Towards Accurate Identification and Removal of *Shirorekha* from *Off-line* Handwritten Devanagari word Documents

Mohammad Idrees Bhat [*,1] , B. Sharada[1], Sk. Md Obaidullah[2], and Mohammad Imran[3]
[1]Department of Studies in Computer Science, University of Mysore, Mysore-570006, Karnataka-INDIA
[2]Department of Computer Science and Engineering, Aliah University, Kolkata-700160, West Bengal-INDIA
[3]NTT DATA Information processing services private limited, whitefield Banglore-560066, Karnataka-INDIA
[1]{mohammadbhatrsh*, sharada}@compsci.uni-mysore.ac.in, { [2]sk.obaidullah, [3]emrangi}@gmail.com

***Abstract* — *Shirorekha* identification and removal is an important and a challenging pre-processing stage in almost all machine interpretations for handwritten Devanagari documents. Within this area of investigation, all studies are designed based on traditional image processing techniques. Which are mainly based on hand-engineering and learn local transformations only. However, it can also be viewed as a supervised classification task in which each pixel, in a document, is examined/ queried so that those classified as shirorekha are removed. For this purpose, we extended this area of investigation by designing an encoder-decoder based convolutional neural network (EDCNN). Which have demonstrated, from various studies, that they learn image intricacies very well. The contribution of this work is three-fold, first, we created our own handwritten word dataset comprising of words with and without shirorekha, such that, effective training takes place. Next, we trained the proposed network with binary as well as in gray scale formats. Finally, we demonstrated that the proposed approach is accurate and generalizable.**

Keywords—Shirorekha; Handwritten Devanagari document; Encoder-Decoder Convolutional Neural Network; pixel-labelling; Generalizable and adaptable.

I. INTRODUCTION

Shirorekha[1] is a horizontal line, running on top of all constituent characters of a word, in a Devanagari script (see Fig.1). For this reason, it is considered a key discriminative feature for automatic machine interpretation [1]. The significance of it can be understood from a statistical analysis carried out in [2].That revealed the fact that the frequency of the occurrence of shirorekha in Devanagari word documents is about 99%. That is why shirorekha is first identified and later, for various attempts, removed/segmented. Like, for example, in analytical-based handwritten word recognition. Furthermore, it plays an important role in script identification [3], [4],skew estimation and correction [5], [6], and text line/block identification and segmentation [7], [8], etc. That is, precise identification and removal of shirorekha leads to accurate results in forthcoming steps of any machine interpretation pipeline. In contrast to machine printed documents, varying individual handwriting styles, script characteristics, imperfect scanning of manuscripts, and cursiveness in the handwriting leads to discontinuous, wavy, skewed and degraded shirorekha. Thus, making its identification and subsequent removal from a handwritten document *a challenging task*.

Although identification and removal of shirorekha can be seen as a simple task, it is often challenging to get accurate results. This is mainly due to the previous works that have revolved around the traditional image processing (IP) techniques. That primarily require some heuristics [9], [10], experimental rules [11] , a priori knowledge of a document [12], [13], and hand engineering [14]. Notwithstanding all the approaches, shirorekha identification and removal is still inaccurate and often produce erroneous errors. As a matter of fact, they cease to be optimal (i.e. they remove essential character information) when there is a slight change in the type of a document with respect to style, skew, deformations, and handwriting. As the approaches are designed such that they learn local transformations, only i.e. they lack flexibility and generalizability.

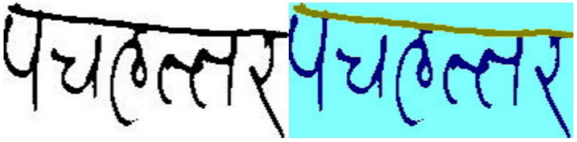

Fig.1 Segmentation of a handwritten Devanagari word into pre-defined classes by using pixel-labeling. Background, Characters, and Shirorekha represented with turquoise blue, Persian blue and citron brown, respectively.

In an alternative way, the identification and removal of shirorekha can be posed/viewed as a supervised classification task. Such that each pixel in a document is examined/queried so that those classified as shirorekha are removed. To the best of our knowledge, no study has tackled this problem as a supervised classification task. This is rather striking, as the classification has a strong foundation and is well-studied in the broad area of machine learning. Note that, viewing the problem in this way both identification and removal can be achieved simultaneously. Which is in contrast to IP based approaches where it is a two-stage process i.e. identification and removal. More importantly, the techniques based on supervised classification learn image intricacies very well [15], [16]. In order to evaluate the effectiveness of the proposed approach, we classify each pixel in the following pre-defined classes: (i) background, (ii) character, and (iii) shirorekha, as shown in Fig.1.

It is now generally agreed that convolutional neural networks (CNNs), and its different architectures achieve state-of-the-art results in different and inverse application domains. This notion is also strengthened with the active interest of researchers towards employing CNNs in core areas of computer vision [15], [16]. For example, in [17], different artifacts of a document image, such as, text, comments, decorations, and background are segmented with CNN-based architecture. The superior performance of CNN is three-fold. First, CNN is robust in learning spatial/contextual information. Next, with this information effect of noise is minimized. Finally, once trained, CNNs offer a generalized/adaptable framework and can be tailored/adapted to different problem domains. Therefore, with this motivation, in this paper, we propose the use of encoder-decoder based CNN (EDCNN) to solve the problem of identification and removal of shirorekha from a handwritten document.

The rest of the paper is categorized as follows. In Section II we describe our proposed EDCNN approach for shirorekha identification and removal. Experimental results and comparative study are given in Section III . Finally, we present conclusion and future work in Section IV .

[1] Constituent of two Sanskrit words, '*Siro*' means something which is present in the upper part and *'rekha*" means line. It is also present in other Indic scripts, like Bangla script.

978-1-7281-9966-5/20/$31.00 ©2020 IEEE
DOI 10.1109/ICFHR2020.2020.00051

II. Proposed Method

A Pre-processing

The main aim of pre-processing is to enhance the quality of an image for further processing. For that reason, at first, word images are resized to a fixed size of 256×256 . The motivation for that is two-fold; first, larger the size, lesser is the shrinking (i.e. low deformations of features/ patterns present in an image) and second, model architecture is size dependent. Next, resized images are median filtered with a window size 3×3 . As stated, we have taken both formats into account (i.e., gray and binary), therefore, Otsu's method [18], has been used for binarisation. Finally, for normalization, a zero-centering technique is employed. That on one hand, increases the learning process and on the other hand, avoids the effect on a non-linearity response with the slight change in filters.

B. Encoder Decoder CNN architecture (EDCNN)

CNN-based networks benefit from various artifacts like local connections, learnable/shared weights, and multiple layers, etc., in order to, learn a data representation of underlying classification problem. Our proposed EDCNN architecture is somewhat similar to [19], that is used for semantic segmentation for natural images. However, in contrast, we decided to choose encoder part empirically that comprises sets of (or stack of) convolution, batch normalization, ReLu and max-pooling layers. That suits the problem at hand and also reduces the number of parameters in the encoder part significantly. Against, each encoder layer[2] there is a corresponding decoder layer[3] comprising a stack of a transposed convolutional layer, batch normalization and ReLu layers. Finally, soft-max classifier after receiving the output from the last decoder layer, computes probabilities for each pixel.

In order to restrict exhaustive search space, we have designed EDCNN by taking into account following constant parameters: (i)3×3 filters and 2×2 max-pooling per encoder layer, (ii) 50% of dropout (iii) in contrast to, image patches, input to the network is set to whole image i.e. 256×256 . Other, parameters like a number of convolution layers (depth) and a number of filters per layer (neurons) are determined empirically. Note, the total number of encoder layers in EDCNN depends on a number of convolutional layers. Adam optimizer [20], is used to learn the weights of a network, with a batch size of 8. Also, for L_2- regularization (weight decay) and initial learning rate 5×10^{-4} and 10^{-3} values were used. Cross-entropy loss function is used after normalizing pixel count with a median frequency technique [15], to avoid dominance of pixels belonging to background class (Table1 shows the total number of pixels belonging to all classes in a Training set). In order to choose an appropriate epoch (Note, we have set the total number of epochs to 100) and avoid over fitting an early stopping criteria [21], with a validation patience of 10 is employed. After, optimal parameters are determined, we added skip connections [22], from one encoder layer to the corresponding decoder layer to observe any improvement in classification accuracy

Once the proposed EDCNN has learned how to classify a pixel in one among: (i) background, (ii) character and (iii) shirorekha, it can be used to segment/remove the shirorekha from any handwritten Devanagari word document. To do so, identified shirorekha pixels can be forwarded into the network and finally be removed.

Table1: The Total number of pixels belonging to predefined classes in a Training set, where class 1=Background, class 2= Character, and class 3 = Shirorekha.

Type	Class Labels		
	1	2	3
Gray	8.5e+07	2.46e+06	1.4578e+07
Binary	8.5097e+07	2.4772e+06	1.3138e+07

III. Experiments

A. Dataset

Identification and removal being the pre-processing step, therefore, there was no dataset available as such. We addressed this limitation, in this paper, and created our own dataset. For this purpose, we used ICDAR'11 legal amount Devanagari dataset [23], and selected complex structured 10 legal amount classes (each with 80 samples). Next, we manually removed the shirorekha from each word sample in each class. Now each class carries word samples with and corresponding word samples without shirorekha (i.e. word samples where shirorekha is removed). Such that, pixel at position (i,j) in a word without shirorekha, gives a ground truth for the pixel at position (i,j) in a word with shirorekha. More often, various attempts are carried out either in gray-scale or in binary formats. Therefore, both the formats are taken into consideration. Finally, we manually labelled each pixel with pre-defined classes: (i) background, (ii) character, and (iii) shirorekha. Note that the combined pixel between shirorekha and character is treated as the pixel for a character class. Table2 gives a brief description about the dataset and Fig.2 shows some word samples from the dataset.

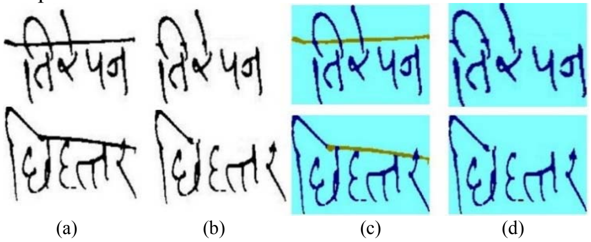

Fig.2 Illustration of the dataset (a) pre-processed and resized word images with shirorekha (b) corresponding word images without shirorekha (c) pixel labels overlaid on a word image with shirorekha, and (d) pixel labels overlaid on a word images without shirorekha.

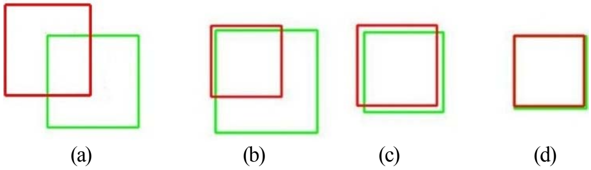

Fig.3 An illustration for the mIoU in which green square indicates ground truth and red square indicates a prediction , respectively (a) bad mIoU with score 0.23, (b) bad mIoU with score 0.412 (c) good mIoU with score 0.752 , and (d) excellent mIoU with score 0.914.

B.Evaluation Metric

We used a mean intersection over union (mIoU) [15], [16], to arrive at the optimal parameters (see Section IIB) of EDCNN and to demonstrate the efficacy of the proposed approach. It computes a ratio

[2] Here encoder layer= convolutional layer + Batch Normalization layer + ReLu + Max-pooling layers

[3] Decoder layer = transposed convolutional layer+ Batch Normalization layer + ReLu

Table3: Mean IoU achieved on validation sets for the empirical evaluation of parameters for proposed EDCNN (bold face font indicates maximum results obtained)

Ratio	Depth	Filters per layer									
		Binary					Gray				
		16	32	64	96	128	16	32	64	96	128
50:25:25	1	67.6	68.4	68.7	69.0	69.1	68.3	70.3	71.8	73.0	74.0
	2	75.0	76.7	78.8	80.9	81.1	73.9	76.1	78.0	79.2	77.1
	3	76.7	81.5	84.3	88.1	88.2	75.6	80.3	82.9	83.0	84.2
	4	74.1	80.8	85.2	**89.0**	88.9	73.5	79.7	83.2	**87.3**	87.1
	5	54.1	49.4	74.4	81.5	82.5	51.4	38.2	73.0	86.2	86.0
60:20:20	1	67.8	68.5	68.8	69.1	69.1	69.0	70.9	72.2	73.2	73.6
	2	76.0	77.5	79.3	80.9	81.2	75.2	77.5	78.8	81.2	81.5
	3	78.7	82.7	85.3	89.0	87.7	77.1	81.4	83.8	87.7	87.3
	4	76.0	81.9	85.6	**89.0**	88.6	75.4	80.7	84.2	**88.8**	88.4
	5	56.6	57.8	75.7	82.7	83.1	57.7	41.2	74.5	80.7	81.9

Table 4: EDCNN results, with and with-out skip connections, on independent test sets.

Ratio	Depth	Filters	Without skip-connections		With skip-connections	
			Binary	Gray	Binary	Gray
50:25:25	4	96	89.9	87.8	**95.1**	**95.1**
60:20:20			89.4	88.9	**95.3**	**95.4**

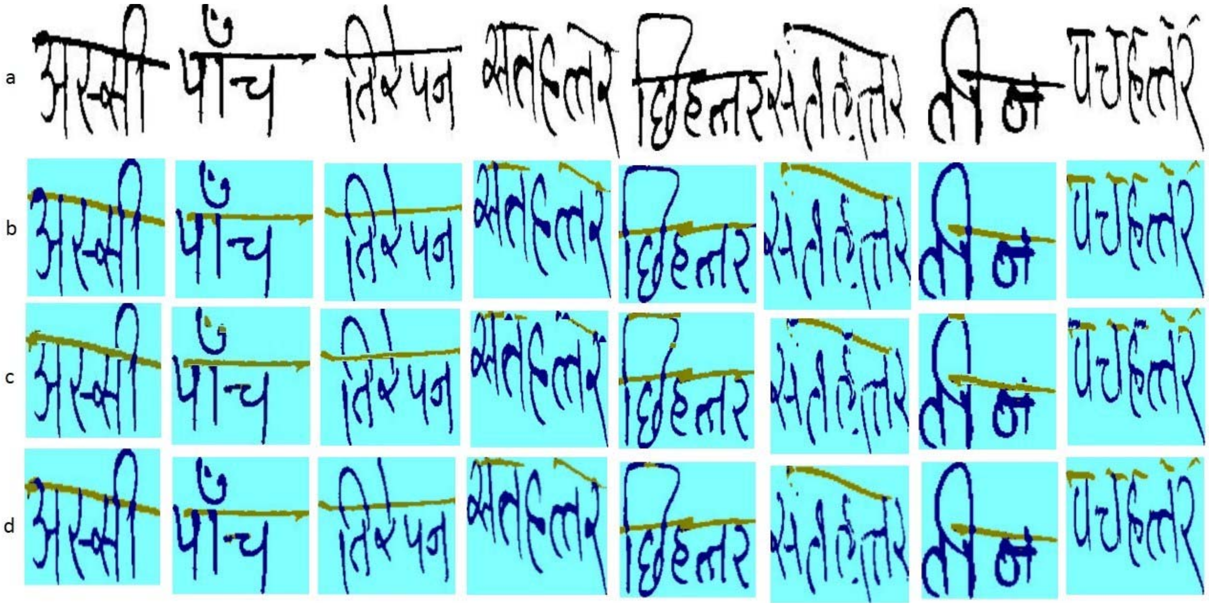

Fig.4 Semantic segmentation results (a) original image (b) ground truth (c) without skip connections (d) with skip-connections

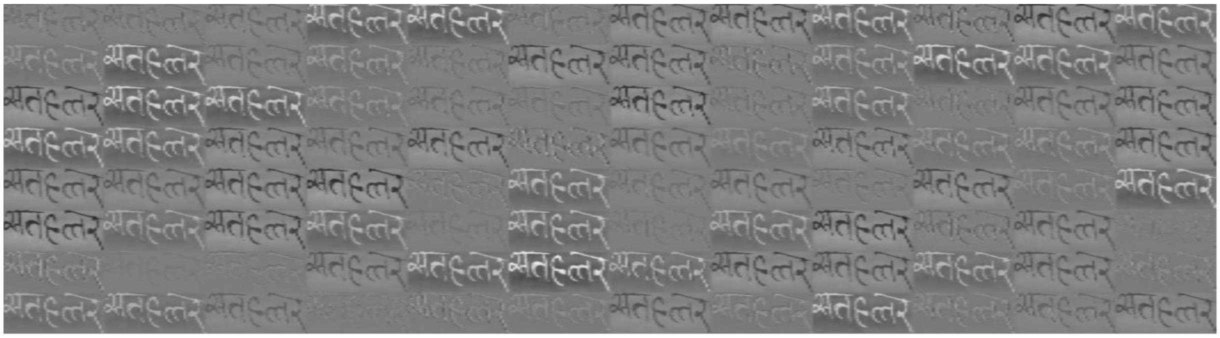

Fig.5 Intermediate representation of convolutional layer 1 with 96 filters of an arbitrarily chosen word sample.

Table2: A brief description of a dataset

Format	Image Size (pixels)	Word samples
Gray-scale	256×256	10×160
Binary	256×256	10×160
Total		**3,200**

between prediction and ground-truth i.e., ratio between the area of intersection/overlap and area of union, as shown in Fig.3. Note, mIoU is computed class-wise and later averaged. Further, mIoU greater than 0.5 is generally considered as good prediction.

C. Netwrok Tunning

As stated in section IIB , some parameters have to be selected empirically. Such that suitable topology for proposed EDCNN is determined. For that reason, we partitioned the dataset (in both formats i.e. gray and binary) in the ratios of $50:25:25$ and $60:20:20$ of training, validation and testing sets, respectively. We performed grid search for a number of convolution + batch normalization+ ReLu + max-pooling layers (encoder layer) over: $1,2,3,4,5$ and number of filters (neurons) per convolution layer over: $16,32,64,96,128$. As stated earlier, we have used validation patience of 10 i.e. training automatically stops when the learning performance does not show any improvement on repetitive 10 epochs, on the corresponding validation set. Note that, we have performed grid search separately for gray-scale and binary formats, as shown in Table 3. From now on, we shall select that topology for proposed EDCNN which shows the best results (in terms of mIoU) on validation set.

Broadly speaking, skip-connections allow gradients to flow uninterrupted from an encoder layer to its corresponding decoder layer. Consequently increases the performance, therefore, with this motivation we added skip-connections to the selected topology. We observe from Table 4 that recognition accuracy has been improved significantly on independent test sets. It is also indicated from Fig4 which compares segmentation results on some common test word samples (i.e. boundaries between classes have been precisely localized). Fig.5 shows the intermediate representation encoded with convolutional layer 1 (conv1) of EDCNN. It shows all 96 filters in 8×12 grid i.e. each image, in a grid, is for one particular channel. Activations are scaled up such that minimum activation is kept to 0 and maximum activation to 1. A bright tile in the grid indicates that it is strongly activated.

Finally, we form our final model (that performs best in terms of mIoU i.e. convolutional layers = 4, a number of filters = 96 and with skip-connections (Table4)) for shirorekha identification and removal from *off-line* handwritten Devanagari word documents, henceforth, called as *ShirorekhaNet* (see Fig.8). Fig.6 shows the normalized confusion matrix obtained in one of the experiment. Therefore, subsequent experiments for comparison with other models is performed with this *ShirorekhaNet.*

D.Comparative study

As stated earlier, shirorekha identification and removal is a pre-processing stage. As a result, there are no results shown in the previous works towards this end, against which we can compare the results obtained with *ShirorekhaNet*. Therefore, we decided to implement two popular methods, the first method is proposed in [11] (Seg1), and another *de-facto* method based on horizontal projections (Seg2,) on our created dataset. The results obtained are compared with the corresponding pixel-labeled images of the test set created in this paper (see Section *IIIA*) and are shown in Table 5. Moreover, Seg1 and Seg2 are applied on those test sets on which *ShirorekhaNet* is applied for fair comparisons. Fig.9 shows, some of the samples from which shirorekha is segmented first from Seg1, Seg2, and later with *ShirorekhaNet*, respectively. One can observe the accuracy and precision with which *ShirorekhaNet* has identified shirorekha. Very often, segmentation based approaches removed essential character information (Fig.9). Therefore with this motivation for shirorekha identification and removal supervised classification approaches *in general*, and deep learning-based architectures, *in particular*, need further investigations towards accurate identification and removal of shirorekha from handwritten Devanagari word documents.

In order to show that *ShirorekhaNet* is generalizable and adaptable to other types of Devanagari word documents, we used the two datasets. The First dataset was created as a part of Indic script identification at the word level [5] (script-dB). Second, was created for holistic recognition of handwritten Devanagari words [9] (holistic-dB). For this experimentation we used 40 words from each dataset. Then we followed the same procedure, for pre-processing, sated in section IIA . Finally, we labelled each pixel according to their belongingness, i.e. background, character, and shirorekha, respectively. Note, we carried experiments for both the formats i.e. gray and binary. A snapshot from the datasets is shown in Fig.7.

Table 5: mIoU obtained with previous work and our proposed method.

Method		Mean Inter-section over Union (mIoU)	
		Binary	Gray-scale
Seg1	50:25:25	86.2	86.4
	60:20:20	87.0	88.1
Seg2	50:25:25	74.1	75.2
	60:20:20	70.4	70.3
ShirorekhaNet	50:25:25	**95.1**	**95.1**
	60:20:20	**95.3**	**95.4**

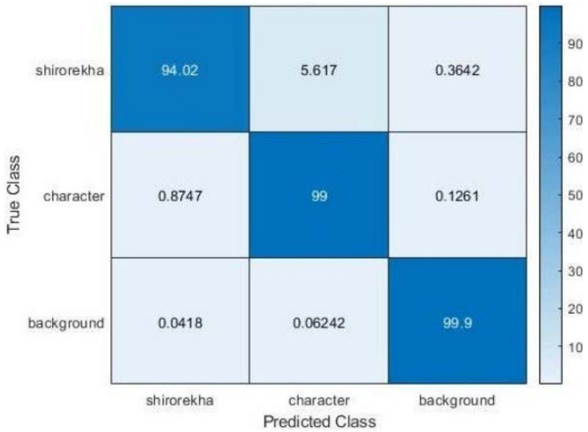

Fig.6 Normalized Confusion matrix obtained in a gray-scale for a 60:20:20 ratio.

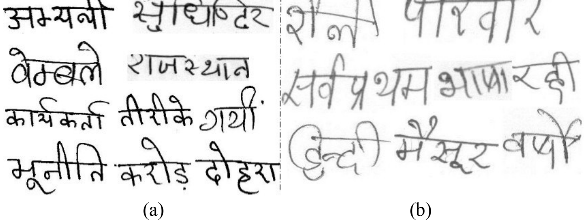

Fig.7 Word samples (a) samples from holistic-dB (b) samples from script-dB.

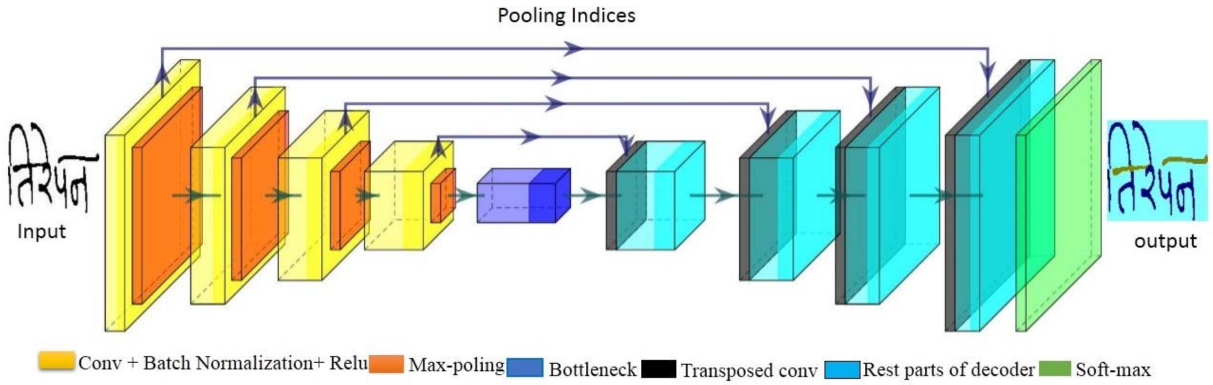

Fig.8 Pictorial representation of the *ShirorekhaNet*

In contrast to our created dataset (see section IIIA), which is based on large corpus, however, for current experimentation we have used limited datasets. This reflects the nature of real-world scenarios/ problem domains. In this situation, transfer learning [24], a paradigm in a machine learning/deep learning that allow us to use existing models in closely related problem domains. That is, instead training/learning from scratch previously learned patterns (pre-trained models) are used to solve a new problem. Since, the overlap of the problem domain is very high (with respect to problem type and number of classes) and dataset is limited, we used a pre-trained model as feature-extractor i.e. we used both the datasets (script-dB and holistic-dB) as a test sets. Table 6 shows the results obtained on Script-dB and holistic-dB, respectively. For comparison, Table 7 shows the results obtained with the Seg1 and Seg2 on the same test sets used in Table 6. It can be observed from Tables 6 and 7 and Figs 10-11 that proposed *ShirorekhaNet* (Fig.8) can be adapted to various types of documents.

Table 6: mIoU with the transfer learning

Dataset	Method	Ratio	mIoU	
			Binary	Gray
Script-dB	*Transfer learning*	50:25:25	94.2	94.1
		60:20:20	94.1	95.3
Holistic-dB		50:25:25	95.4	95.6
		60:20:20	95.7	95.7

Table 7: mIoU with the Segmentation based approaches

Dataset	Method	Ratio	mIoU	
			Binary	Gray
Script-dB	Seg1	50:25:25	82.4	82.3
holistic-dB		60:20:20	82.1	82.4
Script-dB	Seg2	50:25:25	71.5	72.3
holistic-dB		60:20:20	70.9	72.8

IV. Conclusion and Future work

In this paper, an encoder-decoder based convolutional neural network (EDCNN) is proposed for identification and removal of shirorekha from handwritten Devanagari word documents. In contrast to, existing/traditional image processing techniques, we demonstrated that the proposed EDCNN besides being accurate, is generalizable and adaptable. The efficacy of the EDCNN is corroborated on three state-of-the-art datasets. And, the experiments reveal that the essential character information is not removed/segmented irrespective of the type of a document. This reflects the accuracy, generalizability and adaptability of the EDCNN. Nevertheless, a major bottleneck in these approaches when compared with traditional approaches is the availability of an appropriate pixel-labeled dataset and large data for training. However, we are interested to employ one short learning in order to identify and remove shirorekha with a limited number of labelled word samples.

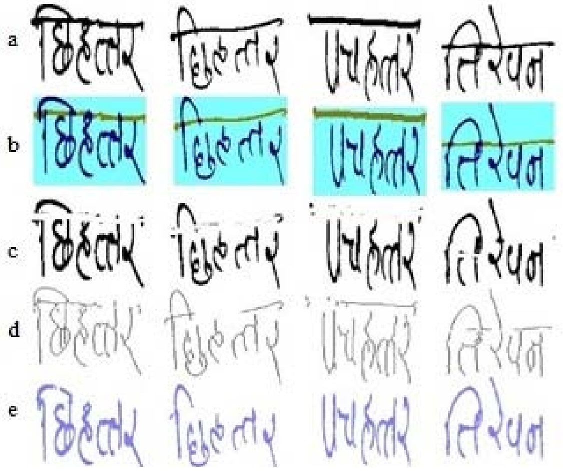

Fig.9 An illustration for segmentation achieved: (a) original image (b) ground truth (c) segmentation achieved with Seg1 (d) segmentation achieved with Seg2, and (d) Segmentation achieved with *ShirorekhaNet.*

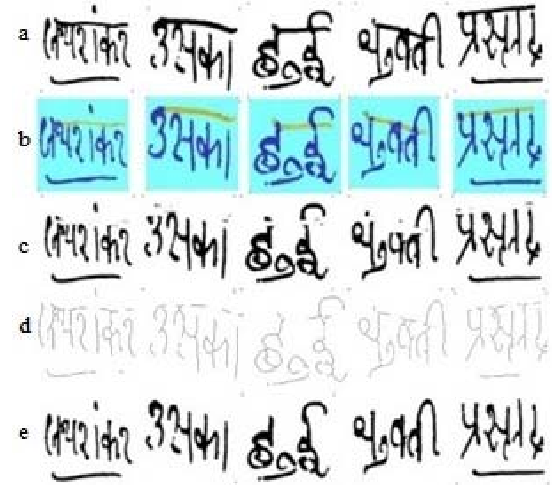

Fig.10 An illustration of segmentation achieved on script-dB (a) original images (b) ground truth (c) segmentation achieved with Seg1(d) segmentation achieved with Seg2 (e) Segmentation achieved with *ShirorekhaNet.*

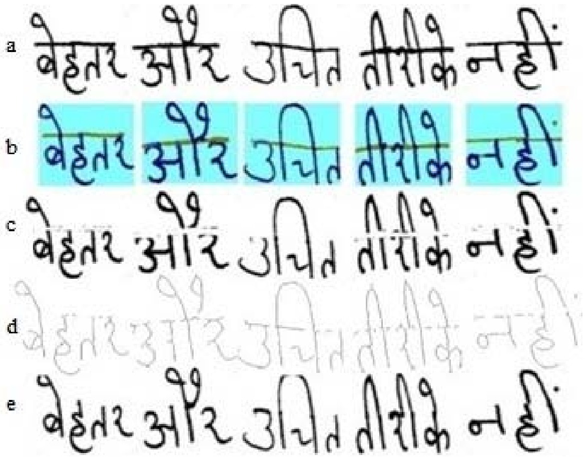

Fig.11 An illustration of segmentation achieved on holistic-dB (a) original images (b) ground truth (c) segmentation achieved with Seg1 (d) segmentation achieved with Seg2 (e) Segmentation achieved with *ShirorekhaNet.*

Acknowledgement

We acknowledge the support of the Department of Studies in Computer Science, High Performance Computing Lab (HPC-UPE) at the University of Mysore, Mysore-570006, Karnataka-INDIA.

References

[1] R. Jayadevan, S. R. Kolhe, P. M. Patil, and U. Pal, “Offline recognition of devanagari script: a survey,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 41, pp. 782–796, 2011.

[2] B. B. Chaudhuri and U. Pal, “An ocr system to read two indian language scripts: bangla and devnagari (hindi),” *in proc. of 4th Int. Conf. Doc. Anal. Recognit* (ICDAR)., vol. 2, pp. 1011–1015, 1997.

[3] P. K. Singh, R. Sarkar, and M. Nasipuri, “Offline script identification from multilingual Indic-script documents : A state-of-the-art,” *Comput. Sci. Rev.*, pp. 1–28, 2014.

[4] S. Ukil, S. Ghosh, S. Obaidullah, and K. C. Kaushik, “Improved word-level handwritten indic script identification by integrating small convolutional neural networks,” *Neural Comput. Appl.*, vol. 32, pp. 2829–2844, 2019.

[5] M. Ravikumar, S. Manjunath, and D. S. Guru, “Analysis and automation of handwritten word level script recognition,” *Adv. Intell. Syst. Comput. Cham,* vol. 369, pp. 213–225, 2015.

[6] B. B. Chaudhuri and U. Pal, “Skew angle detection of digitized indian script documents,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 182–186, 1997.

[7] B. B. Chaudhuri and B. Sumedha, “Handwritten text line identification in indian scripts,” *in proc of 10th Int. Conf. Doc. Anal. Recognit* (ICDAR)., pp. 636–640, 2009.

[8] P. Roy, U. Pal, and J. Lladós, “Morphology based handwritten line segmentation using foreground and background information,” *in proc. of 12th Int. Conf. Front. Handwrit. Recognit* (ICFHR*)*, pp. 241–246, 2008.

[9] P. P. Roy, A. K. Bhunia, A. Das, P. Dey, and U. Pal, “HMM-based Indic handwritten word recognition using zone segmentation,” *Pattern Recognit.*, vol. 60, pp. 1057–1075, 2016.

[10] R. Sarkar, B. Sen, N. Das, and S. Basu, “Handwritten devanagari script segmentation : a non-linear fuzzy approach,” *in* p*roc.of 8th IEEE International Conf. AI Tools Eng* (ICAITE), pp. 6–8, 2015.

[11] B. Shaw, S. K. Parui, and M. Shridhar, “A segmentation based approach to offline handwritten devanagari word recognition,” *in proc.ofIEEE International Conference on Information Technology* (ICIT), pp. 256–257, 2008.

[12] M. Hanmandlu, Pooja Agrawal, and B. Lall, “Segmentation of handwritten hindi text : a structural approach,” *Int. J. Comput. Process. Lang.*, vol. 22, pp. 1–20, 2009.

[13] S. Arora, D. Bhatcharjee, M. Nasipuri, and L. Malik, “A two stage classification approach for handwritten devanagari characters,” *in proc.of IEEE Int. Conf. Comput. Intell. Multimed. Appl.* (ICCIMA) *2007*, vol. 2, pp. 399–403, 2008.

[14] D.S.Guru, M. Suhil, and M. Ravikumar, “Small eigenvalue based skew estimation of handwritten devanagari words,” *in proc. of Int. Conf. Min. Intell. Knowl. Explor. Cham* (MIKE), vol. 9468, pp. 216–225, 2015.

[15] F. Lateef and Y. Ruichek, “Survey on semantic segmentation using deep learning techniques,” *Neurocomputing*, vol. 338, pp. 321–348, 2019.

[16] X. Liu, Z. Deng, and Y. Yang, “Recent progress in semantic image segmentation,” *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 1089–1106, 2019.

[17] K. V Jobin and C. V Jawahar, “Document image segmentation using deep features,” *in proc. of Comput. Vision, Pattern Recognition, Image Process Graph.* (NCVPRIPG), vol. 841, pp. 372–382, 2017.

[18] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Trans. Syst. Man Cybern.*, vol. 9, pp. 62–66, 1979.

[19] V. Badrinarayanan, A. Kendall, R. Cipolla, and S. Member, “SegNet : A deep convolutional encoder-decoder architecture for image segmentation,” *arXiv1511.00561, 2015.*, pp. 1–14.

[20] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimisation,” *arXiv:1412.6980v9*, pp. 1–15, 2017.

[21] L. Prechelt, “Early stopping - But when?,” *Montavon G., Orr G.B., Müller KR. Neural Networks Tricks Trade. Lect. Notes Comput. Sci.* (LNCS), vol. 7700, pp. 53–67, 2012.

[22] K. He and X. Zhang, “Deep residual learning for image recognition,” *arXiv:1512.03385v1*, pp. 100–200, 2015.

[23] R. Jayadevan, S. R. Kolhe, P. M. Patil, and U. Pal, “Database development and recognition of handwritten Devanagari legal amount words,” *in proc.of 11th Int. Conf. Doc. Anal. Recognition* (ICDAR), pp. 304–308, 2011.

[24] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review,” *Neural Comput.*, vol. 29, June, pp. 2352–2449, 2017.