



Report.docx

-  MTech Dhairvi
-  MTech Dhairvi
-  MIT World Peace University, Pune India

Document Details

Submission ID

trn:oid:::28480:107907274

Submission Date

Aug 11, 2025, 1:49 PM GMT+5:30

Download Date

Aug 11, 2025, 2:05 PM GMT+5:30

File Name

Report.docx

File Size

2.4 MB

38 Pages

8,035 Words

49,241 Characters





6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.




Filtered from the Report

- Bibliography

Match Groups

-  **39 Not Cited or Quoted 6%**
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 3%  Internet sources
- 3%  Publications
- 4%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 39 Not Cited or Quoted 6%**
Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%**
Matches that are still very similar to source material
- 0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 3% Internet sources
- 3% Publications
- 4% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

- 1** **Publication**
Prajwal Gaikwad, Sanskruti Pardeshi, Shreya Sawant, Shrushti Rudrawar, Ketaki ... <1%
- 2** **Internet**
varun21290.medium.com <1%
- 3** **Internet**
repository.psa.edu.my <1%
- 4** **Submitted works**
Latvia University of Life Sciences and Technologies on 2023-12-05 <1%
- 5** **Submitted works**
University of Sunderland on 2023-10-24 <1%
- 6** **Internet**
repository-tnmgrmu.ac.in <1%
- 7** **Submitted works**
De La Salle University - Manila on 2025-08-04 <1%
- 8** **Internet**
www.ncbi.nlm.nih.gov <1%
- 9** **Submitted works**
University of Bradford on 2022-03-18 <1%
- 10** **Submitted works**
Government College of Engineering Kannur on 2025-08-05 <1%

11	Submitted works	University of Maryland, Global Campus on 2024-07-31	<1%
12	Publication	Young-Seok Hwang, Stephan Schlueter, Biswajeet Pradhan, Jung-Sup Um. "Unloc...	<1%
13	Internet	core.ac.uk	<1%
14	Publication	Aaishwarya S. Bajaj, Jayesh M. Dhodiya. "Sustainable 4D TSP to reduce carbon em...	<1%
15	Submitted works	North South University on 2025-01-26	<1%
16	Internet	www.mdpi.com	<1%
17	Submitted works	Sheffield Hallam University on 2023-05-23	<1%
18	Internet	link.springer.com	<1%
19	Internet	www.coursehero.com	<1%
20	Publication	Anushiya Rachel Gladston, K. V. Pradeep. "Exploring Solutions for Text-to-Speech ...	<1%
21	Publication	H.L. Gururaj, Francesco Flammini, J. Shreyas. "Data Science & Exploration in Artifi...	<1%
22	Submitted works	Liverpool John Moores University on 2024-03-03	<1%
23	Submitted works	Rochester Institute of Technology on 2012-11-30	<1%
24	Submitted works	University of Hertfordshire on 2022-09-23	<1%

25	Submitted works	University of Sydney on 2024-11-01	<1%
26	Internet	dspace.lboro.ac.uk	<1%
27	Internet	www.aad.gov.au	<1%
28	Internet	www.researchgate.net	<1%
29	Publication	"Artificial Intelligence: Theory and Applications", Springer Science and Business ...	<1%
30	Submitted works	Macquarie University on 2024-11-05	<1%
31	Submitted works	University of Durham on 2009-09-15	<1%
32	Internet	chm.pops.int	<1%
33	Submitted works	Manipal Academy of Higher Education (MAHE) on 2025-07-15	<1%
34	Submitted works	Manipal University Jaipur Online on 2025-03-23	<1%
35	Submitted works	Western Washington University on 2016-03-17	<1%
36	Publication	Wood, Carol. "Individual Differences in Women Based on Occupational Attainmen...	<1%
37	Publication	Harsh Ahlawat, Naveen Aggarwal, Deepti Gupta. "Automatic Speech Recognition:...	<1%
38	Publication	Wayne Patterson. "New Perspectives in Behavioral Cybersecurity II - Human Beha...	<1%

Acknowledgement

6 It is with great personal gratification that I submit this Project Stage-II Report. The process of completing this assignment has been very fulfilling, and along the way, I have been enriched through the support, guidance, and encouragement of numerous great people. I would like to take this platform to thank all those in some way directly or indirectly associated with the completion of this assignment.

30 First of all, I would like to express my sincere thanks to my project guide, Prof. Dr. Preeti Kale, School of Computer Engineering and Technology, for her useful inputs, constant encouragement, and considerate guidance. Her guidance was instrumental in finalizing the direction of my research and ensuring that I was on the correct path to accomplish my goals. Besides being an academic guide, she has been a wonderful mentor—encouraging, inspiring, and guiding patiently and assiduously through the entire process. The freedom she provided to me in exploring and working on my thoughts, tactfully guiding me in the right direction whenever necessary, is something I appreciate the most. Without her encouragement and guidance, it would have never been possible for this work to reach this level, and I am deeply indebted to her.

I would also like to thank Mr. Satish Bhosikar and Mr. Atul Bavdhankar, and the entire Dell Technologies team for the support and guidance they gave me during my one-year internship. Their mentorship, guidance, and willingness to help me at any time of need were a significant factor for my development and learning. I am also privileged to be a full-time Dell Technologies employee, a personal achievement.

22 I am also most thankful to my friends and classmates, who have been my support group throughout the ups and downs of this learning experience. Their arguments, encouragement, and stories have made the process more enjoyable and less daunting. Thank you all for this adventure that I will always remember.

I express my gratitude to Dr. Balaji Patil, Head, School of Computer Engineering and Technology, and all other faculty members whose encouragement and association made learning and research easier.

3 Lastly, but most importantly, I owe a lifelong debt of gratitude to my family—my parents and sister—being my inspiration and source of encouragement. Their faith in my capability, love, and sacrifices have been the guiding lights of all my success. Without their blessings and wishes, this success would have remained a distant dream. Genuine thank you to all of you who have been with me throughout the journey.

Contents

Ch. No.	Name of Sections	Page No.
	Certificate	I
	Certificate Of Completion By Company	II
	Abstract	III
	Acknowledgement	IV
	Contents	V
	List of Figure	VII
	List of Tables	VIII
	Abbreviations	IX
1.	Introduction	1
	1.1 Background of the Project	1
	1.2 Problem Statement	2
	1.3 Organization of report	3
2.	Literature Survey	4
	2.1 Basis of literature Review	4
	2.2 Literature	4
	2.3 Summary of literature	6
	2.4 Research Gap	7
3.	Aim, Objectives and Scope of study	8
	3.1 Aim	
	3.2 Scope	
	3.3 Objective	
4.	Methodology	9
	4.1 Dataset Preparation	9
	4.2 Classification	11
	4.3 Similarity	14
	4.4 XAI	18
5.	Results	19

6.	Conclusion	28
7.	Future Scope	29
	References	30
	Certificate of Attending Conference / Journal Publication	33
	Plagiarism Certificate	34
	Appendices	35
	I. A	35
	II. B	38
	III. C	39

13

List of Figures

Fig. No.	Name of Figure	Page No.
1.1	Languages Across India	2
4	Overall Methodology	9
4.1.1	Image Dataset Preparation	10
4.1.2	Textual Data Preparation	11
4.2.1	Flowchart for Image Classification	12
4.2.2	Flowchart for Text data Classification	13
4.3.1	Flowchart for Overall Similarity Trend	15
4.3.2	Flowchart showing letter wise similarity analysis	16
4.3.3	Flowchart of textual Similarity	18
5.2.1A	Overall Cosine Similarity among Devnagri script	21
5.2.1B	Overall Cosine Similarity among Dravidian script	22
5.2.2A	Letter wise Combinations	22
5.2.2B	Distribution of Cosine Similarity	23
5.2.2C	Letter Wise Combinations	23
5.2.2D	KDE of Cosine Similarity	24
5.2.3A	Document size	25
5.2.3B	Textual similarity among Devnagri script	25
5.2.3C	Textual Similarity among Dravidian Script	25
5.3.1	XAI for Classification	26
5.3.2A	Implementation for Devnagri Script	27
5.3.2B	XAI Implementation for Dravidian Script	27

List of Tables

Table. No.	Name of Figure	Page No.
1.3	Organization of the Report	3
2.2	Literature Review Table	4
5.1.1	Results for Image Classification	20
5.1.2A	Textual Classification for Indo – Aryan Group	20
5.1.2B	Textual Classification for Dravidian Group	20
5.2.1A	Summarization of results (Devnagri)	21
5.2.1B	Summarization of results (Dravidian)	22
5.2.3	Key observations from Textual Similarity	24

Abbreviations

OCR	Optical Character Recognition
CNN	Convolution Neural Networks
ML	Machine Learning
DL	Deep Learning
TL	Transfer Learning
LSTM	Long – Short Term Memory Model
HMM	Hidden Markov Model
XAI	Explainable Ai

3

Ch. 1. Introduction

1.1 Background of the project

India is internationally well known for its unmatched linguistic richness with 22 officially recognized languages and more than 1,600 dialects, some of which have their own unique scripts. These scripts are largely from the Indo-Aryan and Dravidian language families, each with its own typical structural, phonetic, and graphic features. A few examples include Devanagari (used for Hindi, Marathi, Sanskrit), Gujarati, Tamil, Kannada, and Telugu. Such diversity, as enriching as it is in terms of culture, poses humongous challenges in document digitization, archival preservation, and Optical Character Recognition (OCR).

The National Digital Library of India (NDLI), for instance, aims to digitize tens of millions of books and manuscripts in languages but is handicapped by chronic difficulties in accurately digitizing text due to script-level visual vagaries. Similar to the above are sectors such as banking, where forms and cheques are often handwritten or printed in composite scripts such as Marathi, Gujarati, and Hindi, which complicates the process for automated text processing and authentication systems. Furthermore, the Indira Gandhi National Centre for the Arts (IGNCA) contains a huge number of ancient manuscripts, numerous of which consist of low-resource or visually analogous scripts that need proper OCR tools to achieve successful digitization and research.



1.1 Languages Across India

Despite recent improvements in image-based analysis technologies, the majority of OCR systems today are biased towards Latin and Cyrillic scripts, which are relatively less complex. Indian scripts contain compound characters, vowel diacritics, conjuncts, and high contextual glyph variation, and all these combined make the higher error rates in OCR for Indian languages. A 2019 Ministry of Electronics and Information Technology (MeitY) survey [32] indicated that the existing OCR systems for Indian languages are on average as accurate as only 70-80%, far less than that of English OCR systems, which are typically well over 95% accurate.

DL techniques, especially Convolutional Neural Networks (CNNs), in the last few years have held out the promise of bettering classification accuracy by directly learning from image data. Yet, in the absence of a systematic approach to measuring visual and structural similarity between scripts, OCR models become over fitted to one particular dataset and do not gain generalized adaptability across other scripts or handwriting instances. Additionally,

20

the lack of an Explainability module in deep models makes the system difficult to interpret for critical use cases where it is essential to understand the decision-making process of the model, e.g., legal document analysis, historical manuscript digitization, and multilingual e-governance platforms.

1.2 Problem Statement

The existing state of Indian language script categorization and OCR technology is plagued by a myriad of deficiencies that make correct, scalable, and understandable solutions challenging. The primary issues which have been realized are:

- **Visual Similarity among Scripts:** Characters in Indian scripts are often similar to each other, especially for members of the same family. Certain Gujarati letters, for example, are very much similar to their Hindi or Marathi counterparts in Devanagari, and thus get misclassified in computers.
- **Lack of Generalizable OCR Models:** There are no generalizable OCRs for today's scripts, particularly when they are trained on a small dataset. It is very hard for less-researched languages like Manipuri or Konkani, for which there are no large-scale annotated datasets.
- **Poor Multilingual Accuracy:** In actual usage like multilingual signboards, school books, or composite-script bank cheques, OCR engines fail miserably in detecting scripts when there are two or more languages used in one document.
- **Limited Interpretability of DL Models:** Deep learning models, as accurate as they are, are black boxes that provide no or minimal information about the features or patterns behind the predictions. The transparency gap is likely to be dangerous in critical areas such as judiciary records or government documents.
- **Quantitative Visual Similarity Analysis Shortfall:** There exists a fundamental shortage of models for quantitatively comparing visual similarities and dissimilarities between scripts, which is required for transliteration, cross-language information retrieval, and script-aware translation services.

These limitations imply the necessity for a large, image-based, and interpretable model that can:

- Indian scripts classify with high accuracy without regard to visual appearance.
- Offer explainable insights into script classification judgments.
- Provide quantitative inter-script similarity analysis for transliteration, translation, and information retrieval improvement.
- Support scalable, mobile-centric document digitization, heritage conservation, and e-governance applications. The new framework, POSACLE, aims to address these deficiencies by blending deep learning with explainable AI (XAI) supported by an interpretability module specifically developed for the purpose of incorporating transparency as well as stability in multilingual script identification.

1.3 Organization of the Report

This report is structured to guide the reader through the various stages of the research, from foundational understanding to experimental insights and future possibilities.

Chapter	Title	Description
2	Literature Survey	Basis of review, prior work, research gaps.
3	Aim, Objectives, Scope	Defines the aim, detailed scope, and objectives.
4	Methodology	Framework design, data, models, analysis.
7	Results and Discussion	Experimental results, interpretations.
8	Conclusion	Consolidated findings of the research.
9	Future Scope	Opportunities for further exploration.
-	References	Academic and research references cited.
-	Certificates & Appendices	Conference, plagiarism certificates, and extra details.

Table 1.3: Organization of the Report

Ch. 2. Literature review

2.1 Basis of literature Review

The base of this literature review comes from the need to understand the current technological landscape and research methodologies addressing Indian language processing and script analysis, particularly in the context of ML, DL, and image processing methods.

Given that Indian scripts are not only linguistically diverse but also visually complex; the review was guided by the objective to explore how different studies have tackled tasks such as:

- Script recognition and classification
- OCR for multiple Indian languages
- Machine translation and transliteration
- Speech recognition, especially for low-resource languages
- Text detection and summarization in complex or compressed images

The review also extended to ancillary areas such as sentiment analysis, hate speech detection, and speech synthesis, where understanding language structures and representations is critical. Special attention was paid to models that employ transformers, CNNs, and hybrid machine learning techniques, as well as methods emphasizing multilingual support and cross-linguistic analysis.

By evaluating this body of work, the goal was to identify the key methodologies, datasets, application domains, and inherent challenges. This helps situate the current research within existing efforts while highlighting the gaps in script similarity analysis, multilingual OCR, and interpretability of DL for Indian scripts.

2.2 Literature

Research Area	Applications	Techniques	Languages	Key Papers
Hate Speech Detection	Offensive language detection, surveys	Transformer-based embeddings, ML, DL, TL, feature extraction	Kannada, Malayalam, Tamil, Hindi, Dravidian languages	[2], [4]
Unified Automatic Speech Recognition	Multilingual speech recognition	Common labeling scheme, multilingual modeling	Kannada, Telugu, Sanskrit, Malayalam, Tulu	[3], [12]
Machine	Neural machine	mT5 Transformer,	Hindi, Bengali,	[5], [18]

Research Area	Applications	Techniques	Languages	Key Papers
Translation	translation	AdamW optimizer	English, Bhojpuri, Sindhi, Marathi, Urdu	
Text Translation	Indian language translation	LSTM, BLEU score evaluation	Hindi, Odia, Malayalam, Indian Sign Language	[6]
Anomalous Text Identification	Anomaly detection in text	Markov Chain models	Indus Script	[7], [17]
Speech Synthesis	Multilingual text-to-speech synthesis	Multilingual training, byte-pair encoding	Indo-Aryan and Dravidian languages	[8], [15], [16]
Real-Time Translation	Real-time language translation	LSTM, deep learning	Marathi to Gujarati	[9], [14]
POS Tagging	Linguistic analysis for low-resource languages	Rule-based, HMM tagging	Marathi, Assamese, Bengali, Telugu, Konkani, Manipuri	[11]
Low-Resource Speech Recognition	Speech recognition in low-resource settings	Encoder-Decoder Transformer	Gujarati, Tamil, Telugu	[10], [13]
Learner Corpora & Prompt Design	Analyzing learner responses	Quantitative and qualitative analysis	Spanish	[19]
Corpus Linguistics	Methodological foundations across disciplines	Epistemological analysis	General linguistic applications	[20]
Gamified Learning	Educational experiences	Fogg's Behavior Model, Hook Model	English	[21]
Hypertext Fiction	Digital storytelling	Mixed-methods: textual analysis, surveys	English	[22]
Coreference Resolution	Text Understanding, Machine Translation	SPNet, Reinforcement Learning	English	[23]

Research Area	Applications	Techniques	Languages	Key Papers
Transliteration	Transliteration of Hindi text images	Image processing, attention-trained models	Hindi	[24]
Image Captioning	Hindi image caption generation	CNN for feature extraction, GRU	Hindi	[25]
Devanagari Text Detection	Text detection in wild images	SWT, MSER	Devanagari	[26]
OCR for Indian Languages	OCR for 11 Indian languages	Vision Transformer, ConvMixer	Multiple	[27]
Handwritten Character Recognition	Offline handwritten recognition	SVM, Backpropagation	Multiple	[28]
Text Information Extraction	Extraction from images	Connected component analysis, DWT	Multiple	[29]
Summarization of Compressed Images	Text summarization without OCR	Partial decompression	Multiple	[30], [31]

Table 2.2 Literature Review Table

2.3 Summary of literature

The literature surveyed spans a lengthy range of applications and methodologies in Indian language processing and script analysis. Researchers have applied deep learning models, transformers, rule-based approaches, and hybrid frameworks across tasks such as speech recognition, machine translation, text summarization, OCR, transliteration, and sentiment analysis.

For example, OCR models like CMViT and ConvMixer with Vision Transformers have made strides in recognizing multiple Indian scripts, achieving impressive accuracy rates, especially in printed texts. However, wild image recognition—such as text in street signs or inscriptions—still poses significant challenges. Similarly, machine translation research is advancing with transformer models like mT5, enabling cross-language conversion across major Indian languages.

Despite these achievements, the literature also reveals that efforts remain concentrated on a few widely spoken languages (like Hindi, Tamil, and Bengali), while low-resource languages continue to be underrepresented. Furthermore, while techniques like SVMs and LSTMs have been extensively applied to recognition and translation, there remains a gap in visual similarity analysis between scripts, which is crucial for OCR and transliteration accuracy.

2.4 Research Gap

Based on the survey done on the literature, research gaps have been identified are as follows:

- **Limited Focus on Low-Resource Languages**
- **Lack of Visual Similarity Analysis Across Scripts**
- **Generalization Limitations**
- **Challenges in Wild Image Recognition**
- **Limited Multilingual OCR Systems**
- **Emotion and Sentiment Detection.**

Ch. 3. Scope of the work and objective

3.1 Aim

To develop a robust, interpretable, and accurate framework for classifying and analyzing Indian scripts using combined deep learning and classical machine learning techniques for multilingual OCR and script evolution studies.

3.2 Scope

- Processed diverse datasets comprising whole words, isolated letters, and inscribed objects.
- Applied deep learning for automated feature extraction and CNN-based classification.
- Enhanced analysis with handcrafted features processed through Random Forest and Gradient Boosting.
- Performed script similarity analysis using cosine similarity on both learned and handcrafted features.
- Incorporated explainability through Grad-CAM and feature importance metrics.

3.3 Objectives

The objectives are to work on following points.

- i. To classify Indian scripts with high accuracy using visual features.
- ii. To quantify visual similarity between scripts for linguistic insights.
- iii. To create interpretable models that highlight discriminative features of scripts.
- iv. To support applications like multilingual OCR, script differentiation, and digital archiving.

Ch. 4. Methodology

The **POSACLE structure**, as illustrated in Figure 4, provides an overview of the comprehensive methodology employed in this study. This figure offers a structured insight into each methodological phase, outlining the key steps and processes that underpin the research framework. parts.

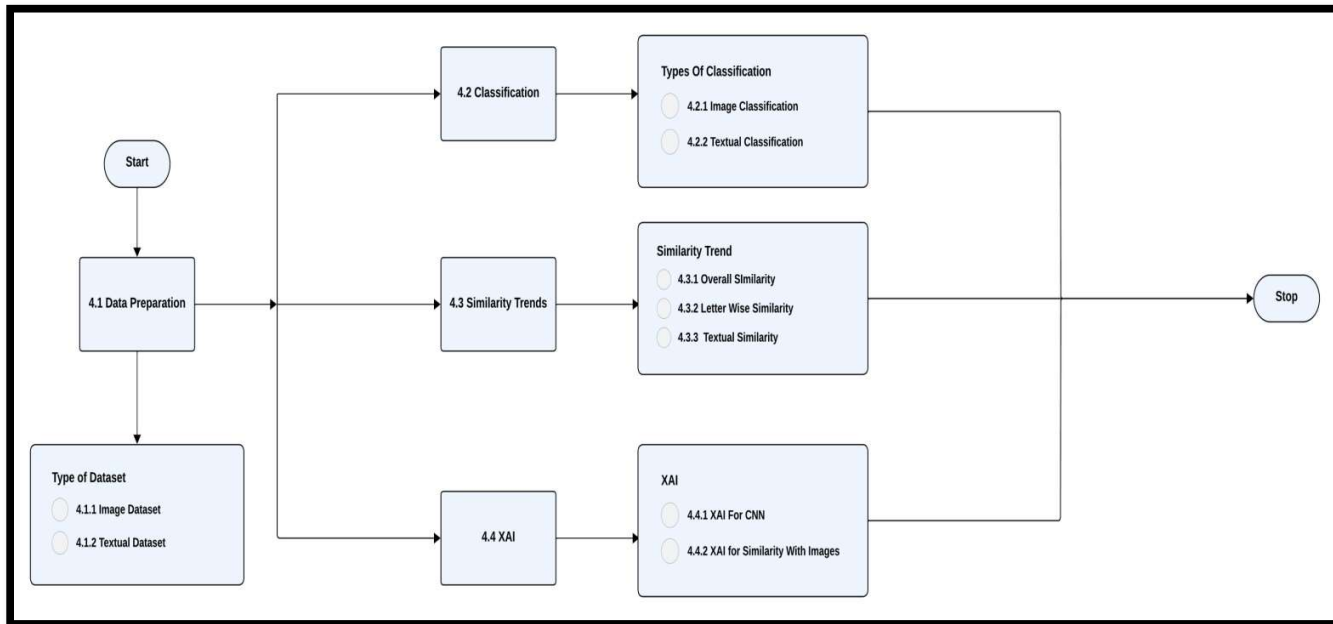


Figure 4: Overall Methodology

4.1 Dataset Preparation

The dataset utilized in this study, which implements the POSACLE system, comprises two primary types:

- Images : 10,000 for each language having same letter in 3 to 4 different fonts.
- Text: 10,000 lines for each language of same content.

A detailed description of the data processing workflow is as follows:

4.1.1 Image Dataset

As shown in following Figure 4.1.1, the image dataset preparation can be explained as below:

- Start with an input image, which can be:
 - A full word image
 - An image containing individual letters
 - An inscribed object (e.g., coins)
- Identify the type of image:
 - If it's a Word Image:
 - Convert the word into individual letter images through a Pre-Process:

Step 1: Extract the word from the image
 Step 2: Split the word into individual letters
 Step 3: Create a blank image using Python
 Step 4: Draw each letter onto the blank image (using Python libraries like PIL/OpenCV)
 These letter images are stored for classification or similarity analysis.

- If it's an Inscribed Image (e.g., object with text):
 Step 1: Use OCR (Optical Character Recognition) to extract letters directly from the image.
 Step 2: These extracted letters are then used for classification or similarity analysis.
- If it's already a Letter Image:
 Directly use it for classification/similarity tasks (no preprocessing needed).
- Images that don't match any of the three categories are discarded.
- Final output: A clean set of individual letter images, ready for use in machine learning models or analysis tasks.

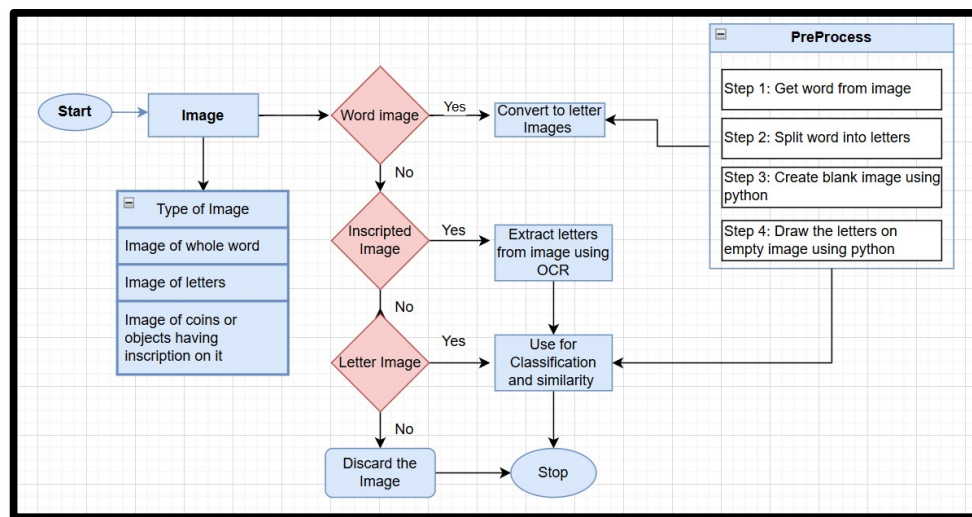


Figure 4.1.1: Image Dataset Preparation

4.1.2 Textual Dataset

Figure 4.1.2 describes the preparation of textual data. It can be interpreted using the following steps:

Step 1: Begin with textual data: One English paragraph containing about 10,000 lines is taken as the source content.

Step 2: Machine Translation: The paragraph is translated into several Indian regional languages using machine translation tools (OpenNMT)

Step 3: Target languages are Hindi, Gujarati, Marathi, Kannada & Tamil

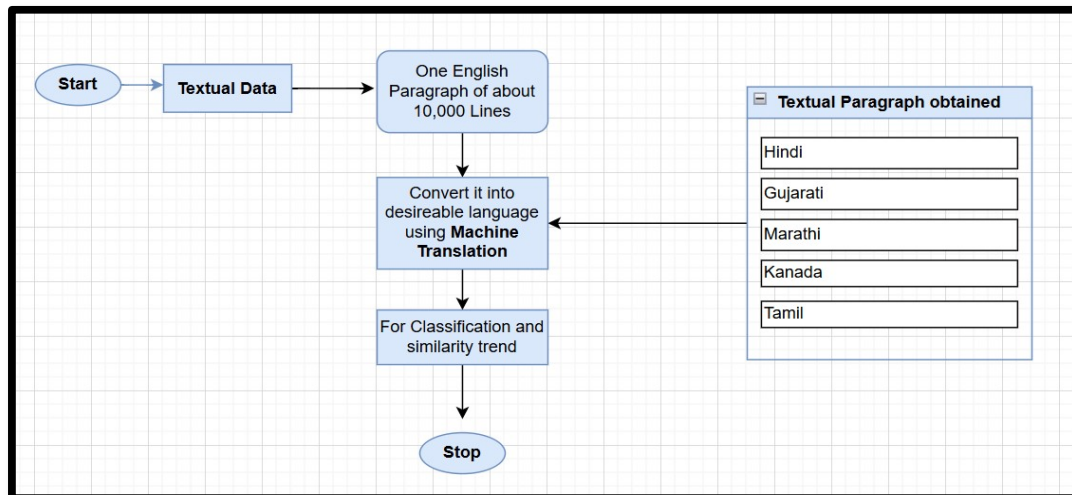


Figure 4.1.2 Textual Data Preparation

The combination of the two types of datasets enabled a complete analysis, bringing together visual and textual components to reveal complex phonetic and structural patterns between languages. The step-by-step approach ensured that every dataset was best suited to meet the needs of the POSACLE system.

This double-edged approach not only powered the findings of the study but also provided a strong base for subsequent studies in multilingual computational linguistics.

4.2 Classification

4.2.1 Image Classification

The first phase is model development and training to discriminate between different languages on the basis of visual representations of the text for these languages. This Image dataset is letter images as defined in data preparation. The complete process is visible in the following Figure 4.2.1.

- **Data:** Data considered here is prepared in section 4.1.1
- **Image Preprocessing:**
 - Standardize image dimensions and batch sizes.
 - Apply advanced data augmentation techniques to enhance variability and robust feature of the dataset.
 - Perform normalization to ensure uniformity across input data, facilitating consistent model performance.
- **CNN Architecture Design:**
 - Construct a CNN model, integrating layers such
 - Input: RGB image (img_width, img_height, 3)
 - Conv-2D (32, 3×3) + ReLU → MaxPooling (2*2)
 - Conv-2D (64, 3×3) + ReLU → MaxPooling (2*2)

- Conv-2D (128, 3×3) + ReLU → MaxPooling (2×2)
- Flatten
- Dense (128) + ReLU
- Dense (1)
- Carefully design each layer to optimize feature extraction and model learning capacity.

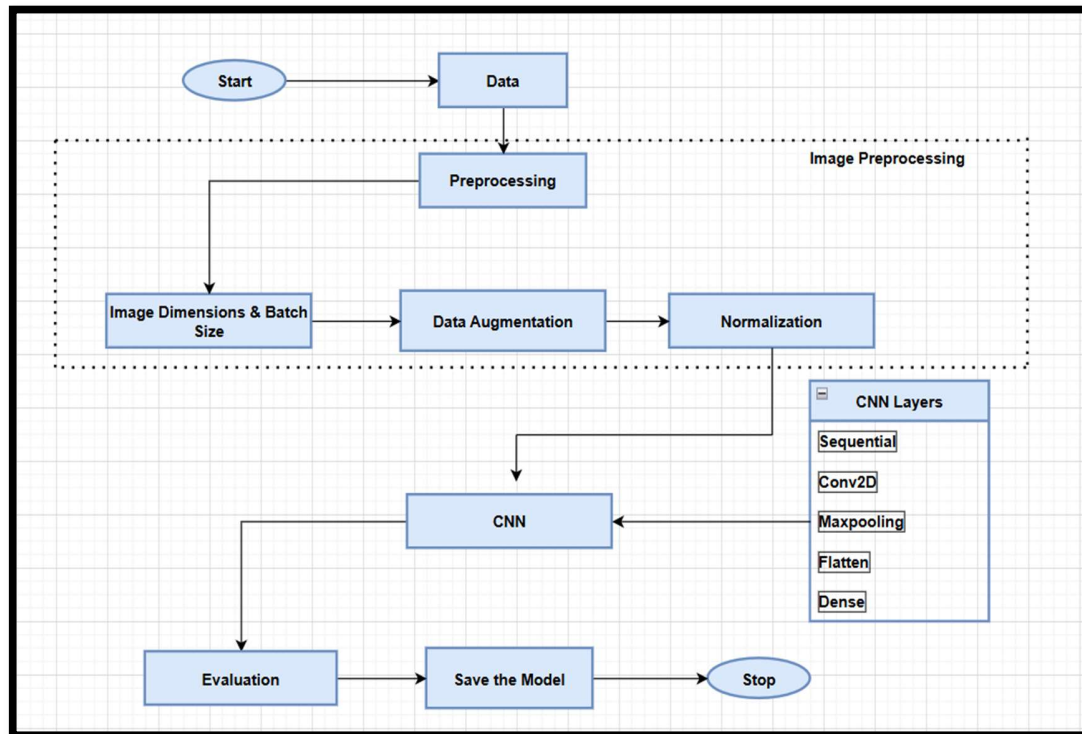


Figure 4.2.1: Flowchart for Image Classification

- **Model Training and Evaluation:**
 - Training phase of the CNN on the preprocessed and normalized dataset.
 - Evaluate its performance metrics to gauge effectiveness and identify any necessary adjustments.
- **Model Saving:** Once performance metrics are satisfactory, save the trained model for deployment in language identification tasks.

This structured approach ensures a comprehensive methodology for training a CNN to identify languages through image analysis.

4.2.2 Textual Classification

Following the successful classification of images, the study progresses to text-based classification prepared in data preprocessing. The architecture for this phase, illustrated in Figure 4.2.2, details the framework employed for processing and categorizing textual inputs.

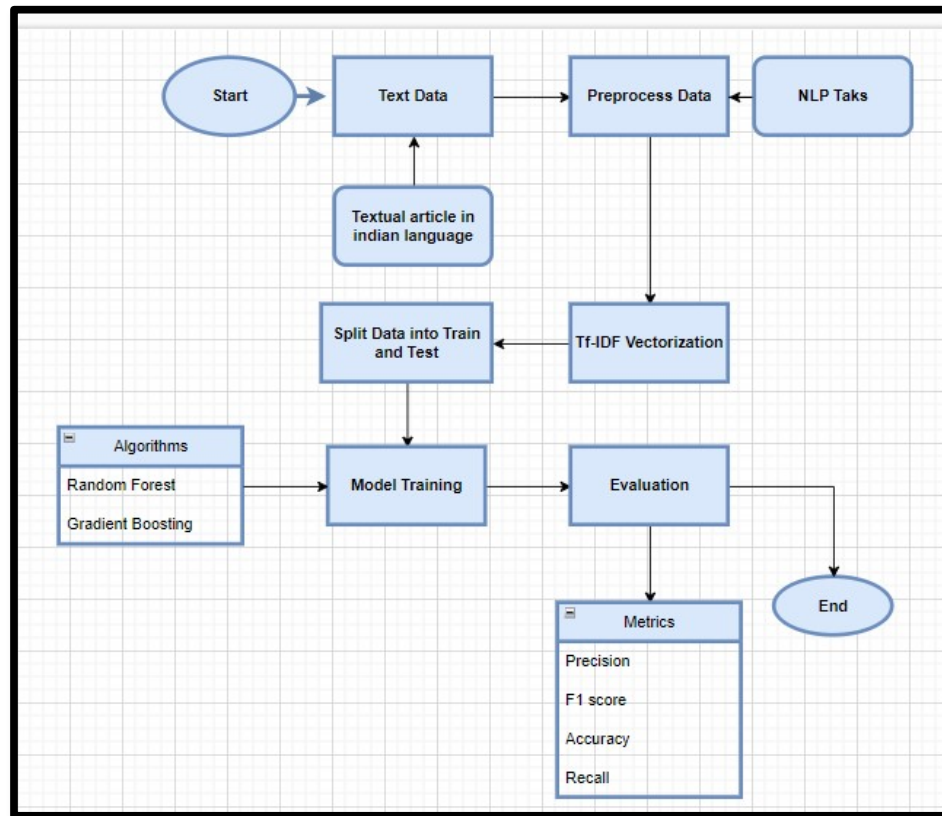


Figure: 4.2.2: Flowchart for Text data Classification

The above flow structure can be explained as:

- **Start:** The process begins here.
- **Text Data:** The input data is the one prepared during process of 4.1.2
- **Preprocess Data:** This step involves first cleaning and then preparing the text data for analysis. Common preprocessing tasks include tokenization, removing stop words, and stemming or lemmatization.
- **Split Data into Train and Test:** The previously processed data is split into two sets: training data for building the model and test data for evaluating its performance.
- **Tf-IDF Vectorization:** The text data is changed into numerical features using Tf-IDF strategy, which helps in quantifying the importance of words in the documents.
- **Algorithms:** Different machine learning algorithms are employed for model training. In this flowchart, two algorithms are highlighted:
 - Random Forest (standard Configurations)
 - Gradient Boosting (standard Configurations)
- **Model Training:** The selected algorithms are used to train the model on the training data.
- **Evaluation:** The trained model is evaluated using the test data. Various metrics features are calculated to assess the model's performance, including:
 - Precision

- F1 Score
- Accuracy
- Recall
- **End:** The process concludes here.

This flowchart provides a systematic approach to handling natural language processing (NLP) tasks, from raw text data to model evaluation.

4.3 Similarity Trend

The trend analysis for similarity here is focused on written script of languages and is not focused on pronunciation or spoken difference among natives. The analysis is carried out through two approaches:

1. Overall Trend: On the basis of overall language alone
2. Letter-wise Similarity: Assesses visual and structural similarity and phonic similarity of individual letters of languages, comparing similar or contrasting qualities of their written forms.
3. Textual Similarity: Language comparison on textual paragraph belonging to a respective language.

Both these methods together give an overall concept of script-based similarities and differences between languages, leading to precise information about written language features.

All the sub methods of similarity utilize Cosine Similarity. Cosine similarity is similarity between two vectors by calculating the cosine of the angle between the two vectors. Cosine similarity can be between -1 and 1 where 1 indicates vectors are equal and 0 indicates vectors are not similar at all. Cosine similarity is utilized in image analysis as well as in text analysis to determine the similarity of two objects or group of texts. Formula for Cosine similarity is

$$\cos\theta = \frac{A \cdot B}{||A|| \cdot ||B||}$$

where $A \cdot B$ is the dot product of the vectors and $||A||$ and $||B||$ represent the vector lengths. It is a measure of similarity by virtue of the vector relationship.

4.3.1 Overall Trend

This flowchart illustrates a general process for image data processing to differentiate between languages. It involves data loading and preprocessing, feature extraction, and image conversion to normalize. This is followed by random sampling and then computation of a cosine similarity matrix and visualization of the result by displaying it as a heat map for the sake of demonstrating patterns and variation across language sets.

Figure 4.3.1 below is a step-by-step process for image data analysis to differentiate among various languages. The process is outlined in the below step-by-step detailed process:

- **Start:** The process begins.
- **Load & Preprocess Data:** The first step is to load data and perform some preliminary preprocessing. It includes tasks such as normalization and cleaning on data prepared in 4.1.1 for analysis.

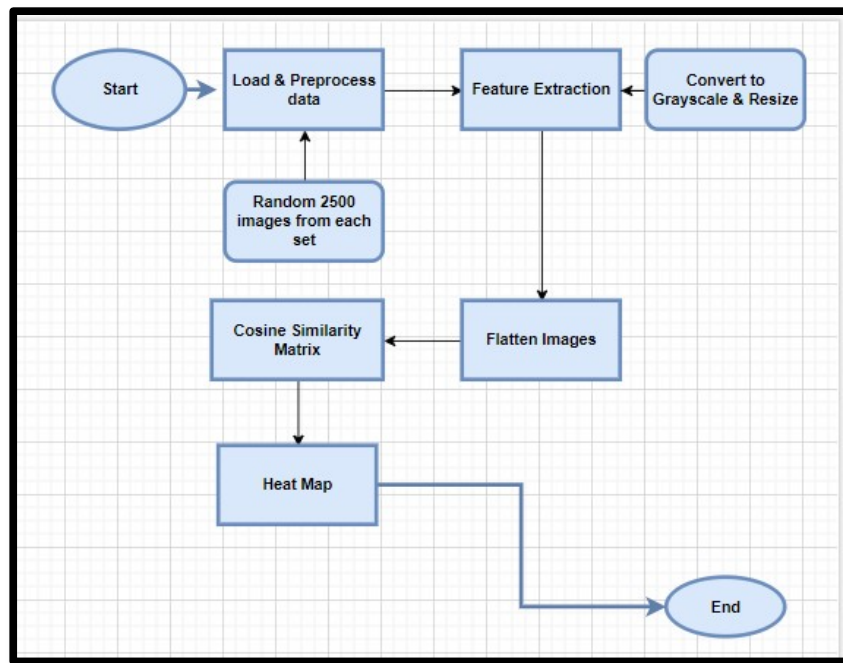


Figure 4.3.1: Flowchart for Overall Similarity Trend

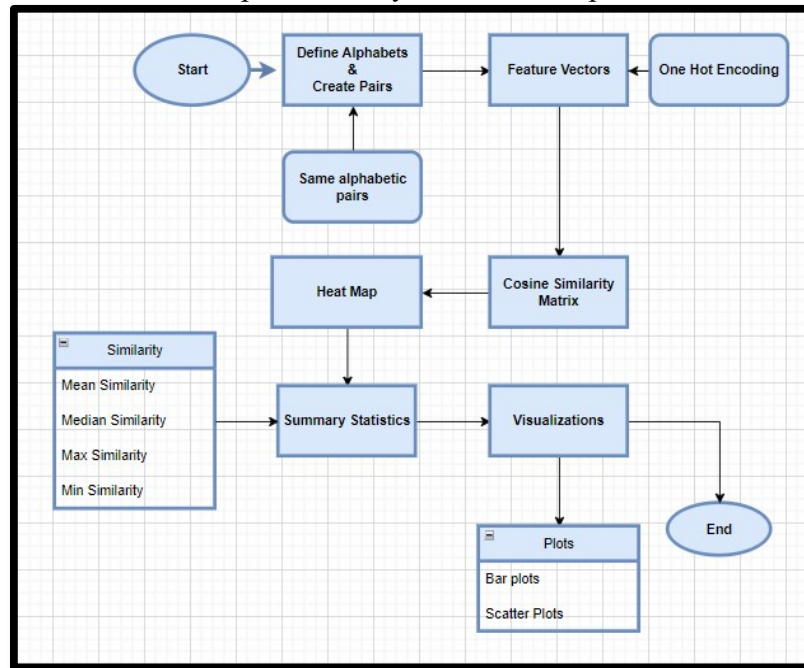
- **Random 2500 Images per Set:** 2500 images are randomly selected from each language set. Random selection helps in managing computational resources and in ensuring analysis representative of the entire dataset.
- **Feature Extraction:** Feature extraction is carried out on the preprocessed data to enhance the process of classification. This is intended to detect and remove major features from within the images to be used in the language identification.
- **Grayscale Conversion & Resizing:** To reduce complexity and computation, and to provide uniformity, all images in the dataset are converted to grayscale. The images also are resized to a common size so that the dataset is uniform.
- **Flatten Images:** Resized grayscale images are flattened into single-dimensional arrays. This is performed for convenience in handling and analysis in later steps.
- **Cosine Similarity Matrix:** A cosine similarity matrix is calculated for the chosen images. The matrix is used to measure the similarity between pairs of images by the cosine of the angle between their respective feature vectors. High similarity values suggest that the images are likely to be similar in feature terms.
- **Heat Map:** The cosine similarity matrix is portrayed as a heat map. This graphical representation of the similarity of patterns and clusters of the data shows how dissimilar or similar one image is to another.
- **End:** The procedure ends.

The whole process is made to process and analyze images systematically to differentiate between languages effectively based on their visual elements.

4.3.2 Letter Wise Similarity

The flowchart represented in Figure 4.3.2 described below depicts a stepwise similarity analysis and visualization of alphabetic characters of different languages. The process consists of multiple

stages, each of which resolves specific analytical and computational tasks, and is described



below:

Figure 4.3.2 : Flowchart showing letter wise similarity analysis

- **Start:** The process is initiated.
- **Define Alphabets & Form Pairs:** Alphabets in various languages are defined, and pairs of alphabetic characters are formed. The pairs are of similar and dissimilar alphabetic characters to allow comparative study.
- **Same Alphabetic Pairs:** "Same alphabetic pairs" are based on the similarity of characters of different linguistic systems with visual or phonetic similarity. For example, the Hindi/Marathi character 'क' is matched with the Gujarati 'ક', since both possess the same phonetic features and the same functions within their respective linguistic systems. To verify whether or not these pairs are accurate, we take into account the intersection of alphabets of both the languages, i.e., ' $A \cap B$ '. This guarantees that only the characters that have actual similarities are matched, making the process more effective.
- **Feature Vectors:** There are feature vectors produced for every pair of characters. The character pairs are represented in a high-dimensional space by these vectors, holding the features to be compared.
- **One-Hot Encoding:** One-hot encoding is a technique to convert the feature vectors. It translates categorical data into binary form. That is to ensure that the features are of the proper type for the subsequent analysis of similarity.
- **Cosine Similarity Matrix:** Cosine similarity matrix is calculated over feature vectors. The matrix calculates cosine angle between pairs of vectors and provides a quantitative measure of similarity between them.

- **Heat Map:** The cosine similarity matrix is graphically represented as a heat map. The graph representation illustrates the similarity of character pairs, and the strength of the color indicates the level of similarity.
- **Summary Statistics:** Summary statistics such as mean, median, maximum, and minimum similarity are calculated from the cosine similarity matrix. The statistics provide an overall view of similarities in and among various alphabet sets.
- **Plots:** The summary statistics are also represented by plotting them with various plots like bar plots and scatter plots. The plots help to describe the data and determine the distribution and relationship of similarities.
- **End:** The procedure ends, giving a complete comparison and visualization of similarities among alphabetic characters of various languages.

This procedure processes and analyzes character pairs systematically by using sophisticated feature extraction, similarity computation, and data visualization techniques in order to uncover subtle patterns and association patterns between various alphabetic systems.

4.3.3 Textual Similarity

The Figure 4.3.3 illustrates the general method taken into account while determining the similarity of the script languages.

- **Start:** It represents the beginning of the process.
- **Text Files:** Input here is that done in section 4.1.2.
- **Preprocess Data:** Preprocessing of text files is done. Preparation of the data to be analyzed is the function of this step. Text normalization, missing value imputation, and removal of noise are some common preprocessing operations.
- **NLP Tasks (Tokenization):** The input is then provided to Natural Language Processing tasks after preprocessing. Tokenization is a task in which the text is segmented into smaller elements, e.g., words or phrases.
- **Analysis:** Analysis is then conducted on the tokenized data. This is where the data is examined to construct useful insights.
- **Sampling:** 100 sentences from each language are sampled for closer analysis. This helps to limit the size of the data and focus on a representative subset.
- **Phonetic Similarity (Soundex):** Phonetic similarity of the sample sentences is confirmed employing the Soundex algorithm. This is to determine words that have the same sound.
- **Contextual Similarity:** Lastly, the sentences are matched for contextual similarity. That is, meaning and context of the words in the sentences.
- **Visualization:** Phonetic and contextual similarity search results are visualized. Pattern detection and data interpretation are facilitated by this step.
- **Dissimilarity Calculation:** The dissimilarity between the sentences is calculated. This is the measure of how dissimilar the sentences are to one another according to the analysis performed.
- **End:** This is the end of the process.

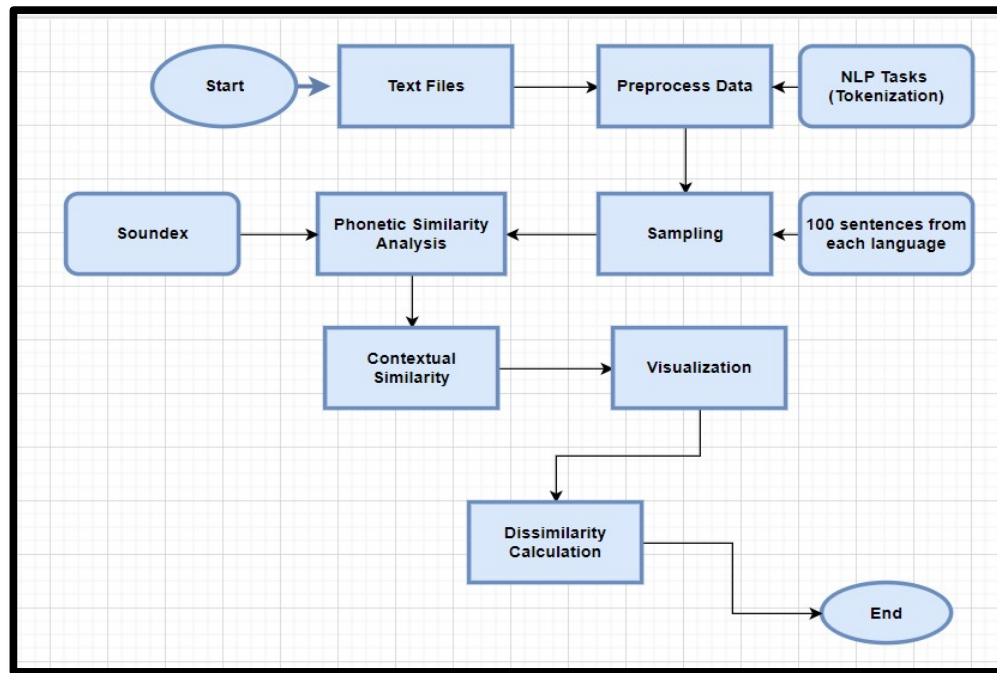


Figure 4.3.3: Flowchart of textual Similarity

This flowchart is a systematic way of handling and working with text data, and each step is performed systematically to get proper and comprehensible results.

4.4 XAI

Explainable AI Visualization to get better understanding the implementation and interpretability.

4.4.1 For Classification

1. Model Preparation:

- We first load the trained CNN model for language classification.
- The final sigmoid activation is removed to access raw logits for gradient computation.

2. Gradient Computation:

- Using TensorFlow's GradientTape, we track gradients of the predicted class score with respect to the feature maps of the last convolutional layer.
- For binary classification, we consider the predicted class (0 or 1) based on the threshold of 0.5.

3. Heatmap Generation:

- The gradients are pooled globally to obtain weights representing the essentiality of each feature map.
- A calculated combination of these feature maps produces the initial heatmap.
- The heatmap is normalized and resized to match the input image dimensions.

4. Visualization:

- The heatmap is color-coded (using a jet colormap) and superimposed on the original image.
- The intensity of colours indicates regions that most influenced the model's prediction.

4.4.2 For Letter Similarity

1. Letter Selection: Randomly select one letter from the first language and one letter from the second language
2. Preprocessing: Normalize and align the letters to ensure consistent positioning and scaling for fair comparison.
3. Pixel Matching and Similarity Computation: Compare the pixel-wise intensities between the two letters to generate a pixel similarity map. This highlights the regions of overlap and differences.
4. Visualization: Create visual outputs

Ch. 5. Results

This segment delineates the findings obtained from the execution of the aforementioned methodology. Each finding is scrutinized in connection to the distinct aims of the research, emphasizing the efficacy of the employed strategies and offering perspectives on the relative results of each method.

5.1 Classification

5.1.1 Image Classification Ref[4.2.1]

- Groups Analyzed:

Groups Analyzed	Precision	F1	Recall	Accuracy	Insights
Gujarati vs. Marathi/Hindi	0.97	0.97	0.96	98.57%	Demonstrates strong model sensitivity to subtle script variations in Indo-Aryan languages.
Tamil vs. Kannada	0.95	0.96	0.96	96.49%	Slightly lower performance reflects higher structural similarity between Dravidian scripts.

Table 5.1.1 : Results for Image Classification

- Inference from Table 5.1.1:
 - This experiment confirms the effectiveness of the image classification approach in script recognition.
 - The results underscore how language family characteristics (Indo-Aryan vs. Dravidian) can impact model accuracy, with each family presenting unique challenges in script differentiation.

5.1.2 Textual Classification Ref[4.2.2]

- Classification of Hindi, Gujarati, and Marathi Texts:**

Model	Precision	F1	Recall	Accuracy	Insights
Random Forest	0.93	0.93	0.93	93.25%	Consistently high performance; effectively classifies textual data across languages.
Gradient Boosting	0.88	0.86	0.85	85.52%	Moderate performance; lags Random Forest in handling multilingual text.

Table 5.1.2A: Textual Classification for Indo – Aryan Group

- Classification of Tamil and Kannada Texts:**

Model	Precision	F1	Recall	Accuracy	Insights
Random Forest	0.99	0.98	0.99	99.06%	Near-perfect classification for both languages.
Gradient Boosting	0.95	0.95	0.95	93.96%	Performs well but falls short of Random Forest's near-perfect results.

Table 5.1.2B: Textual Classification For Dravidian Group

To summarize the Table 5.1.2A and 5.1.2B given above:

- Random Forest:** Achieved higher accuracy and balanced metrics across both Indo-Aryan and Dravidian language groups, excelling in Tamil and Kannada classification.
- Gradient Boosting:** Effective but generally lower in accuracy and metric balance, particularly for Indo – Aryan group.

Inference: Random Forest demonstrates superior performance in handling multilingual text classification tasks.

5.2 Similarity Trend

5.2.1 Overall Similarity Ref[4.3.1]

I. Gujarati & Hindi/Marathi letters

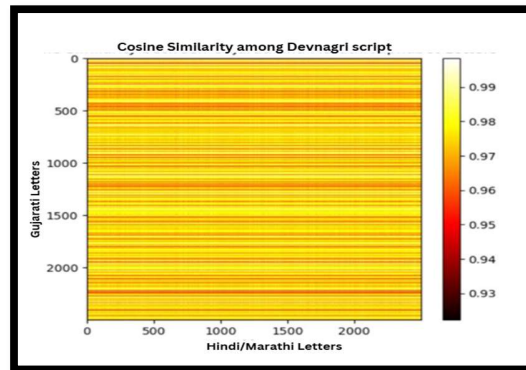


Figure 5.2.1A: Overall Cosine Similarity among Devanagri script

The Figure 5.2.1 illustrates the cosine similarity as defined in methodology, between Gujarati and Hindi/Marathi alphabet letters. As the color bar indicates, the heatmap demonstrates a significant level of similarity across the majority. The yellow regions represent the highest similarity scores close to 0.99, while the darker shades indicate slightly lower similarity. This suggests that the alphabets of these two languages, despite belonging to different scripts, exhibit **substantial structural or visual resemblance**.

Cosine Similarity Range	Percentage	General Interpretation
0.97 - 0.99	40%	Very high similarity
0.94 - 0.96	35%	High similarity
0.93 - 0.94	15%	Moderate to high similarity
Below 0.93	10%	Moderate or lower similarity

Table 5.2.1A: Summarization of results

This Table 5.2.1 presents the cosine similarity trends, showing that a significant portion of letter pairings (75%) exhibit high to very high similarity (0.94 and above), suggesting a strong structural or visual resemblance between the alphabets of Gujarati and Hindi/Marathi. The analysis of the 2500-image dataset reveals that a substantial 75% of the images exhibit a similarity level of 0.93 or above, as presented in Table 5.

II. Tamil and Kannada letters

The cosine similarity heatmap featured in Figure 10 highlights a detailed relationship between Kannada and Tamil letters. In contrast to the consistent similarity observed in Devnagari, the letters of Kannada and Tamil reveal alternating bands of elevated and diminished similarity, thereby indicating certain subsets of characters that possess common characteristics while others exhibit marked distinctions. This variability underscores the intricate visual complexity inherent within Dravidian scripts.

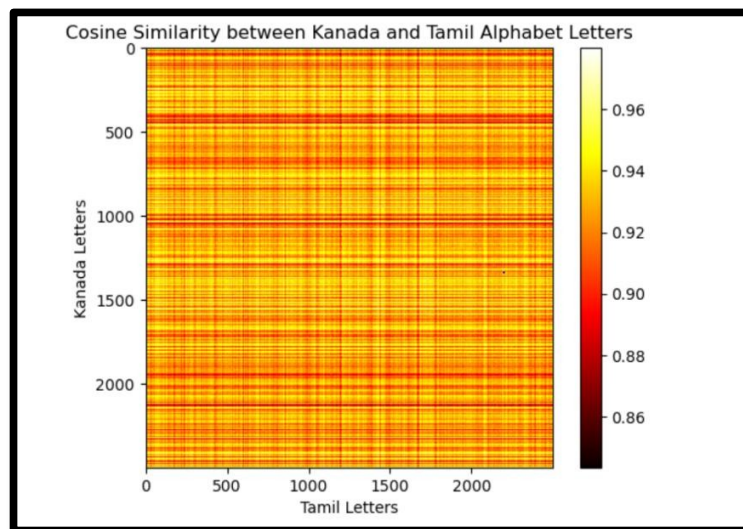


Figure 5.2.1B: Overall Cosine Similarity among Dravadian script

Cosine Similarity Range	Percentage	Interpretation
0.94 - 0.96	20 %	High similarity for certain letters
0.90 - 0.93	35 %	Moderate similarity across some letters
0.86 - 0.89	45 %	Lower similarity indicating distinct forms

Table 5.2.1B: Summarization of results

As shown in above Table 5.2.1B the similarities in structure or visuals of input images for Tamil and Kanada have distinct features in comparison to those found in Devnagari script.

Challenges of this method : As it can be inferred from above results that this method of 4.3.1 with results of 5.2.1 are inefficient. As a result, this study also incorporates the methodology 4.3.2 and 4.3.3 for better and efficient results.

5.2.2 Letter Wise Similarity Ref[3.3.2]

1. Devnagari Script

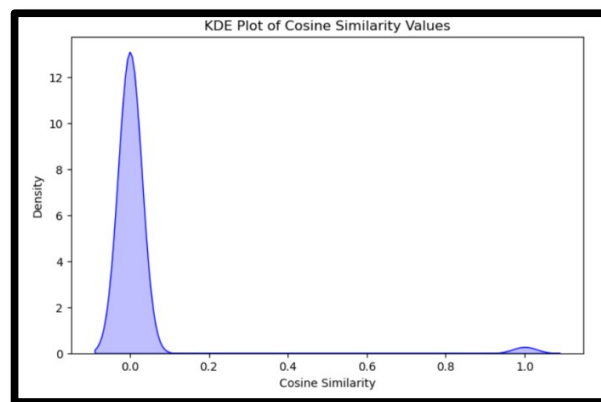
```
# Define the alphabets of Gujarati and Marathi Languages
gujarati_alphabets = ['અ', 'આ', 'ઇ', 'ઈ', 'ઉ', 'ઊ', 'એ', 'ઐ', 'ઓ', 'ઔ', 'અં', 'અઃ', 'ક', 'ખ', 'ગ', 'ઘ', 'ઙ', 'ચ', 'છ', 'જ', 'ઝ', 'ઞ', 'ટ', 'ઠ', 'ડ', 'ઢ', 'ણ', 'ત', 'થ', 'દ', 'ધ', 'ન', 'પ', 'ફ', 'બ', 'ભ', 'મ', 'ય', 'ર', 'લ', 'વ', 'શ', 'ષ', 'સ', 'હ', 'ળ', 'ક્ષ', 'ઞ', 'ઞ', 'ઞ']
marathi_alphabets = ['अ', 'आ', 'इ', 'ई', 'उ', 'ऊ', 'ए', 'ऐ', 'ओ', 'औ', 'अं', 'अः', 'क', 'ख', 'ग', 'घ', 'ङ', 'च', 'छ', 'ज', 'झ', 'ञ', 'ट', 'ठ', 'ड', 'ढ', 'ण', 'त', 'थ', 'द', 'ध', 'न', 'प', 'फ', 'ब', 'भ', 'म', 'य', 'र', 'ल', 'व', 'श', 'ष', 'स', 'ह', 'र्', 'ल्', 'क्ष', 'ञ', 'ञ', 'ञ']

# Define the similar pairs of alphabets based on their phonetic sounds
similar_pairs = [('अ', 'अ'), ('आ', 'आ'), ('इ', 'इ'), ('ई', 'ई'), ('उ', 'उ'), ('ऊ', 'ऊ'), ('ए', 'ए'), ('ऐ', 'ऐ'), ('ओ', 'ओ'), ('औ', 'औ'), ('अं', 'अं'), ('अः', 'अः'), ('क', 'क'), ('ख', 'ख'), ('ग', 'ग'), ('घ', 'घ'), ('ङ', 'ङ'), ('च', 'च'), ('छ', 'छ'), ('ज', 'ज'), ('झ', 'झ'), ('ञ', 'ञ'), ('ट', 'ट'), ('ठ', 'ठ'), ('ड', 'ड'), ('ढ', 'ढ'), ('ण', 'ण'), ('त', 'त'), ('थ', 'थ'), ('द', 'द'), ('ध', 'ध'), ('न', 'न'), ('प', 'प'), ('फ', 'फ'), ('ब', 'ब'), ('भ', 'भ'), ('म', 'म'), ('य', 'य'), ('र', 'र'), ('ल', 'ल'), ('व', 'व'), ('श', 'श'), ('ष', 'ष'), ('स', 'स'), ('ह', 'ह'), ('र्', 'र्'), ('ल्', 'ल्'), ('क्ष', 'क्ष'), ('ञ', 'ञ'), ('ञ', 'ञ'), ('ञ', 'ञ')]
```

Figure 5.2.2A : Letter wise Combinations

The Figure 5.2.2A shares the insights on letter wise comparison for Devanagri script. Also, the mean cosine similarity between the letters of Gujarati, Hindi, and Marathi scripts is calculated to be approximately 0.0204. This low mean similarity value indicates that, on average, the letters across these scripts exhibit minimal structural or visual resemblance, despite their shared linguistic and cultural roots. This suggests that while there may be some overlap or common features between the scripts, the overall differences in letter shapes and forms are significant, which could aid in distinguishing these scripts in classification tasks.

Figure 5.2.2B: Distribution of Cosine Similarity



The KDE plot in Figure 5.2.2B reveals a strong concentration of values near 0.0, indicating that most letter pairs have minimal similarity. A small secondary peak near 1.0 suggests that a few letter pairs share substantial structural similarities, though these are relatively rare. This distribution pattern, with a mean similarity of approximately 0.0204, underscores the distinctiveness of letters across these scripts despite occasional overlaps.

II. Dravadian Script

```
# Define the alphabets of Kannada and Tamil Languages
kannada_alphabets = ['ಅ', 'ಆ', 'ಇ', 'ಈ', 'ಉ', 'ಊ', 'ಋ', 'ಎ', 'ಐ', 'ಓ', 'ಔ', 'ಕ', 'ಖ', 'ಗ', 'ಘ', 'ಙ', 'ಚ', 'ಛ', 'ಟ', 'ಠ', 'ಡ', 'ಢ', 'ತ', 'ಥ', 'ದ', 'ಧ', 'ನ', 'ಪ']
tamil_alphabets = ['அ', 'ஆ', 'இ', 'ஈ', 'உ', 'ஊ', 'எ', 'ஏ', 'ஐ', 'ஓ', 'ஔ', 'ஶ', 'ச', 'ந', 'ஶ', 'ட', 'ண', 'த', 'ந', 'ன', 'ப']

# Check and align the lengths of Kannada and Tamil alphabets Lists
if len(kannada_alphabets) != len(tamil_alphabets):
    max_length = max(len(kannada_alphabets), len(tamil_alphabets))
    kannada_alphabets.extend([''] * (max_length - len(kannada_alphabets)))
    tamil_alphabets.extend([''] * (max_length - len(tamil_alphabets)))

# Construct feature vectors for similar pairs of alphabets
similar_pairs = list(zip(kannada_alphabets, tamil_alphabets))
kannada_vectors = np.array([np.eye(1, len(kannada_alphabets), kannada_alphabets.index(pair[0])) for pair in similar_pairs])
tamil_vectors = np.array([np.eye(1, len(tamil_alphabets), tamil_alphabets.index(pair[1])) for pair in similar_pairs])
```

Figure 5.2.2C: Letter Wise Combinations

The Figure 5.2.2C shares the insights on letter wise comparison for Dravadian script. Also, the mean cosine similarity between the letters of Tamil and Kanada scripts is calculated to be approximately 0.0206. This low mean similarity value indicates that, on average, the letters across these scripts exhibit minimal structural or visual resemblance, despite their shared linguistic and cultural roots. This suggests that while there may be some overlap or common features between the scripts, the overall differences in letter shapes and forms are significant, which could aid in distinguishing these scripts in classification tasks.

shapes and forms are significant, which could aid in distinguishing these scripts in classification tasks.

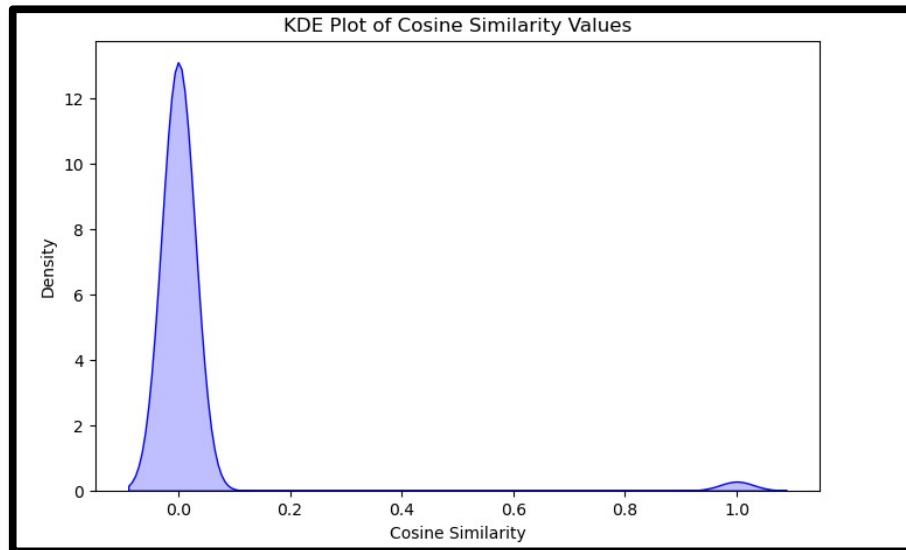


Figure 5.2.2D: KDE of Cosine Similarity

Figure 5.2.2D displays a KDE plot illustrating the cosine similarity clusters. A small secondary peak near 1.0 indicates that a few letter pairs exhibit significant structural similarities, although these occurrences are relatively infrequent. The overall distribution, with a mean similarity of approximately 0.0206, highlights the distinctiveness of the letters in both scripts, despite occasional overlaps.

5.2.3 Textual Similarity Ref[4.3.3]

I. Devanagri Script:

The analysis of *Textual Parity* i.e. comparison of written content in different languages to ensure that the conveyed meaning and structure remain consistent across linguistic translations, among Hindi, Gujarati, and Marathi reveals the following key observations shown in Table 5.2.3:

Aspect	Gujarati & Marathi	Gujarati & Hindi	Marathi & Hindi
Textual Parity	Mean: 0.09 (Low)	Mean: 0.03 (Very Low)	Mean: 0.69 (High)
Dissimilarity Threshold	No dissimilarities below threshold	No dissimilarities below threshold	Dissimilarity Score: 0.32 (Notable distinctions despite phonetic similarity)

Table 5.2.3: Key observations from Textual Similarity

The analysis from Table 5.2.3 reveals that Gujarati and Hindi exhibit the lowest Textual Parity, indicating significant linguistic divergence between these two languages. In

contrast, Marathi and Hindi demonstrate a much closer phonetic correlation, as reflected by their higher similarity score. The document sizes for all three languages are comparable, as shown in Figure 5.2.3A suggesting balanced data representation. Despite the phonetic closeness between Marathi and Hindi, a notable distinction is observed with a dissimilarity score of 0.32, highlighting subtle linguistic differences between these two languages. These results have also been depicted by plot shown in Figure 5.2.3B.

Gujarati	3,445,837 characters
Marathi	3,361,362 characters
Hindi	3,384,385 characters

Figure 5.2.3A Document size

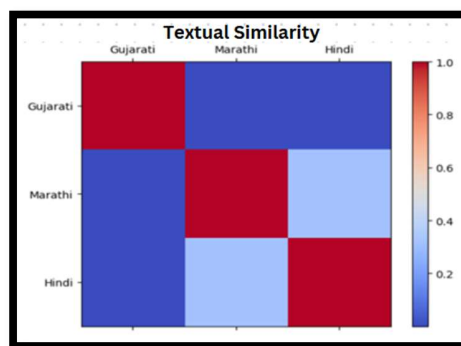


Figure 5.2.3B: Textual similarity among Devanagri script

II Dravadian Script:

The analysis of Textual Parity between Tamil and Kannada reveals the following key points:

1. **Textual Parity:** The **average similarity is 0.11**, indicating a moderate degree of semantic closeness between the two languages.
2. **Article length:**
 - **Tamil: 3,941,467 characters**
 - **Kannada: 32,699,244 characters (significantly longer)**

These findings are illustrated in Figure 5.2.3C below.

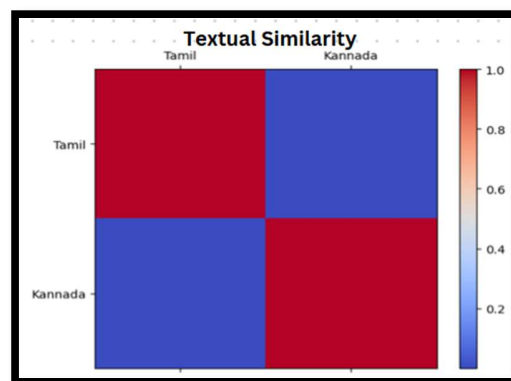


Figure 5.2.3C: Textual Similarity

These similarity metrics directly inform improvements in OCR accuracy, transliteration precision, and cross-language information retrieval systems, thereby enhancing overall multilingual text processing.

5.3XAI

5.3.1 For Classification

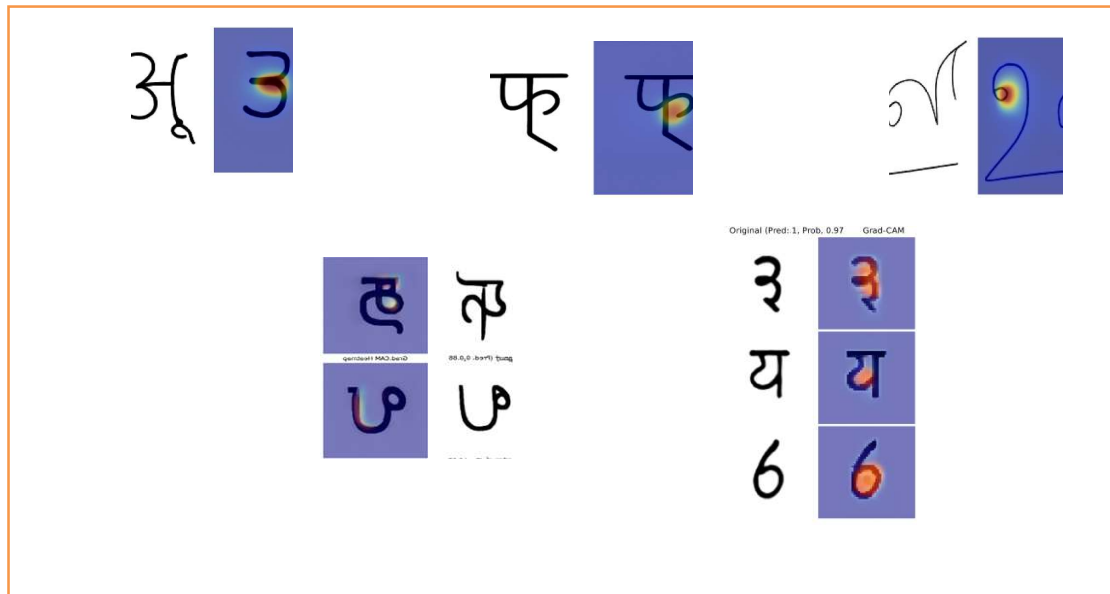


Figure 5.3.1: XAI for Classification

The Figure 5.3.1 explains how the CNN work behind the picture to classify the images.

- Red/Orange Regions: Areas that strongly supported the model's prediction
- Blue Regions: Areas that had minimal influence on the decision
- The visualization helps in checking whether the model focuses on linguistically needed features (like script patterns) rather than irrelevant elements.

5.3.2 For Letter Similarity

To improve the interpretability of similarity assessments between characters from languages within the same language family, we employed Explainable AI (XAI) techniques.

- Initially, all character images were **converted to grayscale, brought back to a uniform resolution**, and then again **normalized** to standard pixel intensity values across inputs.
- A **pixel-wise absolute difference** was calculated between the two images. By inverting these differences, we generated a **pixel similarity map**, where **brighter regions** indicate higher structural similarity, and **darker areas** represent greater dissimilarity.
- To further enhance visual interpretability, we created **overlay images** by blending the two characters. This technique helps in identifying **spatial alignment and structural correspondence** between the letters.

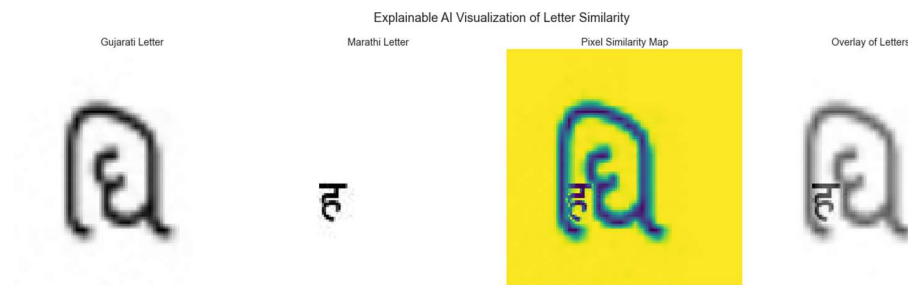
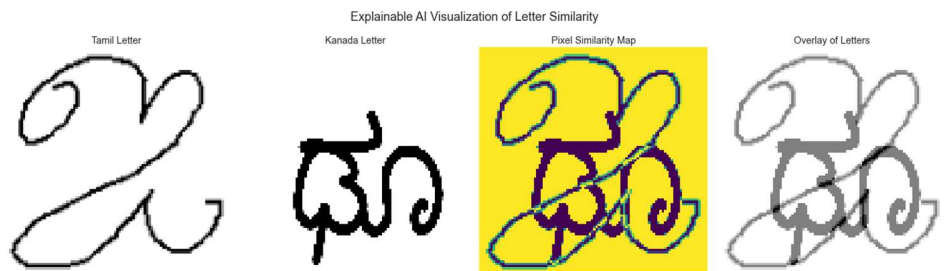


Figure 5.3.2A:
XAI
Implement
ation for
Devnagari
Script

Figure 5.3.2B: XAI
Implementation
for Dravidian
Script



Figures 5.3.2A and 5.3.2B illustrate this process:

- **Figure 5.3.2A** demonstrates the application of XAI for characters from the **Devanagari script**, comparing Gujarati and Marathi letters.
- **Figure 5.3.2B** showcases the analysis for the **Dravidian script**, comparing Tamil and Kannada characters.

These visualizations—particularly the **pixel similarity heatmaps** and **letter overlays**—offer a transparent and intuitive understanding of character resemblance. Unlike traditional numeric metrics (e.g., cosine similarity), this approach supports a **visually explainable evaluation**, aligning with the core principles of modern XAI and providing deeper insight into structural similarities between characters.

Ch. 6. Conclusion

This paper is a significant contribution to multilingual script comparison as viewed from the POSACLE approach, which is an integration of deep learning, machine learning, and similarity measures for classifying and comparing Indian scripts on a fine-grained, character-based level. Using the integration of visual features and phonic similarities, POSACLE can distinguish visually and linguistically complicated scripts—achieving 98.57% classification accuracy for Indo-Aryan scripts and over 96% for Dravidian scripts.

Aside from classification, the model provides deeper understanding of structural relations between Indian languages, discovering patterns that reflect evolutionary or historical relations—especially between the Dravidian scripts, which showed divergence and selective similarities. Adding explainability through XAI techniques like Grad-CAM also enhances model transparency through deeper understanding of what affects the model's decision.

Most importantly, POSACLE is not only precise but also usable in actual practice—providing solutions to the building of OCR, cross-language information retrieval, translation, and linguistic research. It is thus a holistic tool that fills the gap between computational methods and cultural-linguistic preservation.

While the findings of this research are promising, there are a couple of inherent limitations that need to be taken into account:

- **Data Constraints:** The study is limited by the availability of data sets, especially for handwritten and low-resource scripts. This restricts application of the framework to all Indian languages.
- **Algorithmic Limitations:** Reliance on cosine similarity to calculate script similarity is oversimplifying the problem by not taking into account non-linear relationships and context-dependent glyph subtleties that can miss deeper structural patterns.
- **Loss of Spatial Hierarchies:** For similarity comparison, flattening images to vectors resulted in loss of significant spatial hierarchies needed in encoding the full richness of script morphology.
- **Computational Trade-offs:** Deep learning algorithms are computationally expensive, and this can be a problem when deploying such solutions. in low-resource environments, e.g., in real-time systems or mobile devices.

Ch. 7. Future Scope

With this outset, several potential directions for further research can be developed to further increase the scope and strength of the framework:

1. Scaling Script and Language Support

- Add additional scripts of the Brahmic family and additional less-studied Indian languages, like handwriting and cursive scripts, to reach maximum use.

2. Advanced Similarity Metrics

- Exceed cosine similarity by utilizing deep metric learning approaches like triplet networks or graph-based approaches to learn more sophisticated relations among scripts and characters.

3. Hybrid Model Development

- Integrate CNN feature extraction with transformer models for context-pervasive analysis, capturing local visual features and global script structures.

4. Real-World Deployment and Optimization

- Create lean, efficient models that are appropriate for application in mobile OCR scenarios, with emphasis in low-resource language support and offline use.

5. Historical and Cultural Research Applications

- Expanding the system to include historical document analysis to assist linguists and historians to follow script development and cultural exchanges between language families.

6. Enhanced Error Analysis

- Conduct rigorous examinations of mislabeled samples to enhance the model's sensitivity to fine script differences, ideally based on contextual and phonetic information.

Essentially, this study is a robust foundation for technological innovation and linguistic scholarship alike, both contributing to a deeper understanding of India's rich scriptural tradition and facilitating real-world applications in digitization, educational practices, and cultural heritage conservation.

References

- [1] P. Mark and R. Burling, "The Tibeto-Burman languages of Northeast India," 2017. [Online]. Available: <http://dx.doi.org/10.4324/9781315399508>
- [2] K. Sreelakshmi, B. Premjith, B. R. Chakravarthi, and K. P. Soman, "Detection of Hate Speech and Offensive Language CodeMix Text in Dravidian Languages Using Cost-Sensitive Learning Approach," *IEEE Access*, vol. 12, pp. 20064-20090, 2024. [Online]. Available: <http://dx.doi.org/10.1109/ACCESS.2024.3358811>
- [3] C. S. Anoop and A. G. Ramakrishnan, "Exploring a Unified ASR for Multiple South Indian Languages Leveraging Multilingual Acoustic and Language Models," in *Proc. IEEE Spoken Language Technology Workshop (SLT)*, Doha, Qatar, 2023, pp. 830-837. [Online]. Available: <http://dx.doi.org/10.1109/SLT54892.2023.10022380>
- [4] A. Rameez and S. Madhukumar, "Hate Speech Detection in Indian Languages: A Brief Survey," in *Proc. IEEE ICDDs*, 2023, pp. 1-5. [Online]. Available: <http://dx.doi.org/10.1109/ICDDs59137.2023.10434756>
- [5] A. Jha, H. Y. Patil, S. K. Jindal, and S. M. N. Islam, "Multilingual Indian Language Neural Machine Translation System Using mT5 Transformer," in *Proc. 2nd Int. Conf. Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS)*, Nagpur, India, 2023, pp. 1-5. [Online]. Available: <http://dx.doi.org/10.1109/PCEMS58491.2023.10136051>
- [6] V. Rajesh, B. Permual, B. Prasanna, B. Haripriya, R. Sravani, and S. Nandini, "Text Translation for Indian Languages," in *Proc. IEEE ViTECoN*, 2023, pp. 1-5. [Online]. Available: <http://dx.doi.org/10.1109/ViTECoN58111.2023.10157002>
- [7] V. Venkatesh and A. Farghaly, "Identifying Anomalous Indus Texts from West Asia Using Markov Chain Language Models," in *Proc. IEEE ICCNT*, 2023, pp. 1-7. [Online]. Available: <https://doi.org/10.1109/ICCNt56998.2023.10306531>
- [8] A. Prakash and H. A. Murthy, "Exploring the Role of Language Families for Building Indic Speech Synthesizers," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 734-747, 2023. [Online]. Available: <https://doi.org/10.1109/TASLP.2022.3230453>
- [9] A. H. Patil, S. S. Patil, S. M. Patil, and T. P. Nagarhalli, "Real Time Machine Translation System between Indian Languages," in *Proc. IEEE ICOEI*, 2022, pp. 1778-1783. [Online]. Available: <https://doi.org/10.1109/ICOEI53556.2022.9777103>
- [10] D. Baishya and R. Baruah, "Recent Trends in Deep Learning for Natural Language Processing and Scope for Asian Languages," in *Proc. IEEE ICAISS*, 2022, pp. 408-411. [Online]. Available: <https://doi.org/10.1109/ICAISS55157.2022.10010807>
- [11] T. Gore and V. Khatavkar, "Development of Part-of-Speech Tagger for a Low-Resource Endangered Language," in *Proc. IEEE ICAC3N*, 2022, pp. 1531-1535. [Online]. Available: <https://doi.org/10.1109/ICAC3N56670.2022.10074031>
- [12] S. Bhattacharjee et al., "IITG-INDIGO Submissions for Interspeech-2021 Multilingual and Code-Switching ASR Challenges for Low Resource Indian Languages," in *Proc. IEEE INDICON*, 2021, pp. 1-5. [Online]. Available: <https://doi.org/10.1109/INDICON52576.2021.9691549>
- [13] V. M. Shetty and N. J. Sagaya Mary, "Improving the Performance of Transformer Based Low Resource Speech Recognition for Indian Languages," in *Proc. IEEE ICASSP*, 2020, pp. 8279-8283. [Online]. Available: <https://doi.org/10.1109/ICASSP40776.2020.9053808>
- [14] N. Baruah, S. K. Sarma, and S. Borkotokey, "Text Summarization in Indian Languages: A Critical Review," in *Proc. IEEE ICACCP*, 2019, pp. 1-6. [Online]. Available: <https://doi.org/10.1109/ICACCP.2019.8882968>

- [15] G. I. Ahmad, J. Singla, and N. Nikita, "Review on Sentiment Analysis of Indian Languages with a Special Focus on Code Mixed Indian Languages," in Proc. IEEE ICACTM, 2019, pp. 352-356. [Online]. Available: <https://doi.org/10.1109/ICACTM.2019.8776796>
- [16] S. Deleep Kumar, C. Sunitha, and A. Ganesh, "Semantic Representation of Texts in Indian Languages — A Review," in Proc. IEEE ICISC, 2018, pp. 40-42. [Online]. Available: <https://doi.org/10.1109/ICISC.2018.8399119>
- [17] R. Saikia, S. R. Singh, and P. Sarmah, "Effect of Language Independent Transcribers on Spoken Language Identification for Different Indian Languages," in Proc. IEEE IALP, 2017, pp. 214-217. [Online]. Available: <https://doi.org/10.1109/IALP.2017.8300582>
- [18] S. Saini and V. Sahula, "A Survey of Machine Translation Techniques and Systems for Indian Languages," in Proc. IEEE CICT, 2015, pp. 676-681. [Online]. Available: <https://doi.org/10.1109/CICT.2015.123>
- [19] C. H. Sánchez-Gutiérrez, S. Minnillo, P. F. Mira, and A. Hernández, "Prompt response variation in learner corpus research: Implications for data interpretation," *Research Methods in Applied Linguistics*, vol. 3, no. 3, p. 100134, 2024. [Online]. Available: <https://doi.org/10.1016/j.rmal.2024.100134>
- [20] P. Pérez-Paredes and N. Curry, "Epistemologies of corpus linguistics across disciplines," *Research Methods in Applied Linguistics*, vol. 3, no. 3, p. 100141, 2024. [Online]. Available: <https://doi.org/10.1016/j.rmal.2024.100141>
- [21] A. Andayani et al., "The role of systemic functional linguistics in improving students' ability in speaking skills," *Linguistic and Philosophical Investigations*, vol. 23, no. 1, pp. 1–121, 2024. [Online]. Available: <https://philolinginvestigations.com>
- [22] K. Indasari, S. Abubakar, and A. Ahmad, "Digital narratives and the evolution of textual forms: A study of hypertext fiction," *Linguistic and Philosophical Investigations*, vol. 23, no. 1, pp. 202–207, 2024. [Online]. Available: <https://philolinginvestigations.com>
- [23] Z. Jia, T. Zhao, J. Ru, Y. Meng, and B. Xia, "SPNet: A serial and parallel convolutional neural network algorithm for the cross-language coreference resolution," *Computer Speech & Language*, vol. 91, p. 101729, 2025. [Online]. Available: <https://doi.org/10.1016/j.csl.2024.101729>
- [24] Yadav, M., & Kumar, I. (2023). Transliteration from Hindi to English Using Image Processing. <https://doi.org/10.1109/conit59222.2023.10205844>
- [25] Dhir, R., Mishra, S. K., Saha, S., & Bhattacharyya, P. (2019). A Deep Attention based Framework for Image Caption Generation in Hindi Language. *Computación Y Sistemas*, 23(3), 693–701. <https://doi.org/10.13053/CYS-23-3-3269>
- [26] Bhardwaj, M., Singh, S., Khanna, A., Gupta, A., Gupta, D., & Gupta, N. (2020). Detection of Devanagari Text from Wild Images Through Image Processing Techniques (pp. 675–686). Springer, Singapore. https://doi.org/10.1007/978-981-15-1286-5_60
- [27] Gupta, M. K., Dhawan, S. S., & Kumar, A. (2024). CMViT OCR: printed Indian language recognition using CMViT. 28–33. <https://doi.org/10.1109/spin60856.2024.10511992>
- [28] Dholakia, K. (2015). A Survey on Handwritten Character Recognition Techniques for Various Indian Languages. *International Journal of Computer Applications*, 115(1), 17–21. <https://doi.org/10.5120/20114-2159>
- [29] Sethi, G. K., & Bawa, R. K. (2012). Text Information Extraction: in Context of Indian Languages. *Advanced Materials Research*, 5012–5019. <https://doi.org/10.4028/WWW.SCIENTIFIC.NET/AMR.433-440.5012>

- [30] Garain, U., Datta, A., Bhattacharya, U., & Parui, S. K. (2006). Summarization of JBIG2 Compressed Indian Language Textual Images. *International Conference on Pattern Recognition*, 3, 344–347. <https://doi.org/10.1109/ICPR.2006.1090>
- [31] Garain, U., Datta, A., Bhattacharya, U., & Parui, S. K. (2006). Summarization of JBIG2 Compressed Indian Language Textual Images. 3, 344–347. <https://doi.org/10.1109/icpr.2006.1090>
- [32] <https://www.meity.gov.in/static/uploads/2024/02/33-1.pdf>