Recognition of Historical Sanskrit and Kannada Manuscripts using

Convolution Neural Network

**Prabal jain(22BCB0192)** **Dr. Sridhar Raj S**

Btech CSE-Bininformatics **Assistant Professor (Senior Grade)**

**VIT, VelloreVellore, Tamil Nadu, India  VIT, VelloreVellore, Tamil Nadu, India**

[Prabal.jain2022@vitstudent.ac.in](mailto:Prabal.jain2022@vitstudent.ac.in)

Submitted: 20/9/24

Abstract: Document image analysis has emerged as a field of study with growing importance over the last few decades.

Historical documents add another challenge of physical degradation that needs to be tackled in the preprocessing. The main focus of the present work is the classification and identification of Kannada old stone inscription characters. The characters are segmented into lines, words, and characters for easier processing. The segmented images are then preprocessed in order to extract the essential features and remove the redundancies in the image. The preprocessed data is then augmented in order to compensate for the lack of datasets, and the existing dataset is trained in order to create data for the training phase. The machine learning model, Convolutional Neural Network (CNN), is selected. The classifiers based on each model are trained, and the performance of each model is evaluated. The model developed for recognizing Kannada characters achieved a validation accuracy of 92.3%. This outcome demonstrates a significant achievement in processing and digitizing ancient Kannada scripts, considering the complex nature of the language and the diverse characteristics of individual handwriting.

Sanskrit is a 3,500-year-old Indian language and the liturgical language of Hinduism, Buddhism, and Jainism. Due to resemblance in the forms of distinct letters, script complication, non-forte in the representation, and a large number of symbols, the current study on Sanskrit Character Recognition from images of text documents is one of the most challenging. The Sanskrit language is written in the Devanagari script. There are a variety of approaches for recognizing characters in a scanned image [1,2,3,4,5].

This research provides an optical character recognition (OCR) system that enables to analyse the word recognition and translate various types of Sanskrit documents or images into text using deep learning architectures which include Recurrent Neural Networks (RNN), Convolutional Neural Network (CNN)

*Keywords: Historical Kannada documents, : Historical Sanskrit , documents Image recognition, Document analysis, Cultural heritage preservation Convolutional Neural Network*

1. Introduction

The language used in Karnataka is Kannada, a Dravidian language. Thirteen vowels, thirty-four consonants, and two letters that are neither consonants nor vowels make up the forty-nine phonemic letters of Kannada, which are written in a script that is not Latin [1]. Since there are more characters and more character repetitions in Kannada, character identification is a more difficult challenge. Preservation of rich linguistic and cultural history is very much the need of the hour in present times, where rich legacies are gradually being forgotten. Most of this is handwritten or inscribed on stones whose remnants can still be found in archaeological places like Hampi and Halebeedu [2]. In this digital age where tools and means are available to identify and digitally preserve records for posterity, it is required to use the tools to preserve rich cultural legacies.The objective of the work is to devise a system to read and process Kannada script, digitize the information, and store it electronically. Datasets in the form of leaf manuscripts and stone inscriptions are collected from the Kuvempu Institute of Kannada Studies, Manasagangotri,

Mysore, and Hampi

(UNESCO World Heritage Site), Vijayanagara. Various preprocessing steps, namely gray scaling, image binarization, edge detection, cropping, etc., are applied to the collected datasets to remove redundancy in the information contained and make processing faster. The preprocessed images are fed to the optical character recognition system, which uses CNN model algorithms to recognize individual Kannada characters. Comparisons are drawn between images with different noise levels. Finally, images are analyzed, and a conclusion about the model is drawn [3].

THE Historical language of North India. Most of the greatest literature works to come out of India were written in Sanskrit. It is considered the mother tongue of all contemporary languages. The Vedas, which were written in sanskrit, represent the spirit of Indian culture and history. The core thoughts of Buddhism were also written in sanskrit.

In many academic organizations, the existence of historical scientific and mathematical research work published in Sanskrit.

Scientists from all across the world are working on decoding ancient documents. However, one major stumbling block is the lack of adequately scanned and classified Sanskrit Texts. Furthermore, the issue is aggravated by poor maintenance and text quality. As a result, digitising historical documents that are not only significant for study but also represent an important part of India's culture and history. With the emergence of digital content, the need for the development of an Optical Character

Recognition (OCR) System with high performance has become essential. OCR research is a field of Pattern Recognition. Its process of human reading is the driving force behind the development of a machine that can read characters as well as humans.

2. Related Works

In [1], recognition of handwritten characters was focused, and the captured characters are difficult to recognize due to the dissimilar script styles and shape of character.In order to solve this issue, initial training data was performed, which was later supplemented by validation and testing data. In [2], the database was created, and it contains 100,000 words from 600 users compiled. The characteristics of the scripts, as well as the number of symbols, effectively trained the data for recognition. All of the words are included in the database list. In [3], the author proposes a three-stage character segmentation method for separating Kannada characters.

The three-line character segmentation includes separating each into three segments: the top zone, base and compound characters make up the middle zone, and consonant make up the bottom zone. In [4], author demonstrates the implementation of a feature extraction method based on zone metrics. The image is split up into n equal zones, and the character centroid is computed.

The survey paper [5] expresses the tasks associated with the recognition system, including binarization, enhancement, and nternational Journal of Intelligent Systems and Applications in Engineering

layout analysis. It compares existing architectures like U-Nets and encoder-decoder networks and looks into the different historical datasets available, such as Bentham, Washington, and Ratskaporolle used to solve problems with imperfect backgrounds, stains, and uneven illumination. In [6], various binarization methods such as global fixed threshold, Otsu threshold and Markov models are implemented on a set of images with printed text. The results show that the Otsu threshold with little to no denoising performs the best. [7] presents page layout analysis consisting of classifying blocks and subsequently lines that can be performed using ANNs. Block segmentation extracts text regions in the reading order using U Net, while line segmentation involves segmenting text into smaller lines using tools like ARU Net and Kraken. Synthetic data is also made use of here, and test cases included completely randomly generated text, historical words in German, and modern German words with far lower accuracies of about 50%. In [8] presents a novel methodology for segmenting a document page into words. The proposed segmentation is based on evaluate blobs, i.e., the connection of individual letters or characters as a single entity based on a Laplacian on a Gaussian operator. The results show a promising detection of 99.12% of words with 87% accuracy.

The literature reveals substantial work on the preprocessing of ancient scripts, such as noise removal, thinning, binarization, and segmentation. The Devanagri, Tamil, and Kannada stone inscription characters are recognized using Support Vector Machines(SVM) and Neural networks algorithms. But recognition of ancient Kannada words in inscriptions is still an open challenge because some characters used in inscriptions have been partly or completely erased due to weather conditions or war, etc.

In 1973, segmentation-based algorithms were used to recognise Devanagari characters in printed texts for the first time. Many more studies have been conducted on the recognition of manuscripts in various Indian languages; most classical OCRs employ character segmentation to extract symbols and then recognise them using a classifier.

A neural network-based pattern recognition has attracted a lot of attention. The advancement of technology in terms of scanning devices, computation power and new learning strategies, Large neural networks can be developed to better and better approximate a good mapping. Unquestionably the invention of deep learning

(particularly convolutional neural networks) is a paradigm that has already had a significant impact [26, 27, 28].

In current research, machine learning methods such as support vector machines (SVMs) and artificial neural networks (ANNs) [29, 31] are used to categorise characters in images in an OCR system for Indian languages. Existing Indic OCRs perform poorly on deteriorated or badly kept documents or materials, and their digitization capabilities are confined to high-quality text documents [41].

Using LSTM Neural Networks, Kundaikar T., et al. [40] worked on Multi-font Devanagari Text Recognition. In this paper, they used OCR on a scanned document with text in Devanagari script in multiple font styles, notably Nakula, Baloo, Dekko, Biryani, and Aparajita. In 2015, research adopted a segmentation-free strategy for word classification in printed Devanagari documents [39].

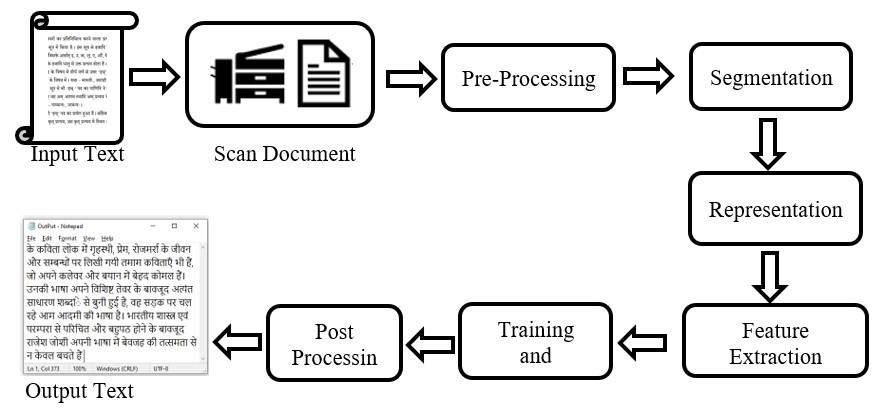
Rajib Ghosh et al. proposed employing horizontal zoning for RNN-based online handwritten word recognition in Devanagari and Bengali scripts. They employed two recently developed Recurrent Neural Network (RNN) models, LSTM and BLSTM, to recognize online handwritten cursive and non-cursive words in Devanagari and

Bengali scripts. [32]

We have effectively implemented Optical Character recognition using recurrent neutral networks, which are a type of deep learning with long-term memory cell [26, 27, 28, 38]. Recent developments in the deep BLSTM network have increased interest in text recognition, obtained in excellent results. [35,36,37], for multiple layered networks, this approach has been success fully applied for Sanskrit Manuscripts

1. Proposed Framework

The design methodology is shown in Fig. 1 and it consists of segmentation, pre-processing, Augmentation, Feature extraction and evaluation of



* 1. Dataset Creation:

Datasets are created using sources from the Vijayanagar Empire and Mysore, Karnataka, India, using ordinary digital cameras. Around ten thousand images are captured in Hampi, Vijayanagar, India, and leaf manuscripts are collected from the Kuvempu Institute of Kannada Studies, University of Mysore, India. Hampi, the UNESCO world heritage site and the second largest city of that time, is the capital of the Vijayanagara Empire. Period: 14th–16th century The stone inscriptions that they carved are relevant to the culture of that time. Stone inscriptions are a source of important historic information, such as about kings, kingdoms, and governance during that period of time. The collected image datasets are then preprocessed using image preprocessing techniques such as gray scaling, binarization, edge detection, skew correction, etc. The output of this stage passes to the segmentation stage, where character segmentation is performed and bounding boxes are obtained for each character. The characters in the image are resized to a set uniform pixel area and Images are labeled and split for training and testing set, and given to the CNN recognition model.

* 1. Preprocessing 3.2.1 Gray Scaling

The RGB of an image will be converted to a grayscale image using average method or weighted method.

The weighted method, also called the luminosity method, the weighs of red, green, and blue according to their wavelengths, as shown in equation 1.

Grayscale = 0.299R + 0.587G +

0.114B (1)

In the proposed method the standard NTSC conversion formula is used for grayscale conversion.

3.2.2Image Binarization Word recognition tasks do not require gradients or color

information. A number of binarization techniques were implemented and analyzed in order to select the ideal technique. The following methods are analyzed:

A. Isodata thresholding: randomly considers a pixel threshold “t” within the range of available intensity values in a grayscale image. It determines two means of either class higher or lower than the pixel threshold ”t” given by mL and mH. The threshold is then updated as the average of mL and mH. This is depicted in Equation . Grayscale = 0.299R + 0.587G +

0.114B (1)

This process is iteratively continued until convergence upon a single threshold, which is then utilized as the image threshold.

B.Mean binarization

This thresholding method utilizes the average of all the grayscale pixel intensity values as its threshold and the image based on its pixel intensity values is likely to be skewed, as shown in equation 3.

t =⅀I(x, y))/n (3)

where I(x, y) represents the pixel intensity value for any pixel at location (x, y), and n represents the number of pixels in the

region.

C.Sauvola Thresholding

The Sauvola method is a local thresholding algorithm that takes into consideration the local mean and standard deviation. It is a good algorithm to use for slightly more complex images. A local threshold is calculated by using a sliding rectangular field of consideration across the image, as shown in equation 4.

t = m(x, y) + k n(x,

y) (4)

Equation 4 depicts this. Here, m(x, y) represents the mean of the local window, n(x, y) represents the variance, and k is a constant set at -0.2.

D. Binarization: Otsu Thresholding

This algorithm minimizes the difference between classes by finding the threshold or weighted average of the difference between two classes (background and foreground). The grayscale range is 0-255 (0or 1 for floating point values). According to Equation 5, all pixels with a value less than 100 will be the background of the image, and all pixels with a value greater than or equal to 100 will be the foreground of the image. This happens if you set it to 100.

σ2W = Wbσ2b + Wfσ2f (5)

Evaluation Metric: PSNR

Peak signal-to-noise ratio (PSNR) is an expression for the ratio between the signal and noise. There is a high positive correlation between PSNR and visual quality, but it cannot be a sole indicator. PSNR is most easily defined via the mean squared error (MSE), as shown in equation 6.

PSNR= (2N-

1)2/MSE

(6)

G. Evaluation Metric: SSIM

The measure of similarity between two images is called Structural Similarity Information Measure (SSIM). Unlike PSNR, which uses the concept of absolute error, SSIM includes the concept of proximity. Brightness time, contrast time, and sampling time are three terms calculated to determine the quality index of the SSIM Index. Equation 7 shows that all exponents are combinations of three terms.

SSIM(x, y) = [l(x, y)] α[c(x, y)] β[s(x, y)] γ

(7)

Where x and y are the images,

The System Similarity Index (SSIM) is a method that calculates how similar two images are to each other. SSIM values range from 0 to 1; where 1 means the values for the well-developed method are generally 0.97, 0.98, and 0.99. Table 1 explains that the PSNR of the Otsu method is comparable to and better than the Sauvola method. The final processing was done using reconstructed image is exactly the original image. SSIM

Table 1: Performance Metric Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| Metric/Method  PSNR  SSIM | Isodata  12.7876  0.7955 0.788 | Otsu  12.4864 | Sauvola  10.5448  0.7300 |

the Otsu method.

3.2.3 Edge Detection and Cropping

Edge detection is an image-processing technique that is used to identify the boundaries (edges) of objects or regions within an image. The resultant image after binarization is considered input, and the sum of all pixels in each row of the image is determined using equation 8. Compete the mean of the sums recorded using equation 9 and crop out the regions that fall below the mean.

𝑐𝑜𝑙𝑠𝑗 𝑎𝑖𝑗

𝑎𝑣𝑔   (8)

𝑛𝑜.𝑜𝑓 𝑟𝑜𝑤𝑠

𝑡𝑒𝑥𝑡𝑎𝑟𝑒𝑎𝑐𝑜𝑙𝑠𝑗 𝑎𝑖𝑗 − 𝑎𝑣𝑔 (9)

3.2.4 Denoising

The captured images contain salt and pepper noise, and to remove such noise, various standard filtering methods are analyzed and their PSNR is calculated. Spatial filtering techniques such as mean, median, and Gaussian filters are analyzed with varied kernel sizes in order to accommodate different types of images.

3.2.4.a Gaussian Smoothing:

This filter focuses on giving more weight to the central pixel, which is unlike the mean and median filters. It uses a 2-D distribution called a point spread function. The Gaussian kernel outputs a weighted average at the center, which permits gentler smoothing and the preservation of edges, as shown in Fig. 2.

3.2.5 Skew Correction

Each word will be converted to slope correction and Hough transform for slope detection. The Equation 10 shows that if the character's angle is positive, the character's slope is clockwise, otherwise the character's slope is counterclockwise. Set it to show a visual tilt angle of -1.5 degrees.

𝑝𝑖𝑥𝑒𝑙 (10)

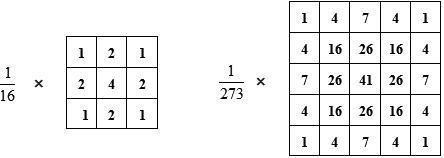


Fig 2: Gaussian filters of size 9 and 25 respectively

The angle for which f (angle) is maximum over a range of angles is the skew angle, and the image is rotated by the skew angle to make it as horizontal as possible.

3.2.6 Line Segmentation

Segmentation is nothing but breaking the whole image into subparts to process them further as shown in Fig. 3

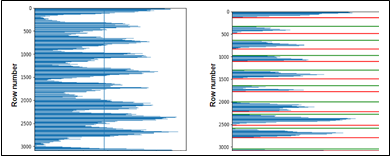


Fig. 3: Line Segmentation

The text zones are detected, and the rows of skewcorrected text are summed up after the image is negated. The negative images contained text zones showing peaks. Finally, the average is taken, and the sum is normalized to this average. Regions with high non-zero values are marked as text zones. The green line in the graph indicates the start of the text zone, and the red line indicates the end of the text zone.

3.2.7 Character Segmentation:

The function of character segmentation is to divide the image of a string of characters into smaller images representing different characters. It is one of the determining factors of the Optical Character Recognition (OCR) system. The linked object collects an image and divides its pixels into objects based on the connectivity of the pixels (e.g., all pixels in a link have a ratio of pixels using multiples and connected to each other). Once each group is defined, each pixel is labeled according to its material, which can be colored or grayscale (color labeling).The 4-way and 8-way communication is possible. When using a 4-way connection, measure the pixel's connectivity by checking the top, bottom, left and right pixel position.

3.3 Convolutional Neural Network:

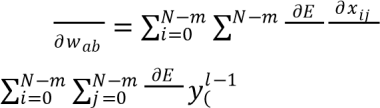
It is a type of neural network that used to process data with a known grid-like topology. The Keras is a deep learning library that simplifies the creation of CNNs. It is composed of two stacks of convolutional and maximum pooling layers. Normalization, dropout, and flattening layers are then applied to convert the feature set to a 1dimensional vector, which is then fed to fully connected dense layer.

𝑙

𝜕𝐸

𝑗=0 𝜕𝑥𝑖𝑗𝑙 𝜕𝑤𝑎𝑏 =

𝑙𝑖 (11)



𝜕𝑥𝑖𝑗

In Equation 11 error function is E, and it compute for the preceding layer with constant a ,b for each neuron output 𝜕𝜕𝐸𝑥𝑖𝑗𝑙 .

3.3 MODEL DESCRIPTION - NEURAL NETWORKS

RNN : Recurrent neural networks (RNNs) can model sequence data using their recurrent connections, which map all previous inputs to each output and construct a memory. Bidirectional RNNs help the network learn context both forward and backward.

Rectified Linear Unit

The purpose of applying the rectifier function is to increase the non-linearity in our images. The reason we want to do is that, images are naturally non-linear (do all negative values to 0 in the feature map).

When you look at any image, you'll find it contains a lot of non-linear features (e.g. the transition between pixels, the borders, the colors, etc.).

The rectifier serves to break up the linearity even further in order to make up for the linearity that we might impose an image when we put it through the convolution operation.

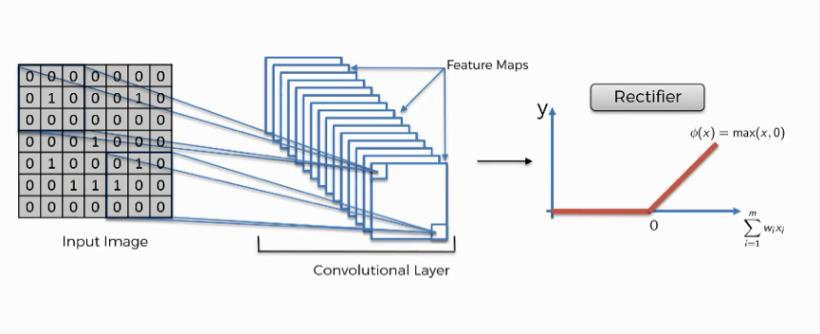
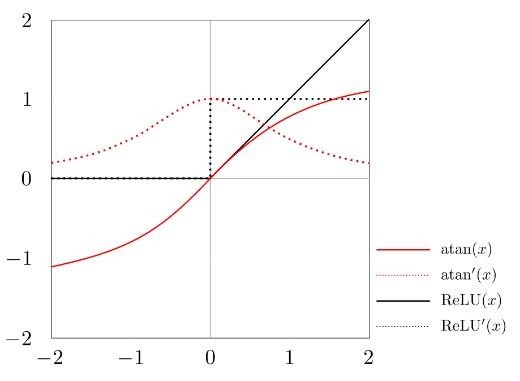


Figure 2. A plot of the atan and ReLU activation function and their derivative

As seen in Fig. 2, the Rectified linear function is essentially a hinge function that is zero for negative input values and the identical function otherwise. The function is incredibly fast to compute and has a simple derivative, with a value of 0 for negative input values and a value of 1 otherwise. Typically, the function is trimmed so that it does not exceed a significant value, such as 20. The absence of differentiability when x = 0 isn't a big deal because it may be adjusted to 0 or 1 arbitrarily.

3.3.1 Input Image:

The Red, Green, and Blue colors planes of input stone inscription images are separated. The ConvNet’s condense the image into a simpler format without loosing essential information using equation 12.

Image Dimensions = 5 (Height) x 5 (Breadth) x 1

(Number of channels, egg. RGB) (12) 3.3.2 Convolution Layer: The Kernel

In the first part of the convolution process, the elements that complete the convolution process are called kernels or filters. Select matrix K, 3x3x1. The kernel moves nine times with step length = 1 (no step) and does the matrix multiplication of K and the viewport P in which it is currently hovering

3.3.3 Padding:

The function of the convolution process is to extract high-level features such as edges from the input image. The first ConvLayer usually saves low-level properties such as edges, colors, and gradient directions. Adding additional layers also allows the architecture to evolve to a higher level, providing us with a network that understands all the images in the dataset.

This process leads to two types of results: one is a situation in which the dimensionality of convolutional features is smaller

than the dimensionality of the input, and the other is a situation in which the dimensionality of convolutional features increases or remains unchanged. This can be done with appropriate padding. After scaling a 5x5x1 image to a 6x6x1 image and applying a 3x3x1 kernel to it, a convolution matrix of size 5x5x1 is obtained. That's why Same Padding was created. On the other hand, if the same method is done without padding, it produces a matrix with the exact dimensions of the kernel (3x3x1).

3.3.4 Normalization Layer:

If the pixel value is negative, make it zero. Otherwise, keep the same value. It should be applied after every convolution layer.

Applying ReLU doesn’t change the dimension.

3.3.5 Pooling Layer:

The pooling layer, like the convolutional layer, is in charge of shrinking the convolutional feature’s spatial size.

Through dimensionality reduction, the amount of computing power needed to process the data will be reduced. Additionally, it helps to extract dominating characteristics that are rotational and positional invariant, retaining the effectiveness of the model training process. Max pooling and average pooling are the two different types of pooling. The maximum value from the area of the image that the kernel has covered is returned by Max on the basis of the increased complexity of capturing low-level details even further, but at the cost of more computational power.

3.3.6 Classification:

Fully Connected Layer (FC Layer)

Implementing a fully connected layer is a generally inexpensive way to learn non-linear combinations of the high-level properties represented by the convolutional layer's output. The fully connected layer is now learning a function in that region that might not be linear. To be used with multi-level perception, the input image must now be flattened into a column vector. The flattened output is fed into a feed-forward neural network, and each training iteration uses back propagation. The model can detect dominant and particular low-level features to classify images using the Softmax Classification approach across multiple epochs.

1. Results & Discussions

4.1 Dataset Collection:

The se was created for Hoysala, Mysore, Belur, and Halibedu places in Karnataka, India as shown in Fig. 4. The 5000 images are collected, and for training, around 2500 images are stored in a database sample.

The result of binarization of the Hoyasala stone Inscriptions image is shown in Fig. 7. A 256-level grayscale binary image is converted to a black and white image. A threshold must be selected for the grayscale image before starting the binarization process. The OTSU binarization method, which results in thresholding, is used in the planning process. If the pixel is higher tan the threshold, it is considered white (255); If it is lower than the threshold, the pixel is considered black

MODAL CODE=

# -\*- coding: utf-8 -\*-

from \_\_future\_\_ import absolute\_import from \_\_future\_\_ import print\_function import random import numpy as np from keras.models import Sequential, Model from keras.layers import Dense, Input, Lambda from keras.optimizers import SGD from keras.layers.convolutional import MaxPooling2D,Conv2D from keras.layers.core import Flatten from keras.layers import Dropout from keras.models import load\_model from keras import backend as K from keras.callbacks import ModelCheckpoint,EarlyStopping from sklearn.metrics import accuracy\_score as accuracy import os



Fig. 7: Image Binarization of Kannada stone inscription

. 4 Smoothening of Hoyasala Stone 8. This removes unnecessary specs and noise in the Inscriptions image and allows for better character segmentation

The smoothening process is done using median filter that averages the pixel value with its neighbours as shown in Fig.



Fig. 8: Smoothening of Kannada Stone Inscriptions Images

Character Segmentation of Hoyasala Connected Components algorithm used is successful in

Stone Inscriptions character segmentation of Hoyasala Stone

Inscriptions

The Fig.9 represents the results obtained after character segmentation of Hoyasala

Inscriptions.



Fig. 9: Character Segmentation

Recognition of Hoysala Stone inscription: The Hoysala Stone Inscriptions are recognized and it is shown in the

Fig. 10

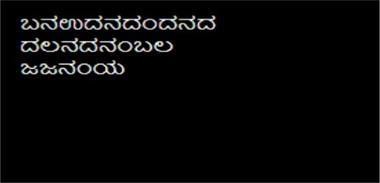


Fig. 10: Result obtained from character segmentation of Kannada stone inscription

A graph depicting model accuracy versus epochs is illustrated in Fig. 11. It demonstrates the relationship between the number of training epochs and the recognition model’s accuracy. The y-axis shows the model’s accuracy on a particular task or dataset, and the x-axis shows the epochs, which are training process iterations.

As the number of epochs rises, so does the model’s accuracy,making it possible for it to recognize characters more precisely. A graph illustrating model loss versus epochs in Fig. 12 demonstrates the relation between the number of training epochs and the loss of the recognition model. The training process's iterations are represented by the epochs on the x-axis, and the model’s loss is shown on the y-axis. It can be seen that as the number of epochs rises, the loss likewise falls, which lowers the frequency of incorrect predictions.

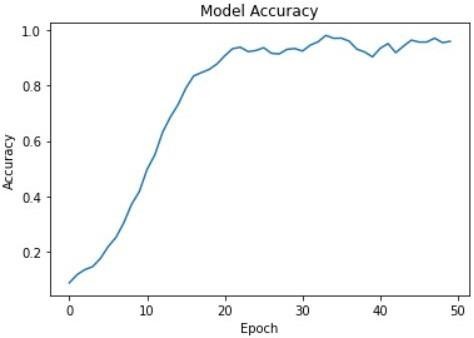


Fig. 11: Epochs v/s Accuracy

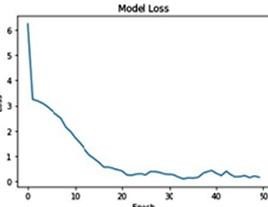


Fig.12: Epochs v/s Loss

4.7. Comparsion Results: 82.8%, respectively, displaying good performance but

slightly lower than the palm leaf. The stone inscription

The recognition rates for different types of inscriptions with high noise obtained a recognition rate of 50.5%, with varying levels of noise are displayed in Table 2. implying significant challenges in accurately processing The palm leaf inscription with low noise achieved a heavily noisy inscriptions. Lastly, the stone inscription

recognition rate of 90.9%, demonstrating an exceptional with moderate noise achieved a recognition rate of

75%,

ability to recognize characters. Stone inscriptions 1 and 2 indicating reasonable performance in handling

with low noise attained recognition rates of 88.4% and moderately noisy data.

Table 2: Comparison of Results

Epigraph Recognition rate (in %)

Palm Leaf inscription - Low Noise 90.9

Stone inscription 1 - Low Noise 88.4

Stone inscription 2 - Low Noise 82.8

Stone inscription - High Noise 50.5

Stone inscription - Moderate Noise 75

The recognition accuracy attained on degraded existing approaches. The projected experimental results documents is shown in Table 3. The Table 3 contains the in Table 3 make it evident that the proposed method has results of the comparison between the proposed and a greater level of recognition accuracy than the other methods already in use

Table 3: Comparative analysis of degraded printed Kannada character recognition

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No. | Techniques | Total datasets experimented | Accuracy (in%) |
| 1 | Neural Network[27] | 2450 broken character dataset synthetically generated | 98.9 |
| 2 | FDA(Fit Discriminant Analysis)[28] | 250 real datasets from historical, 150x49=7350 Synthetically generated dataset | 99.38 |
| 3 | FLD(Fisher Linear Discriminant Analysis)[29] | 21560 both clear and degraded characters(Kannada and English)  [Kailasam, Kasturi,Times new  Roman,Arial | 98.2 |
| 4 | End point algorithm[30] | 100 degraded Kannada characters | 89 |
| 5 | Convolutional Neural Network (Proposed Method). | 16100 degraded Kannada characters extracted from old documents | 95.9 |

5. Conclusion

The Optical Character Recognition (OCR) system uses CNN model algorithms for recognizing historical Kannada characters. The system reduces the amount of information contained in the original dataset during the preprocessing step by using algorithms like OTSU thresholding and Gaussian thresholding. It is observed that though there is considerable skew angle present in captured images, the system is capable of not only calculating the skew angle but also correcting it. This helps in a considerable reduction in human effort and makes processing faster. The average accuracy of the OCR system is 95.9%. The OCR system is found to be best suited for leaf manuscript databases using the CNN model. The system could be modified in the future by using more advanced algorithms that can translate historical Kannada words into present-day Kannada words, which could lead to an easier understanding of historical Kannada documents.

4.2 PROPOSED MODEL

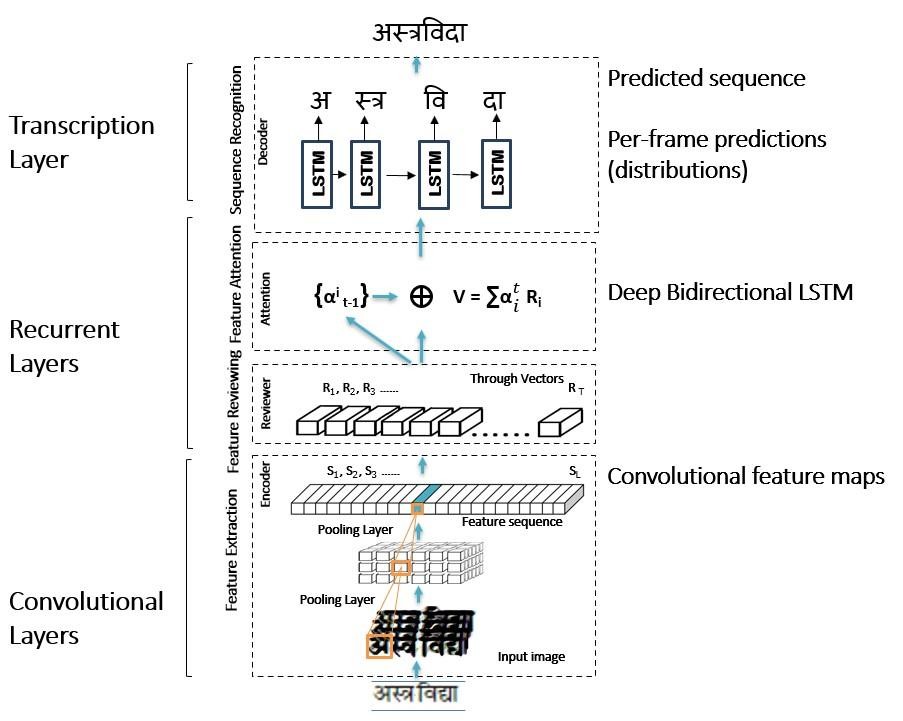


Figure 5 : The network architecture.

From bottom to top, the Proposed network architecture comprises of three components: convolutional layers, recurrent layers, and a transcription layer, as shown in Fig. 5.

Convolutional layers, from the input picture, which extracts a feature sequence;

Recurrent layers, which forecasts each frame's label distribution;

Transcription layer, which converts per-frame predictions into a final label sequence

1. Feature Sequence Extraction

The convolutional layer component of the model is built by combining the convolutional and maxpooling layers from a typical CNN model (fully-connected layers are removed). A sequential feature representation is extracted from an input picture using this component. All photos must be resized to the same height before being supplied into the network. The input for the recurrent layers is therefore a sequence of feature vectors derived from the feature maps produced by the components of convolutional layers. Each feature vector in a feature sequence is formed on the feature maps by column, from left to right. This indicates that the i-th feature vector is the concatenation of all the maps' i-th columns. In our settings, the width of each column is set to a single pixel.

1. Sequence Labeling

As the recurrent layers, a deep bidirectional Recurrent Neural Network is formed on top of the convolutional layers. For each frame xt in the feature sequence S = S1, S2. S3.., SL, the recurrent layers predict a label distribution yt [Fig 4,5]. The recurrent layers have three distinct benefits. To begin with, RNN excels at capturing contextual information inside a sequence. For image-based sequence recognition, using contextual clues is more reliable and useful than considering each sign separately. In the case of scene text recognition, broad characters may necessitate numerous frames to completely represent.

1. Transcription layer

The process of translating RNN's per-frame predictions into a label sequence is known as transcription. The goal of transcription, mathematically, is to discover the label sequence with the best likelihood based on per-frame predictions.

1. Network Training a). SANSKRIT CHARACTERS ARE USED FOR RESEARCH WORK Devanagari Alphabet

|  |  |
| --- | --- |
| Numerals(अंक)        Variant Letters          Conjunct Consonants (संयुक्तक्षर) - Ancient  Worlds | Vowels (स्वर) |

Figure 6. The Language written with the Devanagari alphabet

As show in figure 6 Sanskrit text contains of several compound characters which are formed by different combinations of half letter and full letter consonants. Some examples of compound characters are shown in Fig 6. Since such compound characters are either less frequent or completely absent in Hindi text, Hindi OCRs would not be trained to segment and classify such characters correctly. Subsequently, the Hindi OCRs would display poor results in Sanskrit text.

b). DATASETS

For training and benchmarking, many character recognition algorithms require a large amount of ground-truthed real-world data. A trainable OCR model's generalisation accuracy is directly affected by the quantity and quality of training data.

In our proposed datasets we have taken data from two sources

Existing Sanskrit document image texts and synthetically generated Sanskrit texts

Scan image Characters - Annotated 19,579 lines of Sanskrit text from four different books. Books महात्मा गाांधी का आर्थिक एवां सामार्र् क दर्र् न- Economic and Social Philosophy of Mahatma Gandhi Bhrigu Sanghita By Maharshi Bhrigu श्रीमदभगवद्गीता - Bhagavad Gita - The Song of God By Swami Mukundananda मनमनाभव (श्रीमद्भागवतगीता पर आधाररत)- Manmanabhav (Based on Shrimad Bhagwat Geeta)

To ensure high data quality, the notations were acquired from Sanskrit domain specialists. The relevant statistics are listed in Table 2.

Table 2. Statistics of our annotated datasets

|  |  |  |  |
| --- | --- | --- | --- |
| Book | Pages | Lines | Words |
| Economic and Social Philosophy | 152 | 5490 | 38320 |
| Bhrigu Sanghita | 130 | 4696 | 32774 |
| Bhagavad Gita | 110 | 3973 | 27731 |
| Manmanabhav | 150 | 5418 | 37816 |
| Total | 542 | 19579 | 136642 |

Sanskrit writings synthesised - typefaces from many sources [22,23,24,25]. The list was then reduced to 67 fonts that allow conjunct character rendering. We used 4500 different lines per typeface from old Sanskrit literature. Available at https://sanskritdocuments.org to increase the diversity of synthetic data. [Section A Font Styles ]

c). DATA PREPARATION

For our synthetic data, we employ binary images and trim the output text to remove superfluous whi tespace around the actual text content. There are 170 characters in the vocabulary set. All of the photos have been scaled to 32 pixels in height, with the width altered to match the original aspect ratio.

d). OPTIMIZATION

For optimization, we use the ADADELTA (with decay rate of 0.95) [42] to automatically calculate per-dimension learning rates. There is no need to manually establish a learning rate with ADADELTA. More notably, we discovered that ADADELTA optimization converges faster than the momentum technique. The learning rate is initially set at 1. We train with 16-bit mini-batches and stop after 310k iterations.

Table 5. Robustness of the classifier

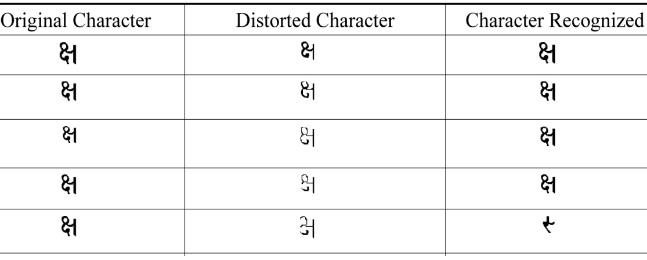
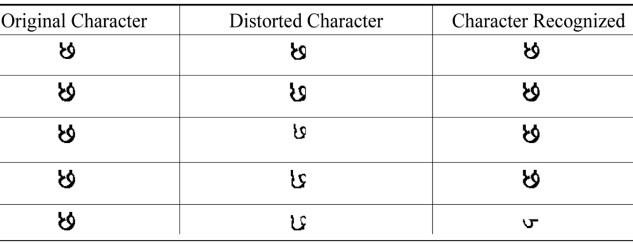
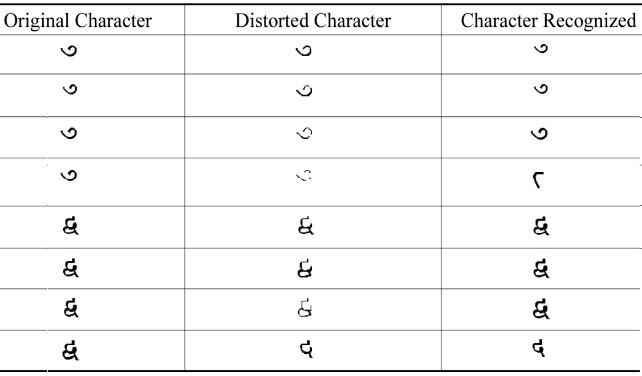
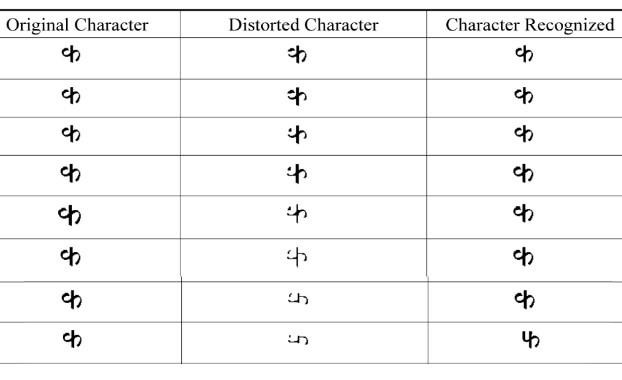


Table 6. Result comparison of OCR Systems for Printed Devanagari Sanskrit Characters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | Feature | Classifier | Data Set Size | Accuracy (%) |
| Govindraju et al. [18] | Gradient | Neural Networks | 4,506 | 84 |
| Kompalli et al. [19] | GSC | Neural Network | 32,413 | 84.77 |
| Bansal et al. [16] | Statistical Structural | Statistical  Knowledge Sources | Unspecified | 87 |
| Huanfeng Ma et al. [20] | Statistical Structural | Hausd or off image comparison | 2,727 | 88.24 |
| Natrajan et al. [25] | Derivatives | HMM | 21,727 | 88.24 |
| Bansal et al. [17] | Filters | Five filters | Unspecified | 93 |
| Dhurandhar et al. [23] | Contours | Interpolation | 546 | 93.03 |
| Kompalli et al. [22] | GSC | K-nearest neighbor | 9,297 | 95 |
| Kompalli et al. [21] | SFSA | Stochastic finite state automa- ton | 10,606 | 96 |
| Dhingra et al. [24] | Gabor | MCE | 30,000 | 98.5 |
| Bhavesh et al. | Gradient | Stochastic  Gradient Descent  (SGD) | 34,215 | 98.64 |

Table 7. Result comparison of OCR Systems for Printed Devanagari words

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | Feature | Classifier | Data Set Size | Accura cy (%) |
| Govindraju et al. [18] | Gradient | Neural Networks | 4,506 | 53 |
| Kompalli et al. [22] | GSC | K-nearest neighbor | 1,882 | 58.51 |
| Kompalli et al. [19] | GSC | Neural Network | 14,353 | 61.8 |
| Ma et al. [20] | Statistical Structural | Hausdorff image comparison | 2,727 | 66.78 |
| Kompalli et al. [21] | SFSA | Stochastic finite state automaton | 10,606 | 87 |
| Bhavesh et al. | Gradient | Stochastic Gradient Descent (SGD) | 15,455 | 97.3 |

We tested our classifier on various printed Sanskrit documents and gathered the results. In fact the testing was done at two levels: individual letter level and paragraph level. The character level testing was performed on approximately 34,215 individual characters. Paragraph level testing was performed on approx. 144 paragraphs of different Sanskrit fonts consisting of approximately 15,455 words (including devanagari characters, numeric digits, and punctuation symbols). The character level performance was excellent with a correct recognition rate of 98.6%. However, at paragraph level the performance dropped and an average accuracy of approximately 93.54% was achieved and is shown in Table 6 and Table 7 respectively.

Two such input sample paragraph used for testing the performance of the classifier is shown in Table 8. It can be observed that the recognition rate is higher for individual than for continuous characters

Table 8. Testing performance of the classifier

|  |  |
| --- | --- |
| Input Image | Output Text |
|  |  |
|  |  |

Model Comparison

Table 9. Performance of cnn’s character accuracy for text in Sanskrit Devanagari Font Style

|  |  |  |  |
| --- | --- | --- | --- |
| cnn |  | Accuracy (%) |  |
| Image 1 | Image 2 | Image 3 |
| Tesseract | 82.15 | 93.14 | 92.14 |
| Indsenz | 73.54 | 85.65 | 86.65 |
| E Aksharayan | 71.61 | 79.21 | 81.21 |
| Alex Net | 80.64 | 84.60 | 86.60 |
| Google Le Net | 81.23 | 86.54 | 87.54 |
| ResNet-50 | 86.15 | 91.24 | 89.24 |
| SVM | 79.40 | 82.40 | 85.40 |
| Bidirectional LSTM | 94.56 | 96.63 | 98.64 |

Table 10. Error analysis of OCRs for text document image in Sanskrit Devanagari Font Style

|  |  |  |  |
| --- | --- | --- | --- |
| OCR |  | Type of error |  |
| Insertion | Substitution | Deletion |
| Tesseract | 2.91 | 3.56 | 3.86 |
| Indsenz | 4.52 | 2.94 | 4.20 |
| E Aksharayan (Hindi) | 4.34 | 5.15 | 8.09 |
| Alex Net | 4.25 | 2.45 | 5.11 |
| Google Le Net | 3.13 | 4.32 | 2.60 |
| ResNet-50 | 4.66 | 3.60 | 5.24 |
| SVM | 4.16 | 5.63 | 6.25 |
| Bidirectional LSTM | 2.14 | 1.21 | 1.86 |

5. References

Rajithkumar B K et al, “[Template matching method for recognition of stone inscripted Kannada characters of different time frames based on correlation analysis”,](https://search.proquest.com/openview/0553b52b90f02d994fa3d7f2ea254a5d/1?pq-origsite=gscholar&cbl=1686344) International Journal of Electrical and Computer Engineering (IJECE) Vol. 4, No. 5, October 2014, pp. 719~729,ISSN: 20888708.

Sridevi T.N, et al. “Deep Convolution Neural Network for Degraded Printed Kannada Character Recognitions”.

Indian

Journal of Computer Science and Engineering. Volume 12 No. 3 May-Jun 2021. DOI:

10.21817/indices/2021/v12i3/211203187

Rajithkumar B K et al, “Extraction of Stone Inscripted Kannada Characters Using Sift Algorithm Based Image

Mosaic”, International Journal of Electronics & Communication Technology, Volume 5, Issue 2, April - June 2014.

Rajithkumar B K et al., “Era Identification and Recognition of Stone In-scripted Kannada

Characters Using Artificial Neural Networks”:2nd National Conference on Innovation in Computing and Communication Technology, March, 2014.

Haoming Zhang. “Ancient Stone Inscription Image Denoising and Inpainting Methods Based on Deep Neural Networks”. [Discrete Dynamics in Nature an](https://www.hindawi.com/journals/ddns/)d Societ[y Vo](https://www.hindawi.com/journals/ddns/)l. 1, 2021. DOI:

[10.1155/2021/7675611](http://dx.doi.org/10.1155/2021/7675611)

Chandrakala, H. T, “Deep Convolution Neural Networks for Recognition of Historical

Handwritten Kannada Characters”, In Frontiers in Intelligent Computing: Theory and Applications (pp. 69-77).

Springer, Singapore. 2021

Thippeswamy, G. “Recognition of Historical Handwritten Kannada Characters Using Local Binary Pattern Features”. International Journal of Natural Computing Research (IJNCR), 2020

F. Lombardi, “Deep learning for historical document analysis and recognition—a survey,” National Library of Medicine, vol. 10, 2020.

M. R. Gupta, N. P. Jacobson, and E. K. Garcia, “OCR binarization and image pre-processing for searching historical documents,” Pattern

Recognition, vol. 40, 2007.

J. Martine, L. Lenc, and P. Kr al, “Building an efficient OCR system for historical documents with little training data,” Neural Computing and Applications, vol. 32, 2020.

R. Manmatha and N. Srimal, “Scale space technique for word segmentation in handwritten documents,” in ScaleSpace Theories in Computer Vision, Springer Berlin Heidelberg, 1999. [12] G. Chen, Q. Chen, X. Zhu, and Y. Chen,

“A study of historical documents denoising,” in 2017 10th International Congress on Image and Signal Processing, Biomedical

Engineering and

Informatics (CISP-BMEI), 2017.

P. Sharan, S. Aitha, A. Kumar, A. Trivedi, A. Augustine, and S. R. K. Sarvadevabhatla, “Palmira: A deep deformable network for instance segmentation of dense and uneven layouts in handwritten manuscripts,” CoRR, 2021.

R. I. Minyue Dai Carrie Yang and M. J. Brown., “Experiments with early modern manuscripts and computer-aided transcription,” Pattern Recognition Letters, 2018.

Kshetry and R. Lal, “Image pre-processing and modified adaptive thresholding for improving OCR,” ArXiv, 2021.

M. Shen and H. Lei, “Improving OCR performance with background image elimination,” in 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2015

T. Blanke, M. Bryant, and M. Hedges, “Open source optical character recognition for historical research,” Journal of Documentation, vol. 68, 2012.

B. J. Bipin Nair, N. Shobharani, N. R. Sreekumar, and G. Ashok, “A two phase denoising approach to remove uneven illumination from ancient note book images,” in 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 1, 2021. K. Saddami, K. Munadi, Y. Away, and F. Arnia,

“Effective and fast binarization method for combined degradation on ancient documents,” Heliyon, vol. 5, 2019.

[20] C. Tensmeyer and T. Martinez, “Historical document image binarization: A review,” SN Computer Science, vol. 1, 2020.

J. A. S anchez, V. Romero, A. H. Toselli, M.

Villegas, and E. Vidal, “A set of benchmarks for handwritten text recognition on historical documents,” Pattern Recognition, vol. 94, 2019.

M. Almeida, R. Lins, R. Bernardino, D. Jesus, and

B. Lima, “A new binarization algorithm for historical documents,” Journal of Imaging, vol. 4, 2018.

W. Xiong and L. Zhou, “An enhanced binarization framework for degraded historical document images,” EURASIP Journal on Image and Video Processing, vol. 13, 2021

S. Lu and C. L. Tan, “Script and language identification in noisy and degraded document images,” IEEE transactions on pattern analysis and machine intelligence, vol. 30, 2008.

S. Vijayarani and A. Sakila, “Multi-language script identification from document images,” International Research

Journal of Modernization in Engineering Technology and Science, vol. 3, 2021 [26] Kumar, H. S et al., “ Versatile OCR for Documents in any Language Printed in Kannada Script”. ACM

Transactions on Asian and Low-Resource

Language Information Processing (TALLIP),2020. [27] Monisha, G. S. et al,: “Effective Survey on Handwriting

Character Recognition”. In Computational Method and Data Engineering. Springer, Singapore.2021

Sandhya, N., & Krishnan, R. “Broken Kannada character recognition a neural network based approach”,

International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT).2016, March: (pp. 20472050). IEEE.

Sandhya, N., Krishnan, R., Babu, D. R., & Rao, N. B.,. “An efficient approach for handling degradation in character recognition.”, International Journal of Advanced Intelligence

Paradigms,”.2019, 14(1-2), 14-29.

Aradhya, V. M., Kumar, G. H., Noushath, S., &Shivakumara, P. “Fisher linear discriminant analysis based technique useful for efficient character recognition”, Fourth International Conference on Intelligent Sensing and Information Processing,2006, (pp. 49-52). IEEE

Sandhya, N., Krishnan, R., &Babu, D. R. “A novel local enhancement technique for rebuilding Broken characters in a degraded Kannada script”. In 2015

IEEE International Advance Computing

[32]. R. Ghosh, A Recurrent Neural Network based deep learning model for offline signature verification and recognition system. Expert Systems With Applications (2020), doi: [https://doi.org/10.1016/j.eswa.2020.114249.](https://doi.org/10.1016/j.eswa.2020.114249)

[33]. A. L. Spitz. Multilingual Document Recognition. In H. Bunke and P. S. P. Wang, editors, Handbook of character Recognition and Document Image Analysis, pages 259–284. World Scientific Publishing Company, 1997.

[34]. Jurgen Schmidhuber. Deep Learning in Neural Networks: An Overview. Neural Networks, 61:85-117, 2015.

[35]. Paul, I. J. L., Sasirekha, S., Vishnu, D. R., & Surya, K. (2019). Recognition of handwritten text using long short term memory (LSTM) recurrent neural network (RNN). doi:10.1063/1.5097522

[36]. A. Dwivedi, R. Saluja and R. K. Sarvadevabhatla, "An OCR for Classical Indic Documents Containing Arbitrarily Long Words," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 2386-2393, doi:

10.1109/CVPRW50498.2020.00288.

[37]. Dutta, K., Krishnan, P., Mathew, M., & Jawahar, C.V. (2018). Offline Handwriting Recognition on Devanagari Using a New Benchmark Dataset. 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), 25-30.

[38]. Bansal, Veena & Sinha, R.M.K.. (2000). Integrating knowledge sources in Devanagari text recognition system.

Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on. 30. 500 - 505.

10.1109/3468.852443.

[39]. T. Karayil, A. Ul-Hasan and T. M. Breuel, "A segmentation-free approach for printed Devanagari script recognition," 2015 13th International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 946950, doi: 10.1109/ICDAR.2015.7333901.

[40]. T. Kundaikar and J. D. Pawar, “Multi-font Devanagari Text Recognition Using LSTM Neural

Networks,” in First International Conference on Sustainable Technologies for Computational Intelligence, 2020, pp. 495–506.

[41]. Holley, Rose. (2009). How good can it get? Analysing and improving OCR accuracy in large scale historic newspaper digitisation programs. D-Lib Magazine: The Magazine of the Digital Library Forum. 15.

[42]. Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. Retrieved from http://arxiv.org/abs/1212.5701

[43]. A. Dwivedi, R. Saluja and R. K. Sarvadevabhatla, "An OCR for Classical Indic Documents Containing Arbitrarily Long Words," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 2386-2393, doi:

10.1109/CVPRW50498.2020.00288.

[44]. V. Frinken and S. Uchida, "Deep BLSTM neural networks for unconstrained continuous handwritten text recognition," 2015 13th International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 911915, doi: 10.1109/ICDAR.2015.7333894.

[45]. Shuohao Li, Anqi Han, Xu Chen, Xiaoqing Yin, and Jun Zhang "Review network for scene text recognition," Journal of Electronic Imaging 26(5), 053023 (17 October 2017). https://doi.org/10.1117/1.JEI.26.5.053023

[46]. Mathew, M., Singh, A.K., Jawahar, C.: Multilingual OCR for indic scripts. In: 2016 12th IAPR Workshop on Document Analysis Systems (DAS), pp. 186–191 (2016)