# Analysis of the key components of segmentation-free bilingual OCR for mobile phones

Deepak Kumar
Department of Electrical Engineering
Indian Institute of Science, Bengaluru, 560012
Email: dipkmr@gmail.com

Ramakrishnan A. G.
Department of Electrical Engineering
Indian Institute of Science, Bengaluru, 560012
Email: agr@iisc.ac.in

***Abstract*—The recognition of text from camera-captured images using mobile phones has tremendous applications. The research work to drive mobile phone-based applications is much needed and ubiquitous. One of the applications is the transliteration of text in the image from one language into another. We need to recognize the text using an OCR engine and then perform transliteration. Here, we focus on the problems encountered while developing an OCR engine to recognize bilingual (Kannada and English) text from camera-captured images. A number of components are involved in building a bilingual OCR engine. We need a large corpus of real-world images to evaluate the OCR engine on camera-captured images. We need a neural network model that can handle text in two different languages without hassle. We need the model to run on mobile phones and recognize the text in the image. In this work, we analyze the challenges involved in achieving high performance. Still, there is scope for improvement in recognizing out-of-vocabulary words, which were not part of training the model.**

***Index Terms*—Mobile phones, Word recognition, Camera-captured images, OCR, Binarization, Segmentation, CRNN, Kannada, English, Devanagari, Transliteration.**

## I. Introduction

An optical character recognition (OCR) is an engine that digitizes digital images of text into machine readable text. A lot of research has produced several OCR engines, which are easily accessible online to perform the digitization of scanned or camera-captured images. Mobile phones have made the task of digitization much easy with cloud computing. However, we have enough scope for research to explore the digitization process to occur within the mobile phones or hand-held devices. The amount of computation available in mobile phones is ever-increasing. There are several applications where machine recognition is necessary to reduce the amount of manual work like in billing and filling the content in different languages.

Here, we are exploring the applicability of OCR for transliteration from a language unknown to the user to a known language [1]. Sometimes while vacationing in a new place, we may feel lost when no information is available in our own language. Unfortunately, we may not have access to wireless communication channels. So we solely depend on the people in the near vicinity. Even then, there is a communication barrier if the local language is unknown to us. In such situations, mobile phones may come handy and help us to communicate and navigate through unseen places in the world. An example to mention here is the mobile reading a signboard in a language unknown to us and transliterating it into a language known to us. The essential component of this application would be an OCR engine running on mobile phones that can handle multiple languages. Figure 1 shows some sample images containing Kannada text that are captured from street display boards using mobile phone cameras.

The amount of text normally present in a scanned document image is more than that in a camera-captured image. Archival documents are usually preserved through the digitization process using a scanner. The text in camera-captured images is often sparsely populated and is only occasionally dense. We need to detect and locate the text in such images [2]–[4] and then perform recognition. A number of techniques have been proposed for extracting text, Tables, etc. from camera-captured images [5]–[7]. The sparsely populated text may not even have a complete paragraph or sentence. In such situations, we pass the detected text boxes as words to an OCR engine. The word or block level recognition of text is convenient. By block, we refer to a group of closely placed words that form a section distinct from the background. Many of the works reported in the literature assume that the words in the camera-captured images are from a highly limited vocabulary of words [8]–[10]. For example, Wang and Belongie [8] add the word to be recognized to a random set of 49 words and use the resulting 50-word dictionary for recognition. Obviously, such systems are not practically useful. We use a vocabulary of size over 400,000 words.

The augmented intelligence of identification of the script

978-1-6654-7350-7/22/$31.00 © 2022 IEEE

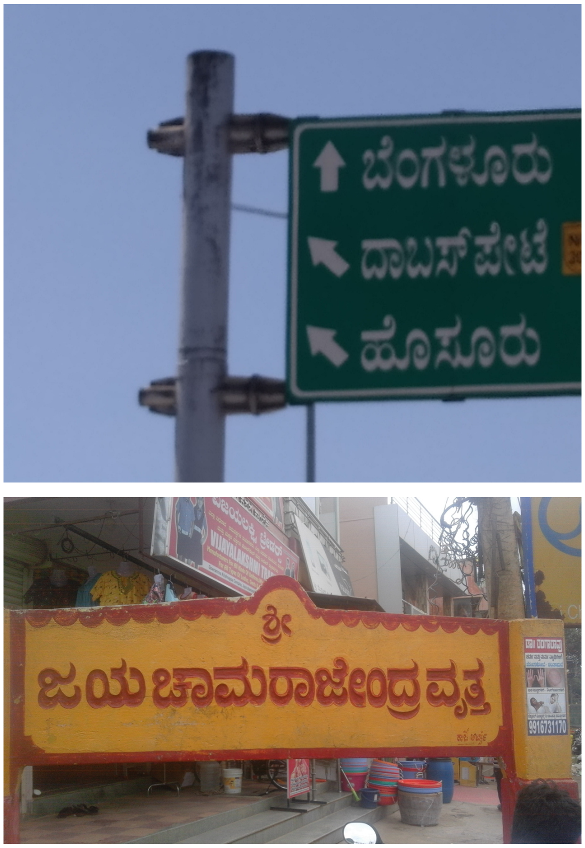


Fig. 1: Images containing text captured using mobile phone cameras.

or language of the text helps in the selection of an OCR engine [11]. We need a block of text to identify the language, which is easier with statistically derived features, but we need the respective language OCR engine to perform recognition. We may know that text is present in a different language, but we may not have used the OCR engine of that language since it may not make sense in our everyday use case. Hence we may not have kept that OCR engine on our mobile device. The missing OCR engine may prevent the recognition of text and leave us in a nowhere situation. Thus in the long run, a multilingual OCR engine is the preferable way to go with mobile phones.

Text recognition from a camera-captured image is a two-step process: text detection [12] and recognition. In this paper, we mainly focus on word recognition from a cropped image. Our contributions are as follows:

- Data augmentation of word images for training and evaluation of a neural network model.
- Comparison of the neural network models trained on gray-scale images and color images.
- Comparison of language models using different sets of vocabulary words.
- Binarization of color images before recognition.

## II. Bilingual OCR

An OCR engine is usually developed to recognize the characters in an image from a single language like English, Hindi, Kannada, or Tamil. In many State capital cities in India, multiple languages are used for communication, and information is conveyed in many languages. Sometimes, we observe that text in many languages are placed one after another in order to serve maximum audience using common languages. Even then, some languages will be missing. Thus we need a model capable of handling multiple languages to the extent possible. Mobile phones are used for different purposes and may become handy in transliterating a signboard present in an unknown language to a known language [13], [14]. The signboards may carry information in many languages, and the text is restricted to word or block levels in different languages. We may select an interesting word from the camera-captured image for the transliteration process. The selected word needs to be recognized irrespective of the script. It is a complex problem because the number of languages is high, and covering all of them is not feasible. Here, we combine English and Kannada words for the data set to train the deep learning model, and the model recognizes the word in the images from these two languages.

### A. Data generation and augmentation

A neural network model requires a large amount of training data and produces good accuracy if sufficient variations are covered in the training data. Capturing a large number of images through a mobile phone is difficult. Hence we need to rely on other data sources for training the neural network model. We have selected two prominent data sources: scanned books and synthetically created images. These sources do not replicate the exact situations that may occur in camera-captured images but provide consistent similarities.

*1) Image corpus created from printed books:* We have selected scanned images from books to train a neural network model [15]–[17]. Each scanned image is a page from a book. Each page consists of multiple text blocks. Since our model requires word-level images for training, we have annotated the images at the level of the word bounding box for each scanned page. With the help of bounding box information and the annotated word, the scanned images are cropped into individual word-level images. The number of word images generated from the book corpus is around 800,000. The images from the book corpus are gray-colored images. The images are mostly clean and may contain a little blur due to degradation of paper over time and scanning error. The neural network model trained with these gray-colored images is termed as ‘gray model’ in our further discussion.

*2) Image corpus created synthetically:* We observe that the book corpus contains long sentences, mostly in gray-color. Conversely, the camera-captured images have limited text with a few words, and are mostly colored. We generated word-level images using Image Magick [18] from Wikipedia data for Kannada [19]–[21]. A total of 400,000 images were synthesized using random colors with random change in orientation. These synthetically generated images are used to train another neural network model, which is referred to as 'color model' hereinafter.

### B. Training of the CRNN

We have trained the CRNN neural network architecture using the word images to recognize the text in those images [22], [23]. The neural network has been used to recognize scene text word images of English dataset [22], [23]. The model recognizes the characters directly from the image without segmenting individual characters in the image. This type of neural network model is helpful when the characters cannot be segmented using a threshold value in the images. The CRNN architecture consists of convolutional layers, recurrent layers, and a transcription layer as shown in Figure 2. The sizes of the input image and the kernel are tabulated in Table I. The gray model is trained on the entire book corpus of around 800,000 words for 50 epochs, and its validation accuracy is 95%. The color model is trained on the synthetic corpus of 400,000 words for 30 epochs, and its validation accuracy is 92%.

The words recognized by the CRNN model are fed to the transliteration application to generate the transliterated word in Devanagari script as shown in Figure 2. The entire operation is performed on an application developed for the purpose on a mobile phone [13]. The mobile application will soon be made available for free download from the Google play store, after the necessary checks for software reliability.

TABLE I: Configuration of CRNN model with parameters.

| Type | Configurations |
|---|---|
| Transcription Layer | - |
| Bidirectional-LSTM Layer 2 | #hidden units: 128 |
| Bidirectional-LSTM Layer 1 | #hidden units: 128 |
| Map-to-Sequence | - |
| BatchNormalization Layer 3 | - |
| Convolution Layer 7 | #maps:512, k:3x3, s:1, p:1 |
| Convolution Layer 6 | #maps:512, k:3x3, s:1, p:1 |
| MaxPooling | Window:2x2, s:2 |
| BatchNormalization Layer 2 | - |
| Convolution Layer 5 | #maps:512, k:3x3, s:1, p:1 |
| Convolution Layer 4 | #maps:256, k:3x3, s:1, p:1 |
| MaxPooling | Window:2x2, s:2 |
| BatchNormalization Layer 1 | - |
| Convolution Layer 3 | #maps:256, k:3x3, s:1, p:1 |
| Convolution Layer 2 | #maps:128, k:3x3, s:1, p:1 |
| Convolution Layer 1 | #maps:64, k:3x3, s:1, p:1 |
| Input Image | 50x300 (HxW) grayscale/color image |

## III. Evaluation of the Results and Discussion

For the purpose of extensive testing of our trained models, we have separately synthetically generated 42,233 test images in each of the three categories, namely, clean grayscale images, clean color images, and blurred grayscale images. The total number of unique words in our corpus is 4,22,330. We chose the number of test images in each category to be 10% of this number, namely 42,233. The words present in all the categories of images are the same. We fix the words in order to evaluate the effect of the characteristics of the image, namely color, contrast and blur. The test set is high in number to identify these differentiating factors. Even a one percent improvement needs an additional 400 words to be correctly recognized which may not be the case in several evaluations. Thus the high number of test

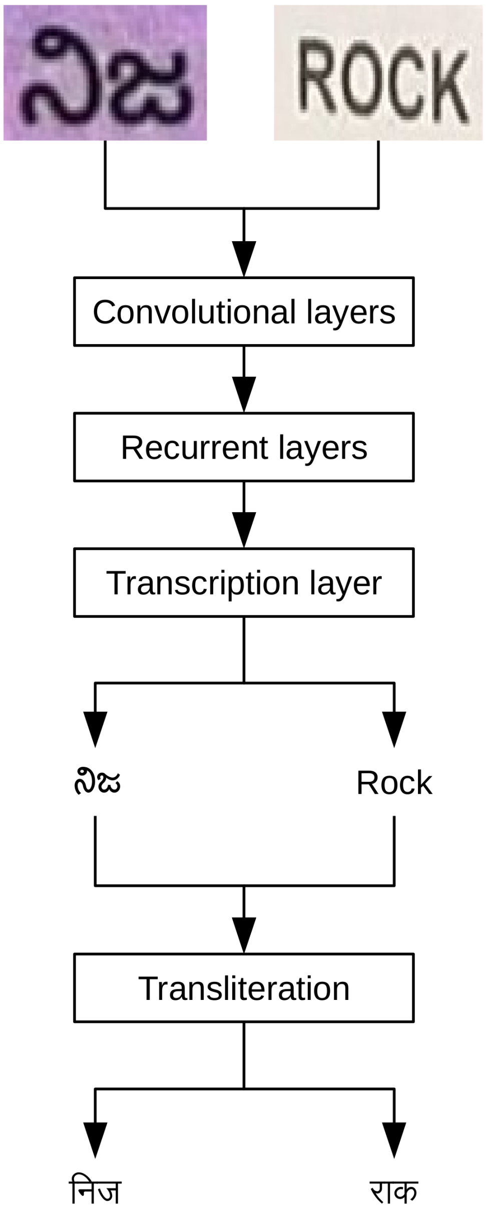
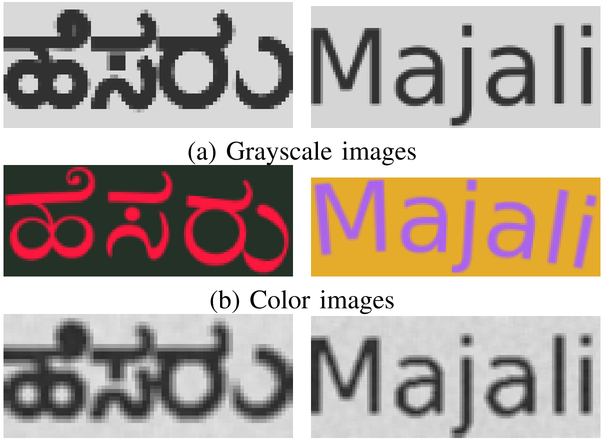


Fig. 2: The block diagram of CRNN architecture with input Kannada and English word images for recognition. The recognized words are transliterated into Devanagari script.

samples acts as a sieve to weed out the wrong strategies. The evaluation of a strategy or its variation consumes a lot of time due to the higher number of test images preventing minor improvements.



(a) Grayscale images

(b) Color images

(c) Blurred grayscale images

Fig. 3: Some of the example Kannada and English word images used to evaluate the gray and color DNN models.

### A. Results on grayscale images

The sample grayscale images generated for evaluation are shown in Figure 3 (a). The images are generated to understand whether the model performance is good when the trained model is trained on different word lists. The language model present in the neural network can capture the transitions between the characters, which is heavily dependent on the vocabulary. The accuracy of the model against edit distance is tabulated in Table II. The edit distance is a measure to count the number of deletion, insertion, and substitution of individual characters between two strings. We observe that the model accuracy is dependent on the vocabulary since the language model does not comprehend the new transitions which are out of vocabulary for the model. Thus, the model provides good accuracy only on the words present in the vocabulary.

TABLE II: The accuracies of the trained models on grayscale images, as a function of the edit distance. No. of test images: 42,233.

| Edit distance | Gray model | Color model |
|---|---|---|
| 0 | 44.52% | 91.52% |
| 1 | 60.40% | 97.61% |
| 2 | 78.64% | 99.35% |
| 3 | 86.50% | 99.69% |
| 4 | 92.74% | 99.81% |

### B. Results on color images

The sample color images generated for evaluation are shown in Figure 3 (b). The images are generated to understand the contrast between the foreground and the background. The color model receives the input image directly whereas the grayscale model receives the grayscale image converted from the color image. The accuracies of the models are tabulated in Table III. We observe that the model needs good contrast between the foreground and the background to obtain higher accuracy. However, the accuracy of the color model is far better since it is trained on color images that address the contrast between the foreground and the background of the image through convolutional filters. Do we need good contrast images? Yes, it gives a boost in the accuracy of the model. The next section discusses improving the contrast in the images.

TABLE III: Accuracies of trained models on color images, as a function of edit distance. No. of test images: 42,233.

| Edit distance | Gray model | Color model |
|---|---|---|
| 0 | 5.03% | 90.93% |
| 1 | 9.08% | 96.65% |
| 2 | 14.02% | 98.71% |
| 3 | 18.22% | 99.37% |
| 4 | 22.87% | 99.66% |

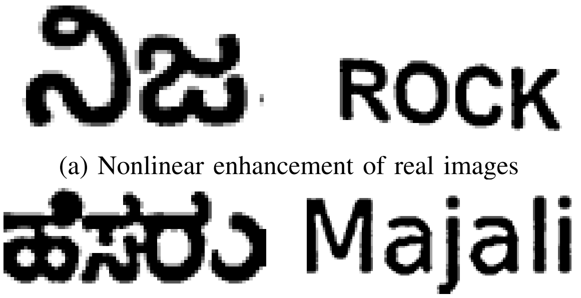
### C. Results on blurred grayscale images

Some sample blurred grayscale images are shown in Figure 3 (c). An important characteristic of any character in a language is the contrast of the edge of the character against the background. The text in the camera-captured images does not have as strong edges as that of scanned images. The edges are smoothened in the characters of the camera-captured images when zoomed to a maximum resolution as shown in the figure. The grayscale images are blurred using a 3x3 Gaussian kernel to provide smoothening effect usually observed in a camera-captured image. The accuracies of the model on blurred grayscale images are tabulated in Table IV as a function of the edit distance. We observe that the model performance is poor when the edges are blurred. It is because the convolutional kernels used in the convolutional layers look for strong edges in the input images, but the images do not have strong edges which results in poor performance.

TABLE IV: The accuracies of the trained models on blurred grayscale images, as a function of the edit distance. No. of test images: 42,233.

| Edit distance | Gray model | Color model |
|---|---|---|
| 0 | 39.48% | 24.41% |
| 1 | 59.03% | 35.17% |
| 2 | 78.10% | 45.22% |
| 3 | 87.21% | 54.19% |
| 4 | 93.40% | 62.87% |

*1) Nonlinear enhancement of blurred images:* There are two options available to us: either we can train the model with the blurred images or remove the blur from the images. We selected the second option since the models were trained on clean images. We could not train the model with the blurred images due to time constraints, since already a



(a) Nonlinear enhancement of real images

(b) Nonlinear enhancement of blurred images

Fig. 4: Nonlinear enhancement of real and blurred images using edge improvement and Sauvola threshold.

considerable amount of time was invested in training the models.

The images are upscaled by three times and the edges are enhanced nonlinearly using a 3x3 kernel with the grayscale value of the pixel as the weight of the kernel [24]–[29]. The enhanced images are segmented using Sauvola binarization to improve the contrast further in the images [30]–[33]. Then we downscale the images by three times to their original sizes. We remove the horizontal offset in the images to match the neural model requirements. The enhanced images are passed to the model for recognition. The accuracies of the two trained models are tabulated in Table V. We observe that the model requires a good contrast image to perform better in accuracy. It is evident from the recognition accuracies of the blurred and enhanced images. Figure 4 shows the improvement after nonlinear enhancement and segmentation of images. The text present in the mobile captured images are noisy and look like shot under low light conditions that may be improved with the present enhancement approach.

TABLE V: The accuracy of the trained models on blurred grayscale images after nonlinear enhancement. No. of test images: 42,233.

| Edit distance | Gray model | Color model |
|---|---|---|
| 0 | 37.34% | 69.22% |
| 1 | 56.87% | 88.21% |
| 2 | 74.19% | 94.99% |
| 3 | 83.77% | 97.59% |
| 4 | 90.67% | 98.72% |

*2) Lexicon-based correction of recognized words:* Even though the accuracy on the enhanced images is better, we performed lexicon-based correction to check for any further improvement [34]. The lexicon is a list of words created from Wikipedia data to train the model. The edit distance of the recognized word was computed with all the words in the lexicon. The edit distance is normalized by the length of the recognized word and the length of the word in the lexicon [35]. The smallest normalized edit distance is considered as the recognized word from the lexicon. The accuracies of the models after lexicon correction are tabulated in Table VI. We observe that the accuracy of the color model on the blurred images is equivalent to that on the color images. We also observe that the lexicon plays an important role in increasing the accuracy of the gray model, in spite of the fact that the word list used to train the gray model is from books that are different from the words in Wikipedia.

TABLE VI: The accuracy of the trained models on blurred grayscale images after nonlinear enhancement and lexicon-based correction. No. of test images: 42,233. Lexicon size: 4,22,330.

| Edit distance | Gray model | Color model |
|---|---|---|
| 0 | 76.16% | 93.40% |
| 1 | 80.35% | 95.41% |
| 2 | 84.65% | 97.20% |
| 3 | 89.79% | 98.17% |
| 4 | 92.45% | 98.82% |

We have tabulated the accuracies as a function of the edit distance, which is not often used for evaluation. The edit distance measure reveals the nature of the models. If we closely examine all the accuracies at the edit distance of 4, we observe the interesting fact that all the accuracies are above 90% except for two cases. In Table III, the accuracy of 22.87% achieved by the gray model is clearly due to the fact that the model is not trained on color images. Similarly, in Table IV, the accuracy of 62.87% achieved by the color model is because the model is not trained on the blurred images. The accuracy of the gray model with the edit distance of 0 in Tables IV and V are similar indicating that the nonlinear enhancement of images has not contributed to the improvement of accuracy, but rather decreased it. The improvement in accuracy for the gray model is observed with lexicon-based correction indicating that the vocabulary of the model does not contain the words from the test set. Another finding from this evaluation is that the gray model has not been trained on binarized images resulting in lower accuracy after improving the images through nonlinear enhancement. Thus, the model needs a similar environment for training and testing the images to provide higher performance.

## IV. Conclusion and Future work

In this paper, we have analyzed and identified the important components that boost the accuracy of a bilingual OCR used to recognize text from scene word images. The accuracy primarily depends on the quality of the processed images input to the network and the completeness of the lexicon or vocabulary. Hence, we need to provide better binarized images for the recognition engine. The recognition engine uses the language model to learn the transitions between the characters. When the words in the test images are present in the vocabulary, the system gives higher accuracy. However, the recognition of out-of-vocabulary words goes for a toss. Thus, there is an imminent requirement to develop systems that require segmentation and recognition of individual characters to boost the recognition confidence of out-of-vocabulary words.

Usually, an OCR is developed to cater to a particular language. Here, a bilingual OCR is developed to handle two languages. Multilingual OCRs need to be addressed in the future rather than a single language-based OCR, since multiple single language OCRs are required to perform the recognition of multi-lingual text. The identification of language or script at the word level [36] is an additional feature that may help improve the accuracy of OCR. We need to also pursue handwritten text captured using mobile phones. However, handling handwritten text is more complex than the normal printed text in a camera-captured image.

## Acknowledgment

The authors thank the Technology Development for Indian Languages, MeitY, Government of India for funding this work under the National Language Translation Mission. Project ref: 11(1)/2020- HCC(TDIL).

## References

[1] A. G. Ramakrishnan, Deepak Kumar, Maithri G. S, Sasidharan Ayyavu, Gangotri Nadiger, "KannadaPado: Mobile-based recognition and cross-lingual transcription of camera captured text in Kannada", in Proc. of International Conference on Electronics, Computing and Communication Technologies, CONECCT 2021, July 2021 DOI:10.1109/CONECCT52877.2021.9622593

[2] S Sabari Raju, Peeta Basa Pati, AG Ramakrishnan, "Text localization and extraction from complex color images," Proc. International Symposium on Visual Computing, Springer, Berlin, Heidelberg, pp. 486-493, 2005.

[3] Thotreingam Kasar, Jayant Kumar, AG Ramakrishnan, "Font and background color independent text binarization," Proc. Second international workshop on camera-based document analysis and recognition, 2007.

[4] Thotreingam Kasar, A. G. Ramakrishnan, "Multi-script and multi-oriented text localization from scene images," Proc. International Workshop on Camera-Based Document Analysis and Recognition, 2011.

[5] A. Kaur, R. Dhir, and G. S. Lehal, "A survey on camera-captured scene text detection and extraction: towards Gurmukhi script." International Journal of Multimedia Information Retrieval, vol. 6, No. 2, pp. 115-142, 2017.

[6] R Amarnath, GS Sindhushree, P Nagabhushan, M. Javed, "Automatic localization and extraction of tables from handheld mobile-camera captured handwritten document images," Journal of Intelligent & Fuzzy Systems, vol. 36, No. 3, pp. 2527-2544, 2019.

[7] K. S. Raghunandan, P. Shivakumara, S. Roy, G. H. Kumar, U. Pal, and T. Lu, "Multi-script-oriented text detection and recognition in video/scene/born digital images," IEEE Trans. cks. Sys. Video Tech., Vol. 29, No. 4, 2019.

[8] K. Wang and S. Belongie, "Word spotting in the wild," Proc. European conference on computer vision, pp. 591-604. Springer, Berlin, Heidelberg, 2010.

[9] A. Mishra, K. Alahari, C.V. Jawahar, "An MRF model for binarization of natural scene text," Proc. ICDAR, pp. 11-16, 2011.

[10] A. K. Bhunia, G. Kumar, P. P. Roy, R. Balasubramanian, U. Pal, "Text recognition in scene image and video frame using color channel selection," Multimedia tools and applications, Vol. 77, No. 7, pp. 8551-8578, 2018.

[11] AG Ramakrishnan, S Kumar Raja, HV Raghu Ram, "Neural network-based segmentation of textures using Gabor features," Proc. 12th IEEE Workshop on Neural Networks for Signal Processing, 2002, pp. 365-374.

[12] OpenCV Text Detection (EAST text detector), https://www.pyimagesearch.com/2018/08/20/opencv-text-detection-east-text-detector/ Last accessed on 21 April 2021.

[13] A. G. Ramakrishnan, Royal, Denzil Sequiera, Shashank S Rao, Shiva Kumar H. R., "Transliteration of Indic languages to Kannada with a user-friendly interface," Proc. 2015 IEEE International Advance Computing Conference (IACC).

[14] A. G. Ramakrishnan, Shashank S Rao, "Open source code for MILE-IISc Transliterator," https://github.com/MILE-IISc/MILE-Transliterator. Last accessed 21 April 2021.

[15] Shiva Kumar H. R., Ramakrishnan A. G., "Lipi Gnani - A versatile OCR for documents in any language printed in Kannada script," ACM Transactions on Asian and Low-Resource Language Information Processing, May 2020.

[16] B. Vijay Kumar, A. G. Ramakrishnan, "Radial basis function and subspace approach for printed Kannada text recognition," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. (ICASSP'04), 2004.

[17] B. Vijay Kumar, A. G. Ramakrishnan, "Machine recognition of printed Kannada text," Proc. Fifth IAPR Workshop on Document Analysis Systems (DAS-02), 2002.

[18] https://imagemagick.org/index.php

[19] https://en.wikipedia.org/

[20] https://textrecognitiondatagenerator.readthedocs.io/en/latest/index.html

[21] https://github.com/mineshmathew/IndicSceneTextRendering

[22] https://github.com/MILE-IISc/CRNN

[23] Baoguang Shi, Xiang Bai, Cong Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition", https://arxiv.org/abs/1507.05717

[24] Deepak Kumar, M. N. Anil Prasad and A. G. Ramakrishnan, "MAPS: Midline analysis and propagation of segmentation," Eighth Indian Conf. on Vision, Graphics and Image Processing (ICVGIP), December 16-19, 2012.

[25] Deepak Kumar and A. G. Ramakrishnan, "OTCYMIST: Otsu-Canny minimal spanning tree for born-digital images," Proc. 10th International Workshop on Document Analysis Systems (DAS 2012), Gold Coast, Queensland, Australia, 2012.

[26] Ram Krishna Pandey, A G Ramakrishnan, "Efficient document-image super-resolution using convolutional neural network," Sadhana, Vol. 43(15), Feb. 2018.

[27] Ram Krishna Pandey, A G Ramakrishnan, "Improving the perceptual quality of document images using deep neural network," Proc. 16th International Symposium on Neural Networks (ISNN 2019), Moscow, Russia, July 10-12, 2019.

[28] Deepak Kumar, A. G. Ramakrishnan, "Power-law transformation for enhanced recognition of born-digital word images," Proc. 2012 International Conference on Signal Processing and Communications (SPCOM), IEEE.

[29] Deepak Kumar, M. N. Anil Prasad and A. G. Ramakrishnan, "NESP: Nonlinear enhancement and selection of plane for optimal segmentation and recognition of scene word images," Document Recognition and Retrieval (DRR) XX, February 5-7, 2013.

[30] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. SMC, vol. 9, pp. 62–66, Jan. 1979.

[31] W. Niblack, "An introduction to digital image processing." New York: Prentice Hall, 1986.

[32] J. J. Sauvola and M. Pietäikinen, "Adaptive document image binarization, Pattern Recognition," vol. 33, no. 2, pp. 225–236, 2000.

[33] D. Bradley and G. Roth, "Adaptive thresholding using the integral image," Journal of Graphics Tools, vol. 12, no. 2, pp. 13–21, 2007.

[34] Suresh Sundaram, Bhargava Urala, AG Ramakrishnan, "Language models for online handwritten Tamil word recognition," Proc. Workshop on Document Analysis and Recognition (DAR 2012), pp. 42-48, Dec. 2012.

[35] Li Yujian, Liu Bo, "A normalized Levenshtein distance metric", IEEE Trans. on PAMI, vol. 29, no. 6, pp. 1091–1095, 2007.

[36] D Dhanya, A G Ramakrishnan, Peeta Basa Pati, "Script identification in printed bilingual documents," Sadhana, Vol. 27, No. 1, pp. 73-82, 2002.