Towards Boosting the Accuracy of Non-Latin Scene Text Recognition

Sanjana Gunna(✉)[0000−0003−3332−8355], Rohit Saluja[0000−0002−0773−3480], and C. V. Jawahar[0000−0001−6767−7057]

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

Abstract. Scene-text recognition is remarkably better in Latin languages than the non-Latin languages due to several factors like multiple fonts, simplistic vocabulary statistics, updated data generation tools, and writing systems. This paper examines the possible reasons for low accuracy by comparing English datasets with non-Latin languages. We compare various features like the size (width and height) of the word images and word length statistics. Over the last decade, generating synthetic datasets with powerful deep learning techniques has tremendously improved scene-text recognition. Several controlled experiments are performed on English, by varying the number of (i) fonts to create the synthetic data and (ii) created word images. We discover that these factors are critical for the scene-text recognition systems. The English synthetic datasets utilize over 1400 fonts while Arabic and other non-Latin datasets utilize less than 100 fonts for data generation. Since some of these languages are a part of different regions, we garner additional fonts through a region-based search to improve the scene-text recognition models in Arabic and Devanagari. We improve the Word Recognition Rates (WRRs) on Arabic MLT-17 and MLT-19 datasets by 24.54% and 2.32% compared to previous works or baselines. We achieve WRR gains of 7.88% and 3.72% for IIIT-ILST and MLT-19 Devanagari datasets.

Keywords: Scene-text recognition · photo OCR · multilingual OCR · Arabic OCR · Synthetic Data · Generative Adversarial Network.

1 Introduction

The task of scene-text recognition involves reading the text from natural images. It finds applications in aiding the visually impaired, extracting information for map services and geographical information systems by mining data from the street-view-like images [2]. The overall pipeline for scene-text recognition involves a text detection stage followed by a text recognition stage. Predicting the bounding boxes around word images is called text detection [6]. The next

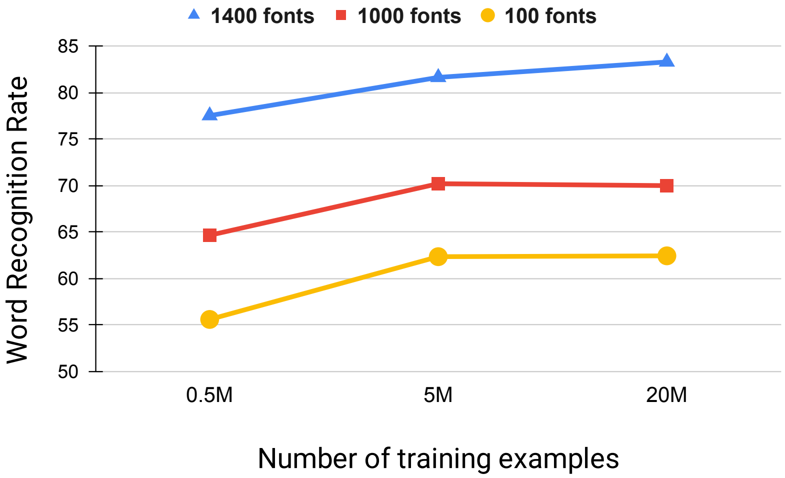

Fig. 1: Comparing STAR-Net's performance on IIIT5K [13] dataset when trained on synthetic data created using a varying number of fonts and training samples.

step involves recognizing text from the cropped text images obtained from the labeled or the predicted bounding boxes [12]. In this work, we focus on improving text recognition in non-Latin languages. Multilingual text recognition has witnessed notable growth due to the impact of globalization leading to international and intercultural communication. Like English, the recognition algorithms proposed for Latin datasets have not successfully recorded similar accuracies on non-Latin datasets. Reading text from non-Latin images is challenging due to the distinct variation in the scripts used, writing systems, scarcity of data, and fonts. In Fig. 1, we illustrate the analysis of Word Recognition Rates (WRR) on the IIIT5K English dataset [13] by varying the number of training samples and fonts used in the synthetic data. The training performed on STAR-Net [11] proves extending the number of fonts leads to better WRR gains than increasing training data. We incorporate the new fonts found using region-based online search to generate synthetic data in Arabic and Devanagari. The motivation behind this work is described in Section 3. The methodology to train the deep neural network on the Arabic and Devanagari datasets is detailed in Section 4. The results and conclusions from this study are presented in Section 5 and 6, respectively. The contributions of this work are as follows:

1. We study the two parameters for synthetic datasets crucial to the performance of the reading models on the IIIT5K English dataset; i) the number of training examples and ii) the number of diverse fonts[1].

[1] We also investigated other reasons for low recognition rates in non-Latin languages, like comparing the size of word images of Latin and non-Latin real datasets but could not find any significant variations (or exciting differences). Moreover, we observe

Table 1: Latin and non-Latin scene-text recognition datasets.

Language	**Datasets**
Multilingual	IIIT-ILST-17 ($3K$ words, 3 languages), MLT-17 ($18K$ scenes, 9 languages), MLT-19 ($20K$ scenes, 10 languages) OCR-on-the-go-19 (1000 scenes, 3 languages), CATALIST-21 (2322 scenes, 3 languages)
Arabic	ARASTEC-15 (260 signboards, hoardings, advertisements), MLT-17,19
Chinese	RCTW-17 ($12K$ scenes), ReCTS-25K-19 ($25K$ signboards), CTW-19 ($32K$ scenes), RRC-LSVT-19 ($450K$ scenes), MLT-17,19
Korean	KAIST-11 ($2.4K$ signboards, book covers, characters) , MLT-17,19
Japanese	DOST-16 ($32K$ images), MLT-17,19
English	SVT-10 (350 scenes), SVT-P-13 (238 scenes, 639 words), IIIT5K-12 ($5K$ words), IC11 (485 scenes, 1564 words), IC13 (462 scenes), IC15 (1500 scenes), COCO-Text-16 ($63.7K$ scenes), CUTE80-14 (80 scenes), Total-Text-19 (2201 scenes), MLT-17,19

2. We share 55 additional fonts in Arabic, and 97 new fonts in Devanagari, which we found using a region-wise online search. These fonts were not used in the previous scene text recognition works.
3. We apply our learnings to improve the state-of-the-art results of two non-Latin languages, Arabic, and Devanagari.

2 Related Work

Recently, there has been an increasing interest in scene-text recognition for a few widely spoken non-Latin languages around the globe, such as Arabic, Chinese, Devanagari, Japanese, Korean. Multi-lingual datasets have been introduced to tackle such languages due to their unique characteristics. As shown in Table 1, Mathew et al. [12] release the IIIT-ILST Dataset containing around $1K$ images each three non-Latin languages. The MLT dataset from the ICDAR'17 RRC contains images from Arabic, Bangla, Chinese, English, French, German, Italian, Japanese, and Korean [15]. The ICDAR'19 RRC builds MLT-19 on top of MLT-17 to containing text from Arabic, Bangla, Chinese, English, French, German, Italian, Japanese, Korean, and Devanagari [14]. Recent OCR-on-the-go and CATALIST[2] datasets include around 1000 and 2322 annotated videos in Marathi, Hindi, and English [19]. Arabic scene-text recognition datasets involve ARASTEC and MLT-17,19 [26]. Chinese datasets cover RCTW,

very high word recognition rates (> 90%) when we tested our non-Latin models on the held-out synthetic datasets, which shows that learning to read the non-Latin glyphs is trivial for the existing deep models. Refer https://github.com/firesans/STRforIndicLanguages for more details.

[2] https://catalist-2021.github.io/

ReCTS-25k, CTW, and RRC-LSVT from ICDAR'19 Robust Reading Competition (RRC) [23,33,31,24]. Korean and Japanese scene-text recognition datasets involve KAIST and DOST [9,7]. Different English datasets are listed in the last row of Table 1 [30,28,20,13,16,10,17,27,3,15,14].
Various models have been proposed for the task of scene-text recognition. Wang et al. [29] present an object recognition module that achieves competitive performance by training on ground truth lexicons without any explicit text detection stage. Shi et al. [21] propose a Convolutional Recurrent Neural Network (CRNN) architecture. It achieves remarkable performances in both lexicon-free and lexicon-based scene-text recognition tasks as is used by Mathew et al. [12] for three non-Latin languages. Liu et al. [11] introduce Spatial Attention Residue Network (STAR-Net) with Spatial Transformer-based Attention Mechanism, which handles image distortions. Shi et al. [22] propose a segmentation-free Attention-based method for Text Recognition (ASTER). Mathew et al. [12] achieves the 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] propose a CNN (and CTC) based method for text localization, script identification, and text recognition and is tested on 11 languages (including Arabic) of MLT-17 dataset. The WRRs are above 65% for Latin and Hangul and are below 47% for the remaining languages (46.2% for Arabic). Therefore, we aim to improve non-Latin recognition models.

3 Motivation and Datasets

This section explains the motivation behind our work. Here we also describe the datasets used for experiments on non-Latin scene text recognition.

Motivation: To study the effect of fonts and training examples on scene-text recognition performance, we randomly sample 100 and 1000 fonts from the set of over 1400 English fonts from previous works [8,5]. For 1400 fonts, we use the datasets available from earlier photo OCR works on synthetic dataset generation [8,5]. For 100 and 1000 fonts, we generate synthetic images by following a simplified methodology proposed by Mathew et al. [12]. Therefore, we create three different synthetic datasets. Moreover, we simultaneously experiment by varying the number of training samples from $0.5M$ to $5M$ to $20M$ samples. By changing the two parameters, we train our model (refer Section 4) on the above synthetic datasets and test them on the IIIT5K dataset. We observe that the Word Recognition Rate (WRR) of the dataset with around $20M$ samples and over 1400 fonts achieves state-of-the-art accuracy on the IIIT5K dataset [11]. As shown in Fig. 1, the WRR of the model trained on $5M$ samples generated using over 1400 fonts is very close to the recorded WRR ($20M$ samples). Moreover, models trained on 1400 fonts outperform the models trained on 1000 and 100 fonts by a margin of 10% because of improved (font) diversity and better but complex dataset generation methods. Also, in Fig.1, as we increase the number of fonts from 1000 to 1400, the WRR gap between the models trained on $5M$ and $20M$ samples moderately improves (from 0% to around 2%). Finally, this

Table 2: Synthetic Data Statistics. μ, σ represent mean, standard deviation.

Language	**# Images**	μ, σ **word length**	**# Fonts**
English	17.5M	5.12, 2.99	>1400
Arabic	5M	6.39, 2.26	140
Devanagari	5M	8.73, 3.10	194

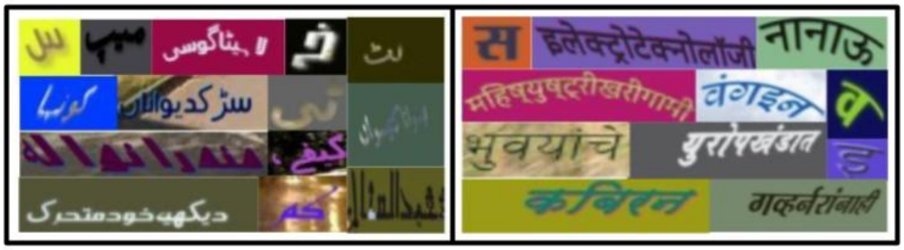

Fig. 2: Synthetic word images in Arabic and Devanagari.

analysis highlights the importance of increasing the fonts in synthetic dataset generation and ultimately improving the scene-text recognition models.

Datasets: As shown in Table 2, we generate over 17M word images in English, 5M word images each in Arabic, and Devanagari, using the tools provided by Mathew et al. [12]. We use 140 and 194 fonts for Arabic and Devanagari, respectively. Previous works use 97 fonts and 85 fonts for these languages [12,1]. Since the two languages are spoken in different regions, we found 55 additional fonts in Arabic and 97 new fonts in Devanagari using the region-wise online search.[3]. We use the additional fonts obtained by region-wise online search, which we will share with this work. As we will see in Section 5, we also perform some of our experiments with these fonts. Sample images of our synthetic data are shown in Fig. 2. As shown in Table 2, English has the lowest average word length among the languages mentioned, while Arabic and Devanagari have comparable average word lengths. Please note that we use over 1400 fonts for English, whereas the number of diverse fonts available for the non-Latin languages is relatively low. We run our models on Arabic and Devnagari test sets from MLT-17, IIIT-ILST, and MLT-19 datasets[4]. The results are summarized in Section 5.

[3] Additional fonts we found using region-based online search are available at: www.sanskritdocuments.org/, www.tinyurl.com/n84kspbx, www.tinyurl.com/7uz2fknu, www.ctan.org/tex-archive/fonts/shobhika?lang=en, www.hindi-fonts.com/, www.fontsc.com/font/tag/arabic, more fonts are shared on https://github.com/firesans/NonLatinPhotoOCR

[4] We could not obtain the ARASTEC dataset we discussed in the previous section.

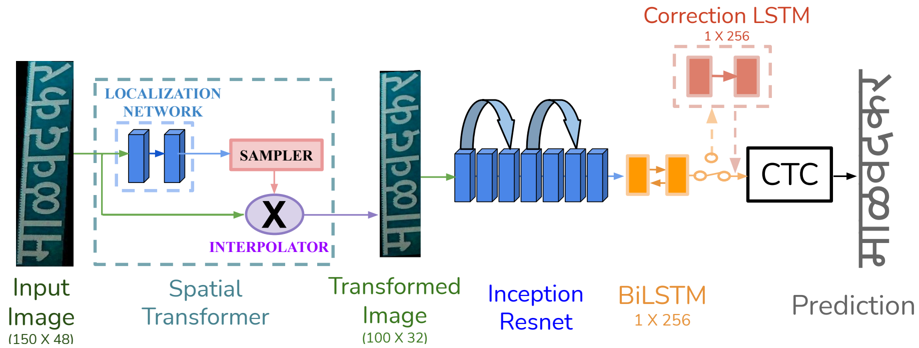

Fig. 3: Model used to train on non-Latin datasets.

4 Underlying Model

We now describe the model we train for our experiments. We use STAR-Net because of its capacity to handle different image distortions [11]. It has a Spatial Transformer network, a Residue Feature Extractor, and a Connectionist Temporal Classification (CTC) layer. As shown in Fig. 3, the first component consists of a spatial attention mechanism achieved via a CNN-based localisation network that helps predict affine transformation parameters to handle image distortions. The second component consists of a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN). The CNN is inception-resnet architecture, which helps in extracting robust image features [25]. The last component provides the non-parameterized supervision for text alignment. The overall end-to-end trainable model consists of 26 convolutional layers [11].

The input to spatial transformer module is of resolution 150×48. The spatial transformer outputs the image of size 100×32 for the next stage (Residue Feature Extractor). We train all our models on $5M$ synthetic word images as discussed in the previous section. We use the batch size of 32 and the ADADELTA optimizer for our experiments [32]. We train each model for 10 epochs and test on Arabic and Devanagari word images from IIIT-ILST, MLT-17, and MLT-19 datasets. Only for the Arabic MLT-17 dataset, we fine-tune our models on training images and test them on validation images to fairly compare with Bušta et al. [1]. For Devanagari, we present the additional results on the IIIT-ILST dataset by fine-tuning our best model on the MLT-19 dataset. We fine-tune all the layers of our model for the two settings mentioned above. To further improve our models, we add an LSTM layer of size 1×256 to the STAR-Net model, pre-trained on synthetic data. The additional layer corrects the model's bias towards the synthetic datasets, and hence we call it correction LSTM. We plug-in the correction LSTM before the CTC layer, as shown in Fig. 3 (top-right). After attaching the LSTM layer, we fine-tune the complete network on the real datasets.

Table 3: Results of our experiments on real datasets. FT means fine-tuned.

Language	Dataset	# Images	Model	CRR	WRR
Arabic	MLT-17	951	Bušta et al. [2]	75.00	46.20
			STAR-Net (85 Fonts) FT	88.48	66.38
			STAR-Net (140 Fonts) FT	89.17	68.51
			STAR-Net (140 Fonts) FT with Correction LSTM	**90.19**	**70.74**
Devanagari	IIIT-ILST	1150	Mathew et al. [12]	75.60	42.90
			STAR-Net (97 Fonts)	77.44	43.38
			STAR-Net (194 Fonts)	77.65	44.27
			STAR-Net (194 Fonts) FT on MLT-19 data	79.45	50.02
			STAR-Net (194 Fonts) FT with Correction LSTM	**80.45**	**50.78**
Arabic	MLT-19	4501	STAR-Net (85 Fonts)	71.15	40.05
			STAR-Net (140 Fonts)	**75.26**	**42.37**
Devanagari	MLT-19	3766	STAR-Net (97 Fonts)	84.60	60.83
			STAR-Net (194 Fonts)	**85.87**	**64.55**

5 Results

Table 3 depicts the performance of our experiments on the real datasets. For the Arabic MLT-17 dataset and Devanagari IIIT-ILST dataset, we achieve recognition rates better than Bušta et al. [1] and Mathew et al. [12]. With STAR-Net model trained on < 100 fonts (refer Section 3), we achieve 13.48% and 20.18% gains in Character Recognition Rate (CRR) and Word Recognition Rate (WRR) for Arabic, and 1.84% and 0.48% improvements for Devanagari over the previous works (compare rows 1, 2 and 5, 6 in the last column of Table 3). The CRR and WRR further improve by training the models on the same amount of training data synthesized with $>=$ 140 fonts (rows 3 and 7 in the last column of Table 3). By fine-tuning the Devanagari model on the MLT-19 dataset, the CRR and WRR gains raise to 3.85% and 7.12%. By adding the correction LSTM layer to the best models, we achieve the highest CRR and WRR gains of 15.19% and 24.54% for Arabic, and 5.25% and 7.88% for Devanagari, over the previous works. The final results for the two datasets discussed above can be seen in rows 3 and 7 of the last column of Table 3.

As shown in Table 3, for the MLT-19 Arabic dataset, the model trained on $5M$ samples generated using 85 fonts achieve the CRR of 71.15% and WRR of 40.05%. Increasing the number of diverse fonts to 140 gives a CRR gain of 4.11% and a WRR gain of 2.32%. For the MLT-19 Devanagari dataset, the model trained on $5M$ samples generated using 97 fonts achieves the CRR of 84.60% and WRR of 60.83%. Increasing the number of fonts to 194 gives a CRR gain of 1.27% and a WRR gain of 3.72%. It is also interesting to note that the WRR of our models on MLT-17 Arabic and MLT-19 Devanagari datasets are very close

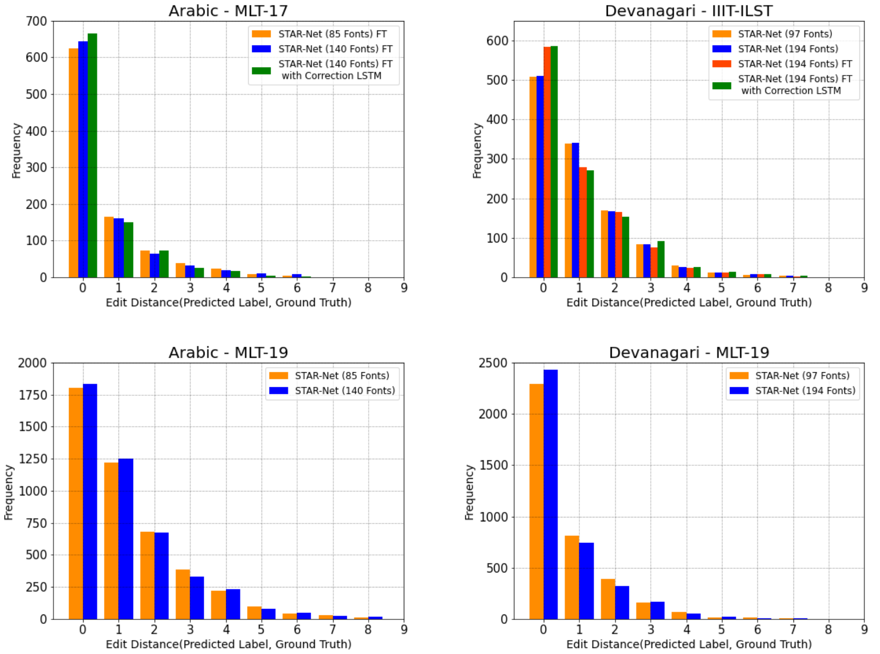

Fig. 4: Histogram of correct words ($x = 0$) and words with x errors ($x > 0$). FT represents the models fine-tuned on real datasets.

to the WRR of the English model trained on $5M$ samples generated using 100 fonts (refer to the yellow curve in Fig. 1). It supports our claim that the number of fonts used to create the synthetic dataset plays a crucial role in improving the photo OCR models in different languages.

To present the overall improvements by utilizing extra fonts and correction LSTM at a higher level, we examine the histograms of edit distance between the pairs of predicted and corresponding ground truth words in Fig. 4. Such histograms are used in one of the previous works on OCR error corrections [18]. The bars at the edit distance of 0 represent the words correctly predicted by the models. The subsequent bars at edit distance $n > 0$ represent the number of words with x erroneous characters. As it can be seen in Fig. 4, overall, with the increase in the number of fonts and subsequently with correction LSTM, i) the number of correct words ($x = 0$) increase for each dataset, and ii) the number of incorrect words ($x > 0$) reduces for many values of x for the different datasets. We observe few exceptions in each histogram where the frequency of incorrect words is higher for the best model than others, e.g., at edit distance of 2 for the Arabic MLT-17 dataset. The differences (or exceptions) show that the recognitions by different models complement each other.

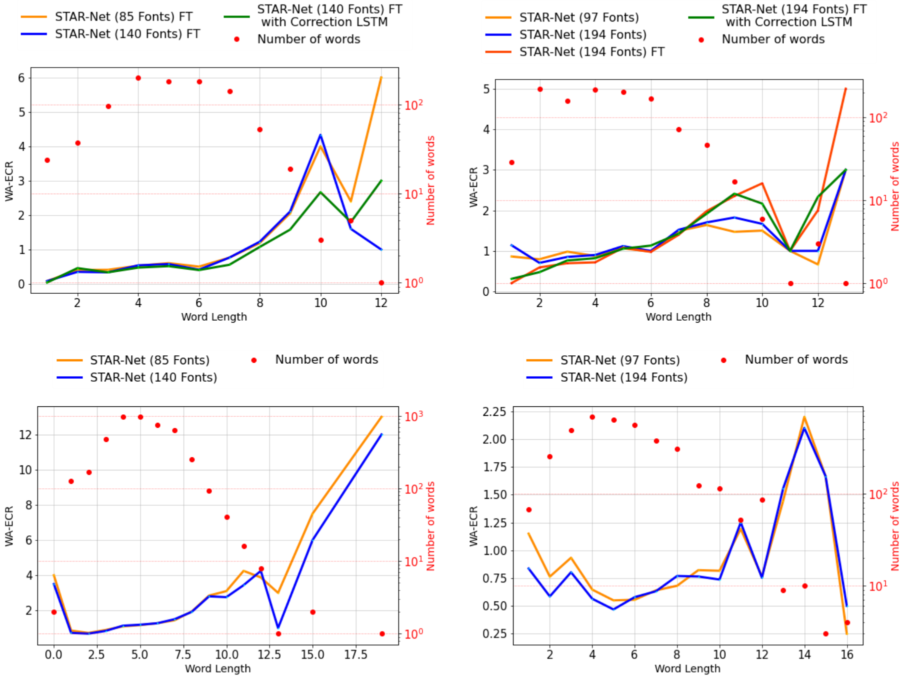

Fig. 5: Clockwise from top-left: WA-ECR of our models tested on MLT-17 Arabic, IIIT-ILST Devanagari, MLT-19 Devanagari, and MLT-19 Arabic datasets.

Another exciting way to compare the output of different OCR systems is Word-Averaged Erroneous Character Rate (WA-ECR), as proposed by Agam et al. [4]. The WA-ECR is the ratio of i) the number of erroneous characters in the set of all l-length ground truth words (e_l), and ii) the number of l-length ground truth words (n_l) in the test set. As shown in the red dots and the right y-axis of the plots in Fig. 5, the frequency of words generally reduces with an increase in word length after $x = 4$. Therefore, the denominator term tends to decrease the WA-ECR for short-length words. Moreover, as the word length increases, it becomes difficult for the OCR model to predict all the characters correctly. Naturally, the WA-ECR tends to increase with the increase in word length for an OCR system. In Fig. 5, we observe that our models trained on $>=$ 140 fonts (blue curves) are having lower WA-ECR across different word lengths as compared to the ones trained on $<$ 100 fonts (orange curves). For the IIIT-ILST dataset, the model, trained on 194 fonts, performs poorly on the long words ($x > 8$ in the top-right plot of Fig. 5), and the correction LSTM further enhances this effect. On the contrary, we observe that the Correction LSTM reduces WA-ECR for the MLT-17 Arabic dataset for word lengths in the range $[6, 11]$ (compare green and blue curves in the top-left plot). Interestingly, the WA-ECR of some of our models drops after word-length of 10 and 14 for the MLT-19 Arabic and MLT-19

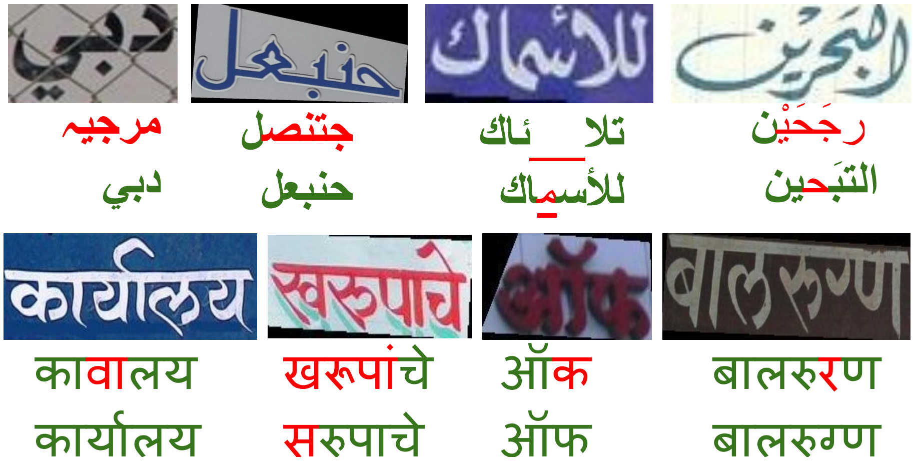

Fig. 6: Real word images in Arabic (top) and Devanagari (bottom), Below the images: predictions from i) baseline model trained on < 100 fonts, ii) model trained on ≥ 140 fonts. Green & red represent correct predictions and errors.

Devanagari datasets (see blue curve in the top-left plot and the two curves in the bottom-right plot of Fig. 5).

In Fig. 6, we present the qualitative results of our models. The green and red colors represent the predictions and errors. As shown, the models trained on over 140 fonts perform better than the models trained on < 100 fonts. Overall, the experiments support our claim that the diversity in fonts used to generate synthetic datasets is crucial for improving the existing non-Latin scene-text recognition systems.

6 Conclusion

We carried out a series of controlled experiments in English to highlight the importance of font diversity and the number of synthetic examples in improving the scene-text recognition accuracy. We augmented the font set of two non-Latin scripts, Arabic and Devanagari, with new fonts obtained by region-based online search. We generated $5M$ synthetic images in two languages. Our experiments show improvements over the previous works and baselines trained on lesser fonts. We further improve our results by introducing the correction LSTM into the models to reduce the bias towards the synthetic data. Finally, we affirm that more fonts are required to improve the existing non-Latin systems. For future work in this area, we plan to employ human designers or Generative Adversarial Networks (GAN) based font generators to boost the accuracy of non-Latin scene-text recognition.

References

1. Bušta, M., Neumann, L., Matas, J.: Deep textspotter: An end-to-end trainable scene text localization and recognition framework. ICCV (2017)
2. Bušta, M., Patel, Y., Matas, J.: E2E-MLT-an Unconstrained End-to-End Method for Multi-Language Scene Text. In: Asian Conference on Computer Vision. pp. 127–143. Springer (2018)
3. Chng, C.K., Chan, C.S.: Total-Text: A Comprehensive Dataset for Scene Text Detection and Recognition. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) **01**, 935–942 (2017)
4. Dwivedi, A., Saluja, R., Kiran Sarvadevabhatla, R.: An OCR for Classical Indic Documents Containing Arbitrarily Long Words. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2020)
5. 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)
6. 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)
7. Iwamura, M., Matsuda, T., Morimoto, N., Sato, H., Ikeda, Y., Kise, K.: Downtown Osaka Scene Text Dataset. In: ECCV. pp. 440–455. Springer (2016)
8. 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)
9. Jung, J., Lee, S., Cho, M.S., Kim, J.H.: Touch TT: Scene Text Extractor using Touchscreen Interface. ETRI Journal **33**(1), 78–88 (2011)
10. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Bigorda, L.G.i., Mestre, S.R., Mas, J., Mota, D.F., Almazàn, J.A., de las Heras, L.P.: ICDAR 2013 Robust Reading Competition. p. 1484–1493. ICDAR '13, IEEE Computer Society, USA (2013)
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. Mathew, M., Jain, M., Jawahar, C.: Benchmarking Scene Text Recognition in Devanagari, Telugu and Malayalam. In: ICDAR. vol. 7, pp. 42–46. IEEE (2017)
13. Mishra, A., Alahari, K., Jawahar, C.V.: Scene text recognition using higher order language priors. In: BMVC (2012)
14. Nayef, N., Patel, Y., Busta, M., Chowdhury, P.N., Karatzas, D., Khlif, W., Matas, J., Pal, U., Burie, J.C., Liu, C.l., 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., Yin, F., Bizid, I., Choi, H., Feng, Y., Karatzas, D., Luo, Z., Pal, U., Rigaud, C., Chazalon, J., 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. Phan, T., Shivakumara, P., Tian, S., Tan, C.: Recognizing Text with Perspective Distortion in Natural Scenes. 2013 IEEE International Conference on Computer Vision pp. 569–576 (2013)
17. Risnumawan, A., Shivakumara, P., Chan, C.S., Tan, C.L.: A robust arbitrary text detection system for natural scene images. Expert Systems with Applications **41**, 8027–8048 (12 2014)

18. Saluja, R., Adiga, D., Chaudhuri, P., Ramakrishnan, G., Carman, M.: Error Detection and Corrections in Indic OCR using LSTMs. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 17–22. IEEE (2017)
19. 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)
20. Shahab, A., Shafait, F., Dengel, A.: Icdar 2011 robust reading competition challenge 2: Reading text in scene images. 2011 International Conference on Document Analysis and Recognition pp. 1491–1496 (2011)
21. Shi, B., Bai, X., Yao, C.: An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and I]ts Application to Scene Text Recognition. IEEE transactions on pattern analysis and machine intelligence **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 Transactions on Pattern Analysis and Machine Intelligence (2018)
23. Shi, B., Yao, C., Liao, M., Yang, M., Xu, P., Cui, L., Belongie, S., Lu, S., Bai, X.: ICDAR2017 Competition on Reading Chinese Text in the Wild (RCTW-17). In: 14th ICDAR. vol. 1, pp. 1429–1434. IEEE (2017)
24. Sun, Y., Ni, Z., Chng, C.K., Liu, Y., Luo, C., Ng, C.C., Han, J., Ding, E., Liu, J., Karatzas, D., 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)
25. 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)
26. 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)
27. Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.: COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images. arXiv preprint arXiv:1601.07140 (2016)
28. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: 2011 International Conference on Computer Vision. pp. 1457–1464 (2011)
29. Wang, K., Babenko, B., Belongie, S.: End-to-End Scene Text Recognition. In: ICCV. pp. 1457–1464. IEEE (2011)
30. Wang, K., Belongie, S.: Word Spotting in the Wild. In: European conference on computer vision. pp. 591–604. Springer (2010)
31. Yuan, T., Zhu, Z., Xu, K., Li, C., Mu, T., Hu, S.: A Large Chinese Text Dataset in the Wild. Journal of Computer Science and Technology **34**(3), 509–521 (2019)
32. Zeiler, M.: Adadelta: An adaptive learning rate method **1212** (12 2012)
33. Zhang, R., Zhou, Y., Jiang, Q., Song, Q., Li, N., Zhou, K., Wang, L., Wang, D., Liao, M., Yang, M., 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)