A Project Report on

Recognition of Brahmi Script and Its Variant Using ML

Submitted by,

Gauri Kardekar	(Exam Seat No. 202101040137)
Gargi Gundawar	(Exam Seat No. 202101070184)
Shrawani Chaudhari	(Exam Seat No. 202101060009)
Sakshi Deshmukh	(Exam Seat No. 202101070114)

Guided by,

Prof. Vinayak Kulkarni

A Report submitted to MIT Academy of Engineering, Alandi(D), Pune,
An Autonomous Institute Affiliated to Savitribai Phule Pune University
in partial fulfillment of the requirements of

BACHELOR OF TECHNOLOGY in
Electronics & Telecommunication Engg.

School of Electronics and Telecommunication Engineering

MIT Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

Alandi (D), Pune – 412105

(2024–2025)

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

CERTIFICATE

It is hereby certified that the work which is being presented in the BTECH Project Report entitled **"Recognition of Brahmi Script and Its Variant Using ML"**, in partial fulfillment of the requirements for the award of the Bachelor of Technology in Electronics & Telecommunication Engg.and submitted to the **School of Electronics and Telecommunication Engineering** of **MIT Academy of Engineering, Alandi(D), Pune, Affiliated to Savitribai Phule Pune University (SPPU), Pune**, is an authentic record of work carried out during Academic Year **2024–2025**, under the supervision of **Prof. Vinayak Kulkarni**, **School of Electronics and Telecommunication Engineering**

Gauri Kardekar	**(Exam Seat No. 202101040137)**
Gargi Gundawar	**(Exam Seat No. 202101070184)**
Shrawani Chaudhari	**(Exam Seat No. 202101060009)**
Sakshi Deshmukh	**(Exam Seat No. 202101070114)**

Prof. Vinayak Kulkarni	**Dr.Gayatri Ambadkar**	**Dr. Dipti Y. Sakhare**
Project Advisor	**Project Coordinator**	**Dean**

Director/Dy. Director(AR)

External Examiner

DECLARATION

We the undersigned solemnly declare that the project report is based on our own work carried out during the course of our study under the supervision of **Prof. Vinayak Kulkarni**.

We assert the statements made and conclusions drawn are an outcome of our project work. We further certify that

1. The work contained in the report is original and has been done by us under the general supervision of our supervisor.
2. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this Institute/University or any other Institute/University of India or abroad.
3. We have followed the guidelines provided by the Institute in writing the report.
4. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Gauri Kardekar	**(Exam Seat No. 202101040137)**
Gargi Gundawar	**(Exam Seat No. 202101070184)**
Shrawani Chaudhari	**(Exam Seat No. 202101060009)**
Sakshi Deshmukh	**(Exam Seat No. 202101070114)**

Abstract

The purpose of this project is to use AI/ML methods to recognize the Ancient Brahmi script automatically. The historical Brahmi script, considered one of the earliest writing systems in the Indian subcontinent, causes problems due to its difficult characters and the deterioration of historical artifacts. Our method requires compiling a specialized collection of Ancient Brahmi inscriptions in various styles and contexts. We want to increase transcription accuracy by combining linguistic and contextual knowledge. This AI/ML-based solution not only helps scholars and linguists read and translate Ancient Brahmi scripts more successfully, but it also adds to the preservation and comprehension of this priceless cultural heritage.

Acknowledgment

We want to express our gratitude towards our respected project guide Prof.Vinayak Kulkarni for his constant encouragement and valuable guidance during the completion of this project work. We also want to express our gratitude towards respected School Dean Dr. Dipti Sakhare for her continuous encouragement.

We would be failing in our duty if we do not thank all the other staff and faculty members for their experienced advice and evergreen co-operation.

202101040137 Gauri Kardekar

202101070184 Gargi Gundawar

202101060009 Shrawani Chaudhari

202101070114 Sakshi Deshmukh

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The ancient Brahmi script is the ancestor of many contemporary Indian scripts, including Bangla, Devanagari, and Gurmukhi. Numerous historical artifacts from the Indian subcontinent contain this script, which is regarded as the mother of many Indian writing systems. Important locations with Brahmi-written inscriptions, such as the Ashoka Pillars and Edicts, Ellora Caves, Ajanta Caves, and Bhattiprolu Stupa, offer priceless insights into early Indian civilization.

Inscriptions in Brahmi script are as old as the Mauryan Empire, which existed in the third century BCE. Deciphering these inscriptions is difficult, though, due to the age of the script and the degradation of many artifacts. More effective, precise recognition techniques are required because manual transcription attempts have not been able to keep up with the intricate characters and faded scripts over time.

With advancements in machine learning and image processing, there is now potential to automate the recognition of Brahmi script. We can greatly increase transcription accuracy by using AI techniques to process and analyze images of Brahmi inscriptions. In order to preserve and make this ancient cultural legacy more widely accessible, this project intends to use machine learning to identify and translate Brahmi script into contemporary scripts like Devanagari.

1.2 Project Idea

The project's goal is to automatically recognise old Brahmi script using AI and ML. Because of the deteriorating ancient artefacts and intricate characters, this script can be challenging to read.A machine learning model is trained on an exclusive dataset of historical Brahmi inscriptions created as part of the research.The model achieves high accuracy in Brahmi script recognition, even in cases where the inscriptions are blurry or damaged. This effort is transforming the reading and transcription of ancient Brahmi inscriptions, helping to preserve these ancient letters.

1.3 Motivation

This effort is motivated by the growing interest in historical linguistics, archaeology, and cultural conservation. Our motivation stems from utilising technology to decipher ancient Brahmi script. Computer model will be trained to recognize these characters, which will save time, provide historical context, and make this information available to everyone. Our mission is to protect our legacy while making it easier to interpret and appreciate these ancient writings.

1.4 Project Challenges

1. **Developing Dataset**: Developing a comprehensive dataset that included Brahmi script images and their corresponding Devanagari characters was challenging due to the unavailability of annotated resources. This process required manual labor to label and match the inscriptions with accurate translations.

2. **Unavailability of Standard Datasets**: No standardized or widely accessible dataset existed for the recognition of Brahmi script. As a result, it was more difficult to train the model efficiently and compare its results to those of other methods or studies.

3. **Preprocessing Variability**: It took considerable adjustment to preprocess the photos in order to improve clarity and standardize various styles, lighting

scenarios, and resolutions. Variability was introduced by inconsistent preprocessing, which also impacted recognition accuracy.

4. **Character Similarities**: The Brahmi characters were difficult to differentiate from one another because of their visual similarity. Gaining high recognition accuracy required the model to learn subtle variations.

5. **Computational Resources**: It took a lot of time and computing resources to train big machine learning models for image recognition, particularly for fine-tuning and optimization.

1.5 Proposed Solution

1. **Data Collection:**
    - Gather pictures of Ancient Brahmi characters from various sources.
    - Ensure high-quality images in a variety of styles and settings.
    - Arranged field visits to Aihole, Karnataka, India and Hampi, Karnataka, India in order to gather photos of Brahmi scripture from historic temples and monuments.

2. **Data Enhancement:**
    - Use preprocessing techniques for images to improve their quality and clarity.

3. **Feature Extraction and Character Identification:**
    - Identify the most important traits and distinctive qualities of Brahmi characters by applying algorithms.

4. **Model Selection:**
    - Select an appropriate AI/ML model for picture recognition.

5. **Training and Testing:**

- Utilizing the prepared dataset, train the selected model with an emphasis on character recognition.
- Assess the performance of the model with a validation dataset.

6. **Model Refinement:**
   - Collect and incorporate feedback from preliminary tests.
   - Based on feedback and testing outcomes, adjust the model's parameters and the training procedure.

7. **User Interface Development:**
   - Provide an easy-to-use software interface for inputting character images and obtaining recognition outcomes.

8. **Continuous Improvement and Deployment:**
   - Using the user interface, deploy the first model to collect data from the real world.
   - Gather user feedback on a regular basis to pinpoint problems and potential areas for development.
   - Update the model regularly with updated information and user-provided ideas.

9. **Documentation and Outreach:**
   - Make thorough documentation that describes the model, technique, and usage guidelines.
   - Communicate the project's significance, methods, and outcomes to the academic and cultural preservation groups.

1.6 Major Contribution

This project made significant contributions to the field of ancient script recognition using machine learning techniques. The key contributions are as follows:

1. **Creation of a Unique Brahmi-Devanagari Dataset**: The creation of an extensive and annotated dataset featuring pictures of Brahmi script and the equivalent Devanagari characters was one of the main contributions. Because there aren't many publicly accessible resources for Brahmi script recognition, this dataset closed a significant gap.

2. **Development of a Machine Learning Model for Brahmi Script Recognition**: Despite incomplete, unclear, or damaged inscriptions, a deep learning-based machine learning model was trained to identify Brahmi characters with excellent accuracy. The transcription of ancient script, which was previously a laborious manual procedure, was automated thanks to this methodology.

3. **Improved Preprocessing Techniques**: The effort greatly improved the model's recognition skills by introducing reliable picture preprocessing techniques to deal with noise, distortions, and the different quality levels of ancient inscriptions.

4. **Integration of Brahmi to Devanagari Conversion**: By effectively incorporating the automatic conversion of known Brahmi letters into contemporary Devanagari script, the initiative made it simpler to analyze and comprehend old literature.

5. **User-Friendly Interface for Script Recognition**: To enable non-expert users to upload pictures of Brahmi inscriptions and obtain precise transcriptions in Devanagari, a user interface was created. Because of this, linguists, scholars, and historians who lacked technical knowledge of machine learning could now use the technology.

6. **Contribution to Cultural Preservation**: The project helped to preserve and revitalize old Indian history by automating the recognition and transcription of Brahmi script, making these texts more accessible to the general public and future generations of researchers.

1.7 Project Report Organization (Chapter wise summary)

The report is structured as follows:

Chapter 2: Literature Review

With an emphasis on Brahmi script, this chapter provides a thorough analysis of relevant research in the subject of ancient script recognition. It discusses the difficulties encountered in the discipline, the approaches taken by earlier researchers, and current frameworks or systems. The project's background and context are established by the literature analysis, which also highlights the gaps that the current research attempts to fill.

Chapter 3: Problem Definition and Scope

This chapter outlines the project's specific focus, which is the use of machine learning techniques to recognize and translate Brahmi writing into Devanagari. It provides an overview of the project's objectives, limits, and scope. The issue is presented in light of historical script recognition and its significance for linguistic study and cultural preservation.

Chapter 4: System Requirement Specification

The functional and non-functional needs of the system created for this project are covered in this chapter. It outlines the necessary hardware and software, such as the deep learning framework, image processing software, and processing power. The intended performance standards, including accuracy levels, system scalability, and user interface parameters, are also covered in this chapter.

Chapter 5: Methodology

This chapter provides a detailed explanation of the development process for the Brahmi script recognition system. This covers the methodology for creating the dataset, the preprocessing methods used to improve the quality of the images, the selection of the machine learning model (e.g., CNN, deep learning models), and the training and assessment processes. With a focus on the crucial choices made along the way, the chapter describes the complete workflow from data collection to model training.

Chapter 6: Implementation

The system's actual implementation is covered in detail in this chapter. Data preprocessing, model design, training, and testing are among the technical and coding facets of model development that are covered. Additionally covered are the user interface's development and the system architecture. The practical procedures used to convert the theoretical framework into a workable solution are explained to readers in this chapter.

Chapter 7: Result Analysis

The system's performance outcomes, including evaluation metrics like accuracy, precision, recall, and F1-score, are presented in this chapter. It offers a thorough evaluation of the system's performance on the dataset of Brahmi scripts and contrasts the outcomes with those of other approaches or standards. The chapter also addresses the model's limitations and any difficulties that arose during testing, providing recommendations for enhancements in light of the results.

Chapter 8: Conclusion

The project's major contributions are outlined in the last chapter, which also considers whether the research goals were achieved. It talks about the project's possible uses in cultural preservation and gives a summary of how it has affected the field of ancient script recognition. The chapter ends with suggestions for further research, such growing the dataset, enhancing model performance, or looking into new uses for the system that has been established.

References and Appendices

All of the references cited in the report are included in this section. It offers a thorough inventory of all the books, articles, research papers, and other materials used in the study. The appendices section contains extra information that supports the report's key points, including data samples, code snippets, and supplemental material.

Chapter 2

Literature Review

2.1 Related work And State of the Art (Latest work)

Title of Publication	**Name of Author**	**Summary**
Analysis of segmentation methods for Brahmi script.	Dr. Ajay P. Singh and Ashwin.	Discusses recognition and segmentation of Brahmi inscriptions using image processing techniques. Focuses on challenges like noise, skew, and overlapping text. Highlights the need for character segmentation in ancient Brahmi script. Projection profile-based method for segmenting lines, words, and characters in Brahmi script.

Title of Publication	**Name of Author**	**Summary**
Handwriting recognition of Brahmi script based on PALI language.	Aniket Suresh Nagane, Chandrashekhar Himmatrao Patil, Shankar Mali.	Focuses on recognizing Brahmi script using geometrical features and SVM learning. Addresses the lack of prior work and a standard dataset. Uses HOG features for gradient orientation and SVM for character recognition. Emphasizes the need for improved algorithms for recognizing ancient Indian scripts.

Title of Publication	Name of Author	Summary
Recognition of Brahmi words by Using Deep Convolutional Neural Network.	N. Gautam, S. S. Chai, and Jais Jose.	The document focuses on character image identification using deep convolutional neural networks (DCNN), specifically for Brahmi words. Character splitting is unnecessary in this case since Brahmi characters are segregated. Preprocessing and distinctive traits are used to improve recognition accuracy and efficiency. The benefits listed include improved accuracy, more efficient recognition procedure, the potential for broader applications, and a meaningful contribution to field research. However, there is a lack of a common data set for benchmarking, as well as difficulties about determining the layout of neural network layers.

Title of Publication	**Name of Author**	**Summary**
The Dataset for Printed Brahmi Word Recognition.	Neha Gautam, Soo See Chai, Megha Gautam.	The document stresses the importance of datasets in recognition systems, particularly for accurate Brahmi word recognition. Due to resource limits, the proposed process entails gathering and organising word samples, as well as manually segmenting characters. Despite its shortcomings, the Brahmi word collection has the potential to improve script recognition in languages such as Hindi and Marathi. The lack of standardisation, complex characters, limited current use, and a lack of recognition tools are among the challenges, signalling prospects for future automation and dataset extension.

Title of Publication	**Name of Author**	**Summary**
Review of Character Recognition technique for Modi Script.	The name of the author is Patil P.A. .	The research looks into character recognition in Marathi's historically significant Modi script. Various strategies are used to segment characters based on pixel density in the algorithm. Despite its traditional significance, Modi Lipi confronts obstacles such as limited modern use and intricacy, which affects practical communication. According to the study, CNN and SVD are the most accurate recognition algorithms, with 99.78 percent and 99.56 percent accuracy, respectively. Variations in handwriting styles cause problems with standardisation and recognition.

Title of Publication	Name of Author	Summary
Machine Learning Algorithms for Handwritten Devanagari Character Recognition.	Mimansha Agrawal,Bhanu Chauhan,Tanisha Agrawal.	The study focuses on using machine learning, specifically methods and formulas established by Bhanu, Devangan, and Agrawal, to recognize handwritten characters more accurately and efficiently. It discusses the difficulties in detecting handwritten characters and the authors' approaches and algorithms for overcoming these difficulties. The research digs into specific machine learning methodologies, with the possibility of investigating neural network topologies and deep learning models. It may also address the algorithms' practical applications in areas such as automated data entry, document digitization, and optical character recognition. The paper's overarching goal is to develop machine learning-based handwriting recognition solutions.

Title of Publication	Name of Author	Summary
Handwritten Character Recognition from Images using CNN - ECOC.	Mayur Bhargab Bora, Dinthisrang Daimary, Khwairakpam Amitab , Debdatta Kanda.	For handwritten character recognition, the paper's methodology combines Convolutional Neural Networks (CNN) with Error-Correcting Output Codes (ECOC). When comparing CNN architectures (LeNet, AlexNet, and ZfNet), AlexNet shines, and when paired with ECOC, it improves accuracy over typical CNN softmax approaches. The researchers trained four CNN architectures with an ECOC classifier on the NIST handwritten character dataset, with the goal of improving CNN character recognition accuracy with ECOC. The best integration of AlexNet was noticed, as well as the higher performance of the ECOC classifier over CNN softmax. By extracting pertinent information and improving discrimination, the combination of CNN and ECOC enhances recognition accuracy.

Title of Publication	**Name of Author**	**Summary**
Segmentation of Characters from Degraded Brahmi Script Images.	Aniket suresh Nagane and Shankar Mali from MIT Arts, Commerce and Science College and Dr. Vishwanat h Karad MIT World Peace University Pune.	The research describes a two-stage mechanism for character segmentation in degraded photos of the Brahmi script: line segmentation and character segmentation. Character segmentation recognizes characters within lines and categorizes them by size based on pixel area, with an accuracy of roughly 90 percent. Line segmentation separates the material into lines to create content structure, highlighting the need of good segmentation. The algorithm employs erosion-based noise removal for noise reduction and character separation, and it has proven effective across a wide range of image qualities. However, there are certain downsides, such as uneven character separation caused by erosion and limited improvement on complicated characters.

Title of Publication	Name of Author	Summary
Brahmi Script Classification using VGG16 Architecture Convolutional Neural Network.	Vincen and Samsuryadi.	This study employs the VGG16 CNN architecture for classifying Brahmi script characters. The process includes preprocessing steps like zero-padding, resizing, and grayscale conversion, followed by data augmentation. The model achieves high accuracy (up to 98%) for classification, with metrics such as precision and recall demonstrating robust performance. The dataset preparation and training strategies highlight improvements in recognition accuracy through data augmentation and hyperparameter tuning.

Title of Publication	**Name of Author**	**Summary**
Classification of Brahmi Script Characters using HOG Features and Multiclass ECOC Model with SVM Binary Learners	Aniket S. Nagane; Chandrashekhar H. Patil; Shankar M. Mali	This study addresses the recognition of Brahmi script characters using HOG (Histogram of Oriented Gradients) for extracting distinctive features from character images. For classification, it employs a multiclass ECOC (Error-Correcting Output Code) framework integrated with SVM binary learners, efficiently handling the complexities of multi-class script recognition. The approach overcomes challenges like varying writing styles and intricate character structures, demonstrating significant improvements in recognition accuracy and efficiency.

Title of Publication	**Name of Author**	**Summary**
Deep Learning-powered Mobile App for Early Brahmi Script Decipherment in Sri Lanka	Sakith Gunasekara; Muhammed Haleef Lafir; Chavindu Dulaj; Lakidu Haputhanthri; Dileeka Alwis	This paper introduces a mobile application powered by deep learning to facilitate the decipherment of early Brahmi scripts found in Sri Lanka. The application employs advanced neural network models to analyze and recognize ancient Brahmi characters, enabling users to interact with and understand this historic script. By combining accessibility with sophisticated AI-driven recognition, the app seeks to make Brahmi script more comprehensible and usable for both scholars and the general public. This innovative approach bridges technology and heritage, contributing to the preservation and understanding of ancient texts.

Title of Publication	Name of Author	Summary
Challenges and Opportunities in Brahmi Script Recognition using Artificial Intelligence	N. Babu; S. Wickramarathna	This paper provides an in-depth review of the challenges associated with Brahmi script recognition, including the highly diverse writing styles, fragmented historical data, and absence of standardized datasets. It emphasizes how artificial intelligence techniques, such as Convolutional Neural Networks (CNNs) for image-based character recognition and Recurrent Neural Networks (RNNs) for handling sequential dependencies, have significantly advanced the accuracy and efficiency of Brahmi script recognition systems. The research highlights the potential of these AI methods to address complex script structures, enabling more robust and scalable solutions for deciphering and analyzing this ancient writing system.

Title of Publication	Name of Author	Summary
Efficient Scalable Template-Matching Technique for Ancient Brahmi Script Image	Kaur, Sandeep; Bharat Bhushan Sagar	The multi-stage template-matching system for Brahmi script identification presented in this research uses area/perimeter-based matching and SIFT-based feature extraction. It improves recognition accuracy across a range of image qualities by using a two-stage pipeline to filter out irrelevant photos and fine-tune matching using multi-scale approaches. In order to maximize character recognition, the dataset is chosen by online scraping and analyzed using a similarity measure system. By tackling problems with image quality and character representation unpredictability, this method greatly improves the recognition of intricate Brahmi characters.

The literature review highlights how Brahmi script recognition algorithms have advanced from conventional segmentation and feature extraction methods to state-of-the-art deep learning models. Recent developments in deep convolutional neural networks (DCNN) have shown themselves to be more successful in managing the intricacies of ancient scripts, even though older techniques like SVMs and geometrical feature extraction were still quite important. Research using CNN and VGG16 architectures has shown better performance and accuracy in identifying and categorizing Brahmi characters. CNN use has continuously surpassed conventional approaches when paired with strong preprocessing and data augmentation strategies. CNN provides a highly accurate and scalable solution for this challenging problem, making it the most efficient technique for identifying and translating Brahmi characters to Devanagari.

2.2 Limitation of State of the Art techniques

- Lack of standardized datasets
- High variability in character shapes
- Low accuracy in complex inscriptions
  One of the major challenges in the process for recognizing and translating Brahmi characters to Devanagari is the absence of consistent databases. Effective model training is hampered by this constraint since the data may not fully capture the variances in Brahmi script. Furthermore, there is a great deal of variation in character shapes, which makes recognition much more difficult. It is difficult for models to generalize since Brahmi characters frequently seem differently depending on regional variances, handwriting styles, and the quality of inscriptions. Furthermore, the complexity of the script and the deteriorated condition of many old texts—where characters may be fading or incomplete—make it challenging to achieve high accuracy in complex inscriptions. All of these elements work against conventional recognition techniques' overall effectiveness.

2.3 Discussion and future direction

Future work should focus on dataset creation and exploring Transformer models for improved recognition.

2.4 Concluding Remarks

The analysis of previous studies and approaches for Brahmi script recognition reveals the major difficulties brought on by the script's complexity and the absence of standardized datasets. Because of the distinctive characteristics of its letters and the variations in inscriptions brought on by deterioration over time, little research has been done on ancient scripts like Brahmi, despite advancements in optical character recognition (OCR) for contemporary scripts.

The state-of-the-art methods currently in use, such as more recent deep learning models like Convolutional Neural Networks (CNNs) and more conventional machine learning techniques like Support Vector Machines (SVM), show promise in resolving these problems. The lack of an extensive dataset and the requirement for efficient preprocessing methods, however, continue to be significant obstacles.

By developing a customized dataset and utilizing deep learning techniques for increased accuracy, this initiative seeks to close the gap in the body of existing research. In order to provide a useful tool for historians, archaeologists, and scholars interested in the preservation of cultural heritage, this work aims to advance the field of ancient script recognition by resolving the limitations noted in the literature.

Chapter 3

Problem Definition and Scope

3.1 Problem statement

To recognize Brahmi and its Variant Using Machine Learning

3.2 Goals and Objectives

1. Collect and preprocess data from multiple sources.

2. Using discriminative feature extraction approaches, extract and represent distinguishing aspects of Brahmi characters from pre-processed photos.

3. Select a relevant machine learning model for training on labelled images in order to predict character labels effectively.

3.3 Scope and Major Constraints

This project is all about using computers to create a smart system that can comprehend and recognise ancient Brahmi text.

3.4 Software Requirements

Software Specification:

1. Visual Studio - Visual Studio 2022 Version 17.8. Released: Nov,2023.

2. Python Library - Python Version 3.13. Released: May,2023.

3. Scikkit Image - Scikit-image Version 0.22.0. Released: Oct, 2023.

4. Sci Learn - scikit-learn version 1.3.2. Released: Oct, 2023.

3.5 Methodology

1. Data Collection: Gather a comprehensive dataset of high-resolution Brahmi character images.

2. Feature extraction: Identify and extract unique features of each character using advanced algorithms.

3. Model Selection: Employ a convolutional neural network (CNN) for effective image recognition.

4. Training: Train CNN with the dataset, allowing it to learn and recognize character patterns.

5. Evaluation: Test the trained model with new images to assess its recognition accuracy.

6. User Interface: Create a user-friendly interface for easy image input and character identification.

7. Continuous Improvement: Regularly update the model with feedback and new images to improve accuracy.

3.6 Expected Outcomes

When the image of Brahmi script or carvings is uploaded, the Brahmi script is converted to Devanagari and the result is obtained.

Chapter 4

System Requirement Specification

4.1 Block diagram/ Proposed System setup

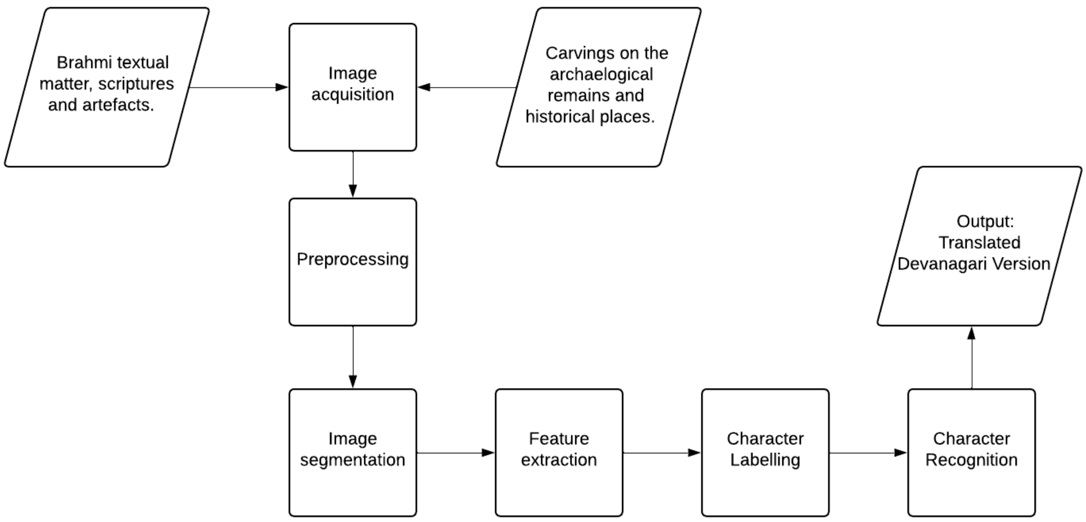

Figure 4.1: Block Diagram

4.1.1 Block Diagram Description

1. **Collecting Textual Carvings and Brahmi Script:** Gathering a varied collection of pictures with Brahmi writing is the first stage. This consists of Brahmi characters, manuscript scans, and old inscriptions. A diversity of examples ensures the recognition system can manage the variations in writing styles across different historical periods and regions.

2. **Segmenting Images to Identify Script Components:** To separate the important parts of the script, the images are split in this phase. Characters, diacritical marks, and other important elements of the Brahmi text may fall under this category. Segmentation is accomplished using methods such as thresholding, contour detection, or deep learning-based approaches (e.g., Convolutional Neural Networks).

3. **Categorizing the Text Components:** After segmentation, the constituent parts are divided into discrete groupings, including modifiers, ligatures, vowels, and consonants. For proper recognition and conversion into Devanagari, this classification is essential. For this job, machine learning methods like as classification algorithms are employed.

4. **Component Recognition and Devanagari Conversion:** The following stage is the recognition phase, in which the Devanagari equivalents of the Brahmi components that have been recognized are compared. A machine learning model that has been trained to identify the segmented text components is used in this procedure. Convolutional Neural Networks (CNNs), a deep learning-based architecture, are one example of the model that has been trained on a dataset that includes known character mappings between Brahmi and Devanagari scripts. The model can identify the Brahmi components and convert them to the appropriate Devanagari symbols by using patterns discovered in the training data.

5. **Displaying the Output:** A textual and visual depiction of the finished product is displayed in Devanagari script. To increase recognition accuracy, the system might have a feedback mechanism that enables manual adjustments or model retraining. The output is made to maintain the original Brahmi script's structural integrity.

4.2 Software Design tools

4.2.1 Use Case Diagram

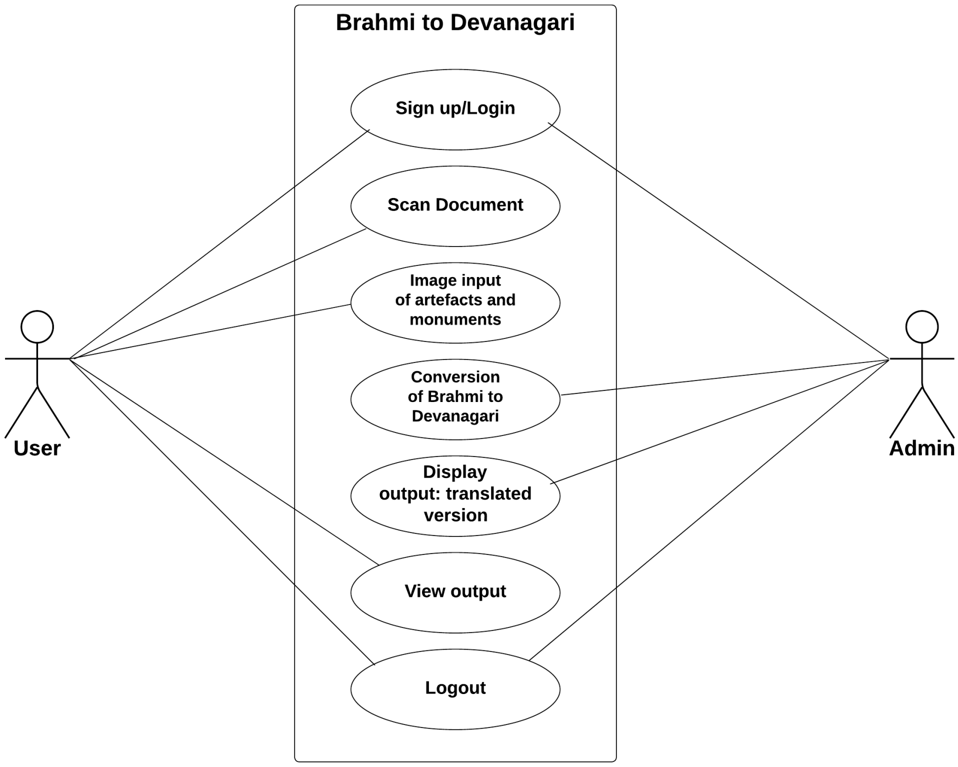

Figure 4.2: Use Case Diagram

Use Case Diagram Description

User:

Sign Up: Users can create a new account by entering the required information.

Log in: Once registered, users can use their credentials to log into the system.

Document Scanning: Users have the ability to scan documents.

Artifact and Monument Image Input: Users can enter images of artefacts and monuments into the system.

View Output: Users can see the processed output as well as information about scanned documents and input photos.

Logout: Users can log out of their accounts, effectively ending the session.

Admin:

Sign Up: Administrators can create accounts by entering the required information.

Conversion of Brahmi to Devanagari: The admin has the power to convert text from the Brahmi script to the Devanagari script. Admins can inspect and show the processed output or information resulting from the conversions.

Logout: Admins can log out of their accounts to securely end the session.

4.2.2 Data Flow Diagram for future

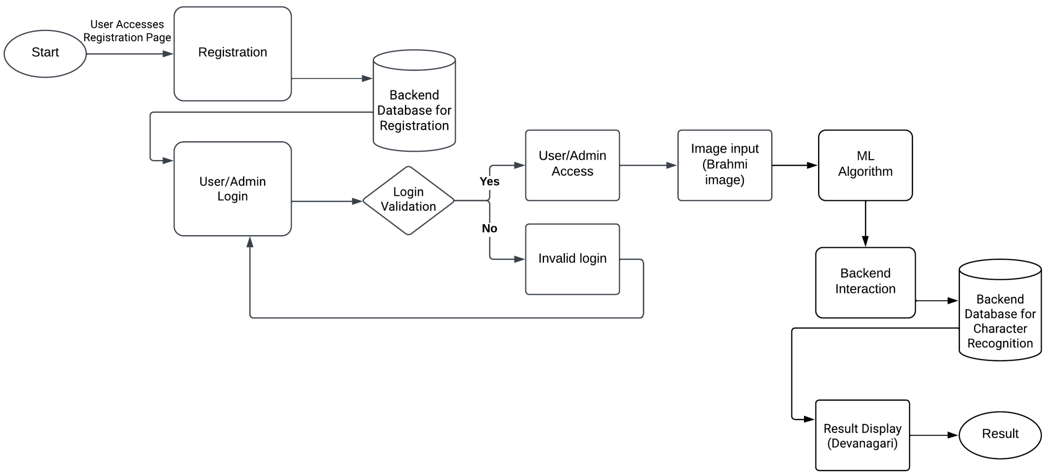

Figure 4.3: Data Flow Diagram

Data Flow Diagram Description

1. Start:

a) The procedure begins with the user accessing the website registration page.

2. Registration:

a) The user enters registration credentials.

b) The registration data is stored in the backend database.

3. Log in:

a) Users and admins log in using their credentials.

b) The login information is verified against the database's stored data.

4. User and Admin Login:

a) Users/admin obtain access to the system after successfully logging in.

5. Image input

a) Through a scanning procedure, users give input photos containing Brahmi script.

6. Machine Learning Algorithm:

a) To recognize the Brahmi script in Devanagari from the dataset, apply a machine learning algorithm.

b) To identify the script, the method use a pre-trained model or a dataset.

7. Backend Interaction:

a) Using backend software/API, the identified data is delivered to the backend.

b) The data are processed by the backend and stored in the database.

8. Results display:

a) The result (Brahmi script recognized in Devanagari) is retrieved from the back-end.

b) The final result is displayed to the user on the website.

9. End:

a) The process ends.

4.3 Software Requirements

Software Specification:

1. Visual Studio - Visual Studio 2022 Version 17.8. Released: Nov,2023.

2. Python Library - Python Version 3.13. Released: May,2023.

3. Scikkit Image - scikit-image Version 0.22.0. Released: Oct, 2023.

4. Sci Learn - scikit-learn version 1.3.2. Released: Oct, 2023.

4.4 Project Planning

Phase 1: Data Collection and Preparation

1. Collect Images: - Gather diverse, high-quality Brahmi character images. - Ensure a variety of styles and conditions.

2. Enhance Data: - Preprocess images to improve clarity and quality.

Phase 2: Model Development

3. Extract Features: - Implement algorithms to identify key features of Brahmi characters.

4. Select Model: - Use a suitable AI model like Convolutional Neural Networks (CNNs).

Phase 3: Training and Testing

5. Train Model: - Train the model with the prepared dataset for character recognition.

6. Validate and Test: - Evaluate model performance with a validation dataset.

Phase 4: Model Refinement

7. Incorporate Feedback: - Use initial test feedback to improve the model.

8. Fine-Tune: Adjust model parameters based on feedback.

Phase 5: User Interface Development

9. Create Interface: - Develop a software interface for character image input and recognition.

Phase 6: Continuous Improvement and Deployment

10. Deploy and Learn: - Deploy the model and gather real-world data.

11. Collect Feedback: Use user feedback for further improvement.

12. Update model: - Regularly update the model with new data.

Phase 7: Documentation and Outreach

13. Document: - Create detailed methodology and usage documentation.

14. Share results: - Disseminate findings to academic and cultural communities.

Semester Plan:

Semester V:

1. Review literature on Brahmi script recognition.

2. Collect Brahmi script images to create a dataset.

3. Implement a baseline recognition system.

Semester VI:

1. Enhance the baseline system.

2. Single Brahmi Character recognition.

Semester VII:

1. Translate Brahmi script to Devanagari.

2. Test on diverse datasets.

Chapter 5

Methodology

5.1 Introduction

To create a mathematical model for recognizing Brahmi script and its variants using the CNN model, we can break down the steps using the uploaded image (which resembles the Brahmi letter 'ba') into the following parts:

5.2 System Architecture

- **Image Collection:** Collect an adequate dataset of pictures of Brahmi characters from online archives, historical manuscripts, and inscriptions. Make sure there is variety in terms of sizes, styles, and orientations. Add artificial instances to the dataset to mimic rare or challenging-to-locate characters.
- **Image Enhancement:** Utilize image processing methods including scaling, denoising, and contrast enhancement to produce clear and uniform images. To improve dataset diversity, use data augmentation techniques including cropping, zooming, flipping, and rotation.

5.3 Mathematical Modeling

- **Feature Identification:** Analyze Brahmi characters to find distinctive features such as symmetry, curves, and strokes. To draw attention to important characteristics, use edge detection algorithms (such as Canny and Sobel).

5.3.1 Mathematical model for feature extraction in CNN

A feature map that highlights significant aspects of the image, like edges or certain character patterns, is the end result of performing convolution. Depending on padding, the convolution process shrinks the image size while preserving its key characteristics.

Detailed Steps for Feature Extraction in CNN

- **Image Matrix for 'ba':** First, the grayscale image of the Brahmi character 'ba' (uploaded image) is resized to 32x32 pixels. Let's assume this resized image $I(x, y)$ is represented as a matrix I, where each element corresponds to a pixel intensity value between 0 and 1 (after normalization).

  For example, if I is the 32x32 matrix of pixel values:

$$I = \begin{bmatrix} 0.0 & 0.1 & 0.3 & \dots & 0.5 \\ 0.2 & 0.4 & 0.1 & \dots & 0.0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.7 & 0.6 & 0.3 & \dots & 0.9 \end{bmatrix}$$

- **Filter Grids (Convolution Operation):** The first convolutional layer applies 32 filters of size 3x3. Each filter matrix, say F_k, slides over the image matrix I, performing element-wise multiplication and summing the results to generate a feature map.

For instance, if a 3x3 filter looks like this:

$$F = \begin{bmatrix} 0.2 & 0.5 & 0.3 \\ 0.1 & 0.4 & 0.2 \\ 0.7 & 0.3 & 0.6 \end{bmatrix}$$

The feature map $M(x, y)$ at position (x, y) is calculated as:

$$M(x, y) = \sum_{i=0}^{2} \sum_{j=0}^{2} I(x + i, y + j) \cdot F(i, j)$$

- **Feature Map for 'ba':** After applying the convolution, the result is a feature map that highlights important features of the image such as edges or specific patterns in the character. The convolution operation reduces the image size, depending on padding, but retains the essential features.

- **Activation Function (ReLU):** Next, the Rectified Linear Unit (ReLU) activation function is applied element-wise to the feature map:

$$\text{ReLU}(M(x, y)) = \max(0, M(x, y))$$

  This introduces non-linearity to the model, helping it learn more complex patterns. All negative values in the feature map are replaced with zero.

- **Pooling Layer:** After applying ReLU, a pooling layer is used to reduce the spatial size of the feature maps. Max-pooling with a 2x2 window and a stride of 2 selects the maximum value from each 2x2 region, significantly reducing the dimensionality while retaining the most important information.

  For example, if part of the feature map is:

$$\begin{bmatrix} 1.2 & 0.8 \\ 0.5 & 0.3 \end{bmatrix}$$

  The result after 2x2 max pooling would be 1.2, the maximum value.

- **Stack of Features:** The next layer would involve stacking these feature maps and repeating the process with a new set of filters (64 in this case). This enables the network to learn deeper hierarchical representations of the Brahmi character 'ba', capturing more complex patterns at higher levels of abstraction.

- **Mathematical Representation of Layers:**
  - **Input Image (I):** A 32x32 matrix of pixel intensities.
  - **Convolution Operation (M_k):** Each filter F_k produces a feature map M_k by convolving with the image:

$$M_k(x, y) = \sum_{i=0}^{2} \sum_{j=0}^{2} I(x + i, y + j) \cdot F_k(i, j)$$

  - **Activation (ReLU):** Apply ReLU to the feature map:

$$A_k(x, y) = \max(0, M_k(x, y))$$

  - **Pooling:** Apply max-pooling to reduce dimensionality:

$$P_k(x', y') = \max\{A_k(2x, 2y), A_k(2x+1, 2y), A_k(2x, 2y+1), A_k(2x+1, 2y+1)\}$$

5.4 Objective Function

- **Model Selection:** For picture recognition, use deep learning models such as Convolutional Neural Networks (CNNs). To recognize Brahmi characters, start with pre-trained models like VGG16, ResNet, or InceptionV3 and refine them.
- **Custom Architecture:** Design a CNN model that is tailored to the dataset by applying spatial hierarchies for feature extraction and transfer learning.
- **Loss Function:** For multi-class classification, use categorical cross-entropy to guarantee that characters are effectively distinguished from one another.

5.5 Approach/Algorithms

- **Data Preparation:** Divide the dataset into subsets for testing, validation, and training. To guarantee that no class is underrepresented, balance the data.

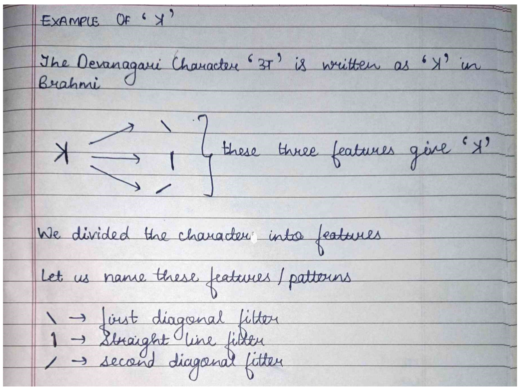

Figure 5.1: CNN example of 'a' in Brahmi

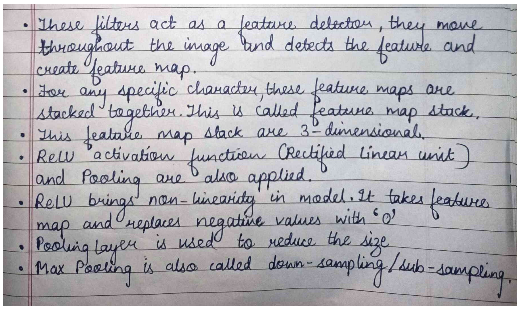

Figure 5.2: CNN example of 'a' in Brahmi

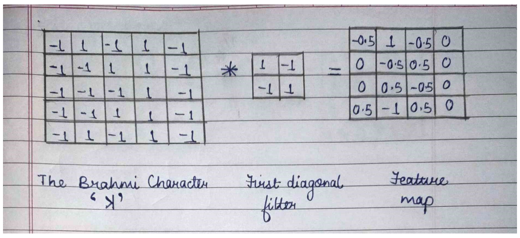

Figure 5.3: Feature maps formed by different filters

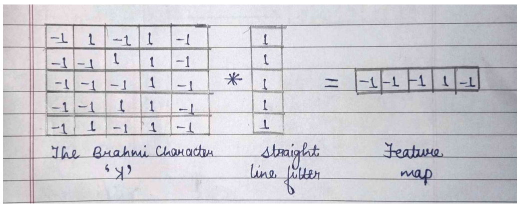

Figure 5.4: Feature maps formed by different filters

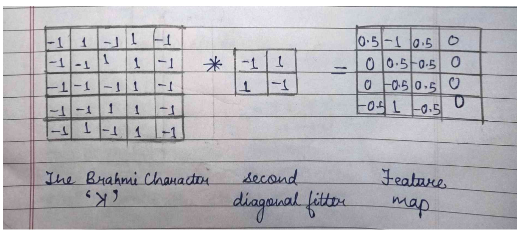

Figure 5.5: Feature maps formed by different filters

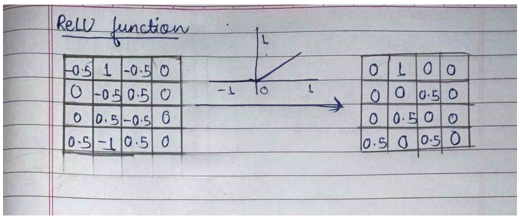

Figure 5.6: ReLU function and MaxPooling

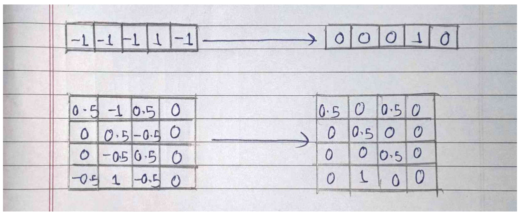

Figure 5.7: ReLU function and MaxPooling

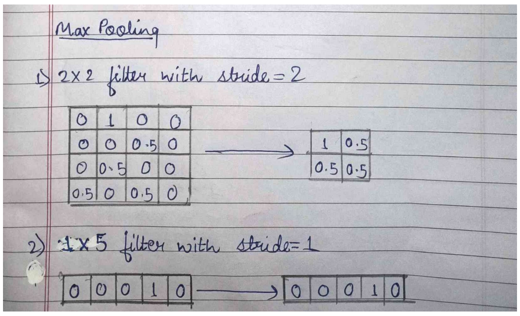

Figure 5.8: ReLU function and MaxPooling

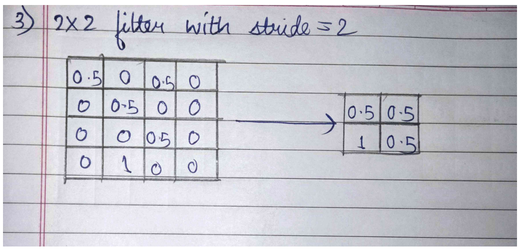

Figure 5.9: MaxPooling

- **Training:** Dropout techniques, learning rate scheduling, and data augmentation are used to train the model. Measures such as F1-score, recall, accuracy, and precision should be tracked.

- **Hyperparameter Optimization:** Perform grid or random search to tune hyperparameters such as learning rate, batch size, and number of layers.

- **Model Evaluation:** Use metrics such as accuracy, precision, recall, F1-score, and a confusion matrix to assess the model on unseen data.

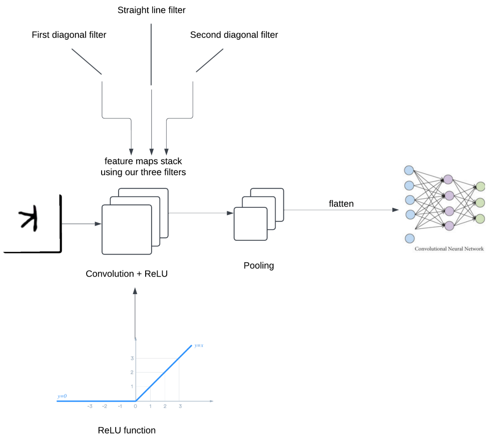

Figure 5.10: CNN Architecture for project

- **Robustness Testing:** To evaluate the robustness of the model, test it on photos that are deformed, noisy, or blurry. Examine incorrect categories to improve training.
- **User Interface Development:** Use frameworks to develop a graphical user interface (GUI) and use Flask to construct a web-based interface. Permit users to examine recognition results after uploading an image.
- **Real-Time Recognition:** Incorporate real-time processing skills to identify characters in live feeds or camera input.
- **Feedback Mechanism:** Allowing users to rectify inaccurate forecasts facilitates ongoing enhancement.
- **Documentation:** For clarity, document the complete process, including data collecting, preprocessing, model training, and testing.

5.6 Loss Function

- The categorical cross-entropy loss function, which calculates the dissimilarity between the anticipated and actual class probabilities, should be used to train the CNN model. For the Brahmi character 'ba,' assuming there are C classes and the correct class index is y, the loss for a single sample is:

$$\text{Loss} = -\sum_{c=1}^{C} y_c \log(\hat{y}_c)$$

Where y_c is the true probability (1 for the correct class, 0 otherwise) and $\hat{y}_c$ is the predicted probability for class c.

5.7 Optimization

- **Optimizer:** Utilize optimization algorithms such as Adam or SGD with momentum to reduce the loss function. For effective convergence, change the learning rate.

- **Weight Updates:** Adjust the weights θ of the CNN using backpropagation. For each weight θ_i, the update rule for the Adam optimizer is:

$$\theta_i \leftarrow \theta_i - \alpha \frac{\hat{m}_i}{\sqrt{\hat{v}_i} + \epsilon}$$

Where α is the learning rate, $\hat{m}_i$ is the biased-corrected first moment estimate, and $\hat{v}_i$ is the biased-corrected second moment estimate.

Chapter 6

Implementation

6.1 System Implementation

6.1.1 Template Matching

Step 1: Template Creation

1. Data Collection: Gather a large collection of Brahmi characters, making sure to include samples of different handwriting styles.

2. Template Generation: To use as templates, make binary pictures of every Brahmi character. The matching Devanagari character is used to label each image.

Step 2: Matching Process

1. Image preprocessing: To use as templates, make binary pictures of every Brahmi character. The matching Devanagari character is used to label each image.

2. Sliding Window: Move the template over the input image using a sliding window method, then compute the correlation coefficient at each location. Match Selection: For each character, choose the position with the highest correlation coefficient.

Step 3: Character Mapping

1. Character Recognition: Using a prepared dictionary, map the identified Brahmi character to its equivalent Devanagari character.

2. Output Generation: Create a cohesive Devanagari text by combining the identified characters.

Figure 6.1: Template matching in Brahmi script. *(Image: Brahmi script, output of the code executed. Source:snapshot of execution of our code*

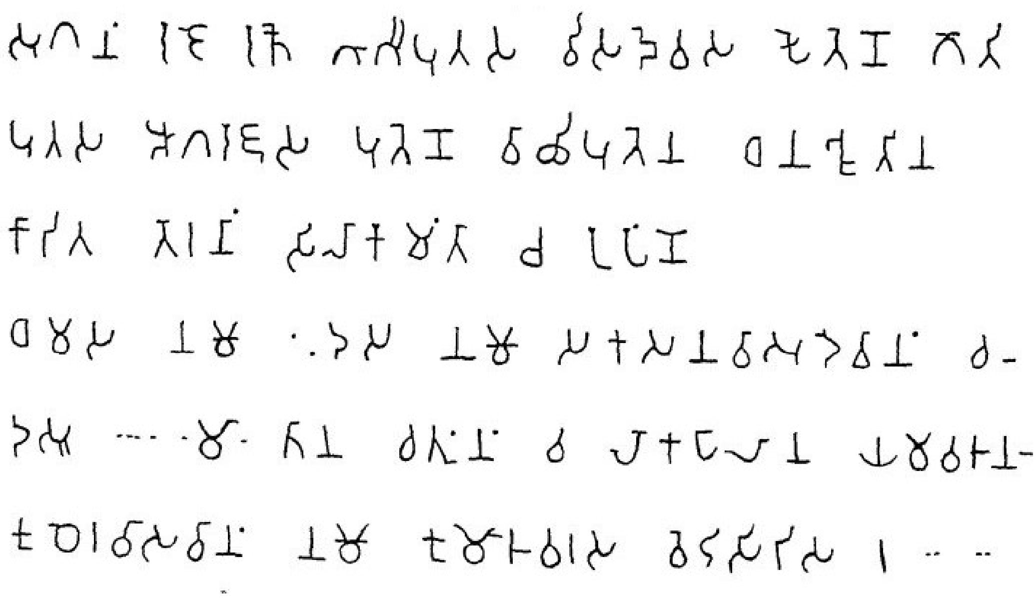

Figure 6.2: Test Image showing Brahmi script. *(Image: Brahmi script. Source: Design India Site. Website accessed on: 10 Oct 2024.)*

6.2 Experiment/Implementation Parameters

6.2.1 Character Recognition Optical

For projects like Brahmi Character Recognition, which transforms old Brahmi script into contemporary Devanagari letters, optical character recognition (OCR) technology is essential for transforming scanned documents into machine-encoded text. Brahmi script is not natively supported by popular OCR systems like Tesseract, despite its usefulness; hence, specialized solutions are required for precise conversion and preservation of old manuscripts and inscriptions. When we attempted to apply OCR to a Brahmi image, we found this. It did not yield any results for Brahmi, but it did work for the English text image.

Figure 6.3: English text used for application of OCR technique. *(Image: English text. Source: courtesy - Gargi.)*

6.2.2 Image Preprocessing

A number of procedures are involved in picture preprocessing in order to get the image ready for additional processing:

```python
img = cv2.imread("C:/Users/91766/Downloads/image.png")
```

```python
print(pytesseract.image_to_string(img))
```

```
Our project title is
Recognition and conversion of
Brahmi Script using ML.
Group Members are:
Shrawani Chaudhari
Gargi Gundawar
Gauri Kardekar
Sakshi Deshmukh
```

Figure 6.4: OCR result for English text. *(Image: English text output of OCR technique. Source: Output image of the code executed by us.)*

```python
print(pytesseract.image_to_string(img))
```

Figure 6.5: OCR result for Brahmi. (*Image: Brahmi text output of OCR technique. Source: Output image of the code executed by us.)*

Noise Removal

To improve the quality of the Brahmi script photographs, noise removal is a crucial step. It uses methods including bilateral filtering, median filtering, and Gaussian blur to eliminate extraneous noise while maintaining the edges.

Changing Brightness

Normalizing the lighting conditions can be achieved by adjusting the image's brightness. By giving each pixel a fixed value, this can be accomplished.

Figure 6.6: Noise removal of Brahmi script on Ashoka Pillar. *(Image: Brahmi script on Ashoka Pillar in Sarnath, c. 250 BCE. Source: Wikipedia. Website accessed on: 10 Oct 2024.)*

Changing Warmth

Altering the warmth improves the image's visual quality by shifting the color balance toward the red or blue end of the spectrum.

Changing to Grayscale

By reducing the RGB image's three color channels to a single intensity channel, grayscale conversion streamlines processing.

Changing the Contrast of the Image

By extending the intensity range, contrast enhancement helps the script's features stand out more.

6.2.3 Edge Detection

An image's boundaries can be found via edge detection. This can be done using a variety of operators:

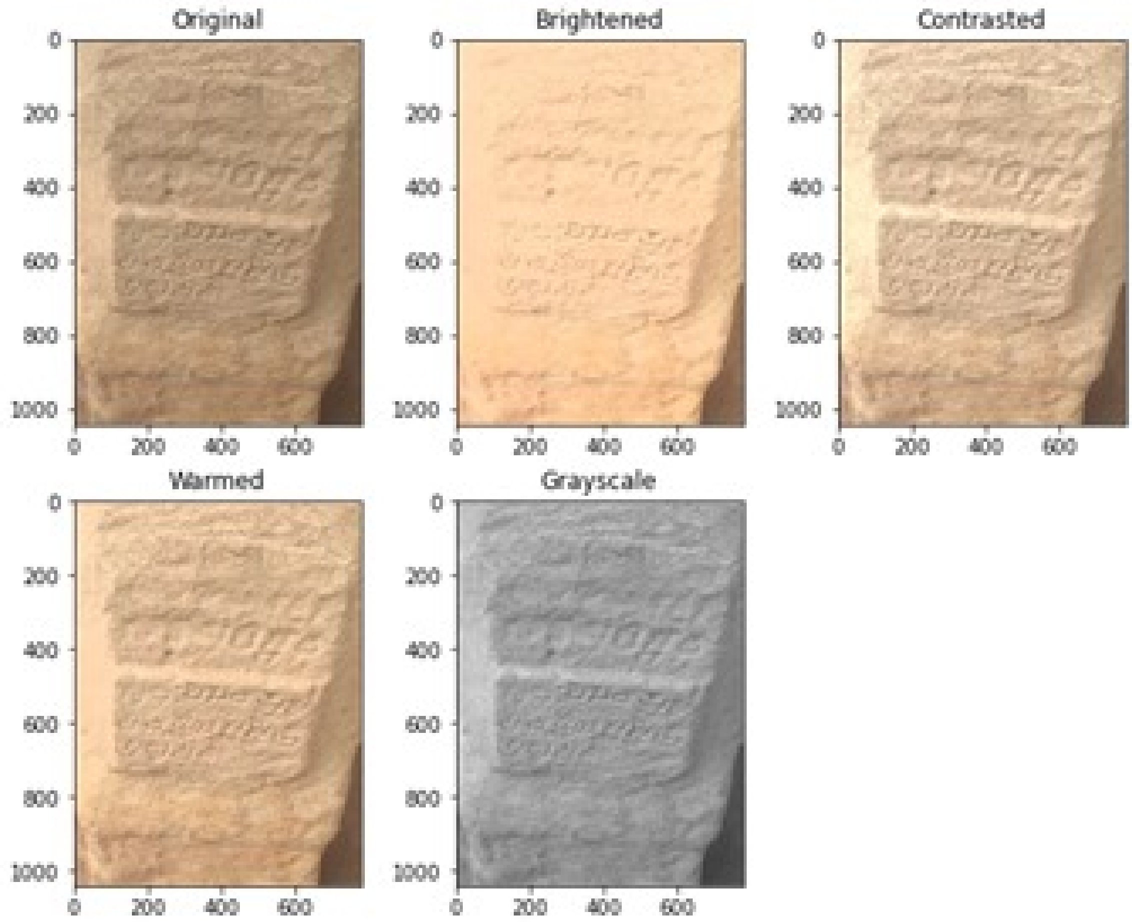

Figure 6.7: Image pre-processing techniques applied on Brahmi script from Aihole, Karnataka. *(Image: Brahmi script from Aihole, Karnataka. Source: captured by us.)*

Robert Operator

Two 2x2 convolution kernels are used by the Robert operator to detect edges. Although it can be susceptible to noise, it is fast and computationally efficient.

Prewitt Operator

In order to approximate the gradient of the image intensity, the Prewitt operator employs two 3x3 convolution kernels.

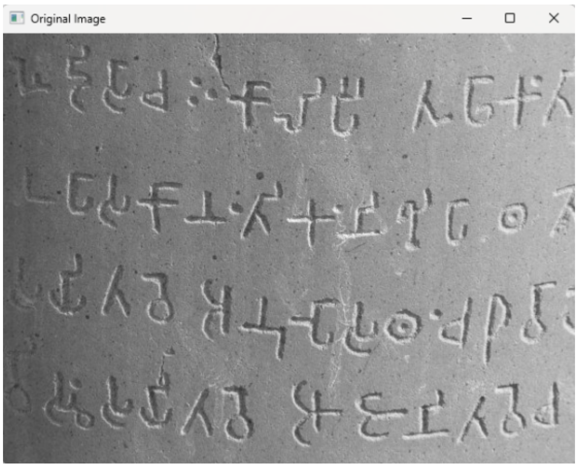

Figure 6.8: Original grayscale Brahmi script image of Ashoka Pillar in Sarnath for edge detection. *(Image: Brahmi script on Ashoka Pillar in Sarnath, c. 250 BCE. Source: Wikipedia. Website accessed on: 10 Oct 2024.)*

Sobel 3x3 Operator

In contrast to the Robert and Prewitt operators, the Sobel 3x3 operator offers superior noise resistance by computing the gradient magnitude using a pair of 3x3 kernels.

Sobel 5x5 Operator

With bigger kernels, the Sobel 5x5 operator expands on the Sobel 3x3 operator and can offer even better edge detection at the expense of additional processing power.

6.2.4 Shape Detection

Shape detection is the process of locating and classifying the image's geometric shapes. Lines, circles, and other shapes can be found using methods like contour analysis and the Hough transform.

6.2.5 High Pass Filter

By weakening the low-frequency elements of the image, a high pass filter is used to accentuate the high-frequency elements, like edges and fine details.

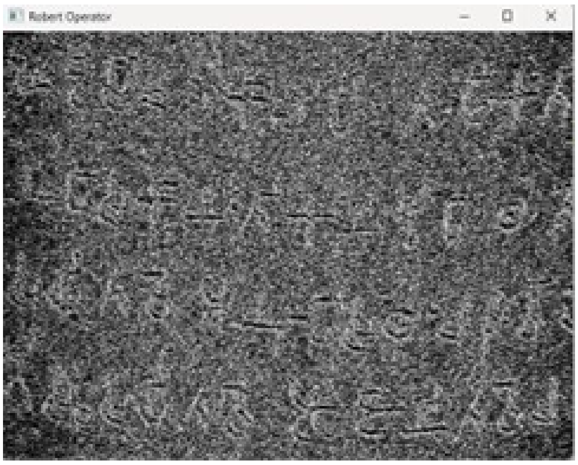

Figure 6.9: Robert's Operator applied on Brahmi script of Ashoka Pillar in Sarnath for edge detection. *(Image: Brahmi script on Ashoka Pillar in Sarnath, c. 250 BCE. Source: Wikipedia. Website accessed on: 10 Oct 2024.)*

6.2.6 Low Pass Filter

By weakening the high-frequency components, a low pass filter reduces noise and fine details in the image.

6.3 User Interface

The Brahmi character recognition system's User Interface (UI) is made to be simple to use and intuitive. Users can upload Brahmi script photos, have the system process them, and see the Devanagari characters that are recognized.

6.4 Data Description

Images of characters written in Brahmi script make up the dataset used for this research. The matching Devanagari character is used to label each image. To guarantee the model's resilience and generalizability, the dataset contains samples of handwriting in a variety of styles.

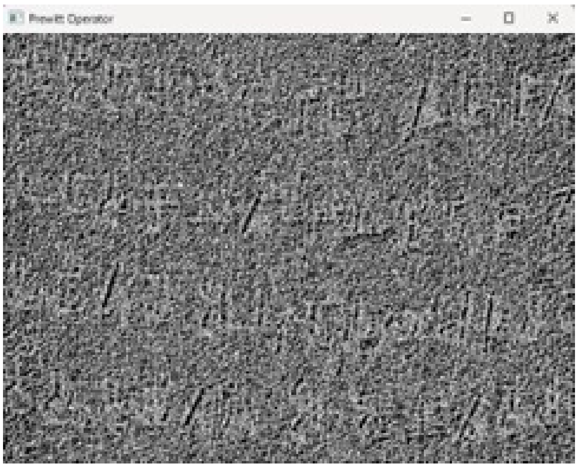

Figure 6.10: Prewitt Operator applied on Brahmi script of Ashoka Pillar in Sarnath for edge detection. *(Image: Brahmi script on Ashoka Pillar in Sarnath, c. 250 BCE. Source: Wikipedia. Website accessed on: 10 Oct 2024.)*

6.5 Functional Implementation

Template matching, optical character recognition, picture preprocessing, and edge detection are some of the steps involved in the functional implementation. Every step is essential to the system's overall functionality.

6.6 Output

From the input Brahmi script images, the system produces the recognized Devanagari text as its output. Metrics including accuracy, precision, recall, and F1-score are used to assess the system's efficacy.

6.7 Standard Industry Practice Adopted

The project complies with accepted industry standards for image processing and machine learning. To guarantee the model's resilience and dependability, strategies including regularization, data augmentation, and cross-validation are used. Furthermore, software development best practices are adhered to, including version control, modular code, and thorough testing.

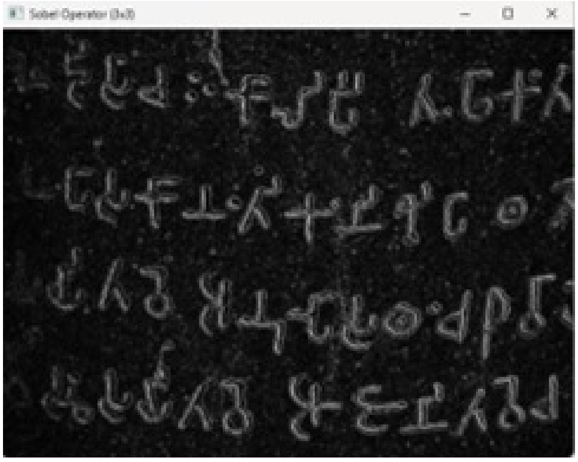

Figure 6.11: Sobel 3x3 Operator applied on Brahmi script of Ashoka Pillar in Sarnath for edge detection. *(Image: Brahmi script on Ashoka Pillar in Sarnath, c. 250 BCE. Source: Wikipedia. Website accessed on: 10 Oct 2024.)*

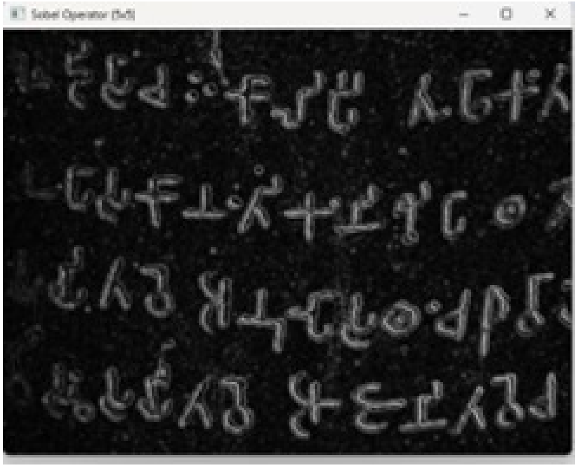

Figure 6.12: Sobel 5x5 Operator applied on Brahmi script of Ashoka Pillar in Sarnath for edge detection. *(Image: Brahmi script on Ashoka Pillar in Sarnath, c. 250 BCE. Source: Wikipedia. Website accessed on: 10 Oct 2024.)*

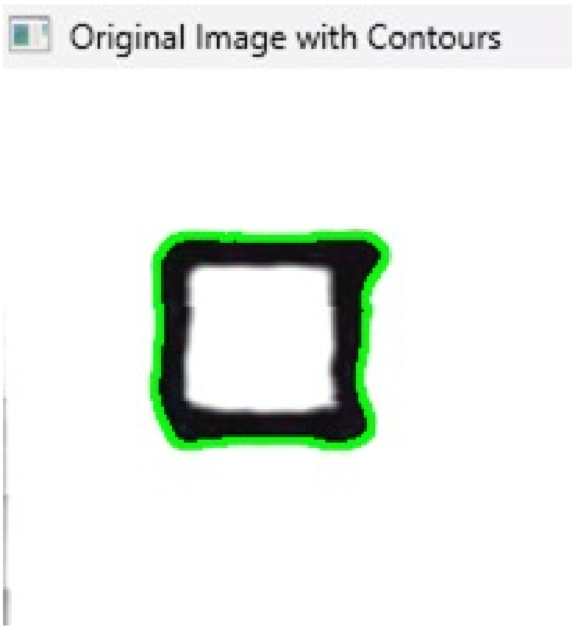

Figure 6.13: Shape detection of Brahmi character('ba'). *(Image: 'ba' character in Brahmi. Source: from our Brahmi database.)*

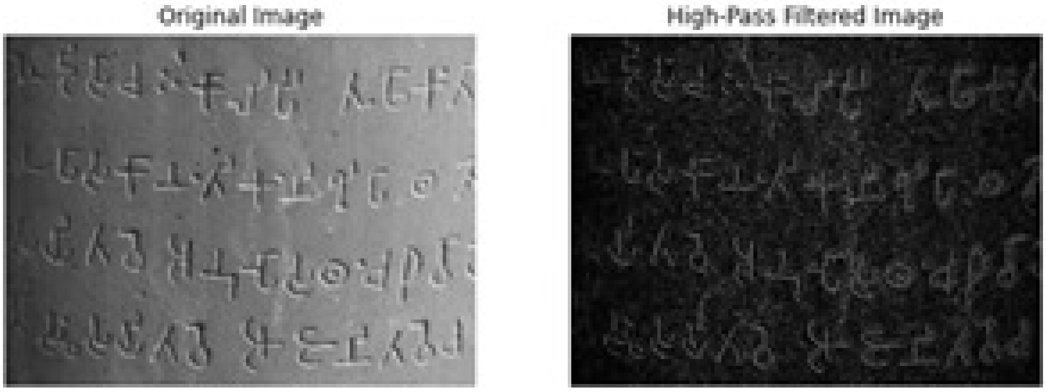

Figure 6.14: High-pass filter applied on Brahmi script of Ashoka Pillar in Sarnath. *(Image: Brahmi script on Ashoka Pillar in Sarnath, c. 250 BCE. Source: Wikipedia. Website accessed on: 10 Oct 2024.)*

Figure 6.15: Low-pass filter applied on Brahmi script of Ashoka Pillar in Sarnath. *(Image: Brahmi script on Ashoka Pillar in Sarnath, c. 250 BCE. Source: Wikipedia. Website accessed on: 10 Oct 2024.)*

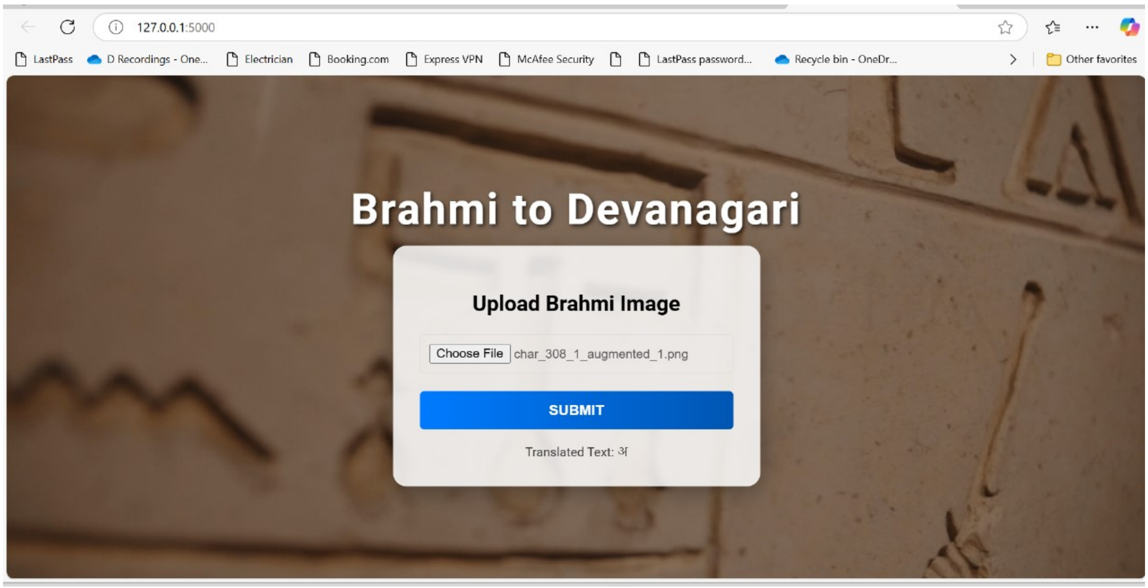

Figure 6.16: User Interface.*(Image:snapshot of our website. Source: snapshot from our system)*

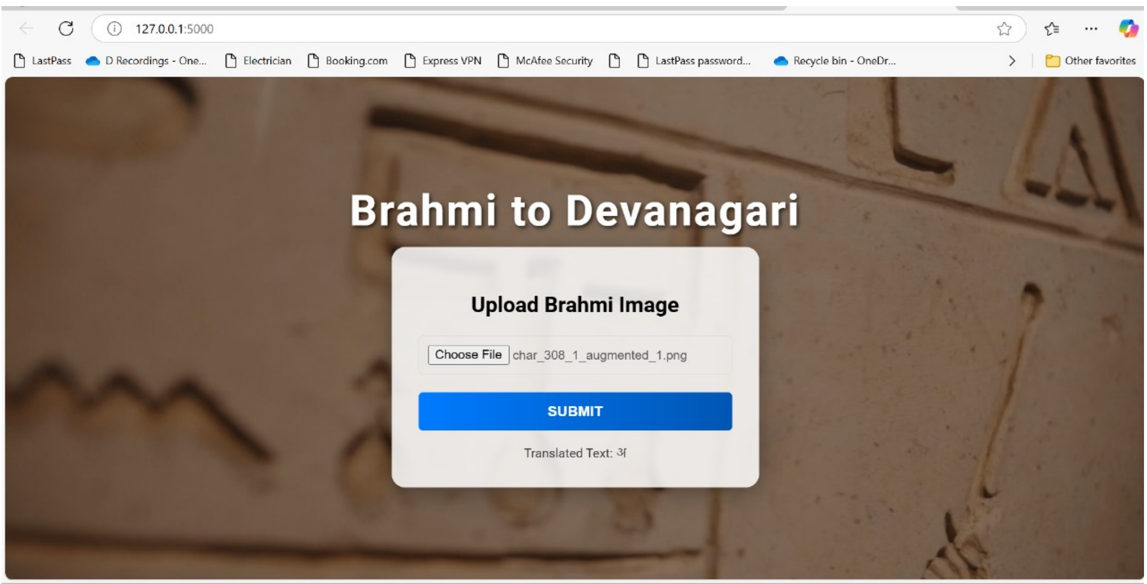

Figure 6.17: User Interface.*(Image:snapshot of our website. Source: snapshot from our system)*

Figure 6.18: Brahmi character (a). *(Image: Brahmi character for 'a'. Source: from our Brahmi database.)*

Figure 6.19: Corresponding Devanagari character (a). *(Image: Output Devanagari character for 'a'. Source: from our Devanagari database.)*

Chapter 7

Result Analysis/Performance Evaluation

7.1 Result Analysis

The performance of the proposed model for the recognition of Brahmi script using machine learning techniques is evaluated. The dataset includes images of Brahmi script characters. The system's performance is measured using accuracy, precision, recall, and F1-score. The experiments are conducted under various conditions to test the robustness of the model. The implementation is carried out using Python with libraries such as TensorFlow and OpenCV.

Table 7.1: Performance Metrics for Brahmi Script Recognition

Metric	Description	Value
Accuracy	Overall accuracy of the model after 10 epochs	83.32%
Loss	Loss value after 10 epochs	0.6478
al_loss	Additional loss metric	1.8806

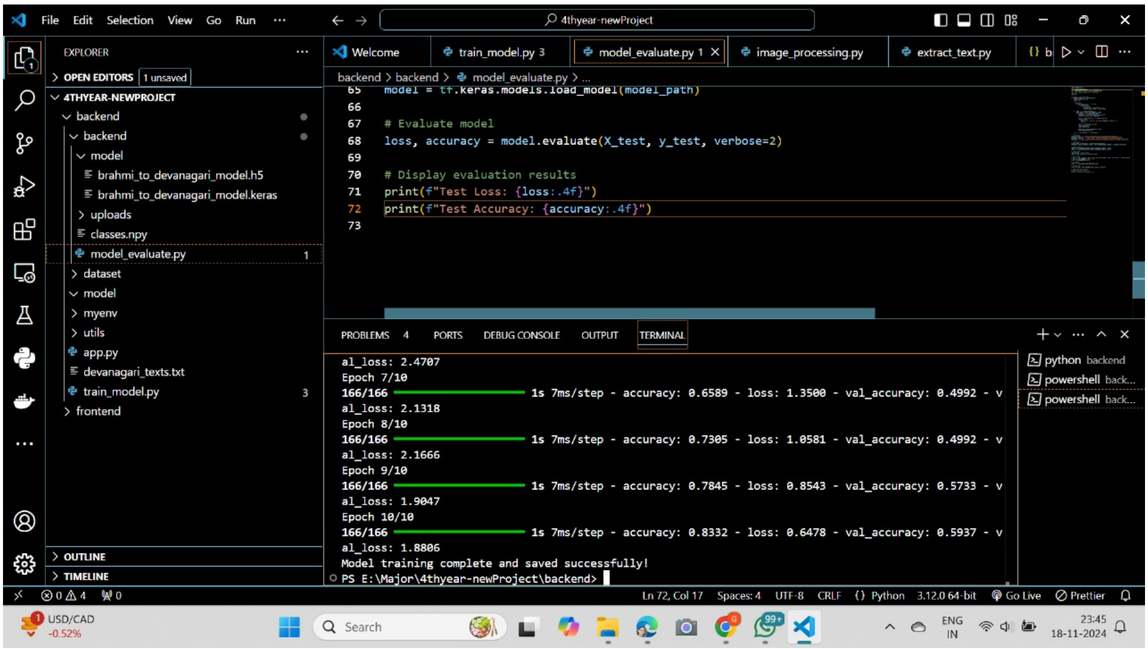

Figure 7.1: Accuracy of the result obtained. *(Image: Snapshot of the result of executed code. Source: Snapshot from our system.)*

Chapter 8

Conclusion

8.1 Conclusion

When we scan a picture of Brahmi script or Brahmi scripture carvings, the text is translated to Devanagari, and we receive the result. The Devanagari script would also be translated into English. We are using this to assist our new generation in easily understanding our archaic language.

8.2 Future Scope

Semester V

1. Conduct a literature review on the topic of Brahmi script recognition.
2. Collect a dataset of Brahmi script images.
3. Develop a method for preprocessing the dataset.
4. Implement a baseline Brahmi script recognition system.

Semester VI

1. Improve the performance of the baseline system by using more advanced techniques.

2. Evaluate the performance of the system on a held-out test set.

3. Develop a method for translating Brahmi characters to Devanagari characters.

Semester VII

1. Develop a method for translating Brahmi script into Devanagari.

2. Test the final system on a variety of datasets.

Appendices

Appendix A

Achievements and Publications

A.1 Conference Paper Submission

We submitted a paper to the *Conference Name 2025 4th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 5.0*.

A.2 Pune Anveshana Competition

We also participated in the Pune Anveshana competition and were finalists in this round.

12/11/24, 10:50 AM Conference Management Toolkit - Submission Summary

Submission Summary

Conference Name	2025 4th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 5.0
Track Name	OTCON2025
Paper ID	460
Paper Title	Recognition and Conversion of Brahmi Characters to Devanagari
Abstract	This work investigates the use of Python libraries to recognize Brahmi characters through the application of template matching. To precisely identify certain Brahmi letters within regulated image sets, the procedure entails scaling, grayscale conversion, and template matching algorithms. When working with more complicated inputs, such handwritten or deteriorated letters, this method's shortcomings become apparent, even though it produces dependable results in well-defined contexts, like printed text. In order to overcome these challenges, we incorporated Convolutional Neural Networks (CNNs) to improve the resilience of the system. Because CNNs can learn complex patterns and variations, they can recognize Brahmi characters in a variety of writing styles and contexts. Utilizing this cutting-edge method greatly increases recognition accuracy and provides a more scalable solution for a range of applications. Furthermore, by mapping recognized Brahmi characters to their equivalent Devanagari characters, the system goes beyond recognition to translation. This feature aids in the digitization and accessibility of historical writings by preserving old scripts and making translation work easier. By bridging the gap between historical linguistics and contemporary technology, our method advances optical character recognition (OCR) for ancient scripts and paves the way for further developments in script translation and preservation.
Created	12/11/2024, 10:44:52 AM
Last Modified	12/11/2024, 10:50:22 AM
Authors	**Gauri Kardekar** (MIT Academy of Engineering,Pune) <gauri.kardekar@mitaoe.ac.in> Gargi Gundawar (MIT Academy of Engineering,Pune) <gargi.gundawar@mitaoe.ac.in> Shrawani Chaudhari (MIT Academy of Engineering,Pune) <shrawani.chaudhari@mitaoe.ac.in> Sakshi Deshmukh (MIT Academy of Engineering,Pune) <deshmukh.sakshi@mitaoe.ac.in> Vinayak Kulkarni (MIT Academy of Engineering,Pune) <vbkulkarni@mitaoe.ac.in>
Primary Subject Area	Smart Computing for Data Science
Submission Files	Research_Paper_BrahmiCharacter.pdf (1.9 Mb, 12/11/2024, 10:44:09 AM)

Figure A.1: Proof of paper submission to the Conference Name 2025 4th OPJU International Technology Conference (OTCON).

Appendix B

Review of a book Brahmi Teacher

The foreword to Shri Pradip Dattuji Wankar's "Brahmi Teacher" emphasises the author's great interest in ancient Indian history and culture, particularly as it relates to historic money. In addition to being a life member of the INTAC chapter in Chandrapur and the Jt-Secretary of the "Chandrapur Coin Society," Wankar's previous book on "Regular Commemorative Coins of Republic India" received high accolades from coin collectors. The preface praises the author for focusing on Brahmi Script while avoiding contentious themes and emphasises the importance of numismatics and ancient scripts in decoding historical narratives. Wankar's life story is explored in the introduction, beginning with his discovery of a Brahmi-scripted book in college and concluding with him taking a deep interest in Indian history and ancient money. It highlights the importance of learning Brahmi and Kharoshti scripts in order to understand ancient India's history and culture. The author expresses gratitude to the Brahmi script study group for their assistance and gives the inspiration for the book to publications such as "Let us Learn Brahmi Script." The author concludes the preface by expressing the hope that the book will be valuable for coin collectors and connoisseurs interested in learning about the complexities of Brahmi writing, as well as increase understanding of ancient Indian history and culture.

Appendix C

Review of book The Palaeography Of India

The appendix of Pandit GauriShankar Hirachand Ojha's "The Palaeography Of India" is a great adjunct to the main body of work. It contains a thorough examination of Chronological Perspectives, going into numerous Indian eras to provide readers with a thorough grasp of the historical settings influencing script evolution. The appendix also includes Case Studies and Detailed Excurses, which provide in-depth analysis of unique issues encountered when reading ancient Indian epigraphs. These practical examples provide scholars and students with actual applications of the theoretical knowledge offered in the book's descriptive section.The Supplementary Exercises section presents additional practical resources, providing readers with additional opportunities to develop their Indian palaeography skills. Furthermore, the appendix includes Further Reading Recommendations, which direct readers to further scholarly works that deepen and enrich the primary text's content. Overall, the appendix supplements the main book by giving a greater range of tools and insights to scholars and students navigating the complex world of Indian palaeography.

Appendix D

Database

D.1 Backend Datasets

There are two datasets in the backend: the Brahmi dataset and the Devanagari dataset.

D.1.1 Brahmi Dataset

The Brahmi dataset contains folders named after each character. Inside each folder, there are photos of the particular character in the Brahmi script.

D.1.2 Devanagari Dataset

Similarly, the Devanagari dataset has folders named after each character, and each folder contains photos of the particular character in the Devanagari script.

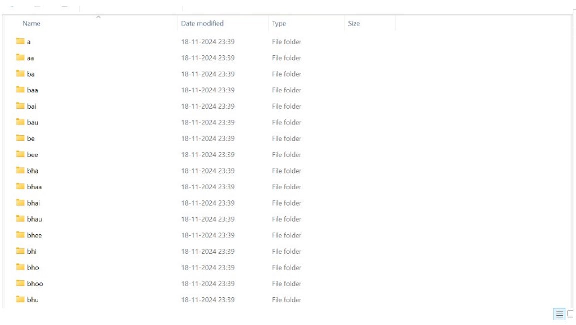

Figure D.1: Brahmi Dataset

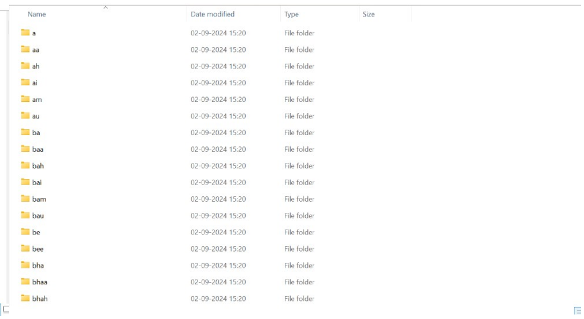

Figure D.2: Brahmi Dataset1

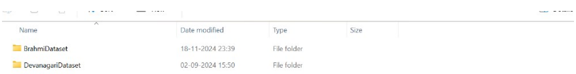

Figure D.3: Dataflow of Dataset

Appendix E

Sample Live Inscriptions

E.1 Live Inscription from Aihole, Karnataka

E.2 Live Inscription from Hampi, Karnataka

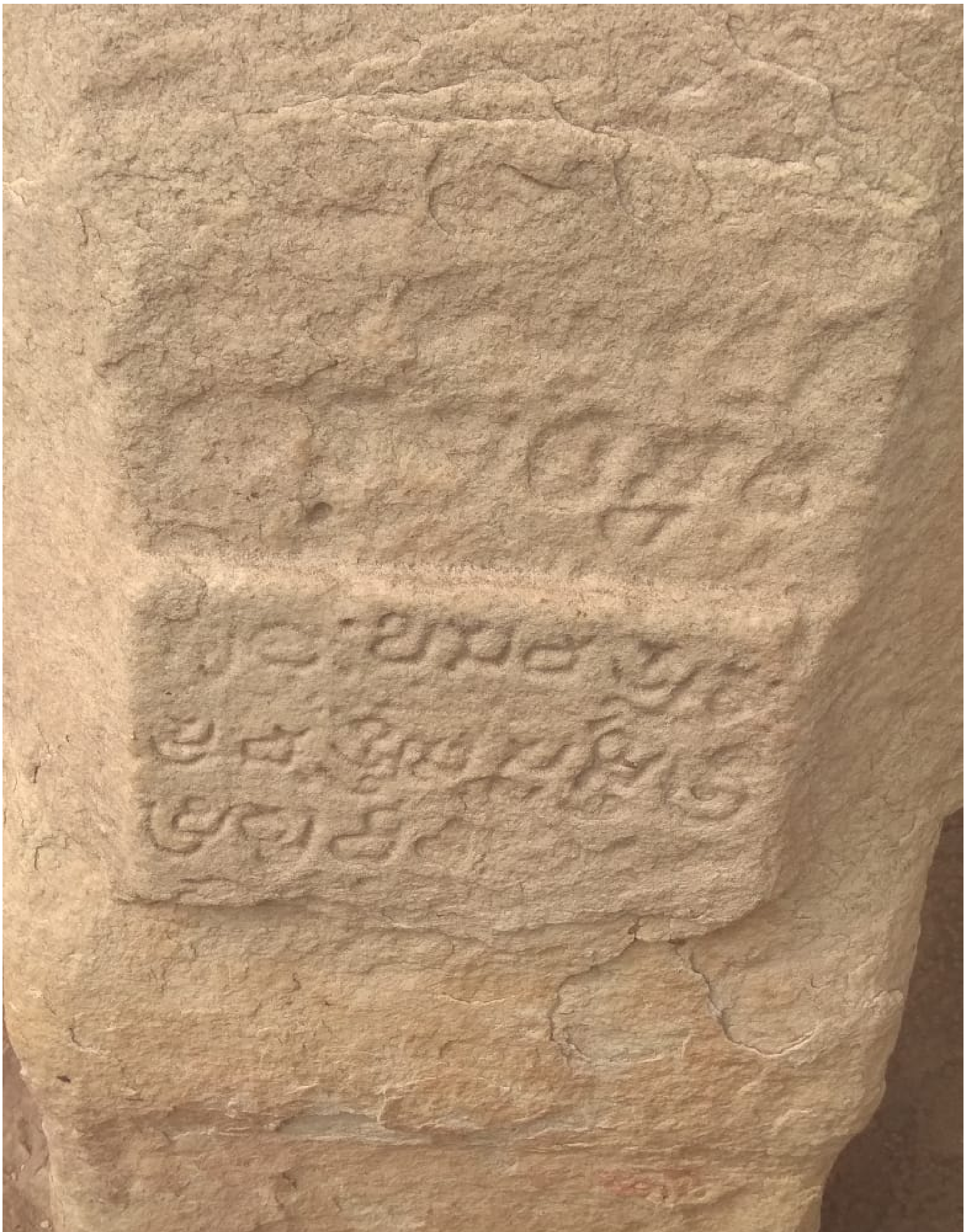

Figure E.1: Aihole, Karnataka *(Image: Live Inscriptions from Aihole,Karnataka. Source: Picture clicked by our team.)*

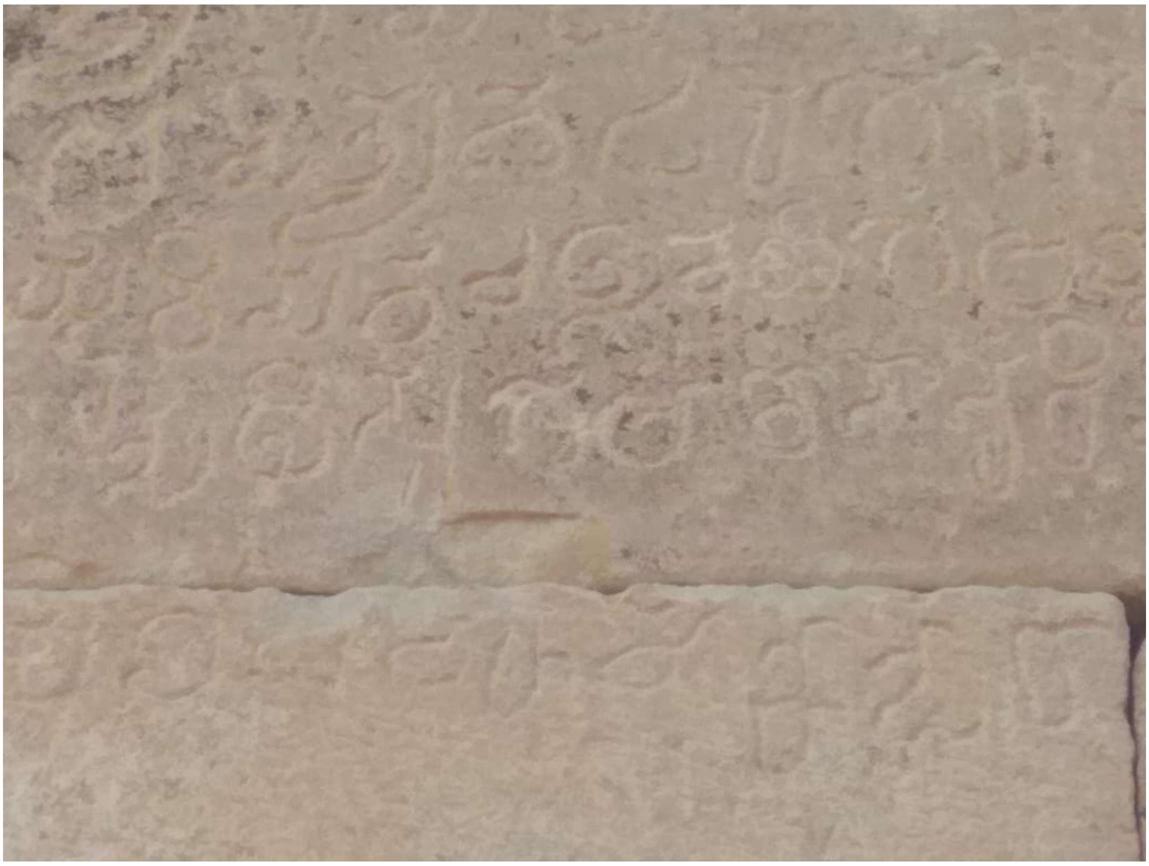

Figure E.2: Hampi, Karnataka*(Image: Live Inscriptions from Hampi,Karnataka. Source: Picture clicked by our team.)*

Chapter 9

References

1. Gautam, N., Sharma, R. S., Hazrati, G. (2016). Handwriting recognition of Brahmi script (an artefact): base of PALI language. In Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 2 (pp. 519-527). Springer International Publishing.

2. Singh, A. P., Kushwaha, A. K. (2019). Analysis of Segmentation Methods for Brahmi Script. DESIDOC Journal of Library Information Technology, 39(2).

3. Gautam, N., Chai, S. S., Jose, J. (2020). Recognition of Brahmi Words by Using Deep Convolutional Neural Network.

4. Gautam, N., Chai, S. S., Gautam, M. (2020). The Dataset for Printed Brahmi Word Recognition. In Micro-Electronics and Telecommunication Engineering: Proceedings of 3rd ICMETE 2019 (pp. 125-133). Springer Singapore.

5. Joseph, S., George, J. (2019). Feature extraction and classification techniques of MODI script character recognition. Pertanika J Sci Technol, 27(4), 1649-1669.

6. Agrawal, M., Chauhan, B., Agrawal, T. (2022). Machine learning algorithms for handwritten Devanagari character recognition: a systematic review. J Sci Technol, 7(01).

7. Moudgil, A., Singh, S., Gautam, V., Rani, S., Shah, S. H. (2023). Handwritten devanagari manuscript characters recognition using capsnet. International Journal

of Cognitive Computing in Engineering, 4, 47-54.

8. Bora, M. B., Daimary, D., Amitab, K., Kandar, D. (2020). Handwritten character recognition from images using CNN-ECOC. Procedia Computer Science, 167, 2403-2409.

9. Albahli, S., Nawaz, M., Javed, A., Irtaza, A. (2021). An improved faster-RCNN model for handwritten character recognition. Arabian Journal for Science and Engineering, 46(9), 8509- 8523.

10. Gharde, S. S., Ramteke, R. J. (2016, December). Recognition of characters in Indian MODI script. In 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC) (pp. 236-240). IEEE.

11.Kaur, S., Sagar, B. B. (2023). Efficient Scalable Template-Matching Technique for Ancient Brahmi Script Image. Computers, Materials Continua, 75(1).

12.Vincen, V., Samsuryadi, S. (2022). Brahmi Script Classification using VGG16 Architecture Convolutional Neural Network. Computer Engineering and Applications Journal, 11(2), 127-136.

13.SRIRAM, S. Virtual Reality Based Enhanced Brahmi Inscription Script Recognition.

14.Gunasekara, S., Lafir, M. H., Dulaj, C., Haputhanthri, L., Alwis, D. (2024, April). Deep Learning-powered Mobile App for Early Brahmi Script Decipherment in Sri Lanka. In 2024 International Research Conference on Smart Computing and Systems Engineering (SCSE) (Vol. 7, pp. 1-6). IEEE.

15.Vincen, V., Samsuryadi, S. (2022). Brahmi Script Classification using VGG16 Architecture Convolutional Neural Network. Computer Engineering and Applications Journal, 11(2), 127-136.