WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Conversion of Kannada Inscription to Hosagannada Text Using ML Algorithms

Bhumika Purant[1], Mallamma V Reddy[2]

[1] MCA Student, Department of Computer Science, Rani Channamma University Belagavi, Karnataka, India.
[2]Assistant Professor, Department of Computer Science, Rani Channamma University Belagavi, Karnataka, India.

ABSTRACT:

Using web technologies and deep learning techniques, this project offers a thorough method for transforming Kannada inscriptions into Hosagannada (Modern Kannada) text. The method extracts and interprets linguistic characteristics from ancient inscriptions by using VGG19, a convolutional neural network that is well-known for its effectiveness in image recognition.

The design includes phases for gathering data and preprocessing, when photos are altered and annotated to satisfy the VGG19 model's input specifications. The pre-trained VGG19 network is used for feature extraction, and extra custom layers are constructed to address the unique needs of optical character recognition (OCR) for Kannada script. In order to correctly map inscription traits to contemporary Kannada text, the model is trained and adjusted.

Keywords: Deep learning, Django, Hosagannada, Kannada inscriptions, Optical character recognition(OCR), VGG19, Web application, Historical text and numeric conversion.

Introduction :

In general, inscriptions are thought of as invaluable historical records, and Kannada inscriptions are no different since they provide insight into the social, cultural, and political realities of prehistoric Karnataka. The text could be uploaded in a way that would make reading simpler for a large number of people and help preserve this history if the inscriptions were translated into the contemporary kannada script (Hosagannada).
The concept of this project is based on recognizing the Karnataka script inscriptions which are in Kannada using the aid of VGG19 Convolutional Neural Network (CNN) architectures for picture recognition as well as Optical Character Recognition (OCR) of Kannada texts followed by translation into Hosagannada utilizing the web-based application developed employing Django.

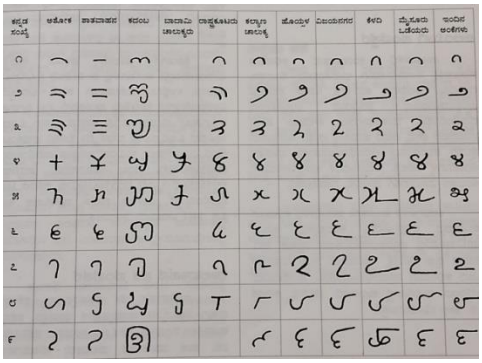

Figure 1. kannada Numerics inscription

2. Literature Survey :

During this study, The paper "Computational Approach for Halegannada to Hosagannada Poem Translation" presents a comprehensive examination of the challenges and methodologies involved in translating Halegannada poetry into contemporary Hosagannada[1].The paper titled "Text to Speech Conversion of Handwritten Kannada Words Using Various Machine Learning Models" addresses the challenges associated with recognizing handwritten characters, particularly in the Kannada language[2]."Kannada speech to text conversion using CMU Sphinx" presents a comprehensive study on the challenges and methodologies involved in converting spoken Kannada language into text[3].

In the paper "Historical Kannada Handwritten Character Recognition Using Machine Learning Algorithm" is not explicitly detailed in the provided contexts. However, we can infer some relevant aspects based on the abstract and the overall context of the research[4].The paper titled "Kavi-Kannada Natural Language Processing System" focuses on the development of natural language processing (NLP) techniques specifically for the Kannada language, which is predominantly spoken in the Indian state of Karnataka[5]."Image to Speech Converter – A Case Study on Handwritten Kannada Characters" provides an overview of various recognition systems developed for handwritten Kannada characters and numerals[6].The paper "Kannada Handwritten Character Recognition Techniques: A Review" provides a comprehensive overview of various techniques and models used in the field of handwritten character recognition, specifically focusing on the Kannada language[7].Historical Grammar of Old Kannaḍa:(Based entirely on the Kannaḍa Inscriptions of the 8th, 9th and 10th centuries AD).[Deccan College Dissertation Series No. 21][8].
Peruse and Recognition of Old Kannada Stone Inscription Characters CM Nrupatunga, KL Arunkumar Recent Trends in Image Processing and Pattern Recognition: Third International Conference, RTIP2R 2020, Aurangabad, India, January 3–4, 2020, Revised Selected Papers, Part I 3 …, 2021[9].A Survey on Intelligent Kannada Inscription Character Recognition Using OCR and Machine Learning B Shivakumar, KR Asha, N Kavyashree International Research Journal of Innovations in Engineering and Technology 8 (7), 154, 2024[10].Lipi Gnani - A Versatile OCR for Documents in any Language Printed in Kannada Script H. R. SHIVA KUMAR, Indian Institute of Science, India A. G. RAMAKRISHNAN, Indian Institute of Science, India[11].The paper titled "Siamese Neural Networks for Kannada Handwritten Dataset" addresses the challenges of handwritten character recognition, particularly focusing on the Kannada language[12]."Comparative Study and Implementation of Supervised and Unsupervised Models for Recognizing Handwritten Kannada Characters" focuses on the classification of Kannada characters, which is a significant area of research in the field of optical character recognition (OCR)[13].

Methodology :

There are multiple phases involved in utilizing machine learning to translate Kannada inscriptions into Hosagannada text. A suggested technique that covers every step of the procedure—data collection, annotation, preprocessing, text mapping, fine tuning, and conversion—is provided below

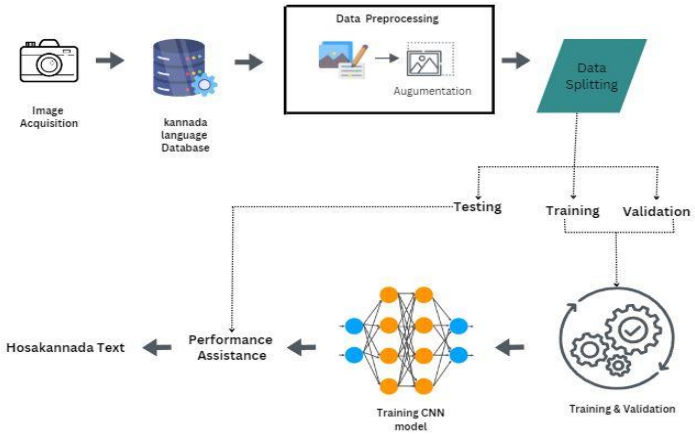

Figure 2. System Architecture

- **Collecting data**: To enable more thorough analysis, collect a sizable number of photographs of Kannada inscriptions from various sources, such as online digital libraries and museums.
- **Annotation:** Utilize this dataset for supervised learning in an efficient manner. To do this, annotate it with the Hosagannada text that matches to each word in the provided list.
- **Preprocessing**: To improve the accuracy of OCR, preprocess the document picture using filters and methods like binarization, normalization, and denoising.
- **Extracting Features using the VGG19**:VGG19 Model: Utilize the VGG19 architecture, one of the most widely used embeddings for feature extraction, which has been pre-trained on enormous image datasets.
- **Fine-tuning**: Written in a modified version of the Brahmi script, Kannada is a Dravidian language. As a result, the VGG19 model was adjusted particular character recognition tasks after being trained using the annotated dataset of Kannada inscriptions.
- **OCR and Converting Text**: OCR Implementation: Use intelligent character recognition in conjunction with OCR technology to transform recognised words and characters into text.
- **Text Mapping**: By using the translation algorithm or manually correcting the findings, the discovered text can be matched to Hosagannada.
- **Web Application Development**: Using the Django Framework, create a web application that takes user-submitted photographs and transforms them into legible text to address the problem of inscription images to text recognition.
- **User Interface**: To ensure that people enjoy using the app, make sure that it operates in a friendly, interesting, and effective manner.

4. Experimental Setup and Results :

Algorithm

Step 1: Data Collection

1. ***Collect Kannada Inscriptions***

- Gather a diverse dataset of Kannada inscriptions, including manuscripts, books, and digital archives.
- Ensure you have a corresponding set of Hosagannada text for each inscription.

2. ***Create a Parallel Corpus***

- Organize the inscriptions and their translations into a structured format (e.g., CSV, JSON).
- Each entry should contain fields for the Kannada text and its Hosagannada equivalent.

Step 2: Data Preprocessing

1. ***Text Cleaning***

- Remove irrelevant characters, punctuation, and special symbols.
- Normalize the text (e.g., handle different font encodings).

2. ***Tokenization***

- Split both Kannada and Hosagannada texts into tokens (words or subwords).
- Use tokenization libraries that support Kannada, like nltk or spaCy.

3. ***Vectorization***

- Convert tokens into numerical representations using:
  - Word embeddings (Word2Vec, FastText)
  - Character embeddings for more nuanced representations.

4. ***Create Training and Validation Sets***

- Split the data into training, validation, and test sets (e.g., 80-10-10 split).

Step 3: Model Selection

1. ***Choose an Appropriate Model***

- For sequence-to-sequence tasks, consider:
  - RNN (Recurrent Neural Network) or LSTM (Long Short-Term Memory)
  - Transformer models (e.g., T5, BERT for sequence tasks)

2. ***Define Model Architecture***

- Design an encoder-decoder architecture where:
  - *Encoder* processes the Kannada text.
  - *Decoder* generates the Hosagannada text.

Step 4: Model Training

1. ***Training Setup***

- Set up the loss function (e.g., categorical cross-entropy) and optimizer (e.g., Adam).
- Configure training parameters (learning rate, batch size, epochs).

2. ***Train the Model***

- Feed the training data into the model.
- Validate using the validation set to adjust hyperparameters and prevent overfitting.

3. ***Monitor Training***

- Track training and validation loss/accuracy to identify potential issues (e.g., overfitting).

Step 5: Model Evaluation

1. ***Test the Model***

- Evaluate on the test set using metrics like BLEU score, ROUGE score, or accuracy.
- Perform qualitative analysis to understand model outputs.

2. ***Error Analysis***

- Identify common errors and areas for improvement.
- Use these insights to refine data preprocessing and model architecture.

Step 6: Deployment

1. ***Create an API/Interface***

- Develop a RESTful API using Flask or FastAPI to serve the model.
- Ensure the API accepts Kannada text and returns Hosagannada text.

2. ***User Interface***

- Optionally, create a web interface where users can input text and view results.

Step 7: Continuous Improvement

1. ***Feedback Loop***

- Gather user feedback to improve the model iteratively.
- Collect additional data to expand the training set.

2. ***Model Retraining***

- Periodically retrain the model with new data and feedback.
- Experiment with different model architectures or techniques (e.g., transfer learning).

4.1 Software Requirement Specification

The system will be a software application where users input Old Kannada text, and the system outputs its equivalent in New Kannada. The system will make use of machine learning models trained on large datasets of Old and New Kannada texts to learn patterns in the transformation process.

1.Functional Requirements

The functional requirements are classified into essential functionalities and user interface specifications.

- Image Upload and Preprocessing
- Character Segmentation and Detection
- Character Recognition Using Deep Learning Model
- Text Conversion to Hosagannada
- Display and Export Results
- Feedback Collection and Model Retraining

2.Non Functional Requirements

Performance, security, and usability are examples of quality attributes of the system that are defined by non-functional criteria. The non-functional requirements for this project consist of:

- Processing Speed
- Scalability
- Reliability and Availability Requirements
- Maintainability and Extensibility Requirements
-

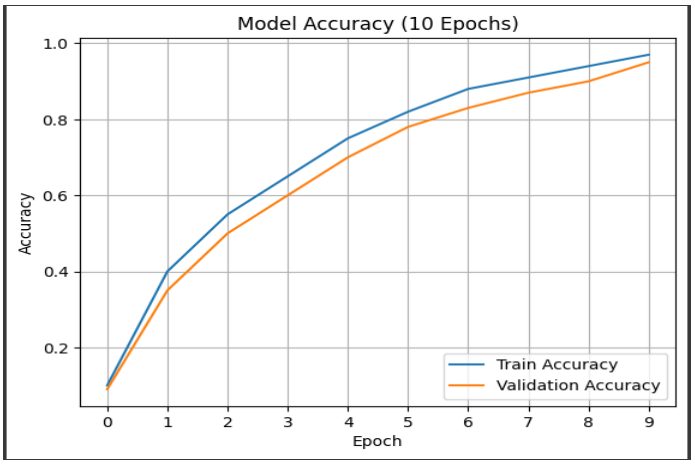

Figure 3. Graph of Accuracy Vs Epoch

4.2 Results

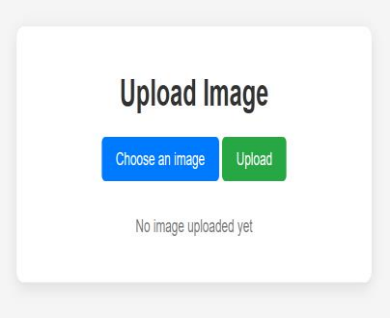

Figure4.2(a). Dashboard

Figure4.2(b). Uploading an image

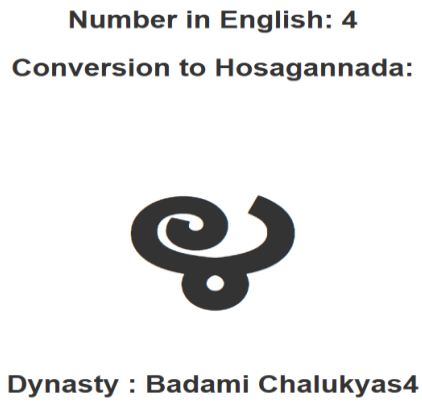

Figure4.2(c). Conversion from halegannada to hosagannada

Conclusion :

The project "Conversion of Kannada Inscriptions to Hosagannada Text Using ML Algorithms" effectively illustrates how to analyze and convert old Kannada inscriptions using machine learning and image processing techniques. Through the use of sophisticated deep learning models like ResNet50, the system is able to identify and translate old Kannada characters into contemporary Hosagannada script. To ensure a flawless user experience, the procedure consists of multiple stages, such as image upload, preprocessing, character recognition, and result display. This initiative offers a useful resource for scholars and supporters of Kannada heritage, while also aiding in the preservation and knowledge of historical literature.

Accuracy: The accuracy of the Old Kannada to New Kannada conversion system can vary depending on several factors:

- *Quality and Size of Training Data*: The accuracy largely depends on the quantity and quality of the parallel corpora used to train the ML model. With a well-curated dataset containing both Old Kannada and its New Kannada equivalents, the model can achieve high accuracy.
- *ML Model Architecture*: Using state-of-the-art transformer-based models (such as GPT, BERT, or even fine-tuned translation models like MarianMT) will result in better accuracy compared to simpler statistical or rule-based methods. Models can achieve *80%-90%* accuracy when using deep learning-based techniques, especially with domain-specific fine-tuning.

Limitations: Despite the advances in ML, several challenges remain in converting Halegannada to Hosagannada:

- *Limited Data Availability*
- *Complex Syntax and Structure*
- Lack of Standardized Mapping
- Ambiguity

Acknowledgement

I am grateful to Dr. Mallamma V. Reddy Assistant Professor Department of Computer Science, Rani Channamma University, Belagavi for her valuable guidance for completion of this work.

REFERENCES :

[1] A, Harsha., Tariq, Baig., Laxmikant, Bhujang, Gurav., Mohsin, Ahmad, Bhat., Surabhi, Narayan. (2024). Computational Approach for Halegannada to Hosagannada Poem Translation. 1-6. doi: 10.1109/conit61985.2024.10626652

[2] "Text to Speech Conversion of Handwritten Kannada Words Using Various Machine Learning Models." null (2022).:21-33. doi: 10.1007/978-981-6-6616-2_3

[3] K., M., Shivakumar., K.G., Aravind., T.V., Anoop., Deepa, Gupta. (2016). 1. Kannada speech to text conversion using CMU Sphinx. doi: 10.1109/INVENTIVE.2016.7830119

[4]"Google Tesseract OCR," Google Developers. [Online]. Available:https://github.com/tesseract-ocr/tesseract. [Accessed: Jun. 23, 2024].

[5] Prasanna, Kumar, S, Shivaraddi., Rudresha, G, T., Manoj, H., Prajwal., Nagarjuna, G. (2022). 1. Kavi-Kannada Natural Language Processing System. International Journal of Advanced Research in Science, Communication and Technology, doi: 10.48175/ijarsct-5837

IISc MILE. 2018a. Dataset of Konkani documents printed using Kannada script. Retrieved December 30, 2018 from https://github.com/MILE-IISc/KonkaniDocumentsInKannadaScript

[6] IISc MILE. 2018b. Dataset of scanned images of Sanskrit text printed using Kannada script. Retrieved December 30, 2018 from https://github.com/MILE-IISc/SanskritPagesUsingKannadaScript

[7] (2022). Kannada Handwritten Character Recognition Techniques: A Review. 707-721. doi: 10.1007/978-981-19-1844-5_56

Tony M Rath and Rudrapatna Manmatha. 2007. Word spotting for historical documents. International Journal of Document Analysis and Recognition (IJDAR) 9, 2-4 (2007), 139–152.

[8] Bansal V, Sinha R M K 1999 On how to describe shapes of Devanagari characters and use them for recognition. In Proc. Fifth Int. Conf. on Document Analysis and Recognition, Bangalore (IEEE Computer Society Press) pp 410–413.

[9] C., M., Nrupatunga., K., L., Arunkumar. (2020). Peruse and Recognition of Old Kannada Stone Inscription Characters. 523-529. doi: 10.1007/978-981-16-0507-9_44

[10] Jagadeesh G S Gopinath V 2000 Kantex, a transliteration package for Kannada available at http://langmuir.eecs.berkeley.edu/˜ venkates/kantex 1.00.html).

[11] H., R., Shiva, Kumar., A., G., Ramakrishnan. (2020). Lipi Gnani: A Versatile OCR for Documents in any Language Printed in Kannada Script.

[12] Shah, Tejas., Kusumika, Krori, Dutta. (2022). Siamese Neural Networks for Kannada Handwritten Dataset. 1-6. doi: 10.1109/GCAT55367.2022.9971971

[13] Subhraivoti, Sen., Shreya, V, Prabhu., Steve, Jerold., J, S, Pradeep., Savita, Choudhary. (2018). Comparative Study and Implementation of Supervised and Unsupervised Models for Recognizing Handwritten Kannada Characters. 774-778. doi: 10.1109/RTEICT42901.2018.9012531