IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 January 2024, accepted 12 January 2024, date of publication 18 January 2024, date of current version 26 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3355470

RESEARCH ARTICLE

Tulu Language Text Recognition and Translation

PRATHWINI[1], ANISHA P. RODRIGUES[2], P. VIJAYA[3], AND ROSHAN FERNANDES[2]

[1]Department of Master of Computer Applications, NMAM Institute of Technology, NITTE (Deemed to be University), Nitte 574110, India
[2]Department of Computer Science and Engineering, NMAM Institute of Technology, NITTE (Deemed to be University), Nitte 574110, India
[3]Department of Mathematics and Computer Science, Modern College of Business and Sciences, Muscat 133, Oman

Corresponding author: Roshan Fernandes (roshan_nmamit@nitte.edu.in)

This work was supported by NITTE (Deemed to be University).

ABSTRACT Language is a primary means of communication, but it is not the only means; knowing a language does, however, assist speed up the process. Many distinct languages are spoken worldwide, and people use them to communicate. This is only one of the many reasons why language is so crucial. Based on the literature survey, it is evident that there is a lack of available translators for the Tulu language. Despite being prevalent predominantly in Karnataka, the Tulu language has not been as widely spoken as other Indian languages until recently, although it gained enough recognition to become the second language in Karnataka. The purpose of our research work aims at translating the English language into the Tulu language. During the evaluation the system was tested on a dataset consisting of handwritten characters during the evaluation process Convolutional Neural Networks used achieved an accuracy rate of 92%. To translate English to the Tulu language, we employed a parallel sentence dataset for the neural approach and a parallel word dataset for the rule-based approach. The rule-based approach resulted in an 89% accuracy rate for word-based analysis and an 81% accuracy rate for sentence-based analysis for the English-to-Tulu language translation. The neural machine translation approach of the Encoder-Decoder model with LSTM is been used to accomplish translation from English to Tulu with a BLEU score of 0.83 and Tulu to English with a BLUE score of 0.65. The model also employed hybrid machine translation to enhance the translation.

INDEX TERMS Machine translation, rule-based method, neural network translation, convolutional neural network (CNN), encoder-decoder model, long short-term memory (LSTM).

I. INTRODUCTION

Language, as a means of communication, encompasses a collection of written symbols and sounds used by individuals within a specific region or country for oral or written expression. It sets humans apart and involves acquiring a complex framework of vocabulary, structure, and syntax for effective communication. Indian languages are categorized into families such as Indo-Iranian or Indo-European, Munda, Dravidian, Austroasiatic, Sino-Tibetan, and Tibeto-Burman. The Indian constitution mentions twenty-two of these languages. India's multilingual nature is evident across its states, although fluency in every language spoken within the nation isn't universal. Tulu, from the Dravidian family, is spoken predominantly in the southern region of Dakshina Kannada and Udupi districts in Karnataka, and in Kasaragod district in southwestern India. Tuluva, the indigenous people, reside in Tulu Nadu. Presently, Tulu is formally acknowledged as the second language of Karnataka, with ongoing efforts to include it in the 8th Schedule of the Constitution. It contains four dialects, mainly used for inter-community communication, trade, and entertainment. Tulu is spoken across various regions like Mangalore, Udupi, Karkala, Belthangady, Kundapura, Kasaragod, Manjeshwar, Puttur Sullia, and Bantwal, with different dialects in each area. Efforts are underway to include Tulu in the Constitution's 8th Schedule. Machine translation (MT), a crucial aspect of natural language processing, has evolved significantly. Rule-based machine translation (RBMT) is an early approach to language translation using predefined linguistic rules. These systems analyze input sentences, break them down into grammatical parts, and then apply linguistic rules to

The associate editor coordinating the review of this manuscript and approving it for publication was Maria Chiara Caschera.

generate the translated output in a target language. RBMT operates in three phases: analyzing the input sentence, transferring the linguistic elements into the target language, and finally, generating the translated sentence. However, RBMT's reliance on manually constructed rules limited its adaptability to handle the complexities of natural languages. This led to the development of more advanced translation methods like neural machine translation (NMT). End-to-end neural machine translation (NMT) has become dominant, departing from earlier rule-based systems that heavily relied on manually crafted translation rules and linguistic knowledge. NMT utilizes computer systems to translate languages and has shown remarkable success compared to earlier rule-based methods, which required extensive linguistic rules for translation between languages. NMT has streamlined translation processes and improved results. Following are some of the significant contributions of the proposed work:

- The data set, consisting of 30,500 manually collected Tulu handwritten characters, was compiled for the proposed work.
- To recognize the Tulu language script, both the CNN algorithm and other machine learning algorithms were utilized.
- A dataset comprising 1458 words and 1000 English-to-Tulu sentences was manually collected. Rule-based and neural machine techniques were employed to translate English to Tulu.
- The English-to-Tulu language translation was accomplished through the application of both rule-based and neural machine techniques.
- To achieve backward translation from Tulu to English, the Encoder-Decoder model with LSTM has been utilized, employing the neural machine translation approach.
- Hybrid machine translation was employed by combining rule-based and neural machine translation to improve the performance of the model.

II. RELATED WORKS

In this section, we elaborate on our efforts in handwritten character recognition and English-to-Tulu language translation, employing various methods. To address the identification of human emotions, we conduct a literature survey to understand and pinpoint the limitations of current techniques. [1] Manimozhi's software focuses on precise recognition of handwritten Tulu characters, aiding in transforming historical manuscripts. Bhat and Seshikala's [2] study evaluates CNN models for Devanagari characters, emphasizing computational efficiency in their specialized architecture. Bhat et al.'s [3] research rejuvenates the ancient Tulu alphabet using CNNs for character recognition, aiming to transform historical manuscripts. Savitha and Antony [4] method utilizes image preprocessing techniques to enhance character recognition accuracy. Rao et al. [5] explore machine-learning methods for Tulu characters, emphasizing CNN's efficiency in recognition. Memon et al. [6] present a user-friendly approach for Kannada character recognition employing deep learning algorithms. Albahli et al.'s [7] summary explores handwritten document recognition and potential future research directions based on an SLR. Bora et al. [8] propose a multi-step approach using enhanced Faster-RCNN for numeral recognition from images. Deore and Pravin [9] combine CNN and ECOC classifiers for accurate OCR of handwritten characters. Khandokar et al. [10] introduce a dataset and a VGG16 model for Devanagari character recognition. Guha et al. [11] explore CNN's capacity in recognizing complex handwritten characters with a dataset achieving 92.91% accuracy. Rani et al. [12] emphasize CNN's role in automatic feature extraction for character recognition, specifically in Devanagari script. Hamdan [13] design a capsule network for effective Kannada character recognition. Vinjit et al. [14] compare various approaches for handwriting recognition, including statistical methods and neural networks. Rani et al. [15] highlight the need for improved accuracy and efficiency in Handwritten Character Recognition. Athira et al. [16] use transfer learning from Devanagari for recognizing handwritten Kannada characters. Yadav et al. [17] tackle the challenge of recognizing confusing characters in Kannada documents using templates and classifiers. Ganji et al. [18] comprehensively cover phases of offline handwritten Hindi character recognition. Ayyob and lyas [19] discuss OCR challenges in recognizing Telugu literature characters due to limited datasets and trained CNNs. Srelekha [20] survey diverse methods used in Malayalam handwriting recognition, highlighting feature extraction and classification techniques. Islam et al. [21] propose an effective Bengali-to-English translation technique, improving machine translation accuracy. Bhadwal et al. [22] propose a system for translating Hindi text into Sanskrit, considering linguistic features of both languages. Sitender and Bawa [23] utilize bilingual dictionaries and rule-based approaches for translation. Kharate et al. [24] discuss challenges in building translation models due to language differences in syntax and morphology. Kodabagi and Angadi [25] focus on neural machine translation (NMT) for Kannada to English, achieving higher accuracy than statistical methods. Salunkhe et al. [26] highlight challenges in translating complex sentences and suggest simplification techniques. [27] Arikpo and Dickson propose a method for localizing and classifying digits into ten classes using Faster-RCNN. Mardhotillah et al. [28] propose a methodology achieving a high accuracy of 97.56% in translating sentences into Kannada. Dhar et al. [29] propose a system using parallel datasets for document translation. Soman et al. [30] develop a transfer-based translator for English to Efik language translation. Prajapati et al.'s [31] paper evaluates CNN models for recognizing Devanagari characters, highlighting the absence of a universal model. Choudhary et al. [32] compare subword segmentation methods for translating English to Dravidian languages. Gogineni et al. [33] assess an NMT architecture's effectiveness for translation, emphasizing the impact of lengthy English sentences. Hegde et al. [34] explore

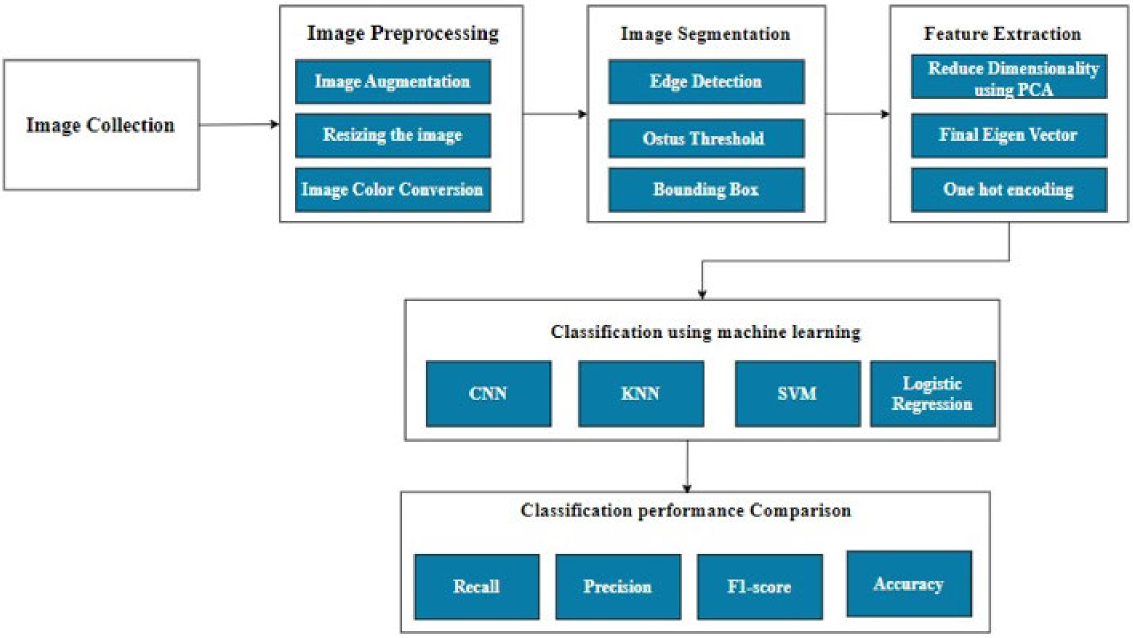


FIGURE 1. Block diagram of Tulu character recognition.

pragmatic methods for examining machine translation between Tulu and Kannada. Fernandes and Rodrigues [35], [36] propose models recognizes handwritten Kannada characters and predict emotions in videos. Dou et al. [37] explored various machine learning algorithms and achieved favorable accuracy results. From the literature survey we have observed that Hybrid algorithms are not considered for the evaluation of the handwritten Tulu character recognition. Expansion of the dataset has to be done to increase the accuracy of the Tulu character recognition model. More Preprocessing techniques can be applied to increase the accuracy of the Tulu character recognition model. From the literature survey we have also found that there are no existing translators for English to Tulu or Tulu to English since Tulu is a local language which is spoken in udupi and Mangaluru region. Our research objective is to create a system that can translate the English language into Tulu, as there are no existing translators available according to the literature survey. Our research endeavors to create a system with the goal to fill the gap in Tulu language translation tools by translating English to Tulu and Tulu to English. To address the lack of Tulu language translators by developing a system that can precisely translate English to Tulu and Tulu to English, as no existing translators were identified during the literature survey.

III. METHODOLOGY

A. TULU CHARACTER RECOGNITION

The proposed method, as represented in Figure 1, follows various steps including Image Pre-processing, Image Segmentation, Feature Extraction, Classification using machine learning algorithms, and Performance Comparison.

1) DATA PREPARATION AND COLLECTION

Figure 1 describes the methodology used for Tulu character recognition such as image collection which involves collecting various handwritten characters of about 62 classes which includes 50 characters and 12 numerical. Total 31000 images which include 500 images of each class are collected.

2) IMAGE PREPROCESSING

Augmentation techniques are being applied with rotation_range = 40, zoom_range = 0.1, and brightness range between 0.5, and 0.8 which are shown in figure 2. Below are the different augmented images. Once the character is detected each character is cropped individually and converted into 28*28 pixels. Each image is converted into a Grayscale and then a binary image to reduce the noise in the image which is shown in figure 3.

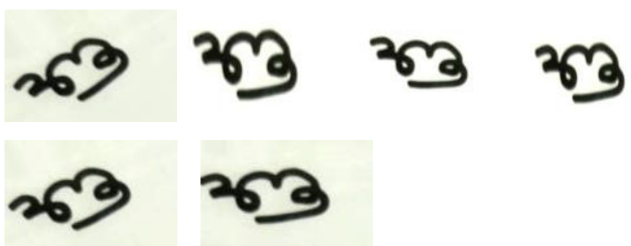


FIGURE 2. Different augmented images.

3) IMAGE SEGMENTATION

After the image preprocessing stage, the contour technique is utilized to identify distinct shapes or patterns within

(a) Ordinary Image (b) Grey Image (c) Binary Image

FIGURE 3. Image color conversion.

Algorithm 1 Image Segmentation

Set image width to 1000
Set image height to 550
Set dim to (width, height)
Resize the image to dim using cv2.INTER_AREA
Convert the resized image to grayscale
Set thresh to 90
Create a binary image using thresholding with the OTSU method on binaryImage
Set kernel to a size (2,1)
Apply morphological operations for refinement
Set dilate_kernel to a rectangular structuring element of size (8,5)
Apply morphological dilation to opening using dilate_kernel
Find contours on dilate with RETR_EXTERNAL and CHAIN_APPROX_SIMPLE modes
if len(cnts) == 2 **then**
  Set cnts to cnts[0]
end if
Set i to 4
for each contour c in cnts **do**
  Set rect to the bounding rectangle of c
  if rect [2] < 20 OR rect [3] < 20 **then**
    continue
  end if
  Set x, y, w, h to rect
  Write the subimage of binaryImage that corresponds to rect to a file named str(i) + ".jpg"
  Increment i by 1
end for

the processed image. This involves detecting these shapes, outlining their boundaries using bounding boxes, and potentially recognizing characters or text segments. Subsequently, these segmented areas are subjected to further analysis like character identification.

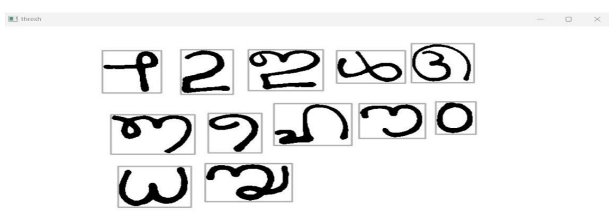


FIGURE 4. Tulu numericals detection.

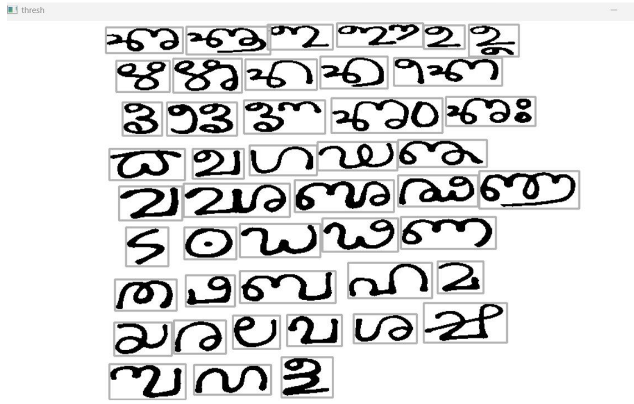


FIGURE 5. Tulu character detection.

Once the character is detected each character is cropped individually and converted into 28*28 pixels. Figure 4 and 5 shows the detection of Tulu numerical and characters.

4) FEATURE EXTRACTION

Principle Component Analysis is applied on each individual cropped image. In this process 784 feature of individual image is fed into PCA since images are in 28*28 pixel size. PCA considers 0.98 information from the each image hence 291 features are considered from the image. Once feature extraction is completed through the use of eigen vectors, the labels are encoded using the one-hot encoding technique. This is done to facilitate their input into machine learning algorithms for subsequent processing.

Algorithm 2 PCA for Dimensionality Reduction

1) Set PCA to keep 98% of the variance:
2) If variance threshold is not provided then
3) Set variance threshold to 0.98
4) Initialize pca with variance threshold
5) Fit the PCA model with the training data and transform
6) xtrain = Apply fit_transform method of pca with x_train as input
7) xtest = Apply transform method of pca with x_test as input
8) Print the shape of xtrain and xtest
9) Print the number of principal components selected by pca
10) Print the number of features in the original dataset

5) CLASSIFICATION USING MACHINE LEARNING ALGORITHMS

In order to train our method on a dataset, we utilized preparation with Convolutional Neural Network to train on bureaucracy data. Multi-class categorization with a total of 62 classes, comprising 50 characters and 12 numerical.

In the analysis, the convolutional neural network displays greater accuracy when contrasted with the other evaluated

TABLE 1. CNN layer hyperparameters.

Layer	Hyperparameters
Conv2D	3 ×3 Kernel size, ReLU activation, 64 filters
Conv2D	3 ×3 Kernel size, ReLU activation, 32 filters
Dropout (Core Layer)	0.3 Neurons
MaxPool2D	2 ×2 pool size
Dense	64 Units, ReLU activation
Dense	62 Units, Softmax activation
Dropout (Core Layer)	0.3 Neurons
Conv2D	3 ×3 Kernel size, ReLU activation, 128 filters
MaxPool2D	2 ×2 pool size

machine learning algorithms. Table 1 and 2 gives the detailed analysis of the CNN hyperparameters.

TABLE 2. Hyperparameter values.

Hyperparameter	Value
Regularization term	kernel regularizer with L2 regularization
Batch size	32
Loss function	Categorical cross-entropy
Learning rate	0.001
Epochs	150
Optimizer	Adam

B. RULE BASED LANGUAGE TRANSLATION FROM ENGLISH TO TULU

The model comprises four modules in sequence. The initial module processes English speech input, passing its output to the subsequent module, aiming to generate Tulu text as the final result.Figure 6 illustrates the system architecture of the language translator.

1) PREPARATION AND COLLECTION OF DATASET

Collection of the dataset was done manually which includes words in english, kannada and tulu language. Though we only make use of english and tulu words in the main translation, we collected the corresponding kannada words which can be used in near future, for further development. The total number of words collected initially was 1481.Training words for rule based machine translation are shown in figure 7.

2) ALGORITHM

English text is taken as input. Spacy was utilized to perform Parts of Speech tagging and Named Entity Recognition on the English text, with the goal of identifying any names or locations present. If such entities were detected, they were transcribed and stored separately. The resulting modified English text was then translated into Kannada. For each word in the Kannada text, a word-to-word translation was performed from Kannada to English, with each English word being checked against a database. If a match was found, the corresponding Tulu word was mapped to the English word. If no match was found, no translation was performed. Finally, the Tulu sentence was constructed based on the Kannada sentence structure, with any previously identified named entities being reintroduced. All of these steps were combined to generate the final Tulu text. Algorithm 3 shows the step by step procedure for the rule based translation.

Algorithm 3 Translate Text Algorithm

```
Initialize ner_dict as an empty dictionary and pos_list as an empty list.
Tokenize and tag the input text using spaCy's nlp() function.
for each token do
    if token has a named entity type then
        Add token to ner_dict.
    else
        Append (token text, POS tag) to pos_list.
    end if
end for
Translate the input text from English to Kannada.
Split the translated text into a list of words (l).
for each word in l do
    Translate word from Kannada to English using Translator.
end for
Filter out stop words and exceptions from the translated words.
Join the filtered words into a single string (string1) and split it into a list (string).
Open and read a CSV file.
for each row in the CSV file do
    for each word in string do
        if word is a key in ner_dict then
            Replace the word in string with ner_dict.
        else
            Construct a regex pattern "∧" and match it against the word.
            if there is a match then
                Replace the word in string with the Tulu word.
            end if
        end if
    end for
end for
```

C. NEURAL MACHINE TRANSLATION

Neural Machine Translation (NMT) is an automated translation method that utilizes neural networks to translate text from one language to another, replacing Statistical Machine Translation (SMT) in many applications. The fundamental concept behind NMT is to use a deep neural network to learn the correlation between the source and target language. The neural network takes the source language text as a input and generates the translated text in the target language as output. This network is trained on a large parallel corpus of sentences

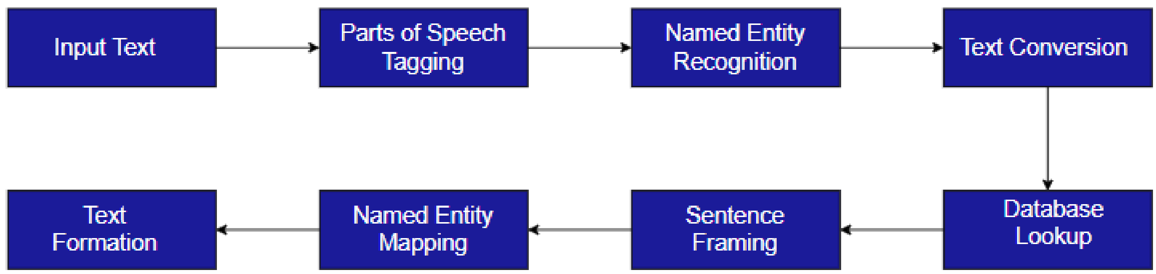


FIGURE 6. Block diagram of rule based language translation.

English	Kannada	Tulu
ancient	ಪ್ರಾಚೀನ	ಪಿರಾವುದ
anger	ಸಿಟ್ಟು	ಕೋಪ
animal	ಪ್ರಾಣಿ	ಪ್ರಾಣಿ
fruit	ಹಣ್ಣು	ಪರ್ನ್ ದ್
annoy	ಕಿರಿಕಿರಿ ಮಾಡು	ಕಿರಿಕಿರಿ ಮನ್ಪು
another	ಇನ್ನೊಂದು	ಕುಡೊಂಜಿ
answer	ಉತ್ತರ	ಜವಾಬು
ant	ಇರುವೆ	ಪಿಜಿನ್
dear	ಜಿಂಕೆ	ಜಿಂಕೆ
any	ಯಾವುದೇ	ಒವುಲಾ
anybody	ಯಾರಾದರೂ	ಎರಾಂಡಲ
anyhow	ಹೇಗಾದರೂ	ಎಂಚಾಂಡಲ
anything	ಯಾವುದಾದರೂ	ಒವುವಾಂಡಲ
anyway	ಏನೇ ಆದರೂ	ದಾದಆಂಡಲಾ
anywhere	ಎಲ್ಲಾದರೂ	ಒಲಾಂಡಲ
apologise	ಕ್ಷಮೆ	ಮಾಪು
ask	ಕೇಳು	ಕೇನ್
appointment	ಭೇಟಿಗಾಗಿ	ತಿಕ್ಕುನು
time	ಸಮಯ	ಪೊರ್ತು
appreciate	ಮೆಚ್ಚು	ಪುಗರ್
approach	ಹತ್ತಿರ	ಕೈತಲ್

FIGURE 7. Training words for rule based machine translation.

in both languages. The process of neural machine translation involves the computer automatically translating sentences from one language to another.The steps involved in neural machine translation are illustrated in Figure 7. The input layer converts the source sentence into numerical vectors. LSTM1 layer encodes the source sentence and generates hidden states capturing its contextual information. LSTM2 layer decodes the encoded information into the target language, utilizing the hidden states from LSTM1 layer. The output layer produces the final translation by computing probabilities for each target word and selecting the word with the highest probability. Figure 8 shows the structure of Neural Machine Translation.

In our methodology for English to Tulu neural machine translation, we incorporated both forward and backward approaches. The forward approach involves training the model to translate from English to Tulu, where the English sentence is the input and the Tulu translation is the target. However, we also recognized the importance of the backward approach, which involves training the model in the reverse direction, from Tulu to English. The backward approach provides additional benefits by enabling the model to capture bidirectional language dependencies and improve the overall translation quality. By training the model to translate from Tulu to English, we ensure that it learns to generate accurate and fluent Tulu translations, leveraging the knowledge obtained from both forward and backward training.

1) DATA COLLECTION AND PREPARATION

To train a neural machine translation model, we have used a 1000 parallel sentence of both english and tulu. Clean the parallel corpus to remove noise and inconsistencies. This involves removing duplicate sentences, correcting typographical errors, and normalizing punctuation, capitalization, and formatting. Training sentences of neural machine translation are shown in Figure 9.

2) TOKENIZATION

The dataset's sentences are tokenized, meaning that each sentence is converted into a sequence of integers where each integer represents a word. Separate tokenizers are utilized for the source and target languages.

3) SEQUENCE PADDING

After tokenization, the sequences are padded to ensure that they have the same length. This is essential since neural networks demand input data to have the same shape. The block diagram of neural machine translation is displayed in Figure 7.

4) ENCODER MODEL

An encoder model is built that takes the source language embeddings as input and generates a fixed-size vector that represents the entire input sequence. This model typically uses an LSTM (Long Short-Term Memory) network, which allows it to handle variable-length input sequences and capture long-term dependencies in the input. The LSTM network processes the input embeddings word by word and produces a sequence of hidden states. The last hidden state of the LSTM network is then used as the fixed-size vector representation of the input sequence.

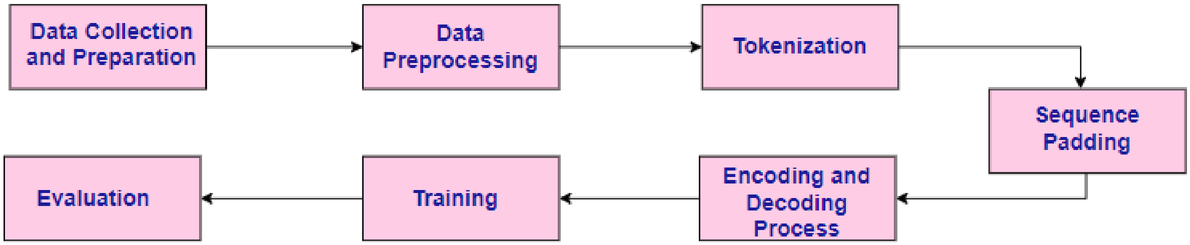


FIGURE 8. Block diagram of neural machine translation.

English	Tulu
How are you?	ಎಂಚ ಉಲ್ಲರ್?
Is everyone good at home?	ಇಲ್ಲಡ್ ಮಾತೆರ್ಲ ಉಷಾರ್ ಉಲ್ಲೆರ?
What are you having today?	ಇನಿ ದಾದಾ ತಿನ್ಪಿನಿ?
Where did you go?	ಒಡೆ ಪೋಯಿನಿ?
Puru went to Bangalore.	ಪುರು ಬೆಂಗಳೂರುಗ್ ಪೋಯೆ.
Does it take a while?	ಮಸ್ತ್ ಪೊರ್ತು ದೆತೊನುನ?
Give me a masala dosa.	ಎಂಕ್ ಮಸಾಲ ದೋಸೆ ಕೊರ್ಲೆ.
I know little tulu.	ಎಂಕ್ ಚುರು ತುಳು ಬರ್ಪುಂಡು.
Pratheeksha likes it.	ಪ್ರತೀಕ್ಷಗ್ ಇಷ್ಟ ಆಪುಂಡು.
We will come back.	ಎಂಕುಲು ಪಿರ ಬರ್ಪೆ.
Is there a Telephone nearby?	ಕೈತಲ್ಡ್ ಟೆಲಿಫೋನ್ ಉಂಡೆ?
May I use your telephone?	ಇರ್ನ ಮೊಬೈಲ್ ಉಪಯೋಗ ಮನ್ಪೊಲಿಯ?
My name is prathwini.	ಎನ್ನ ಪುದರ್ ಪ್ರಥ್ವಿನಿ.
Please give the bill.	ದಯಾಮಲ್ದ್ ರಶೀದಿ ಕೊರ್ಲೆ.
Please wait for a moment.	ದಯಾಮಲ್ದ್ ಚುರು ಕಾಪುಲೆ.
Thank you very much!	ಮಸ್ತ್ ಸೊಲ್ಮೆಲು!
Ganesh went to london.	ಗಣೇಶ ಲಂಡನ್ಗ್ ಪೋಯೆ.
The price is too high.	ಕಾಸ್ ಮಸ್ತ್ ಜಾಸ್ತಿ.

FIGURE 9. Training sentences of neural machine translation.

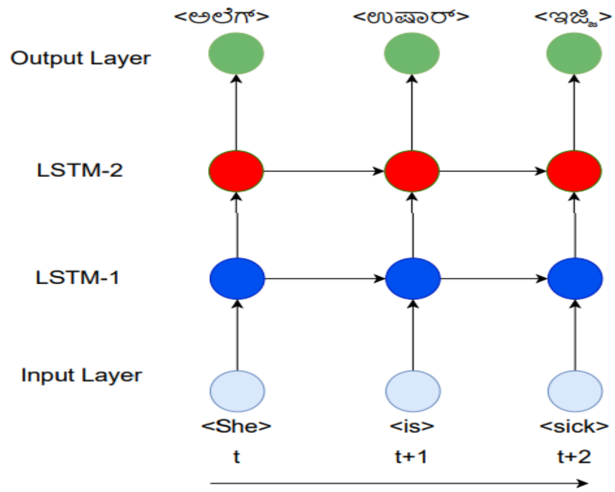


FIGURE 10. Structure of neural machine translation.

5) DECODER MODEL

The decoder model is designed to receive the embeddings of the target language along with the output of the encoder model as inputs, and generate the output sequence word by word. Typically, an LSTM network is used in the decoder model, which enables it to handle output sequences of variable length and generate words based on previous words generated. As the target language embeddings and the encoder model's output are fed into the LSTM network, a sequence of hidden states is produced. In each time step, the LSTM network of the decoder generates a probability distribution over the target vocabulary using the current hidden state and selects the subsequent word in the output sequence based on this distribution. Encoder decoder model is shown in Figure 8.

6) TRAINING

The training process combines training for both the encoder and decoder models, using a loss function that computes the variance between the predicted and real output sequences. The LSTM network weights and other model parameters are optimized using backpropagation through time. After training, the models can be used to translate new input sequences from the source language to the target language by encoding the input sequence using the encoder model and generating the output sequence word by word using the decoder model. Table 3 describes the Hyperparameter of Encoder-Decoder Model.

TABLE 3. Hyperparameter of encoder-decoder model.

Encoder	
Embedding dimension	256
LSTM units	256
Decoder	
Embedding dimension	256
LSTM units	256
Return sequences	True (to return sequences at each timestep)
Return state	True (to return the final internal state of the LSTM)
Training	
Optimizer	Adam optimizer
Loss function	Sparse categorical cross-entropy
Number of epochs	500
Batch size	128

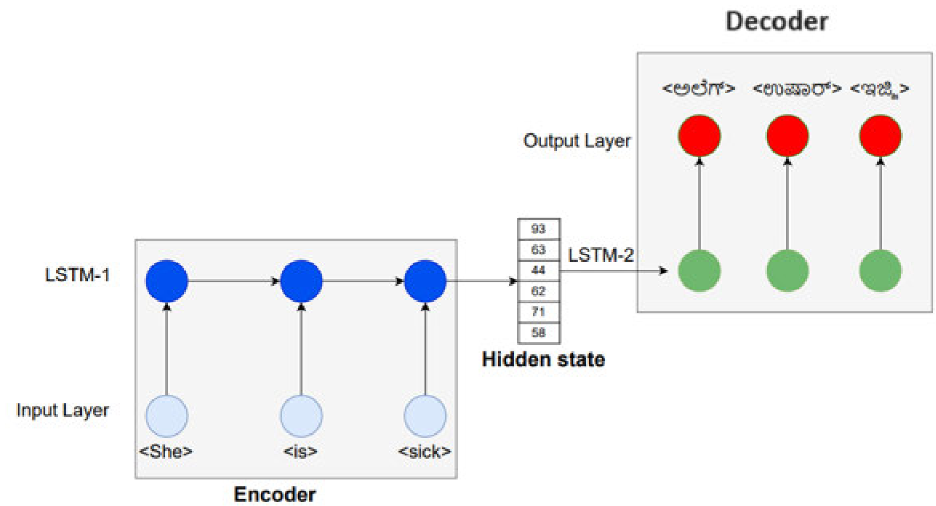


FIGURE 11. Encoder decoder model structure.

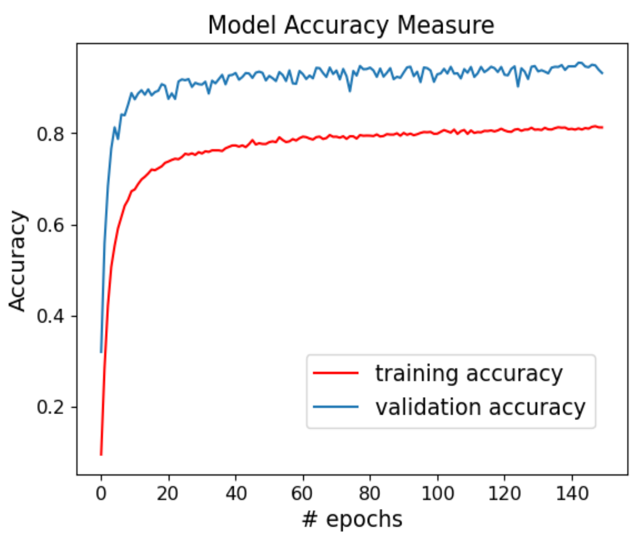


FIGURE 12. CNN model accuracy measure.

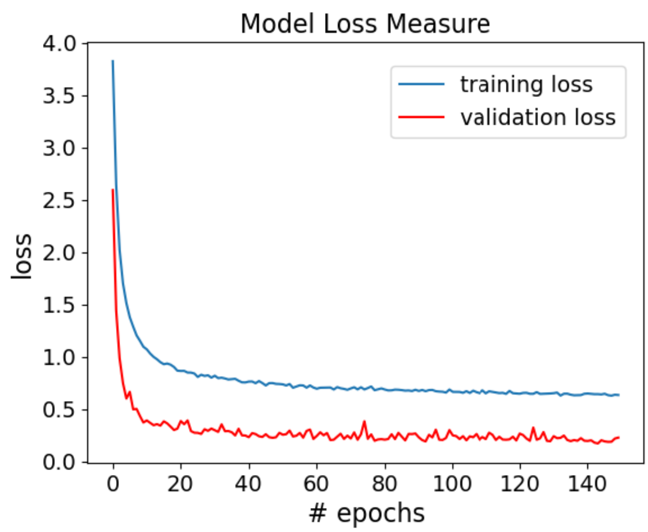


FIGURE 13. CNN model loss measure.

7) EVALUATION

To evaluate the model's performance on the validation set, metrics such as precision, accuracy, F1- score and recall are calculated.

IV. RESULTS AND DISCUSSION

This section presents the experiment results on recognizing handwritten Tulu characters. In the performance analysis, the CNN model achieved an accuracy of 92% with the considered dataset. This accuracy outperforms the other algorithms, as Figure 10(c) depicts. Figures 10(a) and 10(b) illustrate the accuracy and loss measures of the CNN model over 150 epochs.The accuracy measure is determined by analyzing the performance of the model on both the validation and training datasets, providing insights into its generalization capabilities. Figure 10(d) depicts the performance analysis Tulu character recognition model by considering parameters such as f1-score, recall, and precision. In the analysis CNN model has better score compared to the other algorithms.

TABLE 4. Performance analysis of rule based machine translation.

Parameters	Accuracy	Precision	Recall	f1-score
Word Based	89%	95%	94%	94%
Sentence Based	81%	91%	90%	91%

A. CLASSIFICATION PERFORMANCE COMPARISON

The performance of the classification was evaluated using the Precision, F1 Score, and Recall metrics, as defined by equations (1), (2), and (3). Figure 9(d) shows the performance analysis of Tulu character recongition.

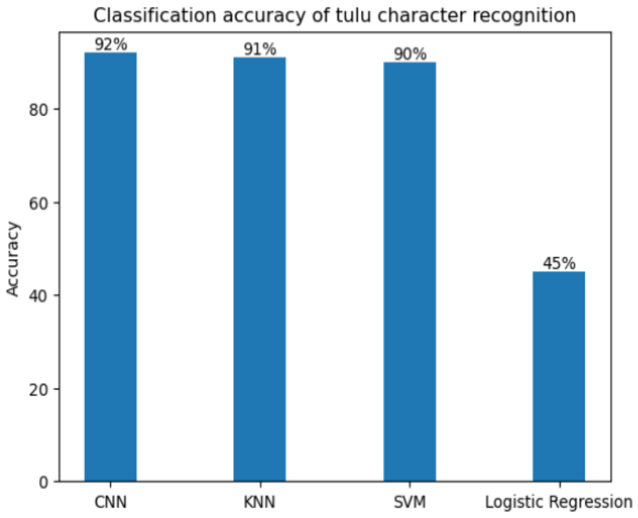


FIGURE 14. Tulu character recognition accuracy classification.

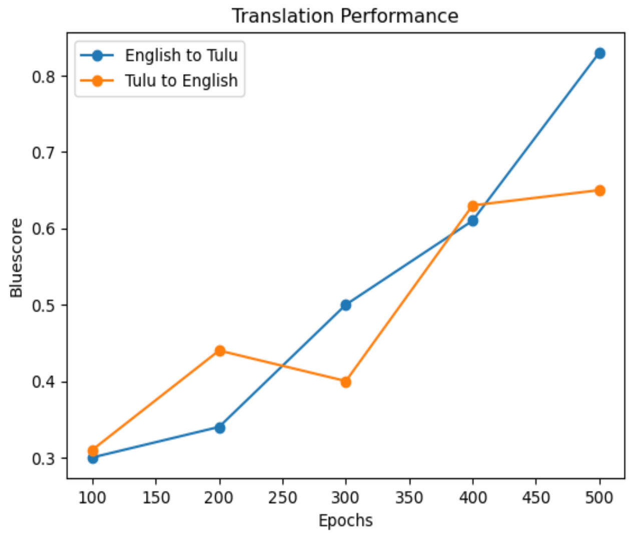


FIGURE 16. Analysis of blue score.

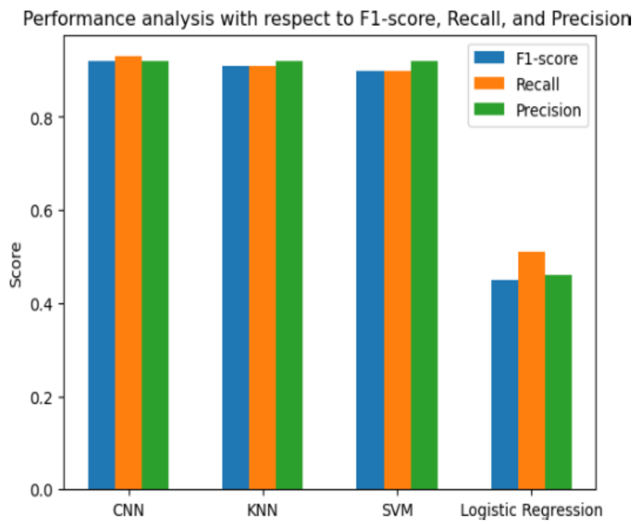


FIGURE 15. Performance analysis of Tulu character.

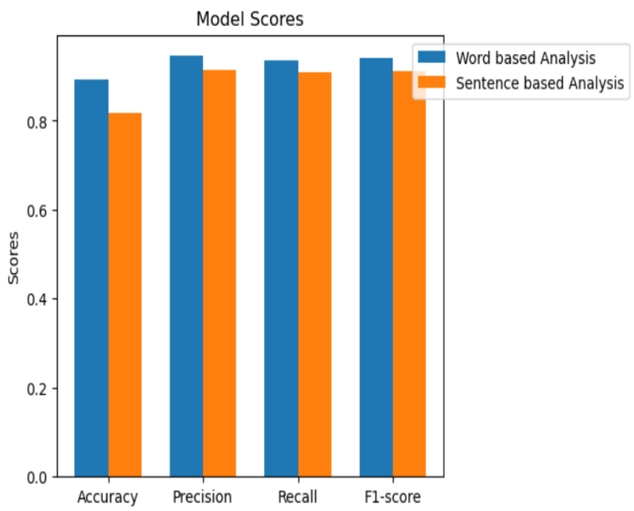


FIGURE 17. Performance analysis of rule based method.

Precision:

$$\text{Precision} = \frac{\text{TP}}{\text{FP} + \text{TP}} \tag{1}$$

Recall:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

F1 Score:

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

where:

- TP = True Positive
- TN = True Negative
- FP = False Positive
- FN = False Negative

Figures 11(c) and 11(d) illustrate the process of recognizing Tulu characters using the trained model. In this process, each Tulu character is inputted into the model, and the model produces the corresponding output in the form of either a Kannada character or a numerical representation. Figure 11(b) shows the experimental result of Text-based translation from English to Tulu model has achieved 89% accuracy on word-based translation and 81% on sentence-based translation considered dataset. For the performance analysis, the following parameters were taken into consideration: accuracy, precision, recall, and F1-score. Comparing word-based analysis to sentence-based analysis, the word-based approach yielded better results. Figure 11(a) depicts the variation in Blue score values for English to Tulu translation over 500 iterations, with a peak value of 0.83. For the training of translation from English to Tulu and viceversa we have considered single line sentence which are simple and easier for the conversion. In contrast, the Blue score values for Tulu to English translation range from 0.31 to 0.65 over the

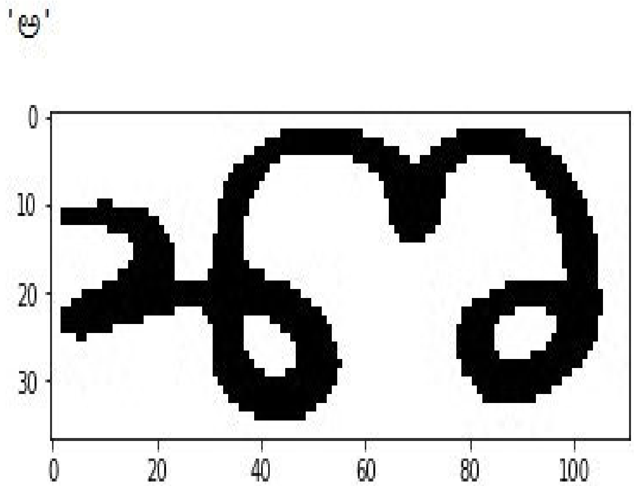


FIGURE 18. 'A' character detection.

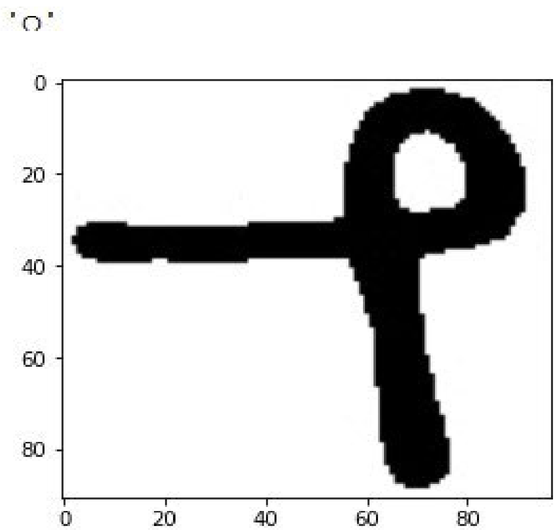


FIGURE 19. 'One' numerical detection.

same iteration range. Table 4 shows the performance analysis of rule-based machine Translation.

V. CONCLUSION AND FUTURE WORK

The research aimed to develop an English-to-Tulu language translator and a literature review of existing translators for Indian and other languages was conducted. Although many approaches provide accurate results for words and simple phrases, accuracy decreases for complex sentences. To improve accuracy, a survey paper was written based on collected research papers, and a dataset was manually collected and tested. A handwritten character recognition system was developed using CNN, achieving 92% accuracy for Tulu characters and numerals. An algorithm using a rule-based method was incorporated into the research work for English-to-Tulu translation, achieving 89% accuracy for simple words and sentences. Neural machine technology was applied to increase efficiency and achieved a blue score of 0.83. The model is currently being evaluated using individual sentences, while the assessment of phrases is slated for future consideration. However, accuracy decreased for complex sentences, indicating the need for more dataset collection to improve the system. Future work includes developing a real-time application with Tulu unicode and a more complex model and phrases can be considered in the testing of the model.

VI. CONFLICT OF INTEREST

The authors do not have any conflict of interest.

REFERENCES

[1] I. Manimozhi and M. challa, "An efficient translation of Tulu to Kannada south Indian scripts using optical character recognition," in *Proc. 5th Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Apr. 2021, pp. 952–957.

[2] S. Bhat and G. Seshikala, "Character recognition of Tulu script using convolutional neural network," in *Advances in Artificial Intelligence and Data Engineering*. Singapore: Springer, 2021, pp. 121–131.

[3] S. S. Bhat et al., "Building dataset and deep learning-based inception model for the character classification of tigalari script," in *Recent Advances in Artificial Intelligence and Data Engineering: Select Proceedings of AIDE 2020*. Singapore: Springer, 2022.

[4] C. K. Savitha and P. J. Antony, "Machine learning approaches for recognition of offline Tulu handwritten scripts," *J. Phys., Conf. Ser.*, vol. 1142, Nov. 2018, Art. no. 012005.

[5] A. S. Rao, S. Sandhya, K. Anusha, C. N. Arpitha, and S. N. Meghana, "Exploring deep learning techniques for Kannada handwritten character recognition: A boon for digitization," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 5, pp. 11078–11093, 2020.

[6] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR)," *IEEE Access*, vol. 8, pp. 142642–142668, 2020.

[7] S. Albahli, M. Nawaz, A. Javed, and A. Irtaza, "An improved faster-RCNN model for handwritten character recognition," *Arabian J. Sci. Eng.*, vol. 46, no. 9, pp. 8509–8523, Sep. 2021.

[8] M. B. Bora, D. Daimary, K. Amitab, and D. Kandar, "Handwritten character recognition from images using CNN-ECOC," *Proc. Comput. Sci.*, vol. 167, pp. 2403–2409, 2020.

[9] S. P. Deore and A. Pravin, "Devanagari handwritten character recognition using fine-tuned deep convolutional neural network on trivial dataset," *Sādhanā*, vol. 45, no. 1, pp. 1–13, Dec. 2020.

[10] I. Khandokar, M. Hasan, F. Ernawan, S. Islam, and M. N. Kabir, "Handwritten character recognition using convolutional neural network," in *J. Phys., Conf. Ser.*, vol. 1918, no. 4, 2021, Art. no. 042152.

[11] R. Guha, N. Das, M. Kundu, M. Nasipuri, and K. C. Santosh, "DevNet: An efficient CNN architecture for handwritten devanagari character recognition," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 34, no. 12, Nov. 2020, Art. no. 2052009.

[12] N. S. Rani, "Robust recognition technique for handwritten Kannada character recognition using capsule networks," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 12, no. 1, p. 383, Feb. 2022.

[13] Y. B. Hamdan, "Construction of statistical SVM based recognition model for handwritten character recognition," *J. Inf. Technol.*, vol. 3, no. 2, pp. 92–107, 2021.

[14] B. M. Vinjit, M. K. Bhojak, S. Kumar, and G. Chalak, "A review on handwritten character recognition methods and techniques," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Jul. 2020, pp. 1224–1228.

[15] N. S. Rani, A. C. Subramani, A. Kumar, and B. R. Pushpa, "Deep learning network architecture based Kannada handwritten character recognition," in *Proc. 2nd Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Jul. 2020, pp. 213–220.

[16] B. N. Athira, "Kannada confusing character recognition and classification using random forest and SVM," in *Proc. 3rd Int. Conf. Signal Process. Commun. (ICPSC)*, May 2021, pp. 537–541.

[17] M. Yadav, R. K. Purwar, and M. Mittal, "Handwritten Hindi character recognition: A review," *IET Image Process.*, vol. 12, no. 11, pp. 1919–1933, 2018.

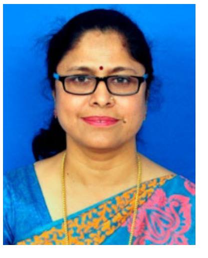
[18] T. Ganji, M. S. Velpuru, and R. Dugyala, "Multi variant handwritten Telugu character recognition using transfer learning," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1042, no. 1, 2021, Art. no. 012026).
[19] M. P. Ayyoob and P. M. Ilyas, "A review on various techniques used to recognize off-line handwritten Malayalam characters," *Malaya J. Matematik (MJM)*, vol. 9, p. 1093, Jan. 2021
[20] S. Sreelekha, "Machine translation between Malayalam and English," *Linguistics J.*, vol. 14, no. 2, pp. 7–31, 2020.
[21] M. A. Islam, M. S. H. Anik, and A. B. M. A. A. Islam, "An enhanced RBMT: When RBMT outperforms modern data-driven translators," *IETE Tech. Rev.*, vol. 39, no. 6, pp. 1473–1484, Nov. 2022.
[22] N. Bhadwal, P. Agrawal, and V. Madaan, "A machine translation system from Hindi to Sanskrit language using rule based approach," *Scalable Comput., Pract. Exper.*, vol. 21, no. 3, pp. 543–554, 2020.
[23] Sitender and S. Bawa, "A Sanskrit-to-English machine translation using hybridization of direct and rule-based approach," *Neural Comput. Appl.*, vol. 33, pp. 2819–2838, 2021.
[24] N. G Kharate and D. V. H. Patil, "Handling challenges in rule based machine translation from Marathi to English," *Int. J. Natural Lang. Comput.*, vol. 8, no. 4, pp. 39–49, Aug. 2019.
[25] M. M. Kodabagi and S. A. Angadi, "A methodology for machine translation of simple sentences from Kannada to English language," in *Proc. 2nd Int. Conf. Contemp. Comput. Informat. (IC3I)*, Dec. 2016, pp. 237–241.
[26] P. Salunkhe, Aniket. D. Kadam, S. Joshi, S. Patil, D. Thakore, and S. Jadhav, "Hybrid machine translation for English to marathi: A research evaluation in machine translation: (Hybrid translator)," in *Proc. Int. Conf. Electr., Electron., Optim. Techn. (ICEEOT)*, Mar. 2016, pp. 924–931.
[27] I. Arikpo and I. Dickson, "Development of an automated English-to-local-language translator using natural language processing," *Int. J. Sci. Eng. Res.*, vol. 9, no. 7, pp. 378–383, 2018.
[28] R. Mardhotillah, B. Dirgantoro, and C. Setianingsih, "Speaker recognition for digital forensic audio analysis using support vector machine," in *Proc. 3rd Int. Seminar Res. Inf. Technol. Intell. Syst. (ISRITI)*, Dec. 2020, pp. 514–519.
[29] P. Dhar, A. Bisazza, and G. van Noord, "Optimal word segmentation for neural machine translation into Dravidian languages," in *Proc. 8th Workshop Asian Transl. (WAT)*, 2021, pp. 181–190.
[30] 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, Jul. 2019.
[31] R. Prajapati, V. V. Parikh, and P. Majumder, "Irlab-daiict@dravidianlangtech-eacl2021: Neural machine translation," in *Proc. 1st Workshop Speech Lang. Technol. Dravidian Lang.*, Apr. 2021, pp. 262–265.
[32] H. Choudhary, S. Rao, and R. Rohilla, "Neural machine translation for low-resourced Indian languages," 2020, *arXiv:2004.13819*.
[33] S. Gogineni, G. Suryanarayana, and S. K. Surendran, "An effective neural machine translation for English to Hindi language," in *Proc. Int. Conf. Smart Electron. Commun. (ICOSEC)*, Sep. 2020, pp. 209–214.
[34] A. Hegde, H. L. Shashirekha, A. K. Madasamy, and B. R. Chakravarthi, "A study of machine translation models for Kannada-Tul," in *Proc. 3rd Congr. Intell. Syst. (CIS)*. Singapore: Springer Nature, vol. 1, Mar. 2023, pp. 145–161.
[35] R. Fernandes and A. P. Rodrigues, "Kannada handwritten script recognition using machine learning techniques," in *Proc. IEEE Int. Conf. Distrib. Comput., VLSI, Electr. Circuits Robot. (DISCOVER)*, Aug. 2019, pp. 1–6.
[36] Prathwini, R. Fernandes, and A. P. Rodrigues, "Emotion detection in multimedia data using convolution neural network," in *Proc. Int. Conf. Artif. Intell. Data Eng. (AIDE)*, Dec. 2022, pp. 157–161.
[37] Z. Dou, Y. Sun, J. Zhu, and Z. Zhou, "The evaluation prediction system for urban advanced manufacturing development," Systems, vol. 11, no. 8, p. 392, 2023.

PRATHWINI was born in Mangaluru, India, in 1998. She received the B.E. degree in information science and engineering and the M.Tech. degree in computer science and engineering from VTU, Belagavi, India, in 2019 and 2023, respectively.

Currently, she is an Assistant Professor with the Department of Master of Computer Applications, NMAM Institute of Technology, NITTE University, India. She is the author of two research articles. Her research interests include natural language processing, machine learning, and deep learning.

ANISHA P. RODRIGUES was born in Mangaluru, India, in 1985. She received the B.E. degree in electrical and electronics engineering, the M.Tech. degree in computer science and engineering, and the Ph.D. degree in big data analytics from VTU, Belagavi, India, in 2006, 2012, and 2020, respectively.

Since 2012, she has been an Associate Professor with the Department of Computer Science and Engineering, NMAM Institute of Technology, NITTE University, India. She is the author of three book chapters and 20 articles. Her research interests include natural language processing, machine learning, deep learning, and big data analytics.



P. VIJAYA was born in Tiruchirappalli, India, in 1969. She received the Diploma degree in electronics and communication, the B.E. degree in electronics and communication engineering, the M.Tech. degree in computer science and engineering, and the Ph.D. degree in the area of information retrieval from the Karpagam Academy of Higher Education, Coimbatore, India, in 1987, 1995, 2002, and 2017, respectively.

Currently, she is with the Department of Mathematics and Computer Science, Bawshar, Muscat, Oman. She is the author of four book chapters and 36 articles. Her research interests include information retrieval, machine learning, deep learning, and big data analytics.

ROSHAN FERNANDES was born in Mangaluru, India, in 1979. He received the B.E. degree in computer science and engineering from Mangalore University, in 2001, and the M.Tech. degree in computer engineering and the Ph.D. degree in the area of mobile web services from VTU, Belagavi, India, in 2007 and 2019, respectively.

Currently, he is a Professor with the Department of Computer Science and Engineering, NMAM Institute of Technology, NITTE University, India. He has a teaching experience of 18 years in this institution. He is the author of four book chapters and 18 articles. His research interests include mobile web services, natural language processing, machine learning, and deep learning.