' predictions (row-2)", from Gujarati, Hindi, Bangla, Tamil, Telugu and Malayalam. Green, red, and "_" represent correct predictions, errors, and missing characters, respectively. (Color figure online)

variety of computer vision tasks and has enjoyed significant success in several commercial applications [9]. Photo-OCR has diverse applications like helping the visually impaired, data mining of street-view-like images for information used in map services, and geographic information systems [2]. Scene-text recognition conventionally involves two steps; i) Text detection and ii) Text recognition. Text detection typically consists of detecting bounding boxes of word images [4]. The text recognition stage involves reading cropped text images obtained from the text detection stage or from the bounding box annotations [13]. In this work, we focus on the task of text recognition.

The multi-lingual text in scenes is a crucial part of human communication and globalization. Despite the popularity of recognition algorithms, non-Latin language advancements have been slow. Reading scene-text in such low resource languages is a challenging research problem as it is generally unstructured and appears in diverse conditions such as scripts, fonts, sizes, and orientations. Hence a large amount of dataset is usually required to train the scene-text recognition models. Conventionally, the synthetic dataset is used to deal with the problem since a large number of fonts are available in such low resource languages [13]. The synthetic data may also serve as an exciting asset to perform controlled experiments, e.g., to study the effect of transfer learning with the change in script or language text. We investigate such effects for transfer from English to two Indian languages in this work, i.e., Hindi and Gujarati. We also explore the transferability of features among six different Indian languages. We share 2500 scene text word images obtained from over 440 scenes in Gujarati and Tamil to demonstrate such effects. In Fig. 1, we illustrate the sample annotated images from our datasets, and IIIT-ILST and MLT datasets, and the predictions of our

models. The overall methodology we follow is that we first generate the synthetic
datasets in the six Indian languages. We describe the dataset generation process
and motivate the work in Sect. 2. We then train the two deep neural networks we
introduce in Sect. 3 on the individual language datasets. Subsequently, we apply
transfer-learning on all the layers of different networks from one language to
another. Finally, as discussed in Sect. 4, we fine-tune the networks on standard
datasets and examine their performance on real scene-text images in Sect. 5. We
finally conclude the work in Sect. 6. The summary of our contributions are as
follows:

1. We investigate the transfer learning of complete scene-text recognition mod-
   els i) from English to two Indian languages and ii) among the six Indian
   languages, i.e., Gujarati, Hindi, Bangla, Telugu, Tamil, and Malayalam.
2. We also contribute two datasets of around 500 word images in Gujarati and
   2535 word images in Tamil from a total of 440 Indian scenes.
3. We achieve gains of 6%, 5%, and 2% in Word Recognition Rates (WRRs) on
   IIIT-ILST Hindi, Telugu, and Malayalam datasets in comparison to previous
   works [13,20]. On the MLT-19 Hindi and Bangla datasets and our Gujarati
   and Tamil datasets, we observe the WRR gains of 8%, 4%, 5%, and 3%,
   respectively, over our baseline models.
4. For the MLT-17 Bangla dataset, we show a striking improvement of 15% in
   Character Recognition Rate (CRR) and 24% in WRR compared to Bušta
   et al. [2], by applying transfer-learning from another Indian language and
   plugging in a novel correction RNN layer into our model.

1.1   Related Work

We now discuss datasets and associated works in the field of photo-OCR.

Works of Photo-OCR on Latin Datasets: As stated earlier, the process of
Photo-OCR conventionally includes two steps: i) Text detection and ii) Text
recognition. With the success of Convolutional Neural Networks (CNN) for
object detection, the works have been extended to text detection, treating words
or lines as the objects [12,27,37]. Liao et al. [10] extend such works to real-time
detection in scene images. Karatzas et al. [8] and Bušta et al. [1] present more
efficient and accurate methods for text detection. Towards reading scene-text,
Wang et al. [30] propose an object recognition pipeline based on a ground truth
lexicon. It achieves competitive performance without the need for an explicit
text detection step. Shi et al. [21] propose a Convolutional Recurrent Neural Net-
work (CRNN) architecture, which integrates feature extraction, sequence mod-
eling, and transcription into a unified framework. The model achieves remark-
able performances in both lexicon-free and lexicon-based scene-text recognition
tasks. Liu et al. [11] introduce Spatial Attention Residue Network (STAR-Net)
with spatial transformer-based attention mechanism to remove image distor-
tions, residue convolutional blocks for feature extraction, and an RNN block
for decoding the text. Shi et al. [22] propose a segmentation-free Attention-
based method for Text Recognition (ASTER) by adopting Thin-Plate-Spline

(TPS) as a rectification unit. It tackles complex distortions and reduces the difficulty of irregular text recognition. The model incorporates ResNet to improve the network's feature representation module and employs an attention-based mechanism combined with a Recurrent Neural Network (RNN) to form the prediction module. Uber-Text is a large-scale Latin dataset that contains around $117K$ images captured from 6 US cities [36]. The images are available with line-level annotations. The French Street Name Signs (FSNS) data contains around $1000K$ annotated images, each with four street sign views. Such datasets, however, contain text-centric images. Reddy et al. [16] recently release RoadText-1K to introduce challenges with generic driving scenarios where the images are not text-centric. RoadText-1K includes 1000 video clips (each 10 s long at 30 fps) from the BDD dataset, annotated with English transcriptions [32].

Works of Photo-OCR on Non-Latin Datasets: Recently, there has been an increasing interest in scene-text recognition for non-Latin languages such as Chinese, Korean, Devanagari, Japanese, etc. Several datasets like RCTW ($12k$ scene images), ReCTS-25k ($25k$ signboard images), CTW ($32k$ scene images), and RRC-LSVT ($450k$ scene images) from ICDAR'19 Robust Reading Competition (RRC) exist for Chinese [23,25,33,35]. Arabic datasets like ARASTEC (260 images of signboards, hoardings, and advertisements) and ALIF ($7k$ text images from TV Broadcast) also exist in the scene-text recognition community [28,31]. Korean and Japanese scene-text recognition datasets include KAIST ($2,385$ images from signboards, book covers, and English and Korean characters) and DOST ($32k$ sequential images) [5,7]. The MLT dataset available from the ICDAR'17 RRC contains $18k$ scene images (around $1 - 2k$ images per language) in Arabic, Bangla, Chinese, English, French, German, Italian, Japanese, and Korean [15]. The ICDAR'19 RRC builds MLT-19 over top of MLT-17 to contain $20k$ scene images containing text from Arabic, Bangla, Chinese, English, French, German, Italian, Japanese, Korean, and Devanagari [14]. The RRC also provides $277k$ synthetic images in these languages to assist the training. Mathew et al. [13] train the conventional encoder-decoder, where Convolutional Neural Network (CNN) encodes the word image features. An RNN decodes them to produce text on synthetic data for Indian languages. Here an additional connectionist temporal classification (CTC) layer aligns the RNN's output to labels. The work also releases an IIIT-ILST dataset for testing that reports Word Recognition Rates (WRRs) of 42.9%, 57.2%, and 73.4% on $1K$ real images in Hindi, Telugu, and Malayalam, respectively. Bušta et al. [2] proposes a CNN (and CTC) based method for text localization, script identification, and text recognition. The model is trained and tested on 11 languages of MLT-17 dataset. The WRRs are above 65% for Latin and Hangul and are below 47% for the remaining languages. The WRR reported for Bengali is 34.20%. Recently, an OCR-on-the-go model and obtain the WRR of 51.01% on the IIIT-ILST Hindi dataset and the Character Recognition Rate (CRR) of 35% on a multi-lingual dataset containing 1000 videos in English, Hindi, and Marathi [20]. Around 2322 videos in these languages recorded with controlled camera movements like tilt, pan, etc., are additionally shared at https://catalist-2021.github.io/.

Table 1. Statistics of synthetic data. μ, σ represent mean, standard deviation.

Language	# Images	Train	Test	μ, σ word length	# Fonts
English	17.5M	17M	0.5M	5.12, 2.99	>1200
Gujarati	2.5M	2M	0.5M	5.95, 1.85	12
Hindi	2.5M	2M	0.5M	8.73, 3.10	97
Bangla	2.5M	2M	0.5M	8.48, 2.98	68
Tamil	2.5M	2M	0.5M	10.92, 3.75	158
Telugu	5M	5M	0.5M	9.75, 3.43	62
Malayalam	7.5M	7M	0.5M	12.29, 4.98	20

Transfer Learning in Photo-OCR: With the advent of deep learning in the last decade, transfer learning became an essential part of vision models for tasks such as detection and segmentation. [17,18]. The CNN layers pre-trained from the Imagenet classification dataset are conventionally used in such models for better initialization and performance [19]. The scene-text recognition works also use the CNN layers from the models pre-trained on Imagenet dataset [11,21,22]. However, to our best knowledge, there are no significant efforts on transfer learning from one language to another in the field of scene-text recognition, although transfer learning seems to be naturally suitable for reading low resource languages. We investigate the possibilities of transfer learning in all the layers of deep photo-OCR models.

2   Datasets and Motivation

We now discuss the datasets we use and the motivation for our work.

Synthetic Datasets: As shown in Table 1, we generate $2.5M$, or more, word images each in Hindi, Bangla, Tamil, Telugu, and Malayalam[1] with the methodology proposed by Mathew et al. [13]. For each Indian language, we use $2M$ images for training our models and the remaining set for testing. Sample images of our synthetic data are shown in Fig. 2. For English, we use the models pretrained on the $9M$ MJSynth and $8M$ SynthText images [3,6]. We generate $0.5M$ synthetic images in English with over 1200 fonts for testing. As shown in Table 1, English has a lower average word length than Indian languages. We list the Indian languages in the increasing order of language complexity, with visually similar scripts placed consecutively, in Table 1. Gujarati is chosen as the entry point from English to Indian languages as it has the lowest word length among all Indian languages. Subsequently, like English, Gujarati does not have a top-connector line that connects different characters to form a word in Hindi and Bangla (refer to Fig. 1 and 2). Also, the number of Unicode fonts available in Gujarati is

---

[1] For Telugu and Malayalam, our models trained on $2.5M$ word images achieved results lower than previous works, so we generate more examples equal to Mathew et al. [13].

Fig. 2. Clockwise from top-left: synthetic word images in Gujarati, Hindi, Bangla, Tamil, Telugu, & Malayalam. Notice that a top-connector line connects the characters to form a word in Hindi or Bangla. Some vowels and characters appear above and below the generic characters in Indian languages, unlike English.

fewer than those available in other Indian languages. Next, we choose Hindi, as Hindi characters are similar to Gujarati characters and the average word length of Hindi is higher than Gujarati. Bangla has comparable word length statistics with Hindi and shares the property of the top-connector line with Hindi. Still, we keep it after Hindi in the list as its characters are visually dissimilar and more complicated than Gujarati and Hindi. We use less than 100 for fonts in Hindi, Bangla, and Telugu. We list Tamil after Bangla because these languages share similar vowels' appearance (see the glyphs above general characters in Fig. 2). Tamil and Malayalam have the highest variability in word length and visual complexity compared to other languages. Please note that we have over 150 fonts available in Tamil.

Real Datasets: We also perform experiments on the real datasets from IIIT-ILST, MLT-17, and MLT-19 datasets (refer to Sect. 1.1 for these datasets). To enlarge scene-text recognition research in complex and straight forward low-resource Indian Languages, we release 500 and 2535 annotated word images in Gujarati and Tamil. We crop the word images from 440 annotated scene images, which we obtain by capturing and compiling Google images. We illustrate sample annotated images of different datasets in Fig. 1. Similar to MLT datasets, we annotate the Gujarati and Tamil datasets using four corner points around each word (see Tamil image at bottom-right of Fig. 1). IIIT-ILST dataset has two-point annotations leading to an issue of text from other words in the background of a cropped word image as shown in the Hindi scene at the top-middle of Fig. 1.

Motivation: As discussed earlier in Sect. 1.1, most of the scene-text recognition works use the pre-trained Convolutional Neural Networks (CNN) layers for improving results. We now motivate the need for transfer learning of the com-
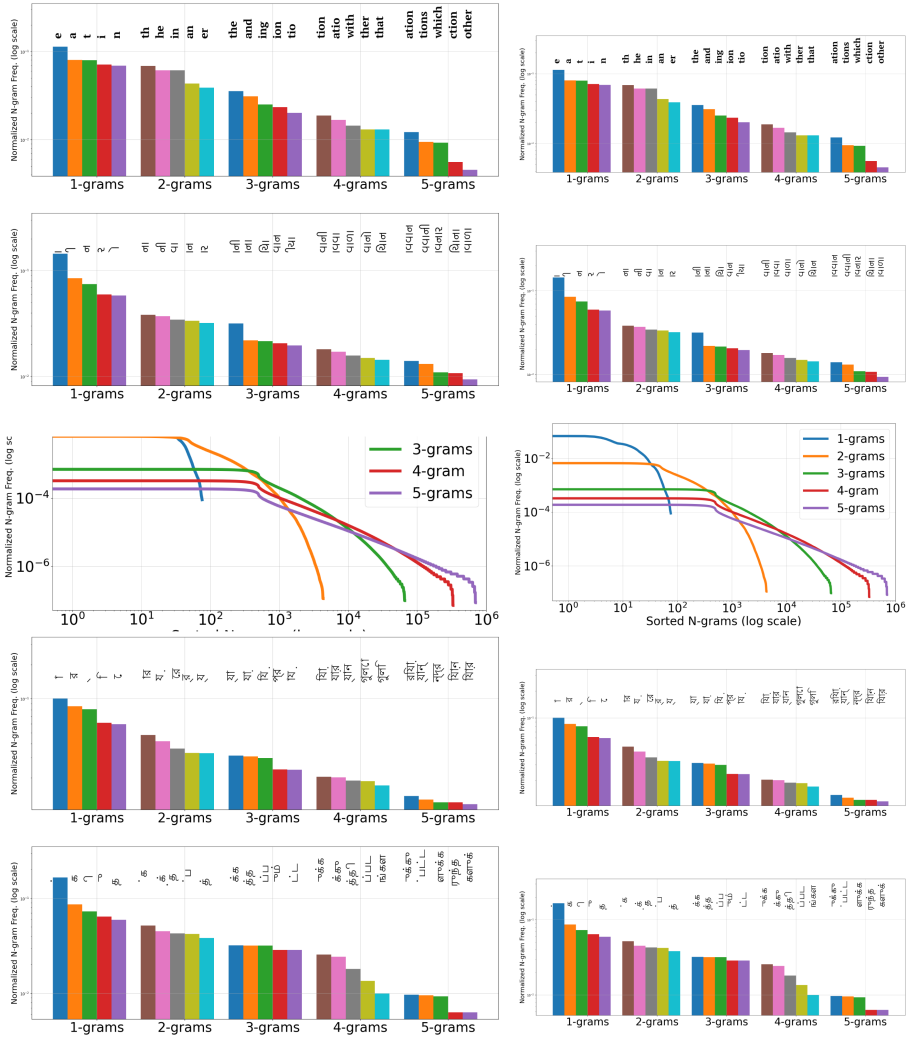
Fig. 3. Distribution of Char. n-grams ($n \in [1, 5]$) from $2.5M$ words in English, Gujarati, Hindi, Bangla, and Tamil (top to bottom): Top-5 (left) and All (right).

plete recognition models discussed in Sect. 1 and the models we use in Sect. 3 among different languages. As discussed in these sections, the Recurrent Neural Networks (RNNs) form another integral component of such reading models. Therefore, we illustrate the distribution of character-level n-grams they learn in Fig. 3[2] for the first five languages we discussed in the previous section (we notice that the last two languages also follow the similar trend). On the left, we show

---

[2] For plots on the right, we use moving average of 10, 100, 1000, 1000, 1000 for 1-grams, 2-grams, 3-grams, 4-grams, and 5-grams, respectively.

the frequency distribution of top-5 n-grams, ($n \in [1,5]$). On the right, we show the frequency distribution of all n-grams with $n \in [1,5]$. We use $2.5M$ words from each language for these plots. We consider both capital and small letters separately for English, as it is crucial for the text recognition task. Despite this, we note that top-5 n-grams are composed of small letters. The Indian languages, however, do not have small and capital letters like English. However, the total number of English letters (given that small letters are different from capitals) is of the same order as Indian languages. The x-values ($\leq$100) for the drops in 1-gram plots (blue curves) of Fig. 3 also illustrates this. So it becomes possible to compare the distributions. Next, we note that most of the top-5 n-grams comprise vowels for all the languages. Moreover, the overall distributions are similar for all the languages. Hence, we propose that the RNN layers' transfer among the models of different languages is worth an investigation.

It is important to note the differences between the n-grams of English and Indian languages. Many of the top-5 n-grams in English are the complete word forms, which is not the case with Indian languages owing to their richness in inflections (or fusions) [29]. Also, note that the second and the third 1-gram for Hindi and Bangla in Fig. 3 (left), known as Halanta, is a common feature of top-5 Indic n-grams. The Halanta forms an essential part of joint glyphs or *aksharas* (as advocated by Vinitha et al. [29]). In Figs. 1 and 2, the vowels, or portions of the joint glyphs for word images in Indian languages, often appear above the top-connector line or below the generic consonants. All this, in addition to complex glyphs in Indian languages, makes transfer learning from English to Indian languages ineffective, which is detailed in Sect. 5. Thus, we also investigate the transferability of features among the Indic scene-text recognition models in the subsequent sections.

3 Models

This section explains the two models we use for transfer learning in Indian languages and a plug-in module we propose for learning the correction mechanism in the recognition systems.

CRNN Model: The first model we train is Convolutional-Recurrent Neural Network (CRNN), which is the combination of CNN and RNN as shown in Fig. 4 (left). The CRNN network architecture consists of three fundamental components, i) an encoder composed of the standard VGG model [24], ii) a decoder consisting of RNN, and iii) a Connectionist Temporal Classification (CTC) layer to align the decoded sequence with ground truth. The CNN-based encoder consists of seven layers to extract feature representations from the input image. The model abandons fully connected layers for compactness and efficiency. It replaces standard squared pooling with 1×2 sized rectangular pooling windows for 3rd and 4th max-pooling layer to yield feature maps with a larger width. A two-layer Bi-directional Long Short-Term Memory (BiLSTM) model, each with a hidden size of 256 units, then decodes the features. During the training phase, the CTC layer provides non-parameterized supervision to align the decoded predictions
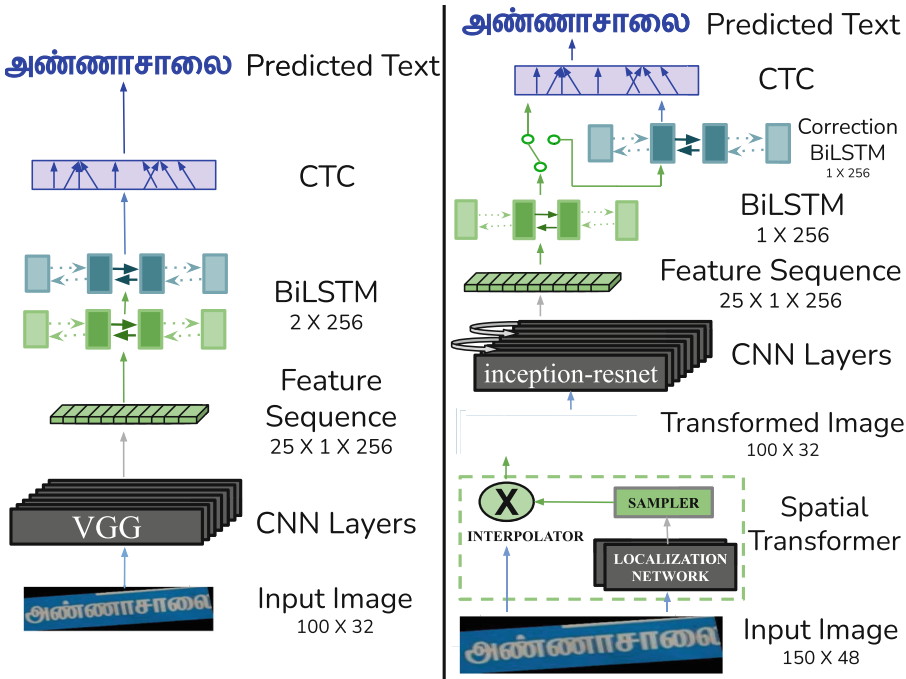
Fig. 4. CRNN model (left) and STAR-Net with a correction BiLSTM (right).

with the ground truth. The greedy decoding is used during the testing stage. We use the PyTorch implementation of the model by Shi et al. [21].

STAR-Net: As shown in Fig. 4 (right), the STAR-Net model consists of three components, i) a Spatial Transformer to handle image distortions, ii) a Residue Feature Extractor consisting of a residue CNN and an RNN, and iii) a CTC layer to align the predicted and ground truth sequences. The transformer consists of a spatial attention mechanism achieved via a CNN-based localization network, a sample, and an interpolator. The localizer predicts the parameters of an affine transformation. The sampler and the nearest-neighbor interpolator use the transformation to obtain a better version of the input image. The transformed image acts as the input to the Residue Feature Extractor, which includes the CNN and a single-layer BiLSTM of 256 units. The CNN used here is based on the inception-resnet architecture, which can extract robust image features required for the task of scene-text recognition [26]. The CTC layer finally provides the non-parameterized supervision for text alignment. The overall model consists of 26 convolutional layers and is end-to-end trainable [11].

Correction BiLSTM: After training the STAR-Net model on a real dataset, we add a correction BiLSTM layer (of size 1×256), an end-to-end trainable module, to the end of the model (see Fig. 4 top-right). We train the complete model again on the same dataset to implicitly learn the error correction mechanism.

Table 2. Results of individual CRNN & STAR-Net models on synthetic datasets.

Language	CRNN-CRR	CRNN-WRR	STAR-Net-CRR	STAR-Net-WRR
English	77.13	38.21	**86.04**	**57.28**
Gujarati	94.43	81.85	**97.80**	**91.40**
Hindi	89.83	73.15	**95.78**	**83.93**
Bangla	91.54	70.76	**95.52**	**82.79**
Tamil	82.86	48.19	**95.40**	**79.90**
Telugu	87.31	58.01	**92.54**	**71.97**
Malayalam	92.12	70.56	**95.84**	**82.10**

4   Experiments

The images, resized to 150×18, form the input of STAR-Net. The spatial transformer module, as shown in Fig. 4 (right), then outputs the image of size 100×32. The inputs to the CNN Layers of CRNN and STAR-Net are of the same size, i.e., 100×32, and the output size is $25 \times 1 \times 256$. The STAR-Net localization network has four plain convolutional layers with 16, 32, 64, and 128 channels. Each layer has the filter size, stride, and padding size of 3, 1, and 1, followed by a 2×2 max-pooling layer with a stride of 2. Finally, a fully connected layer of size 256 outputs the parameters which transform the input image. We train all our models on $2M$ or more synthetic word images as discussed in Sect. 2. We use the batch size of 16 and the ADADELTA optimizer for stochastic gradient descent (SGD) for all the experiments [34]. The number of epochs varies between 10 to 15 for different experiments. We test our models on 0.5 M synthetic images for each language. We use the word images from IIIT-ILST, MLT-17, and MLT-19 datasets for testing on real datasets. We fine-tune the Bangla models on 1200 training images and test them on 673 validation images from the MLT-17 dataset to fairly compare with Bušta et al. [1]. Similarly, we fine-tune only our best Hindi model on the MLT-19 dataset and test it on the IIIT-ILST dataset to compare with OCR-on-the-go (since it is also trained on real data) [20]. To demonstrate generalizability, we also test our models on 3766 Hindi images and 3691 Bangla images available from MLT-19 datasets [14]. For Gujarati and Tamil, we use 75% of word images to fine-tune our models and the remaining 25% for testing.

5   Results

In this section, we discuss the results of our experiments with i) individual models for each language, ii) the transfer learning from English to two Indian languages, and iii) the transfer learning from one Indian language to another.

Performance on Synthetic Datasets: It is essential to compare the results on synthetic datasets of different languages sharing common backgrounds, as it provides a good intuition about the difficulty in reading different scripts. In Tables 2

Table 3. Results of transfer learning (TL) on synthetic datasets. Parenthesis contain results from Table 2. TL among Indic scripts improves STAR-Net results.

Language	CRNN-CRR	CRNN-WRR	STAR-Net-CRR	STAR-Net-WRR
English → Gujarati	92.71 (**94.43**)	77.06 (**81.85**)	97.50 (**97.80**)	90.90 (**91.40**)
English → Hindi	88.11 (**89.83**)	70.12 (**73.15**)	94.50 (**95.78**)	80.90 (**83.93**)
Gujarati → Hindi	**91.98** (89.83)	73.12 (**73.15**)	**96.12** (95.78)	**84.32** (83.93)
Hindi → Bangla	91.13 (**91.54**)	70.22 (**70.76**)	**95.66** (95.52)	**82.81** (82.79)
Bangla → Tamil	81.18 (**82.86**)	44.74 (**48.19**)	**95.95** (95.40)	**81.73** (79.90)
Tamil → Telugu	87.20 (**87.31**)	56.24 (**58.01**)	**93.25** (92.54)	**74.04** (71.97)
Telugu → Malayalam	90.62 (**92.12**)	65.78 (**70.56**)	94.67 (**95.84**)	77.97 (**82.10**)

and 3, we present the results of our experiments with synthetic datasets. As noted in Table 2, the CRNN model achieves the Character Recognition Rates (CRRs) and Word Recognition Rates (WRRs) of i) 77.13% and 38.21% in English and ii) above 82% and 48% on the synthetic dataset of all the Indian languages (refer to columns 1 and 2 of Table 2). The low accuracy on the English synthetic test set is due to the presence of more than 1200 different fonts (refer Sect. 2). Nevertheless, using a large number of fonts in training helps in generalizing the model for real settings [3,6]. The STAR-Net achieves remarkably better performance than CRNN on all the datasets, with the CRRs and WRRs above 90.48 and 65.02 for Indian languages. The reason for this is spatial attention mechanism and powerful residual layers, as discussed in Sect. 3. As shown in columns 3 and 5 of Table 2, the WRR of the models trained in Gujarati, Hindi, and Bangla are higher than the other three Indian languages despite common backgrounds. The experiments show that the scripts in latter languages pose a tougher reading challenge than the scripts in former languages.

We present the results of our transfer learning experiments on the synthetic datasets in Table 3. The best individual model results from Table 2 are included in parenthesis for comparison. We begin with the English models as the base because the models have trained on over 1200 fonts and $17M$ word images as discussed in Sect. 2, and are generic. However, in the first two rows of the table, we note that transferring the layers from the model trained on the English dataset to Gujarati and Hindi is inefficient in improving the results compared to the individual models. The possible reason for the inefficiency is that Indic scripts have many different visual and slightly different n-gram characteristics from English, as discussed in Sect. 2. We then note that as we try to apply transfer learning among Indian languages with CRNN (rows 3–7, columns 1–2 in Table 3), only some combinations work well. However, with STAR-Net (rows 3–7, columns 3–4 in Table 3), transfer learning helps improve results on the synthetic dataset from a simple language to a complex language[3]. For Malayalam, we observe that the individual STAR-Net model is better than the one transferred from Telugu, perhaps due to high average word length (refer Sect. 2).

---

[3] We also discovered experiments on transfer learning from a tricky language to a simple one to be effective but slightly lesser than the reported results.

Table 4. Results on real datasets. FT indicates fine-tuned.

Language	Dataset	# Images	Model	CRR	WRR
Gujarati	Ours	125	CRNN	84.93	64.80
			STAR-Net	85.63	64.00
			STAR-Net Eng→Guj	78.48	60.18
			STAR-Net Hin→Guj	**88.47**	**69.60**
Hindi	IIIT-ILST	1150	Mathew et al. [13]	75.60	42.90
			CRNN	78.84	46.56
			STAR-Net	78.72	46.60
			STAR-Net Eng→Hin	77.43	44.81
			STAR-Net Guj→Hin	79.12	47.79
			OCR-on-the-go [20]	–	51.09
			STAR-Net Guj→Hin FT[a]	**83.64**	**56.77**
Hindi	MLT-19	3766	CRNN	86.56	64.97
			STAR-Net	86.53	65.79
			STAR-Net Guj→Hin	**89.42**	**72.96**
Bangla	MLT-17	673	Bušta et al. [2]	68.60	34.20
			CRNN	71.16	52.74
			STAR-Net	71.56	55.48
			STAR-Net Hin→Ban	72.16	57.01
			W/t Correction BiLSTM	**83.30**	**58.07**
Bangla	MLT-19	3691	CRNN	81.93	74.26
			STAR-Net	82.80	77.48
			STAR-Net Hin→Ban	**82.91**	**78.02**
Tamil	Ours	634	CRNN	**90.17**	70.44
			STAR-Net	89.69	71.54
			STAR-Net Ban→Tam	89.97	**72.95**
Telugu	IIIT-ILST	1211	Mathew et al. [13]	**86.20**	57.20
			CRNN	81.91	58.13
			STAR-Net	82.21	59.12
			STAR-Net Tam→Tel	82.39	**62.13**
Malayalam	IIIT-ILST	807	Mathew et al. [13]	**92.80**	73.40
			CRNN	84.12	70.36
			STAR-Net	91.50	72.73
			STAR-Net Tel→Mal	92.70	**75.21**

[a] Fine-tuned on MLT-19 dataset as discussed earlier. We fine-tune all the layers.

Performance on Real Datasets: Table 4 depicts the performance of our models on the real datasets. At first, we observe that for each Indian language, the overall performance of the individual STAR-Net model is better than the individual CRNN model (except for Gujarati and Hindi, where the results are very close). Based on this and similar observations in the previous section, we present the results of transfer learning experiments on real datasets only with the STAR-Net model[4]. Next, similar to the previous section, we observe that the transfer learning from English to Gujarati and Hindi IIIT-ILST datasets (rows 3

[4] We also tried transfer learning with CRNN; STAR-Net was more effective.

Fig. 5. CNN Layers visualization in the **Top:** CRNN models trained on Hindi, English→Hindi, and Gujarati→Hindi; and **Bottom:** STAR-Net models trained on Gujarati, English→Gujarati, and Hindi→Gujarati. Red boxes indicate the regions where the features for the model transferred from English are activated (as white), whereas the features from the other two models are not. (Color figure online)

and 8 in Table 4) is not as effective as individual models in these Indian languages (rows 2 and 7 in Table 4). Finally, we observe that the performance improves with the transfer learning from a simple language to a complex language, except for Hindi→Gujarati, for which Hindi is the only most straightforward choice. We achieve performance better than the previous works, i.e., Bušta et al. [1], Mathew et al. [13], and OCR-on-the-go [20]. Overall, we observe the increase in WRRs by 6%, 5%, 2% and 23% on IIIT-ILST Hindi, Telugu, and Malayalam, and MLT-17 Bangla datasets compared to the previous works. On the MLT-19 Hindi and Bangla datasets, we achieve gains of 8% and 4% in WRR over the baseline individual CRNN models. On the datasets we release for Gujarati and Tamil, we improve the baselines by 5% and 3% increase in WRRs. We present the

qualitative results of our baseline CRNN models as well as best transfer learning models in Fig. 1. The green and red colors represent the correct predictions and errors, respectively. "_" represents the missing character. As can be seen, most of the mistakes are single-character errors.

Since we observe the highest gain of 23% in WRR (and 4% in CRR) for the MLT-17 Bangla dataset (Table 4), we further try to improve these results. We plug in the correction BiLSTM (refer Sect. 3) to the best model (row 18 of Table 4). The results are shown in row 19 of Table 4. As shown, the correction BiLSTM improves the CRR further by a notable margin of 11% since the BiLSTM works on character level. We also observe the 1% WRR gain, thereby achieving the overall 24% WRR gain (and 15% CRR gain) over Bušta et al. [1].

Features Visualization: In Fig. 5 for the CRNN model (top three triplets), we visualize the learned CNN layers of the individual Hindi model, the "English →Hindi" model, and the "Gujarati→Hindi" model. The red boxes are the regions where the first four CNN layers of the model transferred from English to Hindi are different from the other two models. The feature visualization again strengthens our claim that transfer from the English reading model to any Indian language dataset is inefficient. We notice a similar trend for the Gujarati STAR-Net models, though the initial CNN layers look very similar to word images (bottom three triplets in Fig. 5). The similarity also demonstrates the better learnability of STAR-Net compared to CRNN, as observed in previous sections.

6 Conclusion

We generated 2.5M or more synthetic images in six different Indian languages with varying complexities to investigate the language transfers for two scene-text recognition models. The underlying view is that the transfer of image features is standard in deep models, and the transfer of language text features is a plausible and natural choice for the reading models. However, we observe that transferring the generic English photo-OCR models (trained on over 1200 fonts) to Indian languages is inefficient. Our models transferred from one Indian language to another perform better than the previous works or the new baselines we created for individual languages. We, therefore, set the new benchmarks for scene-text recognition in low-resource Indian languages. The proposed Correction BiLSTM, when plugged into the STAR-Net model and trained end-to-end, further improves the results.

References

1. Bušta, M., Neumann, L., Matas, J.: Deep textspotter: an end-to-end trainable scene text localization and recognition framework. In: ICCV (2017)
2. Bušta, M., Patel, Y., Matas, J.: E2E-MLT - an unconstrained end-to-end method for multi-language scene text. In: Carneiro, G., You, S. (eds.) ACCV 2018. LNCS, vol. 11367, pp. 127–143. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21074-8_11

3. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2315–2324 (2016)

4. Huang, Z., Zhong, Z., Sun, L., Huo, Q.: Mask R-CNN with pyramid attention network for scene text detection. In: WACV, pp. 764–772. IEEE (2019)

5. Iwamura, M., Matsuda, T., Morimoto, N., Sato, H., Ikeda, Y., Kise, K.: Downtown Osaka scene text dataset. In: Hua, G., Jégou, H. (eds.) ECCV 2016, Part I. LNCS, vol. 9913, pp. 440–455. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46604-0_32

6. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. In: Workshop on Deep Learning, NIPS (2014)

7. Jung, J., Lee, S., Cho, M.S., Kim, J.H.: Touch TT: scene text extractor using touchscreen interface. ETRI J. **33**(1), 78–88 (2011)

8. Karatzas, D., et al.: ICDAR 2015 competition on robust reading. In: 2015 ICDAR, pp. 1156–1160. IEEE (2015)

9. Lee, C.Y., Osindero, S.: Recursive recurrent nets with attention modeling for OCR in the wild. In: CVPR, pp. 2231–2239 (2016)

10. Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: Textboxes: a fast text detector with a single deep neural network. In: AAAI, pp. 4161–4167 (2017)

11. Liu, W., Chen, C., Wong, K.Y.K., Su, Z., Han, J.: STAR-Net: a spatial attention residue network for scene text recognition. In: BMVC, vol. 2 (2016)

12. Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., Yan, J.: FOTS: fast oriented text spotting with a unified network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5676–5685 (2018)

13. Mathew, M., Jain, M., Jawahar, C.: Benchmarking scene text recognition in Devanagari, Telugu and Malayalam. In: ICDAR, vol. 7, pp. 42–46. IEEE (2017)

14. Nayef, N., et al.: ICDAR2019 robust reading challenge on multi-lingual scene text detection and recognition-RRC-MLT-2019. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1582–1587. IEEE (2019)

15. Nayef, N., et al.: Robust reading challenge on multi-lingual scene text detection and script identification - RRC-MLT. In: 14th ICDAR, vol. 1, pp. 1454–1459. IEEE (2017)

16. Reddy, S., Mathew, M., Gomez, L., Rusinol, M., Karatzas, D., Jawahar, C.: RoadText-1K: text detection & recognition dataset for driving videos. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 11074–11080. IEEE (2020)

17. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497 (2015)

18. Romera, E., Alvarez, J.M., Bergasa, L.M., Arroyo, R.: ERFnet: efficient residual factorized ConvNet for real-time semantic segmentation. IEEE Trans. Intell. Transp. Syst. **19**(1), 263–272 (2017)

19. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)

20. Saluja, R., Maheshwari, A., Ramakrishnan, G., Chaudhuri, P., Carman, M.: OCR On-the-Go: robust end-to-end systems for reading license plates and street signs. In: 15th IAPR International Conference on Document Analysis and Recognition (ICDAR), pp. 154–159. IEEE (2019)

21. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11), 2298–2304 (2016)

22. Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X.: ASTER: an attentional scene text recognizer with flexible rectification. IEEE Trans. Pattern Anal. Mach. Intell. **41**, 2035–2048 (2018)
23. Shi, B., et al.: ICDAR2017 competition on reading Chinese text in the wild (RCTW-17). In: 14th ICDAR, vol. 1, pp. 1429–1434. IEEE (2017)
24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
25. Sun, Y., et al.: ICDAR 2019 competition on large-scale street view text with partial labeling - RRC-LSVT. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1557–1562. IEEE (2019)
26. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, Inception-Resnet and the impact of residual connections on learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
27. Tian, Z., Huang, W., He, T., He, P., Qiao, Yu.: Detecting text in natural image with connectionist text proposal network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part VIII. LNCS, vol. 9912, pp. 56–72. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_4
28. Tounsi, M., Moalla, I., Alimi, A.M., Lebouregois, F.: Arabic characters recognition in natural scenes using sparse coding for feature representations. In: 13th ICDAR, pp. 1036–1040. IEEE (2015)
29. Vinitha, V., Jawahar, C.: Error detection in indic OCRs. In: 2016 12th IAPR Workshop on Document Analysis Systems (DAS), pp. 180–185. IEEE (2016)
30. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: ICCV, pp. 1457–1464. IEEE (2011)
31. Yousfi, S., Berrani, S.A., Garcia, C.: ALIF: a dataset for Arabic embedded text recognition in TV broadcast. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 1221–1225. IEEE (2015)
32. Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., Darrell, T.: BDD100K: a diverse driving video database with scalable annotation tooling. arXiv preprint arXiv:1805.04687, vol. 2, no. 5, p. 6 (2018)
33. Yuan, T., Zhu, Z., Xu, K., Li, C., Mu, T., Hu, S.: A large Chinese text dataset in the wild. J. Comput. Sci. Technol. **34**(3), 509–521 (2019)
34. Zeiler, M.: ADADELTA: an adaptive learning rate method, p. 1212 (December 2012)
35. Zhang, R., et al.: ICDAR 2019 robust reading challenge on reading Chinese text on signboard. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1577–1581. IEEE (2019)
36. Zhang, Y., Gueguen, L., Zharkov, I., Zhang, P., Seifert, K., Kadlec, B.: Uber-text: a large-scale dataset for optical character recognition from street-level imagery. In: SUNw: Scene Understanding Workshop-CVPR, vol. 2017 (2017)
37. Zhou, X., et al.: East: an efficient and accurate scene text detector. In: CVPR, pp. 2642–2651 (2017)