VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi, Karnataka-590018

SAN-ENG: SANSKRIT TO ENGLISH TRANSLATOR USING MACHINE LEARNING

(CEC/CS/2022/P20)

A PROJECT REPORT

Submitted by

SHETTY RAMAKRISHNA MOHAN	**4CB19CS096**
ROHAN S BHAT	**4CB19CS086**
RANJITH V SHETTY	**4CB19CS082**
ANIRUDDHA	**4CB19CS014**

In the partial fulfillment of the requirement for the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Under the guidance of

Ms. Saritha M

Assistant Professor

Department of Computer Science & Engineering

CANARA ENGINEERING COLLEGE

BENJANAPADAVU, BANTWAL- 574219, D.K. ,KARNATAKA

2022-2023

CANARA ENGINEERING COLLEGE
BANTWAL, D.K. 574219 – KARNATAKA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that the project work entitled "**SAN-ENG: SANSKRIT TO ENGLISH TRANSLATOR USING MACHINE LEARNING**" carried out by **Mr. SHETTY RAMAKRISHNA MOHAN** bearing USN **4CB19CS096**, **Mr. ROHAN S BHAT** bearing USN **4CB19CS086**, **Mr. RANJITH V SHETTY** bearing USN **4CB19CS082**, **Mr. ANIRUDDHA** bearing USN **4CB19CS014**, a bonafide students of **CANARA ENGINEERING COLLEGE, BENJANAPADAVU** in partial fulfillment for the award of **BACHELOR OFENGINEERING in COMPUTER SCIENCE AND ENGINEERING** of the **VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI** during the year **2022-2023**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribedfor the said Degree.

Asst. Prof.Saritha M Project Guide Dept. of CSE	**Dr. Ganesh V. Bhat** Principal, CEC	**Dr. Sunil Kumar B L** Professor & Head Dept. of CSE

Name of the Examiners	**Signature with Date**
1	
2	

ACKNOWLEDGEMENT

Any Achievement, be in scholastic or otherwise does not depend solely on the individual effort but on the guidance, encouragement and co-operation of intellectuals, elders and friends. A number of personalities in their own capacity have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

Our sincere thanks to internal guide **Asst**. **Prof. Saritha M, Department of Computer Science and Engineering, CEC, Mangalore,** for her valuable guidance and support throughout the course of project work and for being a constant source of inspiration.

We extend our thanks to **Prof. Ajay Shet**, Project coordinator, **Department of Computer Science and Engineering, CEC, Mangalore,** for his valuable suggestions throughout all the phases of the Project Work.

With profound sense of gratitude and respect, we thank **Dr. Sunil Kumar B L, HOD, Department of Computer Science and Engineering** and **Dr. Ganesh V. Bhat, Principal, Canara Engineering College Benjanapadavu Mangalore,** for all the facilities and help.

We thank the management of **Canara Engineering College** for the support throughout the course of Bachelor Degree and all the facilities they have provided.
Last, but certainly not the least we thank all teaching and non-teaching staff of CEC for guiding us in the right path. Most importantly we wish to thank our parents for their support and encouragement.

We are also thankful to college library for providing us with necessary books as and when required, which made our task of tracing the logic of the source codes much easier.

SHETTY RAMAKRISHNA MOHAN **4CB19CS096**
ROHAN S BHAT **4CB19CS086**
RANJITH V SHETTY **4CB19CS082**
ANIRUDDHA **4CB19CS014**

ABSTRACT

Sanskrit to English translation is a challenging task due to the differences in morphology and structure between the two languages. In recent years, the use of OCR and Encoder-Decoder models has shown promising results in this field. OCR is used to extract text from input images, while Encoder-Decoder models such as LSTM-based models are used for sequence-to-sequence translation. Attention mechanisms have further improved the accuracy of the models by allowing them to focus on important parts of the input text. However, there is still room for improvement in Sanskrit to English translation, such as creating more extensive datasets, fine-tuning pre-trained models, using hybrid models, and developing multi-lingual models. This abstract provides an overview of the current state of the art in Sanskrit to English translation using OCR and Encoder-Decoder models, as well as potential areas for future research.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Neural Machine Translation is a popular machine learning technique used for translating texts from one language to another. Currently, tools are available to translate most popular languages like German, Hindi, Punjabi, and Spanish, yet there is still no successful tool for translating Sanskrit. Sanskrit is an important language for the Hindu religion and is still used to this day. In order to create an effective automated written translator of full sentences, San-Eng has been developed. This machine translation model, Encoder-Decoder LSTM, works by understanding each word in a sentence based on its understanding of previous words and the context of the sentence. Previous research has found two methods for translating Sanskrit to English, Statistical Machine Translation and a general algorithm of top-down processing.

The aim of San-Eng is to develop an automated Sanskrit-English sentence translation model. It will use the Encoder-Decoder LSTM model, which is a specialized recurrent neural network that takes context into account when translating. Previous research has shown that Sanskrit is a viable language for natural language processing, as its grammar and syntax are easier to understand. Two methods have been explored for translating Sanskrit to English: Statistical Machine Translation and top-down processing. San-Eng will combine these methods to create an effective translation model.

1.2 PROBLEM STATEMENT

For a given Sanskrit text image apply the image pre-processing technique, extract the Sanskrit text from image using OCR, then apply NMT to accurately identify Sanskrit characters and words to translate it to English.

1.3 SCOPE OF THE PROJECT

These are the following main scopes of our project:

- **Language Support**: The language translator should be designed to support a specific set of source and target languages, depending on the intended use case.

- **Input and Output Format**: The system should be able to handle different types of inputs such as text, speech, or images, and produce output in the desired format such as text or speech.
- **Translation Accuracy**: The project should aim to achieve a high degree of accuracy in translating the input language to the target language while preserving the meaning and context.
- **Language Model Selection**: The project should select appropriate language models and algorithms based on the language pair and the type of input.

1.4 OBJECTIVE

The objectives of a language translator project are to accurately translate text or speech from one language to another while preserving the meaning and context, and to facilitate cross-cultural communication and collaboration.

CHAPTER 2

LITERATURE SURVEY

S. Kondo, Et al. "**Machine Translation with Pre-specified Target-side Words Using a Semi-autoregressive Model"** the author describes a Japanese-to-English translation system that uses a semi-autoregressive model called Recover SAT to insert specified words (RTVs) into the output sentence. The system sorts the RTVs in the order of their corresponding words or phrases in the source sentence, resulting in over 96% inclusion of all RTVs in the output sentences. The paper uses the ASPEC dataset for Japanese-to-English translation and evaluates system outputs using BLEU score and consistency score. BLEU score measures n-gram matching rate with the reference, and consistency score is the ratio of translations that satisfy exact match of all given constraints. Future work will focus on determining the best order to insert RTVs. [1]

R. K. Chakrawarti, **"Machine translation model for effective translation of Hindi poetries into English"** the author proposes a hybrid machine translation approach for Word Sense Disambiguation of Hindi lyrics into English, which combines rule-based and statistical machine translation. The proposed approach uses a trie-based dictionary and handcrafted rules to identify accurate words, and erroneous words are corrected by providing a list of valid words. The system can be further improved by using machine translation trained with data from various sources. The proposed approach uses a tokenization process and tree-based algorithm for translation and outlines four stages for successful conversion. The paper highlights the difficulty of translating a Hindi poem into English and suggests future research to analyse the quality of translations dealing with complex sentences and various varieties of the same word. [2]

A. Pathak, **"Neural Machine Translation for Indian Language"** the author discusses the use of Neural Machine Translation (NMT) systems to bridge communication barriers between people of different linguistic backgrounds. The NMT systems were trained, tested, and evaluated for English to Tamil, English to Hindi, and English to Punjabi translations, using parallel corpora. The paper shows that NMT produces fluent translations and the translation quality can be improved by increasing the size of the training corpus, optimizing system parameters, and selecting an appropriate score function for computing attention. The study highlights the importance of linguistic resources and context analysis in machine translation systems. [3]

B. Premjith, **"Neural Machine Translation System for English to Indian Language Translation Using MTIL Parallel Corpus"** the author discusses the development of a neural machine translation system for translating between English and four Indian languages, namely Malayalam, Hindi, Tamil, and Punjabi. The system uses deep neural networks, including LSTM networks and Bi-RNN, to model the translation process. The paper emphasizes the importance of using large, diverse corpora to improve translation quality and incorporating linguistic features. The system was evaluated using BLEU scores and manual metrics such as adequacy, fluency, and overall ranking. The paper also notes that the length of sentences should be appropriate for deep learning architectures to extract long dependencies present in long sentences. The corpus developed for this study is a contribution to the machine translation research community. [4]

Sitender, **"Sanskrit to universal networking language EnConverter system based on deep learning and context-free grammar"** the author proposes an extension to an existing machine translation system (MTS) for Sanskrit that includes a stemmer, neural network for POS tagging, Sanskrit grammar, and CYK parser. The proposed system has seven layers, each performing a different task, and includes a new Sanskrit to UNL EnConverter system. The proposed system has reported an average BLEU score and average fluency score with overall efficiency. In the future, deep neural networks could be used to generate all UNL relations automatically. Additionally, the proposed parse tree generation algorithm and POS tagging technique may be used for parsing and POS tagging for other languages as well. The paper also reviews various machine translation systems based on the UNL approach and neural machine translation (NMT) systems. NMT is an extension of statistical machine translation and performs end-to-end translation. RNNs were proposed and used an encoder-decoder approach for translation. Several companies, including Microsoft, Google, Facebook, and Amazon, have launched their machine translation systems using NMT approach. [5]

Rivera-Trigueros, **"Machine translation systems and quality assessment: a systematic review"** the author presents a systematic literature review of Machine Translation (MT) systems for the English-Spanish language combination. The review focuses on the most employed MT systems, their architectures, and the quality assessment procedures used to determine their performance. The research findings indicate that neural MT is the predominant paradigm, with Google Translator being the most used system. The majority of the works assessed MT using either automatic or human evaluation, and more than half of the works included error classification and analysis. Despite the limited sample size due to selection criteria, it was concluded that MT is a growing area with great potential for overcoming

language barriers and increasing productivity. Additionally, the study observes that Deep Learning has not been widely used in MT research and suggests that it should be included in future research for comparison with the current predominant system. [6]

O. Hellwig, **"Obtaining More Expressive Corpus Distributions for Standardized Ancient Languages"** the author presents a latent variable model that quantifies the influence of early authoritative works on the lexis of their literary successors in ancient languages. The model is applied to a corpus of pre-Renaissance Latin texts and is evaluated based on its ability to capture intellectual lineages and structures of word reuse. Results show that the model provides meaningful linguistic distributions that better describe certain aspects of language development than plain corpus distributions. Future extensions could consider non-temporal influence factors such as the geographic origin or genre of a text. The model has potential for application to other ancient text traditions to better understand their intellectual and diachronic structures. [7]

S. Scarlata, **"The Treebank of Vedic Sanskrit"** the author presents the first treebank of Vedic Sanskrit, a morphologically rich ancient Indian language, containing 4,000 sentences spanning 600 years of metrical and prose texts. The sentences are annotated with the Universal Dependencies scheme, with attention given to certain syntactic constructions. A syntactic label based on neural networks is provided to support the initial annotation of the treebank, and can be useful for setting up a full syntactic parser of Vedic Sanskrit. The paper also discusses the benefits of using Vedic Sanskrit in machine learning for natural language understanding and sentiment analysis, and describes the Universal Dependencies annotation scheme and its advantages. Finally, the paper describes the unique aspects of the Vedic Sanskrit treebank's annotation and suggests future applications of the treebank, including the use of neural parser architectures for morphologically rich languages. [8]

O. Hellwig, **"Dating and Stratifying a Historical Corpus with a Bayesian Mixture Mode"** the author proposes and evaluates a Bayesian mixture model for dating texts based on their linguistic features, with a focus on the Vedic Sanskrit corpus. The evaluation reveals linguistic features strongly correlated with time, but also potential problems in reconciling quantitative results with philological insights. The proposed model is applicable to any corpus with a disputed historical structure and can be evaluated through a case study on the Rigveda. The model is a prototype that can be extended in various aspects, with plans to replace inflexible admixture models with a Hierarchical Dirichlet Process and combine it with a Markov Random

Field to induce textual strata from the data. The paper provides an overview of textual chronology in machine learning, Vedic Sanskrit, and Bayesian mixture models. [9]

S. Sellmer, **"Detecting Diachronic Syntactic Developments in Presence of Bias Terms"** the author introduces an infinite relational model to study diachronic syntactic changes in Vedic Sanskrit texts. The model groups syntactic constituents based on their structural similarities and diachronic distributions, while controlling for register and intellectual affiliation. The results of the model are discussed for four syntactic structures in the texts, and the framework is shown to identify chronological signals from Middle and Late Vedic texts. However, controlling for the content of passages and qualitative scrutiny are necessary for a clearer picture. The paper also discusses historical syntax in machine learning and the Infinite Relational Model, which is used to model relationships between entities in a database. The paper proposes a data-driven framework to track chronological changes in Vedic syntax and suggests extending the framework to account for contextualized word embedding. [10]

P. Dhar, **"Optimal Word Segmentation for Neural Machine Translation into Dravidian Languages**" the author explores the effectiveness of Linguistically Motivated Vocabulary Reduction (LMVR) and Sentence Piece (SP) for sub word segmentation in Neural Machine Translation (NMT) from English to four Dravidian languages. Results indicate that SP is the best choice for segmentation and that larger sub word vocabularies lead to higher translation quality. The study found interesting differences among the four target languages, with Kannada being the most challenging. While LMVR results in shorter sub words, SP remains the best option for all language pairs. Further research is needed to improve translation quality in Dravidian languages. [11]

M. Singh,**"Machine Translation Systems for Indian Languages: Review of Modelling Techniques, Challenges, Open Issues and Future Research Directions"** the author provides a review of different modelling techniques used in machine translation and compares research on different Indic language pairs based on their modelling techniques. The paper suggests that more accurate and efficient techniques like Neural Machine Translation (NMT) and Hybrid Machine Translation (HBMT) need to be implemented for better translation results. The paper also highlights the minimal research that has been done on the Sanskrit language despite its ancient scientific and comprehensive literature, and suggests possible solutions to the linguistic and technical challenges of processing Sanskrit language. A comparison of research work on different Indic language pairs based on modelling techniques has been performed, and it was found that the use of SMT-based MTS is more as compared to other techniques, while the

Neural and Hybrid approaches perform better. The paper outlines the open issues, technical and linguistic challenges, and future research directions of Machine Translation for processing Sanskrit language. [12]

O. Hellwig, **"Dating Sanskrit texts using linguistic features and neural networks"** the author explores the diachronic nature of Classical Sanskrit and uses machine learning algorithms to provide approximate dates of composition for Sanskrit texts. The paper argues that a quantitative approach can detect traces of diachronic change in Classical Sanskrit and derive historical dates from them. The paper evaluates Book 6 of the Mahabharata, which has been debated in Indological research. The algorithm predicts the hidden value based on input features and quantifies the prediction errors. The paper suggests that the relationship between linguistic features and times contains non-linear interactions, which are better captured by a multi-layer neural network than by linear regression. The paper goes beyond unsupervised methods by coupling linguistic features directly with historical time, using a supervised approach to dating and text stratification. [13]

R. Haque **,"Investigating Terminology Translation in Statistical and Neural Machine Translation: A Case Study on English-to-Hindi and Hindi-to English"** the author compares the performance of phrase-based statistical machine translation (PB-SMT) and neural machine translation (NMT) systems for English-to-Hindi and Hindi-to-English translations in the legal domain. The authors propose an error typology for terminology translation errors and evaluate the systems based on this typology. Results show that NMT performs better than PB-SMT in terms of lexical, reordering, and morphological errors. NMT also performs well in translating unknown terms using open-vocabulary translation. However, NMT can sometimes produce strange or non-existent wordforms or repetitions. The paper emphasizes the need to consider the diversity of term translations in automated terminology translation evaluation processes. The study also measures agreement in manual classification of terminology translation errors. [14]

S. R. Laskar, **"Multimodal Neural Machine Translation for English to Hindi"** The author describes their participation in the WAT2020 translation task for multi-modal translation from English to Hindi, and reports that their multi-modal NMT system achieved higher scores than their text-only NMT system on both the challenge and evaluation test sets. The multi-modal NMT system combined textual and visual features to improve translation quality for low resource languages, and utilized a pre-trained CNN with VGG19 for the extraction of global and local features from provided image datasets. GloVe was used to pretrain on monolingual

data of English-Hindi and generate global vectors of word embedding. The author aims to further improve the performance of their multi-modal NMT system in future work. [15]

S. Nehrdich, **"Sanskrit Word Segmentation Using Character-level Recurrent and Convolutional Neural Networks"** the author presents a novel end-to-end neural network model for tokenizing Sanskrit that incorporates both local phonetic and distant semantic features. The model outperforms previous approaches to Sanskrit word segmentation and is also shown to beat the state of the art for German compound splitting, demonstrating its language agnosticity. The models are character-based and operate in a sequence labeling framework, with convolutional and recurrent elements. The paper has three main contributions: introducing novel models for SWS that beat state of the art models, demonstrating that the models work on par with sequence-to-sequence models but require less time for training and inference, and publishing a new dataset for Sanskrit word splitting that consists of more than 560,000 sentences with manually validated splits. The authors plan to expand their research by exploring joint learning of splits, lexical and morphological annotations, and hypothesize that CTC and segmental NNs with modified objectives may be suitable for this task. [16]

W. Lu , **"Word sense disambiguation based on dependency constraint knowledge"** the author describes a word sense disambiguation (WSD) method based on dependency constraint knowledge, which utilizes dependency parsing to construct a knowledge base and compute posterior probabilities for each sense of an ambiguous word. This method has achieved the best performance among unsupervised and knowledge-based methods in the SemEval dataset. The paper discusses three categories of WSD methods, including supervised, unsupervised, and knowledge-based methods, and highlights the advantages of the knowledge-based method. The paper suggests that further work is needed to enrich the dependency constraint knowledge base and evaluate the types of reserved dependency constraint cells more carefully to improve the method's performance. Overall, the method presented in this paper is demonstrated to be superior to other unsupervised and knowledge-based methods in experiments. [17]

A. R. Pal, **"Word Sense Disambiguation in Bangla Language Using Supervised Methodology with Necessary Modifications"** the author proposes a supervised methodology for word sense disambiguation in Bangla language, using Naive Bayes as a baseline and incorporating lemmatization and bootstrapping. The method achieves an accuracy of 84% using the Bangla POS tagged corpus and Bangla WordNet. The paper also discusses the challenge of differentiating functional and non-functional words in Bangla, and the need for manual intervention in the learning process. This work is a start to devising new strategies for

South Asian languages, which have complex syntax and morphology. The method incorporates sense-resolute test data into the training sets to improve accuracy in subsequent executions. Manual intervention is required to correct misclassified instances, as the model is not always accurate. [18]

W. Lu, **"The Vedic corpus as a graph. An updated version of Bloomfield's Vedic Concordance"** the author discusses an updated and extended version of Bloomfield's Vedic Concordance, which has been transformed into a strict XML structure to enable programmatic access to the data. The paper presents case studies demonstrating how the new version of the Concordance can be used in textual studies and suggests possible research applications such as studying the types of links and inferring the direction of edges in the graph to provide further insights into the structure and development of the Vedic canon. The paper also discusses the importance of mantra citations for reconstructing the history of Vedic literature and presents a lean user interface based on PHP and Xpath for querying and extending the XML version of the Concordance. The paper concludes by discussing the need to merge the data into a format that is platform independent and easily processed by programming languages. [19]

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 PURPOSE

Facilitating Communication: A San-Eng translator helps to bridge the language gap between Sanskrit speakers and English speakers. It allows people to communicate and understand each other, especially in cases where Sanskrit is used in religious or academic contexts.

Promoting Cross-Cultural Understanding: Sanskrit is an ancient language that has a rich cultural heritage. A Sanskrit to English translator helps to promote cross-cultural understanding by making Sanskrit literature, history, philosophy, and spirituality accessible to a wider audience.

Enhancing Language Learning: Sanskrit is considered one of the mother tongues of many Indian languages. A San-Eng translator can be helpful for language learners who want to learn Sanskrit vocabulary and grammar, as well as to understand the nuances of the language.

Understanding ancient Indian texts: Sanskrit is an ancient Indian language that is the liturgical language of Hinduism, Buddhism, and Jainism, and is also the language of many ancient Indian texts, including the Vedas, Upanishads, and Bhagavad Gita. A Sanskrit to English translator can help scholars, researchers, and students who are studying these texts to better understand their meaning and significance.

3.2 SCOPE OF THE PROJECT

These are the following main scopes of our project:

- **Religious Studies:** Sanskrit is an ancient language used in many religious texts, such as the Hindu scriptures, Buddhist texts, and Jain texts. As it can help people understand the meaning and significance of religious concepts and terminology.
- **Cultural Exchange:** By making Sanskrit literature, history, and philosophy accessible to a wider audience, a translator can help people from different cultural backgrounds appreciate and understand Indian culture.
- **Legal and Administrative Documentation:** Sanskrit is also used in legal and administrative documentation in India, particularly in traditional systems such as Ayurveda and Jyotish. Includes translating legal and administrative documentation to facilitate crossborder transactions and to ensure legal compliance.

- **Business and Industry:** Sanskrit is also used in many industries, particularly in the fields of Ayurvedic medicine, yoga, and astrology. The scope of a Sanskrit to English translator includes translating marketing materials, product descriptions, and industry-specific terminology to help businesses reach a wider audience and expand their markets.

3.3 DEFINITIONS, ACRONYMS & ABBREVIATIONS

SMT - Statistical machine translation

NMT - Neural machine translation.

OCR - Optical Character Recognition.

Statistical machine translation: Statistical machine translation (SMT) is a machine translation method that uses statistical models to automatically translate text from one language to another. It involves analysing and processing large amounts of bilingual text to learn the statistical patterns of language and translate based on probability. The approach uses algorithms and techniques from both computer science and statistics to build models that can capture the relationship between words and phrases in different languages.

Neural machine translation: Neural machine translation (NMT) is a machine translation approach that uses artificial neural networks to translate text from one language to another. NMT is based on the principles of deep learning, a subset of machine learning, which involves training neural networks on large amounts of data to learn patterns and relationships in the input data. In NMT, a neural network is trained on bilingual corpora to learn the mapping between the source and target languages. The network is then able to generate translations by processing the source language input and producing a target language output.

Optical Character Recognition: Optical Character Recognition (OCR) is a process of converting scanned images or documents into text that can be easily searched, edited, and shared. OCR technology enables the automated recognition of printed or handwritten characters in an image or document, and their conversion into machine-readable text.

3.4 OVERVIEW OF THE DOCUMENT

The SRS will include two sections:

- Overall Description: This section will describe major parts of the system and their connections.

- Specific Requirements: This section will describe the function of the system and the constraint faced by the system.

3.5 OVERALL DESCRIPTION

3.5.1 PRODUCT PERSPECTIVE

- **Data acquisition**: Gather a large amount of parallel Sanskrit-English data for training the machine learning model.
- **Data pre-processing**: Clean and standardize the data to prepare it for machine learning.
- **Feature engineering**: Design features that represent the Sanskrit and English texts in a way that can be learned by the machine learning model.
- **Model selection and training**: Select an appropriate machine learning model and train it on the pre-processed data.
- **Evaluation and testing**: Assess the accuracy and performance of the trained model using standard metrics and evaluation methods.
- **Integration and deployment**: Integrate the trained model into a usable product, such as a web or mobile application.

3.5.2 PRODUCT FUNCTIONS

- **Image recognition**: Use optical character recognition (OCR) algorithms to identify the Sanskrit text in the image.
- **Text extraction**: Extract the identified Sanskrit text from the image for translation.
- **Translation engine**: Use machine learning algorithms to translate the extracted Sanskrit text into English.
- **Output text**: Display the translated English text to the user.
- **Training and updating**: Continuously improve the OCR and translation engines through user feedback and updating the machine learning models.
- **Integration and deployment**: Make the image-to-text translation system available as a web or mobile application or integrate it into existing translation software.

3.5.3 USER CLASSES AND CHARACTERISTICS

- **Language Learners**: One group of users for a Sanskrit to English translator is language learners who are interested in learning Sanskrit as a second language. They may be interested in using the tool to translate Sanskrit vocabulary and grammar, as well as to read and understand Sanskrit literature and poetry.
- **Researchers and Academics**: Researchers and academics who study Sanskrit literature, history, philosophy, and religion may also use a Sanskrit to English translator to translate Sanskrit texts and manuscripts. They may be interested in understanding the context and meaning of the text, as well as analysing the linguistic and cultural significance of the text.
- **Spiritual Practitioners**: Sanskrit is the language of many ancient Indian scriptures and is used in many spiritual practices such as yoga and meditation. Spiritual practitioners may use a Sanskrit to English translator to better understand the meaning of Sanskrit words and concepts used in their practice.

3.5.4 DESIGN AND IMPLEMENTATION CONSTRAINTS

- **Accuracy**: As Sanskrit is a complex language with a rich cultural history, designing a translator that accurately reflects the meaning and context of the original text can be challenging. There are multiple interpretations of Sanskrit texts, and there may be different regional variations of the language. These factors can make it difficult to create a translation tool that accurately captures the intended meaning of the original text.
- **Grammar Rules**: Sanskrit has a complex grammar system, including declensions, conjugations, and cases, which can make it difficult to design a translator that correctly handles all possible grammatical structures. A translator must be designed to recognize and apply the correct grammar rules to ensure accurate translations.
- **Cultural and Linguistic Context**: Sanskrit has a rich cultural and linguistic context that is embedded in its literature, poetry, and religious texts. Designing a translator that takes into account these cultural and linguistic nuances can be challenging, as it requires a deep understanding of Sanskrit culture and language.
- **Technical Constraints**: Implementing a Sanskrit to English translator requires advanced NMT techniques and technologies, such as machine learning, deep learning, and artificial intelligence.

3.6 SPECIFIC REQUIREMENTS

3.6.1 EXTERNAL INTERFACE REQUIREMENTS

3.6.1.1 USER INTERFACES

- **Input Button**: The first component of the interface would be an input button where users can enter the Sanskrit image they want to translate into English. This should be prominent and easily visible to the user.
- **Output box**: The translated English text should be displayed in a separate output box, which should be clearly labelled and located near the input box. The output box should be large enough to display the translated text without requiring the user to scroll or resize the window.
- **Translate button**: A prominent "Translate" button should be placed near the input box to initiate the translation process. This button should be large and easy to click, with clear text labelling.
- **Error messages**: If the user enters text that is not recognized by the translator or if there is an error during the translation process, a clear error message should be displayed in the output box.

3.6.1.2 HARDWARE INTERFACES

- **Input device**: This could be a keyboard that allows users to input text or image in Sanskrit. Processing unit: This could be a computer or a mobile device that performs the translation process. It could include a central processing unit (CPU), a graphics processing unit (GPU), and memory to store data and algorithms.
- **Translation software**: This is the main component of the system and would be responsible for converting Sanskrit text or speech to English. The software could be built using machine learning algorithms, NMT techniques, and other tools.
- **Output device**: This could be a screen that displays the translated text in English.

3.6.1.3 SOFTWARE INTERFACES

- **Tesseract**: Tesseract is a Python wrapper for Google's Tesseract-OCR engine, which is an open-source OCR (Optical Character Recognition) engine used to recognize text in images. Tesseract allows Python developers to use Tesseract's OCR capabilities within their Python code

3.6.2 FUNCTIONAL REQUIREMENTS

- **User-Friendly Interface**: Should have a user-friendly interface that allows users to input text or speech in Sanskrit and receive a translated output in English. The interface should be easy to navigate, and the translation process should be intuitive.
- **Accuracy**: The translation software or tool should accurately translate Sanskrit text to English, without losing the meaning or context of the original text.
- **Speed**: A Sanskrit to English translator should also be able to translate text quickly and efficiently. Users should be able to input the text to be translated, and the software should provide the translation within a reasonable time frame.
- **Offline Access**: The translator should provide offline access, allowing users to use the application without an internet connection
- **Operating System**: The system can run on any modern operating system, such as Windows, MacOS, or Linux.
- **Tesseract**: is an open-source OCR (Optical Character Recognition) software that is capable of recognizing text in various languages, including Sanskrit. Tesseract is known for its accuracy and reliability in text recognition and has been widely used in various applications, including document scanning, image processing, and text-to-speech systems.
- **Google Colab**: is a cloud-based platform for developing and running Python code. It is a free service offered by Google that allows users to write and execute Python code in a web browser, using a Jupyter notebook interface.

3.6.3 PERFORMANCE REQUIREMENTS

- **Speed:** The translator should be able to translate text quickly and efficiently, with minimal delay or lag time. Users should be able to input the text to be translated and receive the translated output within a reasonable time frame.
- **Accuracy**: The primary performance requirement for a Sanskrit to English translator is accuracy. The translated output should accurately reflect the meaning and context of the original text, without losing or distorting any of the original meaning.
- **Scalability:** The translator should be able to handle a large volume of text, without experiencing any performance issues or delays. It should be able to handle text of

varying lengths and complexities, including long-form texts, poetry, and technical terminology.

3.6.4 DESIGN CONSTRAINTS AND ATTRIBUTES

- **Availability of Sanskrit text data**: The system's effectiveness and accuracy will depend on the availability and quality of Sanskrit text data used for training and improving the machine learning algorithms.
- **Computational resources:** The system's performance will depend on the computational resources available for training and inference, such as processing power and memory.
- **Accuracy:** The system must have a high level of accuracy in recognizing Sanskrit text in images and translating it to English text.
- **Speed:** The system should be fast enough to provide near real-time translation of Sanskrit text to English text.
- **User-friendliness:** The system should have a user-friendly interface that is easy to use and understand, even for non-technical users.
- **Adaptability:** The system should be adaptable and flexible, allowing for continuous improvement through user feedback and updates to the machine learning algorithms.
- **Accessibility:** The system should be accessible to a wide range of users, including those with disabilities or who use assistive technologies.

CHAPTER 4

SYSTEM DESIGN

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. The design of a system is perhaps the most critical factor affecting the quality of the software, and has a major impact on the later phases, particularly testing and maintenance. The design activity is often divided into separate phase i.e. system design and detailed design.

4.1 ARCHITECTURAL DESIGN

Architecture has emerged as a crucial part of design process. The architectural design of a system is abstract, distilling away details of implementation, algorithm and data representation and concentration on behaviour and interaction of "black box" elements. Software Architecture is developed as the first step towards designing a system that has collection of desired properties.

An architecture description is a formal description that illustrates the structure and the behaviour of the system. It defines the system components or the building blocks that must work together in tandem, to implement the overall system. Figure 4.1 shows the architectural design of virtual proctor system.

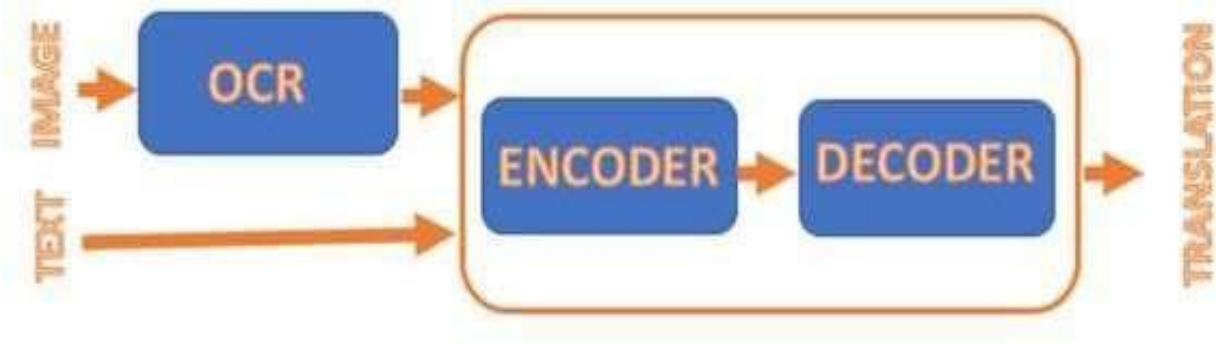


Fig 4.1 Architectural Design for San-Eng Translator

4.2 USE-CASE DIAGRAM

The main representation of system/software requirements for a new, developing software is a use case diagram. Use cases describe expected behavior, not the precise way it will be accomplished. By describing all externally observable system behavior, it is a useful tool for explaining system behavior to users. It depicts a list of actions or event steps, typically defining the interactions between a role and a system, to achieve a goal.

The Use-Case Diagram of the proposed system is shown in figure 4.2. Users are first required to choose an input type which can either be text or an image. In case of image input,

text extraction from the input image is carried out and the extracted Sanskrit text is fed to the translation model to get equivalent English translations.

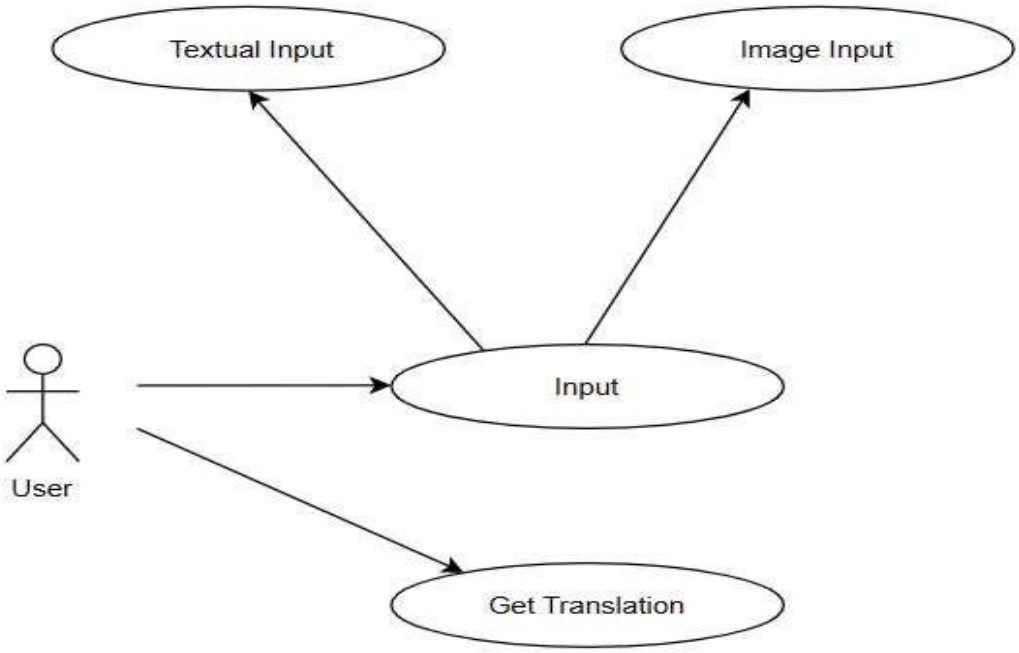


Fig 4.2 Use-Case Diagram for San-Eng Translator

4.3 MODULAR DESIGN DIAGRAM

A modular design is a method of product design that combines or integrates smaller, independently functioning elements to create finished goods. A complex product can be separated into smaller, simpler components that are separately designed and produced using the modular design approach. The final product is created by combining each of these parts individually.

The product is a web-based application. It involves obtaining user input in either image format or in textual form. In case of textual format input type, pre-processing of the input is carried out to take care of special symbols and numeric data. The processed Sanskrit text is then used for translation purpose. Encoder-decoder model trained for the process of translation does the job and equivalent translation is generated and does its best to preserve the meaning. In case of image input, text is extracted from the image and the text is used for the process of translation. If there are words apart from the ones the model has been trained on, result is displayed by the model along with a warning.

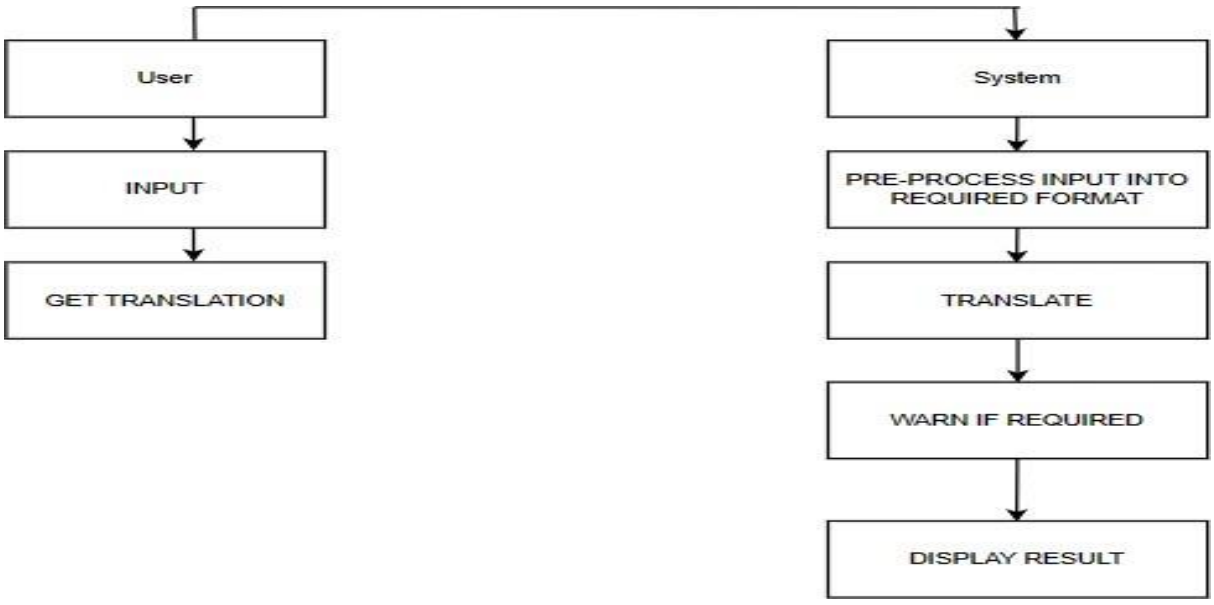


Fig 4.3 Modular Design Diagram for San-Eng Translator

4.4 DATA FLOW DIAGRAM

The classic visual representation of how information moves through a system is a data flow diagram (DFD). It demonstrates how information enters and exits the system, what modifies the data, and where information is kept.

Sanskrit to English translation process proceeds in a sequence of steps. The inputs can be either an image or in a textual format. In case of image input, text extraction is carried out and only the text then is used for translation purpose. The translation process includes preprocessing to remove special symbols and numbers from the input texts and the generated preprocessed text is then fed to an encoder-decoder model trained for the purpose of translation to get relevant translations of the same. In case any new words are encountered in the process apart from the vocabulary that the model has been trained on, translation is given by the model along with a warning message.

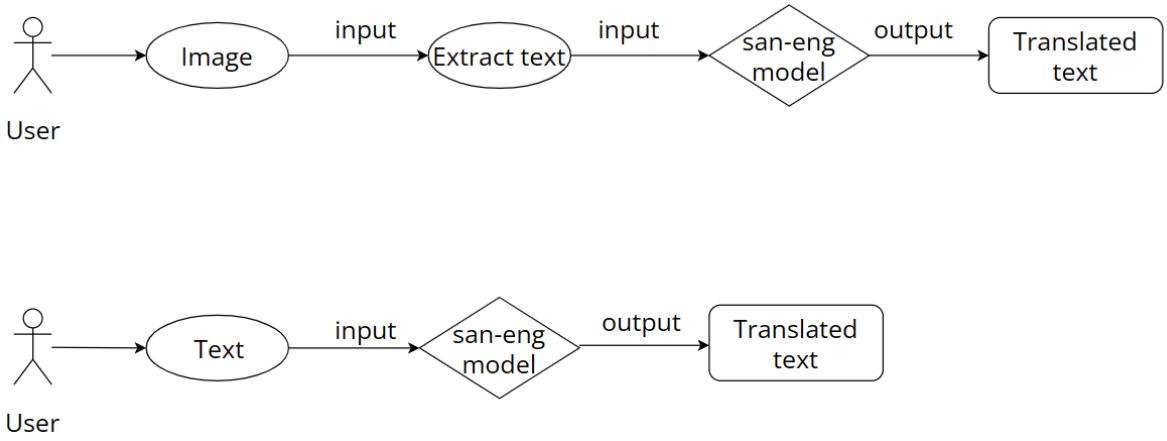


Fig. 4.4 Data Flow Diagram for San-Eng Translator

4.5 SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram that demonstrates the relationship and sequential order of operations. It displays item interactions that are timed. It shows the classes and objects involved in the scenario as well as the flow of messages that must be exchanged for the objects to work as intended.

The sequence flow diagram for the San-Eng translator is shown in Fig 4.5. The sequence is as follows:

- User chooses an input type as image or text.
- If input type is Image, OCR does the text extraction.
- Else if input type is text, OCR is skipped.
- Pre-processor then processes the input/extracted text and removes special symbols, numbers.
- Translator then takes the pre-processed text, translated it to English and returns the result to the user

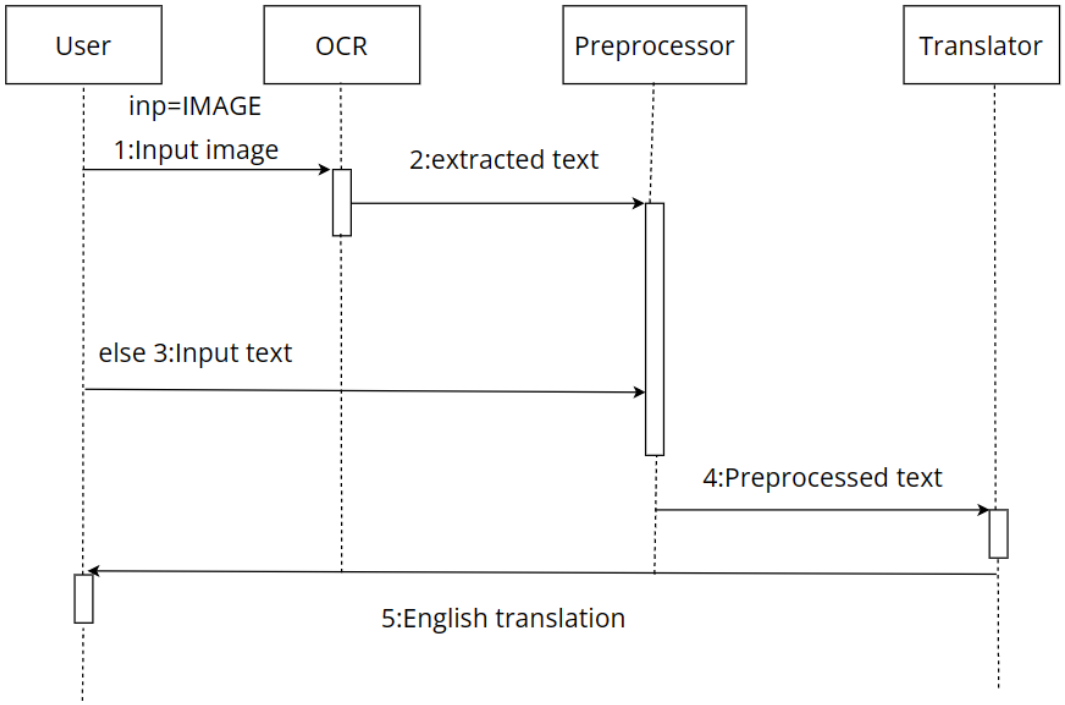


Fig. 4.5 Sequence Diagram for San-Eng Translator

4.6 SYSTEM FLOW DIAGRAM

One of the graphical representations of the data flow in a system used in software engineering is the system flow diagram. The diagram is made up of numerous steps that show where the system's input and output come from and go to. The graphic makes it feasible to manage how the system decides which events to process and how data enters the system. The system flow diagram, which leaves out the minor components and shows the important components of the system in order of importance, is essentially a visual depiction of data flow.

The complete system flow design includes:

- Collect the user query through the user interface either as an image or as a text.
- In case of image text extraction is carried out and the text retrieved is used for the purpose of translation.
- Pre-process the text by removing special symbols and numbers.
- Trained encoder-decoder model then takes this processed text as input, transforms each Sanskrit word into a vector and maps it to a matching English translation.
- Model follows sequence to sequence word mapping approach and gives relevant translation.
- If some words encountered don’t see a matching translation, translation is returned along with a warning message.

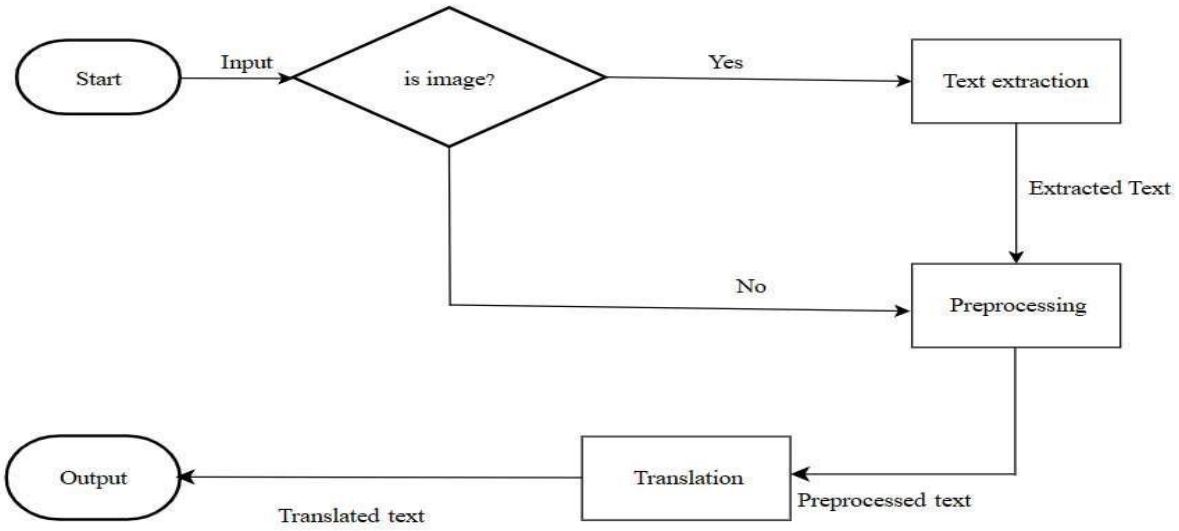


Figure 4.6: System Flow Diagram for San-Eng Translator

4.7 ACTIVITY DIAGRAM FOR USE CASE

Activity diagram depicts the progression of one activity across a system or process. It is referred to as a "behaviour diagram" because it outlines what ought to occur in the modelled system and is used to depict the various dynamic characteristics of a system.

The activity diagram for the use case ‘Translation’ is shown in fig 4.7. Input to this phase is the pre-processed Sanskrit text. This text is then one hot encoded and given to the encoder that returns a context vector. Context vector is decoded by the decoder into one hot encoded vectors again which are mapped to English words by the translator model and the result it then displayed.

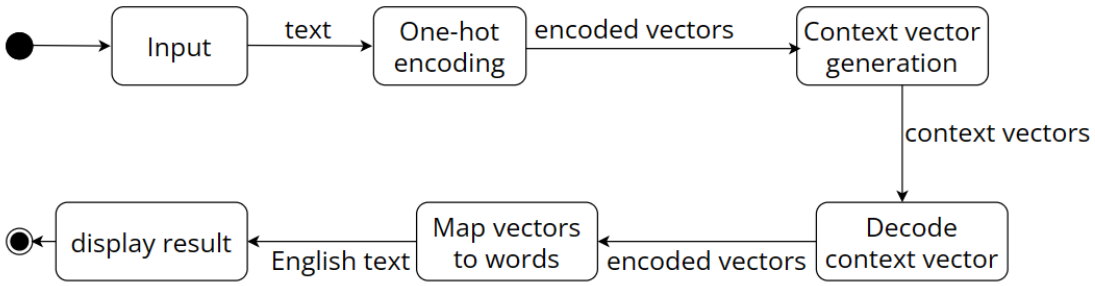


Fig 4.7 Activity Diagram for San-Eng Translator

4.8 USER INTERFACE FROM DESIGN

The goal of user interface design is to foresee what users would need to perform and make sure that the interface has parts that are simple to use, access, and comprehend. Information architecture, interaction design, and graphic design ideas are all combined in UI.

UI for the current system has been built using streamlit library. Streamlit is an open-source app framework in Python language. It helps us create web apps for data science and machine learning in a short time.

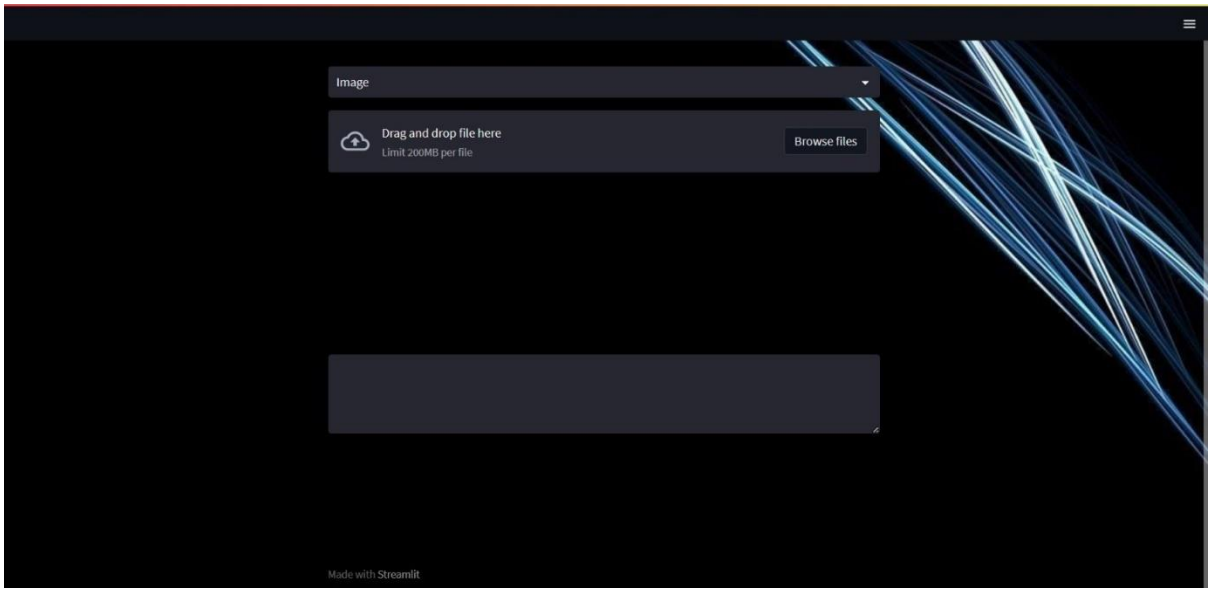


Fig 4.8 User Interface Design

CHAPTER 5

IMPLEMENTATION & TESTING

5.1 DATA COLLECTION

The first step is to collect a large corpus of Sanskrit and English texts. This corpus will serve as the training data for the machine learning algorithm. The data can be collected from various sources such as books, websites, and other documents.

5.2 DATA PREPROCESSING

Once the data is collected, it needs to be pre-processed. This involves cleaning the data, removing any unwanted characters, and formatting it in a way that the machine learning algorithm can process.

5.3 LANGUAGE IDENTIFICATION

The next step is to identify the language of each sentence in the corpus. This is necessary to separate the Sanskrit and English sentences, so that the machine learning algorithm can learn to translate from Sanskrit to English.

5.4 TRAINING MACHINE LERANING MODEL

The next step is to train the machine learning model. This involves using a technique called supervised learning, where the model is trained on a labelled dataset. The labelled dataset consists of pairs of Sanskrit and English sentences, where the English sentence is the correct translation of the Sanskrit sentence.

5.5 TESTING AND EVALUATION

After training the model, it needs to be tested and evaluated. This involves using a separate set of data to test the accuracy of the model's translations.

5.6 DEPLOYMENT

Once the model has been tested and evaluated, it can be deployed for use. This involves integrating the model into an application or website that users can access to translate Sanskrit sentences to English.

CHAPTER 6

RESULTS

6.1 HOME PAGE

A home page is the main web page of a website. The term may also refer to the start page shown in a web browser when the application first opens. Usually, the home page is located at the root of the website's domain or subdomain.

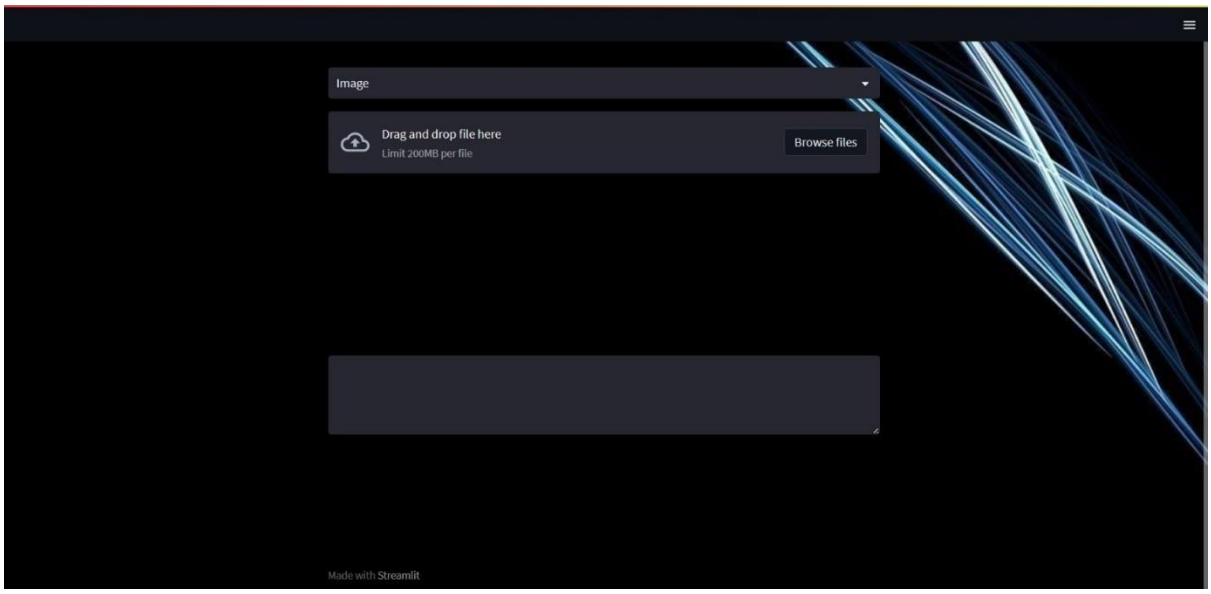


Fig 6.1 Home Page

6.2 TEXT EXTRACTION

To convert the Sanskrit text into English text you have to enter the text in the below box and press enter to translate the text.

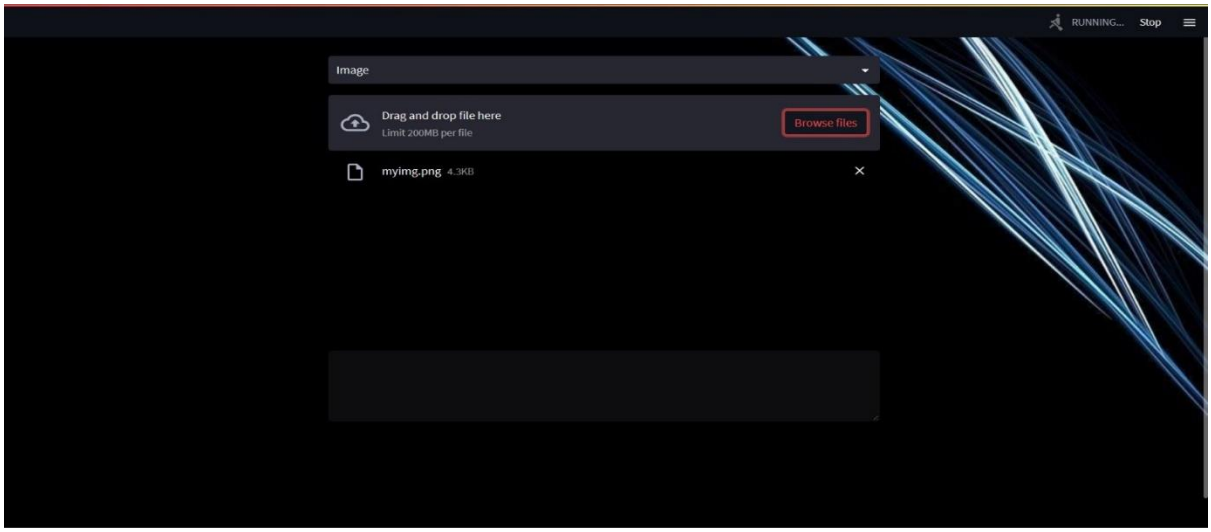


Fig 6.2 Text Extraction

6.3 TRANSLATION

It is used to convert the image of Sanskrit text to English text by dragging and dropping the image below the option and press enter to translate the image text.

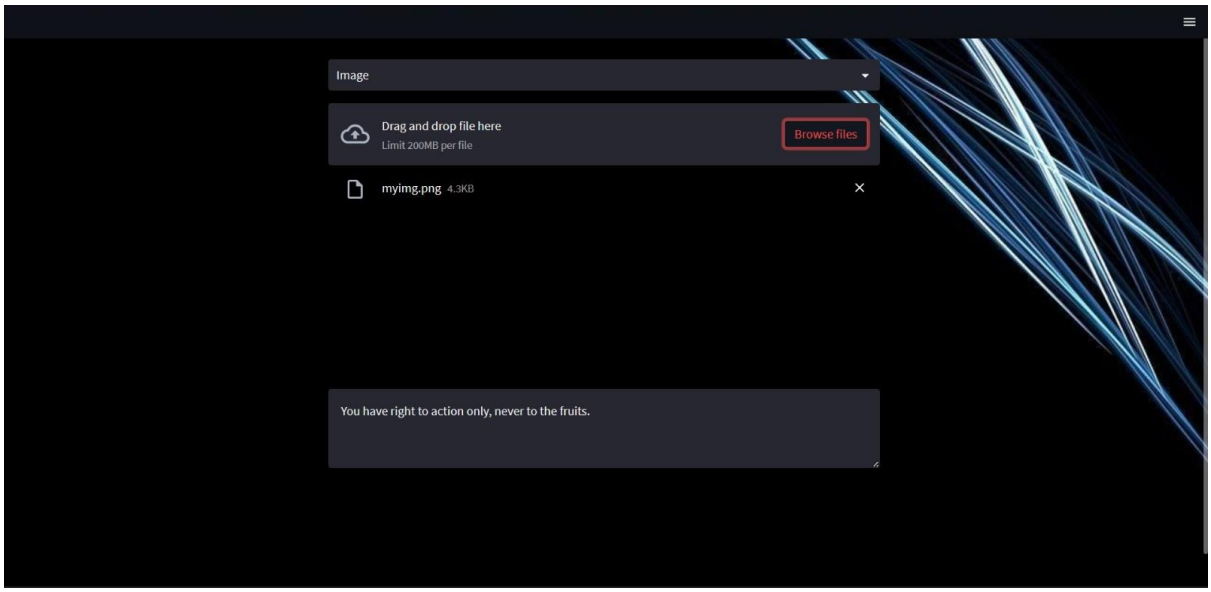


Fig 6.3 Translation

CHAPTER 7

CONCLUSION AND FUTURE WORK

The use of OCR and Encoder-Decoder models for Sanskrit to English translation is a promising approach, given the morphological and structural differences between the two languages. OCR helps in accurate text extraction, while Encoder-Decoder models enable end-to-end translation, contextual understanding, and speed and efficiency.

Future work in this area could include creating more extensive datasets for training, fine-tuning pre-trained models, using hybrid models combining rule-based approaches and deep learning, and developing multi-lingual models for Sanskrit and other languages. With these advancements, the accuracy and efficiency of Sanskrit to English translation using OCR and Encoder-Decoder models can be further improved, making it a valuable tool for language learning, cultural exchange, and cross-language communication.

BIBLIOGRAPHY

[1] S. Kondo, "Machine Translation with Pre-specified Target-side Words Using a Semi-autoregressive Model," no. 2019, pp. 68–73, 2021.

[2] R. K. Chakrawarti, J. Bansal, and P. Bansal, "Machine translation model foreffective translation of Hindi poetries into English," *J. Exp. Theor. Artif. Intell.*, vol.34, no. 1, pp. 95–109, 2022, doi: 10.1080/0952813X.2020.1836033.

[3] A. Pathak and P. Pakray, "Neural machine translation for Indian languages," *J. Intell. Syst.*, vol. 28, no. 3, pp. 465–477, 2019, doi: 10.1515/jisys-2018-0065.

[4] B. Premjith, M. A. Kumar, and K. P. Soman, "Neural machine translation system for English to Indian language translation using MTIL parallel corpus," *J. Intell. Syst.*, vol. 28, no. 3, pp. 387–398, 2019, doi: 10.1515/jisys-2019-2510.

[5] Sitender and S. Bawa, "Sanskrit to universal networking language EnConverter system based on deep learning and context-free grammar," *Multimed. Syst.*, 2020, doi: 10.1007/s00530-020-00692-3.

[6] I. Rivera-Trigueros, "Machine translation systems and quality assessment: a systematic review," *Lang. Resour. Eval.*, vol. 56, no. 2, pp. 593–619, 2022, doi: 10.1007/s10579-021-09537-5.

[7] O. Hellwig, S. Sellmer, and S. Nehrdich, "Obtaining more expressive corpus distributions for standardized ancient languages," *CEUR Workshop Proc.*, vol. 2989, pp. 92–107, 2021.

[8] O. Hellwig, S. Scarlata, E. Ackermann, and P. Widmer, "The treebank of Vedic Sanskrit," *Lr. 2020 - 12th Int. Conf. Lang. Resour. Eval. Conf. Proc.*, pp. 5137– 5146, 2020.

[9] O. Hellwig, "Dating and Stratifying a Historical Corpus with a {B}ayesian Mixture Model," *Proc. LT4HALA 2020 - 1st Work. Lang. Technol. Hist. Anc. Lang.*, no. May, pp. 1–9, 2020, [Online]. Available: https://aclanthology.org/2020.lt4hala-1.1

[10] O. Hellwig and S. Sellmer, "Detecting Diachronic Syntactic Developments in Presence of Bias Terms," *Proc. Second Work. Lang. Technol. Hist. Anc. Lang.*, no. June, pp. 10–19, 2022, [Online]. Available: https://aclanthology.org/2022.lt4hala- 1.

[11] P. Dhar, A. Bisazza, and G. Van Noord, "Optimal Word Segmentation for," no. 2012,

pp. 181–190, 2021.

[12] M. Singh, R. Kumar, and I. Chana, "Machine Translation Systems for Indian Languages: Review of Modelling Techniques, Challenges, Open Issues and Future Research Directions," *Arch. Comput. Methods Eng.*, vol. 28, no. 4, pp. 2165–2193, 2021, doi: 10.1007/s11831-020-09449-7.

[13] O. Hellwig, "Dating Sanskrit texts using linguistic features and neural networks," *Indogermanische Forschungen*, vol. 124, no. 1, pp. 1–46, 2019, doi: 10.1515/if- 2019-0001.

[14] R. Haque, M. Hasanuzzaman, and A. Way, "Investigating terminology translation in statistical and neural machine translation: A case study on English-to-Hindi and Hindi-to-English," *Int. Conf. Recent Adv. Nat. Lang. Process. RANLP*, vol. 2019-Septe, no. 2017, pp. 437–446, 2019, doi: 10.26615/978-954-452-056-4_052.

[15] S. R. Laskar, A. F. U. R. Khilji, D. Kaushik, P. Pakray, and S. Bandyopadhyay, "Multimodal Neural Machine Translation for English to Hindi" *WAT 2021 -8th Work. Asian Transl. Proc. Work.*, pp. 155–160, 2021, doi: 10.18653/v1/2021.wat-1.17.

[16] O. Hellwig and S. Nehrdich, "Sanskrit word segmentation using character-level recurrent and convolutional neural networks," *Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018*, pp. 2754–2763, 2018, doi: 10.18653/v1/d18- 1295.

[17] W. Lu, "Word sense disambiguation based on dependency constraint knowledge," *Cluster Comput.*, vol. 22, pp. 7549–7557, 2019, doi: 10.1007/s10586-018-1899-3.

[18] A. R. Pal, D. Saha, N. S. Dash, and A. Pal, "Word Sense Disambiguation in Bangla Language Using Supervised Methodology with Necessary Modifications," *J. Inst. Eng. Ser. B*, vol. 99, no. 5, pp. 519–526, 2018, doi: 10.1007/s40031-018-0337-5.

[19] W. Lu, "The Vedic corpus as a graph. An updated version of Bloomfield's Vedic Concordance," *Cluster Comput.*, vol. 22, pp. 754–775, 2019, doi: 10.1009/s10556-018-1899-3.

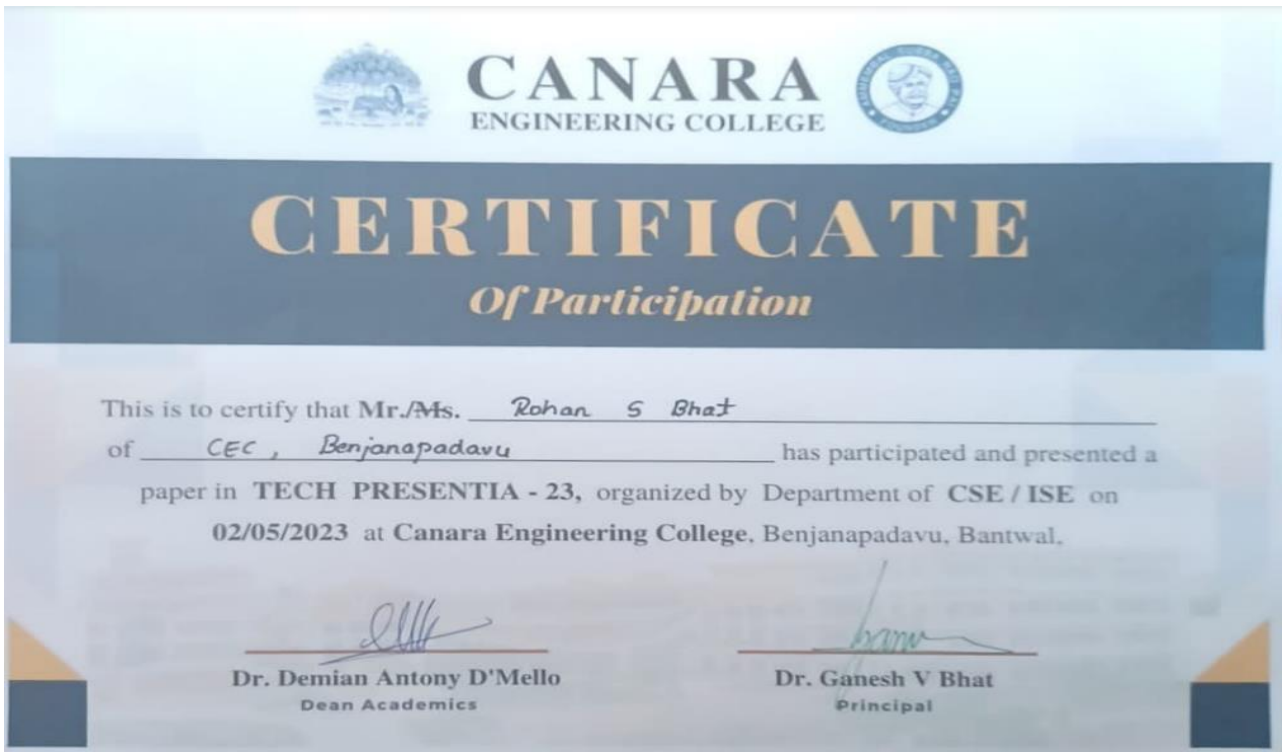
CANARA
ENGINEERING COLLEGE

CERTIFICATE
Of Participation

This is to certify that Mr./Ms. Shetty Ramakrishna Mohan
of CEC, Benjanapadavu has participated and presented a paper in TECH PRESENTIA - 23, organized by Department of CSE / ISE on 02/05/2023 at Canara Engineering College, Benjanapadavu, Bantwal.

Dr. Demian Antony D'Mello
Dean Academics

Dr. Ganesh V Bhat
Principal



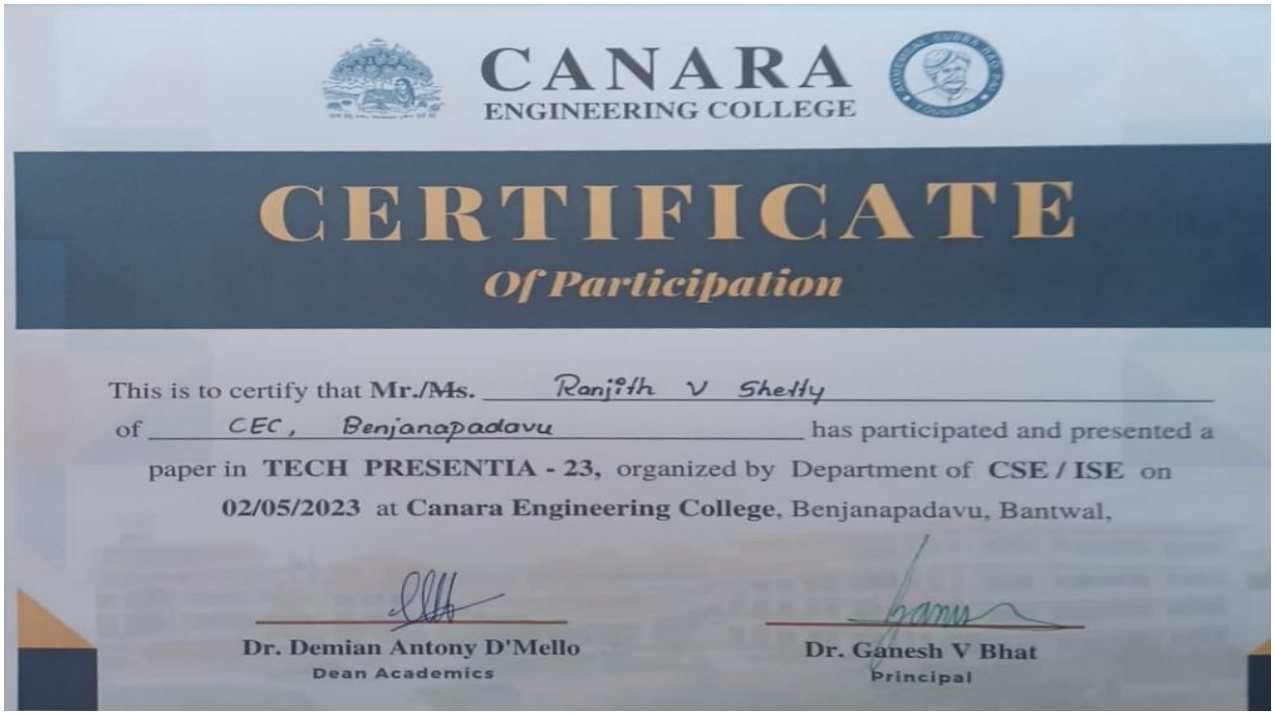
CANARA
ENGINEERING COLLEGE

CERTIFICATE
Of Participation

This is to certify that Mr./Ms. Rohan S Bhat
of CEC, Benjanapadavu has participated and presented a paper in TECH PRESENTIA - 23, organized by Department of CSE / ISE on 02/05/2023 at Canara Engineering College, Benjanapadavu, Bantwal.

Dr. Demian Antony D'Mello
Dean Academics

Dr. Ganesh V Bhat
Principal



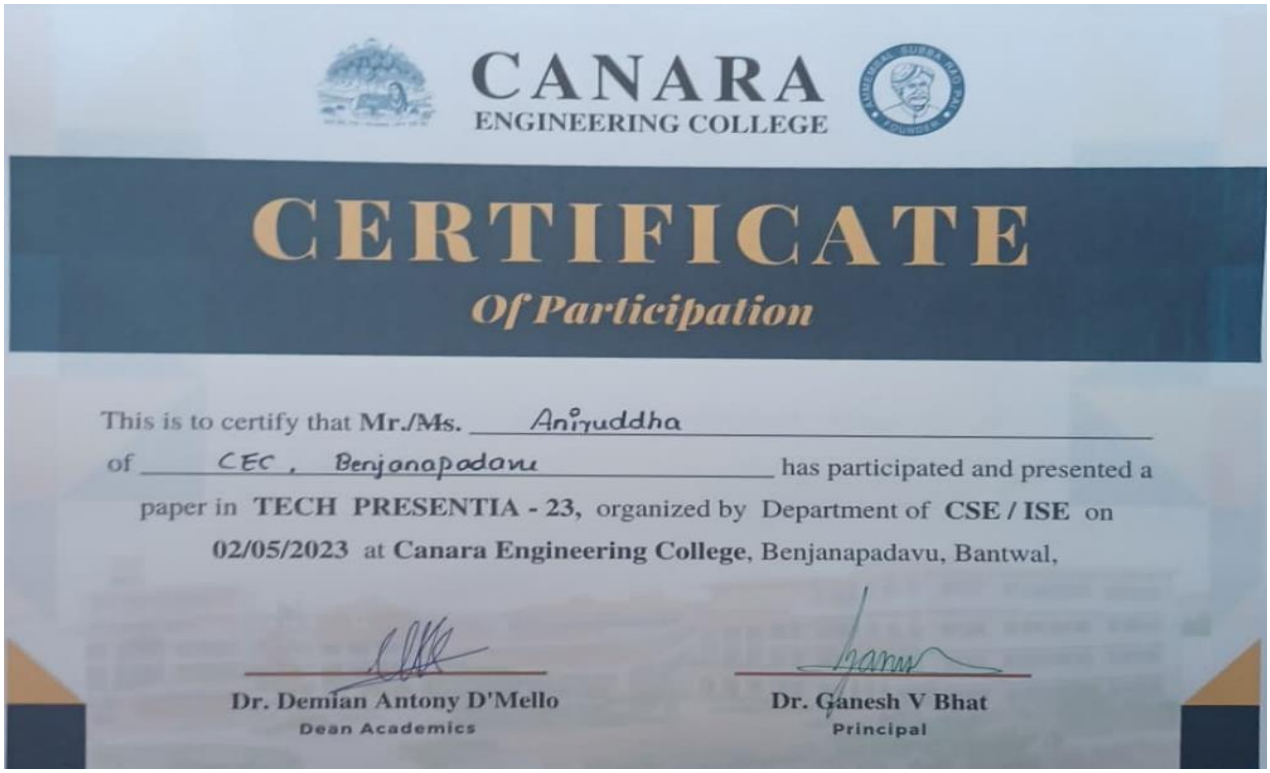
CANARA
ENGINEERING COLLEGE

CERTIFICATE
Of Participation

This is to certify that Mr./~~Ms.~~ Ranjith V Shetty
of CEC, Benjanapadavu has participated and presented a paper in **TECH PRESENTIA - 23,** organized by Department of **CSE / ISE** on **02/05/2023** at **Canara Engineering College**, Benjanapadavu, Bantwal,

Dr. Demian Antony D'Mello
Dean Academics

Dr. Ganesh V Bhat
Principal



CANARA
ENGINEERING COLLEGE

CERTIFICATE
Of Participation

This is to certify that Mr./~~Ms.~~ Aniruddha
of CEC, Benjanapadavu has participated and presented a paper in **TECH PRESENTIA - 23,** organized by Department of **CSE / ISE** on **02/05/2023** at **Canara Engineering College**, Benjanapadavu, Bantwal,

Dr. Demian Antony D'Mello
Dean Academics

Dr. Ganesh V Bhat
Principal