VEDIC SANSKRIT TEXT RECOGNITION SYSTEM

Ambar Dhanave[1] Shubham Dhere[1] Mahadev Mastud[1] Rohit Karne[1] Prof. Dipali Pawar[2]

[1]*Department of Computer Engineering, ZCOER, Pune, Maharashtra, India*
[2]*Assistant Professor, Department of Computer Engineering, ZCOER, Pune, Maharashtra, India*

Abstract

Vedic Sanskrit texts are wonderful supply of information concerning Science, arithmetic, Hindu mythology, Indian civilization, and culture. It's necessary to access the texts simply, to share this data with the soil and to facilitate any analysis of those ancient texts. Throughout this study we are going to analyze the spread of ways and challenges concerning the ways for recognizing the character of the Indic and Nagari script. The popularity of Indic texts is advanced compared to different language scripts. However, many researchers have provided totally different solutions for Indic character recognition additionally, the popularity of Indic texts language texts still presents some different challenges for researchers. This analysis provides variety of solutions to the Indic and Nagari script texts to acknowledge malpractice in some ways though industrial and operational accuracy has not been accomplished. So our objective of this survey is to support Indic and Nagari character recognition to make a degreed-related style develop an academic degree of economic recognition system for writing spiritual Indic text within the future within the same approach as a scientific style to assist and generate direct access to those papers. Indic, to share this data. and earth.

Keywords: *Optical Character Recognition (OCR), Convolutional Neural Networks (CNN), Dataset, Image Pre-processing, Segmentation, Feature Extraction.*

I. Introduction

Sanskrit is gaining importance in various academic fields due to the presence of ancient scientific and mathematical research work written in this language. Scientists all over the world, are spending large amount of time trying to retrieve the important knowledge present in this ancient research manuscripts. However, the lack of accurately digitized versions of Sanskrit manuscripts is a major bottleneck. In addition to this, the poor maintenance and text quality of these manuscripts adds to the problem. Hence, it becomes very important to digitize such ancient manuscripts which are not only important for research but, are also an important part of the Indian culture and heritage. In order to simplify digitization of ancient Sanskrit material, we build an Indic Optical Character Recognition System (OCR), specifically for Sanskrit. In the recent years, several OCRs have been developed for various Indian languages such as Hindi, Devanagari, Gujarati etc. However, very little work has been done to develop good OCRs for Vedic Sanskrit. Even though both Hindi and Sanskrit are written in the Devanagari script, it is important to use a Sanskrit OCR system instead of a Hindi OCR system to digitize Sanskrit text due to the significant difference in complexity between these two languages. Sanskrit text consists of large set of compound characters which are formed by different combinations of half letter and full letter consonants. Some examples of compound characters are shown in Fig 1 and Fig 2. Since such compound characters are either less frequent or not present in Hindi language, So the Hindi OCR systems would not be trained to segment and classify such characters accurately.

Figure 1: Compound characters that are considered as unique classes

Fig.1

Figure 2: Ambiguous situation while segmenting half letters

Fig.2

II. **Literature Survey**

Anoop C. S. and A.G. Ramakrishnan [1] describes CTC-Based finish To- End ASR for the Low Resource Indo-Aryan Language with spectrograph Augmentation 2021 National Conference on Communications (NCC). In this paper author said Pre-processing they have a tendency to Use
Input as text Dataset. The Pre-process means that divide as per feature. Segmentation: they have a tendency to Use SVM formula as a result of we have a tendency to Use knowledge set as matter format. and since of SVM they have a tendency to succeed additional accuracy Feature Extraction: SVM may be a neural network formula. that extracts input as text options and another neural network classifies the Text options. A Sanskrit speech corpus with around 5.5 hours of speech data are used to train the neural network for the purpose of CTC. Accuracy of this method is 92%.

Sheetal S. Pandya[2] describes Pre-processing section of Text Sequence Generation for Gujarati Language Proceedings of the Fifth International Conference on Computing Methodologies and Communication (ICCMC 2021) IEEE Xplore. Gujarati words have completely different meanings reckoning on their region, society, and community. These words known as multi- similar words that usually produce ambiguity in sentences vowels. The system not be able to provide correct answer if the input is poorly worded or ambiguous. Accuracy of this method is 85%.
Agam Dwivedi[3] describes An OCR for Classical Indic Documents Containing Arbitrarily Long Words2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).
Document image texts: We annotated 24,000 lines from three different classical Sanskrit texts – Nirnaya Sindhu, Kavyaprakasha of Mammata, Kshemakutuhalam. The annotations were gathered from Sanskrit domain experts to ensure good data quality. Accuracy of this method is 70%.

Yash Gurav, Priyanka Bhagat [4] describes Devanagari written Character Recognition mistreatment Convolutional Neural Networks 2020 International Conference on Electrical, Communication and laptop Engineering (ICECCE). It uses a self- made Devanagari script dataset which comprises of 29 consonants with no header line (Shirorekha) over them. Data 34604 handwritten pictures. Accuracy of this method is 84%.

Rohit Saluja, Mayur Punjabi [5] describes Sub-word Embeddings for OCR Corrections in Highly Fusional Indic Languages 2019 International Conference on Document Analysis and Recognition (ICDAR).Texts Indic Languages contain a large portion of nonvocabulary (OOV) words due to regular interaction using compound rules. Accuracy of this method is 92%.
Anupama Thakur, Amrit Kaur [6] describes Devanagari Handwritten Character Recognition using Convolutional Neural Networks INTERNATIONAL JOURNAL OF SCIENTIFIC&TECHNOLOGY RESEARCH.
This system was trained using a dataset of 3000 samples. Accuracy of this method is 97.40%.

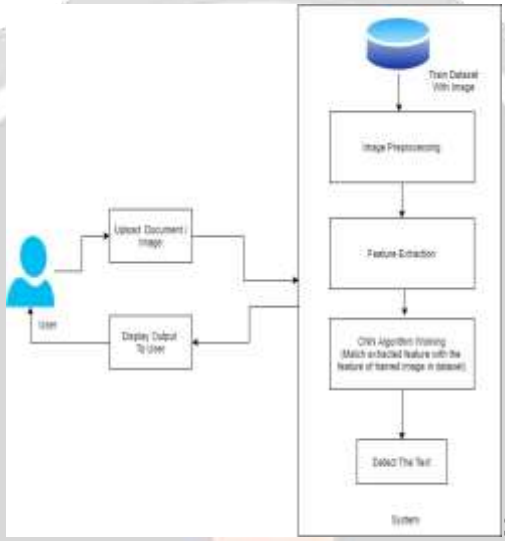
Meduri Avadesh and Navneet Goyal [7] describes Optical Character Recognition for Sanskritic language victimisation Convolution Neural Networks 2018 thirteenth IAPR International Workshop on Document Analysis Systems. The dataset contains 7702 images of sanskrit(Devana gari) letters belonging to 602 classes. Accuracy of this method is 92%.
Rohit Saluja, Devaraj Adiga [8] describes Framework for Accurate Text Detection and Indic OCR Correction 2017 14th IAPR International Conference on Literary Analysis and Recognition.They have edited more than 1100 pages (13 books) in Sanskrit, 190 pages (one book) in Marathi, 50 pages (part of the book) in Hindi and 1000 pages. (12 books) in

English using their outline. Accuracy of this method is 74%.

V. Amrutha Raj, Jyothi; A. Anil kumar [9] describes Grantha script recognition from ancient palm leaves using a contextual histogram IEEE 2017 International Conference on Computers and Communications. The dataset contains 5476 images of sanskrit letters belonging to 408 classes. Accuracy of this method is 96%.

Bipul Pandey, Alok Ranjan [10] describes Multilingual Speaker Recognition Using ANFIS 2010 2nd International Conference on Signal Processing Systems (ICSPS). When proposed system is trained and tested, the average performance achieved turns out to 83.32%. The database used comprises of 25 speakers. From each speaker, 23 input utterances were taken. The database used for the system was prepared by us. The number of errors from a total of 575 input utterances was found to be 75. Accuracy of this method is 73.91%.

III. Architecture

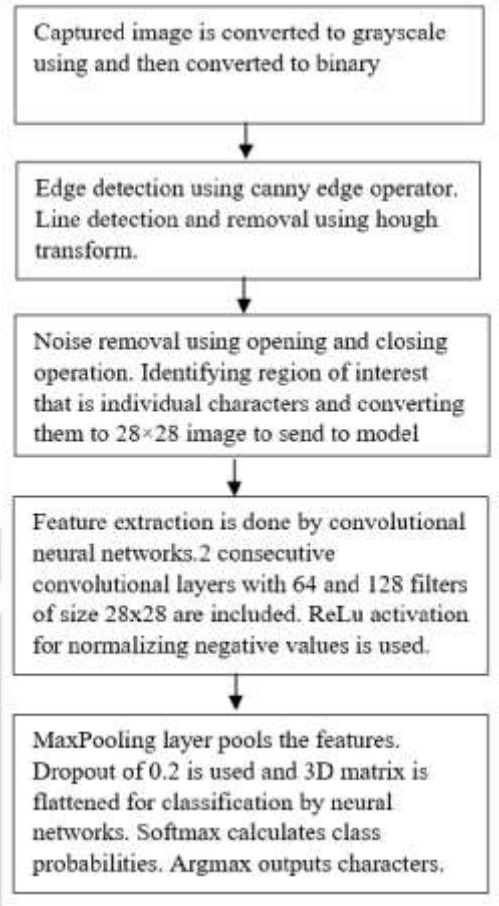


S

system Architecture diagram

IV. Existing System

The system uses Image Processing and deep learning approaches for the process of character spotting, segmentation and feature extraction respectively. Image processing is used to convert images into standard binarized images with the size of 28x28. Feature extraction is performed by the deep learning approach using convolutional neural networks. The flow diagram is shown in the below Fig. 3. A. Image Processing A lot of image processing needs to be done on the input image. Proper segmentation of the character depends on the pre-processing. If the characters are segmented correctly, they increase the overall efficiency of correctly recognizing the characters. The programming is done in python for image processing as well as training the model. CV2 library is used for image processing. The first step is to convert the image to binary because it becomes difficult to perform processing on colour images. For conversion to binary, the image is converted to a grayscale image for this cv2.cvtColor() function is used Fig. 4(a). The grayscale image consists of pixels with values ranging from 0 to 255. The value 0 is assigned to Fig.



Flow Diagram

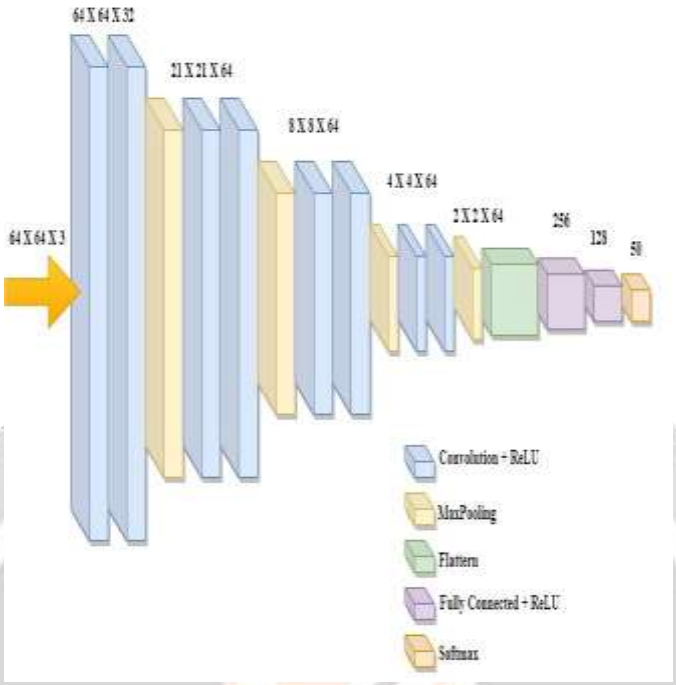
Proposed System

The system Architecture has following functionalities.

- Pre-Processing
  - Read Image
  - Resize Image
  - Remove Noise

- Segmentation
  - Dividing Image
  - Covert image to pixel
- Feature Extraction
  - Filtering
  - Shape Feature
  - Taxture Feature

System Model



Proposed Algorithm Architecture

V. Implementation

1. **Data Collection:** Gather a dataset of Sanskrit text images along with their corresponding ground truth labels (i.e., the actual text in Sanskrit). This dataset will be used for training and evaluating the OCR model.

2. **Data Preprocessing:** Preprocess the Sanskrit text images to enhance their quality and remove noise. Common preprocessing techniques include resizing, grayscale conversion, noise removal, and contrast enhancement.

3. **Character Segmentation:** In order to recognize individual characters accurately, segment the text images into separate characters. This step is crucial for training the OCR model to recognize Sanskrit characters effectively.

4. **Dataset Preparation:** Divide the dataset into training, validation, and testing sets. Typically, the training set contains the majority of the data for model training, while the validation set is used for hyperparameter tuning and model selection. The testing set evaluates the model's performance on unseen data.

5. **CNN Architecture Design**: Design a Convolutional Neural Network (CNN) architecture suitable for OCR tasks. CNNs are widely used for image recognition tasks due to their ability to automatically extract relevant features from images. The architecture should include convolutional layers, pooling layers, and fully connected layers.

6. **Model Training:** Train the CNN model using the prepared dataset. During training, the model learns to recognize Sanskrit characters by adjusting its internal weights based on the provided image-label

pairs. The optimization process usually involves gradient-based optimization algorithms such as stochastic gradient descent (SGD) or Adam.

7. **Model Evaluation:** Evaluate the trained model on the validation set to measure its performance. Metrics like accuracy, precision, recall, and F1 score can be used to assess the OCR model's effectiveness in recognizing Sanskrit characters.

8. **Translation Integration:** Once the OCR model is trained and performs well, integrate a translation component into the system. This can involve using a separate machine translation model or leveraging existing translation APIs. The translated output can be in multiple languages, as desired.

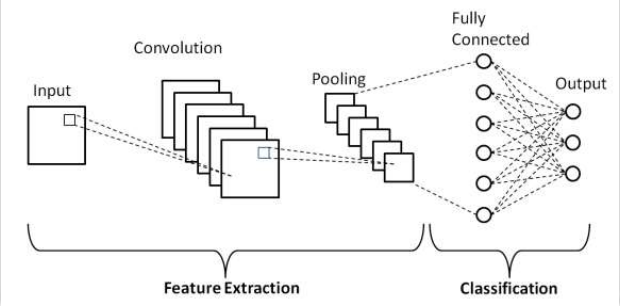
9. **Deployment:** Deploy the OCR and translation system into a production environment where it can process Sanskrit text images and provide translated outputs. This can be achieved through building a web application, creating an API, or integrating it into an existing software system.

10. **Continuous Improvement:** Continuously collect user feedback, monitor the system's performance, and iterate on the OCR and translation models to enhance accuracy and address any shortcomings.

It's worth noting that implementing a complete OCR and translation system involves considerable expertise in deep learning, computer vision, and natural language processing. Additionally, the availability of suitable datasets and computational resources will play a significant role in the system's overall performance.

VI. Algorithms

1) CNN (Convolutional Neural Network)



1. **Input layer**
   As the name says, it's our input image and can be Grayscale or RGB. Every image is made up of pixels that range from 0 to 255. We need to normalize them i.e convert the range between 0 to 1 before passing it to the model.

2. **Convolutional Layer**
   The convolution layer is the layer where the filter is applied to our input image to extract or detect its features. A filter is applied to the image multiple times and creates a feature map which helps in classifying the input image. Let's understand this with the help of an example. For simplicity, we will take a 2D input image with normalized pixels.

3. **Pooling Layer**
   • The pooling layer is applied after the Convolutional layer and is used to reduce the dimensions of the feature map which helps in preserving the important information or features of the input image and reduces the computation time.
   • Using pooling, a lower resolution version of input is created that still contains the large or important elements of the input image.
   • The most common types of Pooling are Max Pooling and Average Pooling.

4. **Fully Connected Layer**

Till now we have performed the Feature Extraction steps, now comes the Classification part. The Fully connected layer (as we have in ANN) is used for classifying the input image into a label. This layer connects the information extracted from the previous steps (i.e Convolution layer and Pooling layers) to the output layer and eventually classifies the input into the desired label.

5. **Output Layer**
   The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into the probability score of each class.

VII. Conclusion

Several OCR systems are studied and various methods, techniques are analyzed. OCR is a challenging field of research and presents new challenges through the use of various methods. Extending it to the Vedic Sanskrit Scripts makes it very difficult due to its large set of letters, compound letters, high degree of similarity between the letters and variations of different writing styles, aging of palm leaves etc. In addition, in CNN architecture, the use of a single convolutional layer is a well-used method but consecutive conversion layers can be used that help extract high-quality features for the purpose of recognition. The accuracy and modification of relevant content requires testing for the satisfactory implementation of the Vedic Sanskrit OCR system. This study identified research gaps and errors in existing systems. In the future, research will be done to reduce the errors mentioned above and to try to develop and design an effective Sanskrit text recognition solution using Neural Networks.

VIII. References

[1] Anoop C. S.; A. G. Ramakrishnan 2021 National Conference on Communications (NCC) CTC-Based End-To-End ASR for the Low Resource Sanskrit Language with Spectrogram Augmentation.

[2] Sheetal S. Pandya ICCMC 2021) IEEE Xplore Preprocessing Phase of Text Sequence Generation for Gujarati Language Proceedings of the Fifth International Conference on Computing Methodologies and Communication.

[3] Agam Dwivedi 2020 IEEE/CVF An OCR for Classical Indic Documents Containing Arbitrarily Long Words Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).

[4] Yash Gurav, Priyanka Bhagat 2020 International Conference on Electrical, Communication and Computer Engineering (ICECCE) Devanagari Handwritten Character Recognition using Convolutional Neural Networks.

[5] Rohit Saluja, Mayur Punjabi Languages2019 International Conference on Document Analysis and Recognition (ICDAR) Sub-word Embeddings for OCR Corrections in Highly Fusional Indic.

[6] Anupama Thakur, Amrit Kaur 2019 INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH Devanagari Handwritten Character Recognition using Convolutional Neural Networks.

[7] Meduri Avadesh ,Navneet Goya 2018 13th IAPR International Workshop on Document Analysis Systems Optical Character Recognition for Sanskrit Using Convolution Neural Networks.

[8] Rohit Saluja, Devaraj Adiga 2017 14th IAPR International Conference on Document Analysis and Recognition A Framework for Document Specific Error Detection and Corrections in Indic OCR .

[9] V. Amrutha Raj, R. L. Jyothi; A. Anil Kumar IEEE 2017 International Conference on Computing Methodologies and Communication Grantha script recognition from ancient palm leaves using histogram of orientation shape context.

[10] Bipul Pandey, Alok Ranjan 2010 2nd International Conference on Signal Processing Systems (ICSPS) Multilingual Speaker Recognition Using ANFIS.