# Implementation of Recognition Model for Ancient Kannada Epigraphs Kannada Epigraphs using Raspberry Pi

Anusha L.S.
Dept. of ECE, RVCE
Visvesvaraya Technological University
Karnataka, India

Abhay A. Deshpande, PhD
Dept. of ECE, RVCE
Visvesvaraya Technological University
Karnataka, India

## ABSTRACT
The decipherment and transcription of ancient manuscripts present substantial challenges due to their frequently fragmented, deteriorated, and stylistically heterogeneous nature. This paper introduces an innovative methodology for ancient text recognition through the application of Convolutional Neural Networks (CNNs) implemented on a Raspberry Pi 3B+ for enhanced sensor integration. The approach leverages advanced deep learning paradigms to preprocess and analyze historical documents, thereby enhancing the precision of text extraction and interpretation. The CNN architecture proposed is trained on a heterogeneous corpus encompassing diverse scripts and languages, thus improving its capacity to generalize across disparate ancient writing systems. Techniques such as data augmentation, transfer learning, and customized network architectures are employed to address challenges such as reduced resolution and significant variability in textual appearance. Empirical results demonstrate that CNN-based methodologies significantly surpass conventional optical character recognition (OCR) approaches, achieving notable advancements in both accuracy and efficiency. This research provides a robust framework for the automated decipherment of ancient texts using the Raspberry Pi 3B+, potentially advancing further historical and philological scholarship.

## Keywords
Ancient text, Deep Learning Algorithm Convolutional Neural Network (CNN), Raspberry pi3b+

## 1. INTRODUCTION
Kannada, the official language for sixty million individuals in India, is written in an abugida script, where each character represents a consonant or vowel phoneme. The inscriptions are typically rendered diagonally from left to right. As one of the ancient Dravidian languages, Kannada boasts a rich and diverse linguistic heritage, encapsulated in a vast corpus of historical manuscripts. The utilization of Convolutional Neural Networks (CNNs) for the recognition of ancient Kannada texts encounters considerable challenges. These include the fragmented and deteriorated condition of the

**Fig1: Evolution of Kannada**

The presence of noise and artifacts within the manuscripts further complicates text extraction, obscuring the clarity of the original content. Manual decipherment necessitates extensive expertise and is labor-intensive, often subject to human error. While the integration of CNNs addresses these issues by enhancing the precision and efficiency of text recognition, significant obstacles remain, such as adapting to diverse script variations and mitigating the impact of degraded text quality. Despite these challenges, the application of CNNs has yielded substantial improvements in both the accuracy and overall efficacy of the decipherment process.

The language has additionally grown numerous times over the years, as demonstrated in Figure 1. The present Kannada language is distinct from outdated publications. The primary aim of utilizing Convolutional Neural Networks (CNNs) for ancient text recognition is to develop a highly effective system that can precisely identify and decode historical scripts and manuscripts. CNNs, known for their ability to capture spatial hierarchies in images, can be trained to recognize intricate and varied features of ancient text characters[1]. This allows for: Enhanced Accuracy-CNNs can effectively handle the complexities of ancient scripts, which often include diverse fonts, sizes, and degradation patterns. Automatic Extraction-Automating the recognition process helps in digitizing and preserving ancient texts, making them more accessible for research and analysis. Historical Analysis-The system aids in the study of historical texts by providing tools for easier interpretation and comparison of different manuscripts. Overall, the use of CNNs can significantly advance the field of text recognition in historical documents, facilitating greater understanding and preservation of ancient cultural heritage[2]. In addition to these capabilities, the hardware integration is instrumental in optimizing the performance and applicability of the CNN-based system. The implementation of this system on

a Raspberry Pi 3B+ coupled with a LCD display augments its functionality and accessibility. The Raspberry Pi 3B+ serves as a compact and economically feasible computing platform, ideally suited for field applications and portable usage. Although its computational resources are relatively modest, they are adequate for the execution of CNN models tailored to text recognition tasks. The display offers an intuitive user interface, facilitating interactive operations such as image uploads, model training, and text recognition directly from the device. This hardware configuration not only enhances user-friendliness but also enables on-site analysis and processing, making it particularly advantageous in environments where conventional computing resources might be unfeasible. Collectively, the synergy between CNNs and the Raspberry Pi 3B+ with a LCD display constitutes a potent and accessible solution for advancing the domain of historical text recognition and preservation.

## 2. HISTORY OF KANNADA TEXT RECOGNITION

The history of Kannada text recognition traces the evolution of methods and technologies used to digitize and interpret the Kannada language, a Dravidian language spoken predominantly in Karnataka, India. Here is a brief overview of its development:

**Early Manual Methods:** Script Analysis: Initially, text recognition efforts involved manual analysis of Kannada script, which required extensive knowledge of the script's unique characters. Early efforts included using typewriters with Kannada keys and rudimentary Optical Character Recognition (OCR) systems designed for printed texts.

**Early Computational Approaches:** Rule-Based Systems: In the late 20th century, rule-based OCR systems were developed that used predefined rules to recognize Kannada characters. These systems struggled with variations in handwriting and different fonts.

**Machine Learning Integration:** Traditional Algorithms: The introduction of machine learning techniques in the 2000s brought improvements through algorithms like Support Vector Machines (SVMs) and Hidden Markov Models (HMMs), which enhanced recognition accuracy by learning from large datasets. Feature Extraction: Techniques such as feature extraction and pattern recognition were used to improve the identification of Kannada script variations and styles[3].

**Advancement with Deep Learning:** Convolutional Neural Networks (CNNs): The 2010s saw the advent of deep learning approaches, particularly CNNs, which significantly improved the accuracy of Kannada text recognition. CNNs could learn complex features of the script from large annotated datasets, handling variations in fonts, sizes, and handwriting more effectively.

Recurrent Neural Networks (RNNs) and Transformers: More recently, the integration of RNNs and Transformer-based models has further advanced recognition capabilities, enabling better handling of context and sequential information in Kannada text [4].

**Current and Future Trends:** Multilingual Models: Modern systems often incorporate multilingual models that support Kannada alongside other languages, enhancing their versatility and accuracy.

Contextual Understanding: Ongoing research aims to improve contextual understanding and semantic analysis of Kannada text, moving beyond mere character recognition to grasping the meaning and overall, the history of Kannada text recognition reflects a progression from manual methods to advanced computational approaches [5], driven by technological advancements and the increasing availability of data.

## 3. METHODOLOGY

A standard character recognition model consists of six key phases: data gathering, Preprocessing is a fundamental phase in the preparation. It involves the application of various techniques to refine and standardize the data, ensuring its suitability for subsequent analysis. This stage is crucial for enhancing data quality, mitigating noise, and normalizing data formats, which collectively contribute to the effectiveness and accuracy of the model training process.

This process is particularly crucial for historical texts, which may be discovered or preserved under suboptimal conditions and may not be readily suitable for direct input into neural network models [6], [7]. Depending on the nature of the epigraphic material, preprocessing involves several stages, including grayscale conversion, binarization, and edge detection. Each stage addresses specific aspects of data enhancement and transformation, facilitating the effective extraction and analysis of textual information from degraded or challenging source materials, and cropping, smoothing, and line and character segmentation.

### 3.1 Data acquisition and preprocessing.

Preprocessing serves as a pivotal stage in image processing, profoundly influencing the precision and effectiveness of subsequent analytical and computational tasks as shown in figure 2.

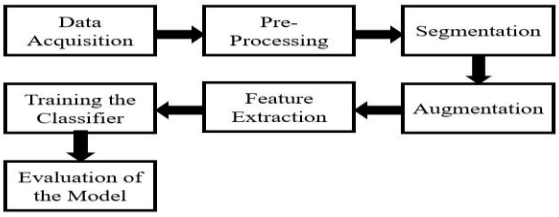


**Fig 2: Process of image processing**

In this phase, first involves the acquisition of input image of stone inscription as shown in figure 3. This phase also involves the strategic application of techniques designed to refine. The extraction of salient features while reducing the computational burden in subsequent stages is achieved by attenuating noise and artifacts, accentuating essential information, and enhancing overall image fidelity [8]. Moreover, preprocessing entails the normalization of image dimensions and orientation, as well as the correction of illumination variances. Additionally, preprocessing encompasses the normalization of image dimensions and orientation, adjustment for illumination discrepancies, and enhancement of the visual clarity of the image. This preparatory step is essential for advancing the accuracy, efficiency, and applicability of image processing [9].

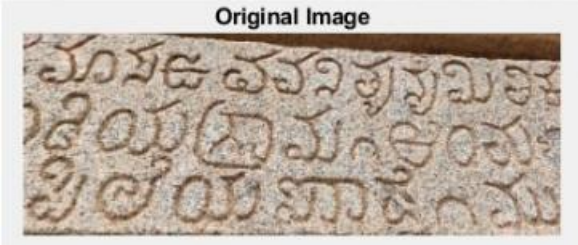


**Fig 3: Input image of stone inscription**

### *3.1.1. Gray scaling Images*

Digital images are represented as matrices of integer values ranging from 0 to 255, which encode pixel intensity levels and facilitate the representation of color saturation. In color images, the red, green, and blue (RGB) components are combined in varying proportions to produce a full spectrum of colors. This process entails the digital acquisition of RGB values, which the computer then integrates to dynamically generate the color output [10]

In the context of neural networks, particularly those employing convolutional operations, the image data is processed through its individual red, green, and blue channels. Each channel is treated as a separate array, necessitating convolution operations to be performed on three distinct arrays rather than one [11]. This approach, while accurate, introduces computational complexity. from figure 4. Consequently, reducing ---image is three-dimensional (3D) pixel values to a one-dimensional (1D) grayscale representation simplifies the convolution process by consolidating the data into a single channel, thus streamlining the calculations, and reducing computational demands.

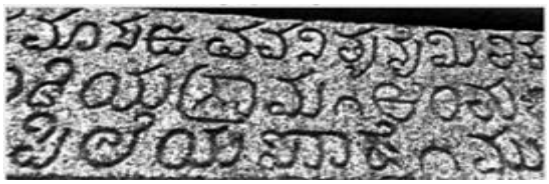

**Fig 4: Gray scaling Image of stone inscription**

Convert a colored image to grayscale is a linear transformation that considers the different contributions of the red, green, and blue channels to the perceived brightness of a pixel. The most used formula for this conversion is:

**Gray=0.299×Red+0.587×Green+0.114×Blue**

This formula is based on the luminance perception of the human eye, which is more sensitive to green light and less sensitive to blue light.

### *3.1.2. Edge detection*

Thresholding is a fundamental technique in the realm of image processing, particularly within document scanning and analysis. The principal aim of binarization is to preserve the essential features of the original image while diminishing the impact of noise [12]. This process of image thresholding transforms grayscale images into a binary format, effectively delineating foreground elements from the background and facilitating clearer, more focused analysis, involves converting an image into a binary format based on pixel intensity values. This process simplifies each pixel's data representation, which accelerates subsequent computational operations [13]. Typically, thresholding requires a grayscale image and a specified threshold value as inputs. The resulting output is a binary image wherein pixels are assigned either a white (foreground) or black (background) value. Specifically, an output pixel is turned to black if its intensity in the input image is less than or equal to the threshold, and white if its intensity in the image is greater than or equal to the threshold. Otsu's thresholding algorithm generally follows these steps: i) processing the given image, ii) computing the histogram of pixel intensity values, iii) determining the optimal threshold value TTT, and iv) binarizing the image by assigning white to pixels exceeding the threshold while retaining the desired levels of saturation and contrast.

After grayscale conversion and binarization, the image undergoes cropping to eliminate extraneous elements and enhance its quality. Cropping involves adjusting or removing the peripheral regions of an image, typically a photograph, to refine its framing or composition, accentuate the primary subject, or modify its size and aspect ratio. Following the cropping process, edge detection is employed to identify regions within the image where there are pronounced variations in brightness, commonly referred to as discontinuities.

The Canny edge detection algorithm has been identified as the most effective method for the model utilized in manuscript and inscription recognition tasks. The Canny edge detection algorithm is renowned for its robustness and precision and is extensively applied across various fields. However, it is crucial to acknowledge that noise pollution within imaging environments can present significant challenges. Such noise can obscure or disrupt the detection of image edges, potentially leading to inaccuracies and incomplete edge features. To address these issues, an image edge identification technique employing the canny algorithm has been proposed to mitigate the effects of salt-and-pepper noise and enhance the extraction of edge information from the regions of interest [14].

The Canny edge detection algorithm is widely recognized as one of the most popular, effective, and sophisticated techniques for edge detection. It employs a multi-stage process to identify and delineate various types of edges within an image. The key steps involved in implementing the canny algorithm are as follows:

Grayscale Conversion: Initially, the image must be converted to grayscale to simplify the edge detection process by eliminating color information.

Noise Reduction: Given that edge identification using derivatives is sensitive to noise, it is essential to apply a noise reduction filter, such as a Gaussian blur, to mitigate noise interference.

Gradient Calculation: The algorithm computes the gradient of the image to determine the magnitude and direction of the edges. This step is crucial for assessing the edge's orientation and strength.

Double Thresholding: The algorithm uses a dual thresholding mechanism to classify pixels into strong, weak, and non-relevant categories. This helps in differentiating between significant edges and noise, as shown in figure 5.

Hysteresis Edge Tracking: Finally, hysteresis edge tracking is employed to link weak pixels to strong pixels if they are connected, thereby ensuring that weak edges are retained only if they are part of a strong edge.

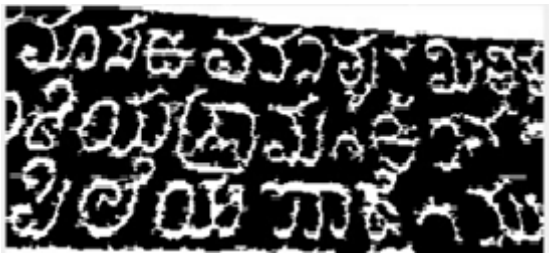

**Fig 5: Edge detection Image of stone inscription**

### *3.1.3. Smoothing*

Smoothing is performed prior to the segmentation of text within datasets of epigraphic materials, such as stone inscriptions. Smoothing is a technique utilized to reduce noise and pixelation in images [15]. Typically, smoothing algorithms employ low-pass filters; however, alternative methodologies

include averaging or applying a median filter to a kernel, which represents a moving window of pixels. The primary objective of image smoothing techniques is to enhance image quality by attenuating noise while preserving the essential structural features of the image.

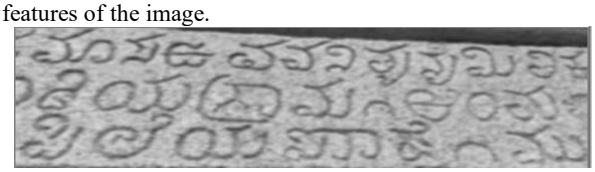

**Fig 6: Smoothened Image of stone inscription**

The Smoothing addresses various types of noise, such as additive, impulsive, and multiplicative noise from figure 6. By effectively mitigating these noise components, smoothing enhances the precision of text segmentation in inscriptions and epigraphs.

### *3.1.4 Character segmentation*

Character segmentation is a key step in the segmentation of fragmented characters is a critical factor in assessing the [16].partition the characters within the text of an image into discrete units. and identifiable units. The necessity of this level of Segmentation is contingent upon specific application of Optical Character Recognition (OCR) technology. When OCR is utilized for text in which characters are discrete and separated, character-level segmentation may not be requisite.

**Fig 7: Character Segmentation Image of stone inscription**

Frequent inaccuracies in this assumption can significantly elevate the error rate of the system from figure 7. At this stage of segmentation, an image of a previously segmented word, consisting of a sequence of characters, is presented. Consequently, the accuracy of text line segmentation is paramount to the overall effectiveness of a document recognition system. Proper text line segmentation ensures that the system can correctly identify and process each word, thereby enhancing the reliability and performance of the recognition process [17].are isolated, character-level segmentation may be superfluous. In such cases, where there is a consistent spacing between letters within words, even if minimal, segmentation can be effectively achieved at a previous stage by employing a very low threshold. [18],[19]Conversely, for OCR systems applied to text where characters are connected, such as in cursive handwriting, character-level segmentation becomes indispensable. Segmentation is governed by geometric and shape-based criteria to ascertain precise delineation points. Subsequently, word image thinning is employed. Thinning techniques are employed to measure the pixel width of strokes and to identify ligatures in Kannada script. Accurately identify, process, classify, and archive intricate and densely packed characters.

## 4. SEPARABLE NEURAL NETWORK ALGORITHM

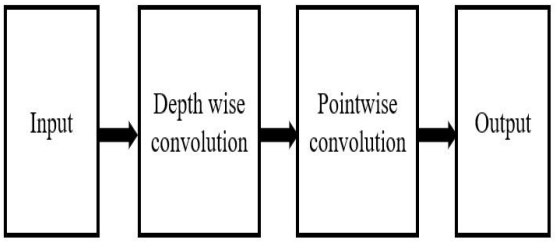


**Fig 8: Separable Neural Network architecture**

In separable Neural Network, we can create a diagram that highlights the key components of the algorithm, particularly focusing on depth wise and pointwise convolutions as shown in figure 8. Separable convolutions reduce the computational complexity by decomposing a standard convolution into two separate processes.

Depth wise Convolution: Applies a single convolutional filter per input channel. Captures spatial features for each channel independently. Does not combine information across channels.

Pointwise Convolution: Applies a 1×11 \times 11×1 convolutional filter across all channels. Combines the information from different channels after the depth wise convolution. Often used to change the number of channels. The initial input to the network, typically an image or feature map. This step involves applying a single convolutional filter to each input channel independently, capturing spatial features without combining channel information. This step applies a 1×11 \times 11×1 convolution across all channels, allowing the network to combine the information from different channels. The final output feature map after the separable convolutions. This algorithm is the core part of the system. It takes the uploaded image of ancient Kannada inscriptions and processes it to convert the text in the image into digital text.

## 5. BLOCK DIAGRAM USING RASPBERRY PI

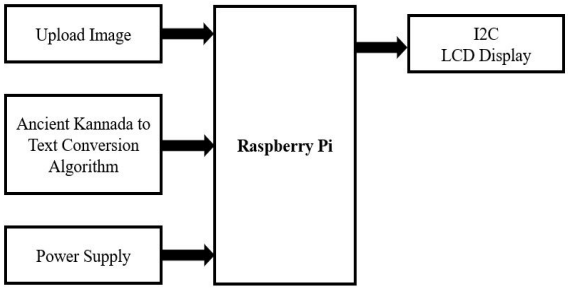


**Fig 9: System Architecture for Kannada Text Conversion using Raspberry Pi**

The system architecture for a project involving the recognition of ancient Kannada epigraphs using a Raspberry

Pi as shown in figure 9. Here is a detailed explanation of each component and its role:

Upload Image: Function: This component represents the mechanism by which images of ancient Kannada inscriptions are uploaded to the system". It could be through various methods like a web interface, a mobile app, or directly through

a USB or camera connected to the Raspberry Pi. It sends the uploaded image to the Raspberry Pi for processing. This involves steps like image preprocessing, character recognition, and text extraction. It is implemented on Raspberry Pi, which performs the necessary computations to convert images to text.

Power Supply: Function: This provides the necessary electrical power to the Raspberry Pi and other connected components. It ensures that the Raspberry Pi operates correctly and continuously. It is connected directly to the Raspberry Pi to power the entire system.

Raspberry Pi: Function: The central processing unit of the system. It manages the image upload, runs the conversion algorithm, and sends output to the display. The Raspberry Pi acts as brain of the system, coordinating all tasks. Receives images from the component and runs the conversion algorithm and receives power from the power supply and sends the converted text to the I2C LCD Display.

I2C LCD Display: Function: This display component is used to show the converted text output. It is connected via I2C (Inter-Integrated Circuit) protocol, which is a communication method allowing multiple devices to connect with the Raspberry Pi. It receives the converted text from the Raspberry Pi and displays it to the user. Image Upload: Users upload images of ancient Kannada inscriptions.

Processing: The Raspberry Pi processes these images using the ancient Kannada to text conversion algorithm.

Power Supply: Ensures continuous operation of the Raspberry Pi and connected components.

Output Display: The converted text is sent to the I2C LCD Display for visual output.

This system provides an automated way to digitize and display ancient Kannada inscriptions, leveraging the processing power of the Raspberry Pi and specialized algorithms for text recognition.

# 6. CONVOLUTIONAL NEURAL NETWORK ALGORITHM

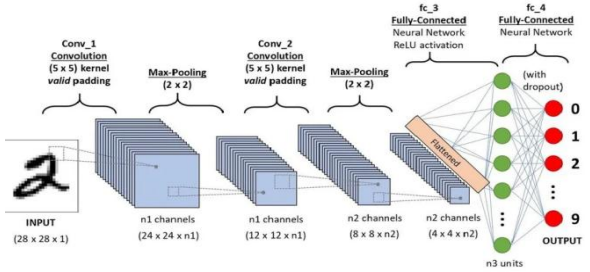


**Fig 10: Convolutional Neural Network Architecture**

Convolutional Neural Network (CNN) architecture as shown in figure 10, we can break down each layer and operation using formulas. Here is a detailed explanation with the corresponding formulas for each layer and operation:

**Input Layer:**

The input is a 28x28 grayscale image (single channel).

First Convolution Layer (Conv_1):

Filter size: 5×55 \ times 55×5

Stride: 1

Padding: Valid (no padding)

Number of filters: n1

The output dimensions after convolution:

1. Output width = (Input width-Filter width/stride) +1--- (1)

2. Output height = (Input height – Filter height/stride) +1 ----- --- (2)

3.Output depth = n1 ------(3)

4. Output dimensions

(28-5+1) X (28-5+1) X n1=24x24xn1 -----(4)

5. First Max-Pooling Layer:

Filter size: 2×22 \ times 22×2

Stride: 2 ------- (5)

6. The output dimensions after pooling:

1. Output width = (Input width /stride)
2. Output height = (Input height /stride)
3. Output depth = Input depth ------ (6)

7. Output dimensions:

(24/2) x (24/2) x n1= 12x12xn1 ------ (7)

8. Second Convolution Layer (Conv_2):

Filter size: 5×55 \ times 55×5

Stride: 1

Padding: Valid (no padding)

Number of filters: n2 -----(8)

9. The output dimensions after convolution:

1.Output width=(Input width-Filter width /stride)+1

2 Output height= (Input height-Filter height/stride)

3.Output depth = n2

10. Output dimensions:

(15+1) x (15+1) x n2=8x8xn2 -------- (9)

10. Second Max-Pooling Layer:

Filter size: 2×22 \ times 22×2

Stride: 2 --------(10)

11. The output dimensions after pooling:

Output width = (Inputwidth-Filter width/stride) +1

Output height = (Input height-Filter Height/stride) +1 -- ---(11)

12. Second Max-Pooling Layer:

Filter size: 2×22 \ times 22×2

Stride: 2 ------ (12)

13. The output dimensions after pooling:

Output width = (Input width/stride)

Output height = (Input height/stride)

Output depth = Input depth

Output dimensions = (8/2) x (8/2) x n2 =4x4xn2 ----- (13)

14. Flattening:

The output of the second max-pooling layer is flattened into a single vector. Flattened dimensions.

=4x4xn2 = 16xn2 -------(14)

15. First Fully-Connected Layer (fc_3):

Number of units: n3

Activation: ReLU

Output dimensions=n3 ------- (15)

This Convolutional neural network algorithm is the core part of the system. It takes the uploaded image of ancient Kannada inscriptions and processes it to convert the text in the image into digital text.

## 7. SYSTEM ARCHITECTURE FOR KANNADA TEXT CONVERSION USING MATLAB AND ARDUINO UNO

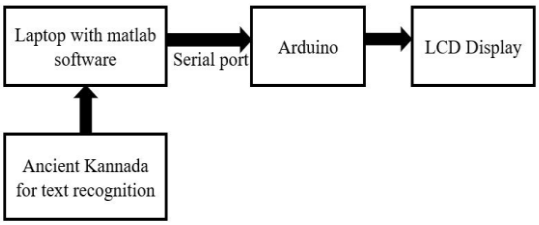


**Fig 11: System Architecture for Kannada Text Conversion using MATLAB and ARDUINO**

Figure 11 illustrates the system architecture for a project involving the recognition of ancient Kannada epigraphs using MATLAB software on a laptop, interfaced with an Arduino Uno to display the converted text on an LCD. Here is a detailed explanation of each component and its role:

Components: Laptop with MATLAB Software:

Function: The laptop is used to run the MATLAB software, which includes the algorithm for converting ancient Kannada inscriptions to text It communicates with the Arduino Uno via a serial port to send the converted text This algorithm, implemented in MATLAB, processes the images of ancient Kannada inscriptions and converts the text in the images into digital text. This involves steps like image preprocessing, character recognition, and text extraction. It runs on the laptop and sends the processed text to the Arduino uno via the MATLAB software. The Arduino uno acts as an intermediary between the laptop and the LCD display. It receives the converted text from the laptop via the serial port and sends it to the LCD display for visualization. Receives data from the laptop via the serial port and sends data to the LCD display.

The LCD display is used to show the converted text. It is controlled by the Connection: It receives the converted text from the Arduino uno and displays it to the user.

Image Processing: The user processes the images of ancient Kannada inscriptions using the MATLAB software on the laptop. The software runs the ancient Kannada to text conversion algorithm to convert the inscriptions into digital text.

Data Transmission: The converted text is sent from the laptop to the Arduino uno via a serial port connection.

Display Output: The Arduino Uno receives the text data and sends it to the LCD display, which then shows the converted text.

This system leverages MATLAB for powerful image processing and text recognition capabilities, using Arduino uno as a bridge to display the processed text on an LCD. This setup allows for effective digitization and display of ancient Kannada inscriptions.

## 8. RESULTS & DISCUSSION

The Figure 12 illustrates a fragment of an archaic stone inscription, meticulously etched with glyphs that exhibit the intricate curves and forms emblematic of ancient Kannada script. The inscriptions, rendered in these timeworn characters, suggest the use of a classical variant of the Kannada language, discernible through the distinctive and elaborate morphology characteristic of this scriptural tradition.


**Fig 12: Original image**

Grayscale images as shown in the figure 13 are frequently employed in image processing due to their ability to streamline computational algorithms by diminishing the data complexity inherent in full-color images.

Operations are more efficaciously implemented on grayscale images, as these procedures benefit from the reduction in dimensionality and the focus on luminance variation rather than chromatic information.

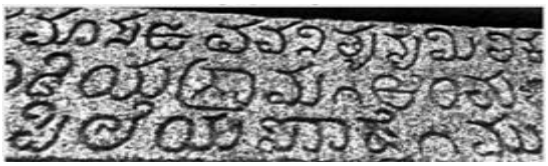

**Fig 13: Gray image**

Edge detection algorithm is widely recognized as one of the most popular, effective, and sophisticated techniques for edge detection. It employs a multi-stage process to identify and delineate various types of edges within an image as shown in figure 14.

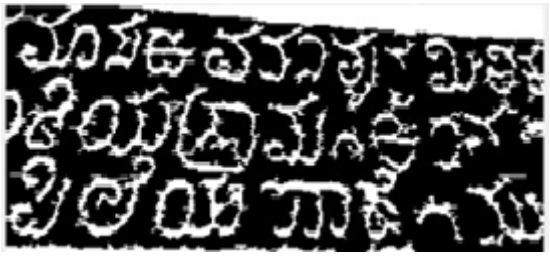

**Fig 14: Edge detection image**

Character segmentation, therefore, constitutes a critical and foundational element within the expansive domain of text recognition, establishing the indispensable groundwork for the precision and reliability of Optical Character Recognition systems. This process is integral to ensuring that subsequent stages of text analysis and interpretation are conducted with a high degree of accuracy, thereby enhancing the overall efficacy of Optical Character Recognition technologies as shown in the figure 15.

**Fig 15: Character segmentation image**

Finally, the GUI exemplifies an intuitive interface designed to facilitate seamless interaction with a Kannada text recognition system as shown in figure 16. It effectively supports the processes of model training, data ingestion, and textual recognition, rendering it an indispensable tool for scholars and practitioners engaged in computational linguistics, machine learning, or the digital preservation of Kannada script. The system's capability to streamline these complex tasks underscores its utility in advancing research and application in the domain of Kannada language processing

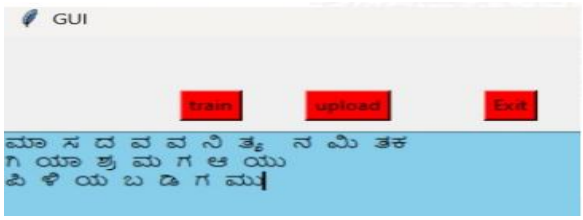


**Fig 16: GUI Output**

The Raspberry Pi, when coupled with LCD display, facilitates autonomous operation of the system, rendering it particularly suitable for field applications or scenarios where portability and user convenience are paramount. This integration enables users to execute tasks such as image upload, model training, and text recognition directly via LCD interface, thereby streamlining the workflow and enhancing operational efficiency as shown in figure 17.

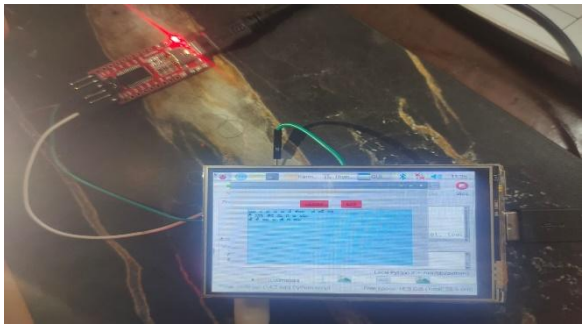

**Fig 17: Hardware Output**

# 9. CONCLUSION

In this study, we have demonstrated the efficacy of Convolutional Neural Networks in the recognition and decipherment of ancient Kannada texts. The results indicate that CNNs, with their inherent ability to capture spatial hierarchies in images, are remarkably adept at identifying and interpreting complex script patterns characteristic of ancient manuscripts. By leveraging deep learning techniques, we have achieved significant improvements in accuracy and processing efficiency compared to traditional methods.

This advancement not only enhances the digitization and preservation efforts of invaluable historical texts but also paves the way for future research into more nuanced and contextually aware recognition systems. The integration of CNNs into ancient text recognition frameworks holds immense potential for the broader field of computational linguistics, offering a robust tool for unraveling the linguistic heritage of civilizations. Future work will focus on refining these models further, addressing challenges such as noise reduction, script variability, and extending the methodology to other ancient scripts.

# 10. REFERENCES

[1] Anusha Leela Somashekharaiah and Abhay Deshpande. 2024. Preprocessing techniques for recognition of ancient Kannada epigraphs. *International Journal of Electrical and Computer Engineering (IJECE)*, Vol 14, No 1, February 2024.

[2]Aİ.Topçuand and UğurTöreyin, Neural Machine Translation Approaches for Post OCR Text Processing, *30th Signal Processing and Communications Applications Conference (SIU),Safranbolu, Turkey*, 2022.

[3]A.İ.TopçuandU and UğurTöreyin, Neural Machine Translation Approaches for Post OCR Text Processing, *30th Signal Processing and Communications Applications Conference (SIU), Safranbolu, Turkey, 2022*

[4].A. Kashyap and A. Kumara B, OCR of Kannada Characters using Deep Learning. *Trends in Electrical ,Electronics, Computer Engineering Conference (TEECCON),* Bengaluru, India, 2022, pp. 35-38,doi:10.1109/TEECCON54414.2022.9854842.

[5].A. Spruck, M. Hawesch, A. Maier, C. Riess, J. Seiler and A, 3D Rendering Frame work for Data Augmentation in Optical Character Recognition, *2021 International Symposiumon Signals, Circuits and Systems (ISSCS),* Iasi, Romania, 2021, pp. 1-4,doi:10.1109/ISSCS52333.2021.9497438.

[6]. A. Spruck, M. Hawesch, A. Maier, C. Riess, J. Seiler and A. Kau, 3D Rendering Frame work for Data Augmentation in Optical Character Recognition, *2021 International Symposiumon Signals, Circuits and Systems (ISSCS),* Iasi, Romania, 2021, pp. 1-4,doi:10.1109/ISSCS52333.2021.9497438.

[7].Chandrakala,H and Thippeswamy,G, Deep Convolutional Neural Networks for Recognition of Historical Handwritten Kannada Characters. 10.1007/978-981-13-9920-6_7, 2020.

[8].Bannigidad, Parashuram, Gudada and Chandrashekar, Historical Kannada Handwritten Character Recognition using K-Nearest Neighbour Technique, 2019.

[9] Panchal, Amoli, Chintan, Shah and Bhargav, Image Binarization Techniques for Degraded Document Images: A Review Binarization techniques, 2017.

[10]. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, and C. Fu, et al. 2017. SSD: Single Shot Multi Box Detector. *Proceedings of Computer Vision and Pattern Recognition.*

[11] A. Soumya and G. H. Kumar, Recognition of Ancient Kannada Epigraphs using Fuzzy-based Approach. *International Conference on Contemporary Computing and Informatics*. Mysore, India, 2014, pp. 657-662, doi: 10.1109/IC3I.2014.7019645.

[12] K. R. Shah and D. Dattatray Badgujar, Devnagari Handwritten Character Recognition (DHCR) for Ancient Documents. *IEEE Conference on Information & Communication Technologies.* Thuckalay, India, 2013, pp. 656-660, doi: 10.1109/CICT.2013.6558176.

[13] C. Naveena and V. N. Manjunath Aradhya, Handwritten Character Segmentation for Kannada Scripts. *World Congression Information and Communication Technologies*. Trivandrum, India, 2012, pp.144149, doi:10.1109/WICT.2012.6409065.

[14] T. Sari, A. Kefali and H. Bahi, An MLP for Binarizing Images of Old Manuscripts. *International Conference on Frontiers in Handwriting Recognition*. Bari, Italy, 2012, pp. 247-251, doi: 10.1109/ICFHR.2012.176.

[15] Indira, K. & Selvi, S. Sethu, Kannada Character Recognition System,2010 - A Review.

[16] M. I. Shah and C. Y. Suen, Word Spotting in Gray Scale Handwritten Pashto Documents. *12th International Conference on Frontiersin Handwriting Recognition*. Kolkata, India,2010, pp.136-141, doi:10.1109/ICFHR.2010.28.

[17] Fischer A, Wthrich M Liwicki M, Frinken V, Bunke H, Viehhauser G and Stolz M, Automatic Transcription of Handwritten Medieval Documents. *Proceedings of Virtual Systems and Multimedia,* 137{142 (2009).

[18] Sauvola J. J. and Pietikainen M. 2000. Adaptive Document Image Binarization. *Pattern Recognition 33(2),225{236 (2000).*

[19] T. Wakahara and Y. Kimura, Affine-Invariant Gray-Scale Character Recognition using GAT Correlation. *Proceedings 15th International Conference on Pattern Recognition.* ICPR-2000, Barcelona, Spain, pp. 417-421 vol.4, doi: 10.1109/ICPR.2000.902947.