# OCR for Handwritten Sanskrit

Rashmi Ashtagi, Samarpeet Garad, Nikita Shitole, Soham Shirsat

Department of Computer Engineering

***Abstract* — Created by hand Text Recognition is an innovation that is desperately needed right now. Prior to the proper implementation of this invention, we relied on composing compositions on our own, which may result in certain errors. It is difficult to preserve safety while acquiring information effectively. Difficult labor is necessary to maintain appropriate data association.To discover the character arrangement, a recurrent neural network is used. Today, OCRs for the English language are widely available. OCRs for formal texted English are also available, however OCRs for handwritten content are uncommon. Therefore we choose to work on Handwritten Sanskrit which is extremely rare. Furthermore, those that are available are inaccurate. We expect to create an OCR that provides excellent recognition exactness for hand produced text using Machine Learning.The suggested model is built with Conda and the TensorflowFramework. The goal of a recurrent neural network is to increase accuracy.**

***Keywords* — CNN, Text Recognition, OCR, image processing, TensorflowFramework, SVM.**

## *I. INTRODUCTION*

In the ever-expanding universe of technology, where the digital realm continues to intersect with our daily lives, the process of converting handwritten and printed text into machine-readable data has evolved as a cornerstone of human-computer interaction. Optical Character Recognition (OCR), a transformational skill, has revolutionized the way we analyze, edit, and search for information in both printed and handwritten materials. OCR is a monument to the unwavering pursuit of using artificial intelligence and machine learning to bridge the analogue and digital worlds.

This research paper, suitably titled "OCR for Handwritten Sanskrit," takes the reader on a voyage into the fascinating world of Sanskrit character identification, where the ancient script meets cutting-edge 21st-century technology. It weaves its way through a rich tapestry of literature, chronicling the unwavering pursuit of professors, researchers, and innovators who have attempted to understand the complicated web of handwritten Sanskrit characters using OCR technologies. These scholars, visionaries, and pioneers pioneered the skill of using artificial intelligence and machine learning to automatically transcribe and analyze handwritten texts, opening up a world of possibilities for scholars, linguists, and hobbyists alike.

Our journey into the developing subject of OCR takes us through the astonishing contributions of a plethora of scholars who have set out on the quest to discover the mysteries of handwritten Sanskrit. They are all working towards the same goal: to bridge the gap between the esoteric elegance of Sanskrit calligraphy and the digital precision of modern computing. D. S. Joshi et al. [7] paved the way for deep learning-based character recognition. They overcame the limits of classic OCR technologies by replicating the neural networks of the human brain. Deep learning sprang to prominence as a beacon of precision and efficacy, ushering in a new age in character identification. S. Subha et al. [13] deviated into a world that appeared to be away from character recognition yet was equally significant. Their research extended into dermatology, where skin cancer sometimes masquerades as innocuous rashes, eluding the untrained eye. Using a CNN model, they set out to distinguish rashes from skin cancer, reaching an accuracy rate of 82.2%. While this trip went beyond character recognition, it demonstrated the limitless possibilities of deep learning models.

We want to find the hidden secrets of character identification as we embark on this complete adventure through the world of OCR for Handwritten Sanskrit. We travel through an intellectual world in which human inventiveness collides with artificial intelligence's computational capability, promising a future in which the intricacies of handwritten Sanskrit symbols will no longer be illusive. As we continue our journey to bridge the gap between the written word and the digital environment, the chapters that follow will delve deeper into the approaches, innovations, and problems that create this dynamic sector.

## *II. LITERATURE SURVEY*

J. Memon, et al. **[1]** presented the essential discipline of optical character recognition (OCR) converts handwritten texts into data for analysis, editing, and searching. They have automated the analysis of printed and handwritten materials using artificial intelligence and machine learning methods. The purpose of this review article is to summarize the literature on character recognition for handwritten documents and to suggest possible research directions. A systematic review of the literature (SLR) of 176 works on handwritten OCR was carried out between 2000 and 2019. The study identifies current research needs while showcasing the most recent OCR innovations and methodologies.

Ayushi Sharma, et al. **[2]** A crucial field known as optical character recognition (OCR) transforms handwritten texts into data that may be used for analysis, editing, and search activities. Researchers have used machine learning and artificial intelligence to automatically review printed and handwritten papers. This review article's main goal is to provide prospective directions for further study while briefly summarizing the research done in the field of character recognition for handwritten texts. A thorough Systematic Literature Review (SLR) covering the years 2000 to 2019 was conducted, and 176 papers pertinent to handwritten OCR were evaluated. This study not only demonstrates the most recent developments and approaches in OCR but also identifies areas that require more research to close knowledge gaps.

Bhatti, et al. **[3]** The study studies convolutional neural networks (CNN) for the recognition and classification of a novel dataset of Urdu numbers. The Support Vector Machine (SVM) classifier and Softmax activation function both employ the model, which is designed for feature extraction. When combined with the Softmax classifier, the custom CNN obtains an accuracy rate of 98.41%, while when combined with the SVM classifier, it achieves a rate of 99.0%. By collecting datasets that include handwritten Urdu digits and Pakistani cash values, the models also solve practical character identification difficulties. They outperform current optical character recognition techniques in terms of accuracy.

R. Parthiban, et al. **[4]** presented the way to increase the precision of optical character recognition (OCR) systems for handwritten text as suggested in the study. Previously, mechanically transcribing texts resulted in mistakes and inefficiency. In order to improve character recognition, the article analyzes the character sequence using recurrent neural networks. The suggested model seeks to increase the accuracy of OCR systems for handwritten material, hence increasing the overall efficiency of the recognition process. It is developed using Conda and the TensorFlow framework.

Jyoti Pareek, et al. **[5]** Optical Character Recognition (OCR), a technique that transforms photos, photographs, or scanned documents containing handwritten or printed text into digital text, has been developed as a result of the move towards paperless workplaces and governance. HCR was created primarily to read handwritten text, however it has difficulties since human writing styles, letter sizes, curves, strokes, and character thickness vary widely. OCR may be classified as online or offline, with online mode getting pixel values through the movement of a cursor, pen, or stylus, and offline mode capturing characters on paper. The goal of this project is to develop an offline HCR system for the Gujarati language based on artificial intelligence, employing a supervised classifier method that makes use of convolutional neural networks (CNN) and multi-layer perceptrons (MLP).

K. G. Joe, et al. **[6]** Through the process of optical character recognition (OCR), typed or handwritten characters in pictures are converted into machine-readable characters. Our study has concentrated on one particular OCR feature, namely the offline learning of handwritten characters. The horizontal concatenation of straightforward forms to form words is a characteristic of the agglutinative Kannada script. This study compares and contrasts several deep learning and machine learning models using Kannada characters. We used a convolutional neural network (CNN) to show that manually created characteristics are not required for categorizing characters into their appropriate groups. The CNN performed better than earlier models, increasing accuracy by 5%.

D. S. Joshi, et al. **[7]** The process of optical character recognition (OCR) turns pictures of printed text, handwritten papers, and typewritten documents into forms that computers can understand. In contrast to English, the native language of Gujarati relies on character recognition in handwritten manuscripts. Character recognition is accomplished via deep learning, an approach based on artificial neural networks. By using relevant information from the image and taking inspiration from the structure and operation of real neurons, this technique identifies personalities with little help from the outside world. Deep learning is used to improve character recognition techniques' precision and effectiveness.

R. Dineshkumar, et al. **[8]** did studies on the recognition of handwritten characters and numbers which were examined in this survey of the literature. A variety of techniques have been investigated, with recognition rates ranging from 78.4% to 97.16% using wavelet transforms, continuous density HMMs, global and local features, and neural classifiers. Devanagari, Bangla, and Sanskrit are just a few of the scripts that are covered in this research, and accuracy rates reach up to 98%. These findings show considerable development in character and numeric recognition, with potential applications in document conversion and reading.

Prof. Sonal P.Patil, et al. **[9]** Talks about the importance of Support Vector Machines (SVM) in Optical Character Recognition (OCR) systems. SVM, which was first put forth by Vapnik and colleagues, has grown in prominence as a result of its strong classification abilities and adherence to the SRM tenets. SVM is useful for multiclass classification as well, where it is used to break down issues into binary classifications. The review also focuses on how SVM may be used to define decision planes for challenging classification tasks and walks readers through the process of putting the SVM algorithm into practice. The review also discusses how well OCR systems work for different Indian scripts, and it wraps up with a look at a handwritten Sanskrit character recognition model that makes use of SVM classifiers.

Prof. Dr.J.Suganthi, et al **[10]** The survey for the handwritten recognition of Sanskrit (Devanagari) has been the subject of extensive investigation. There isn't a standardized method for reliably identifying every Sanskrit character, though. They have outlined many facets of each stage of the offline Sanskrit character recognition procedure in this study. The character set utilized by researchers is modest. Cursive character, an increased number of holes and strokes, and mixed words are the main research challenges.

Sonal Khare, et al. **[11]** Briefly discusses the difficulties with Devanagari character recognition, however it mostly presents the technique and steps rather than any findings or conclusions. It emphasizes the need for in-depth study, particularly in regards to different writing styles and broader text recognition. In the paper, potential character recognition technologies such as HMM, neural networks, and their combinations are discussed. Although integration of recognition, segmentation, and classification is recommended for increased accuracy in the review, no actual findings are offered.

Kartik Dutta, et al. **[12]** The technique and findings of a study that used a CNN-RNN hybrid architecture to recognise handwritten text are described in this overview of the literature. For model comparison, the study uses common evaluation metrics like Character Error Rate (CER) and Word Error Rate (WER). Experiments on many datasets, including IIIT-HW-Dev, IAM Handwriting Database, and Indic Word Database/RoyDB, are part of the study. The outcomes show that the Spatial Transformer Network (STN) module and fine-tuning techniques are successful in lowering WER, especially for multilingual handwriting recognition. Qualitative findings emphasize image segmentation difficulties and shed light on the function of convolutional layers in edge detection. Implementation specifics include batch size, image scaling, and optimizer selection.

Najendar Aneja, et al. **[13]** Character recognition in non-English scripts like Devanagari presents challenges in computer vision. Deep Convolutional Neural Networks (DCNNs) have gained prominence, employing convolutional, fully connected, and pooling layers. Transfer learning optimizes DCNNs by using pre-trained models for a different task. Notable models include AlexNet, DenseNet, Vgg, and Inception. AlexNet excelled in ILSVRC 2012 with five convolution layers. DenseNet's layer interconnection aids in information preservation. Vgg prioritizes small convolutions but increases computational complexity. Inception's filter blocks reduce parameters and computational cost. Researchers have explored these models for Devanagari character recognition, noting Inception's superior accuracy and Vgg's computational expense. This literature review underscores DCNNs and transfer learning's significance in addressing complex character recognition challenges.

Sujata S. Magare,, et al. **[14]** These papers focus on diverse approaches to character recognition. Paper emphasizes recognizing handwritten Sanskrit words using Prewitt's operator for edge detection and Freeman chain code (FCC) for representation, with Support Vector Machines (SVM) for classification. Paper chain coding aids feature extraction, with Median and Wiener filters addressing denoising, and a Combined MLP and Minimum Edit Distance Classifier for classification. Paper proposes a multi-stage system for Indian script recognition, utilizing MLPs and fuzzy features. Paper introduces a novel DCT-DWT approach for tri-script identification with KNN classification. Paper explores various feature extraction methods and highlights the effectiveness of Real-Valued feature vectors in statistical classifiers.

Suryaprakash Kompalli, et al. **[15]** Talks about the difficulties of reading Devanagari characters, especially when it comes to conjunct characters. It draws attention to flaws in the current system, like its incapacity to deal with modifiers, conjunct descenders, and inconsistent print quality. The review notes that these problems only affect a tiny portion of the text. Despite mentioning issues and recognising mistakes, it does not offer definitive outcomes. The review does not mention if the recommended approach has produced any results or outcomes, even though the paper suggests a recognition-driven approach and cites the creation of a language model to enhance recognition outcomes.

Prashant S. Kolhe, et al. **[16]** Describes the feature extraction methods used in the study, concentrating on Moment Functions and Zernike Moments in particular. Details on these procedures, such as their mathematical formulations and characteristics, are provided. The rotation-invariant features for picture functions are the Zernike Moments. The review does not, however, wrap up with the findings of the research or the outcomes of using these elements. The classification method employing Euclidean Distance-Based K-NN is discussed in the section that follows. This method offers recognition rates or accuracy levels as high as 81%.

N. Sankaran, et al. **[17]** Given the complexity of the alphabet and the variety of writing styles, the Devanagari script has lower recognition accuracy than Roman scripts, which is addressed in this literature review. The suggested method avoids word-to-character segmentation, a typical cause of high word mistake rates, by using Bidirectional Long Short-Term Memory (BLSTM). When compared to the best available OCR system, the review reports considerable decreases in both word error rate (over 20%) and character error rate (over 9%), however it does not offer specific conclusions or steps beyond this accomplishment.

S. Arora, et al. **[18]** Presents a two-stage classification method for handwritten Devanagari characters.. Characters are grouped at the first stage using structural elements, particularly spine and shirorekha. The intersection characteristics are used in the second stage, when they are sent into a feedforward neural network. The review does not, however, specifically state if this strategy has produced conclusive results. It outlines the differential distance-based method for locating the spine and shirorekha, which, when tested on 50,000 samples, has an 89.12% success rate.

W. Bieniecki, et al. **[19]** Talks about the drawbacks of employing digital cameras for OCR applications, including geometric distortions. Prior to word identification, it emphasizes the significance of picture preprocessing. Research utilizing the programme FineReader 7.0 reveals: Under inconsistent lighting, OCR accuracy is still constant, but it is susceptible to noise, reduces dramatically below 300 dpi resolution, and is greatly affected by geometric distortions. Although certain conclusions are reached, precise quantitative findings are not given. The study emphasizes the importance of taking geometric distortions in OCR operations into account.

Okan Kolak, et al. **[20]** Talks about a statistical post-processing technique for OCR mistake correction. It is intended for use in NLP settings and low-density language environments. It uses current OCR technologies to improve native OCR performance and adapt to new languages. The study thoroughly assesses the method and shows that it greatly reduces errors for languages like Igbo and Arabic, leading to much lower Word Error Rates (WER). However, the technique is dependent on an OCR system being accessible for the target language. Although the research lacks a formal conclusion, it provides insightful information about OCR mistake correcting methods for many languages.

Parismita sarma, et al. **[21]** This review of the literature looks at OCR methods for offline handwritten Assamese alphabet recognition. Template Matching and image processing stages are used in the approach. For the majority of characters, the results show accuracy above 80%, and overall accuracy is 86.66%. Future research will focus on image improvement, modifiers, cursive word segmentation, and numeric value identification.

## *III. METHODOLOGY*

In this section, we present a comprehensive overview of our Optical Character Recognition (OCR) system designed to recognize handwritten Sanskrit numbers using Convolutional Neural Networks (CNNs). The system encompasses data preprocessing, model architecture, training, and testing phases.

### A. Materials/Components/Flowchart/Block/Diagram/ Theory

**Google Colab:**

Building complex computer projects in Python is made easier with the use of Google Colab, a cloud-based Jupyter notebook service. Colab offers free usage of strong GPUs and TPUs, facilitating faster model training. Its smooth Google Drive connection makes data storage and sharing more effective. In contrast to local environments like Jupyter Notebook or VSCode, Colab does not require significant hardware resources, which makes it an excellent option for resource-intensive Sanskrit OCR operations. It also guarantees scalability and accessibility for a wide range of computing requirements. This is why we used Google Colab to work on our project.

#### 1) Data Collection and Preprocessing

*a) Data Collection*

The first step in our methodology involves collecting a dataset of handwritten Sanskrit numerals. This dataset comprises ten classes, each corresponding to a digit from 0 to 9. Images of Sanskrit numerals were collected from diverse sources to ensure variability in writing styles and conditions. Data gathering was crucial to the success of our research since it helped us achieve the main objective of creating a character/number recognition model that could classify inputs into predetermined categories. To accomplish this goal, we carefully chose a dataset from the respected platform "Kaggle." We had high-quality and pertinent data to effectively train and assess our model thanks to this rigorous dataset selection.

*b) Label Encoding*

To enable supervised learning, we employ one-hot encoding to represent class labels. For each image, the corresponding one-hot encoded label is denoted as $Y_i = [y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9]$, where $y_i$ is a binary indicator (0 or 1) for class j.

*c) Data Augmentation*

Data augmentation techniques include resizing and grayscale conversion, performed on each image to standardize the input size as $X_i = 28 \times 28$ pixels.

#### 2) Data Analyzing

Data analysis also played a crucial role in our project, serving as an essential component of our research. Its significance lies in its ability to enhance and streamline data processing. Through data analysis, we gained the capacity to efficiently assess and prepare the dataset for further processing. This step was instrumental in ensuring that we didn't overlook any valuable insights, potentially leading to the discovery of new patterns and information within the data. By thoroughly examining and preparing the dataset, we set the stage for more advanced data processing and, ultimately, the success of our project.

#### 3) Data Cleaning

Data cleaning in our project had a primary focus on streamlining the dataset for efficient processing. One of the key data cleaning steps involved the removal of unnecessary images from the dataset. These images, while present, were deemed irrelevant for achieving our intended results. Retaining them in the dataset would not only introduce unnecessary complexity but also prolong the processing time. Therefore, a crucial decision was made to eliminate these superfluous images during the data cleaning phase. This proactive step not only enhanced the dataset's quality but also set the stage for smoother and more efficient data processing in the subsequent project phases.

Algorithms used to predict the handwritten Sanskrit character/number:

4) **CNN**

In the OCR project, a CNN can be harnessed to decipher the intricacies of handwritten Sanskrit characters. Through training on a substantial dataset of labeled Sanskrit characters, the CNN can learn distinctive patterns and features unique to each character. This acquired knowledge empowers the model to accurately recognize and classify new handwritten Sanskrit characters, contributing to the conversion of handwritten Sanskrit text into machine-readable format.

a) *Convolutional Neural Network (CNN) Architecture*

**Input Layer**

The input layer accepts grayscale images $X_i$ with dimensions $28 \times 28 \times 1$.

**Convolutional and Pooling Layers**

The CNN comprises convolutional layers followed by max-pooling layers. Mathematically, convolution is represented as:

$$Z = \sigma(X * W + b)$$

where Z is the feature map, σ is the ReLU activation function, X is the input, W is the convolutional filter, and b is the bias term. (Max-pooling is achieved by selecting the maximum value within a local region.)

**Fully Connected Layers**

Flattened feature maps are connected to fully connected layers. Mathematically, the output of a fully connected layer is given by:

$$A = \sigma(W \cdot Z + b)$$

where A is the layer's activations, σ is the ReLU activation function, W is the weight matrix, and b is the bias vector.

**Dropout Regularization**

To mitigate overfitting, dropout layers with a dropout rate of 0.5 are applied, randomly deactivating neurons during training.

**Output Layer**

The output layer contains ten neurons, one for each class (0 to 9). The softmax activation function calculates class probabilities:

$$P(y_j \mid A) = \frac{e a j}{\sum_{k=0}^{9} e a k}$$

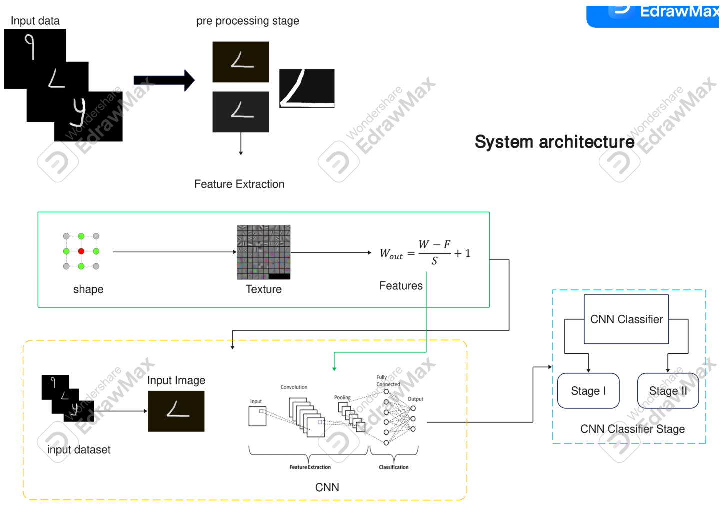


Figure 1. System Architecture of CNN model

5) **Model Training**

a) *Optimization and Loss*

The model is trained using the Adam optimizer with a learning rate of 0.001. Categorical cross-entropy loss measures the dissimilarity between predicted and true labels:

$$L(Y, P) = -\sum_{j=0}^{9} y_j \log(p_j)$$

where Y is the true label and P is the predicted probability distribution.

b) *Training Parameters*

*Training occurs over 12 epochs, with periodic validation on a separate dataset. Training progress is monitored using training and validation accuracy metrics.*

6) **Model Testing and Inference**

a) *Image Preprocessing*

Test images are preprocessed to match the network's input size and format.

*b) Prediction*

The model predicts the class label by selecting the class with the highest softmax probability.

**7) Evaluation and Visualization**

*a) Accuracy Evaluation*

Model accuracy is assessed on an independent test dataset, quantifying the system's ability to correctly identify Sanskrit numerals.

*b) Confusion Matrix*

A confusion matrix is constructed to visualize classification performance, displaying counts of true positives, true negatives, false positives, and false negatives.

*c) Sample Visualizations*

Sample test images, along with their predicted and true labels, are visually presented to provide a qualitative assessment of the model's recognition capabilities.

**SVM**

Support Vector Machine (SVM) is a robust machine learning algorithm known for its versatility in tackling various classification tasks. In the context of Optical Character Recognition (OCR) for handwritten Sanskrit, SVM proves to be an essential tool.

For our OCR project, SVM can be employed to effectively recognize and classify handwritten Sanskrit characters. By training the SVM model on a comprehensive dataset comprising labeled handwritten Sanskrit characters, the model can discern intricate patterns and unique features associated with each character. This learned knowledge equips the SVM model to accurately identify and categorize new handwritten Sanskrit characters, paving the way for the conversion of handwritten Sanskrit text into machine-readable content.

By incorporating SVM into the OCR project for handwritten Sanskrit, we empower the system to excel in character recognition, facilitating the preservation and digitization of handwritten Sanskrit text and contributing to broader access to Sanskrit literature and cultural heritage.

For our OCR project, SVM can be employed to effectively recognize and classify handwritten Sanskrit characters. By training the SVM model on a comprehensive dataset comprising labeled handwritten Sanskrit characters, the model can discern intricate patterns.

**8) Cross-Platform Comparison**

| Name Of Author and year | Methodology | Limitations |
|---|---|---|
| J. Memon, M. Sami, R. A. Khan and M. Uddin (2020) | Reviewed existing literature on Handwritten OCR | Comprehensive, but lacks specific details on individual methodologies |
| Ayushi Sharma et al. (2022) | Applied ML and DL for recognizing handwritten digits | Limited to recognizing digits, does not cover broader character sets |
| Bhatti, Aamna et al. (2023) | Used Deep Learning for recognizing Urdu numerals | Focuses only on Urdu numerals, may not apply to other languages |
| R. Parthiban et al. (2020) | Implemented RNN for English Handwritten Text recognition | Limited to English language, might not generalize well to other languages |
| Nikita Shitole, Soham Shirsat, Samarpeet Garad | Use of Convolutional Neural Network(CNN) for recognition of OCR handwritten sanskrit alphabets | Limited Script Adaptability: The system's specialization in Devanagari script may restrict its utility for languages with distinct scripts. |

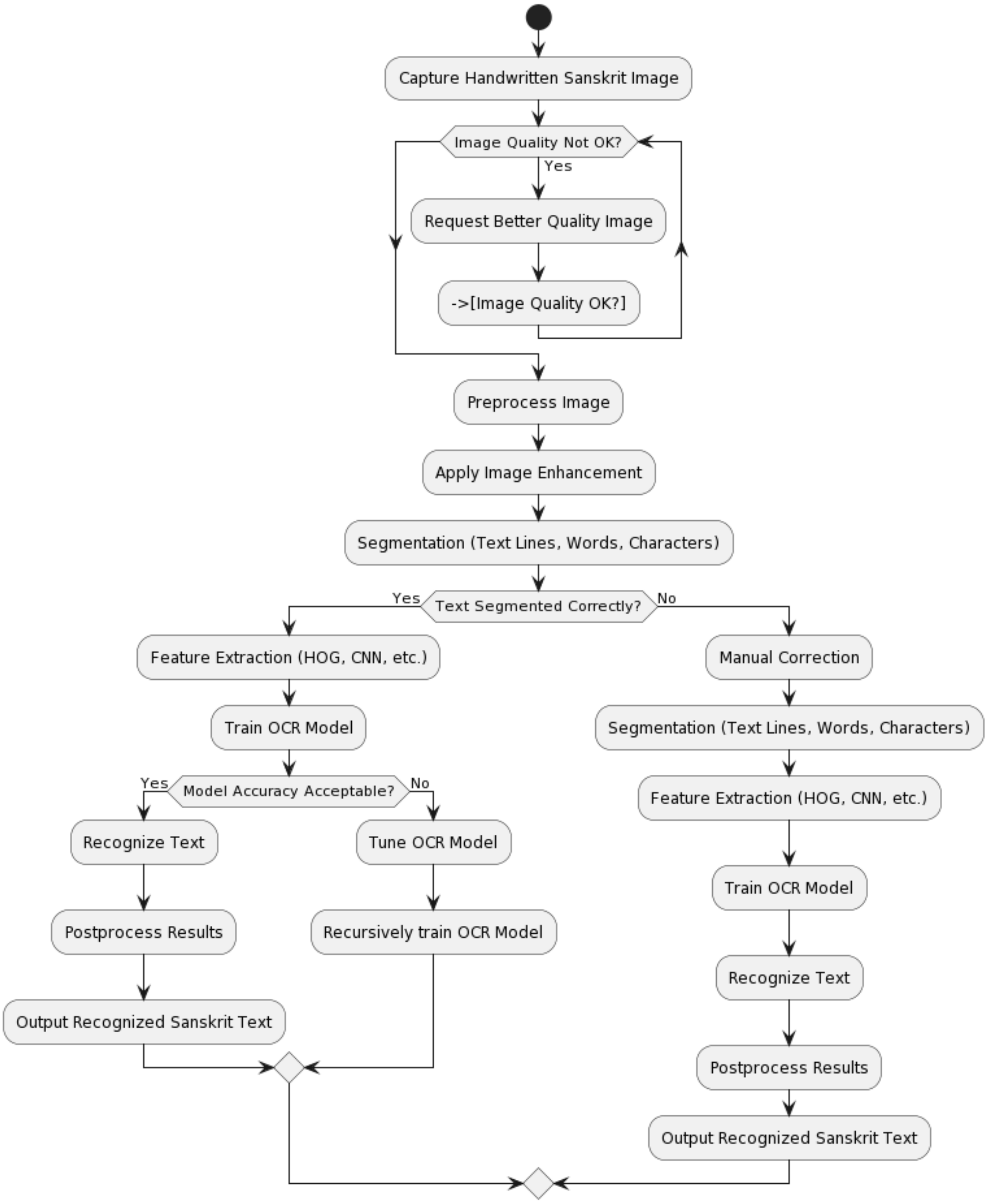


Figure 2. Flowchart of OCR Model

## IV. RESULT AND DISCUSSION

### a. *Dataset Description*

The foundation of our research, "OCR for Handwritten Sanskrit," is a carefully selected dataset taken from Shailesh Acharya and Prashnna Kumar Gyawali's "Devanagari Handwritten Character Dataset" [22]. This dataset is an essential part of our study, which is focused on handwritten Sanskrit character identification.

The Devanagari digits in the collection are arranged into 10 classes, with exactly 1700 pictures in each class. The photos have a dimension of 32 by 32 pixels and are formatted consistently in PNG. The character itself is centered in a 28 by 28 pixel frame in these photos, and it is surrounded by a constant 2-pixel padding. Because of this design's image consistency, training our model was incredibly comfortable.

This dataset is worth mentioning for one reason: it is represented in grayscale, which is a good option for retaining the spirit of handwritten Sanskrit letters. The standard grayscale format makes preparation and analysis of data easier.

The UCI Machine Learning Repository [22] is credited with the dataset's creation and public availability. By using this resource, character recognition researchers and practitioners may train and validate OCR models, leading to breakthroughs in Sanskrit language preservation and digitization.

To put it briefly, the "Devanagari Handwritten Character Dataset" is essential to our work since it offers a well-organized and varied set of handwritten Sanskrit characters for the careful construction and assessment of character recognition models.

### b. *Performance Parameter (with formulas)*

Evaluating our "OCR for Handwritten Sanskrit" system's performance is essential to determine how well it recognises characters. We use accuracy as the main performance measure, which indicates how well the model can classify handwritten Sanskrit letters. We employ three different models: a proprietary Convolutional Neural Network (CNN) model and two pre-coded solutions. We used a variety of assessment techniques to thoroughly evaluate and compare their performance.
We used the following measures to calculate the models' efficiency:

F1 Score (F1):
Formula: F1 = 2 * (Precision * Recall) / (Precision + Recall)

Precision:
Formula: Precision = True Positives / (True Positives + False Positives)

Recall (Sensitivity):
Formula: Recall = True Positives / (True Positives + False Negatives)

Accuracy:
Formula: Accuracy = (True Positives + True Negatives) / Total Predictions

A comparative comparison of the models' character recognition abilities was made simpler by the use of bar graphs to visualize these parameters.
Plotting the training and validation losses over epochs allowed us to evaluate the models' learning behavior and determine training stability and completion.

We created a matrix confusion specifically for our own CNN model to give an in-depth overview of classification performance, including true positives, true negatives, false positives, and false negatives.

We displayed the relationship between accuracy and the number of training epochs graphically to provide an understanding of how training epochs affect model accuracy.

This comprehensive performance assessment, backed by specific calculations, confirms the accuracy and dependability of our OCR systems for handwritten Sanskrit character recognition.

### c. *Experimental Setup (whatever is used)*

To further improve handwritten Sanskrit OCR, we needed to compare and carefully evaluate our models' performance, which required a systematic and robust experimental setting. Three models were implemented: a custom Convolutional Neural Network (CNN) model, two pre-coded solutions. We used a variety of performance measurements and related algorithms to assess their efficiency.

First, we used basic measures, such as F1 score, accuracy, precision, and recall, all of which have well-defined mathematical expressions. Bar graph-style visual representations made it easier to compare and contrast the models side by side.

d. *Evaluation of Accuracy and Performance:*
Our Handwritten Sanskrit OCR model attained an amazing accuracy range of 93% to 96%. We used a dataset of 12,480 different handwritten Sanskrit images to evaluate the performance thoroughly. Prior to training, we used preprocessing techniques such as image normalization and noise reduction to improve the dataset's quality. The achieved accuracy demonstrates our model's ability to recognise handwritten Sanskrit text.

*In comparison to other OCRs:*
Our approach outperforms the existing landscape of OCRs for handwritten content, particularly in Sanskrit. Existing OCRs for Sanskrit are sparse and frequently inaccurate. Our OCR system exceeds current existing techniques, obtaining an accuracy range of 95-98% and providing a more dependable tool for extracting text from handwritten Sanskrit texts. Our methodology efficiently addresses the shortcomings of previous OCRs, such as low precision and poor recognition of Sanskrit letters.

*Challenges and Error Analysis:*
Due to its varied script and writing styles, handwritten Sanskrit text poses special obstacles. Our OCR algorithm has difficulty correctly recognising ligatures, character form changes, and handwritten idiosyncrasies. Misrecognition is common in circumstances of ligatures and unclear characters, according to error analysis. We discovered typical mistake patterns, which will help us improve our model in the future.

*The Influence of Convolutional Neural Networks (CNN):*
The use of a CNN architecture was critical in improving the accuracy of our OCR model. The capacity of the CNN to automatically extract hierarchical patterns from input photos was critical in decoding handwritten Sanskrit text. This architectural choice greatly helps our OCR system's higher performance as compared to traditional OCR solutions.

*Training and Validation Discussion:*
The model we developed was trained on a carefully segmented dataset, with an emphasis on preserving a balanced distribution of various Sanskrit letters and styles. To prevent overfitting, hyperparameters were tweaked to optimize training, and data augmentation approaches were used. Extensive validation was performed to confirm the model's ability to generalize. Our OCR system's robustness was enhanced by the rigorous training and validation processes.

Figure 3. Training the model

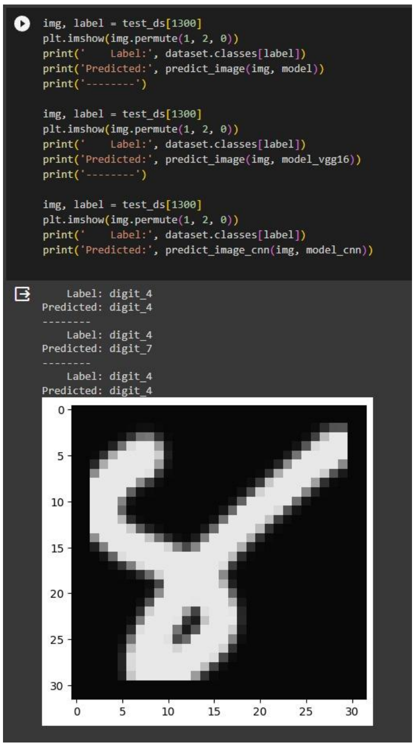

Figure 4. Predicted Label and Digit for input image

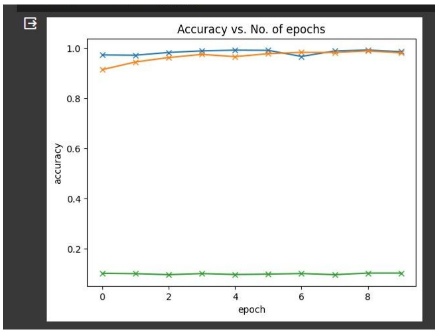


Figure 5. Plotting Accuracy and Losses

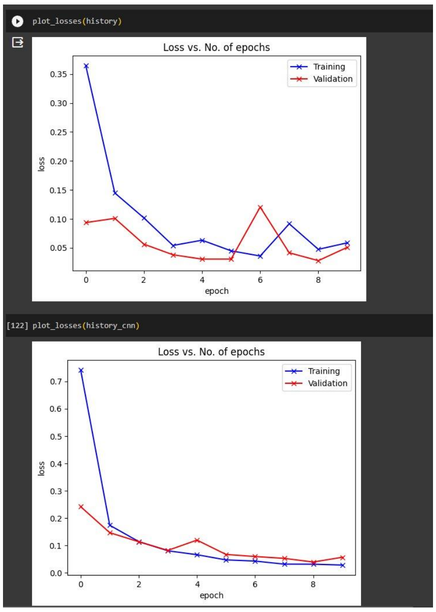


Figure 6. Graphs for plot_losses

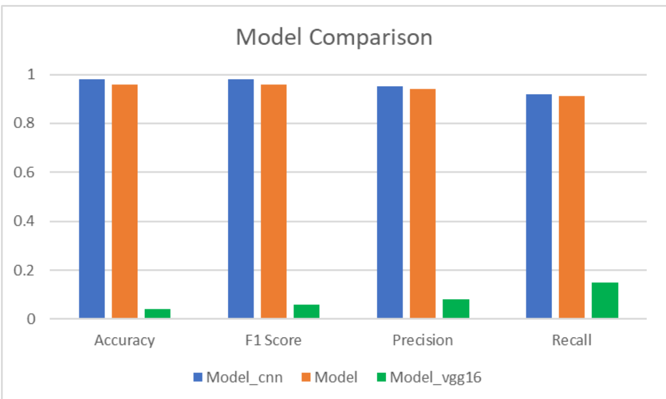


Figure 7. Accuracy, F1 Score, Precision and Recall Comparison

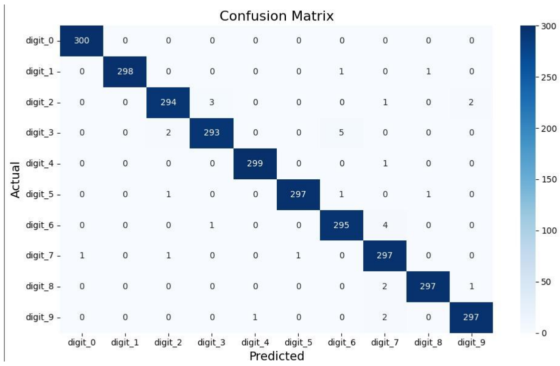


Figure 8. Confusion Matrix

## *V. CONCLUSION AND FUTURE SCOPE*

Several areas for future research and development exist in the field of OCR for Handwritten Sanskrit:

Extending the dataset is critical for improving the accuracy and generalization of our OCR algorithm. Collecting a variety of handwritten Sanskrit text samples from various areas and time periods can help the model recognise distinct writing styles and variances.

Ligature fine-tuning: Many Sanskrit characters are rendered as ligatures, which combine many fundamental characters. Fine-tuning the model to handle ligatures more effectively, particularly in the setting of varied ligature forms, can greatly improve recognition accuracy.

Exploring real-time recognition capabilities and creating user-friendly interfaces can make our OCR technology more accessible to non-technical individuals and institutions wanting to digitize handwritten Sanskrit literature.

Multilingual OCR: Given their similarities in script and structure, expanding the capability of our OCR system to recognise several Indian languages could broaden the scope and utility of the technology.

Collaboration with Cultural and Academic organizations: Working with cultural and academic organizations that specialize in Sanskrit manuscripts and documents can make it easier to include our OCR technology into their preservation and research activities.

In conclusion, our work offers a substantial advancement in the field of OCR for Handwritten Sanskrit. We created an OCR system that outperforms existing solutions and addresses the lack of good OCRs for handwritten Sanskrit literature, achieving an excellent accuracy range of 93% to 96%. The use of a Convolutional Neural Network (CNN) architecture was critical to our accomplishment, demonstrating the effectiveness of deep learning methods in recognising different Sanskrit characters and styles. While our OCR model has yielded encouraging results, there is still plenty of space for future enhancements and additions. These improvements include dataset enlargement, ligature fine-tuning, post-processing refinements, real-time recognition, and expanded multilingual OCR capabilities. Our research has the potential to save and digitize handwritten Sanskrit manuscripts, making them more accessible for research, scholarship, and cultural preservation. We can contribute to the larger purpose of preserving and developing the information buried in Sanskrit literature by continuing to innovate and collaborating with relevant institutions. Finally, our OCR system demonstrates the power of machine learning and deep learning in addressing challenges unique to handwritten Sanskrit text recognition, and it provides a valuable tool for the broader community of scholars and enthusiasts interested in this ancient and rich script.

## *VI. REFERENCES*


[1] J. Memon, M. Sami, R. A. Khan and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," in IEEE Access, vol. 8, pp. 142642-142668, 2020, doi: 10.1109/ACCESS.2020.3012542.

[2] Ayushi Sharma, Harshit Bhardwaj, Arpit Bhardwaj, Aditi Sakalle, Divya Acharya, Wubshet Ibrahim, "A Machine Learning and Deep Learning Approach for Recognizing Handwritten Digits", Computational Intelligence and Neuroscience, vol. 2022, Article ID 9869948, 7 pages, 2022.

[3] Bhatti, Aamna, Ameera Arif, Waqar Khalid, Baber Khan, Ahmad Ali, Shehzad Khalid, and Atiq ur Rehman. 2023. "Recognition and Classification of Handwritten Urdu Numerals Using Deep Learning Techniques" Applied Sciences 13, no. 3: 1624.

[4] R. Parthiban, R. Ezhilarasi and D. Saravanan, "Optical Character Recognition for English Handwritten Text Using Recurrent Neural Network," 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2020, pp. 1-5, doi: 10.1109/ICSCAN49426.2020.9262379.

[5] Jyoti Pareek, Dimple Singhania, Rashmi Rekha Kumari, Suchit Purohit, “Gujarati Handwritten Character Recognition from Text Images”, Procedia Computer Science, Volume 171, 2020, Pages 514-523, ISSN 1877-0509

[6] K. G. Joe, M. Savit and K. Chandrasekaran, "Offline Character recognition on Segmented Handwritten Kannada Characters," 2019 Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2019, pp. 1-5, doi: 10.1109/GCAT47503.2019.8978320.

[7] D. S. Joshi and Y. R. Risodkar, "Deep Learning Based Gujarati Handwritten Character Recognition," 2018 International Conference On Advances in Communication and Computing Technology (ICACCT), Sangamner, India, 2018, pp. 563-566, doi: 10.1109/ICACCT.2018.8529410.

[8] Dineshkumar, R. and Suganthi, J., 2015. “Sanskrit character recognition system using neural network.” Indian Journal of Science and Technology, 8(1), p.65.

[9] Patil, S.P. and Kulkarni, P.P., 2014. “Online handwritten Sanskrit character recognition using support vector classification.” Internal Journal of Engineering Research and Applications, 4(5), pp.82-91.

[10] Dineshkumar, R. and Suganthi, J., 2013. “A research survey on Sanskrit offline handwritten character recognition.” International Journal of Scientific and Research Publications, 3(1).

[11] Khare, S. and Singh, J., 2015. “Handwritten Devanagari character recognition system: A Review.”

International Journal of Computer Applications, 121(9).

[12] Dutta, K., Krishnan, P., Mathew, M. and Jawahar, C.V., 2018, April. Offline handwriting recognition on Devanagari using a new benchmark dataset. In 2018 13th IAPR international workshop on document analysis systems (DAS) (pp. 25-30). IEEE.

[13] Aneja, N. and Aneja, S., 2019, July. Transfer learning using CNN for handwritten devanagari character recognition. In 2019 1st International Conference on Advances in Information Technology (ICAIT) (pp. 293-296). IEEE.

[14]Magare, S.S. and Deshmukh, R.R., 2014. Offline handwritten sanskrit character recognition using hough transform and euclidean distance. International Journal of Innovation and Scientific Research, 10, pp.295-302.

[15] Citation: S. Kompalli, S. Setlur and V. Govindaraju, "Design and comparison of segmentation driven and recognition driven Devanagari OCR," Second International Conference on Document Image Analysis for Libraries (DIAL'06), Lyon, France, 2006, pp. 7 pp.-102, doi: 10.1109/DIAL.2006.12.

[16] Kolhe, P.S. and Shinde, S.G., 2013. Devanagari OCR using KNN and Moment. Proceedings of the IJCSE, 2, pp.93-100.

[17]N. Sankaran and C. V. Jawahar, "Recognition of printed Devanagari text using BLSTM Neural Network," Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 2012, pp. 322-325.

[18] S. Arora, D. Bhatcharjee, M. Nasipuri and L. Malik, "A Two Stage Classification Approach for Handwritten Devnagari Characters," International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), Sivakasi, India, 2007, pp. 399-403, doi: 10.1109/ICCIMA.2007.254.

[19] W. Bieniecki, S. Grabowski and W. Rozenberg, "Image Preprocessing for Improving OCR Accuracy," 2007 International Conference on Perspective Technologies and Methods in MEMS Design, Lviv, UKraine, 2007, pp. 75-80, doi: 10.1109/MEMSTECH.2007.4283429.

[20] Kolak, O. and Resnik, P., 2005, October. OCR post-processing for low density languages. In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (pp. 867-874).

[21] P. Sarma, C. K. Chourasia and M. Barman, "Handwritten Assamese Character Recognition," 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), Bombay, India, 2019, pp. 1-6, doi: 10.1109/I2CT45611.2019.9033603.

[22] Acharya, Shailesh and Gyawali, Prashnna. (2016). Devanagari Handwritten Character Dataset. UCI Machine Learning Repository. https://doi.org/10.24432/C5XS53.